P.S.I.

PAR
JACQUES
DECONCHAT

P.S.I.

102 PROGRAMMES POUR

VG 5000

Autre ouvrage relatif au VG 5000

— VG 5000 pour tous, Jean-Michel Jégo

Ouvrages d'initiation

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

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 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 : 2-86595-193-6

JACQUES
DECONCHAT

102 PROGRAMMES POUR
VG 5000

EDITIONS DU P.S.I.
1984

SOMMAIRE

Avant-propos 8
Organisation de l'ouvrage 9

Niveau 1

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

Niveau 2

Dimensionnement d'un tableau - Recherche d'un élément dans un tableau et dans une chaîne – L'instruction DATA 60
Dé truqué 62
Visions 64
421 66
L'électricien fou 68
La course au 20 70
Musique 72
Poker 74
Le jeu de saute-mouton 76
Lettres 78
Machinamot 80

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

Niveau 3

Manipulation des chaînes
et graphiques haute résolution 103
Un dé 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 1 130
Reines sur l'échiquier 132
Répétitions 134
Mots 136
Permutations 138
Bridge 140
Hanoï 142
Losanges 144
Mathieu 146
Le loup et les agneaux 148

Niveau 4

Les techniques d'animation élémentaire 152
Tirage de dés 154
Jeu de loto 156
La bombe 158
Ardoise magique 160
Affichage publicitaire 162
Evasion 164
Chapeaux 166
Combat aérien 168

Dessin de dés 170
Le jeu du missile 172
Le singe 174
Le jeu du 15 176
Adresse 178
Sous-marin 180
Zombies 182
Diago 184
Décomaque 186
Chasse au sous-marin 188
Clavier 190
Le lézard d'Alice 192
Calculatrice 194

Niveau 5

Les techniques spécifiques du VG 5000 198
Dames 199
Smokey 202
Course de voitures 204
Echecs 206
Danger spatial 208
Tic Tac Toe 210
Horloge 2 212
Anagrammes 214
Œufs 216
Taquin de lettres 218
Cibles 220
Gouttes 222
Obstacles 224
Captures dans l'espace 226
Anomalie 228
Plusieurs en un 230
Parcours du cavalier 232
Mur de briques 234
Othello 236
Quatre à quatre 238

Avant-propos

Vous trouverez dans ce livre plus de 100 programmes (essentiellement des jeux), rédigés en Basic Microsoft 80, pour l'ordinateur individuel VG 5000. La plupart de ces programmes seront d'ailleurs utilisables sans grande modification sur d'autres ordinateurs utilisant aussi un Basic Microsoft.

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

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

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

Les programmes n'utilisent pas toutes les particularités d'un appareil comme le VG 5000, par ailleurs très bien détaillées dans le manuel de base.

Le niveau 5 donne une idée sur l'utilisation de certaines techniques un peu spécifiques, comme l'utilisation de PEEK et POKE, ou les caractères graphiques utilisateur. Ces techniques pourront être reprises avec profit pour améliorer les programmes précédents.

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

Organisation de l'ouvrage

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

- **Le niveau 1** permet l'étude des instructions élémentaires du langage Basic : **LET** (optionnel ici), **PRINT, INPUT, IF...THEN, FOR...NEXT, GOTO, GOSUB, ON...GOTO, ON...GOSUB.**
- **Au niveau 2,** on utilisera les instructions de confection et de manipulation des éléments d'un tableau, et de recherche d'un élément dans une chaîne : **DIM, R(I), MID$, READ, DATA.**
- **Au niveau 3,** on utilise la panoplie complète des instructions de traitement des chaînes de caractères, et des instructions permettant des affichages plus spectaculaires : **MID$, RIGHT$, LEFT$.**
- **Le niveau 4** fait appel à une instruction très particulière qui facilite la réalisation de jeux interactifs : **KEY (O).**
- Enfin, **le niveau 5** nous fait découvrir des instructions plus spécifiques : définition des caractères graphiques utilisateur, identification d'objets sur l'écran, (SETET, PEEK, POKE, ...).

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

Le programme, fonctionnant sur le VG 5000, est ensuite reproduit, ainsi qu'un exemple d'exécution. Les listings ont été obtenus directement d'après les originaux de l'auteur, sur une imprimante matricielle, ce qui devrait garantir contre tout risque de disfonctionnement. Les minuscules (obtenues sur VG 5000 par une touche particulière) ne sont pas utilisées.

Niveau 1

Nom du programme	*Thèmes et particularités*
Le nombre mystérieux	Jeu de déduction, pour un joueur. Plusieurs niveaux.
Cube	Jeu de dessin : perspective sur un écran.
Chaud, froid	Logique et hasard, pour un joueur. Plusieurs niveaux.
Les allumettes	Jeu de réflexion, contre l'ordinateur.
Allumettes à plusieurs	Jeu de réflexion et de stratégie, pour plusieurs joueurs.
Calendrier perpétuel	Confection d'un calendrier, mois par mois.
Bataille navale	Un seul navire à trouver, mais quelques particularités.
Golf	Un parcours de 18 trous, pour un seul joueur.
Solfège	Apprenez le solfège à l'aide de votre ordinateur.
Le jeu du 21	Jeu de hasard, l'ordinateur tient la banque.
Pair - Impair	Un jeu de casino en trois temps (1 ou plusieurs joueurs).
Devine un chiffre	Un jeu très facile, pour un ou plusieurs joueurs.
Le jeu de Wytthof	Un jeu de réflexion assez simple, pour deux joueurs.
Rebonds	Un jeu d'adresse très visuel, pour un joueur.
Jackpot	Jeu de hasard, pour un joueur.
Roulette	Une simulation élégante du jeu très connu.
Couloirs	Un jeu de logique, à découvrir.
Echiquier	Une base graphique pour vos propres créations.
Le jeu de la fève	Un jeu de réflexion, pour deux joueurs.
Jeu de l'oie aléatoire	Un très vieux jeu, revu et corrigé par l'ordinateur, deux joueurs.

Instructions élémentaires

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

– ***RND (1) pour obtenir un nombre aléatoire entre*** Ø et 1.

– ***INT pour prendre la partie entière d'un résultat.***

– ***REM pour placer un commentaire.***

– ***END pour terminer un programme.***
etc.

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

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

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

L'INSTRUCTION LET OU INSTRUCTION D'AFFECTATION

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

- **Si la quantité est numérique,** le nom sera une simple lettre de l'alphabet, ou un ensemble de plusieurs chiffres ou lettres, commençant obligatoirement par une lettre.

- **Si la quantité est alphanumérique,** la même règle s'applique pour le nom, qui devra cependant être suivi du symbole $, et la variable devra être écrite entre guillemets.

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

Sur le VG 5000, 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 1re ligne, en haut, à gauche.

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

- **PRINT TAB(N)** provoque sur la ligne d'impression courante un affichage commençant à la énième colonne à partir de la gauche ($0 \leqslant N \leqslant 39$).

- Un ; (**point-virgule**) en fin de PRINT permet soit d'enchaîner une autre impression sans répéter l'instruction PRINT, soit d'indiquer que la prochaine impression sera, en l'absence d'indication contraire, effectuée exactement à la suite de l'impression précédente.

– La , (**virgule**) provoque une tabulation horizontale automatique prédéfinie.

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

LE POSITIONNEMENT DU CURSEUR SUR L'ECRAN

Pour afficher un texte commençant à un endroit précis de l'écran, le VG 5000 dispose de deux instructions de positionnement du curseur.

- **CURSOR X** permet de positionner le curseur dans une colonne X, dont le numéro doit être compris entre Ø et 39.

- **CURSOR Y** permet de positionner le curseur dans une ligne Y, dont le numéro doit être compris entre Ø et 24.

Ainsi, pour écrire le mot "**BONJOUR**" à la 5e colonne de la 8e ligne, on fera :

CURSOR X 5 : CURSOR Y 8 : PRINT "BONJOUR".

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 le VG 5000 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 **RET,** 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 **RECOMMENCEZ AU DEBUT** 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'utiliser les instructions CURSOR X n et CURSOR Y p.

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 : d'abord, en rupture de séquence conditionnelle, sous la forme :

IF condition = vraie THEN GOTO numéro de ligne.

qui peut être abrégée en :

IF␣ GOTO␣

C'est l'usage le plus courant de ce test, qui permet de renvoyer à une autre partie de 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 $, 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 (c'est-à-dire la quantité dont elle augmente ou diminue à chaque passage) pouvant aussi prendre ses valeurs dans les entiers ou les décimaux.

Ainsi, si l'on écrit :

```
FOR I = 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 une quantité quelconque. Mais, si l'on veut de nouveau revenir dans la même boucle, il faudra assigner à I la valeur de fin de boucle, (par exemple, ici I = 2Ø) sinon, un message d'erreur sera affiché.

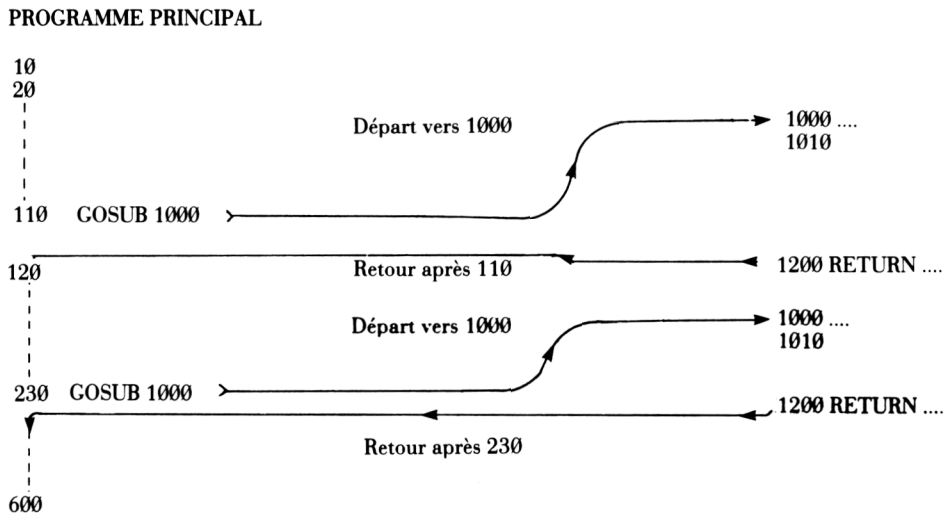
Si ***le pas de la boucle (STEP)*** n'est pas précisé, il sera automatiquement pris égal à 1.

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

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

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

Voici le schéma théorique de fonctionnement :



FIN DU PROGRAMME PRINCIPAL

```
1000
 |
 |
1200  RETURN
```

Sous-Programme 1.

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

Un sous-programme pourra lui-même éventuellement appeler un autre sous-programme, et ainsi de suite.
On peut aussi appeler un sous-programme de façon conditionnelle, en utilisant l'instruction ON␣ GOSUB␣.
Par exemple ON X GOSUB 1000, 2000, 3000 permettra d'exécuter le sous-programme placé en 1000 si X = 1, en 2000 si X = 2, en 3000 si X = 3.

Le nombre mystérieux

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

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

LE PROGRAMME :
La ligne 5Ø permet de fabriquer le nombre à deviner.
Les entrées sont faites par les lignes 7Ø à 8Ø.
La ligne 9Ø teste si le résultat est exact.
La ligne 1ØØ signale un essai trop petit, et la ligne 11Ø un essai trop grand.
Le compteur d'essais est en ligne 12Ø.
Les lignes 2ØØ à 23Ø 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 RET.
On obtient alors l'affichage ESSAI N° 1, suivi de la question VOTRE NOMBRE.
On entre un nombre, et l'on presse RET.
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 RET).
On peut arrêter à tout moment en tapant CTRL et △.

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

```
10 REM LE NOMBRE MYSTERIEUX
20 INIT 4,2
30 INPUT"NOMBRE DE CHIFFRES :";N
40 IF N>7 THENRUN
50 X=INT(RND(1)*10^N+1)
60 J=1
70 PRINT:PRINT"ESSAI NO. ";J
80 INPUT"VOTRE NOMBRE :";M
90 IF M=X THENGOTO 200
100 IF M<X THENPRINT"TROP PETIT"
110 IF M>X THENPRINT"TROP GRAND"
120 J=J+1
130 PRINT
140 GOTO 70
200 INIT 4,7
210 PRINT"VOUS AVEZ TROUVE ";X;" EN ";J;
" ESSAIS"
220 INPUT"AUTRE PARTIE (O/N) :";R$
230 IF R$="O" THENRUN
```

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 signes +, en utilisant deux boucles : la boucle I (3Ø-12Ø), et la boucle I (13Ø-18Ø).
Le nom de l'objet représenté est affiché en ligne 19Ø (ici : CUBE).

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

EXTENSIONS POSSIBLES :
La haute résolution permettra d'avoir un tracé plus précis.
On pourra imaginer un jeu utilisant des parallélépipèdes rectangles de différentes tailles, avec affichage par exemple des 3 côtés, et il faudrait trouver leur volume, etc.

```
10 REM CUBE
20 INIT6,2
30 FOR I=1 TO 10
40 CURSORY2:CURSORX7+I:PRINT"+"
50 CURSORY8:CURSORXI:PRINT"+"
60 CURSORY11:CURSORX7+I:PRINT"+"
70 CURSORY17:CURSORXI:PRINT"+"
80 CURSORY7+I:CURSORX1:PRINT"+"
90 CURSORY7+I:CURSORX11:PRINT"+"
100 CURSORY1+I:CURSORX7:PRINT"+"
110 CURSORY1+I:CURSORX17:PRINT"+"
120 NEXT I
130 FOR I=1 TO 7
140 CURSORY9-I:CURSORXI:PRINT"+"
150 CURSORY9-I:CURSORX10+I:PRINT"+"
160 CURSORY18-I:CURSORXI:PRINT"+"
170 CURSORY18-I:CURSORX10+I:PRINT"+"
180 NEXT I
190 CURSORX25:CURSORY18:PRINT"CUBE"
```

Chaud, froid

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

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

LE PROGRAMME :
Les lignes 40 et 50 définissent la position de l'objet.
La ligne 90 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 RET.
Dès l'affichage suivant (DONNEZ X), on entre la coordonnée X supposée, suivie de RET.
On obtient ensuite l'affichage DONNEZ Y, et l'on entre Y suivi de RET.
On obtient en réponse un texte, qui est variable selon que l'on sera plus ou moins loin de l'objet cherché (cela va de COMPLETEMENT GELE à BRULANT). Lorsque l'on a trouvé, on obtient l'affichage du nombre d'essais effectués.
Pour sortir du programme avant la fin de la partie, il suffit de taper CTRL et △.

EXTENSIONS POSSIBLES :
Si l'on dispose de la couleur, il sera intéressant de trouver un moyen plus visuel pour symboliser la distance, par exemple des pavés allant du rouge au bleu, aux emplacements essayés.
On peut aussi travailler dans un espace à trois dimensions, cacher plusieurs objets, etc.

```
10 REM CHAUD,FROID
20 INIT 2,5
30 INPUT"DIMENSION DE LA GRILLE :";D
40 X=INT(RND(1)*D+1)
50 Y=INT(RND(1)*D+1)
55 INIT 4,1
60 INPUT"DONNEZ X :";A
70 INPUT"DONNEZ Y :";B
80 S=S+1
90 T=ABS(X-A)+ABS(Y-B)
100 IF T=0 THENGOTO 6000
110 PRINT"VOUS ETES ";
120 N=INT(SQR(D)/T)+1
130 IF N>5 THENN=5
140 IF T<=SQR(D) THENON N GOTO 1000,2000
,3000,4000,5000
150 PRINT"COMPLETEMENT GELE"
160 GOTO 60
1000 PRINT"A PEINE TIEDE"
1010 GOTO 60
2000 PRINT"TIEDE"
2010 GOTO 60
3000 PRINT"CHAUD"
3010 GOTO 60
4000 PRINT"TRES CHAUD"
4010 GOTO 60
5000 PRINT"BRULANT"
5010 GOTO 60
6000 INIT5,2
6010 PRINT"VOUS AVEZ TROUVE EN ";S;" COU
PS"
```

Les allumettes

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : contre le VG 5000.
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 le VG 5000. Chacun prend à son tour autant d'allumettes qu'il le désire, dans la limite du nombre maximum autorisé.
Celui qui prend la dernière allumette est considéré comme perdant.

LE PROGRAMME :
Le jeu du joueur commence en ligne 7Ø.
Celui du VG 5000 commence en ligne 11Ø.
La ligne 8Ø contrôle la légalité du coup du joueur.
La ligne 9Ø gère le changement de joueur.
Le coup du VG 5000 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 RET. Puis on entre la prise maximum autorisée, suivie de RET. Si le joueur veut commencer, il frappe ensuite Ø ; sinon il frappe 1, puis RET.
Après chaque coup, le nombre d'allumettes restantes est affiché, et le VG 5000 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 INIT 3,6
30 INPUT"NOMBRE TOTAL :";N
40 INPUT"PRISE MAXIMUM :";I
50 INPUT"QUI COMMENCE (VOUS:0 / MOI:1) :
";D
60 IF D=1 THENGOTO 110
70 INPUT"VOUS RETIREZ :";P
80 IF P<>INT(P) OR P>N THENGOTO 70
90 D=1
100 GOTO 1000
110 S=I+D
120 T=(N-S)/(I+D)
130 IF T=INT(T) THENGOTO 200
140 S=S-D
150 GOTO 120
200 P=ABS(S-D)
210 IF P=0 THENP=D
220 D=0
1000 PRINT P
1010 N=N-P
1020 PRINT"IL RESTE ";N
1030 IF N=0 THENGOTO 2000
1040 GOTO 60
2000 INIT 7,5
2010 IF D=1 THENPRINT"J'AI GAGNE":PLAY"A
BCDE"
2020 IF D<>1 THENPRINT"BRAVO":PLAY"T1150
2CCC"
```

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 au VG 5000 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 30 à 40 permettent les différentes entrées.
Les lignes 70 à 100 réalisent les impressions.
La ligne 120 contrôle le coup de chaque joueur (par rapport au coup précédent).
La ligne 180 gère les tours (remise à 1 du numéro du joueur).
La ligne 160 réalise le test de fin, et la ligne 210 affiche le numéro du joueur gagnant (avec la règle prévue initialement).

POUR JOUER :
Après avoir lancé le programme par RUN, on entre le nombre d'allumettes initial, suivi de RET, puis le nombre de joueurs (et RET).
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 le VG 5000.
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 INIT 3,6
30 INPUT"NOMBRE TOTAL D'ALLUMETTES :";N
40 INPUT"NOMBRE DE JOUEURS :";J
50 F=1
60 K=1
70 INIT 6,3
80 PRINT"TOUR DU JOUEUR :";K
90 PRINT"TOTAL RESTANT :";N
100 PRINT"MAXIMUM A ENLEVER :";F
110 INPUT"VOTRE COUP :";C
120 IF C=0 OR C>F THENGOTO 110
130 N=N-C
140 IF N<0 THENN=N+C:GOTO 110
150 F=C+C
160 IF N=0 THENGOTO 200
170 K=K+1
180 IF K>J THENK=1
190 GOTO 70
200 INIT 1,4
210 PRINT"LE JOUEUR NUMERO ";K;" A GAGNE
":PLAY"CEG"
```

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 1983, suivie de RET.
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 RET, 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 INIT 6,7
30 INPUT"POUR QUELLE ANNEE :";A
40 INPUT"POUR QUEL MOIS    :";M
50 D=INT(365.25*(A+(M<3)))+1
60 D=D+INT(30.6*(M+1-(M<3)*12))-INT((INT
((A+(M<3))/100)-7)*.75)
70 D=D-7*INT(D/7)
80 M=31+(M=4 OR M=6 OR M=9 OR M=11)+(M=2
)*3
90 PRINT
100 PRINT"DIM LUN MAR MER JEU VEN SAM"
110 PRINT
120 FOR A=1 TO M
130 PRINT TAB(4*(D+A-1));A;
140 IF D+A-1=6 THENPRINT:D=D-7
150 NEXT A
160 PLAY"CEG"
```

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 1Ø cases sur 1Ø cases.
A chaque coup, on devra tirer en donnant les coordonnées H (horizontale) et V (verticale) de la case visée, et le VG 5000 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 (4Ø-7Ø) dessine le terrain de jeu.
Les lignes 9Ø-1ØØ définissent l'emplacement du navire.
Les coordonnées du tir sont entrées aux lignes 11Ø et 12Ø, et la ligne 13Ø affiche le coup sur l'écran. La distance est calculée en ligne 14Ø, et la ligne 15Ø regarde si l'on a ou non fait mouche.
La ligne 17Ø intervient pour déplacer le navire si les coups sont rapprochés.
La ligne 18Ø 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 RET et, à la question COLONNE, on répond en donnant le numéro de la colonne verticale suivi de RET.
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 le VG 5000, mettre plusieurs navires, des sous-marins, etc.

```
10 REM BATAILLE NAVALE
20 INIT6,4
30 PRINT "    0  1  2  3  4  5  6  7  8
 9"
40 FOR I=0 TO 9
50 PRINT I;"  +  +  +  +  +  +  +  +  +
 +"
60 PRINT
70 NEXT I
80 S=1
90 X=INT(RND(1)*10)
100 Y=INT(RND(1)*10)
110 CURSORX1:CURSORY21:INPUT"LIGNE :";H
120 CURSORX20:CURSORY21:INPUT"COLONNE :"
;V
130 CURSORX3*V+4:CURSORY2*H+1:TX1,4,0:PR
INTS:SOUND200,2,8:TX0
140 D=INT(SQR((X-H)*(X-H)+(Y-V)*(Y-V)))
150 IF D=0 THEN GOTO 200
160 S=S+1
170 IF D=1 THEN CURSORX1:CURSORY22:PRINT
"PFFFFF...                    ":GOTO 90
180 CURSORX1:CURSORY22:PRINT"DISTANCE
  :";D;"   "
190 CURSORX1:CURSORY21:PRINT"
                        ":GOTO 110
200 INIT6,4
210 PRINT"TROUVE EN ";S;" COUPS":PLAY"AE
C"
```

Golf

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

LE JEU :
Il s'agit d'essayer de réaliser, avec le moins de coups possible, un parcours de golf de 18 trous. Pour cela, il vous faut frapper la balle avec une force plus ou moins grande, jusqu'à obtenir sa chute dans le trou.
Mais attention, il peut y avoir 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 7Ø permet de définir la position du trou, et la ligne 8Ø dessine le terrain de jeu.
Le déplacement à réaliser est calculé en ligne 12Ø.
La boucle I (14Ø-19Ø) réalise le déplacement de la balle.
La ligne 2ØØ regarde si le coup est, ou non, gagnant ; s'il l'est, on recommence (ligne 25Ø) jusqu'à ce que l'on ait réalisé 18 trous (boucle S (4Ø-25Ø) ; fin de boucle en 27Ø, 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 RET, la balle recommence à se déplacer ; elle peut alors soit s'arrêter avant le trou (auquel cas on recommence), soit tomber dedans (on passe alors au trou suivant), soit le dépasser : on devra alors frapper avec une force négative.
Le jeu se termine avec l'affichage 18 TROUS EN ... COUPS.

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

```
10 REM GOLF
20 INIT 6,3
40 FOR S=1 TO 18
50 INIT6,1
60 X=1
70 T=INT(29*RND(1)+9)
80 CURSORY6:PRINT"======================
==============="
90 CURSORXT:CURSORY6:PRINT" ";:CURSORXX:
CURSORY5:PRINT"O"
100 CURSORX15:CURSORY8:PRINT"
      "
110 CURSORY8:PRINT"TROU ";S;:INPUT"   FO
RCE ";R
120 F=INT(R*0.16-(ABS(R)*(SQR(RND(1))+0.
1)^2)/1000)
130 IF F=0 THENGOTO 100
140 FOR I=SGN(F) TO F STEP SGN(F)
150 X=X+SGN(F)
160 IF X<39 AND X>=1 THENCURSORXX-SGN(F)
:CURSORY5:PRINT" ":CURSORXX:CURSORY5:PRI
NT"O"
170 IF X>=38 THENX=38
180 IF X<=1 THENX=1
190 NEXT I
200 IF X=T THENGOTO 230
210 E=E+1
220 GOTO 80
230 CURSORXX:CURSORY5:PRINT" ":CURSORXX:
CURSORY6:PRINT"O":PLAY"ACA"
240 FOR P=1 TO 400:NEXT P
250 NEXT S
260 INIT 2,0
270 PRINT"18 TROUS EN ";E;" COUPS":PLAY"
ACE"
```

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 (5Ø-7Ø) dessine une portée.
La note est choisie en ligne 8Ø et affichée en lignes 9Ø-1ØØ-11Ø.
La ligne 12Ø affiche le score.
La ligne 13Ø affiche le nom de la clé (ici clé de sol) et vous interroge.
La ligne 14Ø s'occupe de la fin de la partie.
Les lignes 15Ø et 16Ø effectuent le décodage de la note entrée.
La ligne 19Ø 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 RET.
Le score vous indiquera si c'est exact ou non, et une nouvelle note sera affichée.
Pour arrêter, il suffira de répondre Ø suivi de RET, 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
30 P=0
40 INIT1,1
50 FOR I=1 TO 5
60 CURSORX1:CURSORY2*I+1:GR0,1,0:PRINT"L
LLLLLLLLLLLLLLLLLLLLLLLLLL";:TX0,0,0
70 NEXT I
80 X=INT(RND(1)*12)
90 CURSORX18:CURSORYX:PRINT"¦"
100 CURSORX18:CURSORYX+1:PRINT"¦"
110 CURSORX17:CURSORYX+2:PRINT"O"
120 CURSORY21:PRINT"SCORE ";S;" SUR ";P;
" ESSAIS"
130 INPUT"CLE DE SOL ; QUELLE EST LA NOT
E :";R$
140 IF R$="0" THENEND
150 Y=(1 AND R$="FA")+(2 AND R$="MI")+(3
 AND R$="RE")+(4 AND R$="DO")
160 Y=Y+(5 AND R$="SI")+(6 AND R$="LA")
170 P=P+1
180 A$=CHR$(76-X):IF X<7 THEN A$=CHR$(71
-X):PLAY"O4"+A$
190 IF Y=X OR Y=X-7 THENS=S+1
200 IF X>=7 THEN PLAY"O3"+A$
210 GOTO 40
```

Le jeu du 21

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu contre le VG 5000.
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 le VG 5000 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 8Ø et 9Ø réalisent le contrôle de celui qui doit jouer (joueur ou VG 5000).
Le sous-programme (1ØØØ-1Ø1Ø) lance les dés.
Les lignes 11Ø-12Ø calculent le total fait par le joueur et l'affichent.
Le total du VG 5000 est calculé en ligne 19Ø et affiché par la ligne 2ØØ.
La ligne 22Ø contrôle la fin du jeu pour le VG 5000.
Les lignes 26Ø-29Ø affichent les résultats, et la ligne 3ØØ cherche quel est le gagnant.
Le gain (ou la perte) sera affiché par la ligne 32Ø, et le total par la ligne 35Ø.
La ligne 38Ø contrôle la fin de partie.

POUR JOUER :
Taper RUN pour lancer le jeu.
A la question VOTRE MISE, entrer la mise initiale, suivie de RET, et taper 1 pour lancer le dé. Le premier tirage est affiché sous la forme VOUS AVEZ... et on devra refaire 1 suivi de RET, tant que l'on désire recommencer à tirer.
Dès que l'on a terminé, on tape Ø et RET pour laisser jouer le VG 5000 (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
50 INIT6,7
60 T=0
70 INPUT"VOTRE MISE :";M
80 INPUT"FAIRE 1 POUR TIRER, 0 POUR FINI
R :";R
90 IF R=0 THENGOTO 150
100 GOSUB 1000
110 T=T+X:SOUND200,2,5
120 PRINT"VOTRE TOTAL EST DE ";T
130 GOTO 80
150 INIT3,1
160 PRINT"LA BANQUE JOUE"
170 S=0
180 GOSUB 1000
190 S=S+X:SOUND140,1,8
200 PRINT"TOTAL DE LA BANQUE ";S
210 FOR I=1 TO 400:NEXT I
220 IF S>T OR S>17 THENGOTO 250
230 GOTO 180
250 INIT2,6:PLAY"ABC"
260 PRINT"VOUS AVEZ ";T
270 PRINT
280 PRINT"LA BANQUE A ";S
290 PRINT
300 IF S>21 OR (S<T AND T<=21) THENGOTO
320
310 M=-M
320 PRINT"GAIN ";M
330 PRINT
340 G=G+M
350 PRINT"VOTRE TOTAL ACTUEL :";G
360 PRINT
370 INPUT"ON CONTINUE (O/N) :";R$
380 IF R$="N" THENEND
390 GOTO 50
1000 X=INT(RND(1)*6+1)
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. Le VG 5000 lance simultanément deux dés, non visibles et pose trois questions : quelle est la parité du total, quel est le total, et enfin donner la valeur de l'un des dés. On peut ensuite soit poursuivre le jeu, soit l'interrompre à tout moment.

LE PROGRAMME :
Les dés sont tirés par un sous-programme placé aux lignes 1000 et 1010.
La ligne 80 tient le compte du nombre de parties, et les lignes 180, 260 et 290 comptabilisent les points marqués.
La parité du total est déterminée par les lignes 130 et 140 (la ligne 140 compare avec la réponse du joueur).
La ligne 400 et les suivantes réalisent l'affichage de fin de partie : total, nombre de points, nombre de parties jouées.
La ligne 440 provoque l'arrêt si l'on répond N à la question ON CONTINUE.

POUR JOUER :
On lance le programme par l'instruction RUN.
Il suffit ensuite de répondre aux diverses questions posées, en suivant scrupuleusement les indications données et en tapant RET après chaque réponse.
Lorsque l'on a terminé une partie, on obtient l'affichage du total actuel et du nombre de parties sous la forme de, par exemple, 6 SUR 10.
On obtient alors l'affichage de la question ON CONTINUE (O/N) et l'on devra répondre N suivi de RET pour arrêter.

EXTENSIONS POSSIBLES :
Si l'on désire jouer à plusieurs, il sera intéressant de prévoir la possibilité de miser, et d'ajouter une gestion du score réalisé par chacun des joueurs.
On pourra aussi prévoir une amélioration sur le plan de la présentation (utilisation de la couleur, etc.).

```
10 REM PAIR-IMPAIR
20 T=0
30 R=0
50 GOSUB 1000
60 A=X
70 GOSUB 1000
80 R=R+1
90 S=A+X
100 INIT7,6
110 INPUT"TOTAL PAIR (0) OU IMPAIR (1) :
";N
120 PRINT
130 P=S/2-INT(S/2)
140 IF N=P OR N-0.5=P THENGOTO 170
150 PRINT"FAUX":PLAY"AAA"
160 GOTO 190
170 PRINT"EXACT":PLAY"CEG"
180 T=T+1
190 PRINT
200 INPUT"TOTAL PRESUME :";N
210 IF N=S THENGOTO 250
220 PRINT:PRINT"PERDU":PLAY"ACA"
230 FOR I=1 TO 1500:NEXT I
240 GOTO 400
250 PRINT
260 T=T+2
270 INPUT"BRAVO - DONNEZ UN DE :";N
280 IF N<>A AND N<>S-A THENGOTO 220
290 T=T+3
300 PRINT
310 PRINT"BRAVO":PLAY"ABCDE":GOTO 230
400 INIT 2,0
410 PRINT"VOUS AVEZ UN TOTAL DE ";T;" SU
R ";R
420 PRINT
430 INPUT"ON CONTINUE (O/N) :";R$
440 IF R$="N" THENEND
450 GOTO 50
1000 X=INT(RND(1)*6+1)
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 deux fois il double sa mise ; s'il apparaît trois fois, il triple sa mise.

LE PROGRAMME :
Le tirage des dés est réalisé par un sous-programme placé en lignes 1000 et 1010.
L'affichage est réalisé par les lignes 150 et 160.
Les lignes 180 à 210 calculent le gain dans les diverses configurations possibles.
La totalisation est effectuée en ligne 230 pour les gains, et en ligne 510 pour les pertes.
La variable R$, testée en ligne 550, détermine la poursuite éventuelle du jeu.

POUR JOUER :
Le lancement du programme par RUN provoque un premier affichage VOTRE MISE ; on répond par un nombre suivi de RET. On obtient alors un deuxième affichage DONNEZ UN CHIFFRE DE 1 A 6 ; on entre un chiffre suivi de RET ; 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 RET).

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 le VG 5000.

```
10 REM DEVINE UN CHIFFRE
20 G=0
30 INIT7,6
40 INPUT"VOTRE MISE :";N
50 PRINT
60 INPUT"DONNEZ UN CHIFFRE DE 1 A 6 :";D
80 IF D<>INT(D) OR D<1 OR D>6 THENRUN
90 GOSUB 1000
100 A=X
110 GOSUB 1000
120 B=X
130 GOSUB 1000
140 C=X
150 PRINT:PRINT:DELIM1,1,0:PRINT"   A
       B            C"
160 PRINT:PRINT"   ";A;"           ";B;"
       ";C
170 T=0
180 IF D=A THENT=T+N:PLAY"GA"
190 IF D=B THENT=T+N:PLAY"GA"
200 IF D=C THENT=T+N:PLAY"GA"
210 IF T=0 THENGOTO 500
220 PRINT:PRINT:PRINT"VOUS GAGNEZ ";T
230 G=G+T
240 GOTO 520
500 PRINT:PRINT:PRINT"PERDU":PLAY"AAA"
510 G=G-N
520 PRINT:PRINT:PRINT"TOTAL ACTUEL :";G
530 PRINT:PLAY"ACE"
540 INPUT"ON CONTINUE (O/N) :";R$
550 IF R$="N" THENEND
560 GOTO 30
1000 X=INT(RND(1)*6+1)
1010 RETURN
```

Le jeu de Wytthof

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

LE JEU :
On dispose d'une grille carrée de 1Ø cases sur 1Ø 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 droite et bas. Le premier joueur qui arrive en (9,9) sera considéré comme gagnant, ou perdant, selon la convention faite au début.

LE PROGRAMME :
Les lignes 4Ø, 5Ø et 6Ø permettent de définir une position de départ, en dehors de la diagonale.
La ligne 8Ø et la boucle I (9Ø-11Ø) dessinent le terrain de jeu.
La ligne 13Ø imprime la position du point mobile.
Le test de fin est effectué en ligne 14Ø.
Le changement de joueur est assuré par la ligne 18Ø.
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 RET.
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 au VG 5000 de jouer, etc.

```
10 REM LE JEU DE WYTTHOF
40 X=3*INT(RND(1)*5)+8
50 Y=INT(RND(1)*5)
60 IF 3*Y+8=X THENGOTO 50
70 INIT 5,1
80 DELIM1,1,1:PRINT"     0  1  2  3  4
5  6  7  8  9":DELIM5,5,0
90 FOR I=0 TO 9
100 PRINT:DELIM1,1,1:PRINTI;:DELIM5,5,0:
PRINT"  +  +  +  +  +  +  +  +  +  +"
110 NEXT I
120 H=1
130 CURSORXX-1:CURSORY2*Y+2:PRINTCHR$(12
7)
140 IF X=35 AND Y=9 THENPRINT"C'EST FINI
":PLAY"CDE":END
150 CURSORY22:PRINT"JOUEUR ";H;:INPUT"-
DIRECTION (1,2,3) :";D$
160 D=VAL(D$):PLAY"AA"
170 ON D GOSUB 1000,2000,3000
180 H=2-ABS(H-1)
190 GOTO 130
1000 X=X+3
1010 RETURN
2000 GOSUB 1000
3000 Y=Y+1
3010 RETURN
```

Rebonds

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

LE JEU :
Il utilise l'idée de la simulation d'un billard à bandes.
De part et d'autre de l'écran se trouvent la balle et le but visé : une série de chiffres et de lettres, en haut de l'écran, permettent de choisir l'endroit où l'on désire rebondir.
La balle va alors partir vers le haut de l'écran (vers le bas si l'on introduit un signe moins), y rebondir, revenir éventuellement frapper le bas, rebondir, et ainsi de suite jusqu'à atteindre la ligne de but ; si l'on est suffisamment près de la cible on a gagné et le nombre de rebonds sera alors affiché.

LE PROGRAMME :
La ligne 5Ø imprime les chiffres et les lettres en haut de l'écran.
Les lignes 6Ø et 7Ø définissent les positions initiales de la balle et de la cible.
Les lignes 8Ø et 9Ø effectuent leur impression.
Le symbole correspondant à la direction choisie est entré en ligne 1ØØ.
La boucle J (13Ø-17Ø) réalise le déplacement de la balle.
La ligne 16Ø imprime la nouvelle position.
Le changement de direction, lors de la rencontre d'un bord, est effectué par la ligne 15Ø, par renvoi dans une partie du programme indépendante, située aux lignes 22Ø à 24Ø.
La ligne 18Ø effectue l'impression de fin de partie.
Le nombre de rebonds est compté en ligne 23Ø.

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 RET.
La trajectoire est alors affichée, suivie s'il y a lieu de la mention BRAVO ; X REBONDS.
On peut lancer une autre partie en tapant une touche quelconque, et s'arrêter à tout moment en tapant CTRL et △.

EXTENSIONS POSSIBLES :
Il sera très intéressant, en utilisant un graphisme plus fin, de dessiner un véritable billard, avec son tapis vert, et d'indiquer la direction par un segment, symbolisant la queue de billard, se déplaçant autour de la bille. On peut aussi ajouter des trous, mettre plusieurs billes, sonoriser les rebonds, ou les heurts avec d'autres balles, les comptabiliser pour faire une partie à plusieurs joueurs, etc.

```
10 REM REBONDS
40 INIT3,3
50 DELIM3,2,0:PRINT"-0123456789ABCDEFGHI
JKLMNOPQRSTUVWXYZ-";:DELIM3,3,0
60 X=INT(RND(1)*18+2)
70 Y=INT(RND(1)*18+2)
80 CURSORYX:DELIM3,1,0:PRINT"*";:DELIM3,
3,0
90 CURSORX38:CURSORYY:DELIM3,5,0:PRINT"+
";:DELIM3,3,0
100 CURSORY23:INPUT"VOTRE COUP :";R$
110 S=0
120 T=X
130 FOR J=3 TO 37
140 T=T-X/(ASC(R$)-ASC("0")+7*(R$>="A"))
150 IF T>=22 OR T<=1 THENGOTO 220
160 CURSORXJ:CURSORYT:DELIM3,1,0:PRINT"*
";:DELIM3,3,0:DISPLAY20:PLAY"G":FORK=1TO
100:NEXTK
170 NEXT J
180 IF ABS(T-Y)<3 THENPRINT"BRAVO :";S;"
 REBONDS":PLAY"CEG"
190 CURSORX1:CURSORY22:INPUT"AUTRE PARTI
E (O/N) :";R$
200 IF R$="N" THENEND
210 RUN
220 X=-X
230 S=S+1
240 GOTO 170
```

Jackpot

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

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

LE PROGRAMME :
La boucle I (6Ø-8Ø) permet le dessin de la machine, la ligne 9Ø affichant les emplacements (vides) des chiffres.
La mise sera entrée en ligne 1ØØ, et testée en ligne 11Ø : si on mise Ø, la partie est terminée (départ vers la ligne 2ØØØ).
Les lignes 12Ø, 14Ø et 16Ø permettent le tirage des 3 chiffres, en appelant un sous-programme qui se trouve en ligne 1ØØØ.
Le résultat est affiché par la ligne 17Ø, puis testé en ligne 18Ø, 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 lequel vos mises seront décomptées.
Puis on tape RET, et on entre une première mise (différente de Ø), suivie de RET.
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 RET 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
40 INPUT"POT DE DEBUT :";P
50 INIT6,4
60 FOR I=1 TO 3
70 CURSORX8:CURSORYI+4:DELIM1,1,0:PRINT"
                              ";:DELIM6,6,0
80 NEXT I
90 CURSORX11:CURSORY6:DELIM2,2,0:PRINT"
   ";:DELIM1,1,0:CURSORX19:DELIM2,2,0:PR
INT"    ";:DELIM1,1,0
95 CURSORX27:DELIM2,2,0:PRINT"    ";:DEL
IM1,1,0
100 CURSORX1:CURSORY11:INPUT"VOTRE MISE
:";M
110 IF M=0 OR M>P THENGOTO 2000
120 GOSUB 1000
130 A=X
140 GOSUB 1000
150 B=X
160 GOSUB 1000
170 CURSORY6:CURSORX12:PRINTA;" ";:CURSO
RX20:PRINTB;" ";:CURSORX28:PRINTX;" "
180 IF A=B AND A=X THENGOTO 300
190 N=-M
200 CURSORY13:PRINT"MISE :";M;" - RESULT
AT :";N;"    "
210 INPUT"TAPER - RET - POUR CONTINUER";
R$
220 P=P+N
230 GOTO 50
300 N=M*50:PLAY"ACE"
310 IF A=7 THENN=M*500:PLAY"ABCDEFG"
320 GOTO 200
1000 X=INT(RND(1)*9+1):SOUND255-(X+9)*10
,10,4
1010 RETURN
2000 INIT7,4
2010 PRINT"VOUS AVEZ ";P
2020 PRINT"A BIENTOT":PLAY"ACE"
```

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 boule K (3Ø-9Ø) dessine le terrain de jeu, auquel la ligne 1ØØ rajoute les trois symboles P, ØØ et I.
La boucle K (11Ø-22Ø) réalise le déplacement de la boule, l'arrêt étant contrôlé par la ligne 19Ø.
La ligne 18Ø réalise une légère temporisation.
La ligne 23Ø tire la position d'arrêt.
La ligne 25Ø gère les essais suivants.

POUR JOUER :
Il suffit de faire RUN, et de regarder.
Une fois la boule arrêtée, il suffira de frapper une touche quelconque (sauf "N") pour la relancer. Taper N pour sortir du programme.

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

```
10 REM ROULETTE
20 INIT2,4
30 FOR K=1 TO 4
40 FOR I=1 TO 9
50 S=I+I
60 PRINT TAB(4*I)(S AND K=1)+(S+18 AND K
=2)+(S-1 AND K=3)+(S+17 AND K=4);
70 NEXT I
80 PRINT:PRINT:PRINT:PRINT
90 NEXT K
100 CURSORY2:PRINT"P":CURSORY6:PRINT"00"
:CURSORY10:PRINT"I"
110 FOR K=1 TO 4
120 FOR I=1 TO 10
130 IF I=10 AND K<4 THENA=1:B=4*K-1:GOTO
 160
140 A=4*I+1:B=4*K-3
150 IFK*I=40 THENGOTO 210
160 CURSORXA:CURSORYB:PRINTCHR$(127)
170 X=X-1:SOUND250,2,1
180 FOR J=1 TO 400:NEXT J
190 IF X=1 THENPLAY"ACE":GOTO 250
200 CURSORXA:CURSORYB:PRINT" "
210 NEXT I
220 NEXT K
230 X=INT(RND(1)*38)
240 GOTO 100
250 CURSORY20:INPUT"AUTRE ESSAI (O/N) :"
;R$
260 IF R$="N" THENEND
270 RUN
```

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 le VG 5000 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 du VG 5000.
Dans la version retenue ici, la réponse est alors immédiatement affichée, et le chemin suivi depuis le début du jeu, ainsi que les sorties correspondantes, sont affichés sur l'écran. Ceci présente l'avantage de permettre à plusieurs joueurs d'exercer leur sagacité en essayant de prévoir, pour une entrée donnée, ce que va répondre le VG 5000.

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 INIT7,6:PRINT:PRINT
30 GR1,7,0:PRINT"LLLLLLLLLLLLLLLLLLLLL
LL";:TX0:PRINT
40 PRINT:PRINT"B      ";:GR1,7,0:PRINT"LL
LLLLLLLLL";:TX0,0,0:PRINT"  X    2"
50 PRINT:GR1,7,0:PRINT"LLLLLLL";:TX0,0,0
:PRINT"  X      ";:GR1,7,0:PRINT"LLLLLLLL
LL";:TX0:PRINT
60 PRINT:PRINT"A      ";:GR1,7,0:PRINT"LL
LLLLLLLLL";:TX0,0,0:PRINT"  X    1":PRIN
T
70 GR1,7,0:PRINT"LLLLLLLLLLLLLLLLLLLLL
LL";:TX0,0,0
80 I=0:J=0:K=0:CLEAR 500
90 R$="":T$=""
100 CURSORX1:CURSORY13:INPUT"ENTREE 1 OU
 2 :";C$
110 IF C$="" OR C$>="3" OR LEN(C$)<>1 TH
ENGOTO 100
120 R$=R$+C$
130 PRINT:PRINT R$
140 IF C$="1" THENGOTO 190
150 J=NOT(J)
160 IF J<>0 THENGOTO 250
170 S$="B":PLAY"E"
180 GOTO 220
190 I=NOT(I)
200 IF I<>0 THENGOTO 250
210 S$="A":PLAY"A"
220 T$=T$+S$
230 CURSORY17:PRINT"SORTIE EN ";S$:PRINT
:PRINT T$
240 GOTO 100
250 K=NOT(K)
260 IF K<>0 THENGOTO 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 : il donne, en effet, la possibilité de réaliser un affichage graphique de bonne qualité pour des jeux utilisant un damier de 8 cases (on pourra facilement l'étendre à 1Ø cases sur 1Ø 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 :
Les lignes 3Ø, 4Ø et 5Ø définissent trois variables A$, B$ et C$ qui contiennent les éléments de dessin. La ligne 6Ø imprime les lettres horizontales et la première ligne du dessin.
La boucle I (7Ø-14Ø) permet l'impression, le motif à imprimer étant affiché ou non selon la valeur de (I + J)/2 ; les lignes 1ØØ et 11Ø réalisent 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 3*J, J variant de 1 à 8, et en hauteur par 2*I.

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 INIT7,7
30 A$="ppp"
40 B$=CHR$(127)+CHR$(127)+CHR$(127)
50 C$="CCC"
60 DELIM4,4,0:PRINT"     A  B  C  D  E
F  G  H ";:DELIM7,7,0
70 FOR I=1 TO 8
80 FOR J=2 TO 9
90 IF (I+J)/2=INT((I+J)/2) THENGOTO 120
100 CURSORX3*J:CURSORY2*I-1:GR1,7,0:PRIN
TA$;
110 CURSORX3*J:CURSORY2*I:GR1,7,0:PRINTB
$;:CURSORX3*J:CURSORY2*I+1:PRINTC$;
120 NEXT J
130 TX4,4,0:CURSORX1:CURSORY2*I:PRINT I;
" "
140 NEXT I
```

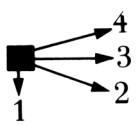
Le jeu de la fève

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

LE PROGRAMME :
La ligne 3Ø et la boucle I (4Ø-6Ø) dessinent le terrain de jeu.
La ligne 1ØØ réalise le test de fin de partie.
La ligne 13Ø permet d'exécuter l'un des sous-programmes de déplacement.
Les lignes 14Ø et 15Ø 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 RET.
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 INIT2,5
30 PRINT"    0  1  2  3  4  5  6  7  8
9"
40 FOR I=1 TO 9
50 PRINT:PRINTI;"  +  +  +  +  +  +  +
+  +  +"
60 NEXT I
70 X=5:Y=1
80 J=1
90 CURSORXX:CURSORY2*Y:TX4,4,0:PRINT"+"
100 IF X=32 AND Y=9 THEN GOTO 200
110 CURSORY22:PRINT"JOUEUR ";J;:INPUT"-D
IRECTION(1,2,3,4):";I$
120 I=VAL(I$):SOUND250,2,1
130 ON I GOSUB 1000, 2000, 3000, 4000
140 J=J+1
150 IF J=3 THENGOTO 80
160 GOTO 90
200 PRINT"LE JOUEUR ";J;" A GAGNE":PLAY"
ACE"
210 END
1000 Y=Y-(Y<9)
1010 RETURN
2000 GOSUB 1000
3000 X=X-3*(X<32)
3010 RETURN
4000 GOSUB 3000
4010 Y=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 le VG 5000 ; 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 1Ø8Ø 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 1Ø2Ø à 1Ø8Ø redessinent à chaque affichage le terrain de jeu en entier.

POUR JOUER :
On lancera le jeu par RUN. Le terrain de jeu est alors dessiné, puis on voit apparaître l'affichage JOUEUR 1 :...
On devra alors appuyer sur une touche quelconque pour que le coup du joueur numéro 1 soit pris en compte ; ne pas s'inquiéter du temps de réaction, qui pourra sembler un peu long.
Le déplacement indiqué par les dés sera exécuté, puis au bout de quelque temps, un second déplacement sera éventuellement effectué : c'est le déplacement "CHANCE". On obtiendra alors l'affichage JOUEUR 2 :...
Le joueur 2 appuie à son tour sur une touche quelconque et ainsi de suite...
Le joueur gagnant sera signalé par un affichage BRAVO en fin de partie.
On peut aussi s'arrêter à tout moment en tapant CTRL et △.

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

```
10 REM JEU DE L'OIE ALEATOIRE
20 X=0:Y=0
30 J=1
40 I=X
50 INIT2,2
60 GOSUB 1000
70 J=ABS(J-1)
80 I=INT(RND(1)*6+1)
90 IF X=35 OR Y=35 THENCURSORY20:PRINT"B
RAVO":PLAY"ACE":END
100 SOUND250,1,1:CURSORY12:PRINT"JOUEUR
";J+1;" - VOTRE JEU EST :";I
110 PRINT:INPUT"APPUYEZ SUR - RET - POUR
 AVANCER";R$
120 GOSUB 1000
130 I=INT(RND(1)*4)
140 F=INT(RND(1)*5+1)
150 ON F GOSUB 200,300,400,500,160
160 GOTO 60
200 I=I+3
210 RETURN
300 I=-I
310 RETURN
400 I=2*I
410 RETURN
500 IF J=0 THENX=INT(RND(1)*29+1):RETURN
510 Y=INT(RND(1)*29+1):RETURN
1000 IF J=0 THENX=X-I*(X+I<=35)
1010 Y=Y-J*I*(Y+I<=35)
1020 CURSORY3:PRINT"123456789ABCDEFGHIJK
LMNOPQRSTUVWXYZ"
1030 CURSORX1:CURSORY5:DELIM1,1,0:PRINT"
                                    <***"
1040 CURSORXX:CURSORY5:PRINT"1"
1050 CURSORX1:CURSORY6:DELIM4,4,0:PRINT"
                                    <***"
1060 CURSORXY:CURSORY6:PRINT"2"
1070 FOR I=1 TO 400:NEXT I
1080 RETURN
```

Niveau 2

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

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

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

Les structures de tableaux sont très couramment utilisées en programmation, se révélant même indispensables dans un grand nombre de cas. Si vous voulez, par exemple, entrer vos recettes et dépenses pour tous les jours du mois (on supposera pour simplifier des mois de trente jours) vous pourriez définir 30 variables appelées PREMIER JOUR, DEUXIEME JOUR, etc. Il est plus facile, et plus rationnel d'utiliser une boucle I, I variant de 1 à 3Ø, 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 le VG 5000 que si vous l'avez auparavant prévenu que vous alliez utiliser un tel tableau, en donnant une instruction DIM R (3Ø) pour les recettes, et DIM D (3Ø) 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 (3Ø, 2) (ou DIM RD (2, 3Ø)). Ainsi, avec DIM RD (3Ø, 2) on pourra convenir que RD (I, 1) représente les recettes, et RD (I, 2) les dépenses. Des tableaux de plus de 2 dimensions sont également possibles.

On peut aussi concevoir des tableaux dont les éléments sont des chaînes de caractères, et non plus des nombres, avec 255 caractères au maximum.

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

Sur le VG 5000 il n'est pas obligatoire de dimensionner un tableau qui contient moins de 10 éléments.

Ne pas oublier qu'il sera nécessaire de réserver de la place en mémoire, si nécessaire, par une instruction CLEAR.

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 3e élément numérique du tableau R. PRINT R$ (5) permet l'affichage du 5e 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", (A$, 3, 1) donne le caractère N ; de plus MID$ (A$, 3, 3) donne la sous-chaîne "NJO".

LE TRAITEMENT DE DONNEES MULTIPLES INSTRUCTIONS READ, DATA, RESTORE

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

Ainsi, on pourra écrire :

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

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

Dé truqué

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

LE JEU :
On veut essayer de fabriquer un dé truqué selon sa propre convenance. La méthode proposée ici consiste à définir au départ une probabilité de réalisation pour chaque type de tirage (soit la probabilité d'obtenir un 1, un 2, un 3, etc. que chacune de ces probabilités soit comprise entre 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 du VG 5000 supposé de qualité convenable (on pourra aisément utiliser tout autre générateur, si besoin est). La boucle I (7Ø-14Ø) s'occupe de l'entrée de chacune des probabilités, avec vérification à chaque entrée et sur le total (ligne 11Ø) ; chaque probabilité est conservée dans une variable indicée E (I) : ligne 9Ø du programme.
Le lancer de dé truqué est effectué à l'aide d'une autre boucle I (18Ø-2ØØ), la ligne permettant le truquage se trouvant en 19Ø. Un nouveau tirage pourra être effectué sur pression de RET (ligne 22Ø).

POUR JOUER :
Après le lancement de l'exécution par RUN, on obtient l'affichage de la question PROBABILITE DU DE 1, à laquelle on devra répondre par un nombre compris entre 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 RET pour effectuer les tirages suivants. On arrête le programme en tapant CTRL et △, ou STOP.

EXTENSIONS POSSIBLES :
Il sera évidemment bien 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 définissant le dé truqué dans une variable A$, etc. On pourra aussi envisager de présenter les résultats correspondant à une série de lancers de dés sous une forme spectaculaire, qui mette le truquage en évidence...

```
10 REM DE TRUQUE
40 INIT6,4
50 P=0
60 DIM E(6)
70 FOR I=1 TO 6
80 CURSORX17:CURSORY3:PRINT"
"
90 CURSORY3:PRINT"PROBABILITE DU DE ";I;
:INPUT E(I)
100 P=P+E(I)
110 IF E(I)<0 OR E(I)>=1 OR P>1.1 THEN 2
40
120 CURSORX6*I-4:CURSORY6:PRINT"DE ";I
130 CURSORX6*I-4:CURSORY8:PRINT E(I)
140 NEXT I
150 IF P<0.9 THENSOUND250,1,3:RUN
160 FOR I=1 TO 500:NEXT I
170 N=INT(RND(1)*6+1)
180 I=1
190 IF N=I AND RND(1)>E(I) THEN GOTO 170
200 I=I+1:IF I<6 THEN GOTO 190
210 CURSORX1:CURSORY15:PRINT"TIRAGE :";N
;"     ":PLAY"ACE"
220 PRINT:INPUT"FAIRE - RET - POUR UN AU
TRE TIRAGE";R$
230 GOTO 170
240 CURSORY4:TX1,0,1:PRINT"IMPOSSIBLE :0
<= P <=1":PLAY"AAA"
250 P=P-E(I)
260 FOR J=1 TO 1500:NEXT J
270 CURSORY4:PRINT"
      "
280 GOTO 90
```

Visions

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

LE JEU :
Le VG 5000 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 (4Ø-12Ø) se charge de réaliser les dessins.
La ligne 5Ø permet de transformer en nombre chaque lettre du mot entré au début (ligne 3Ø). Ce nombre sera ensuite traduit en base 2 par une boucle J (lignes 6Ø à 11Ø) ; selon le résultat de cette transformation, un point sera imprimé, ou non (ligne 9Ø).

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 le VG 5000 dès que l'on aura frappé RET.
Pour recommencer, il suffira de frapper une touche quelconque, et STOP pour finir.

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 VG 5000 deviendra de plus en plus savant. On peut aussi réaliser un jeu à deux joueurs dans lequel VG 5000 vous demande de trouver le mot qui correspond à un certain dessin, etc.

```
10 REM VISIONS
20 INIT4,6
30 INPUT"QUEL DESSIN :";R$
40 FOR I=1 TO LEN(R$)
50 N=9*ASC(MID$(R$,I,1))-340
60 J=1
70 S=N/2-INT(N/2)
80 N=INT(N/2):SOUNDN+1,1,2
90 IF S<>0 THENCURSORXJ+15:CURSORYI+5:TX
1,0,0:PRINTCHR$(127)
100 J=J+1
110 IF J<10 THENGOTO 70
120 NEXT I
130 PRINT:PRINT
140 INPUT"APPUYEZ SUR - RET -";R$
150 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 40.
La ligne 130 affiche le numéro de joueur, le nombre d'essais déjà effectués, et le numéro de la partie en cours.
Un sous-programme placé en ligne 4000, utilisé 3 fois dans la boucle J (200-250), réalise le lancer de dés.
Les retirages sont effectués par les lignes 1000, 2000 et 3000.
La variable R$ gère les 2e et 3e tirages. L'impression du résultat est réalisée dans tous les cas par les lignes 140-150.
Les tests pour le nombre d'essais et pour le changement de joueur sont effectués aux lignes 260 et 280.

POUR JOUER :
On lance l'exécution du programme par RUN ce qui conduit à l'affichage NOMBRE DE JOUEURS. On entre le nombre de joueurs, et l'on tape RET. 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 RET. 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 RET : par exemple, pour retirer A et C on tapera AC suivi de RET.
Les joueurs suivants procéderont de la même façon.

EXTENSIONS POSSIBLES :
La plus évidente consiste à assurer la gestion des résultats des joueurs, en se référant aux règles de gain usuelles du jeu du 421. On pourra envisager l'affichage après chaque tour d'un tableau récapitulant les résultats des joueurs.
On pourra aussi améliorer l'affichage des dés (se référer aux divers programmes d'affichage des dés se trouvant dans le même ouvrage).

```
10 REM 421
20 INIT6,4
40 INPUT"NOMBRE DE JOUEURS :";N
50 P=1
60 I=1
70 K=0
80 INIT 5,3
90 R$="ABC"
100 GOTO 200
110 K=K+1
120 CURSORX15:CURSORY1:TX4,7,1:PRINT"- 4
 2 1 - ":CURSORX15:PRINT"- 4 2 1 - "
130 CURSORY4:TX7,0,0:PRINT"JOUEUR NO. ";
I;" - ESSAI ";K;" - PARTIE ";P
140 CURSORX8:CURSORY7:TX1,2,0:PRINTCHR$(
A+48)+" ";:CURSORX18:PRINTCHR$(B+48)+" "
;:CURSORX28:PRINTCHR$(C+48)+" "
150 CURSORX8:CURSORY9:TX7,6,0:PRINT"A ";
:CURSORX18:PRINT"B ";:CURSORX28:PRINT"C
"
160 TX7,0,0:R$="":CURSORY14:PRINT"
                    "
170 CURSORY14:INPUT"ABC :";R$
180 IF R$="" THENGOTO 270
190 R$=R$+"DDD"
200 FOR J=1 TO 3
210 GOSUB 4000
220 X$=MID$(R$,J,1)
230 X=ASC(X$)-64
240 ON X GOSUB 1000,2000,3000,4000
250 NEXT J
260 IF K<3 THENGOTO 110
270 I=I+1
280 IF I<=N THENGOTO 70
290 P=P+1
300 GOTO 60
1000 A=D
1010 RETURN
2000 B=D
2010 RETURN
3000 C=D
3010 RETURN
4000 D=INT(RND(1)*6+1):SOUND20*D+100,9,4
0
4010 RETURN
```

L'électricien fou

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

LE JEU :
Un immeuble comportant 1Ø fenêtres, numérotées de Ø à 9, apparaît sur l'écran. Le joueur devra à chaque essai manœuvrer 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 1Ø fenêtres, avec le moins possible d'essais.

LE PROGRAMME :
Une table A, en ligne 5Ø, contiendra l'état (allumé ou éteint) des 1Ø fenêtres.
La boucle I (7Ø-9Ø) et l'autre boucle I (11Ø-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 à manœuvrer.
Les lignes 19Ø à 29Ø définissent le résultat de la manœuvre.

POUR JOUER :
RUN provoque l'affichage de l'immeuble, avec toutes les fenêtres allumées.
A la question INTERRUPTEUR NUMERO, on répond en indiquant le numéro de l'interrupteur à manœuvrer, suivi de RET.
Le résultat de la manœuvre est alors affiché sur l'écran (des fenêtres s'allument ou s'éteignent).
On continue ainsi jusqu'à ce que toutes les fenêtres soient éteintes, et l'on obtient alors l'affichage du nombre d'essais réalisés.

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

```
10 REM L'ELECTRICIEN FOU
40 M=0
50 DIM A(10)
60 INIT1,1
70 FOR I=1 TO 7
80 CURSORYI:DELIM4,4,0:PRINT
90 NEXT I
100 T=0
110 FOR I=1 TO 10
120 T=T+A(I)
130 CURSORX2*I+8:CURSORY3:PRINT" "
140 IF A(I)=0 THENCURSORX2*I+8:CURSORY3:
TX3,0,0:PRINTCHR$(127):TX7,0,0
150 CURSORX2*I+8:CURSORY6:TX5,1,0:PRINTC
HR$(I+47):CURSORX2*I+8:PRINTCHR$(I+47)
160 NEXT I
170 TX7,0,0:IF T=10 THENGOTO 310
180 CURSORY14:PLAY"A":INPUT"INTERRUPTEUR
 NUMERO :";N
190 N=N+1
200 M=M+1
210 IF A(N)=1 THENGOTO 290
220 A(N)=1
230 R=TAN(RND(1)+N/RND(1)-N)-SIN(RND(1)/
N)+336*SIN(8*N)
240 N=INT(10*(R-INT(R)))
250 IF N=0 THENN=INT(RND(1)*10+1)
260 IF A(N)=1 THENGOTO 290
270 A(N)=1
280 GOTO 100
290 A(N)=0
300 GOTO 230
310 PRINT:PRINT"TROUVE EN ";M;" ESSAIS":
PLAY"ACE"
```

La course au 20

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

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

LE PROGRAMME :
La boucle I (30-50) dessine 4 rangées de 20 étoiles.
La variable F, dimensionnée en ligne 60, servira à enregistrer le nombre d'étoiles prises dans chaque rangée.
La variable T contiendra le nombre total d'étoiles enlevées (ligne 70).
La ligne 120 va contrôler qu'il reste suffisamment d'étoiles sur la ligne indiquée.
Les boucles I (150-190) et J (160-180) suppriment les étoiles ; la ligne 220 teste la fin de partie.

POUR JOUER :
Lancer le programme par RUN ; le terrain de jeu s'affiche, puis les inscriptions JOUEUR 1 et LIGNES, puis CASES. On répond en entrant un numéro de ligne (entre 1 et 4) suivi de RET, et le nombre d'étoiles à enlever suivi de RET. Les étoiles sont alors supprimées, et c'est au tour du deuxième joueur.
La partie se termine, lorsque la dernière étoile a été enlevée, par l'affichage du joueur gagnant.

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

```
10 REM LA COURSE AU 20
20 INIT 6,4
30 FOR I=1 TO 4
40 CURSORY2*I:PRINTI;" ";:DELIM1,1,0:PRI
NT" ********************  ";:DELIM6,6,0:
PRINT
50 NEXT I
60 DIM F(4)
70 B=1:T=0
80 CURSORY12:PRINT"JOUEUR ";B
90 CURSORY14:PRINT"                    ":CURS
ORY14:INPUT"LIGNE :";L
100 IF L>4 THENGOTO 80
110 CURSORY16:PRINT"              ":CURSORY
16:INPUT"CASES :";C
120 IF C+F(L)>20 THENGOTO 80
130 F(L)=F(L)+C
140 T=T+C
150 FOR I=1 TO 4
160 FOR J=0 TO C
170 CURSORX5+F(I)+J*(I=L):CURSORY2*I:PRI
NT" ";
180 NEXT J
190 NEXT I
200 B=B+1:PRINT:SOUND240,2,50
210 IF B=3 THENB=1
220 IF T<80 THENGOTO 80
230 PRINT"LE JOUEUR ";B;" GAGNE":PLAY"AC
EG"
```

Musique

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

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

LE PROGRAMME :
La boucle I (5Ø-7Ø) dessine la portée. Une deuxième boucle I (8Ø-12Ø) affiche les numéros des lignes.
La ligne 13Ø définit le nom des notes, et la ligne 14Ø réalise le tirage aléatoire d'une note.
Le nom de la note est calculé à la ligne 15Ø et il est affiché par la ligne 16Ø, suivi de la mention GRAVE ou AIGUE, selon le cas.
La réponse sera entrée en ligne 19Ø.
La ligne 2ØØ s'occupe du décodage de la réponse.
La ligne 21Ø affiche la note.
Les lignes 23Ø et 24Ø regardent si l'on a gagné ou perdu et affichent le message correspondant.
La ligne 25Ø 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 RET.
La note est alors dessinée et le message BRAVO ou PERDU apparaît, dans ce cas, à l'écran.
On doit taper la touche RET pour continuer, et s'arrêter en tapant STOP.

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
40 INIT6,6
50 FOR I=1 TO 5
60 CURSORX4:CURSORY2*I+4:GR1,6,0:PRINT"L
LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL";
70 NEXT I
80 FOR I=1 TO 14
90 CURSORX1:CURSORYI+2:TX4
100 IF I<=9 THEN PRINT I
110 IF I>9 THEN PRINT" ";CHR$(I+55)
120 NEXT I
130 N$="SI LA SOLFA MI RE DO "
140 X=INT(RND(1)*14+1)
150 T=X-1-7*INT((X-1)/7)
160 CURSORX1:CURSORY18:TX0:PRINT"NOTE ";
MID$(N$,3*T+1,3);
170 IF X<=7 THEN PRINT" AIGUE"
180 IF X>7 THEN PRINT" GRAVE"
190 PRINT:INPUT"NUMERO DE LA LIGNE :";L$
200 L=-(ASC(L$)-48)*(L$<":")-(ASC(L$)-55
)*(L$>="A")
210 CURSORX14:CURSORYX:PRINT"¦":CURSORX1
4:CURSORYX+1:PRINT"¦":CURSORX13:CURSORYX
+2:PRINT"O"
220 CURSORX25:CURSORY20:TX5,4,0
230 IF L=X THEN PRINT"BRAVO":PLAY"ACE"
240 IF L<>X THEN PRINT"PERDU":PLAY"AAA"
250 TX0:PRINT:INPUT"TAPEZ - RET - POUR C
ONTINUER";R$
260 GOTO 40
```

Poker

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu contre le VG 5000.
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, le VG 5000.
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 du VG 5000 de jouer.

LE PROGRAMME :
Les lignes 3Ø et 4Ø définissent les cartes. Une variable N est dimensionnée à 15, en ligne 2Ø, pour recevoir 1Ø cartes joueur et 5 cartes VG 5000.
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 23Ø-34Ø.
Une autre boucle I (16Ø-18Ø) réalise le deuxième tirage du joueur, la ligne 21Ø permettant le deuxième affichage, et la boucle I (23Ø-3ØØ) 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 RET s'il veut arrêter la partie. Pour jouer, il répondra simplement RET s'il s'estime servi, ou il entrera les numéros des cartes qu'il refuse suivis de RET.
Par exemple, s'il tape 235 et RET, son nouveau jeu conservera les cartes 1 et 4. Son jeu sera alors affiché ainsi que celui du VG 5000, ce dernier n'ayant pas, dans ce programme, la possibilité de reprendre des cartes.
Il suffira ensuite de faire RET 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 du VG 5000, l'introduction des règles sur les paris et même, pourquoi pas, des possibilités de bluff.

```
10 REM POKER
20 DIM N(15)
30 V$="7 8 9 10VADAROAS"
40 C$="TRCACOPI"
50 INIT6,4
60 FOR I=1 TO 15
70 N(I)=INT(RND(1)*32)
80 J=1:IF I=1 THEN GOTO 110
90 IF N(I)=N(J) THENGOTO 70
100 J=J+1:IF J<I THEN GOTO 90
110 NEXT I
120 K=0:P=0
130 CURSORY12:PRINT"CARTES REFUSEES":PLA
Y"AEC"
140 GOSUB 230
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 INIT6,4
200 P=1:PLAY"AAA"
210 GOSUB 230
220 K=K+10
230 FOR I=1 TO 5
240 J=2*INT(N(I+K)/8)+1
250 F=2*(N(I+K)-8*INT(N(I+K)/8))+1
260 R$=MID$(V$,F,2)
270 CURSORX7*I-4:CURSORYK:PRINTI
280 CURSORX7*I-5:CURSORYK+2:PRINTR$;":";
MID$(C$,J,2)
290 CURSORX7*I-5:CURSORYK+3:GR5,6,0:PRIN
T"LLLLL";:TX0
300 NEXT I
310 IF P=1 THENP=0:RETURN
320 IF K=10 THENPRINT:PRINT:INPUT"FAIRE
- RET - POUR CONTINUER";R$:GOTO 50
330 CURSORX18:CURSORY12:INPUT R$
340 RETURN
```

Le jeu de saute-mouton

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

LE JEU :
Bien que l'idée en soit très simple, il ne sera pas forcément évident de découvrir une stratégie convenable. On dispose de 3 moutons noirs et de 3 moutons gris, représentés par des symboles graphiques, 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 17Ø et 18Ø s'occupent du texte et des entrées.
La ligne 19Ø contrôle la validité du coup. Le changement de joueur est réalisé en ligne 22Ø.

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 RET, puis le numéro de la case d'arrivée, suivi de RET.
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 INIT2,2
30 DIM A(8)
40 A(1)=0:A(8)=0
50 FOR I=1 TO 3
60 A(I+1)=1
70 A(I+4)=-1
80 NEXT I
90 H=1
100 FOR I=1 TO 8
110 TX0,0,0:CURSORX4*I:CURSORY8:PRINT I
120 CURSORX4*I+1:CURSORY6:PRINT" "
130 IF A(I)=1 THENCURSORX4*I+1:CURSORY6:
TX1,0,0:PRINTCHR$(127)
140 IF A(I)=-1 THENCURSORX4*I+1:CURSORY6
:TX4,0,0:PRINT CHR$(127)
150 NEXT I
160 J=1-(H=-1)
170 TXJ-2*(J=2):CURSORY13:PRINT"JOUEUR "
;J:TX0,0,0:SOUND100+50*J,5,3
180 PRINT:INPUT"DEBUT :";D:INPUT"FIN   :
";F
190 IF A(D)<>H OR A(F) <>0 THENGOTO 170
200 A(D)=0
210 A(F)=H
220 H=-H
230 GOTO 100
```

Lettres

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

LE JEU :
La règle est fortement inspirée d'un jeu de lettres très connu en France, pratiqué notamment dans une émission télévisée (les chiffres et les lettres). Le VG 5000 choisit au hasard un certain nombre de symboles (voyelles et consonnes) dans l'alphabet latin, et chacun des joueurs devra indiquer la longueur du mot qu'il a réussi à obtenir (l'appareil ne vérifiera pas si le mot est correct !).
Le score total de chaque joueur est affiché ensuite.

LE PROGRAMME :
La boucle I (90-120) choisit dans l'alphabet entre 7 et 12 lettres, en fonction de la variable aléatoire L, tirée en ligne 60. Pour éviter d'avoir trop de consonnes, on utilise en ligne 110 une petite astuce, qui consiste à refaire au hasard un tirage dans la table A$, contenant des voyelles (on peut en avoir plus ou moins en utilisant d'autres RND).
La boucle 130-180 demande à chacun des joueurs la longueur de son mot. Les lignes 190 et 200 comparent les résultats et totalisent le score de chaque joueur, affiché par la ligne 210.

POUR JOUER :
On lance l'exécution du jeu avec RUN.
Les lettres sont alors affichées, puis la question JOUEUR 1 LONGUEUR à laquelle le premier joueur répond en donnant la longueur du mot qu'il a obtenu (l'autre joueur devra toutefois vérifier si le mot est correct) ; puis le joueur 2 fera de même et le score de chaque joueur sera alors affiché.
Pour continuer la partie, il suffira d'appuyer sur la touche RET.
On devra faire STOP 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 le VG 5000, auquel on aurait auparavant entré une liste de mots assez longue, dont il afficherait les lettres dans le désordre, avec éventuellement quelques lettres de complément choisies au hasard ?

```
10 REM LETTRES
30 S=0
40 J=0
50 INIT6,4
60 L=INT(RND(1)*6+7)
70 A$="AEIOUEA"
80 CURSORY6:PRINT"LETTRES :"
90 FOR I=1 TO L
100 CURSORX2*I+8:CURSORY6:DISPLAY200:PRI
NTCHR$(INT(RND(1)*26+65))
110 IF RND(1)<0.4 THENCURSORX2*I+8:CURSO
RY6:PRINTMID$(A$,INT(RND(1)*7+1),1)
120 NEXT I
130 I=1
140 CURSORY10:PRINT"JOUEUR :";I;:INPUT"
- LONGUEUR :";L
150 CURSORX25:CURSORY10:PRINT"      "
160 IF I=1 THENP=L
170 I=I+1
180 IF I<3 THENGOTO 140
190 S=S+(P AND P>L)
200 J=J+(L AND L>P)
210 CURSORY13:PRINT"SCORES - JOUEUR 1 :"
;S;" - JOUEUR 2 :";J:PLAY"ACE"
220 PRINT:INPUT"TAPEZ - RET - POUR CONTI
NUER";R$
230 GOTO 50
```

Machinamot

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

LE JEU :
Ce très court programme permet au VG 5000 de faire la démonstration de ses capacités à manipuler des chaînes de caractères : on part simplement d'un préfixe et d'un suffixe d'origine grecque, que l'on va assembler au hasard, on tombera bien sûr, de temps à autre, sur des mots absurdes ou même légèrement équivoques.
Un nouveau mot est confectionné à chaque essai, mais il est possible d'obtenir plusieurs fois le même mot.

LE PROGRAMME :
Deux variables A$ et B$ sont dimensionnées à 15 en lignes 5Ø et 6Ø : 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 (7Ø-9Ø) lit les lignes DATA (16Ø-19Ø) concernant les préfixes suivis des suffixes.
La ligne 1ØØ effectue le choix d'un préfixe au hasard.
La ligne 11Ø effectue le choix d'un suffixe au hasard.
Le mot obtenu en ligne 12Ø est affiché par la ligne 13Ø, puis on recommence (ligne 15Ø).

POUR JOUER :
Lancer l'exécution du programme par RUN. Un premier mot sera immédiatement affiché, puis un nouveau mot sera affiché à la suite du précédent.
Taper STOP 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 au VG 5000...

```
10 REM MACHINAMOT
20 CLEAR 500
40 INIT6,2
50 DIM A$(15)
60 DIM B$(15)
70 FOR I=1 TO 15
80 READ A$(I),B$(I)
90 NEXT I
100 I=INT(RND(1)*15+1)
110 J=INT(RND(1)*15+1)
120 C$=A$(I)+B$(J)
130 PRINT C$:SOUND130,1,1
140 FOR S=1 TO 1500:NEXT S
150 GOTO 100
160 DATA"ANDRO","CRATE","COSMO","PHILE",
"MICRO","PHOBE","HELIO","TROPE"
170 DATA"SEXO","GYNE","STEREO","MANE","P
ALEO","NAUTE","MACRO","PHAGE"
180 DATA"HYDRO","FUGE","STENO","PATHE","
PATHO","CARDE","OLEO","DERME"
190 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 œuvre sur ce type d'appareil.
Un petit générateur aléatoire situé en fin de programme est utilisé pour modifier de façon arbitraire chaque lettre du message ; on devra bien 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 (8Ø-15Ø) est utilisée soit pour effectuer le codage, soit pour effectuer le décodage.
La ligne 1ØØ appelle le générateur aléatoire.
La ligne 11Ø réalise la transposition des lettres.
Les lignes 12Ø et 13Ø vérifient que l'on reste bien dans les limites autorisées.
Les résultats seront imprimés par la ligne 16Ø.
Les lignes 1ØØØ et 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 RET.
Choisir l'option C pour CODER, et l'option D pour DECODER, et taper RET.
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 INIT6,4
30 INPUT"DONNEZ VOTRE CODE :";N
40 A=1/N
50 PRINT:INPUT"VOTRE MESSAGE :";A$
60 PRINT:INPUT"VOULEZ-VOUS (C)ODER OU (D
)ECODER :";R$
70 B$=""
80 FOR I=1 TO LEN(A$)
90 T=ASC(MID$(A$,I,1))
100 GOSUB 1000
110 T=T-A*((R$="C")-(R$="D"))
120 IF T>90 THENT=T-26
130 IF T<65 THENT=T+26
140 B$=B$+CHR$(T)
150 NEXT I
160 PRINT:PRINT"RESULTAT :";B$:PLAY"AAA"
: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 à son wagon, qui apparaissent tous deux en début de partie, en deux endroits plus ou moins éloignés de l'écran. On devra pour cela déplacer la locomotive en indiquant la suite des mouvements à effectuer (sous forme codée : H pour une case vers le haut ; B pour une case vers le bas ; D pour une case vers la droite et G pour une case vers la gauche).
On tapera, par exemple : BBGDHHH.

LE PROGRAMME :
Les lignes 40 et 50 dessinent la locomotive, la ligne 60 le wagon, et les roues sont en ligne 70.
La position initiale de la locomotive est définie par les lignes 90 et 100, celle du wagon par les lignes 110 et 120.
La ligne 130 évite qu'ils soient initialement trop proches l'un de l'autre.
Les lignes 150 et 160 réalisent l'affichage du wagon et les lignes 170, 180 et 190 celui de la locomotive.
Le test de fin est en ligne 200.
Les mouvements à effectuer sont entrés dans une variable alphanumérique R$, en ligne 210, et sont décodés dans une boucle I (240-270) pour être exécutés en ligne 280.

POUR JOUER :
RUN permet d'afficher les positions de départ de la locomotive et du wagon.
On tape alors la série des déplacements à effectuer, sous la forme DDDHHHBHGG, suivie de RET.
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
40 D$="l!z"
50 A$="z"+CHR$(127)+CHR$(127)
60 B$="upz"
70 C$="O O"
80 N=0
90 LY=INT(RND(1)*33)
100 LX=INT(RND(1)*20+1)
110 WY=INT(RND(1)*33)
120 WX=INT(RND(1)*20+1)
130 IF ABS(LX-WX)<4 THENGOTO 90
140 INIT6,4
150 CURSORXWY:CURSORYWX:GR1,6,0:PRINT B$
;
160 TX1,0,0:CURSORXWY:CURSORYWX+1:PRINTC
$
170 CURSORXLY:CURSORYLX-1:GR3,6,0:PRINTD
$;
180 CURSORXLY:CURSORYLX:PRINTA$;
190 CURSORXLY:CURSORYLX+1:TX1,0,0:PRINTC
$
200 IF LX=WX AND WY=LY-3 THENGOTO 300
210 TX0,0,0:CURSORY22:INPUT"CHEMIN A SUI
VRE (DGGHBB...):";R$
220 X=0:Y=0
230 N=N+1
240 FOR I=1 TO LEN(R$)
250 X=X-(MID$(R$,I,1)="B" AND LX<22)+(MI
D$(R$,I,1)="H" AND LX>0)
260 Y=Y-(MID$(R$,I,1)="D" AND LY<37)+(MI
D$(R$,I,1)="G" AND LY>0)
270 SOUND80,2,9:NEXT I
280 LX=LX+X:LY=LY+Y
290 GOTO 140
300 CURSORY22:PRINT"BRAVO ; TROUVE EN ";
N;" COUPS"
310 PLAY"ACE"
```

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 (7Ø-14Ø)avec toutes les vérifications de validité nécessaires. Le nombre de tirages demandés est introduit en ligne 16Ø.
La boucle J (17Ø-25Ø) comptabilise les tirages.
La boucle I (19Ø-21Ø) assure la conformité de chaque tirage avec les probabilités introduites en début de jeu.
Les résultats sont comptabilisés en ligne 22Ø, et affichés grâce aux lignes 23Ø et 24Ø.

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 proche de 1.
Ensuite, à l'affichage de la question NOMBRE DE TIRAGES, on introduira un entier suffisamment grand, suivi de RET. L'affichage des résultats mettra d'autant plus de temps à apparaître que l'entier introduit sera grand.
Les résultats sont affichés sous les numéros des tirages : ainsi 32.3 apparaissant sous le 1 signifie que le tirage de 1, auquel on avait affecté une probabilité de Ø.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
40 DIM E(6)
50 INIT7,7
60 P=0
70 FOR I=1 TO 6
80 CURSORY3:PRINT"
                  "
90 CURSORY3:PRINT"PROBABILITE DU DE ";I;
:INPUT E(I)
100 P=P+E(I)
110 IF E(I)<0 OR E(I)>=1 OR P>1.1 THENGO
TO 300
120 CURSORX6*I-5:CURSORY6:PRINT"DE ";I
130 CURSORX6*I-5:CURSORY8:PRINT E(I)
140 NEXT I
150 IF P<0.9 THENRUN
160 PRINT:INPUT"NOMBRE DE TIRAGES :";P
170 FOR J=1 TO P
180 N=INT(RND(1)*6+1)
190 I=1
200 IF (N=I) AND (RND(1)>(E(I)-INT(E(I))
)) THENGOTO 180
210 I=I+1:IF I<7 THEN GOTO 200
220 E(N)=E(N)+1
230 CURSORX6*N-3:CURSORY15:TX5,4,0:PRINT
 N
240 CURSORX6*N-4:CURSORY17:TX4,4,0:PRINT
 INT(E(N))
250 NEXT J
260 END
300 CURSORY4:PRINT"IMPOSSIBLE :0<= P <=1
"
310 P=P-E(I)
320 FOR J=1 TO 2000:NEXT J
330 CURSORY4:PRINT"
  "
340 GOTO 90
```

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 le VG 5000 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 au VG 5000 dans quelle colonne (1re, 2e ou 3e à partir de la gauche) se trouve la carte choisie ; les cartes sont alors redistribuées et le joueur indique de nouveau la colonne correspondant à sa carte ; ceci est fait encore une fois et, au 4e affichage, le VG 5000 est en mesure de deviner la carte choisie.

LE PROGRAMME :
La boucle I (60-120) permet de choisir, au hasard, 21 cartes sur 52, et la boucle emboîtée J (80-100) permet de s'assurer que l'on ne réalise pas deux fois le même tirage. La ligne 140 contient les couleurs des cartes ainsi que les noms des cartes figures (VA, DA, RO, AS), dans une variable alphanumérique B$.
La boucle I (150-240) réalise l'affichage du jeu.
La ligne 160 décode la couleur de la carte.
La ligne 170 choisit la position d'affichage.
La ligne 180 décode la hauteur.
L'affichage proprement dit est effectué de la ligne 190 à la ligne 230.
La ligne 260 permet au joueur d'entrer le numéro de la colonne où se trouve sa carte. La boucle J (300-320) réalise le brouillage, et les boucles emboîtées J (350-390) et I (360-380) fabriquent la nouvelle variable A$ à utiliser pour le prochain affichage.

POUR JOUER :
On lance le programme par RUN.
On obtient un premier affichage de 21 cartes réparties sur 3 colonnes.
On indique alors le numéro (1, 2 ou 3) de la colonne où se trouve la carte choisie, suivi de RET.
On obtient un 2e affichage et l'on fait de même ; puis un 3e affichage et l'on fait de même.
Au 4e affichage, la carte choisie est automatiquement affichée.
On sortira du programme en tapant STOP à la place du numéro de la colonne.

EXTENSIONS POSSIBLES :
On peut envisager une amélioration de l'affichage en écrivant les noms des cartes en entier, ou même, en dessinant de véritables cartes, en couleur.
On pourra aussi essayer de programmer d'autres tours, utilisant le même principe de base, mais avec davantage de cartes, etc.

```
10 REM TOUR DE CARTES
20 CLEAR500
40 INIT3,6
50 A$=""
60 FOR I=1 TO 21
70 B$=CHR$(INT(RND(1)*52+38))
80 FOR J=1 TO LEN(A$)
90 IF B$=MID$(A$,J,1) THENI=I-1:GOTO 120
100 NEXT J
110 A$=A$+B$
120 NEXT I
130 H=3:P=1
140 B$="TRCACOPIVADAROAS"
150 FOR I=1 TO LEN(A$)*P-11*(P=0)
160 S=INT((ASC(MID$(A$,I,1))-38)/13)
170 J=7*INT((I-1)/7)
180 T=ASC(MID$(A$,I,1))-37-13*S
190 IF P=0 THENCURSORY0:PRINT"VOTRE CART
E EST :                    "
200 CURSORXJ*P+INT(J/2)-16*(P=0)+1:CURSO
RY(I-J)*P
210 IF T<=9 THENPRINT STR$(T+1);
220 IF T>9 THENPRINTMID$(B$,ABS(2*T-11),
2);
230 CURSORX(J+3)*P+2+INT(J/2)-21*(P=0):C
URSORY(I-J)*P:PRINTMID$(B$,2*S+1,2)
240 NEXT I
250 IF P=0 THENPLAY"ABCDE":END
260 CURSORY16:PLAY"ACA":INPUT"QUELLE COL
ONNE :";N
270 INIT3,4
280 H=H-1
290 R$="20120":B$=""
300 FOR J=0 TO 2
310 B$=B$+MID$(A$,7*VAL(MID$(R$,N+J,1))+
1,7)
320 NEXT J
330 IF H=0 THENA$=B$:P=0:GOTO 140
340 A$=""
350 FOR J=0 TO 2
360 FOR I=1 TO 7
370 A$=A$+MID$(B$,3*I-2+J,1)
380 NEXT I
390 NEXT J
400 GOTO 140
```

Le jeu de Marienbad

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

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

LE PROGRAMME :
La ligne 6Ø dimensionne une variable A (I) qui contiendra le nombre d'allumettes de chaque tas.
La boucle I (7Ø-1ØØ) permet d'entrer ce nombre.
La boucle I (13Ø-16Ø) imprime les tas.
La ligne 17Ø assure l'affichage du joueur.
La ligne 19Ø, ainsi que la ligne 2ØØ, contrôlent si le coup est autorisé.
La ligne 27Ø contrôle la fin de la partie.

POUR JOUER :
On lance le programme avec RUN, puis on entre le nombre de tas (entre 1 et 8), suivi de RET.
On entrera ensuite le nombre d'allumettes dans chaque tas (faire à chaque entrée N suivi de RET). 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 RET. 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 au VG 5000 de servir de partenaire, etc.

```
10 REM LE JEU DE MARIENBAD
20 INIT6,4
30 INPUT"NOMBRE DE TAS ( <= 7 ) :";N
40 IF N>=8 THENPLAY"AAA":RUN
50 S=0
60 DIM A(N)
70 FOR I=1 TO N
80 PRINT:PRINT"TAS ";I;:INPUT" - NOMBRE
D'ALLUMETTES :";A(I)
90 S=S+A(I)
100 NEXT I
110 H=1
120 INIT3,5
130 FOR I=1 TO N
140 CURSORX5*I+2:CURSORY4:TX1,0,0:PRINT
A(I)
150 CURSORY6:TX6,4,0:PRINT"TAS":CURSORX5
*I+2:CURSORY6:PRINT I
160 NEXT I
170 CURSORY10:TX0,6,0:PRINT"J O U E U R"
;STR$(H)+"  "
180 TX0,0,0:PRINT:INPUT"DANS QUEL TAS :"
;T
190 IF T>N THEN 120
200 IF A(T)=0 THENGOTO 120
210 CURSORY14:INPUT"QUANTITE A ENLEVER :
";O
220 IF O>A(T) OR O=0 THENGOTO 200
230 A(T)=A(T)-O
240 S=S-O
250 H=H+1
260 IF H=3 THENH=1
270 IF S=0 THENTX1,4,1:PRINT:PRINT"LE JO
UEUR ";H;" A GAGNE":PLAY"ACE":END
280 GOTO 120
```

L'indésirable

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : hasard et réflexion (1 joueur).
Difficulté du jeu : moyenne.

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

LE PROGRAMME :
Une table B est dimensionnée à 16, en ligne 5Ø, pour recevoir les numéros des cases déjà visitées.
La boucle I (7Ø-11Ø) dessine le terrain de jeu.
La position initiale du locataire est tirée au hasard en ligne 12Ø.
Le numéro de la case à visiter est donné en ligne 13Ø.
Les lignes 14Ø à 22Ø contrôlent les diverses réponses possibles.
La boucle I (18Ø-2ØØ) dessine les cases visitées.
La ligne 23Ø dessine le locataire, effacé en ligne 26Ø.
Les lignes 27Ø et 28Ø assurent son déplacement.

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

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

```
10 REM L'INDESIRABLE
20 CLEAR 500
30 A$=CHR$(127)
40 B$=A$+A$+A$
50 DIM B(16)
60 INIT6,4
70 FOR I=1 TO 16
80 S=4+4*(I-1-4*INT((I-1)/4))
90 CURSORXS:CURSORY2+4*INT((I-1)/4):TX1,
0,0:PRINT B$
100 CURSORXS:CURSORY3+4*INT((I-1)/4):TX2
,0,0:PRINTA$;:TX0,4,0:PRINTCHR$(64+I);:T
X2,0,0:PRINTA$:TX0,0,0
110 NEXT I
120 X=INT(RND(1)*16+1):S=1:K=0
130 SOUND240,1,5:CURSORY20:INPUT"CASE VI
SITEE :";C$:C=ASC(C$)-64
140 IF B(C)=1 THENCURSORY22:PRINT"DEJA V
ISITEE":PLAY"AAA":GOTO 130
150 IF C=X THENCURSORY22:PRINT"VOUS L'AV
EZ EU EN ";S;" COUPS. BRAVO":PLAY"ACE":K
=1:GOTO 230
160 IF X-5=CORX-4=CORX-3=CORX-1=CORX+1=C
ORX+3=CORX+4=CORX+5=C THENGOTO 230
170 CURSORY22:PRINT"IL N'EST PAS LA":PLA
Y"ACA":FORI=1 TO 500:NEXTI
180 S=S+1:B(C)=1:FOR I=1 TO 16
190 IF B(I)=1 THENCURSORX4+4*(I-1-4*INT(
(I-1)/4)):CURSORY2+4*INT((I-1)/4):TX5,0,
0:PRINT B$
200 NEXT I
210 IF S=9 THENCURSORY22:PRINT"VOUS AVEZ
 PERDU. IL ETAIT EN ";CHR$(X+64):K=1:GOT
O 230
220 CURSORY22:PRINT"
                    ":GOTO 130
230 CURSORX5+4*(X-1-4*INT((X-1)/4)):CURS
ORY1+4*INT((X-1)/4):TX0,4,0:PRINT"%"
240 FOR I=1 TO 900:NEXT I
250 IF K=1 THENPLAY"ECA":END
260 CURSORX5+4*(X-1-4*INT((X-1)/4)):CURS
ORY1+4*INT((X-1)/4):TX0,0,0:PRINT" "
270 Z=INT(10*RND(1)+1)-5:IF ABS(Z)=2 THE
NGOTO 270
280 X=X+Z:IF X>16 OR X<1 THENX=X-Z:GOTO
270
290 GOTO 180
```

Compose

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

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

LE PROGRAMME :
La boucle I (5Ø-7Ø) dessine la portée. La variable N$, en ligne 8Ø, contient les noms des notes.
La deuxième boucle I (9Ø-17Ø) permet de faire le choix d'une note de façon aléatoire, à la ligne 1ØØ ; son nom est calculé en ligne 11Ø et la note est dessinée par la ligne 12Ø.
La ligne 18Ø permet de recommencer en tapant RET.

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 RET provoquera l'affichage d'une nouvelle série de notes, et une pression sur la touche STOP provoque la fin de la partie.

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

```
10 REM COMPOSE
40 INIT6,6
50 FOR I=1 TO 5
60 CURSORX1:CURSORY2*I+3:GR1,6,0:PRINT"L
LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL";:T
X0,0,0
70 NEXT I
80 N$="SI LA SOLFA MI RE DO ":E$="BAGFED
C"
90 FOR I=1 TO 13
100 X=INT(RND(1)*14+1)
110 D$=MID$(N$,3*(X+7*(X>7))-2,3)
120 GR4,6,0:CURSORX3*I-2:CURSORYX-1:PRIN
T"j";:CURSORX3*I-2:CURSORYX:PRINT"j";:CU
RSORX3*I-2:CURSORYX+1:TX4,0,0:PRINT"O"
130 TX0:CURSORX3*I-2:CURSORY18:PRINT D$
140 IF X<3 THENPLAY"O4"+MID$(E$,X,1):GOT
O 170
150 IF X<=9 THENPLAY"O3"+MID$(E$,X+7*(X>
7),1):GOTO 170
160 PLAY"O2"+MID$(E$,X+7*(X>7),1)
170 NEXT I
180 CURSORY21:INPUT"TAPEZ - RET - POUR C
ONTINUER";R$
190 GOTO 40
```

Jour de la semaine

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

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

LE PROGRAMME :
Les lignes 10 à 70 contrôlent les entrées. La ligne 40 s'occupe de la fin de partie. La ligne 90 permet une correction pour les dates antérieures à 1582 (mise en service du calendrier actuel).
Les lignes 100 à 110 permettront l'affichage en clair du mois dans la réponse (sur 4 lettres).
La formule utilisée apparaît aux lignes 130 et 140.
Les lignes 150 et 160 permettront l'affichage en clair du jour dans la réponse. La ligne 170 imprime les résultats.
La ligne 120 intervient pour tenir compte des mois de janvier et février.

POUR JOUER :
Lancer le programme par RUN, puis entrer le jour suivi de RET, le mois (entre 1 et 12) suivi de RET, et l'année (sous la forme 1845, par exemple) suivie de RET.
Le résultat est très rapidement affiché.
On tape RET et le programme revient sur la question JOUR.
Pour continuer, il suffit de retaper un jour, un mois, une année.
Pour terminer, il faut répondre 0 à la question JOUR.

EXTENSIONS POSSIBLES :
On pourra entrer les mois en clair et faire afficher le nom du jour en entier.
On peut envisager de faire afficher les dates et fêtes particulières (Noël, Pâques, les Rameaux, Pentecôte...).
On peut écrire un programme qui afficherait les jours fériés pour une année donnée, etc.

```
10 REM JOUR DE LA SEMAINE
20 INIT2,5:CLEAR300
30 INPUT"JOUR.....:";J
40 IF J=0 THENEND
50 PRINT:INPUT"MOIS.....:";M
60 PRINT:INPUT"ANNEE.....:";A
70 B=A
80 S=INT(A/100)-INT(A/400)
90 IF A<=1582 THENS=2
100 M$="JANVFEVRMARSAVRIMAI JUINJUILAOUT
SEPTOCTONOVEDECE"
110 M$=MID$(M$,4*M-3,4)
120 IF M<3 THENM=M+12:A=A-1
130 S=J-1+INT(13*(M+1)/5)+INT(5*A/4)-S
140 S=S-7*INT(S/7)
150 J$="DIMANCHELUNDI   MARDI   MERCREDI
JEUDI   VENDREDISAMEDI  "
160 J$=MID$(J$,8*S+1,8)
170 CURSORY15:TX1,4,0:PRINT"LE ";J;" ";M
$;B;" ETAIT UN ";J$:PLAY"ACE"
180 TX0,0,0:PRINT:PRINT:INPUT"TAPEZ - RE
T - POUR CONTINUER";R$
190 RUN
```

Le jeu de la vie

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

LE JEU :
La version présentée ici respecte les règles posées à l'origine par le mathématicien J.H. CONWAY.
Une certaine population est définie au début du jeu soit par le VG 5000 (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 40 et 50, pour enregistrer l'état de la population. Les boucles F (230-310) et G (240-300) sont utilisées en début de jeu pour définir la population initiale (lignes 250 et 260). Le calcul de la nouvelle population est ensuite réalisé à l'aide d'une quadruple boucle : 80-210, 90-200, 110-160, 120-150. La ligne 130 réalise la gestion du nouvel état suivant les règles.
L'affichage de ce nouvel état est obtenu en réutilisant la double boucle d'affichage F (230-310) et G (240-300).
La ligne 320 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 STOP. 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 (MANUEL)
20 INIT6,6
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 THENGOTO 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 THENC(F,G)=1
200 NEXT G
210 NEXT F
220 INIT 3,3
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I<>0 THENGOTO 280
260 TX0,0,0:CURSORX1:CURSORY20:PRINT"LIG
NE ";F;";COLONNE ";G;";TAPEZ 1 OU 0"
270 CURSORY21:INPUT C(F,G)
280 CURSORX2*G:CURSORY2*F:TX1,0,0:IF C(F
,G)=1 THENPRINT"*"
290 B(F,G)=C(F,G)
300 NEXT G
310 NEXT F
320 GOTO 80
```

```
10 REM LE JEU DE LA VIE (AUTOMATIQUE)
20 INIT6,6
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 THENGOTO 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 THENC(F,G)=1
200 NEXT G
210 NEXT F
220 INIT 3,3
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I<>0 THENGOTO 270
260 IF RND(1)<0.5 THENC(F,G)=1
270 CURSORX2*G:CURSORY2*F:TX1,0,0:IF C(F
,G)=1 THENPRINT"*"
280 B(F,G)=C(F,G)
290 NEXT G
300 NEXT F
310 GOTO 80
```

Niveau 3

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

Manipulation des 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 du VG 5000 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.

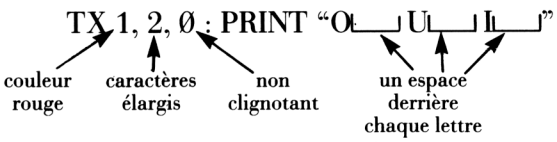
AFFICHAGE GRAND FORMAT

Pour améliorer la présentation sur l'écran d'un certain nombre de programmes, il est possible de modifier la taille des caractères affichés, qu'il s'agisse des caractères standards, des caractères graphiques, ou des caractères définis par l'utilisateur. Les caractères peuvent être agrandis en largeur, en hauteur, ou même les deux à la fois, et peuvent éventuellement être affichés en vidéo inverse (clignotant ou non).

L'AGRANDISSEMENT EN LARGEUR

Il se fait en précisant, derrière le paramètre de couleur utilisé derrière les instructions **TX** ou **ET**, un deuxième paramètre : 2 pour un agrandissement en largeur, 6 pour un agrandissement en largeur avec inversion vidéo ; un paramètre supplémentaire, 1 ou Ø, indiquera si le caractère est clignotant ou non. Mais il faudra prévoir en outre, lors de l'affichage du texte, de réserver de la place pour les caractères élargis ; ceci se fait en espaçant régulièrement les caractères dans l'instruction PRINT.

Exemple :



L'AGRANDISSEMENT EN HAUTEUR

C'est le même principe, mais le deuxième paramètre vaudra 1 ou 5 (inversion vidéo).

Pour réserver de la place en hauteur, il faudra afficher le texte à imprimer sur deux lignes différentes.

Exemple :

```
TX 1, 1, O : PRINT "OUI" : PRINT "OUI"
```

DOUBLE LARGEUR ET DOUBLE HAUTEUR

On combine les deux procédés, avec pour valeurs des paramètres 3 ou 7 (inversion vidéo).

Exemple :

```
TX 1, 3, Ø : PRINT "O␣U␣I␣" : PRINT "O␣U␣I␣"
```

Le VG 5000 dispose à l'origine de deux jeux de caractères : les caractères **standards,** accessibles par **TX,** et les caractères **graphiques,** accessibles par **GR.**

L'instruction GR utilise 3 paramètres : le 1er définit la couleur du caractère, le second la couleur du fond (mais ceci colore toute la ligne) et le 3e (Ø ou 1) le clignotement éventuel.

Pour éviter de colorer tout le reste de la ligne après un affichage en **GR,** on mettra un point virgule (;) en fin d'affichage.

Exemple :

GR 1, 4, O : PRINT "ABC" ;

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 jeux faisant appel au lancer d'un ou de plusieurs dés.

LE PROGRAMME :
La ligne 5Ø utilise le générateur aléatoire du VG 5000 pour tirer un nombre de 1 à 6.
La boucle J (7Ø-9Ø) dessine les bords du dé, et les lignes 6Ø et 1ØØ dessinent le haut et le bas.
Les lignes 12Ø à 17Ø sont utilisées pour le dessin des points.
Les lignes 18Ø et 19Ø permettront le cas échéant de refaire un tirage.

POUR JOUER :
Lancer le programme par RUN.
On entre un entier pour initialiser le générateur, et un dé s'affiche, en couleurs, et en grand format.
A la question : AUTRE PARTIE (O/N) on répond O, suivi de RET, pour relancer le dé.
Remarquer l'utilisation qui est faite des caractères élargis, en ligne 18Ø.

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 A$=CHR$(127)
40 INIT 7,6
50 A=INT(RND(1)*6+1)
60 CURSORX10:CURSORY5:GR5,7,0:PRINT"WCCC
CCCCCCCCk";
70 FOR J=0 TO 10
80 CURSORX10:CURSORY6+J:PRINT"U";:CURSOR
X22:CURSORY6+J:PRINT"j";
90 NEXT J
100 CURSORX10:CURSORY17:PRINT"upppppppppp
ppz";
110 TX1,0,0:PLAY"ACE"
120 IF A<>1 THENCURSORX13:CURSORY8:PRINT
A$
130 IF A>=4 THENCURSORX19:CURSORY8:PRINT
A$
140 IF A=6 THENCURSORX13:CURSORY11:PRINT
A$:CURSORX19:CURSORY11:PRINTA$
150 IF A/2<>INT(A/2) THENCURSORX16:CURSO
RY11:PRINTA$
160 IF A>=4 THENCURSORX13:CURSORY14:PRIN
TA$
170 IF A<>1 THENCURSORX19:CURSORY14:PRIN
TA$
180 TX3,6,0:CURSORY20:INPUT"A U T R E
P A R T I E ( O / N )";R$
190 IF LEFT$(R$,1)="O" THENGOTO 40
```

Plus grand, plus petit

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

LE JEU :
Il s'inspire d'un principe légèrement différent de celui du Mastermind ; il s'agit également de deviner un nombre (de 1 à 7 chiffres), mais les indications dont on dispose sont cette fois d'une part le nombre de chiffres qui, dans le nombre essayé, sont supérieurs à ceux du nombre à deviner, et, d'autre part, le nombre de chiffres qui sont inférieurs. Si, par exemple, le nombre à deviner est 3Ø54, 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 lignes 6Ø et 7Ø, 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-16Ø) où l'on fait la comparaison "lettre à lettre".
L'affichage du résultat est réalisé grâce à la ligne 17Ø.
Le compteur du nombre d'essais est en ligne 18Ø.

POUR JOUER :
Le lancement du programme par RUN provoque l'affichage NOMBRE DE CHIFFRES.
On entre un nombre inférieur à 8 et on appuie sur la touche RET.
Le numéro de l'essai s'affiche, ainsi que la question NOMBRE ESSAYE ; on répond en entrant un nombre, suivi de RET.
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 le VG 5000 affichera VOUS AVEZ TROUVE... EN... ESSAIS. Puis AUTRE PARTIE (O/N) ; on fera O suivi de RET 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 le VG 5000 qui tiendra alors le rôle de décodeur.

```
10 REM PLUS GRAND, PLUS PETIT
20 INIT6,3
30 INPUT"NOMBRE DE CHIFFRES :";C
40 IF C>8 THENGOTO 20
50 E=1:F=1
60 X$=STR$(INT(RND(1)*9^C+10^(C-E)))
70 X$=RIGHT$(X$,LEN(X$)-1)
80 CURSORY3:PRINT"ESSAI NO.  NOMBRE ESSA
YE    >     <  "
90 GR1,6,0:PRINT"CCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCC";
100 TX0:CURSORX1:CURSORYF+4:PRINTTAB(4)E
;TAB(10);:INPUT P$
110 IF P$=X$ THENGOTO 200
120 A=0:B=0
130 FOR I=1 TO C
140 IF MID$(P$,I,1)>MID$(X$,I,1) THENA=A
+1
150 IF MID$(P$,I,1)<MID$(X$,I,1) THENB=B
+1
160 NEXT I
170 CURSORX27:CURSORYF+4:PRINTA;TAB(34);
B
180 E=E+1:F=F+1:IF F=18 THENF=1
190 GOTO 100
200 PRINT:PRINT"VOUS AVEZ TROUVE ";X$;"
EN ";E;" ESSAIS":PLAY"ACE"
210 PRINT:INPUT"AUTRE PARTIE ( O/N )";R$
220 IF LEFT$(R$,1)="O" THENGOTO 20
```

Ordre

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

LE JEU :
Les 1Ø premières lettres de l'alphabet sont affichées dans le désordre ; il faut réussir à les remettre dans l'ordre, avec le moins de coups possible, par des permutations successives de deux des lettres ; mais ceci a aussi pour inconvénient de faire permuter autour d'elles les deux lettres qui entourent celles-ci, suivant le mécanisme expliqué ci-dessous :

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

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

LE PROGRAMME :
Les boucles emboîtées I (6Ø-12Ø) et J (8Ø-1ØØ) définissent une variable alphanumérique A$, contenant les 1Ø premières lettres de l'alphabet dans le désordre.
La ligne 14Ø affiche A$, et la ligne 15Ø affiche les numéros des lettres à utiliser pour la permutation.
On entre la permutation en ligne 19Ø.
Les lignes 2ØØ et 21Ø transforment la variable alphanumérique entrée en une variable numérique.
La ligne 22Ø appelle une première fois le sous-programme d'inversion.
La ligne 25Ø appelle une deuxième fois le sous-programme d'inversion.
La ligne 26Ø comptabilise les essais.
La ligne 18Ø 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
40 INIT6,6
50 A$=""
60 FOR I=1 TO 10
70 R$=CHR$(INT(RND(1)*10+65))
80 FOR J=1 TO I-1
90 IF R$=MID$(A$,J,1) THENI=I-1:GOTO 120
100 NEXT J
110 A$=A$+R$
120 NEXT I
130 S=1
140 CURSORY4:PRINT"RESULTAT :";:TX1,4,0:
PRINTA$
150 TX5,0,0:PRINT TAB(11);"0123456789"
160 TX0,0,0:CURSORY9:PRINT"ESSAI NUMERO
";S
170 CURSORX33:CURSORY12:PRINT"        "
180 IF A$="ABCDEFGHIJ" THENPRINT"TROUVE
EN ";S;" COUPS":PLAY"ACE":END
190 CURSORY12:INPUT"INVERSION (I ET J SO
US FORME IJ ) ";R$:R$=LEFT$(R$,2)
200 I=ASC(LEFT$(R$,1))-47
210 J=ASC(RIGHT$(R$,1))-47
220 GOSUB 500
230 IF LEFT$(R$,1)="0" OR LEFT$(R$,1)="9
" THENGOTO 250
240 I=I-1:J=I+2
250 GOSUB 500
260 S=S+1
270 GOTO 140
500 B$=""
510 FOR T=1 TO LEN(A$)
520 IF T=I THENN$=MID$(A$,T,1)
530 IF T=J THENM$=MID$(A$,T,1)
540 NEXT T
550 FOR T=1 TO LEN(A$)
560 IF T=I THENB$=B$+M$:GOTO 590
570 IF T=J THENB$=B$+N$:GOTO 590
580 B$=B$+MID$(A$,T,1)
590 NEXT T
600 A$=B$
610 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$, en ligne 4Ø, contient tous les éléments du dessin.
La boucle I (6Ø-8Ø) fabrique un mot P$ qui contient des tirets à l'emplacement de chacune des lettres du mot initial.
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 RET. 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 RET.
Si la lettre est bonne, elle viendra remplacer un ou plusieurs tirets ; sinon, la potence va commencer à se dessiner.
Le mot exact sera affiché dans le cas où l'on aurait perdu.

EXTENSIONS POSSIBLES :
Il sera intéressant de constituer une liste de mots dans lesquels le VG 5000 pourra piocher au hasard ; on pourra aussi améliorer le dessin (en utilisant SETET, par exemple) et modifier le nombre d'essais permis selon la difficulté du mot.

```
10 REM LE PENDU
20 INIT6,6
30 INPUT"VOTRE MOT :";R$
40 B$="z@uj@UOOO?????@U@@U@+Y@@U@!!!"
50 P$=""
60 FOR I=1 TO LEN(R$)
70 P$=P$+"-"
80 NEXT I
90 P=11
100 INIT3,3
110 CURSORY3:TX0,0,0:PRINTP$
120 FOR I=11 TO P STEP-1
130 CURSORX22:CURSORY12-P:GR1,3,0:PRINTM
ID$(B$,1+3*(I-1),3);:TX0,0,0
140 NEXT I
150 IF P$=R$ THENTX5,6,0:PRINT"B R A V O
 ":PLAY"ACE":END
160 IF P=1 THENPRINT:PRINT"PERDU; LE MOT
 ETAIT :";R$:PLAY"AAA":END
170 CURSORX1:CURSORY20:INPUT"VOTRE LETTR
E :";S$
180 T=0
190 FOR I=1 TO LEN(R$)
200 IF MID$(R$,I,1)=S$ THENGOSUB 300
210 NEXT I
220 IF T<>0 THENGOTO 110
230 P=P-1
240 GOTO 110
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 2Ø cartes sur 32.

LE PROGRAMME :
En ligne 5Ø, on crée un tableau pouvant contenir les 2Ø cartes.
La boucle I (8Ø-13Ø) assure le tirage aléatoire de 2Ø nombres, pris parmi 32 ; la boucle J (1ØØ-12Ø) vérifie que l'on ne réalise pas deux fois le même tirage.
La deuxième boucle I (15Ø-31Ø) assure ensuite l'affichage, les noms des cartes étant affichés en clair, et la couleur étant décodée et affichée en clair, grâce aux lignes 21Ø à 28Ø.

POUR JOUER :
Il suffit de taper RUN pour obtenir immédiatement l'affichage de 4 paquets de 5 cartes chacun (une donne).
Un nouveau RUN permet d'obtenir une nouvelle distribution.

EXTENSIONS POSSIBLES :
On pourra facilement mettre au point un affichage plus graphique en utilisant les caractères graphiques redéfinissables du VG 5000 ; on pourra également se servir de ce programme comme d'un noyau pour la mise au point de programmes de jeux de cartes sur l'ordinateur (qui pourra, si nécessaire, servir de partenaire : voir l'exemple du Poker, dans le même ouvrage).

```
10 REM BELOTE
40 INIT6,6
50 DIM A(20)
60 K=1
70 CURSORX12:CURSORY12:PRINT"PATIENTEZ,
JE TIRE"
80 FOR I=1 TO 20
90 A(I)=INT(RND(1)*32+1)
95 IF I=1 THENGOTO 130
100 FOR J=1 TO I-1
110 IF A(I)=A(J) THENI=I-1:GOTO 130
120 NEXT J
130 NEXT I
140 INIT3,4
150 FOR I=1 TO 20
160 S$="CA"
170 S=1+A(I)-8*INT(A(I)/8)
190 R$=STR$(S+6)+"  ":R$=RIGHT$(R$,LEN(R
$)-1)
200 R$=LEFT$(R$,2)
210 IF S=5 THENR$="VA"
220 IF S=6 THENR$="DA"
230 IF S=7 THENR$="RO"
240 IF S=8 THENR$="AS"
250 IF A(I)<=24 THENS$="TR"
260 IF A(I)<=16 THENS$="PI"
270 IF A(I)<=8 THENS$="CO"
280 CURSORX10*INT((I-1)/5):CURSORY2*K:DE
LIM7,7,0:TX1,0,0:PRINTR$;" ";S$;:DELIM3,
3,0
290 K=K+1:SOUND240,1,4
300 IF K=6 THENK=1
310 NEXT I
```

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 le VG 5000) et doit le transformer à l'aide de 5 opérations qu'il choisit lui-même, ainsi d'ailleurs que les opérateurs.
Le résultat qu'il a obtenu et les opérations qu'il a utilisées seront affichés sur l'écran : le décodeur devra alors essayer de retrouver le nombre initial en effectuant les opérations en sens inverse.

LE PROGRAMME :
La ligne 5Ø permet d'obtenir l'entier de départ compris entre 5 et 2Ø.
Une boucle I (11Ø-2ØØ) permet d'entrer successivement les 5 opérations.
4 sous-programmes sont utilisés pour les calculs (aux lignes 1ØØØ et 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 19Ø).
La ligne 21Ø affiche les 5 opérations utilisées, ainsi que les opérateurs correspondants.

POUR JOUER :
On frappe RUN pour lancer le programme.
Le nombre du début est affiché, puis la question ENTREZ 5 OPERATIONS ; on devra les entrer l'une après l'autre, sous la forme, par exemple : + 4 suivi de RET.
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 le VG 5000 participer davantage en lui confiant le rôle du codage, etc.

```
10 REM UN TAQUIN DE CALCUL
40 INIT6,6
50 X=INT(RND(1)*16+5)
60 PRINT"LE NOMBRE INITIAL EST :";X
70 PRINT:PRINT"ENTREZ 5 OPERATIONS"
80 PRINT:PRINT"SOUS LA FORME +5 , /2 , *
4 ..."
90 F=X
100 DIM R$(5)
110 FOR I=1 TO 5
120 CURSORY10:INPUT"VOTRE OPERATION :";R
$(I)
130 CURSORX16:CURSORY10:PRINT"       "
140 A=ASC(LEFT$(R$(I),1))-41
150 IF A<=0 OR A=3 OR A=5 OR A>6 THENGOT
O 120
160 IF A>3 THENA=(A/2)+1
170 ON A GOSUB 1000,2000,3000,4000
180 CURSORX14:CURSORY12:PRINT"       "
190 CURSORY12:PRINT"NOMBRE OBTENU :";F
200 NEXT I
210 PRINT:PRINT"OPERATIONS :";R$(4);" ";
R$(1);" ";R$(5);" ";R$(3);" ";R$(2)
220 PRINT:PRINT:PRINT"BON COURAGE"
230 PLAY"ACE":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 le VG 5000 qui va chercher à deviner votre mot, mais il joue quelque peu au hasard et ne possède pas votre intuition ; il faudra donc, en règle générale, beaucoup d'essais pour découvrir la solution. Pour augmenter ses chances, on a fait appel à une formule qui lui permet de tirer davantage dans les voyelles puis, dans les consonnes courantes, et, enfin, dans les consonnes plus rares.

LE PROGRAMME :
La ligne 5Ø 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 7Ø (en fait, il suffirait d'introduire sa longueur).
La boucle I (9Ø-12Ø) imprime des tirets aux emplacements des lettres à trouver.
La boucle en ligne 14Ø sert uniquement à simuler le temps de recherche d'une lettre.
La formule, placée en ligne 15Ø, permet de modifier légèrement la probabilité de tirage, en faveur des voyelles et des consonnes usuelles.
La ligne 17Ø provoque un nouveau tirage, et la boucle I (21Ø-24Ø) réimprime les tirets et les lettres trouvées.
La ligne 25Ø incrémente le compteur d'essais.
La ligne 26Ø affiche le nombre d'essais réalisés, en fin de partie.

POUR JOUER :
Lancer le programme par RUN.
Entrer le mot à découvrir, suivi de RET.
Des tirets sont alors affichés (un par lettre) et le VG 5000 propose une première lettre, regarde si elle convient et continue automatiquement, jusqu'à ce qu'il ait trouvé.
On pourra sortir du programme à tout moment en tapant STOP.

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 le VG 5000 ; on pourra aussi imaginer d'ajouter un affichage graphique, par exemple une potence ou une voiture qui avance vers un précipice, etc.

```
10 REM PENDU INVERSE
20 CLEAR 300
40 INIT3,3
50 D$="YAEIOUBCDFGLMNPRSTVWHKJQXZ"
60 C=0:T$=""
70 INPUT"ENTREZ VOTRE MOT :";R$
80 INIT3,3
90 FOR I=1 TO LEN(R$)
100 CURSORXI:CURSORY5:PRINT"-"
110 T$=T$+"-"
120 NEXT I
130 CURSORY9:TX1,4,1:PRINT"JE CHERCHE...
";:TX0,0,0:PRINT"ESSAI ";C
140 FOR J=1 TO 2000:NEXT J
150 X=-6*(RND(1)<0.7)-13*(RND(1)<0.5)-7*
(RND(1)<0.4)
160 X=INT(RND(1)*X+1)
170 IF MID$(D$,X,1)="-" THENGOTO 150
180 E$=MID$(D$,X,1)
190 D$=LEFT$(D$,X-1)+"-"+RIGHT$(D$,26-X)
200 PRINT:PRINT:PRINT"JE PRENDS :";E$:PL
AY"O2CEG"
210 FOR I=1 TO LEN(R$)
220 IF E$=MID$(R$,I,1) THENT$=LEFT$(T$,I
-1)+E$+RIGHT$(T$,LEN(T$)-I):PLAY"ACE"
230 CURSORXI:CURSORY5:PRINTMID$(T$,I,1)
240 NEXT I
250 C=C+1
260 IF T$=R$ THENCURSORY18:PRINT"TROUVE
";T$;" EN ";C-1;" ESSAIS":PLAY"ABCDEFG":
END
270 GOTO 130
```

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 1ØØ qui peuvent être identifiés sans ambiguïté par leurs restes dans les divisions par 3, 5 et 7.
Le programme, proposé ici, est un programme codeur·qui affiche simplement les restes d'un entier donné N dans les divisions indiquées.
Le jeu consiste à essayer de découvrir l'entier (mais, dans cette version, vous n'avez droit qu'à un seul essai).

LE PROGRAMME :
La ligne 5Ø permet de confectionner l'entier.
La boucle I (8Ø-13Ø) calcule les trois restes successifs, et la boucle emboîtée J (1ØØ-12Ø) permet de les afficher sous une forme assez visuelle, aux bons emplacements.
Les lignes 6Ø et 7Ø sont là pour indiquer de quelle division il s'agit.
Le nombre proposé par le joueur est entré à la ligne 14Ø ; il est affiché et testé aux lignes 15Ø et 16Ø (affichage de fin de partie : BRAVO ou PERDU).
Les lignes 17Ø et 18Ø 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 RET.
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 RET.

EXTENSIONS POSSIBLES :
On peut envisager de réaliser le programme décodeur (on donne les restes et le VG 5000 devra vous donner N) et chercher si la propriété peut être étendue (en effectuant davantage de divisions, par exemple).

```
10 REM NICOMAQUE
40 INIT6,6
50 X=INT(RND(1)*100+1)
60 DELIM2,2,0:PRINT"RESTE DANS LA DIVISI
ON PAR ";:DELIM6,6,0
70 TX1,6,0:PRINT:PRINT:PRINTTAB(13);"3 "
;TAB(21);"5 ";TAB(29);"7      "
80 FOR I=1 TO 3
90 A=X-(2*I+1)*INT(X/(2*I+1)):IF A=0 THE
N GOTO 140
100 FOR J=1 TO A
110 TX0:CURSORXJ+8*I+3:CURSORY5:PRINT"*"
120 NEXT J
130 NEXT I
140 TX0:CURSORY10:INPUT"VOTRE ESSAI :";N
150 TX5,3,0:IF N=X THENPRINT:PRINT"B R A
 V O ":PRINT"B R A V O "
160 IF N<>X THENPRINT:PRINT"P E R D U ":
PRINT"P E R D U "
170 TX0,0,0:PLAY"AAA":PRINT:PRINT:INPUT"
AUTRE PARTIE (O/N)";R$
180 IF LEFT$(R$,1)="O" THENRUN
```

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 1Ø, et la course débutera immédiatement.
Il pourra y avoir des ex aequo, car l'on tiendra compte du fait que le VG 5000 s'occupe de faire avancer les chevaux les uns après les autres, en commençant par le numéro 1 ; le classement sera affiché au fur et à mesure des arrivées (et en tenant compte des ex aequo).

LE PROGRAMME :
Une première boucle non formelle (8Ø-17Ø) affiche les chevaux dans leur position de départ.
Une deuxième boucle I (1ØØ-19Ø) les fera avancer ; la position de chaque cheval sera repérée par un chiffre ou une lettre dans une variable alpha-numérique A$.
La ligne 12Ø fera avancer les chevaux.
La ligne 14Ø gère les arrivées.
La ligne 15Ø réalisera les impressions.
Le sous-programme 24Ø-28Ø 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 1Ø, suivi de RET.
La course se déroulera ensuite automatiquement.

EXTENSIONS POSSIBLES :
On pourra dessiner les chevaux avec un graphisme plus soigné, et les différencier par leur couleur.
On pourra aussi imaginer tout un système de paris, qui interviendra avant le départ de chaque course.

```
10 REM COURSE DE CHEVAUX
20 A$=""
30 P=0:A=0:K=0
40 INIT7,7
50 INPUT"NOMBRE DE CHEVAUX :";N:IF N>9 T
HENGOTO 40
60 T=1
70 J=T
80 A$=A$+"0":I=J
90 GOTO 150
100 I=1
110 IF MID$(A$,I,1)="K" THEN GOTO 190
120 B$=CHR$(ASC(MID$(A$,I,1))+INT(RND(1)
*4+1))
130 A$=LEFT$(A$,I-1)+B$+RIGHT$(A$,LEN(A$
)-I)
140 IF B$>="K" THENGOSUB 240
150 CURSORXASC(MID$(A$,I,1))-46:CURSORY2
*I+3:PRINT"    ";:TX4,4,0:PRINTI:TX0,0,0
160 PLAY"A":FOR D=1 TO 800:NEXT D
170 IF K=0 THENJ=J+1:IF J<=N THENGOTO 80
180 IF K=0 THENJ=J+1:IFJ>N THENK=1:GOTO
100
190 I=I+1:IF I<=N THENGOTO 110
200 P=P+T
210 T=0
220 IF A<N THENGOTO 100
230 PLAY"ACE":END
240 A$=LEFT$(A$,I-1)+"K"+RIGHT$(A$,LEN(A
$)-I)
250 CURSORY2*I+3:TX1,4,1:PRINT P
260 T=1:PLAY"AAA"
270 A=A+T
280 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 7Ø définissent les valeurs de départ.
Les lignes 9Ø à 14Ø dessinent le terrain de jeu et la ligne 16Ø affiche le joueur qui a le trait.
La ligne 19Ø contrôle le coup joué, de même que les boucles I et J (2ØØ à 24Ø), et le changement de joueur est assuré par la ligne 33Ø.

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 RET.
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 ou de lettres et d'améliorer les contrôles. On peut aussi définir une stratégie pour le VG 5000.

```
10 REM A TRAVERS
20 DIM N(4,4):K=1
30 FOR I=1 TO 4
40 N(I,1)=ASC("0")+I
50 N(4,I)=ASC("?")+I
60 NEXT I
70 N(4,1)=0
80 INIT3,3
90 FOR U=1 TO 4
100 FOR V=1 TO 4
110 TX1,3,0:IF N(V,U)=0 THENCURSORX4*U +
1:CURSORY4*V-2:PRINT"+ ":CURSORX4*U+1 :C
URSORY4*V-1:PRINT"+ "
120 IF N(V,U)<>0 THENCURSORX4*U+1:CURSOR
Y4*V-2:PRINTCHR$(N(V,U))+" ":CURSORX4*U+
1:CURSORY4*V-1:PRINTCHR$(N(V,U))+" "
130 NEXT V
140 NEXT U
150 R$="LETTRES":IF K=1 THEN R$="CHIFFRE
S"
160 TX0:CURSORX1:CURSORY20:PRINT"JOUEUR
AYANT LES ";R$
170 CURSORY22:DELIM2,2,0
180 INPUT"VOTRE COUP (N,S,E,O) :";R$:DEL
IM3,3,0
190 IF (K=1 AND RIGHT$(R$,1)="O") OR (K=
0 AND RIGHT$(R$,1)="S") THENGOTO 170
200 I=1
210 J=1
220 IF N(I,J)=ASC(LEFT$(R$,1)) THENGOTO
260
230 J=J+1:IF J<5 THEN GOTO 220
240 I=I+1:IF I<5 THEN GOTO 210
250 GOTO 150
260 A=I:B=J
270 I=I-(RIGHT$(R$,1)="S")+(RIGHT$(R$,1)
="N")
280 J=J-(RIGHT$(R$,1)="E")+(RIGHT$(R$,1)
="O")
290 IF (I=0 AND K=0) OR (J>4 AND K=1) TH
ENGOTO 320
300 IF N(I,J)<>0 THENGOTO 150
310 N(I,J)=ASC(LEFT$(R$,1))
320 N(A,B)=0
330 K=ABS(K-1)
340 GOTO 80
```

Mastermind

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

LE JEU :
Devenu un grand classique des jeux sur micro-ordinateur, ce jeu développe le raisonnement déductif ; il faut réussir à trouver le plus vite possible un nombre de n chiffres (on pourra prendre n compris entre 1 et 5), en disposant à chaque essai de renseignements.
En effet, le VG 5000 répondra en indiquant d'une part le nombre de chiffres qui sont, dans le nombre proposé, exactement à la même place que dans le nombre à découvrir, et, d'autre part, le nombre de chiffres qui figurent dans le nombre à découvrir sans être exactement au bon emplacement.

LE PROGRAMME :
Le choix du nombre de chiffres du nombre à découvrir est effectué aux lignes 30 et 40.
Le nombre est fabriqué à la ligne 60 (il pourra y avoir des chiffres répétés).
L'affichage est préparé par la ligne 90.
La ligne 130 s'occupe de la fin de partie.
Une première boucle I (140-230) teste les chiffres qui sont à la bonne place.
Une deuxième boucle J (160-180) teste les chiffres qui sont présents et mal placés.
Les compteurs sont aux lignes 200 et 220.
La ligne 240 affiche le résultat, et la ligne 250 comptabilise les essais.

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

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 INIT6,6
30 INPUT"COMBIEN DE CHIFFRES :";N
40 IF N>6 THENRUN
50 E=1:F=1
60 X$=STR$(INT(RND(1)*9^N+10^(N-1)))
70 X$=RIGHT$(X$,LEN(X$)-1)
80 INIT 6,6
90 PRINT"NO.        NOMBRE        BON
PLACE"
100 GR1,6,0:PRINT"LLLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLL";:TX0
110 CURSORX1:CURSORY3+F:PRINTE;"
                          ";:CURSORX8:INP
UTE$
120 X=0:Y=X
130 IF E$=X$ THENPLAY"ACE":PRINT"TROUVE
";X$;" EN ";E;" ESSAIS":END
140 FOR I=1 TO N
150 IF MID$(X$,I,1)=MID$(E$,I,1) THENGOT
O 220
160 FOR J=1 TO N
170 IF J<>I AND MID$(X$,I,1)=MID$(E$,J,1
) THENGOTO 200
180 NEXT J
190 GOTO 230
200 Y=Y+1
210 GOTO 230
220 X=X+1
230 NEXT I
240 CURSORX23:CURSORY3+F:PRINTX;TAB(32);
Y
250 E=E+1:F=F+1:IF F=18 THENF=1
260 GOTO 110
```

Idem

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

LE JEU :
Le VG 5000 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 le VG 5000 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 2Ø, pour contenir les résultats successifs des deux joueurs.
La ligne 3Ø 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 sera affiché un court instant (ligne 21Ø). La boucle I (26Ø-28Ø) permet de vérifier si tous les chiffres sont identiques.
L'impression et la fin de partie sont effectuées en ligne 29Ø. Les lignes 3ØØ et 31Ø 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 RET ; 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 STOP.

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 DIM N(2):INIT6,6
30 INPUT"NIVEAU (2-6) :";N
40 IF N<2 OR N>6 THENRUN
50 H=1
60 U=10:P=0
70 N(1)=INT((U^N-U^(N-1))*RND(1)+U^(N-1)
)
80 N(2)=N(1)
90 A=INT(2*U^(N-1)*RND(1)+U^(N-1))
100 IF H=1 THENP=P+1
110 INIT6,6
120 IF N(H)<U^(N-1) THENN(H)=U^N-ABS(N(H
))
130 CURSORY3:PRINT"JOUEUR ";H;"   -  ESS
AI ";P
140 CURSORY6:PRINT"NOMBRE :":TX1,4,0:CUR
SORX15-N:CURSORY6:PRINTN(H)
150 CURSORX9:CURSORY7:TX4,0,0:PRINT"7654
321 ":TX0,0,0
160 CURSORX1:CURSORY10:INPUT"VOTRE JEU (
 N ET C SOUS FORME NC )";R$:IF R$="" THE
NGOTO 160
170 N(H)=N(H)+VAL(LEFT$(R$,1))*U^(VAL(RI
GHT$(R$,1))-1)
180 N(H)=N(H)-A
190 IF N(H)>U^N THENN(H)=N(H)-U^N
200 N(H)=INT(N(H))
210 CURSORY13:PRINT"RESULTAT :";:TX5,0,0
:PRINTN(H):TX0,0,0
220 PLAY"ACE"
230 FOR J=1 TO 2000:NEXT J
240 R$=STR$(N(H)):R$=RIGHT$(R$,LEN(R$)-1
)
250 I=0
260 I=I+1
270 IF MID$(R$,I,1)<>MID$(R$,I+1,1)THENG
OTO 300
280 IF I<LEN(R$)-1 THENGOTO 260
290 PRINT:PRINT"BRAVO, JOUEUR ";H:PLAY"A
BCDE":END
300 H=H+1
310 IF H>=3 THENH=1
320 GOTO 100
```

Horloge 1

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 : 1Ø24 signifera 1Ø heures 24 minutes).
Les lignes 8Ø-17Ø 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 la progression de l'heure.
Les lignes 23Ø à 29Ø effectuent l'affichage de deux points clignotants pour les secondes.

POUR JOUER :
Dès le lancement par RUN, le programme se place en attente de l'heure ; on tape et on attend le top pour appuyer sur RET.
L'heure s'affiche alors assez rapidement et l'horloge est en route. Pour l'arrêter, il faudra appuyer sur la touche STOP.
L'heure sera entrée sous forme d'un entier ayant impérativement 4 chiffres, pour 5 heures 6 minutes, on veillera à taper Ø5Ø6 suivi de RET.

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

```
10 REM HORLOGE
20 INIT7,7:B$=CHR$(127)
30 INPUT"HEURE ( FORME XXXX )";A$
40 INIT2,3:TX4,0,0
50 K=1
60 A=VAL(MID$(A$,K,1))
70 IF A<>1 AND A<>4 THENCURSORX6*K-5:CUR
SORY1:PRINTB$+B$+B$
80 FOR J=1 TO 5
90 IF A=0 OR (A>3 AND A<>7 ) THENCURSORX
6*K-5:CURSORYJ:PRINT B$
100 IF A<>5 AND A<>6 THENCURSORX6*K-3:CU
RSORYJ:PRINT B$
110 IF A=0 OR A=2 OR A=6 OR A=8 THENCURS
ORX6*K-5:CURSORYJ+4:PRINTB$
120 IF A<>2 THENCURSORX6*K-3:CURSORYJ+4:
PRINT B$
130 NEXT J
140 IF A>1 THENCURSORX6*K-4:CURSORY5:PRI
NTB$+B$
150 IF A<>1 AND A<>4 AND A<>7 THENCURSOR
X6*K-5:CURSORY9:PRINTB$+B$+B$
160 K=K+1
170 IF K<5 THENGOTO 60
180 A=1+VAL(A$)
190 IF A=2360 THENA=0
200 A$="000"+RIGHT$(STR$(A),LEN(STR$(A))
-1)
210 A$=RIGHT$(A$,4)
220 IF RIGHT$(A$,2)="60" THENA$=RIGHT$(S
TR$(VAL(LEFT$(A$,2))+1),2)+"00"
230 H=1
240 FOR J=1 TO 120
250 FOR I=1 TO 17
260 CURSORX11:CURSORY6:GR1,2,0:IF H=1 TH
ENPRINT"s";
270 TX4,0,0:IF H=0 THENPRINT " "
280 NEXT I
290 H=ABS(H-1)
300 NEXT J
310 PLAY"A6D":GOTO 40
```

Reines sur l'échiquier

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

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

LE PROGRAMME :
La ligne 30 vous permet de choisir la taille de l'échiquier.
Une variable D sera dimensionnée en conséquence, en ligne 40.
Une boucle d'exploration principale I (50-110) contient une sous-boucle J (80-100) qui réalise le contrôle de non-alignement (ligne 90).
Si, au cours de l'exploration, on rencontre un alignement, on revient en arrière, en utilisant les lignes 230 à 270 et on recommence.
La boucle F (130-210) assure l'impression de l'échiquier, et la boucle K (160-180), que l'on pourra éventuellement supprimer, assure l'impression du résultat sous forme d'un code numérique correspondant à la position des pièces dans chaque colonne.
La ligne 220 permet un arrêt après l'affichage de chacune des solutions.

POUR JOUER :
On lance le programme avec RUN, puis on tape la dimension d'échiquier désirée, et on fait RET.
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 RET pour le faire repartir. Pour terminer l'exploration sans avoir eu toutes les solutions, il faudra taper STOP.

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 INIT6,6
30 INPUT"TAILLE DE L'ECHIQUIER :";S
40 DIM D(S)
50 I=1
60 D(I)=1
70 IF I=1 THENGOTO 110
75 J=0
80 J=J+1
90 IF D(I)=D(J) OR (ABS(D(I)-D(J))=I-J)
THENGOTO 240
100 IF J<I-1 THENGOTO 80
110 I=I+1:IF I<=S THENGOTO 60
120 INIT 6,6
130 FOR F=1 TO S
140 CURSORX2*F+1:CURSORY1:PRINT F
150 CURSORX1:CURSORY2*F+1:PRINT F
160 FOR K=1 TO S
170 CURSORX2*K+2:CURSORY2*F+1:PRINT"+"
180 NEXT K
190 CURSORX2*D(F)+2:CURSORY2*F+1:TX1,0,0
:PRINTCHR$(127)
200 TX0,0,0:CURSORXF:CURSORY20:PRINTRIGH
T$(STR$(D(F)),1)
210 NEXT F
220 PLAY"ACE":CURSORY22:INPUT"TAPEZ - RE
T - POUR CONTINUER";R$
230 I=I-1
240 D(I)=D(I)+1
250 IF D(I)<=S THENGOTO 70
260 I=I-1
270 IF I<>0 THENGOTO 240
```

Répétitions

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

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

LE PROGRAMME :
La ligne 5Ø appelle un sous-programme (lignes 1ØØØ à 1Ø6Ø) qui donne la règle du jeu.
La ligne 7Ø fabrique les chiffres, qui sont ensuite stockés dans une variable alphanumérique A$ (ligne 8Ø).
La ligne 1ØØ 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 11Ø.
La ligne 12Ø attend la réponse du joueur, qui est comparée à la réponse correcte en ligne 13Ø.
La ligne 16Ø marque les points.
La ligne 17Ø contrôle le meilleur score.
Les affichages de fin de partie sont effectués de la ligne 2ØØ à la ligne 24Ø.

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 RET. 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 RET.
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 STOP 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 INIT6,6
30 M=0:CLEAR200
40 A$=""
50 GOSUB 1000
60 H=0
70 X=INT(RND(1)*10):B$=RIGHT$(STR$(X),1)
80 A$=A$+B$
90 INIT6,6:TX1,3,0:S=INT(RND(1)*39+1)
100 CURSORXS:CURSORY4:PRINTB$+" ":CURSOR
XS:CURSORY5:PRINTB$+" "
110 FOR I=1 TO 300+400*RND(1):NEXT I
120 INIT6,6:TX0,0,0:CURSORY10:INPUT"VOTR
E SEQUENCE :";R$
130 IF R$<>A$ THENGOTO 190
140 PRINT:PLAY"ACE":PRINT"EXACT; ATTENTI
ON"
150 FOR I=1 TO 500:NEXT I
160 H=H+1
170 IF M<H THENM=H
180 GOTO 70
190 INIT3,3
200 PRINT:PRINT"VOTRE SCORE EST DE ";H
210 PRINT:PRINT"MEILLEUR SCORE ACTUEL :"
;M
220 PRINT:INPUT"AUTRE PARTIE ( O/N )";R$
230 IF R$="N" THENPLAY"AAA":END
240 GOTO 40
1000 INIT6,6:PRINT
1010 PRINT"VOUS DEVEZ REPETER TOUTE LA S
ERIE DES"
1020 PRINT"NOMBRES APPARUS DEPUIS LE DEB
UT DU JEU"
1030 PRINT:PRINT"IL Y EN A UN DE PLUS A
CHAQUE FOIS"
1040 PRINT:PRINT"APPUYEZ SUR - RET - POU
R COMMENCER"
1050 INPUT"";R$
1060 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 (5Ø) 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.
Le nombre d'essais est incrémenté 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 RET, 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 CLEAR 500
30 DIM A$(30)
40 S=0:E=0
50 FOR I=1 TO 30:READ A$(I):NEXT I
60 INIT6,6
70 Z=INT(RND(1)*30+1)
80 IF A$(Z)="" THENGOTO 70
90 R$=A$(Z):A$="":A$(Z)="":V$=R$
100 FOR I=1 TO LEN(R$)
110 Z=INT(RND(1)*LEN(R$)+1)
120 IF MID$(R$,Z,1)="-" THENGOTO 110
130 A$=A$+MID$(R$,Z,1)
140 R$=LEFT$(R$,Z-1)+"-"+RIGHT$(R$,LEN(R
$)-Z)
150 NEXT I
160 PRINT"LES LETTRES SONT :";:TX4,4,0:P
RINTA$:TX0,0,0:PLAY"ABC"
170 CURSORY6:INPUT"VOTRE SOLUTION :";S$
180 IF S$=V$ THENS=S+1
190 E=E+1
200 CURSORY12:INPUT"AUTRE ESSAI ( O/N )"
;R$
210 IF LEFT$(R$,1)="O" THENGOTO 60
220 CURSORY15:PRINT"VOTRE SCORE :";S;" S
UR ";E;" ESSAIS"
230 PLAY"ACA":END
240 DATA"JAMBON","ENERGUMENE","AFFILIE",
"ERUDIT","INSIGNIFIANT","REALISTE"
250 DATA"IMPRESSIONNANT","ALLECHANT","EP
IQUE","INTREPIDE","COUARD","ORDINATEUR"
260 DATA"TREVE","BANDEROLLE","SENAT","AC
TUEL","CONSACRE","DIEDRE"
270 DATA"LIVRE","TERMINE","SIGNATURE","M
ALODORANT","AIMANT","ARBITRAIRE"
280 DATA"CHALEUR","OPPORTUN","PELURE","A
LTRUISTE","RAMONEUR","NOVICE"
```

Permutations

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

LE JEU :
Le VG 5000 vous demande d'abord le nombre de lettres que vous désirez permuter (pas plus de 12). A partir de là, il va vous afficher le nombre correspondant de lettres prises dans le début de l'alphabet, d'abord dans le désordre (DEBUT) puis dans l'ordre (FIN).
Vous devez réussir à remettre dans l'ordre les lettres données au début, en utilisant des permutations sur les n premières lettres à partir de la gauche, avec le moins d'essais possible.

LE PROGRAMME :
La ligne 40 permet de choisir le nombre de lettres, et la ligne 50 contrôle que l'on n'en demande pas plus de 12.
La ligne 80 réalise le brouillage, en tirant au hasard la nouvelle position de chaque lettre. Les lettres sont ensuite ajoutées au fur et à mesure de leur tirage, dans une variable B$ (ligne 100), supprimées dans A$ et replacées en C$, dans le bon ordre (lignes 110-120).
La ligne 140 contrôle le nombre de lettres de B$.
Les affichages sont effectués aux lignes 160, 170 et 180.
Le retournement est fait, lettre par lettre, dans la boucle I (210-230) ; le résultat est placé dans B$ en ligne 240, affiché par la ligne 250 et comparé au résultat final en ligne 260.
Le nombre d'essais sera affiché en fin de partie, grâce à la ligne 270.

POUR JOUER :
Après avoir tapé RUN, on obtient l'affichage de la question NOMBRE DE LETTRES (DE 1 A 12) ; on répond en frappant N, suivi de RET. 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 RET, on obtiendra AEGBDFC.
On poursuivra jusqu'à l'obtention de l'ordre désiré.
Le score sera affiché en fin de partie ainsi qu'une question AUTRE PARTIE (O/N) ; répondre O si l'on désire faire une autre partie.

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

```
10 REM PERMUTATIONS
20 INIT6,6
30 CLEAR200:PRINT
40 INPUT"NOMBRE DE LETTRES ( DE 1 A 12 )
";N
50 IF N>12 THENRUN
60 A$="ABCDEFGHIJKL":C$="            ":B
$=""
70 INIT6,6
80 X=INT(RND(1)*N+1)
90 IF MID$(A$,X,1)=" " THENGOTO 80
100 B$=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$)<>N THENGOTO 80
150 C$=LEFT$(C$,LEN(B$))
160 CURSORY4:PRINT"DEBUT  :";:TX1,5,0:PR
INTB$:CURSORX9:CURSORY3:PRINTB$
170 TX0,0,0:CURSORY6:PRINT"FIN    :";:TX
4,5,0:PRINTC$:CURSORX9:CURSORY5:PRINTC$
180 TX0,0,0:CURSORX1:CURSORY10:PRINT"PER
MUTATION NUMERO :";S:PRINT:INPUT"VOTRE Q
UANTITE ";L:IF L>N THENGOTO 180
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 PLAY"ABC":CURSORY16:PRINT"RESULTAT :
";:TX5,5,0:PRINTB$:CURSORX11:CURSORY15:P
RINTB$:TX0,0,0
260 IF B$<>C$ THENGOTO 180
270 PLAY"ACE":PRINT:PRINT"TROUVE EN ";S-
1;" ESSAIS"
280 PRINT:INPUT"AUTRE PARTIE ( O/N )";R$
290 IF LEFT$(R$,1)="O" THENRUN
```

Bridge

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

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

LE PROGRAMME :
Les lignes 5Ø et 6Ø définissent un jeu de 52 cartes.
La variable B$, en ligne 7Ø, permettra de donner en abrégé le nom des couleurs et aussi d'améliorer la désignation de certaines cartes (1Ø, valet, dame, roi et as).
La boucle J (8Ø-22Ø) réalise le tirage de quatre jeux de 13 cartes.
La boucle I (9Ø-19Ø) réalise tirage et impression. La carte est choisie parmi les 52 disponibles, en ligne 1ØØ.
La ligne 11Ø calcule la couleur de la carte tirée.
La ligne 12Ø vérifie que cette carte n'a pas déjà été tirée.
Les lignes 13Ø et 14Ø s'occupent de l'impression complémentaire dans le cas où l'on tire un 1Ø, un valet, une dame, un roi ou un as.
L'impression est réalisée par les lignes 15Ø, 16Ø et 17Ø.
La ligne 18Ø permet d'éliminer les cartes déjà tirées, et la ligne 2ØØ 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 RET pour obtenir la deuxième série de 13 cartes. On procédera de même pour la troisième et la quatrième.

EXTENSIONS POSSIBLES :
On peut imaginer de représenter les 13 cartes sous une forme plus conventionnelle.
On peut aussi ajouter la possibilité de faire des annonces (et de les comparer à celles du VG 5000, par exemple), ou introduire des possibilités de jeu de la carte, y compris pour le VG 5000.

```
10 REM BRIDGE
20 INIT6,6
30 FOR I=0 TO 14:DELIM5,5,0:PRINT:NEXT I
40 CLEAR300
50 R$="234567891VDRA"
60 R$=R$+R$+R$+R$
70 B$="TRCACOPI0AAOS"
80 FOR J=1 TO 4
90 FOR I=1 TO 13
100 X=INT(RND(1)*LEN(R$)+1)
110 S=2*INT((X-1)/13)+1
120 IF MID$(R$,X,1)="0" THENGOTO 100
130 T=X-13*INT(X/13)
140 T=T-13*(T=0)
150 CURSORX10*J-8:CURSORYI:TX0,4,0:PRINT
MID$(R$,X,1);
160 IF T>=9 THENPRINT MID$(B$,T,1)
170 CURSORX10*J-4:CURSORYI:TX(S+1)/2,0,0
:PRINT MID$(B$,S,2)
180 R$=LEFT$(R$,X-1)+"0"+RIGHT$(R$,LEN(R
$)-X)
190 NEXT I
200 DISPLAY20:PLAY"ACE":CURSORY18:TX0,0,
0:INPUT"FAIRE - RET - POUR CONTINUER";S$
210 CURSORY18:PRINT"
                    "
220 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 de diamètre supérieur.
Le nombre de coups n'est pas comptabilisé par ce programme mais, par contre, les coups illicites seront refusés.

LE PROGRAMME :
La variable D$, en ligne 40, servira à dessiner les disques.
Le nombre de disques est entré en ligne 50 et refusé s'il y a plus de 8 disques.
La boucle I (70-90) permet de fabriquer une variable A (1) qui contient l'état de la première pile sous forme numérique (en ligne 30, trois variables sont prévues pour conserver l'état des trois piles).
Les socles sont dessinés par la ligne 110 et les trois piles par les boucles emboîtées I (130-180) et J (140-170).
Le contrôle du coup est effectué par la ligne 260 et l'état des piles est modifié par les lignes 270 et 280.

POUR JOUER :
Après un RUN qui lance le programme, on entre le nombre de disques demandé, suivi de RET. 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 RET et, à la question TOUR D'ARRIVEE, on répond par le numéro de la tour d'arrivée.
Si le coup n'est pas possible, on sera obligé de recommencer à TOUR DE DEPART. Si le coup est possible, le déplacement est réalisé et l'on recommence.

EXTENSIONS POSSIBLES :
On pourra dessiner des disques de différentes couleurs, introduire un compteur de coups et, surtout, définir un jeu pour l'ordinateur.
On peut aussi essayer de compliquer le jeu avec deux tiges de plus et deux joueurs, chacun ayant des disques de couleur différente, par exemple.

```
10 REM HANOI
20 INIT7,7
30 DIM A(3):A(1)=9:A(2)=9:A(3)=9
40 D$="":FOR I=1 TO 9:D$=D$+CHR$(127):NE
XT I
50 PRINT:INPUT"NOMBRE DE TOURS ( MOINS D
E 8 ):";N
60 IF N>=8 THENRUN
70 FOR I=0 TO N-1
80 A(1)=VAL(STR$(A(1))+STR$(N-I))
90 NEXT I
100 INIT3,0
110 CURSORX1:CURSORY14:TX5:PRINT D$;:CUR
SORX13:PRINTD$;:CURSORX25:PRINTD$
120 CURSORX5:CURSORY14:TX4:PRINT"1";:CUR
SORX17:PRINT"2";:CURSORX29:PRINT"3"
130 FOR I=0 TO N-1
140 FOR J=1 TO 3
150 B$=RIGHT$(STR$(A(J)),LEN(STR$(A(J)))
-1)
160 IF LEN(B$)>I+1 THENCURSORX12*J-10:CU
RSORY13-I:TX1:PRINTLEFT$(D$,VAL(MID$(B$,
I+2,1)))
170 NEXT J
180 NEXT I
190 TX0:CURSORX1:CURSORY16
200 INPUT"TOUR DE DEPART :";D
210 IF D>3 OR D<1 THENGOTO 190
220 IF A(D)=9 THENGOTO 190
230 CURSORX1:CURSORY18:INPUT"TOUR D'ARRI
VEE :";A:IF A=D THENGOTO 230
240 B$=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) THENGOTO 190
270 A(A)=VAL(C$+MID$(B$,LEN(B$),1))
280 A(D)=VAL(LEFT$(B$,LEN(B$)-1))
290 PLAY"ABC":GOTO 100
```

Losanges

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

LE JEU :
Le programme va progressivement construire sur l'écran un losange, puis il va l'effacer progressivement, le reconstruire, et ainsi de suite.
Vous pourrez envisager de l'utiliser pour vous endormir, si les moutons ne vous suffisent plus.
Une version, très légèrement modifiée, vous est également proposée, qui vous permettra de faire apparaître des motifs aléatoires après la construction de chaque losange.

LE PROGRAMME :
Une première boucle K (5Ø-13Ø) définit les dimensions des tracés successifs.
La deuxième boucle I (6Ø-12Ø) trace des bandes de points, (lignes 8Ø à 11Ø).
Si la variable S (ligne 7Ø) est à 1, alors il y aura effacement.
On obtiendra dans le deuxième programme un effacement aléatoire en modifiant la ligne 7Ø.

POUR JOUER :
On commence l'exécution du programme avec RUN, et on regarde.
On peut arrêter à tout moment en tapant STOP.

EXTENSIONS POSSIBLES :
On pourra introduire le son, imaginer sur des principes analogues divers programmes (kaléidoscope, tapisserie...).

```
10 REM LOSANGES
20 INIT7,7
30 A$=CHR$(127)
40 S=0
50 FOR K=0 TO 10
60 FOR I=0 TO K
70 IF S<>0 THENGOTO 160
80 CURSORX21-K+I:CURSORY11-I:TXINT(RND(1
)*7):PRINTA$
90 CURSORX21-K+I:CURSORY11+I:TXINT(RND(1
)*7):PRINTA$
100 CURSORX21+K-I:CURSORY11-I:TXINT(RND(
1)*7):PRINTA$
110 CURSORX21+K-I:CURSORY11+I:TXINT(RND(
1)*7):PRINTA$
120 NEXT I
130 NEXT K
140 S=NOT S
150 GOTO 50
160 CURSORX21-K+I:CURSORY11-I:PRINT" "
170 CURSORX21-K+I:CURSORY11+I:PRINT" "
180 CURSORX21+K-I:CURSORY11+I:PRINT" "
190 CURSORX21+K-I:CURSORY11-I:PRINT" "
200 GOTO 120
```

```
10 REM LOSANGES
20 INIT7,7
30 A$=CHR$(127)
40 S=0
50 FOR K=0 TO 10
60 FOR I=0 TO K
70 IF S<>0 THENON INT(RND(1)*5+1) GOTO 1
60,170,180,190,200
80 CURSORX21-K+I:CURSORY11-I:TXINT(RND(
                                      1)
*7):PRINTA$
90 CURSORX21-K+I:CURSORY11+I:TXINT(RND(1
)*7):PRINTA$
100 CURSORX21+K-I:CURSORY11-I:TXINT(RND(
1)*7):PRINTA$
110 CURSORX21+K-I:CURSORY11+I:TXINT(RND(
1)*7):PRINTA$
120 NEXT I
130 NEXT K
140 S=NOT S
150 GOTO 50
160 CURSORX21-K+I:CURSORY11-I:PRINT" "
170 CURSORX21-K+I:CURSORY11+I:PRINT" "
180 CURSORX21+K-I:CURSORY11+I:PRINT" "
190 CURSORX21+K-I:CURSORY11-I:PRINT" "
200 GOTO 120
```

Mathieu

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

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

LE PROGRAMME :
Les 3 premiers paramètres H, K et L, introduits en ligne 3Ø, définissent le rayon maximum et l'occupation de l'écran.
Les variables aléatoires A et B, choisies entre Ø et 3, de la ligne 4Ø, définissent l'allure de la courbe ; il pourra être intéressant de les introduire manuellement (par exemple : 4Ø INPUT A, 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.
L'affichage des points est réalisé en ligne 1ØØ.
La ligne 12Ø imprime un message de fin.
La ligne 13Ø permet de refaire un nouveau dessin.

POUR JOUER :
On lance le programme en tapant RUN.
Le dessin sera ensuite exécuté automatiquement.
Le message "c'est fini" signale la fin du tracé.
Si on veut un nouveau tracé, on tape RET.
Pour arrêter, faire STOP.

EXTENSIONS POSSIBLES :
On pourra essayer d'utiliser d'autres fonctions pour réaliser des tracés encore plus extraordinaires (par exemple avec simulation de la troisième dimension).

```
10 REM MATHIEU
20 INIT0,0
30 H=2.5:K=4:L=6
40 A=3*RND(1):B=3*RND(1)
50 C=INT(RND(1)*7+1)
60 FOR I=0 TO 6.28 STEP 9.42/180
70 R=H*SIN(3*I)
80 X=21+INT(L*R*COS(A*I))
90 Y=11+INT(K*R*SIN(B*I))
100 GRC,0,0:CURSORXX:CURSORYY:PRINTCHR$(
32);
110 NEXT I
120 TX7,0,0:PRINT"C'EST FINI":PLAY"ACE"
130 INPUT A$
140 RUN
```

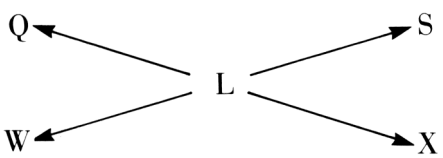
Le loup et les agneaux

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 50 contient les positions de départ.
Les boucles emboîtées I (90-130) et J (100-130) dessinent le damier, et les lignes 150 à 190 placent le loup et les agneaux.
La ligne 200 gère le changement de joueur.
La ligne 220 affiche JEU DU LOUP si c'est le joueur qui manipule le loup qui a le trait, et la ligne 230 JEU DES AGNEAUX dans le cas contraire.
Les lignes 250 à 270 analysent la réponse, en fonction du joueur concerné.

POUR JOUER :
RUN déclenche l'arrêt sur l'attente d'une entrée : 0 et RET 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 X, suivant la direction choisie :



On devra veiller à ne pas sortir de l'échiquier et à ne pas venir sur l'adversaire, aucun contrôle n'ayant été prévu.

EXTENSIONS POSSIBLES :

On pourra prévoir un repérage des cases, un graphisme plus soigné, des contrôles de non dépassement des limites du terrain, de non venue sur une case déjà occupée...

On pourra aussi envisager de jouer sur un damier de 10 cases sur 10 cases, ce qui augmente légèrement la difficulté du jeu pour les agneaux. Mais l'extension la plus intéressante à réaliser sera certainement la réalisation d'un programme permettant au VG 5000 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 le VG 5000).

```
10 REM LE LOUP ET LES AGNEAUX
20 INIT6,6
30 INPUT"AGNEAUX (0) OU LOUP (1)";S
40 INIT6,6:CLEAR200
50 A$="8183858714"
60 PRINT"     A   B   C   D   E   F   G
  H"
70 FOR I=1 TO 8:CURSORY2*I:PRINT I:NEXT
I
80 C$=CHR$(127)
90 FOR I=1 TO 8
100 FOR J=1 TO 8
110 TX1,3,0:B$=C$+" "+C$+" ":IF (I+J)/2<
>INT((I+J)/2) THENTX0:B$="    "
120 CURSORX4*J:CURSORY2*I:PRINT B$:CURSO
RX4*J:CURSORY2*I-1:PRINT B$
130 NEXT J,I
140 TX0
150 CURSORX4*VAL(MID$(A$,10,1))+1:CURSOR
Y2*VAL(MID$(A$,9,1)):TX4,4,0:PRINT"L"
160 CURSORX4*VAL(MID$(A$,2,1))+1:CURSORY
2*VAL(MID$(A$,1,1)):TX2,4,0:PRINT"1"
170 CURSORX4*VAL(MID$(A$,4,1))+1:CURSORY
2*VAL(MID$(A$,3,1)):TX2,4,0:PRINT"2"
180 CURSORX4*VAL(MID$(A$,6,1))+1:CURSORY
2*VAL(MID$(A$,5,1)):TX2,4,0:PRINT"3"
190 CURSORX4*VAL(MID$(A$,8,1))+1:CURSORY
2*VAL(MID$(A$,7,1)):TX2,4,0:PRINT"4"
200 S=ABS(S-1):SOUND100+100*S,5,5
210 CURSORY20:TX0:PRINT"
                    "
220 CURSORY20:IF S=1 THENINPUT"JEU DU LO
UP (Q,S,W,X):";R$
230 IF S=0 THENINPUT"JEU DES AGNEAUX (ND
 OU NG):";R$:GOTO 280
240 C=10
250 A=VAL(RIGHT$(A$,2))+(C+1 AND R$="X")
+(C-1 AND R$="W")
260 A=A-(C+1 AND R$="Q")-(C-1 AND R$="S"
)
270 GOTO 300
280 C=2*VAL(MID$(R$,1,1))
290 A=VAL(MID$(A$,C-1,2))-(9 AND RIGHT$(
R$,1)="D")-(11 AND RIGHT$(R$,1)="G")
300 A$=LEFT$(A$,C-2)+RIGHT$(STR$(A),2)+R
IGHT$(A$,LEN(A$)-C)
310 GOTO 70
```

Niveau 4

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

Les techniques d'animation élémentaire

SAISIE AU VOL D'UN CARACTERE FRAPPE AU CLAVIER : KEY (Ø)

Le principe d'un jeu graphique, en animation, est de laisser le jeu se dérouler tout seul, d'une certaine manière, pendant un certain temps, tant qu'aucune intervention n'a été faite par le joueur.

Il faut donc disposer de la possibilité d'observer s'il y a eu ou non intervention du joueur, sans pour cela interrompre le déroulement du programme.

Dans le Basic du VG 5000, cette propriété est offerte, par la **fonction KEY (Ø)** qui effectue une strutation des touches du clavier et enregistre le code ASCII de la dernière touche frappée.

On pourra utiliser KEY (Ø) de deux façons :

- soit dans une boucle (réalisation d'un mouvement). On saisit KEY (Ø) et on le teste pour une éventuelle sortie de la boucle.
- soit dans une attente : tant que KEY (Ø) est nul (on n'a rien frappé) on attend, et l'on sort dès qu'une touche a été frappée.

Il sera quelquefois nécessaire de taper plusieurs fois la touche pour qu'elle soit prise en compte.

DEPLACEMENTS D'OBJETS DE BAS EN HAUT SUR L'ECRAN : SCROLL

Il est difficile, si l'on travaille en Basic, de gérer le déplacement simultané de plusieurs objets sur l'écran (sinon on obtient vite une animation saccadée, à cause d'un temps de déplacement prohibitif).

Le VG 5000 réalise automatiquement un **"Scrolling"** (c'est-à-dire un déplacement de l'ensemble de l'image) vers le haut, lorsque l'on écrit sur la dernière ligne en bas de l'écran.

Cette propriété a été mise à profit dans plusieurs jeux : cela permet en effet, de déplacer d'une ligne vers le haut les divers objets (textes ou graphiques) présents sur l'écran à un moment donné. La ligne du haut est bien évidemment perdue.

Si un objet doit rester immobile sur l'écran, il sera nécessaire de l'effacer dans sa nouvelle position, et de le replacer à son emplacement initial.

Tirage de dés

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

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

LE PROGRAMME :
La ligne 4Ø saisit le code du caractère frappé au clavier et le transforme en nombre (ligne 6Ø). Si ce n'est pas possible, le programme refuse l'entrée, en ligne 5Ø.
Une variable D non dimensionnée reçoit la valeur indiquée par chaque dé.
Une boucle J (9Ø-11Ø) est utilisée pour dessiner un fond, et une autre boucle I (8Ø-14Ø) sert au tirage (tirage en ligne 12Ø et impression en ligne 13Ø).
La ligne 15Ø offre la possibilité de refaire un tirage ; elle n'est pas indispensable.

POUR JOUER :
On lance le programme par l'ordre RUN et on obtient l'affichage NOMBRE DE DES A LANCER auquel on répond en frappant un chiffre.
Le résultat est alors automatiquement affiché ; il suffit de presser une touche quelconque pour recommencer le programme.
Pour arrêter, il est nécessaire de taper STOP.

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 INIT6,6
30 CURSORX1:CURSORY1:PRINT"NOMBRE DE DES
 A LANCER :";
40 A=KEY(0):IF KEY(0)=0 OR KEY(0)=13 THE
NGOTO 40
50 IF A<49 OR A>57 THENGOTO 40
60 N=A-48
70 INIT6,6
80 FOR I=1 TO N
90 FOR J=1 TO 3
100 CURSORX4*I-3:CURSORY8+J:TX1,4,0:PRIN
T"   "
110 NEXT J
120 D(I)=INT(RND(1)*6+1)
130 CURSORX4*I-2:CURSORY10:TX4,0,0:PRINT
RIGHT$(STR$(D(I)),1)
140 NEXT I
150 DISPLAY20:PLAY"ABC":GOTO 30
```

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 2Ø, 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 (14Ø-17Ø) affiche le résultat, et la ligne 18Ø affiche le numéro complémentaire.
Les lignes 2ØØ à 22Ø s'occupent du cas où l'on voudrait refaire une autre partie.

POUR JOUER :
Il suffit de taper RUN pour lancer le programme et obtenir l'affichage d'un premier loto.
La question VOULEZ-VOUS UN AUTRE TIRAGE (O/N) s'affiche en bas de l'écran ; on répondra O si l'on désire continuer.

EXTENSIONS POSSIBLES :
On peut imaginer un programme qui pourrait tenir compte, soit de ses propres résultats, soit de résultats entrés par un joueur, pour améliorer le tirage, en tenant compte des probabilités de sortir un numéro donné.
On peut aussi envisager un programme qui réaliserait des cartons pour le jeu de loto plus usuel, tel qu'il est encore couramment pratiqué dans le Midi de la France.

```
10 REM JEU DE LOTO
20 DIM N(7)
30 INIT7,7
40 FOR I=1 TO 7
50 N(I)=INT(RND(1)*49+1)
60 IF I=1 THENGOTO 100
70 FOR J=1 TO I-1
80 IF N(I)=N(J) THENI=I-1:NEXTJ:GOTO 50
90 NEXT J
100 NEXT I
110 CURSORY3:TX4,6,0
120 PRINT"    T I R A G E   D U   L O T
O     "
130 PRINT"  - - - - - - - - - - - - - - -
- -   "
140 FOR I=1 TO 6
150 CURSORX6*I-5:CURSORY12:TX1,3,0:PRINT
MID$(STR$(N(I)),2,1)+" "+MID$(STR$(N(I))
,3,1)+" "
160 CURSORX6*I-5:CURSORY11:PRINTMID$(STR
$(N(I)),2,1)+" "+MID$(STR$(N(I)),3,1)+"
"
170 NEXT I
180 CURSORY20:TX0:PRINT"NUMERO COMPLEMEN
TAIRE :";:TX5,6,0:PRINTMID$(STR$(N(7)),2
,1)+" "+MID$(STR$(N(7)),3,1)+" "
190 CURSORY22:DISPLAY20:PLAY"ABC":TX0
200 PRINT"VOULEZ-VOUS UN AUTRE TIRAGE (
O/N )"
210 A=KEY(0):IF A=0 THENGOTO 210
220 IF A=111 THENGOTO 30
```

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 (I), 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 et 50.
La boucle I (70-110) dessine le détecteur.
Le signal de proximité est calculé en ligne 130 et la ligne 140 permet l'affichage des coordonnées de la chambre à découvrir, dans le cas où l'on a perdu.
La ligne 150 permet l'affichage, à la fois du signal et du temps restant à courir.
La ligne 160 réalise le décompte du temps.
La ligne 170 contrôle si l'on a trouvé ou non.
La boucle I (210-250) permet d'entrer les nouvelles coordonnées, en utilisant le sous-programme placé aux lignes 300 à 350.

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 RET, 45 puis RET, 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), soit jusqu'à l'apparition de 3 nombres (x,y,z) sans sortie du programme, ce qui signifie alors que l'on a perdu.

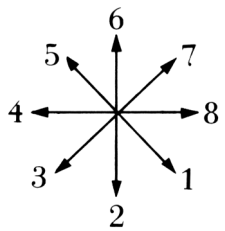
EXTENSIONS POSSIBLES :
Il sera sans doute indispensable d'améliorer le graphisme, d'ajouter une explosion (sonore et visuelle) pour symboliser l'éclatement de la bombe, d'afficher symboliquement la position dans l'hôtel, etc.

```
10 REM LA BOMBE
20 L=100:T=200
30 A=INT(RND(1)*L)
40 B=INT(RND(1)*L)
50 C=INT(RND(1)*L)
60 INIT6,6
70 FOR I=0 TO 17
80 CURSORXI+1:CURSORY4:TX4:PRINTCHR$(127
)
90 CURSORXI+1:CURSORY5:TX4:PRINTCHR$(127
)
100 CURSORXI+1:CURSORY6:TX4:PRINTCHR$(12
7)
110 NEXT I
120 S=ABS((A/L+B+C*L)-(R(1)/L+R(2)+R(3)*
L))
130 S$=LEFT$(STR$(S),7):S=VAL(S$)
140 IF T=0 THENPRINT"PERDU; LA SOLUTION
:";A;"  ";B;"  ";C:PLAY"ACE":END
150 CURSORX3:CURSORY5:PRINTS;:CURSORX13:
PRINTT
160 PLAY"A":T=T-INT(SQR(L))
170 IF S=0 THENCURSORY21:TX3,6,0:PRINT"B
 R A V O "IF"AEC":END
180 CURSORX1:CURSORY10:PRINT" - X  :"
190 CURSORX1:CURSORY12:PRINT" - Y  :"
200 CURSORX1:CURSORY14:PRINT" - Z  :"
210 FOR I=1 TO 3
220 R$=""
230 GOSUB 300
240 R(I)=VAL(R$)
250 NEXT I
260 GOTO 70
300 CURSORX8:CURSORY2*I+8:TX1:PRINT"?" ;
R$;"    ":SOUND240,5,5:TX0
310 A=KEY(0):IF A=0 THENGOTO 310
320 FOR J=1 TO 300:NEXT J
330 IF A=13 THENRETURN
340 R$=R$+CHR$(A)
350 GOTO 300
```

Ardoise magique

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

LE JEU :
Comme le titre peut sembler l'indiquer, il s'agit, tout simplement d'utiliser l'écran du téléviseur comme une ardoise, en manipulant un point sur l'écran suivant 8 directions, à l'aide des touches de 1 à 8, suivant la convention :



On dispose aussi de la possibilité de déplacer le point sur l'écran sans laisser de trace en appuyant sur la touche 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 sur une touche de A à H ; et il est possible d'effacer tout l'écran en appuyant sur P.

LE PROGRAMME :
La ligne 3Ø contient le choix des possibilités de couleur.
La ligne 5Ø imprime le point élémentaire - (ici, un carré de couleur).
Les lignes 6Ø à 9Ø contrôlent les entrées.
La ligne 13Ø appelle l'un des quatre sous-programmes effectuant le déplacement.
La ligne 14Ø vérifie que l'on ne sort pas des limites de l'écran.
La ligne 16Ø effectue l'effacement s'il y a lieu.
Le sous-programme 4ØØ-41Ø choisit la couleur ; 3ØØ-31Ø réalise ou non l'effacement.

POUR JOUER :
On lance le programme par RUN ; un carré apparaît vers le milieu de l'écran. On devra taper Ø pour le déplacer sans afficher.
Si on frappe une lettre de A à H, la couleur du point change.
La touche P efface tout, et l'on recommence.
On peut sortir à tout moment, en tapant STOP.

EXTENSIONS POSSIBLES :
On pourra, bien sûr, envisager d'écrire un programme identique en utilisant un véritable point, prévoir un sous-programme qui permette la sauvegarde du dessin réalisé pour une utilisation ultérieure, etc.

```
10 REM ARDOISE MAGIQUE
20 INIT7,7
30 C$="01234567"
40 X=10:Y=20:B=4
50 TXB:CURSORXY:CURSORYX:PRINTCHR$(127)
60 A=KEY(0):IF A=0 THENGOTO 60
70 IF A>=97 AND A<=123 THENGOSUB 400
80 IF A=48 THENGOSUB 300
90 IF A=112 THENGOTO 20
100 F=-2*((A-48)<5)-1
110 C=X:D=Y
120 U=(A-48)+2*(F-1)
130 ON U GOSUB 1000, 2000, 3000, 4000
140 E=-(X>=0 AND X<25 AND Y>0 AND Y<40)
150 IF E=0 THENGOSUB 5000
160 IF S<>0 THENCURSORXD:CURSORYC:PRINT"
 "
170 GOTO 50
300 S=ABS(S-1):FORJ=1 TO 200:NEXT J
310 RETURN
400 B=VAL(MID$(C$,A-96,1))
410 RETURN
1000 Y=Y+F
2000 X=X+F
2010 RETURN
3000 GOSUB 2000
4000 Y=Y-F
4010 RETURN
5000 X=C:Y=D
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 à 9Ø.
L'affichage sera tronqué à 38 caractères au plus, grâce aux lignes 16Ø et 17Ø.
Le défilement vers la gauche est préparé à la ligne 18Ø, et le défilement vers la droite à la ligne 19Ø.
Le texte sera affiché en lignes 16Ø et 17Ø, et la durée de l'affichage (vitesse de défilement) est choisie par la ligne 2ØØ.
La ligne 22Ø permettra de changer, si besoin est, les conditions du défilement.

POUR JOUER :
On lancera le programme par RUN, et à l'affichage de la question ENTREZ VOTRE TEXTE, on introduira le texte à faire défiler, suivi de RET.
On obtient ensuite l'affichage DEFILEMENT A DROITE (D) OU A GAUCHE (G) auquel on répondra en tapant, soit D, soit G, suivi de RET.
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 Ø à 4Ø ou plus).
Le texte commencera à défiler dès que l'on aura frappé RET.
Si l'on désire changer les conditions du défilement, il suffit d'appuyer sur la touche Ø pour retrouver les questions SENS et VITESSE. On tapera STOP 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 INIT6,6
30 B$="":C$="":CLEAR 1000
40 PRINT"ENTREZ VOTRE TEXTE":PRINT:PRINT
CHR$(96):X=POS(0)+1:Y=POS(255)
50 A=KEY(0):IF A=0 THENGOTO 50
60 IF A=13 THENGOTO 100
70 B$=B$+CHR$(A)+" ":C$=C$+CHR$(A)+" "
80 CURSORXX-1:CURSORYY+1:PRINT CHR$(A);C
HR$(96);:X=POS(0):IF X>38 THENX=2:Y=Y+1
90 FOR J=1 TO 100:NEXT J:GOTO 50
100 CURSORX1:CURSORY10:PRINT"DEFILEMENT:
A DROITE (D); A GAUCHE (G)":INPUTS$
110 PRINT:C$=B$
120 INPUT"VITESSE ( DE 0 A 50 )";V
130 V=51-V
140 INIT6,6
150 C$="  "+C$+"  ":T=LEN(C$)
160 CURSORX1:CURSORY12:TX1,3,0:PRINTMID$
(C$,1,38)
170 CURSORX1:CURSORY11:PRINTMID$(C$,1,38
)
180 IF LEFT$(S$,1)="G" THENC$=RIGHT$(C$,
T-2)+LEFT$(C$,2)
190 IF LEFT$(S$,1)="D" THENC$=RIGHT$(C$,
2)+LEFT$(C$,T-2)
200 FOR J=0 TO 10*V
210 A=KEY(0)
220 IF A=48 THENNEXTJ:TX0:INIT6,6:GOTO 1
00
230 NEXT J
240 GOTO 160
```

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 le VG 5000 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 VG 5000 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 1ØØ et 11Ø ; sa trajectoire est tracée par la ligne 8Ø, et sa position actuelle (sur l'écran, un "&") est imprimée par la ligne 12Ø.
La ligne 13Ø regarde s'il est touché ou non.
La ligne 14Ø effectue le contrôle de fin de partie.

POUR JOUER :
On commence l'exécution du programme par RUN.
Un premier tir a lieu, et ce n'est qu'après que le joueur apparaît, représenté par un "&".
Les 4 touches de direction (←, ↓ , ↑ ,→) servent à déplacer le joueur.
S'il est touché, on obtient, 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 STOP.

EXTENSIONS POSSIBLES :
La couleur et un graphisme plus soigné amélioreront considérablement l'attrait du jeu. On pourra aussi rendre le tir des bombes plus aléatoire, introduire des obstacles à contourner, ajouter des contrôles, faire participer plusieurs joueurs, etc.

```
10 REM EVASION
20 INIT 6,6
30 X=12:Y=1
40 H=(Y-3+5*RND(1))
50 K=INT(23*RND(1)+1)
60 IF Y<3 THENGOTO 80
70 GR1,6,0:CURSORXH:CURSORYK:DISPLAY240:
PRINT"?";
80 TX2:SOUND200,2,9:CURSORXY:CURSORYX:PR
INT"."
90 A=KEY(0)
100 X=X-(A=10)+(A=9)
110 Y=Y-(A=7)+(A=8)
120 CURSORXY:CURSORYX:TX4,4,0:PRINT"&"
130 IF Y>3 AND (ABS(H-Y)+ABS(K-X))<=2 TH
ENTX0:PRINT"PERDU":PLAY"ECA":END
140 IF Y<38 THENGOTO 40
150 PRINT"BRAVO":PLAY"ABCDE"
```

Chapeaux

REFERENCES :
Difficulté de programmation : assez facile.
Catégorie de jeu : jeu de réflexe, pour deux joueurs.
Difficulté du jeu : facile.

LE JEU :
L'idée en est très simple : deux personnages se déplacent l'un vers l'autre sur l'écran et se croisent. Au moment où ils se croisent, il est possible qu'ils échangent leurs chapeaux. Il s'agit d'être le premier à dire si oui ou non il y a eu échange de chapeaux lors du croisement.

LE PROGRAMME :
Les lignes 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.
La ligne 7Ø permet, s'il y a lieu, l'échange des chapeaux.
La variable T de la ligne 8Ø 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 la ligne 16Ø.
Le test, permettant ou non le changement des couvre-chefs, est effectué en ligne 18Ø.
Les lignes 2ØØ à 26Ø 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 Ø.
S'il y a eu changement de chapeau, les touches à appuyer seront 1 (à gauche) et Ø (à droite).
S'il n'y a pas eu de changement, il faudra, pour gagner, être le premier à appuyer sur 2 (à gauche) et sur 9 (à droite).
L'affichage indiquera alors BRAVO GAUCHE ou BRAVO DROITE, ou, encore, PERDU GAUCHE ou PERDU DROITE.

EXTENSIONS POSSIBLES :
Il sera possible de prévoir un affichage couleur, avec une animation des personnages.
On pourra aussi compliquer le jeu à plaisir avec plusieurs objets échangés de façon aléatoire (il faudrait alors en trouver la liste complète).

```
10 REM CHAPEAUX
20 B$="m^m"
30 C$="^m^"
40 A$="fY@@j@@j@??????---j@Uz@u"
50 X=1:Y=31
60 T=1
70 D$=C$:C$=B$:B$=D$
80 T=ABS(T-1)
90 INIT7,7
100 CURSORXX:CURSORY0:GR1,7,0:PRINTC$;
110 CURSORXY:CURSORY0:PRINTB$;
120 FORI=1 TO 8
130 CURSORXX:CURSORYI:GR4,7,0:PRINTMID$(
A$,3*I-2,3);
140 CURSORXY:CURSORYI:GR2,7,0:PRINTMID$(
A$,3*I-2,3);
150 NEXT I
160 X=X+1:Y=Y-1
170 FORI=1 TO 100:NEXT I:SOUND220,1,5
180 IF X=Y AND RND(1)<0.5 THENGOTO 70
190 R=KEY(0):IF R=0 THENGOTO 90
200 TX0,0,0:PRINT :PRINT
210 IF X<=Y THENGOTO 240
220 IF (R=49 AND T=1) OR (R=50 AND T=0)
THENPRINT"BRAVO GAUCHE":PLAY"ACE":END
230 IF (R=48 AND T=1) OR (R=57 AND T=0)
THENPRINT"BRAVO DROIT":PLAY"AEC":END
240 IF R=50 OR R=49 THENPRINT"PERDU GAUC
HE"
250 IF R=57 OR R=48 THENPRINT"PERDU DROI
T"
260 PLAY"AAA"
```

Combat aérien

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

LE JEU :
Au centre de l'écran, se trouve le viseur de votre appareil, et l'avion ennemi se déplace devant vous, d'une façon qui pourra vous sembler au début incohérente.
Vous pouvez vous déplacer dans toutes les directions, grâce aux flèches de mouvement, et vous allez essayer d'amener l'appareil ennemi au centre de votre viseur.
Vous tirez sur lui en frappant Ø, et vous voyez vos projectiles qui convergent sur lui. Mais il continue à se déplacer, même pendant votre tir, et il n'est pas du tout évident que vous réussirez à l'avoir.
La partie continuera à se dérouler jusqu'à ce que vous l'ayez touché (affichage de GAGNE) ou qu'il soit sorti de votre champ de vision.

LE PROGRAMME :
La ligne 5Ø dessine l'appareil ennemi dont le déplacement aléatoire sera réalisé par les lignes 7Ø et 8Ø.
Les lignes 1ØØ et 11Ø vous permettent le pilotage de votre appareil.
La ligne 15Ø dessine le viseur de votre appareil.
Le sous-programme, gérant le tir, va de la ligne 18Ø à la ligne 23Ø et le test de fin de partie est effectué en ligne 22Ø.
La ligne 16Ø regarde si vous avez tiré ou non.

POUR JOUER :
Dès que le programme a été lancé par l'ordre RUN, vous devez manœuvrer pour essayer de faire en sorte que l'appareil ennemi soit cadré dans votre viseur.
Pour cela, vous disposez des flèches de déplacement (touche ↑, ↓, →, ←) et vous pouvez tirer à tout moment en pressant la touche 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 touchez l'avion ennemi.
On peut aussi arrêter à tout moment en frappant STOP.

EXTENSIONS POSSIBLES :
Il sera assez facile de faire un affichage au graphisme plus soigné, utilisant la couleur. On pourrait envisager de stabiliser la position de l'avion ennemi pendant le tir (il suffit de modifier la ligne 16Ø).
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 INIT6,6:PAGE
30 A=5:B=15
40 N=21
50 CURSORXB:CURSORYA:TX1:PRINTCHR$(127)+
"-"+CHR$(127)
60 U=A:V=B
70 A=A+2*INT(RND(1)*2-1)
80 B=B+2*INT(RND(1)*2-1)
90 FOR I=1 TO 100:R=KEY(0):NEXT I
100 B=B-2*(R=7)+2*(R=8)
110 A=A-2*(R=10)+2*(R=9)
120 IF B<1 OR B>37 THENB=ABS(B-37)
130 IF A<3 OR A>22 THENA=ABS(A-22)
140 CURSORXV:CURSORYU:PRINT"   "
150 CURSORX18:CURSORY9:TX4:PRINT"* *":CU
RSORX18:CURSORY10:PRINT" X ":CURSORX18:C
URSORY11:PRINT"* *"
160 IF R=48 OR N<21 THENGOSUB 180
170 GOTO 50
180 DISPLAY240:TX5,4,0:CURSORX29-N:CURSO
RYN:PRINT">":CURSORXN+9:CURSORYN:PRINT"<
"
190 DISPLAY20:TX0,0,0:CURSORX29-N:CURSOR
YN:PRINT" ":CURSORXN+9:CURSORYN:PRINT" "
200 N=N-1
210 IF N>9 THENRETURN
220 IF (A=9 OR A=10 OR A=11) AND (B=18 O
R B=19 OR B=20) THENPRINT"GAGNE...":PLAY
"ACE":END
230 GOTO 40
```

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 5 dés sur l'écran, en utilisant une technique de dessin un peu particulière.
Chaque réponse à la question NOMBRE DE DES provoque immédiatement un nouveau lancer du nombre de dés indiqué.

LE PROGRAMME :
La ligne 4Ø permet de contrôler s'il y a eu ou non action sur une touche de 1 à 5.
Les lignes 7Ø et 8Ø initialisent les deux variables de dessin des dés.
La boucle I (9Ø-18Ø) et les lignes 1ØØ et 11Ø dessinent le cadre du dé.
La ligne 12Ø réalise le tirage, et la boucle J (13Ø-17Ø) 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 5. Les dés demandés sont alors représentés, et la question NOMBRE DE DES est de nouveau posée. Il suffira de taper STOP 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 INIT 6,6:CLEAR 100
30 CURSORX1:CURSORY15:PRINT"NOMBRE DE DE
S"
40 A=KEY(0):IF A=0 THENGOTO 40
50 N=A-48:IF N<1 OR N>5 THENRUN
60 INIT7,7
70 A$="        "+CHR$(127)+"  "+CHR$(127)
+"        "+CHR$(127)+CHR$(127)+"   "+CH
R$(127)
80 B$="1213143245155255555"
90 FOR I=1 TO N
100 CURSORX8*I-7:CURSORY2:GR4,7,0:PRINT"
WCCCCCk";
110 CURSORX8*I-7:CURSORY8:PRINT"upppppz"
;
120 X=INT(RND(1)*6+1)
130 FOR J=1 TO 5
140 IF J<=3 THENS=5*VAL(MID$(B$,3*(X-1)+
J,1))-4
150 IF J<=3 THENCURSORX8*I-6:CURSORY2*J+
1:TX1,0,0:PRINTMID$(A$,S,5)
160 CURSORX8*I-7:CURSORYJ+2:GR4,7,0:PRIN
T"U";:CURSORX8*I-1:CURSORYJ+2:PRINT"j";
170 NEXT J
180 NEXT I
190 TX0:DISPLAY20:PLAY"ACE":GOTO 30
```

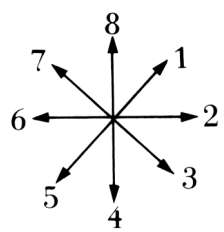
Le jeu du missile

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : contre le VG 5000.
Difficulté du jeu : moyenne.

LE JEU :
Le VG 5000 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 agrandi), et de revenir sain et sauf sur votre base ; mais, dès que le VG 5000 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 3Ø dessine la base ennemie à survoler (sous forme d'un H agrandi).
La ligne 4Ø dessine votre base de départ.
Les lignes 6Ø et 8Ø choisissent l'instant de départ du missile.
La ligne 1ØØ dessine votre avion.
Les lignes 12Ø et 13Ø définissent son mouvement suivant 8 directions, en utilisant les sous-programmes en lignes 23Ø, 24Ø, 26Ø et 27Ø.
La ligne 14Ø assure le départ du missile qui est dessiné par la ligne 15Ø et dont le déplacement est géré par les lignes 16Ø et 17Ø.
Les lignes 18Ø à 21Ø 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 STOP 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 INIT6,6:M=0
30 CURSORX17:CURSORY5:TX5,7,0:PRINT"H ":
CURSORX17:CURSORY4:PRINT"H "
40 TX4,3,0:CURSORX1:CURSORY20:PRINTCHR$(
127)+" ":CURSORX1:CURSORY19:PRINTCHR$(12
7)+" "
50 U=15:X=18
60 S=INT(RND(1)*U+6)
70 Y=1
80 S=S-1
90 V=33
100 CURSORXY:CURSORYX:TX0:PRINT"Y"
110 A=KEY(0):IF A=0 THENGOTO 110
120 F=-2*((A-48)<5)-1
130 ON A-48+2*(F-1) GOSUB 230, 240, 260,
 270
140 IF S<>0 THENGOTO 80
150 CURSORXV:CURSORYU:TX1,4,0:PRINT"<"
160 U=U-SGN(U-X)
170 V=V-SGN(V-Y)
180 IF U=X AND V=Y THENPRINT"TOUCHE...":
PLAY"ECA":END
190 IF (U=4 OR U=5) AND (V=17 OR V=18) T
HENGOTO 20
200 IF X<=4 AND Y>=18 THENM=1
210 IF M=1 AND X=18 AND Y<=4 THEN PRINT"
MISSION ACCOMPLIE":PLAY"ABCDE":END
220 GOTO 100
230 X=X-F
240 Y=Y+F
250 RETURN
260 GOSUB 240
270 X=X+F
280 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 le VG 5000 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 Ø).
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 contient 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 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 Ø 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 RET.

Les points sont alors calculés par VG 5000, et la série de lettres suivante est alors affichée.
Pour sortir du programme, il faudra taper P au lieu de Ø 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:CLEAR 200
30 V$="AEIOU"
40 C$="BCDGLMNPRSTV"
50 X=INT(RND(1)*4+3)
60 M$=""
70 INIT6,6
80 FOR I=1 TO X
90 E$=CHR$(INT(RND(1)*26+65))
100 IF RND(1)<0.4 THENE$=MID$(C$,INT(LEN
(C$)*RND(1)+1),1)
110 IF RND(1)<0.5 THENE$=MID$(V$,INT(LEN
(V$)*RND(1)+1),1)
120 M$=M$+E$
130 NEXT I
140 CURSORY6:TX1,1,0:PRINTM$:PRINTM$:PLA
Y"ABC"
150 FOR J=1 TO 500:NEXT J
160 TX0:K=KEY(0)
170 IF K=80 OR K=112 THENCURSORY10:PRINT
"SCORE DE A :";A;" ; SCORE DE B :";B:PLA
Y"ACE":END
180 IF K<>48 AND K<>49 THENGOTO 50
190 PRINT:INPUT"(J)USTE OU (F)AUX :";R$
200 A=A-(R$="J" AND K=48) + (R$="F" AND
K=49)
210 B=B-(R$="J" AND K=49) + (R$="F" AND
K=48)
220 GOTO 50
```

Le jeu du 15

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

LE JEU :
C'est un jeu pour deux joueurs, assez ancien, qui était autrefois pratiqué dans les foires. On dispose d'un tapis de jeu de 9 cases, et chacun mise à son tour un jeton de sa couleur, sur une case disponible.
Le premier des deux joueurs qui, avec exactement 3 jetons, arrive à faire un total de 15, a gagné.

LE PROGRAMME :
La boucle I (5Ø-8Ø) dessine le terrain de jeu, et les lignes 9Ø à 1ØØ s'occupent de la gestion du tour des joueurs (numéro 1 ou 2 pour chaque joueur, et jeton O ou X correspondant).
La ligne 11Ø attend le coup joué.
La boucle I (12Ø-14Ø) vérifie la validité du coup.
La ligne 15Ø teste la fin de partie, en cas de match nul, et la ligne 16Ø refuse les coups non conformes (avec retour à la situation initiale, ligne 9Ø).
Les lignes 17Ø et 18Ø s'occupent de l'affichage du coup, et la ligne 19Ø 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 21Ø à la ligne 3ØØ : 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 31Ø.

POUR JOUER :
Lancer le jeu en tapant RUN.
Le terrain de jeu s'affiche, puis la question JOUEUR 1 : X, et le joueur qui a choisi les X devra donner le numéro de la case où il désire jouer si elle est libre, son coup y sera enregistré.
On obtient, ensuite, l'affichage JOUEUR 2 : O et le deuxième joueur procède de la même façon. Si l'un des joueurs gagne (total de 15 sur 3 coups), il obtient l'affichage BRAVO et la partie s'arrête.
En cas de match nul, la machine affiche C'EST FINI.
On fera une autre partie en tapant RUN.

EXTENSIONS POSSIBLES :
Le plus intéressant sera sans doute d'essayer de définir une stratégie machine (en essayant d'éviter que la machine ne joue toujours le 5 lorsqu'elle doit commencer).
On peut aussi, bien sûr, améliorer le graphisme et introduire des effets sonores pour les différents messages.

```
10 REM LE JEU DU QUINZE
20 INIT 7,7:CLEAR 200
30 F=0:R$="123456789"
40 O$="":X$=""
50 FOR I=0 TO 24 STEP 3
60 CURSORXI+1:CURSORY1:GR5,7,0:PRINT"WCk
";:CURSORXI+1:CURSORY2:PRINT"U@j";:CURSO
RXI+1:CURSORY3:PRINT"upz";
70 CURSORXI+1:CURSORY4:TX2,4,0:PRINT(I+3
)/3;" "
80 NEXT I
90 TX0:CURSORY10:PRINT"JOUEUR ";F+1;
100 PRINT" : ";CHR$(79-9*(F=0))
110 A=KEY(0):A=A-48:IF A<1 OR A>9 THENGO
TO 110
120 FOR I=1 TO LEN(R$)
130 IF A=VAL(MID$(R$,I,1)) THENGOSUB 170
140 NEXT I
150 IF VAL(R$)=0 THENCURSORY20:PRINT"C'E
ST FINI":PLAY"AEC":END
160 GOTO 90
170 IF F=1 THENO$=O$+RIGHT$(STR$(A),1):C
URSORX3*I-1:CURSORY2:TX4:PRINT"O"
180 IF F=0 THENX$=X$+RIGHT$(STR$(A),1):C
URSORX3*I-1:CURSORY2:TX1:PRINT"X"
190 R$=LEFT$(R$,I-1)+"0"+RIGHT$(R$,LEN(R
$)-I):SOUND150+80*F,3,8
200 T$=X$:IF F=1 THENT$=O$
210 IF LEN(T$)<3 THENGOTO 310
220 FOR I=1 TO 3
230 IF I=LEN(T$) THENGOTO 310
240 FOR J=I+1 TO LEN(T$)
250 IF J=LEN(T$) THENGOTO 310
260 FOR K=J+1 TO LEN(T$)
270 IF VAL(MID$(T$,I,1))+VAL(MID$(T$,J,1
))+VAL(MID$(T$,K,1))=15 THENGOTO 330
280 NEXT K
290 NEXT J
300 NEXT I
310 TX0:F=ABS(F-1)
320 RETURN
330 TX6,7,1:R$="B R A V O   J O U E U R
"
340 CURSORY14:PRINTR$+CHR$(79-9*(F=0))+"
 "
350 CURSORY15:PRINTR$+CHR$(79-9*(F=0))+"
 "
360 PLAY"ABCDE"
```

Adresse

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

LE JEU :
Des chiffres, entre Ø 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 1Ø points. La durée de la partie est aléatoire et le score sera affiché en fin de partie.

LE PROGRAMME :
Les lignes 5Ø et 6Ø 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 le VG 5000 est choisi aux lignes 7Ø et 8Ø, et le chiffre est tiré en ligne 9Ø.
La ligne 1ØØ réalise l'impression, et la ligne 11Ø détermine la durée d'affichage. La ligne 13Ø définit la durée du jeu, et la fin de partie est traitée par la ligne 15Ø.
La boucle I (2ØØ-23Ø) réalise le dessin grand format du chiffre frappé par le joueur.
Le contrôle de l'exactitude du résultat est effectué par la ligne 24Ø.

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 1Ø 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 STOP.

EXTENSIONS POSSIBLES :
On peut envisager de modifier le jeu pour faire participer plusieurs joueurs, en modifiant éventuellement l'affichage et en codant les touches du clavier mises à la disposition de chacun des joueurs.
On pourra aussi enlever des points à chaque résultat faux, modifier les règles de calcul du score et la durée du jeu, etc.

```
10 REM ADRESSE
20 CLEAR 300
30 T=100:S=0:C$=CHR$(127)
40 INIT7,0
50 C$=C$+C$+C$+C$+"     "+C$+C$+" "+C$
60 B$="144413333313121131314413312131121
411333314141141 31"
70 X=INT(RND(1)*15+1)
80 Y=INT(RND(1)*15+15)
90 N=INT(RND(1)*10):N$=RIGHT$(STR$(N),1)
+" "
100 CURSORXY:CURSORYX:TX1,3,0:PRINTN$:CU
RSORXY:CURSORYX+1:PRINTN$
110 FOR I=0 TO RND(1)*450+300:NEXT I
120 SOUND200,5,1:A=KEY(0):IF A<>0 THENGO
TO 190
130 T=T-N
140 CURSORXY:CURSORYX:PRINT"  ":CURSORXY
:CURSORYX+1:PRINT"  "
150 IF T>=0 THENGOTO 70
160 INIT7,0
170 TX4,4,0:PRINT:PRINT"VOTRE SCORE EST
DE ";S;" SUR 100"
180 PLAY"ACE":END
190 U=A-48
200 FOR I=1 TO 5
210 A=3*VAL(MID$(B$,5*U+I,1))-2
220 CURSORX2:CURSORYI:TX3:PRINTMID$(C$,A
,3)
230 NEXT I
240 IF U=N THENS=S+10:PLAY"A"
250 GOTO 130
```

Sous-marin

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

LE JEU :
Vous manœuvrez en surface un chasseur de sous-marins, et des sous-marins passent très régulièrement en dessous de votre unité, à différentes profondeurs.
Vous pouvez, à tout moment, larguer des grenades sous-marines et vous pouvez même envisager de toucher plusieurs fois le même sous-marin.
Votre score sera affiché en permanence, sur la gauche de l'écran (rapport nombre de coups au but sur nombre de sous-marins apparus).

LE PROGRAMME :
La ligne 7Ø permet au VG 5000 de choisir la profondeur du sous-marin.
La ligne 9Ø dessine votre bâtiment.
Les lignes 1ØØ et 11Ø affichent le score et dessinent la surface de l'eau.
La ligne 12Ø dessine le sous-marin.
Le tir est effectué par un programme spécifique placé aux lignes 2ØØ à 24Ø.
Le déplacement (très rapide) de votre navire est réalisé par les lignes 17Ø et 18Ø.
Le test, permettant de savoir si l'on a touché ou non, est effectué en ligne 14Ø.

POUR JOUER :
Taper RUN pour commencer la partie.
Le bâtiment sera déplacé grâce aux flèches droite et gauche.
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 STOP ou d'attendre 99 sous-marins.

EXTENSIONS POSSIBLES :
Il sera assez élémentaire d'améliorer la qualité graphique du jeu, d'utiliser la couleur et les sons, de faire apparaître plusieurs sous-marins se déplaçant à des vitesses différentes, éventuellement même, de permettre aux sous-marins de tirer...

```
10 REM SOUS-MARIN
20 INIT6,6:P=0:C=15:S=5
30 FOR J=0 TO 3
40 CURSORX0:CURSORYJ:DELIM7,7,0
50 NEXT J
60 TX1:CURSORX1:CURSORYS:PRINT"      "
70 T=33:S=INT(RND(1)*8+5):P=P+0.01
80 X=1
90 CURSORXC:CURSORY3:GR5,7,0:PRINT"@@@lt
]E@@@";
100 CURSORX1:CURSORY4:GR4,6,0:PRINT"CCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC";
110 CURSORX1:CURSORY18:TX1:PRINT"SCORE :
";P
120 CURSORXT:CURSORYS:GR3,6,0:PRINT"dxp@
";
130 IF X=1 THENY=C+2*X
140 IF S=X AND (T+1=Y OR T+2=Y) THENP=P+
1:CURSORXT:CURSORYS:TX1:PRINT"***":PLAY"
AAA"
150 T=T-1:IF P=99 THENCURSORY18:TX2,2,0:
PRINT"C ' E S T   F I N I ":PLAY"ABC":EN
D
160 IF T<1 THENGOTO 60
170 A=KEY(0)
180 C=C-3*((A=7 AND C<27)-(A=8 AND C>1))
190 IF A<>48 AND X=1 THENGOTO 90
200 CURSORXY:CURSORYX
210 TX1:PRINT" ":X=X+1
220 IF X>S THENGOTO 80
230 CURSORXY:CURSORYX:PRINT"O"
240 GOTO 90
```

Zombies

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

LE JEU :
Les ZOMBIES sont des êtres malfaisants qui vont essayer de vous détruire, mais vous allez, vous aussi, essayer de les utiliser à votre profit, pour débarrasser votre espace vital des trous noirs qui l'obscurcissent. Sur l'écran, est tracé un cadre qui définit votre espace vital. Dans ce cadre, vont apparaître de façon aléatoire, les ZOMBIES (ce sont des Z inversés), les trous noirs (des carrés noirs) et le joueur (un O). Les ZOMBIES convergent vers le joueur pour essayer de le capturer, mais celui-ci se déplace plus rapidement qu'eux, et il va essayer de leur faire avaler tous les trous noirs, sans se faire prendre.
La partie se termine, soit avec la disparition du joueur, soit par l'élimination de tous les ZOMBIES.

LE PROGRAMME :
Un tableau T à deux dimensions permettra d'enregistrer la position des trous noirs (ligne 2Ø).
La position des ZOMBIES est définie par un tableau Z non dimensionné (c'est inutile en dessous de 1Ø).
Les lignes 5Ø à 12Ø (boucle I) dessinent le cadre, les ZOMBIES et les trous noirs.
La ligne 13Ø définit la position initiale du joueur.
Une deuxième boucle I (15Ø-24Ø) réalise le déplacement du joueur.
Les lignes 29Ø à 34Ø contrôlent le déplacement du joueur.

POUR JOUER :
Après avoir lancé l'exécution par RUN, le joueur devra se déplacer en utilisant les 4 flèches et en veillant à ne pas sortir des limites du terrain.
Il devra manœuvrer de façon à faire venir les ZOMBIES sur les trous noirs, sans se faire lui-même coincer.
Pour arrêter la partie avant la fin, il suffira de taper STOP.

EXTENSIONS POSSIBLES :
Il serait intéressant d'agrandir le terrain de jeu, d'augmenter le nombre de ZOMBIES ou de trous noirs, d'améliorer les contrôles, de placer le joueur de façon plus aléatoire, éventuellement de modifier la règle.
On pourra aussi, bien sûr, améliorer la qualité du dessin.

```
10 REM ZOMBIES
20 DIM T(24,22)
40 INIT7,7
50 FOR I=0 TO 21
60 TX4:CURSORXI+2:CURSORY0:PRINTCHR$(127
)
70 CURSORX1:CURSORYI:PRINTCHR$(127);:CUR
SORX24:PRINTCHR$(127)
80 IF RND(1)<0.5 THENCURSORXI+2:M=INT(RN
D(1)*20+1):TX1,4,0:CURSORYM:PRINT"#":T(I
+2,M)=1
90 S=INT(RND(1)*14+1)
100 Z(INT(I/3)+1)=5+S
110 CURSORXI+2:CURSORY21:TX4:PRINTCHR$(1
27)
120 NEXT I
130 K=INT(RND(1)*18+1)
140 P=0
150 FOR I=1 TO 4
160 P=P+Z(I)
170 IF Z(I)=0 THENGOTO 240
180 TX1:CURSORXZ(I+4):CURSORYZ(I):PRINT"
 "
190 Z(I)=Z(I)-SGN(Z(I)-S)
200 Z(I+4)=Z(I+4)-SGN(Z(I+4)-K)
210 IF T(Z(I+4),Z(I))=1 THENZ(I)=0:GOTO
230
220 CURSORXZ(I+4):CURSORYZ(I):TX5,4,0:PR
INT"Z"
230 IF Z(I)=S AND Z(I+4)=K THENPRINT"VOU
S AVEZ PERDU...":PLAY"ABA":END
240 NEXT I
250 IF P=0 THENPRINT"BRAVO:VOUS AVEZ DET
RUIT TOUS LES ZOMBIES":PLAY"ABCDE":END
260 TX0:CURSORXK:CURSORYS:PRINT"O"
270 FOR I=1 TO 1500:NEXT I
280 CURSORXK:CURSORYS:PRINT" "
290 A=KEY(0)
300 K=K-2*((A=7)-(A=8))
310 IF K<=1 OR K>=24 THENPRINT"DOMMAGE..
.":PLAY"AEC":END
320 S=S-2*((A=10)-(A=9))
330 IF S<=0 OR S>=21 THENK=25:GOTO 310
340 GOTO 140
```

Diago

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

LE JEU :
On lance le programme par RUN et on obtient un premier affichage qui vous renseigne sur la performance à réaliser (par exemple, 5 points noirs ou 17 points noirs...), puis le temps est affiché et décompté, à partir de 1Ø secondes. Le jeu commence lorsque l'on arrive à Ø seconde. Des ronds apparaissent alors en bas et à gauche de l'écran, formant une ligne diagonale, et il vous faudra appuyer sur la touche Ø pour arrêter l'affichage des points dès que vous estimez que le nombre de points demandé 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 Ø 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 19Ø à 22Ø.

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 Ø, la diagonale apparaît et commence à se déplacer sur l'écran.
Si l'on estime que le nombre de points réclamé a été atteint, on devra presser la touche Ø.
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 INIT 6,6
30 X=INT(RND(1)*18+3)
40 S=10
50 PRINT:PRINT"LE BON REFLEXE EST A ";X
60 CURSORX1:CURSORY10:PRINT"ATTENTION ;
IL RESTE ";S;" SECONDES "
70 S=S-1
80 PLAY"A":FOR I=1 TO 1200:NEXT I
90 IF S<>0 THENGOTO 60
100 INIT7,7:U=0
110 CURSORY25:PRINTTAB(U+1);"O"
120 FORI=1 TO 30:NEXT I
130 U=U+1
140 A=KEY(0)
150 IF A=48 THENGOTO 180
160 IF U>22 THENPRINT"PERDU":PLAY"AEC":G
OTO 200
170 GOTO 110
180 IF U<>X THENU=23:GOTO 160
190 PRINT"BRAVO":PLAY"ABCDE"
200 PRINT:PRINT"AUTRE ESSAI ( O/N )"
210 A=KEY(0):IF A=0 THENGOTO 210
220 IF A=79 OR A=111 THENRUN
```

Décomaque

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

LE JEU :
C'est une réplique au programme Nicomaque, proposé dans les pages précédentes ; vous entrez les restes successivement dans les divisions par 3, 5 et 7 d'un entier compris entre 1 et 100, et le VG 5000 calcule et affiche cet entier. Le procédé retenu peut d'ailleurs être généralisé (à vous de trouver comment).

LE PROGRAMME :
Les lignes 30 à 50 affichent quelques explications.
La ligne 90 attend que vous frappiez un reste, qui sera refusé s'il est incompatible avec la division par le nombre affiché.
La ligne 100 emmagasine les restes dans une variable non dimensionnée A (S).
La ligne 130 réaffiche les restes entrés.
Les boucles I (140-180) et J (150-170) dessinent des petits carrés selon les restes introduits.
Les lignes 210 à 230 gèrent la fin de partie.

POUR JOUER :
On lance le programme par RUN. Un texte de présentation est affiché, puis le message "RESTE DANS LA DIVISION PAR 3" s'affiche.
On doit répondre 0, 1 ou 2, et l'on obtient le message "RESTE DANS LA DIVISION PAR 5" etc.
La solution est affichée dès que l'on a donné les 3 restes.
Pour une autre partie, taper O à la question VOULEZ-VOUS UN AUTRE ESSAI (O/N).
Tapez N pour finir.

EXTENSIONS POSSIBLES :
On peut étendre le programme aux entiers de 1 à 1000, en utilisant les restes par 3, 5, 7 et 11 ; on pourra rajouter du son et de la couleur pour rendre le programme plus attrayant.

```
10 REM DECOMAQUE
20 INIT0,0:TX7
30 PRINT"VOUS DEVEZ ENTRER LES TROIS RES
TES"
40 PRINT:PRINT "DANS LES DIVISIONS PAR 3
 , 5 ET 7"
50 PRINT :PRINT "D'UN NOMBRE ENTIER INFE
RIEUR A 100"
60 I=3:S=1
70 CURSORX1:CURSORY10:PLAY"ABC"
80 TX1:PRINT"RESTE DANS LA DIVISION PAR
";:TXI-1:PRINTI
90 A=KEY(0):IF A<48 OR A>47+I THENGOTO 9
0
100 A(S)=A-48
110 S=S+1:I=I+2
120 IF I<8 THENGOTO 70
130 CURSORY16:TX7,1,0:R$="RESTE :   PAR
3      PAR 5      PAR 7":PRINTR$:PRINTR$
140 FOR I=1 TO 3:IF A(I)=0 THENGOTO 180
150 FOR J=1 TO A(I)
160 CURSORX2*J+10*I-3:CURSORY14;TX2*I:PR
INTCHR$(127):SOUND150+30*I,2,5
170 NEXT J
180 NEXT I
190 T=A(1)*70+A(2)*21+A(3)*15
200 IF T>105 THENT=T-105:GOTO 200
210 CURSORY19:TX3,5,0:R$="LA SOLUTION ES
T"+STR$(T):PRINTR$:PRINTR$:DISPLAY20:PLA
Y"ACE"
220 CURSORY22:TX7:PRINT"VOULEZ-VOUS UN A
UTRE ESSAI ( O/N )"
230 A=KEY(0):IF A=0 THENGOTO 220
240 IF A=79 OR A=111 THENRUN
```

Chasse au sous-marin

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

LE JEU :
Vous êtes à bord d'un bâtiment de surface, chargé de traquer un sous-marin caché quelque part dans votre secteur ; vous devez vous déplacer jusqu'à venir au-dessus du sous-marin et tirer à une certaine profondeur ; mais, attention : le sous-marin, lui aussi, est mobile ; il ne coupera ses moteurs que lorsque vous serez exactement à sa verticale, pour ne pas être repéré.

LE PROGRAMME :
La boucle I (60-120) affiche le terrain.
La ligne 140 définit la position initiale du sous-marin ; la ligne 150 permet son déplacement, contrôlé par la ligne 160 (on reste dans les limites du terrain).
La ligne 180 affiche votre position.
Les lignes 220 à 240 contrôlent votre déplacement.
La distance est calculée en ligne 250, affichée en ligne 260 et testée en lignes 270 et 280.
Le test de fin de partie est effectué à la ligne 310.

POUR JOUER :
On lance le programme par RUN et on entre la dimension ; la grille de jeu s'affiche. Vous êtes au départ en position (0, 0) et vous vous déplacez avec les flèches.
Après chaque déplacement, la nouvelle distance du sous-marin s'affiche ; à chaque coup le sous-marin se déplace. Si vous réussissez à vous placer à sa verticale, la distance 0 est affichée et il ne bouge plus.
Vous tirez en donnant une profondeur entière (entre 2 et 4, bornes comprises). Si vous le ratez, il se replace n'importe où, et on recommence ; si vous le coulez, vous obtenez l'affichage BRAVO et la partie s'arrête.
On peut arrêter à tout moment le programme en utilisant la touche STOP.

EXTENSIONS POSSIBLES :
Il est possible de prévoir une amélioration du graphisme, 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 INIT7,7
30 PRINT:INPUT"DIMENSION DE LA GRILLE :"
;H
40 IF H>10 THENRUN
50 INIT6,6:H=H-1
60 FOR I=0 TO H
70 TX0:CURSORX2*I+3:CURSORY0:PRINT I
80 CURSORX1:CURSORY2*I+2:PRINTRIGHT$(STR
$(I),1)
90 FOR J=0 TO H
100 TX4:CURSORX2*J+4:CURSORY2*I+2:PRINT"
+"
110 NEXT J
120 NEXT I
130 I=1:X=0:Y=0
140 V=INT(RND(1)*H):U=INT(RND(1)*H):PLAY
"GGG"
150 U=U+I:V=V+I
160 IF (U<0 OR U>H OR V<0 OR V>H) THENI=
-I:PLAY"GGG"
170 CURSORX2*Y+2:CURSORY2*X+2:IF Y<>0 TH
ENTX4:PRINT"+"
180 CURSORX2*Y+4:CURSORY2*X+2:TX1:PRINTC
HR$(127)
190 CURSORX2*Y+6:CURSORY2*X+2:TX4:IF Y<>
H THENPRINT"+"
200 CURSORX2*Y+4:CURSORY2*X:IF X<>0 THEN
PRINT"+"
210 CURSORX2*Y+4:CURSORY2*X+4:IF X<>H TH
ENPRINT"+"
220 A=KEY(0):IF A=0 THENGOTO 220
230 Y=Y-(A=7 AND Y<H)+(A=8 AND Y>0)
240 X=X-(A=10 AND X<H)+(A=9 AND X>0)
250 D=INT(SQR(ABS(X-V)^2+ABS(Y-U)^2))
260 CURSORX30:CURSORY10:PRINT"DIST.";STR
$(D)+" ":SOUND240,3,7
270 IF D=0 THENGOTO 300
280 IF D<3 THENGOTO 150
290 GOTO 170
300 CURSORX30:CURSORY12:INPUT"PROF.";D
310 IF D=INT(RND(1)*3+2) THENPRINT"BRAVO
":PLAY"ACE":END
320 CURSORX30:CURSORY12:PRINT"          "
:PLAY"AAA"
330 GOTO 140
```

Clavier

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

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

LE PROGRAMME :
Une variable A$ dimensionnée en ligne 5Ø, contiendra la liste des touches concernées. Les valeurs des touches sont entrées de la ligne 6Ø à la ligne 11Ø par appel d'un sous-programme situé en 3ØØ.
Les boucles emboîtées J (12Ø-17Ø) et I (13Ø-16Ø) dessinent le clavier.
La ligne 18Ø définit la position du point imprimé par la ligne 19Ø.
La touche frappée est testée aux lignes 22Ø-23Ø et le point est effacé par la ligne 25Ø.
Les affichages de fin sont réalisés aux lignes 27Ø à 29Ø.

POUR JOUER :
On lance le programme par RUN. L'on dispose alors de quelques secondes pour frapper la touche qui occupe la colonne au-dessus de l'étoile, puis une autre étoile est affichée, etc.
La durée de la partie n'est pas limitée ; il suffit de taper Ø pour obtenir l'affichage du score, et du temps total.
Faire RUN pour une autre partie.

EXTENSIONS POSSIBLES :
On pourra envisager de faire clignoter les lettres, ou de les afficher en couleur, au lieu d'indiquer la colonne par une balle. On pourra afficher les lettres devinées justes en bas de l'écran, et éventuellement s'entraîner à taper un texte qui défile en haut de l'écran, lettre par lettre.

```
10 REM CLAVIER
20 CLEAR 300
30 INIT
40 T=0:S=0:P=0
50 DIM A$(10,3)
60 S$="AZERTYUIOP":J=1
70 GOSUB 300
80 S$="QSDFGHJKLM":J=2
90 GOSUB 300
100 S$="WXCVBN":J=3
110 GOSUB 300
120 FOR J=1 TO 3
130 FOR I=1 TO 10+3*(J=3)
140 CURSORX3*I+J:CURSORY5*J:TX1,3,0:PRIN
TA$(I,J)+" "
150 CURSORX3*I+J:CURSORY5*J-1:PRINTA$(I,
J)+" "
160 NEXT I
170 NEXT J
180 X=INT(RND(1)*10+1):Y=INT(RND(1)*3+1)
:IF Y=3 AND X>6 THENGOTO 180
190 CURSORX3*X+Y:CURSORY5*Y+2:TX4,3,0:PR
INT"* ":CURSORX3*X+Y:CURSORY5*Y+1:PRINT"
* ":SOUND240,3,4
200 F=1:L=100+140*RND(1)
210 A=KEY(0):T=T+1:IF A=0 THENF=F+1:IFF<
LTHENGOTO 210
220 IF A-32=ASC(A$(X,Y)) THENS=S+1
230 IF A=48 THENGOTO 270
240 P=P+1
250 CURSORX3*X+Y:CURSORY5*Y+2:PRINT"  ":
CURSORX3*X+Y:CURSORY5*Y+1:PRINT"  "
260 GOTO 180
270 CURSORY20:TX0:PRINTS;" TROUVES JUSTE
S SUR ";P;" ESSAIS"
280 PRINT:PRINT"VOTRE TEMPS :";T
290 PLAY"ABCDE":END
300 FOR I=1 TO LEN(S$):A$(I,J)=MID$(S$,I
,1):NEXT I:RETURN
```

Le lézard d'Alice

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

LE JEU :
Il s'inspire très vaguement d'une idée de LEWIS CARROLL, dans *Alice au Pays des Merveilles*. Dans l'histoire originale, Alice grandissait ou diminuait, alors que dans le jeu, pour des raisons d'encombrement de la mémoire, c'est un lézard qui change de taille.
Deux verres apparaissent sur l'écran : l'un contient le liquide qui fait grandir, l'autre le liquide qui fait rapetisser (malheureusement ce n'est pas toujours le même verre). Le joueur devra indiquer la quantité de liquide qu'il désire boire et le lézard changera de taille en fonction de cette quantité. Le but du jeu est de parvenir à lui rendre sa taille initiale.

LE PROGRAMME :
La boucle I (60-90) dessine les verres.
Les deux boucles emboîtées I (180-220) et J (190-210) les remplissent avec la quantité indiquée en ligne 170.
La variation de taille du lézard est définie par la ligne 260, avec contrôle par les lignes 270 et 280.
La boucle I (290-310) dessine le corps du lézard, les lignes 320 et 330 dessinent l'œil et les pattes.
Un message gagnant sera affiché par la ligne 350 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 l'affichage QUELLE QUANTITE ; on entre alors une quantité (de 0 à 130) et on frappe RET.
Le verre désigné se remplit sous vos yeux, puis, après quelques instants, le lézard est affiché. On devra alors frapper une touche quelconque ; si le lézard n'avait pas la bonne taille, aucun message n'est affiché et l'on revient au dessin des verres.

EXTENSIONS POSSIBLES :
Puisqu'il s'agit d'un jeu essentiellement graphique, il sera intéressant d'améliorer les dessins, d'introduire la couleur, de modifier le caractère aléatoire du choix des verres, etc.
Pourquoi ne pas relire *Alice au Pays des Merveilles*, les idées n'y manquent pas.

```
10 REM LE LEZARD D'ALICE
40 INIT7,7
50 N=18:TX5
60 FOR I=1 TO 9
70 CURSORX3:CURSORYI+4:PRINTCHR$(127):CU
RSORX10:CURSORYI+4:PRINTCHR$(127)
80 CURSORX13:CURSORYI+4:PRINTCHR$(127):C
URSORX20:CURSORYI+4:PRINTCHR$(127)
90 NEXT I
100 FOR I=3 TO 10:CURSORXI:CURSORY14:PRI
NTCHR$(127);:CURSORXI+10:PRINTCHR$(127)
110 NEXT I
120 TX1:CURSORX4:CURSORY15:PRINT"GAUCHE"
;:CURSORX14:PRINT"DROIT"
130 TX0:CURSORY17:PRINT"QUEL COTE :";
140 A=KEY(0):IF A<>100 AND A<>103 THENGO
TO 140
150 U=2-10*(A=100):R$="GAUCHE":IF A=100
THENR$="DROIT"
160 PRINT R$
170 PRINT:INPUT"QUELLE QUANTITE :";Q
180 FOR I=1 TO SQR(Q/2)
190 FOR J=0 TO 5
200 TX4:CURSORXU+J+2:CURSORY14-I:PRINTCH
R$(127)
210 NEXT J
220 SOUND160+6*I,2,6:NEXT I
230 A=KEY(0):IF A=0 THENGOTO 230
240 INIT0,0
250 H=2*(RND(1)<0.5)-1
260 N=N-H*INT(SQR(Q))
270 IF N<5 THENN=5
280 IF N>35 THENN=35
290 FOR I=0 TO N
300 GR1,0,0:CURSORXI+3:CURSORY15:PRINT"L
";:SOUND250,1,1
310 NEXT I
320 CURSORX5:CURSORY16:PRINT"&)";:CURSOR
XN-5:PRINT"&)";
330 CURSORXN-2:CURSORY14:TX1:PRINT"O"
340 A=KEY(0):IF A=0 THENGOTO 340
350 IF N=28 THENPRINT"BRAVO":PLAY"ABCDE"
:END
360 GOTO 40
```

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 15Ø s'occupent de la gestion des entrées.
La boucle I (16Ø-19Ø) réalise 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 2ØØ et 21Ø.
Le nombre total des symboles entrés est contrôlé par la ligne 22Ø.
Le résultat est inscrit grâce à un programme fonctionnant en automatique qui se trouve aux lignes 27Ø à 39Ø (l'automatisme est géré par la ligne 23Ø, dans le programme principal).
Les sous-programmes placés aux lignes 4ØØ-41Ø, 42Ø-43Ø, 44Ø-45Ø et 46Ø-47Ø effectuent l'opération indiquée, selon la valeur de Z (ligne 31Ø).
Chaque résultat est mémorisé en A (F-1) (ligne 33Ø).

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 STOP, puis RUN.
Taper STOP pour sortir du programme.

EXTENSIONS POSSIBLES :
On devra, tout d'abord, améliorer l'affichage du signe − (représenté, ici, par un trait), puis on pourra améliorer le dessin des chiffres (en utilisant, par exemple, le générateur du VG 5000), on peut aussi envisager une extension à plus de 8 chiffres, autoriser des opérations plus complexes (COS, SIN, ...), ajouter des mémoires, etc.

```
10 REM CALCULATRICE
20 C$=CHR$(127):C$=C$+C$+C$+C$+"    "+C$
+C$+" "+C$
30 B$="144413333313121131314413312131121
411333314141141 31"
40 F=1:L=0:INIT6,6:DIM A(50)
50 N=0:T=0:K=0:R$=""
60 A=KEY(0):IF A=0 THEN GOTO 60
70 SOUND240,4,1:IF K=1 THEN N=0:R$="":K=
0
80 IF A=61 THEN GOTO 260
90 X=(1 AND A=43)+(2 AND A=45)+(3 AND A=
42)+(4 AND A=47)
100 R$=R$+CHR$(A)
110 IF X<>0 THEN GOTO 260
120 IF R$="-" THEN A=ASC(R$):GOTO 200
130 IF A<46 OR A>57 OR A=47 THEN GOTO 60
140 IF N=0 THEN INIT6,6
150 IF A=46 OR A=45 THEN GOTO 200
160 FOR I=1 TO 5
170 S=3*VAL(MID$(B$,I+5*(A-48),1))-2
180 CURSORXN+1:CURSORYI+3:TX4:PRINTMID$(
C$,S,3)
190 NEXT I
200 IF A=45 THEN INIT6,6:CURSORX1:CURSOR
Y7:TX4,2,0:PRINT"- "
210 IF A=46 THEN CURSORXN+1:CURSORY7:TX4
,1,0:PRINT",":CURSORXN+1:PRINT","
220 TX4:N=N-4*(N<28)
230 IF T<>0 THEN T=T+1:IF T<= LEN(R$) TH
EN GOTO 380
240 IF K=1 AND L<>1 THEN GOTO 50
250 T=0:GOTO 60
260 IF R$="-" AND A(F)=0 THEN GOTO 120
270 N=0:K=1
280 IF (A=42 OR A=47) AND VAL(R$)=0 THEN
 A(F)=1:GOTO 300
290 A(F)=VAL(R$)
```

```
300 IF X<>0 THEN Z=X
310 ON Z GOSUB 400, 420, 440, 460
320 F=F+2+(A<>61)
330 A(F-1)=VAL(R$)
340 IF VAL(R$)=0 THEN A(F-1)=A(F)
350 R$=STR$(VAL(R$))
360 IF LEFT$(R$,1)=" " THEN R$=RIGHT$(R$
,LEN(R$)-1)
370 T=1
380 A=ASC(MID$(R$,T,1))
390 GOTO 140
400 R$=STR$(A(F-1)+A(F)):A(F)=VAL(R$):L=
1
410 RETURN
420 IF F<>1 THEN R$=STR$(A(F-1)-A(F)):A(
F)=VAL(R$):L=1
430 RETURN
440 IF F<>1 THEN R$=STR$(A(F-1)*A(F)):A(
F)=VAL(R$):L=1
450 RETURN
460 IF F<>1 THEN R$=STR$(A(F-1)/A(F)):A(
F)=VAL(R$):L=1
470 RETURN
```

Niveau 5

Nom du programme	*Thèmes et particularités*
Dames	Pour deux joueurs : classique.
Smokey	Fumez sans risque, avec ce programme original pour un joueur.
Course de voitures	Un jeu de réflexe pas trop difficile, pour un joueur.
Echecs	Un programme assez simplifié, pour deux joueurs.
Danger spatial	Serez-vous assez adroit pour éviter toutes les météorites ?
Tic Tac Toe	Un classique, mais pas sans intérêt (deux joueurs).
Horloge 2	Utilisation du crayon optique de l'ordinateur.
Anagrammes	Un bon jeu de réflexe : pour deux joueurs.
Oeufs	Ne laissez pas faire la pondeuse diabolique (un joueur).
Taquin de lettres	Un proverbe est caché dedans : à vous de le découvrir.
Cibles	Un peu d'adresse, beaucoup de chance : très visuel.
Gouttes	Un véritable gag visuel. A exploiter.
Obstacles	Beaucoup d'obstacles à éviter dans ce jeu d'adresse pour un joueur.
Captures dans l'espace	Votre mission serait facile, sans ces damnées météorites.
Anomalie	Exercez votre perspicacité dans la reconnaissance des symboles.
Plusieurs en un	Un terrain de jeu et les pions, pour deux joueurs.
Parcours du cavalier	Très spectaculaire, et instructif à plus d'un titre.
Mur de briques	Un grand classique dans une version très complète.
Othello	Pour deux joueurs ; il retourne les pions à votre place.
Quatre à quatre	Un jeu de réflexion un peu difficile, pour deux joueurs.

Les techniques spécifiques du VG 5000

LES INSTRUCTIONS D'EXPLORATION DES CASES MEMOIRES. PEEK

Cette instruction (utilisée avec l'instruction complémentaire POKE) permet de lire (et d'écrire) directement dans l'une quelconque des cases mémoires adressables par le microprocesseur. On pourra donc, en utilisant judicieusement des **PEEK** et des **POKE**, remédier aux insuffisances du langage Basic, en allant même jusqu'à l'écriture, à partir du Basic, de programmes en langage machine.

Sur le VG 5000, la fonction **PEEK** est surtout utilisée pour lire directement le fichier d'affichage, le fichier débute à l'adresse 16 384, et chaque case affichée à l'écran utilise deux positions mémoires : la première contient le code ASCII du caractère affiché, la seconde contient des informations sur le mode, la couleur, le clignotement, etc.

On utilise donc la formule :

PEEK (16 384 + 80 x X + 2 × Y)

pour lire le contenu de la ligne **X** et de la colonne **Y** de l'écran - (case contenant le code caractère).

PEEK (16 385 + 80 x X + 2 × Y)

permet de lire les indications complémentaires (couleur, etc.).

LES CARACTERES DEFINIS PAR L'UTILISATEUR : SETET ET SETEG

Le VG 5000 offre à l'utilisateur la possibilité de redéfinir assez facilement 96 caractères en mode texte, et 96 caractères en mode graphique. Chaque caractère est défini dans une matrice 8 × 1Ø, par la donnée de 1Ø nombres en hexadécimal, suivant une méthode très élémentaire.

On écrira, par exemple :

SETET Ø65, "ØØ/38/44/44/44/7C/44/44/ØØ/ØØ"

Code du caractère redéfini — soit 20 caractères, de Ø à F.

Les caractères ainsi redéfinis seront identifiés sur l'écran par leur code, et peuvent être manipulés de la même façon que les autres (agrandissement, clignotement, couleur, etc.).

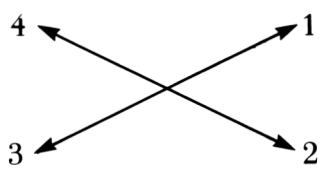
Dames

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

LE JEU :
C'est le jeu classique de dames à la française, mais un peu incomplet puisqu'il ne prévoit pas la possibilité de faire des dames. La règle du jeu reste classique, mais les contrôles ont été un peu simplifiés. Les mouvements des pièces sont entrés en une seule manœuvre (position de départ et direction du déplacement) et les prises sont réalisées automatiquement, si elles sont possibles.
Par contre le changement de joueur étant réalisé à chaque coup, il faudra éventuellement taper Ø pour pouvoir rejouer s'il y avait eu saut ou si le coup n'était pas possible.

LE PROGRAMME :
Les boucles I (8Ø-17Ø) et J (9Ø-16Ø) dessinent le damier et définissent les positions initiales des pions noirs et des pions blancs. Les positions sont dans une table R (X, Y) et la couleur dans une autre table C (X, Y).
L'impression des pions proprement dite est effectuée par les boucles emboîtées J (19Ø-25Ø) et I (2ØØ-24Ø).
La ligne 27Ø indique le joueur qui a le trait.
Le coup est entré en ligne 27Ø.
La ligne 28Ø réalise le changement de joueur (on tape Ø).
La ligne 3ØØ regarde si le déplacement est possible.
La ligne 33Ø traite le cas d'un déplacement et la ligne 34Ø celui d'un saut.
La ligne 38Ø 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 de couleur apparaît en dessous pour indiquer que le trait appartient au joueur correspondant.
On joue en indiquant par un mot de 3 lettres (pouvant commencer par un Ø) la ligne et la colonne de la pièce à déplacer, puis la direction du déplacement (codée de 1 à 4) suivant la convention :



Par exemple 9D1 signifie que le O situé en ligne 9, colonne D, se déplace dans la direction 1 (ne pas oublier de faire RET).
De même ØG3 signifiera pièce noire, ligne Ø, colonne G, direction 3.

Le changement de joueur étant effectué automatiquement il faudra taper Ø pour rejouer si le coup est refusé ou s'il y a eu prise (toujours terminer par RET). 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 au VG 5000 de se mêler avec de bonnes chances de réussite à vos passionnantes parties.

```
10 REM DAMES
20 DIM R(10,10):DIM C(10,10):SETET66,"00
3C7EFFFFFFFF7E3C00":SETET65,"00000000000
000000000"
30 A$=CHR$(124)+CHR$(124)+CHR$(124)
40 B$=CHR$(127)+CHR$(127)+CHR$(127)
50 C$=CHR$(67)+CHR$(67)+CHR$(67)
60 INIT6,6
70 PRINT"   A  B  C  D  E  F  G  H  I  J
"
80 FOR I=1 TO 10
90 FOR J=1 TO 10
100 IF (I+J)/2=INT((I+J)/2) THEN GOTO 13
0
110 GR1,6,0:CURSORX3*J:CURSORY2*I-1:PRIN
TA$;:CURSORX1:CURSORY2*I:TX0:PRINTI-1;:C
URSORX3*J:GR1,6,0:PRINTB$;
120 CURSORX3*J:CURSORY2*I+1:PRINTC$;
130 R(I,J)=65:IF (I+J)/2<>INT((I+J)/2) T
HEN GOTO 160
140 IF J<5 THEN R(I,J)=66:C(I,J)=4
150 IF J>6 THEN R(I,J)=66:C(I,J)=3
160 NEXT J
170 NEXT I
180 H=1
190 FOR J=1 TO 10
200 FOR I=1 TO 10
210 IF (I+J)/2<>INT((I+J)/2) THEN GOTO 2
40
220 X=3*J+1:Y=2*I
230 CURSORXX:CURSORYY:ETC(I,J):PRINTCHR$
(R(I,J))
240 NEXT I
250 NEXT J
```

```
260 TX0:CURSORX25:CURSORY22:PRINT"
":SOUND150+70*H,8,6
270 CURSORX1:CURSORY22:PRINT"JOUEUR ";:E
T3-(H=1):PRINTCHR$(66);:TX0:INPUT" ;VOTR
E COUP :";D$
280 IF D$="0" THEN H=ABS(H-1):GOTO 260
290 GOSUB 400
300 IF R(L,C)=65 OR ((C(L,C)=3 AND H=1)
OR (C(L,C)=4 AND H=0)) THEN GOTO 260
310 T=(1 AND (D=1 OR D=2)) - (1 AND (D=4
 OR D=3))
320 S=(1 AND (D=3 OR D=2)) - (1 AND (D=1
 OR D=4))
330 IF R(L+S,C+T)=65 THEN R(L,C)=65:R(L+
S,C+T)=66:C(L+S,C+T)=3-(H=1):GOTO 380
340 IF C(L+S,C+T)=3-(H=0) AND R(L+S,C+T)
=66 AND R(L+2*S,C+2*T)=65 THEN GOTO 360
350 GOTO 260
360 R(L,C)=65:R(L+S,C+T)=65:R(L+2*S,C+2*
T)=66:C(L+2*S,C+2*T)=3-(H=1)
370 GOTO 190
380 H=ABS(H-1)
390 GOTO 190
400 L=VAL(LEFT$(D$,1))+1:C=ASC(MID$(D$,2
,1))-64:D=VAL(RIGHT$(D$,1))
410 RETURN
```

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 sur la droite. Il doit dans un premier temps se positionner sur la colonne de fumée, puis à partir de là, il va automatiquement avaler la fumée, à condition qu'il réussisse à se maintenir sur sa trace.

LE PROGRAMME :
La ligne 90 fait apparaître la fumée au bas de l'écran ; la ligne 100 permet de changer (vers la droite ou vers la gauche) la direction de la colonne, et la ligne 120 fait monter la colonne.
Le joueur apparaît grâce à la ligne 140, et son déplacement est effectué par les lignes 180 et 190.
Les lignes 250 et 260 se chargent de l'affichage du résultat.

POUR JOUER :
On doit lancer l'exécution du programme par RUN.
La fumée commence à s'élever, tandis que le joueur, apparu sous forme d'un rond coloré 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 SETET 065,"001C3E7F7F7F3E1C0000"
30 SETET 066,"0000367F7F3E1C080000"
40 S=0
50 X=24:Y=INT(RND(1)*24+5)
60 U=0
70 V=15
80 INIT2,2
90 CURSORXY:CURSORYX:ET1:PRINT"B"
100 Y=Y-(RND(1)<0.5 AND Y<24)+(RND(1)<0.
5 AND Y>0)
110 T=1
120 X=X-T
130 IF X=0 THEN GOTO 250
140 CURSORXV:CURSORYU:ET4:PRINT"A"
150 I=1:IF T=0 THEN GOTO 170
160 I=I+1:A=KEY(0):IF I<300 AND A=0 THEN
 GOTO 160
170 TX0:CURSORXV:CURSORYU:PRINT" "
180 U=U-(A=10 AND T=1 AND X<24)
190 U=U-(T=0)
200 V=V-(A=7 AND V<24)+(A=8 AND V>0)
210 C=PEEK(16385+80*U+2*V):TX0
220 IF C<>1 THEN GOTO 90
230 T=0:S=S+1:SOUND240,1,5
240 IF U<23 THEN GOTO 140
250 TX0:IF T=0 THEN PRINT"BRAVO":PLAY"AB
CDEFG":GOTO 270
260 PRINT"PERDU":PLAY"AAA"
270 PRINT: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 affichée en permanence, et votre score est affiché dès que vous avez un accident.

LE PROGRAMME :
L'impression de la route est assurée par la boucle I (7Ø-1ØØ) et la ligne 9Ø imprime la couleur de la route.
La ligne 11Ø fait avancer la route, et la ligne 12Ø imprime la voiture.
La ligne 16Ø contrôle son mouvement, et la ligne 17Ø détermine si la route fait ou non des virages.
La ligne 18Ø regarde si l'on a touché le bord de la route.
La ligne 19Ø incrémente le score, et la ligne 2ØØ efface l'ancienne position de la voiture (une trace claire apparaît à l'emplacement de sa trajectoire).
La ligne 23Ø affiche le temps avant l'accident.

POUR JOUER :
Lancer l'exécution du programme par l'ordre RUN, le dessin de la route commence à évoluer, et votre véhicule, représenté par une voiture, apparaît.
Pour le déplacer vers la droite ou vers la gauche, vous devrez utiliser les touches → et ← (flèches de déplacement du curseur). Il vous faudra éviter le plus longtemps possible de toucher une bordure noire, ce qui provoquera l'arrêt du programme et l'affichage de la mention ACCIDENT, suivie de l'indication sur le score réalisé.
On pourra arrêter le déroulement de la partie à tout moment en pressant la touche STOP.

EXTENSIONS POSSIBLES :
On peut envisager de jouer à plusieurs, chacun son tour, en enregistrant le meilleur score.
On peut aussi envisager d'élargir la piste, et de faire courir plusieurs voitures.
On pourra aussi s'inspirer des jeux d'arcades qui sont particulièrement nombreux sur ce thème.

```
10 REM COURSE DE VOITURES
20 SETET065,"18BDFFBD24243CBDFF99"
30 T=0:S=0
40 X=24:Y=11
50 A=5:B=10
60 INIT7,7
70 FOR I=0 TO 6
80 CURSORXY+I:CURSORYX:TX7:PRINT" ";
90 IF I<2 OR I>4 THEN CURSORXY+I:CURSORY
X:TX2:PRINT CHR$(127);
100 NEXT I
110 PRINT
120 CURSORXB:CURSORYA:ET1:PRINT"A"
130 I=1
140 I=I+1:K=KEY(0):T=T+0.1:IF I<10 AND K
=0 THEN GOTO 140
150 C=B
160 B=B-(K=7)+(K=8)
170 Y=Y-(2*RND(1) AND Y<25)+(2*RND(1) AN
D Y>3)
180 IF PEEK(16384+80*(A+1)+2*B)=127 THEN
 GOTO 220
190 S=S+1
200 CURSORXC:CURSORYA:TX0:PRINT" "
210 GOTO 70
220 TX0:PRINT"ACCIDENT :SCORE :";S
230 PRINT:PRINT"TEMPS :";INT(T):PLAY"ABA
ABAABAABA"
```

Echecs

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexion, pour deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
Le programme propose seulement à deux joueurs de disputer une partie en se servant de l'écran du téléviseur comme plateau de jeu. Il n'y a pas de contrôle de la validité des coups, mais seulement gestion de la position des pièces sur l'échiquier. Les pions et les pièces sont dessinés à l'aide des caractères utilisateurs. La couleur permet de distinguer chaque camp.

LE PROGRAMME :
Les lignes 100 à 200 dessinent l'échiquier. Les lignes 210 et 220 définissent les conditions initiales, à l'aide de deux tableaux : R (I, J) pour les codes des pièces, et C (I, J) pour la couleur.
Les boucles emboîtées J (240-290) et I (250-280) dessinent les pièces.
La ligne 320 permet l'entrée de la ligne et de la colonne de départ, avec contrôle en ligne 350.
La ligne 370 permet d'entrer colonne et ligne d'arrivée.
La ligne 400 note la nouvelle position.
La ligne 410 assure le changement de joueur.

POUR JOUER :
On lance l'exécution par RUN.
L'échiquier est dessiné, puis les pièces, et enfin l'inscription JOUEUR BLEU suivie de DEPART (C et L) ; on entre la colonne et la ligne de départ, sous la forme E 7 par exemple, on obtient alors l'affichage de ARRIVEE (C et L) et on répond de même. Le déplacement est alors effectué, et c'est au tour de l'autre joueur.
Pour arrêter, il suffit de taper STOP.

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 au VG 5000 de se comporter en partenaire honorable. Bon courage !

```
10 REM ECHECS
20 DIM R(8,8):DIM C(8,8)
30 SETET065,"00001028446C387C0000":SETET
066,"0000103810384438000O":SETET068,"000
05428106C7C7C0000"
40 SETET067,"0000547C38387C7C0000":SETET
069,"0000001038381O7C0000":SETET070,"000
010387C18387C0000"
50 SETET071,"0000000000000000000"
60 A$="ppp"
70 B$=CHR$(127)+CHR$(127)+CHR$(127)
80 C$="CCC"
90 INIT7,7
100 PRINT"   A  B  C  D  E  F  G  H"
110 FOR I=1 TO 8
120 FOR J=1 TO 8
130 IF (I+J)/2=INT((I+J)/2) THEN GOTO 15
0
140 GR0,7,0:CURSORX3*J:CURSORY2*I-1:PRIN
TA$;:CURSORX3*J:CURSORY2*I:PRINTB$;:CURS
ORX3*J:CURSORY2*I+1:PRINTC$;
150 IF I=2 OR I=7 THEN R(I,J)=69
160 IF I<>2 AND I<>7 THEN R(I,J)=71
170 C(I,J)=1
180 IF I=7 OR I=8 THEN C(I,J)=4
190 NEXT J
200 TX0:CURSORX1:CURSORY2*I:PRINTI:NEXT
I
210 R(1,1)=67:R(1,2)=70:R(1,3)=65:R(1,4)
=68:R(1,5)=66:R(1,6)=65:R(1,7)=70:R(1,8)
=67
220 FOR I=1 TO 8:R(8,I)=R(1,I):NEXT I
230 H=1
240 FOR J=1 TO 8
250 FOR I=1 TO 8
260 X=1+3*J:Y=2*I
270 CURSORXX:CURSORYY:ETC(I,J):PRINTCHR$
(R(I,J))
280 NEXT I
290 NEXT J
300 CURSORX28:CURSORY6:TX0:PRINT"
 ":CURSORX18:CURSORY20:PRINT"    ":CURSO
RX19:CURSORY22:PRINT"    "
310 CURSORX1:CURSORY18:R$="ROUGE":IFH=1
THENR$="BLEU "
320 PRINT"JOUEUR ";R$:PLAY"AEC":CURSORX1
:CURSORY20:INPUT"DEPART (C ET L) :";D$
330 CURSORX28:CURSORY6:PRINT D$
340 GOSUB 500
350 IF R(L,C)=71 OR (C(L,C)=1 AND H=1) O
R (C(L,C)=4 AND H=0) THEN GOTO 310
360 T=R(L,C):S=C(L,C):R(L,C)=71
370 CURSORX1:CURSORY22:INPUT"ARRIVEE (C
ET L) :";D$
380 CURSORX31:CURSORY6:PRINT"- ";D$
390 GOSUB 500
400 R(L,C)=T:C(L,C)=S
410 H=ABS(H-1)
420 GOTO 240
500 L=VAL(RIGHT$(D$,1)):C=ASC(LEFT$(D$,1
))-64
510 RETURN
```

Danger spatial

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

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

LE PROGRAMME :
La position de chaque météorite est déterminée par les lignes 6Ø et 8Ø ; elle est imprimée par les lignes 7Ø et 9Ø.
L'avance des météorites est réalisée par la ligne 1ØØ.
Les lignes 1ØØ et 11Ø vérifient si vous avez ou non heurté une météorite.
La ligne 13Ø dessine votre vaisseau spatial, et la ligne 16Ø permet de le déplacer.
Le compteur se trouve en ligne 17Ø.

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 STOP. 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
40 INIT 2,2
50 X=18:Z=0:Y=X
60 H=INT(RND(1)*39+1)
70 IF RND(1)<0.7 THEN CURSORXH:CURSORY24
:TX1:PRINTCHR$(127);
80 K=INT(RND(1)*39+1)
90 IF RND(1)<0.6 THEN CURSORXK:CURSORY24
:TX1:PRINTCHR$(127);
100 PRINT:B=PEEK(16384+960+2*X):C=PEEK(1
6384+962+2*X)
110 IF B=127 OR C=127 THEN GOTO 190
120 CURSORXY-1:CURSORY11:TX0:PRINT"     "
:CURSORXY-1:CURSORY10:PRINT"     "
130 CURSORXX:CURSORY12:TX4:PRINT"##"
140 A=KEY(0)
150 Y=X
160 X=X-(A=7 AND X<39)+(A=8 AND X>0)
170 Z=Z+1
180 GOTO 60
190 PRINT"LE SCORE EST DE ";Z:PLAY"ACE"
```

Tic tac toe

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

LE JEU :
Un grand classique des jeux d'écoliers, proposé ici dans une version ne comportant pas le jeu machine ; chaque joueur devra placer à son tour un O ou un X (ici un cœur ou un carreau) dans une case libre de la grille de jeu, et le premier qui réussit un alignement de 3 symboles identiques a gagné (l'alignement peut être horizontal, vertical, ou suivant l'une des deux diagonales).

LE PROGRAMME :
Le terrain de jeu est dessiné à l'aide d'une boucle I (40-80).
La ligne 100 s'occupe d'afficher le symbole cœur ou carreau correspondant au joueur qui a le trait.
Un symbole clignotant (ligne 130) est affiché en haut et à gauche du terrain, même si la case est déjà occupée ; les lignes 160 à 180 permettent de déplacer le symbole sur le terrain de jeu.
La ligne 190 vérifie si la case où l'on veut jouer est libre, et si l'on désire jouer.
La ligne 200 affiche le coup, et la ligne 210 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 0, le coup sera refusé si la case n'est pas libre.
Si la case est libre, le symbole qui correspond à ce joueur est inscrit, et on change de joueur.
Lorsque la partie est finie, il convient de taper STOP 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 SETET065,"0000367F7F7F3E1C0800":SETET
066,"00081C3E7F3E1C080000":INIT7,6
30 FOR I=1 TO 14:TX5,4,0:PRINT"
     ":NEXT I
40 FOR I=1 TO 3
50 FOR J=1 TO 3
60 TX7,3,0:CURSORX4*I-1:CURSORY4*J-2:PRI
NTCHR$(127)+CHR$(127):CURSORX4*I-1:CURSO
RY4*J-1:PRINTCHR$(127)+CHR$(127)
70 NEXT J
80 NEXT I
90 K=0
100 CURSORX1:CURSORY18:TX0:PRINT"JOUEUR
";:ET0:PRINTCHR$(65+K)
110 I=1:J=1
120 ET0:C=PEEK(16302+320*J+8*I)
130 CURSORX4*I-1:CURSORY4*J-2:TX0,3,0:PR
INT"XX":CURSORX4*I-1:CURSORY4*J-1:PRINT"
XX":FORF=1TO300:NEXTF
140 IF C=32 OR C=127 THEN CURSORX4*I-1:C
URSORY4*J-2:PRINT"  ":CURSORX4*I-1:CURSO
RY4*J-1:PRINT"  ":GOTO 160
150 TX7:CURSORX4*I-1:CURSORY4*J-2:ET4+3*
(C=193),3,0:PRINTCHR$(C)+" ":CURSORX4*I-
1:CURSORY4*J-1:PRINTCHR$(C)+" "
160 A=KEY(0)
170 J=J-(A=10 AND J<3)+(A=9 AND J>1)
180 I=I-(A=7 AND I<3)+(A=8 AND I>1)
190 IF (C<>32 AND C<>127) OR A<>48 THEN
GOTO 120
200 CURSORX4*I-1:CURSORY4*J-2:ET1+3*K,3,
0:PRINTCHR$(65+K)+" ":CURSORX4*I-1:CURSO
RY4*J-1:PRINTCHR$(65+K)+" "
210 K=ABS(K-1)
220 IF K=1 THEN PLAY"ABC":GOTO 100
230 PLAY"AEC":GOTO 100
```

Horloge 2

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : simulation.
Difficulté du jeu : aucune.

LE JEU :
Ce programme simule une horloge digitale grand format en utilisant les caractères d'origine du VG 5000, notablement agrandis. Il présente certaines facilités de manipulation pour la mise à l'heure, et éventuellement l'arrêt ou le départ de l'horloge.

LE PROGRAMME :
Les lignes 30 à 120 dessinent cinq zones particulières, et représentent les symboles associés (défilement en avant par <, en arrière par > et arrêt : STOP).
La ligne 130 sépare les heures et les minutes. Les heures sont affichées par la ligne 110, et les minutes par la ligne 120. Les boucles J (150-220) et I (160-180) effectuent l'affichage clignotant des secondes.
La saisie des informations en provenance du clavier est faite en ligne 190.
Les informations sont traitées par les lignes 290, 310, 330, 350 et 370.
La ligne 250 contrôle le passage à 60 minutes.
La ligne 270 contrôle le passage à 24 heures.

POUR JOUER :
Faire RUN, l'heure est immédiatement affichée sous la forme 00 = 00 ; les secondes clignotent au milieu de l'écran.
Deux symboles < et >, sous les heures, permettent la mise à l'heure : il suffit d'appuyer sur les touches A, B, C, D qui leur correspondent pour faire défiler les heures en avant ou en arrière. De même pour les minutes.
On appuie sur la touche = pour arrêter l'horloge dans l'attente du prochain top sonore. La même touche = fera repartir l'horloge.
Faire STOP pour arrêter le programme. On pourra régler l'horloge en jouant sur la valeur de I, en ligne 160.

EXTENSIONS POSSIBLES :
On peut améliorer l'affichage des secondes, ajouter une possibilité de réveil, prévoir un agenda automatique, faire apparaître un coucou chantant au moment du passage de l'heure.

```
10 REM HORLOGE
20 H$="00":M$="00":INIT3,6
30 GR4,3,0:CURSORX9:CURSORY8:PRINT"WCCCC
CCCCCCCCCCCk";
40 FOR I=0 TO 4:CURSORX9:CURSORY9+I:PRIN
T"U";:CURSORX26:PRINT"j";:NEXT I
50 CURSORX9:CURSORY14:PRINT"uppppppppppp
pppppz";
60 TX5,4,0:FORI=0 TO 5:CURSORX9:CURSORY1
5+I:PRINT"                    ":NEXT I
70 TX2,3,0:CURSORX10:CURSORY17:PRINT"<
 > ":CURSORX10:CURSORY18:PRINT"<    > "
80 CURSORX20:CURSORY17:PRINT"<    > ":CUR
SORX20:CURSORY18:PRINT"<    > "
90 TX0,6,0:CURSORX10:CURSORY19:PRINT"A "
;:CURSORX14:PRINT"B ";:CURSORX20:PRINT"C
 ";:CURSORX24:PRINT"D "
100 TX0,7,0:CURSORX17:CURSORY17:PRINT"=
":CURSORX17:CURSORY18:PRINT"= ":TX1,3,0
110 CURSORX10:CURSORY10:PRINTLEFT$(H$,1)
+"   "+RIGHT$(H$,1)+" ":CURSORX10:CURSOR
Y11:PRINTLEFT$(H$,1)+"   "+RIGHT$(H$,1)+
" "
120 CURSORX20:CURSORY10:PRINTLEFT$(M$,1)
+"   "+RIGHT$(M$,1)+" ":CURSORX20:CURSOR
Y11:PRINTLEFT$(M$,1)+"   "+RIGHT$(M$,1)+
" "
130 M=VAL(M$):H=VAL(H$)
140 F=1
150 J=1
160 FOR I=1 TO 15
170 TX4,2,0:R$=":":CURSORX17:CURSORY11:
IF F=1 THEN R$="  "
180 PRINT R$:TX1,3,0:NEXT I
190 P=KEY(0)-60:IF P<>1 AND P<>37 AND P<
>38 AND P<>39 AND P<>40 THEN GOTO 210
200 P=P+35*(P>1):ON P GOTO 370, 290, 310
, 330, 350
210 F=ABS(F-1)
220 J=J+1:IF J<118 THEN GOTO 160
230 M=M+1
240 M$=STR$(M):M$="0"+RIGHT$(M$,LEN(M$)-
1):M$=RIGHT$(M$,2)
250 IF M=60 THEN M$="00":H=H+1
260 H$=STR$(H):H$="0"+RIGHT$(H$,LEN(H$)-
1):H$=RIGHT$(H$,2)
270 IF H=24 THEN H$="00"
280 GOTO 110
290 H=H+1:IF H>23 THEN H=0
300 GOTO 360
310 H=H-1:IF H<0 THEN H=23
320 GOTO 360
330 M=M+1:IF M>59 THEN M=0
340 GOTO 360
350 M=M-1:IF M<0 THEN M=59
360 GOTO 240
370 P=KEY(0):IF P=0 OR P<>61 THEN GOTO 3
70
380 GOTO 240
```

Anagrammes

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

LE JEU :
Des lettres sont choisies au hasard par le VG 5000, 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 la presser dès qu'ils voient s'afficher au bas de l'écran un mot déjà présent sur l'écran ; leur score est alors évalué, et un message correspondant à leur résultat sera délivré.

LE PROGRAMME :
Le niveau du jeu est entré à la ligne 5Ø. La boucle I (8Ø-1ØØ) réalise le choix des lettres.
Une deuxième boucle I (12Ø-19Ø) effectue le brouillage, et la ligne 14Ø permet de détecter la pression d'une touche.
La ligne 2ØØ affiche un nouveau mot, et fait monter d'une ligne sur l'écran les mots déjà obtenus.
La ligne 24Ø cherche quel est celui des deux joueurs qui a pressé une touche.
La boucle I (25Ø-32Ø) explore les mots présents sur l'écran ligne par ligne, alors que la boucle J (27Ø-3ØØ) saisit un par un les caractères de ces mots. On obtient ainsi un nouveau mot qui est comparé au mot initial en ligne 31Ø et le résultat BRAVO apparaît si c'est exact.
Sinon, la partie s'arrête sur l'affichage PERDU (ligne 33Ø).

POUR JOUER :
On lance le programme par RUN.
On devra ensuite entrer le niveau de difficulté (en principe de 1 à 3) et faire RET. 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, améliorer le son et la couleur, etc.

```
10 REM ANAGRAMMES
40 INIT7,7
50 INPUT"NIVEAU ( DE 1 A 3 ) :";N
60 IF N<1 OR N>3 THEN GOTO 40
70 S$=""
80 FOR I=1 TO N+3
90 S$=S$+CHR$(INT(RND(1)*26+65))
100 NEXT I
110 A$="":B$=S$
120 FOR I=1 TO LEN(S$)
130 N=INT(RND(1)*LEN(S$)+1)
140 D=KEY(0)
150 IF D=49 OR D=48 THEN GOTO 240
160 IF MID$(B$,N,1)="-" THEN GOTO 130
170 A$=A$+MID$(B$,N,1)
180 B$=LEFT$(B$,N-1)+"-"+RIGHT$(B$,LEN(B
$)-N)
190 NEXT I
200 TXINT(RND(1)*7):CURSORX1:CURSORY24:P
RINT A$
210 FOR I=1 TO 50*LEN(B$):NEXT I
220 S$=A$
230 GOTO 110
240 TX0:H$="JOUEUR DE DROITE":IF D=49 TH
EN H$="JOUEUR DE GAUCHE"
250 FOR I=0 TO 22
260 B$=""
270 FOR J=1 TO LEN(S$)
280 R$=CHR$(PEEK(16384+80*I+2*J))
290 B$=B$+R$
300 NEXT J
310 IF B$=S$ THEN CURSORXLEN(S$)+5:CURSO
RYI:PRINT"BRAVO , ";H$:PLAY"ACE":END
320 NEXT I
330 CURSORXLEN(S$)+5:CURSORY24:PRINT"PER
DU , ";H$:PLAY"AAA"
```

Œufs

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

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

LE PROGRAMME :
L'oiseau est dessiné en deux parties (pour le mouvement) aux lignes 7Ø et 12Ø ; son déplacement aléatoire est géré par le sous-programme 21Ø-3ØØ (en particulier les lignes 26Ø à 29Ø).
La ligne 13Ø permet de reboucler le programme sur la ligne 7Ø.
Le déplacement du joueur est régi par les lignes 15Ø et 16Ø, et le test permettant de détecter la présence d'un œuf est effectué en ligne 17Ø.
La ligne 19Ø permet l'affichage du score.
La fin de jeu est détectée en ligne 23Ø.

POUR JOUER :
On lancera le programme par RUN.
L'oiseau commence alors à se déplacer au hasard sur l'écran, et se met à pondre ses œufs. Lorsque le symbole qui représente le joueur (un carré de couleur) apparaît sur l'écran, celui-ci peut se déplacer en utilisant les 4 touches de déplacement. Il vous faut dès lors essayer de récupérer un maximum d'œufs avant la fin de la partie.
On pourra arrêter de jouer à tout moment en tapant STOP.

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

```
10 REM OEUFS
20 SETET65,"00003C7EFFFF7E3C0000"
30 INIT7,7
40 R=18:S=10
50 P=200:T=0
60 X=INT(RND(1)*36+1):Y=INT(RND(1)*21)
70 GR4,7,0:CURSORXX:CURSORYY:PRINT"dLX";
80 I=1
90 A=KEY(0):I=I+1:IF A=0 AND I<40 THEN G
OTO 90
100 IF A<>0 THEN GOSUB 140
110 GOSUB 210
120 CURSORXX:CURSORYY:GR4,7,0:PRINT"VCi"
;:FORI=1 TO 20:NEXT I
130 GOSUB 210:GOTO 70
140 CURSORXR:CURSORYS:TX1:PRINT" "
150 R=R-(A=7 AND R<36)+(A=8 AND R>1)
160 S=S-(A=10 AND S<21)+(A=9 AND S>0)
170 G=PEEK(16384+80*S+2*R)
180 T=T-(G=193)
190 CURSORX15:CURSORY22:TX0:PRINT"SCORE
:";T
200 RETURN
210 A=X:B=Y
220 P=P-1
230 IF P=0 THEN PLAY"ABC":TX0:PRINT"C'ES
T FINI...":END
240 CURSORXR:CURSORYS:TX1:PRINTCHR$(127)
250 CURSORXA:CURSORYB:PRINT" ";:ET2:PRIN
TCHR$(65);:TX1:PRINT" "
260 X=ABS(X+INT(1+2*(RND(1)<0.5)))
270 Y=ABS(Y+INT(1+2*(RND(1)<0.5)))
280 IF X>36 OR X<1 THEN X=1
290 IF Y>21 THEN Y=1
300 RETURN
```

Taquin de lettres

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

LE JEU :
Qui d'entre nous n'a pas, un jour ou l'autre, eu dans les mains un de ces petits puzzles où l'on essaie de déplacer d'une case à une autre des petits carrés de plastique colorés, jusqu'à l'obtention d'un résultat bien précis ? En son temps le 14-15 de SAM LOYD, a eu autant de succès que notre célèbre RUBIK'S.
Le taquin qui vous est proposé ici utilise 35 lettres, disposées au hasard dans un carré de 6 cases sur 6 cases, et ces lettres peuvent, replacées dans leur ordre initial, vous permettre de reconstituer un proverbe très connu. Une case coloriée vous permet de déplacer les lettres, et un compteur affiche en permanence le nombre d'essais : arriverez-vous à retrouver le proverbe ?

LE PROGRAMME :
La phrase à reconstituer (qui pourra bien sûr être modifiée) est placée dans une variable R$ en ligne 40.
Le mélange des lettres et l'impression sont faits par la boucle I (50-130).
La ligne 180 réalise l'affichage du carré coloré ; celui-ci est déplacé par les lignes 200 et 210 en utilisant les 4 flèches.
Les lignes 220 et 230 réimpriment la lettre concernée à l'emplacement du carré coloré.
La ligne 250 comptabilise le nombre d'essais qui est imprimé grâce à la ligne 150.

POUR JOUER :
On lance le programme par RUN.
La partie sera évidemment d'autant plus difficile que l'on ne connaît pas le proverbe à reconstituer. On peut déplacer les lettres et les faire passer à l'emplacement du carré coloré, à l'aide des 4 touches de direction (→, ←, ↓, ↑).
Le nombre d'essais est affiché en permanence, à droite du puzzle.
On peut arrêter à tout moment en tapant STOP.

EXTENSIONS POSSIBLES :
Le même principe peut être appliqué à toute sorte de taquins (manipulation de nombres, de couleurs, de sons éventuellement). On peut aussi prévoir plusieurs phrases, avec un choix effectué par un générateur aléatoire, etc.

```
10 REM TAQUIN DE LETTRES
20 INIT7,7:CLEAR 300
30 FOR I=1 TO 16:TX1,4,0:CURSORX10:CURSO
RYI:PRINT"                    ":NEXT I
40 A$="":R$="ONASOUVENTBESOIND'UNPLUSPET
ITQUESOI"
50 FOR I=1 TO 35
60 X=INT(RND(1)*35+1)
70 IF MID$(R$,X,1)="-" THEN GOTO 60
80 A$=A$+MID$(R$,X,1)
90 R$=LEFT$(R$,X-1)+"-"+RIGHT$(R$,LEN(R$
)-X)
100 F=ASC(MID$(A$,I,1))
110 TX4,7,0:X=I-6*INT(I/6):Y=INT(I/6)
120 CURSORX2*X+12:CURSORY2*Y+3:PRINTCHR$
(F)+" ":CURSORX2*X+12:CURSORY2*Y+4:PRINT
CHR$(F)+" "
130 NEXT I
140 X=0:Y=0:S=0:C=32
150 CURSORX1:CURSORY20:TX0:PRINT"NOMBRE
D'ESSAIS :";S
160 TX2,7,0:CURSORX2*X+12:CURSORY2*Y+3:P
RINT"  ":CURSORX2*X+12:CURSORY2*Y+4:PRIN
T"  "
170 T=KEY(0)
180 IF T=0 THEN CURSORX2*X+12:CURSORY2*Y
+3:PRINT"+ ":CURSORX2*X+12:CURSORY2*Y+4:
PRINT"+ ":GOTO 160
190 I=X:J=Y
200 X=X-(T=7 AND X<5)+(T=8 AND X>0)
210 Y=Y-(T=10 AND Y<5)+(T=9 AND Y>0)
220 C=PEEK(16648+160*Y+4*X)
230 TX4,7,0:CURSORX2*I+12:CURSORY2*J+3:P
RINTCHR$(C)+" ":CURSORX2*I+12:CURSORY2*J
+4:PRINT CHR$(C)+" "
240 IF I=X AND J=Y THEN PLAY"AAA":GOTO 1
50
250 S=S+1:PLAY"ACE"
260 GOTO 150
```

Cibles

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

LE JEU :
Il consiste simplement à tirer sur des cibles, en essayant de marquer le maximum de points ; le nombre de cibles (de 1 à 6) sera choisi en début de partie, et le score réalisé sera affiché en permanence. Si l'on touche le bord d'une cible, la partie s'arrête. On marquera d'autant plus de points que le nombre sur lequel on tombe sera plus grand.

LE PROGRAMME :
Le nombre de cibles est entré en ligne 40. La boucle I (80-100) effectue l'impression des cibles.
La ligne 120 affichera le score en permanence.
La balle est affichée par la ligne 130.
La boucle I (150-200) réalise l'impression des chiffres contenus dans les cibles, et permet le déplacement de la boule (ligne 190).
La ligne 220 regarde si l'on a tiré, et la ligne 230 regarde si la boule est encore sur la ligne de départ.
La ligne 240 fait progresser la boule d'une case vers la droite, et la ligne 250 vérifie si l'on a ou non atteint les cibles ; puis les lignes 260 et 270 regardent ce que l'on a touché.
La ligne 280 incrémente le score.
Les lignes 290 et 300 s'occupent de la fin de partie.

POUR JOUER :
Lancer le programme par RUN, et taper le nombre de cibles désiré suivi de RET. 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 0.
Le score sera modifié en fonction de votre résultat, et une nouvelle balle apparaît. Il est possible d'arrêter à tout moment en tapant STOP, 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 SETET65,"003C7EFFFFFFFF7E3C00"
30 INIT7,7
40 PRINT:INPUT"NOMBRE DE CIBLES :";N
50 IF N>6 THEN GOTO 30
60 P=10:J=30:S=0:B=1
70 INIT 2,2
80 FOR I=1 TO N
90 CURSORXJ:CURSORY3*I-1:GR4,2,0:PRINT"C
k";:CURSORXJ+1:CURSORY3*I:PRINT"j";:CURS
ORXJ:CURSORY3*I+1:PRINT"pz";
100 NEXT I
110 K=1:A=18
120 CURSORX1:CURSORY22:TX0:PRINT"SCORE :
";S;" - BALLE :";B
130 CURSORXK:CURSORYA:ET1:PRINTCHR$(65)
140 D=A
150 FOR I=1 TO N
160 U=INT(RND(1)*P)
170 CURSORXJ:CURSORY3*I:TX5:PRINT CHR$(U
+48)
180 M=KEY(0)
190 A=A-(M=10)+(M=9)
200 NEXT I
210 CURSORXK:CURSORYD:PRINT" "
220 IF M=48 THEN B=B+1:GOTO 240
230 IF K=1 THEN GOTO 130
240 K=K+1
250 IF K<J THEN GOTO 130
260 C=PEEK(16384+80*A+2*K)
270 IF C<48 OR C>57 THEN PRINT"PERDU..."
:PLAY"ECA":END
280 S=S+C-48:PLAY"ACE"
290 P=P-1
300 IF P=1 THEN PLAY"ABC":PRINT "C'EST F
INI":END
310 GOTO 110
```

Gouttes

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

LE JEU :
C'est un véritable programme de dessin animé que l'on peut considérer, en fait, comme un interlude ; un récipient est dessiné sur l'écran, vu en coupe, et il contient déjà des obstacles. Une goutte d'eau, symbolisée par un rond coloré, apparaît au-dessus du récipient et tombe en remplissant le récipient au fur et à mesure. Sa trajectoire, lorsqu'elle rencontre un obstacle, est entièrement déterminée par le hasard : elle part à droite ou à gauche, jusqu'à ce qu'il n'y ait plus aucune possibilité. Alors, une autre goutte tombe, et ainsi de suite, jusqu'au débordement final.

LE PROGRAMME :
Les deux boucles emboîtées I (60-130) et J (70-120) dessinent le récipient et les obstacles.
La ligne 170 dessine la goutte et la ligne 210 envoie dans un sous-programme (370-390) qui permet de déterminer s'il y a ou non rencontre avec un obstacle.
Si c'est non, on va en 290, et la goutte continue de tomber.
Si c'est oui, la goutte part à droite ou à gauche (ligne 230) selon la valeur définie pour K en ligne 150.
On regarde de nouveau s'il y a ou non obstacle, ou si la goutte peut continuer à tomber (lignes 240, 250 ou 260).
Selon le résultat du test, on utilisera 270, 290, 310 ou 340.

POUR JOUER :
Le RUN déclenche le dessin du récipient et des obstacles, la goutte apparaît, se met à tomber et finit par se trouver une place. Une autre goutte apparaît et l'on recommence.
Le programme s'arrête de lui-même lorsque le récipient déborde ; on peut l'arrêter à tout moment en tapant STOP.

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
40 SETET65,"003C7EFFFFFFFF7E3C00"
50 INIT7,7
60 FOR I=12 TO 20
70 FOR J=1 TO 16
80 CURSORX1:CURSORYI:TX2:PRINTCHR$(127);
:CURSORX18:PRINTCHR$(127)
90 CURSORXJ+1:CURSORY20:PRINTCHR$(127)
100 IF I<>12 AND I<>20 THEN TX5:CURSORXJ
+1:CURSORYI:PRINTCHR$(127)
110 IF I<>20 AND RND(1)<0.7 THEN CURSORX
INT(RND(1)*15+2):CURSORYI:PRINT"  "
120 NEXT J
130 NEXT I
140 Y=INT(RND(1)*7+6):X=6
150 K=-2*(RND(1)<0.5)-1
160 I=0
170 CURSORXY:CURSORYX:ET4:PRINTCHR$(65)
180 FOR J=0 TO 100:NEXT J
190 CURSORXY:CURSORYX:TX0:PRINT" "
200 U=X+1:V=Y
210 GOSUB 370
220 IF J=0 THEN GOTO 290
230 U=X:V=Y+K
240 GOSUB 370
250 IF J=0 THEN GOTO 310
260 IF I=0 THEN GOTO 340
270 CURSORXY:CURSORYX:ET4:PRINTCHR$(65)
280 GOTO 140
290 X=X+1
300 GOTO 150
310 Y=Y+K
320 I=1
330 GOTO 170
340 K=-K
350 I=1
360 GOTO 230
370 C=PEEK(16384+80*U+2*V):SOUND230,2,3
380 J=-(C<>32):IF U=24 THEN CURSORX20:CU
RSORY15:TX0:PRINT"C'EST FINI...":PLAY"AE
C":END
390 RETURN
```

Obstacles

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

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

LE PROGRAMME :
La ligne 8Ø fait avancer les piétons vers la voiture du joueur qui est imprimée grâce à la ligne 7Ø.
Les lignes 8Ø, 9Ø, 1ØØ fabriquent la route (les lignes 9Ø et 1ØØ 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 12Ø affiche le score.
La ligne 14Ø permettra de déplacer le véhicule.
Les lignes 15Ø et 16Ø permettent de contrôler s'il y a ou non un obstacle et arrêtent le programme s'il y a lieu.
Le score est incrémenté en ligne 17Ø.

POUR JOUER :
On lance le programme en tapant RUN.
La route et les piétons apparaissent en bas de l'écran et montent vers la voiture. La voiture peut se déplacer à l'aide des flèches de déplacement horizontal (touches → et ←).
La rencontre avec un piéton ou la bordure interrompt le déroulement du programme et provoque l'affichage du score.
On peut, cependant, sortir du programme à tout moment en utilisant les touches STOP.

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

```
10 REM OBSTACLES
20 SETET65,"99FFA5243C3C185A7E5A"
30 SETET66,"18241818187C9A19242466"
50 INIT7,7
60 S=0:Y=5
70 CURSORXY:CURSORY0:ET1:PRINTCHR$(65)
80 CURSORX1:CURSORY24:TX4:PRINTCHR$(127)
+"          "+CHR$(127);
90 IF RND(1)<0.7 THEN CURSORXINT(RND(1)*
8+2):ET2:PRINTCHR$(66);
100 IF RND(1)<0.8 THEN CURSORXINT(RND(1)
*8+2):ET5:PRINTCHR$(66)
110 CURSORX15:CURSORY5:TX0:PRINT"
    "
120 CURSORX15:CURSORY6:PRINT"SCORE :";S
130 A=KEY(0)
140 Y=Y-(A=7 AND Y<10)+(A=8 AND Y>1)
150 C=PEEK(16384+2*Y)
160 IF C=127 OR C=194 THEN PLAY"ABC":END
170 S=S+1
180 GOTO 70
```

Captures dans l'espace

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

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

LE PROGRAMME :
La ligne 7Ø permet de faire apparaître les météorites.
La ligne 9Ø fera apparaître le minerai d'urane. La ligne 1ØØ réalise une légère temporisation et assure le déplacement sur l'écran.
Les lignes 11Ø à 13Ø 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 12Ø).
Votre vaisseau spatial est imprimé par les lignes 14Ø et 15Ø et son déplacement est contrôlé par la ligne 17Ø.

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 STOP.

EXTENSIONS POSSIBLES :
On peut rendre le jeu de plus en plus difficile en augmentant progressivement le nombre de météorites, ou encore, en minutant la durée de la partie.
On peut aussi prévoir un affichage permanent du score, introduire un deuxième vaisseau spatial manœuvré par une puissance étrangère et qui cherche aussi à s'emparer de l'urane...

```
10 REM CAPTURES DANS L'ESPACE
20 SETET65,"0000001824E7BD240000"
30 SETET66,"000099A5FF7E3C3C4200"
40 INIT0,0
50 X=15:Z=0
60 S=INT(RND(1)*38+1)
70 CURSORXS:CURSORY24:TX6,4,0:PRINT"*";
80 S=INT(RND(1)*38+1)
90 IF RND(1)<0.7 THEN CURSORXS:CURSORY24
:ET1:PRINTCHR$(66);
100 FOR I=1 TO 15:NEXT I:TX0:PRINT
110 C=PEEK(17344+2*X)
120 IF C=194 THEN TX7:PRINT"SCORE OBTENU
 :";Z:PLAY"ACE":END
130 IF CHR$(C)="*" THEN Z=Z+1:PLAY"ABC"
140 CURSORXX-1:CURSORY11:TX0:PRINT"   "
150 CURSORXX-1:CURSORY12:PRINT" ";:ET2:P
RINTCHR$(65);:TX0:PRINT" "
160 A=KEY(0)
170 X=X-(A=7 AND X<39)+(A=8 AND X>1)
180 GOTO 60
```

Anomalie

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

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

LE PROGRAMME :
La boucle I (50-80) imprime les chiffres de repérage des grilles.
La boucle principale I (100-160) imprime les symboles des deux grilles, dans un ordre différent.
La ligne 110 choisit le caractère à imprimer.
Le sous-programme (1000-1040) s'occupe de faire l'impression à un emplacement libre (on l'appelle deux fois pour les deux grilles).
Les lignes 170 et 180 changent, de façon aléatoire, l'un des caractères.
Les coordonnées du point essayé sont entrées aux lignes 190 et 200, et le contrôle est effectué par les lignes 210 et 220.
Les lignes 230 et 240 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 RET, et son numéro de colonne, suivi de RET.
Le programme répondra alors BRAVO ou PERDU.

EXTENSIONS POSSIBLES :
Il sera possible d'utiliser des symboles graphiques de diverses couleurs, ce qui pourra simplifier la recherche.
On peut aussi envisager d'utiliser des dessins plus sophistiqués, améliorer le contrôle du résultat, admettre la bonne réponse sur la grille de droite ou celle de gauche, introduire un minutage, etc.

```
10 REM ANOMALIE
20 INIT6,6
40 INIT6,6:T=5
50 FOR I=1 TO 5
60 CURSORX2*I+1:CURSORY0:TX1,4,0:PRINTI:
CURSORX1:CURSORY2*I:PRINTI
70 CURSORX2*I+19:CURSORY0:PRINTI:CURSORX
19:CURSORY2*I:PRINTI
80 NEXT I
90 TX6
100 FOR I=1 TO 22
110 D=INT(26*RND(1)+1)
120 L=2:S=T
130 GOSUB 1000
140 L=11
150 GOSUB 1000
160 NEXT I
170 S=0:FOR J=1 TO 5:FORI=1 TO 5:POKE163
89+4*I+160*J,5:POKE16425+4*I+160*J,3:NEX
T I,J
180 TX3:GOSUB 1000
190 CURSORX1:CURSORY16:TX0:INPUT"LIGNE..
...:";I
200 PRINT:INPUT"COLONNE...:";T
210 U=2*I:V=2*T+20
220 C=PEEK(16384+80*U+2*V)
230 IF D+64=C THEN PRINT"BRAVO...":PLAY"
ACE":END
240 IF D+64<>C THEN PRINT"PERDU ; LA LET
TRE A TROUVER ETAIT ";:TX4:PRINTCHR$(64+
D):PLAY"AAA":END
1000 U=2*INT(RND(1)*T+1):V=2*INT(RND(1)*
S+L)
1010 C=PEEK(16384+80*U+2*V)
1020 IF C<>32 AND S<>0 THEN GOTO 1000
1030 CURSORXV:CURSORYU:PRINT CHR$(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 possibles avec ce programme qui va se comporter comme un crayon électronique vous permettant d'inscrire des symboles O ou X, en n'importe quel emplacement de votre feuille-écran de télévision. On pourra, par exemple, l'utiliser pour un jeu de morpion (le premier à réaliser un alignement de 5 pions identiques gagne) ou pour un GO (on réalise des encadrements des pions adverses) ou tout autre jeu, selon son imagination.

LE PROGRAMME :
Le terrain de jeu est délimité par un cadre 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 13Ø et 14Ø contrôlent si le déplacement est possible (case libre et non dépassement des limites).
La ligne 19Ø permet d'écrire sur une case libre.

POUR JOUER :
RUN provoque l'affichage des limites du terrain et du numéro du joueur, ainsi que des symboles correspondants (O ou X) que l'on devra déplacer sur l'écran à l'aide des 4 touches de direction (→, ←, ↓, ↑).
Il n'est pas possible de venir sur une case déjà occupée et il faudra appuyer sur la touche Ø pour inscrire son propre symbole dans une case disponible.
Dès que cela a été fait, le symbole du second joueur apparaît dans la colonne de gauche, à peu près au milieu, et le joueur procédera de la même manière que son adversaire.
Pour sortir du programme, il suffira de taper STOP.

EXTENSIONS POSSIBLES :
On pourra utiliser l'écran tout entier comme terrain de jeu et réaliser des contrôles en fonction des différents jeux que l'on désire pratiquer : contrôle d'alignements de 5 pièces pour le morpion, contrôle du nombre de pièces capturées pour le GO.

```
10 REM PLUSIEURS EN UN
20 INIT6,6
30 TX3,4,0:FORI=0 TO 22 : PRINTCHR$(127)
+"                              ":NEXT I
40 J=0
50 J=ABS(J-1)
60 X=12:Y=1
70 CURSORXY:CURSORYX:TX1+3*J,4,0:PRINTCH
R$(79+9*J)
80 K=KEY(0):IF K=0 THEN GOTO 80
90 A=X:B=Y
100 X=X-(K=10 AND X<22)+(K=9 AND X>0)
110 Y=Y-(K=7 AND Y<28)+(K=8 AND Y>2)
120 CURSORXB:CURSORYA:TX3,-4*(B>1),0:PRI
NT" "
130 C=PEEK(16384+80*X+2*Y)
140 IF C<>79 AND C<>88 THEN GOTO 160
150 X=A:Y=B
160 I=1
170 I=I+1:K=KEY(0):IF K=0 AND I<15 THEN
GOTO 170
180 IF K<>48 OR Y=1 THEN GOTO 70
190 CURSORXY:CURSORYX:TX1+3*J:PRINTCHR$(
79+9*J)
200 SOUND150+70*J,4,1
210 GOTO 50
```

Parcours du cavalier

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

LE JEU :
Il se présente, ici, comme un jeu de hasard total, mais provient, en réalité, d'un problème assez difficile à résoudre : comment déplacer un cavalier sur un échiquier de façon qu'il passe une fois, et une seule, sur chacune des 64 cases du damier.
Le cavalier se déplace suivant les règles communément admises aux échecs, et sa position de départ est définie par le joueur ; par contre, ses déplacements sont effectués au hasard. Arrivera-t-il à parcourir les 64 cases ?

LE PROGRAMME :
La ligne 60 définit les divers déplacements possibles, dans deux variables alphanumériques A$ et B$; la boucle emboîtée I (70-100) dessine le terrain de jeu.
On entre les coordonnées de la case de départ (ligne, puis colonne) aux lignes 110 et 120.
La ligne 140 imprime le numéro du coup.
La ligne 150 permet le tirage au hasard d'un déplacement du cavalier.
La boucle F (160-210) cherche si le déplacement est possible ; si ce n'est pas le cas, on essaie tous les autres (toujours dans le même ordre).
Les lignes 170 et 180 effectuent le déplacement.
La ligne 190 contrôle que l'on ne sorte pas du terrain de jeu.
La ligne 200 envoie en ligne 230 si le coup est possible.
La ligne 220 gère la fin de partie (cavalier bloqué) et affiche le score.
Les lignes 230, 240 et 250 réalisent le coup et incrémentent le score.

POUR JOUER :
On lance le programme par l'ordre RUN. Le terrain de jeu s'affiche ; on choisit alors la position de départ en tapant d'abord un numéro de ligne (entre 1 et 8), suivi de RET puis son numéro de colonne, suivi de RET. 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 STOP.

EXTENSIONS POSSIBLES :
On peut imaginer de nombreuses variantes sur ce thème, ne serait-ce qu'en commençant par faire varier la taille des échiquiers, on peut aussi améliorer considérablement le graphisme. Mais, le plus intéressant est d'arriver à définir un algorithme donnant la ou les solutions du problème initial...

```
10 REM PARCOURS DU CAVALIER
20 CLEAR 200
30 INIT6,6
40 S=1
50 T=0
60 A$="56653223":B$="653223"+A$+"65":A$=
A$+A$
70 FOR I=1 TO 8
80 CURSORX3*I+1:CURSORY0:TX4,4,0:PRINTI
90 CURSORX1:CURSORY2*I:PRINTI;:TX0:PRINT
"  .  .  .  .  .  .  .  ."
100 NEXT I
110 CURSORX1:CURSORY18:INPUT"LIGNE DE DE
PART :";I
120 PRINT:INPUT"COLONNE DE DEPART :";J
130 IF S>9 THEN S=0:T=T+1
140 CURSORX3*J+2:CURSORY2*I:TX1,4,0:PRIN
TRIGHT$(STR$(S),1);:IF T<>0 THEN CURSORX
3*J+1:PRINTRIGHT$(STR$(T),1)
150 Z=INT(RND(1)*8+1)
160 F=Z
170 A=I+VAL(MID$(A$,F,1))-4
180 B=J+VAL(MID$(B$,F,1))-4
190 IF A<1 OR B<1 OR A>8 OR B>8 THEN GOT
O 210
200 IF PEEK(16388+160*A+6*B)=46 THEN GOT
O 230
210 F=F+1:IF F<=Z+8 THEN GOTO 170
220 CURSORX25:CURSORY18:PRINT"TOTAL :";T
*10+S:PLAY"ACE":END
230 I=A:J=B
240 S=S+1:PLAY"AE"
250 GOTO 130
```

Mur de briques

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

LE JEU :
S'il est un jeu que l'on peut présenter comme un grand classique des jeux d'arcades pour micro-ordinateur, c'est bien le mur de briques (en anglais BREAKOUT).
Il est offert, ici, dans une version simplifiée, mais offre néanmoins de bonnes performances : affichage du score en fin de partie, rebonds aléatoires, etc.
La règle en est très élémentaire : vous disposez d'une raquette qui peut être déplacée horizontalement pour renvoyer une balle (matérialisée par un point) qui va démolir les briques dessinées en haut de l'écran.
Votre problème est d'en démolir le plus grand nombre possible, sans rater la balle, ce qui provoquerait la fin de partie et l'affichage du score.

LE PROGRAMME :
La boucle I (50-80) dessine le terrain de jeu et les briques sont mises en place par les boucles emboîtées I (90-130) et J (100-120).
La ligne 150 dessine la raquette dont la position est contrôlée par la ligne 170.
Les lignes 220 à 250 regardent si l'on a touché le mur, les briques ou la raquette.
Les rebonds sont gérés en conséquence avec des éventuels aléas, par les lignes 260 et 300.
La ligne 190 affiche le score en permanence.

POUR JOUER :
Le programme sera lancé par l'instruction RUN, ce qui provoque l'affichage du terrain de jeu puis, très rapidement, l'envoi de la balle. Le joueur devra alors essayer de déplacer sa raquette à l'aide des touches de déplacement horizontal (→ et ←).
Si, à un moment quelconque, sa balle ne touche pas la raquette, le score sera affiché et la partie s'arrête.
On peut aussi terminer à tout moment par STOP.

EXTENSIONS POSSIBLES :
On peut envisager d'agrandir le terrain de jeu, d'augmenter le nombre de briques, de prévoir plusieurs balles, d'introduire des effets sonores.
On peut aussi, en s'inspirant des jeux d'arcades, introduire de petits ouvriers qui viendront reconstruire votre mur de briques, au fur et à mesure que vous le démolissez.

```
10 REM MUR DE BRIQUES
20 A$=CHR$(127)
40 INIT7,7
50 FOR I=1 TO 18
60 CURSORX9+I:CURSORY0:PRINT A$
70 CURSORX10:CURSORYI:PRINTA$;:CURSORX27
:PRINTA$
80 NEXT I
90 FOR I=1 TO 7
100 FOR J=11 TO 26
110 CURSORXJ:CURSORYI:TXI-1,4,0:PRINT"#"
120 NEXT J
130 NEXT I
140 H=-1:K=-0.5:J=18:I=17:Y=J
150 CURSORXY-2:CURSORY19:TX0:PRINT" "+CH
R$(96)+CHR$(96)+CHR$(96)+" "
160 A=KEY(0)
170 Y=Y-(A=7 AND Y<28)+(A=8 AND Y>9)
180 IF T=1 THEN CURSORXN:CURSORYM:PRINT"
 "
190 CURSORX1:CURSORY20:PRINT"SCORE ";S
200 IF I>22 THEN PLAY"AAA":END
210 I=I+H:J=J+K
220 M=INT(I):N=INT(J):C=PEEK(16384+80*M+
2*N)
230 IF C=32 THEN CURSORXN:CURSORYM:PRINT
"O":T=1:GOTO 150
240 IF C=127 THEN GOTO 280
250 IF C=35 THEN F=71-PEEK(16385+80*M+2*
N):CURSORXN:CURSORYM:PRINT"O":T=1:GOTO 3
00
260 H=-H
270 K=RND(1)-RND(1)
280 K=-K:T=0:SOUND150,1,5
290 GOTO 150
300 S=S+F:SOUND250,2,1
310 K=-(K-(2*(RND(1)<0.5)+1)*RND(1)/3)
320 H=-H
330 GOTO 150
```

Othello

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

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

LE PROGRAMME :
Les boucles emboîtées P (5Ø-12Ø) et Q (7Ø-11Ø) dessinent le terrain de jeu et la ligne 14Ø affiche un X ou un O, selon le joueur qui commence.
La boucle P (2ØØ-32Ø) contrôle la validité du coup (la case est-elle libre : lignes 3ØØ et 31Ø), en utilisant une partie de programme qui se trouve aux lignes 36Ø à 42Ø (a-t-on pris un ou plusieurs ennemis en ligne 36Ø et retournement éventuel autorisé par la ligne 38Ø : on refait un balayage dans toutes les directions, si l'encadrement s'est révélé possible).
La ligne 2ØØ définit les 8 directions d'exploration.
Les lignes 24Ø et 25Ø calculent les déplacements à effectuer pour explorer ces 8 directions.
La ligne 23Ø réalise les déplacements.
La ligne 34Ø assure le changement de joueur, s'il y a eu retournement (contrôlé en ligne 33Ø).

POUR JOUER :
Après RUN, chaque joueur entrera à son tour les coordonnées (ligne, RET, puis colonne, RET) 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 le VG 5000.
Après avoir lancé le programme par RUN, il convient d'entrer la dimension de la grille (4, RET ou 6, 8 ...).

```
10 REM OTHELLO
20 INIT7,7:PRINT:INPUT"DIMENSION DE LA G
RILLE :";A
30 IF A/2<>INT(A/2) THEN GOTO 20
40 INIT7,7
50 FOR P=1 TO A
60 TX5,4,0:CURSORX2*P:CURSORY0:PRINT RIG
HT$(STR$(P),1)
70 FOR Q=1 TO A
80 CURSORX1:CURSORY2*P:TX5,4,0:PRINT RIG
HT$(STR$(P),1);:CURSORX2*Q:TX0:PRINT"."
90 IF Q=A/2+1 AND P=A/2 THEN CURSORX2*Q-
2:CURSORY2*P:TX1:PRINT"O ";:TX4:PRINT"X"
100 IF Q=A/2+1 AND P=A/2+1 THEN CURSORX2
*Q-2:CURSORY2*P:TX4:PRINT"X ";:TX1:PRINT
"O"
110 NEXT Q
120 NEXT P
130 K=1
140 CURSORX23:CURSORY6:TX0:PRINT"JOUEUR
";:TX1+3*K,4,0:PRINTCHR$(79+9*K):SOUND15
0+60*K,3,7
150 CURSORX31:CURSORY8:TX0:PRINT"      "
:CURSORX31:CURSORY10:PRINT"      "
160 CURSORX23:CURSORY8:INPUT"LIGNE  :";U
170 IF U=0 THEN GOTO 340
180 CURSORX23:CURSORY10:INPUT"COLONNE:";
V
190 H=1
200 FOR P=1 TO 8
210 S=0
220 W=0
230 T=2*U:R=2*V
240 F=(P>3 AND P<7)-(P<3 OR P>7)
250 G=(P>1 AND P<5)-(P>=6)
260 Q=46-42*(S=1)-33*(S=2)
270 IF S=0 THEN GOTO 290
280 CURSORXR:CURSORYT:TX1+3*K:PRINT CHR$
(Q)
290 T=T+2*F:R=R+2*G
300 C=PEEK(16384+80*T+2*R)
310 IF C=79 OR C=88 THEN GOTO 360
320 NEXT P
330 IF H<>0 THEN GOTO 140
340 K=ABS(K-1)
350 GOTO 140
360 IF (C=79 AND K=1) OR (C=88 AND K=0)
THEN W=1
370 IF S<>0 THEN CURSORXR:CURSORYT:TX1+3
*K:PRINT CHR$(Q)
380 IF W=1 AND ((C=79 AND K=1) OR (C=88
AND K=0)) THEN GOTO 290
390 IF W=0 OR S<>0 THEN GOTO 320
400 S=1-(K=0)
410 H=-H*(S=0)
420 GOTO 220
```

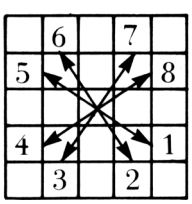
Quatre à quatre

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

LE JEU :
Il pourra être pratiqué à un ou plusieurs joueurs. Dans les deux cas, le but est d'arriver à échanger les pions X avec les pions O ; si l'on joue en solitaire, cet échange devra être réalisé avec le moins de coups possible. Les pions sont déplacés alternativement en respectant les règles de déplacement du cavalier aux échecs ; il n'y a pas de prise, et on ne peut passer son tour.

LE PROGRAMME :
La variable alphanumérique A$, en ligne 40, contient la table des coups possibles (déplacement du cavalier).
La boucle I (60-100) dessine le terrain de jeu, et les positions de départ.
La ligne 120 permet d'afficher O ou X, selon le joueur qui a le trait.
La ligne 150 permet l'affichage du curseur clignotant (ici un pavé de couleur) qui indique sur quelle case on se trouve.
Les lignes 180 et 190 permettent le déplacement du curseur.
La ligne 200 regarde si l'on a appuyé sur une touche de 1 à 8, donnant la direction du déplacement.
La ligne 210 réimprime la figure qui était sous le curseur. La ligne 240 transforme en nombre le symbole chiffre frappé. La ligne 230 réalise le contrôle de départ.
Les lignes 250 et 260 effectuent le déplacement. La ligne 290 réalise le contrôle d'arrivée. Le résultat est imprimé par la ligne 300, le changement de joueur est réalisé par la ligne 310 et un "." est imprimé à l'emplacement de départ. Le sous-programme 330-340 est utilisé pour les divers contrôles.

POUR JOUER :
On lance le programme par RUN. Le terrain de jeu est affiché, ainsi que le symbole permettant d'identifier le joueur qui a le trait. Le joueur peut choisir le pion qu'il désire déplacer en manœuvrant le curseur clignotant grâce aux touches de déplacement du curseur.
Pour déplacer son pion, il suffit, lorsque le curseur est sur ce pion, de taper la direction du déplacement, suivant le code :



Le coup sera refusé si la case de départ n'est pas conforme ou si la case d'arrivée est occupée. Sinon, le coup est réalisé, et le trait passe à l'autre joueur, qui déplace le curseur clignotant pour l'amener sur l'un des pions, et joue de la même façon.
Le joueur gagnant est celui qui réussit à amener le premier tous ses pions dans les cases initialement occupées par l'adversaire. On peut interrompre la partie à tout moment en tapant STOP.

EXTENSIONS POSSIBLES :
On peut améliorer les contrôles, prévoir un test de fin, utiliser un terrain de jeu différent ; on pourra aussi envisager l'écriture d'un programme permettant au VG 5000 de participer au jeu avec de bonnes chances de l'emporter.

```
10 REM QUATRE A QUATRE
20 INIT6,6
30 FOR I=1 TO 19:TX3,4,0:PRINT"
              ":NEXT I
40 H=1:A$="5665322356"
50 TX0,3,0:FOR I=1 TO 5:CURSORX4*I:CURSO
RY1:PRINT CHR$(48+I)+" ":CURSORX4*I:CURS
ORY2:PRINT CHR$(48+I)+" ":NEXT I
60 FOR I=1 TO 5:CURSORX1:CURSORY3*I+1:PR
INTCHR$(48+I)+" ":CURSORX1:CURSORY3*I+2:
PRINT CHR$(48+I)+" "
70 FOR J=1 TO 5:CURSORX4*J:CURSORY3*I+1:
PRINT". ":CURSORX4*J:CURSORY3*I+2:PRINT"
. "
80 IF I<3 AND J<3 THEN CURSORX4*J:CURSOR
Y3*I+1:PRINT"O ":CURSORX4*J:CURSORY3*I+2
:PRINT"O "
90 IF I>3 AND J>3 THEN CURSORX4*J:CURSOR
Y3*I+1:PRINT"X ":CURSORX4*J:CURSORY3*I+2
:PRINT"X "
100 NEXT J,I
110 X=1:Y=1
120 R$="J O U E U R   X ":IF H=1 THEN R$
="J O U E U R   O "
130 CURSORX1:CURSORY21:PRINT R$:CURSORX1
:CURSORY22:PRINT R$:GOSUB 330
140 I=J
150 TX5,7,0:CURSORX4*X:CURSORY3*Y+1:PRIN
T"  ":CURSORX4*X:CURSORY3*Y+2:PRINT"  ":
TX0,3,0
```

```
160 A=X:B=Y
170 R=KEY(0)
180 Y=Y-(R=10 AND Y<5)+(R=9 AND Y>1)
190 X=X-(R=7 AND X<5)+(R=8 AND X>1)
200 IF R<57 AND R>48 THEN GOTO 230
210 R$=CHR$(46-33*(J=79)-42*(J=88))+" ":
CURSORX4*A:CURSORY3*B+1:PRINTR$:CURSORX4
*A:CURSORY3*B+2:PRINT R$
220 GOTO 120
230 IF (I=46) OR (I=88 AND H=1) OR (I=79
 AND H=0) THEN GOTO 210
240 J=R-48
250 S=Y:Y=Y+VAL(MID$(A$,J,1))-4
260 T=X:X=X+VAL(MID$(A$,J+2,1))-4
270 IF X<1 OR X>5 OR Y<1 OR Y>5 THEN Y=S
:X=T:A=T:B=S:J=88+9*(H=1):GOTO 210
280 GOSUB 330
290 IF J<> 46 THEN GOTO 320
300 R$=CHR$(88+9*(H=1)):CURSORX4*X:CURSO
RY3*Y+1:PRINT R$:CURSORX4*X:CURSORY3*Y+2
:PRINT R$
310 H=ABS(H-1):I=1:PLAY"ABC"
320 GOTO 210
330 J=PEEK(16544+240*Y+8*X)
340 RETURN
```

Achevé d'imprimer en octobre 1984
sur les presses de l'imprimerie Laballery et Cie - 58500 Clamecy
Dépôt légal : octobre 1984 — N° d'impression : 410030 — N° d'édition : 86595-193-1
ISBN : 2-86595-193-6

Enseigner et distraire est le double objectif de cet ouvrage. Ces 102 programmes de jeux vous entraînent à la découverte du VG 5000.

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.

Le dernier niveau constitue une illustration claire de l'utilisation du Basic étendu.

DES LIVRES POUR VOTRE ORDINATEUR

P.S.I.

ISBN : 2.86595.193.6
110 FF

Couverture GIBRAT

102 PROGRAMMES POUR VG 5000

PAR JACQUES DECONCHAT

P.S.I.