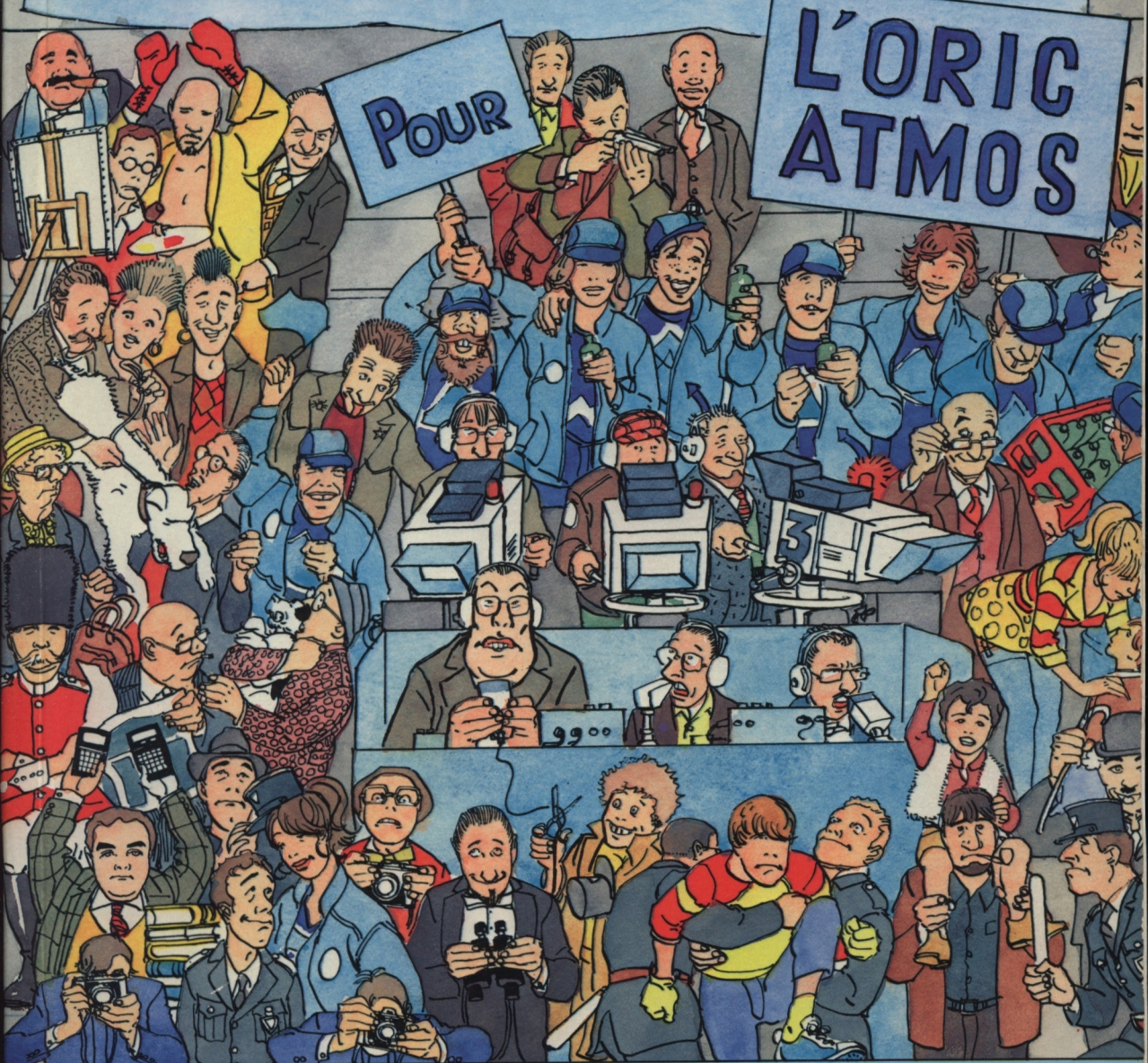
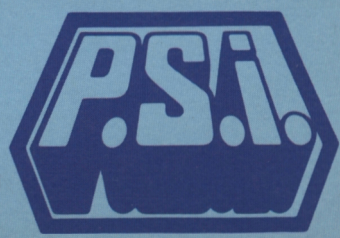




102 PROGRAMMES



PAR
JACQUES
DECONCHAT



**102 PROGRAMMES POUR
ORIC-ATMOS**

Autres ouvrages relatifs à l'Oric

- La découverte de l'Oric — Daniel-Jean David
- Oric pour tous — Jacques Boisgontier et Sophie Brébion
- 52 programmes Oric pour tous — Jacques Boisgontier
- Clefs pour l'Oric — Emmanuel Flesselles
- Le cahier du Basic sur Oric — Jacques Boisgontier et Sophie Brébion
- L'Oric à l'affiche — Jean-François Séhan
- L'assembleur de l'Oric 1 et Oric-Atmos — Marcel Henrot

Ouvrages d'initiation

- Visa pour l'informatique — Jean-Michel Jégo
- Visa pour le Basic — Jean-Michel Jégo
- Mon ordinateur — Jean-Claude Barbance
- Le logotron informatique — Jean-Pierre Petit
- L'ordinateur apprivoisé — François Picard et Danielle Shaw
- Minitel mode d'emploi — Dominique Schraen et Sibylle Rochas
- Macintosh le magnifique — Merl K. Miller et Mary A. Myers, traduit par Raymond Borraz

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

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

**JACQUES
DECONCHAT**

**102 PROGRAMMES POUR
ORIC-ATMOS**



EDITIONS DU P.S.I.

1984

SOMMAIRE

Avant-propos	8
Organisation de l'ouvrage	9

Niveau 1

Instructions élémentaires	12
Le nombre mystérieux	18
Cube	20
Chaud, froid	22
Les allumettes	24
Allumettes à plusieurs	26
Calendrier perpétuel	28
Bataille navale	30
Golf	32
Solfège	34
Le jeu du 21	36
Pair - Impair	38
Devine un chiffre	40
Le jeu de Wytthof	42
Rebonds	44
Jackpot	46
Roulette	48
Couloirs	50
Echiquier	52
Le jeu de la fève	54
Jeu de l'oie aléatoire	56

Niveau 2

Dimensionnement d'un tableau - Recherche d'un élément dans un tableau et dans une chaîne - L'instruction DATA	60
Dé truqué	62
Visions	64
421	66
L'électricien fou	68
La course au 20	70
Musique	72
Poker	74
Le jeu de saute-mouton	76

Lettres	78
Machinamot	80
Espion	82
Accrochez les wagons	84
Tirage d'un dé truqué	86
Tour de cartes	88
Le jeu de Marienbad	90
L'indésirable	92
Compose	94
Jour de la semaine	96
Le jeu de la vie	98

Niveau 3

Manipulation des affichages grand format –	
Mode haute résolution	102
Un dé	104
Plus grand, plus petit	106
Ordre	108
Le pendu	110
Belote	112
Un taquin de calcul	114
Pendu inversé	116
Nicomaque	118
Course de chevaux	120
A travers	122
Mastermind	124
Idem	126
Horloge	128
Reines sur l'échiquier	130
Répétitions	132
Mots	134
Permutations	136
Bridge	138
Hanoï	140
Losanges	142
Mathieu	144
Le loup et les agneaux	146

Niveau 4

Les techniques d'animation élémentaire –	
Les caractères graphiques	150
Tirage de dés	152
Jeu de loto	154
La bombe	156
Ardoise magique	158

Affichage publicitaire	160
Evasion	162
Chapeaux	164
Combat aérien	166
Dessin de dés	168
Le jeu du missile	170
Le singe	172
Le jeu du 15	174
Adresse	176
Sous-marin	178
Zombies	180
Diago	182
Décomaque	184
Chasse au sous-marin	186
Clavier	188
Le lézard d'Alice	190
Calculatrice	192

Niveau 5

Les techniques évoluées de programmation	196
Smokey	198
Course de voitures	200
Echecs	202
Danger spatial	204
Tic Tac Toe	206
Chenille	208
Dames	210
Quatre à quatre	213
Anagrammes	216
Œufs	218
Taquin de lettres	220
Cibles	222
Gouttes	224
Obstacles	226
Captures dans l'espace	228
Anomalie	230
Plusieurs en un	232
Parcours du cavalier	234
Mur de briques	236
Othello	238

Avant-propos

Vous trouverez dans ce livre plus de 100 programmes (essentiellement des jeux), rédigés en Basic, pour l'ordinateur individuel ORIC-ATMOS. La plupart de ces programmes seront d'ailleurs utilisables sans grande modification sur l'ORIC 1.

Les programmes proposés sont très courts, et l'ordre dans lequel ils sont présentés n'est bien évidemment pas arbitraire : il s'agit d'initier le lecteur, à travers la réalisation et l'utilisation de programmes de jeux de bonne qualité, à la connaissance et à la maîtrise progressive de cet outil extraordinaire qu'est un ordinateur individuel.

La brièveté des programmes proposés permettra de les taper facilement, sans risque d'erreur. Vous pourrez bien sûr les utiliser comme base pour la réalisation de jeux plus performants.

Les programmes sont abondamment commentés pour vous aider à comprendre leur fonctionnement, et les pages de présentation vous permettront de retrouver facilement les caractéristiques propres des instructions Basic qu'ils utilisent.

Les programmes n'utilisent pas toutes les particularités d'un appareil comme l'ORIC, par ailleurs très bien détaillées dans le manuel de base.

Vous trouverez, à la fin de chaque texte de présentation, quelques suggestions pour des améliorations possibles, ou des modifications, qui pourraient rendre le jeu proposé plus attrayant, plus performant ou plus complet.

Organisation de l'ouvrage

Les jeux sont classés par ordre de difficulté croissante, suivant 5 niveaux :

- **Le niveau 1** permet l'étude des instructions élémentaires du langage BASIC : **LET** (optionnel ici), **PRINT**, **INPUT**, **IF...THEN**, **FOR...NEXT**, **GOTO**, **GOSUB**, **ON...GOTO**, **ON...GOSUB**.
- **Au niveau 2**, on utilisera les instructions de confection et de manipulation des éléments d'un tableau, et de recherche d'un élément dans une chaîne : **DIM**, **R(I)**, **MID\$**, **READ**, **DATA**.
- **Au niveau 3**, on utilise la panoplie complète des instructions de traitement des chaînes de caractères, et les instructions permettant l'affichage grand format et le graphique haute résolution : **MID\$**, **RIGHT\$**, **LEFT\$**, **HIRES**, **CURSET**.
- **Le niveau 4** permet de faire connaissance avec le deuxième jeu de caractères (caractères semi-graphiques de type TELETEL), et introduit une instruction particulièrement utile dans les jeux interactifs : **KEY\$**.
- Enfin, **le niveau 5** nous fait découvrir des instructions plus spécifiques : définition des caractères graphiques, identification d'objets sur l'écran, mise en mémoire directe et rappel : **SCRN (x, y)**, **PEEK**, **POKE...**

Chaque jeu est présenté d'une façon très systématique, par une courte description, suivie de l'étude détaillée de chacune des lignes importantes du programme. Après quoi les indications nécessaires au bon fonctionnement du jeu sont données, ainsi que quelques éléments pouvant conduire à des extensions ou des améliorations du jeu proposé.

Le programme, fonctionnant sur l'ORIC-ATMOS, est ensuite reproduit, ainsi qu'un exemple d'exécution. Les listings ont été obtenus directement d'après les originaux de l'auteur, sur une imprimante matricielle, ce qui devrait garantir contre tout risque de dysfonctionnement. L'emploi des minuscules a été systématiquement évité (on travaille en position **CAPS**).

Niveau 1

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Le nombre mystérieux	Jeu de déduction, pour un joueur. Plusieurs niveaux.
Cube	Jeu de dessin : perspective sur un écran.
Chaud, froid	Logique et hasard, pour un joueur. Plusieurs niveaux.
Les allumettes	Jeu de réflexion, contre l'ordinateur.
Allumettes à plusieurs	Jeu de réflexion et de stratégie, pour plusieurs joueurs.
Calendrier perpétuel	Confection d'un calendrier, mois par mois.
Bataille navale	Un seul navire à trouver, mais quelques particularités.
Golf	Un parcours de 18 trous, pour un seul joueur.
Solfège	Apprenez le solfège à l'aide de votre ordinateur.
Le jeu du 21	Jeu de hasard, l'ordinateur tient la banque.
Pair - Impair	Un jeu de casino en trois temps (un ou plusieurs joueurs).
Devine un chiffre	Un jeu très facile, pour un ou plusieurs joueurs.
Le jeu de Wytthof	Un jeu de réflexion assez simple, pour deux joueurs.
Rebonds	Un jeu d'adresse très visuel, pour un joueur.
Jackpot	Jeu de hasard, pour un joueur.
Roulette	Une simulation élégante du jeu très connu.
Couloirs	Un jeu de logique, à découvrir.
Echiquier	Une base graphique pour vos propres créations.
Le jeu de la fève	Un jeu de réflexion, pour deux joueurs.
Jeu de l'oie aléatoire	Un très vieux jeu, revu et corrigé par l'ordinateur, deux joueurs.

Instructions élémentaires

Nous ne retiendrons ici que les instructions fondamentales, ou instructions de programmation ; beaucoup d'autres commandes, ou fonctions particulières, seront utilisées à l'occasion, par exemple :

- *RND (1) pour obtenir un nombre aléatoire entre 0 et 1.*
 - *INT pour prendre la partie entière d'un résultat.*
 - *REM pour placer un commentaire.*
 - *END pour terminer un programme.*
- etc.

Les instructions fondamentales de la programmation, utilisées dès le niveau 1, et présentes dans pratiquement tous les programmes sont :

LET (optionnelle) ; **PRINT** ; **INPUT** ; **GOTO** ; **IF** **THEN** ;
FOR ... NEXT ; **GOSUB ... RETURN**.

soit seulement sept instructions de base (et deux instructions dérivées) que nous allons maintenant examiner plus en détail.

L'INSTRUCTION LET, OU INSTRUCTION D'AFFECTATION

C'est l'instruction par excellence des langages de programmation évoluée, puisque son rôle essentiel est de permettre, pour le programmeur, une gestion transparente de la mémoire. Cela signifie, en pratique, que pour ranger une quantité en mémoire, par exemple 12, il n'est pas nécessaire de savoir dans quelle case de la mémoire on va la déposer : il suffit de lui affecter un nom autorisé, et c'est par ce nom que la quantité sera par la suite retrouvée et manipulée.

- **Si la quantité est numérique**, le nom sera une simple lettre de l'alphabet, ou un ensemble de plusieurs chiffres ou lettres, commençant obligatoirement par une lettre.
- **Si la quantité est alphanumérique**, la même règle s'applique pour le nom, qui devra cependant être suivi du symbole \$, et la variable devra être écrite entre guillemets.

Exemple : LET A1 = 12 ; la variable numérique A1 prend la valeur 12
LET NOM \$ = "RENOIR" ; la variable alphanumérique
NOM\$ prend la valeur RENOIR.

Sur l'ORIC, comme dans la plupart des Basic, le LET est devenu optionnel, aucune ambiguïté n'étant possible, on écrira donc : A = 12 ; NOM\$ = "TOTO" ; etc.

L'INSTRUCTION PRINT, OU INSTRUCTION DE SORTIE (ECRAN)

C'est une instruction de dialogue, qui permet à la machine d'afficher sur l'écran un texte, le résultat d'un calcul, ou une variable. C'est le seul moyen que possède l'ordinateur individuel pour faire connaître ses résultats. Ainsi, PRINT A1 provoque l'affichage de 12 (si l'on a fait auparavant A1 = 12), PRINT A\$ provoque l'affichage de BONJOUR (si l'on a fait A\$ = "BONJOUR"). Utilisé après un effacement d'écran, le PRINT provoque l'affichage sur la 1^{re} ligne, en haut, à gauche.

Un certain nombre de paramètres peuvent être utilisés pour obtenir sur l'écran un affichage convenable :

- **PRINT TAB(N)** provoque sur la ligne d'impression courante un affichage commençant à la *énième* colonne à partir de la gauche ($0 \leq N \leq 39$).
- Un ; (**point virgule**) en fin de PRINT permet soit d'enchaîner une autre impression sans répéter l'instruction PRINT, soit d'indiquer que la prochaine impression sera, en l'absence d'indication contraire, effectuée exactement à la suite de l'impression précédente.
- La , (**virgule**) provoque une tabulation horizontale automatique pré-définie.

Lors de la frappe, l'**instruction PRINT** pourra être abrégée en tapant ? (**point d'interrogation**). Lors d'une impression, tout nombre est précédé d'un signe ; mais le signe positif n'est pas affiché, un espace précèdera donc automatiquement l'impression de tout résultat positif.

Sur l'ORIC-ATMOS, on peut définir la ligne et la colonne d'écriture avec l'instruction **PRINT a X, Y** ; l'écran utilise 28 lignes de 40 caractères, la ligne du haut (inscription **CAPS**) n'étant pas accessible, on ne pourra utiliser que les lignes 0 à 26.

Par exemple : PRINT A 12, 21 ; "BONJOUR" affiche le mot BONJOUR à la 12^e colonne de la 21^e ligne. Les 2 premières colonnes sont en principe réservées aux instructions de papier et d'encre ; l'affichage commencera en général à la deuxième colonne.

Une autre commande permet d'obtenir un affichage identique, malgré un fonctionnement légèrement différent : il s'agit de **PLOT X, Y, A\$**. Ainsi PLOT 12, 21, "BONJOUR" affiche le mot BONJOUR à la 12^e ligne de la 21^e colonne.

Avertissement important : ce livre, écrit pour l'ORIC-ATMOS, fait fréquemment usage de l'instruction **PRINT A** ; cette instruction n'est pas présente sur l'ORIC 1 ; elle pourra facilement, dans certains cas, être remplacée par **PLOT**, comme indiqué précédemment. Sinon, il conviendra de définir la position du curseur, avant d'afficher.

Ainsi **PRINT A, X, Y** ; "HELLO" sera remplacé sur ORIC 1 par : **POKE # 269, X : POKE # 268, Y : PRINT "HELLO"**.

En dehors de ce cas, l'adaptation des programmes sur ORIC 1 ne devrait pas poser de problème.

L'INSTRUCTION INPUT, OU INSTRUCTION D'ENTREE (CLAVIER)

Cette instruction, qui provoque un arrêt et une attente de l'appareil, *permettra d'entrer un nombre, ou un texte, qui sera affecté à la variable suivant l'ordre INPUT*. Ceci est fait en mode conversationnel, c'est-à-dire au fur et à mesure des besoins ; en effet, lorsque l'ORIC rencontre une instruction **INPUT**, il s'arrête et vous rend la main ; on peut alors lui entrer une information de nature convenable (nombre ou texte), et lui signaler que cette entrée est terminée à l'aide de la touche **RETURN**, ce qui lui rendra la main.

Si l'information entrée n'est pas de nature convenable (vous tapez, par exemple, un texte sur un **INPUT X**, qui attend un nombre), le message **REDO FROM START** est affiché, et l'appareil se remet en attente.

Il est possible, comme pour le **PRINT**, d'afficher des messages avant l'entrée de la variable.

On pourra, par exemple, écrire :

```
INPUT "VOTRE NOM, S'IL VOUS PLAIT" ; A$
```

L'INSTRUCTION GOTO OU INSTRUCTION DE RENVOI INCONDITIONNEL

Une des particularités du langage Basic, qui le rend facilement accessible, est que *chaque ligne de programme doit être précédée d'un numéro*, ces lignes étant ensuite exécutées par l'ordinateur individuel en séquences, dans le sens des numéros croissants. Pour diverses raisons, il pourra être nécessaire au cours du déroulement d'un programme, de ne pas suivre cet ordre séquentiel très rigide ; on utilisera alors *l'instruction GOTO, qui permet de détourner le fonctionnement du programme de son cours normal* quitte d'ailleurs à le reprendre ensuite par un autre **GOTO**. Le renvoi doit, par ailleurs, impérativement se faire sur un numéro de ligne existant dans le programme (**GOTO 20**, par exemple).

Pour donner un peu de souplesse à cette instruction de renvoi, il existe une instruction très puissante, qui permet de renvoyer à n'importe quel numéro de ligne, selon la valeur d'une variable numérique : *c'est l'instruction **ON** **GOTO*** ; on écrira par exemple :

```
ON X GOTO 100, 200, 300 et le renvoi sera fait à la ligne 100 si X vaut 1, à la ligne 200 si X vaut 2, et 300 si X vaut 3.
```


L'INSTRUCTION IF... THEN... OU INSTRUCTION DE TEST

On peut l'utiliser de deux façons : d'abord, en rupture de séquence conditionnelle, sous la forme :

IF condition = vraie **THEN** numéro de ligne.

C'est l'usage le plus courant de ce test, qui permet de renvoyer à une autre partie de programme, si par exemple $X = 0$, ou si $A = B$, etc. Les **opérateurs logiques NOT, OR** ou **AND** peuvent d'ailleurs être utilisés dans l'écriture de la condition.

On pourra aussi utiliser le test sous la forme :

IF condition **THEN** instruction ;

l'instruction présente à la suite de **THEN** peut être une affectation $A = 12$, une impression **PRINT A**, une entrée **INPUT B \$**, ou même un autre test **IF ... THEN ...**.

On peut par exemple écrire :

IF A > 3 THEN IF B < 7 THEN PRINT A + B

L'INSTRUCTION FOR...NEXT, OU INSTRUCTION DE BOUCLES

On est très souvent conduit, en programmation, à vouloir exécuter un nombre entier de fois une certaine suite d'instructions (par exemple, faire avancer un mobile de la case 6 à la case 20, etc.). Plutôt que d'utiliser un test et un renvoi, il sera préférable de faire appel à une **boucle**, la variable de boucle prenant ses valeurs dans les entiers ou les décimaux, et le **pas** de cette variable (c'est-à-dire la quantité dont elle augmente ou diminue à chaque passage) pouvant aussi prendre ses valeurs dans les entiers ou les décimaux.

Ainsi, si l'on écrit :

```
FOR I = 0 TO 20 STEP 2  
  Instructions  
NEXT I
```

Les instructions contenues dans la boucle seront exécutées pour $I = 0$, pour $I = 2$, ... et ceci jusqu'à $I = 20$, soit 11 fois. La variable I peut être utilisée pour faire un calcul dans la boucle, mais ne devra pas être modifiée dans la boucle. On peut, bien sûr, sortir de la boucle, avant $I = 20$, par un test portant une quantité quelconque.

Si **le pas de la boucle** (STEP) n'est pas précisé, il sera automatiquement pris égal à 1.

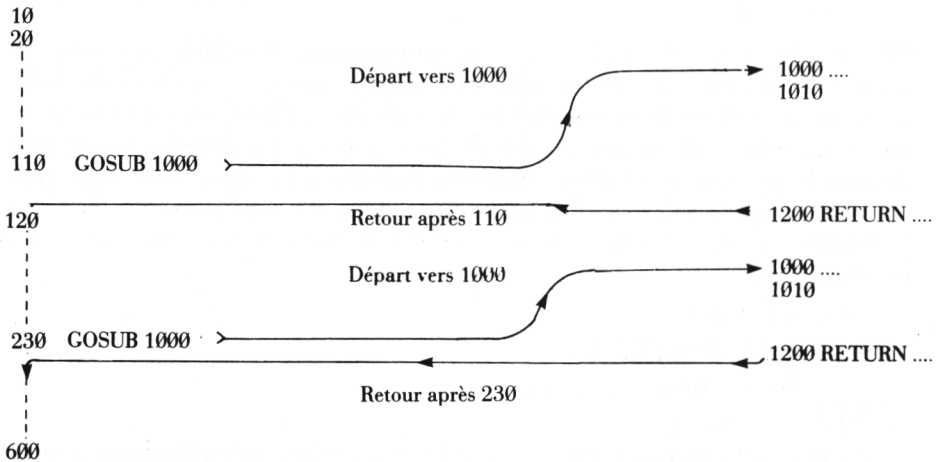
Il est possible d'emboîter plusieurs boucles de noms différents, et d'omettre l'indication de la variable dans le **NEXT**.

L'INSTRUCTION GOSUB, OU INSTRUCTION D'APPEL, DE SOUS-PROGRAMME

Lorsqu'une même suite d'instructions doit être exécutée plusieurs fois, et appelée à partir de plusieurs endroits du programme, on évitera de la réécrire plusieurs fois. On la rédige une fois pour toutes, et on la place en dehors du programme proprement dit (en lui donnant, par exemple, un numéro de ligne supérieur au numéro de la dernière ligne du programme). Indépendamment des difficultés que l'on pourra rencontrer au niveau du nom des variables, lorsque l'on sera désireux d'utiliser cette séquence, il suffira de l'appeler par l'instruction **GOSUB** numéro de ligne. Le programme ira dès lors "faire un détour" dans la partie du programme appelé, mais il ne pourra revenir et poursuivre son déroulement normal que si il rencontre dans le programme appelé une instruction **RETURN**, ou instruction de retour.

Voici le schéma théorique de fonctionnement :

PROGRAMME PRINCIPAL



FIN DU PROGRAMME PRINCIPAL

1000
1200 RETURN

Sous-Programme 1.

Le Sous-Programme n'est écrit qu'une fois, mais appelé aussi souvent que l'on veut.

Un sous-programme pourra lui-même éventuellement appeler un autre sous-programme, et ainsi de suite.

On peut aussi appeler un sous-programme de façon conditionnelle, en utilisant l'instruction `ON \square GOSUB \square`

Par exemple `ON X GOSUB 1000, 2000, 3000` permettra d'exécuter le sous-programme placé en 1000 si $X = 1$, en 2000 si $X = 2$, en 3000 si $X = 3$.

Le nombre mystérieux

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu de déduction pour un joueur.

Difficulté du jeu : facile.

LE JEU :

On demande à l'ordinateur de fabriquer un nombre de n chiffres (avec n compris entre 1 et 7).

On devra ensuite essayer de le découvrir, avec le moins d'essais possible, en recevant les seuls renseignements TROP GRAND ou TROP PETIT.

LE PROGRAMME :

La ligne 50 permet de fabriquer le nombre à deviner.

Les entrées sont faites par les lignes 70 à 80.

La ligne 90 teste si le résultat est exact.

La ligne 100 signale un essai trop petit, et la ligne 110 un essai trop grand.

Le compteur d'essais est en ligne 120.

Les lignes 200 à 230 s'occupent de l'affichage du résultat, et de la gestion de la partie suivante.

POUR JOUER :

On lancera le programme par RUN.

On obtient un premier affichage NOMBRE DE CHIFFRES auquel on répond par un entier entre 1 et 7, suivi de RETURN.

On obtient alors l'affichage ESSAI N° 1, suivi de la question VOTRE NOMBRE.

On entre un nombre, et l'on presse RETURN.

On obtient alors une réponse TROP PETIT, ou TROP GRAND, ou encore, si l'on a trouvé juste, un affichage de VOUS AVEZ TROUVEZ... EN... ESSAIS. Si le résultat n'est pas exact, on revient à ESSAI N° 2, et VOTRE NOMBRE.

En fin de partie, on obtient la question AUTRE PARTIE (O/N) ; on répond O pour faire une autre partie (suivi de RETURN).

On peut arrêter à tout moment en tapant CTRL et C.

EXTENSIONS POSSIBLES :

On pourra imaginer de faire deviner au joueur autre chose qu'un chiffre (par exemple, un mot, en utilisant l'ordre alphabétique, ou encore autre chose...).

```

10 REM LE NOMBRE MYSTERIEUX
20 CLS
30 INPUT"NOMBRE DE CHIFFRES :";N
40 IF N>7 THEN RUN
50 X=INT(RND(1)*10^N+1)
60 J=1
70 PRINT:PRINT"ESSAI NO. ";J
80 INPUT"VOTRE NOMBRE :";M
90 IF M=X THEN 200
100 IF M<X THEN PRINT"TROP PETIT":PING
110 IF M>X THEN PRINT"TROP GRAND":PING
120 J=J+1
130 PRINT:PRINT
140 GOTO 70
200 CLS:EXPLODE
210 PRINT"VOUS AVEZ TROUVE ";X;" EN ";J;" ESSAIS"
220 INPUT"AUTRE PARTIE (O/N)";R$
230 IF R$="O" THEN RUN

```

NOMBRE DE CHIFFRES : ? 3

ESSAI NO. 1
VOTRE NOMBRE : ? 500
TROP PETIT

ESSAI NO. 2
VOTRE NOMBRE : ? 800
TROP PETIT

ESSAI NO. 3
VOTRE NOMBRE : ? 900
TROP GRAND

ESSAI NO. 4
VOTRE NOMBRE : ?

Cube

REFERENCES :

Difficulté de programmation : aucune.

Catégorie de jeu : dessin.

Difficulté du jeu : aucune.

LE JEU :

Le programme proposé se contente de réaliser à l'aide d'instructions PRINT une vue d'un cube en perspective cavalière.

Il pourrait être par exemple incorporé à un jeu basé sur les capacités de reconnaissance des formes géométriques élémentaires.

LE PROGRAMME :

Le dessin du cube est réalisé par l'impression de points, en utilisant deux boucles : la boucle I (30-120), et la boucle I (130-180).

Le nom de l'objet représenté est affiché en ligne 190 (ici : CUBE).

POUR JOUER :

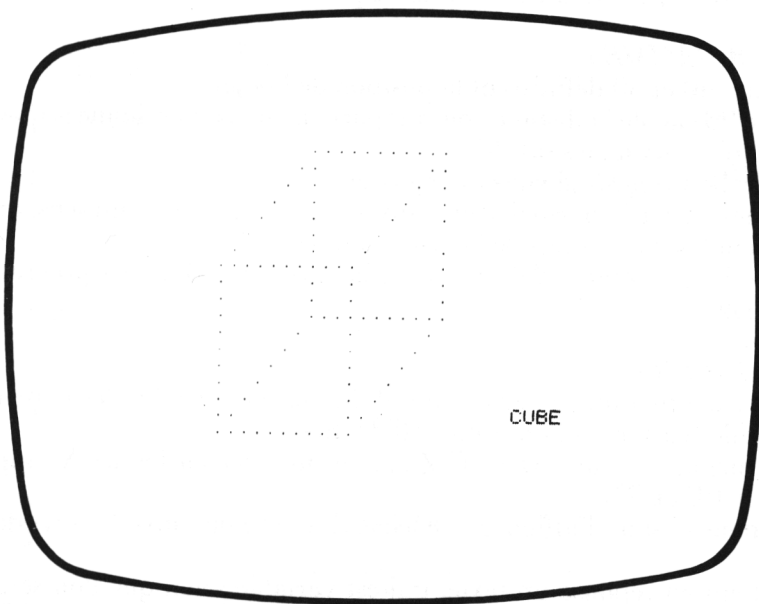
Il suffit de lancer l'exécution par RUN, et de regarder.

EXTENSIONS POSSIBLES :

La haute résolution permettra d'avoir un tracé plus précis.

On pourra imaginer un jeu utilisant des parallélépipèdes rectangles de différentes tailles, avec affichage par exemple des 3 côtés, et il faudrait trouver leur volume, etc.

```
10 REM CUBE
20 CLS:PAPER 2
30 FOR I=2 TO 11
40 PRINT@7+I,2;". "
50 PRINT@1,9;". "
60 PRINT@7+I,12;". "
70 PRINT@1,19;". "
80 PRINT@2,7+I;". "
90 PRINT@12,7+I;". "
100 PRINT@9,I;". "
110 PRINT@19,I;". "
120 NEXT I
130 FOR I=2 TO 9
140 PRINT@I,11-I;". "
150 PRINT@10+I,11-I;". "
160 PRINT@I,21-I;". "
170 PRINT@10+I,21-I;". "
180 NEXT I
190 PRINT@25,18;"CUBE":PING
```



Chaud, froid

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

Un des plus vieux jeux du monde, toujours aussi pratiqué, mais cette fois, c'est l'ordinateur qui vous sert de partenaire : il va en effet cacher un objet à un certain emplacement d'une grille dont vous aurez vous-même choisi les dimensions. Pour le trouver, il ne vous donnera que quelques indications sous la forme **TU BRULES**, ou **C'EST FROID**, etc.

Vous devez réussir à découvrir l'objet caché, avec le moins d'essais possible (votre score sera affiché en fin de partie).

LE PROGRAMME :

Les lignes 40 et 50 définissent la position de l'objet.

La ligne 90 cacule la distance par rapport aux deux coordonnées que vous avez entrées aux lignes 60-70.

La ligne 100 regarde si vous avez gagné.

Les lignes 120-130 normalisent la distance (qui doit avoir un sens, quelles que soient les dimensions de la grille initiale).

La ligne 140 effectue le départ, s'il y a lieu, vers l'un des sous-programmes d'affichage.

POUR JOUER :

On lance le programme par l'ordre **RUN**, puis on introduit les dimensions de la grille (carrée) et l'on frappe **RETURN**.

Dès l'affichage suivant (**DONNEZ X**), on entre la coordonnée **X** supposée, suivie de **RETURN**.

On obtient ensuite l'affichage **DONNEZ Y**, et l'on entre **Y** suivi de **RETURN**.

On obtient en réponse un texte, qui est variable selon que l'on sera plus ou moins loin de l'objet cherché (cela va de **COMPLETEMENT GELE** à **BRULANT**). Lorsque l'on a trouvé, on obtient l'affichage du nombre d'essais effectués.

Pour sortir du programme avant la fin de la partie, il suffit de taper **CTRL** et **C**.

EXTENSIONS POSSIBLES :

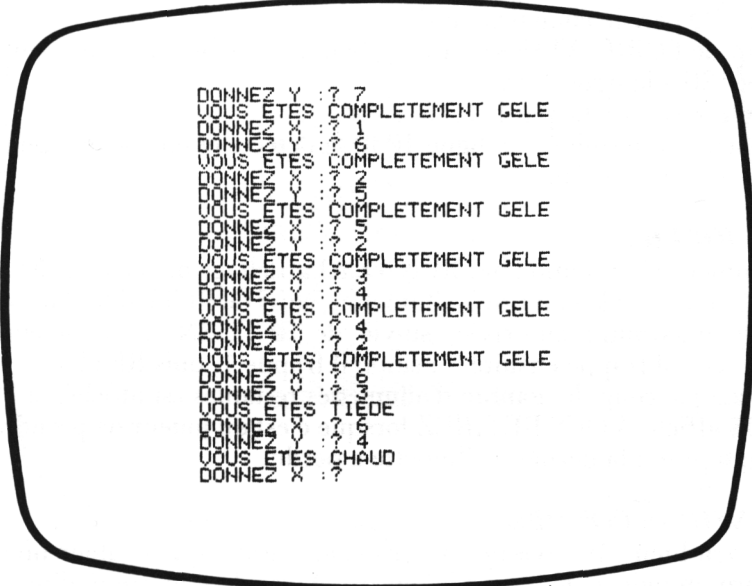
Si l'on dispose de la couleur, il sera intéressant de trouver un moyen plus visuel pour symboliser la distance, par exemple des pavés allant du rouge au bleu, aux emplacements essayés.

On peut aussi travailler dans un espace à trois dimensions, cacher plusieurs objets, etc.


```

10 REM CHAUD,FROID
20 CLS:PAPER 6
30 INPUT"DIMENSION DE LA GRILLE ";D
40 X=INT(RND(1)*D+1)
50 Y=INT(RND(1)*D+1)
55 CLS
60 INPUT"DONNEZ X ";A
70 INPUT"DONNEZ Y ";B
80 S=S+1
90 T=ABS(X-A)+ABS(Y-B)
100 IF T=0 THEN 6000
110 PRINT"VOUS ETES ";:PING
120 N=INT(SQR(D)/T)+1
130 IF N>5 THEN N=5
140 IF T<=SQR(D) THEN ON N GOTO 1000,2000,3000,4000,5000
150 PRINT"COMPLETEMENT GELE"
160 GOTO 60
1000 PRINT"A PEINE TIEDE"
1010 GOTO 60
2000 PRINT"TIEDE"
2010 GOTO 60
3000 PRINT"CHAUD"
3010 GOTO 60
4000 PRINT"TRES CHAUD"
4010 GOTO 60
5000 PRINT"BRULANT"
5010 GOTO 60
6000 CLS:PAPER 7
6010 PRINT"VOUS AVEZ TROUVE EN ";S;" COUPS":EXPLODE

```



```

DONNEZ Y : ? 7
VOUS ETES : ? 7 COMPLETEMENT GELE
DONNEZ X : ? 1
DONNEZ Y : ? 6
VOUS ETES : ? 6 COMPLETEMENT GELE
DONNEZ X : ? 2
DONNEZ Y : ? 2
VOUS ETES : ? 2 COMPLETEMENT GELE
DONNEZ X : ? 2
DONNEZ Y : ? 2
VOUS ETES : ? 2 COMPLETEMENT GELE
DONNEZ X : ? 3
DONNEZ Y : ? 3
VOUS ETES : ? 3 COMPLETEMENT GELE
DONNEZ X : ? 4
DONNEZ Y : ? 4
VOUS ETES : ? 4 COMPLETEMENT GELE
DONNEZ X : ? 2
DONNEZ Y : ? 2
VOUS ETES : ? 2 COMPLETEMENT GELE
DONNEZ X : ? 6
DONNEZ Y : ? 6
VOUS ETES : ? 6 TIEDE
DONNEZ X : ? 7
DONNEZ Y : ? 7
VOUS ETES : ? 7 CHAUD
DONNEZ X : ? 7

```

Les allumettes

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : contre l'ORIC-ATMOS.

Difficulté du jeu : moyenne.

LE JEU :

Le joueur devra en début de partie choisir une certaine quantité d'allumettes, ainsi que la prise maximum autorisée. Il décide ensuite s'il commence, ou s'il laisse commencer l'ORIC-ATMOS. Chacun prend à son tour autant d'allumettes qu'il le désire, dans la limite du nombre maximum autorisé.

Celui qui prend la dernière allumette est considéré comme perdant.

LE PROGRAMME :

Le jeu du joueur commence en ligne 70.

Celui de l'ORIC-ATMOS commence en ligne 110.

La ligne 80 contrôle la légalité du coup du joueur.

La ligne 90 gère le changement de joueur.

Le coup de l'ORIC-ATMOS est calculé dans une boucle non formelle, de la ligne 120 à la ligne 150.

Le résultat est affiché par la ligne 1020.

Si la partie est terminée, la ligne 1030 le détecte et envoie en ligne 2000, pour le message de fin.

POUR JOUER :

On commence l'exécution par RUN, et, à l'affichage du message NOMBRE TOTAL, on entre le nombre d'allumettes, suivi de RETURN. Puis on entre la prise maximum autorisée, suivie de RETURN. Si le joueur veut commencer, il frappe ensuite 0 ; sinon il frappe 1, puis RETURN.

Après chaque coup, le nombre d'allumettes restantes est affiché, et l'ORIC-ATMOS affiche VOUS RETIREZ lorsque c'est au joueur de prendre.

Celui qui prend la dernière allumette a perdu.

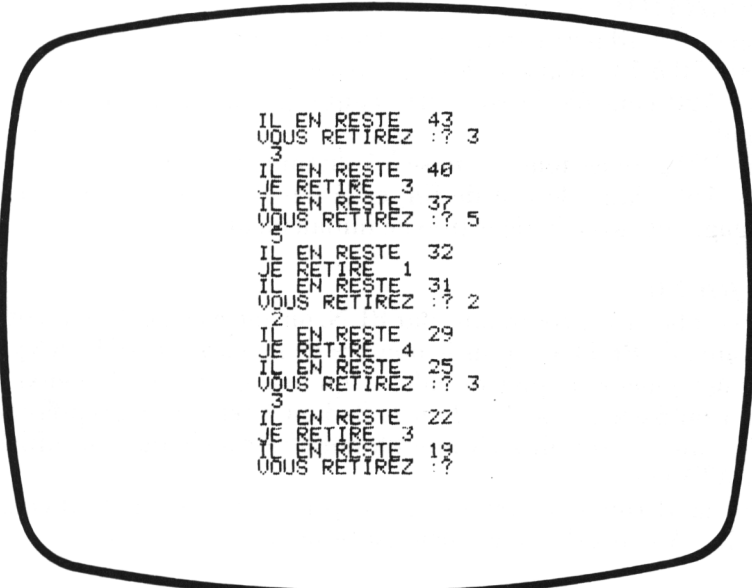
EXTENSIONS POSSIBLES :

On pourra bien sûr envisager un affichage graphique couleur plus spectaculaire, imaginer de laisser le hasard choisir le nombre d'allumettes et la prise maximum, etc.

```

10 REM LES ALLUMETTES
20 CLS
30 INPUT"OMBRE TOTAL :";N
40 INPUT"PRISE MAXIMUM :";I
50 INPUT"QUI COMMENCE (VOUS:0 / MOI:1) :";D
60 IF D=1 THEN 110
70 INPUT"VOUS RETIREZ :";P
80 IF P<>INT(P) OR P<1 OR P>N THEN 70
90 D=1
100 PING:GOTO 1000
110 S=I+D
120 T=(N-S)/(I+D)
130 IF T=INT(T) THEN 200
140 S=S-D
150 GOTO 120
200 P=ABS(S-D)
210 IF P=0 THEN P=D
220 D=0:PRINT"JE RETIRE ";ZAP:WAIT 50
1000 PRINT P
1010 N=N-P
1020 PRINT"IL EN RESTE ";N:WAIT 150
1030 IF N=0 THEN 2000
1040 GOTO 60
2000 CLS:PAPER 6
2010 IF D=1 THEN PRINT"J'AI GAGNE":ZAP:ZAP:ZAP
2020 IF D<>1 THEN PRINT"BRAVO":PING

```



```

IL EN RESTE 43
VOUS RETIREZ :? 3
3
IL EN RESTE 40
JE RETIRE 3
IL EN RESTE 37
VOUS RETIREZ :? 5
5
IL EN RESTE 32
JE RETIRE 1
IL EN RESTE 31
VOUS RETIREZ :? 2
2
IL EN RESTE 29
JE RETIRE 4
IL EN RESTE 25
VOUS RETIREZ :? 3
3
IL EN RESTE 22
JE RETIRE 3
IL EN RESTE 19
VOUS RETIREZ :?

```

Allumettes à plusieurs

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : logique (plusieurs joueurs).

Difficulté du jeu : assez difficile.

LE JEU :

Dans ce jeu pour plusieurs joueurs, le meneur de jeu devra, en début de partie, choisir un nombre d'allumettes, et indiquer à l'ORIC le nombre de joueurs.

À chaque coup, le joueur qui a le trait est autorisé à retirer du tas d'allumettes restantes jusqu'au double de ce que vient de retirer le précédent joueur ; celui qui retire la dernière allumette a gagné (ou perdu, si l'on modifie légèrement les règles).

LE PROGRAMME :

Les lignes 10 à 40 permettent les différentes entrées.

Les lignes 70 à 100 réalisent les impressions.

La ligne 120 contrôle le coup de chaque joueur (par rapport au coup précédent).

La ligne 180 gère les tours (remise à 1 du numéro du joueur).

La ligne 160 réalise le test de fin, et la ligne 210 affiche le numéro du joueur gagnant (avec la règle prévue initialement).

POUR JOUER :

Après avoir lancé le programme par RUN, on entre le nombre d'allumettes initial, suivi de RETURN puis le nombre de joueurs (et RETURN).

Le premier joueur ne pourra enlever qu'une allumette, le deuxième en enlever 1 ou 2 ; le suivant 1, 2 ou 1, 2, 3 ou 4 selon le coup précédent, et ainsi de suite ; le retrait maximum autorisé est à chaque fois affiché par l'ORIC-ATMOS.

Lorsque la dernière allumette a été prise, on obtient l'affichage LE JOUEUR... A GAGNE, et la partie est terminée.

EXTENSIONS POSSIBLES :

On peut convenir que celui qui prend la dernière allumette est éliminé du jeu, et recommencer la partie entre les joueurs restants qui seront ainsi éliminés les uns après les autres.

On peut aussi imaginer une représentation graphique des tas d'allumettes, tirer au hasard le nombre à enlever en début de partie, etc.

```
10 REM ALLUMETTES A PLUSIEURS
20 CLS
30 INPUT"OMBRE TOTAL D'ALLUMETTES :";N
40 INPUT"OMBRE DE JOUEURS :";J
50 F=1:K=1
60 INK K:PAPER 0
70 CLS
80 PRINT"TOUR DU JOUEUR :";K
90 PRINT"TOTAL RESTANT :";N
100 PRINT"MAXIMUM A ENLEVER :";F
110 PING:INPUT"VOTRE COUP :";C
120 IF C=0 OR C>F THEN 110
130 N=N-C
140 IF N<0 THEN N=N+C:GOTO 110
150 F=C+C
160 IF N=0 THEN 200
170 K=K+1
180 IF K>J THEN K=1
190 GOTO 60
200 INK 0:PAPER 7:CLS
210 PRINT"LE JOUEUR NUMERO ";K;" A GAGNE":EXPLODE
```



```
TOUR DU JOUEUR : 3
TOTAL RESTANT : 31
MAXIMUM A ENLEVER : 4
VOTRE COUP : ? 3
```

Calendrier perpétuel

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : gadget publicitaire.

Difficulté du jeu : aucune.

LE JEU :

Ce programme vous permettra de confectionner tous les ans votre calendrier ; vous pourrez ainsi vous en servir pour envoyer à vos amis un calendrier du mois de leur naissance, ou pour leur souhaiter une bonne fête, etc.

LE PROGRAMME :

Les lignes 30 et 40 vous permettront d'entrer les mois et l'année.

Les lignes 50, 60 et 70 calculent le numéro du premier jour du mois.

La ligne 80 calcule le nombre de jours du mois (il n'est pas tenu compte ici des 29 février, pour simplifier).

La boucle A (120-150) réalise l'impression du calendrier proprement dit.

POUR JOUER :

Après avoir lancé le programme par l'instruction RUN, on devra introduire l'année, sous la forme, par exemple 1983, suivie de RETURN.

On obtient alors la question MOIS, à laquelle il convient de répondre par le numéro du mois, suivant l'ordre usuel (1 pour JANVIER, 2 pour FEVRIER, etc.).

On fait ensuite RETURN, et le calendrier est alors affiché sur l'écran.

EXTENSIONS POSSIBLES :

On pourra sans difficulté prévoir l'affichage des 29 février, prévoir une option permettant l'affichage pour une année complète, prévoir l'introduction du nom des mois en clair, etc.

```

10 REM CALENDRIER PERPETUEL
20 CLS
30 INPUT"POUR QUELLE ANNEE ";A
40 INPUT"POUR QUEL MOIS ";M
50 D=INT(365.25*(A+(M<3)))+1
60 D=D+INT(30.6*(M+1-(M<3)*12))-
  INT((INT((A+(M<3))/100)-7)*.75)
70 D=D-7*INT(D/7)
80 M=31+(M=4 OR M=6 OR M=9 OR M=11)+(M=2)*3
90 PRINT
100 PRINT"  DIM LUN MAR MER JEU VEN SAM"
110 PRINT
120 FOR A=1 TO M
130 PRINT TAB(4*(D+A-1)+4);A;
140 IF D+A-1=6 THEN PRINT:D=D-7
150 NEXT A

```

```

POUR QUELLE ANNEE :? 1984
POUR QUEL MOIS :? 7
  DIM LUN MAR MER JEU VEN SAM
  1   2   3   4   5   6   7
  8   9  10  11  12  13  14
 15  16  17  18  19  20  21
 22  23  24  25  26  27  28
 29  30  31

```

Bataille navale

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : réflexion.

Difficulté du jeu : facile.

LE JEU :

Une bataille navale a priori assez simple, avec un seul navire caché au hasard dans une grille de 10 cases sur 10 cases.

A chaque coup, on devra tirer en donnant les coordonnées H (horizontale) et V (verticale) de la case visée, et l'ORIC-ATMOS répond en affichant la distance séparant le navire invisible de la case visée. Mais, si l'on tire trop près, le navire risque de se recacher au hasard.

LE PROGRAMME :

La boucle I (40-70) dessine le terrain de jeu.

Les lignes 90-100 définissent l'emplacement du navire.

Les coordonnées du tir sont entrées aux lignes 110 et 120, et la ligne 130 affiche le coup sur l'écran. La distance est calculée en ligne 140, et la ligne 150 regarde si l'on a ou non fait mouche.

La ligne 170 intervient pour déplacer le navire si les coups sont rapprochés.

La ligne 180 permet l'affichage de la distance.

POUR JOUER :

Le terrain de jeu est affiché après avoir frappé RUN. A l'affichage de LIGNE, on répond en donnant le numéro de la ligne horizontale sur laquelle on désire tirer. On tape ensuite RETURN et, à la question COLONNE, on répond en donnant le numéro de la colonne verticale suivi de RETURN.

Si on a tiré trop près, on obtient l'affichage PFFFF... qui signifie que le navire ennemi a changé d'emplacement.

Si on a tiré sur la bonne case, on obtient l'affichage du nombre total de tirs effectués pour gagner.

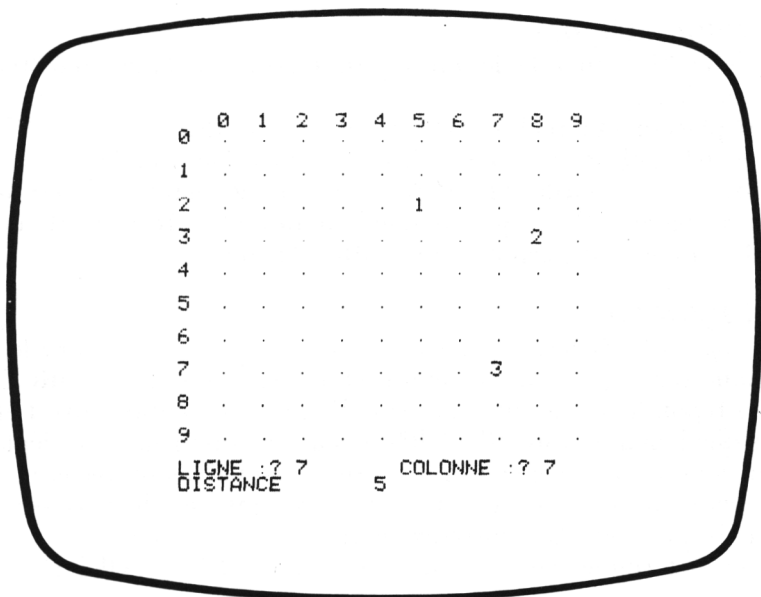
EXTENSIONS POSSIBLES :

Elles sont innombrables, puisqu'aussi bien la bataille navale est l'un des jeux les plus répandus qui soit. On peut introduire deux joueurs, jouer contre l'ORIC, mettre plusieurs navires, des sous-marins, etc.


```

10 REM BATAILLE NAVALE
20 PAPER 6:INK 0:CLS
30 PRINT"  0  1  2  3  4  5  6  7  8  9"
40 FOR I=0 TO 9
50 PRINTI;" . . . . . "
60 PRINT
70 NEXT I
80 S=1
90 X=INT(RND(1)*10)
100 Y=INT(RND(1)*10)
110 PRINT@3,21;"LIGNE :";INPUT H
120 PRINT@20,21;"COLONNE :";INPUT V
130 PRINT@3*V+5,2*H+1;S:EXPLODE
140 D=INT(SQR((X-H)*(X-H)+(Y-V)*(Y-V)))
150 IF D=0 THEN 200
160 S=S+1
170 IF D=1 THEN PRINT@3,22;"PFFFF...."      ":GOTO 90
180 PRINT@3,22;"DISTANCE      ";D;"      "
190 GOTO 110
200 CLS
210 PRINT"TROUVE EN ";S;" COUPS":ZAP:ZAP:ZAP

```



Golf

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez facile.

LE JEU :

Il s'agit d'essayer de réaliser, avec le moins de coups possible, un parcours de golf de 18 trous. Pour cela, il vous faut frapper la balle avec une force plus ou moins grande, jusqu'à obtenir sa chute dans le trou.

Mais attention, il peut y avoir certains aléas sur chaque coup, et vous serez amené souvent à revenir en arrière (il faudra alors faire précéder l'indication de la force par un signe -).

LE PROGRAMME :

La ligne 70 permet de définir la position du trou, et la ligne 80 dessine le terrain de jeu.

Le déplacement à réaliser est calculé en ligne 120.

La boucle I (140-190) réalise le déplacement de la balle.

La ligne 200 regarde si le coup est, ou non, gagnant ; s'il l'est, on recommence (ligne 250) jusqu'à ce que l'on ait réalisé 18 trous (boucle S (40-250) ; fin de boucle en 270, avec affichage de fin).

POUR JOUER :

On obtient l'affichage du terrain en frappant RUN ainsi que le dessin de la balle sur sa position de départ, et le texte : TROU 1. FORCE ; on doit alors répondre en donnant la force désirée (un nombre entre 1 et 300...).

Dès que l'on aura tapé RETURN, la balle recommence à se déplacer ; elle peut alors soit s'arrêter avant le trou (auquel cas on recommence), soit tomber dedans (on passe alors au trou suivant), soit le dépasser : on devra alors frapper avec une force négative.

Le jeu se termine avec l'affichage 18 TROUS EN ... COUPS.

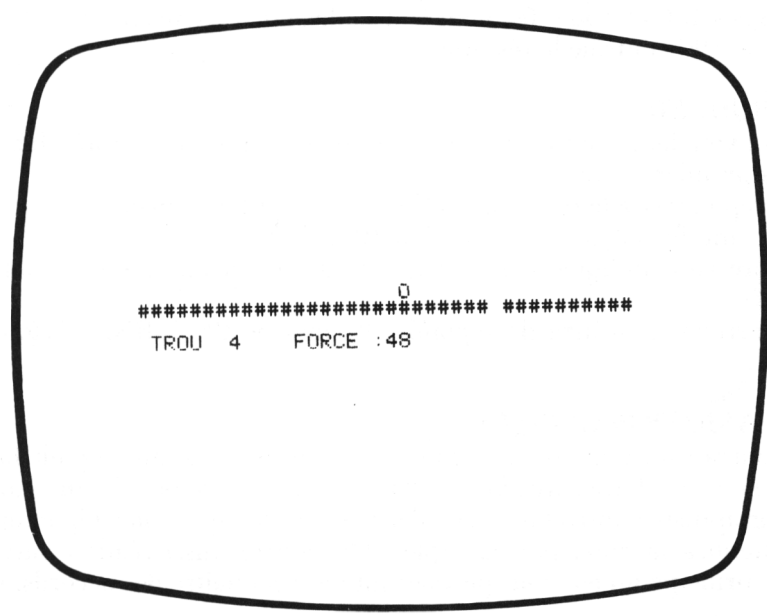
EXTENSIONS POSSIBLES :

Il sera possible d'étendre le jeu pour faire participer plusieurs joueurs, d'améliorer la qualité graphique, d'utiliser une représentation spatiale du terrain (se référer à certains jeux d'arcade), d'avoir un terrain plus accidenté, des obstacles, de changer de club...

```

10 REM GOLF
20 PAPER 3:CLS
30 PRINT CHR$(17)
40 FOR S=1 TO 18
50 FOR I=0 TO 5:PLOT0,I,CHR$(22):NEXT I
60 X=2
70 T=INT(29*RND(1)+9)
80 PLOT 0,6,CHR$(16):PLOT 1,6,CHR$(2)
90 PRINT@T,6;CHR$(255):PRINT@X,5;"0"
100 PRINT@15,8;" "
110 PRINT@3,8;"TROU ";S;:INPUT"   FORCE :";R:SHOOT
120 F=INT(R*0.16-(ABS(R))*(SQR(RND(1))+0.1)^2)/1000)
130 IF F=0 THEN 100
140 FOR I=SGN(F) TO F STEP SGN(F)
150 X=X+SGN(F)
160 IF X<39 AND X>=2 THEN PRINT@X-SGN(F),5;" " :
    PRINT@X,5;"0"
170 IF X>=38 THEN X=38
180 IF X<=2 THEN X=2
190 NEXT I
200 IF X=T THEN 230
210 E=E+1
220 GOTO 80
230 PRINT@X,5;" " :PRINT@X,6;"0"
240 WAIT 250:PRINT@X,6;" " :PING
250 NEXT S
260 PAPER 7:CLS
270 PRINT"18 TROUS EN ";E;" COUPS":ZAP:ZAP:ZAP

```



Solfège

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion.

Difficulté du jeu : facile.

LE JEU :

Ce programme vous propose une initiation élémentaire au solfège, en vous permettant d'apprendre à reconnaître facilement les différentes notes, qui apparaissent les unes après les autres sur une portée ; après l'affichage de la note, vous devez entrer son nom en clair et votre score sera affiché en permanence.

Le programme est initialement prévu pour la clé de sol, mais il sera très facilement adaptable à toute autre clé.

LE PROGRAMME :

La boucle I (50-70) dessine une portée.

La note est choisie en ligne 80 et affichée en lignes 90-100-110.

La ligne 120 affiche le score.

La ligne 130 affiche le nom de la clé (ici clé de sol) et vous interroge.

La ligne 140 s'occupe de la fin de la partie.

Les lignes 150 et 160 effectuent le décodage de la note entrée.

La ligne 190 contrôle le résultat.

POUR JOUER :

Après avoir lancé le programme par RUN, la portée sera affichée, ainsi que la première note.

Vous répondrez à la question NOTE en donnant le nom en clair de la note (do, ré, mi, fa, sol, la ou si), suivi de RETURN.

Le score vous indiquera si c'est exact ou non, et une nouvelle note sera affichée.

Pour arrêter, il suffira de répondre Ø suivi de RETURN, à la question NOTE.

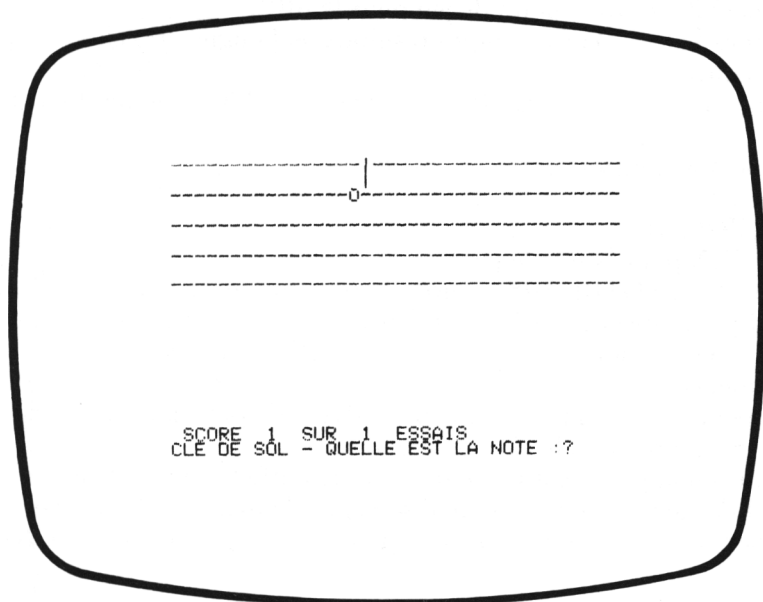
EXTENSIONS POSSIBLES :

On pourra tout d'abord facilement modifier le programme pour lui permettre de vous faire travailler les différentes clés. On pourra aussi améliorer le graphisme, introduire la couleur, le son (lorsque l'on a répondu juste, on entendra la note, par exemple). On pourra aussi réaliser suivant le même principe un jeu vous permettant de reconnaître des accords, etc.

```

10 REM SOLFEGE
20 PAPER 0:INK 7
30 S=0:P=0
40 CLS
50 FOR I=1 TO 5
60 PLOT 0,2*I+1,17
70 NEXT I
80 X=INT(RND(1)*12)
90 PRINT@18,X;"1"
100 PRINT@18,X+1;"!"
110 PRINT@17,X+2;"0"
120 PRINT@3,21;"SCORE ";S;" SUR ";P;" ESSAIS"
130 INPUT"CLE DE SOL - QUELLE EST LA NOTE :";R$
140 IF R$="0" THEN END
150 Y=(1 AND R$="FA")+ (2 AND R$="MI")+ (3
    AND R$="RE")+ (4 AND R$="DO")
160 Y=Y+(5 AND R$="SI")+ (6 AND R$="LA")
170 P=P+1
180 PING
190 IF Y=X OR Y=X-7 THEN S=S+1
200 GOTO 40

```



Le jeu du 21

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu contre l'ORIC-ATMOS.

Difficulté du jeu : facile.

LE JEU :

On devra entrer une mise au début du jeu, et on pourra lancer les dés en tapant 1 ; à chaque lancer, on obtiendra un nouveau total, qui augmentera peu à peu, et se rapprochera de 21. On devra essayer de se rapprocher le plus possible de 21, sans dépasser ce nombre. Lorsque l'on décidera de s'arrêter, il suffira de taper 0 pour laisser jouer l'ORIC-ATMOS qui essaiera alors d'avoir un meilleur score.

A la fin de chaque partie, le gain sera affiché ainsi que le total des gains (ou des pertes).

LE PROGRAMME :

Les lignes 80 et 90 réalisent le contrôle de celui qui doit jouer (joueur ou ORIC).

Le sous-programme (1000-1010) lance les dés.

Les lignes 110-120 calculent le total fait par le joueur et l'affichent.

Le total de l'ORIC est calculé en ligne 190 et affiché par la ligne 200.

La ligne 220 contrôle la fin de jeu pour l'ORIC.

Les lignes 260-280 affichent les résultats, et la ligne 300 cherche quel est le gagnant.

Le gain (ou la perte) sera affiché par la ligne 320, et le total par la ligne 350.

La ligne 380 contrôle la fin de partie.

POUR JOUER :

Taper RUN pour lancer le jeu.

A la question VOTRE MISE, entrer la mise initiale, suivie de RETURN, et taper 1 pour lancer le dé. Le premier tirage est affiché sous la forme VOUS AVEZ... et on devra refaire 1 suivi de RETURN, tant que l'on désire recommencer à tirer.

Dès que l'on a terminé, on tape 0 et RETURN pour laisser jouer l'ORIC-ATMOS (qui tient ici le rôle de la banque).

Le gain (ou la perte, s'il y a le signe -) sera affiché en fin de jeu, ainsi que le total des gains (ou pertes) réalisé depuis le début. Si l'on désire arrêter de jouer, on devra taper N en réponse à la question ON CONTINUE (O/N).

EXTENSIONS POSSIBLES :

Ce jeu sera évidemment plus intéressant s'il est pratiqué à plusieurs, et il sera aisé de modifier le programme en conséquence, on pourra aussi dessiner le dé, ou mieux, utiliser des cartes, ce qui permettra alors de faire un jeu de Black Jack.

```

10 REM LE JEU DU 21
20 G=0
30 PAPER 6
40 INK 4
50 CLS
60 T=0
70 INPUT"VOTRE MISE :";M
80 INPUT"FAIRE 1 POUR TIRER , 0 POUR FINIR";R
90 IF R=0 THEN 150
100 GOSUB 1000
110 T=T+X
120 PRINT"VOTRE TOTAL EST DE :";T;PING:WAIT 100
130 GOTO 80
150 CLS
160 PRINT"LA BANQUE JOUE";PRINT:PAPER 2
170 S=0
180 GOSUB 1000
190 S=S+X
200 PRINT"TOTAL DE LA BANQUE ";S
210 ZAP:WAIT 200
220 IF S>T OR S>17 THEN 250
230 GOTO 180
250 CLS:PAPER 7:INK 0
260 PRINT"VOUS AVEZ ";T
270 ZAP:ZAP:ZAP:WAIT 200
280 PRINT:PRINT"LA BANQUE A ";S
290 PRINT:ZAP:ZAP:ZAP:WAIT 200
300 IF S>21 OR (S<T AND T<21) THEN 320
310 M=-M
320 PRINT"GAIN ";M
330 PRINT
340 G=G+M
350 PRINT"VOTRE TOTAL ACTUEL ";G;EXPLODE
360 PRINT
370 INPUT"ON CONTINUE (O/N) :";R#
380 IF R#="N" THEN END
390 GOTO 30
1000 X=INT(RND(1)*6+1)
1010 RETURN

```

```

VOUS AVEZ 22
LA BANQUE A 19
GAIN -100
VOTRE TOTAL ACTUEL -100
ON CONTINUE (O/N) :? N

```

Pair-impair

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : facile.

LE JEU :

C'est un jeu de hasard, pour un ou plusieurs joueurs. L'ORIC lance simultanément deux dés, non visibles et pose trois questions : quelle est la parité du total, quel est le total, et enfin donner la valeur de l'un des dés. On peut ensuite soit poursuivre le jeu, soit l'interrompre à tout moment.

LE PROGRAMME :

Les dés sont tirés par un sous-programme placé aux lignes 1000 et 1010. La ligne 80 tient le compte du nombre de parties, et les lignes 180, 260 et 290 comptabilisent les points marqués.

La parité du total est déterminée par les lignes 130 et 140 (la ligne 140 compare avec la réponse du joueur).

La ligne 400 et les suivantes réalisent l'affichage de fin de partie : total, nombre de points, nombre de parties jouées.

La ligne 440 provoque l'arrêt si l'on répond N à la question ON CONTINUE.

POUR JOUER :

On lance le programme par l'instruction RUN.

Il suffit ensuite de répondre aux diverses questions posées, en suivant scrupuleusement les indications données et en tapant RETURN après chaque réponse.

Lorsque l'on a terminé une partie, on obtient l'affichage du total actuel et du nombre de parties sous la forme de, par exemple, 6 SUR 10.

On obtient alors l'affichage de la question ON CONTINUE (O/N) et l'on devra répondre N suivi de RETURN pour arrêter.

EXTENSIONS POSSIBLES :

Si l'on désire jouer à plusieurs, il sera intéressant de prévoir la possibilité de miser, et d'ajouter une gestion du score réalisé par chacun des joueurs.

On pourra aussi prévoir une amélioration sur le plan de la présentation (utilisation de la couleur, etc.).


```

10 REM PAIR-IMPAIR
20 T=0:R=0
30 INK 4
40 PAPER 6
50 GOSUB 1000
60 A=X
70 GOSUB 1000
80 R=R+1
90 S=A+X
100 CLS
110 INPUT"TOTAL PAIR (0) OU IMPAIR (1) :";N
120 PRINT
130 P=S/2-INT(S/2)
140 IF N=P OR N-0.5=P THEN 170
150 PRINT"FAUX":ZAP:WAIT 100
160 GOTO 190
170 PRINT"EXACT":PING
180 T=T+1
190 PRINT
200 INPUT"TOTAL PRESUME :";N
210 IF N=S THEN 250
220 PRINT"PERDU"
230 EXPLODE:WAIT 300
240 GOTO 400
250 PRINT
260 T=T+2
270 PING:INPUT"BRAVO . DONNEZ UN DE :";N
280 IF N<>A AND N<>S-A THEN 220
290 T=T+3
300 PRINT
310 ZAP:ZAP:ZAP:PRINT"BRAVO":GOTO 230
400 CLS:PAPER7:INK 0
410 PRINT"VOUS AVEZ UN TOTAL DE ";T;" SUR ";R
420 PRINT
430 INPUT"ON CONTINUE (O/N) :";R$
440 IF R$="N" THEN END
450 GOTO 40
1000 X=INT(RND(1)*6+1)
1010 RETURN

```

```

TOTAL PAIR (0) OU IMPAIR (1) : ? 1
EXACT
TOTAL PRESUME : ? 7
BRAVO . DONNEZ UN DE : ? 3
PERDU

```

Devine un chiffre

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : facile.

LE JEU :

C'est un jeu de hasard, de conception très simple, mais néanmoins assez passionnant : le joueur introduit sa mise, et parie sur un chiffre, de 1 à 6 ; trois dés sont alors lancés ; si le chiffre joué apparaît deux fois, il double sa mise ; s'il apparaît trois fois, il triple sa mise.

LE PROGRAMME :

Le tirage des dés est réalisé par un sous-programme placé en lignes 1000 et 1010.

L'affichage est réalisé par les lignes 150 et 160.

Les lignes 180 à 210 calculent le gain dans les diverses configurations possibles.

La totalisation est effectuée en ligne 230 pour les gains, et en ligne 510 pour les pertes.

La variable R\$, testée en ligne 550, détermine la poursuite éventuelle du jeu.

POUR JOUER :

Le lancement du programme par RUN provoque un premier affichage VOTRE MISE ; on répond par un nombre suivi de RETURN. On obtient alors un deuxième affichage DONNEZ UN CHIFFRE DE 1 A 6 ; on entre un chiffre suivi de RETURN ; les 3 dés, repérés par les lettres A, B, C s'affichent alors, ainsi qu'un message indiquant votre gain ou votre perte. Puis une indication sur votre total actuel apparaît, ainsi que la question ON CONTINUE (O/N) ; taper O pour continuer, N pour arrêter (suivi de RETURN).

EXTENSIONS POSSIBLES :

On pourra facilement introduire des contrôles (mise minimum, coup proposé de 1 à 6, etc.). On pourra aussi dessiner des dés avec un graphisme plus élaboré, introduire la possibilité de jouer à plusieurs, faire même participer l'ORIC-ATMOS.

```

10 REM DEVINE UN CHIFFRE
20 G=0
30 PAPER 5
40 INK 3
50 CLS
60 INPUT"VOTRE MISE :";N
70 PRINT
80 INPUT"DONNEZ UN CHIFFRE DE 1 A 6 :";D
90 GOSUB 1000
100 A=X
110 GOSUB 1000
120 B=X
130 GOSUB 1000
140 C=X
150 PRINT:PRINT:PRINT"      A          B          C"
160 PRINT:PRINT"      ";A;"          ";B;"          ";C"
170 T=0
180 IF D=A THEN T=T+N:ZAP
190 IF D=B THEN T=T+N:ZAP
200 IF D=C THEN T=T+N:ZAP
210 IF T=0 THEN 500
220 PRINT:PRINT:PRINT"VOUS GAGNEZ ";T
230 G=G+T
240 GOTO 520
500 PRINT:PRINT:PRINT"PERDU          "
510 G=G-N:PING
520 PRINT:PRINT:PRINT"TOTAL ACTUEL :";G
530 PRINT
540 INPUT"ON CONTINUE (O/N) :";R$
550 IF R$="N" THEN END
560 GOTO 30
1000 X=INT(RND(1)*6+1)
1010 RETURN

```

```

VOTRE MISE :? 100
DONNEZ UN CHIFFRE DE 1 A 6 :? 1

      A          B          C
      2          4          3

PERDU

TOTAL ACTUEL :-100
ON CONTINUE (O/N) :? N

```

Le jeu de Wytthof

REFERENCES :

Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : moyenne.

LE JEU :

On dispose d'une grille carrée de 10 cases sur 10 cases. Un point apparaît, en dehors de la diagonale et les joueurs devront le déplacer à chaque tour, suivant seulement trois directions : droite, bas et diagonale, bas et droite. Le premier joueur qui arrive en (9,9) sera considéré comme gagnant, ou perdant, selon la convention faite au début.

LE PROGRAMME :

Les lignes 40, 50 et 60 permettent de définir une position de départ, en dehors de la diagonale.
La ligne 80 et la boucle I (90-110) dessinent le terrain de jeu.
La ligne 130 imprime la position du point mobile.
Le test de fin est effectué en ligne 140.
Le changement de joueur est assuré par la ligne 180.
Les directions de déplacement disponibles, 1, 2 et 3, sont décodées par trois sous-programmes qui commencent aux lignes 1000, 2000 et 3000.

POUR JOUER :

Taper RUN pour lancer l'exécution du programme. Le terrain de jeu s'affiche, ainsi que la position de départ.
Puis on obtient l'affichage JOUEUR 1 et en dessous la question DIRECTION (1, 2, 3), à laquelle on répond en tapant l'un de ces trois nombres, suivi de RETURN.
C'est ensuite au tour du joueur 2.
Le jeu se termine lorsque l'un des joueurs arrive en case (9, 9).

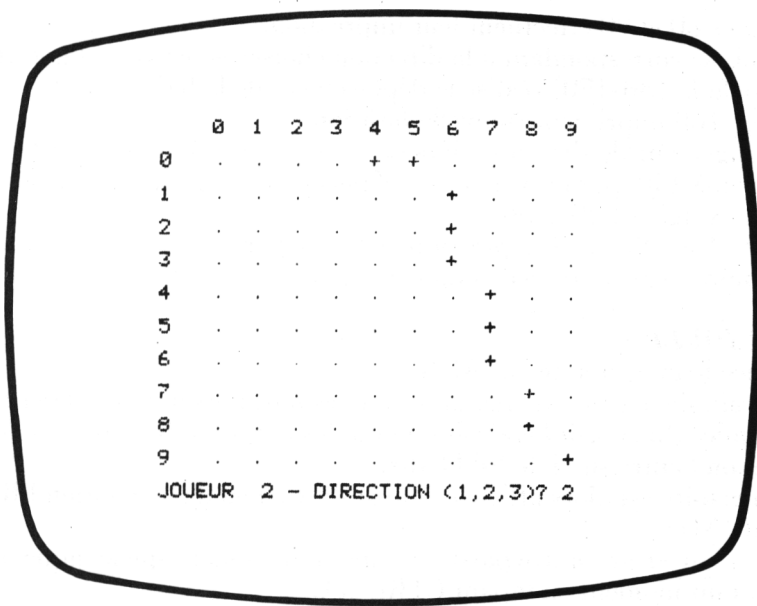
EXTENSIONS POSSIBLES :

On peut tout d'abord améliorer les contrôles (la sortie du terrain n'est pas interdite, par exemple), améliorer ensuite l'affichage (graphisme, couleur), étendre le jeu à plusieurs joueurs avec un terrain agrandi, introduire un monstre qui va essayer de capturer un des joueurs, prévoir un programme qui permettrait à l'ORIC-ATMOS de jouer, etc.

```

10 REM LE JEU DE WYTHHOF
20 PAPER 6
30 INK 4
40 X=3*INT(RND(1)*5)+5
50 Y=INT(RND(1)*5)
60 IF 3*Y+5=X THEN 50
70 CLS
80 PRINT"      0  1  2  3  4  5  6  7  8  9"
90 FOR I=0 TO 9
100 PRINT:PRINT I;" . . . . . ."
110 NEXT I
120 H=1
130 PRINT@X+2,2*Y+2;"+"
140 IF X=32 AND Y=9 THEN ZAP:ZAP:ZAP:END
150 PRINT@3,22;"JOUEUR ";H);:INPUT"- DIRECTION (1,2,3)";D$
160 D=VAL(D$):PING
170 ON D GOSUB 1000,2000,3000
180 H=2-ABS(H-1)
190 GOTO 130
1000 X=X+3
1010 RETURN
2000 GOSUB 1000
3000 Y=Y+1
3010 RETURN

```



Rebonds

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu d'adresse.

Difficulté du jeu : moyenne.

LE JEU :

Il utilise l'idée de la simulation d'un billard à bandes.

De part et d'autre de l'écran se trouvent la balle et le but visé : une série de chiffres et de lettres, en haut de l'écran, permettent de choisir l'endroit où l'on désire rebondir.

La balle va alors partir vers le haut de l'écran (vers le bas si l'on introduit un signe moins), y rebondir, revenir éventuellement frapper le bas, rebondir, et ainsi de suite jusqu'à atteindre la ligne de but ; si l'on est suffisamment près de la cible on a gagné et le nombre de rebonds sera alors affiché.

LE PROGRAMME :

La ligne 50 imprime les chiffres et les lettres en haut de l'écran.

Les lignes 60 et 70 définissent les positions initiales de la balle et de la cible.

Les lignes 80 et 90 effectuent leur impression.

Le symbole correspondant à la direction choisie est entré en ligne 100.

La boucle J (130-170) réalise le déplacement de la balle.

La ligne 160 imprime la nouvelle position.

Le changement de direction, lors de la rencontre d'un bord, est effectué par la ligne 150, par renvoi dans une partie du programme indépendante, située aux lignes 220 à 240.

La ligne 180 effectue l'impression de fin de partie.

Le nombre de rebonds est compté en ligne 230.

POUR JOUER :

On lance le programme par RUN.

Le terrain de jeu est affiché, ainsi que les positions initiales de la balle et de la cible. On choisit l'endroit où l'on désire faire rebondir la balle (de 1 à z), et on l'entre suivi de RETURN.

La trajectoire est alors affichée, suivie s'il y a lieu de la mention BRAVO ; X REBONDS.

On peut lancer une autre partie en tapant une touche quelconque, et s'arrêter à tout moment en tapant CTRL et C.

EXTENSIONS POSSIBLES :

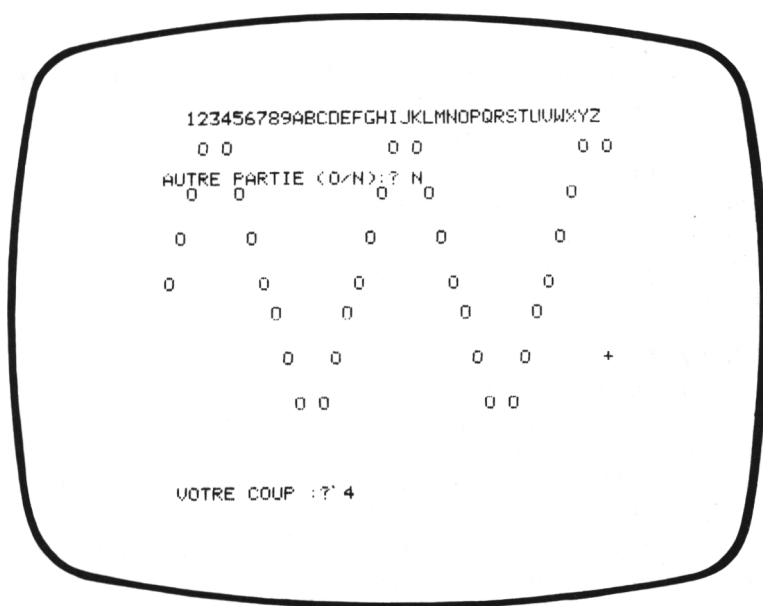
Il sera très intéressant, en utilisant un graphisme plus fin, de dessiner un véritable billard, avec son tapis vert, et d'indiquer la direction par un segment, symbolisant la queue de billard, se déplaçant autour de la bille.

On peut aussi ajouter des trous, mettre plusieurs billes, sonoriser les rebonds, ou les heurts avec d'autres balles, les comptabiliser pour faire une partie à plusieurs joueurs, etc.

```

10 REM REBONDS
20 PRINT CHR$(17)
30 PAPER 3:INK 1
40 CLS
50 PRINT" 123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ"
60 X=INT(RND(1)*18+2)
70 Y=INT(RND(1)*18+2)
80 PRINT@2,X;"0"
90 PRINT@39,Y;"+"
100 PRINT@3,25;"VOTRE COUP :":INPUT R$
110 S=0
120 T=X
130 FOR J=3 TO 39
140 T=T-X/(ASC(R$)-ASC("0")+7*(R$="A"))
150 IF T>=22 OR T<=0 THEN 220
160 PRINT@J,T;"0"
170 NEXT J
180 IF ABS(T-Y)<3 THEN PRINT"BRAVO :":S:" REBONDS":EXPLODE
190 INPUT"AUTRE PARTIE (O/N):":R$
200 IF R$="N" THEN END
210 GOTO 30
220 X=-X
230 S=S+1:PING
240 GOTO 170

```



Jackpot

REFERENCES :

Difficulté de programmation : facile.
Catégorie de jeu : hasard (un joueur).
Difficulté du jeu : facile.

LE JEU :

C'est une variante simplifiée d'un jeu de hasard classique, dans lequel le joueur doit faire une mise. S'il obtient certaines combinaisons de symboles, il gagne (plus ou moins, selon sa mise et la combinaison obtenue). Ici, les symboles sont remplacés par des chiffres, et le fait d'avoir trois chiffres identiques permet de gagner. Il existe de plus une combinaison particulière qui permet de remporter le Jackpot.

LE PROGRAMME :

La boucle I (60-80) permet le dessin de la machine, la ligne 90 affichant les emplacements (vides) des chiffres.
La mise sera entrée en ligne 100, et testée en ligne 110 : si on mise 0, la partie est terminée (départ vers la ligne 2000).
Les lignes 120, 130 et 140 permettent le tirage des 3 chiffres, en appelant un sous-programme qui se trouve en ligne 1000.
Le résultat est affiché par les lignes 150 à 170, puis testé en ligne 180, et en ligne 310, si l'on a gagné.

POUR JOUER :

Lancer le programme par RUN ; à l'affichage de la question POT DE DEBUT, on répond en entrant un nombre suffisamment élevé, sur lequel vos mises seront décomptées.
Puis on tape RETURN, et on entre une première mise (différente de 0), suivie de RETURN.
Les trois chiffres tirés sont alors affichés un bref instant, ainsi que la mise faite, et le résultat obtenu (gain ou perte).
On peut ensuite miser de nouveau, faire RETURN et ainsi de suite.
Pour terminer, il suffit de répondre 0 à la question VOTRE MISE.
Le restant de votre pot initial sera alors affiché.

EXTENSIONS POSSIBLES :

On pourra envisager d'améliorer sérieusement les contrôles, utiliser la couleur et des possibilités graphiques plus élaborées pour simuler davantage un véritable Jackpot.


```

10 REM JACKPOT
20 CLS:PAPER 6:INK 4
30 PRINT@3,0;"POT DE DEBUT ":";INPUT P
40 CLS
50 FOR I=1 TO 3:PLOT 1,I+4,CHR$(5):NEXT I
60 FOR I=1 TO 3
70 FOR J=0 TO 19:PRINT@10+J,I+4:CHR$(126):NEXT J
80 NEXT I
90 PRINT@14,6;"   ":";PRINT@19,6;"   ":";PRINT@24,6;"   "
100 PRINT@3,11;"VOTRE MISE ":";INPUT M
110 IF M=0 THEN 2000
120 GOSUB 1000
130 A=X:GOSUB 1000
140 B=X:GOSUB 1000
150 PLOT13,6,CHR$(0)+STR$(A)+CHR$(5):PING
160 PLOT18,6,CHR$(0)+STR$(B)+CHR$(5):PING
170 PLOT23,6,CHR$(0)+STR$(C)+CHR$(5):PING
180 IF A=B AND A=X THEN 300
190 N=-M
200 PRINT@3,11;"MISE ":";M;" - RESULTAT ":";N;" "
210 INPUT"TAPER - 0 RETURN - POUR CONTINUER";R$
220 P=P+N
230 GOTO 40
300 N=M*50:ZAP:ZAP:ZAP
310 IF A=7 THEN N=M*500:EXPLODE
320 GOTO 200
1000 X=INT(RND(1)*9+1)
1010 RETURN
2000 CLS
2010 PRINT:PRINT"VOUS AVEZ ";P
2020 PRINT:PRINT"A BIENTOT"

```

APPUYER SUR 0 - RETURN -?

3 5 0

MISE : 10 - RESULTAT :-10
TAPER - 0 RETURN - POUR CONTINUER?

Roulette

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : hasard (plusieurs joueurs).

Difficulté du jeu : facile.

LE JEU :

Il est surtout spectaculaire par son aspect graphique : une boule, représentée par un carré noir, se déplace sur un terrain de jeu comportant des cases numérotées de 1 à 36, ainsi que trois cases supplémentaires, le 00, une case P (comme pair) et une case I (comme impair).

La boule parcourt au moins une fois chaque case, puis s'arrête au deuxième tour, on peut ensuite la relancer pour une autre partie.

LE PROGRAMME :

La boule K (50-110) dessine le terrain de jeu, auquel la ligne 120 rajoute les trois symboles P, 00 et I.

La boucle K (130-190) réalise le déplacement de la boule, l'arrêt étant contrôlé par la ligne 170.

Le sous-programme 300-310 réalise une légère temporisation.

La ligne 200 tire la position d'arrêt.

La ligne 260 gère les essais suivants.

POUR JOUER :

Il suffit de faire RUN, et de regarder.

Une fois la boule arrêtée, il suffira de frapper une touche quelconque (sauf "N") pour la relancer. Taper N pour sortir du programme.

EXTENSIONS POSSIBLES :

On pourra essayer de se rapprocher davantage d'une véritable roulette, de différentes manières : mieux dessiner le terrain de jeu, introduire des couleurs, simuler le ralentissement de la boule, introduire différentes possibilités de pari, et assurer la gestion des gains correspondants, tenir les comptes de plusieurs joueurs, et annoncer les résultats sous une forme plus usuelle.

```

10 REM ROULETTE
20 PRINT CHR$(17)
30 PAPER 4:INK 3
40 CLS
50 FOR K=1 TO 4
60 FOR I=1 TO 9
70 S=I+I
80 PRINT TAB(4*I)(S AND K=1)+(S+18 AND K=2)+(S-1
  AND K=3)+(S+17 AND K=4);
90 NEXT I
100 PRINT:PRINT:PRINT:PRINT
110 NEXT K
120 PRINT@2,2;"P":PRINT@2,6;"00":PRINT@2,10;"I"
130 FOR K=1 TO 4
140 FOR I=1 TO 10
150 IF I=10 THEN 220
160 PRINT@4*I+1,4*K-3;"0":PING:WAIT 80 :PRINT@4*I+1,4*K-3;" ";
170 X=X-1:IF X=1 THEN ZAP:ZAP:ZAP:GOTO 250
180 NEXT I
190 NEXT K
200 X=INT(RND(1)*39)
210 GOTO 120
220 IF K=4 THEN 240
230 PRINT@2,4*K-1;"0":PING:WAIT 80 :PRINT@2,4*K-1;" ";
240 GOTO 170
250 PRINT CHR$(8);"0"
260 PRINT@2,20;"AUTRE ESSAI (O/N)":INPUT R$
270 IF R$="N" THEN END
280 GOTO 40

```

```

      2  4  6  8  10  12  14  16  18
P      0
      20 22 24 26 28 30 32 34 36
00
      1  3  5  7  9  11  13  15  17
I
      19 21 23 25 27 29 31 33 35

```

AUTRE ESSAI (O/N)? N

Couloirs

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

Il fait appel à des notions de logique élémentaire. Le joueur peut se présenter à l'entrée, soit du couloir numéro 1, soit du couloir numéro 2. Dans ces deux entrées sont disposés des tourniquets qui basculent à votre passage et vous laissent passer, une fois vers la droite, et la fois suivante vers la gauche. De plus, un tourniquet analogue a été disposé dans le couloir central, pourrez-vous prévoir où (sortie A ou B) va s'effectuer votre prochaine sortie ?

LE PROGRAMME :

Les lignes 30 à 70 dessinent les couloirs (et les trois tourniquets).

La ligne 100 attend votre choix.

Le décodage du chemin est fait par l'ORIC-ATMOS en utilisant trois variables logiques I, J et K, représentant les états des trois tourniquets (lignes 130 à 270).

POUR JOUER :

Après la frappe de RUN, le couloir est dessiné sur l'écran, les tourniquets étant visibles, et l'on devra taper 1 ou 2 en réponse à la question de l'ORIC-ATMOS.

Dans la version retenue ici, la réponse est alors immédiatement affichée, et le chemin suivi depuis le début du jeu, ainsi que les sorties correspondantes, sont affichés sur l'écran. Ceci présente l'avantage de permettre à plusieurs joueurs d'exercer leur sagacité en essayant de prévoir, pour une entrée donnée, ce que va répondre l'ORIC-ATMOS.

EXTENSIONS POSSIBLES :

Elles seront très nombreuses ; il est très facile d'imaginer un couloir plus complexe avec d'autres entrées et d'autres sorties, et d'autres tourniquets. On peut imaginer un affichage graphique des résultats en dessinant, un bref instant, l'état des tourniquets après la réponse du joueur. On peut aussi ajouter un test des réponses (jeu à plusieurs), décompter les points de chaque joueur, etc.).

```

10 REM COULOIRS
20 CLS
30 PRINT"-----"
40 PRINT"B           X  2"
50 PRINT"----- X -----"
60 PRINT"A           X  1"
70 PRINT"-----"
80 I=0:J=0:K=0
90 R$="":T$=""
100 PRINT@2,13;"ENTREE 1 OU 2 :? 2";:INPUT C$
110 IF C$="" OR C$>="3" OR C$="0" OR LEN(C$)<>1 THEN 100
120 R$=R$+C$
130 PRINT:PRINT R$
140 IF C$="1" THEN 190
150 J=NOT(J)
160 IF J<>0 THEN 250
170 S$="B":ZAP
180 GOTO 220
190 I=NOT(I)
200 IF I<>0 THEN 250
210 S$="A":PING
220 T$=T$+S$
230 PRINT@2,17;"SORTIE EN ";S$:PRINT:PRINT T$
240 GOTO 100
250 K=NOT(K)
260 IF K<>0 THEN 170
270 GOTO 210

```

```

B ----- X  2
----- X -----
A ----- X  1
-----

```

```

ENTREE 1 OU 2 :? 2
12112222212212
SORTIE EN A
BAABBABBBABBA

```

Echiquier

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : graphisme.

Difficulté du jeu : variable, selon utilisation.

LE JEU :

Le programme proposé vient en appoint du programme Echecs ; il donne, en effet, la possibilité de réaliser un affichage graphique de bonne qualité pour des jeux utilisant un damier de 8 cases (on pourra facilement l'étendre à 10 cases sur 10 cases). Un repérage a été prévu, horizontalement, par des lettres (de A à H), et, verticalement, par des nombres (de 1 à 8).

LE PROGRAMME :

La ligne 30 imprime les lettres horizontalement. Les deux boucles emboîtées I(40-110) et J(50-100) réalisent le dessin à l'aide de carrés hachurés. Le motif à imprimer est affiché ou non suivant la valeur de $(I+J)/2$, en ligne 60.

La ligne 120 met le programme en attente.

POUR JOUER :

L'ordre RUN provoque l'affichage du damier sur l'écran.

Pour utiliser le programme, il conviendra de tenir compte du repérage des cases utilisables qui sont les centres des carrés noirs ou gris. Les cases sont obtenues en largeur par une tabulation de la forme $4 \times J$, J variant de 1 à 8, et en hauteur par $3 \times I - 1$.

EXTENSIONS POSSIBLES :

On pourra utiliser ce programme pour toutes sortes de jeux se déroulant sur un damier (échecs, dames anglaises, le loup et les agneaux, etc.).

```

10 REM ECHIQUIER
20 PAPER 1:INK 3:CLS
30 PRINT"   A   B   C   D   E   F   G   H"
40 FOR I=1 TO 8
50 FOR J=1 TO 8
60 IF (I+J)/2=INT((I+J)/2) THEN 100
70 PRINT@4*J+1,3*I-1;CHR$(126)+CHR$(126)+CHR$(126):PRINT I
80 PRINT@4*J+1,3*I;CHR$(126)+CHR$(126)+CHR$(126)
90 PRINT@4*J+1,3*I+1;CHR$(126)+CHR$(126)+CHR$(126)
100 NEXT J
110 NEXT I
120 PRINT@2,0;:INPUT R$

```

	A	B	C	D	E	F	G	H
1		■		■		■		■
2	■		■		■		■	
3		■		■		■		■
4	■		■		■		■	
5		■		■		■		■
6	■		■		■		■	
7		■		■		■		■
8	■		■		■		■	

Le jeu de la fève

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : réflexion (deux joueurs).

Difficulté du jeu : moyenne.

LE JEU :

Présentant une certaine analogie avec le jeu de WYTTHOF, cette version offre en fait 4 possibilités de déplacement ; la position de départ est ici fixée en (0, 0), et pourrait éventuellement être modifiée.

Les directions de déplacement disponibles sont : droite, bas, diagonale bas et droite, diagonale bas et gauche.

On joue en principe avec la règle : le premier arrivé en (9, 9) a perdu.

LE PROGRAMME :

La ligne 30 et la boucle I (40-60) dessinent le terrain de jeu.

La ligne 100 réalise le test de fin de partie.

La ligne 130 permet d'exécuter l'un des sous-programmes de déplacement.

Les lignes 140 et 150 assurent le changement de joueur.

POUR JOUER :

On lance le programme par RUN, et le terrain de jeu s'affiche, ainsi que la question DIRECTION (1, 2, 3, 4) ; les 4 directions proposées correspondent aux déplacements suivants.



Le joueur 1 entre son déplacement, et frappe RETURN.

Le mouvement correspondant est réalisé, puis on obtient l'affichage JOUEUR 2 et DIRECTION.

Le joueur 2 procède de même, et ainsi de suite, jusqu'à ce que l'un des joueurs arrive en (9, 9) auquel cas le jeu se termine.

EXTENSIONS POSSIBLES :

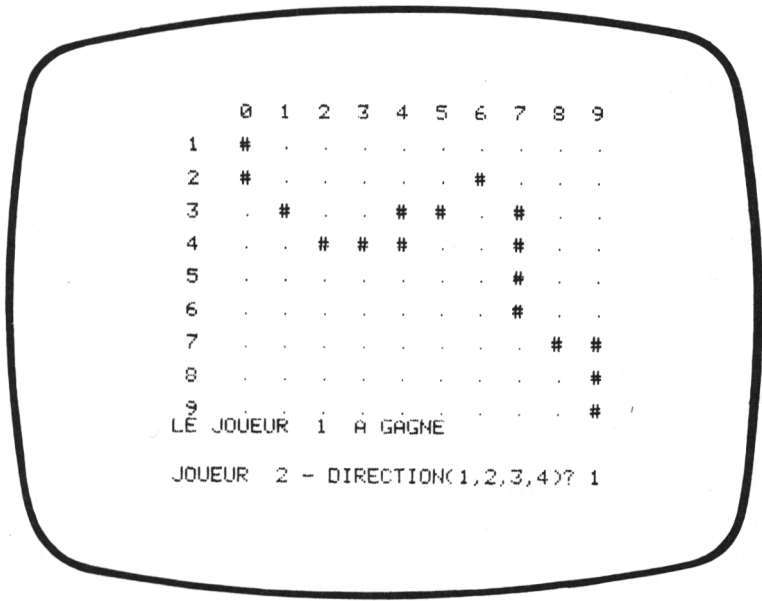
On peut imaginer d'introduire en début de partie des cases pièges (chacun essayant alors de faire venir l'autre sur l'une de ces cases).

On peut aussi jouer avec plusieurs symboles, si possible de couleurs différentes, déplacés simultanément, envisager des terrains de jeu triangulaires, aux dimensions arbitraires, etc.


```

10 REM LE JEU DE LA FEVE
20 CLS
30 PRINT"      0  1  2  3  4  5  6  7  8  9"
40 FOR I=1 TO 9
50 PRINT:PRINT I;"  .  .  .  .  .  .  .  .  .  ."
60 NEXT I
70 X=5:Y=1
80 J=1
90 PRINT@X+2,2*Y;"#"
100 IF X=32 AND Y=9 THEN 200
110 PRINT @2,22;"JOUEUR ";J;:INPUT"-
    DIRECTION(1,2,3,4)";I$
120 I=VAL(I$):PING
130 ON I GOSUB 1000,2000,3000,4000
140 J=J+1
150 IF J=3 THEN 80
160 GOTO 90
200 PRINT"LE JOUEUR ";J;
    " A GAGNE          ":ZAP:ZAP:ZAP
210 END
1000 Y=Y+1
1010 RETURN
2000 GOSUB 1000
3000 X=X+3
3010 RETURN
4000 GOSUB 3000
4010 Y=Y-1
4020 RETURN

```



Jeu de l'oie aléatoire

REFERENCES :

Difficulté de programmation : difficile.
Catégorie de jeu : hasard (deux joueurs).
Difficulté du jeu : facile.

LE JEU :

Le jeu reprend la très ancienne idée du jeu de l'oie, mais en y ajoutant la possibilité offerte par l'ordinateur de choisir au hasard les cases "à problème".

La règle du jeu est sans grand intérêt, puisque les joueurs n'interviennent à aucun moment dans le déroulement de la partie, qui est entièrement gérée par l'ORIC-ATMOS ; il suffira d'appuyer sur une touche après chaque lancer de dé, et de regarder ce qui se passe.

LE PROGRAMME :

La ligne 60 permet d'appeler le sous-programme 1000 à 1110 qui réalise l'affichage du terrain de jeu, le contrôle de la position des joueurs et l'affichage de cette position. Cet affichage sera d'ailleurs refait en cours de partie, pour tenir compte du déplacement réellement effectué (qui n'est pas nécessairement celui indiqué par le dé).

La ligne 70 s'occupe du changement de joueur, et la ligne 80 effectue le lancer de dé.

La ligne 90 traite le cas où l'un des joueurs termine.

La ligne 130 tire un nombre aléatoire entre 0 et 3, qui servira s'il y a lieu pour déterminer le vrai déplacement du joueur, en envoyant dans l'un des sous-programmes de "chance".

Les lignes 1000 et 1010 font avancer les pions, et les lignes 1020 à 1080 redessinent à chaque affichage le terrain de jeu en entier.

POUR JOUER :

On lancera le jeu par RUN. Le terrain de jeu est alors dessiné, puis on voit apparaître l'affichage JOUEUR 1 :...

On devra alors appuyer sur une touche quelconque pour que le coup du joueur numéro 1 soit pris en compte ; ne pas s'inquiéter du temps de réaction, qui pourra sembler un peu long.

Le déplacement indiqué par les dés sera exécuté, puis au bout de quelque temps, un second déplacement sera éventuellement effectué : c'est le déplacement "CHANCE". On obtiendra alors l'affichage JOUEUR 2.

Le joueur 2 appuie à son tour sur une touche quelconque et ainsi de suite...

Le joueur gagnant sera signalé par un affichage BRAVO en fin de partie.

On peut aussi s'arrêter à tout moment en tapant CTRL et C.

EXTENSIONS POSSIBLES :

On peut agrandir le terrain de jeu, introduire davantage d'aléas, changer leur nature. On pourrait aussi choisir les cases "chance" en début de jeu, ajouter des affichages en conséquence, etc.

```

10 REM JEU DE L'OIE ALEATOIRE
20 PAPER 6:INK 4:CLS:PRINT CHR$(17)
30 PLOT 0,5,CHR$(17):PLOT 0,6,CHR$(18)
40 PLOT1,5,CHR$(3):PLOT1,6,CHR$(0)
50 X=0:Y=0:J=1:I=X
60 GOSUB 1000
70 J=ABS(J-1)
80 I=INT(RND(1)*6+1)
90 IF X=35 OR Y=35 THEN PRINT@2,20;"BRAVO":EXPLODE:END
100 PING:PRINT@2,12;"JOUEUR ";J+1;" - VOTRE COUP :";I
110 INPUT "APPUYER SUR 0 ET - RETURN-";R$
120 GOSUB 1000
130 I=INT(RND(1)*4)
140 F=INT(RND(1)*5+1)
150 ON F GOSUB 200,300,400,500,160
160 GOTO 60
200 I=I+3
210 RETURN
300 I=-I
310 RETURN
400 I=2*I
410 RETURN
500 IF J=0 THEN X=INT(RND(1)*30):RETURN
510 Y=INT(RND(1)*30):RETURN
1000 IF J=0 THEN X=X-I*(X+I<=35)
1010 Y=Y-J*I*(Y+I<=35)
1020 PRINT@2,3;"0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ"
1030 FOR I=0 TO 35
1040 PRINT@I+2,5;" ";
1050 IF I=X THEN PRINT@I+2,5;"1";
1060 PRINT@I+2,6;" ";
1070 IF I=Y THEN PRINT@I+2,6;"2";
1080 NEXT I
1090 WAIT 100
1100 PRINT
1110 RETURN

```

0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ

2

1

JOUEUR 2 - VOTRE COUP : 6
APPUYER SUR 0 ET - RETURN-? 0

Niveau 2

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Dé truqué	A utiliser, si vous êtes malhonnête, dans d'autres programmes.
Visions 421	Un jeu sur les mots, pour l'ordinateur. Une simulation simplifiée, plusieurs joueurs.
L'électricien fou La course au 20 Musique	Logique et difficile, pour un seul joueur. Jeu de réflexion, pour deux joueurs ou plus. Révisez vos notions de solfège, grâce à l'ordinateur.
Poker	Une base de départ, pour mettre au point votre programme.
Le jeu de saute-mouton Lettres	Un jeu de réflexion, pour deux joueurs. Un jeu sur les mots, assez simple.
Machinamot	Laissez l'ordinateur créer des vocables nouveaux.
Espion Accrochez les wagons Tirage d'un dé truqué Tour de cartes	Dites-lui votre amour en code. Pour les plus jeunes ; très visuel. Révisez vos connaissances en statistiques. Votre ordinateur a plus d'un tour dans son sac.
Le jeu de Marienbad	Un jeu de logique pour deux joueurs, dans une version étendue.
L'indésirable	Vous êtes propriétaire. Arriverez-vous à mettre la main sur ce locataire indésirable ?
Compose Jour de la semaine	Faire de la musique, c'est d'abord l'écrire... Profitez-en pour faire le point sur les anniversaires.
Le jeu de la vie	Un grand classique, en deux versions : manuelle et automatique.

Dimensionnement d'un tableau Recherche d'un élément dans un tableau et dans une chaîne L'instruction DATA

L'INSTRUCTION DIM, OU INSTRUCTION DE DIMENSIONNEMENT D'UN TABLEAU

Les structures de tableaux sont très couramment utilisées en programmation, se révélant même indispensables dans un grand nombre de cas. Si vous voulez, par exemple, entrer vos recettes et dépenses pour tous les jours du mois (on supposera pour simplifier des mois de trente jours) vous pourriez définir 30 variables appelées PERMIER JOUR, DEUXIEME JOUR, etc. Il est plus facile, et plus rationnel d'utiliser une boucle I, I variant de 1 à 30, et une variable R (I) pour les recettes, et D (I) pour les dépenses ; ce faisant, vous avez, sans le savoir, utilisé une structure de tableau à une dimension ; ceci ne pourra être admis par l'ORIC-ATMOS que si vous l'avez auparavant prévenu que vous alliez utiliser un tel tableau, en donnant une instruction DIM R (30) pour les recettes, et DIM D (30) pour les dépenses.

On pourrait aussi imaginer de conserver dépenses et recettes dans un même tableau, en lui donnant cette fois deux dimensions : par exemple DIM RD (30, 2) (ou DIM RD (2, 30)). Ainsi, avec DIM RD (30, 2) on pourra convenir que RD (I, 1) représente les recettes, et RD (I, 2) les dépenses. Des tableaux de plus de 2 dimensions sont également possibles.

On peut aussi concevoir des tableaux dont les éléments sont des chaînes de caractères, et non plus des nombres, avec 255 caractères au maximum. Ainsi DIM NOM\$ (12) réserve 12 emplacements en mémoire, pour des mots de 255 caractères au maximum.

Sur l'ORIC-ATMOS il n'est pas obligatoire de dimensionner un tableau qui contient moins de 10 éléments.

LA RECHERCHE D'UN ELEMENT DANS UN TABLEAU : R (I)

Si l'on a un tableau à une dimension, numérique ou alphanumérique, défini par une instruction comme DIM R (30) ou DIM N\$ (20), le I^{ème} élément de ce tableau est repéré par R (I) (ou N\$ (I)) ; il sera alors manipulé exactement comme une variable ordinaire (affectation, calcul, tests, etc.).

Ainsi INPUT R (3) permet d'entrer à la main la valeur du 3^e élément numérique du tableau R. PRINT R\$ (5) permet l'affichage du 5^e élément du tableau alphanumérique R\$, etc.

LA RECHERCHE D'UN ELEMENT DANS UNE CHAINE : MID\$

Pour extraire le I^{ème} élément de la chaîne alphanumérique A\$, on utilisera l'instruction MID\$.

Ainsi, si A\$ = "BONJOUR", (A\$, 3, 1) donne le caractère N ; de plus MID\$ (A\$, 3, 3) donne la sous-chaîne "NJO".

LE TRAITEMENT DE DONNEES MULTIPLES INSTRUCTIONS READ, DATA, RESTORE

Pour remplir facilement un tableau, numérique ou alphanumérique, il sera intéressant d'utiliser des lignes DATA, qui contiennent les données inscrites les unes à la suite des autres, séparées par une virgule.

Ainsi, on pourra écrire :

DATA LEON, JULES, 13, RUE MARTIN...

Ces données peuvent être lues dans des boucles FOR... NEXT, au moyen de l'instruction READ. On peut recommencer la lecture au début des DATA, à l'aide de l'instruction RESTORE.

Dé truqué

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : facile.

LE JEU :

On veut essayer de fabriquer un dé truqué selon sa propre convenance. La méthode proposée ici consiste à définir au départ une probabilité de réalisation pour chaque type de tirage (soit la probabilité d'obtenir un 1, un 2, un 3, etc. que chacune de ces probabilités soit comprise entre 0 et 1, et que la somme de toutes les probabilités soit un nombre très voisin de 1).

LE PROGRAMME :

Il utilise le générateur aléatoire de l'ORIC-ATMOS supposé de qualité convenable (on pourra aisément utiliser tout autre générateur, si besoin est). La boucle I (70-140) s'occupe de l'entrée de chacune des probabilités, avec vérification à chaque entrée et sur le total (ligne 110) ; chaque probabilité est conservée dans une variable indicée E (I) : ligne 90 du programme.

Le lancer de dé truqué est effectué à l'aide d'une autre boucle I (180-200), la ligne permettant le trucage se trouvant en 190. Un nouveau tirage pourra être effectué sur pression de 0-RETURN (ligne 220).

POUR JOUER :

Après le lancement de l'exécution par RUN, on obtient l'affichage de la question PROBABILITE DU DE 1, à laquelle on devra répondre par un nombre compris entre 0 et 1. Il conviendra de ne pas oublier que le total de toutes les probabilités entrées ne devra en aucun cas dépasser 1.1 (tolérance de 0.1).

Des messages d'erreurs pourront être affichés et le programme sera éventuellement relancé. Lorsque toutes les probabilités auront été définies, le premier tirage sera automatiquement effectué, et il suffira de presser les touches 0-RETURN pour effectuer les tirages suivants. On arrête le programme en tapant CTRL et C.

EXTENSIONS POSSIBLES :

Il sera évidemment bien tentant d'utiliser surnoisement ce petit programme pour améliorer vos chances de gain dans un jeu de hasard utilisant des lancers de dés. On pourra alors envisager de figer les données définissant le dé truqué dans une variable A\$, etc. On pourra aussi envisager de présenter les résultats correspondant à une série de lancers de dés sous une forme spectaculaire, qui mette le trucage en évidence...


```

10 REM DE TRUQUE
20 PAPER 1:INK 0:CLS
30 PLOT 0,6,CHR$(19):PLOT1,6,CHR$(1):PLOT0,8,
  CHR$(23):PLOT1,8,CHR$(2)
50 P=0
60 DIM E(6)
70 FOR I=1 TO 6
80 PRINT@17,3;"
90 PRINT@2,3;"PROBABILITE DU DE ";I;:INPUT E(I)
100 P=P+E(I)
110 IF E(I)<0 OR E(I)>1 OR P>1.1 THEN 240
120 PRINT@6*I-4,6;"DE ";I
130 PRINT@6*I-4,8;E(I)
140 NEXT I
150 IF P<0.9 THEN PING:RUN
160 WAIT 150:ZAP:ZAP:ZAP
170 N=INT(RND(1)*6+1)
180 FOR I=1 TO 6
190 IF N=I AND RND(1)>E(I) THEN 170
200 NEXT I
210 PRINT@2,15;"TIRAGE ";N;"
220 PRINT:INPUT"FAIRE 0 - RETURN -
  POUR UN AUTRE TIRAGE";R$
230 GOTO 170
240 PRINT@2,4;"IMPOSSIBLE :0<= P <=1"
250 P=P-E(I)
260 PING:WAIT 200
270 PRINT@2,4;"
280 GOTO 90

```

```

PROBABILITE DU DE 6 ? 0.4
DE 1 DE 2 DE 3 DE 4 DE 5 DE 6
1 .3 1 .05 .05 .4

```

```

TIRAGE : 2
FAIRE 0 - RETURN - POUR UN AUTRE TIRAG
E?

```

Visions

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : dessin.

Difficulté du jeu : facile.

LE JEU :

L'ORIC-ATMOS vous permet, pour la première fois, d'entrer dans son intimité en vous laissant entrevoir sa propre représentation du monde ! Vous allez, en effet, lui proposer des mots, n'importe lesquels, et il va vous répondre en vous proposant une représentation graphique de ce que représente ce mot pour lui. Le jeu sera beaucoup plus amusant si vous le pratiquez à plusieurs, et vous pourrez quelquefois avoir des surprises.

LE PROGRAMME :

Une boucle I (40-120) se charge de réaliser les dessins.

La ligne 50 permet de transformer en nombre chaque lettre du mot entré au début (ligne 30). Ce nombre sera ensuite traduit en base 2 par une boucle J (lignes 60 à 110) ; selon le résultat de cette transformation, un point sera imprimé, ou non (ligne 90).

POUR JOUER :

On débute l'exécution du programme par RUN, ce qui amène l'affichage de la question QUEL DESSIN. On répond en entrant un mot quelconque qui sera transformé en dessin par l'ORIC-ATMOS dès que l'on aura frappé RETURN.

Pour recommencer, il suffira de frapper 0 et RETURN, et CTRL et C pour finir. Je vous laisse savourer le dessin qu'il a réalisé lorsqu'il a connu le prénom de l'auteur de ce livre.

EXTENSIONS POSSIBLES :

La même idée peut être reprise en introduisant des éléments de symétrie dans les dessins, en ajoutant la couleur. On peut aussi imaginer de coder par cette méthode un certain nombre de dessins, que l'on introduirait au fur et à mesure dans la machine et l'on pourra alors créer un jeu dans lequel ORIC deviendra de plus en plus savant. On peut aussi réaliser un jeu à deux joueurs dans lequel ORIC vous demande de trouver le mot qui correspond à un certain dessin, etc.

```

10 REM VISIONS
20 CLS
30 INPUT"QUEL DESSIN ?";R$
40 FOR I=1 TO LEN(R$)
50 N=9*ASC(MID$(R$,I,1))-340
60 J=1:K=INT(RND(1)*6+1)
70 S=N/2-INT(N/2)
80 N=INT(N/2):PLOT1,I+5,CHR$(K)
90 IF S<>0 THEN PLOTJ+15,I+5,CHR$(127)
100 J=J+1
110 IF J<10 THEN 70
120 NEXT I
130 PRINT@2,22;:PING
140 INPUT"APPUYER SUR 0 - RETURN -";R$
150 GOTO 20

```

QUEL DESSIN ? JACQUES



421

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : moyenne.

LE JEU :

Chacun des joueurs devra, à son tour, lancer 3 dés, et peut ensuite choisir, sur les deux coups suivants, s'il désire relancer ou non 1, 2 ou 3 dés. Il marque alors un score correspondant au nombre indiqué par les dés, suivant les règles choisies, et le joueur suivant procède de la même manière.

LE PROGRAMME :

Le nombre de joueurs est tout d'abord entré en ligne 30.

La ligne 130 affiche le numéro de joueur et le nombre d'essais déjà effectués.

Un sous-programme placé en ligne 4000, utilisé 3 fois dans la boucle J (200-250), réalise le lancer de dés.

Les retirages sont effectués par les lignes 1000, 2000 et 3000.

La variable R\$ gère les 2^e et 3^e tirages. L'impression du résultat est réalisée dans tous les cas par les lignes 140-150.

Les tests pour le nombre d'essais et pour le changement de joueur sont effectués aux lignes 260 et 280.

POUR JOUER :

On lance l'exécution du programme par RUN ce qui conduit à l'affichage NOMBRE DE JOUEURS. On entre le nombre de joueurs, et l'on tape RETURN. L'écran s'efface et affiche JOUEUR N° 1. ESSAI A, PARTIE P, ainsi que le tirage réalisé, et les noms des dés (A, B, C).

Si l'on ne désire pas refaire de tirage, il suffit de frapper Ø et RETURN.

La main passe alors au joueur suivant.

Pour effectuer un autre tirage, il suffit de taper le nom du ou des dés à retirer, et de faire RETURN : par exemple, pour retirer A et C on tapera AC suivi de RETURN.

Les joueurs suivants procéderont de la même façon.

EXTENSIONS POSSIBLES :

La plus évidente consiste à assurer la gestion des résultats des joueurs, en se référant aux règles de gain usuelles du jeu du 421. On pourra envisager l'affichage après chaque tour d'un tableau récapitulatif des résultats des joueurs.

On pourra aussi améliorer l'affichage des dés (se référer aux divers programmes d'affichage des dés se trouvant dans le même ouvrage).

```

10 REM 421
20 P=1
30 CLS:INPUT"OMBRE DE JOUEURS :":N
40 I=1
50 CLS
60 PAPER 0:INK 7:PLOT0,4,CHR$(22):PLOT1,4,CHR$(0)
70 FOR L=1 TO 5:PLOT0,L+5,CHR$(19):PLOT1,L+5,CHR$(1):NEXT L
80 K=0
90 R$="ABC"
100 GOTO 200
110 K=K+1
120 PRINT@17,1;CHR$(27);"J";"421":PRINT@17,2;CHR$(27);"J";"421":ZAP:ZAP:ZAP
130 PRINT@2,4;"JUEUR NO.":I;" ESSAI":K;" PARTIE":P
140 PRINT@7,7;A:PRINT@17,7;B:PRINT@27,7;C
150 PRINT@8,9;"A":PRINT@18,9;"B":PRINT@28,9;"C"
160 R$="":PRINT@2,14;" "
170 PRINT@2,14;"ABC :":INPUT R$
180 IF R$="0" THEN 270
190 R$=R$+"DDD"
200 FOR J=1 TO 3
210 GOSUB 4000
220 X$=MID$(R$,J,1)
230 X=ASC(X$)-64
240 ON X GOSUB 1000,2000,3000,4000
250 NEXT J
260 IF K<3 THEN 110
270 I=I+1:PING:WAIT 50
280 IF I<=N THEN 80
290 P=P+1
300 GOTO 40
1000 A=D
1010 RETURN
2000 B=D
2010 RETURN
3000 C=D
3010 RETURN
4000 D=INT(RND(1)*6+1)
4010 RETURN

```

421

JUEUR NO. 1	ESSAI 1	PARTIE
4	4	6
A	B	C

ABC :? C

L'électricien fou

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez difficile.

LE JEU :

Un immeuble comportant 10 fenêtres, numérotées de 0 à 9, apparaît sur l'écran. Le joueur devra à chaque essai manœuvrer l'un des interrupteurs, en indiquant son numéro (de 0 à 9).

En principe, la fenêtre correspondante devrait s'éteindre, mais il est aussi possible qu'une autre s'allume (ou s'éteigne). Il s'agit, bien sûr, d'essayer d'obtenir l'extinction des 10 fenêtres, avec le moins possible d'essais.

LE PROGRAMME :

Une table A, en ligne 50, contiendra l'état (allumé ou éteint) des 10 fenêtres.

La boucle I (70-90) et l'autre boucle I (110-160) dessinent l'immeuble dans son état initial.

La ligne 170 regarde si tout est éteint, auquel cas l'on ira en ligne 310 afficher le nombre d'essais.

La ligne 180 attend l'indication de l'interrupteur à manœuvrer.

Les lignes 190 à 290 définissent le résultat de la manœuvre.

POUR JOUER :

RUN provoque l'affichage de l'immeuble, avec toutes les fenêtres allumées.

A la question INTERRUPTEUR NUMERO, on répond en indiquant le numéro de l'interrupteur à manœuvrer, suivi de RETURN.

Le résultat de la manœuvre est alors affiché sur l'écran (des fenêtres s'allument ou s'éteignent).

On continue ainsi jusqu'à ce que toutes les fenêtres soient éteintes, et l'on obtient alors l'affichage du nombre d'essais réalisés.

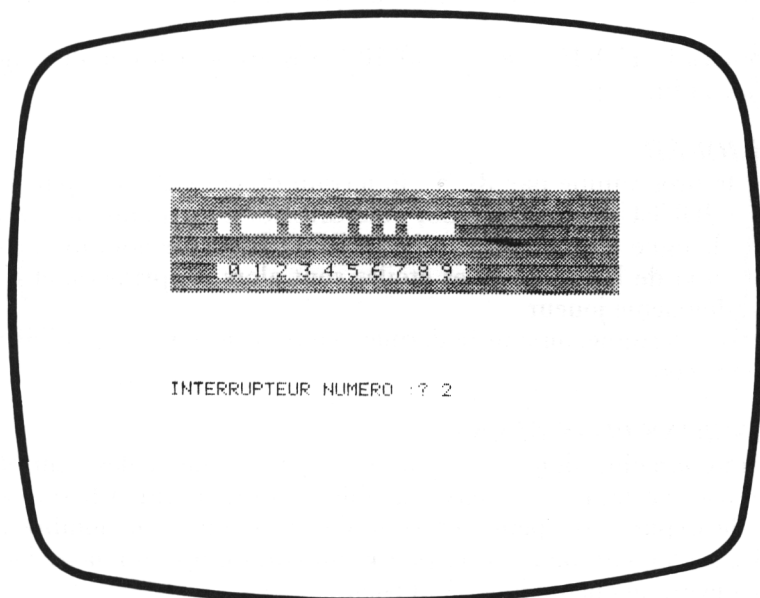
EXTENSIONS POSSIBLES :

On pourra envisager un graphisme très amélioré avec si possible l'utilisation de la couleur. On pourra aussi définir un jeu à deux joueurs, augmenter le nombre de fenêtres, modifier la formule définissant l'allumage ou l'extinction...

```

10 REM L'ELECTRICIEN FOU
20 PAPER 7
30 INK 4
40 M=0:PRINT CHR$(17)
50 DIM A(10)
60 CLS:PLOT1,6,CHR$(2)
70 FOR I=1 TO 7
80 PLOT0,I,CHR$(16)
90 NEXT I
100 T=0
110 FOR I=1 TO 10
120 T=T+A(I)
130 PRINT@2*I+5,3;" "
140 IF A(I)=0 THEN PLOT2*I+4,3,CHR$(4)+CHR$(255)
150 PRINT@2*I+4,6:I-1
160 NEXT I
170 IF T=10 THEN 310
180 PRINT@2,14;"INTERRUPTEUR NUMERO :";PING:INPUT N
190 N=N+1
200 M=M+1
210 IF A(N)=1 THEN 290
220 A(N)=1
230 R=TAN(RND(1)+N/RND(1)-N)-SIN(RND(1)/N)+336*SIN(8*N)
240 N=INT(10*(R-INT(R)))
250 IF N=0 THEN N=INT(RND(1)*10+1)
260 IF A(N)=1 THEN 290
270 A(N)=1
280 GOTO 100
290 A(N)=0
300 GOTO 230
310 PRINT:PRINT"TROUVE EN ";M;" ESSAIS":EXPLODE

```



La course au 20

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : logique (deux joueurs, ou plus).

Difficulté du jeu : moyenne.

LE JEU :

Il est possible d'imaginer pour ce jeu diverses règles, et nous n'avons retenu ici que la règle usuelle pour deux joueurs, qui en fait un jeu apparenté aux jeux de déplacement, ou d'allumettes : 4 tas de 20 étoiles vont apparaître en début de partie, et chaque joueur peut prendre autant d'étoiles qu'il le désire ; mais ceci dans un seul tas à la fois. Celui qui prend la dernière étoile est considéré comme perdant.

LE PROGRAMME :

La boucle I (30-50) dessine 4 rangées de 20 étoiles.

La variable F, dimensionnée en ligne 70, servira à enregistrer le nombre d'étoiles prises dans chaque rangée.

La variable T contiendra le nombre total d'étoiles enlevées (ligne 60).

La ligne 120 va contrôler qu'il reste suffisamment d'étoiles sur la ligne indiquée.

Les boucles I (150-190) et J (160-180) suppriment les étoiles ; la ligne 220 teste la fin de partie.

POUR JOUER :

Lancer le programme par RUN ; le terrain de jeu s'affiche, puis les inscriptions JOUEUR 1 et LIGNES, puis CASES. On répond en entrant un numéro de ligne (entre 1 et 4) suivi de RETURN, et le nombre d'étoiles à enlever suivi de RETURN. Les étoiles sont alors supprimées, et c'est au tour du deuxième joueur.

La partie se termine, lorsque la dernière étoile a été enlevée, par l'affichage du joueur gagnant.

EXTENSIONS POSSIBLES :

Outre une amélioration de la qualité du graphique et des contrôles, de nombreuses variantes sont envisageables, en modifiant si besoin est, les règles de départ : on peut autoriser l'enlèvement d'un nombre impair d'étoiles seulement, ou alterner pair et impair, ou tirer au hasard le nombre à enlever, ou encore jouer à plusieurs...


```

10 REM LA COURSE AU 20
20 CLS
30 FOR I=1 TO 4
40 PRINT@2,2*I;I;:FOR K=1 TO 20:PLOT6+K,2*I,"*":NEXT K
50 NEXT I
60 T=0
70 DIM F(4)
80 B=1
90 PRINT@2,12;"JOUEUR ";B:PING
100 PRINT:INPUT"LIGNE :";L
110 PRINT:INPUT"CASES :";C
120 IF C+F(L)>20 THEN 90
130 F(L)=F(L)+C
140 T=T+C
150 FOR I=1 TO 4
160 FOR J=F(I)-C+1 TO F(I)
170 IF I=L THEN PRINT@27-J,2*I;" ":SHOOT:WAIT 30
180 NEXT J
190 NEXT I
200 B=B+1
210 IF B=3 THEN B=1
220 IF T<80 THEN 90
230 PRINT"LE JOUEUR ";B;" GAGNE":EXPLODE

```

```

1 *****
2 *****
3 *****
4 *****

```

```

JOUEUR 2
LIGNE :? 1
CASES :? 10

```

Musique

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu éducatif.

Difficulté du jeu : moyenne.

LE JEU :

Il se présente comme un exercice d'apprentissage du solfège : l'appareil annonce une note (en clé de sol), suivie de la mention GRAVE ou AIGUE, selon sa place sur la portée. Il faudra répondre en indiquant sur quelle ligne la note sera affichée.

La note est alors dessinée sur la portée, suivie d'un message BRAVO ou PERDU, selon que l'on avait trouvé juste ou non.

LE PROGRAMME :

La boucle I (50-70) dessine la portée. Une deuxième boucle I (80-120) affiche les numéros des lignes.

La ligne 130 définit le nom des notes, et la ligne 140 réalise le tirage aléatoire d'une note.

Le nom de la note est calculé à la ligne 150 et il est affiché par la ligne 160, suivi de la mention GRAVE ou AIGUE, selon le cas.

La réponse sera entrée en ligne 190.

La ligne 200 s'occupe du décodage de la réponse.

La ligne 210 affiche la note.

Les lignes 220 et 230 regardent si l'on a gagné ou perdu et affichent le message correspondant.

La ligne 240 permet de faire un nouvel essai.

POUR JOUER :

On lance le programme par RUN.

On obtient alors l'affichage de la portée, ainsi que le nom de la note à placer et la question NUMERO DE LA LIGNE.

On répond en donnant un numéro de ligne (de 1 à E) suivi de RETURN.

La note est alors dessinée et le message BRAVO ou PERDU apparaît, dans ce cas, à l'écran.

On doit taper les touches Ø-RETURN pour continuer; et s'arrêter en tapant CTRL et C.

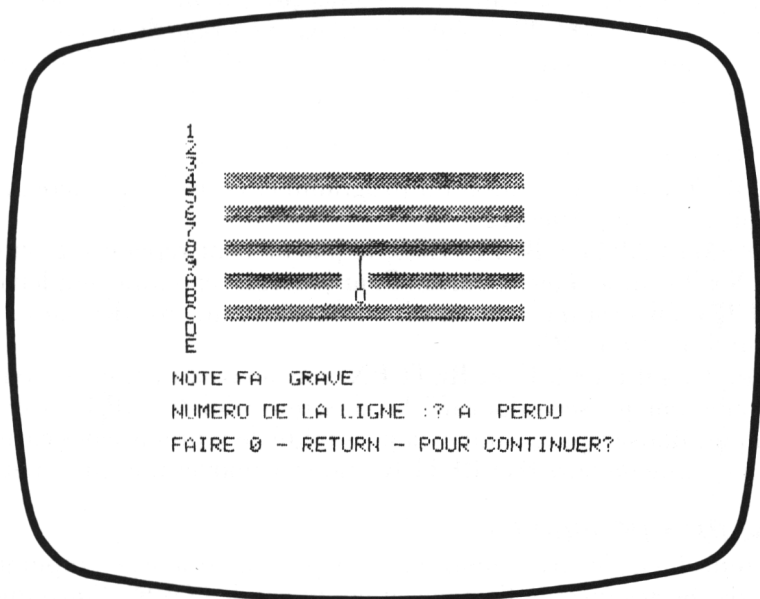
EXTENSIONS POSSIBLES :

On peut étendre à une autre clé que la clé de sol, on peut introduire de la couleur, des sons, etc.

```

10 REM MUSIQUE
20 PAPER 0
30 INK 6
40 CLS
50 FOR I=1 TO 5
60 PRINT@6,2*I+4;CHR$(27);"Q"
70 NEXT I
80 FOR I=1 TO 14
90 PRINT@2,I+2;
100 IF I<=9 THEN PRINT I
110 IF I>9 THEN PRINT " ";CHR$(I+55)
120 NEXT I
130 N$="SI LA SOLFA MI RE DO "
140 X=INT(RND(1)*14+1)
150 T=X-1-7*INT((X-1)/7)
160 PRINT@2,18;"NOTE ";MID$(N$,3*T+1,3);
170 IF X<=7 THEN PRINT" AIGUE"
180 IF X>7 THEN PRINT " GRAVE"
190 PRINT:INPUT"NUMERO DE LA LIGNE :";L$
200 L=- (ASC(L$)-48)*(L$<"")-(ASC(L$)-55)*(L$>"A")
210 PLOT15,X,CHR$(3)+"!":PLOT15,X+1,CHR$(3)+"!":
  PLOT15,X+2,CHR$(3)+"0"
220 IF L=X THEN PRINT@27,20;"BRAVO":ZAP:ZAP:ZAP
230 IF L<>X THEN PRINT@27,20;"PERDU":EXPLODE
240 PRINT:INPUT"FAIRE 0 - RETURN -
  POUR CONTINUER";R$
250 GOTO 40

```



Poker

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu contre l'ORIC-ATMOS.

Difficulté du jeu : facile.

LE JEU :

C'est une simulation d'une partie de poker traditionnelle, effectuée avec un jeu de 32 cartes, dans laquelle le joueur aura, pour partenaire, l'ORIC-ATMOS.

Le joueur recevra 5 cartes, repérées par des numéros de 1 à 5, et pourra décider de changer certaines cartes en annonçant celles qu'il écarte. Sa nouvelle donne sera alors affichée, et c'est ensuite au tour de l'ORIC-ATMOS de jouer.

LE PROGRAMME :

Les lignes 50 et 60 définissent les cartes. Une variable N est dimensionnée à 15, en ligne 40, pour recevoir 10 cartes joueur et 5 cartes ORIC.

Le tirage sera effectué par une boucle I (80-140) et le contrôle par une boucle J (110-130).

L'affichage des cartes est réalisé par le sous-programme 500-600.

Une autre boucle I (200-220) réalise le deuxième tirage du joueur, la ligne 240 permettant le deuxième affichage, et la boucle I (500-570) où l'on entre directement après, réalisera l'affichage du jeu machine.

POUR JOUER :

L'ordre RUN lance l'exécution du programme, et les 5 premières cartes du joueur sont alors affichées.

A la question CARTES REFUSEES, le joueur peut répondre F suivi de RETURN s'il veut arrêter la partie. Pour jouer, il répondra simplement 0 et RETURN s'il s'estime servi, ou il entrera les numéros des cartes qu'il refuse suivis de RETURN.

Par exemple, s'il tape 235 et RETURN, son nouveau jeu conservera les cartes 1 et 4. Son jeu sera alors affiché ainsi que celui de l'ORIC, ce dernier n'ayant pas, dans ce programme, la possibilité de reprendre des cartes.

Il suffira ensuite de faire 0 et RETURN pour démarrer une autre partie.

EXTENSIONS POSSIBLES :

Elles seront, bien sûr, très nombreuses, depuis un affichage graphique en couleur des cartes jusqu'à l'amélioration du jeu de l'ORIC, l'introduction des règles sur les paris et même, pourquoi pas, des possibilités de bluff.

```

10 REM POKER
20 PAPER 0
30 INK 6
40 DIM N(15)
50 V$="7 8 9 10VADAR0AS"
60 C$="TRCACOPI"
70 CLS
80 FOR I=1 TO 15
90 N(I)=INT(RND(1)*32)
100 IF I=1 THEN 140
110 FOR J=1 TO I-1
120 IF N(I)=N(J) THEN 90
130 NEXT J
140 NEXT I
150 K=0:P=0
160 PRINT@2,12;"CARTES REFUSEES":PING
170 GOSUB 500
180 IF R$="F" THEN END
190 IF R$="0" THEN 230
200 FOR I=1 TO LEN(R$)
210 N(VAL(MID$(R$,I,1)))=N(I+5)
220 NEXT I
230 CLS:P=1
240 GOSUB 500
250 K=K+10
500 FOR I=1 TO 5
510 J=2*INT(N(I+K)/8)+1
520 F=2*(N(I+K)-8*INT(N(I+K)/8))+1
530 R$=MID$(V$,F,2)
540 PRINT@7*I-4,K;I
550 PRINT@7*I-5,K+2;R$;" ";MID$(C$,J,2)
560 PRINT@7*I-5,K+3;"-----"
570 NEXT I
580 IF P=1 THEN P=0 : RETURN
590 PRINT@2,16;"FAIRE 0 - RETURN -";:
  INPUTR$:IF K= 10 THEN EXPLODE:GOTO 70
600 RETURN

```

```

  1      2      3      4      5
  9 :CO  DA:PI  8 :TR  10:PI  VA:CO
  ---

```

```

  1      2      3      4      5
  VA:TR  RD:TR  DA:CA  RD:CA  AS:CA
  ---

```

```

FAIRE 0 - RETURN -?

```

Le jeu de saute- mouton

REFERENCES :

Difficulté de programmation : facile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : facile.

LE JEU :

Bien que l'idée en soit très simple, il ne sera pas forcément évident de découvrir une stratégie convenable. On dispose de 3 moutons rouges et de 3 moutons bleus, représentés par des pavés de couleur, qui occupent 8 cases ; les moutons peuvent se déplacer d'une case à une case libre. Plusieurs règles pourront intervenir, selon votre propre estimation de l'intérêt de telle ou telle : soit l'on contraint les moutons à se placer sur la première case disponible, en sautant éventuellement par-dessus les moutons adverses, soit on leur laisse choisir leur case d'arrivée, ce qui peut rendre les parties interminables, à vous de choisir.

LE PROGRAMME :

La boucle I (50-80) initialise une table A (I), qui contient les positions des moutons. La boucle I (100-150) dessine le terrain de jeu et les moutons. Les lignes 170 et 180 s'occupent du texte et des entrées. La ligne 190 contrôle la validité du coup. Le changement de joueur est réalisé en ligne 220.

POUR JOUER :

On commence l'exécution du programme par RUN. On obtient alors un affichage JOUEUR 1, qui indique le numéro du joueur, suivi de la question DEBUT, FIN.

On répond en indiquant le numéro du pion à déplacer suivi de RETURN, puis le numéro de la case d'arrivée, suivi de RETURN.

Le déplacement est alors réalisé ; puis le deuxième joueur procède de la même façon. Le gagnant est en principe celui qui réussit à placer ses trois moutons côte à côte sur les emplacements initialement occupés par l'adversaire.

EXTENSIONS POSSIBLES :

On commencera par rajouter un contrôle du joueur gagnant.

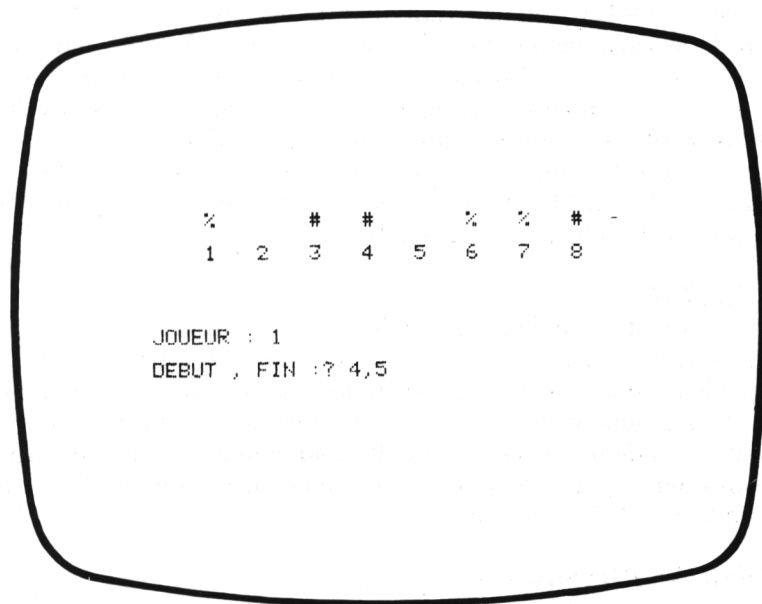
On peut également améliorer la présentation en dessinant de vrais moutons, rendre le jeu plus attractif en les numérotant, et en imposant un certain ordre à l'arrivée.

Enfin, on pourra définir une stratégie pour l'ordinateur, en figeant les règles.

```

10 REM LE JEU DE SAUTE-MOUTON
20 CLS
30 DIM A(8)
40 A(1)=0:A(8)=0
50 FOR I=1 TO 3
60 A(I+1)=1
70 A(I+4)=-1
80 NEXT I
90 H=1
100 FOR I=1 TO 8
110 PRINT @4*I+1,8;I
120 PLOT4*I+1,6,CHR$(7)+" ":ZAP
130 IF A(I)=1 THEN PLOT4*I+1,6,CHR$(1)+CHR$(255)
140 IF A(I)=-1 THEN PLOT4*I+1,6,CHR$(6)+CHR$(255)
150 NEXT I
160 J=1-(H=-1)
170 PLOT1,13,CHR$(6+5*(J=2)):PRINT@2,13;"JOUEUR :";J:PING
180 PRINT:INPUT"DEBUT , FIN :";D,F
190 IF A(D)<>H OR A(F)<>0 THEN 170
200 A(D)=0
210 A(F)=H
220 H=-H
230 GOTO 100

```



Lettres

REFERENCES :

Difficulté de programmation : facile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :

La règle est fortement inspirée d'un jeu de lettres très connu en France, pratiqué notamment dans une émission télévisée (les chiffres et les lettres). L'ORIC-ATMOS choisit au hasard un certain nombre de symboles (voyelles et consonnes) dans l'alphabet latin, et chacun des joueurs devra indiquer la longueur du mot qu'il a réussi à obtenir (l'appareil ne vérifiera pas si le mot est correct !).

Le score total de chaque joueur est affiché ensuite.

LE PROGRAMME :

La boucle I (90-120) choisit dans l'alphabet entre 7 et 12 lettres, en fonction de la variable aléatoire L, tirée en ligne 60. Pour éviter d'avoir trop de consonnes, on utilise en ligne 110 une petite astuce, qui consiste à refaire au hasard un tirage dans la table A\$, contenant des voyelles (on peut en avoir plus ou moins en utilisant d'autres RND).

La boucle 130-180 demande à chacun des joueurs la longueur de son mot. Les lignes 190 et 200 comparent les résultats et totalisent le score de chaque joueur, affiché par la ligne 210.

POUR JOUER :

On lance l'exécution du jeu avec RUN.

Les lettres sont alors affichées, puis la question JOUEUR 1 LONGUEUR à laquelle le premier joueur répond en donnant la longueur du mot qu'il a obtenu (l'autre joueur devra toutefois vérifier si le mot est correct) ; puis le joueur 2 fera de même et le score de chaque joueur sera alors affiché.

Pour continuer la partie, il suffira d'appuyer sur les touches RETURN.

On devra faire CTRL et C pour finir.

EXTENSIONS POSSIBLES :

On pourra bien entendu songer à introduire une temporisation avant de demander leurs réponses aux joueurs.

On pourra aussi entrer non point la longueur du mot, mais le mot lui-même, l'adversaire devant alors répondre à la question CORRECT OU NON ? On peut aussi étendre le jeu à plus de deux joueurs. Et pourquoi ne pas essayer de définir un jeu contre l'ORIC-ATMOS, auquel on aurait auparavant entré une liste de mots assez longue, dont il afficherait les lettres dans le désordre, avec éventuellement quelques lettres de complément choisies au hasard.


```

10 REM LETTRES
20 S=0:J=0
30 CLS
40 PAPER 6:INK 4
50 FOR I=1 TO 3:PLOT 0,I+5,CHR$(19):PLOT 1,I+5,CHR$(1):NEXT I
60 L=INT(RND(1)*6+7)
70 A$="AEIOUEA"
80 PRINT@2,6;"LETTRES : "
90 FOR I=1 TO L
100 PRINT@8+2*I,6;CHR$(INT(RND(1)*26+65))
110 IF RND(1)<0.4 THEN PRINT@8+2*I,6;MID$(A$,
    INT(RND(1)*7+1),1)
120 NEXT I
130 I=1
140 PRINT@2,10;"JOUEUR (";I;":PING:INPUT" - LONGUEUR :";L
150 PRINT@27,10;"
160 IF I=1 THEN P=L
170 I=I+1
180 IF I<3 THEN 140
190 S=S+(P AND P>L)
200 J=J+(L AND L>P)
210 PRINT@2,13;"SCORES - JOUEUR 1:(";S;") - JOUEUR 2:(";J
220 PRINT:INPUT"TAPER 0 - RETURN - POUR CONTINUER";R$
230 GOTO 50

```

LETTRES U O U E O B E C E

JOUEUR : 2 - LONGUEUR : ?

SCORES - JOUEUR 1: 0 - JOUEUR 2: 5

TAPER 0 - RETURN - POUR CONTINUER?

Machinamot

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu de hasard (sur des mots).

Difficulté du jeu : aucune.

LE JEU :

Ce très court programme permet à l'ORIC-ATMOS de faire la démonstration de ses capacités à manipuler des chaînes de caractères : on part simplement d'un préfixe et d'un suffixe d'origine grecque, que l'on va assembler au hasard ; on tombera bien sûr, de temps à autre, sur des mots absurdes ou même légèrement équivoques.

Un nouveau mot est confectionné à chaque essai, mais il est possible d'obtenir plusieurs fois le même mot.

LE PROGRAMME :

Deux variables A\$ et B\$ sont dimensionnées à 15 en lignes 50 et 60 : l'une contiendra les préfixes, l'autre les suffixes (si l'on désire rajouter des possibilités, il suffira de modifier cette limitation à 15).

La boucle I (70-90) lit les lignes DATA (160-180) concernant les préfixes suivis des suffixes.

La ligne 100 effectue le choix d'un préfixe au hasard.

La ligne 110 effectue le choix d'un suffixe au hasard.

Le mot obtenu en ligne 120 est affiché par la ligne 130, puis on recommence (ligne 150).

POUR JOUER :

Lancer l'exécution du programme par RUN. Un premier mot sera immédiatement affiché, puis un nouveau mot sera affiché à la suite du précédent.

Tapir CTRL et C pour sortir du programme.

EXTENSIONS POSSIBLES :

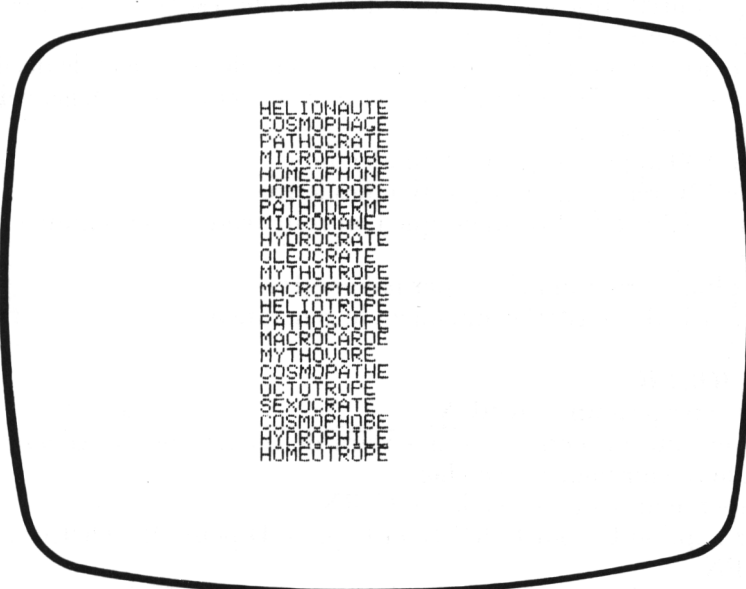
On pourra très facilement augmenter la liste des mots disponibles.

On pourra aussi utiliser un procédé analogue pour faire confectionner des phrases à l'ordinateur, lui faire inventer des poèmes, etc. On pourra aussi transformer le programme initial en un jeu de société qui consisterait, par exemple, à prendre des paris sur le prochain mot qui va apparaître. Et confier, éventuellement, la gestion des paris à l'ORIC.

```

10 REM MACHINAMOT
20 PAPER 1
30 INK 3
40 CLS
50 DIM A$(15)
60 DIM B$(15)
70 FOR I=1 TO 15
80 READ A$(I),B$(I)
90 NEXT I
100 I=INT(RND(1)*15+1)
110 J=INT(RND(1)*15+1)
120 C#=A$(I)+B$(J)
130 PRINT C#
140 PING:WAIT 250
150 GOTO 100
160 DATA ANDRO,CRATE,COSMO,PHILE,MICRO,PHOBE,
HELIO,TROPE,SEXO,GYNE
170 DATA STEREO,MANE,PALEO,NAUTE,MACRO,PHAGE,
HYDRO,FUGE,STENO,PATHE
180 DATA PATHO,CARDE,OLEO,DERME,MYTHO,SCOPE,
HOMEQ,VORE,OCTO,PHONE

```



HELIONAUTE
COSMOPHAGE
PATHOCRATE
MICROPHOBE
HOMEOPHONE
HOMEOTROPE
PATHODERME
MICROMANE
HYDROCRATE
OLEOCRATE
MYTHOTROPE
MACROPHOBE
HELIOTROPE
PATHOSCOPE
MACROCARDE
MYTHOVORE
COSMOPATHE
OCTOTROPE
SEXOCRATE
COSMOPHOBE
HYDROPHILE
HOMEOTROPE

Espion

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu de réflexion.

Difficulté du jeu : facile.

LE JEU :

Le programme peut servir simultanément de codeur et de décodeur de message secret, suivant une méthode très facile à mettre en œuvre sur ce type d'appareil.

Un petit générateur aléatoire situé en fin de programme est utilisé pour modifier de façon arbitraire chaque lettre du message ; on devra bien surveiller à utiliser toujours le même nombre (noyau) pour commencer le codage ou le décodage de deux messages identiques.

LE PROGRAMME :

Le noyau initial (qui sera un entier quelconque) est entré en ligne 30 ; le message est entré à la ligne 50.

La ligne 60 permet de choisir entre les options de codage et de décodage.

La boucle I (80-150) est utilisée soit pour effectuer le codage, soit pour effectuer le décodage.

La ligne 100 appelle le générateur aléatoire.

La ligne 110 réalise la transposition des lettres.

Les lignes 120 et 130 vérifient que l'on reste bien dans les limites autorisées.

Les résultats seront imprimés par la ligne 160.

Les lignes 1000 et 1030 contiennent le générateur.

POUR JOUER :

Lancer le programme par RUN.

Entrer ensuite un entier N qui servira de noyau pour le générateur (il faudra le retenir pour le décodage).

Puis taper le message, suivi de RETURN.

Choisir l'option C pour CODER, et l'option D pour DECODER, et taper RETURN.

Le message codé ou décodé, selon le cas, sera alors affiché.

EXTENSIONS POSSIBLES :

On pourra étendre le codage à d'autres signes que les lettres de l'alphabet, utiliser éventuellement plusieurs générateurs, introduire dans le message un codage du ou des nombres ayant servi de générateurs, etc.

```

10 REM ESPION
20 CLS
30 INPUT"DONNEZ VOTRE CODE :";N
40 A=1/N
50 PRINT:INPUT"VOTRE MESSAGE :";A$
60 PRINT:INPUT"VOULEZ-VOUS (C)ODER OU (D)ECODER :";R$
70 B$=""
80 FOR I=1 TO LEN(A$)
90 T=ASC(MID$(A$,I,1))
100 GOSUB 1000
110 T=T-A*((R$="C")-(R$="D"))
120 IF T>90 THEN T=T-26
130 IF T<65 THEN T=T+26
140 B#=B#+CHR$(T)
150 NEXT I
160 PRINT:PRINT"RESULTAT :";B$:EXPLODE:END
1000 A=(A+3.14159)^8
1010 A=A-INT(A)
1020 A=INT(26*A+1)
1030 RETURN

```

```

DONNEZ VOTRE CODE :? 1234
VOTRE MESSAGE :? BONJOUR MADAME
VOULEZ-VOUS (C)ODER OU (D)ECODER :? C
RESULTAT :DYUKMUP;KBBBKF

```

```

DONNEZ VOTRE CODE :? 1234
VOTRE MESSAGE :? DYUKMUP;KBBBKF
VOULEZ-VOUS (C)ODER OU (D)ECODER :? D
RESULTAT :BONJOURMADAME

```

Accrochez les wagons

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : solitaire.

Difficulté du jeu : facile.

LE JEU :

Il faut essayer d'accrocher, le plus rapidement possible, la locomotive avec son wagon, qui apparaissent tous deux en début de partie, en deux endroits plus ou moins éloignés de l'écran. On devra pour cela déplacer la locomotive en indiquant la suite des mouvements à effectuer (sous forme codée : H pour une case vers le haut ; B pour une case vers le bas ; D pour une case vers la droite et G pour une case vers la gauche).

On tapera, par exemple : BBGDHHH.

LE PROGRAMME :

Les lignes 40 et 50 dessinent la locomotive, la ligne 60 le wagon, et les roues sont en ligne 70.

La position initiale de la locomotive est définie par les lignes 90 et 100, celle du wagon par les lignes 110 et 120.

La ligne 130 évite qu'ils soient initialement trop proches l'un de l'autre.

Les lignes 150 et 160 réalisent l'affichage du wagon et les lignes 170, 180 et 190 celui de la locomotive.

Le test de fin est en ligne 200.

Les mouvements à effectuer sont entrés dans une variable alphanumérique R\$, en ligne 210, et sont décodés dans une boucle I (240-270) pour être exécutés en ligne 280.

POUR JOUER :

RUN permet d'afficher les positions de départ de la locomotive et du wagon.

On tape alors la série des déplacements à effectuer, sous la forme DDDHHHBHGC, suivie de RETURN.

La locomotive sera alors affichée dans sa nouvelle position, et l'on recommence, jusqu'à ce qu'elle soit accrochée au wagon.

On obtiendra alors l'affichage du score.

EXTENSIONS POSSIBLES :

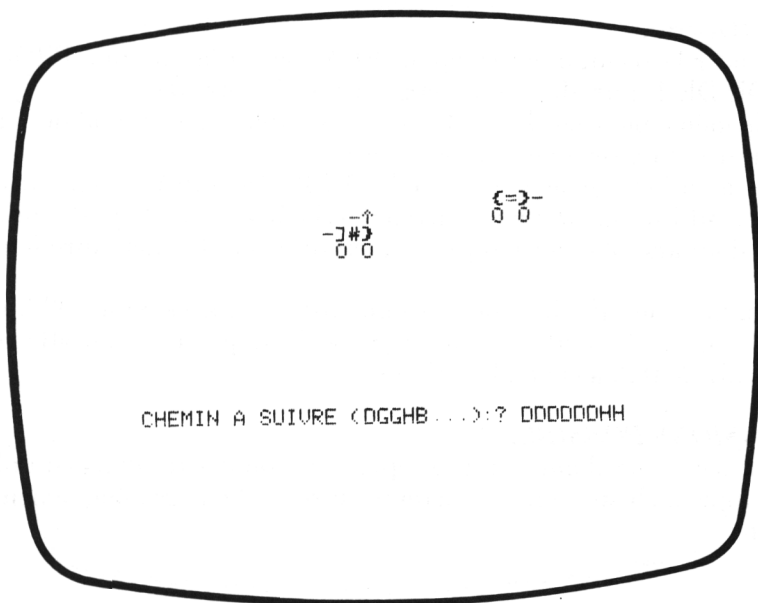
Il sera intéressant de prévoir la perte de la partie si, par accident, au cours du déplacement, la locomotive vient en collision avec le wagon.

On pourra aussi prévoir plusieurs wagons, et un ordre d'accrochage, jouer à plusieurs avec plusieurs locomotives, dessiner les voies ferrées, introduire des obstacles à contourner obligatoirement d'un certain côté, etc.

```

10 REM ACCROCHEZ LES WAGONS
20 PAPER 6
30 INK 4
40 D$=CHR$(1)+" ^"
50 A$=CHR$(1)+"-J#"
60 B$=CHR$(3)+"(=)-"
70 C$="0 0"
80 N=0
90 LY=INT(RND(1)*33+3)
100 LX=INT(RND(1)*19+2)
110 WY=INT(RND(1)*33+3)
120 WX=INT(RND(1)*19+2)
130 IF ABS(LX-WX)<4 THEN 90
140 CLS
150 PLOTWY,WX,B$
160 PLOTWY+1,WX+1,C$
170 PLOTLY,LX-1,D$
180 PLOTLY-1,LX,A$
190 PLOTLY+1,LX+1,C$
200 IF LX=WX AND WY=LY-5 THEN 300
210 PRINT@2,22;"CHEMIN A SUIVRE (DGGHB...):";INPUT R$
220 X=0:Y=0
230 N=N+1
240 FOR I=1 TO LEN(R$)
250 X=X-(MID$(R$,I,1)="B" AND LX<22)+(MID$(R$,I,1)="H"
AND LX>1)
260 Y=Y-(MID$(R$,I,1)="D" AND LY<37)+(MID$(R$,I,1)="G"
AND LY>2)
270 NEXT I
280 LX=LX+X:LY=LY+Y
290 GOTO 140
300 PRINT@2,22;"BRAVO ; TROUVE EN ";N;" COUPS"
310 EXPLODE

```



Tirage d'un dé truqué

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion.

Difficulté du jeu : moyenne.

LE JEU :

Il reprend une idée d'un programme précédent (DE TRUQUE) dans une application statistique : on fabrique un dé volontairement truqué, que l'on va lancer un certain nombre de fois, et l'on dresse un tableau des résultats.

LE PROGRAMME :

Les probabilités désirées pour chaque tirage sont introduites par une boucle I (70-140) avec toutes les vérifications de validité nécessaires. Le nombre de tirages demandés est introduit en ligne 160.

La boucle J (170-250) comptabilise les tirages.

La boucle I (190-210) assure la conformité de chaque tirage avec les probabilités introduites en début de jeu.

Les résultats sont comptabilisés en ligne 220, et affichés grâce aux lignes 230 et 240.

POUR JOUER :

Commencer l'exécution en frappant RUN, puis à la question PROBABLITE DU DE 1, introduire un nombre compris entre 0 et 1.

Faire de même pour les dés 2, 3, etc., sans oublier que le total des probabilités introduites devra être proche de 1.

Ensuite, à l'affichage de la question NOMBRE DE TIRAGE, on introduira un entier suffisamment grand, suivi de RETURN. L'affichage des résultats mettra d'autant plus de temps à apparaître que l'entier introduit sera grand.

Les résultats sont affichés sous les numéros des tirages : ainsi 32.3 apparaissant sous le 1 signifie que le tirage de 1, auquel on avait affecté une probabilité de 0.3, a été réalisé 32 fois.

EXTENSIONS POSSIBLES :

On peut envisager d'améliorer la représentation des résultats en utilisant par exemple un histogramme, ou toute autre méthode graphiquement plus parlante.


```

10 REM DE TRUQUE
20 CLS
30 PAPER 6:INK 4
40 FOR I=1 TO 5:PLOT0,4+I,CHR$(18):PLOT0,13+I,CHR$(19):NEXT I
50 DIM E(6)
60 P=0
70 FOR I=1 TO 6
80 PRINT@2,3;" "
90 PRINT@2,3;"PROBABILITE DU DE ";I);INPUT E(I)
100 P=P+E(I)
110 IF E(I)<0 OR E(I)>1 OR P>1.1 THEN 300
120 PRINT@6*I-4,6;"DE ";I
130 PRINT@6*I-4,8;E(I)
140 NEXT I
150 IF P<0.9 THEN RUN
160 PRINT@2,11;"NOMBRE DE TIRAGES :";INPUT P:PRINT CHR$(17)
170 FOR J=1 TO P
180 N=INT(RND(1)*6+1)
190 FOR I=1 TO 6
200 IF N=I AND RND(1)>E(I)-INT(E(I)) THEN 180
210 NEXT I
220 E(N)=E(N)+1
230 PRINT@6*N-2,15;N
240 PRINT@6*N-3,17;E(N)
250 NEXT J
260 EXPLODE:END
300 PRINT@2,4;"IMPOSSIBLE :0<= P <=1"
310 P=P-E(I)
320 WAIT 200
330 PRINT@2,4;"
340 GOTO 90

```

```

PROBABILITE DU DE 6 ? 0.3
DE 1 DE 2 DE 3 DE 4 DE 5 DE 6
.1 .05 .05 .3 .2 .3
NOMBRE DE TIRAGES :? 200
1 2 3 4 5 6
14.1 13.05 11.05 69.3 31.2 63.3

```

Tour de cartes

REFERENCES :

Difficulté de programmation : difficile.
Catégorie de jeu : jeu de prestidigitation.
Difficulté du jeu : facile.

LE JEU :

C'est un tour assez facile mais, ici, les cartes sont manipulées sous vos yeux par l'ORIC-ATMOS qui tient le rôle du prestidigitateur ; 21 cartes, prises au hasard dans un paquet de 52, sont affichées, et le joueur devra retenir mentalement l'une des cartes présentées. Puis, il devra indiquer à l'ORIC-ATMOS dans quelle colonne (1^{re}, 2^e ou 3^e à partir de la gauche) se trouve la carte choisie ; les cartes sont alors redistribuées et le joueur indique de nouveau la colonne correspondant à sa carte ; ceci est fait encore une fois et, au 4^e affichage, l'ORIC-ATMOS est en mesure de deviner la carte choisie.

LE PROGRAMME :

La boucle I (60-120) permet de choisir, au hasard, 21 cartes sur 52, et la boucle emboîtée J (80-100) permet de s'assurer que l'on ne réalise pas deux fois le même tirage. La ligne 140 contient les couleurs des cartes ainsi que les noms des cartes figures (VA, DA, RO, AS), dans une variable alphanumérique B\$.

La boucle I (150-240) réalise l'affichage du jeu.

La ligne 160 décode la couleur de la carte.

La ligne 170 choisit la position d'affichage.

La ligne 180 décode la hauteur.

L'affichage proprement dit est effectué de la ligne 190 à la ligne 230.

La ligne 260 permet au joueur d'entrer le numéro de la colonne où se trouve sa carte. La boucle J (300-320) réalise le brouillage, et les boucles emboîtées J (350-390) et I (360-380) fabriquent la nouvelle variable A\$ à utiliser pour le prochain affichage.

POUR JOUER :

On lance le programme par RUN.

On obtient un premier affichage de 21 cartes réparties sur 3 colonnes.

On indique alors le numéro (1, 2 ou 3) de la colonne où se trouve la carte choisie, suivi de RETURN.

On obtient un 2^e affichage et l'on fait de même ; puis un 3^e affichage et l'on fait de même.

Au 4^e affichage, la carte choisie est automatiquement affichée.

On sortira du programme en tapant CTRL et C à la place du numéro de la colonne.

EXTENSIONS POSSIBLES :

On peut envisager une amélioration de l'affichage en écrivant les noms des cartes en entier, ou même, en dessinant de véritables cartes, en couleur.

On pourra aussi essayer de programmer d'autres tours, utilisant le même principe de base, mais avec davantage de cartes, etc.

```
10 REM TOUR DE CARTES
20 CLS:PAPER 5:INK 0
30 FOR I=1 TO 9:PLOT0,I,CHR$(19):PLOT12,I,CHR$(18):PLOT26,I,CHR$(17)
40 NEXT I
50 A$="":PRINT CHR$(17)
60 FOR I=1 TO 21
70 B$=CHR$(INT(RND(1)*52+38))
80 FOR J=1 TO LEN(A$)
90 IF B$=MID$(A$,J,1) THEN 70
100 NEXT J
110 A$=A$+B$
120 NEXT I
130 H=3:P=1
140 B$="TRCACOPIVADARORS"
150 FOR I=1 TO LEN(A$)*P-11*(P=0)
160 S=INT((ASC(MID$(A$,I,1))-38)/13)
170 J=7*INT((I-1)/7)
180 T=ASC(MID$(A$,I,1))-37-13*S
190 IF P=0 THEN CLS:PRINT@2,0;"VOTRE CARTE EST : "
200 PRINT@((J+1)*P+INT((2*J+3)/2)-16*(P=0),(I-J)*P);:IF P=1 THEN ZAP
210 IF T<=9 THEN PRINT STR$(T+1)
220 IF T>9 THEN PRINT MID$(B$,ABS(2*T-11),2)
230 PRINT@((J+5)*P+1+INT((2*J+3)/2)-21*(P=0),(I-J)*P);MID$(B$,2*S+1,2)
240 NEXT I
250 IF P=0 THEN EXPLODE:END
260 PRINT@2,16;"QUELLE COLONNE :";:PING:INPUT N
270 CLS:FOR I=1 TO 9:PLOT0,I,CHR$(19):PLOT12,I,CHR$(18):
PLOT26,I,CHR$(17):NEXT I
280 H=H-1
290 R$="20120":B$=""
300 FOR J=0 TO 2
310 B$=B$+MID$(R$,7*VAL(MID$(R$,N+J,1))+1,7)
320 NEXT J
330 IF H=0 THEN A$=B$:P=0:GOTO 140
340 A$=""
350 FOR J=0 TO 2
360 FOR I=1 TO 7
370 A$=A$+MID$(B$,3*I-2+J,1)
380 NEXT I
390 NEXT J
400 GOTO 140
```

5	CO	DA	PI	AS	CA
10	PI	PI	PI	VA	PI
3	PI	CA	AS	CO	AS 4
6	CO	PI	CO	AS	0
0	TR	AS	TR	RO	CO

QUELLE COLONNE : ? 2

Le jeu de Marienbad

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : logique (2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

Un jeu rendu très célèbre grâce à un film qui le popularisa, présenté toutefois ici dans une version plus conforme à ses origines, à savoir le jeu de NIM.

En effet, il est possible, en début de partie, de choisir le nombre d'allumettes que l'on veut mettre dans chaque tas.

Chaque joueur joue à son tour, et celui qui retire la dernière allumette a perdu.

LE PROGRAMME :

La ligne 60 dimensionne une variable A (I) qui contiendra le nombre d'allumettes de chaque tas.

La boucle I (70-100) permet d'entrer ce nombre.

La boucle I (130-160) imprime les tas.

La ligne 170 assure l'affichage du joueur.

La ligne 190, ainsi que la ligne 200, contrôlent si le coup est autorisé.

La ligne 270 contrôle la fin de la partie.

POUR JOUER :

On lance le programme avec RUN, puis on entre le nombre de tas (entre 1 et 8), suivi de RETURN.

On entrera ensuite le nombre d'allumettes dans chaque tas (faire à chaque entrée N suivi de RETURN). On obtiendra alors l'affichage des différents tas, avec le numéro du joueur concerné (1 ou 2), ainsi que le message DANS QUEL TAS ; on répond en tapant le numéro du tas dans lequel on désire faire un retrait suivi de RETURN. On obtient ensuite la question QUANTITE A ENLEVER, et l'on répond en indiquant le nombre d'allumettes que l'on désire enlever du tas concerné.

Puis ce sera au tour du deuxième joueur, et l'on jouera alternativement jusqu'à ce qu'il n'y ait plus d'allumettes disponibles.

En fin de partie, on obtiendra l'affichage du numéro du joueur gagnant.

EXTENSIONS POSSIBLES :

On peut désigner chaque joueur par son nom, faire un graphisme plus élaboré, écrire un programme permettant à l'ORIC de servir de partenaire, etc.

```

10 REM LE JEU DE MARIENBAD
20 CLS:PAPER 4:INK 6
30 INPUT"NOMBRE DE TAS ( <=7 ) :";N
40 IF N>=8 THEN RUN
50 S=0
60 DIM A(N)
70 FOR I=1 TO N
80 PRINT:PRINT"TAS ";I:;INPUT" - NOMBRE D'ALLUMETTES :";A(I)
90 S=S+A(I)
100 NEXT I
110 H=1
120 CLS:PAPER 2:INK 4
130 FOR I=1 TO N
140 PRINT@5*I+2,4;A(I)
150 PRINT@2,6;"TAS":PRINT@5*I+2,6;I
160 NEXT I
170 PRINT@2,10;"JOUEUR ";H:PING
180 PRINT:INPUT"DANS QUEL TAS :";T
190 IF T>N THEN 120
200 IF A(T)=0 THEN 120
210 PRINT@2,14;"QUANTITE A ENLEVER :";;INPUT O
220 IF O>A(T) OR O=0 THEN 200
230 A(T)=A(T)-O
240 S=S-O
250 H=H+1
260 IF H=3 THEN H=1
270 IF S=0 THEN PRINT:PRINT"LE JOUEUR ";H;"
A GAGNE":EXPLODE:END
280 GOTO 120

```

```

          1   3   5   7
TAS  1   2   3   4

JOUEUR  1
DANS QUEL TAS : ? 2
QUANTITE A ENLEVER : ? 1

```

L'indésirable

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : hasard et réflexion (1 joueur).

Difficulté du jeu : moyenne.

LE JEU :

Vous êtes le propriétaire d'un immeuble de 16 pièces vides, et vous n'arrivez pas à mettre la main sur votre locataire indésirable. Vous visitez les pièces au hasard, mais, si vous visitez la pièce voisine, votre locataire, prévenu par le bruit, va déménager. Il peut même se réinstaller dans une pièce que vous avez déjà visitée, et marquée comme vide. Il vous faudra arriver à le déloger en moins de 9 coups pour gagner.

LE PROGRAMME :

Une table B est dimensionnée à 16, en ligne 50, pour recevoir les numéros des cases déjà visitées.

La boucle I (60-100) dessine le terrain de jeu.

La position initiale du locataire est tirée au hasard en ligne 110.

Le numéro de la case à visiter est donné en ligne 120.

Les lignes 130 à 220 contrôlent les diverses réponses possibles.

La boucle I (170-200) dessine les cases visitées.

La ligne 230 dessine le locataire, effacé en ligne 260.

Les lignes 270 et 280 assurent son déplacement.

POUR JOUER :

On lance l'exécution par RUN.

A la question CASE VISITEE, on répond en tapant une lettre, de A à P. Si on a trouvé le locataire, celui-ci est affiché, et la partie est terminée ; sinon, deux cas peuvent se produire : soit l'on est suffisamment loin, et le locataire ne bouge pas, soit l'on est à une case, il apparaît un court instant, et il se recache, aléatoirement, dans une case voisine. Les cases déjà visitées se colorent en rouge, et vous ne pouvez pas y revenir. Si, au bout de 9 coups, vous n'avez pas trouvé, la position du locataire s'affiche, et vous avez perdu.

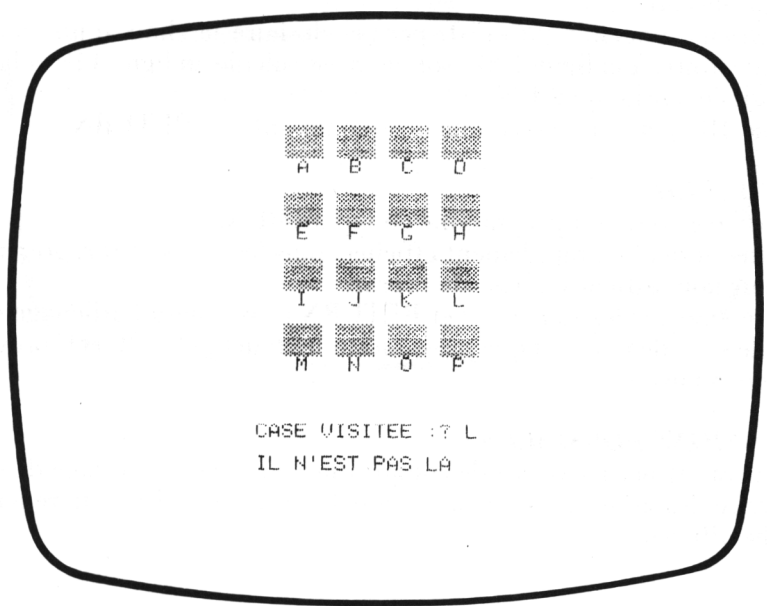
EXTENSIONS POSSIBLES :

On pourra, bien sûr, jouer sur plus de 16 cases, utiliser des caractères graphiques pour représenter le locataire et le propriétaire, utiliser un algorithme plus subtil pour le déplacement du locataire, lui interdire éventuellement de revenir dans des cases déjà visitées, etc.

```

10 REM L'INDESIRABLE
20 CLS:PAPER 6:INK 4
30 B# = CHR$(255)+CHR$(255)+CHR$(255)
40 FOR I=1 TO 16 STEP 4:PLOT1,I+1,CHR$(1):PLOT1,I+2,CHR$(1):NEXT I
50 DIM B(16)
60 FOR I=1 TO 16
70 PRINT@4+4*(I-1-4*INT((I-1)/4)),2+4*INT((I-1)/4);B#
80 PRINT@4+4*(I-1-4*INT((I-1)/4)),3+4*INT((I-1)/4);B#
90 PRINT@5+4*(I-1-4*INT((I-1)/4)),4+4*INT((I-1)/4);CHR$(I+64)
100 NEXT I
110 X=INT(RND(1)*16+1):S=1:K=0
120 PRINT@2,20;"CASE VISITEE ";:INPUT C#:C=ASC(C#)-64
130 IF B(C)=1 THEN PRINT @2,22;"DEJA VISITEE":ZAP:ZAP:ZAP:GOTO 120
140 IF C=X THEN PRINT@2,22;"VOUS L'AVEZ EU EN ";S;" COUPS":PING:K=1:
    GOTO 230
150 IF X-5=CORX-4=CORX-3=CORX-1=CORX+1=CORX+3=CORX+4=CORX+5=C THEN 230
160 PRINT@2,22;"IL N'EST PAS LA":PING:WAIT 150
170 S=S+1:B(C)=1:FOR I=1 TO 16
180 IF B(I)=0 THEN 200
190 PLOT3+4*(I-1-4*INT((I-1)/4)),2+4*INT((I-1)/4),CHR$(0)+B#+CHR$(1)
200 NEXT I
210 IF S=9 THEN PRINT@2,22;"VOUS AVEZ PERDU.C'ETAIT ";CHR$(64+X):K=1:
    GOTO 230
220 PRINT@2,22;"                               ":GOTO 120
230 PRINT@5+4*(X-1-4*INT((X-1)/4)),1+4*INT((X-1)/4);"%
240 SHOOT:WAIT 250:SHOOT
250 IF K=1 THEN EXPLODE:WAIT 200:END
260 PRINT@5+4*(X-1-4*INT((X-1)/4)),1+4*INT((X-1)/4);" "
270 Z=INT(RND(1)*10+1)-5:IF ABS(Z)=2 THEN 270
280 X=X+Z:IF X>16 OR X<1 THEN X=X-Z:GOTO 270
290 GOTO 170

```



Compose

REFERENCES :

Difficulté de programmation : facile.
Catégorie de jeu : composition musicale.
Difficulté du jeu : aucune, a priori.

LE JEU :

Il s'agit, en fait, d'un programme capable de composer, de façon très élémentaire, de la musique, et ceci d'une façon complètement aléatoire. L'ordinateur va, dans un premier temps, dessiner une portée, puis il affiche une succession de notes (allant du DO grave au SI aigu) et donne leur nom. Une nouvelle série de notes sera ensuite affichée, et ainsi de suite.

LE PROGRAMME :

La boucle I (50-70) dessine la portée. La variable N\$, en ligne 80, contient les noms des notes.

La deuxième boucle I (90-170) permet de faire le choix d'une note de façon aléatoire, à la ligne 100 ; son nom est calculé en ligne 110 et la note est dessinée par la ligne 120.

La ligne 180 permet de recommencer en tapant 0 et RETURN.

POUR JOUER :

Lancer l'exécution du programme en tapant RUN.

La portée sera alors rapidement affichée, puis les notes sont représentées, avec leur nom affiché en bas de l'écran.

Une pression sur les touches 0 et RETURN provoquera l'affichage d'une nouvelle série de notes, et une pression sur les touches CTRL et C provoque la fin de la partie.

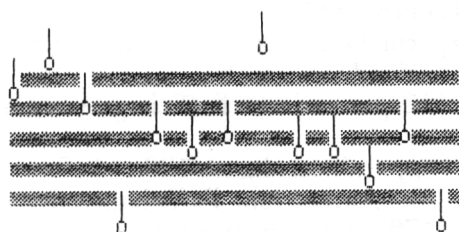
EXTENSIONS POSSIBLES :

On pourra affiner les règles de composition, afficher davantage de notes, introduire des dièses et des bémols, améliorer le graphisme, faire jouer la mélodie affichée.


```

10 REM COMPOSE
20 CLS:R$="1210080605030112100806050301"
30 PAPER 3
40 INK 4
50 FOR I=1 TO 5
60 PLOT0,3+2*I,CHR$(17)
70 NEXT I
80 N$="SI LA SOLFA MI RE DO "
90 FOR I=1 TO 13
100 X=INT(RND(1)*14+1)
110 D$=MID$(N$,3*(X+7*(X>7))-2,3)
120 PRINT@3*I-1,X-1;"|":PRINT@3*I-1,X;"|":PRINT@3*I-1,X+1;"0"
130 PRINT@3*I-1,18;D$
140 MUSIC 1,3-(X<=7),VAL(MID$(R$,2*X-1,2)),0
150 PLAY1,0,4,2000
160 WAIT 150:PLAY0,0,0,0
170 NEXT I
180 PRINT@2,21;"TAPER 0 - RETURN - POUR CONTINUER";:INPUT P$
190 GOTO 20

```



MI SOLRE DO SI LA SI LA LA LA FA SI DO

TAPER 0 - RETURN - POUR CONTINUER?

Jour de la semaine

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : utilitaire.

Difficulté du jeu : facile.

LE JEU :

Le programme vous permet de retrouver, sans aucun calcul, le jour de la semaine correspondant à une date donnée.

La date sera entrée sous forme de trois nombres, par exemple 11, 3, 1956 pour le 11 mars 1956, et le résultat sera affiché en clair.

LE PROGRAMME :

Les lignes 10 à 70 contrôlent les entrées. La ligne 40 s'occupe de la fin de partie. La ligne 90 permet une correction pour les dates antérieures à 1582 (mise en service du calendrier actuel).

Les lignes 100 à 110 permettront l'affichage en clair du mois dans la réponse (sur 4 lettres).

La formule utilisée apparaît aux lignes 130 et 140.

Les lignes 150 et 160 permettront l'affichage en clair du jour dans la réponse. Les lignes 170 à 190 impriment les résultats.

POUR JOUER :

Lancer le programme par RUN, puis entrer le jour suivi de RETURN, le mois (entre 1 et 12) suivi de RETURN, et l'année (sous la forme 1845, par exemple) suivie de RETURN.

Le résultat est très rapidement affiché.

On tape 0 et RETURN et le programme revient sur la question JOUR.

Pour continuer, il suffit de retaper un jour, un mois, une année.

Pour terminer, il faut répondre 0 à la question JOUR.

EXTENSIONS POSSIBLES :

On pourra entrer les mois en clair et faire afficher le nom du jour en entier.

On peut envisager de faire afficher les dates et fêtes particulières (Noël, Pâques, les Rameaux, Pentecôte...).

On peut écrire un programme qui afficherait les jours fériés pour une année donnée, etc.

```

10 REM JOUR DE LA SEMAINE
20 CLS:PAPER 7:INK 0
30 PRINT:INPUT"JOUR.....":J
40 IF J=0 THEN END
50 PRINT:INPUT"MOIS.....":M
60 PRINT:INPUT"ANNEE.....":A
70 B=A
80 S=INT(A/100)-INT(A/400)
90 IF A<1582 THEN S=2
100 M$="JANVFEVRMARS AVRIL MAI JUINJUILAOUTSEPTOCTONOVEDECE"
110 M$=MID$(M$,4*M-3,4)
120 IF M<3 THEN M=M+12:A=A-1
130 S=J-1+INT(13*(M+1)/5)+INT(5*A/4)-S
140 S=S-7*INT(S/7)
150 J$="DIMANCHELUNDI MARDI MERCREDIJEUDI VENDREDISAMEDI "
160 J$=MID$(J$,8*S+1,8)
170 PLOT0,15,CHR$(17):PLOT 1,15,CHR$(0)
180 PRINT@2,15;"LE ";J;" ";CHR$(27);"L";M$:CHR$(27);"H";B;" ETAIT UN ";
190 PRINT CHR$(27);"W";J$:EXPLODE
200 PRINT:PRINT:INPUT"POUR CONTINUER , TAPER 0 - RETURN -":R$
210 RUN

```

JOUR.....? 11
MOIS.....? 6
ANNEE.....? 1984

LE 11 JUIN 1984 ETAIT UN LUNDI

POUR CONTINUER , TAPER 0 - RETURN -?

Le jeu de la vie

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu solitaire (2 versions).

Difficulté du jeu : facile.

LE JEU :

La version présentée ici respecte les règles posées à l'origine par le mathématicien J.H. CONWAY.

Une certaine population est définie au début du jeu soit par l'ORIC-ATMOS (version automatique) soit par vous-même (version manuelle). Cette population va évoluer sous vos yeux, en respectant certaines règles, très élémentaires de naissance, de vie et de mort. L'évolution se poursuivra jusqu'à ce que l'on décide d'arrêter le programme.

LE PROGRAMME :

Deux tables (B et C) sont utilisées en ligne 50, pour enregistrer l'état de la population. Les boucles F (230-300) et G (240-290) sont utilisées en début de jeu pour définir la population initiale (lignes 250 et 260). Le calcul de la nouvelle population est ensuite réalisé à l'aide d'une quadruple boucle : 80-210, 90-200, 110-160, 120-150. La ligne 130 réalise la gestion du nouvel état suivant les règles.

L'affichage de ce nouvel état est obtenu en réutilisant la double boucle d'affichage F (230-300) et G (240-290).

La ligne 310 permet de recommencer.

POUR JOUER :

Dans la version automatique, il suffit de lancer le programme par RUN et de regarder. Pour arrêter, il faudra taper CTRL et C. Dans la version manuelle, la configuration de départ sera entrée, après le lancement du programme par RUN, en tapant 0 si l'on ne veut rien mettre sur la ligne et la colonne indiquée, et 1 si l'on désire y mettre une cellule. Puis l'évolution se déroulera ensuite de façon automatique.

EXTENSIONS POSSIBLES :

Il sera facile d'augmenter les dimensions des tables B et C, en redéfinissant la variable A. On pourra aussi introduire la couleur, améliorer la précision du dessin en remplaçant les PRINT par un affichage haute résolution.

```

10 REM LE JEU DE LA VIE (AUTOMATIQUE)
20 CLS:PAPER 7:INK 0:PRINT CHR$(17)
30 INPUT"DIMENSION :";A
40 CLS:PAPER 3:INK 1
50 DIM B(A,A):DIM C(A,A)
60 I=0
70 GOTO 220
80 FOR F=1 TO A
90 FOR G=1 TO A
100 S=0
110 FOR I=-1 TO 1
120 FOR J=-1 TO 1
130 IF F+J=0 OR G+I=0 OR F+J=A+1 OR G+I=A+1 THEN 150
140 S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 S=2*S-B(F,G)
180 C(F,G)=0
190 IF S>=5 AND S<=7 THEN C(F,G)=1
200 NEXT G
210 NEXT F
220 CLS
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I<>0 THEN 270
260 IF RND(1)<0.5 THEN C(F,G)=1
270 IF C(F,G)=1 THEN PRINT@2*G,2*F;"*"
280 B(F,G)=C(F,G)
290 NEXT G
300 NEXT F

```

```

* * * *
* * * *      *
  * * *
    * * * * *
* * *      *
  * * *      *
* * * * * *
* * *      *

```

```

10 REM LE JEU DE LA VIE (MANUEL)
20 CLS:PAPER 7:INK 0:PRINT CHR$(17)
30 INPUT"DIMENSION :";A
40 CLS:PAPER 3:INK 1
50 DIM B(A,A):DIM C(A,A)
60 I=0
70 GOTO 220
80 FOR F=1 TO A
90 FOR G=1 TO A
100 S=0
110 FOR I=-1 TO 1
120 FOR J=-1 TO 1
130 IF F+J=0 OR G+I=0 OR F+J=A+1 OR G+I=A+1 THEN 150
140 S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 S=2*S-B(F,G)
180 C(F,G)=0
190 IF S>=5 AND S<=7 THEN C(F,G)=1
200 NEXT G
210 NEXT F
220 CLS
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I<>0 THEN 280
260 PRINT@2,20;"LIGNE ";F;";COLONNE ";G;" - TAPER 1 OU 0"
270 PRINT@2,21;:INPUT C(F,G)
280 IF C(F,G)=1 THEN PRINT@2*G,2*F;"*"
290 B(F,G)=C(F,G)
300 NEXT G
310 NEXT F
320 PING:GOTO 80

```

```

*      * * *
** *****
*****
*****
** *****
*      * * *

```

LIGNE 10 ;COLONNE 3 - TAPER 1 OU 0

Niveau 3

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Un dé	Tirage d'un dé, avec graphisme
Plus grand, plus petit	Un jeu de déduction intéressant, pour un joueur.
Ordre	Un peu d'astuce pour remettre en ordre, le plus vite possible.
Le pendu	Un jeu de lettres très classique (2 joueurs).
Belote	Pour simuler une donne, avec un jeu de 32 cartes.
Un taquin de calcul	Un bon entraînement (2 joueurs).
Pendu inversé	Cette fois, c'est l'ordinateur qui essaie de deviner le mot.
Nicomaque	Pour se perfectionner en calcul mental (1 ou plusieurs joueurs).
Course de chevaux	Assez visuel : un jeu de hasard, pour plusieurs joueurs.
A travers	Un jeu de stratégie pour 2 joueurs.
Mastermind	Avec des chiffres, mais très classique. Un seul joueur.
Idem	Un jeu de calcul mental, avec plusieurs niveaux, pour plusieurs joueurs.
Horloge	Ne jetez pas votre vieille télévision : transformez-la en horloge.
Reines sur l'échiquier	Un vieux problème, pour vous aider à réfléchir.
Répétitions	Retrouver une séquence de chiffres de plus en plus longue.
Mots	Une idée très simple, mais ce n'est pas si facile de trouver !
Permutations	Un jeu de réflexion, pas toujours évident.
Bridge	Mettez au point vos propres problèmes de bridge.
Hanoï	Un conseil : commencez petit !!!
Losanges	Pour vous aider à vous endormir...
Mathieu	Vous ne savez pas dessiner ? Laissez faire votre ordinateur.
Le loup et les agneaux	Un jeu de réflexion pour 2 joueurs.

Manipulation des affichages grand format Mode haute résolution

TRAITEMENT DES CHAINES DE CARACTERES : MID\$. LEFT\$.
RIGHT\$

Nous avons eu l'occasion de rencontrer, dès le niveau II, la fonction essentielle utilisée en Basic pour manipuler les chaînes : il s'agit de MID\$ (A\$, I, J) qui permet d'extraire, dans la chaîne A\$, une sous-chaîne de longueur J, commençant au I^{ème} caractère.

Ainsi si,

A\$ = "MONSIEUR"

On aura :

MID\$ (A\$, 2, 3) = "ONS"

Pour faciliter les manipulations, il est possible d'utiliser les deux fonctions LEFT\$ et RIGHT\$.

LEFT\$ (A\$, I) permet d'extraire les I premiers caractères à partir de la gauche du mot A\$.

Ainsi, LEFT\$ (A\$, 2) = "MO", avec l'exemple précédent (on pourrait d'ailleurs utiliser MID\$ (A\$, 1, 2) pour obtenir le même résultat.

RIGHT\$ (A\$, J) permet d'extraire les J derniers caractères (on part de la droite du mot A\$).

Ainsi RIGHT\$ (A\$, 3) = "EUR", toujours avec le même exemple.

Il est plus difficile de se passer de la fonction RIGHT\$, mais il serait possible de faire MID\$ (A\$, LEN (A\$) - 3, 3) pour obtenir le même résultat.

Le Basic de l'ORIC-ATMOS ne permet pas le remplacement direct d'une sous-chaîne par une affectation du type MID\$ (-, -) = " ". Il sera nécessaire, pour remplacer la I^{ème} lettre de la chaîne A\$, d'écrire :

A\$ = LEFT\$ (A\$, I, 1) + "nouvelle lettre" + RIGHT\$ (A\$, LEN (A\$) - I).

Des formules analogues pourront être utilisées pour remplacer dans A\$ une sous-chaîne par une autre, pas nécessairement de même longueur, ce qui permet une grande souplesse.

AFFICHAGE GRAND FORMAT

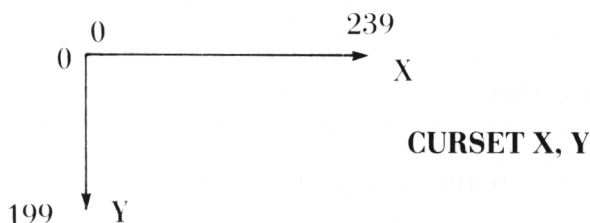
Le Basic de l'ORIC-ATMOS dispose de l'intéressante possibilité de réaliser des affichages double hauteur, ce qui permettra d'améliorer très facilement l'aspect graphique de certains jeux.

Il suffit pour cela d'afficher le caractère concerné, précédé d'un caractère **CHR\$ (10)** sur deux lignes consécutives, avec néanmoins des règles un peu contraignantes sur la parité de la ligne inférieure. On pourra d'ailleurs utiliser cette possibilité avec les caractères graphiques définis par l'utilisateur.

LE MODE HAUTE RESOLUTION

Sur l'ORIC, le mode haute résolution est accessible par l'instruction **HIRES** (on reviendra au mode normal en tapant **TEXT**).

On dispose alors de la possibilité d'allumer ou d'éteindre 200×240 points, avec l'instruction **CURSET**, en tenant compte du repérage suivant :



Un troisième paramètre détermine l'appartenance (fond, ou écriture) du point concerné.

L'affichage d'un texte dans l'écran haute résolution se fait à l'aide de l'instruction **CHAR** (qui travaille caractère par caractère).

Un dé

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : tirage d'un dé (avec graphisme).

Difficulté du jeu : aucune.

LE JEU :

Le programme propose simplement le lancer d'un dé aléatoire, et sa représentation graphique en grand format.

Il pourra facilement être utilisé en sous-programme, dans tout type de jeu faisant appel au lancer d'un ou de plusieurs dés.

LE PROGRAMME :

La ligne 50 utilise le générateur aléatoire de l'ORIC-ATMOS pour tirer un nombre de 1 à 6.

La boucle J (70-100) dessine les bords du dé, et les lignes 60 et 110 dessinent le haut et le bas.

Les lignes 120 à 170 sont utilisées pour le dessin des points.

Les lignes 180 et 190 permettront le cas échéant de refaire un tirage.

POUR JOUER :

Lancer le programme par RUN.

On entre un entier pour initialiser le générateur, et un dé s'affiche, en couleurs, et en grand format.

A la question : AUTRE TIRAGE (O/N) on répond O, suivi de RETURN, pour relancer le dé.

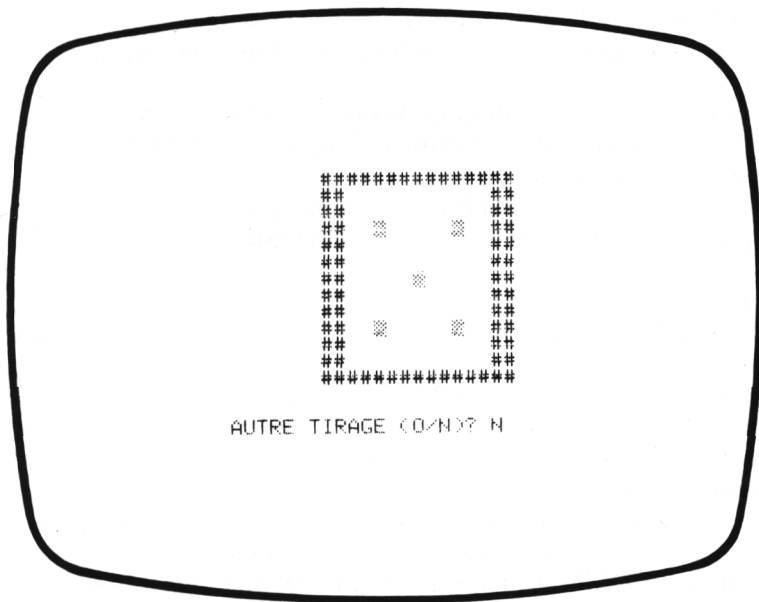
EXTENSIONS POSSIBLES :

On pourra introduire une petite musique agrémentant chaque lancer, faire défiler les points comme si le dé roulait sur une table, etc.

```

10 REM UN DE
20 CLS
30 PAPER 5
40 INK 4
50 A=INT(RND(1)*6+1)
60 PRINT@9,5;CHR$(27);"R";" " " " :PRINT CHR$(27);"U"
70 FOR J=0 TO 10
80 PRINT@9,6+J;CHR$(27);"R";" " " " :PRINTCHR$(27);"U"
90 PRINT@22,6+J;CHR$(27);"R";" " " " :PRINTCHR$(27);"U"
100 NEXT J
110 PRINT@9,17;CHR$(27);"R";" " " " :PRINT CHR$(27);"U"
120 IF A<>1 THEN PRINT@13,8;CHR$(255)
130 IF A=4 THEN PRINT@19,8;CHR$(255)
140 IF A=6 THEN PRINT@13,11;CHR$(255):PRINT@19,11;CHR$(255)
150 IF A/2<>INT(A/2) THEN PRINT@16,11;CHR$(255)
160 IF A=4 THEN PRINT@13,14;CHR$(255)
170 IF A<>1 THEN PRINT@19,14;CHR$(255)
180 PRINT@2,20;"AUTRE TIRAGE (O/N)";:PING:INPUT R$
190 IF LEFT$(R$,1)="O" THEN RUN

```



Plus grand, plus petit

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu de déduction (un seul joueur).

Difficulté du jeu : difficile.

LE JEU :

Il s'inspire d'un principe légèrement différent de celui du Mastermind ; il s'agit également de deviner un nombre (de 1 à 7 chiffres), mais les indications dont on dispose sont cette fois d'une part le nombre de chiffres qui, dans le nombre essayé, sont supérieurs à ceux du nombre à deviner, et, d'autre part, le nombre de chiffres qui sont inférieurs. Si, par exemple, le nombre à deviner est 3054, et que l'on essaie 4251, la machine répondra 2 plus grands (ce sont le 4 et le 2), et 1 plus petit (c'est le 1).

LE PROGRAMME :

La ligne 60 interdit de demander des nombres de plus de 8 chiffres.

Le nombre à deviner est placé en ligne 80, dans une variable alphanumérique X\$.

Les comparaisons avec le nombre essayé sont faites à la ligne 130 (qui teste si l'on a trouvé juste), et dans la boucle I (150-180) où l'on fait la comparaison "lettre à lettre".

L'affichage du résultat est réalisé grâce à la ligne 190.

Le compteur du nombre d'essais est en ligne 200.

POUR JOUER :

Le lancement du programme par RUN provoque l'affichage NOMBRE DE CHIFFRES.

On entre un nombre inférieur à 8 et on appuie sur la touche RETURN.

Le numéro de l'essai s'affiche, ainsi que la question NOMBRE ESSAYE ; on répond en entrant un nombre, suivi de RETURN.

Le résultat est alors affiché sous la forme > et <.

S'il n'est pas exact, on recommence (attention à entrer uniquement des nombres ayant autant de chiffres que le nombre à deviner).

Si l'on a trouvé l'ORIC-ATMOS affichera VOUS AVEZ TROUVE... EN... ESSAIS. Puis AUTRE PARTIE (O/N) ; on fera O suivi de RETURN pour commencer une nouvelle partie.

EXTENSIONS POSSIBLES :

On pourra, comme pour le Mastermind, imaginer un jeu analogue avec des lettres, des couleurs, ou même en utilisant d'autres symbolismes.

On peut aussi prévoir de laisser jouer l'ORIC-ATMOS qui tiendra alors le rôle de décodeur.

```

10 REM PLUS GRAND , PLUS PETIT
20 PAPER 6
30 INK 4
40 CLS
50 INPUT "NOMBRE DE CHIFFRES :":C
60 IF C>8 THEN 40
70 E=1:F=1
80 X#=STR$(INT(RND(1)*9^C+10^(C-E)))
90 X#=RIGHT$(X#,LEN(X#)-1)
100 PRINT@2,3:"ESSAI NO.  NOMBRE ESSAYE
110 PLOT @,4,CHR$(17)
120 PRINT@6,E+4:E:PRINT @16,E+4: INPUT P#
130 IF P#=X# THEN 220
140 A=0:B=0
150 FOR I=1 TO C
160 IF MID$(P#,I,1)>MID$(X#,I,1) THEN A=A+1
170 IF MID$(P#,I,1)<MID$(X#,I,1) THEN B=B+1
180 NEXT I
190 PRINT@29,E+4:A:PRINT@36,E+4:B:PING
200 E=E+1:F=F+1:IF F=18 THEN F=1
210 GOTO 120
220 PRINT:PRINT"VOUS AVEZ TROUVE ";X#;" EN ";E)
    " ESSAIS":EXPLODE
230 PRINT:INPUT"AUTRE PARTIE (O/N)":R#
240 IF LEFT$(R#,1)="O" THEN RUN

```

NOMBRE DE CHIFFRES :? 4

ESSAI NO.	NOMBRE ESSAYE	>	<
1	? 1234	0	2
2	? 5678	0	2
3	? 1200	0	4
4	? 9934	2	0
5	? 6734		

VOUS AVEZ TROUVE 6734 EN 5 ESSAIS

AUTRE PARTIE (O/N)? N

Ordre

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de réflexion.

Difficulté du jeu : moyenne.

LE JEU :

Les 10 premières lettres de l'alphabet sont affichées dans le désordre ; il faut réussir à les remettre dans l'ordre, avec le moins de coups possible, par des permutations successives de deux des lettres ; mais ceci a aussi pour inconvénient de faire permuter autour d'elles les deux lettres qui entourent celles-ci, suivant le mécanisme expliqué ci-dessous :

```
B A D G H I J C F E
0 1 2 3 4 5 6 7 8 9 : permutation 83
      ↑-----↑↑↑
      B A D F H J I E G C
      0 1 2 3 4 5 6 7 8 9 : résultat
```

LE PROGRAMME :

Les boucles emboîtées I (60-120) et J (80-100) définissent une variable alphanumérique A\$, contenant les 10 premières lettres de l'alphabet dans le désordre.

La ligne 140 affiche A\$, et la ligne 150 affiche les numéros de lettres à utiliser pour la permutation.

On entre la permutation en ligne 190.

Les lignes 200 et 210 transforment la variable alphanumérique entrée en une variable numérique.

La ligne 220 appelle une première fois le sous-programme d'inversion.

La ligne 250 appelle une deuxième fois le sous-programme d'inversion.

La ligne 260 comptabilise les essais.

La ligne 180 contrôle les résultats.

EXTENSIONS POSSIBLES :

Lancer le programme par RUN.

Le mot initial est affiché, chaque lettre étant repérée par un numéro de 1 à 9.

A la question INVERSION (I/J), on répond en entrant un nombre de deux chiffres xy ; x est le numéro de la première lettre, et y le numéro de la seconde. Les deux inversions sont alors successivement effectuées, et l'on recommence.

Le nombre de coups sera affiché en fin de partie.

EXTENSIONS POSSIBLES :

On peut imaginer un jeu contre l'ordinateur, qui évaluerait le score du joueur en fonction de ses propres performances ; on peut aussi modifier les règles qui régissent les permutations.

```

10 REM ORDRE
20 CLS
30 PAPER 3
40 INK 4
50 A$=""
60 FOR I=1 TO 10
70 R$=CHR$(INT(RND(1)*10+65))
80 FOR J=1 TO I-1
90 IF R$=MID$(A$,J,1) THEN 70
100 NEXT J
110 A$=A$+R$
120 NEXT I
130 S=1
140 PRINT@2,4;"RESULTAT "; A$;PING
150 PRINT@12,5;"0123456789"
160 PRINT@2,9;"ESSAI NUMERO ";S
170 PRINT@35,12;" "
180 IF A$="ABCDEFGHIJ" THEN PRINT "TROUVE EN ";S;" COUPS";EXPLODE
190 PRINT@2,12;"INVERSION (I ET J SOUS FORME IJ)";INPUT R$
200 I=ASC(LEFT$(R$,1))-47
210 J=ASC(RIGHT$(R$,1))-47
220 GOSUB 500
230 IF LEFT$(R$,1)="0" OR LEFT$(R$,1)="9" THEN 250
240 I=I-1;J=I+2
250 GOSUB 500
260 S=S+1
270 GOTO 140
500 B$=""
510 FOR T=1 TO LEN(A$)
520 IF T=I THEN N$=MID$(A$,T,1)
530 IF T=J THEN M$=MID$(A$,T,1)
540 NEXT T
550 FOR T=1 TO LEN(A$)
560 IF T=I THEN B$=B$+M$;GOTO 590
570 IF T=J THEN B$=B$+N$;GOTO 590
580 B$=B$+MID$(A$,T,1)
590 NEXT T
600 A$=B$
610 RETURN

```

```

RESULTAT :ABIEFDJCGH
          0123456789

```

```

ESSAI NUMERO 5

```

```

INVERSION (I ET J SOUS FORME IJ) ?

```

Le pendu

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de lettres (version deux joueurs).

Difficulté du jeu : moyenne.

LE JEU :

Il serait sans doute assez difficile de retrouver le nom de l'inventeur de ce jeu très à la mode, mais il est certain qu'il existe depuis des temps immémoriaux, quelquefois pratiqué sous des formes assez différentes (par exemple, reconstituer un petit cochonnet en lançant des dés).

L'apparition des jeux sur micro-ordinateur et la magie des écrans de télévision lui ont fait connaître un renouveau certain : dans la version retenue, l'un des joueurs doit entrer un mot de longueur quelconque. L'ordinateur affichera alors des tirets à l'emplacement des lettres de ce mot et demande au deuxième joueur de proposer une lettre. Chaque réponse correcte fait apparaître la lettre à son emplacement exact dans le mot, et chaque réponse fautive fait progresser le dessin de la potence.

Pour gagner, il suffit de réussir à trouver le mot inconnu avant d'être pendu.

LE PROGRAMME :

Le mot à deviner est introduit en début de programme (ligne 30) ; une variable B\$, en ligne 40, contient tous les éléments du dessin.

La boucle I (60-80) fabrique un mot P\$ qui contient des tirets à l'emplacement de chacune des lettres du mot initial.

La boucle I (120-140) s'occupe du dessin (plus ou moins complet, selon la valeur de P).

Les lignes 130-160 affichent le résultat et arrêtent le jeu.

La lettre essayée est entrée en ligne 170 et testée dans une boucle I (190-210) ; le sous-programme 300-320 remplace chaque tiret par la lettre correspondante.

POUR JOUER :

Après le lancement du programme par RUN, l'un des joueurs doit entrer son mot et taper RETURN. Des tirets sont affichés à l'emplacement de chacune des lettres du mot ; et la question VOTRE LETTRE apparaît sur l'écran ; l'autre joueur répond alors en proposant une lettre suivie de RETURN.

Si la lettre est bonne, elle viendra remplacer un ou plusieurs tirets ; sinon, la potence va commencer à se dessiner.

Le mot exact sera affiché dans le cas où l'on aurait perdu.

EXTENSIONS POSSIBLES :

Il sera intéressant de constituer une liste de mots dans lesquels l'ORIC-ATMOS pourra piocher au hasard ; on pourra aussi améliorer le dessin (avec les caractères graphiques, par exemple) et modifier le nombre d'essais permis selon la difficulté du mot.

Belote

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de cartes.

Difficulté du jeu : facile.

LE JEU :

Le jeu proposé est plutôt une simulation, réalisée sur 32 cartes, de la distribution de 4 paquets de 5 cartes. Il pourra être utilisé pour l'étude de problèmes faisant intervenir des donnes de 20 cartes sur 32.

LE PROGRAMME :

En ligne 50, on crée un tableau pouvant contenir les 20 cartes.

La boucle I (80-140) assure le tirage aléatoire de 20 nombres, pris parmi 32 ; la boucle J (110-130) vérifie que l'on ne réalise pas deux fois le même tirage.

La deuxième boucle I (160-310) assure ensuite l'affichage, les noms des cartes étant affichés en clair, et la couleur étant décodée et affichée en clair, grâce aux lignes 210 à 280.

POUR JOUER :

Il suffit de taper RUN pour obtenir immédiatement l'affichage de 4 paquets de 5 cartes chacun (une donne).

Un nouveau RUN permet d'obtenir une nouvelle distribution.

EXTENSIONS POSSIBLES :

On pourra facilement mettre au point un affichage plus graphique en utilisant les caractères graphiques redéfinissables d'ORIC-ATMOS ; on pourra également se servir de ce programme comme d'un noyau pour la mise au point de programmes de jeu de cartes sur l'ordinateur (qui pourra, si nécessaire, servir de partenaire : voir l'exemple du Poker, dans le même ouvrage).

```

10 REM BELOTE
20 CLS:PRINT CHR$(17)
30 PAPER 2
40 INK 0
50 DIM A(20)
60 K=1
70 PRINT@12,12;CHR$(27);"L";"PATIENTEZ - JE TIRE"
80 FOR I=1 TO 20
90 A(I)=1 + INT(RND(1)*32)
100 IF I=1 THEN 140
110 FOR J=1 TO I-1
120 IF A(I)=A(J) THEN 90
130 NEXT J
140 NEXT I
150 CLS
160 FOR I=1 TO 20
170 S$="CA"
180 S=1+A(I)-8*INT(A(I)/8)
190 R$=STR$(S+6)+" " :R$=RIGHT$(R$,LEN(R$)-1)
200 R$=LEFT$(R$,2)
210 IF S=5 THEN R$="VA"
220 IF S=6 THEN R$="DA"
230 IF S=7 THEN R$="RO"
240 IF S=8 THEN R$="AS"
250 IF A(I)<=24 THEN S$="TR"
260 IF A(I)<=16 THEN S$="PI"
270 IF A(I)<=8 THEN S$="CO"
280 PRINT@10*INT((I-1)/5)+2,2*K;R$;" " ;S$
290 K=K+1:PING:WAIT 50
300 IF K=6 THEN K=1
310 NEXT I

```

AS CO	VA CO	RO TR	RO CA
9 CO	7 TR	RO PI	VA CA
10 TR	8 CA	AS TR	VA TR
7 CA	8 PI	DA TR	DA PI
7 PI	DA CA	10 CO	VA PI

Un taquin de calcul

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : difficile.

LE JEU :

Il se pratique à deux joueurs, l'un tenant le rôle du codeur, l'autre celui du décodeur.

Le codeur part d'un nombre (proposé par l'ORIC-ATMOS) et doit le transformer à l'aide de 5 opérations qu'il choisit lui-même, ainsi d'ailleurs que les opérateurs.

Le résultat qu'il a obtenu et les opérations qu'il a utilisées seront affichés sur l'écran : le décodeur devra alors essayer de retrouver le nombre initial en effectuant les opérations en sens inverse.

LE PROGRAMME :

La ligne 50 permet d'obtenir l'entier de départ compris entre 5 et 20.

Une boucle I (110-200) permet d'entrer successivement les 5 opérations.

4 sous-programmes sont utilisés pour les calculs (aux lignes 1000 à 4000).

Après chaque calcul, le résultat est affiché (seul, le codeur devra regarder l'écran pendant cette opération) et le résultat définitif restera affiché pour le décodeur à la fin des calculs (ligne 190).

La ligne 210 affiche les 5 opérations utilisées, ainsi que les opérateurs correspondants.

POUR JOUER :

On frappe RUN pour lancer le programme.

Le nombre du début est affiché, puis la question ENTREZ 5 OPERATIONS ; on devra les entrer l'une après l'autre, sous la forme, par exemple : + 4 suivi de RETURN.

Le résultat sera affiché après chaque opération entrée. A la fin, les opérations entrées sont toutes affichées sur une même ligne, et le message BON COURAGE est envoyé à l'adresse du décodeur (il n'y a pas de contrôle au décodage).

EXTENSIONS POSSIBLES :

On pourra, bien sûr, songer à améliorer la partie décodage du programme en introduisant des contrôles sur le résultat ; on peut aussi envisager de laisser l'ORIC-ATMOS participer davantage en lui confiant le rôle du codage, etc.

```

10 REM UN TAQUIN DE CALCUL
20 PAPER 5
30 INK 0
40 CLS
50 X=INT(RND(1)*16+5)
60 PRINT:PRINT"NOMBRE INITIAL :";X
70 PRINT:PRINT"ENTREZ 5 OPERATIONS"
80 PRINT"SOUS LA FORME +5 , /2 , *4 ..."
90 F=X
100 DIM R$(5)
110 FOR I=1 TO 5
120 PRINT @18,6;" "
130 PRINT@2,6;"VOTRE OPERATION :";INPUT R$(I)
140 A=ASC(LEFT$(R$(I),1))-41
150 IF A<=0 OR A=3 OR A=5 OR A>6 THEN 120
160 IF A>3 THEN A=(A/2)+1
170 ON A GOSUB 1000,2000,3000,4000
180 PRINT@16,10;" "
190 PRINT@2,10;"NOMBRE OBTENU :";F:PING
200 NEXT I
210 PRINT:PRINT"OPERATIONS :";R$(4);" ";R$(1);"
";R$(5);" ";R$(3);" ";R$(2)
220 PRINT:PRINT:PRINT"BON COURAGE....":EXPLODE
230 END
1000 F=F*VAL(RIGHT$(R$(I),LEN(R$(I))-1))
1010 RETURN
2000 F=F+VAL(RIGHT$(R$(I),LEN(R$(I))-1))
2010 RETURN
3000 F=F-VAL(RIGHT$(R$(I),LEN(R$(I))-1))
3010 RETURN
4000 F=F/VAL(RIGHT$(R$(I),LEN(R$(I))-1))
4010 RETURN

```

```

NOMBRE INITIAL : 16
ENTREZ 5 OPERATIONS
SOUS LA FORME +5 , /2 , *4 ...
VOTRE OPERATION : ? -50

NOMBRE OBTENU : 10
OPERATIONS : +28 -4 -50 /3 *8

BON COURAGE....

```

Pendu inversé

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu sur les mots.

Difficulté du jeu : facile.

LE JEU :

Le Pendu est un jeu de lettres très connu, présenté par ailleurs dans ce recueil, dans lequel il s'agit de deviner un mot caché par l'ordinateur et où chaque lettre non découverte vous rapproche un peu plus de la potence. Ici, c'est l'ORIC-ATMOS qui va chercher à deviner votre mot, mais il joue quelque peu au hasard et ne possède pas votre intuition ; il faudra donc, en règle générale, beaucoup d'essais pour découvrir la solution. Pour augmenter ses chances, on a fait appel à une formule qui lui permet de tirer davantage dans les voyelles puis, dans les consonnes courantes, et, enfin, dans les consonnes plus rares.

LE PROGRAMME :

La ligne 50 permet d'enregistrer, dans une variable alphanumérique D\$, les lettres de l'alphabet, dans l'ordre : voyelles, consonnes usuelles, autres consonnes. Le mot à deviner est introduit en ligne 70 (en fait, il suffirait d'introduire sa longueur).

La boucle I (90-120) imprime des tirets aux emplacements des lettres à trouver.

La temporisation en ligne 140 sert uniquement à simuler le temps de recherche d'une lettre.

La formule, placée en ligne 150, permet de modifier légèrement la probabilité de tirage, en faveur des voyelles et des consonnes usuelles.

La ligne 170 provoque un nouveau tirage, et la boucle I (210-240) réimprime les tirets et les lettres trouvées.

La ligne 250 incrémente le compteur d'essais.

La ligne 260 affiche le nombre d'essais réalisés, en fin de partie.

POUR JOUER :

Lancer le programme par RUN.

Entrer le mot à découvrir, suivi de RETURN.

Des tirets sont alors affichés (un par lettre) et l'ORIC-ATMOS propose une première lettre, regarde si elle convient et continue automatiquement, jusqu'à ce qu'il ait trouvé.

On pourra sortir du programme à tout moment en tapant CTRL et C.

EXTENSIONS POSSIBLES :

Il sera sans doute intéressant, pour de jeunes enfants, de prévoir une version non automatique où le joueur répondra OUI ou NON pour chaque lettre proposée par l'ORIC-ATMOS ; on pourra aussi imaginer d'ajouter un affichage graphique, par exemple une potence ou une voiture qui avance vers un précipice, etc.

```

10 REM PENDU INVERSE
20 CLS
30 PAPER 6
40 INK 4
50 D$="YAEIOUBCDFGLMNPSTVWHKJQXZ"
60 C=1:T$=""
70 INPUT"ENTREZ VOTRE MOT :";R$
80 CLS
90 FOR I=1 TO LEN(R$)
100 PRINT@I+2,5;"-"
110 T$=T$+"-"
120 NEXT I
130 PRINT@2,9;"JE CHERCHE.....; ESSAI ";C
140 WAIT 500
150 X=-6*(RND(1)<0.7)-13*(RND(1)<0.5)-7*(RND(1)<0.4)
160 X=INT(X*RND(1)+1)
170 IF MID$(D$,X,1)="-" THEN 150
180 E$=MID$(D$,X,1)
190 D$=LEFT$(D$,X-1)+"-"+RIGHT$(D$,26-X)
200 PRINT:PRINT:PRINT"JE PRENDS :";E$:PING
210 FOR I=1 TO LEN(R$)
220 IF E$=MID$(R$,I,1)
    THEN T$=LEFT$(T$,I-1)+E$+RIGHT$(T$,LEN(T$)-I):ZAP
230 PRINT@2+I,5;MID$(T$,I,1)
240 NEXT I
250 C=C+1
260 IF T$=R$ THEN PRINT@2,18;
    "TROUVE EN ";C-1;" COUPS":EXPLODE:END
270 GOTO 130

```

VAOROUILLEUR

JE CHERCHE.....; ESSAI 21

JE PRENDS :R

TROUVE EN 21 COUPS

Nicomaque

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de réflexion (calcul mental).

Difficulté du jeu : moyenne.

LE JEU :

Il utilise une propriété un peu particulière des entiers de 1 à 100 qui peuvent être identifiés sans ambiguïté par leurs restes dans les divisions par 3, 5 et 7.

Le programme proposé ici, est un programme codeur qui affiche simplement les restes d'un entier donné N dans les divisions indiquées.

Le jeu consiste à essayer de découvrir l'entier (mais, dans cette version, vous n'avez droit qu'à un seul essai).

LE PROGRAMME :

La ligne 50 permet de confectionner l'entier.

La boucle I (80-130) calcule les trois restes successifs, et la boucle emboîtée J (100-120) permet de les afficher sous une forme assez visuelle, aux bons emplacements.

Les lignes 60 et 70 sont là pour indiquer de quelle division il s'agit.

Le nombre proposé par le joueur est entré à la ligne 140 ; il est affiché et testé aux lignes 150 et 160 (affichage de fin de partie : BRAVO ou PERDU).

Les lignes 170 et 180 s'occupent de gérer la prochaine partie, selon la réponse du joueur.

POUR JOUER :

On lance le programme en tapant RUN.

Les trois restes seront alors affichés sur la même ligne, symbolisés par des étoiles (l'absence d'étoiles signifiant : reste 0).

A la question VOTRE ESSAI, on répondra en donnant un entier compris entre 0 et 100, suivi de RETURN.

On obtiendra, en réponse, un affichage PERDU ou BRAVO, puis la question AUTRE PARTIE (O/N).

Pour faire un nouvel essai, on répondra O, suivi de RETURN.

EXTENSIONS POSSIBLES :

On peut envisager de réaliser le programme décodeur (on donne les restes et l'ORIC-ATMOS devra vous donner N) et chercher si la propriété peut être étendue (en effectuant davantage de divisions, par exemple).


```

10 REM NICOMAUQUE
20 CLS
30 PAPER 3
40 INK 0
50 X=INT(RND(1)*100+1)
60 PRINT"RESTE DANS LA DIVISION PAR"
70 PRINT:PRINT TAB(12);"3";TAB(20);"5";TAB(28);"7"
80 FOR I=1 TO 3
90 A=X-(2*I+1)*INT(X/(2*I+1))
100 FOR J=1 TO A
110 IF A<>0 THEN PRINT@J+8*I+3.5;"*":ZAP
120 NEXT J
130 NEXT I
140 PRINT@2,10;"VOTRE ESSAI :?";INPUT N
150 IF N=X THEN PRINT:PRINT"BRAVO":EXPLODE
160 IF N<>X THEN PRINT:PRINT"PERDU":PING
170 PRINT:INPUT"AUTRE PARTIE (O/N) :?";R$
180 IF LEFT$(R$,1)="O" THEN RUN

```

RESTE DANS LA DIVISION PAR

3 5 7

*

VOTRE ESSAI :? 15

BRAVO

AUTRE PARTIE (O/N) :? N

Course de chevaux

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : hasard (plusieurs joueurs).

Difficulté du jeu : aucune.

LE JEU :

On entrera un nombre de chevaux compris entre 1 et 10, et la course débutera immédiatement.

Il pourra y avoir des ex-aequo, car l'on tiendra compte du fait que l'ORIC-ATMOS s'occupe de faire avancer les chevaux les uns après les autres, en commençant par le numéro 1 ; le classement sera affiché au fur et à mesure des arrivées (et en tenant compte des ex-aequo).

LE PROGRAMME :

Une première boucle non formelle (80-170) affiche les chevaux dans leur position de départ.

Une deuxième boucle I (100-180) les fera avancer ; la position de chaque cheval sera repérée par un chiffre ou une lettre dans une variable alphanumérique A\$.

La ligne 120 fera avancer les chevaux.

La ligne 140 gère les arrivées.

La ligne 150 réalisera les impressions.

Le sous-programme 240-280 fait avancer le compteur des arrivées, et imprime les résultats.

POUR JOUER :

On lance le programme par RUN.

A la question NOMBRE DE CHEVAUX : on répond par un entier de 1 à 7, suivi de RETURN.

La course se déroulera ensuite automatiquement.

EXTENSIONS POSSIBLES :

On pourra dessiner les chevaux avec un graphisme plus soigné, et les différencier par leur couleur.

On pourra aussi imaginer tout un système de paris, qui interviendra avant le départ de chaque course.

```

10 REM COURSE DE CHEVAUX
20 CLS:PRINT CHR$(17)
30 PAPER 7
40 INK 0
50 INPUT"OMBRE DE CHEVAUX :";N:IF N>7 THEN 20
60 A$="":P=0:A=0:T=1:K=0:J=T
70 FOR I=1 TO N:PLOT1,2*I+3,CHR$(I-1):NEXT I
80 A$=A$+"0":I=J
90 GOTO 150
100 FOR I=1 TO N
110 IF MID$(A$,I,1)="K" THEN 180
120 B#=CHR$(ASC(MID$(A$,I,1))+INT(RND(1)*4+1))
130 A$=LEFT$(A$,I-1)+B#+RIGHT$(A$,LEN(A$)-I)
140 IF B#>="K" THEN GOSUB 240
150 PRINT@ASC(MID$(A$,I,1))-46,2*I+3;" ";I
160 ZAP:WAIT 150
170 IF K=0 THEN J=J+1:IF JK=N THEN 80 ELSE K=1:
    GOTO 100
180 NEXT I
190 P=P+T
200 T=0
210 PING
220 IF A<N THEN 100
230 EXPLODE:END
240 A$=LEFT$(A$,I-1)+"K"+RIGHT$(A$,LEN(A$)-I)
250 PRINT@2,2*I+3;CHR$(27);"L":;
    PRINT CHR$(27);"@";P:;PRINT CHR$(27);"H";
260 PRINT CHR$(27);CHR$(63+I):T=1
270 A=A+T:PING:WAIT 30:PING:WAIT 30:PING:WAIT 30
280 RETURN

```

NOMBRE DE CHEVAUX : ? 6

```

1
2
3
4
5
6

```

A travers

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : moyenne.

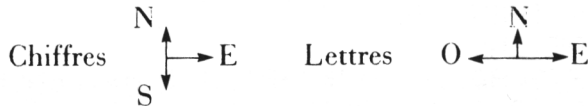
LE JEU :

Sur l'écran apparaît un carré de 9 cases, avec sur le bord vertical des lettres et sur le bord horizontal des chiffres.

L'un des joueurs manipule les chiffres, l'autre les lettres. 3 déplacements sont possibles pour les chiffres, et 3 autres pour les lettres. Les lettres peuvent se déplacer vers le haut, à droite et à gauche et les chiffres vers la droite, en haut ou en bas.

Le gagnant est le premier qui arrive à faire sortir tous ses symboles (les lettres sortent en haut, et les chiffres à droite).

Il n'y a pas de prise, et les sauts ne sont pas autorisés ; on ne peut pas non plus se placer sur une case déjà occupée.



LE PROGRAMME :

Les positions des chiffres et des lettres sont enregistrées dans une table N (4, 4) ; les lignes 30 à 70 définissent les valeurs de départ.

Les lignes 90 à 150 dessinent le terrain de jeu et la ligne 160 affiche le joueur qui a le trait.

La ligne 190 contrôle le coup joué, de même que les boucles I et J (200 à 240), et le changement de joueur est assuré par la ligne 330.

POUR JOUER :

Le terrain de jeu sera affiché après la frappe de RUN et l'ordinateur attend le coup du joueur désigné.

On entrera d'abord le chiffre ou la lettre représentant le pion à déplacer, suivi d'une indication de direction sous la forme N (pour Nord), S (pour Sud), E (pour Est) ou O (pour Ouest).

On pourra ainsi faire 2E suivi de RETURN.

Si le coup est non valide, il ne sera pas exécuté, et le code du joueur restera à l'affichage ; s'il est autorisé, le déplacement sera effectué, et l'on obtient l'affichage correspondant au deuxième joueur.

Le premier qui aura réussi à faire disparaître ses 3 pièces aura gagné.

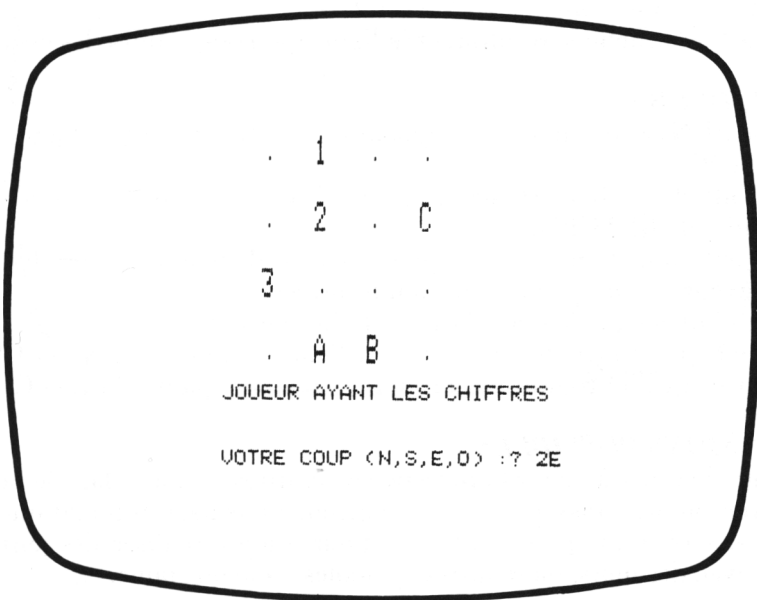
EXTENSIONS POSSIBLES :

On peut envisager d'étendre le jeu avec davantage de chiffres ou de lettres et d'améliorer les contrôles. On peut aussi définir une stratégie pour l'ORIC-ATMOS.

```

10 REM A TRAVERS
20 DIM N(4,4):K=1:A#=CHR$(27)
30 FOR I=1 TO 4
40 N(I,1)=ASC("0")+I
50 N(4,I)=ASC("?")+I
60 NEXT I
70 N(4,1)=0
80 CLS:PAPER 6:INK 4
90 FOR U=1 TO 4
100 PLOT1,4*U-1,CHR$(17):PLOT1,4*U,CHR$(17)
110 PLOT22,4*U-1,CHR$(22):PLOT22,4*U,CHR$(22)
120 FOR V=1 TO 4
130 IF N(V,U)=0 THEN B$="." ELSE B#=CHR$(N(V,U))
140 PLOT4*U,4*V-1,CHR$(10)+B$:PLOT4*U,4*V,CHR$(10)+B#
150 NEXT V:NEXT U
160 PRINT@2,18;"JOUEUR AYANT LES ";
170 IF K=0 THEN PING:PRINT "LETTRES" ELSE ZAP:PRINT "CHIFFRES"
180 PRINT@2,22;"VOTRE COUP (N,S,E,O) :";:INPUT R#
190 IF (K=1 AND RIGHT$(R$,1)="O")
    OR (K=0 AND RIGHT$(R$,1)="S") THEN 180
200 FOR I=1 TO 4
210 FOR J=1 TO 4
220 IF N(I,J)=ASC(LEFT$(R$,1)) THEN 260
230 NEXT J
240 NEXT I
250 GOTO 160
260 A=I:B=J
270 I=I-(RIGHT$(R$,1)="S")+(RIGHT$(R$,1)="N")
280 J=J-(RIGHT$(R$,1)="E")+(RIGHT$(R$,1)="O")
290 IF (I=0 AND K=0) OR (J>4 AND K=1) THEN 320
300 IF N(I,J)<>0 THEN 160
310 N(I,J)=ASC(LEFT$(R$,1))
320 N(A,B)=0
330 K=ABS(K-1)
340 GOTO 80

```



Mastermind

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : logique (un seul joueur).

Difficulté du jeu : difficile.

LE JEU :

Devenu un grand classique des jeux sur micro-ordinateur, ce jeu développe le raisonnement déductif ; il faut réussir à trouver le plus vite possible un nombre de *n* chiffres (on pourra prendre *n* compris entre 1 et 6), en disposant à chaque essai de renseignements.

En effet, l'ORIC répondra en indiquant d'une part le nombre de chiffres qui sont, dans le nombre proposé, exactement à la même place que dans le nombre à découvrir, et, d'autre part, le nombre de chiffres qui figurent dans le nombre à découvrir sans être exactement au bon emplacement.

LE PROGRAMME :

Le choix du nombre de chiffres du nombre à découvrir est effectué aux lignes 30 et 40.

Le nombre est fabriqué à la ligne 60 (il pourra y avoir des chiffres répétés). L'affichage est préparé par la ligne 90.

La ligne 130 s'occupe de la fin de partie.

Une première boucle I (140-230) teste les chiffres qui sont à la bonne place.

Une deuxième boucle J (160-180) teste les chiffres qui sont présents et mal placés.

Les compteurs sont aux lignes 200 et 220.

La ligne 240 affiche le résultat, et la ligne 250 comptabilise les essais.

POUR JOUER :

On tape RUN et l'on choisit la longueur du nombre désiré (de 1 à 6), suivie de RETURN.

Après l'affichage NO. NOMBRE BON PLACE, on introduit son propre essai, suivi de RETURN.

On obtient immédiatement la réponse (en cas de double chiffre, les résultats peuvent être un peu délicats à interpréter).

On introduit ensuite le nouvel essai, et ainsi de suite.

Lorsque l'on a trouvé, on obtiendra l'affichage TROUVE EN ... COUPS.

On peut sortir du programme à tout moment en tapant CTRL et C.

EXTENSIONS POSSIBLES :

On peut chercher à éviter la présence de chiffres doubles dans le nombre à trouver, on peut aussi prévoir un changement de page automatique, avec rappel des résultats précédents ; on pourra aussi réaliser des jeux analogues avec des mots, ou d'autres symboles (couleur, taille...).

```

10 REM MASTERMIND
20 CLS:PAPER 6:INK 4
30 INPUT"COMBIEN DE CHIFFRES :":N
40 IF N>6 THEN 20
50 E=1:F=1
60 X$=STR$(INT(RND(1)*9^N+10^(N-1)))
70 X$=RIGHT$(X$,LEN(X$)-1)
80 PAPER 1:INK 3:CLS
90 PRINT:PRINT"NO.          NOMBRE          BON   PLACE"
100 PLOT0,2,CHR$(22)
110 PRINT@2,3+F;E;TAB(8);:INPUT E$
120 X=0:Y=X
130 IF E$=X$ THEN PRINT"TROUVE EN ";E;" COUPS":EXPLODE:END
140 FOR I=1 TO N
150 IF MID$(X$,I,1)=MID$(E$,I,1) THEN 220
160 FOR J=1 TO N
170 IF J<>I AND MID$(X$,I,1)=MID$(E$,J,1) THEN 200
180 NEXT J
190 GOTO 230
200 Y=Y+1
210 GOTO 230
220 X=X+1
230 NEXT I
240 PRINT@24,3+F;X;TAB(8);Y:PING
250 E=E+1:F=F+1:IF F=18 THEN F=1
260 GOTO 110

```

NO.	NOMBRE	BON	PLACE
1	1234	0	2
2	5678	1	1
3	8909	0	0
4	7100	2	1
5	7174	3	0
6	7172		

TROUVE EN 6 COUPS

Idem

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de calcul mental (2 joueurs).

Difficulté du jeu : selon le niveau choisi.

LE JEU :

L'ORIC-ATMOS choisit un nombre de n chiffres (selon le niveau de difficulté indiqué au début). Ce nombre (le même au début pour les 2 joueurs) sera régulièrement décrémenté à chaque tour, d'une quantité également choisie au hasard par l'ORIC-ATMOS en début de jeu ; vous devez, à chaque coup, choisir une colonne (les dizaines, les centaines, etc.) et ajouter dans cette colonne un certain entier ; le but est d'arriver à obtenir à un moment l'affichage d'un nombre ayant tous ses chiffres identiques.

Chaque résultat sera affiché un court instant, après quoi la main passera au joueur suivant, qui repartira de son précédent résultat.

LE PROGRAMME :

Une variable numérique N est dimensionnée à 2 en ligne 40, pour contenir les résultats successifs des deux joueurs.

La ligne 50 permettra d'entrer le niveau de difficulté choisi.

Le nombre initial est confectionné par la ligne 70. La quantité qui interviendra dans les décrémentations est obtenue en ligne 90.

La ligne 120 teste si le résultat n'est pas trop petit.

La ligne 130 affiche le numéro du joueur.

Le nombre est affiché en ligne 140 et le jeu du joueur est entré en ligne 160 (nombre de deux chiffres).

La ligne 170 incrémente la colonne choisie, et la ligne 180 décrémenté le nombre de la quantité choisie au départ.

Le résultat sera affiché un court instant (ligne 210). La boucle I (260-280) permet de vérifier si tous les chiffres sont identiques.

L'impression et la fin de partie sont effectuées en ligne 290. Les lignes 300 et 310 s'occupent du changement de joueur.

POUR JOUER :

Lancer l'exécution du programme par RUN.

Le nombre initial sera affiché pour le joueur n° 1, qui joue en tapant xy , suivi de RETURN ; x est le chiffre à ajouter, et y est le numéro de la colonne où il faut l'ajouter.

Le résultat (provisoire s'il est trop petit) sera alors affiché un court instant, et le trait passe au deuxième joueur, et ainsi de suite. On peut arrêter à tout moment en tapant CTRL et C.

EXTENSIONS POSSIBLES :

On pourra étendre le jeu à plus de 2 joueurs, améliorer le choix du nombre intervenant dans la décrémentation, améliorer les divers contrôles, laisser les différents résultats affichés, etc.


```

10 REM IDEM
20 CLS
30 PAPER 1:INK 0
40 H=1:DIM N(2):A$=CHR$(27)
50 INPUT"NIVEAU (2-7) :";N
60 U=10:P=0
70 N(1)=INT((U^N-U^(N-1))*RND(1)+U^(N-1))
80 N(2)=N(1)
90 A=INT(2*U^(N-1)*RND(1)+U^(N-1))
100 IF H=1 THEN P=P+1
110 CLS
120 IF N(H)<U^(N-1) THEN N(H)=U^N-ABS(N(H))
130 PRINT@2,3;"JOUEUR ";H;"      ESSAI ";P:PING
140 PRINT@2,6;"NOMBRE  ";":PRINT@17-N,6;A$;"S";N(H);A$;"Q"
150 PRINT@11,7;A$;"V";"7654321 ";A$;"Q"
160 PRINT@2,10;"VOTRE JEU (N ET C SOUS LA FORME NC)";:INPUT R$
170 N(H)=N(H)+VAL(LEFT$(R$,1))*U^(VAL(RIGHT$(R$,1))-1)
180 N(H)=N(H)-A
190 IF N(H)>U^N THEN N(H)=N(H)-U^N
200 N(H)=INT(N(H))
210 PRINT@2,13;"RESULTAT  ";:A$;"J";N(H):
    PRINT@12,14;A$;"J";N(H)
220 EXPLODE
230 WAIT 180
240 R$=STR$(N(H))
250 R$=RIGHT$(R$,LEN(R$)-1)
260 FOR I=1 TO LEN(R$)-1
270 IF MID$(R$,I,1)<>MID$(R$,I+1,1) THEN 300
280 NEXT I
290 PRINT:PRINT"BRAVO , JOUEUR ";H:ZAP:ZAP:ZAP:END
300 H=H+1
310 IF H>=3 THEN H=1
320 GOTO 100

```

```

      JOUEUR 2      ESSAI 4

      NOMBRE :      748
                7654321

      VOTRE JEU (N ET C SOUS LA FORME NC)? 1
      RESULTAT :      629

```

Horloge

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : dessin.

Difficulté du jeu : aucune.

LE JEU :

Transformer votre écran de télévision en horloge numérique à affichage grand format, grâce à ce programme qui vous permettra d'animer une vitrine, de faire une horloge fantaisie, et qui pourra se prêter à bien d'autres usages.

LE PROGRAMME :

La ligne 30 vous permettra d'entrer l'heure sous forme d'un entier de 4 chiffres (par exemple : 1024 signifiera 10 heures 24 minutes).

Les lignes 80-170 réalisent le dessin, en décalant l'impression au fur et à mesure (à l'aide de la variable K, incrémentée en ligne 160).

Les lignes 180 à 220 assurent la progression de l'heure.

Les lignes 230 à 290 effectuent l'affichage de deux points clignotants pour les secondes.

POUR JOUER :

Dès le lancement par RUN, le programme se place en attente de l'heure ; on tape et on attend le top pour appuyer sur RETURN.

L'heure s'affiche alors assez rapidement et l'horloge est en route. Pour l'arrêter, il faudra appuyer sur les touches CTRL et C.

L'heure sera entrée sous forme d'un entier ayant impérativement 4 chiffres, pour 5 heures 6 minutes, on veillera à taper 0506 suivi de RETURN.

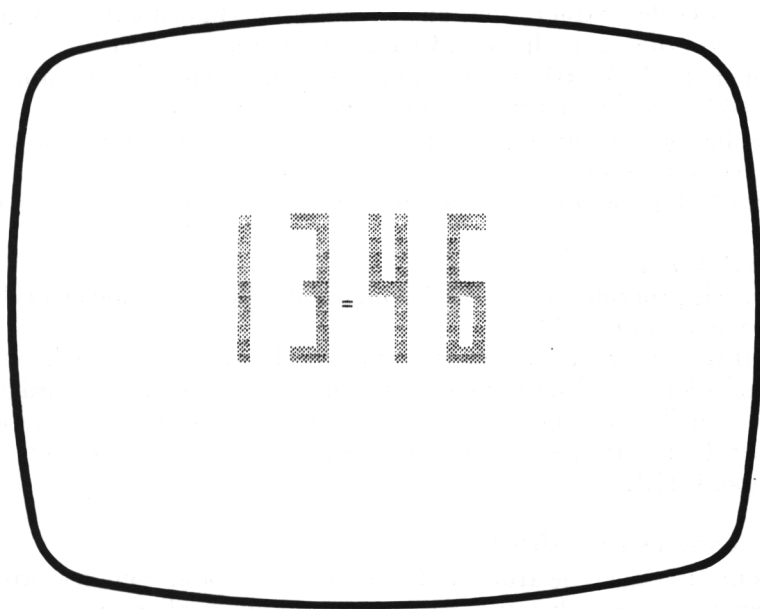
EXTENSIONS POSSIBLES :

Il sera intéressant de modifier le programme pour en faire un réveil, avec alarme ; on peut aussi améliorer le graphisme, introduire la couleur.

```

10 REM HORLOGE
20 CLS:B#=CHR$(255):PRINT CHR$(17)
30 INPUT"HEURE ( FORME XXXX )":A$
40 PAPER 6:INK 1:CLS
50 K=1
60 A=VAL(MID$(A$,K,1))
70 IF A<>1 AND A<>4 THEN PRINT@6*K-3,1;B$+B$+B$
80 FOR J=1 TO 5
90 IF A=0 OR (A>3 AND A<>7) THEN PRINT@6*K-3,J;B$
100 IF A<>5 AND A<>6 THEN PRINT@6*K-1,J;B$
110 IF A=0 OR A=2 OR A=6 OR A=8 THEN PRINT@6*K-3,J+4;B$
120 IF A<>2 THEN PRINT@6*K-1,J+4;B$
130 NEXT J
140 IF A>1 THEN PRINT@6*K-2.5;B$+B$
150 IF A<>1 AND A<>4 AND A<>7 THEN PRINT@6*K-3,9;B$+B$+B$
160 K=K+1
170 IF K<5 THEN 60
180 A=1 + VAL(A$)
190 IF A=2360 THEN A=0
200 A$="000"+RIGHT$(STR$(A),LEN(STR$(A))-1)
210 A$=RIGHT$(A$,LEN(A$)-3)
220 IF RIGHT$(A$,2)="60"
    THEN A$=RIGHT$(STR$(VAL(LEFT$(A$,2))+1),2)+"00"
230 H=1
240 FOR J=1 TO 128
250 FOR I=1 TO 32
260 IF H=1 THEN PRINT@13,6;"=" ELSE PRINT@13,6;" "
270 NEXT I
280 H=ABS(H-1)
290 NEXT J
300 PING:GOTO 40

```



Reines sur l'échiquier

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

Le programme utilise un algorithme très performant pour résoudre un problème qui a donné du fil à retordre à des chercheurs, même aussi célèbres que le mathématicien GAUSS, qui n'avait pas pu découvrir toutes les solutions. Il s'agit de trouver toutes les manières de disposer des reines sur un échiquier, de telle sorte qu'aucune ne puisse en prendre une autre. Le programme proposé est très souple, puisque vous disposez même de la possibilité de choisir la taille de votre échiquier, et les résultats seront affichés à la fois sous forme visuelle et sous forme numérique.

LE PROGRAMME :

La ligne 30 vous permet de choisir la taille de l'échiquier.

Une variable D sera dimensionnée en conséquence, en ligne 40.

Une boucle d'exploration principale I (50-110) contient une sous-boucle J (80-100) qui réalise le contrôle de non-alignement (ligne 90).

Si, au cours de l'exploration, on rencontre un alignement, on revient en arrière, en utilisant les lignes 230 à 270 et on recommence.

La boucle F (130-210) assure l'impression de l'échiquier, et la boucle K (160-180), que l'on pourra éventuellement supprimer, assure l'impression du résultat sous forme d'un code numérique correspondant à la position des pièces dans chaque colonne.

La ligne 220 permet un arrêt après l'affichage de chacune des solutions.

POUR JOUER :

On lance le programme avec RUN, puis on tape la dimension de l'échiquier désirée, et on fait RETURN.

Il suffit d'attendre, avec patience (plus l'échiquier sera grand, plus l'attente sera longue). La première solution sera affichée puis le programme s'arrête, et il faut appuyer sur les touches Ø et RETURN pour le faire repartir. Pour terminer l'exploration sans avoir toutes les solutions, il faudra taper CTRL et C.

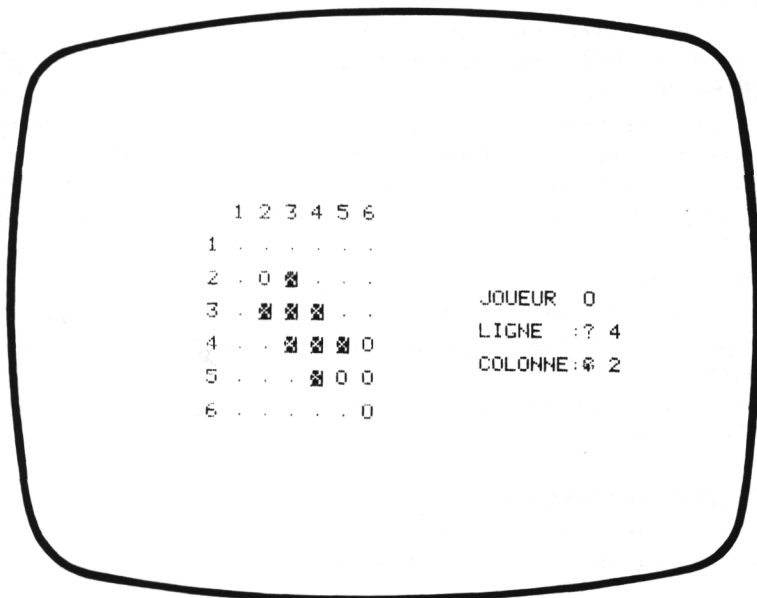
EXTENSIONS POSSIBLES :

On peut envisager de trouver des algorithmes pour tous les problèmes analogues (position de tours, déplacement d'un cavalier, etc.).

```

10 REM REINES SUR L'ECHIQUEUR
20 CLS:A#=CHR$(27):PRINT CHR$(17)
30 INPUT"TAILLE DE L'ECHIQUEUR :";S
40 DIM D(S)
50 I=1
60 D(I)=1
70 IF I=1 THEN 110
80 FOR J=1 TO I-1
90 IF D(I)=D(J) OR (ABS(D(I)-D(J))=I-J) THEN 240
100 NEXT J
110 I=I+1:IF I<=S THEN 60
120 CLS
130 FOR F=1 TO S
140 PRINT@2*F+2,0;F
150 PRINT@2,2*F;F
160 FOR K=1 TO S
170 PRINT@2*K+3,2*F;"+"
180 NEXT K
190 PRINT@2*D(F)+3,2*F;A#;"R";CHR$(255);
    PRINT@2*D(F)+4,2*F;A#"W"
200 PRINT@F+2,20;RIGHT$(STR$(D(F)),1)
210 NEXT F
220 ZAP:ZAP:ZAP:PRINT@2,22;
    "FAIRE 0 - RETURN - POUR CONTINUER";:INPUT R#
230 I=I-1
240 D(I)=D(I)+1
250 IF D(I)<=S THEN 70
260 I=I-1
270 IF I<>0 THEN 240

```



Répétitions

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu de mémoire (1 ou plusieurs joueurs).

Difficulté du jeu : moyenne.

LE JEU :

Des nombres vont apparaître un par un.

Le joueur devra à chaque coup les répéter tous, depuis le début (ce qui signifie qu'il en aura un de plus à taper à chaque fois).

Il devra essayer de tenir le plus longtemps possible. Le meilleur score est mémorisé après chaque partie, ce qui permettra éventuellement de jouer à plusieurs.

LE PROGRAMME :

La ligne 60 appelle un sous-programme (lignes 1000 à 1060) qui donne la règle du jeu.

La ligne 80 fabrique les chiffres, qui sont ensuite stockés dans une variable alphanumérique A\$.

La ligne 90 permet l'impression du dernier chiffre au hasard sur les cinquième et sixième lignes de l'écran, et la durée de l'affichage est déterminée par la ligne 100.

La ligne 120 attend la réponse du joueur, qui est comparée à la réponse correcte en ligne 130.

La ligne 160 marque les points.

La ligne 170 contrôle le meilleur score.

Les affichages de fin de partie sont effectués de la ligne 200 à la ligne 240.

POUR JOUER :

Le lancement du programme par RUN provoque l'affichage de la règle du jeu, puis d'un chiffre qui disparaît. L'appareil pose alors la question VOTRE SEQUENCE.

On lui répond en redonnant le chiffre, suivi de RETURN. Si c'est exact, on a un court instant l'affichage de EXACT. ATTENTION, puis un nouveau chiffre est affiché, qui s'efface, et de nouveau la question VOTRE SEQUENCE. On retape alors les deux chiffres déjà obtenus, suivis de RETURN.

On continue ainsi (3 chiffres, puis 4, 5, etc.) jusqu'à ce que l'on fasse une erreur, ce qui provoque l'affichage du score (et du meilleur score obtenu jusqu'à présent), suivi de la question AUTRE PARTIE (O/N) ; on tape N pour arrêter.

On peut aussi quitter le programme à tout moment, en tapant CTRL et C avant l'affichage d'un chiffre.

EXTENSIONS POSSIBLES :

On pourra réaliser des jeux analogues et plus ou moins perfectionnés en remplaçant les chiffres par des lettres, ou des couleurs, ou des sons...

```

10 REM REPETITIONS
20 CLS:B#=CHR$(27)
30 PAPER 3:INK 4
40 M=0
50 A$=""
60 GOSUB 1000
70 H=0
80 X=INT(RND(1)*10):A#=A#+RIGHT$(STR$(X),1)
90 CLS:Z=RND(1)*38+2:PRINT@Z,5;B$;"J":X:PRINT@Z,6;B$;"J":X
100 WAIT 130+RND(1)*250
110 CLS:PAPER 0:INK 2
120 PRINT@2,10;"VOTRE SEQUENCE :":INPUT R$
130 IF R$(<>)A$ THEN 190
140 PRINT:PING:PRINT"EXACT . ATTENTION"
150 WAIT 200
160 H=H+1
170 IF M<H THEN M=H
180 GOTO 80
190 CLS:PRINT:ZAP
200 PRINT"VOTRE SCORE EST DE ";H
210 PRINT:PRINT"MEILLEUR SCORE ACTUEL ";M
220 PRINT:PRINT"AUTRE PARTIE ( O/N )":INPUT R$
230 IF R$="N" THEN EXPLODE:END
240 GOTO 50
1000 CLS:PAPER 3:INK 4
1010 PRINT:PRINT"VOUS DEVEZ REPETER TOUTE LA SERIE DES"
1020 PRINT"NOMBRES APPARUS DEPUIS LE DEBUT DU JEU"
1030 PRINT:PRINT"IL Y EN A UN DE PLUS A CHAQUE FOIS"
1040 PRINT:PRINT"APPUYEZ SUR 0 - RETURN - POUR COMMENCER"
1050 INPUT R$
1060 RETURN

```

VOUS DEVEZ REPETER TOUTE LA SERIE DES
NOMBRES APPARUS DEPUIS LE DEBUT DU JEU
IL Y EN A UN DE PLUS A CHAQUE FOIS
APPUYEZ SUR 0 - RETURN - POUR COMMENCE
R

VOTRE SCORE EST DE 0
MEILLEUR SCORE ACTUEL : 0
AUTRE PARTIE (O/N) ? 0

Mots

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu de lettres (pour 1 ou plusieurs joueurs).

Difficulté du jeu : moyenne.

LE JEU :

Il est difficile d'imaginer plus simple dans son principe que ce jeu, qui se contente d'afficher dans le désordre les lettres prises dans une trentaine de mots. Mais essayez de jouer, et vous verrez que ce n'est pas si facile, la difficulté variant évidemment selon la liste de mots que vous aurez introduits en DATA au préalable.

LE PROGRAMME :

La ligne 20 permet de dimensionner la variable A\$(I) qui contiendra les mots.

La boucle I (30-50) saisit la liste des mots (inscrits en DATA aux lignes 240 à 280).

La ligne 70 choisit un mot au hasard ; la boucle I (100-150) contrôle si une lettre n'a pas déjà été tirée, et reprend les lettres dans le désordre.

La ligne 160 affiche les lettres, et la ligne 170 attend votre proposition, qui est testée en ligne 180.

Le nombre d'essais est incrémenté en ligne 190.

La fin de partie est contrôlée par la ligne 200, et le score est affiché en ligne 220.

POUR JOUER :

On lance le programme par RUN. Une série de lettres est affichée dans le désordre, et il faut essayer de retrouver le mot correspondant.

On tape le mot supposé, puis RETURN, et une nouvelle série de lettres sera affichée, si l'on répond O à la question AUTRE ESSAI (O/N).

Le score et le nombre d'essais sont affichés en fin de partie.

EXTENSIONS POSSIBLES :

On peut imaginer de minuter le temps d'apparition du mot, ou de faire défiler les lettres l'une après l'autre sur l'écran.

On peut aussi corser la difficulté en introduisant des JOKER, jouer à plusieurs, tenir le compte des points, etc.


```

10 REM MOTS
20 DIM A$(30):S=0:E=0
30 FOR I=1 TO 30
40 READ A$(I)
50 NEXT I
60 CLS
70 Z=INT(RND(1)*30+1)
80 IF A$(Z)="" THEN 70
90 R#=A$(Z):A$="" :A$(Z)="" :V#=R#
100 FOR I=1 TO LEN(R#)
110 Z=INT(RND(1)*LEN(R#)+1)
120 IF MID$(R#,Z,1)="-" THEN 110
130 A#=A#+MID$(R#,Z,1)
140 R#=LEFT$(R#,Z-1)+"-"+RIGHT$(R#,LEN(R#)-Z)
150 NEXT I
160 PRINT"LES LETTRES SONT :",A$:ZAP:ZAP:ZAP
170 PRINT@2,6;"VOTRE SOLUTION ( ? ) :";INPUT S#
180 IF S#=V# THEN S=S+1
190 E=E+1
200 PRINT@2,12;"AUTRE ESSAI ( O/N ) :";INPUT R#
210 IF LEFT$(R#,1)="O" THEN 60
220 PRINT@2,15;"VOTRE SCORE :";S;" SUR ";E;" ESSAIS"
230 EXPLODE : END
240 DATA JAMBON,ENERGUMENE,AFFILIE,ERUDIT,INSIGNIFIANT,REALISTE
250 DATA IMPRESSIONNANT,ALLECHANT,EPIQUE,INTREPIDE,COUARD,ORDINATEUR
260 DATA TREVE,BANDEROLLE,SENAT,ACTUEL,CONSACRE,DIEDRE
270 DATA LIVRE,TERMINE,SIGNATURE,MALODORANT,AIMANT,ARBITRAIRE
280 DATA CHALEUR,OPPORTUN,PELURE,ALTRUISTE,RAMONEUR,NOVICE

```

LES LETTRES SONT :EEULPR

VOTRE SOLUTION : ? PELURE

AUTRE ESSAI (O/N) : ? N

VOTRE SCORE : 1 SUR 2 ESSAIS

Permutations

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion.

Difficulté du jeu : moyenne.

LE JEU :

L'ORIC-ATMOS vous demande d'abord le nombre de lettres que vous désirez permuter (pas plus de 12). A partir de là, il va vous afficher le nombre correspondant de lettres prises dans le début de l'alphabet, d'abord dans le désordre (DEBUT) puis dans l'ordre (FIN).

Vous devez réussir à remettre dans l'ordre les lettres données au début, en utilisant des permutations sur les n premières lettres à partir de la gauche, avec le moins d'essais possible.

LE PROGRAMME :

La ligne 40 permet de choisir le nombre de lettres, et la ligne 50 contrôle que l'on n'en demande pas plus de 12.

La ligne 80 réalise le brouillage, en tirant au hasard la nouvelle position de chaque lettre. Les lettres sont ensuite ajoutées au fur et à mesure de leur tirage, dans une variable B\$ (ligne 100), supprimées dans A\$ et re-placées en C\$, dans le bon ordre (lignes 110-120).

La ligne 140 contrôle le nombre de lettres de B\$.

Les affichages sont effectués aux lignes 160, 170 et 180.

Le retournement est fait, lettre par lettre, dans la boucle I (210-230) ; le résultat est placé dans B\$ en ligne 240, affiché par la ligne 250 et comparé au résultat final en ligne 260.

Le nombre d'essais sera affiché en fin de partie, grâce à la ligne 270.

POUR JOUER :

Après avoir tapé RUN, on obtient l'affichage de la question NOMBRE DE LETTRES (DE 1 A 12) ; on répond en frappant N, suivi de RETURN. L'affichage suivant présente l'ordre initial (DEBUT) et l'ordre à obtenir (FIN).

On joue en indiquant le nombre de lettres à permuter à partir de la gauche, par exemple : sur un affichage initial de GEABDFC, si l'on tape 3 et RETURN, on obtiendra AEGBDFC.

On poursuivra jusqu'à l'obtention de l'ordre désiré.

Le score sera affiché en fin de partie ; ainsi qu'une question AUTRE PARTIE (O/N) ; répondre O si l'on désire faire une autre partie.

EXTENSIONS POSSIBLES :

On pourra utiliser le même type de jeu pour, par exemple, reconstituer un dessin caché dont les diverses parties seraient repérées par des lettres. On peut aussi réaliser des permutations plus complexes (par exemple le fait de permuter deux lettres permute les deux voisines de la lettre la plus à gauche, etc.).

```

10 REM PERMUTATIONS
20 D$=CHR$(27)
30 CLS
40 PRINT:INPUT"OMBRE DE LETTRES ( DE 1 A 12 ) :";N
50 IF N>12 THEN 30
60 A$="ABCDEFGHJKLM":C$="          ":B$=""
70 CLS:PAPER 6:INK 4
80 X=INT(RND(1)*N+1)
90 IF MID$(A$,X,1)=" " THEN 80
100 B$=MID$(A$,X,1)+B$
110 C$=LEFT$(C$,X-1)+MID$(A$,X,1)+RIGHT$(C$,12-X)
120 A$=LEFT$(A$,X-1)+" "+RIGHT$(A$,12-X)
130 S=1
140 IF LEN(B$)<>N THEN 80
150 C$=LEFT$(C$,LEN(B$))
160 PRINT@2,4;"DEBUT  ";D$;"J";B$:PRINT@9,3;D$;"J";B$
170 PRINT@2,6;"FIN    ";D$;"J";C$:PRINT@9,5;D$;"J";C$
180 PRINT@2,10;"PERMUTATION NUMERO ";S:PRINT:PING:
    INPUT"VOTRE QUANTITE ";L
190 IF L>N THEN 180
200 S=S+1:A$=""
210 FOR I=1 TO L
220 A$=A$+MID$(B$,L-I+1,1)
230 NEXT I
240 B$=A$+RIGHT$(B$,LEN(B$)-L)
250 ZAP:ZAP:ZAP:PRINT@2,15;"RESULTAT  ";D$;"J";B$:
    PRINT@12,16;D$;"J";B$
260 IF B$<>C$ THEN 180
270 EXPLODE:PRINT:PRINT"TROUVE EN ";S-1;" ESSAIS"
280 PRINT:INPUT"AUTRE PARTIE ( O/N ):";R$
290 IF LEFT$(R$,1)="0" THEN 30

```

```

DEBUT  : BCDE
FIN    : ABCDE

PERMUTATION NUMERO  7
VOTRE QUANTITE ? 5

RESULTAT : ABCDE

TROUVE EN  7  ESSAIS
AUTRE PARTIE ( O/N ): ? N

```

Bridge

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : variable, selon le problème posé.

LE JEU :

Vous ne trouverez pas, dans ce programme, un partenaire pour vos bridges mais, plus exactement, une simulation pour l'affichage d'une donne ; cet affichage est réalisé jeu par jeu. On peut, à partir de là, se poser toutes sortes de problèmes : annonces, jeu de la carte, etc.

LE PROGRAMME :

Les lignes 50 et 60 définissent un jeu de 52 cartes.

La variable B\$, en ligne 70, permettra de donner en abrégé le nom des couleurs et aussi d'améliorer la désignation de certaines cartes (10, valet, dame, roi et as).

La boucle J (80-220) réalise le tirage de quatre jeux de 13 cartes.

La boucle I (90-190) réalise tirage et impression. La carte est choisie parmi les 52 disponibles, en ligne 100.

La ligne 110 calcule la couleur de la carte tirée.

La ligne 120 vérifie que cette carte n'a pas déjà été tirée.

Les lignes 130 et 140 s'occupent de l'impression complémentaire dans le cas où l'on tire un 10, valet, une dame, un roi ou un as.

L'impression est réalisée par les lignes 150, 160 et 170.

La ligne 180 permet d'éliminer les cartes déjà tirées, et la ligne 200 réalise un arrêt entre chaque affichage.

POUR JOUER :

On lance le programme par RUN.

Une première série de 13 cartes est affichée verticalement. Il suffira d'appuyer sur les touches Ø et RETURN pour obtenir la deuxième série de 13 cartes. On procédera de même pour la troisième et la quatrième.

EXTENSIONS POSSIBLES :

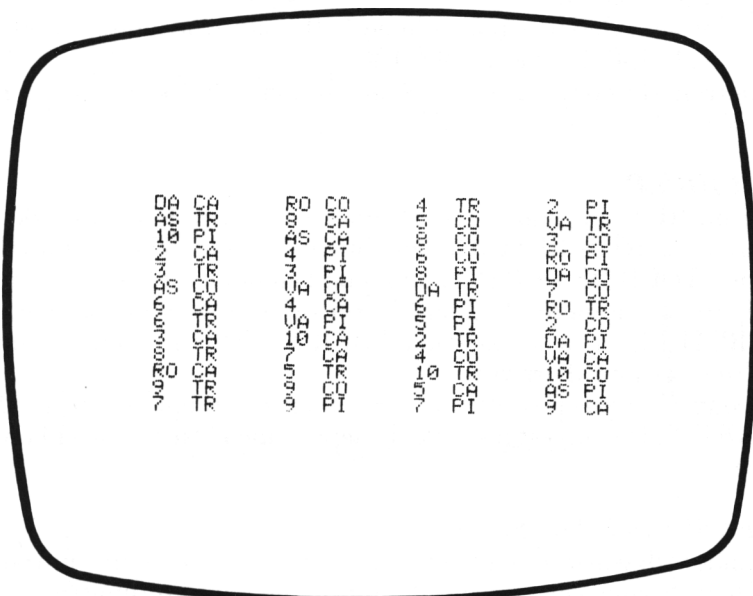
On peut imaginer de représenter les 13 cartes sous une forme plus conventionnelle.

On peut aussi ajouter la possibilité de faire des annonces (et de les comparer à celles de l'ORIC-ATMOS, par exemple), ou introduire des possibilités de jeu de la carte, y compris pour l'ORIC-ATMOS.

```

10 REM BRIDGE
20 C$=CHR$(27):C$(1)=C$+"W":C$(3)=C$+"0":
   C$(5)=C$+"S":C$(7)=C$+"R":C$=C$+"V"
30 CLS
40 PAPER 6:INK 0
50 R$="234567891VDRA"
60 R$=R$+R$+R$+R$
70 B$="TRCACOPI0AAS"
80 FOR J=1 TO 4
90 FOR I=1 TO 13
100 X=INT(RND(1)*LEN(R$)+1)
110 S=2*INT((X-1)/13)+1
120 IF MID$(R$,X,1)="0" THEN 100
130 T=X-13*INT(X/13)
140 T=T-13*(T=0)
150 PRINT@10#J-8,I;C$(S);MID$(R$,X,1);
160 IF T>=9 THEN PRINT MID$(B$,T,1)
170 PRINT@10#J-4,I;MID$(B$,S,2);C$
180 R$=LEFT$(R$,X-1)+"0"+RIGHT$(R$,LEN(R$)-X):ZAP
190 NEXT I
200 PRINT@2,18;"FAIRE 0 - RETURN -
   POUR CONTINUER":INPUT S$
210 PRINT@2,18;"
220 NEXT J

```



Hanoi

REFERENCES :

Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :

Il est présenté, ici, dans une version pour un joueur. On dispose d'un certain nombre de disques (8 au plus) et l'on doit les transférer de la tour numéro 1 à la tour numéro 3, en respectant une seule règle : à aucun moment, un disque ne peut être recouvert par un disque de diamètre supérieur.

Le nombre de coups n'est pas comptabilisé par ce programme mais, par contre, les coups illicites seront refusés.

LE PROGRAMME :

La variable D\$, en ligne 40, servira à dessiner les disques.

Le nombre de disques est entré en ligne 50 et refusé s'il y a plus de 8 disques.

La boucle I (70-90) permet de fabriquer une variable A(1) qui contient l'état de la première pile sous forme numérique (en ligne 30, trois variables sont prévues pour conserver l'état des trois piles).

Les socles sont dessinés par la ligne 110 et les trois piles par les boucles emboîtées I (130-190) et J (140-180).

Le contrôle du coup est effectué par la ligne 260 et l'état des piles est modifié par les lignes 270 et 280.

POUR JOUER :

Après un RUN qui lance le programme, on entre le nombre de disques demandés, suivi de RETURN. On obtient le dessin de l'état initial et, à la question TOUR DE DEPART, on répond en donnant le numéro de la tour de départ. Ce numéro (1, 2 ou 3) sera refusé si la tour est vide. Puis on tape RETURN et, à la question TOUR D'ARRIVEE, on répond par le numéro de la tour d'arrivée.

Si le coup n'est pas possible, on sera obligé de recommencer à TOUR DE DEPART. Si le coup est possible, le déplacement est réalisé et l'on recommence.

EXTENSIONS POSSIBLES :

On pourra dessiner des disques de différentes couleurs, introduire un compteur de coups et, surtout, définir un jeu pour l'ordinateur.

On peut aussi essayer de compliquer le jeu avec deux tiges de plus et deux joueurs, chacun ayant des disques de couleur différente, par exemple.

```

10 REM HANOI
20 CLS:E#=CHR$(27):PAPER 7:INK 0
30 DIM A(3):A(1)=9:A(2)=9:A(3)=9
40 D$="":FOR I=1 TO 9:D#=D#+CHR$(255):NEXT I
50 INPUT"NOMBRE DE TOURS ( MOINS DE 8 ):";N
60 IF N>=8 THEN 50
70 FOR I=0 TO N-1
80 A(1)=VAL(STR$(A(1))+STR$(N-I))
90 NEXT I
100 CLS
110 PLOT1,14,CHR$(0):PRINT@2,14;D$:PRINT@14,14;D$:
PRINT@26,14;D$
120 PRINT@6,14;"1":PRINT@18,14;"2":PRINT@30,14;"3"
130 FOR I=0 TO N-1
140 FOR J=1 TO 3
150 B#=RIGHT$(STR$(A(J)),LEN(STR$(A(J)))-1)
160 S#=LEFT$(D$,VAL(MID$(B#,I+2,1)))
170 IF LEN(B#)>I+1 THEN PRINT@12*J-10,13-I;E$;"B";S#
180 NEXT J
190 NEXT I
200 PRINT@2,16;"TOUR DE DEPART :";INPUT D
210 IF D>3 OR D<1 THEN 200
220 IF A(D)=9 THEN 200
230 PRINT:INPUT"TOUR D'ARRIVEE :";A
240 B#=RIGHT$(STR$(A(D)),LEN(STR$(A(D)))-1)
250 C#=RIGHT$(STR$(A(A)),LEN(STR$(A(A)))-1)
260 IF MID$(B#,LEN(B#),1)>MID$(C#,LEN(C#),1)
THEN 200
270 A(A)=VAL(C#+MID$(B#,LEN(B#),1))
280 A(D)=VAL(LEFT$(B#,LEN(B#)-1))
290 ZAP:ZAP:ZAP:GOTO 100

```



TOUR DE DEPART : ? 3
TOUR D'ARRIVEE : ? 2

Losanges

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : interlude.

Difficulté du jeu : aucune.

LE JEU :

Le programme va progressivement construire sur l'écran un losange, puis il va l'effacer progressivement, le reconstruire, et ainsi de suite.

Vous pourrez envisager de l'utiliser pour vous endormir, si les moutons ne vous suffisent plus.

Une version, très légèrement modifiée, vous est également proposée, qui vous permettra de faire apparaître des motifs aléatoires après la construction de chaque losange.

LE PROGRAMME :

Une première boucle K (50-130) définit les dimensions des tracés successifs.

La deuxième boucle I (60-120) trace des bandes de points, en utilisant les instructions CURSET (lignes 80 à 110).

Si la variable S (ligne 70) est à 1, alors il y aura effacement (on utilise CURSET, mais en se servant des couleurs du fond pour effacer).

On obtiendra dans le deuxième programme un effacement aléatoire en modifiant la ligne 70.

POUR JOUER :

On commence l'exécution du programme avec RUN, et on regarde.

On peut arrêter à tout moment en tapant CTRL et C, et on revient au mode TEXT.

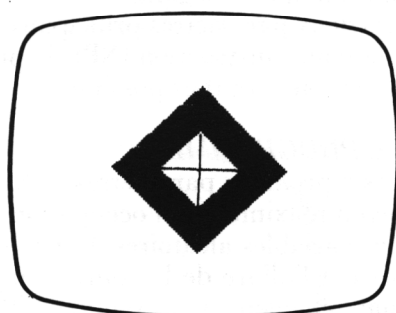
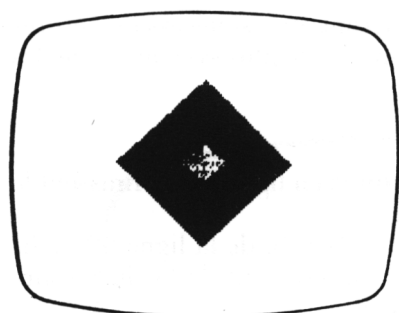
EXTENSIONS POSSIBLES :

On pourra introduire le son, utiliser l'instruction DRAW qui permettra un tracé beaucoup plus rapide, imaginer sur des principes analogues divers programmes (kaléidoscope, tapisserie...).


```

10 REM LOSANGES
20 HIRES
30 PAPER 3: INK 1
40 CLS: S=0
50 FOR K=0 TO 40
60 FOR I=0 TO K
70 IF S<>0 THEN ON INT(RND(1)*5+1) GOTO 160,170,180,190,200
80 CURSET 120-K+I,100-I,1
90 CURSET 120-K+I,100+I,1
100 CURSET 120+K-I,100+I,1
110 CURSET 120+K-I,100-I,1
120 NEXT I
130 NEXT K
140 S=NOT S
150 GOTO 50
160 CURSET 120-K+I,100-I,2
170 CURSET 120-K+I,100+I,2
180 CURSET 120+K-I,100+I,2
190 CURSET 120+K-I,100-I,2
200 GOTO 120

```



```

10 REM LOSANGES (SANS ALEAS)
20 HIRES
30 PAPER 6: INK 4
40 CLS: S=0
50 FOR K=0 TO 40
60 FOR I=0 TO K
70 IF S<>0 THEN 160
80 CURSET 120-K+I,100-I,1
90 CURSET 120-K+I,100+I,1
100 CURSET 120+K-I,100+I,1
110 CURSET 120+K-I,100-I,1
120 NEXT I
130 NEXT K
140 S=NOT S
150 GOTO 50
160 CURSET 120-K+I,100-I,2
170 CURSET 120-K+I,100+I,2
180 CURSET 120+K-I,100+I,2
190 CURSET 120+K-I,100-I,2
200 GOTO 120

```

Mathieu

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : dessin aléatoire.

Difficulté du jeu : aucune.

LE JEU :

On utilise la définition d'une courbe en coordonnées polaires et la fonction CURSET pour obtenir sur l'écran des dessins aléatoires (de type fleur), plus ou moins élégants.

Les deux paramètres principaux A et B pourront être, si nécessaire, entrés à la main (instruction INPUT) pour faire une étude plus systématique des différentes courbes possibles.

LE PROGRAMME :

Les 3 premiers paramètres Z, K et T, introduits en ligne 30, définissent le rayon maximum et l'occupation de l'écran.

Les variables aléatoires A et B, choisies entre 0 et 3, de la ligne 60, définissent l'allure de la courbe ; il pourra être intéressant de les introduire manuellement (par exemple : 60 INPUT A, B).

La boucle H (70-120) réalise le tracé, point par point. Le rayon R est calculé en ligne 80, et le passage en coordonnées cartésiennes est effectué aux lignes 90 et 100.

L'affichage des points est réalisé en ligne 110.

Les lignes 130 et 140 impriment un message de fin.

La ligne 160 permet de refaire un nouveau dessin.

POUR JOUER :

On lance le programme en tapant RUN.

Le dessin sera ensuite exécuté automatiquement.

Le message "c'est fini" signale la fin du tracé.

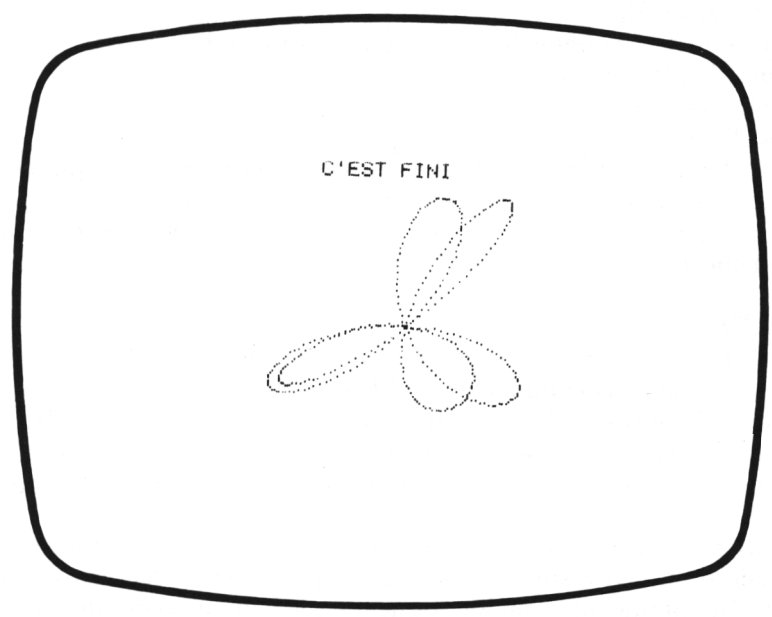
Si on veut un nouveau tracé, on tape 0.

Pour arrêter, faire CTRL et C, et TEXT.

EXTENSIONS POSSIBLES :

On pourra essayer d'utiliser d'autres fonctions pour réaliser des tracés encore plus extraordinaires (par exemple avec simulation de la troisième dimension).

```
10 REM MATHIEU
20 HIRES
30 Z=7:K=9:T=3
40 CLS
50 C=INT(RND(1)*8):F=7-C:PAPER C:INK F
60 A=3*RND(1):B=3*RND(1)
70 FOR H=0 TO 6.28 STEP 3.14/180
80 R=Z*SIN(H*T)
90 X=120+INT(K*R*COS(A*H))
100 Y=100+INT(K*R*SIN(B*H))
110 CURSET X,Y,1
120 NEXT H
130 A$="C'EST FINI"
140 CURSET 80,20,3:FOR I=1 TO 10:CHAR ASC(
    MID$(A$,I,1)),0,1:CURMOV6,0,3:NEXT I
150 PING
160 INPUT"FAIRE 0 - RETURN - POUR CONTINUER":R$
170 IF LEFT$(R$,1)="0" THEN 20
```



Le loup et les agneaux

REFERENCES :

Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :

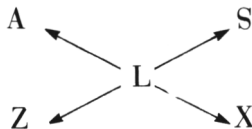
C'est un jeu assez connu qui peut être pratiqué indifféremment sur un damier 8 sur 8, ou 10 sur 10. Il est présenté, ici, dans une version pour un damier de 8 cases ; l'un des joueurs choisit le loup qui se déplace suivant les diagonales, en restant sur les cases noires, d'une case à chaque coup, et qui peut avancer ou reculer. Les agneaux se déplacent également sur les cases noires, en diagonale, mais n'ont pas le droit de reculer. Les cases déjà occupées sont interdites (pas de prises), et il n'y a pas de saut. Le loup gagne s'il réussit à arriver sur la ligne du bas avant que tous les agneaux n'occupent celle du haut ; les agneaux gagnent dans ce dernier cas, ou s'ils arrivent à bloquer le loup.

LE PROGRAMME :

Une variable A\$, en ligne 30 contient les positions de départ.
La boucle I (80-130) dessine le damier, et les lignes 150 à 190 placent le loup et les agneaux.
La ligne 200 gère le changement de joueur.
La ligne 220 affiche JEU DU LOUP si c'est le joueur qui manipule le loup qui a le trait, et la ligne 230 JEU DES AGNEAUX dans le cas contraire.
Les lignes 250 à 260 et 280 à 300 analysent la réponse, en fonction du joueur concerné.

POUR JOUER :

RUN déclenche l'arrêt sur l'attente d'une entrée : Ø et RETURN si l'on veut que les agneaux commencent, et 1 pour le loup. Après quoi, le damier s'affiche et la phrase JEU DU LOUP (A, S, Z, X) pour le loup et JEU DES AGNEAUX (ND et NG) pour les agneaux.
Les agneaux répondent par un numéro, suivi de D ou G (pour droite ou gauche).
Ainsi 2D provoque le déplacement de l'agneau numéro 2 qui monte en diagonale d'une case sur la droite. Le déplacement effectué est affiché, puis on obtient une question identique pour le loup qui répondra en tapant A, S, Z ou X, suivant la direction choisie :

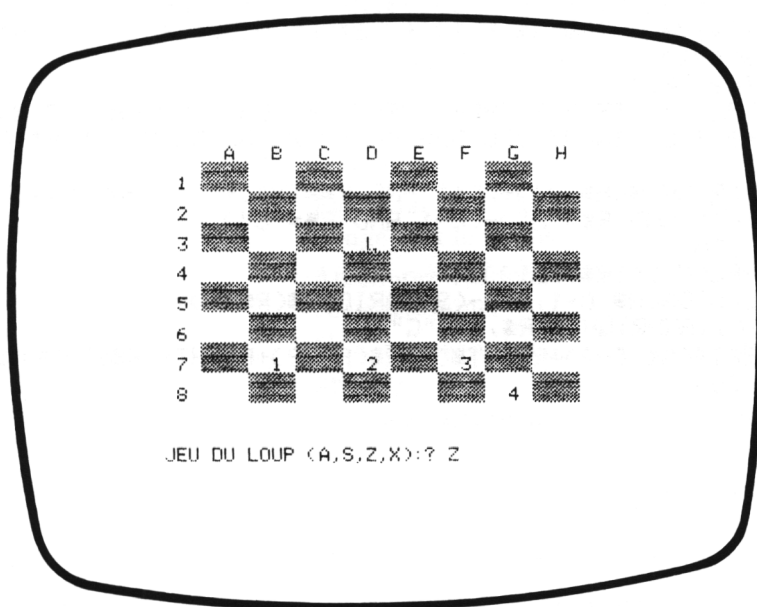


On devra veiller à ne pas sortir de l'échiquier et à ne pas venir sur l'adversaire, aucun contrôle n'ayant été prévu.

EXTENSIONS POSSIBLES :

On pourra prévoir un repérage des cases, un graphisme plus soigné, des contrôles de non dépassement des limites du terrain, de non venue sur une case déjà occupée...

On pourra aussi envisager de jouer sur un damier de 10 cases sur 10 cases, ce qui augmente légèrement la difficulté du jeu pour les agneaux. Mais l'extension la plus intéressante à réaliser sera certainement la réalisation d'un programme permettant à l'ORIC-ATMOS de se comporter en adversaire honorable, soit avec le loup, soit avec les agneaux (et, dans ce cas, il sera possible de définir une stratégie gagnante pour l'ORIC-ATMOS.



```

10 REM LE LOUP ET LES AGNEAUX
20 INPUT"AGNEAUX (0) OU LOUP (1) :";S
30 A$="0183858714":D$=CHR$(27):PRINT CHR$(17)
40 CLS:PAPER 6:INK 4
50 PRINT"  A  B  C  D  E  F  G  H"
60 FOR I=1 TO 8:PRINT@2,2*I;I:NEXT I
70 G$=CHR$(255):F$=G$+G$+G$+G$:F$=D$+"A"+F$
80 FOR I=1 TO 8
90 FOR J=1 TO 8
100 T=(I+J)/2:PRINT@4*J,2*I;
110 IF T=INT(T) THEN PRINT F$:PRINT@4*J,2*I-1;F$
    ELSE PRINT@4*J+1,2*I;"
120 NEXT J
130 NEXT I
140 FOR I=1 TO 9:PLOT0,I+17,CHR$(19):NEXT I
150 PRINT@4*VAL(MID$(A$,10,1))+3,2*VAL(MID$(A$,9,1));"L"
160 PRINT@4*VAL(MID$(A$,2,1))+3,2*VAL(MID$(A$,1,1));"1"
170 PRINT@4*VAL(MID$(A$,4,1))+3,2*VAL(MID$(A$,3,1));"2"
180 PRINT@4*VAL(MID$(A$,6,1))+3,2*VAL(MID$(A$,5,1));"3"
190 PRINT@4*VAL(MID$(A$,8,1))+3,2*VAL(MID$(A$,7,1));"4"
200 S=ABS(S-1)
210 PRINT@2,20;"
220 PRINT@2,20;:IF S=0 THEN INPUT"JEU DU LOUP (A,S,Z,X):";R$
230 IF S=1 THEN INPUT"JEU DES AGNEAUX (ND OU NG):";R$:GOTO 280
240 C=10
250 A=VAL(RIGHT$(A$,2))+(C+1 AND R$="X")+(C-1 AND R$="Z")
260 A=A-(C+1 AND R$="A")-(C-1 AND R$="S")
270 GOTO 300
280 C=2*VAL(MID$(R$,1,1))
290 A=VAL(MID$(A$,C-1,2))-(9ANDRIGHT$(R$,1)="
    "D")-(11ANDRIGHT$(R$,1)="G")
300 A$=LEFT$(A$,C-2)+RIGHT$(STR$(A),2)+RIGHT$(A$,LEN(A$)-C)
310 GOTO 70

```

Niveau 4

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Tirage de dés	Suffisamment souple pour être utilisé partout.
Jeu de loto	Gagnez sans problème, au prochain tirage.
La bombe	Saurez-vous la découvrir à temps ? (1 joueur).
Ardoise magique	Mettez au point vos créations graphiques, sans papier ni crayon.
Affichage publicitaire	Si vous ne savez pas quoi mettre dans votre vitrine.
Evasion	Il vous faudra beaucoup de chance pour réussir.
Chapeaux	Soyez très observateur, et vous réussirez (2 joueurs).
Combat aérien	Pas si facile de descendre un appareil qui bouge tout le temps.
Dessin de dés	Améliorez vos jeux de dés en incorporant ce programme.
Le jeu du missile	Une mission bien périlleuse (un seul joueur).
Le singe	Un jeu de lettres pour 2 joueurs, où il faut de bons réflexes.
Le jeu du 15	Un jeu de réflexion utilisant des chiffres (2 joueurs).
Adresse	Jeu de réflexe pour 1 joueur, très visuel.
Sous-marin	Un classique des jeux vidéos (1 joueur).
Zombies	Un peu farfelu, mais très visuel (1 joueur).
Diago	Etes-vous vraiment sûr de vos réflexes ?
Décomaque	Pour vous éviter de sécher trop longtemps sur "Nicomaque".
Chasse au sous-marin	Il vous faudra pas mal de réflexion, et beaucoup de chance.
Clavier	Apprenez à taper à la machine en vous amusant.
Le lézard d'Alice	Baucoup de hasard, mais très spectaculaire.
Calculatrice	Un programme de simulation très réaliste.

Les techniques d'animation élémentaire Les caractères graphiques

SAISIE AU VOL D'UN CARACTERE FRAPPE AU CLAVIER : KEY\$

Le principe d'un jeu graphique, en animation, est de laisser le jeu se dérouler tout seul, d'une certaine manière, pendant un certain temps, tant qu'aucune intervention n'a été faite par le joueur.

Il faut donc disposer de la possibilité d'observer s'il y a eu ou non intervention du joueur, sans pour cela interrompre le déroulement du programme.

Dans le Basic de l'ORIC-ATMOS, cette propriété est offerte, par la **fonction KEY\$** qui effectue une structation des touches du clavier et enregistre le nom de la dernière touche frappée.

On pourra utiliser KEY\$ de deux façons :

- soit dans une boucle (réalisation d'un mouvement). On saisit KEY\$ et on le teste pour une éventuelle sortie de la boucle.
- soit dans une attente : tant que KEY\$ est vide (on n'a rien frappé) on attend, et l'on sort dès qu'une touche a été frappée ; on peut aussi, pour cette fonction d'attente, utiliser GETA\$.

Il sera quelquefois nécessaire de taper plusieurs fois la touche pour qu'elle soit prise en compte.

DEPLACEMENTS D'OBJETS DE BAS EN HAUT SUR L'ECRAN : SCROLL

Il est difficile, si l'on travaille en Basic, de gérer le déplacement simultané de plusieurs objets sur l'écran (sinon on obtient vite une animation saccadée, à cause d'un temps de déplacement prohibitif).

L'ORIC-ATMOS réalise automatiquement un "**Scrolling**" (c'est-à-dire un déplacement de l'ensemble de l'image) vers le haut, lorsque l'on écrit sur la dernière ligne en bas de l'écran.

Cette propriété a été mise à profit dans plusieurs jeux : cela permet en effet, de déplacer d'une ligne vers le haut les divers objets (textes ou graphiques) présents sur l'écran à un moment donné. La ligne du haut est bien évidemment perdue.

Si un objet doit rester immobile sur l'écran, il sera nécessaire de l'effacer dans sa nouvelle position, et de le replacer à son emplacement initial.

LES CARACTERES GRAPHIQUES DE L'ORIC-ATMOS

L'ORIC-ATMOS dispose d'origine d'un jeu de caractères graphiques, de type TELETEL, accessibles directement en mode **LORES 1**, mais disponibles en mode **TEXT**, dans une instruction **PRINT**, en les faisant précéder d'un **CHR\$ (9)**.

Cette possibilité est utilisée pour améliorer l'aspect graphique de plusieurs programmes.

Tirage de dés

REFERENCES :

Difficulté de programmation : aucune.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : aucune.

LE JEU :

Le programme proposé ne réalise pas un jeu en lui-même, mais pourra être utilisé en sous-programme pour de nombreux jeux.

Il offre simplement la possibilité de lancer de 1 à 9 dés, par simple pression d'une touche. Les dés sont ensuite affichés avec un graphisme très simplifié.

LE PROGRAMME :

La ligne 40 saisit le caractère frappé au clavier et la ligne 80 le transforme en nombre (si ce n'est pas possible, le programme refuse l'entrée).

Une variable D non dimensionnée reçoit la valeur indiquée par chaque dé, (ligne 100).

Une boucle I en ligne 90 est utilisée pour dessiner un fond, et une autre boucle I (90-140) sert au tirage (tirage en ligne 100 et impression en ligne 120).

La ligne 170 offre la possibilité de refaire un tirage ; elle n'est pas indispensable.

POUR JOUER :

On lance le programme par l'ordre RUN et on obtient l'affichage **NOMBRE DE DES A LANCER** auquel on répond en frappant un chiffre.

Le résultat est alors automatiquement affiché ; il suffit de presser une touche quelconque pour recommencer le programme.

Pour arrêter, il est nécessaire de taper F.

EXTENSIONS POSSIBLES :

On pourra, bien sûr, utiliser le canevas fourni par ce programme pour la réalisation de tout jeu se servant de dés (421, YAM'S, CRAPS, etc.).

```
10 REM TIRAGE DE DES
20 CLS:PAPER 6:INK 4:E#=CHR$(27)+"J"
30 PRINT@2,2;"NOMBRE DE DES A LANCER :";
40 GET A#
50 FOR I=1 TO 4:PLOT0,7+I,CHR$(18):PLOT1,7+I,CHR$(0):NEXT I
60 IF A#<"1" OR A#>"9" THEN 30
70 PRINT A#
80 N=VAL(A#)
90 FOR I=1 TO N
100 D(I)=INT(RND(1)*6+1)
110 FOR J=1 TO 2
120 PRINT@4*I-2,8+J;E#;RIGHT$(STR$(D(I)),1)
130 NEXT J
140 NEXT I
150 PING:GET A#
160 IF A#="F" THEN END
170 GOTO 20
```

NOMBRE DE DES A LANCER :5-

4 5 2 6 6

Jeu de loto

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : hasard (plusieurs joueurs).

Difficulté du jeu : facile.

LE JEU :

il s'agit de réaliser un tirage complet de loto, c'est-à-dire d'obtenir 6 numéros, compris entre 1 et 49, plus un numéro complémentaire, tous différents, qui pourraient être, par exemple, utilisés pour remplir votre prochaine grille de loto.

LE PROGRAMME :

On utilisera un tableau N(7), dimensionné en ligne 40, pour enregistrer les résultats.

Le tirage des nombres est effectué dans une boucle I (60-120), une deuxième boucle J (90-110) étant utilisée pour contrôler qu'il n'y ait pas deux tirages identiques.

Le tirage (ou le retraitage) est effectué en ligne 70.

La boucle I (160-190) affiche le résultat, et les lignes 200 à 220 affichent le numéro complémentaire.

Les lignes 230 et 240 s'occupent du cas où l'on voudrait refaire une autre partie.

POUR JOUER :

Il suffit de taper RUN pour lancer le programme et obtenir l'affichage d'un premier loto.

La question **VOULEZ-VOUS UN AUTRE TIRAGE (O/N)** s'affiche en bas de l'écran ; on répondra O si l'on désire continuer.

EXTENSIONS POSSIBLES :

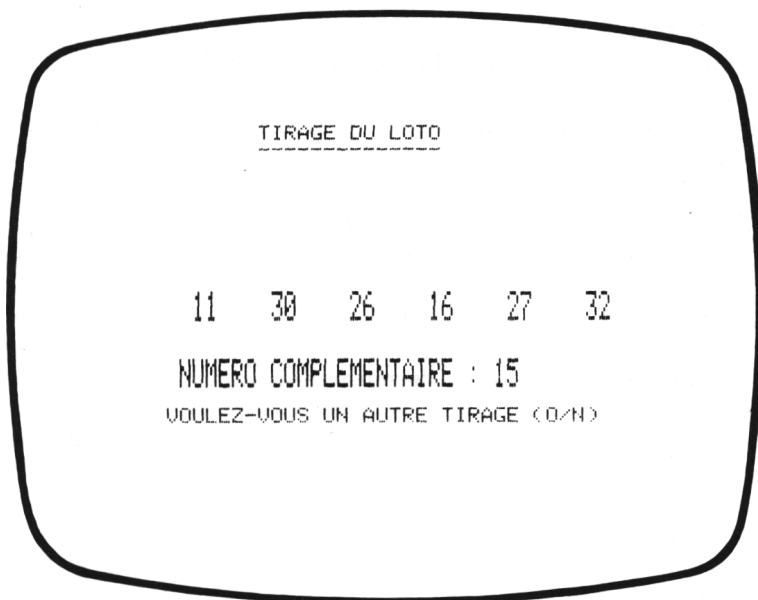
On peut imaginer un programme qui pourrait tenir compte, soit de ses propres résultats, soit de résultats entrés par un joueur, pour améliorer le tirage, en tenant compte des probabilités de sortir un numéro donné.

On peut aussi envisager un programme qui réaliserait des cartons pour le jeu de loto plus usuel, tel qu'il est encore couramment pratiqué dans le midi de la France.

```

10 REM JEU DE LOTO
20 PAPER 0:INK 6
30 D$=CHR$(27):E$=D$+"J"
40 DIM N(7)
50 CLS
60 FOR I=1 TO 7
70 N(I)=INT(RND(1)*49+1)
80 IF I=1 THEN 120
90 FOR J=1 TO I-1
100 IF N(I)=N(J) THEN 70
110 NEXT J
120 NEXT I
130 PRINT@8,3;D$;"L";"TIRAGE DU LOTO"
140 PRINT@8,4;D$;"L";"-----"
150 PLOT1,13,3:PLOT1,14,3
160 FOR I=1 TO 6
170 PRINT@6*I-4,13;E$;N(I)
180 PRINT@6*I-4,14;E$;N(I)
190 NEXT I
200 A$="NUMERO COMPLEMENTAIRE :"+STR$(N(7))
210 PLOT1,17,1:PLOT1,18,1
220 PRINT@2,17;E$;A$:PRINT@2,18;E$;A$
230 PING:PRINT:PRINT"VOULEZ-VOUS UN AUTRE TIRAGE (O/N)"
240 GET A$:IF A$="O"THEN 50

```



La bombe

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

Vous êtes sensé trouver, à l'aide d'un détecteur spécialisé, un colis radioactif piégé disposé par un voyageur mal intentionné, dans une chambre d'hôtel. Seul problème, mais de taille : l'hôtel comporte un million de chambres (100*100*100).

Votre détecteur affiche, en permanence, un signal qui vous renseigne sur la proximité de la bombe, et le temps qui vous reste est affiché en permanence, décompté de 10 en 10.

Pourrez-vous découvrir la bombe avant l'explosion ?

LE PROGRAMME :

Une variable numérique R, non dimensionnée, contient les trois coordonnées définissant la position du joueur.

La position de la chambre contenant le colis piégé est définie par un tirage aléatoire de 3 variables A, B, C, en ligne 50.

La boucle I (70-110) dessine le détecteur.

Le signal de proximité est calculé en lignes 120-130 et la ligne 140 permet l'affichage des coordonnées de la chambre à découvrir, dans le cas où l'on a perdu.

La ligne 150 permet l'affichage, à la fois du signal et du temps restant à courir.

La ligne 160 réalise le décompte du temps.

La ligne 170 contrôle si l'on a trouvé ou non.

La boucle I (210-250) permet d'entrer les nouvelles coordonnées, en utilisant le sous-programme placé aux lignes 300 à 340.

POUR JOUER :

Après le lancement du programme par RUN, le signal et le temps (en principe 200) sont affichés.

Le joueur est alors en position $x=0$, $y=0$, $z=0$, et on devra introduire successivement les trois coordonnées x, y, z de la chambre à explorer.

On fera, par exemple, 13 puis RETURN, 45 puis RETURN, etc.

Le nouveau signal est alors affiché, et le temps est décrétement de 10.

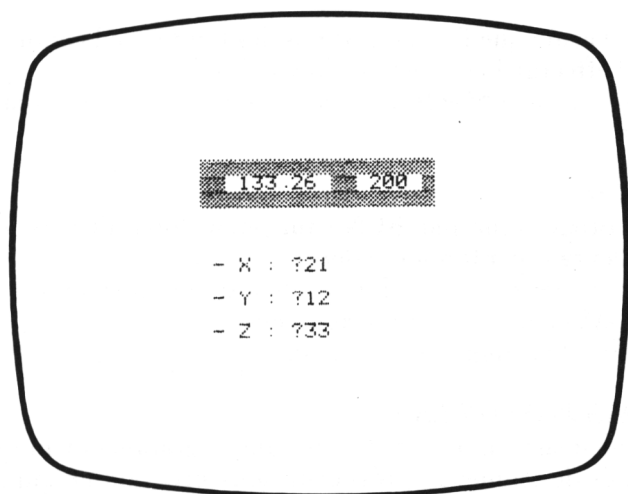
On continuera ainsi, soit jusqu'au bon résultat (arrêt du jeu et fin de programme), soit jusqu'à l'apparition de 3 nombres (x,y,z) sans sortie du programme, ce qui signifie alors que l'on a perdu.

On peut taper F en entrée si l'on veut arrêter avant la fin.

EXTENSIONS POSSIBLES :

Il sera sans doute indispensable d'améliorer le graphisme, d'ajouter une explosion (sonore et visuelle) pour symboliser l'éclatement de la bombe, d'afficher symboliquement la position dans l'hôtel, etc.

```
10 REM LA BOMBE
20 PAPER 6:INK 4:D#=CHR$(27):E#=CHR$(126)
30 FOR I=1 TO 3:PLOT1,3+I,1:NEXT I
40 L=100:T=200
50 A=INT(RND(1)*L):B=INT(RND(1)*L):C=INT(RND(1)*L)
60 CLS
70 FOR I=0 TO 17
80 PRINT@I+2,4;E$
90 PRINT@I+2,5;E$
100 PRINT@I+2,6;E$
110 NEXT I
120 S=ABS((A/L+B+C*L)-(R(1)/L+R(2)+R(3)*L))
130 S#=LEFT$(STR$(S),7):S=VAL(S#)
140 IF T=0 THEN PRINT"PERDU .
    LA SOLUTION :";A;" ";B;" ";C:EXPLODE
150 PRINT@4,5;S:PRINT@14,5;T
160 ZAP:ZAP:ZAP:T=T-INT(SQR(L))
170 IF S=0 THEN PRINT@2,21;"BRAVO":PING:END
180 PRINT@2,10;" - X : "
190 PRINT@2,12;" - Y : "
200 PRINT@2,14;" - Z : "
210 FOR I=1 TO 3
220 R#=""
230 GOSUB 300
240 R(I)=VAL(R#)
250 NEXT I
260 GOTO 70
300 PRINT@9,2*I+8;"?";R#;:PRINT"    "
310 GET A#:IF A#="F" THEN END
320 IF ASC(A#)=13 THEN SHOOT:WAIT 20:RETURN
330 R#=R#+A#
340 GOTO 300
```



Ardoise magique

REFERENCES :

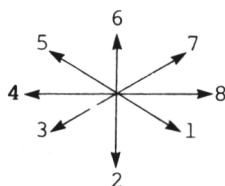
Difficulté de programmation : facile.

Catégorie de jeu : dessin.

Difficulté du jeu : facile.

LE JEU :

Comme le titre peut sembler l'indiquer, il s'agit, tout simplement d'utiliser l'écran du téléviseur comme une ardoise, en manipulant un point sur l'écran suivant 8 directions, à l'aide des touches de 1 à 8, suivant la convention :



On dispose aussi de la possibilité de déplacer le point sur l'écran sans laisser de trace en appuyant sur la touche 0 (ce qui permet l'effacement partiel) et de laisser de nouveau une trace en réappuyant sur 0.

On peut choisir la couleur du point en appuyant sur une touche de A à H ; et il est possible d'effacer tout l'écran en appuyant sur P.

LE PROGRAMME :

La ligne 30 contient le choix des possibilités de couleur.

La ligne 50 imprime le point élémentaire.

Les lignes 60 à 90 contrôlent les entrées.

La ligne 130 appelle l'un des quatre sous-programmes effectuant le déplacement.

La ligne 140 vérifie que l'on ne sort pas des limites de l'écran.

La ligne 160 effectue l'effacement s'il y a lieu.

Le sous-programme 400-410 choisit la couleur ; 300-310 réalise ou non l'effacement.

POUR JOUER :

On lance le programme par RUN ; un point apparaît vers le milieu de l'écran. On devra taper 0 pour le déplacer sans afficher.

Si on frappe une lettre de A à H, la couleur du point change.

La touche P efface tout, et l'on recommence.

On peut sortir à tout moment, en tapant F.

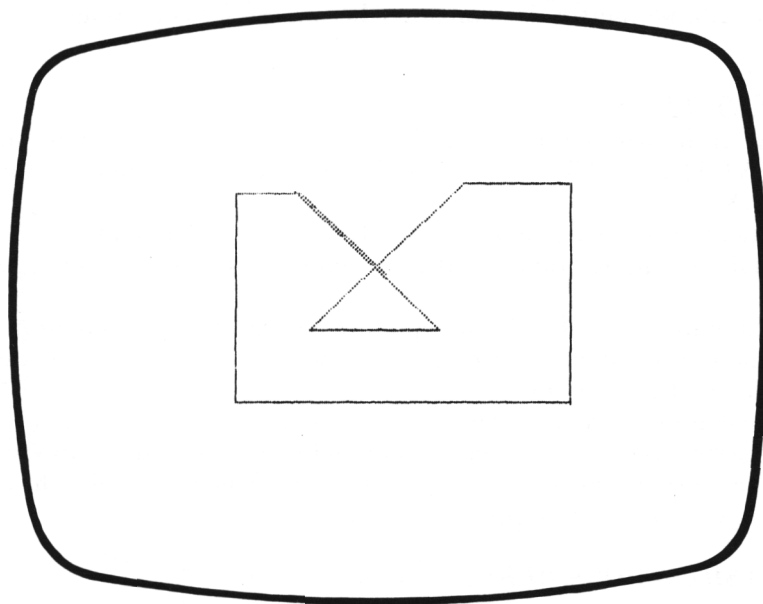
EXTENSIONS POSSIBLES :

On pourra, bien sûr, envisager d'écrire un programme identique en utilisant des tracés de segments, prévoir un sous-programme qui permette la sauvegarde du dessin réalisé pour une utilisation ultérieure, etc.


```

10 REM ARDOISE MAGIQUE
20 HIRES:PAPER 4:INK 2
30 C$="01234567"
40 X=120:Y=100
50 CURSET X,Y,1:INK C
60 GET A$:IF A$="F" THEN END
70 IF A$>="A" AND A$<="H" THEN GOSUB 400
80 IF A$="0" THEN GOSUB 300
90 IF A$="P" THEN 20
100 F=-2*(VAL(A$)<5)-1
110 A=X:B=Y
120 U=VAL(A$)+2*(F-1)
130 ON U GOSUB 1000,2000,3000,4000
140 D=-(X>0 AND X<239 AND Y>0 AND Y<199)
150 IF D=0 THEN GOSUB 5000
160 IF S<>0 THEN CURSET A,B,0
170 GOTO 50
300 S=ABS(S-1)
310 RETURN
400 C=VAL(MID$(C$,ASC(A$)-64,1))
410 RETURN
1000 Y=Y+F
2000 X=X+F
2010 RETURN
3000 GOSUB 2000
4000 Y=Y-F
4010 RETURN
5000 X=A:Y=B
5010 RETURN

```



Affichage publicitaire

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : affichage publicitaire.

Difficulté du jeu : aucune.

LE JEU :

Un texte quelconque devra être introduit en début de programme. Il pourra être relativement long (1 ou 2 lignes) ; on choisira ensuite le sens et la vitesse de défilement, à la suite de quoi le texte va apparaître (une seule ligne à la fois) et commencera à défiler, à la vitesse choisie, et dans le sens choisi.

LE PROGRAMME :

Le texte sera introduit lettre par lettre aux lignes 60 à 90.

L'affichage sera tronqué à 18 caractères au plus, grâce à la ligne 150.

Le défilement vers la gauche est préparé à la ligne 180, et le défilement vers la droite à la ligne 190.

Le texte sera affiché aux lignes 160 et 170, et la durée de l'affichage (vitesse de défilement) est choisie par la ligne 200.

La ligne 220 permettra de changer, si besoin est, les conditions du défilement.

POUR JOUER :

On lancera le programme par RUN, et à l'affichage de la question ENTREZ VOTRE TEXTE, on introduira le texte à faire défiler, suivi de RETURN.

On obtient ensuite l'affichage DEFILEMENT A DROITE (D) OU A GAUCHE (G) auquel on répondra en tapant, soit D, soit G, suivi de RETURN.

Puis à la question VITESSE, on répondra en introduisant un petit nombre pour une vitesse rapide, et un grand nombre pour une vitesse lente (par exemple, de 0 à 40 ou plus).

Le texte commencera à défiler dès que l'on aura frappé RETURN.

Si l'on désire changer les conditions du défilement, il suffit d'appuyer sur la touche O pour retrouver les questions SENS et VITESSE. On tapera CTRL et C pour interrompre le programme.

EXTENSIONS POSSIBLES :

On pourra introduire de la couleur, réaliser un affichage ondulant, faire remonter les lignes sur l'écran, etc.

```

10 REM AFFICHAGE PUBLICITAIRE
20 PAPER 6:INK 4:CLS
30 B$="":D$=CHR$(27):E$=D$+"J"
40 PRINT"ENTREZ VOTRE TEXTE":PING
50 PRINT:PRINT CHR$(17)
60 GET A$
70 IF ASC(A$)=13 THEN 100
80 B$=B$+A$
90 PRINT A$;:GOTO 60
100 PRINT:PRINT"DEFILEMENT A DROITE (D)
    OU A GAUCHE (G)":INPUT S$
110 PRINT
120 INPUT"VITESSE (DE 0 A 50) :":V
130 V=51-V
140 CLS:T=LEN(B$)
150 A$=MID$(B$,1,18)
160 PRINT@2,13;E$;" "+A$+" "
170 PRINT@2,14;E$;" "+A$+" "
180 IF LEFT$(S$,1)="G" THEN B$=RIGHT$(B$,T-1)+LEFT$(B$,1)
190 IF LEFT$(S$,1)="D" THEN B$=RIGHT$(B$,1)+LEFT$(B$,T-1)
200 FOR J=0 TO 10*V
210 C$=KEY$
220 IF C$="0" THEN 100
230 NEXT J
240 GOTO 150

```

ENTREZ VOTRE TEXTE

IL ETAIT UNE FOIS...
 DEFILEMENT A DROITE (D) OU A GAUCHE (G)
 ? D

VITESSE (DE 0 A 50) :? 20

Evasion

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : solitaire.

Difficulté du jeu : facile.

LE JEU :

Votre seul problème : réussir à traverser l'écran, sans que l'ORIC-ATMOS vous ait auparavant touché par l'un des nombreux obus explosifs qu'il va tirer sur vous.

Vous pouvez vous déplacer dans 4 directions (à l'aide des 4 flèches), et vous devez réussir à sortir du terrain par le bord droit, sans, bien sûr, avoir été touché une seule fois.

LE PROGRAMME :

La ligne 30 définit la position de départ du joueur.

La ligne 40 choisit la coordonnée horizontale de la case où ORIC va tirer (liée à votre position), et la ligne 50 choisit la coordonnée verticale (arbitraire) de cette case.

Le déplacement du joueur est contrôlé par les lignes 120 et 130 ; sa trajectoire est tracée par la ligne 100, et sa position actuelle (sur l'écran, un #) est imprimée par la ligne 140.

La ligne 150 regarde s'il est touché ou non.

La ligne 160 effectue le contrôle de fin de partie.

POUR JOUER :

On commence l'exécution du programme par RUN.

Un premier tir a lieu, et ce n'est qu'après que le joueur apparaît, représenté par un #.

Les 4 touches de direction (←, ↓, ↑, →) servent à déplacer le joueur.

S'il est touché, on obtient un bruit d'explosion et la partie s'arrête.

S'il réussit à sortir par la droite de l'écran sans être touché, on obtient une série d'applaudissements, et la partie est également terminée.

On peut terminer à tout moment en tapant CTRL et C ; ne pas oublier de faire TEXT.

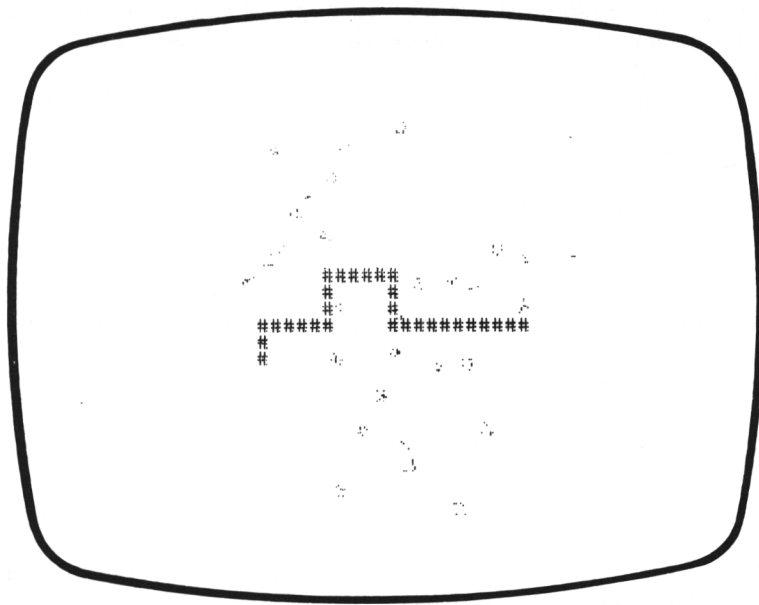
EXTENSIONS POSSIBLES :

La couleur et un graphisme plus soigné amélioreront considérablement l'attrait du jeu. On pourra aussi rendre le tir des bombes plus aléatoire, introduire des obstacles à contourner, ajouter des contrôles, faire participer plusieurs joueurs, etc.

```

10 REM EVASION
20 PAPER 0:HIRES:INK 1
30 X=2:Y=12
40 H=6*(X-1)+14*RND(1)
50 K=190*RND(1)+5
60 IF X<3 THEN 110
70 FOR J=1 TO 7*RND(1)+3
80 CURSET H+6*RND(1)-2,K+6*RND(1)-2,1
90 NEXT J
100 ZAP:CURSET6*X,8*Y,1:CHAR 46,0,1
110 A#=KEY$
120 X=X-(A#=CHR$(9))+(A#=CHR$(8))
130 Y=Y-(A#=CHR$(10))+(A#=CHR$(11))
140 CURSET6*X,8*Y,1:CHAR 35,0,1
150 IF X>3 AND(ABS(H-6*X)+ABS(K-8*(25-Y))<=10)
    THENEXPLODE:WAIT 100:END
160 IF X<38 THEN 40
170 ZAP:PAPER 2:ZAP:PAPER 4:ZAP:PAPER 7:PING

```



Chapeaux

REFERENCES :

Difficulté de programmation : assez facile.
Catégorie de jeu : jeu de réflexe, pour deux joueurs.
Difficulté du jeu : facile.

LE JEU :

L'idée en est très simple : deux personnages se déplacent l'un vers l'autre sur l'écran et se croisent. Au moment où ils se croisent, il est possible qu'ils échangent leurs chapeaux. Il s'agit d'être le premier à dire si oui ou non il y a eu échange de chapeaux lors du croisement.

LE PROGRAMME :

Les lignes 30 et 40 contiennent le dessin des deux chapeaux (dans deux variables alphanumériques B\$ et C\$).

La ligne 50 contient, dans une autre variable A\$, les symboles qui permettront de dessiner les personnages.

La ligne 70 permet, s'il y a lieu, l'échange des chapeaux.

La variable T de la ligne 80 sert à repérer s'il y a eu échange ou non.

Le dessin des deux personnages est réalisé par une boucle I (120-150), et leur déplacement est réalisé par la ligne 160.

Le test, permettant ou non le changement des couvre-chefs, est effectué en ligne 180.

Les lignes 200 à 250 déterminent si le premier joueur ayant appuyé sur une touche a gagné ou perdu.

POUR JOUER :

Lancer le programme par RUN.

Les personnages commencent immédiatement à se déplacer l'un vers l'autre.

Le joueur de gauche dispose des touches 1 et 2 et celui de droite, des touches 9 et 0.

S'il y a eu changement de chapeau, les touches à appuyer seront 1 (à gauche) et 0 (à droite).

S'il n'y a pas eu de changement, il faudra, pour gagner, être le premier à appuyer sur 2 (à gauche) et sur 9 (à droite).

L'affichage indiquera alors BRAVO GAUCHE ou BRAVO DROITE, ou, encore PERDU GAUCHE ou PERDU DROITE.

EXTENSIONS POSSIBLES :

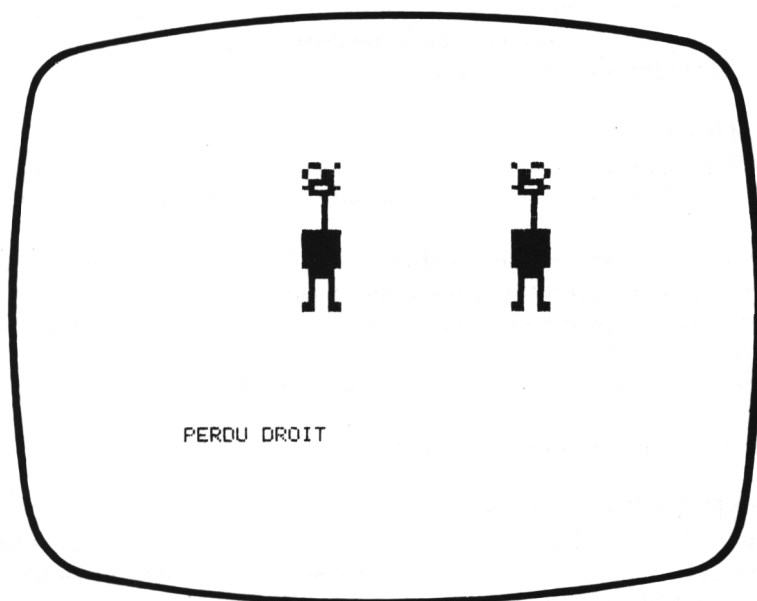
Il sera possible de prévoir un affichage couleur, avec une animation des personnages.

On pourra aussi compliquer le jeu à plaisir avec plusieurs objets échangés de façon aléatoire (il faudrait alors en trouver la liste complète).

```

10 REM CHAPEAUX
20 D$=CHR$(27):E$=D$+"I":PRINT CHR$(17)
30 B$=E$+"6I6"
40 C$=E$+"I6I"
50 A$="NS= J J _____K_L7J 5^ J"
60 X=0:Y=30:T=1
70 G$=C$:C$=B$:B$=G$
80 T=ABS(T-1)
90 PAPER 7:INK 0:CLS
100 PRINT@X+3,0;C$
110 PRINT@Y+3,0;B$
120 FOR I=1 TO 8
130 PRINT@X+2,I;D$;"A";E$+MID$(A$,3*I-2,3)
140 PRINT@Y+2,I;D$;"B";E$+MID$(A$,3*I-2,3)
150 NEXT I
160 X=X+1:Y=Y-1
170 PING:WAIT 120
180 IF X=Y AND RND(1)<0.5 THEN 70
190 R$=KEY$:IF R$="" THEN 90
200 PLOT0,16,19:PLOT1,16,4
210 IF X<=Y THEN 230
220 IF (R$="1"ANDT=1)OR(R$="2"ANDT=0)
THENPRINT@2,16;"BRAVO DROIT":ZAP:END
230 IF (R$="0"ANDT=1)OR(R$="9"ANDT=0)
THENPRINT@2,16;"BRAVO GAUCHE":ZAP:END
240 IF R$="2" OR R$="1" THEN PRINT@2,16;
"PERDU GAUCHE":EXPLODE
250 IF R$="9" OR R$="0" THEN PRINT@2,16;
"PERDU DROIT":EXPLODE

```



Combat aérien

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexe (1 joueur).

Difficulté du jeu : difficile.

LE JEU :

Au centre de l'écran, se trouve le viseur de votre appareil, et l'avion ennemi se déplace devant vous, d'une façon qui pourra vous sembler au début incohérente.

Vous pouvez vous déplacer dans toutes les directions, grâce aux flèches de mouvement, et vous allez essayer d'amener l'appareil ennemi au centre de votre viseur.

Vous tirez sur lui en frappant Ø, et vous voyez vos projectiles qui convergent sur lui. Mais, il continue à se déplacer, même pendant votre tir, et il n'est pas du tout évident que vous réussirez à l'avoir.

La partie continuera à se dérouler jusqu'à ce que vous l'ayez touché (affichage de GAGNE) ou qu'il soit sorti de votre champ de vision.

LE PROGRAMME :

La ligne 70 dessine l'appareil ennemi dont le déplacement aléatoire sera réalisé par les lignes 90 et 100.

Les lignes 120 et 130 vous permettent le pilotage de votre appareil.

La ligne 170 dessine le viseur de votre appareil.

Le sous-programme, gérant le tir, va de la ligne 200 à la ligne 250 et le test de fin de partie est effectué en ligne 240.

La ligne 180 regarde si vous avez tiré ou non.

POUR JOUER :

Dès que le programme a été lancé par l'ordre RUN, vous devez manœuvrer pour essayer de faire en sorte que l'appareil ennemi soit cadré dans votre viseur.

Pour cela, vous disposez des flèches de déplacement (touche ↑, ↓, →, ←) et vous pouvez tirer à tout moment en pressant la touche O.

L'appareil ennemi peut continuer à se déplacer de manière aléatoire, même lorsque vous avez tiré.

La partie se terminera avec l'affichage de GAGNE, si vous touchez l'avion ennemi.

On peut aussi arrêter à tout moment en frappant CTRL et C.

EXTENSIONS POSSIBLES :

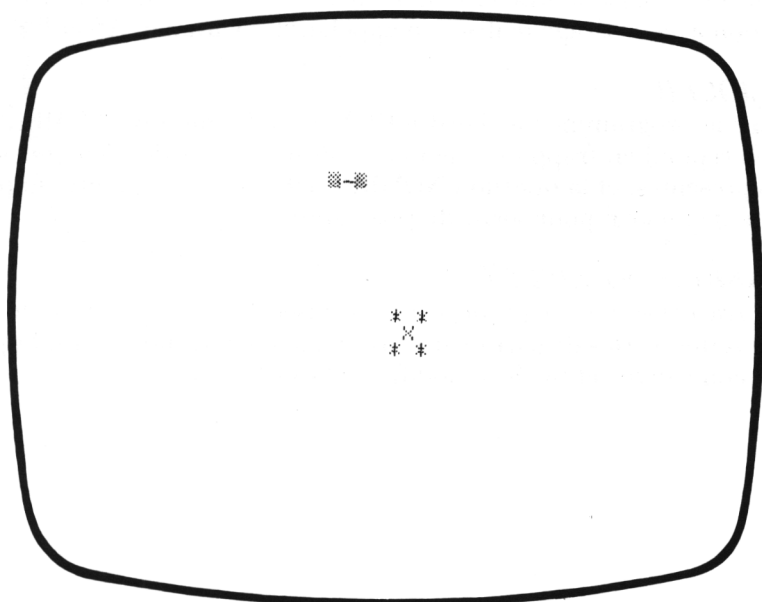
Il sera assez facile de faire un affichage au graphisme plus soigné, utilisant la couleur. On pourrait envisager de stabiliser la position de l'avion ennemi pendant le tir (il suffit de modifier la ligne 180).

On peut aussi introduire plusieurs avions, compter les points ou le temps, tenir compte de sa propre réserve de carburant, etc.


```

10 REM COMBAT AERIEN
20 D$=CHR$(27):E$=D$+"A":F$=D$+"G"
30 PAPER 4:INK 7:PRINT CHR$(17)
40 CLS
50 A=5:B=15
60 N=21
70 PRINT@B,A:CHR$(255)+"-"+CHR$(255)
80 U=A:V=B
90 A=A+2*INT(RND(1)*2)-1
100 B=B+2*INT(RND(1)*2)-1
110 A$=KEY$
120 B=B-2*(A$=CHR$(9))+2*(A$=CHR$(8))
130 A=A-2*(A$=CHR$(10))+2*(A$=CHR$(11))
140 IF B<2 OR B>37 THEN B=ABS(B-37)
150 IF A<3 OR A>22 THEN A=ABS(A-22)
160 PRINT@V,U;"  "
170 PRINT@17,9;E$;"* *";F$:PRINT@17,10;E$;" X ";F$:
PRINT@17,11;E$;"* *";F$
180 IF A$="0" OR N<21 THEN GOSUB 200
190 GOTO 70
200 PRINT@29-N,N;"^":PRINT@N+9,N;"^"
210 PRINT@29-N,N;" ":PRINT@N+9,N;" "
220 N=N-1:SHOOT
230 IF N>9 THEN RETURN
240 IF(A=9ORA=10ORA=11)AND(B=18ORB=19ORB=20)
THENPRINT"GAGNE":EXPLODE:END
250 GOTO 60

```



Dessin de dés

REFERENCES :

Difficulté de programmation : difficile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :

Le programme proposé est surtout intéressant par son côté graphique puisqu'il permet de représenter, de façon très convenable, de 1 à 3 dés sur l'écran, en utilisant une technique de dessin un peu particulière. Chaque réponse à la question NOMBRE DE DES provoque immédiatement un nouveau lancer du nombre de dés indiqué.

LE PROGRAMME :

La ligne 50 permet de contrôler s'il y a eu ou non action sur une touche de 1 à 3.
Les lignes 90 et 100 initialisent les deux variables de dessin des dés.
La boucle I (110-200) et les lignes 120 et 130 dessinent le cadre du dé.
La ligne 140 réalise le tirage, et la boucle J (150-190) dessine les points en relation avec le tirage réalisé (en appelant les variables A\$ et B\$).

POUR JOUER :

On lance le programme par l'ordre RUN, et, à la question NOMBRE DE DES, on répond en frappant une touche de 1 à 3. Les dés demandés sont alors représentés, et la question NOMBRE DE DES est de nouveau posée. Il suffira de taper F pour sortir du programme.

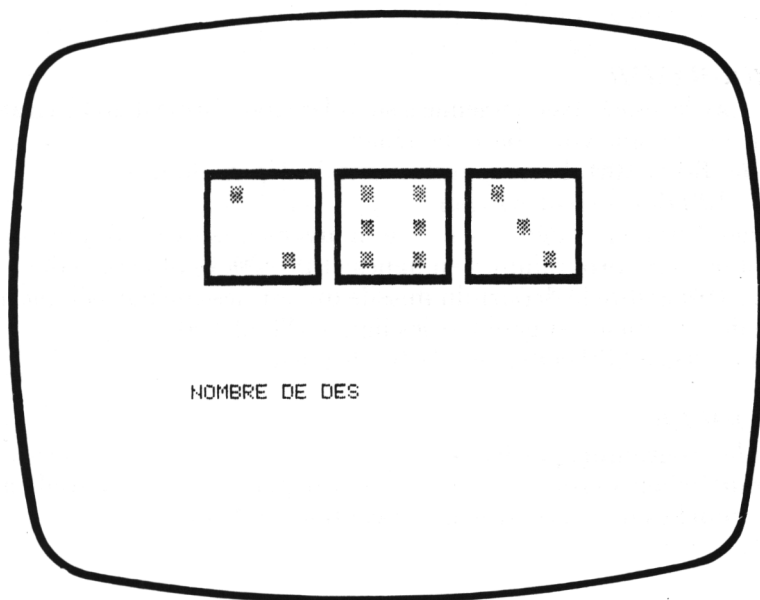
EXTENSIONS POSSIBLES :

On pourra conserver la partie dessin utilisée dans ce programme pour réaliser toutes sortes de jeux de dés, avec une représentation graphique de qualité convenable et facile à modifier si besoin était.

```

10 REM DESSIN DE DES
20 D$=CHR$(27):E$=D$+"I":PRINT CHR$(17)
30 PAPER 3:INK 4:CLS
40 PRINT@2,15;"NOMBRE DE DES"
50 GET A$:IF A$="F" THEN END
60 CLS:FOR I=1 TO 9:PLOT0,I,17:PLOT1,I,7:NEXT I
70 N=VAL(A$):IF N<1 OR N>3 THEN 30
80 F$=CHR$(255)
90 A$="      "+F$+"      "+F$+"      "+F$+"F$+"      "+F$
100 B$="121314324515525555"
110 FOR I=1 TO N
120 PRINT@10*I-8,2;E$+"?////////0"
130 PRINT@10*I-8,8;E$+"J\////////^"
140 X=INT(RND(1)*6+1)
150 FOR J=1 TO 5
160 PRINT@10*I-8,J+2;E$+"5":PRINT@10*I,J+2;E$+"J"
170 IF JK=3 THEN S=5*VAL(MID$(B$,3*(X-1)+J,1))-4
180 IF JK=3 THEN PRINT@10*I-6,2*J+1;D$;"H":MID$(A$,S,5)
190 NEXT J
200 NEXT I
210 PING:GOTO 40

```



Le jeu du missile

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : contre l'ORIC-ATMOS.

Difficulté du jeu : moyenne.

LE JEU :

L'ORIC-ATMOS est en alerte : vous êtes aux commandes d'un avion chargé d'une mission d'espionnage sur son territoire ; votre but est de survoler une base ennemie (représenté par un H agrandi), et de revenir sain et sauf sur votre base ; mais, dès que l'ORIC-ATMOS vous aura repéré, il va envoyer à votre poursuite un missile très perfectionné qui n'aura de cesse de vous détruire.

Votre avion peut se déplacer dans toutes les directions, grâce aux touches de 1 à 8 ; réussirez-vous à remplir votre mission ?

LE PROGRAMME :

La ligne 50 dessine la base ennemie à survoler (sous forme d'un H agrandi).

La ligne 60 dessine votre base de départ.

Les lignes 80 et 100 choisissent l'instant de départ du missile.

La ligne 120 dessine votre avion.

Les lignes 140 et 150 définissent son mouvement suivant 8 directions, en utilisant les sous-programmes en lignes 1000, 2000, 3000 et 4000.

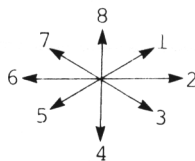
La ligne 160 assure le départ du missile qui est dessiné par la ligne 170 et dont le déplacement est géré par les lignes 180 et 190.

Les lignes 200 à 230 contrôlent la fin de partie.

POUR JOUER :

Lancer le programme par RUN.

Le terrain de jeu s'affiche et vous pouvez déplacer votre appareil en utilisant les touches de 1 à 8, avec la convention suivante :



Vous devez lui faire contourner le territoire ennemi (si vous, ou le missile, touchez ce territoire, une deuxième partie commence).

Si le missile vous détruit, vous obtenez un message TOUCHE, et la partie s'arrête.

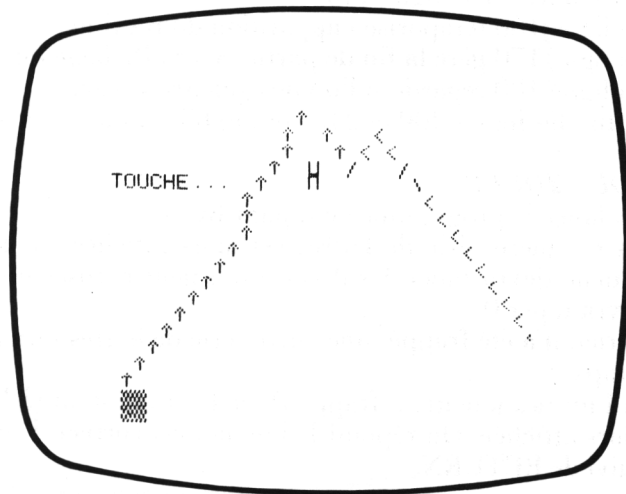
Taper CTRL et C pour arrêter la partie.

EXTENSIONS POSSIBLES :

Il sera sans doute nécessaire de rajouter divers contrôles dans le programme (l'avion a-t-il survolé la base ennemie, est-il revenu à sa propre base...).

On peut aussi prévoir divers niveaux de jeu en faisant, par exemple, varier la vitesse du missile, prévoir l'utilisation de leurres par l'avion, l'explosion du missile au bout d'un certain temps, etc.

```
10 REM LE JEU DU MISSILE
20 D$=CHR$(27):E$=D$+"J":F$=CHR$(255)
30 PAPER 3:INK 4:PRINT CHR$(17)
40 CLS:M=0:PLOT1,19,2:PLOT1,20,2
50 PRINT@16,5;E$;"H":PRINT@16,6;E$;"H"
60 PRINT@2,19;E$;F$;F$:PRINT@2,20;E$;F$;F$
70 U=15:X=18
80 S=INT(RND(1)*U+6)
90 Y=2
100 S=S-1
110 V=33
120 PRINT@Y,X;D$;"A";"^"
130 A$=KEY$:IF A$="" THEN 130
140 F=-2*(VAL(A$)<5)-1
150 ON VAL(A$)+2*(F-1) GOSUB 1000,2000,3000,4000
160 IF S<>0 THEN 100
170 PRINT@V,U;D$;"F";"<"
180 U=U-SGN(U-X)
190 V=V-SGN(V-Y)
200 IF U=X AND V=Y THEN PRINT"TOUCHE...":EXPLODE:WAIT 50:END
210 IF (U=5ORX=5)AND(V=16ORV=17ORY=16ORY=17)THEN 40
220 IF X<=5 AND Y>=16 THEN M=1
230 IF M=1 AND X>18 AND Y<=4
    THEN PRINT"MISSION ACCOMPLIE":ZAP:ZAP:ZAP:END
240 GOTO 120
1000 X=X-F
2000 Y=Y+F
2010 RETURN
3000 GOSUB 2000
4000 X=X+F
4010 RETURN
```



Le singe

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : deux joueurs.

Difficulté du jeu : difficile.

LE JEU :

Il s'inspire de l'idée déjà ancienne du singe qui, frappant au hasard sur une machine à écrire, a des chances non nulles de finir par écrire un texte signifiant.

Le rôle du singe est, ici, tenu par l'ORIC-ATMOS qui va afficher des groupes de 3 à 6 lettres pendant un temps relativement court.

Si l'un des joueurs estime qu'il peut réussir à faire un mot avec les lettres proposées, il doit taper rapidement sur sa touche de jeu (1 ou 0).

Le mot devra alors être accepté ou refusé par l'adversaire qui répondra JUSTE ou FAUX.

Chaque mot juste rapporte 1 point, chaque mot faux enlève 1 point.

LE PROGRAMME :

La ligne 40 contient une liste V\$ des voyelles courantes, et une liste C\$ des consonnes courantes.

Il sera possible de modifier ces deux listes pour augmenter, par exemple, la fréquence d'apparition de certaines lettres (on pourra ainsi mettre deux E dans V\$ ou deux R dans C\$, etc.).

La ligne 50 définit la longueur du mot (en principe, de 3 à 6 lettres, mais on peut également modifier cette longueur).

La boucle I (80-130) permet la confection d'un "mot" en prenant une lettre, soit dans l'alphabet complet, soit dans la liste C\$ (ligne 100), soit dans la liste V\$ (ligne 110).

La ligne 150 temporise l'apparition du résultat.

La ligne 170 gère la fin de partie, avec affichage du score.

La ligne 180 regarde si l'un des joueurs a réagi.

Enfin, les lignes 200 et 210 comptabilisent les points de chaque joueur.

POUR JOUER :

On lance le programme en tapant RUN.

La première série de lettres est alors affichée sur l'écran. Le joueur de gauche devra taper 1 s'il voit qu'il peut réaliser un mot, celui de droite devra taper 0.

Si rien n'a été frappé, une autre série de lettres sera affichée après peu de temps.

Si l'un des joueurs a frappé 0 ou 1, la question (J)USTE ou (F)AUX est alors affichée. On répond J si le mot est correct, F si le mot est incorrect, suivi de RETURN.

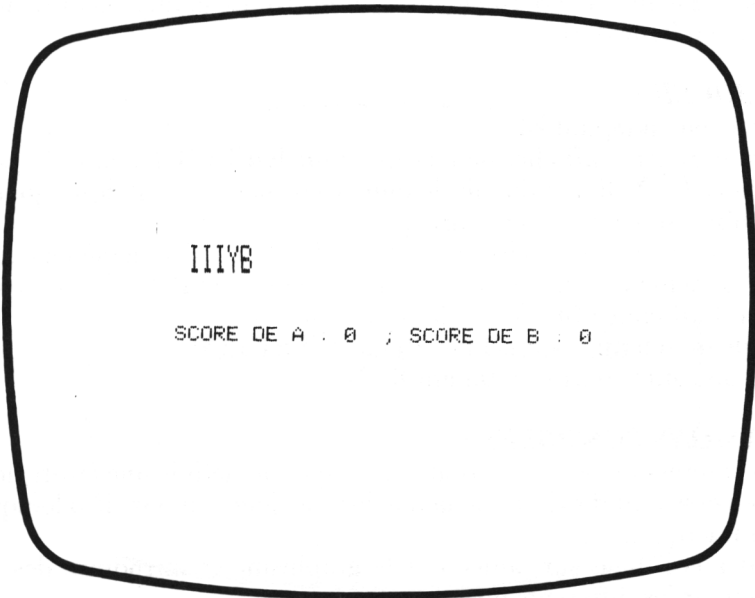
Les points sont alors calculés par ORIC, et la série de lettres suivante est alors affichée.

Pour sortir du programme, il faudra taper P au lieu de 0 ou 1 ; le score final est alors affiché.

EXTENSIONS POSSIBLES :

On pourra modifier la longueur des mots, la fréquence d'apparition des lettres, le décompte des points, la règle du jeu, etc.

```
10 REM LE SINGE
20 D$=CHR$(27):F$=D$+"J"
30 PAPER 6:INK 4
40 A=0:B=0:V$="AEIOU":C$="BCDGLMNPQRSTV"
50 X=INT(RND(1)*4+3)
60 M$=""
70 CLS:PLOT1,5,1:PLOT1,6,1
80 FOR I=1 TO X
90 E$=CHR$(INT(RND(1)*26+65))
100 IF RND(1)<0.4 THEN E$=MID$(C$,INT(LEN(C$)*RND(1)+1),1)
110 IF RND(1)<0.5 THEN E$=MID$(V$,INT(LEN(V$)*RND(1)+1),1)
120 M$=M$+E$
130 NEXT I
140 PRINT@2,5;F$;M$:PRINT@2,6;F$;M$:PING
150 WAIT 200
160 A$=KEY$
170 IF A$="P" THEN PRINT@2,10;"SCORE DE A :";A$;" ;
    SCORE DE B :";B$:EXPLODE:END
180 IF A$<>"0" AND A$<>"1" THEN 50
190 PRINT:INPUT"(J)USTE OU (F)AUX :";R$
200 A=A-(R$="J" AND A$="0")+(R$="F" AND A$="0")
210 B=B-(R$="J" AND A$="1")+(R$="F" AND A$="1")
220 GOTO 50
```



Le jeu du 15

REFERENCES :

Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :

C'est un jeu pour deux joueurs, assez ancien, qui était autrefois pratiqué dans les foires. On dispose d'un tapis de jeu de 9 cases, et chacun mise à son tour un jeton de sa couleur, sur une case disponible.

Le premier des deux joueurs qui, avec exactement 3 jetons, arrive à faire un total de 15, a gagné.

LE PROGRAMME :

Les lignes 50 à 80 dessinent le terrain de jeu, et les lignes 90 à 100 s'occupent de la gestion du tour des joueurs (numéro 1 ou 2 pour chaque joueur, et jeton O ou X correspondant).

La ligne 110 attend le coup joué.

La boucle I (120-140) vérifie la validité du coup.

La ligne 150 teste la fin de partie, en cas de match nul, et la ligne 160 refuse les coups non conformes (avec retour à la situation initiale, ligne 90).

Les lignes 170 et 180 s'occupent de l'affichage du coup, et la ligne 190 l'enregistre dans une variable R\$. La partie la plus intéressante du programme, qui pourrait servir à définir une stratégie machine, va de la ligne 210 à la ligne 300 : on utilise 3 boucles emboîtées pour vérifier si l'un des joueurs a obtenu, avec 3 pions, un total de 15. Le changement de joueur s'effectue en ligne 310.

POUR JOUER :

Lancer le jeu en tapant RUN.

Le terrain de jeu s'affiche, puis la question JOUEUR 1 : X, et le joueur, qui a choisi les X, devra donner le numéro de la case où il désire jouer ; si elle est libre, son coup y sera enregistré.

On obtient, ensuite, l'affichage JOUEUR 2 : O et le deuxième joueur procède de la même façon. Si l'un des joueurs gagne (total de 15 sur 3 coups), il obtient l'affichage BRAVO et la partie s'arrête.

En cas de match nul, la machine affiche C'EST FINI.

On fera une autre partie en tapant RUN.

EXTENSIONS POSSIBLES :

Le plus intéressant sera sans doute d'essayer de définir une stratégie machine (en essayant d'éviter que la machine ne joue toujours le 5 lorsqu'elle doit commencer).

On peut aussi, bien sûr, améliorer le graphisme et introduire des effets sonores pour les différents messages.

Adresse

REFERENCES :

Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (pour 1 joueur).
Difficulté du jeu : difficile.

LE JEU :

Des chiffres, entre 0 et 9, vont apparaître au hasard, en divers emplacements de l'écran. Ils restent affichés pendant un temps très court. Le joueur doit réussir à appuyer sur la touche numérique correspondante, au moment où le chiffre est affiché.

Le chiffre tapé par le joueur sera alors affiché en grand format sur la gauche de l'écran ; et chaque réponse exacte rapporte 10 points. La durée de la partie est aléatoire et le score sera affiché en fin de partie.

LE PROGRAMME :

Les lignes 50 et 60 contiennent, dans deux variables alphanumériques C\$ et B\$, les références qui seront utilisées pour dessiner les chiffres en grand format.

L'emplacement du chiffre tiré par l'ORIC-ATMOS est choisi aux lignes 70 et 80, et le chiffre est tiré en ligne 90.

La ligne 100 réalise l'impression, et la ligne 110 détermine la durée d'affichage. La ligne 130 définit la durée du jeu, et la fin de partie est traitée par la ligne 150.

La boucle I (200-230) réalise le dessin grand format du chiffre frappé par le joueur.

Le contrôle de l'exactitude du résultat est effectué par la ligne 240.

POUR JOUER :

Le programme sera lancé par RUN, puis un premier chiffre apparaît, après une courte pause, vers le milieu de l'écran. Le joueur devra réagir suffisamment vite, tant qu'il reste à l'affichage, et le retaper. Le chiffre qu'il a tapé se dessinera alors en haut à gauche de l'écran, en grand format.

Si le résultat est exact, le joueur marquera 10 points.

Le temps de déroulement de la partie sera fonction des chiffres qui apparaîtront à l'affichage.

En fin de partie, le score sera affiché. Il sera possible de s'arrêter à tout moment en tapant CTRL et C.

EXTENSIONS POSSIBLES :

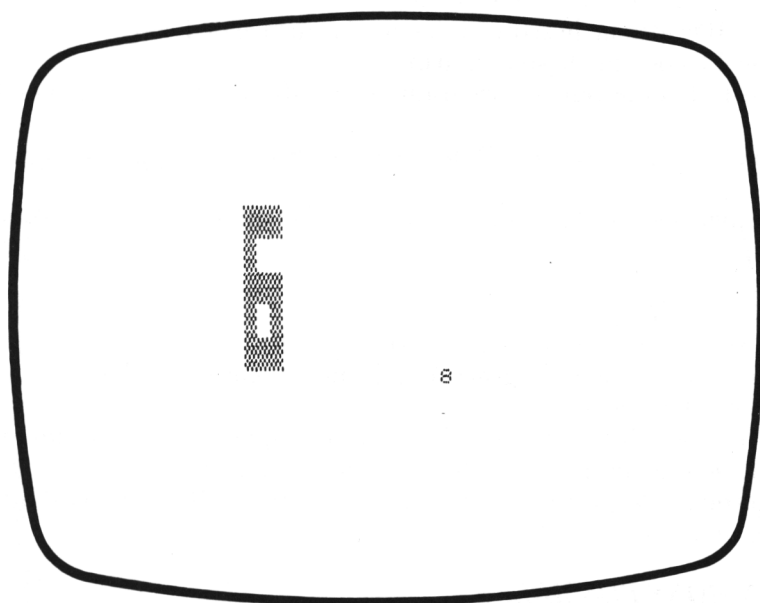
On peut envisager de modifier le jeu pour faire participer plusieurs joueurs, en modifiant éventuellement l'affichage et en codant les touches du clavier mises à la disposition de chacun des joueurs.

On pourra aussi enlever des points à chaque résultat faux, modifier les règles de calcul du score et la durée du jeu, etc.

```

10 REM ADRESSE
20 D$=CHR$(27):E$=CHR$(255):F$=D$+"A"+D$+"J":G$=D$+"H"
30 PAPER 6:INK 4:PRINT CHR$(17)
40 T=100:S=0:CLS
50 C$=E$+E$+E$+E$+" "+E$+E$+" "+E$
60 B$="14441333331312113131441331213112141133331414114131"
70 X=INT(RND(1)*15)
80 Y=INT(RND(1)*15+15)
90 N=INT(RND(1)*10)
100 PRINT@Y,X;N
110 WAIT RND(1)*120+80
120 A$=KEY$:IF A$(">") THEN 190
130 T=T-N
140 PRINT@Y,X;" "
150 IF T>=0 THEN 70
160 CLS
170 PRINT:PRINT"VOTRE SCORE EST DE ";S;" SUR 100"
180 ZAP:ZAP:ZAP:END
190 U=VAL(A$)
200 FOR I=1 TO 5
210 A=3*VAL(MID$(B$,5*U+I,1))-2
220 PRINT@2,2*I-1;F$;MID$(C$,A,3);G$:
    PRINT@2,2*I;F$;MID$(C$,A,3);G$
230 NEXT I
240 IF U=N THEN S=S+10
250 GOTO 130

```



Sous-marin

REFERENCES :

Difficulté de programmation : moyenne.
Catégorie de jeu : réflexes (1 seul joueur).
Difficulté du jeu : facile.

LE JEU :

Vous manœuvrez en surface un chasseur de sous-marins, et des sous-marins passent très régulièrement en dessous de votre unité, à différentes profondeurs.

Vous pouvez, à tout moment, larguer des grenades sous-marines et vous pouvez même envisager de toucher plusieurs fois le même sous-marin.

Votre score sera affiché en permanence, sur la gauche de l'écran (rapport nombre de coups au but sur nombre de sous-marins apparus).

LE PROGRAMME :

La ligne 70 permet à l'ORIC-ATMOS de choisir la profondeur du sous-marin.

La ligne 90 dessine votre bâtiment.

La ligne 100 affiche le score et dessine la surface de l'eau.

La ligne 110 dessine le sous-marin.

Le tir est effectué par un programme spécifique placé aux lignes 190 à 230.

Le déplacement (très rapide) de votre navire est réalisé par les lignes 160 et 170.

Le test, permettant de savoir si l'on a touché ou non, est effectué en ligne 130.

POUR JOUER :

Taper RUN pour commencer la partie.

Le bâtiment sera déplacé grâce aux flèches droite et gauche.

La touche O autorise le tir.

Le navire, se déplaçant trois fois plus vite que les sous-marins, il sera possible de toucher plusieurs fois le même bâtiment.

Pour terminer la partie, il conviendra de faire CTRL et C ou d'attendre 99 sous-marins.

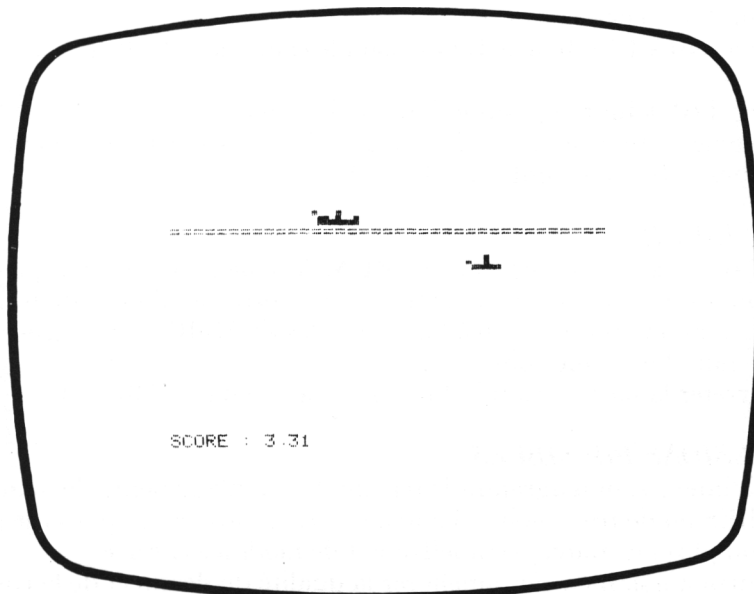
EXTENSIONS POSSIBLES :

Il sera assez élémentaire d'améliorer la qualité graphique du jeu, d'utiliser la couleur et les sons, de faire apparaître plusieurs sous-marins se déplaçant à des vitesses différentes, éventuellement même, de permettre aux sous-marins de tirer...

```

10 REM SOUS-MARIN
20 D#=CHR$(27):E#=D#+"I":F#=D#+"H"
30 PAPER 6:INK 0:CLS
40 P=0:C=17:S=5:PRINT CHR$(17)
50 FOR I=0 TO 3:PLOT0,I,23:NEXT I:
   PLOT1,3,1:PLOT0,4,4
60 PRINT@2,S;" "
70 T=27:S=INT(RND(1)*8+5):P=P+0.01
80 X=1
90 PRINT@C,3;E$;" ITUX "
100 PRINT@2,18;"SCORE :";INT(P*100)/100
110 PRINT@T,S;E$;D$;"C";"DZF "
120 IF X=1 THEN Y=C+2*X
130 IF S=X AND (T+2=Y OR T+3=Y) THEN P=P+1:
   PRINT@T,S;D$;"A";"***":EXPLODE
140 T=T-1:IF P=99 THEN PRINT@2,18:
   "C'EST FINI....":ZAP:ZAP:ZAP:END
150 IF T<3 THEN 60
160 A$=KEY$
170 C=C-3*((A#=CHR$(9) AND C<27)-(A#=CHR$(8)
   AND C>2))
180 IF A$<>" " AND X=1 THEN 90
190 PRINT@Y,X;" "
200 X=X+1
210 IF X>S THEN 80
220 PRINT@Y,X;"0"
230 GOTO 90

```



Zombies

REFERENCES :

Difficulté de programmation : assez difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez facile.

LE JEU :

Les ZOMBIES sont des êtres malfaisants qui vont essayer de vous détruire, mais vous allez, vous aussi, essayer de les utiliser à votre profit, pour débarrasser votre espace vital des trous noirs qui l'obscurcissent. Sur l'écran, est tracé un cadre qui définit votre espace vital. Dans ce cadre, vont apparaître de façon aléatoire, les ZOMBIES (ce sont des Z inversés), les trous noirs (des carrés noirs) et le joueur (un O). Les ZOMBIES convergent vers le joueur pour essayer de le capturer, mais celui-ci se déplace plus rapidement qu'eux, et il va essayer de leur faire avaler tous les trous noirs, sans se faire prendre.

La partie se termine, soit avec la disparition du joueur, soit par l'élimination de tous les zombies.

LE PROGRAMME :

La position des ZOMBIES est définie par un tableau Z non dimensionné (c'est inutile en dessous de 10).

Les lignes 40 à 100 (boucle I) dessinent le cadre, les ZOMBIES et les trous noirs.

La ligne 110 définit la position initiale du joueur.

Une deuxième boucle I (130-220) réalise le déplacement du joueur.

Les lignes 270 à 320 contrôlent le déplacement du joueur.

POUR JOUER :

Après avoir lancé l'exécution par RUN, le joueur devra se déplacer en utilisant les 4 flèches et en veillant à ne pas sortir des limites du terrain.

Il devra manœuvrer de façon à faire venir les ZOMBIES sur les trous noirs, sans se faire lui-même coïncider.

Pour arrêter la partie avant la fin, il suffira de taper CTRL et C.

EXTENSIONS POSSIBLES :

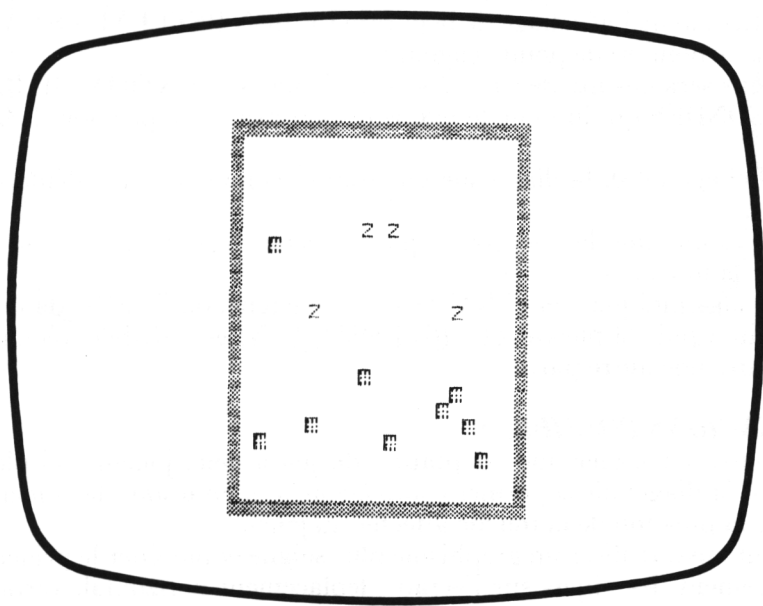
Il serait intéressant d'agrandir le terrain de jeu, d'augmenter le nombre de ZOMBIES ou de trous noirs, d'améliorer les contrôles, de placer le joueur de façon plus aléatoire, éventuellement de modifier la règle.

On pourra aussi, bien sûr, améliorer la qualité du dessin et de la couleur.

```

10 REM ZOMBIES
20 PRINT CHR$(17):E$=CHR$(255)
30 PAPER 6:INK 4:CLS
40 FOR I=0 TO 22:PRINT@I+2,0;E$:PRINT@I+2,23;E$
50 PRINT@2,I;E$:PRINT@24,I;E$
60 IF I<4 OR I>23 THEN 80
70 IF RND(1)<0.5 THEN PLOTI,INT(RND(1)*19+2),CHR$(163)
80 S=INT(RND(1)*14+1)
90 Z(INT(I/3)+1)=5+S
100 NEXT I
110 K=INT(RND(1)*18+3)
120 P=0
130 FOR I=1 TO 4
140 P=P+Z(I)
150 IF Z(I)=0 THEN 220
160 PRINT@Z(I+4),Z(I);" "
170 Z(I)=Z(I)-SGN(Z(I)-S)
180 Z(I+4)=Z(I+4)-SGN(Z(I+4)-K)
190 IF SCRNK(Z(I+4),Z(I))=163 THEN Z(I)=0:GOTO 210
200 PRINT@Z(I+4),Z(I);"Z"
210 IF Z(I)=S AND Z(I+4)=K
    THEN PRINT"VOUS AVEZ PERDU ... ":ZAP:ZAP:ZAP:END
220 NEXT I
230 IF P=0 THEN PRINT"BRAVO .
    VOUS AVEZ DETRUIT TOUS LES ZOMBIES":PING:END
240 PRINT@K,S;"0"
250 WAIT 130
260 PRINT@K,S;" "
270 A$=KEY$
280 K=K-2*((A$=CHR$(9))-(A$=CHR$(8)))
290 IF K<=2 OR K>=24 THEN PRINT"DOMMAGE...":EXPLODE:END
300 S=S-2*((A$=CHR$(10))-(A$=CHR$(11)))
310 IF S<=0 OR S>=22 THEN K=25:GOTO 290
320 GOTO 120

```



Diago

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : réflexes (1 seul joueur).

Difficulté du jeu : difficile.

LE JEU :

On lance le programme par RUN et on obtient un premier affichage qui vous renseigne sur la performance à réaliser (par exemple, 5 points noirs ou 17 points noirs...), puis le temps est affiché et décompté, à partir de 10 secondes. Le jeu commence lorsque l'on arrive à 0 seconde. Des ronds apparaissent alors en bas et à gauche de l'écran, formant une ligne diagonale, et il vous faudra appuyer sur la touche 0 pour arrêter l'affichage des points dès que vous estimez que le nombre de points demandés au début est apparu à l'écran.

On obtient alors un affichage BRAVO ou PERDU selon le résultat obtenu.

LE PROGRAMME :

Le nombre de points noirs sera choisi au hasard, en ligne 40.

Le temps est décompté par les lignes 50 à 110.

Les points sont imprimés et la ligne diagonale est fabriquée par la ligne 140.

La pression de la touche 0 est détectée en ligne 180, et la ligne 190 contrôle la fin de partie (aucun réflexe).

Les affichages de fin sont exécutés par les lignes 190 à 230.

POUR JOUER :

On lancera l'exécution par RUN.

On obtient alors l'affichage de la phrase LE BON REFLEXE EST A... qui indique le nombre de points gagnants.

Le temps sera ensuite décompté sous la forme ATTENTION : IL RESTE ... SECONDES qui indique le temps restant avant l'apparition de la diagonale.

À l'affichage de 0, la diagonale apparaît et commence à se déplacer sur l'écran.

Si l'on estime que le nombre de points réclamés a été atteint, on devra presser la touche 0.

L'affichage indique alors BRAVO si l'on a gagné, ou PERDU dans le cas contraire ; puis la phrase AUTRE ESSAI (O/N) est affichée, répondre O pour faire une autre partie.

EXTENSIONS POSSIBLES :

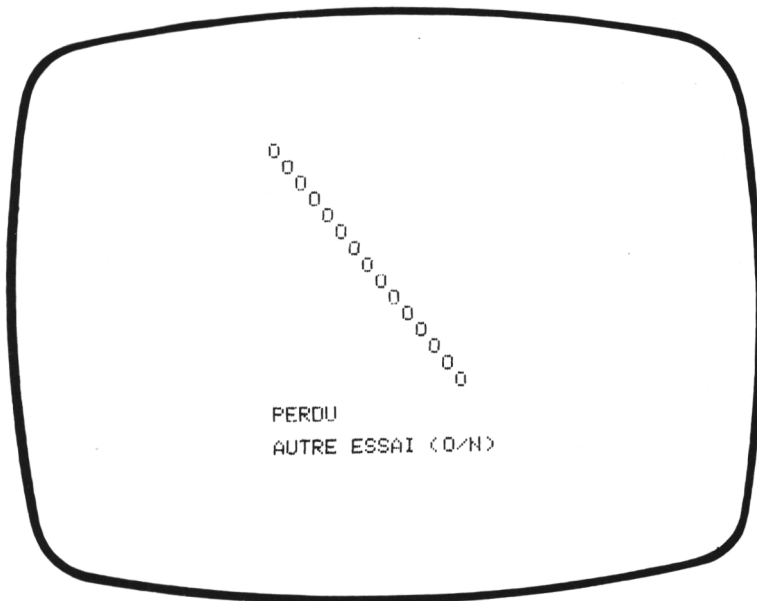
On pourra envisager une adaptation du jeu à deux joueurs ou plus, en laissant la diagonale se former complètement et en testant la coïncidence lors de la pression de la touche affectée au joueur.

On peut aussi réaliser un graphisme plus soigné et modifier le principe du déplacement des objets sur l'écran (déplacement horizontal, vertical ou apparition au hasard...).


```

10 REM DIAGO
20 PAPER 3:INK 0:PRINT CHR$(17)
30 CLS
40 X=INT(RND(1)*18+3)
50 S=10
60 PRINT:PRINT"LE BON REFLEXE EST A "X
70 PRINT@2,10:"ATTENTION . IL RESTE "S;" SECONDES"
80 S=S-1
90 PING:WAIT 120
100 CLS
110 IF S<>0 THEN 60
120 FOR I=0 TO 20:PLOT1,I,1:NEXT I
130 U=0
140 PRINT@2+U,25:"O":PRINT
150 WAIT 10
160 U=U+1
170 A$=KEY$
180 IF A$="0" THEN 210
190 IF U>22 THEN PRINT"PERDU":EXPLODE:GOTO 230
200 GOTO 140
210 IF U<>X THEN U=23:GOTO 190
220 PRINT"BRAVO":ZAP:ZAP:ZAP
230 PRINT:PRINT"AUTRE ESSAI (O/N)"
240 GET A$:IF A$="O" THEN 30

```



Décomaque

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : solution du programme Nicomaque.

Difficulté du jeu : aucune.

LE JEU :

C'est une réplique au programme Nicomaque, proposé dans les pages précédentes ; vous entrez les restes successivement dans les divisions par 3, 5 et 7 d'un entier compris entre 1 et 100, et l'ORIC-ATMOS calcule et affiche cet entier. Le procédé retenu peut d'ailleurs être généralisé (à vous de trouver comment).

LE PROGRAMME :

Les lignes 30 à 50 affichent quelques explications.

La ligne 90 attend que vous frappiez un reste, qui sera refusé s'il est incompatible avec la division par le nombre affiché.

La ligne 100 emmagazine les restes dans une variable non dimensionnée A (S).

Les lignes 130 et 140 réaffichent les restes entrés.

Les boucles I (150-190) et J (160-180) dessinent des petits carrés selon les restes introduits.

Les lignes 220 à 240 gèrent la fin de partie.

POUR JOUER :

On lance le programme par RUN. Un texte de présentation est affiché, puis le message "RESTE DANS LA DIVISION PAR 3" s'affiche.

On doit répondre 0, 1 ou 2, et l'on obtient le message "RESTE DANS LA DIVISION PAR 5" etc.

La solution est affichée dès que l'on a donné les 3 restes.

Pour une autre partie, taper O à la question VOULEZ-VOUS UN AUTRE ESSAI (O/N).

Tapez N pour finir.

EXTENSIONS POSSIBLES :

On peut étendre le programme aux entiers de 1 à 1000, soit en utilisant les restes par 3, 5, 7 et 11 ; on pourra rajouter du son et de la couleur pour rendre le programme plus attrayant.

```

10 REM DECOMAQUE
20 CLS:PAPER 3:INK 4:D#=CHR$(27):E#=D#+"J"
30 PRINT"VOUS DEVEZ ENTRER LES TROIS RESTES"
40 PRINT:PRINT"DANS LES DIVISIONS PAR 3 , 5 , ET 7"
50 PRINT:PRINT"D'UN NOMBRE ENTIER INFERIEUR A 100"
60 I=3:S=1
70 A#="RESTE DANS LA DIVISION PAR "
80 PRINT@2,11;E#;A#;I:PRINT@2,12;E#;A#;I
90 GET B$:IF B#<"0" OR B#>CHR$(47+I) THEN 90
100 A(S)=ASC(B#)-48
110 S=S+1:I=I+2
120 IF I<8 THEN 70
130 A#="RESTE :PAR 3      PAR 5      PAR 7"
140 PRINT@2,17;E#;A#:PRINT@2,18;E#;A#
150 FOR I=1 TO 3
160 FOR J=1 TO A(I)
170 IF A(I)<>0 THEN PRINT@10*I-4+2*J,15;D#;"A":CHR$(255)
180 NEXT J
190 ZAP:NEXT I
200 T=A(1)*70+A(2)*21+A(3)*15
210 IF T>105 THEN T=T-105:GOTO 210
220 PRINT@2,20;"LA SOLUTION EST :";T
230 PRINT@2,22;"VOULEZ-VOUS UN AUTRE ESSAI (O/N)"
240 GET B$:IF B#="O" THEN 20

```

VOUS DEVEZ ENTRER LES TROIS RESTES
DANS LES DIVISIONS PAR 3 , 5 , ET 7
D'UN NOMBRE ENTIER INFERIEUR A 100

RESTE DANS LA DIVISION PAR 7

※ ※ ※ ※ ※ ※ ※ ※

RESTE :PAR 3 PAR 5 PAR 7

LA SOLUTION EST : 88

VOULEZ-VOUS UN AUTRE ESSAI (O/N)

Chasse au sous-marin

REFERENCES :

Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : moyenne.

LE JEU :

Vous êtes à bord d'un bâtiment de surface, chargé de traquer un sous-marin caché quelque part dans votre secteur ; vous devez vous déplacer jusqu'à venir au-dessus du sous-marin et tirer à une certaine profondeur ; mais, attention : le sous-marin, lui aussi, est mobile ; il ne coupera ses moteurs que lorsque vous serez exactement à sa verticale, pour ne pas être repéré.

LE PROGRAMME :

La boucle I (60-120) affiche le terrain.
La ligne 140 définit la position initiale du sous-marin ; la ligne 150 permet son déplacement, contrôlé par la ligne 160 (on reste dans les limites du terrain).
La ligne 180 affiche votre position.
Les lignes 220 à 240 contrôlent votre déplacement.
La distance est calculée en ligne 250, affichée en ligne 260 et testée en lignes 270 et 280.
Le test de fin de partie est effectué à la ligne 310.

POUR JOUER :

On lance le programme par RUN et on entre la dimension ; la grille de jeu s'affiche. Vous êtes au départ en position (0, 0) et vous vous déplacez avec les flèches.

Après chaque déplacement, la nouvelle distance du sous-marin s'affiche ; à chaque coup le sous-marin se déplace. Si vous réussissez à vous placer à sa verticale, la distance 0 est affichée et il ne bouge plus.

Vous tirez en donnant une profondeur entière (entre 2 et 4, bornes comprises). Si vous le ratez, il se replace n'importe où, et on recommence ; si vous le coulez, vous obtenez l'affichage BRAVO et la partie s'arrête.

On peut arrêter à tout moment le programme en utilisant la touche F.

EXTENSIONS POSSIBLES :

Il est possible de prévoir une amélioration du graphique, d'améliorer les possibilités de déplacement du chasseur (un peu lent, dans cette version), de prévoir des parcours plus sophistiqués pour le sous-marin, de mettre un compteur de temps...

```

10 REM CHASSE AU SOUS-MARIN
20 D$=CHR$(27):E$=D$+"A":F$=D$+"D"
30 PAPER 6:INK 4:G$=CHR$(255):PRINT CHR$(17)
40 CLS:PRINT:INPUT"DIMENSION DE LA GRILLE :";H
50 CLS
60 FOR I=0 TO H
70 PRINT@2*I+5,0;I
80 PRINT@2,2*I+2;I
90 FOR J=0 TO H
100 PRINT@2*J+6,2*I+2;". "
110 NEXT J
120 NEXT I
130 I=1:X=0:Y=0
140 V=INT(RND(1)*H):U=INT(RND(1)*H)
150 U=U+I:V=V+I
160 IF (U<0 OR U>H OR V<0 OR V>H) THEN I=-I
170 IF Y<>0 THEN PRINT@2*Y+3,2*X+2;". "
180 PRINT@2*Y+5,2*X+2;E$:G$:F$
190 IF Y<>H THEN PRINT@2*Y+7,2*X+2;". "
200 IF X<>0 THEN PRINT@2*Y+5,2*X;". "
210 IF X<>H THEN PRINT@2*Y+5,2*X+4;". "
220 GET A$:IF A$="F" THEN END
230 Y=Y-(A$=CHR$(9) AND Y<H)+(A$=CHR$(8) AND Y>0)
240 X=X-(A$=CHR$(10) AND X<H)+(A$=CHR$(11) AND X>0)
250 D=INT(SQR(ABS(X-V)^2+ABS(Y-U)^2))
260 PRINT@31,10;"DIST. ";STR$(D)+" ";:PING
270 IF D=0 THEN 300
280 IF D<3 THEN 150
290 GOTO 170
300 PRINT@31,12;"PROF. ";:INPUT D
310 IF D=INT(RND(1)*3+2) THEN PRINT"BRAVO":EXPLODE:END
320 PRINT@31,12;"          ":ZAP:ZAP:ZAP
330 GOTO 140

```

```

      0 1 2 3 4 5 6 7 8
0  . . . . .
1  . . . . .
2  . . . . .
3  . . . . .
4  . . . . . DIST. 0
5  . . . . . PROF. ? 4
6  . . . . .
7  . # . . . . .
8  . . . . .

```

Clavier

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de réflexe (apprentissage du clavier).

Difficulté du jeu : difficile.

LE JEU :

Une reproduction du clavier de votre ordinateur apparaît sur l'écran ; un point apparaît en dessous d'un caractère, qu'il convient de frapper au clavier. Plus l'on frappe vite, et plus le point se déplace rapidement (le temps écoulé depuis le début s'affiche en fin de partie).

LE PROGRAMME :

Une variable A\$ dimensionnée en ligne 60, contiendra la liste des touches concernées. Les valeurs des touches sont entrées de la ligne 70 à la ligne 120 par appel d'un sous-programme situé en 300.

Les boucles emboîtées J (130-170) et I (140-160) dessinent le clavier.

La ligne 180 définit la position du point imprimé par la ligne 190.

La touche frappée est testée aux lignes 220-230 et le point est effacé par la ligne 250.

Les affichages de fin sont réalisés aux lignes 270 à 290.

POUR JOUER :

On lance le programme par RUN. L'on dispose alors de quelques secondes pour frapper la touche qui occupe la colonne au-dessus de l'étoile, puis une autre étoile est affichée, etc.

La durée de la partie n'est pas limitée ; il suffit de taper 0 pour obtenir l'affichage du score, et du temps total.

Faire RUN pour une autre partie.

EXTENSIONS POSSIBLES :

On pourra envisager de faire clignoter les lettres, ou de les afficher en couleur, au lieu d'indiquer la colonne par une balle. On pourra afficher les lettres devinées justes en bas de l'écran, et éventuellement s'entraîner à taper un texte qui défile en haut de l'écran, lettre par lettre.

```

10 REM CLAVIER
20 D$=CHR$(27):E$=D$+"J":PRINT CHR$(17)
30 PAPER 7:INK 0:CLS
40 FOR I=1 TO 3:PLOT1,4*I+1,1:PLOT1,4*I+2,1:
  PLOT1,4*I+3,2:PLOT1,4*I+4,2:NEXT I
50 T=0:S=0:P=0
60 DIM A$(10,3)
70 S$="QWERTYUIOP":J=1
80 GOSUB 300
90 S$="ASDFGHJKL;":J=2
100 GOSUB 300
110 S$="ZXCVBNM,./":J=3
120 GOSUB 300
130 FOR J=1 TO 3
140 FOR I=1 TO 10
150 PRINT@3*I+J,4*J+1;E$;A$(I,J):
  PRINT@3*I+J,4*J+2;E$;A$(I,J)
160 NEXT I
170 NEXT J
180 X=INT(RND(1)*10+1):Y=INT(RND(1)*3+1)
190 PRINT@3*X+Y,4*Y+3;E$;"*":PRINT@3*X+Y,4*Y+4;E$;"*"
200 FOR F=1 TO 80+RND(1)*100:A#=KEY$
210 T=T+1:IF A#="" THEN NEXT F
220 IF A#=A$(X,Y) THEN S=S+1
230 IF A#="0" THEN 270
240 P=P+1:PING
250 PRINT@3*X+Y,4*Y+3;E$;" ":PRINT@3*X+Y,4*Y+4;E$;" "
260 GOTO 180
270 PRINT@2,20;S;" TROUVES JUSTES SUR ";P;" ESSAIS"
280 PRINT:PRINT"VOTRE TEMPS :";T
290 EXPLODE:END
300 FOR I=1 TO 10:A$(I,J)=MID$(S$,I,1):NEXT I:RETURN

```

Q W E R T Y U I O P
A S D F G H J K L ;
Z X C V B N M , . /

22 TROUVES JUSTES SUR 37 ESSAIS
VOTRE TEMPS : 3111

Le lézard d'Alice

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez difficile.

LE JEU :

Il s'inspire très vaguement d'une idée de LEWIS CARROLL, dans "Alice au Pays des Merveilles". Dans l'histoire originale, Alice grandissait ou diminuait, alors que dans le jeu, pour des raisons d'encombrement de la mémoire, c'est un lézard qui change de taille.

Deux verres apparaissent sur l'écran : l'un contient le liquide qui fait grandir, l'autre le liquide qui fait rapetisser (malheureusement ce n'est pas toujours le même verre). Le joueur devra indiquer la quantité de liquide qu'il désire boire et le lézard changera de taille en fonction de cette quantité. Le but du jeu est de parvenir à lui rendre sa taille initiale.

LE PROGRAMME :

La boucle I (60-90) dessine les verres.

Les deux boucles emboîtées I (170-230) et J (180-210) les remplissent avec la quantité indiquée en ligne 150.

La variation de taille du lézard est définie par la ligne 260, avec contrôle par les lignes 270-280 et 290.

La boucle I (300-320) dessine le corps du lézard, les lignes 330 à 380 dessinent l'œil et les pattes.

Un message gagnant sera affiché par la ligne 400 si l'on a trouvé la bonne taille.

POUR JOUER :

On lance le programme par RUN : les deux verres se dessinent, puis apparaît la question QUEL COTE à laquelle on répond en tapant G ou D ; on obtient alors l'affichage QUELLE QUANTITE ; on entre alors une quantité (de 0 à 300) et on frappe RETURN.

Le verre désigné se remplit sous vos yeux, puis, après quelques instants, le lézard est affiché. On devra alors frapper une touche quelconque ; si le lézard n'avait pas la bonne taille, aucun message n'est affiché et l'on revient au dessin des verres, on tapera F pour finir.

EXTENSIONS POSSIBLES :

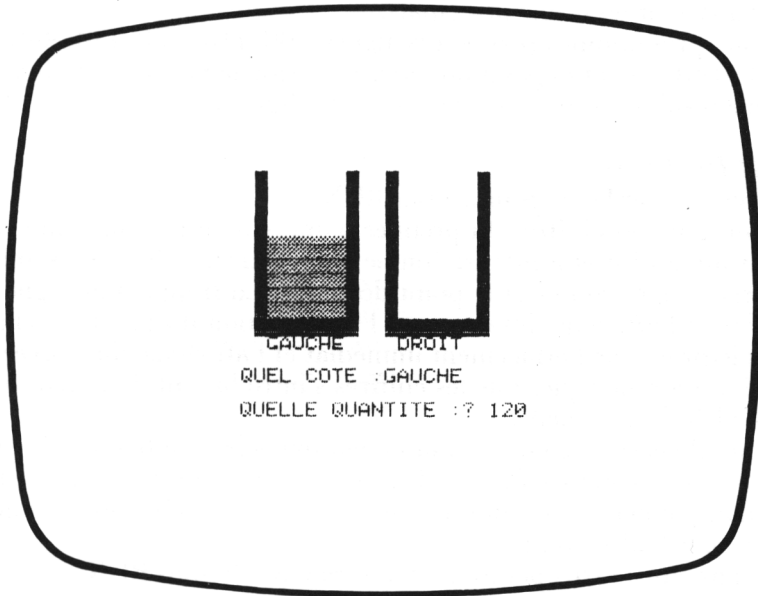
Puisqu'il s'agit d'un jeu essentiellement graphique, il sera intéressant d'améliorer les dessins, d'introduire la couleur, de modifier le caractère aléatoire du choix des verres, etc.

Pourquoi ne pas relire Alice au Pays des Merveilles, les idées n'y manquent pas.


```

10 REM LE LEZARD D'ALICE
20 D$=CHR$(27):E$=D$+"V":F$=CHR$(255):PRINT CHR$(17)
30 PAPER 6:INK 1
40 CLS
50 N=18
60 FOR I=1 TO 9
70 PRINT@3,I+4;F$:PRINT@10,I+4;F$
80 PRINT@13,I+4;F$:PRINT@20,I+4;F$
90 NEXT I
100 FOR I=3 TO 10:PRINT@I,14;F$:PRINT@I+10,14;F$:NEXT I
110 PRINT@4,15;"GAUCHE":PRINT@14,15;"DROIT"
120 PRINT@2,17;"QUEL COTE :";
130 A$=KEY$:IF A$(">")="D" AND A$(">")="G" THEN 130
140 IF A$="D" THEN PRINT"DROIT" ELSE PRINT"GAUCHE"
150 PRINT:PRINT"QUELLE QUANTITE :";:INPUT Q
160 U=4-10*(A$="D")
170 FOR I=1 TO SQR(Q)/2
180 FOR J=0 TO 5
190 PRINT@U+J,14-I;CHR$(148);
200 PING
210 NEXT J
220 PRINTCHR$(9);E$
230 NEXT I
240 ZAP:ZAP:ZAP:GET A$
250 PAPER 0:HIRE$
260 H=2*(RND(1)<0.5)+1
270 N=N+H*INT(SQR(Q))
280 IF N<12 THEN N=12
290 IF N>60 THEN N=60
300 FOR I=0 TO 4*(N+3)
310 CURSETI,100,1:CURSETI,99,1:CURSETI,98,1:CURSETI,97,1
320 NEXT I
330 T=4*(N-INT((SQR(N)/2)))
340 CURSETT,88,2:CURSETT-1,89,2:CURSETT-2,90,2:
CURSETT+1,89,2:CURSETT+2,90,2
350 T=4*SQR(N):CURSETT,102,2:CURSETT-1,103,2:CURSETT-2,104,2
360 T=T+2:CURSETT,102,2:CURSETT+1,103,2:CURSETT+2,104,2
370 T=4*(N-SQR(N)):CURSETT,102,2:CURSETT-1,103,2:CURSETT-2,104,2
380 T=T+2:CURSETT,102,2:CURSETT+1,103,2:CURSETT+2,104,2
390 GET A$:IF A$="F" THEN TEXT:END
400 TEXT:IF N=28 THEN PRINT"BRAVO":EXPLODE:END
410 GOTO 30

```



Calculatrice

REFERENCES :

Difficulté de programmation : très difficile.

Catégorie de jeu : simulation d'une calculatrice.

Difficulté du jeu : moyenne.

LE JEU :

Il s'agit de la simulation, en affichage grand format, du fonctionnement d'une petite calculatrice de poche, de type 4 opérations. Les applications pédagogiques de ce programme sont évidentes et son fonctionnement très simple à maîtriser.

LE PROGRAMME :

Les variables B\$ et C\$, des lignes 20 et 30, contiennent les informations codées qui permettront le dessin des chiffres en grand format.

Les lignes 60 à 150 s'occupent de la gestion des entrées (la boucle I (160-190) réalise la représentation, en grand format, des chiffres sur l'écran, au fur et à mesure des entrées).

Le point décimal et le signe moins sont gérés par les lignes 200 et 210.

Le nombre total des symboles entrés est contrôlé par la ligne 220.

Le résultat est inscrit grâce à un programme fonctionnant en automatique qui se trouve aux lignes 270 à 390 (l'automatisme est géré par la ligne 230, dans le programme principal).

Les sous-programmes placés aux lignes 400-410, 420-430, 440-450 et 460-470 effectuent l'opération indiquée, selon la valeur de Z (ligne 310). Chaque résultat est mémorisé en A (F-1) (ligne 330).

POUR JOUER :

On devra lancer le programme par RUN.

Puis, on tape les chiffres du premier nombre à entrer qui vont s'afficher un par un, au fur et à mesure (on ne pourra entrer plus de 8 symboles, y compris le signe moins et le point décimal). La frappe d'un signe opératoire laisse l'affichage intact, mais l'introduction du prochain chiffre ou symbole provoque l'effacement immédiat et l'affichage du nouveau symbole. Il est à remarquer que les chiffres s'introduisent de gauche à droite, sens de l'écriture usuelle.

L'appareil accepte, comme signes opératoires les symboles +, -, *, / ; le signe = provoquera l'affichage du résultat. Les opérations sont enchaînées automatiquement, le résultat précédent étant pris en compte comme opérateur du calcul suivant.

Pour faire une autre opération, il faudra taper F, puis RUN.

Taper F pour sortir du programme.

EXTENSIONS POSSIBLES :

On devra, tout d'abord, améliorer l'affichage du signe moins (représenté, ici, par un trait), puis on pourra améliorer le dessin des chiffres (en utilisant, par exemple, le générateur de l'ORIC-ATMOS), on peut aussi envisager une extension à plus de 8 chiffres, autoriser des opérations plus complexes (COS, SIN, ...), ajouter des mémoires, etc.



123.567

```

10 REM CALCULATRICE
20 E$=CHR$(255):C$=E$+E$+E$+E$+" "+E$+E$+" "+E$
30 B$="144413333313121131314413312131121411333314141114131"
40 F=1:L=0:PAPER 3:INK 1:CLS:PRINT CHR$(17)
50 N=0:T=0:K=0:R$=""
60 GET A$:IF A$="F" THEN END
70 IF K=1 THEN N=0:R$="":K=0
80 IF A$="" THEN 260
90 X=(1 AND A$="+")+(2 AND A$="-")+(3
  AND A$="*")+(4 AND A$="/")
100 R$=R$+A$
110 IF X<>0 THEN 260
120 IF R$="-" THEN A$=R$:GOTO 200
130 IF A$< "." OR A$>"9" OR A$="/" THEN 60
140 IF N=0 THEN CLS
150 IF A$="." OR A$="-" THEN 200
160 FOR I=1 TO 5
170 S=3*VAL(MID$(B$,I+5*VAL(A$),1))-2
180 PRINT@N+2,I+3;MID$(C$,S,3)
190 NEXT I
200 IF A$="-" THEN CLS:PRINT@3,7;E$;E$
210 IF A$="." THEN PRINT@N+3,8;E$
220 N=N-4*(N<28)
230 IF T<>0 THEN T=T+1:IF T<=LEN(R$) THEN 380
240 IF K=1 AND L<>1 THEN 50
250 PING:T=0:GOTO 60
260 IF R$="-" AND A(F)=0 THEN 120
270 N=0:K=1
280 IF (A$="*" OR A$="/") AND VAL(R$)=0 THEN A(F)=1:GOTO 300
290 A(F)=VAL(R$)
300 IF X<>0 THEN Z=X
310 ON Z GOSUB 400,420,440,460
320 IF A$<>"" THEN F=F+1 ELSE F=2
330 A(F-1)=VAL(R$)
340 IF VAL(R$)=0 THEN A(F-1)=A(F)
350 R$=STR$(VAL(R$))
360 IF LEFT$(R$,1)=" " THEN R$=RIGHT$(R$,LEN(R$)-1)
370 T=1
380 A$=MID$(R$,T,1)
390 GOTO 140
400 R$=STR$(A(F-1)+A(F)):A(F)=VAL(R$):L=1
410 RETURN
420 IF F<>1 THEN R$=STR$(A(F-1)-A(F)):A(F)=VAL(R$):L=1
430 RETURN
440 IF F<>1 THEN R$=STR$(A(F-1)*A(F)):A(F)=VAL(R$):L=1
450 RETURN
460 IF F<>1 THEN R$=STR$(A(F-1)/A(F)):A(F)=VAL(R$):L=1
470 RETURN

```

Niveau 5

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Smokey	Fumez sans risque, avec ce programme original pour un joueur.
Course de voitures	Un jeu de réflexe pas trop difficile, pour un joueur.
Echecs	Un programme assez simplifié, pour deux joueurs.
Danger spatial	Serez-vous assez adroit pour évitez toutes les météorites ?
Tic Tac Toe	Un classique, mais pas sans intérêt (deux joueurs)
Chenille	Un jeu d'action à plusieurs tableaux.
Dames	Pour deux joueurs : classique.
Quatre à quatre	Un jeu de réflexion un peu difficile, pour deux joueurs.
Anagrammes	Un bon jeu de réflexe : pour deux joueurs.
Oeufs	Ne laissez pas faire la pondeuse diabolique (un joueur).
Taquin de lettres	Un proverbe est caché dedans : à vous de le découvrir.
Cibles	Un peu d'adresse, beaucoup de chance : très visuel.
Gouttes	Un véritable gag visuel. A exploiter.
Obstacles	Beaucoup d'obstacles à éviter dans ce jeu d'adresse pour un joueur.
Captures dans l'espace	Votre mission serait facile, sans ces damnées météorites.
Anomalie	Exercez votre perspicacité dans la reconnaissance des symboles.
Plusieurs en un	Un terrain de jeu et les pions, pour deux joueurs.
Parcours du cavalier	Très spectaculaire, et instructif à plus d'un titre.
Mur de briques	Un grand classique dans une version très complète.
Othello	Pour deux joueurs ; il retourne les pions à votre place.

Les techniques évoluées de programmation

LES INSTRUCTIONS D'EXPLORATION DES CASES MEMOIRES : PEEK et POKE

Pour compléter la panoplie des instructions disponibles en standard sur l'ORIC-ATMOS, on sera, dans certains cas, obligé de faire appel à des instructions permettant la manipulation directe dans les mémoires de l'appareil ;

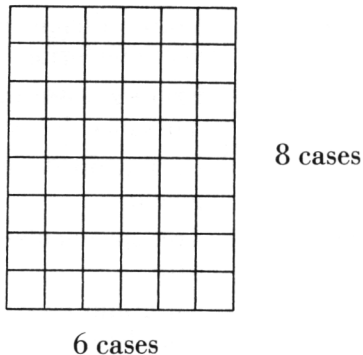
La fonction PEEK permet la lecture directe du contenu de l'une des cases mémoires adressables par le microprocesseur.

La fonction POKE permet, au contraire, l'inscription directe d'une quantité, décimale ou hexadécimale, dans l'une de ces cases mémoires (la quantité est nécessairement un entier compris entre 0 et 255).

LES CARACTERES GRAPHIQUES DEFINIS PAR L'UTILISATEUR

L'utilisateur dispose de la possibilité de remplacer n'importe quel caractère de son choix par un caractère (ou un dessin) qu'il aura lui-même défini, en "POKANT" dans les cases mémoires convenables.

Chaque caractère sera défini par allumage ou extinction d'un point dans un rectangle de 6 cases sur 8, selon le schéma :



Chaque suite de 6 caractères est transformée en un nombre de 0 à 63, huit nombres définissant le dessin élémentaire.

LA RECONNAISSANCE D'UN OBJET SUR L'ECRAN : LECTURE DU CARACTERE PLACE EN COLONNE X, LIGNE Y PAR SCRN (X, Y)

L'instruction SCRN (X, Y) permet de reconnaître le code ASCII du caractère présent sur l'écran en colonne X et en ligne Y.

On pourra reconnaître par cette méthode les caractères agrandis ou les caractères définis par l'utilisateur.

Smokey

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

Une colonne de fumée grise apparaît sur l'écran. Le joueur, représenté par un cœur, devient visible en haut de l'écran. Il doit dans un premier temps se positionner sur la colonne de fumée, puis à partir de là, il va automatiquement avaler la fumée, à condition qu'il réussisse à se maintenir sur sa trace.

LE PROGRAMME :

La ligne 90 fait apparaître la fumée au bas de l'écran ; la ligne 100 permet de changer (vers la droite ou vers la gauche) la direction de la colonne, et la ligne 120 fait monter la colonne.

Le joueur apparaît grâce à la ligne 140, et son déplacement est effectué par les lignes 180 et 190.

Les lignes 250 et 260 se chargent de l'affichage du résultat.

POUR JOUER :

On doit lancer l'exécution du programme par RUN.

La fumée commence à s'élever, tandis que le joueur, apparu sous forme d'un cœur dans un coin de l'écran, va se déplacer sur elle à l'aide des 4 flèches. Dès qu'il arrive sur la fumée, il va automatiquement commencer à l'avalier, et devra essayer de suivre la colonne sans dévier (grâce aux flèches). Il gagne s'il arrive en bas de l'écran avant que la fumée n'arrive en haut.

L'affichage final indique alors BRAVO ou PERDU, ainsi que le score obtenu.

EXTENSIONS POSSIBLES :

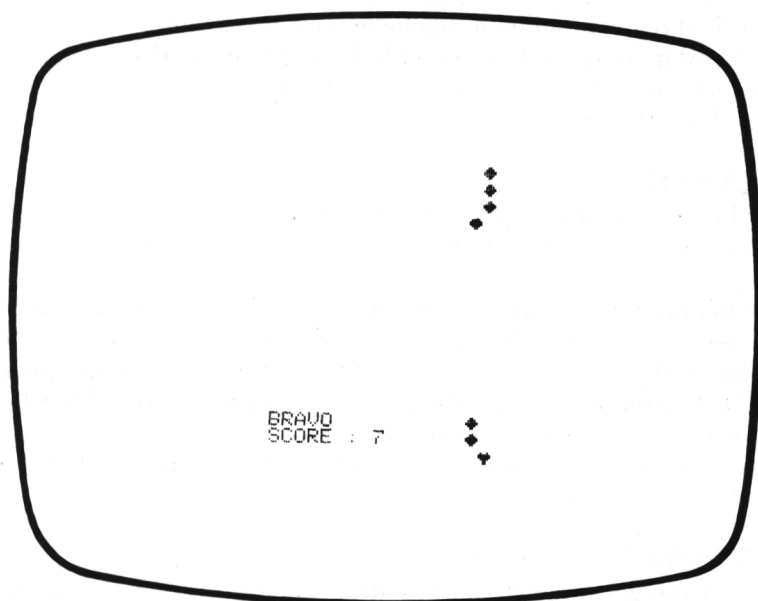
On pourra, tout en conservant l'idée de départ, imposer d'autres règles : par exemple avaler toute la fumée, qui pourra aussi repartir d'un autre endroit si l'on dévie de la colonne initiale.

On pourra aussi améliorer les contrôles de fin de partie, et la qualité du graphisme.


```

10 REM SMOKEY
20 DATA 0,12,30,63,63,30,12,0,0,0,18,63,63,30,12,12
30 FOR I=46856 TO 46871:READ A:POKE I,A:NEXT I
40 PAPER 3:INK 1:PRINT CHR$(17)
50 D$=CHR$(27):E$=D$+"D":F$=D$+"R"
60 S=0:X=24:Y=INT(RND(1)*24)
70 U=0:V=15
80 CLS
90 PRINT@Y+2,X;"a"
100 Y=Y-(RND(1)<0.5 AND Y<24)+(RND(1)<0.5 AND Y>0)
110 T=1
120 X=X-T
130 IF X=0 THEN 250
140 PRINT@V+1,U;E$;"b":F$
150 IF T<>0 THEN FOR I=1 TO 300:A$=KEY$:IF A$<>" " THEN 170
160 IF T<>0 THEN NEXT I
170 PRINT@V+1,U;" "
180 U=U-(A$=CHR$(10) AND T=1 AND X<24)
190 U=U-(T=0)
200 V=V-(A$=CHR$(9) AND V<24)+(A$=CHR$(8) AND V>1)
210 C=SCRN(V+2,U)
220 IF C<>97 THEN 90
230 T=0:S=S+1
240 IF U<23 THEN 140
250 IF T=0 THEN PRINT"BRAVO":ZAP:ZAP:ZAP:GOTO 270
260 PRINT"PERDU....":EXPLODE
270 PRINT"SCORE :";S

```



Course de voitures

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de réflexe (1 joueur).

Difficulté du jeu : moyenne.

LE JEU :

Vous circulez sur une route très sinueuse, ne comportant aucun obstacle, et vous devez essayer de la suivre de plus longtemps possible. La trace de votre parcours reste affichée en permanence, et votre score est affiché dès que vous avez un accident.

LE PROGRAMME :

L'impression de la route est assurée par la boucle I (80-100).

La ligne 110 fait avancer la route, et la ligne 120 imprime la voiture.

La ligne 150 contrôle son mouvement, et la ligne 160 détermine si la route fait ou non des virages.

La ligne 170 regarde si l'on a touché le bord de la route.

La ligne 180 incrémente le score, et la ligne 190 efface l'ancienne position de la voiture (une trace claire apparaît à l'emplacement de sa trajectoire).

La ligne 220 affiche le temps avant l'accident.

POUR JOUER :

Lancer l'exécution du programme par l'ordre RUN, le dessin de la route commence à évoluer, et votre véhicule, représenté par une voiture, apparaît.

Pour le déplacer vers la droite ou vers la gauche, vous devrez utiliser les touches → et ← (flèches de déplacement du curseur). Il vous faudra éviter le plus longtemps possible de toucher une bordure noire, ce qui provoquera l'arrêt du programme et l'affichage de la mention ACCIDENT, suivie de l'indication sur le score réalisé.

On pourra arrêter le déroulement de la partie à tout moment en pressant les touches CTRL et C.

EXTENSIONS POSSIBLES :

On peut envisager de jouer à plusieurs, chacun son tour, en enregistrant le meilleur score.

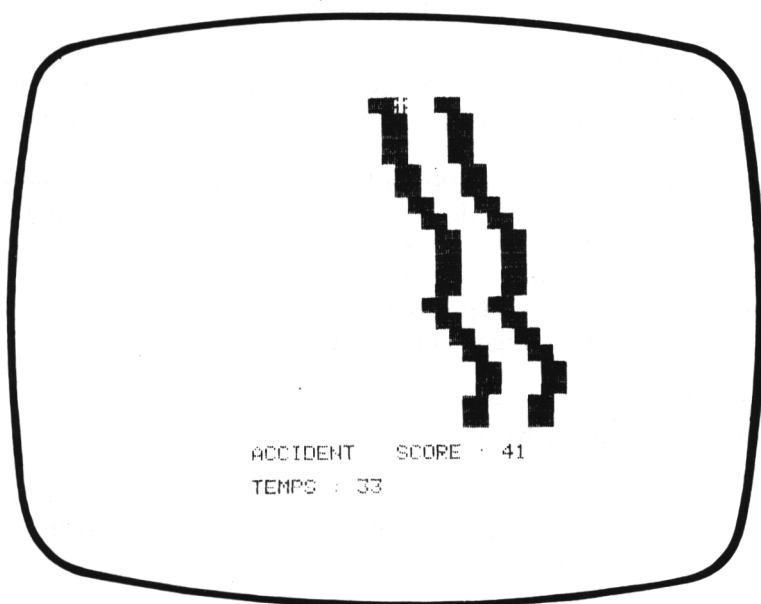
On peut aussi envisager d'élargir la piste, et de faire courir plusieurs voitures.

On pourra aussi s'inspirer des jeux d'arcades qui sont particulièrement nombreux sur ce thème.

```

10 REM COURSE DE VOITURES
20 DATA 12,45,63,45,12,45,63,45
30 FOR I=46856 TO 46863:READ A:POKE I,A:NEXT I
40 PAPER 6:INK 4:D$=CHR$(255)
50 T=0:CLS
60 X=25:Y=11
70 A=5:B=10:S=0
80 FOR I=0 TO 6
90 IF I<2 OR I>4 THEN PRINT@Y+I,X;D$
100 NEXT I
110 PRINT
120 PLOTB,A,225
130 FOR I=1 TO 10:A$=KEY$:T=T+0.1:IF A$="" THEN NEXT I
140 C=B
150 B=B-(A$=CHR$(9))+(A$=CHR$(8))
160 Y=Y+(2*RND(1) AND Y<25)-(2*RND(1) AND Y>3)
170 IF SCRNB(A+1)=127 THEN 210
180 S=S+1
190 PLOTB,A,32
200 GOTO 80
210 PRINT"ACCIDENT . SCORE :";S:EXPLODE
220 PRINT:PRINT"TEMPS :";INT(T)

```



Echecs

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de réflexion, pour deux joueurs.

Difficulté du jeu : difficile.

LE JEU :

Le programme propose seulement à deux joueurs de disputer une partie en se servant de l'écran du téléviseur comme plateau de jeu. Il n'y a pas de contrôle de la validité des coups, mais seulement gestion de la position des pièces sur l'échiquier. Les pions et les pièces sont dessinés à l'aide des caractères graphiques GR\$. La couleur permet de distinguer chaque camp.

LE PROGRAMME :

Les lignes 100 à 200 dessinent l'échiquier. Les lignes 210 et 220 définissent les conditions initiales, à l'aide d'un tableau : R (I, J) pour les codes des pièces.

Les boucles emboîtées J (240-290) et I (250-280) dessinent les pièces.

La ligne 320 permet l'entrée de la ligne et de la colonne de départ, avec contrôle en ligne 350.

La ligne 370 permet d'entrer colonne et ligne d'arrivée.

La ligne 400 note la nouvelle position.

La ligne 410 assure le changement de joueur.

POUR JOUER :

On lance l'exécution par RUN.

L'échiquier est dessiné, puis les pièces, et enfin l'inscription JOUEUR ROUGE suivie de DEPART (C et L) ; on entre la colonne et la ligne de départ, sous la forme E 7 par exemple, on obtient alors l'affichage de ARRIVEE (C et L) et on répond de même. Le déplacement est alors effectué, et c'est au tour de l'autre joueur.

Pour arrêter, il suffit de taper CTRL et C.

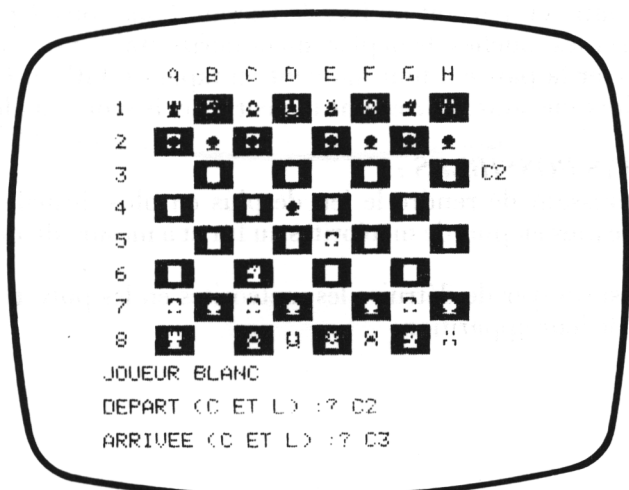
EXTENSIONS POSSIBLES :

On pourra chercher à réaliser un graphisme plus précis, introduire le contrôle du mouvement des pièces. Et pourquoi pas, écrire un programme qui permette à l'ORIC-ATMOS de se comporter en partenaire honorable. Bon courage !

```

10 REM ECHECS
20 DIM R(8,8),C(8,8):DATA12,30,18,33,33,30,30,63,18,12,45,30,12,51,63,63
30 DATA45,45,63,63,12,30,30,63,12,12,30,30,12,30,33,30
40 DATA0,12,30,63,63,30,12,63,5,31,59,63,6,6,63,63,0,0,0,0,0,0
50 FOR I=46856 TO 46911:READ A:POKE I,A:NEXT I
60 PAPER 6:INK 0:CLS
70 B$=CHR$(255)+CHR$(255)+CHR$(255)
80 A$="PPP":C$="###":PRINT CHR$(17)
90 FOR I=1 TO 9:PRINT@2,2*I-1;CHR$(27)+"I":NEXT I
100 PRINT@2,0;"      A B C D E F G H"
110 FOR I=1 TO 8
120 FOR J=1 TO 8
130 IF (I+J)/2=INT((I+J)/2) THEN 150
140 PRINT@3*J+3,2*I-1;A$:PRINTI:PRINT@3*J+3,2*I;B$:PRINT@3*J+3,2*I+1;C$
150 IF I=2 OR I=7 THEN R(I,J)=4
160 IF I<>2 AND I<>7 THEN R(I,J)=6
170 C(I,J)=0
180 IF I=7 OR I=8 THEN C(I,J)=128
190 NEXT J
200 NEXT I
210 R(1,1)=2:R(1,2)=5:R(1,3)=0:R(1,4)=3:R(1,5)=1:
    R(1,6)=0:R(1,7)=5:R(1,8)=2
220 FOR I=1 TO 8:R(0,I)=R(1,I):NEXT I
230 H=1
240 FOR J=1 TO 8
250 FOR I=1 TO 8
260 X=1+3*J:Y=2*I:IF R(I,J)=6 THEN C(I,J)=0
270 PLOTX+3,Y,97+R(I,J)+C(I,J)
280 NEXT I
290 NEXT J
300 PRINT@31,6;"      ":PRINT@20,20;"      ":PRINT@21,22;"      ":PING
310 PRINT@2,18;:IF H=1 THEN PRINT"JOUEUR ROUGE" ELSE PRINT"JOUEUR BLANC"
320 PRINT@2,20;"DEPART (C ET L) :?":INPUT D$
330 PRINT@31,6;D$
340 GOSUB 500
350 IF R(L,C)=6 OR (C(L,C)=0 AND H=1) OR (C(L,C)=128 AND H=0) THEN 310
360 T=R(L,C):S=C(L,C):R(L,C)=6
370 PRINT@2,22;"ARRIVEE (C ET L) :?":INPUT D$
380 PRINT@34,6;"- ":D$
390 GOSUB 500
400 R(L,C)=T:C(L,C)=S
410 H=ABS(H-1)
420 GOTO 240
500 L=VAL(RIGHT$(D$,1)):C=ASC(LEFT$(D$,1))-64
510 RETURN

```



Danger spatial

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de réflexe, pour un joueur.

Difficulté du jeu : moyenne.

LE JEU :

Vous êtes aux commandes d'un engin spatial qui vient de pénétrer dans un nuage de météorites. Vous devez essayer de franchir le nuage sans heurter une seule météorite. Votre score sera affiché en fin de partie, lorsque vous entrez en collision avec une météorite.

LE PROGRAMME :

La position de chaque météorite est déterminée par les lignes 60 et 80 ; elle est imprimée par les lignes 70 et 90.

L'avance des météorites est réalisée par la ligne 100.

Les lignes 100 et 110 vérifient si vous avez ou non heurté une météorite.

La ligne 130 dessine votre vaisseau spatial, et la ligne 160 permet de le déplacer.

Le compteur se trouve en ligne 170.

POUR JOUER :

Le jeu est lancé par l'ordre RUN.

Votre vaisseau apparaît vers le milieu de l'écran, et les météorites viennent à votre rencontre en partant du bas de l'écran. Pour vous déplacer, vous pouvez utiliser les touches de déplacement horizontal (→ et ←).

On peut arrêter la partie à tout moment en tapant CTRL et C. Sinon, le jeu s'arrête lorsque vous heurtez une météorite, et le score est alors affiché.

EXTENSIONS POSSIBLES :

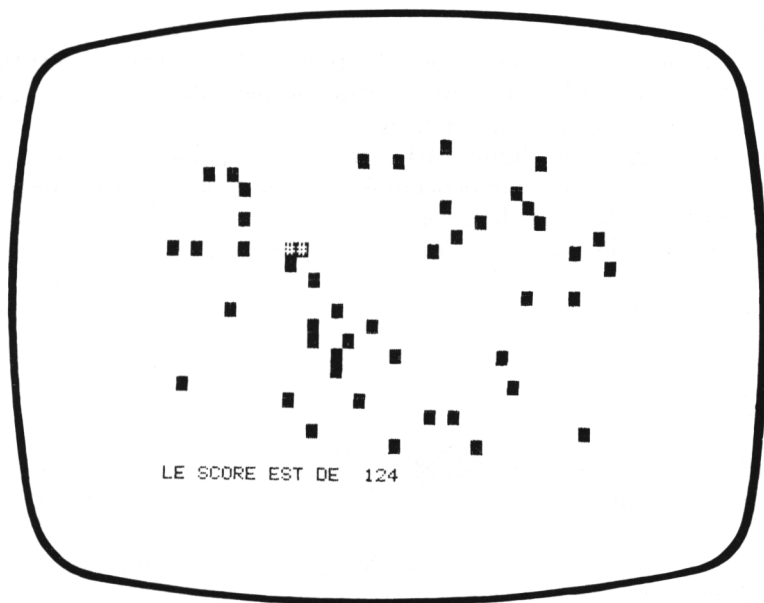
Il serait intéressant de rendre le jeu de plus en plus difficile en faisant apparaître de plus en plus de météorites au fur et à mesure du déroulement de la partie.

On peut aussi essayer de détruire les météorites en les pulvérisant au fur et à mesure de leur apparition.

```

10 REM DANGER SPATIAL
20 D$=CHR$(255)
30 PAPER 6:INK 4
40 CLS
50 X=18:Z=0:Y=X
60 H=INT(RND(1)*38+2)
70 IF RND(1)<0.7 THEN PRINT@H,25;D$;
80 K=INT(RND(1)*38+2)
90 IF RND(1)<0.6 THEN PRINT@K,25;D$
100 PRINT:B=SCRN(X,12):C=SCRN(X+1,12)
110 IF B=127 OR C=127 THEN 190
120 PRINT@Y-1,11;"    ":PRINT@Y-1,10;"    "
130 PLOTX,12,CHR$(163)+CHR$(163)
140 A$=KEY$
150 Y=X
160 X=X-(A$=CHR$(9) AND X<38)+(A$=CHR$(8) AND X>2)
170 Z=Z+1
180 GOTO 60
190 PRINT"LE SCORE EST DE ";Z:EXPLODE

```



Tic tac toe

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion (deux joueurs).

Difficulté du jeu : facile.

LE JEU :

Un grand classique des jeux d'écoliers, proposé ici dans une version ne comportant pas le jeu machine ; chaque joueur devra placer à son tour un Ø ou un X (ici un cœur ou un carreau) dans une case libre de la grille de jeu, et le premier qui réussit un alignement de 3 symboles identiques a gagné (l'alignement peut être horizontal, vertical, ou suivant l'une des deux diagonales).

LE PROGRAMME :

Le terrain de jeu est dessiné à l'aide des boucles I (5Ø-9Ø) et J (6Ø-9Ø).

La ligne 11Ø et J (6Ø-9Ø) s'occupe d'afficher le symbole cœur ou carreau correspondant au joueur qui a le trait.

Un symbole clignotant (ligne 14Ø) est affiché en haut et à gauche du terrain, même si la case est déjà occupée ; les lignes 18Ø et 19Ø permettent de déplacer le symbole sur le terrain de jeu.

La ligne 20Ø vérifie si la case où l'on veut jouer est libre, et si l'on désire jouer.

La ligne 21Ø affiche le coup, et la ligne 22Ø réalise le changement de joueur.

POUR JOUER :

Le terrain de jeu sera affiché dès la frappe de RUN.

Un symbole clignotant apparaît alors en haut et à gauche de l'écran ; le joueur peut le déplacer sur le terrain de jeu à l'aide des 4 flèches (touches ←, → et ↑, ↓).

Lorsqu'il arrive sur la case où il désire jouer, il suffit qu'il appuie sur la touche Ø, le coup sera refusé si la case n'est pas libre.

Si la case est libre, le symbole qui correspond à ce joueur est inscrit, et on change de joueur.

Lorsque la partie est finie, il convient de taper CTRL et C pour sortir du programme. On pourra ensuite faire une autre partie avec RUN.

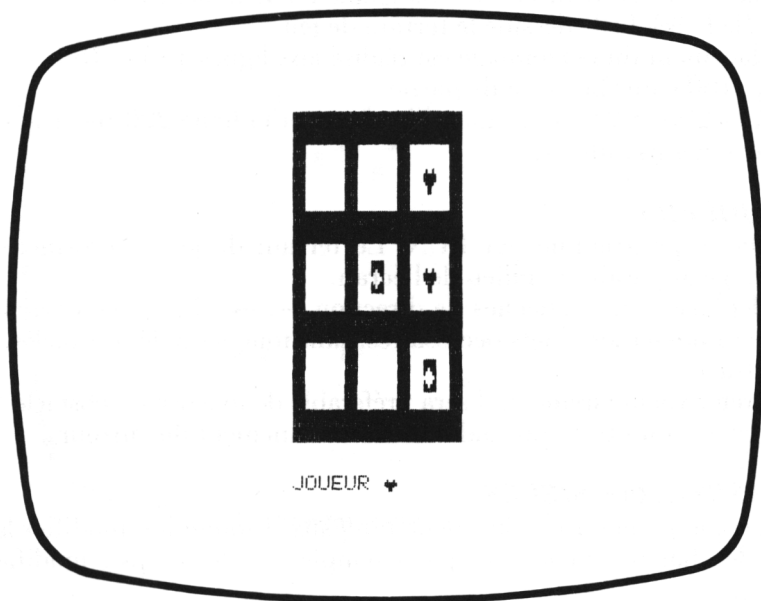
EXTENSIONS POSSIBLES :

On pourrait réaliser après chaque coup un contrôle pour vérifier si l'un des joueurs a gagné. On peut aussi réaliser un jeu machine mais il faudrait éviter qu'il soit par trop systématique (par exemple, la machine jouant toujours son premier coup sur la case centrale).


```

10 REM TIC TAC TOE
20 DATA 0,0,18,63,63,30,12,12,0,0,8,28,62,28,8,0
30 FOR I=46856 TO 46871:READ A:POKE I,A:NEXT I
40 PAPER 3:INK 1:D$=CHR$(27):E$=D$+"J":F$=CHR$(255):
  CLS:PRINT CHR$(17)
50 FOR I=0 TO 12 STEP 4
60 FOR J=2 TO 18
70 PRINT@I+2,J-1;F$
80 IF J<15 THEN PRINT@J,I+2*INT(I/4);F$:PRINT@J,I+2*
  INT(I/4)+1;F$
90 NEXT J:NEXT I
100 K=0
110 PRINT@2,22;"JOUEUR ";CHR$(97+K)
120 I=1:J=1
130 C=SCRN(4*I,6*J-2):D=SCRN(4*I,6*J-3)
140 PRINT@4*I-1,6*J-2;E$;"X":
  PRINT@4*I-1,6*J-3;E$;"X"
150 IF C<>32 THEN PLOT4*I,6*J-2,C:PLOT4*I,6*J-3,D
160 IF C=32 THEN PLOT4*I,6*J-2,32:PLOT4*I,6*J-3,32
170 A$=KEY$
180 J=J-(A$=CHR$(10) AND J<3)+(A$=CHR$(11) AND J>1)
190 I=I-(A$=CHR$(9) AND I<3)+(A$=CHR$(8) AND I>1)
200 IF C<>32 OR A$<>"0" THEN 130
210 PLOT4*I,6*J-2,97+K:PLOT4*I,6*J-3,97+K
220 K=ABS(129-K)
230 IF K=129 THEN PING ELSE ZAP
240 GOTO 110

```



Chenille

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : adresse.

Difficulté du jeu : difficile.

LE JEU :

Il présente la particularité d'offrir plusieurs niveaux de jeu (3 tableaux) : une chenille se déplace très rapidement sur l'écran, en laissant une trace ; elle doit ramasser une certaine quantité d'objets en un temps limité, sans heurter les obstacles déterminés sur son parcours, et sans repasser sur sa propre trace ; le passage au deuxième tableau est réalisé à 2500 points, et le troisième tableau à 5000 points. Mais, on joue en temps limité, et il faudra être très rapide.

LE PROGRAMME :

La ligne 20 permet de représenter un petit personnage sur l'écran.

La ligne 30 assure le passage en mode LORES, et utilise POKE pour accélérer l'action des touches (on pourra réutiliser avec profit ces POKE dans d'autres jeux).

Les lignes 60 et 70 réalisent les changements de tableaux.

La boucle I (80-110) dessine le terrain de jeu.

Le déplacement du personnage est réalisé aux lignes 140 et 150.

La ligne 160 contrôle la fin de partie.

Les lignes 200 et 210 s'occupent du score, et la ligne 220 regarde si l'on dépasse le temps autorisé.

POUR JOUER :

On lance le programme par RUN. Le terrain de jeu se dessine, et un personnage apparaît au milieu de l'écran.

On le déplace avec les flèches de direction ←, →, ↑, ↓, en essayant de lui faire ramasser les objets nécessaires, sans toucher ni les obstacles, ni sa propre trace.

Pour arrêter volontairement, il sera préférable de toucher un obstacle (pour restaurer les conditions normales de fonctionnement du curseur).

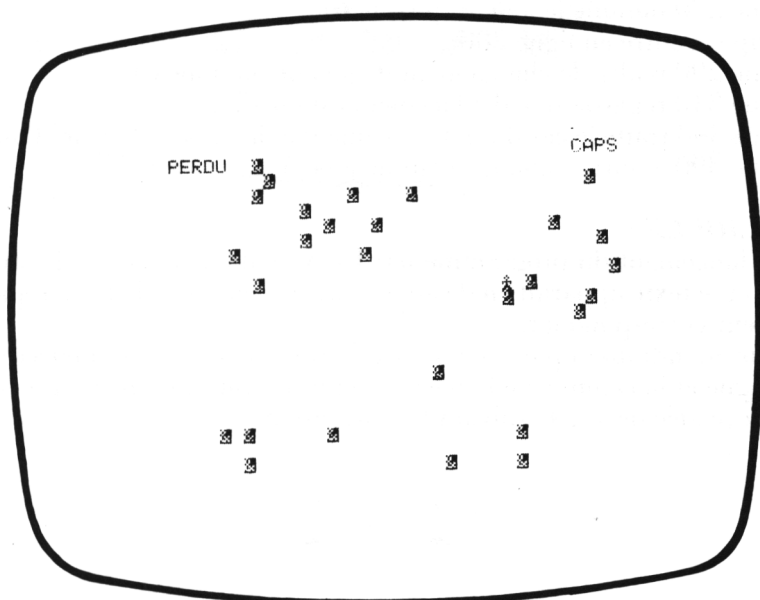
EXTENSIONS POSSIBLES :

On pourra augmenter le nombre de tableaux disponibles, modifier la difficulté de chacun (en jouant, par exemple, sur le temps), modifier les décors ...

```

10 REM CHENILLE
20 DATA 8,20,8,62,42,8,28,54:FOR I=46856
  TO 46863:READ A:POKE I,A:NEXT I
30 LORES 0:PAPER 0:INK 6:POKE#24E,1:POKE#24F,1
40 H=0:W=0:P=0:F=17:C#=CHR$(166)
50 A=10:B=15
60 IF H=1 THEN F=18:C#=CHR$(223)
70 IF H=2 THEN F=19:C#=CHR$(164)
80 FOR I=0 TO 50
90 PLOT INT(RND(1)*38+2),INT(RND(1)*22),F
100 IF I<32-20*(H=2) THEN PLOT INT(RND(1)*38+2),
  INT(RND(1)*22),C#
110 NEXT I
120 PLOTB,A,97
130 X=A:Y=B:A#=KEY$
140 A=A-(A#=CHR$(10) AND A<22)+(A#=CHR$(11) AND A>0)
150 B=B-(A#=CHR$(9) AND B<40)+(A#=CHR$(8) AND B>2)
160 IF SCRNB(B,A)=F OR SCRNB(B,A)=23 THEN 250
170 PRINT@2,0;"TEMPS ";W
180 W=W+1
190 PLOT Y,X,23
200 IF SCRNB(B,A)=ASC(C#) THEN P=P+100
210 PRINT@20,0;"SCORE ";P
220 IF W=500 THEN PRINT"TEMPS ECOULE":PING:WAIT 50:GOTO 250
230 IFP=2500ORP=5000THENH=H+1:ZAP:ZAP:
  LORES0:P=2600-3000*(H=2):W=0:GOTO40
240 GOTO 120
250 PRINT"PERDU":POKE#24E,32:POKE#24F,4:EXPLODE:END

```



Dames

REFERENCES :

Difficulté de programmation : très difficile.

Catégorie de jeu : deux joueurs.

Difficulté du jeu : difficile.

LE JEU :

C'est le jeu classique de dames à la française, mais un peu incomplet puisqu'il ne prévoit pas la possibilité de faire des dames. La règle du jeu reste classique, mais les contrôles ont été un peu simplifiés. Les mouvements des pièces sont entrés en une seule manœuvre (position de départ et direction du déplacement) et les prises sont réalisées automatiquement, si elles sont possibles.

Par contre le changement de joueur étant réalisé à chaque coup, il faudra éventuellement taper 0 pour pouvoir rejouer s'il y avait eu saut ou si le coup n'était pas possible.

LE PROGRAMME :

Les boucles I (80-170) et J (90-160) dessinent le damier et définissent les positions initiales des pions noirs et des pions blancs. Les positions et les couleurs sont dans une table R (X, Y).

L'impression des pions proprement dite est effectuée par les boucles emboîtées J (190-250) et I (200-240).

La ligne 270 indique le joueur qui a le trait.

Le coup est entré en ligne 280.

La ligne 290 réalise le changement de joueur (on tape 0).

La ligne 310 regarde si le déplacement est possible.

La ligne 340 traite le cas d'un déplacement et la ligne 350 celui d'un saut.

La ligne 390 assure le changement de joueur.

POUR JOUER :

Dès le lancement du programme par RUN, le damier est affiché avec ses pions, et le texte apparaît en dessous pour indiquer que le trait appartient au joueur correspondant.

On joue en indiquant par un mot de 3 lettres (pouvant commencer par un 0) la ligne et la colonne de la pièce à déplacer, puis la direction du déplacement (codée de 1 à 4) suivant la convention :



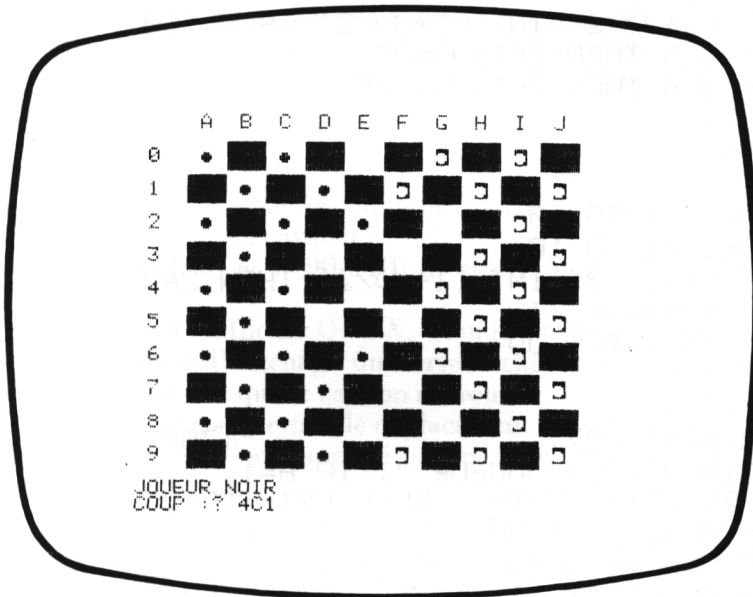
Par exemple 9D1 signifie que le O situé en ligne 9, colonne D, se déplace dans la direction 1 (ne pas oublier de faire RETURN).

De même 0G3 signifiera pion noir, ligne 0, colonne G, direction 3.

Le changement de joueur étant effectué automatiquement, il faudra taper Ø pour rejouer si le coup est refusé ou s'il y a eu prise (toujours terminer par RETURN). Le programme ne contrôle pas l'arrivée à dame, et n'interdit pas les déplacements en arrière.

EXTENSIONS POSSIBLES :

Il sera assez facile de commencer par améliorer les contrôles (changement de joueur, coup en arrière qui ne correspondrait pas à une prise, sortie de terrain). On pourra envisager la création de dames, leur déplacement, et éventuellement écrire un programme permettant à l'ORIC-ATMOS de se mêler avec de bonnes chances de réussite à vos passionnantes parties.



```

10 REM DAMES
20 DIM R(10,10):DATA 0,0,28,62,62,62,28,0
30 FOR I=46856 TO 46863:READ A:POKE I,A:NEXT I
40 B#=CHR$(255)+CHR$(255)+CHR$(255):
   PAPER 6:INK 0:CLS
50 A$="PPP":C$="###":PRINT CHR$(17)
60 FOR I=1 TO 11:PRINT@2,2*I-1;CHR$(27)+"I":NEXT I
70 PRINT@2,0;"      A B C D E F G H I J"
80 FOR I=1 TO 10
90 FOR J=1 TO 10
100 IF (I+J)/2=INT((I+J)/2) THEN 120
110 PRINT@3*J+3,2*I-1;A$:PRINTI-1:
   PRINT@3*J+3,2*I;B$:PRINT@3*J+3,2*I+1;C$
120 R(I,J)=0
130 IF (I+J)/2<>INT((I+J)/2) THEN 160
140 IF J<5 THEN R(I,J)=97
150 IF J>6 THEN R(I,J)=225
160 NEXT J
170 NEXT I
180 H=1
190 FOR J=1 TO 10
200 FOR I=1 TO 10
210 IF (I+J)/2<>INT((I+J)/2) THEN 240
220 X=3*J+4:Y=2*I
230 PLOTX,Y,R(I,J)
240 NEXT I
250 NEXT J
260 PRINT@2,23;"                               ":ZAP
270 PRINT@2,22;"JOUEUR ";:IF H=1
   THEN PRINT"NOIR " ELSE PRINT"ROUGE"
280 PRINT@2,23;"COUP ";:INPUT D$
290 IF D$="0" THEN H=ABS(H-1):GOTO 260
300 L=VAL(LEFT$(D$,1))+1:C=ASC(MID$(D$,2,1))-64:
   D=VAL(RIGHT$(D$,1))
310 IF R(L,C)=0 OR (R(L,C)=97
   AND H=0) OR (R(L,C)=225 AND H=1) THEN 260
320 T=(1 AND (D=1 OR D=2))-(1 AND (D=4 OR D=3))
330 S=(1 AND (D=3 OR D=2))-(1 AND(D=1 OR D=4))
340 IF R(L+S,C+T)=0
   THEN R(L,C)=0:R(L+S,C+T)=97-128*(H=0):GOTO 390
350 IF R(L+S,C+T)=97-128*(H=1)
   AND R(L+2*S,C+2*T)=0 THEN 370
360 GOTO 260
370 R(L,C)=0:R(L+S,C+T)=0:
   R(L+2*S,C+2*T)=97-128*(H=0)
380 GOTO 190
390 H=ABS(H-1):PING
400 GOTO 190

```

Quatre à quatre

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion (un ou deux joueurs).

Difficulté du jeu : difficile.

LE JEU :

Il pourra être pratiqué à un ou plusieurs joueurs. Dans les deux cas, le but est d'arriver à échanger les pions X avec les pions O ; si l'on joue en solitaire, cet échange devra être réalisé avec le moins de coups possible. Les pions sont déplacés alternativement en respectant les règles de déplacement du cavalier aux échecs ; il n'y a pas de prise, et on ne peut passer son tour.

LE PROGRAMME :

La variable alphanumérique A\$, en ligne 50, contient la table des coups possibles (déplacement du cavalier).

Les boucles I (70-110) et J (90-110) dessinent le terrain de jeu, et les positions de départ.

La ligne 130 permet d'afficher O ou X, selon le joueur qui a le trait.

La ligne 160 permet l'affichage du curseur clignotant (ici un pavé de couleur) qui indique sur quelle case on se trouve.

Les lignes 190 et 200 permettent le déplacement du curseur.

La ligne 210 regarde si l'on a appuyé sur une touche de 1 à 8, donnant la direction du déplacement.

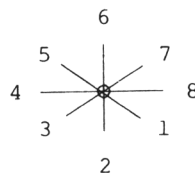
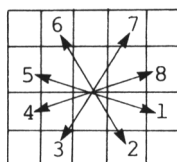
Les lignes 220 et 230 réimpriment la figure qui était sous le curseur. La ligne 240 transforme en nombre le symbole chiffre frappé. La ligne 250 réalise le contrôle de départ.

Les lignes 260 et 270 effectuent le déplacement. La ligne 300 réalise le contrôle d'arrivée. Le résultat est imprimé par la ligne 310, le changement de joueur est réalisé par la ligne 320 et un . est imprimé à l'emplacement de départ. Le sous-programme 340-350 est utilisé pour les divers contrôles.

POUR JOUER :

On lance le programme par RUN. Le terrain de jeu est affiché, ainsi que le symbole permettant d'identifier le joueur qui a le trait. Le joueur peut choisir le pion qu'il désire déplacer en manœuvrant le curseur clignotant grâce aux touches de déplacement du curseur.

Pour déplacer son pion, il suffit, lorsque le curseur est sur ce pion, de taper la direction du déplacement, suivant le code :

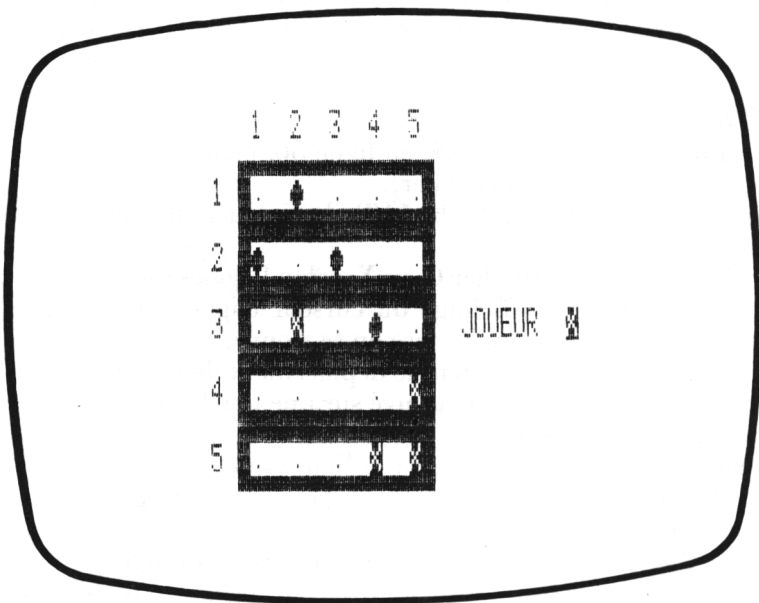


Le coup sera refusé si la case de départ n'est pas conforme ou si la case d'arrivée est occupée. Sinon, le coup est réalisé, et le trait passe à l'autre joueur, qui déplace le curseur clignotant pour l'amener sur l'un des pions, et joue de la même façon.

Le joueur gagnant est celui qui réussit à amener le premier, tous ses pions dans les cases initialement occupées par l'adversaire. On peut interrompre la partie à tout moment en tapant CTRL et C.

EXTENSIONS POSSIBLES :

On peut améliorer les contrôles, prévoir un test de fin, utiliser un terrain de jeu différent ; on pourra aussi envisager l'écriture d'un programme permettant à l'ORIC-ATMOS de participer au jeu avec de bonnes chances de l'emporter.




```

10 REM QUATRE A QUATRE
20 DATA 0,12,30,63,63,63,30,12:FOR I=46856T046863:READA:POKEI,A:NEXT I
30 PAPER 6:INK 4:CLS:PRINTCHR$(17)
40 FOR I=0 TO 24:PRINTCHR$(27);"J":NEXT I
50 H=1:A$="5665322356":PRINT@6,1;"1 2 3 4 5":PRINT@6,2;"1 2 3 4 5"
60 D$=CHR$(255):FOR I=1 TO 20:FORJ=1TO15:PRINT@4+J,3+I;D$:NEXT J:NEXT I
70 FORI=1TO5:I$=RIGHT$(STR$(I),1):PRINT@3,4*I+1;I$:PRINT@3,4*I+2;I$
80 PRINT@6,4*I+1;SPC(13):PRINT@6,4*I+2;". . . ."
90 FOR J=0 TO 1:IF<3 THEN PLOT 6+3*J,4*I+1,97:PLOT6+3*J,4*I+2,97
100 IF I>3 THEN PLOT 15+3*J,4*I+1,216:PLOT15+3*J,4*I+2,216
110 NEXT J:NEXT I
120 X=1:Y=1:B$="JOUEUR "
130 PRINT@22,13;B$:PRINT@22,14;B$:PLOT30,13,216-119*H:PLOT30,14,216-119*H
140 GOSUB 340
150 I=J
160 PRINT@3*X+3,4*Y+1;" ":PRINT@3*X+3,4*Y+2;" "
170 A=X:B=Y
180 R$=KEY$
190 Y=Y-(R$=CHR$(10) AND Y<5)+(R$=CHR$(11) AND Y>1)
200 X=X-(R$=CHR$(9) AND X<5)+(R$=CHR$(8) AND X>0)
210 IF R$<"9" AND R$>"0" THEN 240
220 IF J=97 THEN PLOT3*A+3,4*B+1,97:PLOT3*A+3,4*B+2,97:GOTO 130
230 IFJ=216THENPLOT3*A+3,4*B+1,216:PLOT3*A+3,4*B+2,216:GOTO 130
235 PLOT3*A+3,4*B+2,46:GOTO 130
240 IF (I=46) OR (I=97 AND H=0) OR (I=88 AND H=1) THEN 220
250 J=VAL(R$)
260 S=Y:Y=Y+VAL(MID$(A$,J,1))-4
270 T=X:X=X+VAL(MID$(A$,J+2,1))-4
280 IF X<1 OR X>5 OR Y<1 OR Y>5 THEN Y=S:X=T:A=T:B=S:J=216-119*H:GOTO 220
290 GOSUB 340
300 IF JK>46 THEN 330
310 PLOT3*X+3,4*Y+1,216-119*H:PLOT3*X+3,4*Y+2,216-119*H
320 H=ABS(H-1):I=1:PING
330 GOTO 220
340 J=SCRN(3*X+3,4*Y+2)
350 RETURN

```

Anagrammes

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de réflexion (2 joueurs).

Difficulté du jeu : selon le niveau.

LE JEU :

Des lettres sont choisies au hasard par l'ORIC-ATMOS, le nombre de lettres étant fonction du niveau de difficulté désiré. Puis les lettres du mot ainsi obtenu sont brouillées, et affichées au bas de l'écran ; un nouveau mot est alors confectionné, de façon aléatoire, à partir des mêmes lettres et il est affiché sous le premier, et ainsi de suite ; les deux joueurs (gauche et droit) disposent chacun d'une touche (1 ou 0) et doivent la presser dès qu'ils voient s'afficher au bas de l'écran un mot déjà présent sur l'écran ; leur score est alors évalué, et un message correspondant à leur résultat sera délivré.

LE PROGRAMME :

Le niveau du jeu est entré à la ligne 40. La boucle I (80-100) réalise le choix des lettres.

Une deuxième boucle I (120-190) effectue le brouillage, et la ligne 140 permet de détecter la pression d'une touche.

La ligne 200 affiche un nouveau mot, et fait monter d'une ligne sur l'écran les mots déjà obtenus.

La ligne 240 cherche quel est celui des deux joueurs qui a pressé une touche.

La boucle I (250-320) explore les mots présents sur l'écran ligne par ligne, alors que la boucle J (270-300) saisit un par un les caractères de ces mots. On obtient ainsi un nouveau mot qui est comparé au mot initial en ligne 310 et le résultat BRAVO apparaît si c'est exact.

Sinon, la partie s'arrête sur l'affichage PERDU (ligne 330).

POUR JOUER :

On lance le programme par RUN.

On devra ensuite entrer le niveau de difficulté (en principe de 1 à 3) et faire RETURN. Les mots s'affichent en partant du bas de l'écran, et montent au fur et à mesure ; si l'un des joueurs croit reconnaître dans le mot qui apparaît un des mots déjà affichés, il devra immédiatement presser sa touche (1 ou 0), et son résultat sera affiché après un court instant.

La partie sera alors terminée.

EXTENSIONS POSSIBLES :

On pourra utiliser d'autres affichages que des mots, modifier les temps d'affichage, faire participer plus de deux joueurs, améliorer le son et la couleur, etc.

```

10 REM ANAGRAMMES
20 PAPER 7:INK 4
30 CLS
40 INPUT"NIVEAU ( DE 1 A 3 ) :";N
50 IF N<1 OR N>3 THEN 20
60 PAPER 0:CLS
70 S$=""
80 FOR I=1 TO N+3
90 S$=S$+CHR$(INT(RND(1)*26+65))
100 NEXT I
110 A$="":B$=S$
120 FOR I=1 TO LEN(S$)
130 N=INT(RND(1)*LEN(S$)+1)
140 D$=KEY$
150 IF D$="1" OR D$="0" THEN 240
160 IF MID$(B$,N,1)="-" THEN 130
170 A$=A$+MID$(B$,N,1)
180 B$=LEFT$(B$,N-1)+"-"+RIGHT$(B$,LEN(B$)-N)
190 NEXT I
200 PLOT1,25,INT(RND(1)*6+1):PRINT@2,25;A$:PRINT
210 FOR I=1 TO 20*LEN(B$):NEXT I
220 S$=A$
230 GOTO 110
240 IF D$="1" THEN H$="JOUEUR DE GAUCHE"
    ELSE H$="JOUEUR DE DROITE"
250 FOR I=0 TO 23
260 B$=""
270 FOR J=2 TO LEN(S$)+1
280 R$=CHR$(SCRN(J,I))
290 B$=B$+R$
300 NEXT J
310 IF B$=S$ THEN PRINT@LEN(S$)+6,I;
    "BRAVO , ";H$:ZAP:ZAP:ZAP:END
320 NEXT I
330 PRINT@LEN(S$)+6,24;"PERDU ";H$:EXPLODE

```

```

UDAZC
COUAZ
AOZCU
ACZOU
ZOUAC
ACUOZ
OCZAU
ZUCAU
OUACZ
CZAUO
UAOZC
OACUZ
CAOUZ
OAUOZ
OUZAO
OUACZ
OUCZA PERDU JOUEUR DE DROITE

```

***O*œufs**

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

Un oiseau mystérieux se déplace en battant des ailes au hasard sur l'écran. Il pond des œufs (représentés par des O) que vous (un carré de couleur) devez essayer de récupérer. Votre score (nombre d'œufs récupérés) sera affiché en permanence, et la partie se déroule en un laps de temps défini.

LE PROGRAMME :

L'oiseau est dessiné en deux parties (pour le mouvement) aux lignes 70 et 110 ; son déplacement aléatoire est géré par le sous-programme 210-300 (en particulier les lignes 260 à 290).

La ligne 130 permet de reboucler le programme sur la ligne 70.

Le déplacement du joueur est régi par les lignes 150 et 160, et le test permettant de détecter la présence d'un œuf est effectué en ligne 170.

La ligne 190 permet l'affichage du score.

La fin de jeu est détectée en ligne 230.

POUR JOUER :

On lancera le programme par RUN.

L'oiseau commence alors à se déplacer au hasard sur l'écran, et se met à pondre ses œufs. Lorsque le symbole qui représente le joueur (un carré de couleur) apparaît sur l'écran, celui-ci peut se déplacer en utilisant les 4 touches de déplacement. Il vous faut dès lors essayer de récupérer un maximum d'œufs avant la fin de la partie.

On pourra arrêter de jouer à tout moment en tapant CTRL et C.

EXTENSIONS POSSIBLES :

Sur le même principe, on pourra facilement imaginer d'autres thèmes de jeu. On pourra bien sûr améliorer la couleur, les sons, le graphisme, et éventuellement, définir un jeu à plusieurs joueurs, chacun essayant de récupérer un maximum d'œufs.

Taquin de lettres

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

Qui d'entre nous n'a pas, un jour ou l'autre, eu dans les mains un de ces petits puzzles où l'on essaie de déplacer d'une case à une autre des petits carrés de plastique colorés, jusqu'à l'obtention d'un résultat bien précis ? En son temps le 14-15 de SAM LOYD, a eu autant de succès que notre célèbre RUBIK'S.

Le taquin qui vous est proposé ici utilise 35 lettres, disposées au hasard dans un carré de 6 cases sur 6 cases, et ces lettres peuvent, replacées dans leur ordre initial, vous permettre de reconstituer un proverbe très connu. Une case coloriée vous permet de déplacer les lettres, et un compteur affiche en permanence le nombre d'essais : arriverez-vous à retrouver le proverbe ?

LE PROGRAMME :

La phrase à reconstituer (qui pourra bien sûr être modifiée) est placée dans une variable R\$ en ligne 70.

Le mélange des lettres et l'impression sont faits par la boucle I (80-150).

La ligne 180 réalise l'affichage du carré coloré ; celui-ci est déplacé par les lignes 220 et 230 en utilisant les 4 flèches.

Les lignes 240 et 250 réimpriment la lettre concernée à l'emplacement du carré coloré.

La ligne 270 comptabilise le nombre d'essais, qui est imprimé grâce à la ligne 170.

POUR JOUER :

On lance le programme par RUN.

La partie sera évidemment d'autant plus difficile que l'on ne connaît pas le proverbe à reconstituer. On peut déplacer les lettres et les faire passer à l'emplacement du carré coloré, à l'aide des 4 touches de direction (→, ←, ↓, ↑).

Le nombre d'essais est affiché en permanence, à droite du puzzle.

On peut arrêter à tout moment en tapant CTRL et C.

EXTENSIONS POSSIBLES :

Le même principe peut être appliqué à toutes sortes de taquins (manipulation de nombres, de couleurs, de sons éventuellement). On peut aussi prévoir plusieurs phrases, avec un choix effectué par un générateur aléatoire, etc.

```

10 REM TAQUIN DE LETTRES
20 PAPER6:INK4:CLS:PRINT CHR$(17)
30 FOR I=1 TO 10
40 PRINT@4,I;CHR$(27);"R";"          ";CHR$(27);"V"
50 NEXT I
60 A$=""
70 R$="ONASOUVENTBESOIND'UNPLUSPETITQUESOI"
80 FOR I=1 TO 35
90 X=INT(RND(1)*35+1)
100 IF MID$(R$,X,1)="-" THEN 90
110 A$=A$+MID$(R$,X,1)
120 R$=LEFT$(R$,X-1)+"-"+RIGHT$(R$,LEN(R$)-X)
130 F=ASC(MID$(A$,I,1))
140 PLOTI-6*INT(I/6)+6,INT(I/6)+3,F
150 NEXT I
160 X=0:Y=0:S=0:C=32
170 PRINT@2,20;"NOMBRE D'ESSAIS :";S
180 PRINT@X+6,Y+3;CHR$(255)
190 T$=KEY$
200 IF T$="" THEN PRINT@X+6,Y+3;" ":GOTO 180
210 I=X:J=Y
220 X=X-(T$=CHR$(9) AND X<5)+(T$=CHR$(8) AND X>0)
230 Y=Y-(T$=CHR$(10) AND Y<5)+(T$=CHR$(11) AND Y>0)
240 C=SCRN(X+6,Y+3)
250 PLOTI+6,J+3,C
260 IF I=X AND J=Y THEN ZAP:ZAP:ZAP:GOTO 280
270 S=S+1:PING
280 GOTO 170

```

```

PALENN
TO'BOU
EUDSNU
NTIOST
EEPOUT
UOSTS

```

NOMBRE D'ESSAIS : 27

Cibles

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : hasard.

Difficulté du jeu : facile.

LE JEU :

Il consiste simplement à tirer sur des cibles, en essayant de marquer le maximum de points ; le nombre de cibles (de 1 à 6) sera choisi en début de partie, et le score réalisé sera affiché en permanence. Si l'on touche le bord d'une cible, la partie s'arrête. On marquera d'autant plus de points que le nombre sur lequel on tombe sera plus grand.

LE PROGRAMME :

Le nombre de cibles est entré en ligne 50. La boucle I (80-100) effectue l'impression des cibles.

La ligne 120 affichera le score en permanence.

La balle est affichée par la ligne 130.

La boucle I (150-200) réalise l'impression des chiffres contenus dans les cibles, et permet le déplacement de la boule (ligne 190).

La ligne 220 regarde si l'on a tiré, et la ligne 230 regarde si la boule est encore sur la ligne de départ.

La ligne 240 fait progresser la boule d'une case vers la droite, et la ligne 250 vérifie si l'on a ou non atteint les cibles ; puis les lignes 260 et 270 regardent ce que l'on a touché.

La ligne 280 incrémente le score.

Les lignes 290 et 300 s'occupent de la fin de partie.

POUR JOUER :

Lancer le programme par RUN, et taper le nombre de cibles désirées suivi de RETURN. Les cibles s'affichent, et la partie commence. Votre balle, représentée par un O est affichée en bas à gauche ; vous pouvez la déplacer avec les flèches haut et bas (touches ↑ et ↓), et vous pouvez tirer à tout moment en tapant Ø.

Le score sera modifié en fonction de votre résultat, et une nouvelle balle apparaît. Il est possible d'arrêter à tout moment en tapant CTRL et C, sinon le jeu s'arrête au bout d'un certain nombre de tirs.

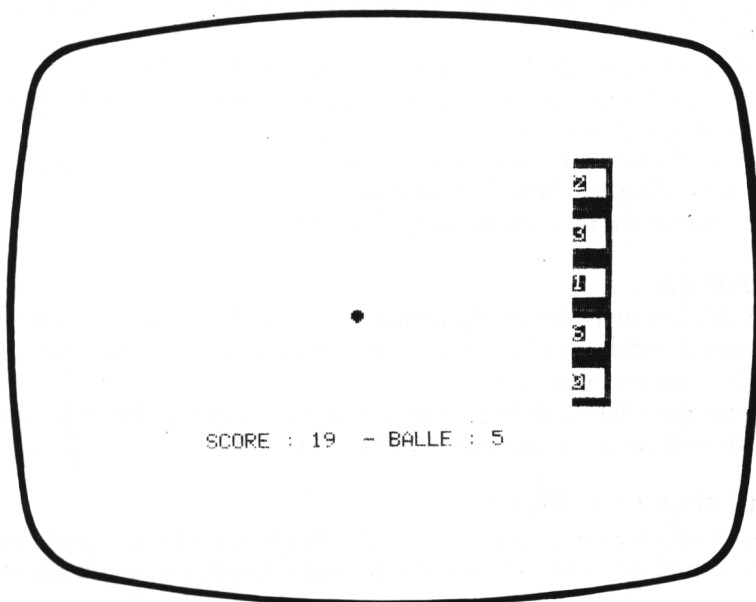
EXTENSIONS POSSIBLES :

On pourra sans difficulté augmenter le nombre de cibles, modifier la règle d'apparition des nombres dans les cibles pour la rendre moins hasardeuse, s'arranger pour que la balle elle-même soit représentée par un chiffre qui devrait coïncider à l'arrivée avec celui de la cible.


```

10 REM CIBLES
20 D$=CHR$(27):E$=D$+"I":PRINT CHR$(17)
30 DATA 0,12,30,63,63,63,30,12:FOR I=46856 TO 46863:READ A
40 POKE I,A:NEXT I
50 PRINT:INPUT"OMBRE DE CIBLES :";N
60 P=10:J=30:S=0:B=1
70 PAPER 3:INK 1:CLS
80 FOR I=1 TO N
90 PRINT @J-1,3*I-1;E$;"//0":PRINT@J+1,3*I;E$;"J":
  PRINT@J-1,3*I+1;E$;"\\^"
100 NEXT I
110 K=3:A=18
120 PRINT@2,20;D$;"BSCORE :";S;" - BALLE :";B
130 PRINT@K-1,A;D$;"E";CHR$(97)
140 D=A
150 FOR I=1 TO N
160 U=INT(RND(1)*P)
170 PLOTJ,3*I,U+176
180 A$=KEY$
190 A=A-(A$=CHR$(10))+(A$=CHR$(11))
200 NEXT I
210 PRINT@K-1,D;" "
220 IF A$="0"THEN B=B+1:SHOOT:GOTO 240
230 IF K=3 THEN 130
240 K=K+1
250 IF K<J THEN 130
260 C=SCRN(K,A):IF C>128 THEN C=C-128
270 IF C<48 OR C>57 THEN PRINT"PERDU...":EXPLODE:END
280 S=S+C-48:ZAP:ZAP:ZAP
290 P=P-1
300 IF P=0 THEN PING:END
310 GOTO 110

```



Gouttes

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : dessin.

Difficulté du jeu : aucune.

LE JEU :

C'est un véritable programme de dessin animé que l'on peut considérer, en fait, comme un interlude ; un récipient est dessiné sur l'écran, vu en coupe, et il contient déjà des obstacles. Une goutte d'eau, symbolisée par un rond gris, apparaît au-dessus du récipient et tombe en remplissant le récipient au fur et à mesure. Sa trajectoire, lorsqu'elle rencontre un obstacle, est entièrement déterminée par le hasard : elle part à droite ou à gauche, jusqu'à ce qu'il n'y ait plus aucune possibilité. Alors, une autre goutte tombe, et ainsi de suite, jusqu'au débordement final.

LE PROGRAMME :

Les deux boucles emboîtées I (50-130) et J (60-120) dessinent le récipient et les obstacles.

La ligne 170 dessine la goutte et la ligne 210 envoie dans un sous-programme (370-390) qui permet de déterminer s'il y a ou non rencontre avec un obstacle.

Si c'est non, ou va en 290, et la goutte continue de tomber.

Si c'est oui, la goutte part à droite ou à gauche (ligne 230) selon la valeur définie pour K en ligne 150.

On regarde de nouveau s'il y a ou non obstacle, ou si la goutte peut continuer à tomber (lignes 240, 250 ou 260).

Selon le résultat du test, on utilisera 270, 290, 310 ou 340.

POUR JOUER :

Le RUN déclenche le dessin du récipient et des obstacles, la goutte apparaît, se met à tomber et finit par se trouver une place. Une autre goutte apparaît et l'on recommence.

Le programme s'arrête de lui-même lorsque le récipient déborde ; on peut l'arrêter à tout moment en tapant CTRL et C.

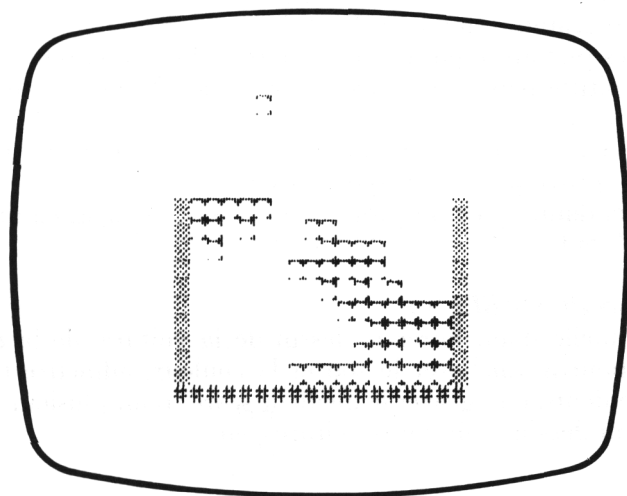
EXTENSIONS POSSIBLES :

On peut, bien entendu, ajouter la couleur et le son et aussi envisager un jeu dans lequel un petit personnage devrait échapper aux gouttes ou à la noyade.

```

10 REM GOUTTES
20 DATA 0,28,62,127,127,127,62,28:FOR I=46856 TO 46863:READ A
   :POKEI,A:NEXT I
30 LORES 0
40 INK 1:PAPER 0
50 FOR I=11 TO 20
60 FOR J=1 TO 16
80 PLOT J+2,20,18
90 IF I=11 THEN NEXT I
100 IF I<>20 THEN PLOT J+2,I,19
110 IF I<>20 AND RND(1)<0.7 THEN PLOT INT(RND(1)*15+3),I,21
120 NEXT J
130 NEXT I
140 Y=INT(RND(1)*7+6):X=6
150 K=-2*(RND(1)<0.5)-1
160 I=0
170 PLOTY,X,225
180 WAIT 30
190 PLOTY,X,16
200 U=X+1:V=Y
210 GOSUB 370
220 IF J=0 THEN 290
230 U=X:V=Y+K
240 GOSUB 370
250 IF J=0 THEN 310
260 IF I=0 THEN 340
270 PLOTY,X,225
280 GOTO 140
290 X=X+1
300 GOTO 150
310 Y=Y+K
320 I=1
330 GOTO 170
340 K=-K
350 I=1
360 GOTO 230
370 C=SCRN(V,U):J=-((C<>16AND C<>19)
380 IF U=24 THEN PRINT@18,15;"C'EST FINI...":EXPLODE:END
390 IF J=0 THEN PING
400 RETURN

```



Obstacles

REFERENCES :

Difficulté de programmation : assez difficile.

Catégorie de jeu : jeu de réflexe (pour un joueur).

Difficulté du jeu : moyenne.

LE JEU :

Des piétons se déplacent sur l'écran pour venir à la rencontre d'une petite voiture qui dispose de la possibilité de se déplacer vers la gauche ou vers la droite pour les éviter et qui devra tenir le plus longtemps possible.

La distance parcourue sera affichée en permanence, et le jeu se termine lorsque la voiture rencontre, soit le bord du terrain, soit un piéton.

LE PROGRAMME :

La ligne 80 fait avancer les piétons vers la voiture du joueur qui est imprimée grâce à la ligne 70.

Les lignes 80, 90, 100 fabriquent la route (les lignes 90 et 100 bien qu'identiques sont toutes deux indispensables, car la valeur de RND ne sera, en principe, pas la même sur ces deux lignes).

La ligne 120 affiche le score.

La ligne 140 permettra de déplacer le véhicule.

Les lignes 150 et 160 permettent de contrôler s'il y a ou non un obstacle et arrêtent le programme s'il y a lieu.

Le score est incrémenté en ligne 170.

POUR JOUER :

On lance le programme en tapant RUN.

La route et les piétons apparaissent en bas de l'écran et montent vers la voiture. La voiture peut se déplacer à l'aide des flèches de déplacement horizontal (touches → et ←).

La rencontre avec un piéton ou la bordure interrompt le déroulement du programme et provoque l'affichage du score.

On peut, cependant, sortir du programme à tout moment en utilisant les touches CTRL et C.

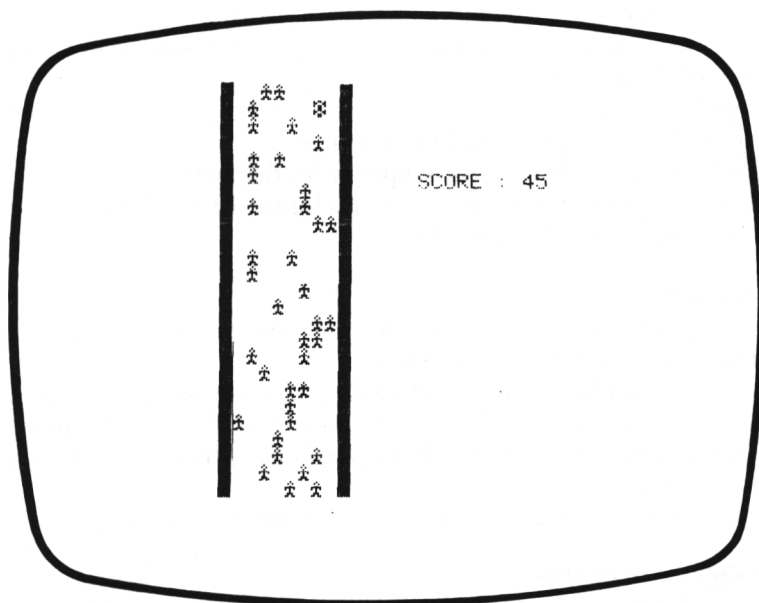
EXTENSIONS POSSIBLES :

Il sera assez facile d'améliorer le dessin de la voiture, de la route, des piétons. On pourra améliorer le son et la couleur, admettre un certain nombre d'accidents, enregistrer le meilleur score (pour plusieurs joueurs), éventuellement dessiner plusieurs voitures, etc.

```

10 REM OBSTACLES
20 DATA 45,51,45,12,45,51,45,12,8,20,8,62,42,8,28,54
30 FOR I=46856 TO 46871:READ A:POKE I,A:NEXT I
40 PAPER 0:INK 2:PRINT CHR$(17)
50 LORES 0
60 S=0:Y=5
70 PRINT@Y-1,2;CHR$(27);"A";CHR$(97);CHR$(27);"B":
  PLOT Y,1,32
80 PRINT@2,25;CHR$(255)+"          "+CHR$(255);
90 IF RND(1)<0.7 THEN PLOT INT(RND(1)*8+3),25,98
100 IF RND(1)<0.8 THEN PLOT INT(RND(1)*8+3),25,98
110 PRINT:PRINT:PRINT@15,5;"          "
120 PRINT@17,6;"SCORE :";S
130 A$=KEY$
140 Y=Y-(A$=CHR$(9) AND Y<11)+(A$=CHR$(8) AND Y>2)
150 C=SCRN(Y,2)
160 IF C=98 OR C=127 THEN EXPLODE:END
170 S=S+1
180 GOTO 70

```



Captures dans l'espace

REFERENCES :

Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (1 seul joueur).
Difficulté du jeu : moyenne.

LE JEU :

Votre vaisseau spatial navigue dans un dangereux champ de météorites et votre mission est de vous emparer de la plus grande quantité possible de minéral d'urane, un métal indispensable que l'on ne peut trouver que dans ces lieux particulièrement dangereux.

Vous devez arriver à manœuvrer pour recueillir l'urane, sans vous faire heurter par une météorite.

LE PROGRAMME :

La ligne 70 permet de faire apparaître les météorites.

La ligne 90 fera apparaître le minerai d'urane et la ligne 100 assure leur déplacement sur l'écran.

Les lignes 110 à 130 contrôlent le déroulement du jeu (a-t-on heurté une météorite ou a-t-on capturé un bloc d'urane).

La rencontre d'une météorite provoque la fin de partie.

Votre vaisseau spatial est imprimé par les lignes 140 et 150 et son déplacement est contrôlé par la ligne 170.

POUR JOUER :

On lance le programme par RUN et les météorites commencent à monter sur l'écran, ainsi que les blocs d'urane.

Les touches de déplacement horizontal (→ et ←) seront utilisées pour déplacer votre vaisseau et récupérer l'urane, mais le moindre heurt avec une météorite provoquera la fin de la partie et l'affichage de la quantité d'urane capturée.

On pourra aussi arrêter à tout moment en tapant CTRL et C.

EXTENSIONS POSSIBLES :

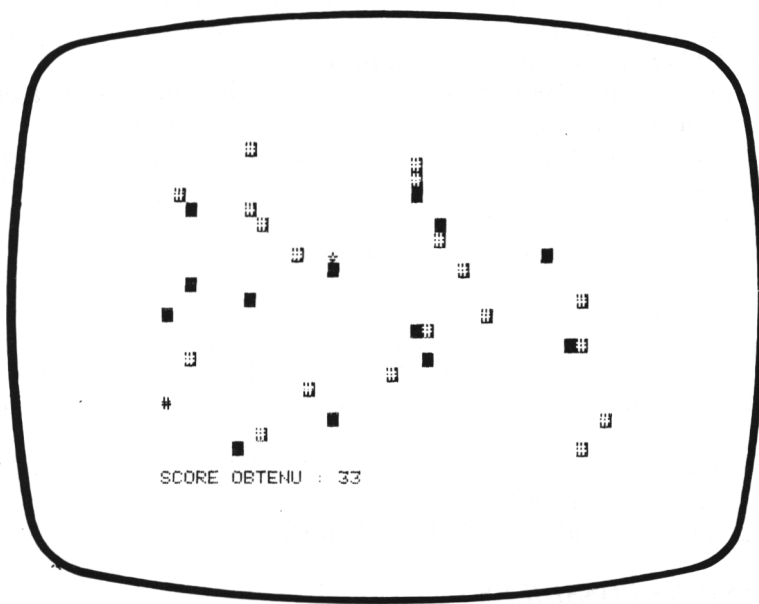
On peut rendre le jeu de plus en plus difficile en augmentant progressivement le nombre de météorites, ou encore, en minutant la durée de la partie.

On peut aussi prévoir un affichage permanent du score, introduire un deuxième vaisseau spatial manœuvré par une puissance étrangère et qui cherche aussi à s'emparer de l'urane...

```

10 REM CAPTURES DANS L'ESPACE
20 DATA 0,0,8,8,54,64,28,8:FOR I=46856 TO 46863:
  READ A:POKE I,A:NEXT I
30 PAPER 0:INK 2:PRINT CHR$(17)
40 LORES 0
50 X=15:Z=0
60 S=INT(RND(1)*38+2)
70 PLOTS,25,163
80 S=INT(RND(1)*38+2)
90 IF RND(1)<0.7 THEN PLOTS,25,127
100 WAIT 3:PRINT@2,25;"":PRINT
110 C=SCRN(X,12)
120 IF C=127 THEN PRINT"SCORE OBTENU :";Z:
  EXPLODE:END
130 IF C=163 THEN Z=Z+1:ZAP
140 PRINT@X-1,11;"  "
150 PRINT@X-1,12;CHR$(27);"A";CHR$(97);CHR$(27);"B"
160 A$=KEY$
170 X=X-(A$=CHR$(9) AND X<39)+(A$=CHR$(8) AND X>3)
180 GOTO 60

```



Anomalie

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

Il repose essentiellement sur une bonne capacité d'observation. Deux grilles de 25 cases chacune sont dessinées sur l'écran. Des symboles identiques sont affichés en des endroits différents des deux grilles, mais l'un des symboles a été modifié sur la grille de droite. Le jeu consiste à trouver lequel, en entrant ses coordonnées dans l'ordre ligne-colonne.

LE PROGRAMME :

La boucle I (50-80) imprime les chiffres de repérage des grilles.

La boucle principale I (90-150) imprime les symboles des deux grilles, dans un ordre différent.

La ligne 100 choisit le caractère à imprimer.

Le sous-programme (1000-1040) s'occupe de faire l'impression à un emplacement libre (on l'appelle deux fois pour les deux grilles).

Les lignes 160 et 170 changent, de façon aléatoire, l'un des caractères.

Les coordonnées du point essayé sont entrées aux lignes 180 et 190, et le contrôle est effectué par les lignes 200 et 210.

Les lignes 220 et 230 s'occupent de l'affichage du résultat.

POUR JOUER :

Après avoir lancé le programme par l'instruction RUN, on devra attendre un certain temps avant d'obtenir l'affichage des deux grilles. On devra alors chercher, par comparaison, quel est le caractère qui a été modifié (sur la grille de droite).

On entre alors son numéro de ligne, suivi de RETURN, et son numéro de colonne, suivi de RETURN.

Le programme répondra alors BRAVO ou PERDU.

EXTENSIONS POSSIBLES :

Il sera possible d'utiliser des symboles graphiques de diverses couleurs, ce qui pourra simplifier la recherche.

On peut aussi envisager d'utiliser des dessins plus sophistiqués, améliorer le contrôle du résultat, admettre la bonne réponse sur la grille de droite ou celle de gauche, introduire un minutage, etc.


```

10 REM ANOMALIE
20 PAPER 3
30 INK 3
40 CLS:PRINT CHR$(17):T=5
50 FOR I=1 TO 5
60 PRINT@2*I+3,0;I:PRINT@2,2*I;I
70 PRINT@2*I+21,0;I:PRINT@20,2*I;I
80 NEXT I
90 FOR I=1 TO 22
100 D=INT(26*RND(1)+1)
110 L=2:S=T
120 GOSUB 1000
130 L=11
140 GOSUB 1000
150 NEXT I
160 S=0:INK 1
170 GOSUB 1000
180 PING:PRINT@2,16;"LIGNE.....":INPUT I
190 PRINT:INPUT"COLONNE...":T
200 U=2*I:V=2*T+20
210 C=SCRN(V+2,U)
220 IF D+64=C THEN PRINT"BRAVO..." ELSE PRINT"PERDU"
230 PRINT:PRINT"LA LETTRE A TROUVER ETAIT ";
    CHR$(D+64):ZAP:ZAP:ZAP:END
1000 U=2*(INT(RND(1)*T+1)):V=2*INT(RND(1)*S+L)
1010 C=SCRN(V+2,U)
1020 IF C<>32 AND S<>0 THEN 1000
1030 PRINT@V+2,U:CHR$(64+D)
1040 RETURN

```

	1 2 3 4 5		1 2 3 4 5
1	X R K W P	1	F K D A F
2	P T F F	2	P X T P L
3	I U X L G	3	L P A N
4	A R D	4	G W I
5	L N D A A	5	D R X A R

```

LIGNE.....? 3
COLONNE.....? 2
BRAVO...
LA LETTRE A TROUVER ETAIT P

```

Plusieurs en un

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : deux joueurs.

Difficulté du jeu : variable, selon l'utilisation.

LE JEU :

Plusieurs jeux sont en réalité possibles avec ce programme qui va se comporter comme un crayon électronique vous permettant d'inscrire des symboles O ou X, en n'importe quel emplacement de votre feuille-écran de télévision. On pourra, par exemple, l'utiliser pour un jeu de morpion (le premier à réaliser un alignement de 5 pions identiques gagne) ou pour un GO (on réalise des encadrements des pions adverses) ou tout autre jeu, selon son imagination.

LE PROGRAMME :

Le changement de joueur est effectué par la ligne 50.

Le joueur qui a le trait voit le symbole qu'il a choisi (O ou X) se placer en limite du cadre (ligne 70).

Les lignes 100 et 110 permettent aux deux joueurs de déplacer le symbole sur le terrain.

Les lignes 130 et 140 contrôlent si le déplacement est possible (case libre et non dépassement des limites).

La ligne 190 permet d'écrire sur une case libre.

POUR JOUER :

RUN provoque l'affichage des limites du terrain et du numéro du joueur, ainsi que des symboles correspondants (O ou X) que l'on devra déplacer sur l'écran à l'aide des 4 touches de direction (→, ←, ↓, ↑).

Il n'est pas possible de venir sur une case déjà occupée et il faudra appuyer sur la touche Ø pour inscrire son propre symbole dans une case disponible. Dès que cela a été fait, le symbole du second joueur apparaît dans la colonne de gauche, à peu près au milieu, et le joueur procédera de la même manière que son adversaire.

Pour sortir du programme, il suffira de taper CTRL et C.

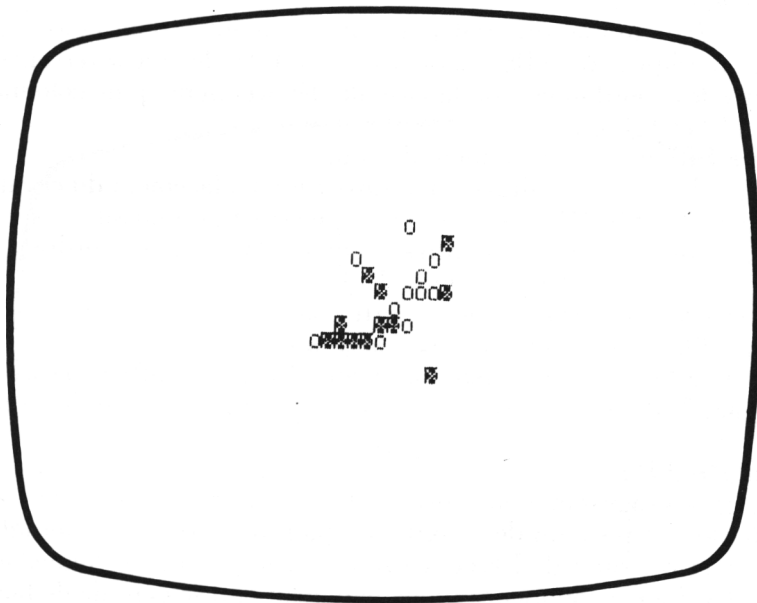
EXTENSIONS POSSIBLES :

On pourra utiliser l'écran tout entier comme terrain de jeu et réaliser des contrôles en fonction des différents jeux que l'on désire pratiquer : contrôle d'alignements de 5 pièces pour le morpion, contrôle du nombre de pièces capturées pour le GO.

```

10 REM PLUSIEURS EN UN
20 PAPER 6:INK 4
30 CLS
40 J=0
50 J=ABS(J-1)
60 X=12:Y=0
70 PLOT Y+2,X,79-137*(J=1)
80 A$=KEY$:IF A$="" THEN 80
90 A=X:B=Y
100 X=X-(A$=CHR$(10) AND X<24)+(A$=CHR$(11) AND X>0)
110 Y=Y-(A$=CHR$(9) AND Y<37)+(A$=CHR$(8) AND Y>1)
120 PLOTB+2,A,32
130 C=SCRN(Y+2,X)
140 IF C<>79 AND C<>216 THEN 160
150 X=A:Y=B
160 FOR I=1 TO 10:A$=KEY$
170 IF A$="" THEN NEXT I
180 IF A$<>"0" OR Y=0 THEN 70
190 PLOTY+2,X,79-137*(J=1)
200 IF J=0 THEN PING ELSE ZAP
210 GOTO 50

```



Parcours du cavalier

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

Il se présente, ici, comme un jeu de hasard total, mais provient, en réalité, d'un problème assez difficile à résoudre : comment déplacer un cavalier sur un échiquier de façon qu'il passe une fois, et une seule, sur chacune des 64 cases du damier.

Le cavalier se déplace suivant les règles communément admises aux échecs, et sa position de départ est définie par le joueur ; par contre, ses déplacements sont effectués au hasard. Arrivera-t-il à parcourir les 64 cases ?

LE PROGRAMME :

La ligne 60, définit les divers déplacements possibles, dans deux variables alphanumériques A\$ et B\$; la boucle I (70-100) dessine le terrain de jeu. On entre les coordonnées de la case de départ (ligne, puis colonne) aux lignes 110 et 120.

La ligne 140 imprime le numéro du coup.

La ligne 150 permet le tirage au hasard d'un déplacement du cavalier.

La boucle Z (160-210) cherche si le déplacement est possible ; si ce n'est pas le cas, on essaie tous les autres (toujours dans le même ordre).

Les lignes 170 et 180 effectuent le déplacement.

La ligne 190 contrôle que l'on ne sorte pas du terrain de jeu.

La ligne 200 envoie en ligne 230 si le coup est possible.

La ligne 220 gère la fin de partie (cavalier bloqué) et affiche le score.

Les lignes 230, 240 et 250 réalisent le coup et incrémentent le score.

POUR JOUER :

On lance le programme par l'ordre RUN. Le terrain de jeu s'affiche ; on choisit alors la position de départ en tapant d'abord un numéro de ligne (entre 1 et 8), suivi de RETURN puis son numéro de colonne, suivi de RETURN. Le jeu commence immédiatement et il se déroule de façon entièrement automatique jusqu'à ce que le cavalier soit bloqué. On peut, bien sûr, arrêter à tout moment par CTRL et C.

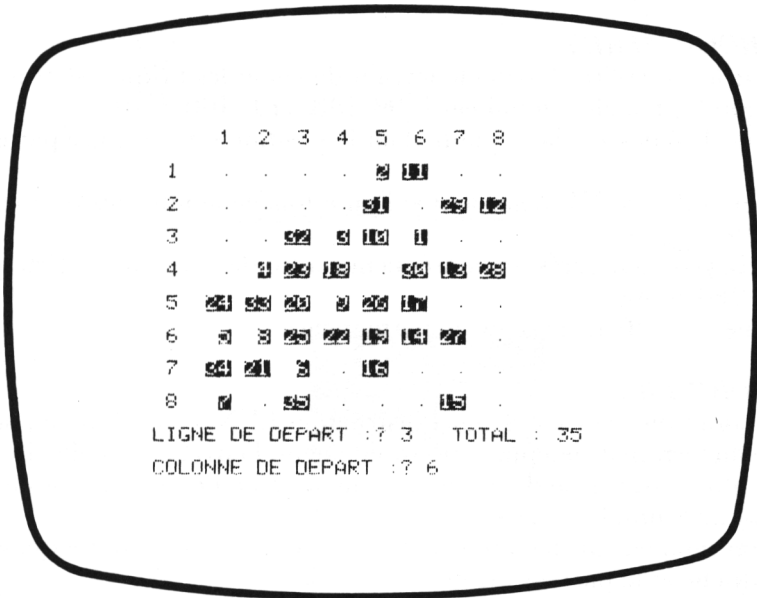
EXTENSIONS POSSIBLES :

On peut imaginer de nombreuses variantes sur ce thème, ne serait-ce qu'en commençant par faire varier la taille des échiquiers, on peut aussi améliorer considérablement le graphisme. Mais, le plus intéressant est d'arriver à définir un algorithme donnant la ou les solutions du problème initial...

```

10 REM PARCOURS DU CAVALIER
20 CLS
30 PAPER 6
40 INK 4
50 S=1:T=0
60 A$="56653223":B$="653223"+A$+"65":A$=A$+A$
70 FOR I=1 TO 8
80 PRINT@3*I+3,0:I
90 PRINT@2,2*I:I;" . . . . ."
100 NEXT I
110 PRINT@2,18:"LIGNE DE DEPART :";:INPUT I
120 PRINT:INPUT"COLONNE DE DEPART :";J
130 IF S>9 THEN S=0:T=T+1
140 PLOT3*J+4,2*I,S+176:IF T<>0 THEN PLOT3*J+3,2*I,T+176
150 Z=INT(RND(1)*8+1)
160 FOR Z=Z TO Z+8
170 A=I+VAL(MID$(A$,Z,1))-4
180 B=J+VAL(MID$(B$,Z,1))-4
190 IF A<1 OR B<1 OR A>8 OR B>8 THEN 210
200 IF SCRN(3*B+4,2*A)=46 THEN 230
210 NEXT Z
220 PRINT@25,18:"TOTAL :":T*10+S:EXPLODE:END
230 I=A:J=B
240 S=S+1:ZAP
250 GOTO 130

```



Mur de briques

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexe (pour 1 joueur).

Difficulté du jeu : difficile.

LE JEU :

S'il est un jeu que l'on peut présenter comme un grand classique des jeux d'arcades pour micro-ordinateur, c'est bien le mur de briques (en anglais BREAKOUT).

Il est offert, ici, dans une version simplifiée, mais offre néanmoins de bonnes performances : affichage du score en fin de partie, rebonds aléatoires, etc.

La règle en est très élémentaire : vous disposez d'une raquette qui peut être déplacée horizontalement pour renvoyer une balle (matérialisée par un point) qui va démolir les briques dessinées en haut de l'écran.

Votre problème est d'en démolir le plus grand nombre possible, sans rater la balle, ce qui provoquerait la fin de partie et l'affichage du score.

LE PROGRAMME :

La boucle I (50-80) dessine le terrain de jeu et les briques sont mises en place par les boucles emboîtées I (90-130) et J (100-120).

La ligne 150 dessine la raquette dont la position est contrôlée par la ligne 170.

Les lignes 220 à 250 regardent si l'on a touché le mur, les briques ou la raquette.

Les rebonds sont gérés en conséquence avec des éventuels aléas, par les lignes 260 et 300.

La ligne 190 affiche le score en permanence.

POUR JOUER :

Le programme sera lancé par l'instruction RUN, ce qui provoque l'affichage du terrain de jeu puis, très rapidement, l'envoi de la balle. Le joueur devra alors essayer de déplacer sa raquette à l'aide des touches de déplacement horizontal (→ et ←).

Si, à un moment quelconque, sa balle ne touche pas la raquette, le score sera affiché et la partie s'arrête.

On peut aussi terminer à tout moment par CTRL et C.

EXTENSIONS POSSIBLES :

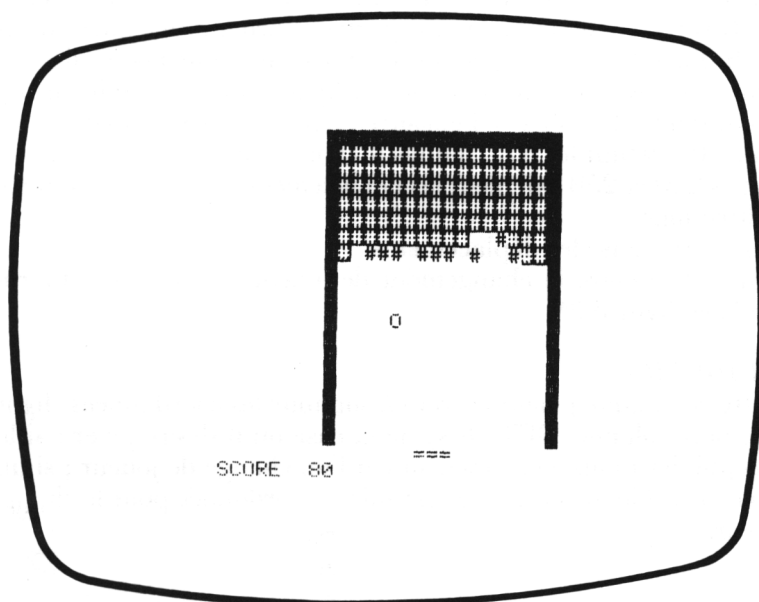
On peut envisager d'agrandir le terrain de jeu, d'augmenter le nombre de briques, de prévoir plusieurs balles, d'introduire des effets sonores.

On peut aussi, en s'inspirant des jeux d'arcades, introduire de petits ouvriers qui viendront reconstruire votre mur de briques, au fur et à mesure que vous le démolissez.

```

10 REM MUR DE BRIQUES
20 D$=CHR$(255):PRINT CHR$(17)
30 PAPER 6:INK 4
40 CLS
50 FOR I=1 TO 18
60 PRINT@9+I,0:D$
70 PRINT@10,I:D$:PRINT@27,I:D$
80 NEXT I
90 FOR I=1 TO 7
100 FOR J=11 TO 26
110 PLOTJ,I,163
120 NEXT J
130 NEXT I
140 H=-1:K=-0.5:J=18:I=17:Y=J
150 PRINT@Y-2,19;" === "
160 A$=KEY$
170 Y=Y-(A$=CHR$(9) AND Y<28)+(A$=CHR$(8) AND Y>8)
180 IF T=1 THEN PLOTJ,I,32
190 PRINT@2,20;"SCORE ";S
200 IF I>22 THEN EXPLODE:GET A$:GOTO 30
210 I=I+H:J=J+K
220 C=SCRN(J,I)
230 IF C=32 THEN PLOTJ,I,79:T=1:GOTO 150
240 IF C=127 THEN 280
250 IF C=163 THEN PLOTJ,I,79:T=1:GOTO 300
260 H=-H
270 K=RND(1)-RND(1)
280 K=-K:T=0
290 GOTO 150
300 S=S+10:ZAP
310 K=-K-(2*(RND(1)<0.5)+1)*RND(1)/3)
320 H=-H
330 GOTO 150

```



Othello

REFERENCES :

Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : difficile.

LE JEU :

Devenu un grand classique des jeux sur micro-ordinateur, le jeu d'Othello ou Reversi gagne, même si, comme ici, il est pratiqué à deux joueurs, à être géré sur ordinateur, avec visualisation : en effet, le jeu consistant essentiellement en des retournements des pions de l'adversaire, ceci peut se révéler, à la longue, très fastidieux, surtout si vous êtes un joueur assidu. Le principe est que le joueur possédant les X doit pour pouvoir jouer, réussir à encadrer une ou plusieurs des pièces de son adversaire, dans une ou plusieurs directions ; s'il y réussit, les O seront alors retournés pour devenir des X (le retournement pourra être fait horizontalement, verticalement ou suivant les diagonales).

Le joueur ayant les O procédera de la même façon, et ceci jusqu'à ce que le damier soit rempli ou que la personne ne puisse plus jouer (si un joueur ne peut retourner des pions, il doit passer son tour).

LE PROGRAMME :

Les boucles emboîtées P (50-120) et Q (70-110) dessinent le terrain de jeu et la ligne 140 affiche un X ou un O, selon le joueur qui commence.

La boucle P (200-320) contrôle la validité du coup (la case est-elle libre : lignes 300 et 310), en utilisant une partie de programme qui se trouve aux lignes 360 à 420 (a-t-on pris un ou plusieurs ennemis en ligne 360 et retournement éventuel autorisé par la ligne 380 : on refait un balayage dans toutes les directions, si l'encadrement s'est révélé possible).

La ligne 200 définit les 8 directions d'exploration.

Les lignes 240 et 250 calculent les déplacements à effectuer pour explorer ces 8 directions.

La ligne 230 réalise les déplacements.

La ligne 340 assure le changement de joueur, s'il y a eu retournement (contrôlé en ligne 330).

POUR JOUER :

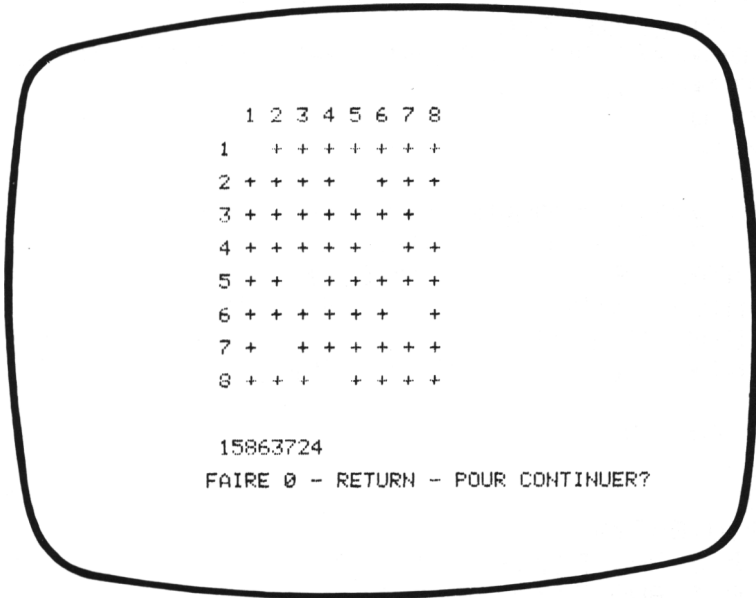
Après RUN, chaque joueur entrera à son tour les coordonnées (ligne, RETURN, puis colonne, RETURN) de la case où il désire jouer ; si le coup est accepté, les pions sont retournés et l'on change de joueur ; si un coup n'est pas possible, on entre 0 en première coordonnée pour le changement de joueur.

EXTENSIONS POSSIBLES :

Avant tout, bien sûr, améliorer l'affichage avec des couleurs, une grille plus aérée.

On pourra aussi imaginer un jeu machine pour l'ORIC-ATMOS.

Après avoir lancé le programme par RUN, il convient d'entrer la dimension de la grille (4, RETURN ou 6, 8 ...).



```

10 REM OTHELLO
20 INPUT "DIMENSION DE LA GRILLE ":A
30 IF A/2<>INT(A/2) THEN RUN
40 PAPER 6:INK 4:CLS:PRINT CHR$(17)
50 FOR P=1 TO A
60 PRINT@2*P+2,0;RIGHT$(STR$(P),1)
70 FOR Q=1 TO A
80 PRINT@2,2*P;RIGHT$(STR$(P),1):PRINT@2*Q+2,2*P;". "
90 IF Q=A/2+1 AND P=A/2 THEN PLOT2*Q,2*P,79:
    PLOT2*Q+2,2*P,216
100 IF Q=A/2+1 AND P=A/2+1 THEN PLOT2*Q,2*P,216:
    PLOT2*Q+2,2*P,79
110 NEXT Q
120 NEXT P
130 K=1
140 PRINT@23,6;"JOUEUR ":IF K=1 THEN PLOT 31,6,216
    ELSE PLOT 31,6,79:PING
150 PRINT@31,8;"      ":PRINT@31,10;"      "
160 PRINT@23,8;"LIGNE  ":INPUT U
170 IF U=0 THEN 340
180 PRINT@23,10;"COLONNE:":INPUT V
190 H=1
200 FOR P=1 TO 8
210 S=0
220 W=0
230 T=2*U:R=2*V
240 F=(P>3 AND P<7)-(P<3 OR P>7)
250 G=(P>1 AND P<5)-(P>=6)
260 Q=46-170*(S=1)-33*(S=2)
270 IF S=0 THEN 290
280 PLOT R+2,T,0
290 T=T+2*F:R=R+2*G
300 C=SCRN(R+2,T)
310 IF C=79 OR C=216 THEN 360
320 NEXT P
330 IF H<>0 THEN 140
340 K=ABS(K-1)
350 GOTO 140
360 IF (C=79 AND K=1)OR(C=216 AND K=0) THEN W=1
370 IF S<>0 THEN PLOT R+2,T,0
380 IF W=1 AND((C=79 AND K=1)OR(C=216 AND K=0))
    THEN 290
390 IF W=0 OR S<>0 THEN 320
400 S=1-(K=0)
410 H=-H*(S=0)
420 GOTO 220

```




Enseigner et distraire est le double objectif de cet ouvrage.
Ces 102 programmes de jeux vous entraînent à la
découverte de l'ORIC-ATMOS.

La classification par niveau de ces programmes
vous permet d'acquérir progressivement des connaissances
nouvelles et une plus grande maîtrise du Basic.

Pour chaque niveau, la description des jeux proposés
est suivie d'un programme abondamment commenté
et d'un exemple d'utilisation.

Le dernier niveau constitue une illustration claire
de l'utilisation du Basic étendu.

**DES LIVRES
POUR VOTRE
ORDINATEUR**



ISBN 2.86595.179.0
110 FF

Couverture GIBRAT



102 PROJES PROGRAMMES

PAR JACQUES DECONCHAT

