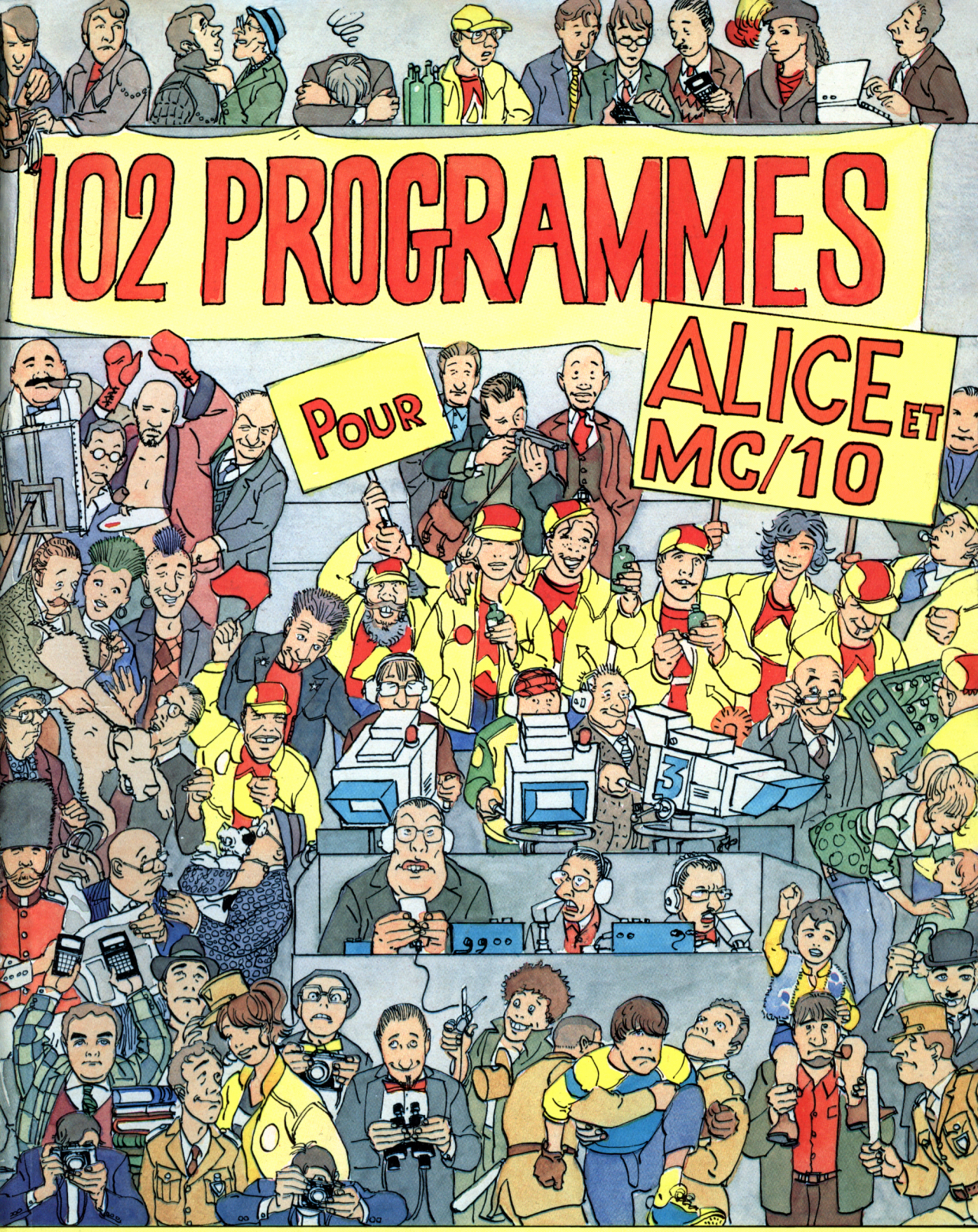
# 102 PROGRAMMES

POUR ALICE ET MC/10



PAR
JACQUES
DECONCHAT

P.S.I.

P.S.I.

PROGRAMMES

# 102
## PROGRAMMES POUR
# ALICE
## ET
# TRS-MC/10

La collection « **PROGRAMMES** » est constituée actuellement de vingt-deux titres. Certains disposent d'une disquette pour Apple II (*).

(*) Bibliothèque scientifique en Pascal — Hervé Haut
(*) Mathématiques et statistiques — Hervé Haut
(*) Modèles pratiques de décision — tomes 1 et 2 — Jean-Pierre Blanger
— Jeux, trucs et comptes pour PET/CBM — Michel Benelfoul
— Jeux, trucs et comptes pour TRS-80 — Michel Benelfoul et Denis Chauvin
— Jeux, trucs et comptes pour T07 — Michel Benelfoul
— Jeux, trucs et comptes pour TI-99/4A — Michel Benelfoul
— Jeux, trucs et comptes pour Commodore 64 — Michel Benelfoul et Cyril Cambien
— Etudes pour ZX-81 — tomes 1 et 2 — Jean-François Séhan
— Variations pour PC-1211 — Jean-François Séhan
— Récréations pour TI-57 — tomes 1 et 2 — Jacques Deconchat
— Collège, poquettes et maths — Jacques Deconchat
— 102 Programmes pour Sinclair et Timex — Jacques Deconchat
— 102 Programmes pour Commodore 64 — Jacques Deconchat
— 102 Programmes pour TI-99/4A — Jacques Deconchat
— Le TI-99/4A à l'affiche — Jean-François Séhan
— Le VIC à l'affiche — Jean-François Séhan
— Suites pour PC-1500 — Jean-François Séhan
— L'Oric à l'affiche — Jean-François Séhan
— Le PC-1251 à l'écran — Pierrick Moigneau
— 102 Programmes pour Alice et MC/10 — Jacques Deconchat
— Le Spectrum à l'affiche — Jean-François Séhan

Autre ouvrage relatif à l'Alice :

— La découverte de l'Alice et MC/10 — Maurice Charbit
Collection « MATERIELS »

RAPPELS

**Les séries :**

En fait, il faudrait parler de niveaux, puisque la couleur attachée à chaque ouvrage permet de situer la « force » de celui-ci selon le code suivant :

*Série VERTE* : ouvrage d'initiation ne nécessitant que des connaissances de base.
*Série BLEUE* : suppose une connaissance élémentaire du sujet traité.
*Série ROUGE* : ouvrage d'approfondissement, niveau de complexité moyen.
*Série NOIRE* : ouvrage d'approfondissement, niveau de complexité élevé.

**Les collections :**

Les ouvrages d'Edition du PSI, actuellement au nombre de 100, sont répartis en collections :

**« LANGAGES », « MATERIELS », « PROGRAMMES », « GUIDES PRATIQUES », « MEMENTOS », « UTILISATIONS DE L'ORDINATEUR », « LOGIGUIDES »** et pour l'initiation, outre quelques livres hors collection, **« ... POUR TOUS »**.

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 1er 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.

*© Editions du P.S.I. B.P. 86 - 77402 Lagny Cedex 1984*
ISBN France : 2-86595-131-6
ISBN étranger : 2-86595-145-6

JACQUES
DECONCHAT

# 102
## PROGRAMMES POUR
# ALICE
## ET
# TRS-MC/10

EDITIONS DU P.S.I.
1984

# SOMMAIRE


| | pages |
|---|---|
| Avant-propos | 8 |
| Organisation de l'ouvrage | 9 |
| **NIVEAU 1** | 11 |
| Instructions élémentaires | 13 |
| Le nombre mystérieux | 18 |
| Cube | 20 |
| Chaud, froid | 22 |
| Les allumettes | 24 |
| Allumettes à plusieurs | 26 |
| Calendrier perpétuel | 28 |
| Bataille navale | 30 |
| Golf | 32 |
| Le jeu du 21 | 34 |
| Pair - Impair | 36 |
| Devine un chiffre | 38 |
| Le jeu de Wytthof | 40 |
| Rebonds | 42 |
| Jackpot | 44 |
| Roulette | 46 |
| Couloirs | 48 |
| Echiquier | 50 |
| Le jeu de la fève | 52 |
| Jeu de l'oie aléatoire | 54 |
| **NIVEAU 2** | 57 |
| Dimensionnement d'un tableau - Recherche d'un élément dans un tableau et dans une chaîne | 58 |
| Dé truqué | 60 |
| Visions | 62 |
| 4 2 1 | 64 |
| L'électricien fou | 66 |
| La course au 20 | 68 |
| Dessin | 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 |
| Le jeu de Marienbad | 88 |
| Compose | 90 |
| Jour de la semaine | 92 |
| Tour de cartes | 94 |
| Le jeu de la vie | 97 |
| **NIVEAU 3** | 101 |
| Manipulation des chaînes et graphiques haute résolution | 102 |
| Un dé | 104 |
| Losanges | 106 |
| Plus grand, plus petit | 108 |
| Ordre | 110 |
| Le pendu | 112 |
| Belote | 114 |
| Un taquin de calcul | 116 |
| Pendu inversé | 118 |

Nicomaque 120
Course de chevaux 122
A travers 124
Mastermind 126
Idem 128
Horloge 130
Reines sur l'échiquier 132
Répétitions 134
Mots 136
Permutations 138
Bridge 140
Hanoï 142
Solfège 144
Le loup et les agneaux 146

**NIVEAU 4** 149

Animation 150
Tirage des 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
Adresse 174
Sous-marin 176
Zombies 178
Mathieu 180
Diago 182
Décomaque 184
Chasse au sous-marin 186
Le lézard d'Alice 188
Clavier 190
Le jeu du 15 192
Calculatrice 194

**NIVEAU 5** 197

Les instructions d'exporation des cases mémoires : **PEEK** et **POKE** 198
Othello 199
Smokey 202
Course de voitures 204
Echecs 206
Danger spatial 208
Tic Tac Toe 210
Anagrammes 212
Oeufs 214
Quatre à quatre 216
Dames 219
Taquin de lettres 222
Cibles 224
Gouttes 226
Obstacles 228
Captures dans l'espace 230
Anomalie 232
Plusieurs en un 234
Parcours du cavalier 236
Mur de briques 238

# AVANT-PROPOS

Vous trouverez dans ce livre plus de 100 programmes, rédigés en Basic Microsoft (Microcolor 1.0) pour l'ordinateur individuel Alice et le Tandy MC/10.

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 assez élémentaires, 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 programmes plus performants ; les programmes sont abondamment commentés pour vous aider à comprendre leur fonctionnement, et les pages de présentation vous permettront de retrouver rapidement les caractéristiques propres des instructions Basic qu'ils utilisent.

Vous trouverez, à la fin de chaque programme, 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 d'étudier les 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 : **MID$, RIGHT$, LEFT$.**

- **Au niveau 4**, on utilise une instruction très particulière, qui facilite la réalisation de jeux interactifs : **INKEY$.**

- Enfin, **le niveau 5** nous fait découvrir les instructions frontières du Basic, qui permettent la manipulation directe dans les mémoires de l'appareil : **PEEK** et **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 pour des extensions ou des améliorations possibles du jeu proposé.

Le programme, fonctionnant sur Alice, est ensuite proposé, suivi d'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 disfonctionnement.

**LES CONVENTIONS FAITES POUR L'IMPRESSION DES PROGRAMMES**

Le système particulier de codage retenu pour Alice est tel que les caractères graphiques et les couleurs sont repérés par des numéros compris entre 128 et 255, ce qui permet de les introduire sans perturber le codage des caractères et symboles usuels qui, en ASCII, reçoivent des numéros compris entre 0 et 128.

Mais l'imprimante associée à Alice ne peut interpréter (et imprimer) de façon simple que les 128 premiers codes : les codes des couleurs (ou les symboles les remplaçant) n'apparaissent donc pas à l'impression.

Comme la couleur, une des particularités les plus intéressantes d'Alice, a été abondamment utilisée dans cet ouvrage, il a fallu se servir d'un subterfuge pour signaler au lecteur, dans les listes de programme, quelles sont les couleurs utilisées, et quels symboles graphiques doivent être frappés au clavier.

La couleur peut être introduite dans les programmes de deux façons :

- soit dans une variable alphanumérique A$,
- soit directement dans une instruction PRINT.

Dans les deux cas, la présence de couleur est signalée par une instruction REM, qui précise que les nombres apparaissant dans A$ ou dans PRINT sont en réalité des références de couleurs, les symboles graphiques à frapper étant énumérés immédiatement après.

Par exemple, si on trouve, dans le programme :

A$ = "6" : REM couleur 6 ; SHIFT Q,
cela signifie que la variable A$ contient non pas le nombre 6, mais le symbole graphique Shift Q (un pavé de couleur) frappé dans la couleur numéro 6.

De même PRINT "686" : REM couleur 6 et 8 ; SHIFT A, puis SHIFT Q et SHIFT F signifient qu'il faudra, après la frappe du 1er guillemet, se mettre dans la couleur numéro 6, frapper le symbole graphique SHIFT A ; se mettre dans la couleur 8 et frapper SHIFT Q, puis revenir dans la couleur 6 et frapper SHIFT F.

A l'allumage, les symboles frappés sont dans la couleur Ø (curseur clignotant vert).

On change de couleur en frappant CONTROL Ø, autant de fois qu'il est nécessaire.

Les numéros des couleurs correspondent à l'ordre dans lequel on les obtient, en frappant CONTROL Ø.

| Numéro | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| Couleur | noir | vert | jaune | bleu roi | rouge | ivoire | bleu pâle | mauve | orange |

Les caractères qui apparaissent en minuscules sur les listes de programme ont été tapés en contraste inversé : le passage en contraste inversé est obtenu en tapant SHIFT Ø, le retour à l'écriture normale s'obtient en tapant une seconde fois SHIFT Ø.

# NIVEAU 1

| *N°* | *Nom du programme* | *Thème et particularités* |
|---|---|---|
| 1 | **Le nombre mystérieux** | Jeu de déduction, pour un joueur. Plusieurs niveaux |
| 2 | **Cube** | Jeu de dessin : perspective sur un écran |
| 3 | **Chaud, froid** | Logique et hasard, pour un joueur. Plusieurs niveaux. |
| 4 | **Les allumettes** | Jeu de réflexion, contre l'ordinateur. |
| 5 | **Allumettes à plusieurs** | Jeu de réflexion et de stratégie, pour plusieurs joueurs. |
| 6 | **Calendrier perpétuel** | Confection d'un calendrier, mois par mois. |
| 7 | **Bataille navale** | Un seul navire à trouver, mais quelques particularités |
| 8 | **Golf** | Un parcours de 18 trous, pour un seul joueur. |
| 9 | **Le jeu du 21** | Jeu de hasard, l'ordinateur tient la banque. |
| 10 | **Pair - Impair** | Un jeu de casino en trois temps (1 ou plusieurs joueurs). |
| 11 | **Devine un chiffre** | Un jeu très facile, pour un ou plusieurs joueurs. |
| 12 | **Le jeu de Wytthof** | Un jeu de réflexion assez simple, pour deux joueurs. |
| 13 | **Rebonds** | Un jeu d'adresse très visuel, pour un joueur. |
| 14 | **Jackpot** | Jeu de hasard, pour un joueur. |
| 15 | **Roulette** | Une simulation élégante du jeu très connu. |
| 16 | **Couloirs** | Un jeu de logique, à découvrir. |
| 17 | **Echiquier** | Une base graphique pour vos propres créations. |
| 18 | **Le jeu de la fève** | Un jeu de réflexion, pour deux joueurs. |
| 19 | **Jeu de l'oie aléatoire** | Un très vieux jeu, revu et corrigé par l'ordinateur, 2 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 (Ø) 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 I, 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 évolués, 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 Alice, comme dans la plupart des Basic, le LET est devenu optionnel, aucune ambiguïté n'étant possible, on écrira donc : A = 12 ; NOM$ = "TOTO" ; etc.

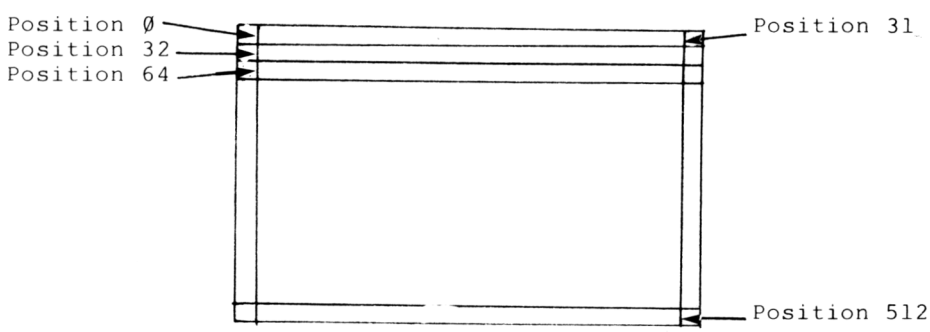
## L'INSTRUCTION PRINT, OU INSTRUCTION DE SORTIE (ECRAN)

***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 1ère 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 Nième colonne à partir de la gauche ($0 \leqslant N \leqslant 31$).
- 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 prédéfinie.

Lorsque la mise en page sur l'écran nécessite une tabulation verticale, il faudra faire appel à l'instruction PRINT@xxx, : le nombre qui suit PRINT est compris entre 0 et 512, et il définit la première position d'impression par rappot à la mémoire d'écran. La position en haut à gauche est la position Ø, la dernière position possible, en bas à droite, étant la position 512, suivant le schéma ci-dessous :



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 Alice 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 ENTER, 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** 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, et d'introduire des commandes de mouvement du curseur.

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 2Ø, 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 1ØØ, 2ØØ, 3ØØ et le renvoi sera fait à la ligne
1ØØ si X vaut 1, à la ligne 2ØØ si X vaut 2, et 3ØØ si X
vaut 3.
```

**L'INSTRUCTION IF⌊___⌋ THEN⌊____⌋**
**OU INSTRUCTION DE TEST**

On peut l'utiliser de deux façons :

- soit 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 du programme, si par exemple X = Ø, 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 présentée à la suite de THEN peut être une affectation A = 12, une impression PRINT A, une entrée INPUT B Ø, 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 2Ø, 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 (cest-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 = ⎫  Ø TO 2Ø STEP 2
        ⎬  Instructions
NEXT I  ⎭
```

Les instructions contenues dans la boucle seront exécutées pour I = Ø, pour I = 2, ... et ceci jusqu'à I = 2Ø, 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 = 2Ø, par un test portant sur une quantité quelconque.

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

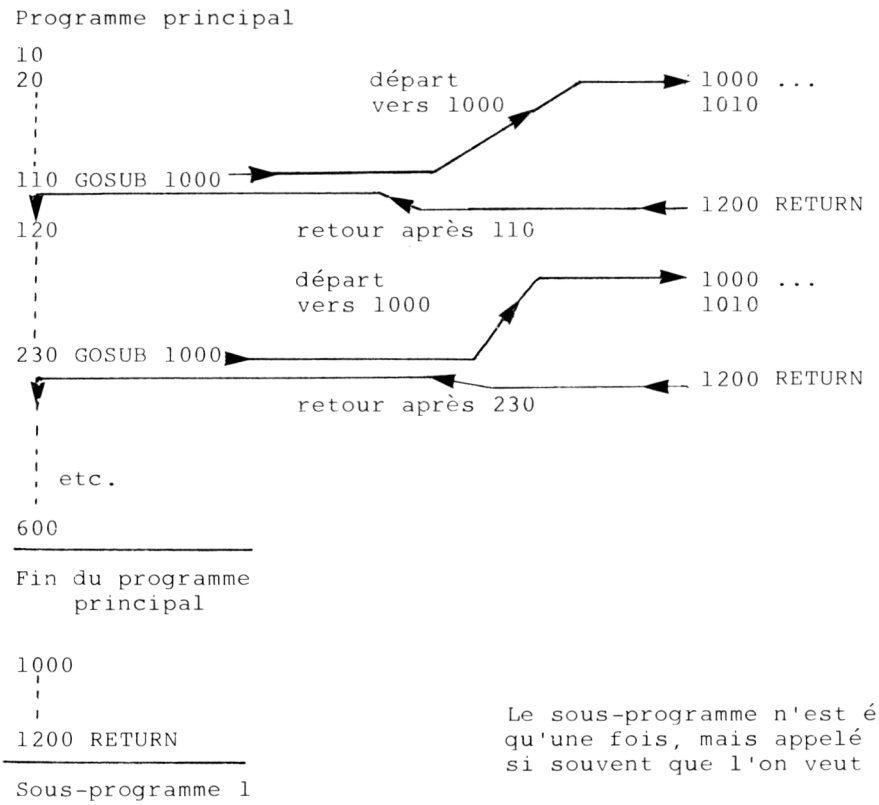
Il est possible d'emboîter plusieurs boucles de noms différents, et d'omettre l'indication de la variable dans le NEXT.

**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'instruc-**

**tion 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 :




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⌊___⌋GOSUB⌊___⌋ .

***Par exemple*** ON X GOSUB 1ØØØ, 2ØØØ, 3ØØØ permettra d'exécuter le sous-programme placé en 1ØØØ si X = 1, en 2ØØØ si X = 2, en 3ØØØ si X = 3.

# Le nombre mystérieux

***REFERENCES*** : Difficulté de programmation : facile.
Catégorie de jeu : jeu de déduction pour un joueur.
Difficulté du jeu : facile

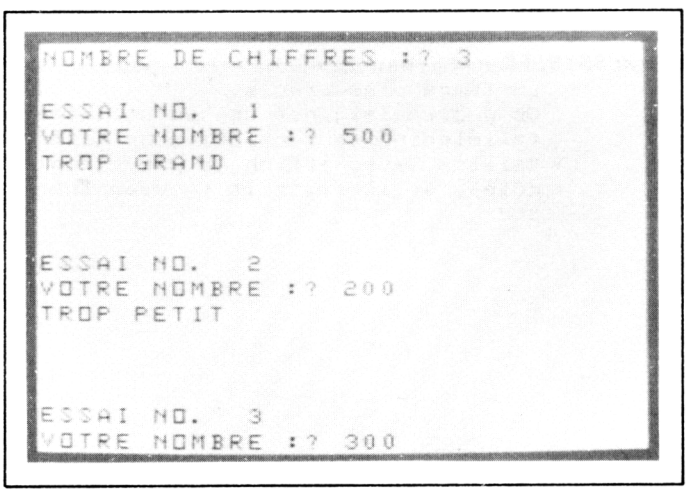
***LE JEU*** : On demande à l'ordinateur de fabriquer un nombre de n chiffres (avec n compris entre 1 et 7).
On devra ensuite essayer de le découvrir, avec le moins d'essais possibles, en recevant les seuls renseignements TROP GRAND ou TROP PETIT.

***LE PROGRAMME*** : La ligne 40 permet de fabriquer le nombre à deviner.
Les entrées sont faites par les lignes 60 à 70.
La ligne 80 teste si le résultat est exact.
La ligne 90 signale un essai trop petit, et la ligne 100 un essai trop grand.
Le compteur d'essais est en ligne 110.
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 ENTER.
On obtient alors l'affichage ESSAI N° 1, suivi de la question VOTRE NOMBRE ?
On entre un nombre, et l'on presse ENTER.
On obtient alors une réponse TROP PETIT, ou TROP GRAND, ou encore, si l'on a trouvé juste un affichage de VOUS AVEZ TROUVE...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 ENTER).
On peut arrêter à tout moment en tapant BREAK.

***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
15 CLS
20 INPUT"NOMBRE DE CHIFFRES :";N
30 IF N > 7 THEN RUN
40 X = INT(RND(0)*10^N+1)
50 J =1
60 PRINT : PRINT "ESSAI NO. ";J
70 INPUT"VOTRE NOMBRE :";M
80 IF M = X THEN 200
90 IF M < X THEN PRINT "TROP PETIT"
100 IF M > X THEN PRINT "TROP GRAND"
110 J = J + 1
120 PRINT:PRINT
130 GOTO 60
200 CLS
210 PRINT"VOUS AVEZ TROUVE ";X;" EN ";J;" ESSAIS"
220 INPUT"AUTRE PARTIE (O/N)";R$
230 IF R$ = "O" THEN RUN
```



# Cube

***REFERENCES*** : Difficulté de programmation : aucune.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

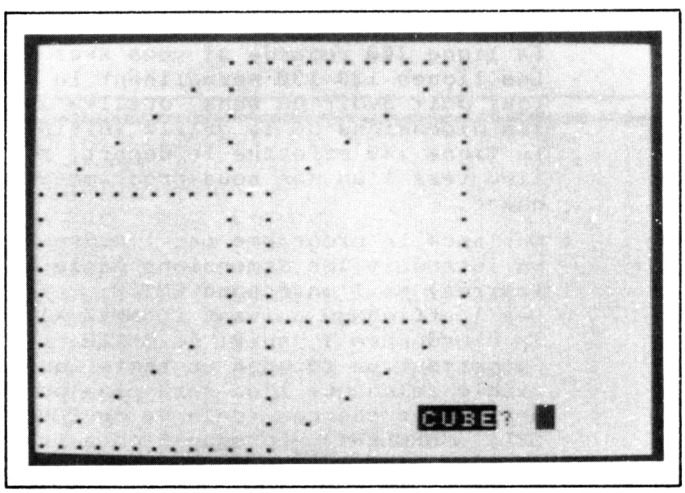
***LE JEU*** : Le programme proposé se contente de réaliser à l'aide d'instructions PRINT une vue d'un cube en perspective cavalière.
Il pourrait être par exemple incorporé à un jeu basé sur les capacités de reconnaissance des formes géométriques élémentaires.

***LE PROGRAMME*** : Le dessin du cube est réalisé par l'impression de points, en utilisant trois boucles : la boucle I (2Ø-7Ø), I (8Ø-17Ø) et la boucle I (14Ø-19Ø).

***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
15 CLS
20 FOR I=0 TO 12
30 PRINT@(10+I),".";
40 PRINT@(160+I),".";
50 PRINT@(330+I),".";
60 PRINT@(480+I),".";
70 NEXT I
80 FOR I=0 TO 9
90 PRINT@(32*I+10),".";
100 PRINT@(32*I+22),".";
110 PRINT@(32*I+160),".";
120 PRINT@(32*I+172),".";
130 NEXT I
140 FOR I=0 TO 5
150 PRINT@(32*(5-I)+2*I),".";
160 PRINT@(32*(15-I)+2*I),".";
170 PRINT@(32*(5-I)+2*I+12),".";
180 PRINT@(32*(15-I)+2*I+12),".";
190 NEXT I
200 PRINT@468,"cube";:INPUTL
```



# Chaud, froid

**REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

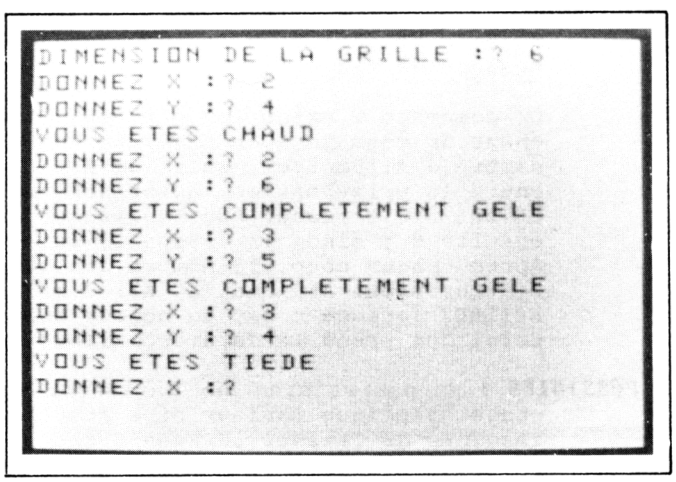
**LE JEU** : Un des plus vieux jeu du monde, toujours aussi pratiqué, mais cette fois, c'est l'ordinateur qui vous sert de partenaire : il va en effet cacher un objet à un certain emplacement d'une grille dont vous aurez vous-même choisi les dimensions. Pour le trouver, il ne vous donnera que quelques indications, sous la forme TU BRULES, ou C'EST FROID, etc. Vous devez réussir à découvrir l'objet caché, avec le moins d'essais possibles (votre score sera affiché en fin de partie).

**LE PROGRAMME** : Les lignes 40 et 50 définissent la position de l'objet.
La ligne 90 calcule 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 ENTER.
Dès l'affichage suivant (DONNEZ X), on entre la coordonnée Y, suivi de ENTER.
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 BREAK.

**EXTENSIONS POSSIBLES** : Si l'on dispose de la couleur, il sera intéressant de trouver un moyen plus visuel de 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 CLS1
30 INPUT"DIMENSION DE LA GRILLE :";D
40 X=INT(RND(D))
50 Y=INT(RND(D))
60 INPUT"DONNEZ X :";A
70 INPUT"DONNEZ Y :";B
80 S=S+1
90 T=ABS(X-A)+ABS(Y-B)
100 IFT=0THEN6000
110 PRINT"VOUS ETES ";
120 N=INT(SQR(D)/T)+1
130 IFN>5THENN=5
140 IFT<=SQR(D)THENONNGOTO1000,2000,3000,4000,5000
150 PRINT"COMPLETEMENT GELE"
160 GOTO60
1000 PRINT"A PEINE TIEDE"
1010 GOTO60
2000 PRINT"TIEDE"
2010 GOTO60
3000 PRINT"CHAUD"
3010 GOTO60
4000 PRINT"TRES CHAUD"
4010 GOTO60
5000 PRINT"BRULANT"
5010 GOTO60
6000 CLS7
6010 PRINT"VOUS AVEZ TROUVE EN";S;"COUPS"
```



# Les allumettes

**REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : contre Alice.
Difficulté du jeu : moyenne.

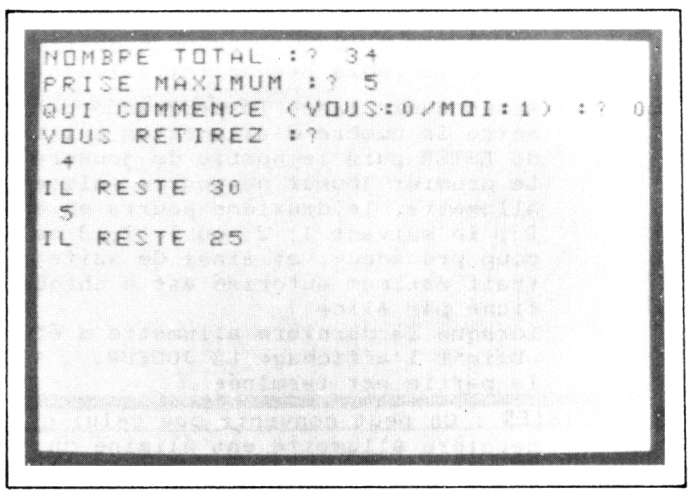
**LE JEU** : Le joueur devra en début de partie choisir une certaine quantité d'allumettes, ainsi que la prise maximum autorisée. Il décide ensuite s'il commence, ou s'il laisse commencer Alice. Chacun prend à tour de rôle autant d'allumettes qu'il le désire, dans la limite du nombre maximum autorisé.
Celui qui prend la dernière allumette est considéré comme perdant.

**LE PROGRAMME** : Le jeu du joueur commence en ligne 7Ø.
Celui d'Alice commence en ligne 11Ø.
La ligne 8Ø contrôle la légalité du coup du joueur.
La ligne 9Ø gère le changement de joueur.
Le coup d'Alice est calculé dans une boucle non formelle, de la ligne 12Ø à la ligne 15Ø.
Le résultat est affiché par la ligne 1Ø2Ø.
Si la partie est terminée, la ligne 1Ø3Ø le détecte et envoie en ligne 2ØØØ, 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 ENTER. Puis on entre la prise maximum autorisée, suivie de ENTER. Si le joueur veut commencer, il frappe ensuite Ø ; sinon il frappe 1, puis ENTER.
Après chaque coup, le nombre d'allumettes restantes est affiché, et Alice 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 CLS1
30 INPUT"NOMBRE TOTAL :";N
40 INPUT"PRISE MAXIMUM :";I
50 INPUT"QUI COMMENCE (VOUS:0/MOI:1) :";D
60 IFD=1THEN110
70 PRINT@96,"VOUS RETIREZ :";:INPUTP
80 IFP<>INT(P)ORP<1ORP>IORP>NTHEN70
90 D=1
100 GOTO1000
110 S=I+D
120 T=(N-S)/(I+D)
130 IFT=INT(T)THEN200
140 S=S-D
150 GOTO120
200 P=ABS(S-D)
210 IFP=0THENP=D
220 D=0
1000 PRINTP
1010 N=N-P
1020 PRINT"IL RESTE";N
1030 IFN=0THEN2000
1040 GOTO60
2000 CLS3
2010 IFD=1THENPRINT"J'AI GAGNE":SOUND100,7:SOUND140,7:SOUND120,7
2020 IFD<>1THENPRINT"BRAVO":SOUND5,20
```



# Allumettes à plusieurs

***REFERENCES*** : Difficulté de programmation : facile.
Catégorie de jeu : logique (plusieurs joueurs)
Difficulté du jeu : assez difficile.

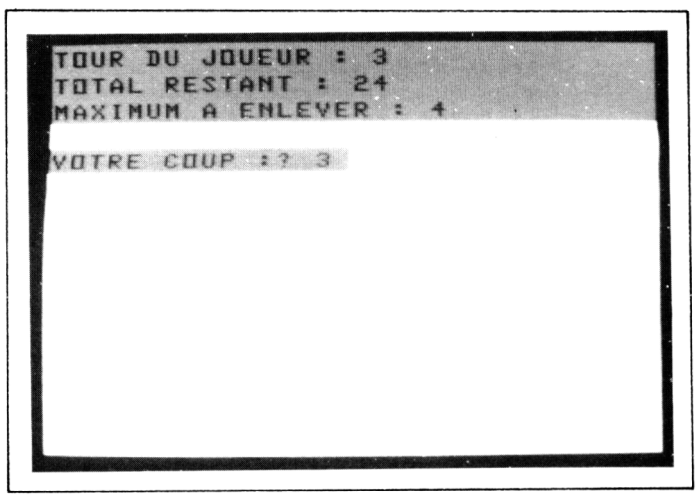
***LE JEU*** : Dans ce jeu pour plusieurs joueurs, le meneur de jeu devra, en début de partie, choisir un nombre d'allumettes, et indiquer à Alice le nombre de joueurs.
A chaque coup, le joueur qui a le trait est autorisé à retirer du tas d'allumettes restantes jusqu'au double de ce que vient de retirer le précédent joueur ; celui qui retire la dernière allumette a gagné (ou perdu si l'on modifie légèrement les règles).

***LE PROGRAMME*** : Les lignes 1Ø à 4Ø permettent les différentes entrées.
Les lignes 7Ø à 1ØØ réalisent les impressions.
La ligne 12Ø contrôle le coup de chaque joueur (par rapport au coup précédent).
La ligne 17Ø gère les tours (remise à 1 du numéro du joueur).
La ligne 15Ø réalise le test de fin, et la ligne 21Ø 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 ENTER puis le nombre de joueurs (et ENTER).
Le premier joueur ne pourra enlever qu'une allumette, le deuxième pourra 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 Alice.
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 CLS4
30 INPUT"NOMBRE TOTAL D'ALLUMETTES :";N
40 INPUT"NOMBRE DE JOUEURS :";J
50 F=1
60 K=1
70 CLS5
80 PRINT"TOUR DU JOUEUR :";K
90 PRINT"TOTAL RESTANT :";N
100 PRINT"MAXIMUM A ENLEVER :";F
110 PRINT@128,"VOTRE COUP :";:INPUTC
120 IFC=0ORC>FTHEN110
130 N=N-C
135 IFN<0THENN=N+C:GOTO110
140 F=C+C
150 IFN=0THEN200
160 K=K+1
170 IFK>JTHENK=1
180 GOTO70
200 CLS6
210 PRINT"LE JOUEUR NUMERO";K;"A GAGNE"
```



# Calendrier perpétuel

**REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : gadget publicitaire.
Difficulté du jeu : aucune.

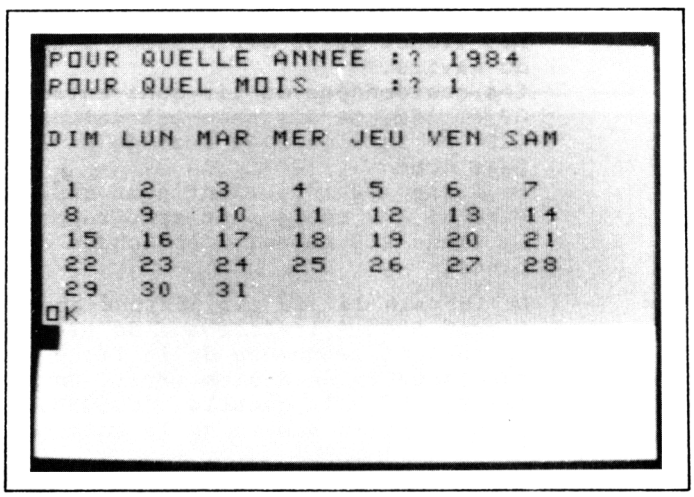
**LE JEU** : Ce programme vous permettra de confectionner tous les ans votre calendrier ; vous pourrez ainsi vous en servir pour envoyer à vos amis un calendrier du mois de leur naissance, ou pour leur souhaiter une bonne fête, etc.

**LE PROGRAMME** : Les lignes 3Ø et 4Ø vous permettront d'entrer le mois et l'année.
Les lignes 5Ø, 6Ø et 7Ø calculent le numéro du premier jour du mois.
La ligne 8Ø calcule le nombre de jours du mois (il n'est pas tenu compte ici des 29 février pour simplifier).
La boucle A (12Ø-15Ø) 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, 1984, suivie de ENTER, 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 ENTER, 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 CLS8
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=4ORM=6ORM=9ORM=11)+(M=2)*3
90 PRINT
100 PRINT"DIM LUN MAR MER JEU VEN SAM"
110 PRINT
120 FORA=1TOM
130 PRINTTAB(4*(D+A-1));A;
140 IFD+A-1=6THENPRINT:D=D-7
150 NEXTA
```



# Bataille navale

***REFERENCES*** : Difficulté de programmation : facile.
Catégorie de jeu : réflexion.
Difficulté du jeu : facile.

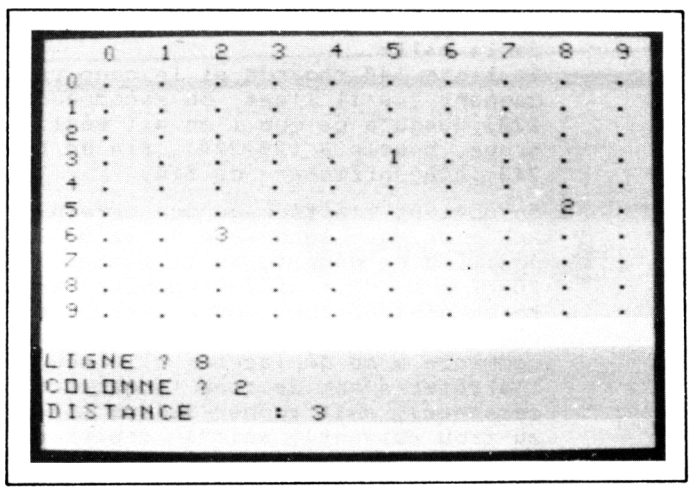
***LE JEU*** : Une bataille navale a priori assez simple, avec un seul navire caché au hasard dans une grille de 10 cases sur 10 cases.
A chaque coup, on devra tirer en donnant les coordonnées H (horizontale) et V (verticale) de la case visée, et Alice répond en affichant la distance séparant le navire 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 (3Ø-5Ø) dessine le terrain de jeu.
Les lignes 6Ø-7Ø définissent l'emplacement du navire.
Les coordonnées du tir sont entrées aux lignes 8Ø et 1ØØ. La distance est calculée en ligne 12Ø, et la ligne 17Ø regarde si l'on a ou non fait mouche.
La ligne 19Ø intervient pour déplacer le navire si les coups sont trop rapprochés.
La ligne 23Ø 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 ENTER, et, à la question COLONNE, on répond en donnant le numéro de la colonne verticale suivi de ENTER.
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 ALICE, mettre plusieurs navires, des sous-marins, etc.

```
10 REM BATAILLE NAVALE
20 CLS3:PRINT"   0  1  2  3  4  5  6  7  8  9"
30 FORI=0TO9
40 PRINTI;".  .  .  .  .  .  .  .  .  ."
50 NEXTI
55 S=1
60 X=INT(RND(10)-1)
70 Y=INT(RND(10)-1)
80 PRINT@384,"LIGNE ";:INPUTH
100 INPUT"COLONNE ";V
105 PRINT@(32*(H+1)+3*V+2),S;
107 IFV=9THEN120
110 FORI=3*V+6TO30STEP3:PRINTTAB(I)".";:NEXTI
120 D=INT(SQR((X-H)*(X-H)+(Y-V)*(Y-V)))
170 IFD=0THEN220
180 S=S+1
190 IFD=1THENPRINT@448,"PFFFF.....":GOTO60
200 PRINT@448,"DISTANCE     :";D
210 GOTO80
220 CLS8
230 PRINT"TROUVE EN";S;"COUPS"
```



# Golf

***REFERENCES*** : Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez facile.

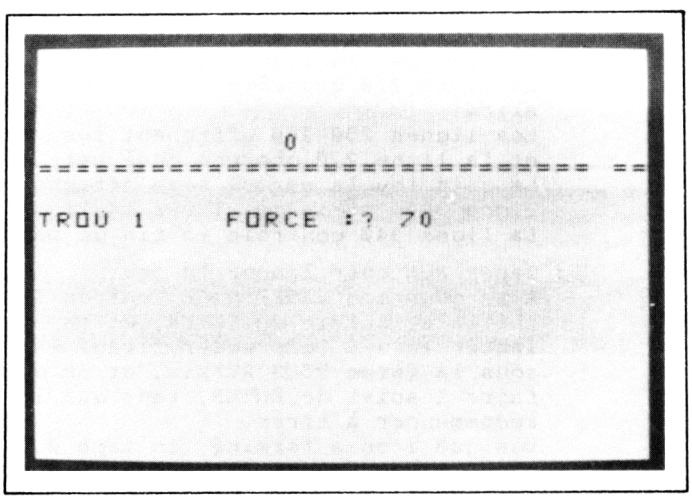
***LE JEU*** : Il s'agit d'essayer de réaliser, avec le moins de coups possibles, un parcours de golf de 18 trous. Pour cela, il vous faut frapper la balle avec une force plus ou moins grande, jusqu'à obtenir sa chute dans le trou. Mais attention, il peut y avoir un certain aléa sur chaque coup, et vous serez amené souvent à revenir en arrière (il faudra alors faire précéder l'indication de la force par un signe -).

***LE PROGRAMME*** : La ligne 4Ø permet de définir la position du trou, et la boucle I (5Ø-8Ø) dessine le terrain de jeu.
Le déplacement à réaliser est calculé en ligne 95.
La boucle I (1ØØ-13Ø) réalise le déplacement de la balle.
La ligne 14Ø regarde si le coup est, ou non gagnant ; s'il l'est, on recommence (ligne 22Ø) jusqu'à ce que l'on ait réalisé 18 trous, boucle S (2Ø-22Ø). Fin de boucle en 24Ø, 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 3ØØ...). Dès que l'on aura tapé ENTER, la balle commence 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 FORS=1TO18
30 CLS
35 X=1
40 T=INT(RND(24)+6)
50 FORI=0TO31
60 PRINT@(128+I),"="
70 IFI=TTHENPRINT@(128+T)," "
80 NEXTI
85 PRINT@96,"";:PRINTTAB(X)"0"
90 PRINT@192,"TROU"S;:INPUT"   FORCE :";F
95 F=INT(F*.16-(ABS(F)*(SQR(RND(0))+.1)^2)/1000)
97 IFF=0THEN85
100 FORI=SGN(F)TOFSTEPSGN(F)
110 X=X+SGN(F)
120 IFX<30ANDX>=1THENPRINT@96,"";:PRINTTAB(X)"0"
125 IFX>=30THENX=30
127 IFX<=1THENX=1
130 NEXTI
140 IFX=TTHEN190
150 E=E+1
160 GOTO50
190 PRINT@96
200 PRINT@(128+X),"0":FORI=X+1TO31:PRINT@(128+I),"=":NEXTI
210 FORI=1TO1500:NEXTI
220 NEXTS
230 CLS8
240 PRINT"18 TROUS EN";E;"COUPS"
```



# Le jeu du 21

**REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : jeu contre Alice.
Difficulté du jeu : facile.

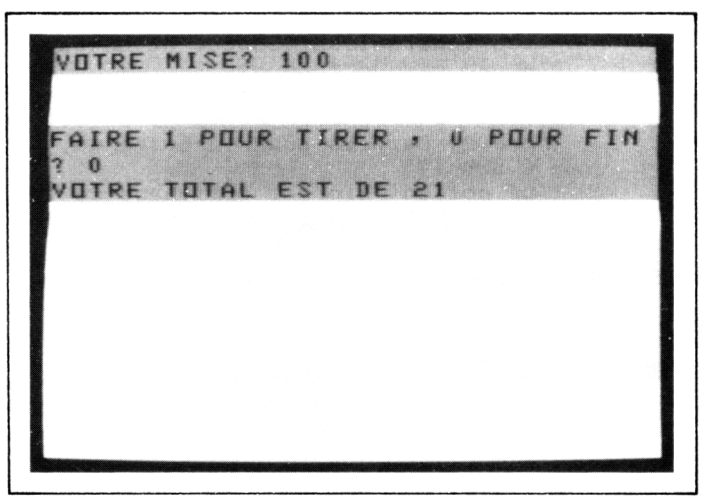
**LE JEU** : On devra entrer une mise au début du jeu, et on pourra lancer les dés en tapant 1 ; à chaque lancer, on obtiendra un nouveau total, qui augmentera peu à peu, et se rapprochera de 21. On devra essayer de se rapprocher le plus possible de 21, sans dépasser ce nombre. Lorsque l'on décidera de s'arrêter, il suffira de taper Ø pour laisser jouer Alice 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 5Ø, 6Ø et 7Ø réalisent le contrôle de celui qui doit jouer (joueur ou Alice).
Le sous-programme (1ØØØ-1Ø1Ø) lance les dés.
Les lignes 9Ø et 1ØØ calculent le total fait par le joueur et l'affichent.
Le total d'Alice est calculé en ligne 18Ø et affiché par la ligne 19Ø.
La ligne 21Ø contrôle la fin du jeu pour Alice.
Les lignes 25Ø-26Ø affichent les résultats et la ligne 27Ø cherche quel est le gagnant.
Le gain (ou la perte) sera affiché par la ligne 3ØØ, et le total par la ligne 32Ø.
La ligne 34Ø contrôle la fin de partie.

**POUR JOUER** : Taper RUN pour lancer le jeu.
A la question VOTRE MISE, entrer la mise initiale, suivie de ENTER, et taper 1 pour lancer le dé. Le premier tirage est affiché sous la forme VOUS AVEZ... et on devra refaire 1 suivi de ENTER, tant que l'on désire recommencer à tirer.
Dès que l'on a terminé, on tape Ø et ENTER pour laisser jouer Alice (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 JEU DU 21
20 G=0
30 CLS2:T=0
40 INPUT"VOTRE MISE";M
50 PRINT@96,"FAIRE 1 POUR TIRER , 0 POUR FIN"
60 INPUTR
70 IFR=0THEN150
80 GOSUB1000
90 T=T+X
100 PRINT@160,"VOTRE TOTAL EST DE"T
110 GOTO50
150 CLS5:PRINT"LA BANQUE JOUE"
160 S=0
170 GOSUB1000
180 S=S+X
190 PRINT@96,"TOTAL DE LA BANQUE"S
200 FORI=1TO1500:NEXTI
210 IF(S>T)OR(S>17)THEN250
220 GOTO170
250 CLS8:PRINT@96,"VOUS AVEZ"T
260 PRINT@160,"LA BANQUE A"S
270 IF(S>21)OR((S<T)AND(T<=21))THEN300
280 M=-M
300 PRINT@224,"GAIN"M
310 G=G+M
320 PRINT@288,"VOTRE TOTAL ACTUEL"G
330 PRINT@352,"ON CONTINUE (O/N)";:INPUTR$
340 IFR$="N"THENEND
350 GOTO30
1000 X=INT(RND(6))
1010 RETURN
```



# Pair - impair

***REFERENCES*** : Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

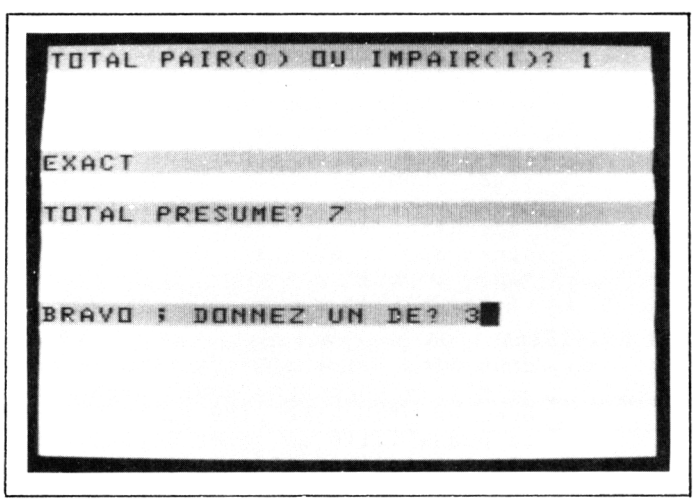
***LE JEU*** : C'est un jeu de hasard, pour un ou plusieurs joueurs. Alice lance simultanément deux dés, non visibles et pose 3 questions : quelle est la parité du total, quel est le total, et enfin donner la valeur de l'un des dés, on peut ensuite soit poursuivre le jeu, soit l'interrompre à tout moment.

***LE PROGRAMME*** : Les dés sont tirés par un sous-programme placé aux lignes 1ØØØ et 1Ø1Ø.
La ligne 6Ø tient compte du nombre de parties et les lignes 15Ø, 2ØØ et 23Ø comptabilisent les points marqués.
La parité du total est déterminée par les lignes 1ØØ et 11Ø (la ligne 11Ø compare avec la réponse du joueur).
La ligne 41Ø réalise l'affichage de fin de partie : nombre de points, nombre de parties jouées.
La ligne 43Ø 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 ENTER 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 1Ø.
On obtient alors l'affichage de la question ON CONTINUE (O/N) et l'on devra répondre N suivi de ENTER 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 de 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 GOSUB1000
40 A=X
50 GOSUB1000
60 R=R+1
70 S=A+X
80 CLS8
90 INPUT"TOTAL PAIR(0) OU IMPAIR(1)";N
100 P=S/2-INT(S/2)
110 IF(N=P)OR(N-.5=P)THEN140
120 PRINT@128,"FAUX "
130 GOTO160
140 PRINT@128,"EXACT"
150 T=T+1
160 PRINT@192,"TOTAL PRESUME";:INPUTN
170 IFN=STHEN200
180 PRINT@256,"PERDU"
185 FORI=1TO1500:NEXTI
190 GOTO400
200 T=T+2
210 PRINT@320,"BRAVO ; DONNEZ UN DE";:INPUTN
220 IF(N<>A)AND(N<>S-A)THEN180
230 T=T+3
240 PRINT@384,"BRAVO":GOTO185
400 CLS4
410 PRINT"VOUS AVEZ UN TOTAL DE"T"SUR"R
420 PRINT@96,"ON CONTINUE (O/N) ";:INPUTR$
430 IFR$="N"THENEND
440 GOTO30
1000 X=INT(RND(6))
1010 RETURN
```



# Devine un chiffre

*REFERENCES* : Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

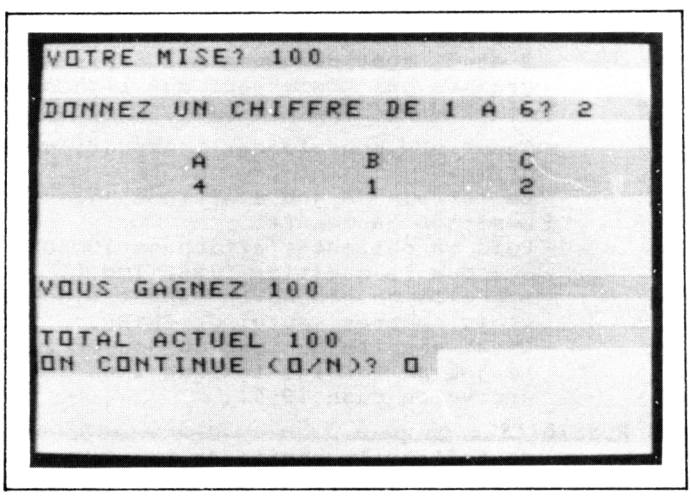
*LE JEU* : C'est un jeu de hasard, de conception très simple, mais néanmoins assez passionnant : le joueur introduit sa mise, et parie sur un chiffre, de 1 à 6 ; trois dés sont alors lancés ; si le chiffre joué apparaît une fois le joueur gagne sa mise. S'il apparaît deux fois il double sa mise ; s'il 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 120 et 130.
Les lignes 150 à 180 calculent le gain dans les diverses configurations possibles.
La totalisation est effectuée en ligne 200 pour les gains, et en ligne 510 pour les pertes.
La variable R$, testée en ligne 540 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 ENTER. On obtient alors un deuxième affichage DONNEZ UN CHIFFRE DE 1 A 6 ; on entre un chiffre suivi de ENTER ; 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 ENTER).

*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 Alice.

```
10 REM DEVINE UN CHIFFRE
20 G=0
30 CLS3
40 INPUT"VOTRE MISE";N
50 PRINT@64,"DONNEZ UN CHIFFRE DE 1 A 6";:INPUTD
60 GOSUB1000
70 A=X
80 GOSUB1000
90 B=X
100 GOSUB1000
110 C=X
120 PRINT@128,"         A         B         C      "
130 PRINT@160,"";TAB(7)A;TAB(16)B;TAB(24)C
140 T=0
150 IFD=ATHENT=T+N
160 IFD=BTHENT=T+N
170 IFD=CTHENT=T+N
180 IFT=0THEN500
190 PRINT@288,"VOUS GAGNEZ"T
200 G=G+T
210 GOTO520
500 PRINT@288,"PERDU"
510 G=G-N
520 PRINT@352,"TOTAL ACTUEL"G"
530 INPUT"ON CONTINUE (O/N)";R$
540 IFR$="N"THENEND
550 GOTO30
1000 X=INT(RND(6))
1010 RETURN
```



# Le jeu de Wytthof

*REFERENCES* : Difficulté de programmation : moyenne.
Catégorie de jeu : reflexion (deux joueurs).
Difficulté du jeu : moyenne.

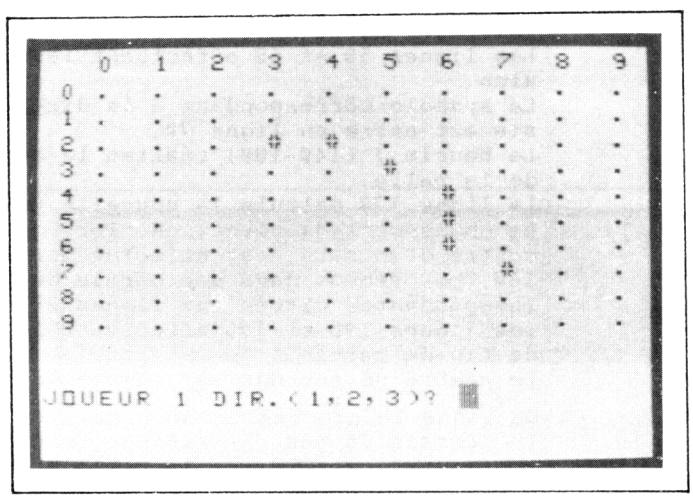
*LE JEU* : On dispose d'une grille carrée de 10 cases sur 10 cases. Un point apparaît, en dehors de la diagonale et les joueurs devront le déplacer à chaque tour, suivant seulement trois directions : droite, bas et diagonale bas et droite. Le premier joueur qui arrive en (9,9) sera considéré comme gagnant,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

*LE PROGRAMME* : Les lignes 2Ø et 3Ø permettent de définir une position de départ, en dehors de la diagonale.
La ligne 4Ø et la boucle I (5Ø-7Ø) dessinent le terrain de jeu.
La ligne 8Ø imprime la position du point mobile.
Le test de fin est effectué en ligne 9Ø.
Le changement de joueur est assuré par la ligne 12Ø.
Les directions de déplacement disponibles, 1, 2 et 3, sont décodées par trois sous-programmes qui commencent aux lignes 1ØØØ, 2ØØØ, et 3ØØØ.

*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 ENTER.
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 Alice de jouer, etc.

```
10 REM LE JEU DE WYTTHOF
20 X=3*INT(RND(5)-1)+3:Y=INT(RND(5))
30 IFX=YTHEN20
40 CLS:PRINT"   0  1  2  3  4  5  6  7  8  9"
50 FORI=0TO9
60 PRINTI;".  .  .  .  .  .  .  .  .  ."
70 NEXTI
75 H=1
80 PRINT@(32*Y+X),"#"
82 IFX=30THEN90
85 FORI=X+3TO30STEP3:PRINT@32*Y+I,".":NEXTI
90 IF(X=30)AND(Y=10)THENEND
100 PRINT@416,"JOUEUR"H;:INPUT"DIR.(1,2,3)";D
110 ONDGOSUB1000,2000,3000
120 H=2-ABS(H-1)
130 GOTO80
1000 X=X+3
1010 RETURN
2000 GOSUB1000
3000 Y=Y+1
3010 RETURN
```



# Rebonds

***REFERENCES*** : Difficulté de programmation : facile.
Catégorie de jeu : jeu d'adresse
Difficulté du jeu : moyenne.

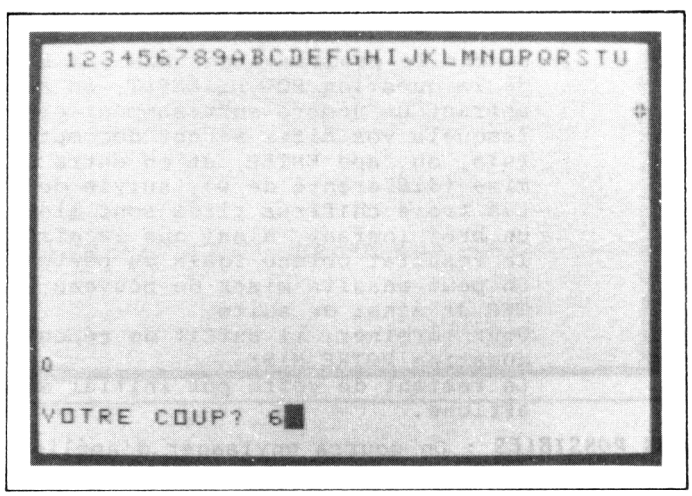
***LE JEU*** : Il utilise l'idée de la simulation d'un billard à bandes.
De part et d'autre de l'écran se trouvent la balle et le but visé : une série de chiffres et de lettres, en haut de l'écran, permettent de choisir l'endroit où l'on désire rebondir.
La balle va alors partir vers le haut de l'écran (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 2Ø imprime les chiffres et les lettres en haut de l'écran.
Les lignes 3Ø et 4Ø définissent les positions initiales de la balle et de la cible.
Les lignes 5Ø et 6Ø effectuent leur impression.
Le symbole correspondant à la direction choisie est entré en ligne 7Ø.
La boucle J (14Ø-18Ø) réalise le déplacement de la balle.
La ligne 17Ø calcule la nouvelle position.
Le changement de direction, lors de la rencontre d'un bord, est effectué par la ligne 16Ø, par renvoi dans une partie du programme indépendante, située aux lignes 25Ø et 27Ø.
Les lignes 19Ø et 195 effectuent l'impression de fin de partie.
Le nombre de rebonds est compté en ligne 26Ø.

***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 ENTER.
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 BREAK.

***EXTENSIONS POSSIBLES*** : Il sera intéressant 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 CLS:PRINT" 123456789ABCDEFGHIJKLMNOPQRSTU"
30 X=INT(RND(12)+2)
40 Y=INT(RND(13)+1)
50 PRINT@32*X,"0"
60 PRINT@(32*Y+31),"#"
70 PRINT@448,"VOTRE COUP";:INPUTR$
120 S=0
130 T=X
135 PRINT@32*X," "
140 FORJ=1TO31
150 T=T-X/(ASC(R$)-ASC("0")+7*(R$="A")+1)
160 IF(T<=1)OR(T>=15)THEN250
170 PRINT@(32*INT(T)),"";TAB(J)"0"
175 FORF=1TO500:NEXTF
177 PRINT@(32*INT(T)),"";TAB(J)" "
178 PRINT@(32*Y+31),"#"
180 NEXTJ
190 IFABS(T-Y)<3THENPRINT"BRAVO":SOUND150,8:SOUND170,8:SOUND130,8
195 PRINTS;"REBONDS"
200 INPUT"AUTRE PARTIE (O/N)";R$
210 IFR$="N"THENEND
220 RUN
250 X=-X
260 S=S+1
270 GOTO180
```



# Jackpot

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

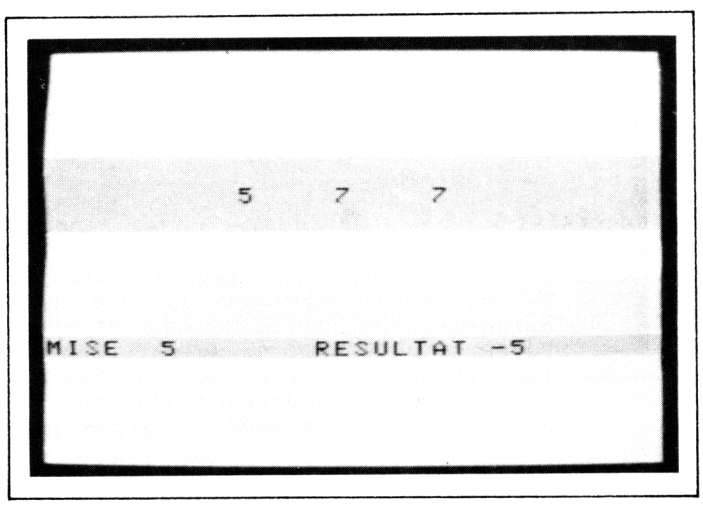
***LE JEU*** : C'est une variante simplifiée d'un jeu de hasard classique, dans lequel le joueur doit faire une mise, et tirer sur une poignée pour lancer le jeu. S'il obtient certaines combinaisons de symboles, il gagne (plus ou moins, selon sa mise et la combinaison obtenue). Ici, les symboles sont 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 (4Ø-6Ø) permet le dessin de la machine, la ligne 7Ø affichant les emplacements vides des chiffres.
La mise sera entrée en ligne 8Ø et testée en ligne 9Ø : si on mise Ø, la partie est terminée (départ vers la ligne 2ØØØ).
Les lignes 1ØØ, 12Ø et 14Ø permettent le tirage des 3 chiffres, en appelant un sous-programme qui se trouve en ligne 1ØØØ.
Le résultat est affiché par la ligne 15Ø, puis testé en ligne 16Ø, et en ligne 31Ø, 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 lesquels vos mises seront décomptées.
Puis, on tape ENTER, et on entre une première mise (différente de Ø), suivie de ENTER.
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 ENTER et ainsi de suite.
Pour terminer, il suffit de répondre Ø 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
30 INPUT"POT DE DEBUT";P
35 CLS4
40 FOR I = 4 TO 6
50 PRINT@32*I
60 NEXT I
70 PRINT@160,TAB(10);"*";TAB(15);"*";TAB(20);"*"
80 PRINT@352,"VOTRE MISE :";:INPUT M
90 IF M =0 THEN 2000
100 GOSUB 1000
110 A=X
120 GOSUB 1000
130 B=X
140 GOSUB 1000
150 PRINT@160,TAB(9);A;TAB(14);B;TAB(19);X
160 IF A=B AND A=X THEN 300
170 N=-M
180 PRINT@352,"MISE "M;"       RESULTAT ";N
190 FOR I=1 TO 1500:NEXT I
200 P=P+N
210 GOTO35
300 N=M*50
310 IF A =7 THEN N=M*500
320 GOTO180
1000 X = INT(RND(9))
1010 RETURN
2000 CLS 6
2010 PRINT"VOUS AVEZ"P
2020 PRINT"A BIENTOT"
```



# Roulette

***REFERENCES*** : Difficulté de programmation : moyenne.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : facile.

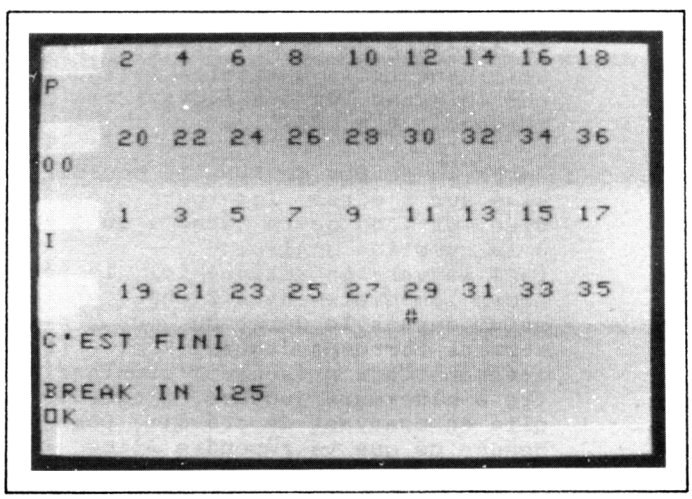
***LE JEU*** : Il est surtout spectaculaire par son aspect graphique : une boule, représentée par un carré noir, se déplace sur un terrain de jeu comportant des cases numérotées de 1 à 36, ainsi que trois cases supplémentaires, le ØØ, une case P (comme pair) et une case I (comme impair).
La boule parcourt au moins une fois chaque case, puis s'arrête au deuxième tour ; on peut ensuite la relancer pour une autre partie.

***LE PROGRAMME*** : La boucle K (2Ø-7Ø) dessine le terrain de jeu, auquel la ligne 8Ø rajoute les trois symboles P, ØØ et I.
La boucle K (95-14Ø) réalise le déplacement de la boule, l'arrêt étant contrôlé par les lignes 115 et 125.
La ligne 125 affiche le message de fin et termine le jeu.
La ligne 15Ø tire la position arrêt.

***POUR JOUER*** : Il suffit de faire RUN, et de regarder.
Une fois la boule arrêtée, il suffira de frapper une touche quelconque (sauf BREAK), pour la relancer. Taper BREAK 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
15 CLS4
20 FOR K=0 TO 3
30 FOR I=1 TO 9
35 S=I+I
40 PRINT@(96*K+3*I),S-18*(K=1)+(K=2)-17*(K=3)
50 NEXT I
70 NEXT K
80 PRINT@32,"P":PRINT@128,"00":PRINT@224,"I"
95 FORK=0 TO 3
100 FOR I=1 TO 10
105 PRINT@(96*K+34-66*(K)=1)*(I=1))
110 PRINT@(96*K+3*I+33-(I=10)),"#"
115 X = X-1
120 GOSUB 200
125 IF X=1 THEN PRINT@352,"C'EST FINI":END
130 NEXT I
140 NEXT K
150 X=INT(RND(38)-1)
160 GOTO95
200 FOR J=1 TO 500:NEXT J
210 RETURN
```



# Couloirs

**REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

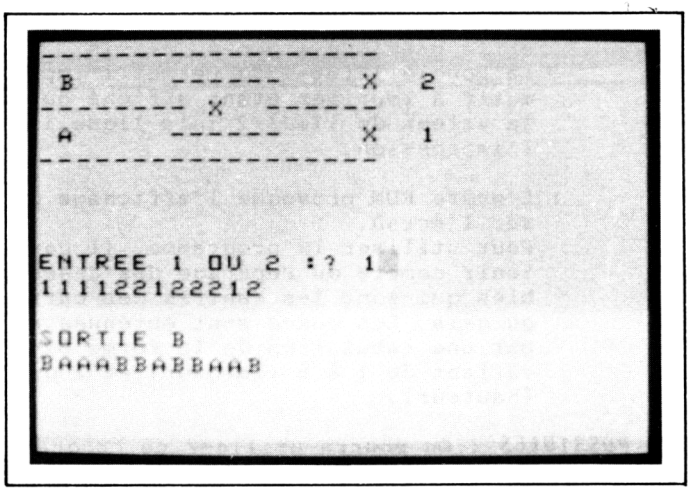
**LE JEU** : Il fait appel à des notions de logique élémentaire. Le joueur peut se présenter à l'entrée, soit du couloir numéro 1, soit du couloir numéro 2. Dans ces deux entrées, sont disposés des tourniquets qui basculent à votre passage et vous laissent passer, une fois vers la droite, et la fois 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 3Ø à 7Ø dessinent les couloirs (et les trois tourniquets).
La ligne 1ØØ attend votre choix.
Le décodage du chemin est fait par Alice en utilisant trois variables logiques I, J et K, représentant les états des trois tourniquets (lignes 13Ø à 27Ø).

**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 d'Alice.
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 affichées 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 Alice.

**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@256,"ENTREE 1 OU 2 :";:INPUTC$
110 IF C$="" OR C$>="3" OR LEN(C$)<>1 THEN 100
120 R$=R$+C$
130 PRINT@288,R$
140 IF C$="1" THEN190
150 J=NOT(J)
160 IF J<>0 THEN 250
170 S$="B"
180 GOTO 220
190 I=NOT(I)
200 IF I<>0 THEN 250
210 S$="A"
220 T$=T$+S$
230 PRINT@352,"SORTIE ";S$:PRINT@384,T$
240 GOTO 100
250 K=NOT(K)
260 IF K<>0 THEN 170
270 GOTO 210
```



# Echiquier

***REFERENCES*** : Difficulté de programmation : difficile.
Catégorie de jeu : graphisme.
Difficulté du jeu : variable, selon utilisation.

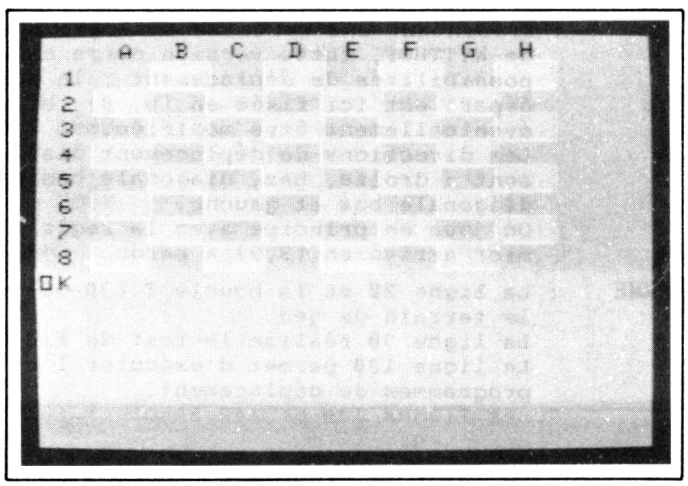
***LE JEU*** : Le programme proposé vient en appoint du programme ECHECS pour être utilisé avec une éventuelle extension de mémoire ; il donne, en effet, la possibilité de réaliser un affichage graphique de bonne qualité pour des jeux utilisant un damier de 8 cases sur 8 cases (on pourra facilement l'étendre à 10 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 4Ø définit une variable B$ qui contient les 3 éléments du dessin.
La ligne 6Ø imprime les lettres horizontales et la première ligne du dessin.
La boucle I (8Ø-15Ø) permet l'impression, le motif à imprimer étant affiché ou non selon la valeur de (I+J)/2 ; la ligne 11Ø réalise l'impression.

***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 32*I + 3*J,J variant de 1 à 8 (largeur) et I de 1 à 8 (hauteur).

***EXTENSIONS POSSIBLES*** : On pourra utiliser ce programme pour toute sorte de jeux se déroulant sur un damier (échecs, dames anglaises, le loup et les agneaux, etc.).

```
10 REM ECHIQUIER
20 CLS
40 B$="444":REM COULEUR 4 ; 3 FOIS SHIFT Q
60 PRINT"    A  B  C  D  E  F  G  H"
80 FOR I=1 TO 8
85 PRINT@(32*I),I
90 FOR J=1 TO 8
100 IF(I+J)/2=INT((I+J)/2)THEN 140
110 PRINT@(32*I+3*J),B$
140 NEXTJ
150 NEXTI
```



# 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 (Ø, Ø), 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 2Ø et la boucle I (3Ø-5Ø) dessinent le terrain de jeu.
La ligne 9Ø réalise le test de fin de partie.
La ligne 12Ø permet d'exécuter l'un des sous-programmes de déplacement.
Les lignes 13Ø et 14Ø assurent le changement de joueur.

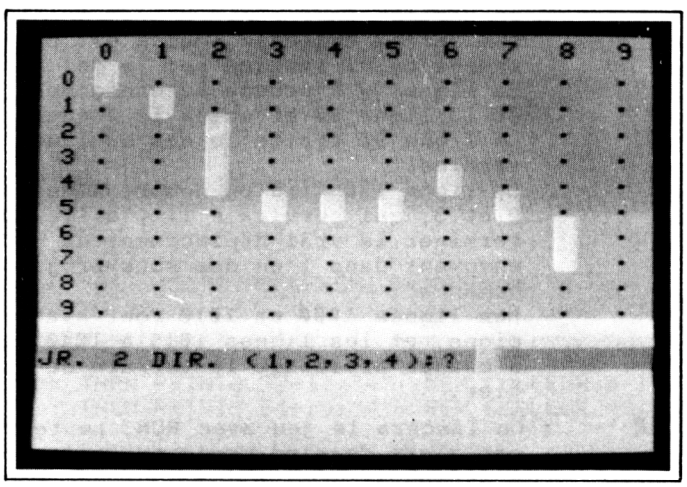
**POUR JOUER** : On lance le programme par RUN, et le terrain de jeu s'affiche, ainsi que la question DIRECTION (1, 2, 3, 4) ; les 4 directions proposées correspondent aux déplacements suivants :




Le joueur 1 entre son déplacement, et tape ENTER.
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 8:PRINT"   0  1  2  3  4  5  6  7  8  9"
30 FOR I=0 TO 9
40 PRINTI;".  .  .  .  .  .  .  .  .  ."
50 NEXT I
60 X=5:Y=1
70 J=1
80 PRINT@(32*Y+X-2),"4";:REM COULEUR 4 ; SHIFT Q
90 IF X=32 AND Y=10 THEN 200
100 PRINT@384,"JR."J;:INPUT"DIR. (1,2,3,4):";I$
110 I=VAL(I$)
120 ON I GOSUB 1000,2000,3000,4000
130 J=J+1
140 IF J=3 THEN 70
150 GOTO 80
200 PRINT@448,"LE JOUEUR"J"A GAGNE":SOUND200,9:SOUND220,7:
                                    SOUND180,13
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 Alice ; il suffira d'appuyer sur une touche après chaque lancer de dé, et de regarder ce qui se passe.

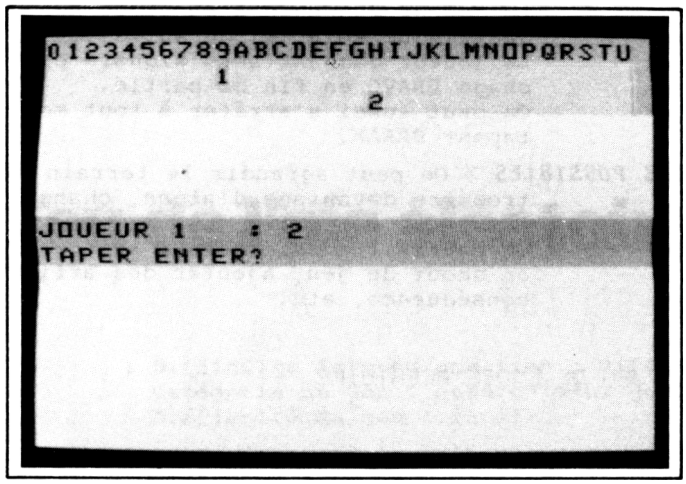
**LE PROGRAMME** : La ligne 6Ø permet d'appeler le sous-programme 1ØØØ à 1Ø6Ø qui réalise l'affichage du terrain de jeu, le contrôle de la position des joueurs et l'affichage de cette position. Cet affichage sera d'ailleurs refait en cours de partie, pour tenir compte du déplacement réellement effectué (qui n'est pas nécessairement celui indiqué par le dé).
La ligne 7Ø s'occupe du changement de joueur et la ligne 8Ø effectue le lancer de dé.
La ligne 9Ø traite le cas où l'un des joueurs termine.
La ligne 13Ø tire un nombre aléatoire entre Ø 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 1ØØØ et 1Ø1Ø font avancer les pions, et les lignes 1Ø15 à 1Ø4Ø redessinent à chaque affichage le terrain de jeu en entier.

**POUR JOUER** : On lancera le jeu avec 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 quelques 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 BREAK.

***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 LE JEU DE L'OIE ALEATOIRE
15 CLS 8
20 X=0
30 Y=0
40 J=1
50 I=X
60 GOSUB 1000
70 J=ABS(J-1)
80 I=INT(RND(6))
85 IF X>30 THEN X=26+RND(4)
87 IF Y>30 THEN Y=26+RND(4)
90 IF X=30 OR Y=30 THEN PRINT@352,"BRAVO":SOUND160,9:
                                    SOUND180,7:SOUND150,9:END
100 PRINT@224,"JOUEUR";J+1;"   :";I
110 INPUT"TAPER ENTER";R$
120 GOSUB 1000
130 I=RND(4)-1
140 F=RND(4)
150 ON F GOSUB200,300,400,500
160 GOTO 60
200 RETURN
300 I=-I
310 RETURN
400 I=I+I
410 RETURN
500 IF J=0 THEN X=RND(30)-1:RETURN
510 Y=RND(30)-1:RETURN
1000 IF J=0 THEN X=X-I*(X+I<=35)
1010 Y=Y-J*I*(Y+I<=35)
1015 PRINT@0,"0123456789ABCDEFGHIJKLMNOPQRSTU"
1020 FOR I=0 TO 30
1025 IF I=X THEN PRINT@(32+I),"1"
1027 IF I=Y THEN PRINT@(64+I),"2"
1030 IF I<>X THEN PRINT@(32+I),"4":REM COULEUR 4 ; SHIFT 0
1035 IF I<>Y THEN PRINT@(64+I),"4":REM COULEUR 4 ; SHIFT 0
1040 NEXT I
1050 FOR I=1 TO 1000:NEXT I
1060 RETURN
```


0123456789ABCDEFGHIJKLMNOPQRSTU
1
2
JOUEUR 1 : 2
TAPER ENTER?

# NIVEAU 2

| N° | Nom du programme | Thème et particularités |
|---|---|---|
| 1 | **Dé truqué** | A utiliser, si vous êtes malhonnête, dans d'autres programmes |
| 2 | **Visions** | Un jeu sur les mots, pour l'ordinateur |
| 3 | **4 2 1** | Une simulation simplifiée, plusieurs joueurs |
| 4 | **L'électricien fou** | Logique et difficile, pour un seul joueur. |
| 5 | **La course au 20** | Jeu de réflexion, pour 2 joueurs ou plus. |
| 6 | **Dessin** | Une méthode de codage originale. |
| 7 | **Musique** | Révisez vos notions de solfège, grâce à l'ordinateur. |
| 8 | **Poker** | Une base de départ, pour mettre au point votre programme. |
| 9 | **Le jeu de saute-mouton** | Un jeu de réflexion, pour deux joueurs. |
| 10 | **Lettres** | Un jeu sur les mots, assez simple. |
| 11 | **Machinamot** | Laissez l'ordinateur créer des vocables nouveaux. |
| 12 | **Espion** | Dites-lui votre amour en code. |
| 13 | **Accrochez les wagons** | Pour les plus jeunes ; très visuel. |
| 14 | **Tirage d'un dé truqué** | Révisez vos connaissances en statistiques. |
| 15 | **Le jeu de Marienbad** | Un jeu de logique pour 2 joueurs, dans une version étendue. |
| 16 | **Compose** | Faire de la musique, c'est d'abord l'écrire... |
| 17 | **Jour de la semaine** | Profitez-en pour faire le point sur les anniversaires. |
| 18 | **Tour de cartes** | Votre ordinateur a plus d'un tour dans son sac. |
| 19-20 | **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 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 PREMIER 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 Alice 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, er 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.

Ainsi DIM NOM$ (12) réserve 12 emplacements en mémoire, pour des mots de 255 caractères au maximum.

Sur Alice 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 (3Ø) ou DIM N$ (2Ø), le Ième 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 3ème élément numérique du tableau R. PRINT R$ (5) permet l'affichage du 5ème élément du tableau alphanumérique R$, etc.

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

Pour extraire le Ième élément de la chaîne alphanumérique A$, on utilisera l'instruction MID$.

Ainsi, si A$ = "BONJOUR", MID$ (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 les 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 Ø et 1, et que la somme de toutes les probabilités soit un nombre très voisin de 1).

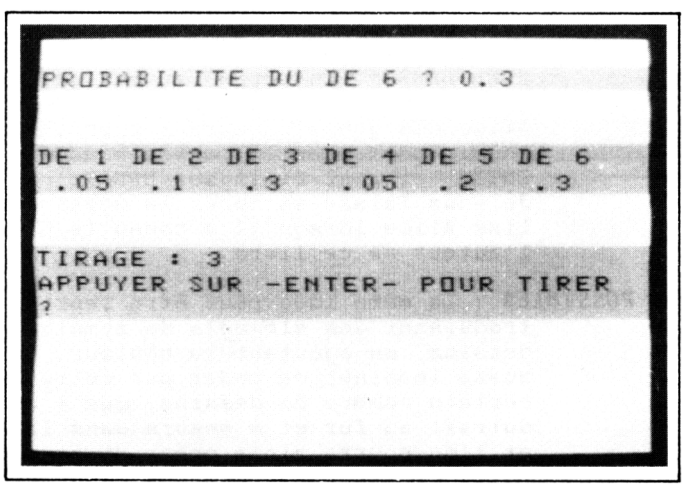
***LE PROGRAMME*** : Il utilise le générateur aléatoire d'Alice supposé de qualité convenable (on pourra aisément utiliser tout autre générateur, si besoin était). La boucle I (4Ø-1ØØ) s'occupe de l'entrée de chacune des probabilités, avec vérification à chaque entrée et sur le total (ligne 7Ø) ; chaque probabilité est conservée dans une variable indicée E (I) : ligne 5Ø du programme.
Le lancer de dé truqué est effectué à l'aide d'une autre boucle I (14Ø-16Ø), la ligne permettant le trucage se trouvant en 15Ø.
Un nouveau tirage pourra être effectué sur pression de ENTER (lignes 18Ø-19Ø).

***POUR JOUER*** : Après le lancement de l'exécution par RUN, on obtient l'affichage de la question PROBABILITE DU DE 1, à laquelle on répond par un nombre compris entre Ø 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 Ø.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 la touche ENTER pour effectuer les tirages suivants. On arrête le programme en tapant BREAK.

***EXTENSIONS POSSIBLES*** : Il sera bien évidemment tentant d'utiliser sournoisement 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 dé-

finissant le dé truqué dans une variable A$, etc. On pourra aussi envisager de présenter les résultats correspondants à une série de lancers de dés sous une forme spectaculaire qui mette le trucage en évidence.

```
10 REM DE TRUQUE
15 CLS8
20 P=0
30 DIM E(6)
40 FOR I=1 TO 6
50 PRINT@32,"PROBABILITE DU DE"I;:INPUTE(I)
60 P=P+E(I)
70 IF E(I)<0 OR E(I)>1 OR P>1.1 THEN 200
80 PRINT@(128+5*I-5),"DE"I
90 PRINT@(160+5*I-5),E(I)
100 NEXT I
110 IF P<0.9 THEN RUN
120 FOR I=1 TO 500:NEXT I
130 N=RND(6)
140 FOR I=1 TO 6
150 IF N=I AND RND(0)>E(I) THEN 130
160 NEXT I
170 PRINT@256,"TIRAGE :"N
180 PRINT"APPUYER SUR -ENTER- POUR TIRER"
190 INPUTR$:GOTO130
200 PRINT@352,"IMPOSSIBLE :0<=P<=1"
210 P=P-E(I)
220 FOR J=1 TO 1500:NEXT J
230 PRINT@352,""
240 GOTO 50
```



# Visions

***REFERENCES*** : Difficulté de programmation : facile.
Catégorie de jeu : dessin
Difficulté du jeu : facile.

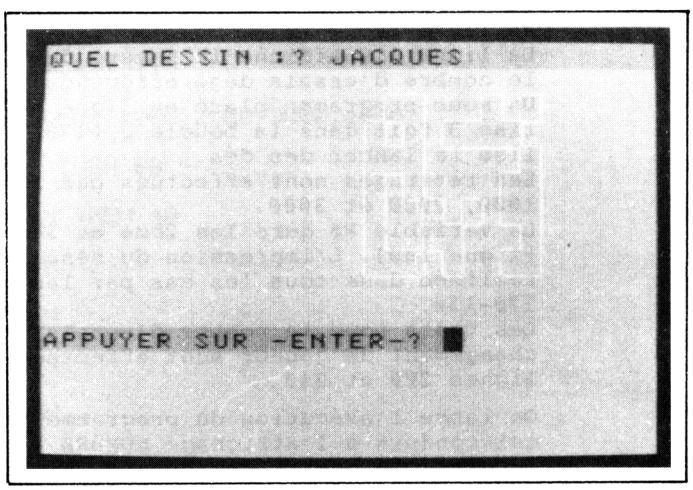
***LE JEU*** : Alice vous permet, pour la première fois, d'entrer dans son intimité en vous laissant entrevoir sa propre représentation du monde ! Vous allez, en effet, lui proposer des mots, n'importe lesquels, et il va vous répondre en vous proposant une représentation graphique de ce que représente 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 Alice dès que l'on aura frappé ENTER.
Pour recommencer, il suffira de taper sur ENTER, et pour finir sur BREAK.
Je vous laisse savourer le dessin qu'a réalisé Alice 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 Alice deviendra de plus en plus savant. On peut aussi réaliser un jeu à deux joueurs, dans lequel Alice vous demande de trouver le mot qui correspond à un certain dessin; etc.

```
10 REM VISIONS
20 CLS 6
30 PRINT@0,"QUEL DESSIN :";:INPUTR$
40 FOR I=1 TO LEN(R$)
50 N=9*ASC(MID$(R$,I,1))-340
60 J=1
70 S=N/2-INT(N/2)
80 N=INT(N/2)
90 IF S<>0 THEN PRINT@(32*I+J+48),"4";:REM COULEUR 4 ; SHIFT Q
100 J=J+1
110 IF J<10 THEN 70
120 NEXT I
130 PRINT@352,"APPUYER SUR -ENTER-";:INPUTR$
140 GOTO 20
```



# 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.

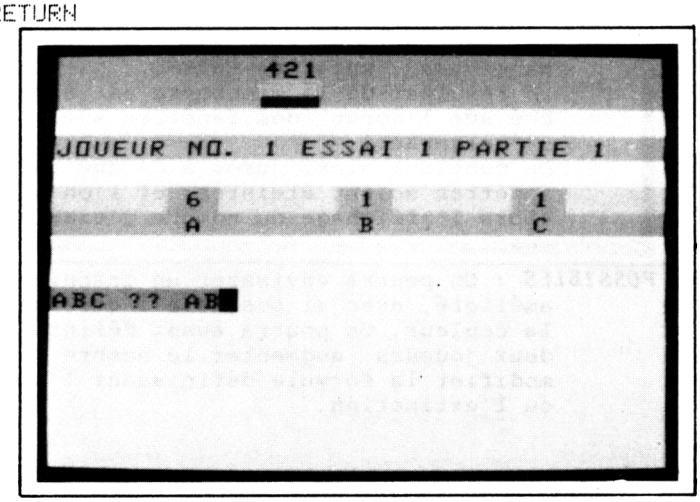
***LE PROGRAMME*** : Le nombre de joueurs est tout d'abord entré en ligne 2Ø.
La ligne 11Ø affiche le numéro de joueur, et le nombre d'essais déjà effectués.
Un sous-programme placé en ligne 4ØØØ, utilisé 3 fois dans la boucle J (17Ø-21Ø) réalise le lancer des dés.
Les retirages sont effectués par les lignes 1ØØØ, 2ØØØ et 3ØØØ.
La variable R$ gère les 2ème et 3ème tirages (ligne 16Ø). L'impression du résultat est réalisée dans tous les cas par les lignes 12Ø-13Ø.
Les tests pour le nombre d'essais et pour le changement de joueur sont effectués aux lignes 22Ø et 24Ø.

***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 ENTER. 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 ENTER. 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 ENTER : par exemple, pour retirer A et C on tapera AC suivi de ENTER.
Les joueurs suivants procèderont 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 récapitulant les

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

```
10 REM 421
20 INPUT"NOMBRE DE JOUEURS ";N
25 P=1
30 I=1
40 K=0
50 CLS2
60 R$="ABC"
70 GOTO 170
80 K=K+1
90 PRINT"            421"
100 PRINT"            444":REM COULEUR 4 ; 12 ESPACES , 3 FOIS
                                                    SHIFT R
110 PRINT@96,"JOUEUR NO."I"ESSAI"K"PARTIE"P
120 PRINT@160,"888888";A;"888888";B;"888888";C;"888888":
                                    REM COULEUR 8 ; 24 FOIS
SHIFT Q
130 PRINT@192,"        A         B         C"
135 R$=""
140 PRINT@288,"ABC ?";:INPUTR$
150 IF R$="" THEN 230
160 R$=R$+"DDD"
170 FOR J=1 TO 3
180 GOSUB 4000
185 X$=MID$(R$,J,1)
190 X=ASC(X$)-64
200 ON X GOSUB 1000,2000,3000,4000
210 NEXT J
220 IF K<3 THEN 80
230 I=I+1
240 IF I<=N THEN 40
245 P=P+1
250 GOTO 30
1000 A=D
1010 RETURN
2000 B=D
2010 RETURN
3000 C=D
3010 RETURN
4000 D=RND(6)
4010 RETURN
```



# L'électricien fou

***REFERENCES*** : Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez difficile.

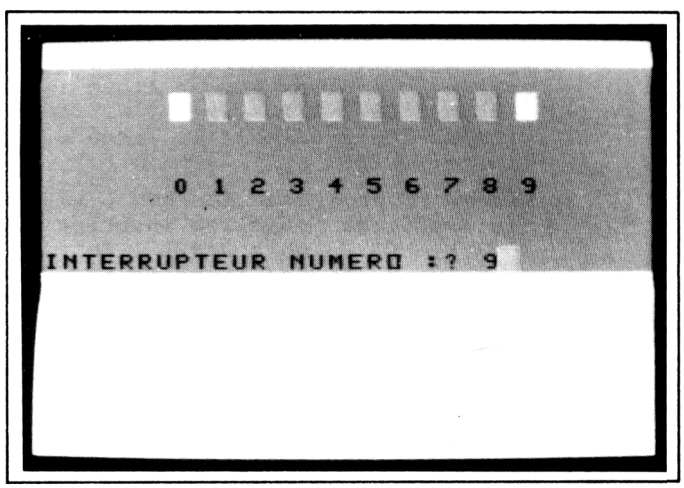
***LE JEU*** : Un immeuble comportant 10 fenêtres, numérotées de Ø à 9, apparaît sur l'écran. Le joueur devra à chaque essai manoeuvrer l'un des interrupteurs, en indiquant son numéro (de Ø à 9).
En principe, la fenêtre correspondante devrait s'éteindre, mais il est aussi possible qu'une autre s'allume (ou s'éteigne). Il s'agit, bien sûr, d'essayer d'obtenir l'extinction des 10 fenêtres, avec le moins d'essais possibles.

***LE PROGRAMME*** : Une table A, en ligne 3Ø, contiendra l'état (allumé ou éteint) des 10 fenêtres.
La boucle I (5Ø-7Ø) et l'autre boucle I (1ØØ-16Ø) dessinent l'immeuble dans son état initial.
La ligne 17Ø regarde si tout est éteint, auquel cas l'on ira en ligne 31Ø afficher le nombre d'essais.
La ligne 18Ø attend l'indication de l'interrupteur manoeuvré.
Les lignes 19Ø à 29Ø définissent le résultat de la manoeuvre.

***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 manoeuvrer, suivi de ENTER.
Le résultat de la manoeuvre 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 M=0
30 DIM A(10)
40 CLS2
50 FOR I=1 TO 7
60 PRINT@32*I,""
70 NEXT I
90 T=0
100 FOR I=1 TO 10
110 T=T+A(I)
120 PRINT@(69+2*I),"3";:REM COULEUR 3 ; SHIFT Q
130 IF A(I)=0 THEN PRINT@(69+2*I),"2";:REM COULEUR 2 ; SHIFT Q
150 PRINT@(160+2*I+5),RIGHT$(STR$(I-1),1)
160 NEXT I
170 IF T=10 THEN 310
180 PRINT@256,"INTERRUPTEUR NUMERO :";:INPUTN
190 N=N+1
200 M=M+1
210 IF A(N)=1 THEN 290
220 A(N)=1
230 R=TAN(RND(0)+N/RND(0)-N)-SIN(RND(0)/N)+336*SIN(8*N)
240 N=INT(10*(R-INT(R)))
250 IF N=0 THEN N=RND(10)
260 IF A(N)=1 THEN 290
270 A(N)=1
280 GOTO 90
290 A(N)=0
300 GOTO 230
310 PRINT@384,"TROUVE EN"M"ESSAIS"
```



# La course au 20

***REFERENCES*** : Difficulté de programmation : facile.
Catégorie de jeu : logique (deux joueurs, ou plus).
Difficulté du jeu : moyenne.

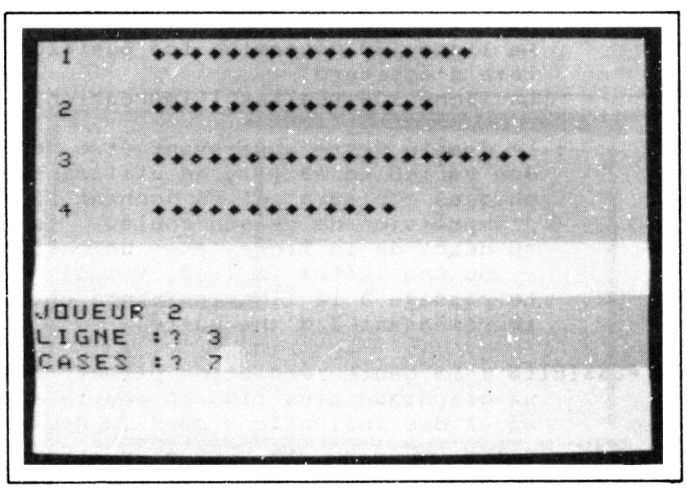
***LE JEU*** : Il est possible d'imaginer pour ce jeu diverses règles, et nous n'avons retenu ici que la règle usuelle pour deux joueurs, qui en fait un jeu apparenté aux jeux de déplacement, ou d'allumettes : 4 tas de 20 étoiles vont apparaître en début de partie, et chaque joueur peut prendre autant 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 (3Ø-6Ø) dessine 4 rangées de 20 étoiles.
La variable F, dimensionnée en ligne 8Ø, servira à enregistrer le nombre d'étoiles prises dans chaque rangée.
La variable T contiendra le nombre total d'étoiles enlevées (ligne 7Ø).
La ligne 12Ø va contrôler qu'il reste suffisamment d'étoiles sur la ligne indiquée.
Les boucles I (15Ø-19Ø) et J (16Ø-18Ø) suppriment les étoiles ; la ligne 22Ø 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 ENTER, et le nombre d'étoiles à enlever suivi de ENTER. Les étoiles sont alors supprimées, et c'est le 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 de seulement un nombre impair d'étoiles, 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 8
30 FOR I=1 TO 4
40 PRINTI;"   ********************"
50 PRINT
60 NEXT I
70 T=0
80 DIM F(4)
90 B=1
100 PRINT@320,"JOUEUR"B
110 INPUT"LIGNE :";L
112 IF L<>INT(L) OR L=0 OR L>4 THEN 100
115 INPUT"CASES :";C
120 IF C=0 OR C+F(L)>20 THEN 100
130 F(L)=F(L)+C
140 T=T+C
150 FOR I=1 TO 4
160 FOR J=0 TO F(I)
170 PRINT@(64*I-38-J)," "
180 NEXT J
190 NEXT I
200 B=B+1
210 IF B=3 THEN B=1
220 IF T<80 THEN 100
230 PRINT@352,"LE JOUEUR"B"A GAGNE"
```



# Dessin

**REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : dessin.
Difficulté du jeu : moyenne.

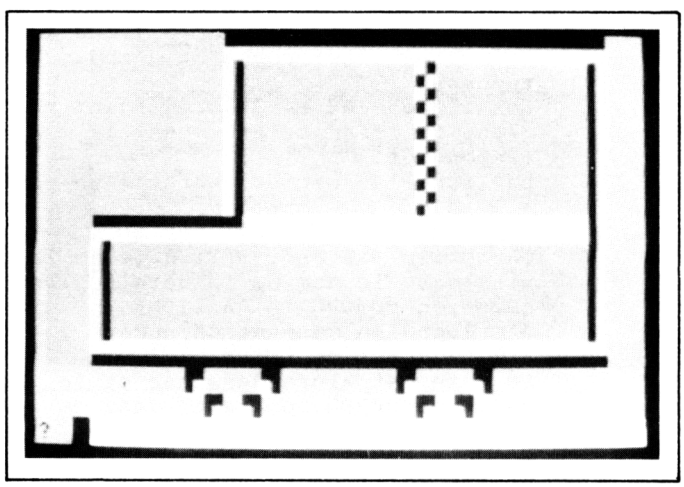
**LE JEU** : C'est surtout une méthode un peu originale de codage d'un dessin, qui pourra être avantageusement utilisée dans certains programmes.

**LE PROGRAMME** : Une table A$ est dimensionnée à 16 en ligne 30, pour contenir chaque ligne de dessin (il y a 16 lignes d'affichage). Le codage du dessin est effectué dans les lignes concernées, en remplissant chacune des variables A$ (I), selon le code retenu (voir plus loin).
Les lignes 40 à 120 contiennent le dessin codé.
La boucle I (130-200) assure le passage d'une ligne à la suivante.
La boucle N (140-180) réalise le décodage de A$ (I).
La ligne 160 détermine les positions de la tête d'écriture.
La ligne 170 réalise l'impression.

**POUR JOUER** : Le dessin devra auparavant être défini dans des variables A$ (I), en utilisant les graphiques couleurs, et en donnant la position d'impression de chaque couleur, par rapport au début de la ligne, avec un chiffre de 1 à 9, ou une lettre (A : 10, B : 11, C : 12, etc.)
Le passage à la ligne suivante est fait en incrémentant I d'une unité.

**EXTENSIONS POSSIBLES** : La haute résolution permettra de définir un graphique plus fin, on pourra aussi envisager des indications pour la couleur, imaginer aussi un jeu dans lequel chaque joueur devrait essayer de coder un dessin donné le plus rapidement possible, etc.

```
10 REM DESSIN
20 CLS 2
30 DIM A$(16)
40 A$(1)="A4444444444444444444":REM COULEUR 4 ; FAIRE A ,
                                    PUIS 20 FOIS SHIFT Q
50 A$(2)="A4K4T4":REM COULEUR 4 ; FAIRE A , SHIFT Y , PUIS K ,
                                   SHIFT A , PUIS T    SHIFT G
60 FOR I=3TO7:A$(I)=A$(2):NEXT I
70 A$(8)="38888888888888888888888888888":REM COULEUR 8 ;
   FAIRE 3 , PUIS 8 FOIS SHIFT X , 18 FOIS SHIFT Q , ET SHIFT G
80 A$(9)="34T4":REM COULEUR 4 ; FAIRE 3 , PUIS SHIFT Y ,
                                     PUIS T , PUIS SHIFT G
90 A$(10)=A$(9):A$(11)=A$(9):A$(12)=A$(9)
100 A$(13)="3444444444444444444444444444":REM COULEUR 4 ;
                          FAIRE 3 , PUIS 27 FOIS SHIFT R
110 A$(14)="84C4J4N4":REM COULEUR 4 ; FAIRE 8 , PUIS SHIFT V ,
    PUIS C , PUIS SHIFT C , PUIS J , PUIS SHIFT V , PUIS N ET
    SHIFT C
120 A$(15)="94B4K4M4":REM COULEUR 4;FAIRE 9 , PUIS SHIFT V ,
    PUIS B , PUIS SHIFT C , PUIS K , PUIS SHIFT V , PUIS M ET
    SHIFT C
130 FOR I=1 TO 15
140 FOR N=1 TO LEN(A$(I))
145 IF A$(I)="" THEN 200
150 P=ASC(MID$(A$(I),N,1))
160 IF P<91 THEN PRINT TAB(P-48+7*(P>64));:GOTO 180
170 PRINTMID$(A$(I),N,1);
180 NEXT N
190 PRINT
200 NEXT I
210 INPUT"";A
```



# Musique

**REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : jeu éducatif.
Difficulté du jeu : moyenne.

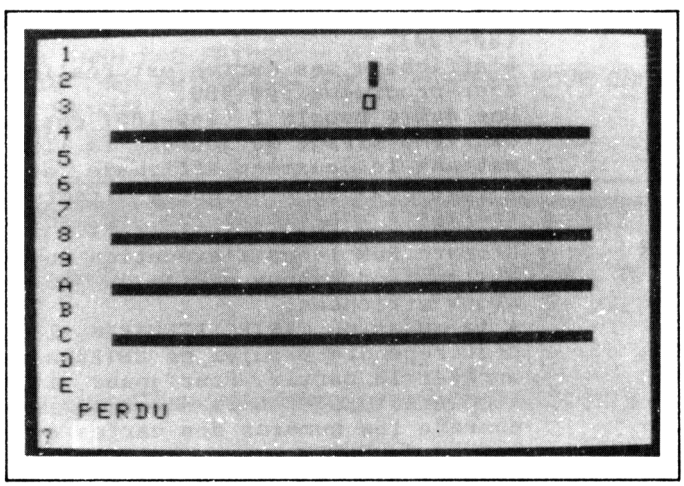
**LE JEU** : Il se présente comme un exercice d'apprentissage du solfège : l'appareil annonce une note (en clé de sol), suivie de la mention GRAVE ou AIGUE,selon sa place sur la portée.
Il faudra répondre en indiquant sur quelle ligne la note sera affichée.
La note est alors dessinée sur la portée, suivie d'une message BRAVO ou PERDU, selon que l'on avait trouvé juste ou non.

**LE PROGRAMME** : Une première boucle I (2Ø-27) affiche les numéros des lignes.
La boucle I (3Ø-5Ø) dessine la portée.
La ligne 6Ø définit le nom des notes , et la ligne 7Ø réalise le tirage aléatoire d'une note.
Le nom de la note est calculé à la ligne 8Ø, et il est affiché par la ligne 9Ø suivi de la mention GRAVE ou AIGUE, selon le cas.
La réponse sera entrée en ligne 12Ø.
La ligne 13Ø s'occupe du décodage de la réponse.
Les lignes 14Ø et 145 regardent si l'on a gagné ou perdu et affichent le message correspondant.
La ligne 17Ø 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 ENTER.
La note est alors dessinée et le message BRAVO ou PERDU apparaît, dans ce cas, à l'écran.
On doit taper la touche ENTER pour continuer, et s'arrêter en tapant BREAK.

**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
15 CLS
20 FOR I=1 TO 14
22 IF I<=9 THEN PRINT I
25 IF I>9 THEN PRINT " ";CHR$(I+55)
27 NEXT I
30 FOR I=1 TO 5
40 PRINT@(64*I+36),"000000000000000000000000000":REM COULEUR 0 ;
   FAIRE 25 FOIS SHIFT R
50 NEXT I
60 N$="SI LA SOLFA MI RE DO "
70 X=RND(13)+1
80 T=X-1-7*INT((X-1)/7)
90 PRINT@450,"NOTE ";MID$(N$,3*T+1,3);
100 IF X<=7 THEN PRINT " AIGUE";
110 IF X>7 THEN PRINT " GRAVE";
120 INPUT"-NO. LIGNE :";L$
130 L=-(ASC(L$)-48)*(L$<":")-(ASC(L$)-55)*(L$>="A")
140 PRINT@(32*X-47),"0";:REM COULEUR 0 ; SHIFT Y
145 PRINT@(32*X-15),"O";
150 IF L=X THEN PRINT@450,"BRAVO";
160 IF L<>X THEN PRINT@450,"PERDU";
170 PRINT:INPUTA$:RUN
```



# Poker

***REFERENCES*** : Difficulté de programmation : difficile.
Catégorie de jeu : jeu contre Alice.
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 Alice.
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 d'Alice de jouer.

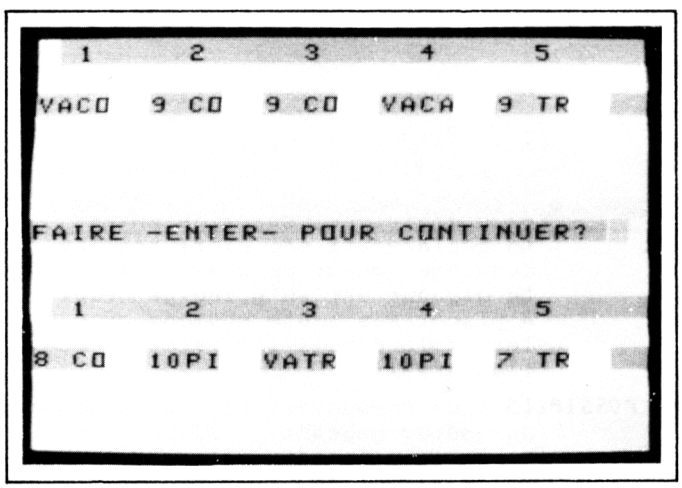
***LE PROGRAMME*** : Les lignes 3Ø et 4Ø définissent les cartes, Une variable N est dimensionnée à 16, en ligne 5Ø pour recevoir 10 cartes joueur et 5 cartes Alice.
Le tirage sera effectué par une boucle I (6Ø-11Ø) et le contrôle par une boucle J (8Ø-1ØØ).
L'affichage des cartes est réalisé par le sous-programme 5ØØ-58Ø.
Une autre boucle I (16Ø-18Ø) réalise le deuxième tirage du joueur, la ligne 2ØØ permettant le deuxième affichage, et la boucle I (5ØØ-55Ø) 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 Ø suivi de ENTER s'il veut arrêter la partie. Pour jouer, il répondra simplement ENTER s'il s'estime servi, ou il entrera les numéros des cartes qu'il refuse suivis de ENTER.
Par exemple, s'il tape 235 et ENTER, son nouveau jeu conservera les cartes 1 et 4. Son jeu sera alors affiché ainsi que celui d'Alice, ce dernier n'ayant pas, dans ce programme, la possibilité de reprendre des cartes.
Il suffira ensuite de faire ENTER 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 d'Alice. L'introduction des règles sur les paris et même pourquoi pas, des possibilités de bluff.

```
10 REM POKER
20 CLS 8
30 V$="7 8 9 10VADAROAS"
40 C$="TRCACOPI"
50 DIM N(16)
60 FOR I=1 TO 16
70 N(I)=RND(32)-1
75 IF I=1 THEN 110
80 FOR J=1 TO I-1
90 IF N(I)=N(J) THEN 70
100 NEXT J
110 NEXT I
120 K=0:P=0
130 GOSUB 500
140 PRINT@224,"CARTES REFUSEES :";
145 INPUTR$
150 IF R$="0" THEN END
160 FOR I=1 TO LEN(R$)
170 N(VAL(MID$(R$,I,1)))=N(I+5)
180 NEXT I
190 CLS 8
195 P=1
200 GOSUB 500
210 K=K+10
500 FOR I=1 TO 5
510 J=2*(N(I+K)-4*INT(N(I+K)/4)+1)-1
520 F=2*(N(I+K)-7*INT(N(I+K)/7)+1)-1
525 R$=MID$(V$,F,2)
530 PRINT@(32*K+6*I-5),I
540 PRINT@(32*K+6*I+58),R$;MID$(C$,J,2);"--"
550 NEXT I
560 IF P=1 THEN P=0:RETURN
570 IF K=10 THEN PRINT@224,"FAIRE -ENTER- POUR CONTINUER";:
    INPUT A$:RUN
580 RETURN
```



# Le jeu de saute-mouton

***REFERENCES*** : Difficulté de programmation : facile.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : facile.

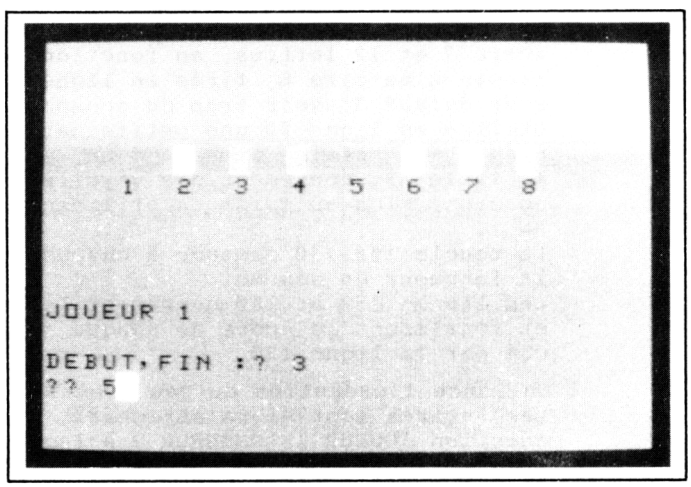
***LE JEU*** : Bien que l'idée en soit très simple, il ne sera pas forcément évident de découvrir une stratégie convenable. On dispose de 3 moutons noirs et 3 moutons gris, représentés par des carrés, qui occupent 8 cases ; les moutons peuvent se déplacer d'une case à une case libre. Plusieurs règles 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 (5Ø-8Ø) initialise une table A (I) qui contient les positions des moutons.
La boucle I (1ØØ-15Ø) dessine le terrain de jeu et les moutons.
Les lignes 16Ø et 17Ø s'occupent du texte et des entrées.
La ligne 18Ø contrôle la validité du coup.
Le changement de joueur est réalisé en ligne 21Ø.

***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 ENTER, puis le numéro de la case d'arrivée, suivi de ENTER.
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
95 PRINT@128,""
100 FOR I=1 TO 8
110 PRINT@(160+3*I),I
130 IF A(I)=1 THEN PRINT@(129+3*I),"4";:REM COULEUR 4 ; SHIFT 0
140 IF A(I)=-1 THEN PRINT@(129+3*I),"8";:REM COULEUR 8 ; SHIFT 0
150 NEXT I
155 J=1-(H=-1)
160 PRINT@320,"JOUEUR"J
170 PRINT@384,"DEBUT,FIN :";:INPUTD,F
180 IF A(D)<>H OR A(F)<>0 THEN 160
190 A(D)=0
200 A(F)=H
210 H=-H
220 GOTO 95
```



# Lettres

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

***LE JEU*** : La règle est fortement inspiré d'un jeu de lettres très connu en France, pratiqué notamment dans une émission télévisée (les chiffres et les lettres).
Alice 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.

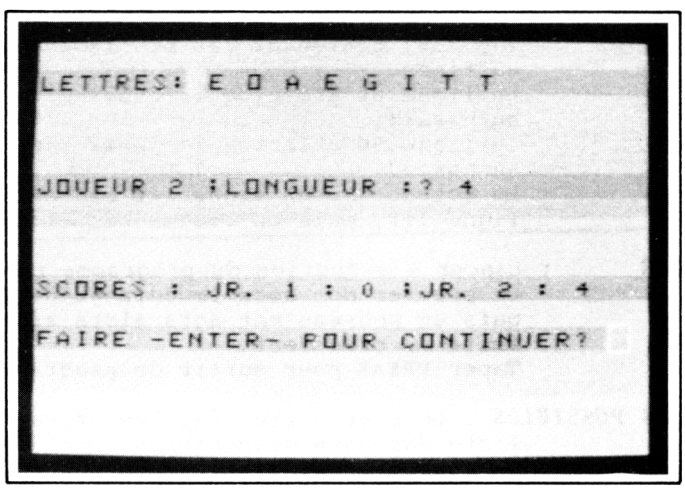
***LE PROGRAMME*** : La boucle I (6Ø-9Ø) choisit dans l'alphabet entre 7 et 12 lettres, en fonction de la variable aléatoire L, tirée en ligne 30.
Pour éviter d'avoir trop de consonnes, on utilise en ligne 8Ø une petite astuce, qui consiste à refaire au hasard un tirage dans la table A$, contenant des voyelles (on peut en avoir plus ou moins en utilisant d'autres RND).
La boucle 1ØØ-14Ø demande à chacun des joueurs la longueur de son mot.
Les lignes 15Ø et 16Ø comparent les résultats et totalisent le score de chaque joueur, affiché par la ligne 17Ø.

***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 (la correction du mot devra toutefois être contrôlée par l'autre joueur) ; 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 la touche ENTER.
On devra faire BREAK 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 Alice, 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 prises au hasard !

```
10 REM LETTRES
20 S=0:J=0
25 CLS6
30 L=RND(6)+6
40 A$="AEIOUEA"
50 PRINT@32,"LETTRES:";
60 FOR I=1 TO L
70 PRINT@(32+2*I+7),CHR$(RND(26)+64)
80 IF RND(0)<0.4 THEN PRINT@(32+2*I+7),MID$(A$,RND(7),1)
90 NEXT I
100 I=1
110 PRINT@160,"JOUEUR"I;:INPUT";LONGUEUR :";L
120 IF I=1 THEN P=L
130 I=I+1
140 IF I<3 THEN 110
150 S=S+(P AND P>L)
160 J=J+(L AND L>P)
170 PRINT@288,"SCORES : JR. 1 :"S";JR. 2 :"J
180 PRINT@352,"FAIRE -ENTER- POUR CONTINUER";:INPUT A$
190 GOTO 25
```



# Machinamot

***REFERENCES*** : Difficulté de programmation : facile.
Catégorie de jeu : jeu de hasard (sur les mots).
Difficulté du jeu : aucune.

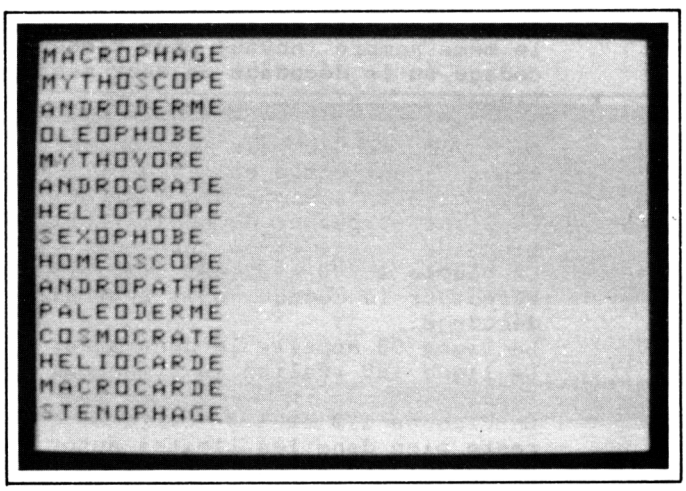
***LE JEU*** : Ce très court programme permet à Alice de faire la démonstration de ses capacités à manipuler des chaînes de caractères : on part simplement d'un préfixe et d'un suffixe d'origine grecque, que l'on va assembler au hasard, on tombera bien sûr, de temps à autre, sur des mots absurdes ou 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 dimensionnés à 15 en ligne 30 et 40 : 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 (50-70) lit les lignes DATA (160-170-180) contenant les préfixes suivis des suffixes.
La ligne 80 efefctue le choix d'un préfixe au hasard.
La ligne 90 effectue le choix d'un suffixe au hasard.
La mot obtenu en ligne 100 est affiché par la ligne 110, 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 alors affiché à la suite du précédent.
Taper BREAK 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 Alice.

```
10 REM MACHINAMOT
20 CLS 3
30 DIM A$(15)
40 DIM B$(15)
50 FOR I=1 TO 15
60 READ A$(I),B$(I)
70 NEXT I
80 I=RND(15)
90 J=RND(15)
100 C$=A$(I)+B$(J)
110 PRINT C$
120 FOR S=1 TO 1200:NEXT S
150 GOTO 80
160 DATA ANDRO,CRATE,COSMO,PHILE,MICRO,PHOBE,HELIO,TROPE,SEXO,
        GYNE,STEREO,MANE
170 DATA PALEO,NAUTE,MACRO,PHAGE,HYDRO,FUGE,STENO,PATHE,PATHO,
        CARDE,OLEO,DERME
180 DATA MYTHO,SCOPE,HOMEO,VORE,OCTO,PHONE
```



# Espion

***REFERENCES*** : Difficulté de programmation : facile.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : facile.

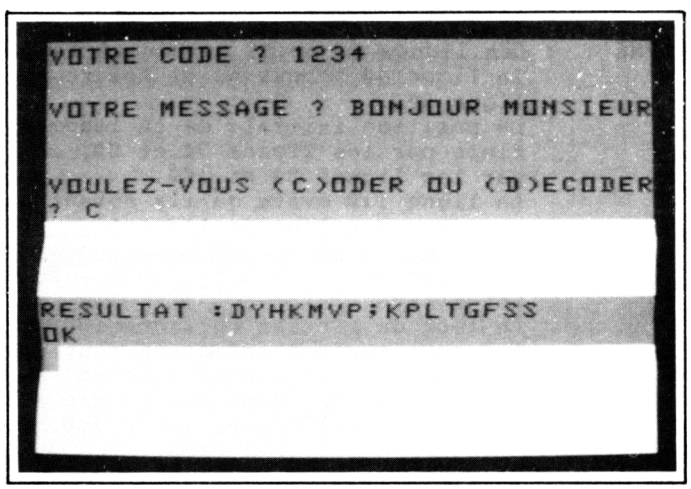
***LE JEU*** : Le programme peut servir simultanément de codeur et de décodeur de message secret, suivant une méthode très facile à mettre en oeuvre sur ce type d'appareil.
Un petit générateur aléatoire situé en fin de programme est utilisé pour modifier de façon arbitraire chaque lettre du message ; on devra bien sûr, veiller 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 3Ø ; le message est entré à la ligne 5Ø.
La ligne 6Ø permet de choisir entre les options de codage et de décodage.
La boucle I (7Ø-14Ø) est utilisée soit pour effectuer le codage, soit pour effectuer le décodage.
La ligne 9Ø appelle le générateur aléatoire.
La ligne 1ØØ réalise la transposition des lettres.
Les lignes 11Ø et 12Ø vérifient que l'on reste bien dans les limites autorisées.
Les résultats seront imprimés par la ligne 15Ø.
Les lignes 1ØØØ à 1Ø3Ø 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 ENTER.
Choisir l'option C pour CODER, et l'option D pour DECODER, et taper ENTER.
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 5
30 INPUT"VOTRE CODE ";N
40 A=1/N
45 PRINT
50 INPUT"VOTRE MESSAGE ";A$
55 PRINT
60 INPUT"VOULEZ-VOUS (C)ODER OU (D)ECODER";R$
65 B$=""
70 FOR I=1 TO LEN(A$)
80 T=ASC(MID$(A$,I,1))
90 GOSUB 1000
100 T=T-A*((R$="C")-(R$="D"))
110 IF T>90 THEN T=T-26
120 IF T<65 THEN T=T+26
130 B$=B$+CHR$(T)
140 NEXT I
150 PRINT@320,"RESULTAT :";B$:END
1000 A=(A+3.14159)^8
1010 A=A-INT(A)
1020 A=INT(26*A+1)
1030 RETURN
```



# 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.

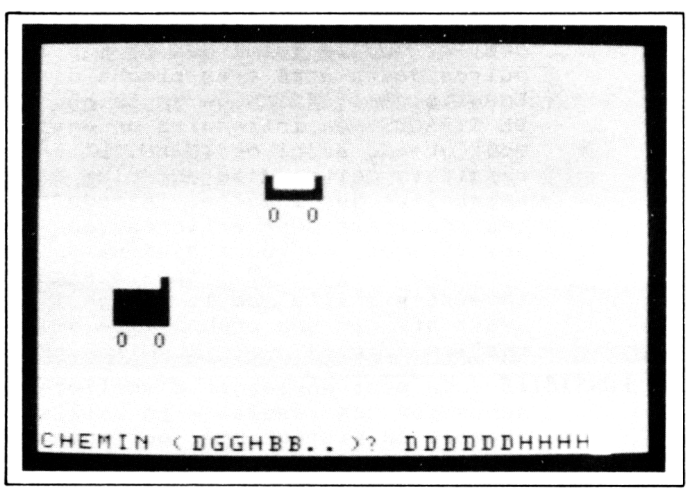
*LE PROGRAMME* : Les lignes 2Ø et 3Ø dessinent la locomotive, la ligne 4Ø le wagon, et les roues sont en ligne 5Ø.
La position initiale de la locomotive est définie par les lignes 7Ø et 8Ø, celle du wagon par les lignes 9Ø et 1ØØ.
La ligne 11Ø évite qu'ils soient initialement trop proches l'un de l'autre.
Les lignes 13Ø et 14Ø réalisent l'affichage du wagon et les lignes 15Ø à 17Ø celui de la locomotive.
Le test de fin est en ligne 18Ø.
Les mouvements à effectuer sont entrés dans une variable alphanumérique R$, en ligne 19Ø, et sont décodés dans une boucle I (22Ø-24Ø), pour être exécutés en ligne 25Ø.

*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 DDDHHHBHGG, suivi de ENTER.
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 D$="000":REM COULEUR 0 ; SHIFT R ,PUIS SHIFT R ET SHIFT S
30 A$="000" : REM COULEUR 0 ; 3 FOIS SHIFT B
40 B$="888" : REM COULEUR 8 ; SHIFT Z , PUIS SHIFT R ET SHIFT S
50 C$="0 0"
60 N=0
70 LY=RND(25)
80 LX=RND(13)+1
90 WY=RND(25)
100 WX=RND(15)
110 IF ABS(LX-WX)<4 THEN 70
120 CLS
130 PRINT@(32*WX+WY),B$
140 PRINT@(32*(WX+1)+WY),C$
150 PRINT@(32*(LX-1)+LY),D$
160 PRINT@(32*LX+LY),A$
170 PRINT@(32*(LX+1)+LY),C$
180 IF LX=WX AND WY=LY-3 THEN 300
190 PRINT@480,"CHEMIN (DGGHBB..)";:INPUTR$
200 X=0:Y=0
210 N=N+1
220 FOR I=1 TO LEN(R$)
230 X=X-(MID$(R$,I,1)="B"AND LX<14)+(MID$(R$,I,1)="H"AND LX>1)
235 Y=Y-(MID$(R$,I,1)="D"AND LY<28)+(MID$(R$,I,1)="G"AND LY>0)
240 NEXT I
250 LX=LX+X:LY=LY+Y
260 GOTO 120
300 PRINT"BRAVO; TROUVE EN"N"COUPS"
310 SOUND180,9:SOUND195,7:SOUND160,9
```



# Tirage d'un dé truqué

*REFERENCES* : Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : moyenne.

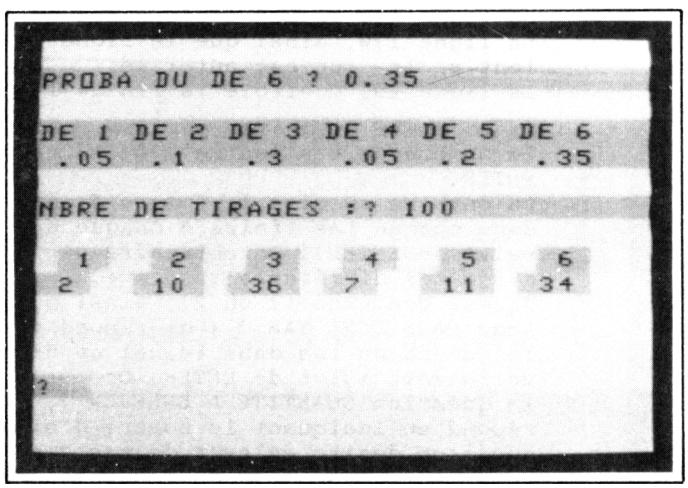
*LE JEU* : Il reprend une idée d'un programme précédent (DE TRUQUE) dans une application statistique : on fabrique un dé volontairement truqué, que l'on va lancer un certain nombre de fois, et l'on dresse un tableau des résultats.

*LE PROGRAMME* : Les probabilités désirées pour chaque tirage sont introduites par une boucle I (4Ø-1ØØ), avec toutes les vérifications de validité nécessaires. Le nombre de tirages demandés est introduit en ligne 115.
La boucle J (12Ø-18Ø) comptabilise les tirages.
La boucle I (14Ø-16Ø) assure la conformité de chaque tirage avec les probabilités introduites en début de jeu.
Les résultats sont comptabilisés en ligne 17Ø, et affichés grâce aux lignes 175 et 178.

*POUR JOUER* : Commencer l'exécution en frappant RUN, puis, à la question PROBABILITE DU DE 1, introduire un nombre compris entre Ø et 1.
Faire de même pour les dés 2, 3, etc., sans oublier que le total des probabilités introduites devra être très proche de 1.
Ensuite, à l'affichage de la question NOMBRE DE TIRAGES, on introduira un entier suffisamment grand, suivi de ENTER. 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, qui apparaissent en caractères inversés : ainsi 32.3 apparaissant sous le 1 inversé signifie que le tirage 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
15 CLS 8
20 P=0
30 DIM E(6)
40 FOR I=1 TO 6
50 PRINT@32,"PROBA DU DE"I;:INPUTE(I)
60 P=P+E(I)
70 IF E(I)<0 OR E(I)>1 OR P>1.1 THEN 200
80 PRINT@(96+5*I-5),"DE"I
90 PRINT@(128+5*I-5),E(I)
100 NEXT I
110 IF P<0.9 THEN RUN
115 PRINT@192,"NBRE DE TIRAGES :";:INPUTP
120 FOR J=1 TO P
130 N=RND(6)
140 FOR I=1 TO 6
150 IF N=I AND RND(0)>E(I)-INT(E(I)) THEN 130
160 NEXT I
170 E(N)=E(N)+1
175 PRINT@(256+5*N-4),N;
178 PRINT@(288+5*N-5),INT(E(N));
180 NEXT J
190 PRINT@416,"";:INPUTA$:RUN
200 PRINT@32,"IMPOSSIBLE :0<=P<=1"
210 P=P-E(I)
220 FOR J=1 TO 1700:NEXT J
240 GOTO 50
```



# Le jeu de Marienbad

**REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : logique (2 joueurs).
Difficulté du jeu : moyenne.

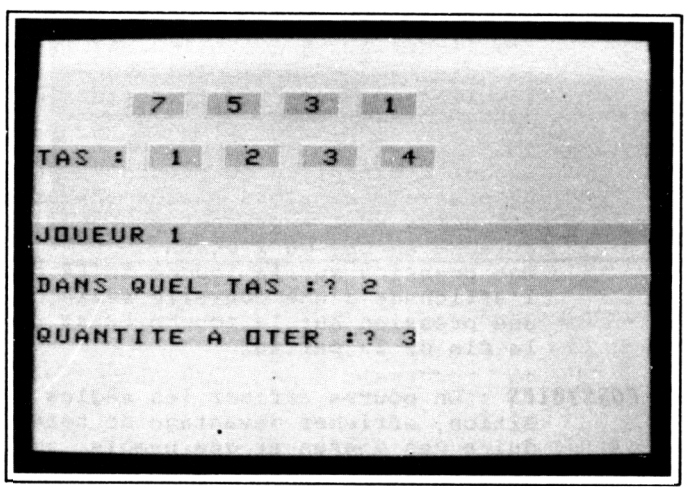
**LE JEU** : Un jeu rendu très célèbre grâce à un film qui le popularisa, présenté toutefois ici dans une version plus conforme à ses origines, à savoir le jeu de NIM.
En effet, il est possible, en début de partie, de choisir le nombre de tas que l'on veut faire et le nombre d'allumettes que l'on veut mettre dans chaque tas.
Chaque joueur joue à son tour, et celui qui retire la dernière allumette a perdu.

**LE PROGRAMME** : La ligne 5Ø dimensionne une variable A (I) qui contiendra le nombre d'allumettes de chaque tas.
La boucle I (6Ø-9Ø) permet d'entrer ce nombre.
La boucle I (11Ø-14Ø) imprime les tas.
La ligne 15Ø assure l'affichage du joueur.
La ligne 17Ø, ainsi que le ligne 18Ø, contrôlent si le coup est autorisé.
La ligne 25Ø contrôle la fin de la partie.

**POUR JOUER** : On lance le programme par RUN, puis on entre le nombre de tas (entre 1 et 8), suivi de ENTER.
On entrera ensuite le nombre d'allumettes dans chaque tas (faire à chaque entrée N suivi de ENTER). 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 ENTER. 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 Alice de servir de partenaire, etc.

```
10 REM LE JEU DE MARIENBAD
15 CLS2
20 INPUT"NOMBRE DE TAS  <= 7";N
30 IF N>=8 THEN RUN
40 S=0
50 DIM A(N)
60 FOR I=1 TO N
70 PRINT@96,"TAS"I;:INPUT"NBRE D'ALLUMETTES :";A(I)
80 S=S+A(I)
90 NEXT I
95 H=1
100 CLS4
110 FOR I=1 TO N
120 PRINT@(65+4*I),A(I);
130 PRINT@128,"TAS :";:PRINT@(128+4*I+2),I;
140 NEXT I
150 PRINT@224,"JOUEUR"H
160 PRINT@288,"DANS QUEL TAS :";:INPUTT
170 IF T>N THEN 100
175 IF A(T)=0 THEN 100
180 PRINT@352,"QUANTITE A OTER :";:INPUTO
190 IF O>A(T) OR O=0 OR O<>INT(O) THEN 180
210 A(T)=A(T)-O
220 S=S-O
230 H=H+1
240 IF H=3 THEN H=1
250 IF S=0 THEN PRINT@384,"LE JOUEUR"H"A GAGNE":SOUND190,9:
    SOUND210,7:SOUND160,9:END
260 GOTO 100
```



# Compose

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

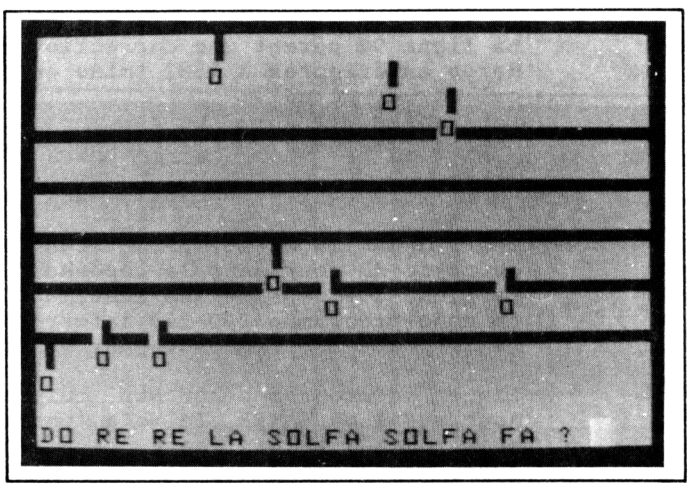
***LE JEU*** : Il s'agit, en effet, d'un programme capable de composer, de façon très élémentaire, de la musique, et ceci d'une façon complètement aléatoire. L'ordinateur va, dans un premier temps, dessiner une portée, puis il affiche une succession de notes (allant du DO grave au SI aigu) et donne leur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

***LE PROGRAMME*** : La boucle I (3Ø-5Ø) dessine la portée.
La variable N$, en ligne 6Ø contient les noms des notes.
La deuxième boucle I (8Ø-13Ø) permet de faire le choix d'une note de façon aléatoire, à la ligne 9Ø ; son nom est calculé en ligne 1ØØ et la note est dessinée par les lignes 11Ø et 115.
Les lignes 123 à 125 calculent la hauteur de la note, jouée en ligne 128.
Les lignes 14Ø et 15Ø permettent de recommencer en tapant ENTER.

***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 la touche ENTER provoquera l'affichage d'une nouvelle série de notes, et une pression sur la touche BREAK 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.

```
10 REM COMPOSE
20 CLS
30 FOR I=1 TO 5
40 PRINT@(64*I+32),"000000000000000000000000000000000":REM
   COULEUR 0 ; 32 FOIS SHIFT R
50 NEXT I
60 N$="SI LA SOLFA MI RE DO "
80 FOR I=1 TO 9
90 X=RND(13)+1
100 D$=MID$(N$,3*(X+7*(X>7))-2,3)
110 PRINT@(32*(X-2)+3*I-3),"0";:REM COULEUR 0 ; SHIFT Y
115 PRINT@(32*(X-1)+3*I-3),"O";
120 PRINT@(480+3*I-3),D$;
123 S=212*(X=2)+206*(X=3)+200*(X=4)+196*(X=5)+189*(X=6)
124 S=S+180*(X=7)+175*(X=8)+165*(X=9)+154*(X=10)+141*(X=11)
125 S=S+135*(X=12)+119*(X=13)+102*(X=14)
128 SOUNDABS(S),9
130 NEXT I
140 INPUTR$
150 RUN
```



# Jour de la semaine

***REFERENCES*** : Difficulté de programmation : moyenne.
Catégorie de jeu : utilitaire.
Difficulté du jeu : facile.

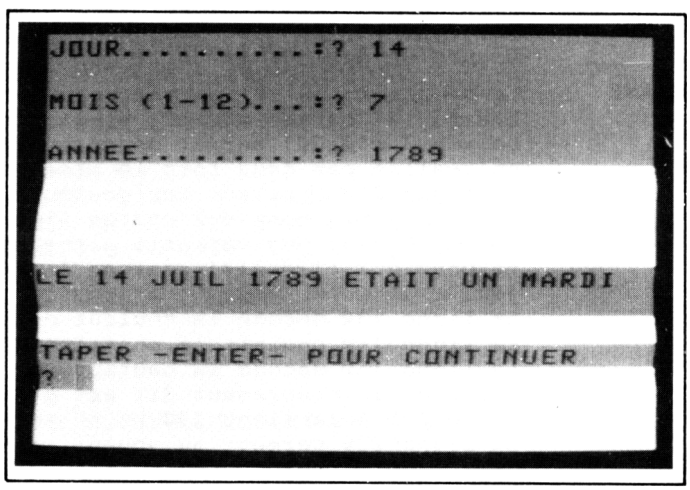
***LE JEU*** : Le programme vous permet de retrouver, sans aucun calcul, le jour de la semaine correspondant à une date donnée.
La date sera entrée sous forme de trois nombres, par exemple 11, 3, 1956 pour le 11 mars 1956, et le résultat sera affiché en clair.

***LE PROGRAMME*** : Les lignes 1Ø à 7Ø contrôlent les entrées.
La ligne 4Ø s'occupe de la fin de partie.
La ligne 9Ø permet une correction pour les dates antérieures à 1582 (mise en service du calendrier actuel).
Les lignes 1ØØ à 11Ø permettront l'affichage en clair du mois dans la réponse (sur 4 lettres).
La formule utilisée apparaît aux lignes 13Ø et 14Ø.
Les lignes 15Ø et 16Ø permettront l'affichage en clair du jour dans la réponse.
La ligne 17Ø imprime les résultats.
Le sous-programme 2ØØ-22Ø intervient pour tenir compte des mois de janvier et février.

***POUR JOUER*** : Lancer le programme apr RUN, puis entrer le jour suivi de ENTER, le mois (entre 1 et 12) suivi de ENTER, et l'année (sous la forme 1845, par exemple) suivie de ENTER.
Le résultat est très rapidement affiché 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 Ø 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 de 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 7
30 INPUT"JOUR.........:";J:PRINT
40 IF J=0 THEN END
50 INPUT"MOIS (1-12)...:";M:PRINT
60 INPUT"ANNEE........:";A
70 B=A
80 S=INT(A/100)-INT(A/400)
90 IF A<=1582 THEN S=2
100 M$="JANVFEVRMARSAVRIMAI JUINJUILAOUTSEPTOCTONOVEDECE"
110 M$=MID$(M$,4*M-3,4)
120 IF M<3 THEN GOSUB 200
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 PRINT@288,"LE"J;M$;B;"ETAIT UN ";J$
175 SOUND190,9
180 PRINT@384,"TAPER -ENTER- POUR CONTINUER":INPUTR$
190 GOTO 20
200 M=M+12
210 A=A-1
220 RETURN
```



# 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 Alice 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 Alice dans quelle colonne (1ère, 2ème, 3ème à partir de la gauche) se trouve la carte choisie ; les cartes sont alors redistribuées et le joueur indique de nouveau la colonne correspondante à sa carte ; ceci est fait à nouveau encore une fois et, au 4ème affichage, Alice est en mesure de deviner la carte choisie.

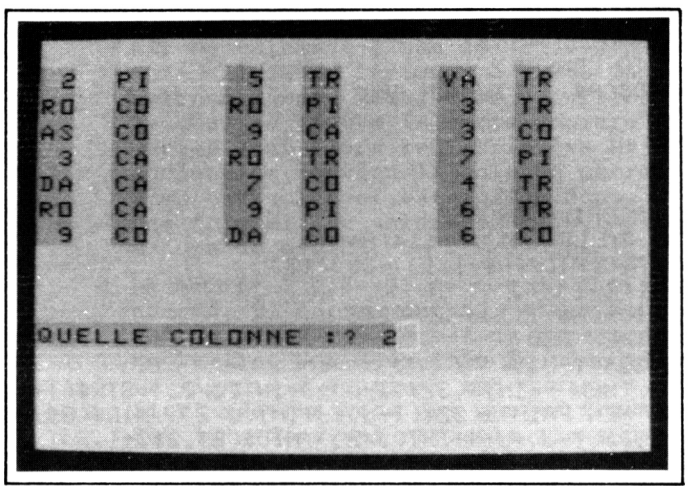
**LE PROGRAMME** : La boucle I (3Ø-9Ø) permet de choisir, au hasard, 21 cartes sur 52, et la boucle emboîtée J (5Ø-7Ø) permet de s'assurer que l'on ne réalise pas deux fois le même tirage.
La ligne 11Ø contient les couleurs des cartes, ainsi que les noms des cartes figures (VA, DA, RO, AS), dans une variable alphanumérique B$.
La boucle I (12Ø-21Ø) réalise l'affichage du jeu.
La ligne 13Ø décode la couleur de la carte, la ligne 14Ø choisit la position d'affichage la ligne 15Ø décode la hauteur.
L'affichage proprement dit est effectué de la ligne 17Ø à la ligne 19Ø.
La ligne 23Ø permet, au joueur, d'entrer le numéro de la colonne où se trouve la carte.
La boucle J (27Ø-29Ø) réalise le brouillage, et les boucles emboîtées J (32Ø-36Ø) et I (33Ø-35Ø) fabriquent la nouvelle variable A$ à utiliser pour le prochain affichage.
Les lignes 38Ø à 43Ø réalisent le tirage et l'affichage de la carte devinée.

**POUR JOUER** : On lance le programme apr 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 ENTER.
On obtient un 2ème affichage et l'on fait de même ; puis un 3ème affichage et l'on fait

de même.
Au 4ème affichage, la carte choisie est automatiquement affichée.
On sortira du programme en tapant BREAK 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
15 CLS 6
20 A$=""
30 FOR I=1 TO 21
40 B$=CHR$(RND(52)+37)
50 FOR J=1 TO LEN(A$)
60 IF B$=MID$(A$,J,1) THEN 40
70 NEXT J
80 A$=A$+B$
90 NEXT I
100 H=3:P=1
110 B$="TRCACOPIVADAROAS"
120 FOR I=1 TO LEN(A$)*P-11*(P=0)
130 S=INT((ASC(MID$(A$,I,1))-38)/13)
140 J=7*INT((I-1)/7)
150 T=ASC(MID$(A$,I,1))-37-13*S
160 IF P=0 THEN 380
165 PRINT@(32*(I-J)),"";
170 IF T<=9 THEN PRINT@(32*(I-J)+J+INT(J/2)),STR$(T+1);
180 IF T>9 THEN PRINT@(32*(I-J)+J+INT(J/2)),MID$(B$,ABS(2*T-11),2);
190 PRINT@(32*(I-J)+J+4+INT(J/2)),MID$(B$,2*S+1,2);
210 NEXT I
230 PRINT@352,"QUELLE COLONNE :";:INPUTN
240 CLS6
250 H=H-1
260 R$="20120":B$=""
270 FOR J=0 TO 2
280 B$=B$+MID$(A$,7*VAL(MID$(R$,N+J,1))+1,7)
290 NEXT J
300 IF H=0 THEN A$=B$:P=0:GOTO 110
310 A$=""
320 FOR J=0 TO 2
330 FOR I=1 TO 7
340 A$=A$+MID$(B$,3*I-2+J,1)
350 NEXT I
360 NEXT J
370 GOTO 110
380 REM CHOIX FINAL
390 PRINT@0,"VOTRE CARTE EST :"
400 IF T<=9 THEN PRINT@20,STR$(T+1);
410 IF T>9 THEN PRINT@20,MID$(B$,ABS(2*T-11),2);
420 PRINT " ";MID$(B$,2*S+1,2)
430 NEXT I
```


2 PI
RO CO
AS CO
3 CA
DA CA
RO CA
9 CO
5 TR
RO PI
9 CA
RO TR
7 CO
9 PI
DA CO
VA TR
3 TR
3 CO
7 PI
4 TR
6 TR
6 CO
QUELLE COLONNE :? 2

# Le jeu de la vie

***REFERENCES*** : Difficulté de programmation : difficile.
Catégorie de jeu : jeu solitaire (2 versions)
Difficluté du jeu : facile.

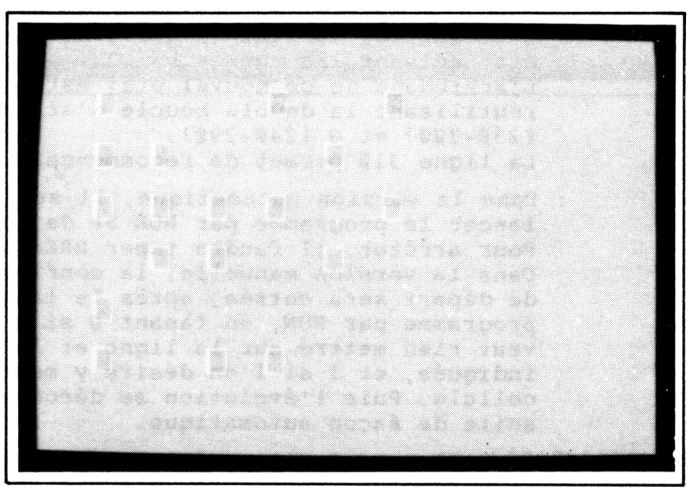
***LE JEU*** : La version présentée ici respecte les règles posées à l'origine par le mathématicien J. H. CONWAY.
Une certaine population est définie au début du jeu soit par Alice (version automatique) soit par vous-même (version manuelle). Cette population va évoluer sous vos yeux, en respectant certaines 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 aux lignes 4Ø et 5Ø, pour enregistrer l'état de la population.
Les boucles F (23Ø-3ØØ) et G (24Ø-29Ø) sont utilisées en début de jeu pour définir la population initiale (lignes 25Ø et 26Ø).
Le calcul de la nouvelle population est ensuite réalisé à l'aide d'une quadruple boucle : 8Ø-21Ø, 9Ø-2ØØ, 11Ø-16Ø, 12Ø-25Ø.
La ligne 13Ø 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 (23Ø-3ØØ) et G (24Ø-29Ø).
La ligne 31Ø permet de recommencer.

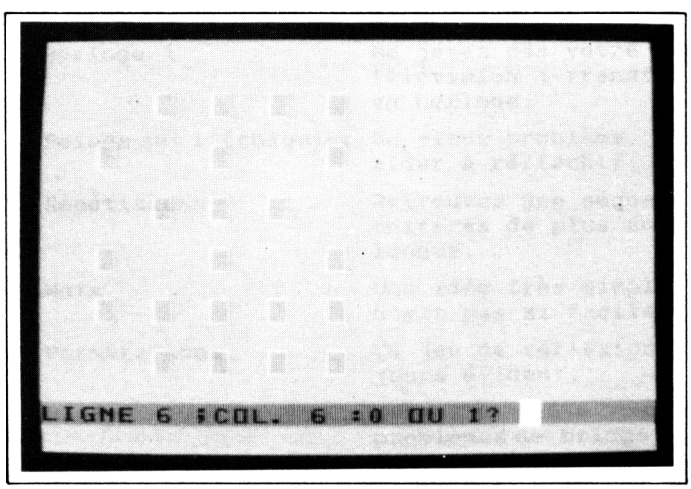
***POUR JOUER*** : Dans la version automatique, il suffit de lancer le programme par RUN et de regarder. Pour arrêter, il faudra taper BREAK.
Dans la version manuelle, la configuration de départ sera entrée, après le lancement du programme par RUN, en tapant Ø si l'on ne veut rien mettre sur la ligne 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, si l'on dispose d'une mémoire plus importante, 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
30 INPUT"DIMENSION ";A
40 DIM B(A,A)
50 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 8
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I<>0 THEN 270
260 IF RND(0)<0.5 THEN C(F,G)=1
270 IF C(F,G)=1 THEN PRINT@(64*F+3*G),"6"; : REM COULEUR 6 ; SHIFT 0
280 B(F,G)=C(F,G)
290 NEXT G
300 NEXT F
310 GOTO 80
```



```
10 REM LE JEU DE LA VIE (MANUEL)
20 CLS
30 INPUT"DIMENSION ";A
40 DIM B(A,A)
50 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 8
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I<>0 THEN 270
260 PRINT@448,"LIGNE"F";COL."G":0 OU 1";:INPUT C(F,G)
270 IF C(F,G)=1 THEN PRINT@(64*F+3*G),"6";: REM COULEUR 6 ; SHIFT Q
280 B(F,G)=C(F,G)
290 NEXT G
300 NEXT F
310 GOTO 80
```



# NIVEAU 3

| N° | Nom du programme | Thème et particularités |
|---|---|---|
| 1 | **Un dé** | Tirage d'un dé, avec graphisme |
| 2 | **Losanges** | Pour vous aider à vous endormir |
| 3 | **Plus grand, plus petit** | Un jeu de déduction intéressant, pour un joueur. |
| 4 | **Ordre** | Un jeu d'astuce pour remettre en ordre, le plus vite possible. |
| 5 | **Le pendu** | Un jeu de lettres très classique (deux joueurs). |
| 6 | **Belote** | Pour simuler une donne, avec un jeu de 32 cartes. |
| 7 | **Un taquin de calcul** | Un bon entraînement (2 joueurs). |
| 8 | **Pendu inversé** | Cette fois, c'est l'ordinateur qui essaie de deviner le mot. |
| 9 | **Nicomaque** | Pour se perfectionner en calcul mental (1 ou plusieurs joueurs). |
| 10 | **Course de chevaux** | Assez visuel : un jeu de hasard pour plusieurs joueurs. |
| 11 | **A travers** | Un jeu de stratégie, pour 2 joueurs. |
| 12 | **Mastermind** | Avec des chiffres, mais très classique. Un seul joueur. |
| 13 | **Idem** | Un jeu de calcul mental, avec plusieurs niveaux, pour plusieurs joueurs. |
| 14 | **Horloge 1** | Ne jetez pas votre vieille télévision : transformez-la en horloge. |
| 15 | **Reines sur l'échiquier** | Un vieux problème, pour vous aider à réfléchir. |
| 16 | **Répétitions** | Retrouvez une séquence de chiffres de plus en plus longue. |
| 17 | **Mots** | Une idée très simple, mais ce n'est pas si facile de trouver ! |
| 18 | **Permutations** | Un jeu de réflexion, pas toujours évident. |
| 19 | **Bridge** | Mettez au point vos propres problèmes de bridge. |
| 20 | **Hanoï** | Un conseil : commencez petit !!! |
| 21 | **Solfège** | Apprenez le solfège avec Alice. |
| 22 | **Le loup et les agneaux** | Un jeu de réflexion, pour 2 joueurs. |

# Manipulation des chaînes et graphiques 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 dans le Basic Microsoft 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 Ième 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 d'Alice ne permet pas le remplacement direct d'une sous-chaîne par une affectation du type MID$ (-.-) = " ". Il sera nécessaire, pour remplacer la Ième lettre de la chaîne A$ d'écrire :

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

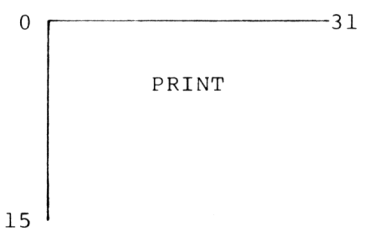
Des formules analogues pourront être utilisées pour remplacer dans A$ une sous-chaîne par une autre, pas nécessairement de même longueur, ce qui permet une grande souplesse.

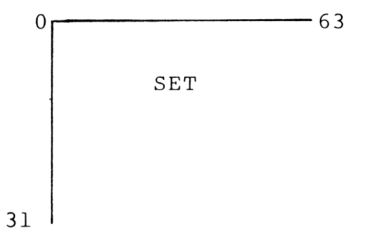
## LES GRAPHIQUES "HAUTE RESOLUTION"

Alice dispose de la possibilité (un peu limitée, il est vrai) d'allumer ou d'éteindre des points de couleur sur l'écran, avec une résolution de 64 x 32 points (soit de Ø à 63 dans le sens de la largeur, et de Ø à 31 dans le sens de la hauteur).

Il conviendra, au préalable, de mettre l'écran en noir en utilisant l'instruction CLS Ø.

Le repérage de l'écran est le même que pour l'instruction PRINT, avec 4 fois plus de points disponibles.




L'instruction utilisée (SET) demande 3 paramètres : ce sont dans l'ordre, la position en largeur, la position en hauteur, et la couleur.

Ainsi, un point allumé en couleur orange par SET (50, 21, 8) sera sur la 21ème ligne, en 50 ème colonne.

Le même point pourra être éteint par :

RESET (50, 21)

La couleur de ce point peut être détectée par une instruction particulière : POINT (50, 21) qui renvoie un nombre de Ø à 8 correspondant à la couleur, ou qui renvoie -1 si un caractère occupe la position indiquée.

# Un dé

**REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : tirage d'un dé (avec graphisme).
Difficulté du jeu : aucune.

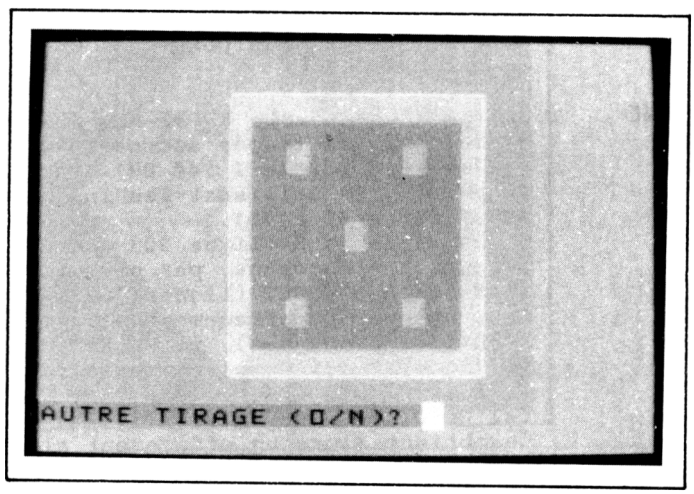
**LE JEU** : Le programme propose simplement le lancer d'un dé aléatoire, et sa représentation graphique en grand format.
Il pourra facilement être utilisé en sous-programme, dans tout type de jeu faisant appel au lancer d'un ou de plusieurs dés.

**LE PROGRAMME** : La ligne 3Ø utilise le générateur aléatoire d'Alice pour tirer un nombre de 1 à 6.
La boucle J (5Ø-7Ø) dessine les bords du dé, et les lignes 4Ø et 8Ø dessinent le haut et le bas.
Les lignes 1ØØ, 11Ø, 115, 12Ø, 13Ø et 14Ø sont utilisées pour le dessin des points.
Les lignes 15Ø et 16Ø permettront le cas échéant de refaire un tirage.

**POUR JOUER** : On lance le dé par RUN, le résultat est affiché sous forme symbolique, comme sur un véritable dé. La question "AUTRE TIRAGE (O/N) ? est ensuite affichée.
Si l'on répond "O" et ENTER, un autre lancer sera effectué.

**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 3
30 A=RND(6)
40 PRINT@74,"8888888888888";:REM COULEUR 8 ; 13 FOIS SHIFT Q
50 FOR J=0 TO 8
60 PRINT@106+32*J,"8           8";:REM COULEUR 8 ; SHIFT Q ,
      PUIS 11 ESPACES ET SHIFT Q
70 NEXT J
80 PRINT@394,"8888888888888";:REM COULEUR 8 ; 13 FOIS SHIFT Q
90 IF A<>1 THEN PRINT@141,"7";:REM COULEUR 7 ; SHIFT Q
100 IF A>=4 THEN PRINT@147,"7";:REM COULEUR 7 ; SHIFT Q
110 IF A=6 THEN PRINT@237,"7";:REM COULEUR 7 ; SHIFT Q
115 IF A=6 THEN PRINT@243,"7";:REM COULEUR 7 ; SHIFT Q
120 IF A/2<>INT(A/2) THEN PRINT@240,"7";:REM COULEUR 7 ; SHIFT Q
130 IF A>=4 THEN PRINT@333,"7";:REM COULEUR 7 ; SHIFT Q
140 IF A<>1 THEN PRINT@339,"7";:REM COULEUR 7 ; SHIFT Q
150 PRINT@448,"AUTRE TIRAGE (O/N)";:INPUTR$
160 IF LEFT$(R$,1)="O" THEN RUN
```



# Losanges

***REFERENCES*** : Difficulté de programmation : facile.
Catégorie de jeu : interlude.
Difficulté du jeu : aucune.

***LE JEU*** : Le programme va construire progressivement 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 des losanges.

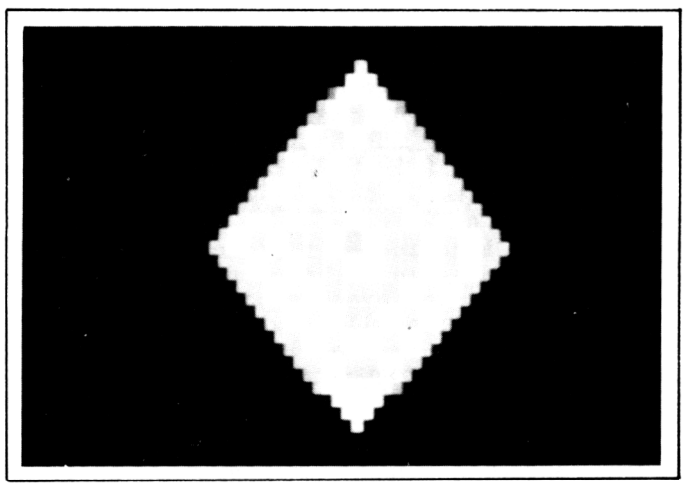
***LE PROGRAMME*** : Une première boucle K (3Ø-85) définit les dimensions des tracés successifs.
La deuxième boucle I (4Ø-8Ø) trace les bandes de points, en utilisant les instructions SET (lignes 6Ø à 75).
Si la variable S (ligne 5Ø) est à 1, alors il y aura effacement, par utilisation des instructions RESET (lignes 1ØØ à 115).
Pour obtenir un effacement aléatoire (deuxième programme), il suffira de modifier la ligne 5Ø qui contient normalement :
IF S <> Ø THEN 1ØØ, en écrivant 50 ON RND (4) GOTO 1ØØ, 1Ø5, 11Ø, 115.
On obtient alors un effacement aléatoire.

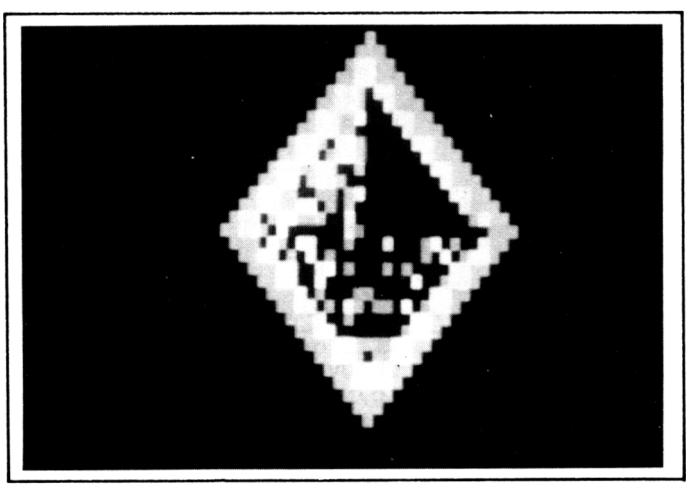
***POUR JOUER*** : On commence l'exécution du programme avec RUN.
Il suffit, dès lors, d'attendre et de regarder. On pourra arrêter le déroulement à tout moment, en tapant BREAK.

***EXTENSIONS POSSIBLES*** : On pourra immédiatement, bien sûr, songer à introduire la couleur et, éventuellement, les sons.
On peut, à partir de là, imaginer toute une série de programmes fonctionnant sur le même principe (kaléidoscope, tapisseries, etc.).

```
10 REM LOSANGES
20 CLS 0
30 FOR K=0 TO 15
40 FOR I=0 TO K
50 IF S<>0 THEN 100
60 SET(31-K+I,15-I,INT((K+1)/2))
65 SET(31-K+I,15+I,INT((K+1)/2))
70 SET(31+K-I,15+I,INT((K+1)/2))
75 SET(31+K-I,15-I,INT((K+1)/2))
80 NEXT I
85 NEXT K
90 S=ABS(S-1)
95 GOTO 30
100 RESET(31-K+I,15-I)
105 RESET(31-K+I,15+I)
110 RESET(31+K-I,15+I)
115 RESET(31+K-I,15-I)
120 GOTO 80
```

```
10 REM LOSANGES
20 CLS 0
30 FOR K=0 TO 15
40 FOR I=0 TO K
45 IF S=0 THEN 60
50 ON RND(4) GOTO 100,105,110,115
60 SET(31-K+I,15-I,INT((K+1)/2))
65 SET(31-K+I,15+I,INT((K+1)/2))
70 SET(31+K-I,15+I,INT((K+1)/2))
75 SET(31+K-I,15-I,INT((K+1)/2))
80 NEXT I
85 NEXT K
90 S=ABS(S-1)
95 GOTO 30
100 RESET(31-K+I,15-I)
105 RESET(31-K+I,15+I)
110 RESET(31+K-I,15+I)
115 RESET(31+K-I,15-I)
120 GOTO 80
```





# Plus grand, pus petit

**REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : jeu de déduction (un seul joueur).
Difficulté du jeu : difficile.

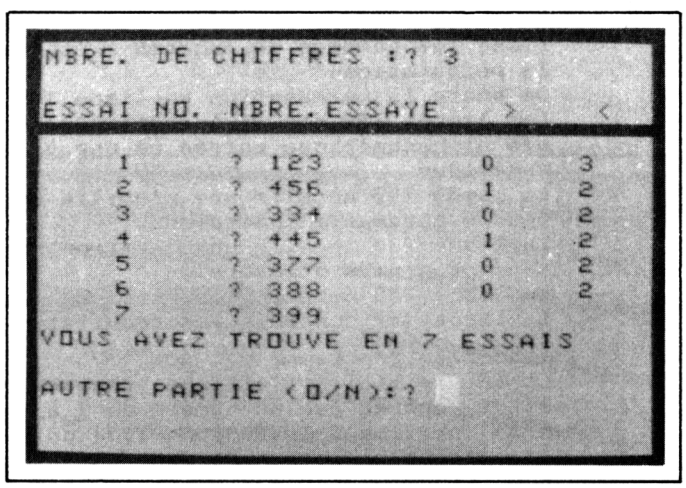
**LE JEU** : Il s'inspire d'un principe légèrement différent de celui du MASTERMIND ; il s'agit également de deviner un nombre (de 1 à 7 chiffres) mais les indications dont on dispose sont cette fois d'une part le nombre de chiffres qui, dans le nombre essayé, sont supérieurs à ceux du nombre à deviner, 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 4Ø interdit de demander des nombres de plus de 8 chiffres.
Le nombre à deviner est placé, en ligne 6Ø, dans une variable alphanumérique X$.
Les comparaisons avec le nombre essayé sont faites à la ligne 11Ø (qui teste si l'on a trouvé juste), et dans la boucle I (13Ø-17Ø), où l'on fait la comparaison "lettre à lettre".
L'affichage du résultat est réalisé grâce à la ligne 18Ø.
La ligne 185 permet de contrôler l'affichage.
Le compteur du nombre d'essais est en ligne 19Ø.

**POUR JOUER** : Le lancement du programme par RUN provoque l'affichage NBRE DE CHIFFRES.
On entre un nombre inférieur à 8 et on appuie sur la touche ENTER.
Le numéro de l'essai s'affiche, ainsi que la question NBRE ESSAYE ; on répond en entrant un nombre, suivi de ENTER.
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 Alice affichera VOUS AVEZ TROUVE...EN...ESSAIS. Puis AUTRE PARTIE (O/N) ? ; on fera O ou N suivi de ENTER 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 Alice qui tiendra le rôle de décodeur.

```
10 REM PLUS GRAND , PLUS PETIT
20 CLS
30 INPUT"NBRE. DE CHIFFRES :";C
40 IF C>8 THEN RUN
50 E=1
60 X$=STR$(INT(RND(0)*9^C+10^(C-E)))
70 X$=RIGHT$(X$,LEN(X$)-1)
80 PRINT@64,"ESSAI NO. NBRE.ESSAYE   >    <"
90 PRINT""
100 PRINT@(131+32*F),E;:INPUT"    ";P$
110 IF P$=X$ THEN 200
120 A=0:B=0
130 FOR I=1 TO C
140 IF MID$(P$,I,1)>MID$(X$,I,1) THEN A=A+1
150 IF MID$(P$,I,1)<MID$(X$,I,1) THEN B=B+1
170 NEXT I
180 PRINT@(150+32*F),A;:PRINT@(155+32*F),B;
185 F=F+1 : IF F=11 THEN F=0
190 E=E+1
195 GOTO 100
200 PRINT"VOUS AVEZ TROUVE EN"E"ESSAIS"
210 PRINT:SOUND190,9:SOUND210,7:SOUND180,9
220 INPUT"AUTRE PARTIE (O/N):";R$
230 IF LEFT$(R$,1)="O" THEN RUN
```



# 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 possibles, par des permutations successives de deux des lettres ; mais ceci a aussi pour inconvénient de faire permuter autour d'elle les deux lettres qui entourent celle-ci, suivant le mecanisme expliqué ci-dessous :

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

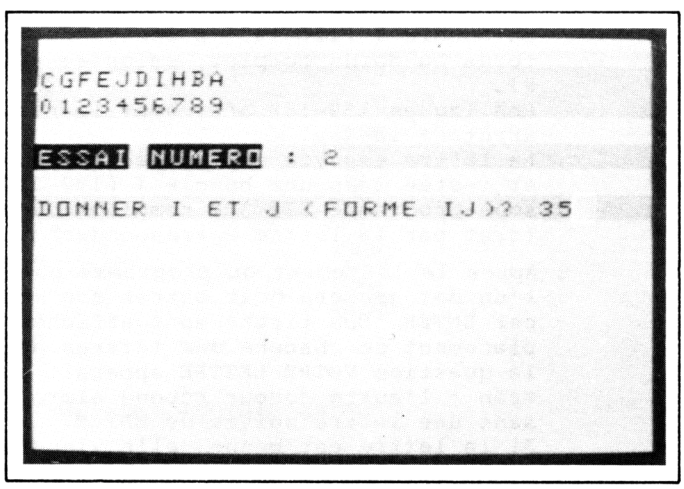
```
B A H F D J I E G C
Ø 1 2 3 4 5 6 7 8 9 : résultat
```

***LE PROGRAMME*** : Les boucles emboîtées I (4Ø-1ØØ) et J (6Ø-8Ø) définissent une variable alphanumérique A$, contenant les 10 premières lettres de l'alphabet dans le désordre.
La ligne 12Ø affiche A$, et la ligne 13Ø affiche les numéros de lettres à utiliser pour la permutation.
On entre la permutation en ligne 14Ø.
Les lignes 16Ø et 17Ø transforment la variable alphanumérique entrée en une variable numérique.
La ligne 18Ø appelle une première fois le sous-programme d'inversion.
La ligne 23Ø appelle une deuxième fois le sous-programme d'inversion.
La ligne 25Ø comptabilise les essais.
La ligne 26Ø contrôle les résultats.

***POUR JOUER*** : 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 6
30 A$=""
40 FOR I=1 TO 10
50 R$=CHR$(RND(10)+64)
60 FOR J=1 TO I-1
70 IF R$=MID$(A$,J,1) THEN 50
80 NEXT J
90 A$=A$+R$
100 NEXT I
110 S=1
120 PRINT@32,A$
130 PRINT"0123456789"
135 PRINT@128,"essai numero :"S
140 PRINT@192,"DONNER I ET J (FORME IJ)";:INPUTR$
160 I=ASC(LEFT$(R$,1))-47
170 J=ASC(RIGHT$(R$,1))-47
180 GOSUB 500
210 IF LEFT$(R$,1)="0" OR LEFT$(R$,1)="9" THEN 250
220 I=I-1:J=I+2
230 GOSUB 500
250 S=S+1
260 IF A$="ABCDEFGHIJ" THEN PRINT@416,"TROUVE EN"S"COUPS":END
265 SOUND190,4
270 GOTO 120
500 B$=""
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$:GOTO 600
570 IF T=J THEN B$=B$+N$:GOTO 600
580 B$=B$+MID$(A$,T,1)
600 NEXT T
610 A$=B$
620 RETURN
```



# 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 fausse 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 3Ø) ; une variable B$, en ligne 35, contient tous les éléments du dessin.
La boucle I (12Ø-14Ø) s'occupe du dessin (plus ou moins complet, selon la valeur de P).
Les lignes 15Ø-16Ø affichent le résultat et arrêtent le jeu.
La lettre essayée est entrée en ligne 17Ø et testée dans une boucle I (19Ø-21Ø) ; le sous-programme 3ØØ-32Ø remplace chaque tiret par la lettre correspondante.

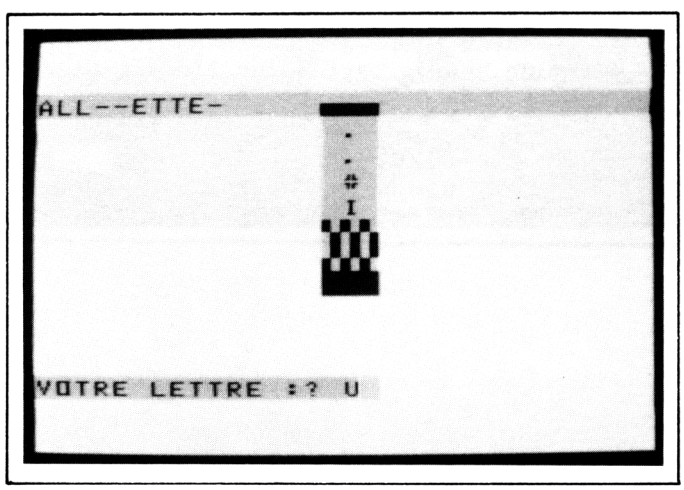
***POUR JOUER*** : Après le lancement du programme par RUN, l'un des joueurs doit entrer son mot et taper ENTER. Des tirets sont affichés à l'emplacement de chacune des lettres du mot ; et la question VOTRE LETTRE apparaît sur l'écran ; l'autre joueur répond alors en proposant une lettre suivie de ENTER.
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 en DATA une liste de mots dans laquelle ALICE pourra piocher au hasard ; on pourra améliorer le dessin, et jouer sur le nombre d'essais permis.

```
10 REM LE PENDU
20 CLS 8
30 INPUT"VOTRE MOT :";R$
35 B$="000000000000000 I  #  .  . 000":REM COULEUR 0 ; TAPER
   SHIFT S , ESPACE ,SHIFT Z , SHIFT Y , ESPACE , SHIFT G
40 REM ( SUITE ) 3 FOIS SHIFT B , 6 FOIS SHIFT A , ESPACE , I ,
   2 ESPACES , # ,2 ESPACES , . , 2 ESPACES  , . , ESPACE
45 REM ( SUITE ) ET 3 FOIS SHIFT R ; SOIT 30 CARACTERES EN TOUT
50 P$=""
60 FOR I=1 TO LEN(R$)
70 P$=P$+"-"
80 NEXT I
90 P=11
100 CLS8
110 PRINT@64,P$
120 FOR I=11 TO P STEP -1
130 PRINT@(32*(12-I)+15),MID$(B$,1+3*(I-1),3);
140 NEXT I
150 IF P$=R$ THEN PRINT@416,"BRAVO":SOUND200,9:SOUND220,7:
                                                SOUND190,9:END
160 IF P=1 THEN PRINT@416,"PERDU; LE MOT ETAIT ";R$:SOUND120,10
                                                        :END
170 PRINT@416,"VOTRE LETTRE :";:INPUT S$
180 T=0
190 FOR I=1 TO LEN(R$)
200 IF MID$(R$,I,1)=S$ THEN GOSUB 300
210 NEXT I
220 IF T<>0 THEN 100
230 P=P-1
240 GOTO 100
300 T=1
310 P$=LEFT$(P$,I-1)+S$+RIGHT$(P$,LEN(P$)-I)
320 RETURN
```



# Belote

***REFERENCES*** : Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de cartes.
Difficulté du jeu : facile.

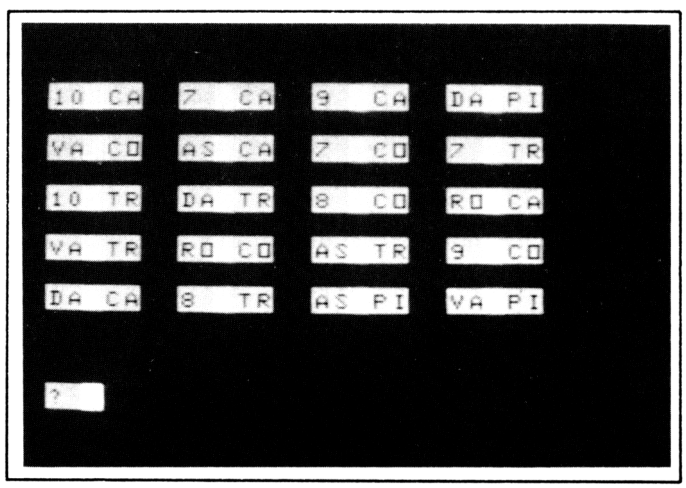
***LE JEU*** : Le jeu proposé est plutôt une simulation, réalisée sur 32 cartes, de la distribution de 4 paquets de 5 cartes. Il pourra être utilisé pour l'étude de problèmes faisant intervenir des donnes de 20 cartes sur 32.

***LE PROGRAMME*** : En ligne 2Ø, on crée un tableau pouvant contenir les 20 cartes.
La boucle I (5Ø-1ØØ) assure le tirage aléatoire de 20 nombres, pris parmi 32 ; la boucle J (7Ø-9Ø) vérifie que l'on ne réalise pas deux fois le même tirage.
La deuxième boucle I (11Ø-26Ø) assure ensuite l'affichage, les noms des cartes étant affichés en clair, et la couleur étant décodée et affichée en clair, grâce aux lignes 14Ø, 2ØØ, 21Ø et 22Ø.

***POUR JOUER*** : Il suffit de taper RUN pour obtenir immédiatement l'affichage de 4 paquets de 5 cartes chacun (une donne).
On tapera ENTER pour obtenir une nouvelle distribution.

***EXTENSIONS POSSIBLES*** : On pourra facilement mettre au point un affichage plus graphique ; 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 DIM A(20)
30 K=1
40 CLS 3
45 PRINT@232,"Patientez . Je tire";
50 FOR I=1 TO 20
60 A(I)=RND(32)
65 IF I=1 THEN 100
70 FOR J=1 TO I-1
80 IF A(I)=A(J) THEN 60
90 NEXT J
100 NEXT I
105 CLS 0
110 FOR I=1 TO 20
120 S$="CA"
130 S=1+A(I)-8*INT(A(I)/8)
140 R$=MID$("7 8 9 10VADAROAS",2*S-1,2)
200 IF A(I)<=24 THEN S$="TR"
210 IF A(I)<=16 THEN S$="PI"
220 IF A(I)<=8 THEN S$="CO"
230 PRINT@(64*K+7*INT((I-1)/5)),R$;"4";:REM COULEUR 4 ; SHIFT Q
235 PRINTS$;
240 K=K+1
250 IF K=6 THEN K=1
260 NEXT I
270 PRINT@448,:INPUTR$:RUN
```



# Un taquin de calcul

*REFERENCES* : Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : difficile.

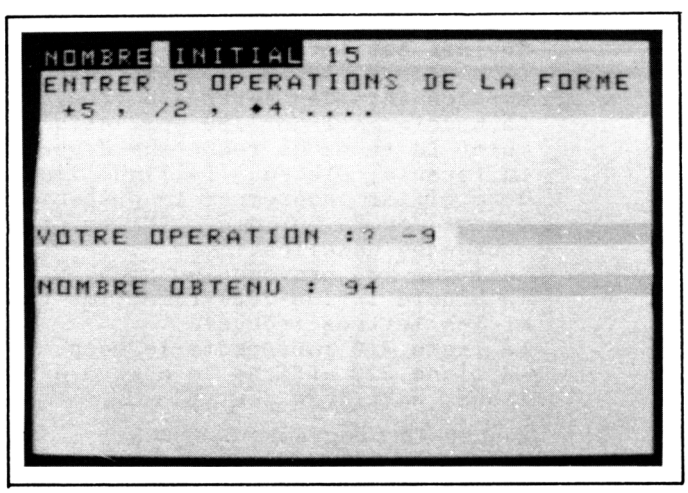
*LE JEU* : Il se pratique à deux joueurs, l'un tenant le rôle du codeur, l'autre celui du décodeur. Le codeur part d'un nombre (proposé par Alice) et doit le transformer à l'aide de 5 opérations qu'il choisit lui-même, ainsi d'ailleurs que les opérateurs.
Le résultat qu'il a obtenu et les opérations qu'il a utilisées seront affichées sur l'écran ; le décodeur devra alors essayer de retrouver le nombre initial en effectuant les opérations en sens inverse.

*LE PROGRAMME* : La ligne 3Ø permet d'obtenir l'entier de départ compris entre 5 et 20.
Une boucle I (9Ø-16Ø) permet d'entrer successivement les 5 opérations.
4 sous-programmes sont utilisés pour les calculs (aux lignes 1ØØØ...4ØØØ).
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 15Ø).
La ligne 17Ø affiche les 5 opérations utilisées, ainsi que les opérateurs correspondants.

*POUR JOUER* : On frappe RUN pour lancer le programme.
Le nombre de 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 ENTER.
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 Alice participer davantage en lui confiant le rôle du codage, etc.

```
10 REM UN TAQUIN DE CALCUL
20 CLS 4
30 X=RND(16)+4
40 PRINT"nombre initial"X
50 PRINT"ENTRER 5 OPERATIONS DE LA FORME"
60 PRINT" +5 , /2 , *4 ...."
70 F=X
80 DIM R$(5)
90 FOR I=1 TO 5
100 PRINT@224,"VOTRE OPERATION :";:INPUT R$(I)
110 A=ASC(LEFT$(R$(I),1))-41
120 IF A<=0 OR A=3 OR A=5 OR A>6 THEN 100
130 IF A>3 THEN A=(A/2)+1
140 ON A GOSUB 1000,2000,3000,4000
150 PRINT@288,"NOMBRE OBTENU :"F
160 NEXT I
165 PRINT@224,""
170 PRINT@352,"OPERATIONS :"R$(4);" ";R$(1);" ";R$(5);" ";
    R$(3);" ";R$(2)
180 PRINT"         bon coura9e"
190 END
1000 F=F*VAL(RIGHT$(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
```



# Pendu inversé

***REFERENCES*** : Difficulté de programmation : moyenne.
Catégorie de jeu : jeu sur les mots.
Difficulté du jeu : facile.

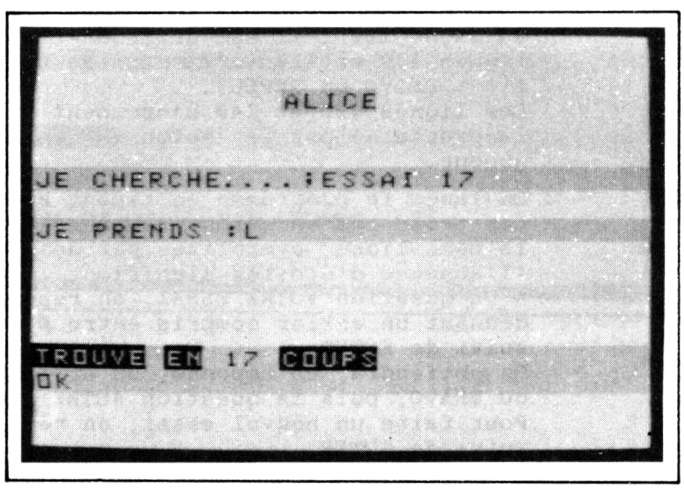
***LE JEU*** : Le PENDU est un jeu de lettres très connu, présenté par ailleurs dans ce recueil, dans lequel il s'agit de deviner un mot caché par l'ordinateur et où chaque lettre non découverte vous rapproche un peu plus de la potence. Ici, c'est Alice qui va chercher à deviner votre mot, mais il joue 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 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 2Ø 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 4Ø.
La boucle I (5Ø-8Ø) imprime des tirets aux emplacements des lettres à trouver.
La boucle en ligne 1ØØ sert uniquement à simuler le temps de recherche d'une lettre.
La formule, placée à la ligne 11Ø, permet de modifier légèrement la probabilité de tirage, en faveur des voyelles et des consonnes usuelles.
La ligne 13Ø provoque un nouveau tirage, et la boucle I (17Ø-2ØØ) réimprime les tirets et les lettres trouvées.
La ligne 21Ø incrémente le compteur d'essais.
La ligne 22Ø affiche le nombre d'essais réalisés, en fin de partie.

***POUR JOUER*** : Lancer le programme par RUN.
Entrer le mot à découvrir, suivi de ENTER.
Des tirets sont alors affichés (un par lettre) et Alice propose une première lettre, regarde si elle convient et continue ceci automatiquement, jusqu'à ce qu'il ait trouvé.
On pourra sortir du programme à tout moment, en tapant BREAK.

***EXTENSIONS POSSIBLES*** : Il sera sans doute intéressant, pour de jeunes enfants, de prévoir une version non automatique où le joueur répondrait OUI ou NON pour chaque lettre proposée par Alice, 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
15 CLS 6
20 D$="YAEIOUBCDFGLMNPRSTVWHKQXZJ"
30 C=1:T$=""
40 INPUT"ENTRER VOTRE MOT :";R$
45 CLS 3
50 FOR I=1 TO LEN(R$)
60 PRINT@(76+I),"-";
70 T$=T$+"-"
80 NEXT I
90 PRINT@160,"JE CHERCHE....;ESSAI"C
100 FOR J=1 TO 1600:NEXT J
110 X=-6*(RND(0)<0.7)-13*(RND(0)<0.5)-7*(RND(0)<0.4)
120 X=INT(X*RND(0)+1)
130 IF MID$(D$,X,1)="-" THEN 110
140 E$=MID$(D$,X,1)
150 D$=LEFT$(D$,X-1)+"-"+RIGHT$(D$,26-X)
160 PRINT@224,"JE PRENDS :";E$
165 SOUND 230,3
170 FOR I=1 TO LEN(R$)
180 IF E$=MID$(R$,I,1) THEN T$=LEFT$(T$,I-1)+E$+RIGHT$(T$,
    LEN(T$)-I)
190 PRINT@(76+I),MID$(T$,I,1);
200 NEXT I
210 C=C+1
220 IF T$=R$ THEN PRINT@384,"trouve en"C-1"coups":SOUND190,9:
    SOUND220,8:SOUND180,9:END
230 GOTO 90
```



# Nicomaque

***REFERENCES*** : Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion (calcul mental).
Difficulté du jeu : moyenne.

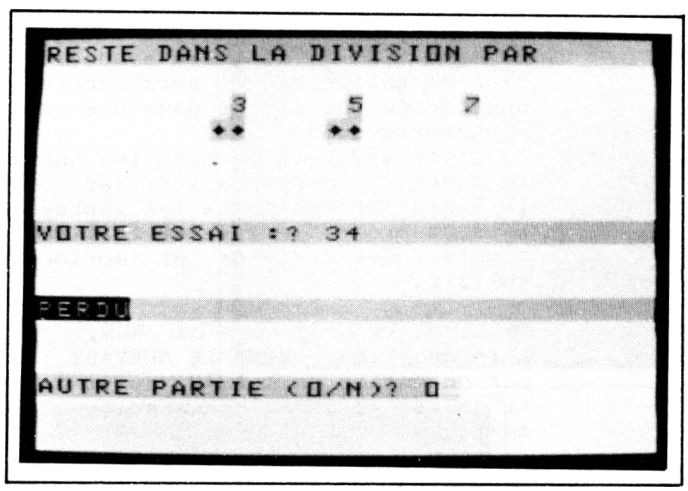
***LE JEU*** : Il utilise une propriété un peu particulière des entiers de 1 à 100 qui peuvent être identifiés sans ambiguïté par leurs restes dans les divisions par 3, 5 et 7.
Le programme, proposé ici, est un programme codeur qui affiche simplement les restes d'un entier donné N dans les divisions indiquées.
Le jeu consiste à essayer de découvrir l'entier (mais, dans cette version, vous n'avez droit qu'à un essai).

***LE PROGRAMME*** : La ligne 3Ø permet de confectionner l'entier X.
La boucle I (4Ø-9Ø) calcule les trois restes successifs, et la boucle emboîtée J (6Ø-8Ø) permet de les afficher sous une forme assez visuelle, aux bons emplacements.
Les lignes 35 et 38 sont là pour indiquer de quelle division il s'agit.
Le nombre proposé par le joueur est entré à la ligne 1ØØ ; il est affiché et testé aux lignes 11Ø et 12Ø (affichage de fin de partie : BRAVO ou PERDU).
Les lignes 13Ø et 14Ø 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 Ø).
A la question VOTRE ESSAI, on répondra en donnant un entier compris entre Ø et 1ØØ, suivi de ENTER.
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 ENTER.

***EXTENSIONS POSSIBLES*** : On peut envisager de réaliser le programme décodeur (on donne les restes et Alice devra vous donner N) et chercher si la propriété peut être étendue (en effectuant davantage de divisions, par exemple).

```
10 REM NICOMAQUE
20 CLS 8
30 X=RND(100)
35 PRINT"RESTE DANS LA DIVISION PAR"
38 PRINT@74,"3";:PRINT@80,"5";:PRINT@86,"7";
40 FOR I=1 TO 3
50 A=X-(2*I+1)*INT(X/(2*I+1))
55 IF A=0 THEN 90
60 FOR J=1 TO A
70 PRINT@(98+J+6*I),"*";
80 NEXT J
90 NEXT I
100 PRINT@224,"VOTRE ESSAI :";:INPUTN
110 IF N=X THEN PRINT@320,"bravo":SOUND180,9:SOUND220,7:SOUND170,9
120 IF N<>X THEN PRINT@320,"Perdu":SOUND120,18
130 PRINT@416,"AUTRE PARTIE (O/N)";:INPUTR$
140 IF LEFT$(R$,1)="O" THEN RUN
```



# Course de chevaux

***REFERENCES*** : Difficulté de programmation : moyenne.
Catégorie de jeu : hasard ( plusieurs joueurs).
Difficulté du jeu : aucune.

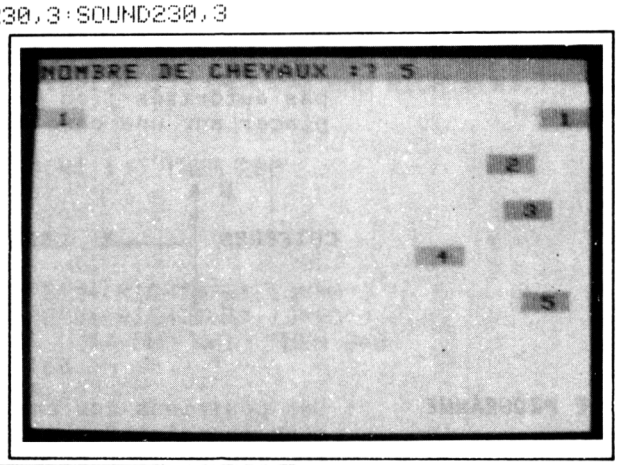
***LE JEU*** : On entrera un nombre de chevaux compris entre 1 et 10, et la course débutera immédiatement. Il pourra y avoir des ex-aequo, car l'on tiendra compte du fait que Alice s'occupe de faire avancer les chevaux les uns après les autres, en commençant par le numéro 1 ; le classement sera affiché au fur et à mesure des arrivées (et en tenant compte des ex-aequo).

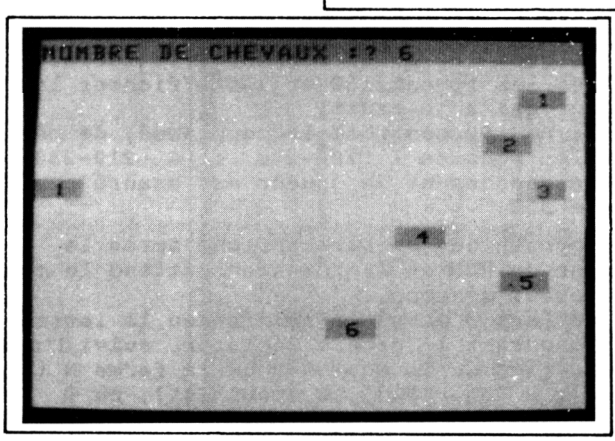
***LE PROGRAMME*** : Une première boucle I (70-160) affiche les chevaux dans leur position de départ.
Une deuxième boucle I (100-160), utilisant le même NEXT I, les fera avancer ; la position de chaque cheval sera repérée par un chiffre ou une lettre dans une variable alphanumérique A$.
La ligne 120 fera avancer les chevaux.
La ligne 130 gère les arrivées.
La ligne 140 réalisera les impressions.
Le sous-programme 200-240 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 10, suivi de ENTER.
La course se déroulera ensuite automatiquement.

***EXTENSIONS POSSIBLES*** : On pourra dessiner des 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 8
30 INPUT"NOMBRE DE CHEVAUX :";N
35 A$=""
40 P=0
50 T=1
60 A=P
70 FOR I=T TO N
80 A$=A$+"0"
90 GOTO 135
100 FOR I=1 TO N
110 IF MID$(A$,I,1)="G" THEN 160
120 B$=CHR$(ASC(MID$(A$,I,1))+RND(4))
125 A$=LEFT$(A$,I-1)+B$+RIGHT$(A$,LEN(A$)-I)
130 IF B$>="G" THEN GOSUB 200
135 PRINT@(64*I+ASC(MID$(A$,I,1))-46),"8888";:REM COULEUR 8 ;
                                                    SHIFT 0
140 PRINT I;
145 SOUND 230,1
150 FOR K=1 TO 800:NEXT K
160 NEXT I
170 P=P+T
180 T=0
190 IF A<N THEN 100
195 SOUND170,12:SOUND210,9:SOUND190,12:PRINT@448,"":END
200 A$=LEFT$(A$,I-1)+"G"+RIGHT$(A$,LEN(A$)-I)
210 PRINT@(64*I),P;
215 SOUND230,3:SOUND230,3:SOUND230,3
220 T=1
230 A=A+T
240 RETURN
```

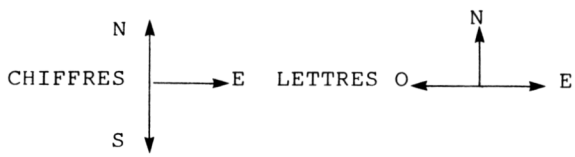


# A travers

**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.



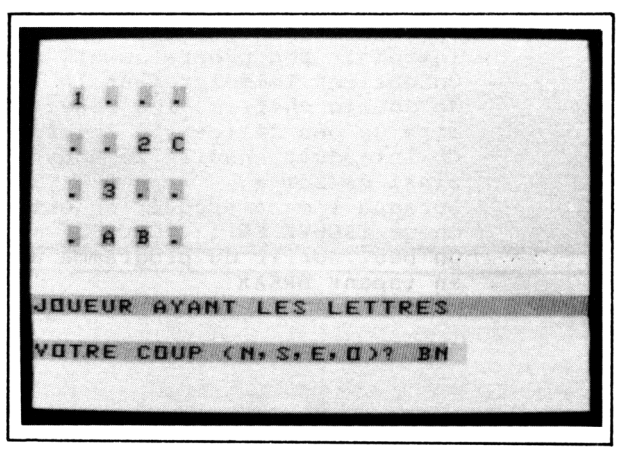

**LE PROGRAMME** : Les positions des chiffres et des lettres sont enregistrées dans une table N (4,4) ; les lignes 1Ø à 8Ø définissent les valeurs de départ.
Les lignes 1ØØ à 15Ø dessinent le terrain de jeu et les lignes 16Ø et 17Ø affichent le joueur qui a le trait.
La ligne 19Ø contrôle le coup joué, de même que les boucles I (2ØØ-24Ø) et J (21Ø-23Ø) et le changement de joueur est assuré par la ligne 35Ø.

**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 ENTER. Si le coup est invalide, 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 et de lettres et d'améliorer les contrôles. On peut aussi définir une stratégie pour Alice.

```
10 REM A TRAVERS
20 K=1
30 DIM N(4,4)
40 FOR I=1 TO 4
50 N(I,1)=ASC("0")+I
60 N(4,I)=ASC("?")+I
70 NEXT I
80 N(4,1)=0
90 CLS 8
100 FOR U=1 TO 4
110 FOR V=1 TO 4
120 IF N(V,U)=0 THEN PRINT@(64*V+2*U),".";
130 IF N(V,U)<>0 THEN PRINT@(64*V+2*U),CHR$(N(V,U));
140 NEXT V
150 NEXT U
160 PRINT@352,"JOUEUR AYANT LES ";:IF K=0 THEN PRINT "LETTRES"
170 IF K=1 THEN PRINT"CHIFFRES"
180 PRINT@416,"VOTRE COUP (N,S,E,O)";:INPUTR$
190 IF (K=1 AND RIGHT$(R$,1)="O") OR (K=0 AND RIGHT$(R$,1)="S")
                                                        THEN 160
200 FOR I=1 TO 4
210 FOR J=1 TO 4
220 IF N(I,J)=ASC(LEFT$(R$,1)) THEN 280
230 NEXT J
240 NEXT I
250 GOTO 160
280 A=I:B=J
290 I=I-(RIGHT$(R$,1)="S")+(RIGHT$(R$,1)="N")
300 J=J-(RIGHT$(R$,1)="E")+(RIGHT$(R$,1)="O")
310 IF (I=0 AND K=0) OR (J>4 AND K=1) THEN 340
320 IF N(I,J)<>0 THEN 160
330 N(I,J)=ASC(LEFT$(R$,1))
340 N(A,B)=0
350 K=ABS(K-1)
360 GOTO 90
```



# Mastermind

***REFERENCES*** : Difficulté de programmation : difficile.
Catégorie de jeu : logique (un seul joueur).
Difficulté du jeu : difficile.

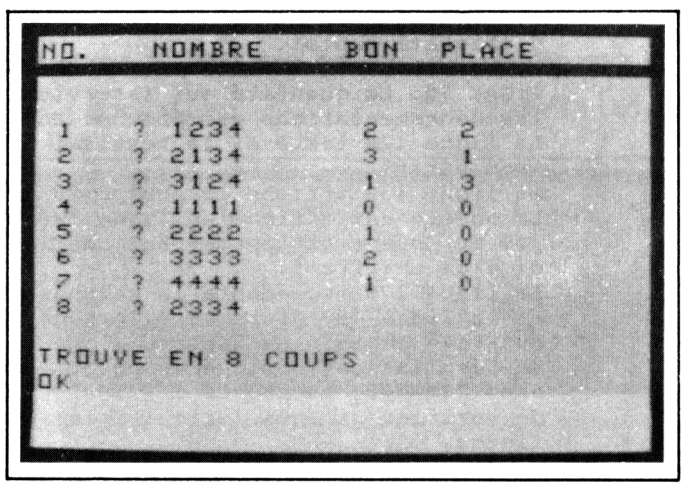
***LE JEU*** : Devenu un grand classique des jeux sur micro-ordinateur, ce jeu développe le raisonnement déductif ; il faut réussir à trouver le plus vite possible un nombre de n chiffres (on pourra prendre n compris entre 1 et 5) en disposant à chaque essai de renseignements. En effet, Alice répondra en indiquant d'une part 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 3Ø et 4Ø.
Le nombre est fabriqué à la ligne 6Ø (il pourra y avoir des chiffres répétés).
L'affichage est préparé par la ligne 9Ø.
La ligne 12Ø s'occupe de la fin de partie.
Une première boucle I (13Ø-26Ø) teste les chiffres qui sont à la bonne place.
Une deuxième boucle J (15Ø-17Ø) teste les chiffres qui sont présents et mal placés.
Les compteurs sont aux lignes 22Ø et 25Ø.
La ligne 27Ø affiche le résultat, et la ligne 28Ø comptabilise les essais.
La ligne 29Ø s'occupe de l'affichage.

***POUR JOUER*** : On tape RUN et l'on choisit la longueur du nombre désiré (de 1 à 6), suivie de ENTER.
Après l'affichage NO. NOMBRE BON PLACE, on introduit son propre essai, suivi de ENTER.
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 BREAK.

***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 4
30 INPUT"COMBIEN DE CHIFFRES :";N
40 IF N>6 THEN RUN
50 E=1:F=1
60 X$=STR$(INT(RND(0)*9^N+10^(N-1)))
70 X$=RIGHT$(X$,LEN(X$)-1)
80 CLS4
90 PRINT"NO.   NOMBRE    BON  PLACE"
95 PRINT"0000000000000000000000000000000000":REM COULEUR 0 ;
                                                32 FOIS SHIFT B
100 PRINT@(32*F+64),E;TAB(5);:INPUTE$
110 X=0:Y=X
120 IF E$=X$ THEN PRINT:PRINT"TROUVE EN"E"COUPS":SOUND190,9:
                                                SOUND220,7:SOUND170,9:END
130 FOR I=1 TO N
140 IF MID$(X$,I,1)=MID$(E$,I,1) THEN 250
150 FOR J=1 TO N
160 IF J<>I AND MID$(X$,I,1)=MID$(E$,J,1) THEN 220
170 NEXT J
180 GOTO 260
220 Y=Y+1
230 GOTO 260
250 X=X+1
260 NEXT I
270 PRINT@(32*F+80),X;:PRINT@(32*F+85),Y;
275 SOUND230,2:SOUND230,2
280 E=E+1
290 F=F+1:IF F>10 THEN F=1
300 GOTO 100
```



# Idem

***REFERENCES*** : Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de calcul mental (2 joueurs).
Difficulté du jeu : selon le niveau choisi.

***LE JEU*** : Alice 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 Alice 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.

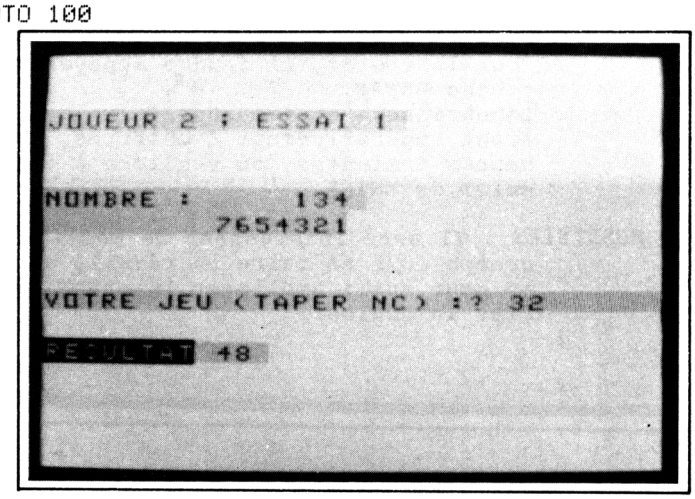
***LE PROGRAMME*** : Une variable numérique N est dimensionnée à 2 en ligne 4Ø, pour contenir les résultats successifs des deux joueurs.
La ligne 5Ø permettra d'entrer le niveau de difficulté choisi.
Le nombre initial est confectionné par la ligne 7Ø. La quantité qui interviendra dans les décrémentations est obtenue en ligne 9Ø.
La ligne 12Ø teste si le résultat n'est pas trop petit.
La ligne 13Ø affiche le numéro du joueur.
Le nombre est affiché en ligne 14Ø et le jeu du joueur est entré en ligne 16Ø (nombre de deux chiffres).
La ligne 17Ø incrémente la colonne choisie, et la ligne 18Ø décrémente le nombre de la quantité choisie au départ.
Le résultat est affiché un court instant (lignes 19Ø-2ØØ. La boucle I (22Ø-24Ø) permet de vérifier si tous les chiffres sont identiques.
L'impression et la fin de partie sont effectuées en ligne 25Ø. Les lignes 26Ø et 27Ø 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 ENTER ; 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 BREAK.

***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 4
30 H=1
40 DIM N(2)
50 INPUT"NIVEAU (2-7) :";N
60 U=10:P=0
70 N(1)=INT((U^(N-1))*RND(0)+U^(N-1))
80 N(2)=N(1)
90 A=INT(2*U^(N-1)*RND(0)+U^(N-1))
100 IF H=1 THEN P=P+1
110 CLS 4
120 IF N(H)<U^(N-1) THEN N(H)=U^N-ABS(N(H))
130 PRINT@64,"JOUEUR"H;" ESSAI"P;
140 PRINT@160,"NOMBRE :";TAB(15-N);N(H);
150 PRINT@192,TAB(9);"7654321";
160 PRINT@288,"VOTRE JEU (TAPER NC) :";:INPUTR$
170 N(H)=N(H)+VAL(LEFT$(R$,1))*U^(VAL(RIGHT$(R$,1))-1)
180 N(H)=N(H)-A
185 IF N(H)>U^N THEN N(H)=N(H)-U^N
187 N(H)=INT(N(H))
190 PRINT@352,"resultat"N(H);
200 FOR J=1 TO 1900:NEXT J
210 R$=STR$(N(H))
215 R$=RIGHT$(R$,LEN(R$)-1)
220 FOR I=1 TO LEN(R$)-1
230 IF MID$(R$,I,1)<>MID$(R$,I+1,1) THEN 260
240 NEXT I
250 PRINT"BRAVO , JOUEUR"H:SOUND200,9:SOUND230,7:SOUND190,9:END
260 H=H+1
270 IF H>=3 THEN H=1
280 GOTO 100
```



# Horloge

***REFERENCES*** : Difficulté de programmation : difficile.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

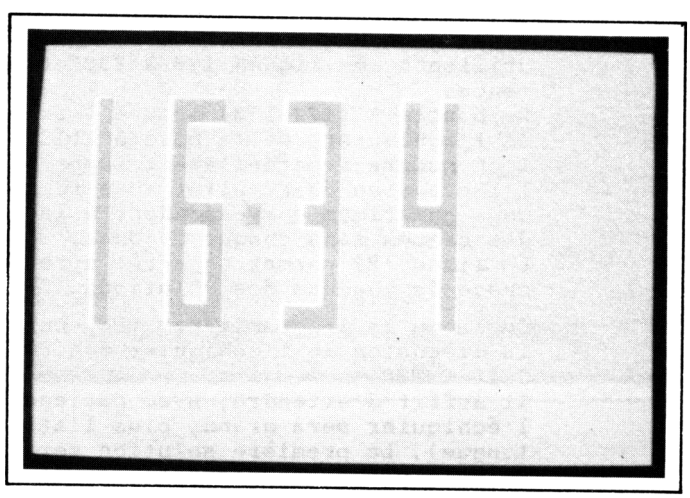
***LE JEU*** : Transformer votre écran de télévision en horloge numérique à affichage grand format, grâce à ce programme qui vous permettra d'animer une vitrine, de faire une horloge fantaisie, et qui pourra se prêter à bien d'autres usages.

***LE PROGRAMME*** : La ligne 3Ø vous permettra d'entrer l'heure sous forme d'un entier de 4 chiffres (par exemple : 1024 signifiera 10 heures 24 minutes).
La ligne 6Ø prépare la variable à afficher.
Les lignes 8Ø-15Ø réalisent le dessin, en décalant l'impression au fur et à mesure (à l'aide de la variable K, incrémentée en ligne 16Ø.
Les lignes 18Ø à 22Ø assurent le changement d'heure.
Les boucles J (24Ø-3ØØ) et I (25Ø-28Ø) assurent la temporisation et l'affichage symbolique des secondes (lignes 26Ø et 27Ø).

***POUR JOUER*** : Dès le lancement par RUN, le programme se place en attente de l'heure ; on tape au prochain top sonore et on attend le top pour appuyer sur ENTER. L'heure s'affiche alors assez rapidement et l'horloge est en route. Pour l'arrêter, il faudra appuyer sur la touche BREAK.
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 ENTER.

***EXTENSIONS POSSIBLES*** : Il sera intéressant de modifier le programme pour en faire un réveil, avec alarme, on peut aussi améliorer le graphisme, modifier la couleur.

```
10 REM HORLOGE
20 CLS 8
30 INPUT"HEURE (FORME XXXX) :";A$
40 CLS 8
50 K=1
60 A=VAL(MID$(A$,K,1))
70 IF A<>1 AND A<>4 THEN PRINT@(59+6*K),"333";:REM COULEUR 3 ;
                                                  3 FOIS SHIFT Q
80 FOR J=1 TO 5
90 IF A=0 OR (A>3 AND A<>7) THEN PRINT@(32*(J+1)+6*K-5),"3";:REM
                                                  COULEUR 3 ; SHIFT Q
100 IF A<>5 AND A<>6 THEN PRINT@(32*(J+1)+6*K-3),"3";:REM
                                                  COULEUR 3 ; SHIFT Q
110 IF A=0 OR A=2 OR A=6 OR A=8 THEN PRINT@(32*(J+5)+6*K-5),"3";
                                                  :REM COULEUR 3 ;SHIFT Q
120 IF A<>2 THEN PRINT@(32*(J+5)+6*K-3),"3";:REM COULEUR 3 ;
                                                  SHIFT Q
130 NEXT J
140 IF A>1 THEN PRINT@(192+6*K-4),"33";:REM COULEUR 3 ; 2 FOIS
                                                  SHIFT Q
150 IF A<>1 AND A<>4 AND A<>7 THEN PRINT@(320+6*K-5),"333";:REM
                                                  COULEUR 3 ; 3 FOIS SHIFT Q
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 59
250 FOR I=1 TO 80
260 IF H=1 THEN PRINT@203,"4";:REM COULEUR 4 ; SHIFT Q
270 IF H=0 THEN PRINT@203," ";
280 NEXT I
290 H=ABS(H-1)
300 NEXT J
310 SOUND220,2: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 célèbre 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 retenu 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 graphique et sous forme numérique.

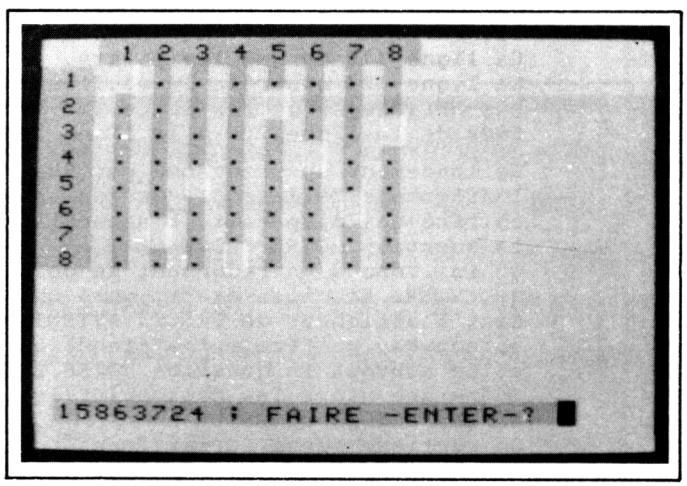
**LE PROGRAMME** : La ligne 30 vous permet de choisir la taille de l'échiquier.
Une variable D sera dimensionnée en conséquence, en ligne 40.
Une boucle d'exploration principale I (50-100) contient une sous-boucle J (70-90) qui réalise le contrôle de non-alignement (ligne 80).
Si, au cours de l'exploration, on rencontre un alignement, on revient en arrière, en utilisant les lignes 190 à 230 et on recommence.
La boucle F (120-170) assure l'impression de l'échiquier, et la boucle K(135-145), 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 180 permet un arrêt après l'affichage de chacune des solutions.

**POUR JOUER** : On lance le programme par RUN, puis on tape la dimension de l'échiquier désiré, et on fait ENTER.
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 la touche ENTER pour le faire repartir. Pour terminer avant l'exploration

de toutes les solutions, il faudra taper BREAK.

***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'ECHIQUIER
20 CLS
30 INPUT"TAILLE DE L'ECHIQUIER :";S
40 DIM D(S)
50 I=1
60 D(I)=1
65 IF I=1 THEN 100
70 FOR J=1 TO I-1
80 IF (D(I)=D(J)) OR (ABS(D(I)-D(J))=I-J) THEN 200
90 NEXT J
100 I=I+1 : IF I<=S THEN 60
110 CLS 3
120 FOR F=1 TO S
130 PRINT@(2*F+1),F;
135 FOR K=1 TO S
140 PRINT@(32*F),F;:PRINT@(32*F+2*K+2),".";
145 NEXT K
150 PRINT@(32*F+2*D(F)+2),"4";:REM COULEUR 4 ; SHIFT Q
160 PRINT@(448+F),RIGHT$(STR$(D(F)),1);
170 NEXT F
180 INPUT" ; FAIRE -ENTER-";R$
190 I=I-1
200 D(I)=D(I)+1
210 IF D(I)<=S THEN 65
220 I=I-1
230 IF I<>0 THEN 200
```



# Répétitions

**REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : jeu de mémoire (1 ou plusieurs joueurs).
Difficulté du jeu : moyenne.

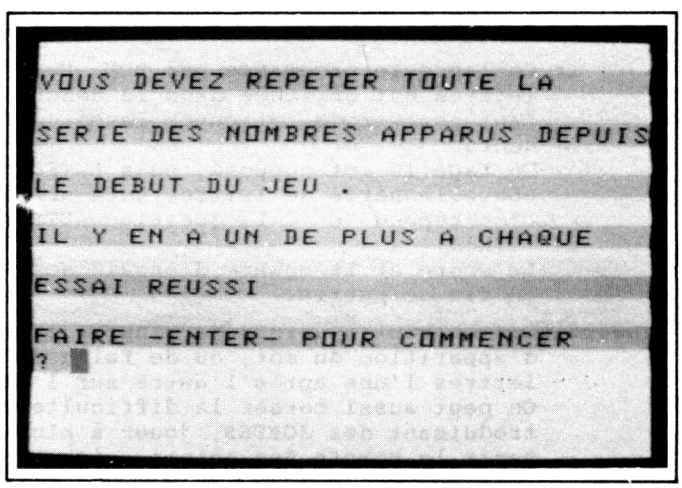
**LE JEU** : Des nombres vont apparaître un par un. Le joueur devra à chaque coup les répéter tous, depuis le début (ce qui signifie qu'il en aura un de plus à taper à chaque fois). Il devra essayer de tenir le plus longtemps possible. Le meilleur score est mémorisé après chaque partie, ce qui permettra éventuellement de jouer à plusieurs.

**LE PROGRAMME** : La ligne 3Ø appelle un sous-programme (lignes 1ØØØ à 1Ø8Ø) qui donne la règle du jeu.
La ligne 5Ø fabrique les chiffres, qui sont ensuite stockés dans une variable alphanumérique A$ (ligne 6Ø).
La ligne 7Ø permet l'impression du dernier chiffre au hasard sur la quatrième ligne de l'écran, et la durée de l'affichage est déterminée par la ligne 75.
La ligne 85 attend la réponse du joueur, qui est comparée à la réponse correcte en ligne 9Ø.
La ligne 11Ø marque les points.
La ligne 12Ø contrôle le meilleur score.
Les affichages de fin de partie sont effectués de la ligne 2ØØ à la ligne 25Ø.

**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 ENTER. 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 ENTER.
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 O pour continuer.
On peut aussi quitter le programme à tout moment, en tapant BREAK 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 M=0
25 A$=""
30 GOSUB 1000
40 H=0
50 X=RND(10)-1
60 A$=A$+RIGHT$(STR$(X),1)
65 CLS 4
70 PRINT@(96+RND(29)),X;
75 FOR I=1 TO 300 + RND(400):NEXT I
80 CLS 4
85 PRINT@288,"VOTRE SEQUENCE :";:INPUTR$
90 IF R$<>A$ THEN 200
100 PRINT@352,"exact.....attention.....";
105 FOR I=1 TO 500:NEXT I
110 H=H+1
120 IF M<H THEN M=H
130 GOTO 50
200 CLS 4
210 PRINT"VOTRE SCORE EST DE"H
220 PRINT"MEILLEUR SCORE ACTUEL"M
230 PRINT@192,"AUTRE PARTIE (O/N) :";:INPUTR$
240 IF R$="N" THEN END
250 GOTO 25
1000 CLS 8
1010 PRINT@32,"VOUS DEVEZ REPETER TOUTE LA"
1020 PRINT@96,"SERIE DES NOMBRES APPARUS DEPUIS";
1030 PRINT@160,"LE DEBUT DU JEU ."
1040 PRINT@224,"IL Y EN A UN DE PLUS A CHAQUE"
1050 PRINT@288,"ESSAI REUSSI"
1060 PRINT@352,"FAIRE -ENTER- POUR COMMENCER"
1070 INPUT R$
1080 RETURN
```



# Mots

***REFERENCES*** : Difficulté de programmation : facile.
Catégorie de jeu : jeu de lettres (pour 1 ou plusieurs joueurs).
Difficulté du jeu : moyenne.

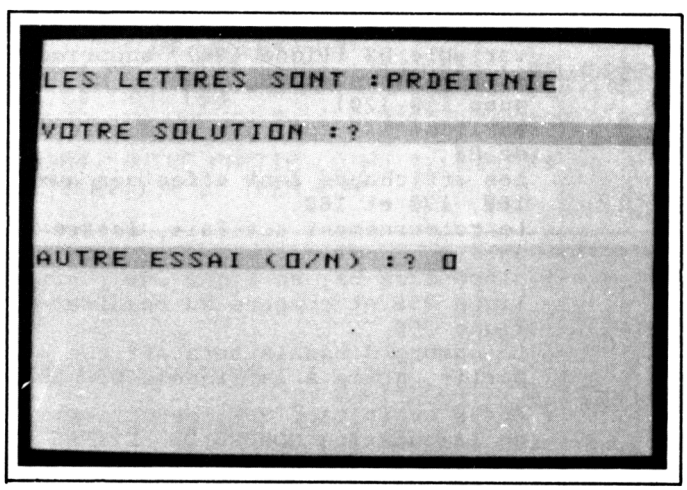
***LE JEU*** : Il est difficile d'imaginer plus simple dans son principe que ce jeu, qui se contente d'afficher dans le désordre les lettres prises dans une trentaine de mots. Mais essayez de jouer, et vous verrez que ce n'est pas si facile, la difficulté variant évidemment selon la liste de mots que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

***LE PROGRAMME*** : La ligne 3Ø permet de dimensionner la variable A$ (I) qui contiendra les mots.
La boucle I (4Ø-6Ø) saisit la liste des mots (inscrits en DATA aux lignes 24Ø à 28Ø).
La ligne 7Ø choisit un mot au hasard ; la boucle I (1ØØ-15Ø) contrôle si une lettre n'a pas déjà été tirée, et reprend les lettres dans le désordre.
La ligne 16Ø affiche les lettres, et la ligne 17Ø attend votre proposition, qui est testée en ligne 18Ø.
Les essais sont incrémentés en ligne 19Ø.
La fin de partie est contrôlée par la ligne 2ØØ, et le score est affiché en ligne 22Ø.

***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 ENTER, 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 JOKERS, jouer à plusieurs, tenir le compte des points, etc.

```
10 REM MOTS
20 CLS4
30 DIM A$(30):S=0:E=0
40 FOR I=1 TO 30
50 READ A$(I)
60 NEXT I
65 CLS 4
70 Z=RND(30)
80 IF A$(Z)="" THEN 70
90 R$=A$(Z):A$="":A$(Z)="":V$=R$
100 FOR I=1 TO LEN(R$)
110 Z=RND(LEN(R$))
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@32,"LES LETTRES SONT :";A$;
170 PRINT@96,"VOTRE SOLUTION :";:INPUTS$
180 IF S$=V$ THEN S=S+1
190 E=E+1
200 PRINT@256,"AUTRE ESSAI (O/N) :";:INPUTR$
210 IF LEFT$(R$,1)="O" THEN 65
220 PRINT@320,"VOTRE SCORE :"S"SUR"E"ESSAIS"
230 END
240 DATA JAMBON,ENERGUMENE,AFFILIE,ERUDIT,INSIGNIFIANT,REALISTE
250 DATA IMPRESSIONANT,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
```



# Permutations

***REFERENCES*** : Difficulté de programmation : moyenne
Catégorie de jeu : réflexion.
Difficulté de programmation : moyenne.

***LE JEU*** : Alice 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 en début, en utilisant des permutations sur les n premières lettres à partir de la gauche, avec le moins d'essais possibles.

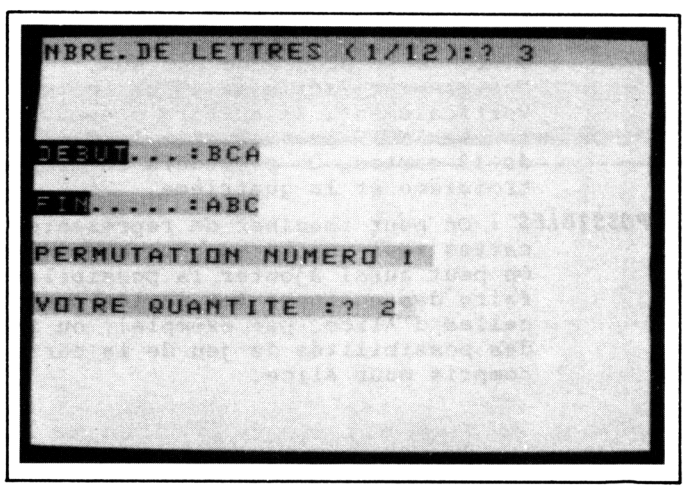
***LE PROGRAMME*** : La ligne 3Ø permet de choisir le nombre de lettres, et la ligne 4Ø contrôle que l'on n'en demande pas plus de 12.
La ligne 8Ø réalise le brouillage, en tirant au hasard la nouvelle position de chaque lettre. Les lettres sont ensuite ajoutées au fur et à mesure de leur tirage, dans une variable B$ (ligne 1ØØ), supprimées dans A$ et replacées en C$, dans le bon ordre (lignes 11Ø-12Ø).
La ligne 14Ø contrôle le nombre de lettres de B$.
Les affichages sont effectués aux lignes 16Ø, 17Ø et 18Ø.
Le retournement est fait, lettre par lettre, dans la boucle I (21Ø-23Ø) ; le résultat est placé dans B$, en ligne 24Ø ; affiché par la ligne 25Ø et comparé au résultat final en ligne 26Ø.
Le nombre d'essais sera affiché en fin de partie, grâce à la ligne 27Ø.

***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 ENTER.
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 ENTER, on obtiendra AEGBDFC.
On poursuivra jusqu'à l'obtention de l'ordre désiré.
Le score sera affiché en fin de partie ;

ainsi qu'une autre 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 CLS 8
30 INPUT"NBRE.DE LETTRES (1/12):";N
40 IF N>12 THEN RUN
50 A$="ABCDEFGHIJKL"
60 C$="            "
70 B$=""
80 X=RND(N)
90 IF MID$(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$))
160 PRINT@128,"debut...:";B$;
170 PRINT@192,"fin.....:";C$;
180 PRINT@256,"PERMUTATION NUMERO"S;
185 PRINT@320,"VOTRE QUANTITE :";:INPUTL
190 S=S+1
200 A$=""
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$)-L)
250 PRINT@384,"RESULTAT :";B$;:SOUND220,1:SOUND220,1:SOUND220,1
260 IF B$<>C$ THEN 180
270 PRINT@320,"TROUVE EN"S-1"ESSAIS":SOUND200,9:SOUND230,8:
                                                    SOUND180,9
280 PRINT@384,"AUTRE PARTIE (O/N):";:INPUTR$
290 IF LEFT$(R$,1)="O" THEN RUN
```



# Bridge

***REFERENCES*** : Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : variable, selon le problème posé.

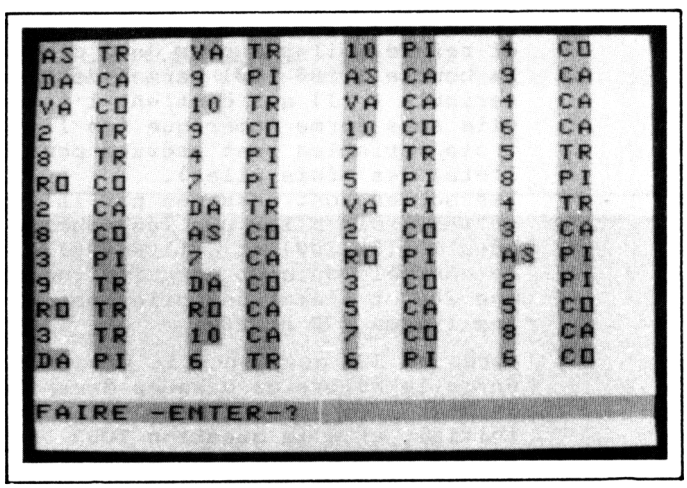
***LE JEU*** : Vous ne trouverez pas dans ce programme, un partenaire pour vos bridges mais, plus exactement une simulation pour l'affichage d'une donne ; cet affichage est réalisé jeu par jeu, on peut, à partir de là, se poser toute sorte de problèmes : annonces, jeu de la carte, etc.

***LE PROGRAMME*** : Les lignes 3Ø et 4Ø définissent un jeu de 52 cartes.
La boucle J (5Ø-19Ø) réalise le tirage de quatre jeux de 13 cartes.
La boucle I (6Ø-17Ø) réalise tirage et impression. La carte est choisie parmi les 52 disponibles, en ligne 8Ø.
La ligne 9Ø calcule la couleur de la carte tirée.
La ligne 1ØØ vérifie que cette carte n'a pas déjà été tirée.
Les lignes 11Ø et 12Ø s'occupent de l'impression complémentaire dans le cas où l'on tire un 10, un valet, une dame, un roi ou un as. L'impression est réalisée par les lignes 13Ø, 14Ø et 15Ø.
La ligne 16Ø permet d'éliminer les cartes déjà tirées, et la ligne 18Ø 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 la touche ENTER pour obtenir la deuxième série de 13 cartes. On procèdera 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 d'Alice, par exemple), ou introduire des possibilités de jeu de la carte, y compris pour Alice.

```
10 REM BRIDGE
15 CLS 8
20 CLEAR 200
30 R$="234567891VDRA"
40 R$=R$+R$+R$+R$
50 FOR J=1 TO 4
60 FOR I=1 TO 13
70 F=I-13*INT((I-1)/13)
80 X=RND(52)
90 S=2*INT((X-1)/13)+1
100 IF MID$(R$,X,1)="0" THEN 80
110 T=X-13*INT(X/13)
120 T=T-13*(T=0)
130 PRINT@(32*I+8*J-40),MID$(R$,X,1);
140 IF T>=9 THEN PRINTMID$("0AAOS",T-8,1);
150 PRINT@(32*I+8*J-37),MID$("TRCACOPI",S,2);
160 R$=LEFT$(R$,X-1)+"0"+RIGHT$(R$,LEN(R$)-X)
170 NEXT I
180 PRINT@448,"FAIRE -ENTER-";:INPUT A
190 NEXT J
```



# Hanoï

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

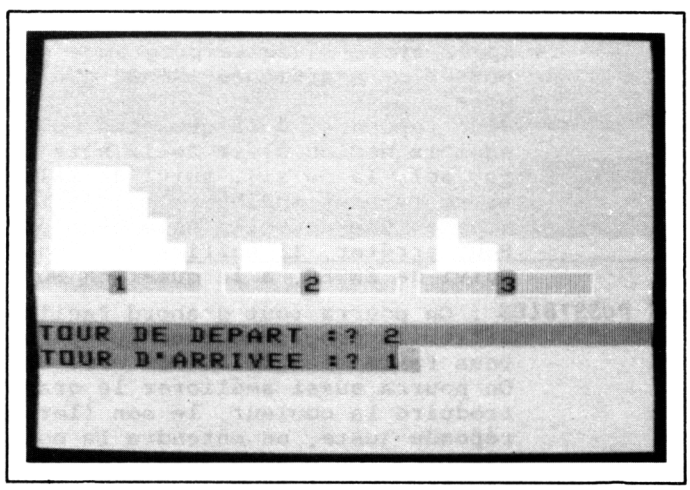
***LE PROGRAMME*** : La variable D$, en ligne 5Ø, servira à dessiner les disques.
Le nombre de disques est entré en ligne 6Ø et refusé s'il y a plus de 8 disques.
La boucle I (8Ø-1ØØ) permet de fabriquer une variable A (I) qui contient l'état de la pile sous forme numérique (en ligne 3Ø, trois variables sont prévues pour conserver l'état des trois piles).
Les socles sont dessinés par la ligne 13Ø et les trois piles par les boucles emboîtées I (15Ø-2ØØ) et J (16Ø-19Ø).
Le contrôle du coup est effectué par la ligne 26Ø et l'état des piles est modifié par les lignes 27Ø et 28Ø.

***POUR JOUER*** : Après un RUN qui lance le programme, on entre le nombre de disques demandés, suivi de ENTER. 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 ENTER 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 d'une couleur différente, par exemple.

```
10 REM HANOI
20 CLS 8
30 DIM A(3)
40 A(1)=9:A(2)=9:A(3)=9
50 D$="555555555":REM COULEUR 5 ; SHIFT Q
60 INPUT"NBRE. DE TOURS (<8):";N
70 IF N>=8 THEN RUN
80 FOR I=0 TO N-1
90 A(1)=VAL(STR$(A(1))+STR$(N-I))
100 NEXT I
110 CLS 8
120 C$="666666666":REM COULEUR 6 ; SHIFT Q
130 PRINT@288,C$;:PRINT@298,C$;:PRINT@308,C$;
140 PRINT@292,"1";:PRINT@302,"2";:PRINT@312,"3";
150 FOR I=0 TO N-1
160 FOR J=1 TO 3
170 B$=RIGHT$(STR$(A(J)),LEN(STR$(A(J)))-1)
180 IF LEN(B$)>I+1 THEN PRINT@(32*(8-I)+10*J-9),LEFT$(D$,VAL
                                                    (MID$(B$,I+2,1)));
190 NEXT J
200 NEXT I
210 PRINT@352,"TOUR DE DEPART :";:INPUTD
220 IF A(D)=0 THEN 210
230 PRINT@384,"TOUR D'ARRIVEE :";:INPUTA
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 210
270 A(A)=VAL(C$+MID$(B$,LEN(B$),1))
280 A(D)=VAL(LEFT$(B$,LEN(B$)-1))
290 GOTO 110
```



# Solfège

***REFERENCES*** : Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : facile.

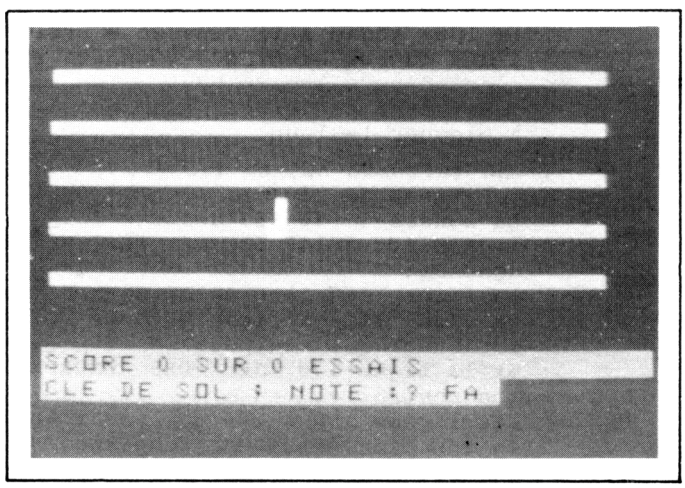
***LE JEU*** : Ce programme vous propose une initiation élémentaire au solfège, en vous permettant d'apprendre à reconnaître facilement les différentes notes, qui apparaissent les unes après les autres sur une portée ; après l'affichage de la note, vous devez entrer son nom en clair et votre score sera 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 (4Ø-7Ø) dessine une portée.
La note est choisie en ligne 8Ø et affichée en lignes 9Ø et 95.
La ligne 1ØØ affiche le score.
La ligne 11Ø affiche le nom de la clé (ici clé de sol) et vous interroge.
La ligne 12Ø s'occupe de la fin de la partie.
Les lignes 13Ø et 14Ø effectuent le décodage de la note entrée.
La ligne 16Ø 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 ENTER.
Le score vous indiquera si c'est exact ou non, et une nouvelle note sera affichée.
Pour arrêter, il suffira de répondre O suivi de ENTER, 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 S=0:P=0
30 CLS0
40 FORI=1TO5
50 FORJ=1TO58
60 SET(J,4*I,8)
65 NEXTJ
70 NEXTI
80 I=INT(RND(12)-1)
90 SET(25,2*I,2):SET(25,2*I+1,2)
95 SET(24,2*I+2,2):SET(25,2*I+2,2)
100 PRINT@416,"SCORE"S"SUR"P"ESSAIS"
110 PRINT@448,"CLE DE SOL ; NOTE :";:INPUTR$
120 IFR$="0"THENEND
130 Y=(1ANDR$="FA")+(2ANDR$="MI")+(3ANDR$="RE")
140 Y=Y+(4ANDR$="DO")+(5ANDR$="SI")+(6ANDR$="LA")
150 P=P+1
160 IFY=IORY=I-7THENS=S+1
170 GOTO30
```



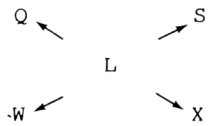
# Le loup et les agneaux

***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.

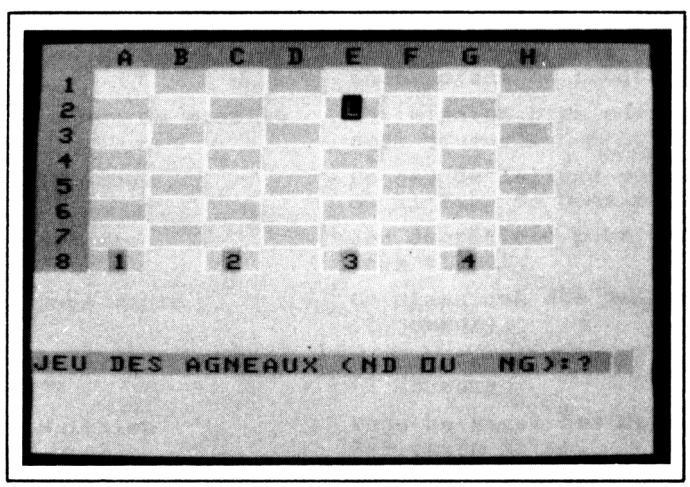
***LE PROGRAMME*** : Une variable A$, en ligne 17, contient les positions de départ.
La boucle I (40-90) dessine le damier, et les lignes 100 à 140 placent le loup et les agneaux.
La ligne 150 gère le changement de joueur.
La ligne 160 affiche "JEU DU LOUP" si c'est le joueur qui manipule le loup qui a le trait et la ligne 170 "JEUX DES AGNEAUX" dans le cas contraire.
Les lignes 190 à 200 analysent la réponse, en fonction du joueur concerné.

***POUR JOUER*** : RUN déclenche l'arrêt sur l'attente d'une entrée : 0 et ENTER 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 (Q, S, W, 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 Q, S, W ou Z, 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 est certainement la réalisation d'un programme permettant à Alice 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 Alice).



```
10 REM LE LOUP ET LES AGNEAUX
15 CLS:INPUT"AGNEAUX(0) OU LOUP(1):";S
17 A$="8183858714"
20 CLS8
30 PRINT"    A  B  C  D  E  F  G  H"
40 FOR I=1 TO 8
50 FOR J=1 TO 8
60 IF (I+J)/2=INT((I+J)/2) THEN 80
70 PRINT@32*I,I;:PRINT@(32*I+3*J),"666";:REM COULEUR 6 ; SHIFT Q
80 NEXT J
90 NEXT I
100 PRINT@(32*VAL(MID$(A$,9,1))+3*VAL(MID$(A$,10,1))+1),"l";
                  :REM TAPER L EN CONTRASTE INVERSE (SHIFT 0 )
110 PRINT@(32*VAL(MID$(A$,1,1))+3*VAL(MID$(A$,2,1))+1),"1";
120 PRINT@(32*VAL(MID$(A$,3,1))+3*VAL(MID$(A$,4,1))+1),"2";
130 PRINT@(32*VAL(MID$(A$,5,1))+3*VAL(MID$(A$,6,1))+1),"3";
140 PRINT@(32*VAL(MID$(A$,7,1))+3*VAL(MID$(A$,8,1))+1),"4";
150 S=ABS(S-1)
160 IF S=0 THEN PRINT@384,"JEU DU LOUP (Q,S,W,X):";:INPUTR$
170 IF S=1 THEN PRINT@384,"JEU DES AGNEAUX (ND OU  NG):";
                                                :INPUTR$:GOTO 220
180 C=10
190 A=VAL(RIGHT$(A$,2))+(C+1 AND R$="X")+(C-1 AND R$="W")
200 A=A-(C+1 AND R$="Q")-(C-1 AND R$="S")
210 GOTO 240
220 C=2*VAL(MID$(R$,1,1))
230 A=VAL(MID$(A$,C-1,2))-(9 AND RIGHT$(R$,1)="D")-(11 AND
                                                RIGHT$(R$,1)="G")
240 A$=LEFT$(A$,C-2)+RIGHT$(STR$(A),2)+RIGHT$(A$,LEN(A$)-C)
250 GOTO 20
```

# NIVEAU 4

| *N°* | *Nom du programme* | *Thème et particularités* |
|---|---|---|
| 1 | **Tirage de dés** | Suffisamment souple pour être utilisé partout. |
| 2 | **Jeu de loto** | Gagnez sans problème, au prochain tirage. |
| 3 | **La bombe** | Saurez-vous la découvrir à temps? (1 joueur). |
| 4 | **Ardoise magique** | Mettez au point vos créations graphiques, sans papier ni crayon. |
| 5 | **Affichage publicitaire** | Si vous ne savez pas quoi mettre dans votre vitrine. |
| 6 | **Evasion** | Il vous faudra beaucoup de chance pour réussir. |
| 7 | **Chapeaux** | Soyez très observateur, et vous réussirez (2 joueurs). |
| 8 | **Combat aérien** | Pas si facile de descendre un appareil qui bouge tout le temps. |
| 9 | **Dessin de dés** | Améliorez vos jeux de dés en incorporant ce programme. |
| 10 | **Le jeu du missile** | Une mission bien périlleuse (un seul joueur). |
| 11 | **Le singe** | Un jeu de lettres pour 2 joueurs où il faut de bons réflexes. |
| 12 | **Adresse** | Jeu de réflexe pour un joueur, très visuel. |
| 13 | **Sous-marin** | Un classique des jeux vidéos (1 joueur). |
| 14 | **Zombies** | Un peu farfelu, mais très visuel, (1 joueur). |
| 15 | **Matthieu** | Vous ne savez pas dessiner. Laissez faire Alice. |
| 16 | **Diago** | Etes-vous vraiment sûr de vos réflexes ? (1 joueur). |
| 17 | **Decomaque** | Pour vous éviter de sécher trop longtemps sur "Nicomaque". |
| 18 | **Chasse au sous-marin** | Il vous faudra pas mal de réflexion, et beaucoup de chance. |
| 19 | **Le lézard d'Alice** | Très spectaculaire. |
| 20 | **Clavier** | Apprenez à taper à la machine en vous amusant. |
| 21 | **Calculatrice** | Un programme de simulation très réaliste. |

# Animation

## SAISIE AU VOL D'UN OU PLUSIEURS CARACTERES FRAPPES AU CLAVIER : INKEY$

Le principe d'un jeu graphique, en animation, est que le jeu doit 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.

Cette possibilité est disponible par **la fonction INKEY$** qui enregistre le nom de la dernière touche frappée. (Il est nécessaire qu'il y ait frappe : on ne devra pas laisser le doigt appuyé sur la touche concernée).

Le INKEY$ sera souvent utilisé dans une boucle d'attente, tant que INKEY$ est vide (rien n'a été frappé), on attend, et l'on sort de la boucle dès qu'une touche a été frappée. Une petite temporisation sera avantageusement utilisée dans certains programmes, pour faciliter la lecture de la touche.

Dans les programmes de jeu, sauf indication contraire, les touches retenues pour les déplacements dans les 4 directions sont :

← (on frappe en réalité Q pour les déplacements vers la gauche).

→ (on frappe en réalité S pour les déplacements vers la droite).

↑ (on frappe en réalité Z pour les déplacements vers le haut).

↓ (on frappe en réalité W pour les déplacements vers le bas).

On pourra facilement modifier les programmes, si l'on désire utiliser d'autres touches.

## DEPLACEMENT D'OBJETS DE BAS EN HAUT SUR L'ECRAN : SIMULATION D'UN " SCROLL "

Alice réalise automatiquement un "Scrolling" (une rotation

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 de nombreux 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 alors 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.

Pour réaliser un "Scroll", il suffit, dès lors, d'incorporer dans le programme une ligne "PRINT @ 48Ø "

Si l'on désire conserver la couleur initiale de l'écran, il faudra incorporer dans les guillemets 32 fois SHIFT @ dans la couleur désirée.

# Tirage des dés

***REFERENCES*** : Difficulté de programmation : aucune.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : aucune.

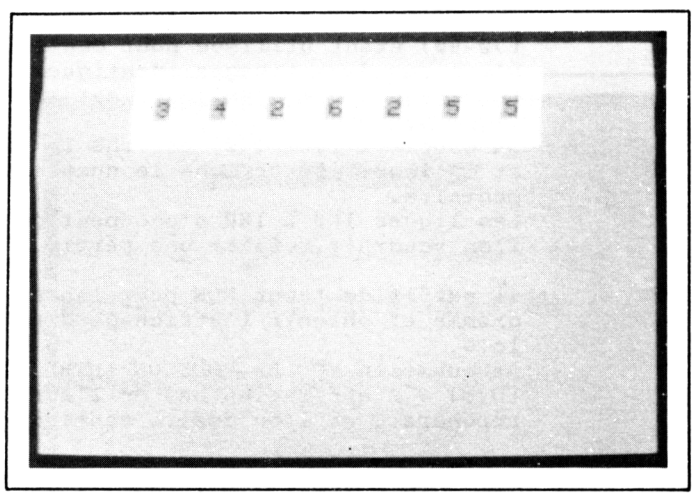
***LE JEU*** : Le programme proposé ne réalise pas un jeu en lui-même, mais pourra être utilisé en sous-programme pour de nombreux jeux.
Il offre simplement la possibilité de lancer de 1 à 9 dés, par simple pression d'une touche. Les dés sont ensuite affichés avec un graphisme très simplifié.

***LE PROGRAMME*** : La ligne 4Ø saisit le caractère frappé au clavier et les lignes 45 et 5Ø le transforment 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 1ØØ).
Une boucle J (7Ø-9Ø) est utilisée pour dessiner le fond, et une autre boucle I (6Ø-12Ø) sert au tirage (tirage en ligne 1ØØ et impression en ligne 11Ø).
La ligne 14Ø 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 alors de presser une touche quelconque pour recommencer le programme.
Pour arrêter, il est nécessaire de taper BREAK.

***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 8
30 PRINT"NOMBRE DE DES A LANCER :";
40 A$=INKEY$:IF A$="" THEN 40
45 IF A$<"1" OR A$>"9" THEN 20
50 N=VAL(A$)
55 CLS 3
60 FOR I=0 TO N-1
70 FOR J=1 TO 3
80 PRINT@(32*J+3*I+5),"555";:REM COULEUR 5 ; SHIFT Q
90 NEXT J
100 D(I)=RND(6)
110 PRINT@(64+3*I+6),RIGHT$(STR$(D(I)),1);
120 NEXT I
130 IF INKEY$="" THEN 130
140 GOTO 20
```



# Jeu de loto

**REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : facile.

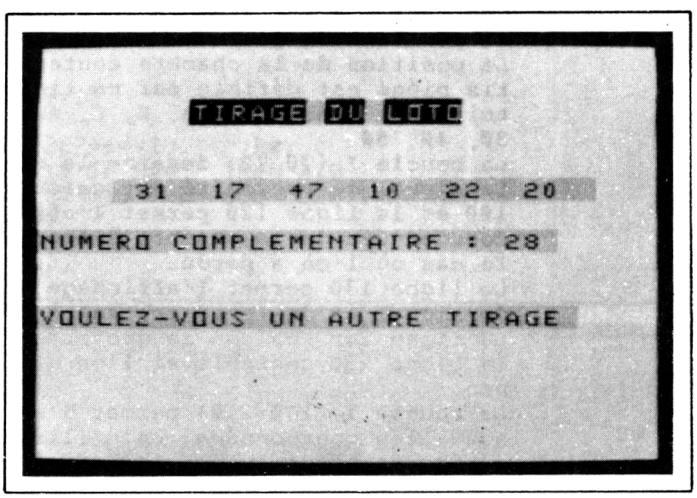
**LE JEU** : Il s'agit de réaliser un tirage complet de loto, c'est-à-dire d'obtenir 6 numéros, compris entre 1 et 49, plus un numéro complémentaire, tous différents, qui pourraient être, par exemple, utilisés pour remplir votre prochaine grille de loto.

**LE PROGRAMME** : On utilisera un tableau N (7), dimensionné en ligne 3Ø, pour enregistrer les résultats.
Le tirage des nombres est effectué dans une boucle I (4Ø-1ØØ), une deuxième boucle J (7Ø-9Ø) étant utilisée pour contrôler qu'il n'y a pas deux tirages identiques.
Le tirage (ou le retirage) est effectué en ligne 5Ø.
La boucle I (12Ø-14Ø) affiche le résultat, et la ligne 15Ø affiche le numéro complémentaire.
Les lignes 16Ø à 18Ø s'occupent du cas où l'on voudrait refaire une 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 des 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 du loto plus usuel, tel qu'il est encore couramment pratiqué dans le midi de la France.

```
10 REM JEU DE LOTO
20 CLS 3
30 DIM N(7)
40 FOR I=1 TO 7
50 N(I)=RND(49)
60 IF I=1 THEN 100
70 FOR J=1 TO I-1
80 IF N(I)=N(J) THEN 50
90 NEXT J
100 NEXT I
110 PRINT@72,"tira9e du loto";
120 FOR I=1 TO 6
130 PRINT@(160+4*I),N(I);" ";
140 NEXT I
150 PRINT@224,"NUMERO COMPLEMENTAIRE :";N(7);
160 PRINT@320,"VOULEZ-VOUS UN AUTRE TIRAGE ";
170 A$=INKEY$:IF A$="" THEN 170
180 IF A$="O" THEN RUN
```



# 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 ?

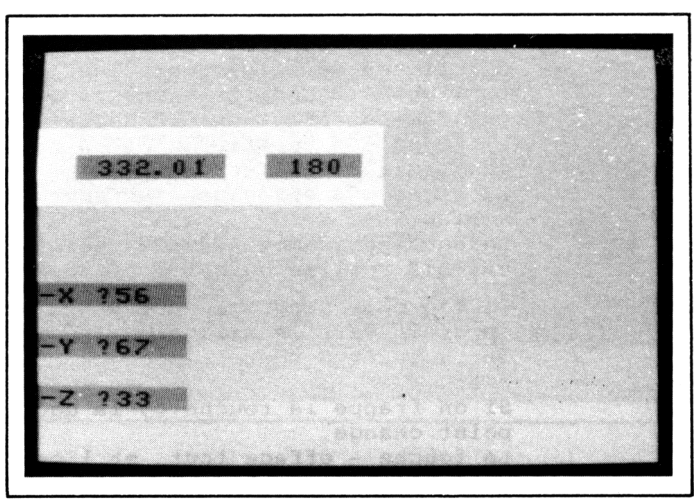
**LE PROGRAMME** : Une variable numérique R, non dimensionnée, contient les trois coordonnées définissant la position du joueur.
La position de la chambre contenant le colis piégé est définie par un tirage aléatoire de 3 variables A, B, C, aux lignes 30, 40, 50.
La boucle I (70-90) dessine le détecteur.
Le signal de proximité est calculé en ligne 100 et la ligne 120 permet l'affichage des coordonnées de la chambre à découvrir, dans le cas où l'on a perdu.
La ligne 130 permet l'affichage, à la fois du signal et du temps restant à courir.
La ligne 140 réalise le décompte du temps.
La ligne 150 contrôle si l'on a trouvé ou non.
La boucle I (170-210) permet d'entrer les nouvelles coordonnées, en utilisant le sous-programme placé aux lignes 400 à 440.

**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 ENTER, 45, puis ENTER, etc.
Le nouveau signal est alors affiché, et le temps est décrémenté de 10.
On continuera ainsi, soit jusqu'au bon

résultat (arrêt du jeu et fin de programme), ou jusqu'à l'apparition de 3 nombres (x, y, z) sans sortie du programme, ce qui signifie alors que l'on a perdu.

***EXTENSIONS POSSIBLES*** : Il sera sans doute indispensable d'améliorer le graphisme, d'ajouter une explosion (sonore et visuelle) pour symboliser l'explosion, d'afficher symboliquement la position dans l'hôtel, etc.

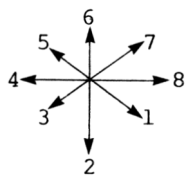
```
10 REM LA BOMBE
20 L=100:T=200
30 A=RND(L)-1
40 B=RND(L)-1
50 C=RND(L)-1
60 CLS 8
70 FOR I=0 TO 17
75 PRINT@96+I,"5";:REM COULEUR 5 ; SHIFT Q
80 PRINT@128+I,"5";:REM COULEUR 5 ; SHIFT Q
85 PRINT@160+I,"5";:REM COULEUR 5 ; SHIFT Q
90 NEXT I
100 S=ABS((A/L+B+C*L)-(R(1)/L+R(2)+R(3)*L))
110 S$=LEFT$(STR$(S),7):S=VAL(S$)
120 IF T=0 THEN PRINT@224,"PERDU ,SOL. :";A;B;C:SOUND160,18:END
130 PRINT@130,S;:PRINT@140,T;
140 T=T-INT(SQR(L))
150 IF S=0 THEN PRINT@224,"bravo":SOUND200,9:SOUND220,7:
                                                SOUND180,9:END
160 PRINT@288,"-X :    ";:PRINT@352,"-Y :    ";:PRINT@416,
                                                  "-Z :    ";
170 FOR I=1 TO 3
180 R$=""
190 GOSUB 400
200 R(I)=VAL(R$)
210 NEXT I
215 SOUND120,15
220 GOTO 70
400 PRINT@(227+64*I),"?";R$;
410 A$=INKEY$:IF A$="" THEN 410
420 IF ASC(A$)=13 THEN SOUND220,1:RETURN
430 R$=R$+A$
440 GOTO 400
```



# 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 :



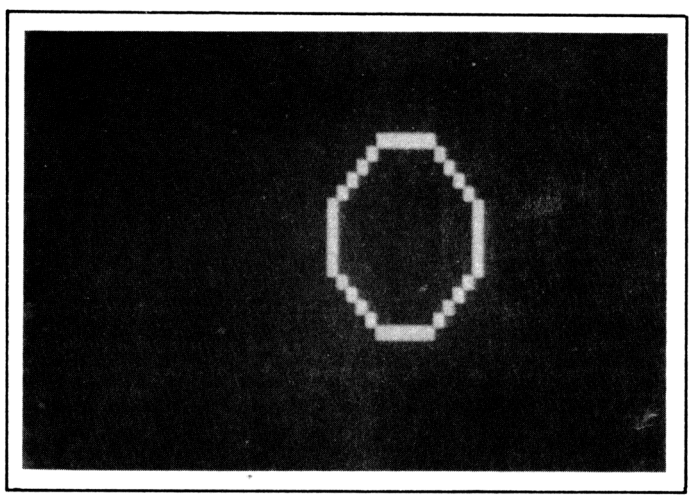

On dispose aussi de la possibilité de déplacer le point sur l'écran sans laisser de trace en appuyant sur la touche Ø (ce qui permet l'effacement partiel) et de laisser de nouveau une trace en réappuyant sur Ø.
On peut choisir la couleur du point en appuyant plusieurs fois de suite sur : (touche :) ; il est possible d'effacer tout l'écran en appuyant sur-(signe -).

**LE PROGRAMME** : La ligne 3Ø définit la position de départ.
La ligne 6Ø imprime le point élémentaire.
Les lignes 8Ø à 1ØØ contrôlent les entrées.
La ligne 14Ø appelle l'un des quatre sous-programmes effectuant le déplacement.
La ligne 15Ø vérifie que l'on ne sort pas des limites de l'écran.
La ligne 17Ø effectue l'effacement s'il y a lieu.
Le sous-programme 4ØØ-42Ø choisit la couleur, 5ØØ-51Ø 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 Ø pour le déplacer sans afficher.
Si on frappe la touche :, la couleur du point change.
La touche - efface tout, et l'on recommence.
On peut sortir à tout moment, en tapant BREAK.

***EXTENSIONS POSSIBLES*** : On pourra prévoir un sous-programme qui permette la sauvegarde du dessin réalisé pour une utilisation ultérieure, ajouter des possibilités pour tirer des traits, introduire le son, etc.

```
10 REM ARDOISE MAGIQUE
20 CLS 0
30 X=32:Y=15
40 S=0
50 C=1
60 SET(X,Y,C)
70 A$=INKEY$:IF A$="" THEN 70
80 IF A$="0" THEN 500
90 IF A$="-" THEN RUN
100 IF A$=":" THEN 400
110 R=VAL(A$)
120 F=-2*(R<5)-1
130 A=X:B=Y
140 ON (R+2*(F-1)) GOSUB 1000,2000,3000,4000
150 D=X>0 AND X<63 AND Y>0 AND Y<31
160 IF D=0 THEN GOSUB 5000
170 IF S=1 THEN RESET(A,B)
180 GOTO 60
400 C=C+1
410 IF C=9 THEN C=0
420 GOTO 60
500 S=ABS(S-1)
510 GOTO 60
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.

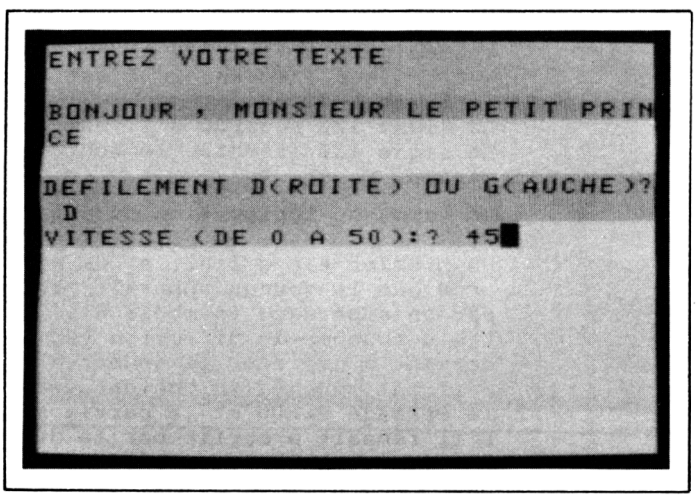
**LE JEU** : Un texte quelconque devra être introduit en début de programme. Il pourra être relativement long (4 ou 5 lignes) ; on choisira ensuite le sens et la vitesse de défilement à la suite de quoi le texte va apparaître (une seule ligne à la fois) et commencera à défiler, à la vitesse choisie, et dans le sens choisi.

**LE PROGRAMME** : Le texte sera introduit lettre par lettre aux lignes 5Ø et 75.
L'affichage sera tronqué à 3Ø caractères au plus, grâce à la ligne 14Ø.
Le défilement vers la gauche est préparé à la ligne 15Ø, et le défilement vers la droite à la ligne 16Ø.
Le texte sera affiché en ligne 14Ø, et la durée de l'affichage (vitesse de défilement) est choisie par la ligne 17Ø.
La ligne 19Ø 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 ENTRER VOTRE TEXTE, on introduira le texte à faire défiler, suivi de ENTER.
On obtient ensuite l'affichage DEFILEMENT A DROITE (D) OU A GAUCHE (G) auquel on répondra en tapant, soit D, soit G, suivi de ENTER.
Puis à la question VITESSE, on répondra en introduisant un petit nombre pour une vitesse rapide, et un grand pour une vitesse lente (par exemple, de 0 à 40 ou plus).
Le texte commencera à défiler dès que l'on aura frappé ENTER.
Si l'on désire changer les conditions du défilement, il suffit d'appuyer sur la touche 0 pour retrouver les questions SENS et VITESSE. On tapera BREAK 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 CLS 4:CLEAR 200
25 B$=""
30 PRINT"ENTREZ VOTRE TEXTE"
40 PRINT
50 A$=INKEY$:IF A$="" THEN 50
60 IF ASC(A$)=13 THEN 80
65 B$=B$+A$
70 PRINTA$;
75 GOTO 50
80 PRINT@160,"DEFILEMENT D(ROITE) OU G(AUCHE)";:INPUTS$
100 PRINT@224,"VITESSE (DE 0 A 50):";:INPUTV
110 V=51-V
120 CLS 8
130 T=LEN(B$)
140 PRINT@224," ";MID$(B$,1,30);" "
150 IF LEFT$(S$,1)="G" THEN B$=RIGHT$(B$,T-1)+LEFT$(B$,1)
160 IF LEFT$(S$,1)="D" THEN B$=RIGHT$(B$,1)+LEFT$(B$,T-1)
170 FOR J=0 TO 10*V
180 C$=INKEY$
190 IF C$="0" THEN 80
200 NEXT J
210 GOTO 140
```



# 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 Alice 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.

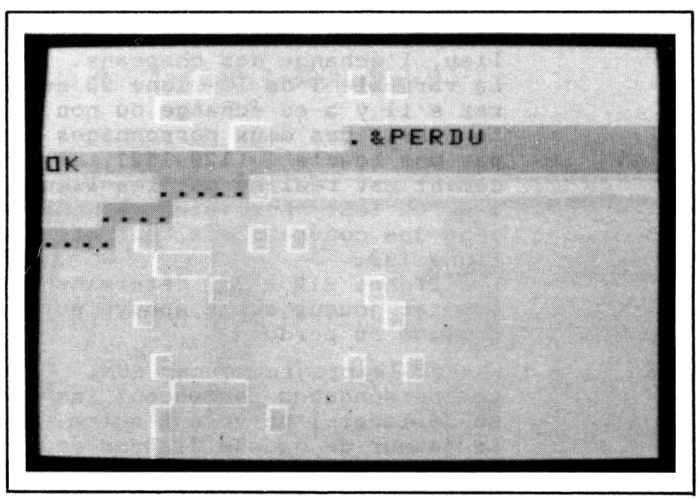
***LE PROGRAMME*** : La ligne 3Ø définit la position de départ du joueur.
La ligne 4Ø choisit la coordonnée "horizontale" de la case où Alice va tirer (liée à votre position), et la ligne 5Ø choisit la coordonnée "verticale" (arbitraire) de cette case.
Le déplacement du joueur est contrôlé par les lignes 9Ø et 1ØØ ; sa trajectoire est tracée par la ligne 7Ø et sa position actuelle (sur l'écran, un 8 est imprimé) par la ligne 11Ø.
La ligne 12Ø regarde s'il est touché ou non.
La ligne 13Ø 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 ampersand (symbole 8).
Les 4 touches de direction (5, 6, 7, 8) servent à déplacer le joueur.
S'il est touché, on obtient, sur l'écran, le message PERDU et la partie s'arrête.
S'il réussit à sortir par la droite de l'écran sans être touché, on obtient le message BRAVO, et la partie est également terminée.
On peut terminer à tout moment en tapant BREAK.

***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 obs-

tacles à contourner, ajouter des contrôles faire participer plusieurs joueurs, etc.

```
10 REM EVASION
20 CLS 4
30 X=7:Y=0
40 H=Y+RND(9)-5
45 IF H<0 THEN 40
50 K=RND(15)
55 IF H<5 OR H>31 THEN 70
60 PRINT@(32*K+H),"3";:REM COULEUR 3 ; SHIFT Q
70 PRINT@(32*X+Y),".";
80 A$=INKEY$
90 Y=Y-(A$="S")+(A$="Q")
100 X=X-(A$="W")+(A$="Z")
110 PRINT@(32*X+Y),"&";
115 IF Y<4 THEN 130
120 IF ABS(H-Y)+ABS(K-X)<=1 THEN PRINT"PERDU":SOUND140,15:END
130 IF Y<31 THEN 40
140 PRINT"BRAVO":SOUND200,9:SOUND220,7:SOUND180,9
```



# 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 des chapeaux lors du croisement.

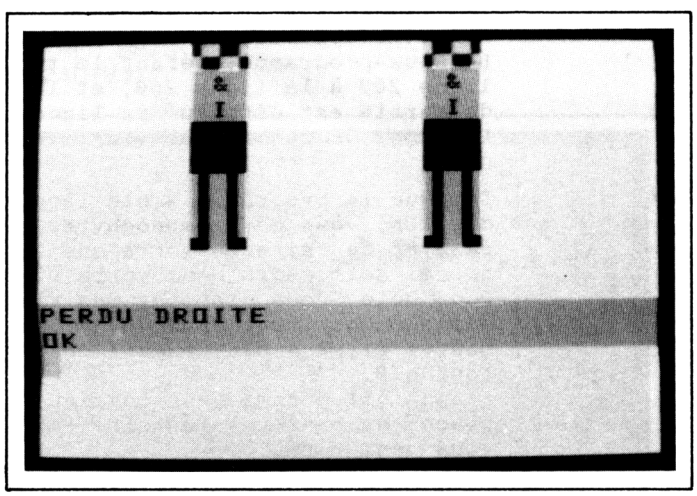
***LE PROGRAMME*** : Les lignes 2Ø et 3Ø contiennent le dessin des deux chapeaux (dans deux variables alphanumériques B$ et C$).
La ligne 4Ø contient, dans une autre variable A$, les symboles qui permettront de dessiner les personnages.
Les lignes 6Ø à 8Ø permettront, s'il y a lieu, l'échange des chapeaux.
La variable T de la ligne 9Ø sert à repérer s'il y a eu échange ou non.
Le dessin des deux personnages est réalisé par une boucle I (12Ø-15Ø), et leur déplacement est réalisé par les lignes 16Ø et 17Ø. Le test, permettant ou non le changement des couvre-chefs, est effectué en ligne 19Ø.
Les lignes 21Ø à 25Ø 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 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 B$="444":REM COULEUR 4 ; SHIFT F , PUIS SHIFT A , ET SHIFT F
30 C$="444":REM COULEUR 4 ; SHIFT A , PUIS SHIFT F , ET SHIFT A
40 A$=" &  I 0000000000000000":REM COULEUR 0 ; ESPACE , & ,
   2 ESPACES , I , ESPACE , 6 FOIS SHIFT Q , SHIFT Y , ESPACE ,
                                                    SHIFT G
45 REM ( SUITE ) SHIFT Y , ESPACE , SHIFT G , SHIFT S , ESPACE ,
                                                  ET SHIFT Z
50 X=1:Y=27:T=1
60 D$=C$
70 C$=B$
80 B$=D$
90 T=ABS(T-1)
100 CLS 8
110 PRINT@X,C$;:PRINT@Y,B$;
120 FOR I=1 TO 7
130 PRINT@(32*I+X),MID$(A$,3*I-2,3);
140 PRINT@(32*I+Y),MID$(A$,3*I-2,3);
150 NEXT I
160 X=X+1
170 Y=Y-1
180 FOR I=1 TO 300 : NEXT I
190 IF X=Y AND RND(5)<3 THEN 60
200 R$=INKEY$:IF R$="" THEN 100
210 IF (R$="1"AND T=1)OR(R$="2"AND T=0) THEN PRINT@320,"BRAVO
                                                  GAUCHE":GOTO 250
220 IF (R$="0"AND T=1) OR (R$="9"AND T=0) THEN PRINT@320,"BRAVO
                                                  DROITE":GOTO 250
230 IF R$="2" OR R$="1" THEN PRINT@320,"PERDU GAUCHE":SOUND140,
                                                       15:END
240 IF R$="9" OR R$="0" THEN PRINT@320,"PERDU DROITE":SOUND140,
                                                       15:END
250 SOUND220,9:SOUND240,7:SOUND200,9
```



# Combat aérien

**REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : réflexe (1 joueur).
Difficulté du jeu : difficile.

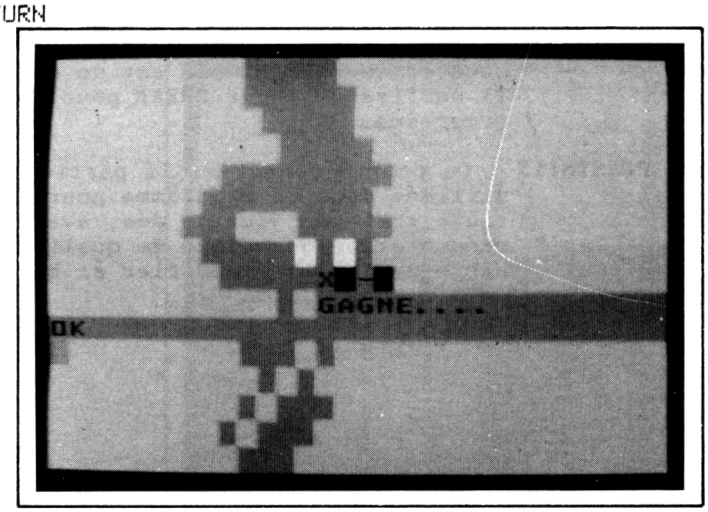
**LE JEU** : Au centre de l'écran, se trouve le viseur de votre appareil, et l'avion ennemi se déplace devant vous, d'une façon qui pourra vous sembler au début incohérente.
Vous pouvez vous déplacer dans toutes les directions, grâce aux flèches de mouvement, et vous allez essayer d'amener l'appareil ennemi au centre de votre viseur.
Vous tirerez 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).

**LE PROGRAMME** : La ligne 3Ø dessine l'appareil ennemi dont le déplacement aléatoire sera réalisé par les lignes 5Ø et 6Ø.
Les lignes 8Ø et 9Ø vous permettent le pilotage de votre appareil.
Les lignes 12Ø à 13Ø dessinent le viseur de votre appareil.
Le sous-programme, gérant le tir, va de la ligne 2ØØ à la ligne 26Ø, et le test de fin de partie est effectué en ligne 24Ø.
La ligne 75 regarde si vous avez tiré ou non.

**POUR JOUER** : Dès que le programme a été lancé par l'ordre RUN, vous devez manoeuvrer pour essayer de faire en sorte que l'appareil ennemi soit cadré dans votre viseur.
Pour cela, vous disposez des flèches de déplacement (touches 5, 6, 7, 8) et vous pouvez tirer à tout moment en pressant la touche Ø.
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 avez touché l'avion ennemi.
On peut aussi arrêter à tout moment en frappant BREAK.

***EXTENSIONS POSSIBLES*** : Il sera assez facile de faire un affichage au graphisme plus soigné, utilisant la couleur. On pourrait aussi envisager de stabiliser la position de l'avion ennemi pendant le tir (il suffit de modifier la ligne 75).
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 CLS 8
25 A=5:B=14
27 N=15
30 PRINT@(32*A+B),"000";:REM COULEUR 0 ; SHIFT Q , - , SHIFT Q
40 U=A:V=B
50 A=A+2*INT(RND(0)*2)-1
60 B=B+2*INT(RND(0)*2)-1
70 A$=INKEY$
75 IF A$="0" OR N<15 THEN GOSUB 200
80 A=A-2*(A$="W")+2*(A$="Z")
90 B=B-2*(A$="S")+2*(A$="Q")
100 GOSUB 300
120 PRINT@237,"555";:REM COULEUR 5 ; SHIFT Q , ESPACE , SHIFT Q
125 PRINT@270,"X";
130 PRINT@301,"555";:REM COULEUR 5 ; SHIFTQ , ESPACE , SHIFT Q
140 GOTO 30
200 PRINT@(32*N+24-N),"4";:REM COULEUR 4 ; SHIFT F
210 PRINT@(32*N+N+4),"4";:REM COULEUR 4 ; SHIFT A
215 PRINT@(32*N+24-N),"8";:REM COULEUR 8 ; SHIFT Q
220 PRINT@(32*N+N+4),"8";:REM COULEUR 8 ; SHIFT Q
225 N=N-1
230 IF N>8 THEN RETURN
240 IF(A=8ORA=9ORA=10)AND(B=13ORB=14ORB=15) THEN PRINT"GAGNE...."
                                :SOUND220,9:SOUND240,7:SOUND200,9:END
250 GOSUB 300
260 GOTO 27
300 IF A<0 THENA=0
310 IF A>15 THEN A=15
320 IF B<0 THEN B=0
330 IF B>28 THEN B=28
340 PRINT@(32*U+V),"   ";
350 RETURN
```



# Dessin de dés

***REFERENCES*** : Difficulté de programmation : difficile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

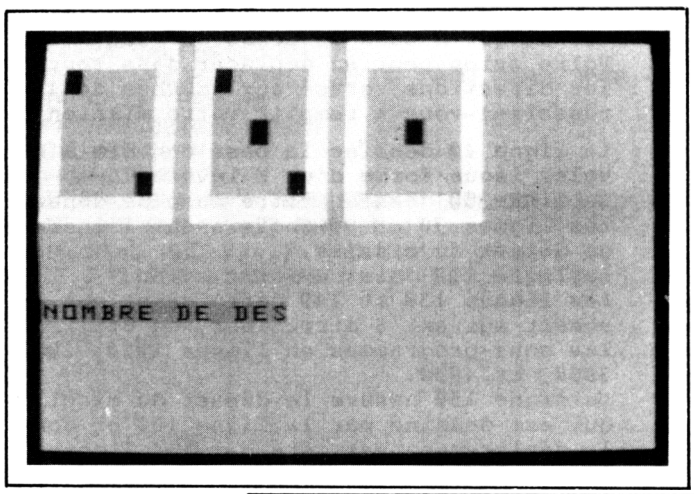
***LE JEU*** : Le programme proposé est surtout intéressant par son côté graphique puisqu'il permet de représenter, de façon très convenable, de 1 à 4 dés sur l'écran, en utilisant une technique de dessin un peu particulière.
Chaque réponse à la question NOMBRE DE DES provoque immédiatement un nouveau lancer du nombre de dés indiqué.

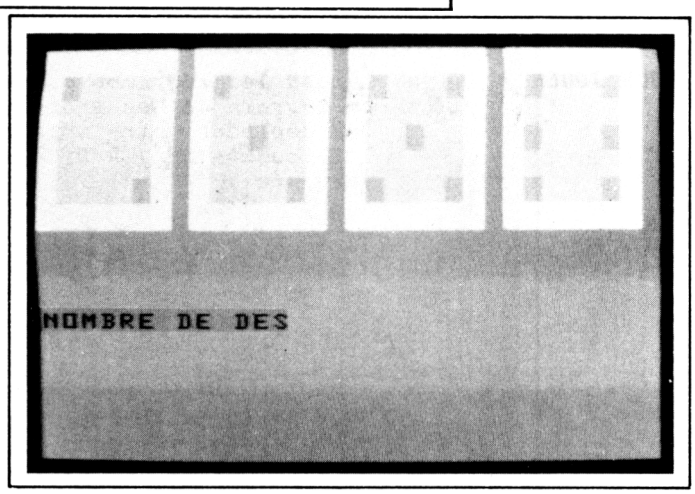
***LE PROGRAMME*** : Les lignes 4Ø et 5Ø permettent de contrôler s'il y a eu ou non action sur une touche de 1 à 4.
Les lignes 6Ø et 7Ø initialisent les deux variables de dessin des dés.
La boucle I (8Ø-17Ø) et les lignes 9Ø et 1ØØ dessine le cadre du dé.
La ligne 11Ø réalise le tirage, et la boucle J (12Ø-16Ø) 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 4. Les dés demandés sont alors représentés, et la question NOMBRE DE DES est de nouveau posée. Il suffira de taper BREAK pour sortir du programme.

***EXTENSIONS POSSIBLES*** : On pourra conserver la partie dessin utilisée dans ce programme pour réaliser toute sorte de jeux de dés, avec une représentation graphique de qualité convenable et facile à modifier si besoin était.

```
10 REM DESSIN DE DES
20 CLS 3
30 PRINT@320,"NOMBRE DE DES";
40 N$=INKEY$:IF N$="" THEN 40
50 N=VAL(N$):IF N<1 OR N>4 THEN RUN
55 CLS 3
60 A$="22222228228222222222882228":REM COULEURS 2 ET 8 ; 25 FOIS
                                                    SHIFT Q
70 B$="1213143245155255555"
80 FOR I=1 TO N
90 PRINT@(8*I-8),"5555555";:REM COULEUR 5 ; 7 FOIS SHIFT Q
100 PRINT@(192+8*I-8),"5555555";:REM COULEUR 5 ; 7 FOIS SHIFT Q
110 X=RND(6)
120 FOR J=1 TO 5
130 IF J<=3 THEN S=5*VAL(MID$(B$,3*(X-1)+J,1))-4
140 IF J<=3 THEN PRINT@(32*(2*J-1)+8*I-7),MID$(A$,S,5);
150 PRINT@(32*J+8*I-8),"5";:REM COULEUR 5 ; SHIFT Q
155 PRINT@(32*J+8*I-2),"5";:REM COULEUR 5 ; SHIFT Q
160 NEXT J
165 PRINT@(64+8*I-7),"22222";:REM COULEUR 2 ; 5 FOIS SHIFT Q
167 PRINT@(128+8*I-7),"22222";:REM COULEUR 2 ; 5 FOIS SHIFT Q
170 NEXT I
180 GOTO 30
```



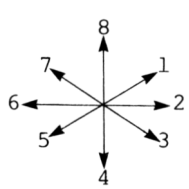
# Le jeu du missile

**REFERENCES** : Difficulté de programamtion : moyenne.
Catégorie de jeu : contre Alice.
Difficulté du jeu : moyenne.

**LE JEU** : Alice 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 (représentée par un H inversé), et de revenir sain et sauf sur votre base ; mais, dès que Alice 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 4Ø dessine la base ennemie à survoler (sous forme d'un H inversé).
La ligne 5Ø dessine votre base de départ.
Les lignes 7Ø et 9Ø choisissent l'instant de départ du missile.
La ligne 11Ø dessine votre avion.
Les lignes 13Ø et 14Ø définissent son mouvement suivant 8 directions, en utilisant les sous-programmes en lignes 1ØØØ, 2ØØØ, 3ØØØ, et 4ØØØ.
La ligne 15Ø assure le départ du missile qui est dessiné par la ligne 16Ø et dont le déplacement est géré par les lignes 17Ø et 18Ø.
Les lignes 19Ø à 22Ø 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 :

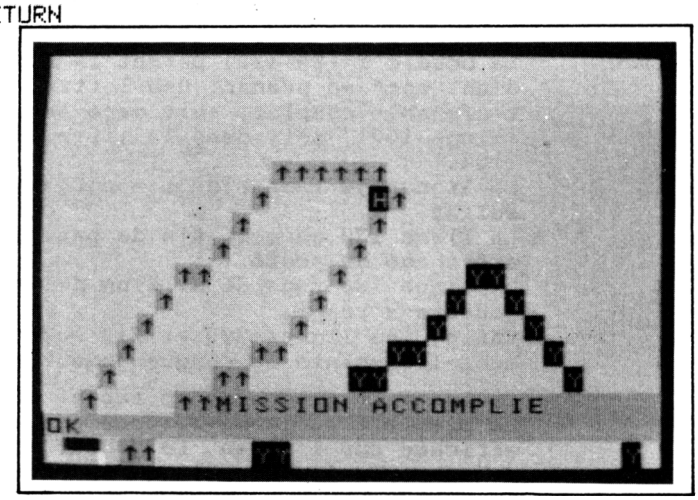



Vous devez lui faire contourner le territoire ennemi (si vous, ou le missile, touchez ce territoire, une deuxième partie

commence). Si le missile vous détruit, vous obtenez un message TOUCHE, et la partie s'arrête.
Taper BREAK 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 CLS 3
30 M=0
40 PRINT@177,"h";
50 PRINT@480,"444";:REM COULEUR 4 ; 3 FOIS SHIFT X
60 U=15:X=14
70 S=INT(RND(0)*U)
80 Y=1
90 S=S-1
100 V=30
110 PRINT@(32*X+Y),"^";
120 A$=INKEY$:IF A$="" THEN 120
130 F=-2*(VAL(A$)<5)-1
140 ON VAL(A$)+2*(F-1) GOSUB 1000,2000,3000,4000
150 IF S<>0 THEN 90
160 PRINT@(32*U+V),"y";
170 U=U-SGN(U-X)
180 V=V-SGN(V-Y)
190 IF U=X AND V=Y THEN PRINT"TOUCHE":SOUND140,15:END
200 IF U=5 AND Y>=17 THEN M=1
210 IF X<=5 AND Y>=17 THEN M=1
220 IF M=1 AND X=15 AND Y<=3 THEN PRINT@425,"MISSION ACCOMPLIE"
                                                    :SOUND220,9:SOUND
240,7:SOUND200,9:END
230 GOTO 110
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 Alice 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.

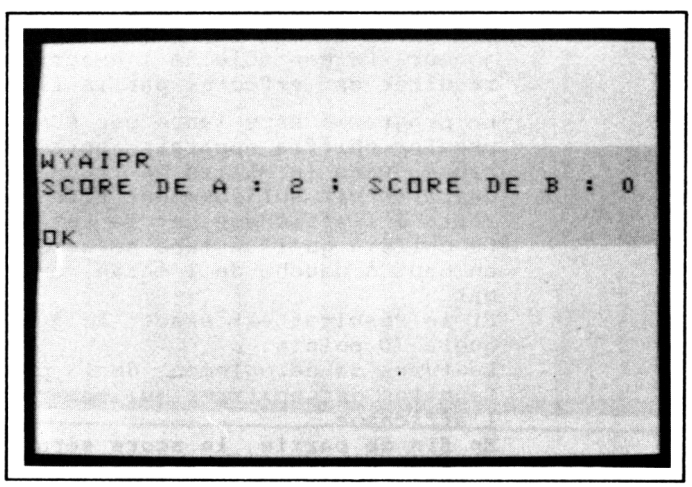
**LE PROGRAMME** : La ligne 3Ø contient une liste V$ des voyelles courantes, et la ligne 4Ø une liste C$ des consonnes courantes.
Il sera possible de modifier ces deux listes pour augmenter, par exemple, la fréquence d'apparition de certaines lettres (on pourra ainsi mettre deux E dans V$ ou deux R dans C$, etc.
La ligne 5Ø définit la longueur du mot (en principe, de 3 à 6 lettres, mais on peut également modifier cette longueur).
La boucle I (8Ø-13Ø) permet la confection d'un "mot" en prenant une lettre, soit dans l'alphabet complet, soit dans la liste C$ (ligne 1ØØ), soit dans la liste V$ (ligne 11Ø).
La ligne 15Ø temporise l'apparition du résultat.
La ligne 17Ø gère la fin de partie, avec affichage du score.
La ligne 18Ø regarde si l'un des deux joueurs a réagi.
Enfin, les lignes 2ØØ et 21Ø 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 Ø.
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 ENTER.
Les points sont alors calculés par Alice, et la série de lettres suivantes 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 A=0:B=0
30 V$="AEIOU"
40 C$="BCDGLMNPRSTV"
50 X=RND(4)+2
60 M$=""
70 CLS 3
80 FOR I=1 TO X
90 E$=CHR$(RND(26)+64)
100 IF RND(0)<0.4 THEN E$=MID$(C$,INT(LEN(C$)*RND(0)+1),1)
110 IF RND(0)<0.5 THEN E$=MID$(V$,INT(LEN(V$)*RND(0)+1),1)
120 M$=M$+E$
130 NEXT I
140 PRINT@128,M$
150 FOR J=1 TO 800: NEXT J
160 A$=INKEY$
170 IF A$="P" THEN PRINT@160,"SCORE DE A :"A"; SCORE DE B :"B
                    :SOUND200,9:SOUND22 0,7:SOUND180,9:END
180 IF A$<>"0" AND A$<>"1" THEN 50
190 PRINT@160,"J(USTE) OU F(AUX) ";:INPUTR$
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
```



# Adresse

**REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (pour 1 joueur).
Difficulté du jeu : difficile.

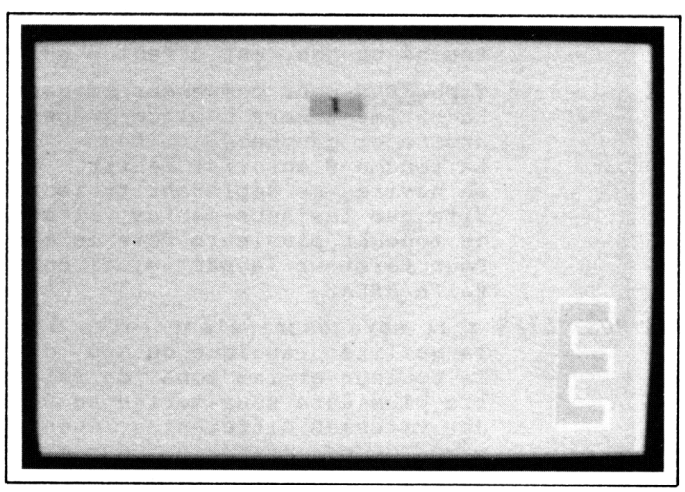
**LE JEU** : Des chiffres, entre 0 et 9, vont apparaître au hasard, en divers emplacements de l'écran. Ils restent affichés pendant un temps très court. Le joueur doit réussir à appuyer sur la touche numérique correspondante, au moment où le chiffre est affiché. Le chiffre tapé par le joueur sera alors 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 40 et 50 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 Alice est choisi aux lignes 60 et 70, et le chiffre est tiré en ligne 80.
La ligne 90 réalise l'impression, et la ligne 100 détermine la durée d'affichage.
La ligne 120 définit la durée du jeu, et la fin de partie est traitée par la ligne 140. La boucle I (210-240) réalise le dessin grand format du chiffre frappé par le joueur. Le contrôle de l'exactitude du résultat est effectué par la ligne 250.

**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 BREAK.

***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.

```
10 REM ADRESSE
15 CLS 8
20 T=100
30 S=0
40 C$="33338888338":REM COULEURS 3 ET 8 ; FAIRE 11 FOIS SHIFT Q
                                    DANS LES COULEURS INDIQUEES
50 B$="1444133333131211313144133121311214113333141411413 1"
60 X=RND(15)-1
70 Y=RND(15)+14
80 N=RND(10)-1
90 PRINT@(32*X+Y),N;
100 FOR I=1 TO RND(400)+200:NEXT I
110 A$=INKEY$:IF A$<>"" THEN 200
120 T=T-N
130 PRINT@(32*X+Y),"888";:REM COULEUR 8 ; 3 FOIS SHIFT Q
140 IF T>=0 THEN 60
150 CLS 8
160 PRINT"SCORE"S"/ 100"
170 SOUND200,9:END
200 U=VAL(A$)
210 FOR I=1 TO 5
220 A=3*VAL(MID$(B$,5*U+I,1))-2
230 PRINT@(32*I+2),MID$(C$,A,3);
240 NEXT I
250 IF U=N THEN S=S+10
260 GOTO 120
```



# Sous-marin

**REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : réflexes (1 seul joueur).
Difficulté du jeu : facile.

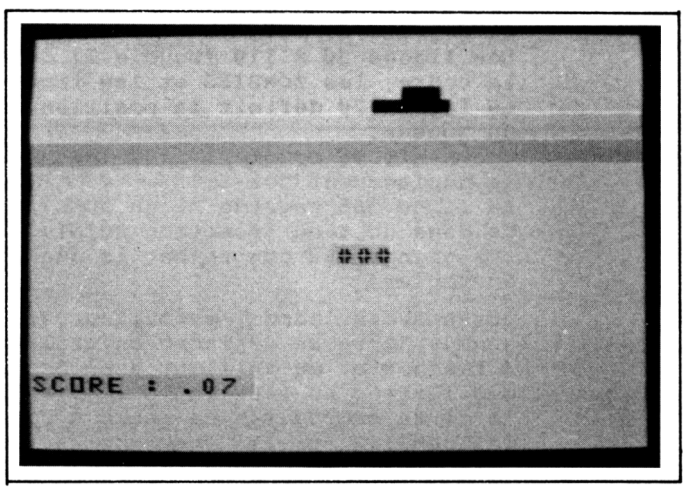
**LE JEU** : Vous manoeuvrez en surface un chasseur de sous-marins, et des sous-marins passent très régulièrement en dessous de votre unité, à différentes profondeurs.
Vous pouvez, à tout moment, larguer des grenades sous-marines et vous pouvez même envisager de toucher plusieurs fois le même sous-marin.
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 5Ø permet à Alice de choisir la profondeur du sous-marin.
La ligne 8Ø dessine votre bâtiment.
Les lignes 9Ø et 1ØØ affichent le score et dessinent la surface de l'eau.
La ligne 11Ø dessine le sous-marin.
Le tir est effectué par un programme spécifique placé aux lignes 23Ø à 27Ø.
Le déplacement de votre navire est réalisé par les lignes 2ØØ et 21Ø.
Le test, permettant de savoir si l'on a touché ou non, est effectué en ligne 13Ø.

**POUR JOUER** : Taper RUN pour commencer la partie.
La bâtiment sera déplacé grâce aux flèches droite et gauche (S ou D).
La touche Ø 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 BREAK.

**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 P=0:C=15
30 CLS 4
40 T=27
50 S=RND(6)+4
60 P=P+0.01
70 X=1
80 PRINT@(64+C),"4444444444";:REM COULEUR 4 ; 3 FOIS SHIFT Q ,
   SHIFT R , SHIFT S , SHIFT B , SHIFT Z , ET 3 FOIS SHIFT Q
90 PRINT@416,"SCORE :"P;
100 PRINT@96,"3333333333333333333333333333333333":REM COULEUR 3 ;
                                                    32 FOIS SHIFT Q
110 PRINT@(32*S+T),"555";:REM COULEUR 5 ; SHIFT X , PUIS SHIFT D
                                                    ET SHIFT X
120 IF X=1 THEN Y=C+2*X
130 IF S=X AND(Y=T+1 OR Y=T+2) THENP=P+1:PRINT@(32*S+T),"###";
                                                    :SOUND140,15
140 T=T-1:IF P=99 THEN PRINT@320,"C'EST FINI";:SOUND200,9
                                :SOUND220,7 :SOUND180,9: END
145 PRINT@(32*S+T+1),"444";:REM COULEUR 4 ; 3 FOIS SHIFT Q
150 IF T<0 THEN 40
200 A$=INKEY$
210 C=C-3*((A$="S"AND C<27)-(A$="Q"AND C>0))
220 IF A$<>"0" AND X=1 THEN 80
230 PRINT@(32*X+Y),"4";:REM COULEUR 4 ; SHIFT Q
240 X=X+1
250 IF X>S THEN 70
260 PRINT@(32*X+Y),"0";
270 GOTO 80
```



# 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 0). 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.

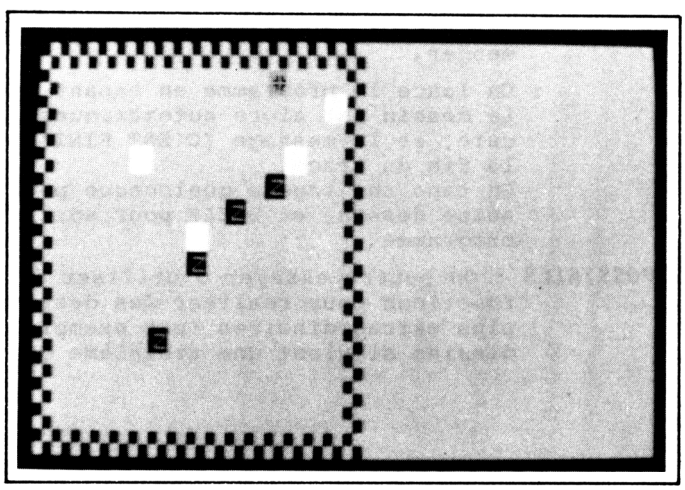
***LE PROGRAMME*** : La position des ZOMBIES est définie par un tableau Z non dimensionné (c'est inutile en dessous de 10).
Les lignes 3Ø à 11Ø (boucle I) dessinent le cadre, les ZOMBIES et les trous noirs.
La ligne 12Ø définit la position initiale du joueur.
Une deuxième boucle I (13Ø-19Ø) réalise le déplacement des Z.
La ligne 165 regarde si un ZOMBIE est tombé dans un trou (fonction POINT).
Les lignes 26Ø 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 manoeuvrer de façon à faire venir les ZOMBIES sur les trous noirs, sans se faire lui-même coincer.
Pour arrêter la partie, il suffira de taper BREAK.

***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 (pratiquement inexistants), de placer le joueur de façon plus aléatoire, éventuellement de modifier la règle.
On pourra aussi, bien sûr, améliorer la qualité du dessin.

```
10 REM ZOMBIES
20 CLS 3
30 FOR I=0 TO 15
40 PRINT@I+1,"3";:REM COULEUR 3 ; SHIFT F
45 PRINT@32*I,"3";:REM COULEUR 3 ; SHIFT F
50 PRINT@(32*I+16),"3";:REM COULEUR 3 ; SHIFT F
60 Z=RND(15)
70 IF RND(0)<0.4 THEN PRINT@(32*Z+I),"5";:REM COULEUR 5 ; SHIFT Q
80 S=RND(10)
90 Z(INT(I/2)+1)=4+S
100 PRINT@480+I,"3";:REM COULEUR 3 ; SHIFT F
110 NEXT I
120 K=RND(14)+1
130 FOR I=1 TO 4
135 IF Z(I)=0 THEN 190
140 PRINT@(32*Z(I)+Z(I+4)),"3";:REM COULEUR 3 ; SHIFT Q
150 Z(I)=Z(I)-SGN(Z(I)-S)
160 Z(I+4)=Z(I+4)-SGN(Z(I+4)-K)
165 IF POINT(2*Z(I),2*Z(I+4))=5 THEN Z(I)=0:GOTO 190
170 PRINT@(32*Z(I)+Z(I+4)),"z";
180 IF Z(I)=S AND Z(I+4)=K THEN PRINT@310,"PERDU":SOUND150,20:END
190 NEXT I
200 PRINT@(32*S+K),"#";
210 FOR I=1 TO 1500 :A$=INKEY$:IF A$="Q" OR A$="S" OR A$="Z"
                                              OR A$="W" THEN 220
215 NEXT I
220 PRINT@(32*S+K),"3";:REM COULEUR 3 ; SHIFT Q
230 IF Z(1)=0 AND Z(2)=0 AND Z(3)=0 AND Z(4)=0 THEN PRINT"BRAVO"
                                                  :SOUND230,15:END
235 K=K-2*((A$="S")-(A$="Q"))
240 IF K<=0 OR K>=16 THEN PRINT@310,"PERDU":SOUND150,20:END
250 S=S-2*((A$="W")-(A$="Z"))
260 IF S<=0 OR S>=16 THEN PRINT@310,"PERDU":SOUND150,20:END
270 GOTO 130
```



# Mathieu

***REFERENCES*** : Difficulté de programmation : facile
Catégorie de jeu : dessin aléatoire.
Difficulté du jeu : aucune.

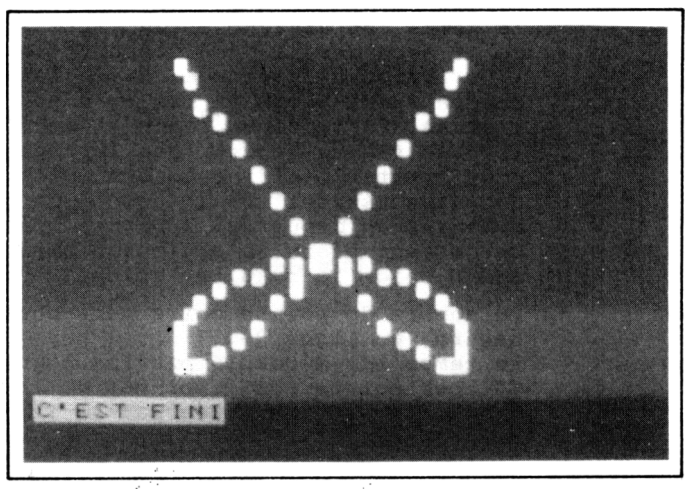
***LE JEU*** : On utilise la définition d'une courbe en coordonnées polaires, et la fonction SET, pour obtenir sur l'écran les dessins aléatoires (de type fleur) plus ou moins élégants;
Les deux paramètres principaux A et B pourront être entrés à la main (instruction INPUT) pour faire une étude plus systématique des diverses courbes possibles.

***LE PROGRAMME*** : La ligne 3Ø définit les 2 paramètres H et K qui déterminent la dimension de la courbe (occupation de l'écran) : on pourra bien entendu les modifier.
La ligne 4Ø tire deux variables aléatoires A et B, entre 0 et 3. L'allure des courbes dépend de ces 2 variables ; on peut modifier leur intervalle de valeur, ou encore les entrer à la main (4Ø INPUT A et 45 INPUT B).
La boucle I (6Ø-11Ø) réalise le tracé point par point, le rayon R est calculé en ligne 7Ø et le passage en coordonnées cartésiennes est effectué aux lignes 8Ø et 9Ø.
La ligne 5Ø tire la couleur des points, qui sont imprimés en ligne 1ØØ.
La ligne 12Ø affiche un message de fin, et les lignes 13Ø et 14Ø permettent de recommencer.

***POUR JOUER*** : On lance le programme en tapant RUN.
Le dessin est alors automatiquement exécuté, et le message "C'EST FINI" signale la fin du tracé.
On tape une touche quelconque pour un autre dessin, et BREAK pour sortir du programme.

***EXTENSIONS POSSIBLES*** : On pourra essayer d'utiliser d'autres fonctions pour réaliser des dessins encore plus extraordinaires (par exemple, des dessins simulant une troisième dimension).

```
10 REM MATHIEU
20 CLS 0
30 H=3:K=5
40 A=3*RND(0):B=3*RND(0)
50 C=RND(8)
60 FOR I=0 TO 6.28 STEP 9.42/180
70 R=H*SIN(3*I)
80 X=30+INT(K*R*COS(A*I))
90 Y=15+INT(K*R*SIN(B*I))
100 SET(X,Y,C)
110 NEXT I
120 PRINT@416,"C'EST FINI";:SOUND200,9:SOUND220,7:SOUND180,9
130 A$=INKEY$:IF A$="" THEN 130
140 RUN
```



# Diago

**REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : réflexes (1 seul joueur)
Difficulté du jeu : difficile.

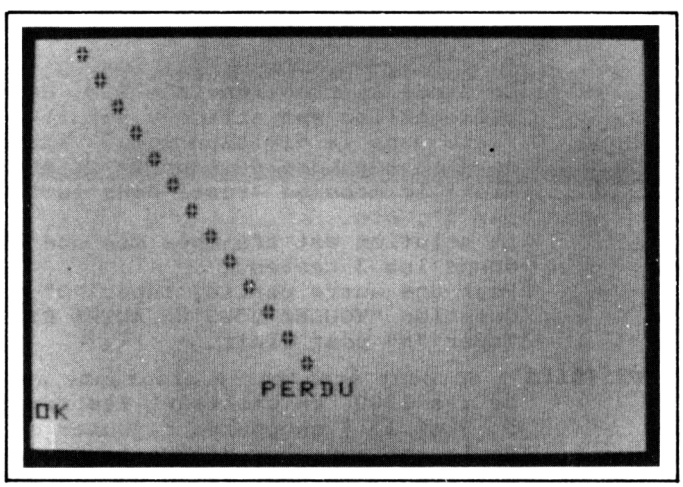
**LE JEU** : On lance le programme par RUN et on obtient un premier affichage qui vous renseigne sur la performance à réaliser (par exemple, 5 points noirs ou 13 points noirs...) ; puis le temps est affiché et décompté, à partir de 10 secondes. Le jeu commence lorsque l'on arrive à 0 seconde. Des points noirs apparaissent alors en bas et à gauche de l'écran, formant une ligne diagonale, et il vous faudra appuyer sur la touche espace 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 3Ø.
Le temps est décompté par les lignes 4Ø à 9Ø. Les points sont imprimés et la ligne diagonale est fabriquée par la ligne 11Ø.
La pression de la touche espace est détectée en ligne 15Ø, et la ligne 16Ø contrôle la fin de partie (aucun réflexe).
Les affichages de fin sont exécutés par les lignes 18Ø à 21Ø.

**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.
A 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 espace.
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 CLS 4
30 X=RND(12)+3
40 S=10
45 PRINT"LE BON REFLEXE EST A"X
50 PRINT@32,"attention :"S"secondes";
60 S=S-1
70 SOUND230,10:FOR I=1 TO 700:NEXT I
80 CLS 4
90 IF S<>0 THEN 45
100 U=0
110 PRINT@480+U,"#";
120 FOR I=1 TO 70:NEXT I
130 U=U+1
140 A$=INKEY$
150 IF A$=" " THEN 180
160 IF U>16 THEN PRINT@460,"PERDU";:SOUND150,20:END
170 PRINT:GOTO110
180 IF U<>X THEN PRINT@460,"PERDU";:SOUND150,20:END
190 PRINT@374,"BRAVO":SOUND200,9:SOUND220,7:SOUND180,9
200 INPUT"AUTRE ESSAI (O/N)";R$
210 IF LEFT$(R$,1)="O" THEN RUN
```



# Décomaque

**REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : solution du programme Nicomaque.
Difficulté du jeu : aucune.

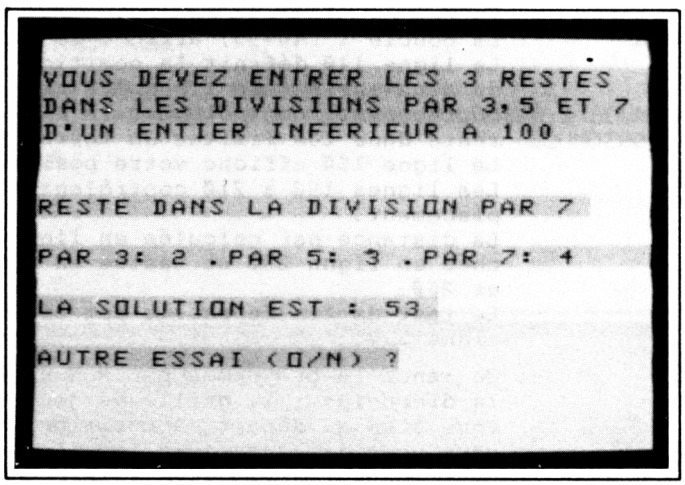
**LE JEU** : C'est une réplique au programme Nicomaque, proposé dans les pages précédentes ; vous entrez les restes successivement dans les divisions par 3, 5 et 7 d'un entier compris entre 1 et 1ØØ, et Alice calcule et affiche cet entier. Le procédé retenu peut d'ailleurs être généralisé (à vous de trouver comment).

**LE PROGRAMME** : Les lignes 3Ø à 5Ø affichent quelques lignes d'explication.
La ligne 8Ø attend que vous frappiez un reste, qui sera refusé s'il est incompatible avec la division par le nombre affiché.
La ligne 9Ø emmagasine les restes dans une variable non dimensionnée A (S).
La ligne 12Ø réaffiche les restes entrés, et les lignes 13Ø et 14Ø calculent la solution, qui est imprimée par la ligne 15Ø.
Les lignes 16Ø à 18Ø 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)?
Taper "N" pour finir.

**EXTENSIONS POSSIBLES** : On peut étendre le programme aux entiers de 1 à 1000, 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. On peut faire un jeu pour plusieurs joueurs en introduisant une temporisation avant l'affichage du résultat, etc.

```
10 REM DECOMAQUE
20 CLS 8
30 PRINT@32,"VOUS DEVEZ ENTRER LES 3 RESTES"
40 PRINT"DANS LES DIVISIONS PAR 3,5 ET 7"
50 PRINT"D'UN ENTIER INFERIEUR A 100"
60 I=3 : S=1
70 PRINT@192,"RESTE DANS LA DIVISION PAR"I;
80 A$=INKEY$:IF A$="" OR A$<"0" OR A$>CHR$(47+I) THEN 80
90 A(S)=ASC(A$)-48
100 S=S+1:I=I+2
110 IF I<8 THEN 70
120 PRINT@256,"PAR 3:"A(1)".PAR 5:"A(2)".PAR 7:"A(3)
130 T=A(1)*70+A(2)*21+A(3)*15
140 IF T>105 THEN T=T-105:GOTO 140
150 PRINT@320,"LA SOLUTION EST :"T;
155 SOUND220,9:SOUND240,7:SOUND200,9
160 PRINT@384,"AUTRE ESSAI (O/N) ?";
170 A$=INKEY$:IF A$="" THEN 170
180 IF A$="O" THEN RUN
```



# 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.

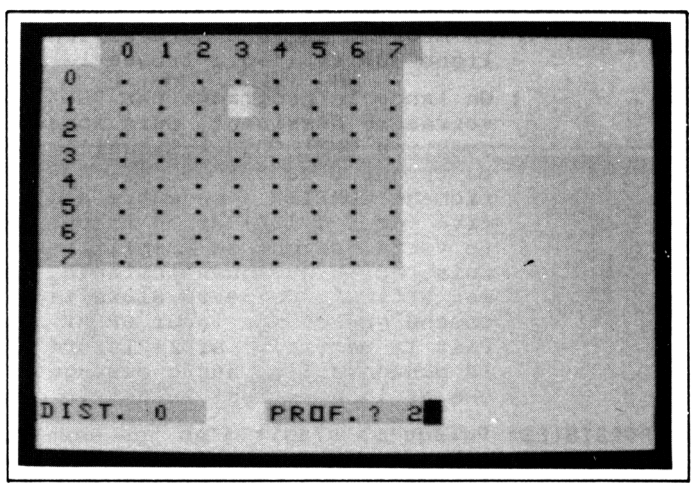
***LE PROGRAMME*** : La boucle I (4Ø-95) affiche le terrain.
La ligne 11Ø définit la position initiale du sous-marin ; la ligne 12Ø permet son déplacement, contrôlé par la ligne 13Ø (on reste dans les limites du terrain).
La ligne 15Ø affiche votre position.
Les lignes 19Ø à 21Ø contrôlent votre déplacement.
La distance est calculée en ligne 22Ø, affichée en ligne 23Ø et testée en lignes 25Ø et 26Ø.
Le test de fin de partie est effectué à la ligne 31Ø.

***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 le programme à tout moment en utilisant la touche BREAK.

***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 CLS 3
30 INPUT"DIMENSION DE LA GRILLE";H
35 CLS 3
40 FOR I=0 TO H
50 PRINT@(2*I+3),I;
60 PRINT@(32*I+32),I;
70 FOR J=0 TO H
80 PRINT@(32*I+2*J+35),".";
90 NEXT J
95 NEXT I
100 I=1:X=0:Y=0
110 V=RND(H)-1:U=RND(H)-1
120 U=U+I:V=V+I
130 IF (U<0 OR U>H OR V<0 OR V>H) THEN I=-I
140 PRINT@(32*A+2*B+36),".";
150 PRINT@(32*X+2*Y+36),"4";:REM COULEUR 4 ; SHIFT Q
190 A$=INKEY$:IF A$="" THEN 190
195 A=X:B=Y
200 Y=Y-(A$="S"AND Y<H)+(A$="Q" AND Y>0)
210 X=X-(A$="W" AND X<H)+(A$="Z" AND X>0)
220 D=INT(SQR(ABS(X-V)^2+ABS(Y-U)^2))
230 PRINT@448,"DIST."STR$(D)+"  ";
240 IF D=0 THEN 300
250 IF D<3 THEN 120
260 GOTO 140
300 PRINT@460,"PROF.";:INPUTD
310 IF D=RND(3)+1 THEN PRINT"BRAVO":SOUND220,9:SOUND240,7
                                              :SOUND200,9:END
320 GOTO 110
```



# Le lézard d'Alice

**REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez difficile.

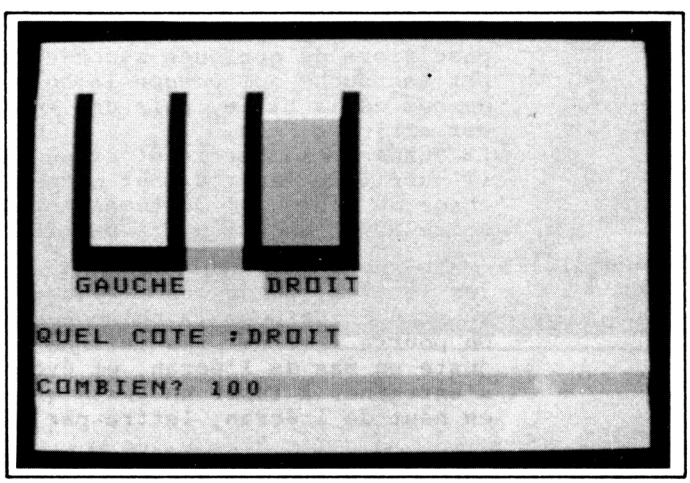
**LE JEU** : Il s'inspire très vaguement d'une idée de LEWIS CAROLL, dans "Alice aux pays des Merveilles". Dans l'histoire originale, Alice grandissait ou diminuait, alors que dans le jeu, pour des raisons d'encombrement de la mémoire, c'est un lézard qui change de taille.
Deux verres apparaissent sur 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 (40-60) dessine les verres.
Les deux boucles emboîtées I (140-180) et J (150-170) les remplissent avec la quantité indiquée en ligne 130.
La variation de taille du lézard est définie par la ligne 220 avec contrôle par les lignes 230 et 240.
La boucle I (250-270) dessine le corps du lézard, les lignes 280 à 320 dessinent l'oeil et les pattes.
Un message gagnant sera affiché par la ligne 330 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 COMBIEN ; on entre alors une quantité (de 0 à 130) et on frappe ENTER.
Le verre désigné se remplit sous vos yeux puis, après quelques instants, le lézard est affiché. On devra alors taper une touche quelconque (sauf BREAK qui arrêterait la partie) ; si le lézard n'avait pas la bonne taille, aucun message n'est affiché et l'on revient au dessin des verres.

**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 aux pays des merveilles, les idées n'y manquent pas.

```
10 REM LE LEZARD D'ALICE
20 CLS
30 N=18:S=6
40 FOR I=1 TO 6
50 PRINT@32*I+34,"0000000000000000";:REM COULEUR 0 ; SHIFT B ,
   4 FOIS ESPACE , SHIFT B , 3 FOIS ESPACE , SHIFT B , 4 FOIS
                                                        ESPACE
55 REM ( SUITE ) SHIFT B
60 NEXT I
70 PRINT@258,"0000000000000000";:REM COULEUR 0 ; 6 FOIS SHIFT B ,
                                 3 FOIS ESPACE , 6 FOIS SHIFT B
80 PRINT@290,"GAUCHE";:PRINT@300,"DROIT";
90 PRINT@352,"QUEL COTE :";
100 A$=INKEY$:IF A$<>"D" AND A$<>"G" THEN 100
110 U=S-18*(A$="D")
125 IF A$="G" THEN PRINT"GAUCHE";
127 IF A$="D"THEN PRINT"DROIT";
130 PRINT@416,"COMBIEN";:INPUTQ
140 FOR I=1 TO SQR(Q)
150 FOR J=0 TO 7
160 SET(U+J,16-I,3)
170 NEXT J
180 NEXT I
190 A$=INKEY$:IF A$="" THEN 190
200 CLS 0
210 H=-2*(RND(0)<0.5)-1
220 N=N+H*INT(SQR(Q))
230 IF N<12 THEN N=12
240 IF N>60 THEN N=60
250 FOR I=0 TO N+3
260 SET(I,S,4)
270 NEXT I
280 SET(N-INT((SQR(N))/2),3,4)
290 SET(SQR(N),7,4)
300 SET(SQR(N)+2,7,4)
310 SET(N-SQR(N),7,4)
320 SET(N-SQR(N)-2,7,4)
330 IF N=28 THEN PRINT@416,"BRAVO":SOUND200,9:SOUND230,7
                                              :SOUND180,9:END
335 SOUND180,3
340 A$=INKEY$:IF A$="" THEN 340
350 GOTO 10
```



# Clavier

***REFERENCES*** : Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexe (apprentissage du clavier).
Difficulté du jeu : difficile.

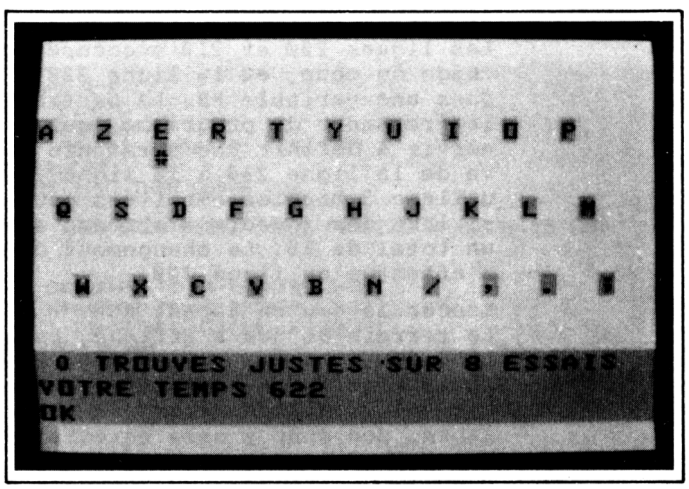
***LE JEU*** : Une reproduction du clavier de votre ordinateur apparaît sur l'écran ; un point apparaît sous l'une des lettres et indique le caractère correspondant qu'il convient de frapper au clavier. Plus on frappe vite et plus le point se déplace rapidement (le temps écoulé depuis le début s'affiche en fin de partie).

***LE PROGRAMME*** : Une variable A$ dimensionnée en ligne 30 contiendra la liste des touches concernées.
Les valeurs des touches sont entrées de la ligne 40 à la ligne 90, par appel d'un sous-programme situé en .400.
Les boucles emboîtées J (100-140) et I (110-130) dessinent le clavier.
La ligne 150 définit la position de la balle, imprimée par la ligne 160.
La touche frappée est testée aux lignes 180, 190 et 200.
Le nombre d'essais est totalisé en ligne 210 et la balle est effacée par la ligne 220.
Les affichages de fin sont réalisés aux lignes 300 et 320.

***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 la balle, puis une autre colonne 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 juste 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 CLS 8
25 T=0:S=0:P=0
30 DIM A$(10,3)
40 S$="AZERTYUIOP":J=1
50 GOSUB 400
60 S$="QSDFGHJKLM":J=2
70 GOSUB 400
80 S$="WXCVBN/,.;":J=3
90 GOSUB 400
100 FOR J=1 TO 3
110 FOR I=1 TO 10
120 PRINT@(96*J+3*I+J-4),A$(I,J);
130 NEXT I
140 NEXT J
150 X=RND(10):Y=RND(3)
160 PRINT@(96*Y+3*X+Y+28),"#";
170 FOR F=1 TO 100+RND(200):A$=INKEY$
175 T=T+1
180 IF A$="" THEN NEXT F
190 IF A$=A$(X,Y) THEN S=S+1
195 SOUND220,2
200 IF A$="0" THEN 300
210 P=P+1
220 PRINT@(96*Y+3*X+Y+28),"8";:REM COULEUR 8 ; SHIFT Q
230 GOTO 150
300 PRINT@384,S"TROUVES JUSTES SUR"P"ESSAIS"
310 PRINT"VOTRE TEMPS"T
320 END
400 FOR I=1 TO 10: A$(I,J)=MID$(S$,I,1):NEXT I
410 RETURN
```



# Le jeu du 15

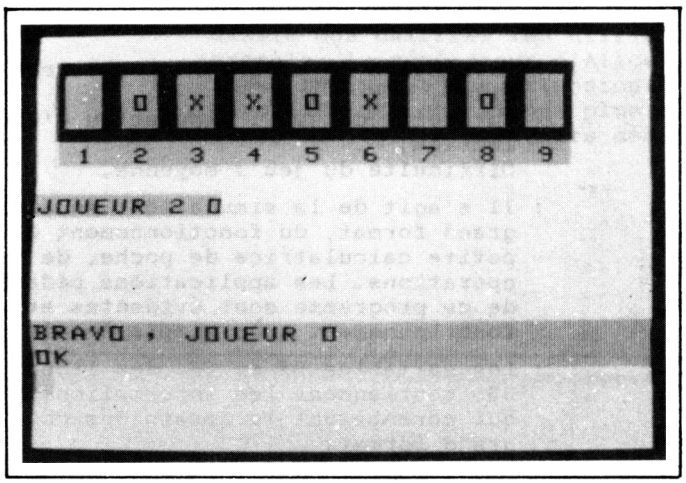
**REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

**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 trois jetons, arrive à faire un total de 15, a gagné.

**LE PROGRAMME** : La boucle I (5Ø-8Ø) dessine le terrain de jeu, et les lignes 9Ø à 11Ø s'occupent de la gestion du tour des joueurs (numéro 1 ou 2 pour chaque joueur, et jeton O ou X correspondant).
La ligne 12Ø attend le coup joué.
La boucle I (13Ø-15Ø) vérifie la validité du coup.
La ligne 16Ø teste la fin de partie, en cas de match nul, et la ligne 17Ø refuse les coups non conformes (avec retour à la situation initiale, ligne 9Ø).
Les lignes 2ØØ et 21Ø s'occupent de l'affichage du coup, et la ligne 22Ø 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 23Ø à la ligne 29Ø : 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 3ØØ.

**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.

***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.
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 JEU DU 15
15 CLS 8
20 F=0
30 R$="123456789"
40 D$="":X$=""
50 FOR I=0 TO 24 STEP 3
55 PRINT@33+I,"000";:REM COULEUR 0 ; SHIFT V , PUIS SHIFT X ET SHIFT C
60 PRINT@65+I,"000";:REM COULEUR 0 ; SHIFT G , ESPACE , SHIFT Y
65 PRINT@97+I,"000";:REM COULEUR 0 ; SHIFT Z , PUIS SHIFT R ET SHIFT S
70 PRINT@129+I,INT((I+3)/3);
80 NEXT I
90 PRINT@192,"JOUEUR"F+1;
100 IF F=1 THEN PRINT"0";
110 IF F=0 THEN PRINT"X";
120 A$=INKEY$:IF A$<"1" OR A$>"9" THEN 120
130 FOR I=1 TO LEN(R$)
140 IF A$=MID$(R$,I,1) THEN GOSUB 200
150 NEXT I
160 IF VAL(R$)=0 THEN PRINT@320,"C'EST FINI...":SOUND140,15:END
165 SOUND240,2
170 GOTO 90
200 IF F=1 THEN O$=O$+A$:T$=O$:PRINT@(63+3*I),"0";
210 IF F=0 THEN X$=X$+A$:T$=X$:PRINT@(63+3*I),"X";
220 R$=LEFT$(R$,I-1)+"0"+RIGHT$(R$,LEN(R$)-I)
230 IF LEN(T$)<3 THEN 300
240 FOR I=1 TO 3
245 IF I=LEN(T$) THEN 300
250 FOR J=I+1 TO LEN(T$)
255 IF J=LEN(T$) THEN 300
260 FOR K=J+1 TO LEN(T$)
270 IF VAL(MID$(T$,I,1))+VAL(MID$(T$,J,1))+VAL(MID$(T$,K,1))=15 THEN 350
275 NEXT K
280 NEXT J
290 NEXT I
300 F=ABS(F-1)
310 RETURN
350 IF F=1 THEN PRINT@352,"BRAVO , JOUEUR 0":SOUND200,9:SOUND220,7:SOUND180,9
360 IF F=0 THEN PRINT@352,"BRAVO , JOUEUR X":SOUND200,9:SOUND220,7:SOUND180,9
```

# Calculatrice

**REFERENCES** : Difficulté de programmation : très difficile.
Catégorie de jeu : simulation d'une calculatrice.
Difficulté du jeu : moyenne.

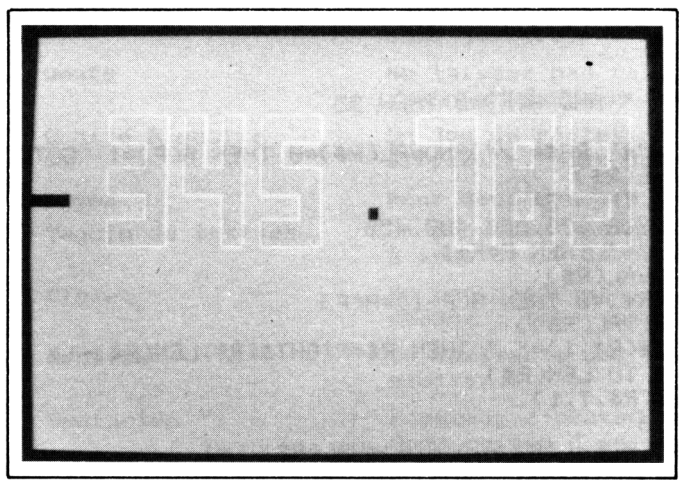
**LE JEU** : Il s'agit de la simulation, en affichage grand format, du fonctionnement d'une petite calculatrice de poche, de type 4 opérations. Les applications pédagogiques de ce programme sont évidentes et son fonctionnement très simple à maîtriser.

**LE PROGRAMME** : Les variables B$ et C$, des lignes 2Ø et 3Ø, contiennent les informations codées qui permettront le dessin des chiffres en grand format.
Les lignes 6Ø à 12Ø s'occupent de la représentation, en grand format, des chiffres sur l'écran , au fur et à mesure des entrées.
Le point décimal et le signe - sont gérés par les lignes 17Ø et 18Ø.
Le nombre total de symboles entrés est contrôlé par la ligne 19Ø.
Le résultat est inscrit grâce à un programme fonctionnant en automatique qui se trouve aux lignes 22Ø à 27Ø (l'automatisme est géré par la ligne 2ØØ, dans le programme principal).
Les sous-programmes placés aux lignes 3ØØ-31Ø, 35Ø-36Ø, 4ØØ-41Ø et 45Ø-46Ø effectuent l'opération indiquée, selon la valeur de Z (ligne 227).
Chaque résultat est mémorisé en A (F-1) (ligne 229).

**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 - 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 BREAK, puis RUN.
Taper BREAK pour sortir du programme.

***EXTENSIONS POSSIBLES*** : On devra, tout d'abord, améliorer l'affichage du signe - (représenté, ici, par un point), puis on pourra améliorer le dessin des chiffres (en utilisant, par exemple, le générateur d'Alice); on peut aussi envisager une extension à 8 chiffres autoriser des opérations plus complexes (COS, SIN...), ajouter des mémoires, etc.).



```
10 REM CALCULATRICE
15 CLS 6
20 C$="444466664464":REM COULEURS 4 ET 6 ; 12 FOIS SHIFT Q
                                    DANS LA COULEUR INDIQUEE
30 B$="144413333313121131314413312131121411333314141141 31"
40 F=1:L=0
50 N=0:T=0:K=0:R$=""
60 A$=INKEY$:IF A$="" THEN 60
65 IF K=1 THEN N=0:R$="":K=0
70 IF A$="=" THEN 220
75 X=(1 AND A$="+")+(2 AND A$="-")+(3 AND A$="*")+(4 AND A$="/")
80 R$=R$+A$
90 IF X<>0 THEN 220
95 IF R$="-" THEN A$=R$: GOTO 170
100 IF A$<"." OR A$>"9" OR A$="/" THEN 60
110 IF N=0 THEN CLS 6
120 IF A$="." OR A$="-" THEN 170
130 FOR I=1 TO 5
140 S=3*VAL(MID$(B$,I+5*VAL(A$),1))-2
150 PRINT@(32*I+N+64),MID$(C$,S,3);
160 NEXT I
170 IF A$="-" THEN PRINT@192,"66";:REM COULEUR 6 ; 2 FOIS SHIFT X
180 IF A$="." THEN PRINT@193+N,"6";:REM COULEUR 6 ; SHIFT W
190 N=N-4*(N<28)
200 IF T<>0 THEN NEXT T
205 IF K=1 AND L<>1 THEN 50
207 T=0
210 GOTO 60
220 IF R$="-" AND A(F)=0 THEN 95
222 N=0:K=1
223 IF (A$="*"ORA$="/")ANDVAL(R$)=0 THEN A(F)=1: GOTO 225
224 A(F)=VAL(R$)
225 IF X<>0 THEN Z=X
227 ON Z GOSUB 300,350,400,450
228 IF A$<>"=" THEN F=F+1
229 A(F-1)=VAL(R$)
230 IF VAL(R$)=0 THEN A(F-1)=A(F)
233 R$=STR$(VAL(R$))
235 IF LEFT$(R$,1)=" " THEN R$=RIGHT$(R$,LEN(R$)-1)
250 FOR T=1 TO LEN(R$)
260 A$=MID$(R$,T,1)
270 GOTO 110
300 R$=STR$(A(F-1)+A(F)):A(F)=VAL(R$):L=1
310 RETURN
350 IF F<>1 THEN R$=STR$(A(F-1)-A(F)):A(F)=VAL(R$):L=1
360 RETURN
400 IF F<>1 THEN R$=STR$(A(F-1)*A(F)):A(F)=VAL(R$):L=1
410 RETURN
450 IF F<>1 THEN R$=STR$(A(F-1)/A(F)):A(F)=VAL(R$):L=1
460 RETURN
```

# NIVEAU 5

| N° | NOM DU PROGRAMME | Thème et particularités |
|---|---|---|
| 1 | **Smokey** | Fumez sans risque, avec ce programme original pour un joueur |
| 2 | **Course de voiture** | Un jeu de réflexe pas trop difficile, pour un joueur. |
| 3 | **Echecs** | Un programme assez simplifié, pour deux joueurs. |
| 4 | **Danger spatial** | Serez-vous assez adroit pour éviter toutes les météorites. |
| 5 | **Tic Tac Toe** | Un classique, mais pas sans intérêt (deux joueurs). |
| 6 | **Anagrammes** | Un bon jeu de réflexes, pour deux joueurs. |
| 7 | **Oeufs** | Ne laissez pas faire la pondeuse diabolique (un joueur). |
| 8 | **Quatre à quatre** | Un jeu de réflexion un peu difficile, pour deux joueurs. |
| 9 | **Dames** | Pour deux joueurs : classique. |
| 10 | **Taquin de lettres** | Un proverbe est caché dedans : à vous de le découvrir. |
| 11 | **Cibles** | Un peu d'adresse, beaucoup de chance : très visuel. |
| 12 | **Gouttes** | Un véritable gag visuel. A exploiter. |
| 13 | **Obstacles** | Beaucoup d'obstacles à éviter dans ce jeu d'adresse pour un joueur. |
| 14 | **Captures dans l'espace** | Votre mission serait facile, sans ces damnés météorites. |
| 15 | **Anomalie** | Exercer votre perspicacité dans la reconnaissance des symboles. |
| 16 | **Plusieurs en un** | Un terrain de jeu et les pions, pour deux joueurs. |
| 17 | **Parcours du cavalier** | Très spectaculaire, et instructif à plus d'un titre. |
| 18 | **Mur de briques** | Un grand classique dans une version très complète. |
| 19 | **Othello** | Pour deux joueurs ; il retourne les pions à votre place. |

# LES INSTRUCTIONS D'EXPLORATION DES CASES MEMOIRES : PEEK ET POKE

Les instructions d'exploration des cases mémoires : **PEEK** et **POKE.**

**La fonction PEEK** permet la lecture directe du contenu de l'une des cases mémoires adressable par le microprocesseur.

**La fonction POKE** permet au contraire l'inscription directe d'une quantité (de 0 à 255, en décima) dans l'une de ces cases mémoires.

Ces deux fonctions permettront, dans un grand nombre de cas, de suppléer aux insuffisances du langage Basic, en allant même jusqu'à l'écriture, à partir du Basic, de programmes en langage machine.

Dans cet ouvrage, ces deux fonctions ont été essentiellement utilisées pour lire l'écran et modifier son contenu.

L'instruction **POINT (x, y)** d'AlICE ne permet en effet que l'identification des couleurs présentes sur l'écran, et non des caractères.

Pour savoir quel est le caractère présent sur l'écran en ligne X et en colonne Y, il faudra utiliser l'instruction PEEK (16384 + 32 x X + Y).

La mémoire d'écran débute en effet à l'adresse 16384 et se termine à l'adresse 16895 (soit à 512 positions).

Le code ainsi obtenu permet d'identifier le caractère (se référer au tableau des codes ASCII d'Alice).

De même, un caractère de code Z ($0 \leqslant Z \leqslant 255$) pourra être affiché en position ligne X, colonne Y par :

```
POKE 16384 + 32 x X + Y, Z
```

# Othello

**REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexion (deux joueurs).
Difficulté du jeu : difficile.

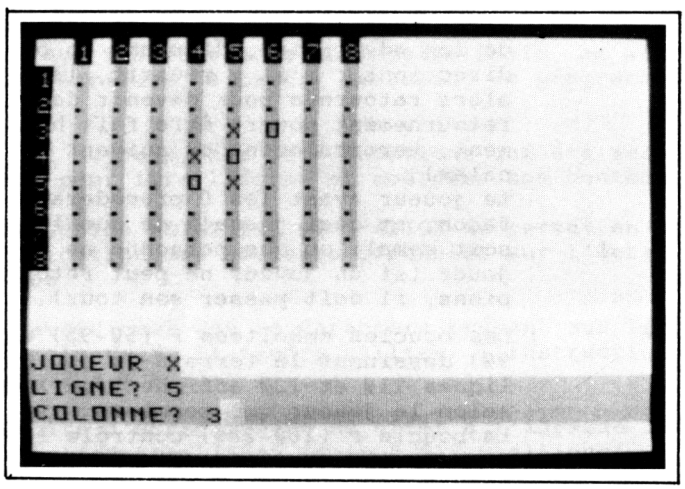
**LE JEU** : Devenu un grand classique des jeux sur micro-ordinateur, le jeu d'OTHELLO ou REVERSI gagne même si comme ici il est pratiqué à deux joueurs, à être géré sur ordinateur, avec visualisation ; en effet, le jeu consistant essentiellement en des retournements des pions de l'adversaire, ceci peut se révéler, à la longue, très fastidieux, surtout si vous êtes un joueur assidu. Le principe est que le joueur, possédant les X, doit pouvoir jouer, réussir à encadrer une ou plusieurs des pièces de son adversaire, dans une ou plusieurs directions ; s'il y réussit, les 0 seront alors retournés pour devenir des X (le retournement pourra être fait horizontalement, verticalement ou suivant les diagonales).
Le joueur ayant les 0 procèdera de la même façon, et ceci jusqu'à ce que le damier soit rempli ou que personne ne puisse plus jouer (si un joueur ne peut retourner des pions, il doit passer son tour).

**LE PROGRAMME** : Les boucles emboîtées P (5Ø-95) et Q (6Ø-9Ø) dessinent le terrain de jeu et les lignes 11Ø et 12Ø affichent un X ou un O selon le joueur qui commence.
La boucle P (16Ø-28Ø) contrôle la validité du coup (la case est-elle libre , : lignes 26Ø et 27Ø), en utilisant une partie de programme qui se trouve aux lignes 33Ø à 39Ø (a-t-on pris un ou plusieurs ennemis en ligne 33Ø et retournement éventuel, autorisé par la ligne 35Ø : on refait un balayage dans toutes les directions , si l'encadrement s'est révélé possible).
La ligne 16Ø définit les 8 directions d'exploration.
Les lignes 2ØØ et 21Ø calculent les déplacements à effectuer pour explorer ces 8 directions.
La ligne 19Ø réalise les déplacements.
La ligne 3ØØ assure le changement de joueur s'il y a eu retournement (contrôlé en ligne

***POUR JOUER*** : Après RUN, chaque joueur entrera à son tour les coordonnées (ligne, ENTER puis colonne ENTER) 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 Ø 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 Alice.

**Nota :** Après avoir lancé le programme par RUN, il convient d'entrer la dimension de la grille (4, ENTER ou 6, 8...).



```
10 REM OTHELLO
20 CLS 8
30 INPUT"DIMENSION DE LA GRILLE ";A
40 IF A/2<>INT(A/2) THEN RUN
45 CLS 8
50 FOR P=1 TO A
55 POKE16384+2*P,48+P
60 FOR Q=1 TO A
70 POKE16384+32*P,48+P:POKE16384+2*Q+32*P,110
80 IF Q=A/2+1 AND P=A/2 THEN POKE16382+2*Q+32*P,88:POKE16384
                                                    +2*Q+32*P,79
85 IF Q=A/2+1 AND P=A/2+1 THEN POKE16382+2*Q+32*P,79:POKE16384
                                                    +2*Q+32*P,88
90 NEXT Q
95 NEXT P
100 K=1
110 IF K=1 THEN PRINT@384,"JOUEUR X";
120 IF K=0 THEN PRINT@384,"JOUEUR O";
125 SOUND180,1
130 PRINT@416,"LIGNE";:INPUTU
135 IF U=0 THEN 300
140 PRINT@448,"COLONNE";:INPUTV
150 H=1
160 FOR P=1 TO 8
170 S=0
180 W=0
190 T=2*U:R=2*V
200 F=(P>3 AND P<7) - (P<3 OR P>7)
210 G=(P>1 AND P<5) - (P>=6)
220 Q=110+22*(S=1)+31*(S=2)
230 IF S=0 THEN 250
240 POKE16384+R+16*T,Q
250 T=T+2*F:R=R+2*G
260 C=PEEK(16384+R+16*T)
270 IF C=88 OR C=79 THEN 330
280 NEXT P
290 IF H<>0 THEN 110
300 K=ABS(K-1)
310 GOTO 110
330 IF (C=79 AND K=1) OR (C=88 AND K=0) THEN W=1
340 IF S<>0 THEN POKE16384+R+16*T,Q
350 IF W=1 AND((C=79 AND K=1) OR (C=88 AND K=0)) THEN 250
360 IF W=0 OR S<> Q0 THEN 280
370 S=1-(K=0)
380 H=-H*(S=0)
390 GOTO 180
```

# Smokey

***REFERENCES*** : Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

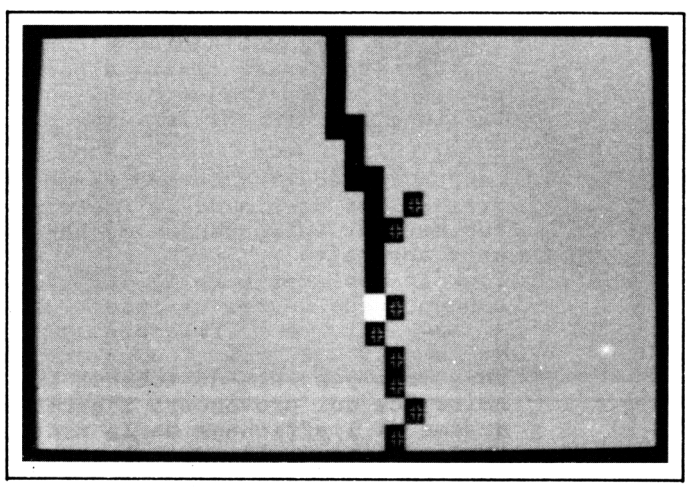
***LE JEU*** : Une colonne de fumée grise apparaît sur l'écran. Le joueur représenté par un point noir devient visible en haut de l'écran. Il doit dans un premier temps se positionner sur la colonne de fumée, puis à partir de là, il va automatiquement avaler la fumée, à condition qu'il réussisse à se maintenir sur sa trace.

***LE PROGRAMME*** : La ligne 5Ø fait apparaître la fumée au bas de l'écran ; la ligne 6Ø permet de changer (vers la droite ou vers la gauche) la direction de la colonne, et la ligne 8Ø fait monter la colonne.
Le joueur apparaît grâce à la ligne 1ØØ et son déplacement est effectué par les lignes 14Ø et 16Ø.
Les lignes 21Ø et 22Ø 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 pavé 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'avaler, 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 CLS 3
30 S=0:X=16:Y=RND(10)+10
40 U=0:V=15
50 POKE 16384+32*X+Y,35
60 Y=Y-(RND(0)<0.5AND Y<30)+(RND(0)<0.5 AND Y>0)
70 T=1
80 X=X-T
90 IF X=0 THEN 200
100 POKE16384+V+32*U,239
110 IF T<>0 THEN FOR I=1 TO 600:A$=INKEY$:IF A$<>"" THEN 130
120 IF T<>0 THEN NEXT I
130 POKE 16384+32*U+V,32
140 U=U-(A$="W"AND(T<>0)ANDX<16)+(A$="Z"AND(T<>0)ANDX>0)
150 U=U-(T=0)
160 V=V-(A$="S"AND V<30)+(A$="Q" AND V>0)
180 C=PEEK(16384+32*U+V)
185 IF C<>35 THEN 50
190 T=0:S=S+1
195 IF U<>16 THEN 100
200 CLS 8
210 IF T=0 THEN PRINT"BRAVO":SOUND220,9:SOUND240,7:SOUND200,9
                                                    :GOTO 230
220 PRINT"PERDU":SOUND150,20
230 PRINT"SCORE"S
```



# Course de voitures

***REFERENCES*** : Difficulté de programmation : difficile
Catégorie de jeu : jeu de réflexe : 1 joueur.
Difficulté du jeu : moyenne.

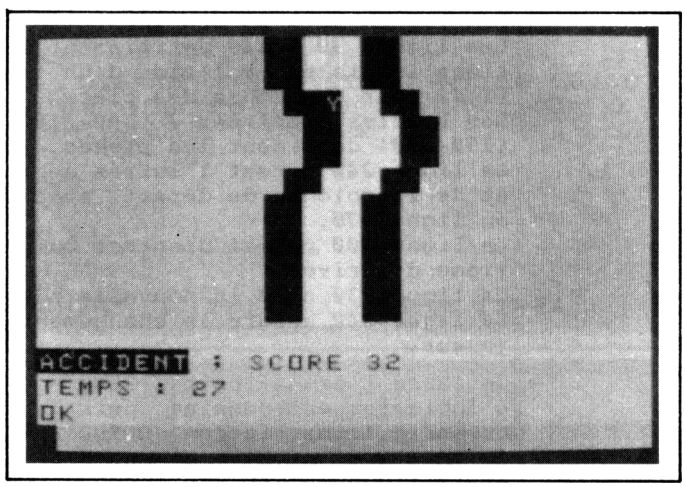
***LE JEU*** : Vous circulez sur une route très sinueuse, ne comportant aucun obstacle, et vous devez essayer de la suivre le plus longtemps possible. La trace de votre parcours reste affiché en permanence, et votre score est affiché dès que vous avez un accident.

***LE PROGRAMME*** : L'impression de la route est assuré par la boucle I (60-80) et les lignes 90 à 110 impriment la couleur.
La ligne 120 fait avancer la route, et la ligne 100 imprime la voiture.
La ligne 130 contrôle son mouvement, et la ligne 140 détermine si la route fait ou non des virages.
La ligne 150 regarde si l'on a touché le bord de la route.
La ligne 165 incrémente le score, et la ligne 160 efface l'ancienne position de la voiture (une trace claire apparaît à l'emplacement de sa trajectoire).
La ligne 210 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 un Y 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 du score réalisé.
On pourra arrêter le déroulement de la partie à tout moment en pressant la touche BREAK.

***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, de faire courir plusieurs voitures. On pourra s'inspirer des jeux d'arcades très nombreux sur ce thème.

```
10 REM COURSE DE VOITURES
15 T=0
20 CLS
30 X=14:Y=11
40 A=5:B=10
50 S=0
60 FOR I=0 TO 6
65 POKE 16384+32*X+Y+I,191
70 IF I<2 OR I>4 THEN POKE 16384+32*X+Y+I,32
80 NEXT I
90 PRINT@480,""
100 POKE16384+32*A+B,25
110 FOR I=1 TO 5:T=T+0.1:A$=INKEY$:IF A$="" THEN NEXT I
120 C=B
130 B=B-(A$="S")+(A$="Q")
140 Y=Y-(2*RND(0) AND Y<25)+(2*RND(0)AND Y>3)
150 IF PEEK(16384+B+32*(A+1))=32 THEN 200
160 POKE16384+C+32*A,223
165 S=S+1
170 GOTO 60
200 PRINT"accident ; SCORE"S:SOUND150,20
210 PRINT"TEMPS :"INT(T)
```



# Echecs

***REFERENCES*** : Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexion pour 2 joueurs.
Difficulté du jeu : difficile.

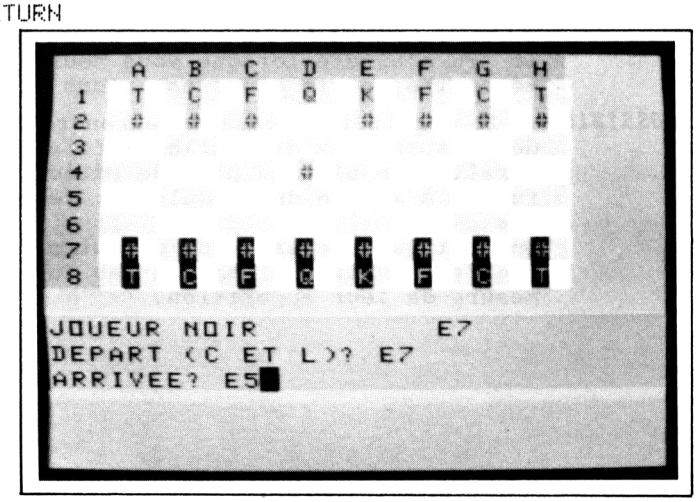
***LE JEU*** : Le programme propose seulement à deux joueurs de disputer une partie en se servan de l'écran du téléviseur comme plateau de jeu. Il n'y a pas de contrôle de la validité des coups, mais seulement gestion de la position des pièces sur l'échiquier. Les pions sont représentés par des P et les autres pièces par leurs initiales (T pour tour, F pour fou, C pour cavalier, Q pour reine, et K pour roi). La couleur (blanc ou noir) permet de distinguer chaque pièce.

***LE PROGRAMME*** : Les lignes 3Ø à 8Ø dessinent l'échiquier.
Les lignes 9Ø à 149 définissent les conditions initiales, à l'aide d'un tableau R (I,J) pour les codes des pièces.
Les boucles emboîtées J (16Ø-21Ø) et I (17Ø-2ØØ) dessinent les pièces.
La ligne 24Ø permet l'entrée de la ligne et de la colonne de départ, avec contrôle en ligne 27Ø.
La ligne 3ØØ permet d'entrer colonne et ligne d'arrivée.
La ligne 33Ø note la nouvelle position.
La ligne 34Ø 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 BLANC, suivie de DEPART (C et L) ; on entre la colonne et la ligne de départ, sous la forme E7 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 BREAK.

***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 Alice de se comporter en partenaire honorable.
Bon courage !

```
10 REM ECHECS
15 DIM R(8,8)
20 CLS 3
30 PRINT"    A  B  C  D  E  F  G  H ";
40 FOR I=1 TO 8
50 PRINT@32*I,I;
60 IF I/2=INT(I/2) THEN PRINT@32*I+3,"888555888555888555888555";
   :REM COULEURS 8 ET 5 ; 24 FOIS SHIFT Q DANS LA COULEUR INDIQUEE
70 IF I/2<>INT(I/2) THEN PRINT@32*I+3,"555888555888555888555888";
   :REM COULEURS 5 ET 8; 24 FOIS SHIFT Q DANS LA COULEUR INDIQUEE
80 NEXT I
90 FOR I=1 TO 8
100 R(2,I)=99:R(7,I)=35
110 NEXT I
120 R(1,1)=84:R(1,2)=67:R(1,3)=70:R(1,4)=81
130 R(1,5)=75:R(1,6)=70:R(1,7)=67:R(1,8)=84
140 R(8,1)=20:R(8,2)=3:R(8,3)=6:R(8,4)=17:R(8,5)=11:R(8,6)=6
145 FOR I=1 TO 8                              :R(8,7)=3:R(8,8)=20
147 FOR J=1 TO 8
148 IF R(I,J)=0 THEN R(I,J)=255+48*((I+J)/2=INT((I+J)/2))
149 NEXT J : NEXT I
150 H=1
160 FOR J=1 TO 8
170 FOR I=1 TO 8
180 X=1+3*J:Y=I
190 POKE 16384+32*Y+X,R(I,J)
200 NEXT I
210 NEXT J
220 PRINT@320,"JOUEUR NOIR"
230 IF H=1 THEN PRINT@320,"JOUEUR BLANC"
240 PRINT@352,"DEPART (C ET L)";:INPUTD$
250 PRINT @340,D$;
260 GOSUB 400
270 IF R(L,C)>200 OR (R(L,C)<65 AND H=1) OR (R(L,C)>64 AND H=0)
280 T=R(L,C):R(L,C)=255+48*((L+C)/2=INT((L+C)/2))      THEN 220
300 PRINT@384,"ARRIVEE";:INPUTD$
310 PRINT@342,"~ ";D$;
320 GOSUB 400
330 R(L,C)=T
340 H=ABS(H-1)
345 PRINT@352,"":PRINT@384,""
350 GOTO 160
400 L=VAL(RIGHT$(D$,1)):C=ASC(LEFT$(D$,1))-64
410 RETURN
```



# Danger spatial

**REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe, pour un joueur.
Difficulté du jeu : moyenne.

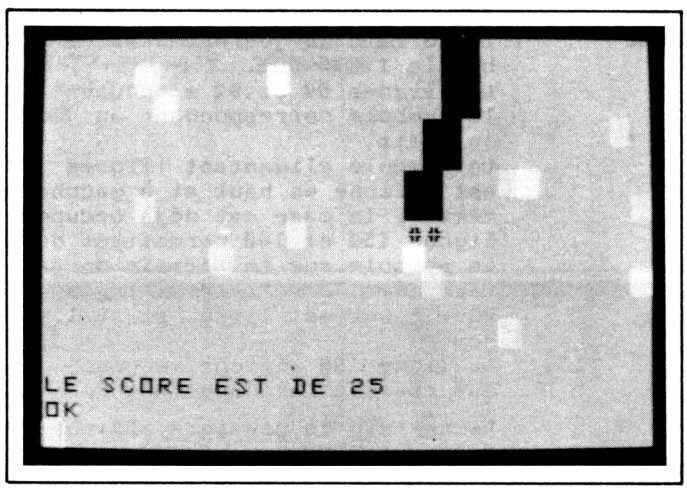
**LE JEU** : Vous êtes aux commandes d'un engin spatial, qui vient de pénétrer dans un nuage de météorites. Vous devez essayer de franchir le nuage sans heurter une seule météorite. Votre score sera affiché en fin de partie, lorsque vous entrez en collision avec une météorite.

**LE PROGRAMME** : La position de chaque météorite est déterminée par les lignes 3Ø et 45 ; elles sont imprimées par les lignes 4Ø et 5Ø.
L'avance des météorites est réalisé par la ligne 6Ø.
Les lignes 7Ø et 8Ø vérifient si vous avez ou non heurté une météorite.
Les lignes 9Ø à 1ØØ dessinent votre vaisseau spatial, et la ligne 13Ø permet de le déplacer.
Le compteur se trouve en ligne 14Ø.

**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 BREAK. 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
15 CLS0
20 X=15:Z=0
30 H=RND(30):F=16*RND(8)-16
40 IF RND(0)<0.7 THEN POKE16836+H,143+F
45 K=RND(30):G=16*RND(7)
50 IF RND(0)<0.6 THEN POKE16836+K,143+G
60 PRINT@480,""
70 B=PEEK(16710+X):C=PEEK(16710+X+1)
80 IF B>144 OR C>144 THEN 200
90 POKE16678+Y,144:POKE16678+Y+1,144
100 POKE16710+X,99:POKE16711+X,99
110 A$=INKEY$
120 Y=X
130 X=X-(A$="S" AND X<30)+(A$="Q"AND X>0)
140 Z=Z+1
145 SOUND220,1
150 GOTO 30
200 PRINT"LE SCORE EST DE"Z
210 SOUND220,9:SOUND240,7:SOUND200,9
```



# Tic tac toe

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

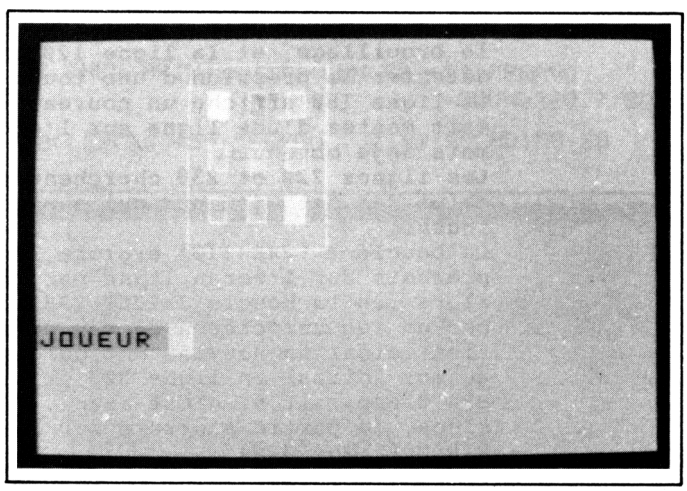
**LE JEU** : Un grand classique des jeux d'écoliers, proposé ici dans une version ne comportant pas le jeu machine ; chaque joueur devra placer à son tour un 0 ou un X dans une case libre de la grille de jeu, et le premier qui réussit un alignement de trois symboles identiques a gagné (l'alignement peut être horizontal, vertical, ou suivant l'une des deux diagonales).

**LE PROGRAMME** : Le terrain de jeu est dessiné à l'aide d'une boucle I (3Ø-6Ø).
Les lignes 8Ø et 9Ø s'occupent d'afficher le symbole correspondant au joueur qui a le trait.
Un symbole clignotant (lignes 11Ø à 13Ø) est affiché en haut et à gauche du terrain, même si la case est déjà occupée ; les lignes 15Ø et 16Ø permettent de déplacer le symbole sur le terrain de jeu.
La ligne 17Ø vérifie si la case où l'on veut jouer est libre, et si l'on désire jouer.
La ligne 18Ø affiche le coup, et la ligne 19Ø 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 était libre, le symbole qui correspond à ce joueur sera inscrit, et on change de joueur.
Lorsque la partie est finie, il convient de taper BREAK 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 CLS 6
30 FOR I=1 TO 7
40 IF I/2<>INT(I/2) THEN PRINT@(32*I+8),"4444444";:REM COULEUR 4
                                                    ; 7 FOIS SHIFT Q
50 IF I/2=INT(I/2) THEN PRINT@(32*I+8),"4646464";:REM COULEURS 4
   ET 6 ; FAIRE SHIFT Q DANS LA COULEUR INDIQUEE
60 NEXT I
70 K=0
80 PRINT@352,"JOUEUR ";:PRINT"8";:REM COULEUR 8 ; SHIFT Q
90 IF K=0 THEN PRINT@352,"JOUEUR ";:PRINT"5";:REM COULEUR 5 ;
                                                    SHIFT Q
100 I=0:J=0
110 C=PEEK(16457+64*J+2*I)
120 POKE16457+2*I+64*J,35
130 POKE16457+2*I+64*J,C
140 A$=INKEY$
150 J=J-(A$="W"AND J<2)+(A$="Z"AND J>0)
160 I=I-(A$="S"AND I<2)+(A$="Q"AND I>0)
170 IF C<>223 OR A$<>"0" THEN 110
180 POKE16457+2*I+64*J,207+K*48
190 K=ABS(K-1)
200 GOTO 80
```



# Anagrammes

**REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexes (2 joueurs).
Difficulté du jeu : selon le niveau.

**LE JEU** : Des lettres sont choisies au hasard par Alice, 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 Ø) et doivent l'appuyer 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.

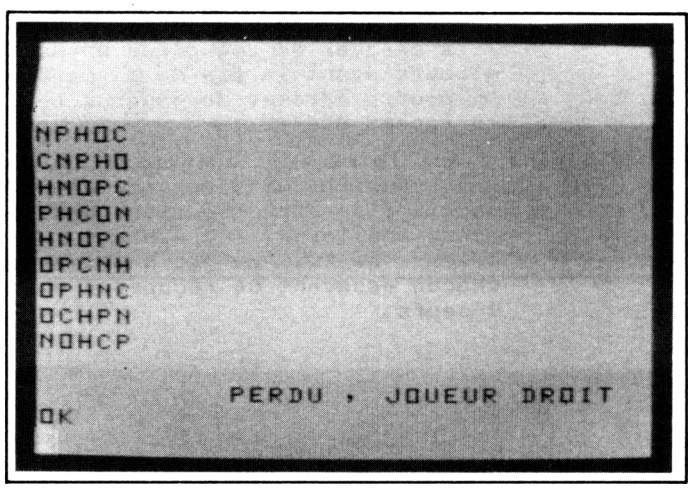
**LE PROGRAMME** : Le niveau du jeu est entré à la ligne 3Ø.
La boucle I (6Ø-8Ø) réalise le choix des lettres.
Une deuxième boucle I (1ØØ-17Ø) effectue le brouillage, et la ligne 12Ø permet de détecter la pression d'une touche.
La ligne 18Ø affiche un nouveau mot, et fait monter d'une ligne sur l'écran les mots déjà obtenus.
Les lignes 22Ø et 23Ø cherchent quel est celui des deux joueurs qui a pressé une touche.
La boucle I (24Ø-31Ø) explore les mots présents sur l'écran ligne par ligne, alors que la boucle J (26Ø-29Ø) saisit un par un les caractères de ces mots. On obtient ainsi un nouveau mot qui est comparé au mot initial en ligne 3ØØ et le résultat BRAVO apparaît si c'est exact.
Sinon, la partie s'arrête sur l'affichage PERDU (ligne 32Ø).

**POUR JOUER** : On lance le programme par RUN.
On devra ensuite entrer le niveau de difficulté (en principe de 1 à 3) et faire ENTER. 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 Ø), 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, etc.

```
10 REM ANAGRAMMES
20 CLS 8
30 INPUT"NIVEAU (DE 1 A 3)";N
40 IF N<1 OR N>3 THEN RUN
50 S$=""
60 FOR I=1 TO N+3
70 S$=S$+CHR$(RND(26)+64)
80 NEXT I
90 A$="":B$=S$
100 FOR I=1 TO LEN(S$)
110 N=RND(LEN(S$))
120 D$=INKEY$
130 IF D$="1" OR D$="0" THEN 220
140 IF MID$(B$,N,1)="-" THEN 110
150 A$=A$+MID$(B$,N,1)
160 B$=LEFT$(B$,N-1)+"-"+RIGHT$(B$,LEN(B$)-N)
170 NEXT I
180 PRINT@448,A$
185 PRINT
190 FOR I=1 TO 30*LEN(B$):NEXT I
200 S$=A$
210 GOTO 90
220 H$="JOUEUR DROIT"
230 IF D$="1" THEN H$="JOUEUR GAUCHE"
240 FOR I=0 TO 12
250 B$=""
260 FOR J=0 TO LEN(S$)-1
265 IF PEEK(16384+32*I+J)>91 THEN 290
270 R$=CHR$(PEEK(16384+32*I+J))
280 B$=B$+R$
290 NEXT J
300 IF B$=S$ THEN PRINT@(32*I+LEN(S$)+5),"BRAVO , ";H$:SOUND200,9
                                                    :SOUND220,7:SOUND180,9:END
310 NEXT I
320 PRINTTAB(LEN(S$)+5);"PERDU , ";H$:SOUND150,20
```



# Œufs

***REFERENCES*** : Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

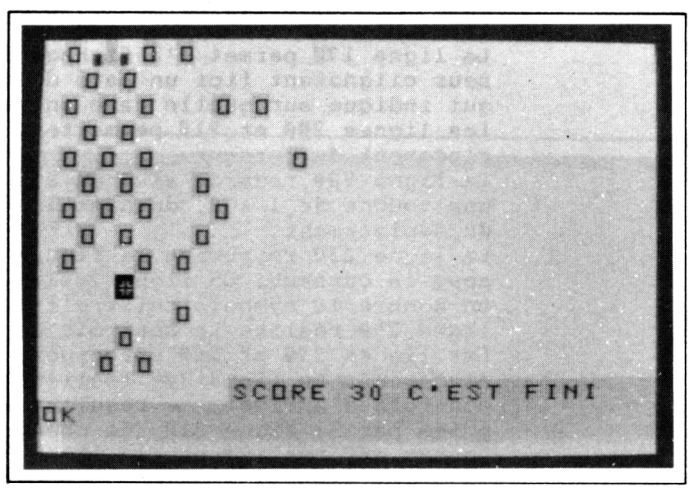
***LE JEU*** : Un oiseau mystérieux se déplace en battant des ailes au hasard sur l'écran. Il pond des oeufs (représentés par des O) que vous ( un " # " ) devez essayer de récupérer. Votre score (nombre d'oeufs récupérés) sera affiché en permanence, et la partie se déroule en un laps de temps fini.

***LE PROGRAMME*** : L'oiseau est dessiné en deux parties (pour le mouvement) aux lignes 60 et 100 ; son déplacement aléatoire est géré par le sous-programme 300-390 (en particulier les lignes 350 à 380).
La ligne 120 permet de reboucler le programme sur la ligne 60.
Le déplacement du joueur est régi par les lignes 160 et 170, et le test permettant de détecter la présence d'un oeuf est effectué en ligne 180.
La ligne 200 permet l'affichage du score.
La fin de jeu est détectée en ligne 320.

***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 oeufs. Lorsque le symbole qui représente le joueur (un " # ") 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'oeufs avant la fin de la partie.
On pourra arrêter de jouer à tout moment en tapant BREAK.

***EXTENSIONS POSSIBLES*** : Sur le même principe, on pourra facilement imaginer d'autres thèmes de jeu. On pourra bien sûr améliorer la couleur, les sons, améliorer le graphisme, et éventuellement, définir un jeu à plusieurs joueurs, chacun essayant de récupérer un maximum d'oeufs.

```
10 REM OEUFS
20 CLS6
30 P=200
40 X=RND(28):Y=RND(14)
50 R=13:S=9:T=0
60 POKE16384+X+32*Y,253:POKE16385+X+32*Y,255:POKE16386+X+32*Y,254
70 FOR I=1 TO 30:A$=INKEY$:IF A$="" THEN NEXT I
80 IF A$<>"" THEN GOSUB 150
90 GOSUB 300
100 POKE16384+X+32*Y,254:POKE16385+X+32*Y,143:POKE16386+X+32*Y,253
110 GOSUB 300
120 GOTO 60
150 POKE16384+R+32*S,223
160 R=R-(A$="S" AND R<30)+(A$="Q" AND R>0)
170 S=S-(A$="W" AND S<16)+(A$="Z" AND S>0)
180 G =PEEK(16384+R+32*S)
190 T=T-(G=79)
200 PRINT@490,"SCORE"T;
210 RETURN
300 A=X:B=Y
310 P=P-1
320 IF P=0 THEN PRINT"C'EST FINI":SOUND200,9:SOUND220,7
                                              :SOUND180,9:END
330 POKE16384+R+32*S,35
340 POKE16384+A+32*B,223:POKE16385+A+32*B,79:POKE16386+A+32*B,223
350 X=ABS(X+INT(1+2*(RND(0)<0.5)))
360 Y=ABS(Y+INT(1+2*(RND(0)<0.5)))
370 IF X>29 THEN X=1
380 IF Y>15 THEN Y=1
390 RETURN
```



# 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 0 ; si l'on joue en solitaire, cet échange devra être réalisé avec le moins de coups possibles. 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.

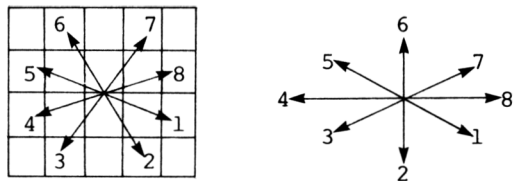
**LE PROGRAMME** : La variable alphanumérique A$, en ligne 40, contient la table des coups possibles (déplacement du cavalier).
Les boucles emboîtées I (50-110) et J (60-100) dessinent le terrain de jeu, et les positions de départ.
Les lignes 130 et 140 permettent d'afficher O ou X, selon le joueur qui a le trait.
La ligne 170 permet l'affichage du curseur clignotant (ici un pavé de couleur) qui indique sur quelle case on se trouve.
Les lignes 200 et 210 permettent le déplacement du curseur.
La ligne 220 regarde si l'on a appuyé sur une touche de 1 à 8, donnant la direction du déplacement.
La ligne 230 réimprime la figure qui était sous le curseur. La ligne 260 transforme en nombre le symbole chiffre frappé. La ligne 250 réalise le contrôle de départ.
Les lignes 270 et 280 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 (ligne 330). Le sous-programme 350-360 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 manoeuvrant le curseur clignotant grâce aux touches :

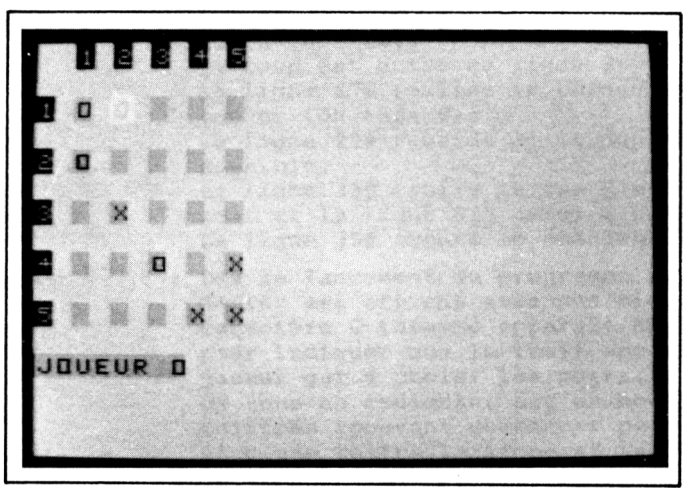
↑ ↓ → ←

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 jusqu'à l'amener sur l'un de ses 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 BREAK.

***EXTENSIONS POSSIBLES*** : On peut utiliser la couleur, 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 Alice de participer au jeu avec de bonnes chances de l'emporter.



```
10 REM QUATRE A QUATRE
20 CLS 3
30 H=1
40 A$="5665322356"
50 FOR I=1 TO 5
60 FOR J=1 TO 5
70 POKE16384+64*I,48+I:POKE16384+2*I,48+I:POKE16384+2*I+64*J,143
80 IF I<3 AND J<3 THEN POKE16384+2*I+64*J,79
90 IF I>3 AND J>3 THEN POKE16384+2*I+64*J,88
100 NEXT J
110 NEXT I
120 X=1:Y=1
130 IF H=1 THEN PRINT@384,"JOUEUR O";
140 IF H=0 THEN PRINT@384,"JOUEUR X";
150 GOSUB 350
160 I=J
170 POKE16384+2*X+64*Y,255
180 A=X:B=Y
190 R$=INKEY$
200 Y=Y-(R$="W"AND Y<5)+(R$="Z" AND Y>1)
210 X=X-(R$="S" AND X<5)+(R$="Q" AND X>1)
220 IF R$<"9" AND R$>"0" THEN 250
230 POKE16384+2*A+64*B,I
240 GOTO 130
250 IF (I=143) OR (I=79 AND H=0) OR (I=88 AND H=1) THEN 230
260 J=VAL(R$)
270 Y=Y+VAL(MID$(A$,J,1))-4
280 X=X+VAL(MID$(A$,J+2,1))-4
290 GOSUB 350
300 IF J<>143 THEN 230
310 POKE16384+2*X+64*Y,79-9*(H=0)
320 H=ABS(H-1)
325 SOUND220,1
330 I=143
340 GOTO 230
350 J=PEEK(16384+2*X+64*Y)
360 RETURN
```

# Dames

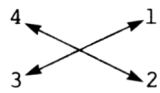
**REFERENCES** : Difficulté de programmation : très difficile.
Catégorie du jeu : deux joueurs.
Difficulté du jeu : difficile.

**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 de pièces sont entrés en une seule manoeuvre (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 Ø pour pouvoir rejouer s'il y avait eu saut ou si le coup n'était pas possible.

**LE PROGRAMME** : La boucle I (4Ø-8Ø) dessine le damier.
Les boucles I (9Ø-15Ø) et J (1ØØ-14Ø) définissent les positions initiales des pions noirs et des pions blancs. Les positions sont dans une table R (X, Y).
L'impression des pions proprement dite est effectuée par les boucles emboîtées J (17Ø-23Ø) et I (18Ø-22Ø).
Les lignes 24Ø et 25Ø indiquent le joueur qui a le trait.
Le coup est entré en ligne 26Ø.
La ligne 27Ø réalise le changement de joueur (on tape Ø).
La ligne 29Ø regarde si le déplacement est possible.
La ligne 32Ø traite le cas d'un déplacement et la ligne 325 celui d'un saut.
La ligne 35Ø assure le changement de joueur.

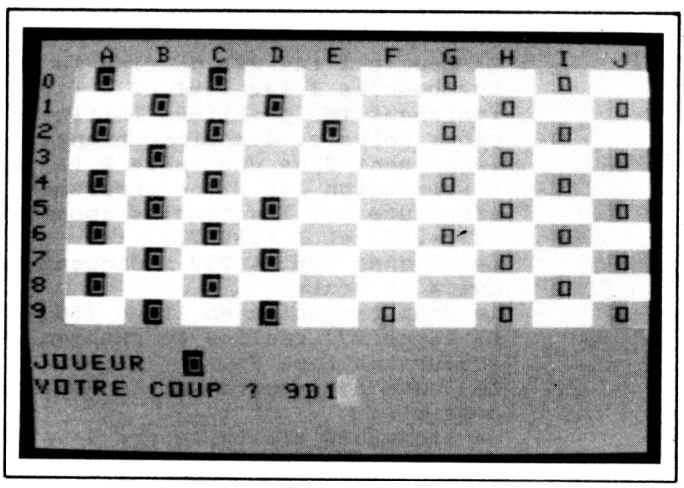
**POUR JOUER** : Dès le lancement du programme par RUN, le damier est affiché avec ses pions, et un caractère O inversé apparaît en-dessous pour indiquer que le trait appartient au joueur qui a choisi les noirs.
On joue en indiquant par un nombre de 3 chiffres (pouvant commencer par un Ø) suivi d'une lettre 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, 9C1 signifie que le O situé en ligne 9, colonne C se déplace dans la direction 1 (ne pas oublier de faire ENTER). De même 0E3 signifiera pièce noire, ligne Ø, colonne E, 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 ENTER). 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 Alice de se mêler avec de bonnes chances de réussite à vos passionnantes parties.



```
10 REM DAMES
15 DIM R(10,10)
20 CLS
30 PRINT"   A  B  C  D  E  F  G  H  I  J "
40 FOR I=0 TO 9
50 PRINT@32*I+32,RIGHT$(STR$(I),1);
60 IF I/2<>INT(I/2) THEN PRINT@(32*I+34),"55544455544455544455544
   44555444";:REM COULEURS 4 ET 5;SHIFT Q DANS LA COULEUR INDIQUEE
70 IF I/2=INT(I/2) THEN PRINT@(32*I+34),"44455544455544455544455
   444555";:REM COULEURS 4 ET 5 ;SHIFT Q DANS LA COULEUR INDIQUEE
80 NEXT I
90 FOR I=0 TO 9
100 FOR J=0 TO 9
110 R(I,J)=207+((I+J)/2=INT((I+J)/2))*16
120 IF J<4 THEN R(I,J)=15
130 IF J>5 THEN R(I,J)=79
140 NEXT J
150 NEXT I
160 H=1
170 FOR J=0 TO 9
180 FOR I=0 TO 9
190 IF (I+J)/2<>INT((I+J)/2) THEN 220
200 X=3*J+3:Y=1+I
210 POKE 16384+X+32*Y,R(I,J)
220 NEXT I
230 NEXT J
240 PRINT@384,"JOUEUR":POKE16776,15
250 IF H=0 THEN PRINT@384,"JOUEUR 0 "
260 INPUT"VOTRE COUP ";D$
270 IF D$="0" THEN 350
280 GOSUB 400
290 IF R(L,C)=191 OR (R(L,C)>64 AND H=1) OR (R(L,C)<63 AND H=0)
                                                    THEN 240
300 T=(1 AND(D=1 OR D=2))-( 1 AND (D=3 OR D=4))
310 S=(1 AND (D=3 OR D=2))-(1AND(D=1 OR D=4))
320 IF R(L+S,C+T)=191 THEN R(L,C)=191:R(L+S,C+T)=79-H*64:GOTO 350
325 IF R(L+S,C+T)=15+H*64 AND R(L+2*S,C+2*T)=191 THEN 340
330 GOTO 240
340 R(L,C)=191:R(L+S,C+T)=191:R(L+2*S,C+2*T)=79-H*64
345 GOTO 170
350 H=ABS(H-1)
355 SOUND220,1
360 GOTO 170
400 L=VAL(LEFT$(D$,1))+1:C=ASC(MID$(D$,2,1))-64:D=VAL(RIGHT$(D$,1))
405 L=L-1:C=C-1
410 RETURN
```

# Taquin de lettres

**REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté de programmation : difficile.

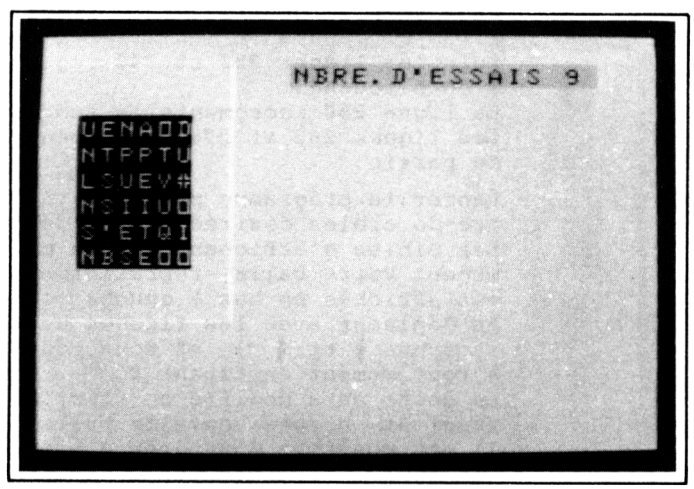
**LE JEU** : Qui d'entre nous n'a pas, un jour ou l'autre eu dans les mains un de ces petits puzzles où l'on essaie de déplacer d'une case à une autre des petits carrés de plastique colorés, jusqu'à l'obtention d'un résultat bien précis ? En son temps le 14-15 de SAM LOYD, a eu autant de succès que notre 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. Un symbole clignotant 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 symbole (un " # " ) ; 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 symbole.
La ligne 260 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 symbole 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 BREAK.

***EXTENSIONS POSSIBLES*** : Le même principe peut être appliqué à toute sorte de taquin (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 CLS3
25 CLEAR 200
30 FOR I=1 TO 10
40 PRINT@32*I,"4444444444";:REM COULEUR 4 ; 10 FOIS SHIFT Q
50 NEXT I
60 A$=""
70 R$="ONASOUVENTBESOIND'UNPLUSPETITQUESOI"
80 FOR I=1 TO 35
90 X=RND(35)
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))+64*(MID$(A$,I,1)<>"'")
140 POKE16482+(I-6*INT(I/6))+32*INT(I/6),F
150 NEXT I
160 X=0:Y=0:S=0:C=32
170 PRINT@45,"NBRE.D'ESSAIS"S;
180 POKE16482+Y+32*X,35
190 T$=INKEY$
200 IF T$="" THEN POKE16482+Y+32*X,C:GOTO 180
210 I=X:J=Y
220 X=X-(T$="W" AND X<5)+(T$="Z" AND X>0)
230 Y=Y-(T$="S" AND Y<5)+(T$="Q" AND Y>0)
240 C=PEEK(16482+Y+32*X)
250 POKE16482+J+32*I,C
260 S=S+1
265 SOUND220,2
270 GOTO 170
```



# Cibles

**REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : hasard.
Difficulté du jeu : facile.

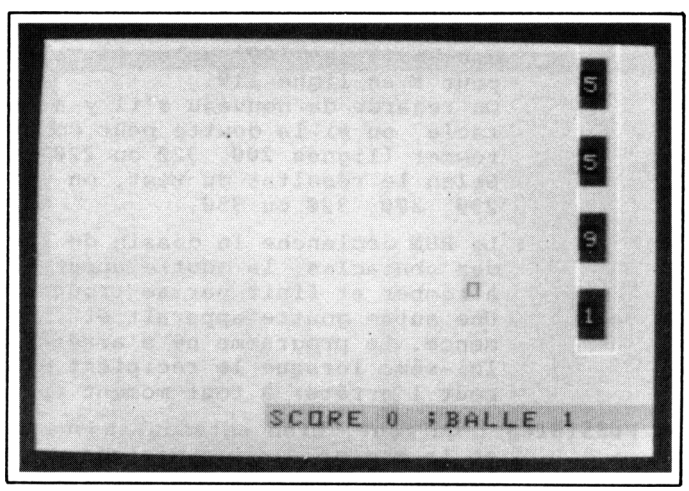
**LE JEU** : Il consiste simplement à tirer sur des cibles, en essayant de marquer le maximum de points ; le nombre de cibles (de 1 à 5) sera choisi en début de partie, et le score réalisé sera affiché en permanence. Si l'on touche le bord d'une cible, la partie s'arrête. On marquera d'autant plus de points que le nombre sur lequel on tombe sera plus grand.

**LE PROGRAMME** : Le nombre de cibles est entré en ligne 3Ø.
La boucle I (65-75) effectue l'impression des cibles.
La ligne 9Ø affichera le score en permanence.
La boule est affichée par la ligne 1ØØ.
La boucle I (12Ø-17Ø) réalise l'impression des chiffres contenus dans les cibles, et permet le déplacement de la boule, (ligne 16Ø).
La ligne 19Ø regarde si l'on a tiré, et la ligne 2ØØ regarde si la boule est encore sur la ligne de départ.
La ligne 21Ø fait progresser la boule d'une case vers la droite, et la ligne 22Ø vérifie si l'on a ou non atteint les cibles ; puis les lignes 23Ø et 24Ø regardent ce que l'on a touché.
La ligne 25Ø incrémente le score.
Les lignes 26Ø et 27Ø s'occupent de la fin de partie.

**POUR JOUER** Lancer le programme par RUN, taper le nombre de cibles désirées suivi de ENTER.
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 BREAK, 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 CLS 6
30 INPUT"NOMBRE DE CIBLES";N
40 P=10:J=28
50 S=0:B=1
60 CLS 6
65 FOR I=0 TO N-1
67 PRINT@96*I+J,"44";:REM COULEUR 4 ; SHIFT R , PUIS SHIFT E
70 PRINT@96*I+J+32,"44";:REM COULEUR 4 ; SHIFT B , PUIS SHIFT G
72 PRINT@96*I+J+64,"44";:REM COULEUR 4 ; SHIFT X , PUIS SHIFT D
75 NEXT I
80 K=0:A=15
90 PRINT@460,"SCORE"S";BALLE"B
100 POKE16384+K+32*A,79
110 D=A
120 FOR I=0 TO N-1
130 U=INT(RND(0)*P)
140 POKE16384+J+96*I+32,U+48
150 A$=INKEY$
160 A=A-(A$="W")+(A$="Z")
170 NEXT I
180 POKE16384+K+32*D,223
190 IF A$="0" THEN B=B+1:GOTO 210
200 IF K=0 THEN 100
210 K=K+1
220 IF K<J THEN 100
230 C=PEEK(16384+K+32*A)
240 IF C<48 OR C>57 THEN PRINT"PERDU":SOUND150,20:END
250 S=S+C-48
260 P=P-1
270 IF P=0 OR B>10 THEN SOUND120,20:END
275 SOUND220,1
280 GOTO 80
```



# Gouttes

**REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

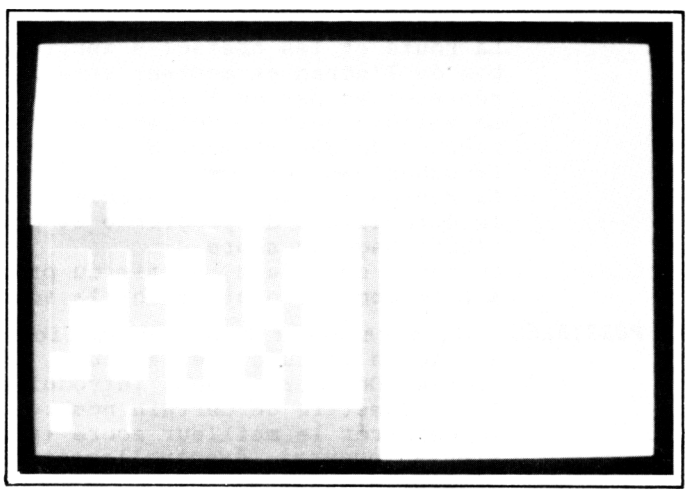
**LE JEU** : C'est un véritable programme de dessin animé que l'on peut considérer, en fait, comme un interlude ; un récipient est dessiné sur l'écran, vu en coupe, et il contient déjà des obstacles. Une goutte d'eau, symbolisée par un pavé bleu apparaît au-dessus du récipient et tombe en remplissant 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 remplissage de l'écran.

**LE PROGRAMME** : Les deux boucles emboîtées I (3Ø-9Ø) et J (4Ø-8Ø) dessinent le récipient et les obstacles.
La ligne 13Ø dessine la goutte.
La ligne 15Ø efface la goutte précédente et la ligne 17Ø envoie dans un sous-programme (4ØØ-42Ø) qui permet de déterminer q'il y a ou non rencontre avec un obstacle.
Si c'est non, on va en 3ØØ et la goutte continue de tomber.
Si c'est oui, la goutte part à droite ou à gauche (ligne 19Ø) selon la valeur définie pour K en ligne 11Ø.
On regarde de nouveau s'il y a ou non obstacle, ou si la goutte peut continuer à tomber (lignes 2ØØ, 32Ø ou 22Ø).
Selon le résultat du test, on utilisera 23Ø, 3ØØ, 32Ø ou 35Ø.

**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 ne s'arrêtera pas de lui-même lorsque le récipient déborde ; on peut l'arrêter à tout moment en tapant BREAK.

**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 CLS 2
30 FOR I=7 TO 15
40 FOR J=1 TO 16
50 POKE16384+32*I,143:POKE16401+32*I,143
55 POKE16864+J,143
60 IF I<>15 AND I<>7 THEN POKE 16384+J+32*I,159
70 IF I<>15 AND RND(0)<0.6 THEN GOSUB 500
80 NEXT J
90 NEXT I
100 Y=RND(7)+5:X=1
110 K=-2*(RND(0)<0.5)-1
120 I=0
130 POKE16384+Y+32*X,175
135 SOUND200,1
140 FOR J=0 TO 150:NEXT J
150 POKE16384+Y+32*X,159
160 U=X+1:V=Y
170 GOSUB 400
180 IF J=0 THEN 300
190 U=X:V=Y+K
200 GOSUB 400
210 IF J=0 THEN 320
220 IF I=0 THEN 350
230 POKE 16384+Y+32*X,175
240 GOTO 100
300 X=X+1
310 GOTO 110
320 Y=Y+K
330 I=1
340 GOTO 130
350 K=-K
360 I=1
370 GOTO 190
400 C=PEEK(16384+V+32*U)
410 J=-(C<>159)
420 RETURN
500 S=INT(RND(0)*13)
510 POKE16385+S+32*I,191
520 RETURN
```



# Obstacles

***REFERENCES*** : Difficulté de programmation : assez difficile.
Catégorie de jeu : jeu de réflexe (pour un joueur).
Difficulté du jeu : moyenne.

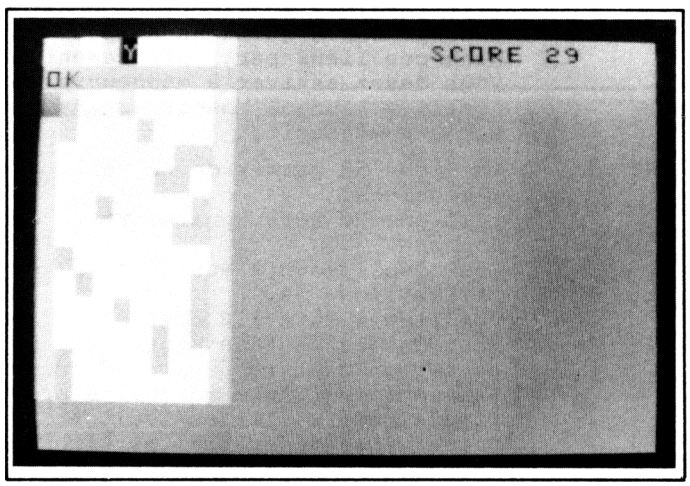
***LE JEU*** : Des obstacles se déplacent sur l'écran pour venir à la rencontre d'une petite voiture symbolisée par un Y inversé, qui dispose de la possibilité de se déplacer vers la gauche ou vers la droite pour les éviter et qui devra tenir le plus longtemps possible. La distance parcourue sera affichée en permanence, et le jeu se termine lorsque la voiture rencontre, soit le bord du terrain, soit un obstacle.

***LE PROGRAMME*** : La ligne 7Ø fait avancer les obstacles vers la voiture du joueur qui est imprimée grâce à la ligne 35.
Les lignes 4Ø, 5Ø et 6Ø fabriquent la route (les lignes 5Ø et 6Ø 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 8Ø affiche le score.
La ligne 1ØØ permettra de déplacer le véhicule.
Les lignes 11Ø et 12Ø permettent de contrôler s'il y a ou non un obstacle et arrêtent le programme s'il y a lieu.
Le score est incrémenté par la ligne 13Ø.

***POUR JOUER*** : On lance le programme en tapant RUN.
La route et les obstacles apparaissent en bas de l'écran et montent vers la voiture représentée par un Y inversé.
La voiture peut se déplacer à l'aide des flèches de déplacement horizontal (touches ← et → ).
La rencontre avec un obstacle interrompt le déroulement du programme et provoque l'affichage du score.
On peut, cependant, sortir du programme à tout moment en utilisant la touche BREAK.

***EXTENSIONS POSSIBLES*** : Il sera assez facile d'améliorer la dessin de la voiture, de la route, des obstacles. On pourra aussi introduire la couleur, admettre un certain nombre d'accidents enregistrer le meilleur score (pour plusieurs joueurs), éventuellement dessiner plusieurs voitures, etc.

```
10 REM OBSTACLES
20 CLS
30 S=0:Y=4
35 PRINT@32+Y,"y";
40 PRINT@448,"8555555558";:REM COULEURS 8 ET 5 ; SHIFT Q DANS
                                                LA COULEUR INDIQUEE
50 PRINT@448+RND(8),"4";:REM COULEUR 4 ; SHIFT Q
60 PRINT@448+RND(8),"4";:REM COULEUR 4 ; SHIFT Q
70 PRINT@480,""
80 PRINT@20,"SCORE"S
90 FOR I=1 TO 10:A$=INKEY$:IF A$="" THEN NEXT I
100 Y=Y-(A$="S" AND Y<9)+(A$="Q" AND Y>0)
110 C=PEEK(16416+Y)
120 IF C=191 OR C=255 THEN END
130 S=S+1
140 SOUND200,1
150 GOTO 35
```



# Captures dans l'espace

**REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (1 seul joueur).
Difficulté du jeu : moyenne.

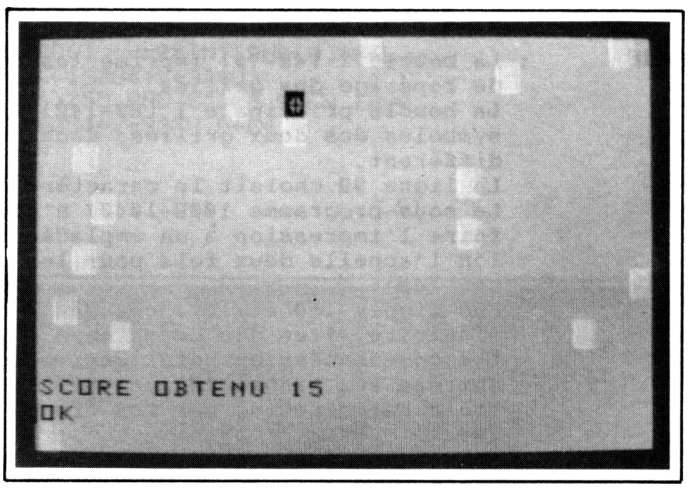
**LE JEU** : Votre vaisseau spatial navigue dans un dangereux champ de météorites et votre mission est de vous emparer de la plus grande quantité possible de minerai d'urane, un métal indispensable que l'on ne peut trouver que dans ces lieux particulièrement dangereux. Vous devez arriver à manoeuvrer pour recueillir l'urane, sans vous faire heurter par une météorite.

**LE PROGRAMME** : La ligne 5Ø permet de faire apparaître les météorites.
La ligne 7Ø fera apparaître le minerai d'urane.
Leur déplacement, sur l'écran, est assuré par la ligne 8Ø.
Les lignes 9Ø à 11Ø 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. (ligne 2ØØ).
Votre vaisseau spatial est imprimé par les lignes 12Ø et 13Ø et son déplacement est contrôlé par la ligne 15Ø.

**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 BREAK.

**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 manoeuvré par une puissance étrangère et qui cherche aussi à s'emparer de l'urane...

```
10 REM CAPTURES DANS L'ESPACE
20 CLS
30 X=7:Z=0
40 S=RND(32)
50 POKE16831+S,191
60 S=RND(32)
70 IF RND(0)<0.5 THEN POKE16831+S,175
80 PRINT@480,""
90 C=PEEK(16544+X)
100 IF C=191 THEN 200
110 IF C=175 THEN Z=Z+1
120 POKE16511+X,143:POKE16512+X,143:POKE16513+X,143
130 POKE16544+X,35
140 FOR I=1 TO 10:A$=INKEY$:IF A$="" THEN NEXT I
150 X=X-(A$="S"AND X<31)+(A$="Q"AND X>0)
160 GOTO 40
200 PRINT"SCORE OBTENU"Z:SOUND220,9:SOUND230,7:SOUND200,9
```



# Anomalie

***REFERENCES*** : Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

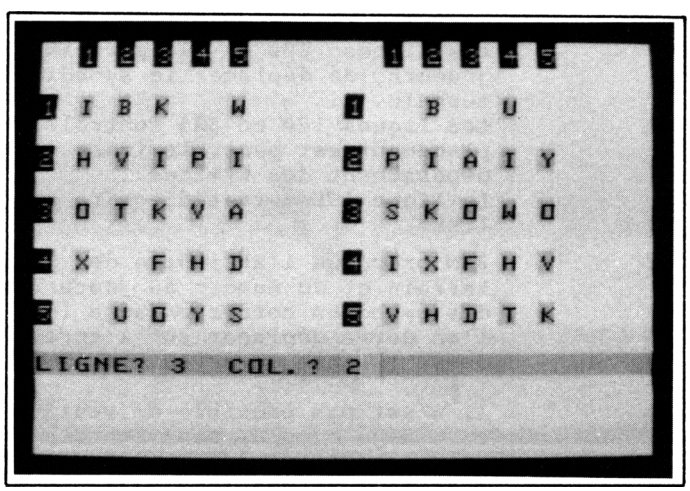
***LE JEU*** : Il repose essentiellement sur une bonne capacité d'observation. Deux grilles de 25 cases chacune sont dessinées sur l'écran. Des symboles identiques sont affichés en des endroits différents des deux grilles, mais l'un des symboles a été modifié sur la grille de droite. Le jeu consiste à trouver lequel, en entrant ses coordonnées dans l'ordre ligne-colonne.

***LE PROGRAMME*** : La boucle I (4Ø-7Ø) imprime les chiffres de repérage des grilles.
La boucle principale I (8Ø-14Ø) imprime les symboles des deux grilles; dans un ordre différent.
La ligne 9Ø choisit le caractère à imprimer.
Le sous-programme 1ØØØ-1Ø4Ø) s'occupe de faire l'impression à un emplacement libre (on l'appelle deux fois pour les deux grilles).
Les lignes 15Ø et 16Ø) changent, de façon aléatoire, l'un des caractères.
Les coordonnées du point essayé sont entrées aux lignes 17Ø et 18Ø, et le contrôle est effectué par les lignes 19Ø et 2ØØ.
Les lignes 21Ø et 22Ø s'occupent de l'affichage du résultat.

***POUR JOUER*** : Après avoir lancé le programme par l'instruction RUN, on obtient 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 ENTER, et son numéro de colonne suivi de ENTER.
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 ANOMALIES
20 CLS 8
30 T=5
40 FOR I=1 TO T
50 POKE16384+2*I,48+I:POKE16400+2*I,48+I
60 POKE16384+64*I,48+I:POKE16400+64*I,48+I
70 NEXT I
80 FOR I=1 TO 22
90 D=RND(26)
100 L=1:S=T
110 GOSUB 1000
120 L=9
130 GOSUB 1000
140 NEXT I
150 S=0
160 GOSUB 1000
170 PRINT@384,"LIGNE";:INPUTI
180 PRINT@394,"COL.";:INPUTT
190 U=2*I:V=2*T+16
200 GOSUB 1010
205 C=C-64
210 IF D=C THEN PRINT@416,"bravo":SOUND200,9:SOUND220,7:SOUND180,9
220 IF D<>C THEN PRINT@416,"PERDU":SOUND150,20:PRINT"C'ETAIT LA
                                                    LETTRE ";CHR$(D+64)
230 END
1000 U=2*RND(T):V=2*INT(RND(0)*S+L)
1010 C=PEEK(16384+V+32*U)
1020 IF (C<>255 AND S<>0) THEN 1000
1030 POKE16384+V+32*U,D+64
1040 RETURN
```



# 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 possible 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'utiser pour un jeu de morpions (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.

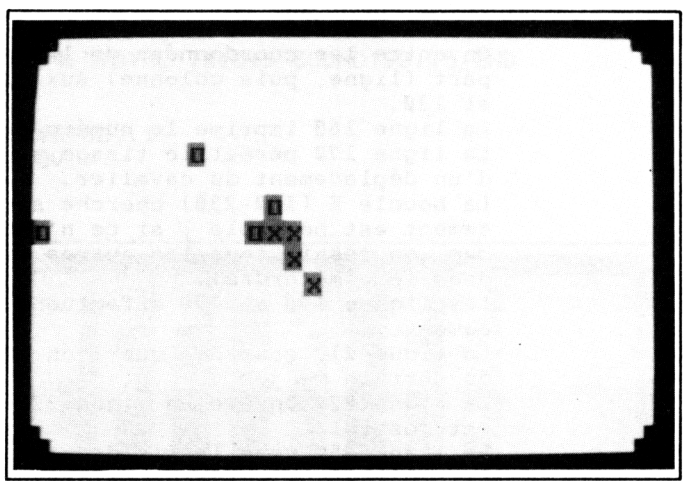
**LE PROGRAMME**

Le terrain de jeu est délimité par 4 symboles mis en place par la ligne 3Ø.
Le changement de joueur est effectué par la ligne 5Ø.
Le joueur qui a le trait voit le symbole qu'il a choisi (O ou X) se placer en limite du cadre (ligne 7Ø).
Les lignes 1ØØ et 11Ø permettent, aux deux joueurs, de déplacer le symbole sur le terrain.
Les lignes 12Ø et 135 contrôlent si le déplacement est possible (case libre et non dépassement des limites).
La ligne 17Ø 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 procèdera de la même manière que son adversaire.
Pour sortir du programme, il suffira de taper BREAK.

***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, etc.

```
10 REM PLUSIEURS EN UN
20 CLS 7
30 POKE16384,225:POKE16415,226:POKE16864,228:POKE16895,232
40 J=0
50 J=ABS(J-1)
60 X=7:Y=0
70 POKE16384+Y+32*X,79-9*(J=1)
80 A$=INKEY$:IF A$="" THEN 80
90 A=X:B=Y
100 X=X-(A$="W" AND X<14)+(A$="Z" AND X>1)
110 Y=Y-(A$="S" AND Y<30)+(A$="Q" AND Y>1)
120 POKE16384+B+32*A,239
130 C=PEEK(16384+Y+32*X)
135 IF C<>79 AND C<>88 THEN 150
140 X=A:Y=B
150 FOR I=1 TO 30:A$=INKEY$:IF A$="" THEN NEXT I
160 IF A$<>"0" OR Y=0 THEN 70
170 POKE16384+Y+32*X,79-9*(J=1)
180 SOUND200,2
190 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 ?

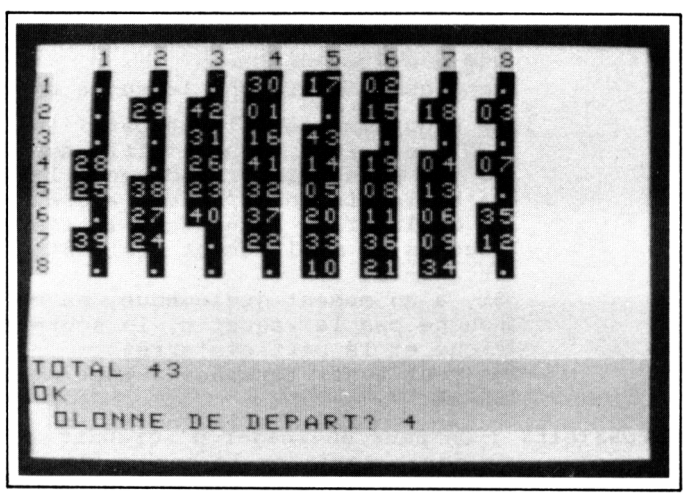
***LE PROGRAMME*** : Les lignes 4Ø et 5Ø définissent les divers déplacements possibles, dans deux variables alphanumériques A$ et B$ ; les deux boucles emboîtées I (6Ø-11Ø) et J (7Ø-1ØØ) dessinent le terrain de jeu.
On entre les coordonnées de la case de départ (ligne, puis colonne) aux lignes 12Ø et 13Ø.
La ligne 16Ø imprime le numéro du coup.
La ligne 17Ø permet le tirage au hasard d'un déplacement du cavalier.
La boucle Z (18Ø-23Ø) 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 19Ø et 2ØØ effectuent le déplacement.
La ligne 21Ø contrôle que l'on ne sorte pas du terrain de jeu.
La ligne 22Ø envoie en ligne 3ØØ si le coup est possible.
La ligne 25Ø gère la fin de partie (cavalier bloqué) et affiche le score.
Les lignes 3ØØ, 31Ø et 32Ø réalisent le coup et incrémente 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 ENTER., puis son numéro de colonne, suivi de ENTER. 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 BREAK.

***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, utiliser la couleur. 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 4
30 S=1:T=0
40 A$="56653223"
50 B$="653223"+A$+"65"
60 FOR I=1 TO LEN(A$)
70 FOR J=1 TO LEN(A$)
80 POKE16384+3*I,112+I:POKE16384+32*I,112+I
90 POKE16384+3*J+32*I,46
100 NEXT J
110 NEXT I
120 PRINT@384,"LIGNE DE DEPART";:INPUTI
130 PRINT@448,"COLONNE DE DEPART";:INPUTJ
140 A$=A$+A$
150 IF S>9 THEN S=0:T=T+1
160 POKE16384+3*J+32*I,S+48:POKE16383+3*J+32*I,T+48
170 Z=RND(8)
180 FOR Z=Z TO Z+8
190 A=I+VAL(MID$(A$,Z,1))-4
200 B=J+VAL(MID$(B$,Z,1))-4
210 IF A<1 OR B<1 OR A>8 OR B>8 THEN 230
220 IF PEEK(16384+3*B+32*A)=46 THEN 300
230 NEXT Z
250 PRINT@384,"TOTAL"T*10+S:SOUND190,15:END
300 I=A:J=B
310 S=S+1
315 SOUND(3*S+62),1
320 GOTO 150
```



# 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.

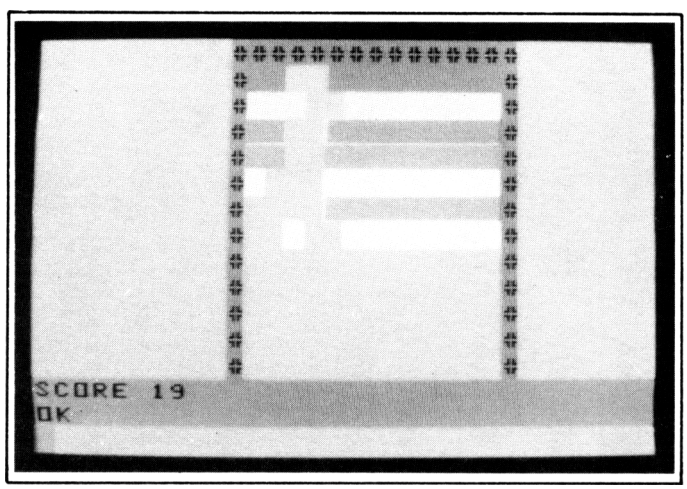
**LE PROGRAMME** La boucle I (4Ø-6Ø) dessine le terrain de jeu et les briques sont mises en place par les boucles emboîtées I (7Ø-11Ø) et J (8Ø-1ØØ).
La ligne 13Ø dessine la raquette dont la position est contrôlée par la ligne 15Ø.
Les lignes 2ØØ à 23Ø regardent si l'on a touché le mur, les briques ou la raquette.
Les rebonds sont gérés en conséquence avec des éventuels aléas, par les lignes 24Ø à 31Ø.
La ligne 17Ø 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 BREAK.

**EXTENSIONS POSSIBLES** : On peut envisager d'agrandir le terrain de jeu, augmenter le nombre de briques, de prévoir plusieurs balles, de dessiner

chaque rangée de briques d'une couleur différente, 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 CLS 8
30 PRINT@10,"################";
40 FOR I=1 TO 12
50 PRINT@32*I+10,"#";:PRINT@32*I+24,"#";
60 NEXT I
70 FOR I=1 TO 7
80 FOR J=11 TO 23
90 POKE16384+J+32*I,127+16*I
100 NEXT J
110 NEXT I
120 H=-1:K=-0.5:J=14:I=13:Y=J
130 PRINT@414+Y,"88888";:REM COULEUR 8 ; SHIFT Q , PUIS 3 FOIS
                                        SHIFT X , PUIS SHIFT Q
140 FOR P=1 TO 15:A$=INKEY$:IF A$="" THEN NEXT P
150 Y=Y-(A$="S" AND Y<23) + (A$="Q" AND Y>11)
160 IF T=1 THEN A=16384+J+32*I:POKEA,255
170 PRINT@416,"SCORE"S;
180 IF I>15 THEN  SOUND150,20:END
190 I=I+H:J=J+K
200 C=PEEK(16384+J+32*I)
210 IF C=255 THEN POKE16384+J+32*I,79:T=1:GOTO130
215 IF C=99 AND I=0 THEN 240
220 IF C=99 THEN 260
230 IF C>=143 AND C<=239 THEN POKE16384+J+32*I,255:T=1:GOTO 280
240 H=-H
250 K=RND(0)-RND(0)
260 K=-K:T=0
270 GOTO 130
280 S=S+1
290 K=-(K-(2*(RND(0)<0.5)+1)*RND(0)/3)
300 H=-H
305 SOUND230,1
310 GOTO 130
```



Achevé d'imprimer en mars 1984
sur les presses de l'imprimerie Laballery et Cie
58500 Clamecy
Dépôt légal : mars 1984

N° d'impression : 402094
N° d'édition : 86595-131-1
ISBN France : 2-86595-131-6
ISBN étranger : 2-86595-145-6



Enseigner et distraire est le double objectif de cet ouvrage. Ces 102 programmes de jeux vous entraînent à la découverte de L'ALICE et du MC-10. La classification par niveaux 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.

DES LIVRES POUR VOTRE ORDINATEUR

P.S.I.

ISBN : 2.86595.131.6
110 FF
****

Couverture GIBRAT

# 102 PROGRAMMES POUR ALICE ET MC-10

PAR JACQUES DECONCHAT

P.S.I.