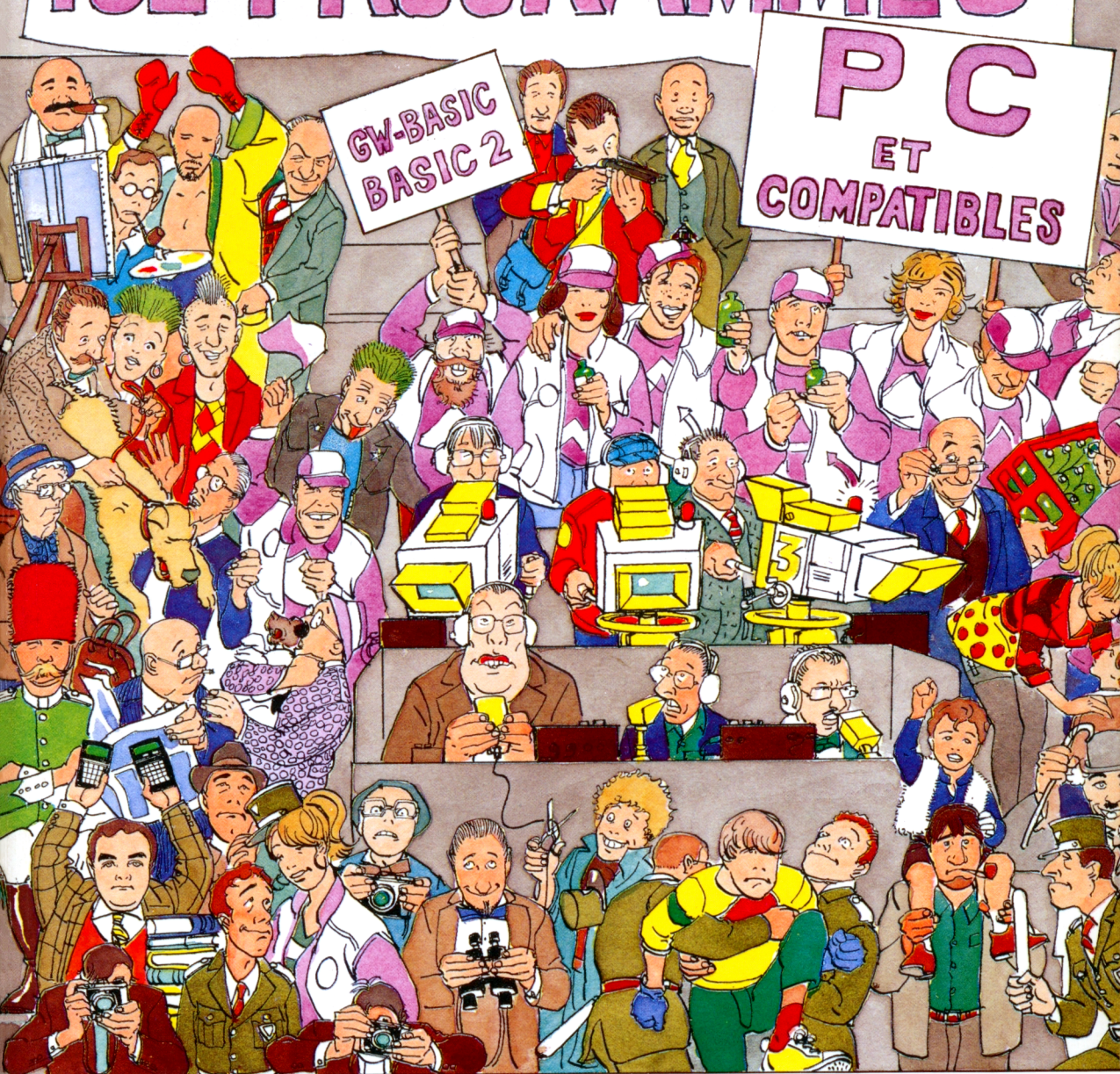




102 PROGRAMMES



PSi.

PAR
JACQUES
DECONCHAT

PSi.

102 PROGRAMMES POUR PC ET COMPATIBLES

CONNAISSEZ-VOUS LA COLLECTION IBM/PC ET COMPATIBLES CHEZ P.S.I. ?

Systèmes MS/DOS et PC/DOS

- Clefs pour MS-DOS, versions 2 à 4 - Yvon Dargery
- MS-DOS pas à pas, versions 2 à 4 - Alain Pinaud
- Le livre de MS/PC-DOS pour PC et AT - François Piette
- Systèmes PC-DOS et MS-DOS, versions 2 à 4 - Jacques Boyer, Jean-Pierre Lamoitier et Michel Treillet

Logiciels professionnels

- Clefs pour Multiplan 3 - Jean-Louis Marx et Alain Thibault
- Multiplan 3 par l'exemple - Jean-Michel Jégo et Alain Gargadennec
- Multiplan Junior par l'exemple - Alain Gargadennec
- Le livre de Framework - René Cohen
- Le livre de Lotus 1.2.3., version 2 - Jean-Pierre Richard
- Introduction à l'analyse financière sur Lotus 1.2.3., version 2 - Eric Baumarti
- Lotus 1.2.3. par l'exemple - Jean-Michel Jégo et Alain Gargadennec
- Clefs pour Lotus 1.2.3., version 2 - Jean-Louis Marx et Alain Thibault
- La pratique de Lotus 1.2.3., et ses macros - Jacques Boisgontier
- Le livre de Word 3 sur PC et compatibles - Pierre Brandeis et Augustin Garcia
- Clefs pour Textor - Claire Desaint
- Programmer en dBASE III et III Plus - René Cohen
- dBASE et ses fichiers III et III Plus - Jacques Boisgontier
- dBASE III Plus en mode direct - René Cohen

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

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

© *Éditions P.S.I. est une filiale de Nathan Éditeur*

GROUPE
DE
LA CITE

6-10, boulevard Jourdan
75014 PARIS
1989
ISBN 2-86595-458-7

**JACQUES
DECONCHAT**

**102 PROGRAMMES POUR
PC ET COMPATIBLES**



EDITIONS P.S.I.
1989

AVERTISSEMENT

Les programmes contenus dans cet ouvrage tournent sur les PC et compatibles équipés de cartes graphiques EGA et CGA.

Ils ne tournent pas sur les compatibles équipés de cartes Hercules comme :

- Amstrad PC 1640
- Thomson TO 16

Attention ! les programmes en Basic2 tournent sur Amstrad PC 1512 mais ne tournent pas sur Amstrad PC 1640 suite à des modifications internes effectuées par Amstrad sur ces machines.

Microsoft Basica, GW-Basic et MS-DOS sont des marques enregistrées de Microsoft Corporation.

IBM est une marque enregistrée d'International Business Machines Corporation.

Amstrad PC est une marque enregistrée d'Amstrad Consumer Electronics.

Basic2 est une marque enregistrée de Locomotive Software Ltd.

Microsoft Basica, GW-Basic et MS-DOS sont des produits de la société Microsoft Corporation, IBM est un produit de la société International Business Machines Corporation, Amstrad PC est un produit de la société Amstrad Consumer Electronics et Basic2 est un produit de la société Locomotive Software Ltd qui en détiennent le copyright à l'échelon mondial.

Ce livre n'est pas le manuel des logiciels Microsoft Basica, GW-Basic, MS-DOS, Basic2 et des matériels IBM, Amstrad PC et son contenu n'engage pas les sociétés Microsoft Corporation, International Business Machines Corporation, Amstrad Consumer Electronics, et Locomotive Software Ltd.

Nous vous rappelons, par ailleurs, les termes de l'article 47 de la loi du 3 juillet 1985 :

"Toute reproduction autre que l'établissement d'une copie de sauvegarde par l'utilisateur, ainsi que toute utilisation d'un logiciel non expressément autorisée par l'auteur ou ses ayants droit, est passible des sanctions prévues par la loi."

SOMMAIRE

Avant-propos	8
Organisation de l'ouvrage	9

Niveau 1

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

Niveau 2

Dimensionnement d'un tableau - Recherche d'un élément dans un tableau et dans une chaîne - L'instruction DATA	58
Un dé – Dessin de cercles	60
Nicomaque	62
L'indésirable	64
Le jeu de saute-mouton	66
La course au 20	68
L'électricien fou	70
Jeu de loto	72
Visions	74
Le jeu de la vie	76
Jour de la semaine	78
421	80
Musique	82
Machinamot	84

Dé truqué	86
Poker	88
Lettres	90
Accrochez les wagons	92
Tour de cartes	94
Compose	96
Tirage d'un dé truqué	98

Niveau 3

Manipulation des chaînes et graphiques haute résolution	102
Bridge	104
Le pendu	106
Mastermind	109
Répétitions	111
Labyrinthix	113
Belote	115
Un taquin de calcul	117
Hanoï	119
Plus grand, plus petit	121
Reines sur l'échiquier	123
Ordre	125
Idem	128
Permutations	130
A travers	133
Le loup et les agneaux	136
Le jeu de Marienbad	139
Pendu inversé	141
Espion	143
Décomaque	145
Horloge	147

Niveau 4

Les techniques d'animation élémentaire	150
Anomalie	152
Gouttes	154
Parcours du cavalier	156
Adresse	158
La bombe	160
Calculatrice	162
Anagrammes	165
Le jeu du quinze	167
Clavier	169
Quatre à quatre	171
Tic Tac Toe	174
Smokey	176

Danger spatial	178
Zombies	180
Taquin de lettres	182
Othello	184
Mur de briques	187
Diago	189
Mots	191
Le singe	193

Niveau 5

Les techniques particulières au PC	199
Course de chevaux	201
Chapeaux	203
Plusieurs en un	205
Œufs	207
Combat aérien	209
Affichage publicitaire	211
Captures dans l'espace	213
Cibles	215
Sous-marin	217
Le jeu du missile	219
Obstacles	222
Courses de voitures	224
Evasion	226
Mathieu	228
Ardoise magique	230
Cube	232
Chasse au sous-marin	234
Dames	236
Dessin de dés	239
Echecs	241
Fais-moi un dessin	244
Von Koch	247
Le lézard d'Alice	249

Conseils de lecture	251
---------------------	-----

Avant-propos

Vous trouverez dans ce livre plus de 100 programmes (essentiellement des jeux), rédigés en GW-Basic et Basic 2, pour les ordinateurs IBM, PC et compatibles. La plupart des compatibles PC sont fournis avec GW-Basic, l'Amstrad PC étant, quant à lui, livré avec Basic 2, qui fonctionne sous GEM. Pour des raisons de place, les commentaires ne concernent que la version GW-Basic ; les numéros de ligne facilitant les références. Chaque programme (sauf un cas) est cependant donné dans les deux versions ; quelques indications supplémentaires précisant le fonctionnement sous Basic 2 lorsqu'il est très différent de celui sous GW-Basic (utilisation de la souris notamment).

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.

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 : **MID\$**, **RIGHT\$**, **LEFT\$**. On découvre la boucle **WHILE... WEND**, la possibilité de définir des fenêtres en mode texte, et l'instruction de gestion du temps **TIME\$**.
- Le niveau 4 fait appel à une instruction très particulière qui facilite la réalisation de jeux interactifs : **INKEY\$**. On y découvre aussi les techniques de **SCROLLING** d'écran, la reconnaissance d'un caractère présent sur l'écran et les interruptions temporisées (**TIMER**).
- Enfin, **le niveau 5** est plus particulièrement consacré aux instructions graphiques : **PSET**, **LINE**, **CIRCLE**, **DRAW** et **PAINT** ; et à l'accès direct à la mémoire : **PEEK** et **POKE**.

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

Des indications sur le fonctionnement de la version Basic 2 sont également données. Puis les listings des programmes dans les deux versions (GW-Basic et Basic 2) sont imprimés d'après les originaux de l'auteur, sur la même imprimante.

Les programmes utilisent presque tous la couleur et, pour le niveau 5 (GW-Basic), ils font appel aux instructions graphiques. Il sera donc préférable de disposer d'un ordinateur avec écran graphique couleur.

La touche désignée par **RETURN** est, sur de nombreux claviers, appelée **ENTER**, ou .

L'arrêt avant la fin du programme est obtenu, en GW-Basic, en tapant **CTRL** et **Break**. En Basic 2, on clique sur **Break** dans la barre des menus, ou on tape **CTRL** et **C**.

Niveau 1

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

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

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

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

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

L'INSTRUCTION LET OU INSTRUCTION D'AFFECTATION

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

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

Exemple :

LET A1 = 12 ; la variable numérique A1 prend la valeur 12

LET NOM \$ = "RENOIR" ; la variable alphanumérique NOM\$ prend la valeur RENOIR.

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

L'INSTRUCTION PRINT OU INSTRUCTION DE SORTIE (ECRAN)

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

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

- **PRINT TAB(N)** provoque sur la ligne d'impression courante un affichage commençant à la *énième* colonne à partir de la gauche (de 1 à 40 ou de 1 à 80, selon le mode d'écran choisi).
- Un ; (**point-virgule**) en fin de PRINT permet soit d'enchaîner une autre impression sans répéter l'instruction PRINT, soit d'indiquer que la prochaine impression sera, en l'absence d'indication contraire, effectuée exactement à la suite de l'impression précédente.
- La , (**virgule**) provoque une tabulation horizontale automatique pré-définie.

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

L'INSTRUCTION LOCATE

Le GW-Basic dispose d'une instruction particulière, d'un emploi très souple, qui *permet de positionner le curseur sur l'écran, en vue de la prochaine écriture*. Il s'agit de l'instruction LOCATE X, Y (,T), qui sera généralement utilisée sous la forme LOCATE X, Y. X correspond à une tabulation verticale (de 1 à 24) et Y à une tabulation horizontale (de 1 à 40 ou 80) en mode texte. Le paramètre T, facultatif, prendra la valeur 0 si l'on veut faire disparaître la trace du curseur (ce qui sera utile pour certains jeux).

Ainsi, pour écrire le mot BONJOUR à la 5^e colonne de la 8^e ligne, on écrira :

LOCATE 5,8 : PRINT "BONJOUR"

Le choix de la largeur d'écran (40 ou 80 colonnes) peut être imposé par l'instruction WIDTH.

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 GW-Basic rencontre une instruction INPUT, il s'arrête et vous rend la main ; on peut alors lui entrer une information de nature convenable (nombre ou texte), et lui signaler que cette entrée est terminée à l'aide de la touche **RETURN**, ce qui lui rendra la main.

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

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

On pourra, par exemple, écrire :

INPUT "VOTRE NOM, S'IL VOUS PLAÎT" ; A\$

On pourra aussi faire appel à une instruction INPUT\$ (N), qui permet d'attendre une entrée jusqu'à ce que le nombre N de caractères demandés ait été frappé. Ainsi R\$ = INPUT\$ (1) permet l'attente d'un seul caractère saisi dans R\$.

L'INSTRUCTION GOTO OU INSTRUCTION DE RENVOI INCONDITIONNEL

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

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

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

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

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

IF condition = vraie THEN numéro de ligne.

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

On pourra aussi utiliser le test sous la forme :

IF condition THEN instruction;

l'instruction 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 $A + B$

Le GW-Basic permet, en outre, de prévoir une autre instruction en cas de non-réalisation du test. C'est la forme dite IF condition THEN instruction 1 ELSE instruction 2.

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

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

Ainsi, si l'on écrit :

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

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

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

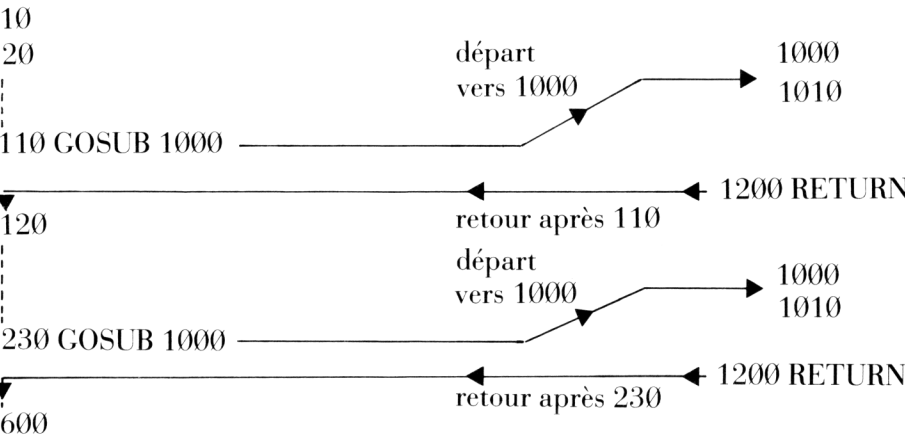
Il est possible d'emboîter plusieurs boucles de noms différents, et d'omettre l'indication de la variable dans le NEXT.

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

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

Voici le schéma théorique de fonctionnement :

Programme principal



Fin du programme principal

1000
:
1200 RETURN

Sous-programme 1

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

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

On peut aussi appeler un sous-programme de façon conditionnelle, en utilisant l'instruction `ON    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 GW-BASIC :

La ligne 50 permet de fabriquer le nombre à deviner.

Les entrées sont faites par les lignes 70 et 80.

La ligne 90 teste si le résultat est exact.

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

Le compteur d'essais est en ligne 120.

Les lignes 150 à 180 s'occupent de l'affichage du résultat, et de la gestion de la partie suivante.

POUR JOUER :

On lancera le programme par RUN.

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

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

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

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 RETURN).

On peut arrêter à tout moment en tapant CTRL et BREAK en GW-Basic et CTRL et C en Basic 2.

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...).

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.


```

10 REM LE NOMBRE MYSTERIEUX
20 SCREEN 1:CLS:RANDOMIZE TIMER
30 INPUT"NOMBRE DE CHIFFRES :";N
40 IF N>7 THEN RUN
50 X=INT(RND*10^N+1)
60 J=1
70 PRINT:PRINT"ESSAI NO. ";J
80 INPUT"VOTRE NOMBRE :";M
90 IF M=X THEN 150
100 IF M<X THEN PRINT"TROP PETIT"
110 IF M>X THEN PRINT"TROP GRAND"
120 J=J+1
130 PRINT:PRINT
140 GOTO 70
150 CLS
160 PRINT"VOUS AVEZ TROUVE ";X;" EN ";J;
" ESSAIS"
170 INPUT"AUTRE PARTIE (O/N)";R$
180 IF R$="O" THEN RUN

```

```

REM Le nombre mystérieux

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "LE NOMBRE MYSTERIEUX"
WINDOW OPEN
WHILE n>7 OR n<=0 OR n<>INT(n)
INPUT AT (1;3) "Nombre de chiffres ( <8 )
: ";n$:n=VAL(n$)
WEND
RANDOMIZE:x=RND(10^n)
j=0:PRINT
REPEAT
j=j+1
PRINT "Essai numéro ";j;TAB(20)" - Votre
nombre :";:INPUT " ",m$;:m=VAL(m$)
IF m<x THEN PRINT COLOR (4) TAB(60)"Trop
petit"
IF m>x THEN PRINT COLOR (2) TAB(60)"Trop
grand"
UNTIL m=x
PRINT:PRINT:PRINT "Vous avez trouvé ";x;"
en ";j;" essais"
r$=INPUT$(1)

```

Les allumettes

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : contre le PC.

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 PC. 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 GW-BASIC :

Le jeu du joueur commence en ligne 70.

Celui du PC commence en ligne 110.

La ligne 80 contrôle la légalité du coup du joueur.

La ligne 90 gère le changement de joueur.

Le coup du PC est calculé dans une boucle non formelle, de la ligne 120 à la ligne 150.

Le résultat est affiché par la ligne 210.

Si la partie est terminée, la ligne 220 le détecte et envoie en ligne 240, pour le message de fin.

POUR JOUER :

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

Après chaque coup, le nombre d'allumettes restantes est affiché, et le PC 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.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM LES ALLUMETTES
20 SCREEN 0:WIDTH 40:COLOR 1,2:CLS
30 INPUT"OMBRE TOTAL :";N
40 INPUT"PRISE MAXIMUM :";I
50 INPUT"QUI COMMENCE (VOUS:0 / MOI:1) :
";D
60 IF D=1 THEN 110
70 INPUT"VOUS RETIREZ :";P
80 IF P<>INT(P) OR P<1 OR P>N THEN 70
90 D=1
100 GOTO 190
110 S=I-D
120 T=(N-S)/(I+D)
130 IF T=INT(T) THEN 160
140 S=S-D
150 GOTO 120
160 P=ABS(S-D)
170 IF P=0 THEN P=D
180 D=0
190 PRINT P
200 N=N-P
210 PRINT"IL RESTE ";N
220 IF N=0 THEN 240
230 GOTO 60
240 COLOR 3,5:PRINT
250 IF D=1 THEN PRINT"J'AI GAGNE":BEEP
260 IF D<>1 THEN PRINT"BRAVO":BEEP:BEEP:
BEEP

```

REM Les allumettes

```

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "LES ALLUMETTES"
WINDOW OPEN
INPUT AT (2;2) "Nombre total d'allumettes
";n;n=VAL(n$)
INPUT AT (2;3) "Prise maximum par coup
";i;i=VAL(i$)
INPUT AT (2;4) "Qui commence ( Vous:0 - M
oi:1 ) ";d;d=VAL(d$)
PRINT:PRINT
REPEAT
FOR j=1 TO n:PRINT COLOR (14) " I";:NEXT:
PRINT
IF d=1 THEN GOSUB machine ELSE GOSUB joue
ur
n=n-p
PRINT "Il reste ";n;" allumettes"
UNTIL n=0
PRINT
IF d=1 THEN PRINT COLOR (4) "J'ai gagné..
.."
IF d<>1 THEN PRINT COLOR (2) "Mes félici
tations"
r$=INPUT$(1):END

```

```

LABEL joueur
p=0
WHILE p<>INT(p) OR p<1 OR p>i OR p>n
PRINT COLOR (2) "Vous enlevez :";:INPUT p
$:p=VAL(p$)
WEND
d=1
RETURN

```

```

LABEL machine
s=i+d
REPEAT
t=(n-s)/(i+d)
s=s-d
UNTIL t=INT(t)
p=ABS(s):IF p=0 THEN p=d
PRINT COLOR (4) "J'en prends :";p
d=0
RETURN

```

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 PC 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 GW-BASIC :

Les lignes 10 à 40 permettent les différentes entrées.

Les lignes 70 à 100 réalisent les impressions.

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

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

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

POUR JOUER :

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

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 PC.

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.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM ALLUMETTES A PLUSIEURS
20 SCREEN 0,1:WIDTH 40:COLOR 2,1:CLS
30 INPUT"OMBRE TOTAL D'ALLUMETTES :";N
40 INPUT"NOMBRE DE JOUEURS :";J
50 F=1
60 K=1
70 CLS
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 THEN 110
130 N=N-C
140 IF N<0 THEN N=N+C:GOTO 110
150 F=C+C
160 IF N=0 THEN 200
170 K=K+1
180 IF K>J THEN K=1
190 GOTO 70
200 COLOR 4,5:PRINT
210 PRINT"LE JOUEUR NUMERO ";K;" A GAGNE
":BEEP:BEEP:BEEP

```

```

REM Allumettes à plusieurs

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "ALLUMETTES A PLUSIEURS"
WINDOW OPEN
INPUT AT (2;5) "Nombre total d'allumettes
: ",n$:n=VAL(n$)
INPUT AT (2;7) "Nombre de joueurs
: ",j$:j=VAL(j$)
f=1:k=0
REPEAT
c=0:k=k+1:IF k>j THEN k=1
PRINT AT (2;10) COLOR (k) "Tour du joueur
";k
PRINT AT (2;12) "Total restant :";n
PRINT AT (30;12) "Maximum autorisé :";f
WHILE c<=0 OR c<>INT(c) OR c>f OR c>n
PRINT AT (14;15) "
INPUT AT (2;15) "Votre coup : ",c$:c=VAL(c
$)
WEND
n=n-c:f=2*c
UNTIL n=0
PRINT AT (2;18) "Le joueur ";k;" gagne"
r$=INPUT$(1)

```

Bataille navale

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : réflexion.

Difficulté du jeu : facile.

LE JEU :

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

A chaque coup, on devra tirer en donnant les coordonnées H (horizontale) et V (verticale) de la case visée, et le PC 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 GW-BASIC :

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

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

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

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

La ligne 180 permet l'affichage de la distance.

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM BATAILLE NAVALE
20 SCREEN 0,1:WIDTH 40:COLOR 2,5:CLS:RANDOMIZE TIMER
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*10)
100 Y=INT(RND*10)
110 LOCATE 22,1:INPUT"LIGNE :";H
120 LOCATE 22,20:INPUT"COLONNE :";V
130 LOCATE 2*H+2,3*V+4:COLOR 1,6:PRINT S
:COLOR 2,5
140 D=INT(SQR((X-H)*(X-H)+(Y-V)*(Y-V)))
150 IF D=0 THEN 200
160 S=S+1
170 IF D=1 THEN LOCATE 23,1:PRINT"PFFF..
."
180 LOCATE 23,1:PRINT"DISTANCE :";D;"
":BEEP
190 GOTO 110
200 COLOR 1,7
210 PRINT"TROUVE EN ";S;" COUPS"

```

```

REM Bataille navale

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "BATAILLE NAVALE"
WINDOW OPEN
a$="[___]":s=0
FOR i=0 TO 9
LOCATE 4*i+7;2:PRINT EFFECTS(8);i:PRINT A
T (2;i+3);i
FOR j=0 TO 9:PRINT AT (4*j+6;i+3);a$
NEXT j,i
RANDOMIZE:x=RND(10)-1:y=RND(10)-1
REPEAT
s=s+1
PRINT AT (1;14);"
"
INPUT AT (1;14) "Ligne :";h$;:h=VAL(h$):I
NPUT AT(20;14) "Colonne :";v$;v=VAL(v$)
PRINT EFFECTS(1) COLOR (4) POINTS(8) MODE
(2) AT (4*v+7;h+3) ;s
d=INT(SQR((x-h)*(x-h)+(y-v)*(y-v)))
PRINT AT (1;15);"Distance :";d;"
"
IF d=1 THEN PRINT AT (1;15);"PFFFF.....
"
":x=RND(10)-1:y=RND(10)-1
UNTIL d=0
PRINT:PRINT "Trouve en ";s;" coups"
r$=INPUT$(1)

```

Roulette

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : hasard (plusieurs joueurs).

Difficulté du jeu : facile.

LE JEU :

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

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

LE PROGRAMME GW-BASIC :

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

La boucle K (130-210) réalise le déplacement de la boule, l'arrêt étant contrôlé par la ligne 190.

La boucle J (ligne 180) réalise une légère temporisation (utilisation d'un sous-programme de ralentissement (270-280)).

La ligne 220 tire la position d'arrêt.

La ligne 240 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.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic. La relance est faite sur la frappe de « 0 », et l'arrêt en tapant une touche quelconque.


```

10 REM ROULETTE
20 SCREEN 0,1:WIDTH 40:COLOR 2,0
30 RANDOMIZE TIMER
40 CLS
50 FOR K=1 TO 4
60 FOR I=1 TO 9
70 S=I+I
80 PRINT TAB(4*I-1)(S AND K=1)+(S+18 AND
  K=2)+(S-1 AND K=3)+(S+17 AND K=4);
90 NEXT I
100 PRINT:PRINT:PRINT:PRINT
110 NEXT K
120 LOCATE 3,1:PRINT"P":LOCATE 7,1:PRINT
  "00":LOCATE 11,1:PRINT"I"
130 FOR K=1 TO 4
140 FOR I=1 TO 10
150 IF I=10 AND K<4 THEN LOCATE 4*K,1,1:
  GOTO 170
160 IF K*I<>40 THEN LOCATE 4*K-2,4*I,1
170 X=X-1
180 GOSUB 270
190 IF X=1 THEN PRINT"*":GOTO 240
200 NEXT I
210 NEXT K
220 X=INT(RND*38)
230 GOTO 120
240 LOCATE 20,1:INPUT"AUTRE ESSAI (O/N)
  :";R$
250 IF R$="N" OR R$="n" THEN END
260 GOTO 40
270 FOR J=1 TO 200-400*(X>0)-(200*(10-X)
  )*(X>0 AND X<8):NEXT J:SOUND 800,1
280 RETURN

```

```

REM Roulette

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "ROULETTE"
WINDOW OPEN
RANDOMIZE
FOR k=1 TO 4
FOR i=1 TO 9
s=2*i:f=6*i+2:g=4*k-1
PRINT AT (f;g) POINTS(20) EFFECTS(64) COL
OR(2);(s AND k=1)+(s+18 AND k=2)+(s-1 AND
  k=3)+(s+17 AND k=4)
NEXT i,k
PRINT POINTS(20) EFFECTS(64) COLOR(2) AT
(1;5);"P";AT (1;9);"00";AT (1;13);"I"
x=RND(76)
FOR h=1 TO 2
FOR k=1 TO 4
FOR i=1 TO 9-(k<4)
IF i=10 AND k<4 THEN f=4*k+3:g=1 ELSE f=4
  *k+1:g=6*i+4
x=x-1:PRINT AT (g;f) COLOR(14) "##"
FOR j=1 TO 250+3200*(1/x):NEXT
  j
IF x<>1 THEN PRINT AT (g;f) " " " ELSE GOT
  O fin
NEXT i,k,h
END

LABEL fin
PRINT AT (1;20);"Un autre essai (O/N) :";
:r$=INPUT$(1)
IF UPPER$(r$)="O" THEN 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 GW-BASIC :

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

La ligne 100 attend votre choix.

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

POUR JOUER :

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

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 PC.

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.).

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM COULOIRS
20 SCREEN 0,1:WIDTH 40:COLOR 4,7:CLS
30 PRINT"-----"
40 PRINT:PRINT"B          X 2"
50 PRINT:PRINT"----- X -----"
60 PRINT:PRINT"A          X 1"
70 PRINT:PRINT"-----"
80 I=0:J=0:K=0:COLOR 1
90 R$="":T$=""
100 LOCATE 13,1:INPUT"ENTREE 1 OU 2 :";C$
110 IF C$="" OR C$="3" OR LEN(C$)<>1 TH
EN 100
120 R$=R$+C$
130 PRINT:PRINT R$
140 IF C$="1" THEN 190
150 J=NOT(J)
160 IF J<>0 THEN 250
170 S$="B"
180 GOTO 220
190 I=NOT(I)
200 IF I<>0 THEN 250
210 S$="A"
220 T$=T$+S$
230 LOCATE 17,1:PRINT"SORTIE EN ";S$:PRI
NT:PRINT T$
240 GOTO 100
250 K=NOT(K)
260 IF K<>0 THEN 170
270 GOTO 210

```

```

REM Couloirs
STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "COULOIRS"
WINDOW OPEN
PRINT AT (2;3) COLOR(4) EFFECTS(1) "====
=====
PRINT AT (2;5) "1      <      " COLOR(4) EF
FECTS(1) "=====
PRINT "          A":PRINT AT (8;6) ">"
PRINT AT (2;7) COLOR(4) EFFECTS(1) "====
=====
PRINT AT (22;7) "<";AT (22;8) ">";AT (2;9
) "2      <      ";
PRINT COLOR(4) EFFECTS(1) "=====
====";
PRINT "          B":PRINT AT (8;10) ">"
PRINT AT (2;11) COLOR(4) EFFECTS(1) "====
=====
i=0:j=0:k=0:r$="":t$=""

LABEL entree
PRINT AT (1;14) "Entrée 1 ou 2 :
":INPUT AT (17;14) "",c$
IF c$<>"1" AND c$<>"2" GOTO entree
IF c$="1" GOTO labi

LABEL labj
j=NOT(j)
IF j<>0 GOTO labk
s$="B":GOTO sortie

LABEL labi
i=NOT(i)
IF i<>0 GOTO labk
s$="A":GOTO sortie

LABEL labk
k=NOT(k)
IF k<>0 THEN s$="B":GOTO sortie
s$="A":GOTO sortie

LABEL sortie
t$=t$+s$
PRINT AT (1;16) "Sortie en :";s$
PRINT:PRINT t$
GOTO entree

```

Chaud, froid

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

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

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

LE PROGRAMME GW-BASIC :

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

La ligne 100 calcule la distance par rapport aux deux coordonnées que vous avez entrées aux lignes 70 et 80.

La ligne 110 regarde si vous avez gagné.

Les lignes 130-140 normalisent la distance (qui doit avoir un sens, quelles que soient les dimensions de la grille initiale).

La ligne 150 effectue le départ, s'il y a lieu, vers l'un des sous-programmes d'affichage.

POUR JOUER :

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

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

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

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

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

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.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en **GW-Basic**.

```

10 REM CHAUD,FROID
20 SCREEN 0,1:WIDTH 40:COLOR 1,3:CLS:RAN
DOMIZE TIMER
30 INPUT"DIMENSION DE LA GRILLE :";D
40 X=INT(RND*D+1)
50 Y=INT(RND*D+1)
60 CLS
70 COLOR 1:INPUT"DONNEZ X :";A
80 INPUT"DONNEZ Y :";B
90 S=S+1
100 T=ABS(X-A)+ABS(Y-B)
110 IF T=0 THEN 280
120 PRINT"VOUS ETES ";
130 N=INT(SQR(D)/T)+1
140 IF N>5 THEN N=5
150 IF T<=SQR(D) THEN ON N GOTO 180,200,
220,240,260
160 COLOR 1:PRINT"COMPLETEMENT GELE"
170 GOTO 70
180 COLOR 15:PRINT"A PEINE TIEDE"
190 GOTO 70
200 COLOR 14:PRINT"TIEDE"
210 GOTO 70
220 COLOR 6:PRINT"CHAUD"
230 GOTO 70
240 COLOR 5:PRINT"TRES CHAUD"
250 GOTO 70
260 COLOR 4:PRINT"BRULANT"
270 GOTO 70
280 COLOR 5,2:PRINT
290 PRINT"VOUS AVEZ TROUVE EN ";S;" COUP
S":BEEP:BEEP:BEEP

```

```

REM Chaud,froid

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "CHAUD ET FROID"
WINDOW OPEN
RANDOMIZE
INPUT "Dimension de la grille : ",d$:d=VAL
(d$)
x=RND(d):y=RND(d):s=0:CLS
REPEAT
INPUT "Donnez X : ",a$:a=VAL(a$)
INPUT "Donnez Y : ",b$:b=VAL(b$)
s=s+1
t=ABS(x-a)+ABS(y-b)
n=INT(d/(t+1))+8/(d-1)
IF n>6 THEN n=6
IF t<>0 THEN ON n GOSUB v1,v2,v3,v4,v5,v6
UNTIL t=0
PRINT:PRINT "Vous avez trouvé en ";s;" es
sais"
r$=INPUT$(1):END

LABEL v1:PRINT COLOR(12) "Complètement ge
lé":RETURN

LABEL v2:PRINT COLOR(11) "A peine tiède":
RETURN

LABEL v3:PRINT COLOR(7) "Tiède":RETURN

LABEL v4:PRINT COLOR(14) "Chaud":RETURN

LABEL v5:PRINT COLOR(2) "Très chaud":RETU
RN

LABEL v6:PRINT COLOR(10) "Brûlant":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 GW-BASIC :

Le tirage des dés est réalisé par un sous-programme placé en lignes 310 et 320.

L'affichage est réalisé par les lignes 140 et 150.

Les lignes 170 à 200 calculent le gain dans les diverses configurations possibles.

La totalisation est effectuée en ligne 220 pour les gains, et en ligne 250 pour les pertes.

La variable R\$, testée en ligne 290, détermine la poursuite éventuelle du jeu.

POUR JOUER :

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

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM DEVINE UN CHIFFRE
20 SCREEN 0,1:WIDTH 40:COLOR 1,7
30 RANDOMIZE TIMER:G=0
40 CLS
50 INPUT"VOTRE MISE :";N
60 PRINT
70 INPUT"DONNEZ UN CHIFFRE DE 1 A 6 :";D
80 GOSUB 310
90 A=X
100 GOSUB 310
110 B=X
120 GOSUB 310
130 C=X
140 PRINT:PRINT:PRINT"      A      B
      C"
150 PRINT:PRINT"      ";A;"      ";B;"
      ";C"
160 T=0
170 IF D=A THEN T=T+N
180 IF D=B THEN T=T+N
190 IF D=C THEN T=T+N
200 IF T=0 THEN 240
210 PRINT:PRINT:PRINT"VOUS GAGNEZ ";T
220 G=G+T
230 GOTO 260
240 PRINT:PRINT:PRINT"PERDU      "
250 G=G-N
260 PRINT:PRINT:PRINT"TOTAL ACTUEL :";G
270 PRINT
280 INPUT"ON CONTINUE (O/N) :";R$
290 IF R$="N" OR R$="n" THEN END
300 GOTO 40
310 X=INT(RND*6+1):SOUND 100*(X+5),5
320 RETURN

REM Devine un chiffre

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "DEVINE UN CHIFFRE"
WINDOW OPEN
g=0:RANDOMIZE
LABEL début
CLS
INPUT AT (1;2) "Combien voulez-vous miser
: ";n$:n=VAL(n$)
INPUT AT (1;4) "Donnez un chiffre entre 1
et 6 : ";d$:d=VAL(d$)
IF d<>INT(d) OR d<1 OR d>6 GOTO début
GOSUB tirage:a$=CHR$(48+x)
GOSUB tirage:b$=CHR$(48+x)
GOSUB tirage:c$=CHR$(48+x)
PRINT AT (10;7) COLOR(2) POINTS(20) EFFEC
TS(1) "A" TAB(15) "B" TAB(23) "C"
PRINT AT (10;10) COLOR(4) POINTS(20);a$;T
AB(15);b$;TAB(23);c$
t=0
IF d$=a$ THEN t=t+n
IF d$=b$ THEN t=t+n
IF d$=c$ THEN t=t+n
g=g+t
IF t=0 THEN PRINT AT (1;14) "Vous perdez
";n:g=g-n ELSE PRINT AT (1;14) "Vous gagn
ez ";t
PRINT AT (1;16) "Le total de vos gains ou
pertes est de ";g
INPUT AT (1;19) "On continue ( O/N) : ";r$
IF UPPER$(r$)="N" THEN END ELSE GOTO débu
t

LABEL tirage:x=RND(6):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 10 cases sur 10 cases. Un point apparaît, en dehors de la diagonale et les joueurs devront le déplacer à chaque tour, suivant seulement trois directions : droite, bas et diagonale, bas et droite. Le premier joueur qui arrive en (9,9) sera considéré comme gagnant, ou perdant, selon la convention faite au début.

LE PROGRAMME GW-BASIC :

Les lignes 40, 50 et 60 permettent de définir une position de départ, en dehors de la diagonale.

La ligne 80 et la boucle I (90-110) dessinent le terrain de jeu.

La ligne 130 imprime la position du point mobile.

Le test de fin est effectué en ligne 140.

Le changement de joueur est assuré par la ligne 180.

Les directions de déplacement disponibles, 1, 2 et 3, sont décodées par trois sous-programmes qui commencent aux lignes 200, 220 et 230.

POUR JOUER :

Taper RUN pour lancer l'exécution du programme. Le terrain de jeu s'affiche, ainsi que la position de départ.

Puis on obtient l'affichage JOUEUR 1 et en dessous la question DIRECTION (1, 2, 3), à laquelle on répond en tapant l'un de ces trois nombres, suivi de RETURN.

C'est ensuite au tour du joueur 2.

Le jeu se termine lorsque l'un des joueurs arrive en case (9, 9).

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.


```

10 REM LE JEU DE WYTHHOF
20 SCREEN 0,1:COLOR 1,7:WIDTH 40
30 RANDOMIZE TIMER
40 X=3*INT(RND*5)+5
50 Y=INT(RND*5)
60 IF 3*Y+5=X THEN 50
70 CLS
80 PRINT"      0  1  2  3  4  5  6  7  8
9"
90 FOR I=0 TO 9
100 PRINT:PRINT I;"  .  .  .  .  .  .  .
.  .  ."
110 NEXT I
120 H=INT(RND*2+1)
130 LOCATE 2*Y+3,X+1:PRINT"+":BEEP
140 IF X=32 AND Y=9 THEN END
150 COLOR H+3:LOCATE 23,1:PRINT"JOUEUR "
;H;:INPUT" - DIRECTION (1,2,3)";D$
160 D=VAL(D$)
170 ON D GOSUB 200,220,230
180 H=2-ABS(H-1)
190 GOTO 130
200 IF X<32 THEN X=X+3
210 RETURN
220 GOSUB 200
230 IF Y<9 THEN Y=Y+1
240 RETURN

```

```

REM Le jeu de Wyththof

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "LE JEU DE WYTHHOF"
WINDOW OPEN
RANDOMIZE
a$="[___]":b$=CHR$(249):x=RND(5)-1
REPEAT:y=RND(5)-1:UNTIL y<>x
FOR i=0 TO 9
PRINT AT (4*i+6;2) EFFECTS(8);i;:PRINT AT
(1;i+3);i
FOR j=0 TO 9:PRINT AT (4*j+5;i+3);a$
NEXT j,i
h=1
REPEAT
PRINT AT (4*x+7;y+3) COLOR(6-2*h) MODE(2)
b$
PRINT AT (1;15) COLOR(2*h)"Joueur ";h
PRINT AT (1;17) "Direction choisie ( 1,2
ou 3) :";d$=INPUT$(1):d=VAL(d$)
ON d GOSUB droite,bas,diago
h=2-ABS(h-1)
UNTIL x=9 AND y=9
PRINT AT (43;12) COLOR(6-2*h) MODE(2) b$
PRINT AT (1;20) "C'est fini..... "
r$=INPUT$(1):END

LABEL droite
x=x+1:IF x>9 THEN x=9
RETURN

LABEL diago
GOSUB droite
GOSUB bas
RETURN

LABEL bas
y=y+1:IF y>9 THEN y=9
RETURN

```

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 GW-BASIC :

La ligne 60 permet de définir la position du trou, et la ligne 70 dessine le terrain de jeu.

Le déplacement à réaliser est calculé en ligne 110.

La boucle I (130-180) réalise le déplacement de la balle.

La ligne 190 regarde si le coup est, ou non, gagnant ; s'il l'est, on recommence (ligne 240) jusqu'à ce que l'on ait réalisé 18 trous (boucle S (40-240) ; fin de boucle en 260, avec affichage de fin).

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM GOLF
20 SCREEN 0,0,0:WIDTH 80:COLOR 1,15:CLS
30 RANDOMIZE TIMER
40 FOR S=1 TO 18
50 CLS
60 X=1:T=INT(64*RND+7)
70 COLOR 2:LOCATE 6,1:FOR I=1 TO 74:PRINT
  CHR$(127);:NEXT I
80 COLOR 1:LOCATE 6,T:PRINT " ":LOCATE 5,
  X:PRINT"0"
90 LOCATE 8,15:PRINT" "
100 LOCATE 8,1:PRINT"TROU ";S;:INPUT"FOR
  CE ";R
110 F=INT(R*.16-(ABS(R)*(SQR(RND)+.1)^2
  /1000)
120 IF F=0 THEN 90
130 FOR I=SGN(F) TO F STEP SGN(F)
140 X=X+SGN(F)
150 IF X<75 AND X>=1 THEN LOCATE 5,X-SGN
  (F):PRINT" ":LOCATE 5,X:PRINT"0"
160 IF X>=74 THEN X=74
170 IF X<=1 THEN X=1
180 NEXT I
190 BEEP:IF X=T THEN 220
200 E=E+1
210 GOTO 70
220 LOCATE 5,X:PRINT" ":LOCATE 6,X:PRINT
  "0"
230 FOR P=1 TO 500:NEXT P
240 NEXT S
250 CLS
260 PRINT"18 TROUS EN ";E;" COUPS"

```

```

REM Golf
STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "GOLF"
WINDOW OPEN
RANDOMIZE
e=0
FOR s=1 TO 18
CLS:x=0:t=RND(64)+7
LOCATE 2;6:FOR i=0 TO 74:PRINT COLOR(11)
MODE(4) CHR$(255);:NEXT
REPEAT
PRINT AT (t+2;6) " ":PRINT AT (x+2;5) COL
OR(2) "o"
REPEAT
PRINT AT (1;10) "Trou ";s;" - Force :
"::INPUT AT (22;10) "",r$
r=VAL(r$):f=INT(r*.16-(ABS(r)*SQR(RND+0.
1)^2)/1000)
UNTIL f<>0
IF f>74-x THEN f=74-x
IF f<=2-x THEN f=-x
FOR i=SGN(f) TO f STEP SGN(f)
x=x+SGN(f)
PRINT AT (x-SGN(f)+2;5) " ":PRINT AT (x+2
;5) COLOR(2) "o"
NEXT i
e=e+1
UNTIL x=t
e=e-1
PRINT AT (x+2;5) " ":PRINT AT (x+2;6) COL
OR(2) "o"

r$=INPUT$(1)
NEXT s
PRINT AT (1;14) "18 trous en ";e;" coups"
r$=INPUT$(1)

```

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 PC 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 GW-BASIC :

Les dés sont tirés par un sous-programme placé aux lignes 370 et 380.

La ligne 70 tient le compte du nombre de parties, et les lignes 170, 250 et 280 comptabilisent les points marqués.

La parité du total est déterminée par les lignes 120 et 130 (la ligne 130 compare avec la réponse du joueur).

La ligne 310 et les suivantes réalisent l'affichage de fin de partie : total, nombre de points, nombre de parties jouées.

La ligne 350 provoque l'arrêt si l'on répond N à la question ON CONTINUE.

POUR JOUER :

On lance le programme par l'instruction RUN.

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

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

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

EXTENSIONS POSSIBLES :

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

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM PAIR-IMPAIR
20 SCREEN 0,1:COLOR 1,15:WIDTH 40
30 RANDOMIZE TIMER:T=0:R=0
40 GOSUB 370
50 A=X
60 GOSUB 370
70 R=R+1
80 S=A+X
90 CLS
100 INPUT"TOTAL PAIR (0) OU IMPAIR (1) :
";N
110 PRINT:BEEP
120 P=S/2-INT(S/2)
130 IF N=P OR N-.5=P THEN 160
140 PRINT"FAUX"
150 GOTO 180
160 PRINT"EXACT"
170 T=T+1
180 PRINT
190 INPUT"TOTAL PRESUME :";N
200 IF N=S THEN 240
210 PRINT"PERDU"
220 FOR I=1 TO 3000:NEXT I
230 GOTO 310
240 PRINT:BEEP
250 T=T+2
260 INPUT"BRAVO . DONNEZ UN DE :";N
270 IF N<>A AND N<>S-A THEN 210
280 T=T+3
290 PRINT:BEEP
300 PRINT"BRAVO":GOTO 220
310 CLS
320 PRINT"VOUS AVEZ UN TOTAL DE ";T;" SU
R ";R
330 PRINT
340 INPUT"ON CONTINUE (O/N) :";R$
350 IF R$="N" THEN END
360 GOTO 40
370 X=INT(RND*6+1)
380 RETURN

```

```

REM Pair-impair

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "PAIR-IMPAIR"
WINDOW OPEN
RANDOMIZE
t=0:r=0
REPEAT
  GOSUB tirage
  a=x
  GOSUB tirage
  r=r+1:s=a+x
  CLS
  REPEAT
    INPUT AT (1;4) "Total (P)air ou (I)mpair
: ",n$:n$=UPPER$(n$)
    UNTIL n$="P" OR n$="N"
    n=(0 AND n$="P")+(1 AND n$="N")
    p=FRAC(s/2)
    IF n=p OR n-0.5=p THEN t=t+1:PRINT AT (1;
6) COLOR(4) "EXACT" ELSE PRINT AT (1;6) C
OLOR(2) "FAUX "
    INPUT AT (1;8) "Donnez le total présumé :
",n$:n=VAL(n$)
    IF n=s THEN GOSUB bravo ELSE PRINT:PRINT
"Perdu"
    PRINT:PRINT"Vous avez ";t;" points en ";r
;" essais"
    PRINT:PRINT "On continue (O/N):";r$=INPU
T$(1):r$=UPPER$(r$)
    UNTIL r$="N"
  END

LABEL bravo
t=t+2:INPUT AT (1;10) "Bravo. Donnez un d
é : ",n$:n=VAL(n$)
IF n=a OR n=s-a THEN t=t+3:PRINT:PRINT "T
rès bien joué" ELSE PRINT:PRINT "Dommage"
RETURN

LABEL tirage
x=RND(6)
RETURN

```

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 GW-BASIC :

La boucle I (60-80) permet le dessin de la machine, la ligne 90 affichant les emplacements (vides) des chiffres.

La mise sera entrée en ligne 100, et testée en ligne 110 : si on mise 0, la partie est terminée (départ vers la ligne 290).

Les lignes 120, 140 et 160 permettent le tirage des 3 chiffres, en appelant un sous-programme qui se trouve en ligne 270.

Le résultat est affiché par la ligne 170, puis testé en ligne 180, et en ligne 250, si l'on a gagné.

POUR JOUER :

Lancer le programme par RUN ; à l'affichage de la question POT DE DEBUT, on répond en entrant un nombre suffisamment élevé, sur lequel vos mises seront décomptées.

Puis on tape RETURN, et on entre une première mise (différente de 0), suivie de RETURN.

Les trois chiffres tirés sont alors affichés un bref instant, ainsi que la mise faite, et le résultat obtenu (gain ou perte).

On peut ensuite miser de nouveau, faire RETURN et ainsi de suite.

Pour terminer, il suffit de répondre 0 à la question VOTRE MISE.

Le restant de votre pot initial sera alors affiché.

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM JACKPOT
20 SCREEN 0,1:COLOR 1,7:WIDTH 40
30 RANDOMIZE TIMER
40 INPUT"POT DE DEBUT :";P
50 CLS
60 FOR I=1 TO 3
70 LOCATE I+4,10:COLOR 3,1:PRINT "
    ":COLOR 0,5
80 NEXT I
90 LOCATE 6,14:PRINT"    ":LOCATE 6,19:PR
INT"    ":LOCATE 6,24:PRINT"    "
100 COLOR 0,7:LOCATE 11,1:INPUT"VOTRE MI
SE :";M
110 IF M=0 THEN 290
120 GOSUB 270
130 A=X
140 GOSUB 270
150 B=X
160 GOSUB 270
170 COLOR 11,5:LOCATE 6,15:PRINT A:LOCAT
E 6,20:PRINT B:LOCATE 6,25:PRINT X
180 IF A=B AND A=X THEN 240
190 N=-M:IF A=B OR A=X OR B=X THEN N=-N
200 COLOR 0,7:LOCATE 11,1:PRINT"MISE :";
:COLOR 4:PRINT M;:COLOR 0:PRINT" - RESUL
TAT :";:COLOR 4:PRINT N;"    "
210 COLOR 0:INPUT"TAPEZ - ENTREE - POUR
CONTINUER";R$
220 P=P+N
230 GOTO 50
240 N=M*50
250 IF A=7 THEN N=M*500
260 GOTO 200
270 X=INT(RND*9+1):SOUND 100*(X+5),2
280 RETURN
290 CLS
300 PRINT"VOUS AVEZ ";P
310 PRINT"A BIENTOT"

```

```

REM Jackpot

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "JACKPOT"
WINDOW OPEN
RANDOMIZE
INPUT AT (1;3) "Somme disponible pour la
partie :";p$:p=VAL(p$)
LABEL partie
CLS
FOR i=1 TO 6
PRINT AT (6;3+i) COLOR (2) MODE(4) "
    "
NEXT i
PRINT AT (1;1) "Le pot contient ";p;" fra
ncs
    "
PRINT AT (9;7) POINTS(20) "    " AT (18;7)
    "    " AT (27;7) "    "
INPUT AT (1;14) "Votre mise pour cet essa
i (0=fin) :";m$:m=VAL(m$)
IF m=0 OR p-m<0 THEN GOTO fin
a=RND(9):b=RND(9):c=RND(9)
a$=CHR$(48+a):b$=CHR$(48+b):c$=CHR$(48+c)
PRINT AT (10;7) EFFECTS(1) COLOR(a) a$ AT
(19;7) COLOR(b) b$ AT (28;7) COLOR(c) c$
n=-m
IF a=b OR a=c OR b=c THEN n=m*2
IF a=b AND b=c THEN n=m*50:IF a=7 THEN n=
10*n
PRINT AT (6;10) "MISE :";m;" > RESULTAT
:";n;"    "
PRINT AT (1;14) "Appuyez sur une touche
    "
r$=INPUT$(1)
p=p+n:GOTO partie

LABEL fin
PRINT AT (1;16) "Il vous reste ";p;" fran
cs
    "
PRINT:PRINT "A bientôt.... "
r$=INPUT$(1):END

```

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 GW-BASIC :

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

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

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

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

POUR JOUER :

Après avoir lancé le programme par l'instruction RUN, on devra introduire l'année, sous la forme, par exemple 1985, suivie de RETURN.

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

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

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.


```

10 REM CALENDRIER PERPETUEL
20 SCREEN 0,1:WIDTH 40:COLOR 1,5:CLS
30 INPUT"POUR QUELLE ANNEE :";A
40 INPUT"POUR QUEL MOIS   :";M
50 D=INT(365.25*(A+(M<3)))+1
60 D=D+INT(30.6*(M+1-(M<3)*12))-INT((INT
((A+(M<3))/100)-7)*.75)
70 D=D-7*INT(D/7)
80 M=31+(M=4 OR M=6 OR M=9 OR M=11)+(M=2
)*3
90 PRINT:COLOR 14,5
100 PRINT" DIM LUN MAR MER JEU VEN SAM"
110 COLOR 3,4:PRINT
120 FOR A=1 TO M
130 PRINT TAB(4*(D+A-1)+2);A;
140 IF D+A-1=6 THEN PRINT:D=D-7
150 NEXT A

```

```

REM Calendrier perpétuel

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "CALENDRIER PERPETUEL"
WINDOW OPEN
j$="DIMANCHE LUNDI   MARDI   MERCREDI J
EUDI   VENDREDI SAMEDI   "
INPUT AT (1;3) "Quelle année :",a$:a=VAL(
a$)
INPUT AT (1;5) "Quel mois   :",m$:m=VAL(
m$)
d=INT(365.25*(a+(m<3)))+1
d=d+INT(30.6*(m+1-(m<3)*12))-INT((INT((a+
(m<3))/100)-7)*0.75)
d=d-7*INT(d/7)
m=31+(m=4 OR m=6 OR m=9 OR m=11)+(m=2)*3
PRINT AT (1;7) COLOR(4) j$
PRINT:PRINT
FOR a=1 TO m
PRINT COLOR(2) TAB(2+9*(d+a-1));a;
IF d+a-1=6 THEN PRINT:PRINT:d=d-7
NEXT a
r$=INPUT$(1)

```

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 WYTHLOF, cette version offre en fait 4 possibilités de déplacement ; la position de départ est ici fixée en (0, 0), et pourrait éventuellement être modifiée.

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

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

LE PROGRAMME GW-BASIC :

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

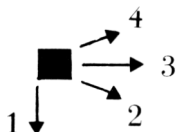
La ligne 90 réalise le test de fin de partie.

La ligne 120 permet d'exécuter l'un des sous-programmes de déplacement.

Les lignes 130 et 140 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 RETURN.

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

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

EXTENSIONS POSSIBLES :

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

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM LE JEU DE LA FEVE
20 SCREEN 0,1:COLOR 1,7:WIDTH 40:CLS
30 PRINT"      0 1 2 3 4 5 6 7 8
9"
40 FOR I=0 TO 9
50 PRINT:PRINT I;" . . . . .
. . ."
60 NEXT I
70 X=0:Y=0:J=0
80 COLOR 4*J:LOCATE 2*Y+3,3*X+6:PRINT"#"
90 IF X=9 AND Y=9 THEN 160
100 COLOR 4*J:LOCATE 23,1:PRINT"JOUEUR "
;J+1;:INPUT"- DIRECTION(1,2,3,4)";I$
110 I=VAL(I$):BEEP
120 ID=0:A=X:B=Y:ON I GOSUB 180,200,210,
230
130 IF ID=1 THEN J=ABS(J-1):X=A:Y=B
140 J=J+1+2*(J=1)
150 GOTO 80
160 COLOR 1:PRINT"LE JOUEUR ";J+1;" A GA
GNE          ":PLAY "CEG"
170 END
180 Y=Y+1:IF Y>9 THEN ID=1
190 RETURN
200 GOSUB 180
210 X=X+1:IF X>9 THEN ID=1
220 RETURN
230 GOSUB 210
240 Y=Y-1:IF Y<0 THEN ID=1
250 RETURN

```

```

REM Le jeu de la fève

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "JEU DE LA FEVE"
WINDOW OPEN
a$=CHR$(124)+"___"+CHR$(124)
FOR i=0 TO 9
PRINT AT (4*i+6;2) EFFECTS(8) i:PRINT AT
(1;i+3) i
FOR j=0 TO 9:PRINT AT (4*j+5;i+3) a$
NEXT j,i
PRINT AT (62;6) "4" AT (61;7) "/" AT (61;
8) "--3" AT (60;9) CHR$(124)+"\"
PRINT AT (60;10) "1 2"
x=0:y=0:j=0
REPEAT
PRINT AT (4*x+6;y+3) COLOR(7) MODE(4) "___"
PRINT AT (1;15) COLOR(2*j+2) "Joueur " j+
1 " "
WHILE d$<>"1" AND d$<>"2" AND d$<>"3" AND
d$<>"4"
PRINT AT (1;17) "Direction (1,2,3,4) : ";
:d$=INPUT$(1)
PRINT d$:d=VAL(d$)
WEND
PRINT AT (4*x+6;y+3) "___"
id=0:a=x:b=y:ON d GOSUB bas,diagbas,droit
,diago
IF id=1 THEN j=j-1:x=a:y=b
j=j+1+2*(j=1):d$=""
UNTIL x=9 AND y=9
PRINT AT (42;12) COLOR(14) MODE(2) "###"
PRINT AT (1;19) COLOR(2*j+2) "Le joueur "
;j+1;" gagne"
r$=INPUT$(1):END

LABEL bas
y=y+1:IF y>9 THEN id=1
RETURN

LABEL diagbas
GOSUB bas
GOSUB droit
RETURN

LABEL droit
x=x+1:IF x>9 THEN id=1
RETURN

LABEL diago
GOSUB droit
y=y-1:IF y<0 THEN id=1
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 GW-BASIC :

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

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

Les lignes 80 et 90 effectuent leur impression.

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

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

La ligne 160 imprime la nouvelle position.

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

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

Le nombre de rebonds est compté en ligne 230.

POUR JOUER :

On lance le programme par RUN.

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

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 Break.

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 billes, les comptabiliser pour faire une partie à plusieurs joueurs, etc.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```
10 REM REBONDS
20 SCREEN 0,1:WIDTH 40:COLOR 1,7
30 RANDOMIZE TIMER
40 CLS
50 PRINT" 0123456789ABCDEFGHIJKLMNQRST
UVWXYZ"
60 X=INT(RND*18+3)
70 Y=INT(RND*18+3)
80 COLOR 4:LOCATE X,1:PRINT"#"
90 COLOR 5:LOCATE Y,40:PRINT"+"
100 COLOR 1:LOCATE 23,1:INPUT"VOTRE COUP
:";R$
110 S=0
120 T=X
130 FOR J=1 TO 39
140 T=T-X/(ASC(R$)-ASC("0")+7*(R$>="A"))
150 IF T>=22 OR T<=0 THEN 220
160 COLOR 4:LOCATE T+1,J+1:PRINT"#"
170 NEXT J
180 COLOR 1:IF ABS(T-Y+1)<3 THEN PRINT"B
RAVO :";S;" REBONDS":PLAY"CEG"
190 INPUT"AUTRE PARTIE (O/N):";R$
200 IF R$="N" OR R$="n" THEN END
210 GOTO 40
220 X=-X
230 S=S+1:BEEP
240 GOTO 170

REM Rebonds
STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "REBONDS"
WINDOW OPEN
RANDOMIZE(TIME)
REPEAT
CLS
PRINT COLOR(4) AT (1;1) " -0-1-2-3-4-5-6-
7-8-9-A-B-C-D-E-F-G-H-I-J-K-L-M-N-O-P-Q-R
-S-T-U-V-W-X-Y-Z"
PRINT AT (1;18);:FOR j=0 TO 74:PRINT COLO
R(4) CHR$(27)+CHR$(19);:NEXT j
x=RND(15)+2:y=RND(15)+2
PRINT AT (75;y) COLOR(11) "##"
PRINT AT (1;x) COLOR(2) CHR$(249)
INPUT AT (1;19) "Votre coup : ";r$:r$=UPP
ER$(r$)
s=0:t=x
FOR j=1 TO 73
PRINT AT (j;t) " "
LABEL calcul:t=t-(x/(ASC(r$)-ASC("0")+7*(
r$>="A")+3))/2
IF t>17 OR t<1.5 THEN x=-x:s=s+1:GOTO cal
cul
PRINT AT (j;t) COLOR(2) " "+CHR$(249)
FOR k=1 TO 100:NEXT k
NEXT j
IF ABS(t-y)<2 THEN PRINT AT (20;19) "Brav
o : ";s;" rebonds"
PRINT AT (55;19) "Autre partie (O/N) ";:r
$=INPUT$(1)
UNTIL UPPER$(r$)="N"
r$=INPUT$(1)
END
```

Echiquier

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : graphisme.

Difficulté du jeu : variable, selon utilisation.

LE JEU :

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

LE PROGRAMME GW-BASIC :

Les lignes 30 à 40 définissent deux variables A\$ et B\$ qui contiennent les éléments de dessin. La ligne 50 imprime les lettres horizontales.

La boucle I (60-120) permet l'impression, le motif à imprimer étant affiché ou non selon la valeur de $(I + J)/2$; les lignes 90 et 100 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.).

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM ECHIQUEUR
20 SCREEN 0,1:COLOR 0,7:WIDTH 40:CLS
30 A$=CHR$(219):B$=CHR$(220)
40 A$=A$+A$:B$=B$+B$:B$
50 PRINT"    A B C D E F G H"
60 FOR I=1 TO 9
70 COLOR 0,7:LOCATE 2*I+1,1:PRINT I
80 FOR J=1 TO 8
90 LOCATE 2*I,3*J+1:COLOR -(3*((I+J) MOD
2)+1)*(I<>9)-7*(I=9),-(4-3*((I+J) MOD 2
))**(I<>1)-7*(I=1):PRINT B$
100 IF I<9 THEN LOCATE 2*I+1,3*J+1:COLOR
3*((I+J) MOD 2)+1:PRINT A$
110 NEXT J
120 NEXT I

```

```

REM Dessin d'un échiquier

STREAM #1:SCREEN GRAPHICS 640 FIXED,300 F
IXED
USER SPACE 640,300
WINDOW FULL ON
WINDOW TITLE "ECHIQUEUR"
WINDOW OPEN
BOX 100;50,400,300 COLOR(8) FILL WITH 4
FOR i=1 TO 8
PRINT AT (15+5*i;1) MODE(2) COLOR(15) CHR
$(64+i) AT (15;1+2*i) COLOR(11) CHR$(48+i
)
FOR j=1 TO 8
BOX 100+40*i;30+28*j,40,28 COLOR((i+j) MO
D 2) FILL WITH 8
NEXT j
NEXT i
r$=INPUT$(1)
END

```

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 GW-BASIC :

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

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

La ligne 120 affiche le score.

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

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

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

La ligne 180 contrôle le résultat.

POUR JOUER :

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

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

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

Pour arrêter, il suffira de répondre 0 suivi de RETURN, à la question NOTE.

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.


```

10 REM SOLFEGE
20 SCREEN 0,1:WIDTH 40:COLOR 1,15:RANDOM
IZE TIMER
30 S=0:P=0:S$="4F4E4D4C3B3A3F3E3D3C2B2A"
40 CLS
50 FOR I=1 TO 5
60 LOCATE 2*I+1,1:FOR J=1 TO 40:PRINT CH
R$(223);:NEXT J
70 NEXT I
80 X=INT(RND*12)+1
90 LOCATE X,18:PRINT CHR$(179)
100 LOCATE X+1,18:PRINT CHR$(179)
110 LOCATE X+2,17:PRINT"O"
120 LOCATE 21,1:PRINT"SCORE ";S;" SUR ";
P;" ESSAIS"
130 INPUT"CLE DE SOL - QUELLE EST LA NOT
E :";R$
140 IF R$="O" THEN END
150 Y=(1 AND R$="FA")+(2 AND R$="MI")+(3
AND R$="RE")+(4 AND R$="DO")
160 Y=Y+(5 AND R$="SI")+(6 AND R$="LA")
170 P=P+1
180 IF Y=X OR Y=X-7 THEN S=S+1:PLAY "O"+
MID$(S$,2*X-1,2)
190 GOTO 40

```

```

REM Solfège

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "SOLFEGE"
WINDOW OPEN
a$=CHR$(124):RANDOMIZE(TIME)
REPEAT
CLS
FOR i=1 TO 5
LOCATE 8;2*i+2:FOR j=1 TO 46:PRINT COLOR(
4) CHR$(95);:NEXT:PRINT
NEXT i
x=RND(12)+2
PRINT COLOR(2) AT (30;x-1) MODE(2) a$ AT
(30;x) a$ AT (30;x+1) a$
PRINT COLOR(2) AT (28;x+1) MODE(2) CHR$(4
7)+CHR$(92) AT (28;x+2) CHR$(92)+CHR$(47)
PRINT AT (12;18) "Score ";s;" en ";p;" es
sais"
INPUT AT (1;20) "Clé de sol - Quelle est
cette note ( 0>fin ) :";r$
r$=UPPER$(r$):IF LEN(r$)>=2 THEN p=p+1
y=(2 AND r$="SOL")+(3 AND r$="FA")+(4 AND
r$="MI")+(5 AND r$="RE")
y=y+(6 AND r$="DO")+(7 AND r$="SI")+(8 AN
D r$="LA")
IF y=x OR y=x-7 THEN s=s+1
UNTIL UPPER$(r$)="O"
END

```

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 PC ; il suffira d'appuyer sur une touche après chaque lancer de dé, et de regarder ce qui se passe.

LE PROGRAMME GW-BASIC :

La ligne 60 permet d'appeler le sous-programme 250 à 350 qui réalise l'affichage du terrain de jeu, le contrôle de la position des joueurs et l'affichage de cette position. Cet affichage sera d'ailleurs refait en cours de partie, pour tenir compte du déplacement réellement effectué (qui n'est pas nécessairement celui indiqué par le dé).

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

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

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

Les lignes 250 et 260 font avancer les pions, et les lignes 270 à 330 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 Break.

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```
10 REM JEU DE L'OIE ALEATOIRE
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER
40 CLS
50 X=0:Y=0:J=1:I=X
60 GOSUB 250
70 J=ABS(J-1)
80 I=INT(RND*6+1)
90 IF X=35 OR Y=35 THEN LOCATE 20,1:PRINT
T"BRAVO":PLAY"CEG":END
100 BEEP:LOCATE 13,1:PRINT"JOUEUR ";J+1;
" - VOTRE COUP :";I
110 INPUT"APPUYER SUR - ENTREE -";R$
120 GOSUB 250
130 I=INT(RND*4)
140 F=INT(RND*5+1)
150 ON F GOSUB 170,190,210,230,160
160 GOTO 60
170 I=I+3
180 RETURN
190 I=-I
200 RETURN
210 I=2*I
220 RETURN
230 IF J=0 THEN X=INT(RND*30):RETURN
240 Y=INT(RND*30):RETURN
250 IF J=0 THEN X=X-I*(X+I<=35)
260 Y=Y-J*I*(Y+I<=35)
270 LOCATE 4,1:PRINT"0123456789ABCDEFGHI
JKLMNPQRSTUVWXYZ"
280 FOR I=0 TO 35
290 LOCATE 6,I+1,0:COLOR 3,1:PRINT" ";
300 IF I=X THEN LOCATE 6,I+1:PRINT"1";
310 LOCATE 7,I+1,0:COLOR 2,4:PRINT" ";
320 IF I=Y THEN LOCATE 7,I+1:PRINT"2";
330 NEXT I
340 FOR I=1 TO 800:NEXT I
350 COLOR 0,6:RETURN

REM Jeu de l'oie aléatoire
STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "JEU DE L'OIE ALEATOIRE"
WINDOW OPEN
RANDOMIZE(TIME)
x=0:y=0:j=1:i=x:k=35
FOR i=0 TO k
PRINT AT (i+1;3) CHR$(48+i-7*(i>9))
NEXT
WHILE x<>k AND y<>k
GOSUB dessin
j=ABS(j-1):i=RND(6)
PRINT AT (1;11) COLOR(2*j+2) "Joueur ";j+
1
PRINT AT (15;11) "Votre coup : ";i
PRINT:INPUT "Pour jouer, appuyer sur - RE
TURN -",r$
a=x:b=y:GOSUB dessin
i=RND(4)-1:f=RND(5)
ON f GOSUB plus_3,opp,double,hasar
WEND
PRINT AT (1;17) POINTS(20) COLOR(14) "BRA
VO"
r$=INPUT$(1):END

LABEL plus_3
i=i+3
RETURN

LABEL opp
i=-i
RETURN

LABEL double
i=2*i
RETURN

LABEL hasar
IF j=0 THEN x=RND(20)-1 ELSE y=RND(20)-1
RETURN

LABEL dessin
IF j=0 THEN x=x+(i AND (x+i<=k)) ELSE y=y
+(i AND (y+i<=k))
FOR i=0 TO k
PRINT AT (i+1;5) COLOR(2) "1";
IF i=x THEN PRINT AT (x+1;5) "1";
PRINT AT (i+1;6) COLOR(4) "1";
IF i=y THEN PRINT AT (y+1;6) "2";
NEXT i
RETURN
```

Le jeu du 21

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu contre le PC.

Difficulté du jeu : facile.

LE JEU :

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

Les lignes 70 et 80 réalisent le contrôle de celui qui doit jouer (joueur ou PC).

Le sous-programme (310-320) lance les dés.

Les lignes 100 et 110 calculent le total fait par le joueur et l'affichent.

Le total du PC est calculé en ligne 170 et affiché par la ligne 180.

La ligne 200 contrôle la fin du jeu pour le PC.

La ligne 230 affiche les résultats, et la ligne 240 cherche quel est le gagnant.

Le gain (ou la perte) sera affiché par la ligne 270, ainsi que le total.

La ligne 280 contrôle la fin de partie.

POUR JOUER :

Taper RUN pour lancer le jeu.

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

Dès que l'on a terminé, on tape 0 et RETURN pour laisser jouer le PC (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.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic, sauf le tirage qui est fait en tapant « 0 » au lieu de « 1 », et arrêté en tapant « N ».

```
10 REM JEU DU 21
20 SCREEN 0,1:COLOR 1,14:WIDTH 80
30 RANDOMIZE TIMER:G=0
40 CLS
50 T=0
60 INPUT"VOTRE MISE :";M
70 COLOR 1:INPUT"FAIRE 1 POUR TIRER , 0
POUR FINIR";R
80 IF R=0 THEN 130
90 GOSUB 310
100 T=T+X:BEEP
110 COLOR 4:PRINT"VOTRE TOTAL EST DE ";T
120 GOTO 70
130 K=1
140 COLOR 1:LOCATE K,40:PRINT "LA BANQUE
JOUE"
150 S=0
160 GOSUB 310
170 S=S+X:K=K+2
180 COLOR 11:LOCATE K,40:PRINT "TOTAL D
E LA BANQUE ";S
190 FOR I=1 TO 500:NEXT I
200 IF S>T OR S>17 THEN 220
210 GOTO 160
220 LOCATE 22,1:BEEP:BEEP:BEEP
230 COLOR 1:PRINT"VOUS AVEZ ";T;TAB(30);
"LA BANQUE A ";S
240 IF S>21 OR (S<T AND T<=21) THEN 260
250 M=-M
260 G=G+M
270 PRINT "GAIN ";M;TAB(30) "VOTRE TOTAL
ACTUEL ";G
280 INPUT"ON CONTINUE (O/N) :";R$
290 IF R$="N" THEN END
300 GOTO 40
310 X=INT(RND*6+1)
320 RETURN

REM Jeu du 21

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "JEU DU 21"
WINDOW OPEN
RANDOMIZE
g=0
REPEAT
CLS:t=0
INPUT AT (1;1) "Combien voulez-vous miser
: ";m$:m=VAL(m$)
REPEAT
GOSUB tirage:t=t+x
PRINT COLOR(12) "Tir :";x;" Total actuel
: ";t
PRINT "Voulez-vous un autre tirage (O/N)
: ";r$=INPUT$(1):PRINT r$
UNTIL UPPER$(r$)="N"
l1=VPOS
PRINT AT (50;1) "LA BANQUE JOUE"
s=0:k=0:PRINT
WHILE s<t AND s<17
GOSUB tirage:s=s+x
PRINT COLOR(14) AT(40;3+k) "La banque tir
e :";x;" Total :";s
k=k+2
FOR i=1 TO 1200:NEXT i
WEND
l2=VPOS
l=MAX(l1,l2)+1
PRINT COLOR(12) AT (1;1) "Votre total est
de ";t
PRINT COLOR(14) AT (40;1) "Celui de la ba
nque ";s
IF s<21 AND (s>=t OR t>21) THEN m=-m
PRINT "Votre gain est de ";m
g=g+m
PRINT COLOR(2) "Votre gain total se monte
à ";g
PRINT "On continue (O/N) :";r$=INPUT$(1)
UNTIL UPPER$(r$)="N"
END

LABEL tirage
x=RND(6)
RETURN
```


Niveau 2

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

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 30, et une variable R (I) pour les recettes, et D (I) pour les dépenses ; ce faisant, vous avez, sans le savoir, utilisé une structure de tableau à une dimension ; ceci ne pourra être admis en MSX que si vous avez auparavant prévenu que vous alliez utiliser un tel tableau, en donnant une instruction DIM R (30) pour les recettes, et DIM D (30) pour les dépenses.

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

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

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

En GW-Basic il n'est pas obligatoire de dimensionner un tableau qui contient moins de 10 éléments.

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

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

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

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

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

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

LE TRAITEMENT DE DONNEES MULTIPLES INSTRUCTIONS READ, DATA, RESTORE

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

Ainsi, on pourra écrire :

DATA LEON, JULES, 13, rue MARTIN...

Ces données peuvent être lues dans 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.

Un dé (dessin de cercles)

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 GW-BASIC :

La ligne 120 utilise le générateur aléatoire du PC pour tirer un nombre de 1 à 6.

La boucle J (80-100) dessine les bords du dé, dans les huit écrans définis.

Les lignes 140-190 sont utilisées pour le dessin des points.

Les lignes 240 et 250 permettront le cas échéant de refaire un tirage.

POUR JOUER :

Lancer le programme par RUN.

Plusieurs dés défilent successivement, puis le dé tiré reste affiché.

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

Les dés des sept écrans précédents sont alors effacés, puis sept autres dés sont lancés avant l'arrêt.

EXTENSIONS POSSIBLES :

On pourra introduire une petite musique agrémentant chaque lancer, faire rouler les dés sur une table, etc.

VERSION BASIC 2 AMSTRAD :

Au lieu d'un programme de lancer de dés, dont l'adaptation en Basic 2 ne pose guère de problèmes, le programme propose une démonstration des capacités d'utilisation de la souris de l'AMSTRAD. La taille du cercle est choisie avec le bouton de droite, et le cercle est dessiné en appuyant sur le bouton de gauche. On quitte le programme en appuyant simultanément sur les deux boutons.

```

10 REM UN DE
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF
40 CLS:A$=CHR$(219)
50 FOR I=1 TO 7:SCREEN ,,I,I:CLS
60 COLOR 0,7:SCREEN 0,1,I,0
70 COLOR ,1:LOCATE 6,11:PRINT"
"
80 FOR J=0 TO 10
90 LOCATE 7+J,11:PRINT" ":LOCATE 7+J,23:
PRINT" "
100 NEXT J
110 LOCATE 18,11:PRINT"          "
120 A=INT(RND*6+1)
130 COLOR INT(RND*7),7
140 IF A<>1 THEN LOCATE 9,14:PRINT A$
150 IF A=4 THEN LOCATE 9,20:PRINT A$
160 IF A=6 THEN LOCATE 12,14:PRINT A$:LO
CATE 12,20:PRINT A$
170 IF A/2 <>INT(A/2) THEN LOCATE 12,17:
PRINT A$
180 IF A>4 THEN LOCATE 15,14:PRINT A$
190 IF A<>1 THEN LOCATE 15,20:PRINT A$
200 NEXT I
210 FOR I=1 TO 7
220 SCREEN ,,I,I:SOUND INT(RND*4000+500)
,1:FOR J=1 TO 1000:NEXT J
230 NEXT I
240 COLOR 4,7:LOCATE 21,1:PRINT"AUTRE TI
RAGE (O/N)":R$=INPUT$(1)
250 IF R$="O" OR R$="o" THEN 50 ELSE SCR
EEN 0,1,0,0

```

```

REM Dessin de cercles
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "DESSIN DE CERCLES"
WINDOW OPEN
RANDOMIZE TIME:j=1
LABEL boucle
PRINT AT (1;1) "TAILLE : " j:b=BUTTON(1)+
BUTTON(2)+2:x=XMOUSE:y=YMOUSE
IF b<1 THEN GOTO boucle ELSE ON b GOSUB d
es,tai,tai,fin:GOTO boucle
LABEL tai
j=j+1+j*(j>63)
RETURN
LABEL des
ELLIPSE x;y,j,0.5 COLOR(RND(15)) FILL WIT
H RND(38)
RETURN
LABEL fin
PRINT AT (1;17) "C'est fini..."
r$=INPUT$(1):END

```

Nicomaque

REFERENCES :

Difficulté de programmation : moyenne.

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

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

LE PROGRAMME GW-BASIC :

La ligne 50 permet de confectionner l'entier.

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

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

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

Les lignes 170 à 190 s'occupent de gérer la prochaine partie, selon la réponse du joueur.

POUR JOUER :

On lance le programme en tapant RUN.

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

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

On obtiendra, en réponse, un affichage PERDU ou BRAVO, puis la question LA MEME, UNE AUTRE ou STOP (O/N/M).

Pour faire un nouvel essai, on répondra A, ou M suivi de RETURN.

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic, sauf l'arrêt, obtenu en tapant « F ».

```

10 REM NICOMAUQUE
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF
40 CLS
50 X=INT(RND*100+1)
60 PRINT"RESTE DANS LA DIVISION PAR"
70 PRINT:PRINT TAB(12);"3";TAB(20);"5";TAB(28);"7"
80 FOR I=1 TO 3
90 A=X-(2*I+1)*INT(X/(2*I+1))
100 FOR J=1 TO A
110 COLOR 1:LOCATE 5,J+8*I+3:IF A<>0 THEN PRINT "*"
120 NEXT J
130 NEXT I
140 COLOR 5:LOCATE 11,1:INPUT"VOTRE ESSAI :";N
150 IF N=X THEN COLOR 4:PRINT:PRINT"BRAVO":PLAY "CEG"
160 COLOR 0:IF N<>X THEN PRINT:PRINT"PERDU":PLAY "CCC"
170 PRINT:PRINT"LA MEME, UNE AUTRE OU ST O?" :R$=INPUT$(1)
180 IF R$="A" OR R$="a" THEN 40
190 IF R$="M" OR R$="m" THEN CLS:GOTO 60

```

```

REM Nicomaque

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 FIXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "NICOMAUQUE"
WINDOW OPEN
RANDOMIZE TIME
LABEL debut
x=RND(100)
LABEL retour
CLS:PRINT AT (1;3) POINTS(16) "Reste dans la division par"
FOR i=3 TO 7 STEP 2
PRINT AT (12*i-20;6) POINTS (20) COLOR(7+i) CHR$(48+i)
NEXT i
FOR i=1 TO 3
BOX 192*i-76;142,26,26 COLOR(i) WIDTH 7 ROUNDED
a=x-(2*i+1)*INT(x/(2*i+1))
FOR j=1 TO a
IF a<>0 THEN PRINT AT (2*j+24*i-14;9) COLOR(8+2*i) POINTS(20) "*"
NEXT j
IF a=0 THEN PRINT AT (24*i-14;9) POINTS(20) "-"
NEXT i
INPUT AT (1;13) "Tapez le nombre essayé ( 0 - 100 ) : ",n$:n=VAL(n$)
IF n=x THEN PRINT AT (1;15) "BRAVO" ELSE PRINT AT (1;15) "PERDU"
PRINT AT (1;17) "Même nombre, un autre ou fin ( M/A/F )":r$=INPUT$(1)
r$=UPPER$(r$):ON INSTR(1,"MAF",r$) GOTO retour,debut,fin
LABEL fin
PRINT AT (1;19) "Le nombre à trouver était " x
r$=INPUT$(1):END

```

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 GW-BASIC :

Une table B est dimensionnée à 16, en ligne 40, pour recevoir les numéros des cases déjà visitées.

La boucle I (50-80) dessine le terrain de jeu.

La position initiale du locataire est tirée au hasard en ligne 90.

Le numéro de la case à visiter est donné en ligne 100.

Les lignes 110 à 180 contrôlent les diverses réponses possibles.

La boucle I (150-170) dessine les cases visitées.

La ligne 200 dessine le locataire, effacé en ligne 260.

Les lignes 240 et 250 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.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM L'INDESIRABLE
20 SCREEN 0,1:WIDTH 40:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF:CLS
40 A$=CHR$(178):B$=A$+A$+A$:DIM B(16)
50 FOR I=1 TO 16
60 LOCATE 2+4*INT((I-1)/4),4+4*(I-1-4*INT((I-1)/4)):COLOR 7:PRINT B$
70 LOCATE 3+4*INT((I-1)/4),4+4*(I-1-4*INT((I-1)/4)):COLOR 5:PRINT A$:COLOR 2:PRINT CHR$(64+I):COLOR 5:PRINT A$
80 NEXT I
90 X=INT(RND*16+1):S=1:K=0
100 LOCATE 21,1:COLOR 7:INPUT"CASE VISITEE : ";C$:C=ASC(C$)-64+32*(C$>="a")
110 IF B(C)=1 THEN LOCATE 23,1:PRINT"DEJA VISITEE":PLAY"CCC":GOTO 100
120 IF C=X THEN LOCATE 23,1:PRINT"VOUS L'AVEZ EU EN ";S: " COUPS . BRAVO ";:PLAY"CEG":K=1:GOTO 200
130 IF X=5=C OR X=4=C OR X=3=C OR X=1=C OR X+1=C OR X+3=C OR X+4=C OR X+5=C THEN GOTO 200
140 LOCATE 23,1:PRINT"IL N'EST PAS LA":;:PLAY"CDE"
150 S=S+1:B(C)=1:FOR I=1 TO 16
160 LOCATE 2+4*INT((I-1)/4),4+4*(I-1-4*INT((I-1)/4)):COLOR 7-6*B(I):PRINT B$
170 NEXT I
180 IF S=9 THEN LOCATE 23,1:PRINT"VOUS AVEZ PERDU . IL ETAIT EN ";CHR$(X+64):K=1:GOTO 200
190 LOCATE 23,1:PRINT"
";: GOTO 100
200 LOCATE 1+4*INT((X-1)/4),5+4*(X-1-4*INT((X-1)/4)),0:COLOR 4:PRINT CHR$(2)
210 FOR I=1 TO 800:NEXT I
220 IF K=1 THEN PLAY"GEC":END
230 LOCATE 1+4*INT((X-1)/4),5+4*(X-1-4*INT((X-1)/4)),0:PRINT " "
240 Z=INT(10*RND+1)-5:IF ABS(Z)=2 THEN 240
250 X=X+Z:IF X>16 OR X<1 THEN X=X-Z:GOTO 240
260 GOTO 150

```

```

REM L'indésirable

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 FIXED
USER SPACE 640,200
WINDOW TITLE "L'INDESIRABLE"
WINDOW OPEN
DIM b(16):FOR i=1 TO 16:b(i)=0:NEXT i
c$="XXXXX":b$=" " :a$=" " :RANDOMIZE TIME
DEF FNa(i)=6+8*(i-1-4*INT((i-1)/4))
DEF FNb(i)=3+5*INT((i-1)/4)
BOX 20;20;280,170 COLOR(2) FILL WITH 9
FOR i=1 TO 16
PRINT AT (FNa(i);FNb(i)) a$ AT (FNa(i);FNb(i)+1) b$+CHR$(64+i)+b$
NEXT i
x=RND(16):s=1:t$=STRING$(30,32)
LABEL debut
PRINT CHR$(7)
PRINT AT (40;2) "Case visitée : ";v$=INPUT$(1)
v$=UPPER$(v$):PRINT v$:c=ASC(v$)-64
IF b(c)=1 THEN PRINT AT (40;4) "Déjà visitée":PRINT CHR$(7):GOTO boucle
IF c=x GOTO fin
IF x=5=c OR x=4=c OR x=3=c OR x=1=c OR x+1=c OR x+3=c OR x+4=c OR x+5=c GOTO affiche
PRINT AT (40;4) "Il n'est pas là":PRINT CHR$(7):CHR$(7):CHR$(7)
LABEL boucle
FOR i=1 TO 6000:NEXT i
s=s+1:b(c)=1
FOR i=1 TO 16
LOCATE FNa(i);FNb(i):IF b(i)=1 THEN PRINT COLOR(4) c$ ELSE PRINT a$
NEXT i
IF s=9 GOTO fin
PRINT AT (40;2) t$ AT (40;4) t$:GOTO debut
LABEL affiche
PRINT AT (FNa(x);FNb(x)) COLOR(15) b$;CHR$(234);b$
FOR p=1 TO 6000:NEXT:PRINT CHR$(7)
PRINT AT (FNa(x)+2;FNb(x)) " "
LABEL tirage
z=RND(10)-5:IF ABS(z)=2 GOTO tirage
x=x+z:IF x>16 OR x<1 THEN x=x-z:GOTO tirage
GOTO boucle

LABEL fin
PRINT AT (FNa(x)+2;FNb(x)) COLOR(11) CHR$(234)
IF c=x THEN PRINT AT (40;6) "Vous l'avez eu en "s" coups"
IF s=9 THEN PRINT AT (40;6)"Perdu. Il était en "CHR$(64+x)
FOR i=1 TO 10:PRINT CHR$(7):NEXT i
r$=INPUT$(1):END

```

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 bleus et de 3 moutons rouges, 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 GW-BASIC :

La boucle I (40-70) initialise une table A (I), qui contient les positions des moutons. La boucle I (90-140) dessine le terrain de jeu et les moutons.

Les lignes 140 à 190 s'occupent du texte et des entrées.

La ligne 190 contrôle la validité du coup. Le changement de joueur est réalisé en ligne 230.

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, puis le numéro de la case d'arrivée séparés par une virgule, suivi de RETURN.

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

EXTENSIONS POSSIBLES :

On commencera par rajouter un contrôle du joueur gagnant.

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

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic, mais les cases de début et de fin sont entrées consécutivement (et non séparées par une virgule).

```
10 REM LE JEU DE SAUTE-MOUTON
20 SCREEN 0,1:WIDTH 40:KEY OFF:COLOR 0,7
:CLS
30 DIM A(8):A(1)=0:A(8)=0
40 FOR I=1 TO 3
50 A(I+1)=1
60 A(I+4)=-1
70 NEXT I
80 H=1
90 FOR I=1 TO 8
100 LOCATE 8,4*I:PRINT I
110 LOCATE 6,4*I+1:PRINT " "
120 IF A(I)=1 THEN LOCATE 6,4*I+1:COLOR
4:PRINT"#":COLOR 0,7
130 IF A(I)=-1 THEN LOCATE 6,4*I+1:COLOR
1:PRINT"%":COLOR 0,7
140 NEXT I
150 J=1-(H=-1):BEEP
160 LOCATE 13,1:PRINT SPACE$(30):LOCATE
15,1:PRINT SPACE$(30)
170 LOCATE 13,1:PRINT"JOUEUR ";:COLOR 7-
3*J:PRINT J
180 COLOR 0:PRINT:INPUT"DEBUT , FIN :";D
,F
190 IF D<>INT(D) OR F<>INT(F) OR D<1 OR
D>8 OR F<1 OR F>8 THEN 160
200 IF A(D)<>H OR A(F)<>0 THEN 160
210 A(D)=0
220 A(F)=H
230 H=-H
240 GOTO 90
```

```
REM Le jeu de saute-mouton

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "LE JEU DE SAUTE-MOUTON"
WINDOW OPEN
a=2:b=4:a(1)=0:a(8)=0
FOR i=1 TO 3:a(i+1)=1:a(i+4)=-1:NEXT
h=1
REPEAT
GOSUB ecrit
j=1-(h=-1):k=(a AND j=1)+(b AND j=2)
REPEAT
PRINT AT (1;15) "
"
PRINT AT (1;13) "Joueur " COLOR(k) CHR$(4
8+j)+ " "+CHR$(27)+CHR$(2+j)
INPUT AT (1;15) "Case de début : ",d$:d=V
AL(d$)
INPUT AT (24;15) "Case de fin : ",f$:f=VA
L(f$)
UNTIL a(d)=h AND a(f)=0
a(d)=0:a(f)=h:h=-h
UNTIL a(1)+a(2)+a(3)=-3 OR a(6)+a(7)+a(8)
=3
GOSUB ecrit
PRINT AT (1;18) "Le joueur " COLOR(k) CHR
$(48+j) COLOR(1) " gagne"
r$=INPUT$(1):END

LABEL ecrit
FOR i=1 TO 8
PRINT AT (4*i;8) COLOR(14) CHR$(48+i)
PRINT AT (4*i;6) " "
IF a(i)=1 THEN PRINT AT (4*i;6) COLOR(a)
CHR$(27)+CHR$(3)
IF a(i)=-1 THEN PRINT AT (4*i;6) COLOR(b)
CHR$(27)+CHR$(4)
NEXT i
RETURN
```

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 GW-BASIC :

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 60).

Les lignes 110-120 vont contrôler qu'il reste suffisamment d'étoiles sur la ligne indiquée.

La boucle J (140-170) supprime les étoiles ; la ligne 200 teste la fin de partie.

POUR JOUER :

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

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

EXTENSIONS POSSIBLES :

Outre une amélioration de la qualité du 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...

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM LA COURSE AU 20
20 SCREEN 0,1:WIDTH 40:COLOR 4,7:CLS
30 FOR I=1 TO 4
40 LOCATE 2*I+1,1:PRINT I;" *****
*****"
50 NEXT I
60 DIM F(4):B=1:T=0
70 COLOR 4*B-3:LOCATE 13,1:PRINT"JOUEUR
";B
80 COLOR 0:PRINT:INPUT"LIGNE :";L
90 IF L>4 THEN 70
100 PRINT:INPUT"CASES :";C
110 IF C+F(L)>20 THEN 70
120 F(L)=F(L)+C
130 T=T+C
140 FOR J=F(L)-C TO F(L)
150 LOCATE 2*L+1,26-J:PRINT" ":SOUND 800
,1
160 FOR I=1 TO 200:NEXT I
170 NEXT J
180 B=B+1
190 IF B=3 THEN B=1
200 IF T<80 THEN 70
210 COLOR 4*B-3:PRINT"LE JOUEUR ";B;" GA
GNE":PLAY"CGE"

```

```

REM La course au 20

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "LA COURSE AU VINGT"
WINDOW OPEN
FOR i=1 TO 4
PRINT AT (1;3*i) POINTS(20) i:f(i)=0
PRINT AT (6;3*i) POINTS(20) COLOR(9+i) ST
RING$(20,"*")
NEXT i
RANDOMIZE(TIME)
t=0:b=RND(2)
REPEAT
PRINT AT (4;14) EFFECTS(1) "Joueur " COL
OR(b+13) POINTS(16) b
LABEL choix
PRINT AT (4;16) STRING$(30,32) AT (4;18)
STRING$(30,32)
INPUT AT (4;16) "Ligne : ",1$:1=VAL(1$)
IF 1<>INT(1) OR 1<1 OR 1>4 GOTO choix
INPUT AT (4;18) "Cases : ",c$:c=VAL(c$)
IF c+f(1)>20 OR c<0 OR c<>INT(c) GOTO choi
x
f(1)=f(1)+c:t=t+c
FOR j=f(1)-c TO f(1)
PRINT AT (46-2*j;3*1) POINTS(20) " "
NEXT j
b=b+1:IF b=3 THEN b=1
UNTIL t=80
PRINT AT (4;20) EFFECTS(1) COLOR(b+13) "L
e joueur " b " gagne"
r$=INPUT$(1)
END

```

L'électricien fou

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez difficile.

LE JEU :

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

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

LE PROGRAMME GW-BASIC :

Une table A, en ligne 40, contiendra l'état (allumé ou éteint) des 10 fenêtres.

La boucle I (50-70) et l'autre boucle I (90-140) dessinent l'immeuble dans son état initial.

La ligne 150 regarde si tout est éteint, auquel cas l'on ira en ligne 290 afficher le nombre d'essais.

La ligne 160 attend l'indication de l'interrupteur à manœuvrer.

Les lignes 170 à 280 définissent le résultat de la manœuvre.

POUR JOUER :

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

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

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

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

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM L'ELECTRICIEN FOU
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF
40 DIM A(10):M=0:CLS
50 FOR I=1 TO 7
60 COLOR 0,1:LOCATE I+1,1:PRINT"
"
70 NEXT I
80 T=0
90 FOR I=1 TO 10
100 T=T+A(I)
110 LOCATE 4,2*I+6:COLOR 0,4:PRINT" "
120 IF A(I)=0 THEN COLOR 0,7:LOCATE 4,2*
I+6:PRINT" "
130 COLOR 1,6:LOCATE 7,2*I+5:PRINT I-1
140 NEXT I
150 IF T=10 THEN 290
160 COLOR 0,7:LOCATE 15,1:BEEP:INPUT"INT
ERRUPTEUR NUMERO :";N
170 N=N+1
180 M=M+1
190 IF A(N)=1 THEN 270
200 A(N)=1
210 R=TAN(RND+N/RND-N)-SIN(RND/N)+336*SI
N(8*N)
220 N=INT(10*(R-INT(R)))
230 IF N=0 THEN N=INT(RND*10+1)
240 IF A(N)=1 THEN 270
250 A(N)=1
260 GOTO 80
270 A(N)=0
280 GOTO 210
290 COLOR 0,7:PRINT:PRINT"TROUVE EN ";M;
" ESSAIS"

```

```

REM L'électricien fou

STREAM #1:SCREEN GRAPHICS 640 FIXED,300 F
IXED
USER SPACE 640,300
WINDOW FULL ON
WINDOW TITLE "L'ELECTRICIEN FOU"
WINDOW OPEN
RANDOMIZE TIME
m=0:k=1
BOX 96;220,436,80 COLOR(12) FILL WITH 7
LABEL début
t=0
FOR i=1 TO 10
t=t+a(i):BOX 80+40*i;256,32,32 COLOR(12)
FILL WITH 7
BOX 80+40*i;256,32,32 WIDTH 3 COLOR(14)
IF a(i)=0 THEN BOX 80+40*i;256,32,32 COLO
R(6) FILL WITH 8
PRINT AT (5*i+12;5) COLOR(10) POINTS(16)
CHR$(47+i)
NEXT i
WHILE t<>10
PRINT AT (1;16) STRING$(40,32) AT (1;16)
"Interrupteur numéro : ";
n$=INPUT$(1):n=VAL(n$):n=n+1:m=m+1
IF a(n)=1 THEN a(n)=0:GOTO calcul
a(n)=1
LABEL calcul
r=TAN(RND+n/RND-n)-SIN(RND/n)+336*SIN(8*n
)
n=INT(ABS(10*(r-INT(r))))
IF n=0 THEN n=RND(10)
IF a(n)=1 THEN a(n)=0:GOTO calcul
a(n)=1:GOTO début
WEND
PRINT CHR$(7):PRINT CHR$(7):PRINT CHR$(7)
PRINT AT (1;18) "Trouvé en " m " essais"
r$=INPUT$(1)
END

```

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 GW-BASIC :

On utilisera un tableau N(7), dimensionné en ligne 40, pour enregistrer les résultats.

Le tirage des nombres est effectué dans une boucle I (60-120), une deuxième boucle J (90-110) étant utilisée pour contrôler qu'il n'y a pas deux tirages identiques.

Le tirage (ou le retraitage) est effectué en ligne 70.

La boucle I (170-190) affiche le résultat, et la ligne 200 affiche le numéro complémentaire.

Les lignes 210 et 220 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.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM JEU DE LOTO
20 SCREEN 0,1:WIDTH 40
30 RANDOMIZE TIMER:KEY OFF
40 DIM N(7):A$=CHR$(200):B$=CHR$(201):C$
=CHR$(205):D$=CHR$(186):E$=CHR$(187):F$=
CHR$(188)
50 COLOR 7,0:CLS
60 FOR I=1 TO 7
70 N(I)=INT(RND*49+1)
80 IF I=1 THEN 120
90 FOR J=1 TO I-1
100 IF N(I)=N(J) THEN 70
110 NEXT J
120 NEXT I
130 G$="":FOR I=1 TO 16:G$=G$+C$:NEXT I
140 COLOR 12:LOCATE 3,11:PRINT B$+G$+E$
150 LOCATE 4,11:PRINT D$:COLOR 7:PRINT
" TIRAGE DU LOTO ";:COLOR 12:PRINT D$
160 LOCATE 5,11:PRINT A$+G$+F$
170 FOR I=1 TO 6
180 COLOR INT(RND*15+1):LOCATE 13,6*I-4:
PRINT N(I)
190 NEXT I
200 LOCATE 20,1:COLOR 0,1:PRINT"NO. COMP
LEMENTAIRE :";:COLOR 4:PRINT N(7)
210 LOCATE 22,1:COLOR 5,6:PLAY"CDE":PRIN
T "VOULEZ-VOUS UN AUTRE TIRAGE ( O/N )"
220 R$=INPUT$(1):IF R$="O" OR R$="o" THE
N 50

```

```

REM Jeu de Loto

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "JEU DE LOTO"
WINDOW OPEN
RANDOMIZE TIME
REPEAT
CLS
BOX 0;0,640,200 COLOR(6) FILL ONLY WITH 8
FOR i=1 TO 7
LABEL tirnombre
n(i)=RND(49):n$(i)=MID$(STR$(n(i)),2)
FOR j=1 TO i-1
IF n(i)=n(j) GOTO tirnombre
NEXT j,i
a$="TIRAGE DU LOTO"
BOX 100;150,440,40 COLOR(5) WIDTH 5 FILL
WITH 4
PRINT AT (26;4) COLOR(2) POINTS(20) MODE(
2) a$
FOR i=1 TO 6
c=RND(8)+7
FOR j=1 TO LEN(n$(i))
PRINT AT (12*i-6+2*j;10) COLOR(c) POINTS(
20) CHR$(ASC(MID$(n$(i),j,1)))
NEXT j,i
PRINT AT (1;15) MODE(2) POINTS(16) "Numér
o complémentaire : " COLOR(4) n(7)
PRINT CHR$(7):PRINT CHR$(7):PRINT CHR$(7)
PRINT AT (1;18) MODE(4) "Voulez-vous un a
utre tirage (O/N)"
r$=INPUT$(1):r$=UPPER$(r$)
UNTIL r$="N"
END

```

Visions

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : dessin.

Difficulté du jeu : facile.

LE JEU :

Le PC 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 GW-BASIC :

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

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

POUR JOUER :

On débute l'exécution du programme par RUN, ce qui amène l'affichage de la question QUEL DESSIN. On répond en entrant un mot quelconque qui sera transformé en dessin par le PC dès que l'on aura frappé RETURN. Pour recommencer, il suffira de frapper une touche quelconque, et CTRL et Break pour finir.

EXTENSIONS POSSIBLES :

La même idée peut être reprise en introduisant des éléments de symétrie dans les dessins, en mettant plusieurs couleurs. 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 PC deviendra de plus en plus savant. On peut aussi réaliser un jeu à deux joueurs dans lequel PC vous demande de trouver le mot qui correspond à un certain dessin, etc.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic. On arrête en tapant RETURN.


```

10 REM VISIONS
20 SCREEN 0,1:WIDTH 40:RANDOMIZE TIMER:C
OLOR 7,0:CLS
30 INPUT"QUEL DESSIN :";R$
40 FOR I=1 TO LEN(R$)
50 N=9*ASC(MID$(R$,I,1))-340
60 J=1
70 S=N/2-INT(N/2)
80 N=INT(N/2):SOUND INT(RND*3000+600),2
90 IF S<>0 THEN LOCATE I+S,J+15:COLOR IN
T(RND*15+1):PRINT CHR$(219):COLOR 7,0
100 J=J+1
110 IF J<10 THEN 70
120 NEXT I
130 PRINT:PRINT
140 INPUT"APPUYER SUR - ENTREE -";R$
150 GOTO 20

```

```

REM Visions
STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "VISIONS"
WINDOW OPEN
REPEAT
t=0
REPEAT
PRINT AT (1;1) EFFECTS(1) "Texte ( < 11
lettres )"
INPUT AT (1;3) "> ",r$:r$=UPPER$(r$)
FOR i=1 TO LEN(r$):t=t+ASC(MID$(r$,i,1)):
NEXT i
CLS:RANDOMIZE (t)
UNTIL LEN(r$)<11
FOR i=1 TO LEN(r$)
j=1:n=9*ASC(MID$(r$,i,1))-340
WHILE j<10
s=(n/2)-INT(n/2):n=INT(n/2):f=RND(15):PRI
NT CHR$(7);
IF s<>0 THEN PRINT AT (4*j+20;2*i) COLOR(
f) POINTS(20) MODE(4) " ";
j=j+1
FOR k=1 TO 400:NEXT k
WEND
NEXT i
PRINT AT (70;1) ANGLE(270)"Dessin de ";r$
;
UNTIL r$=""
END

```

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 PC (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 GW-BASIC :

Deux tables (B et C) sont utilisées en ligne 50, pour enregistrer l'état de la population. Les boucles F (230-300) et G (240-290) sont utilisées en début de jeu pour définir la population initiale (lignes 260 et 270). Le calcul de la nouvelle population est ensuite réalisé à l'aide d'une quadruple boucle : 80-210, 90-200, 110-160, 120-150. La ligne 130 réalise la gestion du nouvel état suivant les règles.

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

La ligne 310 permet de recommencer.

POUR JOUER :

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

EXTENSIONS POSSIBLES :

Il sera facile, 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.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM LE JEU DE LA VIE (AUTOMATIQUE)
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF:CLS
40 INPUT"DIMENSION :";A
50 DIM B(A,A):DIM C(A,A)
60 I=0
70 GOTO 220
80 FOR F=1 TO A
90 FOR G=1 TO A
100 S=0
110 FOR I=-1 TO 1
120 FOR J=-1 TO 1
130 IF F+J=0 OR G+I=0 OR F+J=A+1 OR G+I=
A+1 THEN 150
140 S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 S=2*S-B(F,G)
180 C(F,G)=0
190 IF S>=5 AND S<=7 THEN C(F,G)=1
200 NEXT G
210 NEXT F
220 CLS:K=0
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I<>0 THEN 270
260 IF RND<.5 THEN C(F,G)=1
270 LOCATE 2*F+1,2*G+1:COLOR 4:IF C(F,G)
=1 THEN K=1:PRINT "*" :SOUND INT(RND*3000
+400),1
280 B(F,G)=C(F,G)
290 NEXT G
300 NEXT F
310 IF K=1 THEN 80 ELSE PRINT "C'est fin
i....":PLAY "CGE"

```

```

10 REM LE JEU DE LA VIE (MANUEL)
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 KEY OFF:CLS
40 INPUT"DIMENSION :";A
50 DIM B(A,A):DIM C(A,A)
60 I=0
70 GOTO 220
80 FOR F=1 TO A
90 FOR G=1 TO A
100 S=0
110 FOR I=-1 TO 1
120 FOR J=-1 TO 1
130 IF F+J=0 OR G+I=0 OR F+J=A+1 OR G+I=
A+1 THEN 150
140 S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 S=2*S-B(F,G)
180 C(F,G)=0
190 IF S>=5 AND S<=7 THEN C(F,G)=1
200 NEXT G
210 NEXT F
220 CLS:BEEP:K=0
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I<>0 THEN 280
260 LOCATE 22,1:COLOR 1:PRINT"LIGNE ";F;
";COLONNE ";G;"TAPER 1 OU 0"
270 LOCATE 2*F+1,2*G+1,1:COLOR 0:INPUT "
",C(F,G)
280 LOCATE 2*F+1,2*G+1:COLOR 4:IF C(F,G)
=1 THEN PRINT "*" :K=1 ELSE PRINT " "
290 B(F,G)=C(F,G)
300 NEXT G
310 NEXT F
320 IF K=1 THEN 80 ELSE PRINT "C'est fin
i....":PLAY "CGE"

```

```

REM Le jeu de la vie:version automatique
STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "JEU DE LA VIE AUTOMATIQUE"
WINDOW OPEN
RANDOMIZE TIME
INPUT AT (1;1) "Dimension du terrain : ",
a$:a=VAL(a$)
DIM b(a,a),c(a,a):i=0:GOTO affiche
LABEL calcul
FOR f=1 TO a
FOR g=1 TO a
s=0
FOR i=-1 TO 1
FOR j=-1 TO 1
IF f+j<>0 AND g+i<>0 AND f+j<>a+1 AND g+i
<>a+1 THEN s=s+b(f+j,g+i)
NEXT j,i
s=2*s-b(f,g):c(f,g)=0
IF s>=5 AND s<=7 THEN c(f,g)=1
NEXT g,f
LABEL affiche
CLS:k=0
FOR f=1 TO a
FOR g=1 TO a
IF RND<0.5 AND i=0 THEN c(f,g)=1
IF c(f,g)=1 THEN k=1:PRINT AT (2*g+6;2*f-
1) "*"
b(f,g)=c(f,g)
NEXT g,f
PRINT CHR$(7);IF k=1 GOTO calcul
FOR i=1 TO 10:PRINT CHR$(7);NEXT i
PRINT AT (30;12) "C'est fini....."
r$=INPUT$(1):END

```

```

REM Le jeu de la vie ( version manuelle )
STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "JEU DE LA VIE MANUEL"
WINDOW OPEN
RANDOMIZE TIME
INPUT AT (1;1) "Dimension du terrain : ",
a$:a=VAL(a$)
DIM b(a,a),c(a,a):i=0:GOTO affiche
LABEL calcul
FOR f=1 TO a
FOR g=1 TO a
s=0
FOR i=-1 TO 1
FOR j=-1 TO 1
IF f+j<>0 AND g+i<>0 AND f+j<>a+1 AND g+i
<>a+1 THEN s=s+b(f+j,g+i)
NEXT j,i
s=2*s-b(f,g):c(f,g)=0
IF s>=5 AND s<=7 THEN c(f,g)=1
NEXT g,f
LABEL affiche
CLS:k=0
FOR f=1 TO a
FOR g=1 TO a
IF i<>0 GOTO suite
PRINT AT (40;2) "Ligne " f AT (50;2) "Col
onne " g AT (2*g+6;2*f-1) "-"
PRINT AT (40;6) "Taper 0 ou 1 : ";r$=INP
UT$(1):c(f,g)=VAL(r$)
LABEL suite
IF c(f,g)=1 THEN k=1:PRINT AT (2*g+6;2*f-
1) "*"
b(f,g)=c(f,g)
NEXT g,f
PRINT CHR$(7);IF k=1 GOTO calcul
FOR i=1 TO 10:PRINT CHR$(7);NEXT i
PRINT AT (30;12) "C'est fini....."
r$=INPUT$(1):END

```

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 GW-BASIC :

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.

POUR JOUER :

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

Le résultat est très rapidement affiché.

On tape une touche quelconque 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.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM JOUR DE LA SEMAINE
20 SCREEN 0,1:WIDTH 40:COLOR 0,7:KEY OFF
:CLS
30 INPUT"JOUR.....":J
40 IF J=0 THEN END
50 PRINT:INPUT"MOIS.....":M
60 PRINT:INPUT"ANNEE.....":A
70 B=A
80 S=INT(A/100)-INT(A/400)
90 IF A<=1582 THEN S=2
100 M$="JANVFEVRMARS AVRIMAI JUINJUILAOUT
SEPTOCTONOVEDECE"
110 M$=MID$(M$,4*M-3,4)
120 IF M<3 THEN M=M+12:A=A-1
130 S=J-1+INT(13*(M+1)/5)+INT(5*A/4)-S
140 S=S-7*INT(S/7)
150 J$="DIMANCHELUNDI MARDI MERCREDI
JEUDI VENDREDISAMEDI "
160 J$=MID$(J$,8*S+1,8)
170 LOCATE 16,1:PRINT"LE ";J;" ":COLOR
1:PRINT M$;B;:COLOR 0:PRINT " ETAIT UN "
:COLOR 5:PRINT J$:PLAY"CGE"
180 COLOR 4:PRINT:PRINT:PRINT"APPUYEZ SU
R UNE TOUCHE":R$=INPUT$(1)
190 RUN

```

```

REM Jour de la semaine

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "JOUR DE LA SEMAINE"
WINDOW OPEN
LABEL début
CLS
INPUT AT (1;2) "Jour.....":j$:j=VAL(
j$)
IF j=0 THEN END
INPUT AT (1;4) "Mois.....":m$:m=VAL(
m$)
INPUT AT (1;6) "Année.....":a$:a=VAL(
a$)
b=a:s=INT(a/100)-INT(a/400)
IF a<=1582 THEN s=2
m$="JANVIER FEVRIER MARS AVRIL M
AI JUIN "
m$=m$+"JUILLET AOUT SEPTEMBRE OCTOBRE
NOVEMBRE DECEMBRE "
m$=MID$(m$,9*m-8,9):r$=""
FOR i=1 TO LEN(m$)
IF MID$(m$,i,1)<>" " THEN r$=r$+MID$(m$,i
,1)
NEXT i
IF m<3 THEN m=m+12:a=a-1
s=j-1+INT(13*(m+1)/5)+INT(5*a/4)-s
s=s-7*INT(s/7)
j$="DIMANCHELUNDI MARDI MERCREDIJEUDI
VENDREDISAMEDI "
p$=MID$(j$,8*s+1,8)
PRINT AT (1;13) POINTS(16) "Le " COLOR(2)
j COLOR(4) r$ COLOR(14) b;
PRINT POINTS(16) COLOR(1) " était un " PO
INTS(20) COLOR(15) p$
PRINT:PRINT:PRINT"Appuyez sur une touche"
r$=INPUT$(1)
GOTO début

```

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 GW-BASIC :

Le nombre de joueurs est tout d'abord entré en ligne 40.

La ligne 120 affiche le numéro de joueur et le nombre d'essais déjà effectués.

Un sous-programme placé en ligne 330, utilisé 3 fois dans la boucle J (190-240), réalise le lancer de dés.

Les retirages sont effectués par les lignes 300, 310 et 320.

La variable R\$ gère les 2^e et 3^e tirages. L'impression du résultat est réalisée dans tous les cas par les lignes 130-140.

Les tests pour le nombre d'essais et pour le changement de joueur sont effectués aux lignes 250 et 270.

POUR JOUER :

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

Si l'on ne désire pas refaire de tirage, il suffit de frapper RETURN. La main passe alors au joueur suivant.

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

Les joueurs suivants procéderont de la même façon.

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM 421
20 SCREEN 0,1:COLOR 1,7:WIDTH 40:CLS
30 RANDOMIZE TIMER
40 INPUT"NOMBRE DE JOUEURS :";N
50 P=1
60 I=1
70 K=0:CLS
80 R$="ABC"
90 GOTO 190
100 K=K+1
110 LOCATE 1,17:COLOR 3,1 :PRINT "-421-"
120 COLOR 0,7:LOCATE 4,1:PRINT"Joueur NO
. "I;" - ESSAI "K;" - PARTIE "P
130 LOCATE 7,7:PRINT A :LOCATE 7,17:PRIN
T B:LOCATE 7,27:PRINT C
140 LOCATE 9,8:PRINT"A":LOCATE 9,18:PRIN
T"B":LOCATE 9,28:PRINT"C"
150 R$="":LOCATE 14,1:PRINT"
"
160 LOCATE 14,1:INPUT"ABC :";R$
170 IF R$="" THEN 260
180 R$=R$+"DDD"
190 FOR J=1 TO 3
200 GOSUB 330
210 X$=MID$(R$,J,1)
220 X=ASC(X$)-64+32*(X$>="a")
230 ON X GOSUB 300,310,320,330
240 NEXT J
250 IF K<3 THEN 100
260 I=I+1
270 IF I<=N THEN 70
280 P=P+1
290 GOTO 60
300 A=D:RETURN
310 B=D:RETURN
320 C=D:RETURN
330 D=INT(RND*6+1):SOUND 200*D+400,2:RET
URN

```

```

REM Quatre cent vingt et un

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "421"
WINDOW OPEN
RANDOMIZE(TIME):p=1
REPEAT
INPUT AT (1;3) "Nombre de joueurs ( <=8 )
: ",n$:n=VAL(n$)
UNTIL n>0 AND n<9 AND n=INT(n)
REPEAT
i=1
REPEAT
CLS:k=0:r$="ABC"
PRINT AT (20;3) POINTS(20) COLOR(14) "-42
1-"
GOTO tirage
LABEL essais
r$="":k=k+1
PRINT AT (1;5) EFFECTS(1) "Joueur No. " C
OLOR(16-i) i;
PRINT EFFECTS(1) " - Essai " k " - Part
ie " p
PRINT AT (7;9) POINTS(20) COLOR(a) a AT (
17;9) COLOR(b) b AT (27;9) COLOR(c) c
PRINT AT (8;12) POINTS(16) " A" AT (17;12
) " B" AT (27;12) " C"
PRINT AT (1;15) STRING$(30,32)
PRINT AT (1;15) EFFECTS(1) "A B C : ";:I
NPUT "",r$
r$=UPPER$(r$):IF LEN(r$)=0 GOTO suivant
r$=r$+"DDD"
LABEL tirage
FOR j=1 TO 3
GOSUB tirde
x$=MID$(r$,j,1):x=ASC(x$)-64
ON x GOSUB premier,deuxieme,troisieme,tir
de
NEXT j
IF k<3 GOTO essais
LABEL suivant
i=i+1
UNTIL i>n
p=p+1
PRINT AT (1;18) "On continue ( O/N ) ?":r
$=INPUT$(1)
UNTIL UPPER$(r$)="N"
END

LABEL premier:a=d:RETURN

LABEL deuxieme:b=d:RETURN

LABEL troisieme:c=d:RETURN

LABEL tirde:d=RND(6):PRINT CHR$(7);
RETURN

```

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 GW-BASIC :

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

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

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

La réponse sera entrée en ligne 190.

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

La ligne 210 affiche la note.

Les lignes 230 et 240 regardent si l'on a gagné ou perdu et affichent le message correspondant.

La ligne 250 permet de faire un nouvel essai.

POUR JOUER :

On lance le programme par RUN.

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

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

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

On doit taper une touche quelconque pour continuer, et s'arrêter en tapant CTRL et Break.

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.


```

10 REM MUSIQUE
20 SCREEN 0,1:WIDTH 40:COLOR 1,7
30 RANDOMIZE TIMER:KEY OFF
40 CLS
50 FOR I=1 TO 5
60 LOCATE 2*I+5,5:FOR J=1 TO 35:PRINT CH
R$(219);:NEXT J
70 NEXT I
80 FOR I=1 TO 14
90 LOCATE I+3,1:COLOR 4
100 IF I<=9 THEN PRINT I
110 IF I>9 THEN PRINT " ";CHR$(I+55)
120 NEXT I
130 N$="BAGFEDC":P$="SI LA SOLFA MI RE D
O "
140 X=INT(RND*14+1)
150 T=X-1-7*INT((X-1)/7)
160 COLOR 0:LOCATE 19,1:PRINT"NOTE ";MID
$(P$,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")+32*(L$>"a")
210 LOCATE X+1,16:PRINT CHR$(179):LOCATE
X+2,16:PRINT CHR$(179):LOCATE X+3,15:PR
INT"O"
220 LOCATE 21,26
230 IF L=X THEN COLOR 12:PRINT"BRAVO":IF
X>7 THEN PLAY "O3"+MID$(N$,T+1,1) ELSE
PLAY "O4"+MID$(N$,T+1,1)
240 IF L<>X THEN COLOR 5:PRINT"PERDU"
250 COLOR 1:PRINT:INPUT"TAPER - ENTREE -
POUR CONTINUER";R$
260 GOTO 40

```

REM Musique

```

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "MUSIQUE"
WINDOW OPEN
RANDOMIZE TIME:p=0:s=0
a$=STRING$(70,CHR$(95)):b$=CHR$(124)
REPEAT
CLS
FOR i=1 TO 5
LINE 20;18*i+69,400;18*i+69 COLOR(3)
NEXT i
FOR i=1 TO 14
LOCATE 1;i+1:IF i<=9 THEN PRINT CHR$(48+i
) ELSE PRINT CHR$(55+i)
NEXT i
n$="SI LA SOLFA MI RE DO ":x=RND(14)
t=x-1-7*INT((x-1)/7):s=s+1
PRINT AT (1;16) "Note " MID$(n$,3*t+1,3)
" ";
IF x<=7 THEN PRINT "aigue" ELSE PRINT "gr
ave"
LABEL saisie:PRINT AT (31;18) STRING$(20,
32)
INPUT AT (1;18) "Donnez le numéro de la l
igne : ",1$:1$=UPPER$(1$)
IF INSTR("123456789ABCDE",1$)=0 GOTO sais
ie
l=(ASC(1$)-48 AND 1$<":")+ (ASC(1$)-55 AND
1$>"A")
LINE 190;209-9*x,190;195-9*x COLOR(2) WID
TH(3)
CIRCLE 185;195-9*x,5 COLOR(2) WIDTH(3)
LOCATE 40;18
IF 1<>x THEN PRINT COLOR(14) POINTS(20) "
PERDU" ELSE PRINT COLOR(14) POINTS(20) "B
RAVO":p=p+1
PRINT AT (1;20) p " résultats justes sur
" s " essais"
PRINT AT (1;21) "On continue ( O/N ) ?";:
r$=INPUT$(1)
UNTIL UPPER$(r$)="N"
END

```

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 PC 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 GW-BASIC :

Deux variables A\$ et B\$ sont dimensionnées à 15 en ligne 50 : 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 (60-80) lit les lignes DATA (160-180) concernant les préfixes suivis des suffixes.

La ligne 90 tire au hasard préfixe et suffixe.

Le mot obtenu en ligne 100 est affiché par la boucle L (110-130) et K (120) lettre par lettre, puis on recommence (ligne 150).

POUR JOUER :

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

Taper CTRL et Break pour sortir du programme.

EXTENSIONS POSSIBLES :

On pourra très facilement augmenter la liste des mots disponibles.

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM MACHINAMOT
20 SCREEN 0,1:WIDTH 40:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF
40 CLS
50 DIM A$(15),B$(15)
60 FOR I=1 TO 15
70 READ A$(I),B$(I)
80 NEXT I
90 I=INT(RND*15+1):J=INT(RND*15+1)
100 C$=A$(I)+B$(J)
110 FOR L=1 TO LEN(C$)
120 FOR K=1 TO 26-L:LOCATE 23,K:COLOR IN
T(RND*15+1):PRINT " "+MID$(C$,LEN(C$)-L+
1,1):NEXT
130 SOUND INT(RND*2000)+400,2:NEXT L
140 FOR S=1 TO 1200:NEXT S:PRINT
150 GOTO 90
160 DATA ANDRO,CRATE,COSMO,PHILE,MICRO,P
HOBE,HELIO,TROPE,SEXO,GYNE,STEREO,MANE
170 DATA PALED,NAUTE,MACRO,PHAGE,HYDRO,F
UGE,STENO,PATHE,PATHO,CARDE,OLED,DERME
180 DATA MYTHO,SCOPE,HOMEO,VORE,OCTO,PHO
NE

```

```

REM Machinamot

STREAM #1:SCREEN GRAPHICS
WINDOW FULL ON
WINDOW TITLE "MACHINAMOT"
WINDOW OPEN
RANDOMIZE TIME
DIM a$(15),b$(15):k=1
FOR i=1 TO 15
READ a$(i),b$(i)
NEXT i
LABEL debut
i=RND(15):j=RND(15)
c$=a$(i)+b$(j)
FOR i=1 TO LEN(c$)
m$=MID$(c$,LEN(c$)-i+1,1)
FOR j=1 TO 23-i
PRINT AT(2*j,2*k) COLOR(RND(15)) POINTS(2
0) " " m$;
NEXT j
PRINT CHR$(7);
NEXT i
k=k+1:IF k=11 THEN k=1:CLS
FOR j=1 TO 1000:NEXT j
GOTO debut

DATA ANDRO,CRATE,COSMO,PHILE,MICRO,PHOBE
DATA HELIO,TROPE,SEXO,GYNE,STEREO,MANE
DATA PALED,NAUTE,MACRO,PHAGE,HYDRO,FUGE
DATA STENO,PATHE,PATHO,CARDE,OLED,DERME
DATA MYTHO,SCOPE,HOMEO,VORE,OCTO,PHONE

```

Dé truqué

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME GW-BASIC :

Il utilise le générateur aléatoire du PC supposé de qualité convenable (on pourra aisément utiliser tout autre générateur, si besoin est). La boucle I (50-120) s'occupe de l'entrée de chacune des probabilités, avec vérification à chaque entrée et sur le total (ligne 90) ; chaque probabilité est conservée dans une variable indicée E (I) : ligne 70 du programme.

Le lancer de dé truqué est effectué à l'aide d'une autre boucle I (160-180), la ligne permettant le trucage se trouvant en 170. Un nouveau tirage pourra être effectué sur pression d'une touche (ligne 200).

POUR JOUER :

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

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

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic. On tape 0 pour sortir du programme.

```

10 REM DE TRUQUE
20 SCREEN 0,1:WIDTH 40:RANDOMIZE TIMER
30 COLOR 1,7:CLS
40 P=0:DIM E(6)
50 FOR I=1 TO 6
60 LOCATE 3,18:PRINT"
"
70 LOCATE 3,1:PRINT"PROBABILITE DU DE ";
I;:INPUT E(I)
80 P=P+E(I):A$=SPACE$(5):E$=STR$(E(I)):R
SET A$=E$
90 IF E(I)<0 OR E(I)>1 OR P>1.1 THEN 220
100 COLOR 4:LOCATE 7,6*I-4:PRINT"DE ";I
110 COLOR 1:LOCATE 9,6*I-5:PRINT A$
120 NEXT I
130 IF P<.9 THEN BEEP:RUN 30
140 FOR I=1 TO 500:NEXT I
150 N=INT(RND*6+1)
160 FOR I=1 TO 6
170 IF N=I AND RND>E(I) THEN 150
180 NEXT I
190 LOCATE 15,1:PRINT"TIRAGE :";N;"
":PLAY"CGE"
200 PRINT:INPUT"FAIRE - ENTREE - POUR UN
AUTRE TIRAGE";R$
210 GOTO 150
220 LOCATE 4,1:PRINT"IMPOSSIBLE :0<= P <
=1"
230 P=P-E(I):BEEP
240 FOR J=1 TO 1500:NEXT J
250 LOCATE 4,1:PRINT"
"
260 GOTO 70

REM Dé truqué

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "DE TRUQUE"
WINDOW OPEN
RANDOMIZE(TIME)
LABEL début
CLS:p=0
FOR i=1 TO 6
LABEL tirde
PRINT AT (1;2) STRING$(50,32)
PRINT AT (1;2) "Probabilité de sortir un
";i;" : ";:INPUT "",r$
e(i)=VAL(r$)
IF e(i)<0 OR e(i)>1 OR p+e(i)>1.1 THEN GO
SUB faux:GOTO tirde
p=p+e(i)
PRINT AT (10*i-5;6) EFFECTS(1) "Dé " COLO
R(8+i) i
PRINT AT (10*i-5;9) POINTS(16) COLOR(8+i)
e(i)
NEXT i
IF p<0.9 THEN GOSUB faux:GOTO début
PRINT AT (1;2) STRING$(50,32)
LABEL tirage
n=RND(6)
FOR i=1 TO 6
IF n=i AND RND>e(i) GOTO tirage
NEXT i
PRINT AT (1;14) EFFECTS(1) "Résultat du t
irage " COLOR(8+n) POINTS(16) n " "
PRINT CHR$(7)
PRINT AT (1;16) "Appuyez une touche pour
continuer, 0 pour finir"
r$=INPUT$(1):IF r$="0" THEN END
PRINT AT (1;14) POINTS(16) STRING$(40,32)
AT (1;16) STRING$(40,32)
GOTO tirage

LABEL faux
PRINT CHR$(7) CHR$(7) CHR$(7)
PRINT AT (1;4) "Impossible : 0<= p <= 1"
FOR j=1 TO 6000:NEXT j
PRINT AT (1;4) STRING$(40,32)
RETURN

```

Poker

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu contre le PC.

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 PC.

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 PC de jouer.

LE PROGRAMME GW-BASIC :

Les lignes 50 et 60 définissent les cartes. Une variable N est dimensionnée à 16, en ligne 40, pour recevoir 10 cartes joueur et 5 cartes PC.

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

L'affichage des cartes est réalisé par le sous-programme 260-370.

Une autre boucle I (190-210) réalise le deuxième tirage du joueur, la ligne 240 permettant le deuxième affichage, et la boucle I (260-330) 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 0 suivi de RETURN s'il veut arrêter la partie. Pour jouer, il répondra simplement RETURN s'il s'estime servi, ou il entrera les numéros des cartes qu'il refuse suivis de RETURN.

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

Il suffira ensuite de taper une touche quelconque 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 PC, l'introduction des règles sur les paris et même, pourquoi pas, des possibilités de bluff.

VERSION BASIC 2 AMSTRAD :

Le programme utilise la souris de l'AMSTRAD pour choisir les cartes à retirer. A cet effet, on devra cliquer avec le bouton de gauche sur les cases contenant des chiffres. Les chiffres des cartes à retirer apparaissent en vidéo inverse. On annule le choix en cliquant une deuxième fois. Le clic

sur le bouton de droite valide les choix effectués et provoque un retraitage éventuel, suivi du jeu machine. On rejouera en cliquant à gauche, et on arrête en cliquant à droite.

```

10 REM POKER
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF
40 DIM N(16)
50 V$="7 8 9 10VADARDOAS"
60 C$(0)=CHR$(3):C$(1)=CHR$(4):C$(2)=CHR
$(5):C$(3)=CHR$(6)
70 CLS
80 FOR I=1 TO 16
90 N(I)=INT(RND*32)
100 IF I=1 THEN 140
110 FOR J=1 TO I-1
120 IF N(I)=N(J) THEN 90
130 NEXT J
140 NEXT I
150 K=0:P=0
160 LOCATE 13,1:PRINT"CARTES REFUSEES"
170 GOSUB 260
180 IF R$="0" THEN PLAY "CDE":END
190 FOR I=1 TO LEN(R$)
200 N(VAL(MID$(R$,I,1)))=N(I+5)
210 NEXT I
220 CLS
230 P=1
240 GOSUB 260
250 K=K+10
260 FOR I=1 TO 5
270 J=INT(N(I+K)/8)
280 F=2*(N(I+K)-8*INT(N(I+K)/8))+1
290 R$=MID$(V$,F,2)
300 LOCATE K+1,7*I-3:COLOR 5-3*(K=10):PR
INT I:SOUND 600,1
310 LOCATE K+3,7*I-4:COLOR 0:PRINT R$;"
":COLOR 4+3*(J>1):PRINT C$(J)
320 COLOR 0:LOCATE K+4,7*I-4:PRINT "----
-"
330 NEXT I
340 IF P=1 THEN P=0:RETURN
350 IF K=10 THEN COLOR 4:PRINT:INPUT"FAI
RE - ENTREE - POUR CONTINUER":R$:GOTO 70
360 LOCATE 13,19:INPUT R$
370 RETURN

```

```

REM Poker
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "POKER"
WINDOW OPEN
RANDOMIZE TIME:DIM n(16),t(33):co$="TREFL
E CARREAUCOEUR PIQUE "
v$=" 7 8 9 10 VALETDAVE ROI A
S "
f(0)=3:f(1)=2:f(2)=14:f(3)=12
LABEL début
FOR i=0 TO 32:t(i)=0:NEXT i
FOR i=1 TO 16
REPEAT:n(i)=RND(32)-1:UNTIL t(n(i))<>1:t(
n(i))=1
NEXT i
CLS:r$=" ":r=1:a=0:b=0
PRINT AT (1;10) "Cliquez sur les cases du
haut, gauche pour choisir, droite pour v
alider"
GOSUB tir
FOR i=1 TO LEN(r$)
n(VAL(MID$(r$,i,1)))=n(i+5)
NEXT i
b=1:GOSUB tir:a=a+10:GOSUB tir
PRINT AT (1;21) "Cliquez à gauche pour ti
rer, à droite pour finir";
REPEAT:c=BUTTON(1):d=BUTTON(2):UNTIL c<>-
1 OR d<>-1
IF d<>-1 THEN END ELSE GOTO début
LABEL tir
FOR i=1 TO 5
co=INT(n(i+a)/8):f=5*(n(i+a) MOD 8)+1:s$
=MID$(v$,f,5)
t$=MID$(co$,7*co+1,7)
x=8*i-4:y=a+2-(a>5):PRINT AT (16*i-8;y-1)
i:GOSUB rectangle
IF p=1 THEN s$=" "+s$
PRINT AT (16*i-9;y+3) s$ AT (16*i-10;y+5)
t$;CHR$(7)
NEXT i
IF b=1 OR a=10 THEN b=0:RETURN
GOSUB choix:RETURN
LABEL choix
c=BUTTON(1):d=BUTTON(2):IF c=-1 AND d=-1
GOTO choix
IF d<>-1 THEN RETURN
u=XMOUSE:v=YMOUSE:IF v<168 GOTO choix
FOR i=1 TO 10:NEXT i
ch=-(u>60 AND u<80)-2*(u>188 AND u<206)-3
*(u>315 AND u<335)
ch=ch-4*(u>444 AND u<465)-5*(u>572 AND u<
588):IF ch=0 GOTO choix
BOX 128*ch-68;190,20,11 MODE(3) FILL WITH
8
IF MID$(r$,ch,1)<>" " THEN MID$(r$,ch,1)=
" " ELSE MID$(r$,ch,1)=CHR$(48+ch)
" " ELSE MID$(r$,ch,1)=CHR$(48+ch)
GOTO choix
LABEL rectangle
BOX 16*x-4;190-10*a,20,11
BOX 16*x-26;130-10*a,70,60 COLOR(f(co)) M
ODE(2) FILL WITH 4 ROUNDED
RETURN

```

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 PC 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 GW-BASIC :

La boucle 1 (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 une touche quelconque.

On devra faire CTRL et Break pour finir.

EXTENSIONS POSSIBLES :

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

On pourra aussi entrer non point la longueur du mot, mais le mot lui-même, l'adversaire devant alors répondre à la question CORRECT OU NON ? On peut aussi étendre le jeu à plus de deux joueurs. Et pourquoi ne pas essayer de définir un jeu contre le PC, 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 ?

VERSION BASIC 2 AMSTRAD :

Le programme utilise des fenêtres pour les divers affichages, et chaque joueur devra taper à son tour le mot qu'il a pu former (et non sa longueur). L'ordinateur vérifie que les lettres correspondent bien aux lettres affichées, et calcule lui-même la longueur. On tape « 0 » pour finir, et une touche quelconque pour refaire un tirage.

```
10 REM LETTRES
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF
40 S=0:J=0
50 CLS
60 L=INT(RND*6+7)
70 A$="AEIOUEA"
80 LOCATE 6,1:PRINT"LETTRES :"
90 FOR I=1 TO L
100 COLOR INT(RND*7):LOCATE 6,9+2*I:PRINT
    CHR$(INT(RND*26+65)):BEEP
110 IF RND<.4 THEN LOCATE 6,9+2*I:PRINT
    MID$(A$,INT(RND*7+1),1)
120 NEXT I
130 I=1
140 COLOR 3*I-2:LOCATE 11,1:PRINT"JOUEUR
    :";I;:INPUT" - LONGUEUR :";L
150 LOCATE 11,25:PRINT" "
160 IF I=1 THEN P=L
170 I=I+1
180 IF I<3 THEN 140
190 S=S+(P AND P>L)
200 J=J+(L AND L>P)
210 COLOR 0:LOCATE 14,1:PRINT"SCORES - J
    QUEUR 1 :";S;" - JOUEUR 2 :";J
220 PRINT:INPUT"TAPER - ENTREE - POUR CO
    NTINUER";R$
230 GOTO 50

REM Lettres
FOR i=1 TO 4:CLOSE WINDOW i:CLOSE #i:NEXT
    i
a(1)=1:a(2)=4:a(3)=2
FOR i=1 TO 3:OPEN #i WINDOW a(i):NEXT i
STREAM #1:SCREEN GRAPHICS
WINDOW #2 SIZE 37,10:WINDOW #3 SIZE 37,10
WINDOW SIZE XUSABLE,YUSABLE/3:WINDOW TITL
    E "LETTRES"
WINDOW PLACE 0,120:WINDOW #2 PLACE 0,10:W
    INDOW #2 TITLE "JOUEUR 1"
WINDOW #3 PLACE 320,10:WINDOW #3 TITLE "J
    OUEUR 2"
WINDOW #2 OPEN:WINDOW #3 OPEN:WINDOW OPEN
    RANDOMIZE:j1=0:j2=0:a$="AEIOUEAUAEIO"
REPEAT
    l=RND(6)+6:CLS:f$="":CLS #2:CLS #3
    PRINT AT (12;2) POINTS(16) COLOR(4) "LETT
        RES TIREES"
    FOR i=1 TO 1
        r$=CHR$(RND(26)+64)
        IF RND<0.6 THEN r$=MID$(a$,RND(LEN(a$)),1
            )
        PRINT CHR$(7);:PRINT AT (3*i+3;5) COLOR(R
            ND(12)) POINTS(20) r$
        f$=f$+r$
    NEXT i
    PRINT AT (1;7) "Attention...."
    FOR i=1 TO 28000+RND(20000):NEXT
        CLS:i=RND(2):GOSUB controle
        i=ABS(i-3):GOSUB controle
        j1=j1+(p AND p>1):j2=j2+(1 AND 1>p)
        PRINT #2:PRINT #2,"Score actuel " j1
        PRINT #3:PRINT #3,"Score actuel " j2
        PRINT AT (10;4) COLOR(14) "Taper 0 pour a
            rrêter, une touche pour continuer"
        r$=INPUT$(1)
        UNTIL r$="0"
        PRINT AT (1;6) "Le joueur " 2+(j1>j2) " g
            agne"
    END

    LABEL controle
    PRINT #i+1,"Votre mot : ";:INPUT #i+1,r$:
        r$=UPPER$(r$)
    FOR j=1 TO LEN(r$)
        m$=MID$(r$,j,1):s=INSTR(1,f$,m$):IF s=0 T
            HEN r$="":GOTO suite
    NEXT j
    LABEL suite
    IF i=1 THEN p=LEN(r$) ELSE l=LEN(r$)
    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 : BBCDHHH.

LE PROGRAMME GW-BASIC :

Les lignes 40 à 70 dessinent la locomotive et le wagon.

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 à 190 et 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 (230-270) 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 RETURN.

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

On obtiendra alors l'affichage du score.

EXTENSIONS POSSIBLES :

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

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM ACCROCHEZ LES WAGONS
20 SCREEN 0,1:WIDTH 80:COLOR 0,6
30 RANDOMIZE TIMER:KEY OFF
40 D$=CHR$(126)+CHR$(76)+"^"
50 A$=CHR$(219)+CHR$(219)+CHR$(219)
60 B$="<"+CHR$(95)+">"
70 C$="0 0"
80 N=0
90 LY=INT(RND*75+2)
100 LX=INT(RND*20+2)
110 WY=INT(RND*75+2)
120 WX=INT(RND*20+2)
130 IF ABS(LX-WX)<4 THEN 90
140 CLS
150 LOCATE WX,WY:COLOR 1,6:PRINT B$
160 LOCATE WX+1,WY:COLOR 0,6:PRINT C$
170 LOCATE LX-1,LY:COLOR 3,6:PRINT D$
180 LOCATE LX,LY:COLOR 3,6:PRINT A$
190 LOCATE LX+1,LY:COLOR 0,6:PRINT C$:CO
LOR 4,6
200 IF LX=WX AND WY=LY-3 THEN 300
210 LOCATE 23,1:INPUT"CHEMIN A SUIVRE (D
GGHBB...):";R$
220 N=N+1:X=0:Y=0
230 FOR I=1 TO LEN(R$)
240 IF MID$(R$,I,1)>="a" THEN MID$(R$,I,
1)=CHR$(ASC(MID$(R$,I,1))-32)
250 X=X-(MID$(R$,I,1)="B" AND LX+X<22)+(
MID$(R$,I,1)="H" AND LX+X>2)
260 Y=Y-(MID$(R$,I,1)="D" AND LY+Y<77)+(
MID$(R$,I,1)="G" AND LY+Y>2)
270 NEXT I
280 LX=LX+X:LY=LY+Y
290 GOTO 140
300 LOCATE 23,1:PRINT"BRAVO ; TROUVE EN
";N;" COUPS"
310 PLAY"CGE"

```

```

REM Accrochez les wagons

STREAM #1:SCREEN GRAPHICS 640 FIXED,240 F
IXED
USER SPACE 640,240
WINDOW FULL ON
WINDOW TITLE "ACCROCHEZ LES WAGONS"
WINDOW OPEN
RANDOMIZE TIME:DIM l(20,2),w(20,2):u=30:v
=120:w=56
FOR i=1 TO 20:READ a,b:l(i,1)=a:l(i,2)=b:
NEXT i
FOR i=1 TO 19:READ a,b:w(i,1)=a:w(i,2)=b:
NEXT i
n=0:ly=4*INT(RND*u)+u:lx=4*INT(RND*v)+u
LABEL tirwagon
wy=4*INT(RND*u)+u:wx=4*INT(RND*v)+u
IF ABS(lx-wx)<u+15 GOTO tirwagon
LABEL début
CLS:FOR i=1 TO 19:LINE l(i,1)+lx;200-ly+1
(i,2),l(i+1,1)+lx;200-ly+1(i+1,2):NEXT i
FOR i=1 TO 18:LINE w(i,1)+wx;200-wy+1(i,2
),w(i+1,1)+wx;200-wy+1(i+1,2):NEXT
IF lx=wx AND ly=wy GOTO fin
INPUT AT (1;19) "Chemin (DGGHBB...):",r$
x=0:y=0:n=n+1:r$=UPPER$(r$)
FOR i=1 TO LEN(r$)
x=x-4*((MID$(r$,i,1)="D")-(MID$(r$,i,1)="
G"))
y=y-4*((MID$(r$,i,1)="B")-(MID$(r$,i,1)="
H"))
PRINT CHR$(7);
NEXT i
lx=lx+x:ly=ly+y
IF lx<0 OR lx>585 THEN lx=lx-x
IF ly<0 OR ly>200 THEN ly=ly-y
GOTO début
LABEL fin
PRINT AT (1;19) "Bravo; trouvé en " n " c
oups"
r$=INPUT$(1):END

```

```

DATA 0,10,0,30,10,30,10,27,30,27,30,40,40
,40,40,27,50,27
DATA 50,17,55,10,45,10,42,0,38,0,35,10,25
,10,22,0,18,0,15,10,0,10
DATA 0,10,0,30,5,30,5,15,45,15,45,30,50,3
0,50,12,55,12
DATA 55,10,45,10,42,0,38,0,35,10,25,10,22
,0,18,0,15,10,0,10

```

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

LE PROGRAMME GW-BASIC :

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

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

La ligne 160 décode la couleur de la carte.

La ligne 170 choisit la position d'affichage.

La ligne 180 décode la hauteur.

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

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

POUR JOUER :

On lance le programme par RUN.

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

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

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

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

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

EXTENSIONS POSSIBLES :

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

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM TOUR DE CARTES
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF
40 CLS
50 A$="":C(0)=1:C(1)=4:C(2)=6:C(3)=2
60 FOR I=1 TO 21
70 B$=CHR$(INT(RND*(52+38)))
80 FOR J=1 TO LEN(A$)
90 IF B$=MID$(A$,J,1) THEN 70
100 NEXT J
110 A$=A$+B$
120 NEXT I
130 H=3:P=1
140 B$="TRCACOPIVADARDAS"
150 FOR I=1 TO LEN(A$)*P-11*(P=0)
160 S=INT((ASC(MID$(A$,I,1))-38)/13)
170 J=7*INT((I-1)/7):SOUND INT(RND*3000+400),1
180 T=ASC(MID$(A$,I,1))-37-13*S
190 IF P=0 THEN LOCATE 1,1:PRINT"VOTRE C
ARTE EST : "
200 LOCATE (I-J)*P+1,J*P+INT(J/2)-16*(P=0)+1,0:COLOR C(S)
210 IF T<=9 THEN PRINT STR$(T+1);
220 IF T>9 THEN PRINT MID$(B$,ABS(2*T-11),2);
230 LOCATE (I-J)*P+1,(J+3)*P+2+INT(J/2)-21*(P=0):PRINT MID$(B$,2*S+1,2)
240 NEXT I
250 IF P=0 THEN END
260 LOCATE 17,1,0:COLOR 0,7:INPUT"QUELLE
COLONNE :";N
270 CLS
280 H=H-1
290 R$="20120":B$=""
300 FOR J=0 TO 2
310 B$=B$+MID$(A$,7*VAL(MID$(R$,N+J,1))+1,7)
320 NEXT J
330 IF H=0 THEN A$=B$:P=0:GOTO 140
340 A$=""
350 FOR J=0 TO 2
360 FOR I=1 TO 7
370 A$=A$+MID$(B$,3*I-2+J,1)
380 NEXT I
390 NEXT J
400 GOTO 140

REM Tour de cartes

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "TOUR DE CARTES"
WINDOW OPEN
RANDOMIZE TIME
a$="":f(0)=3:f(1)=2:f(2)=4:f(3)=14
FOR i=1 TO 21
LABEL tircarte
b$=CHR$(RND(52)+37)
FOR j=1 TO LEN(a$)
IF b$=MID$(a$,j,1) GOTO tircarte
NEXT j
a$=a$+b$
NEXT i
h=3:p=1
LABEL debut
b$="TREFLE CARREAUCOEUR PIQUE VALET DA
ME ROI AS "
FOR i=1 TO LEN(a$)*p+(11 AND p=0)
s=INT((ASC(MID$(a$,i,1))-38)/13)
j=7*INT((i-1)/7)
t=ASC(MID$(a$,i,1))-37-13*s
IF p=0 THEN PRINT AT (1;1) "Votre carte e
st : ";STRING$(40,32)
LOCATE 3*j*p+(20 AND p=0)+1;(i-j)*p+1
IF t<=9 THEN PRINT STR$(t+1);ELSE PRINT C
OLOR(f(s)) MID$(b$,ABS(7*t-41),7);
PRINT AT ((3*j+5)*p+(25 AND p=0)+2;(i-j)*
p+1) COLOR(f(s)) MID$(b$,7*s+1,7)
NEXT i
IF p<>0 GOTO question
r$=INPUT$(1)
END
LABEL question
PRINT AT (8;12) POINTS(20) "1" AT (29;12)
"2" AT (51;12) "3"
INPUT AT (1;15) "Quelle colonne : ",n$:n=
VAL(n$)
CLS:h=h-1:r$="20120":b$=""
FOR j=0 TO 2
b$=b$+MID$(a$,7*VAL(MID$(r$,n+j,1))+1,7)
NEXT j
IF h=0 THEN a$=b$:p=0:GOTO debut
a$=""
FOR j=0 TO 2
FOR i=1 TO 7
a$=a$+MID$(b$,3*i-2+j,1)
NEXT i,j
GOTO debut

```

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 GW-BASIC :

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

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

La ligne 180 permet de recommencer en tapant une touche quelconque.

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 une touche quelconque provoquera l'affichage d'une nouvelle série de notes, et une pression sur les touches CTRL et Break provoque la fin de la partie.

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM COMPOSE
20 SCREEN 0,1:WIDTH 40
30 RANDOMIZE TIMER:KEY OFF
40 COLOR 1,7:CLS
50 FOR I=1 TO 5
60 LOCATE 4+2*I,1:FOR J=1 TO 40:PRINT CH
R$(219);:NEXT J
70 NEXT I
80 N$="SI LA SOLFA MI RE DO ":P$="BAGFED
C"
90 FOR I=1 TO 13
100 X=INT(RND*14+1)
110 D$=MID$(N$,3*(X+7*(X>7))-2,3)
120 COLOR 4:LOCATE X,3*I-1:PRINT CHR$(17
9):LOCATE X+1,3*I-1:PRINT CHR$(179):LOCA
TE X+2,3*I-2:PRINT"0"
130 COLOR 0:LOCATE 19,3*I-2:PRINT D$
140 IF X<=7 THEN PLAY "05L4"+MID$(P$,X+7
*(X>7),1):GOTO 160
150 PLAY "04L4"+MID$(P$,X+7*(X>7),1)
160 FOR J=1 TO 800:NEXT J
170 NEXT I
180 LOCATE 22,1:PRINT"APPUYEZ SUR UNE TO
UCHE":R$=INPUT$(1)
190 GOTO 40

```

```

REM Compose

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "COMPOSE"
WINDOW OPEN
RANDOMIZE TIME
a$=STRING$(70,CHR$(95)):e$=CHR$(27)+CHR$(
11)
REPEAT
CLS
FOR i=1 TO 5
PRINT AT (1;2*i+3) a$
NEXT i
n$="SI LA SOLFA MI RE DO "
FOR i=1 TO 17
x=RND(14)
d$=MID$(n$,3*(x+(-7 AND x>7))-2,3)
PRINT AT (4*i-2;x+1) COLOR(4) POINTS(20)
MODE(2) e$
PRINT AT (4*i-2;17) d$
FOR j=1 TO 2000:NEXT
NEXT i
r$=INPUT$(1)
UNTIL UPPER$(r$)="S"
END

```

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 GW-BASIC :

Les probabilités désirées pour chaque tirage sont introduites par une boucle I (50-120) avec toutes les vérifications de validité nécessaires. Le nombre de tirages demandés est introduit en ligne 140.

La boucle J (180-250) comptabilise les tirages.

La boucle I (200-220) assure la conformité de chaque tirage avec les probabilités introduites en début de jeu.

Les résultats sont comptabilisés en ligne 230, et affichés grâce à la ligne 240.

POUR JOUER :

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

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

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

Les résultats sont affichés sous les numéros des tirages : ainsi 32 apparaissant sous le 1 signifie que le tirage de 1, auquel on avait affecté une probabilité de x, 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.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.


```

10 REM TIRAGE D'UN DE TRUQUE
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF
40 DIM E(6):P=0:CLS
50 FOR I=1 TO 6
60 LOCATE 4,1:PRINT"
"
70 COLOR 0:LOCATE 4,1:PRINT"PROBABILITE
DU DE ";I;:INPUT E(I)
80 P=P+E(I):A$=SPACE$(5):E$=STR$(E(I)):R
SET A$=E$
90 IF E(I)<0 OR E(I)>1 OR P>1.1 THEN 270
100 LOCATE 7,6*I-3:PRINT"DE";:COLOR 1,2:
PRINT I:COLOR 4,7
110 LOCATE 9,6*I-5:PRINT A$
120 NEXT I
130 IF P<.9 THEN RUN
140 COLOR 0:PRINT:INPUT"NOMBRE DE TIRAGE
S :";P
150 FOR I=1 TO 6
160 LOCATE 16,6*I-2:COLOR 1,2:PRINT I:CO
LOR 1,7
170 NEXT I
180 FOR J=1 TO P
190 N=INT(RND*6+1)
200 FOR I=1 TO 6
210 IF N=I AND RND>E(I)-INT(E(I)) THEN 1
90
220 NEXT I
230 E(N)=E(N)+1:A$=SPACE$(5):E$=STR$(INT
(E(N))):RSET A$=E$
240 LOCATE 18,6*N-5:PRINT A$
250 NEXT J
260 END
270 LOCATE 5,1:PRINT"IMPOSSIBLE :0<= P <
=1"
280 P=P-E(I):BEEP
290 FOR J=1 TO 1700:NEXT J
300 LOCATE 5,1:PRINT"
"
310 GOTO 70

```

```

REM Tirage d'un dé truqué

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "TIRAGE D'UN DE TRUQUE"
WINDOW OPEN
RANDOMIZE(TIME)
LABEL début
CLS:p=0:a$="###.##"
FOR i=1 TO 6
LABEL tirde
PRINT AT (1;2) STRING$(50,32)
PRINT AT (1;2) "Probabilité de sortir un
";i;" : ";:INPUT "",r$
e(i)=VAL(r$)
IF e(i)<0 OR e(i)>1 OR p+e(i)>1.1 THEN 60
SUB faux:GOTO tirde
p=p+e(i)
PRINT AT (12*i-5;6) EFFECTS(1) "Dé " COLO
R(8+i) i
PRINT AT (12*i-6;9) POINTS(16) COLOR(8+i)
e(i)
NEXT i
IF p<0.9 THEN GOSUB faux:GOTO début
PRINT AT (1;2) STRING$(50,32)
INPUT AT (1;11) "Nombre de tirages : ",p$
:p=VAL(p$)
FOR j=1 TO p
LABEL tirage
n=RND(6)
FOR i=1 TO 6
IF n=i AND RND>e(i)-INT(e(i)) GOTO tirage
NEXT i
e(n)=e(n)+1
PRINT AT (12*n-5;13) n
PRINT AT (12*n-8;15) USING a$ COLOR(8+n)
POINTS(16) e(n)
PRINT CHR$(7)
NEXT j
PRINT AT (1;18) "C'est fini...."
r$=INPUT$(1)
END

LABEL faux
PRINT CHR$(7) CHR$(7) CHR$(7)
PRINT AT (1;4) "Impossible : 0<= p <= 1"
FOR j=1 TO 6000:NEXT j
PRINT AT (1;4) STRING$(40,32)
RETURN

```


Niveau 3

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Bridge	Mettez au point vos propres problèmes de bridge.
Le pendu	Un jeu de lettres très classique (2 joueurs).
Mastermind	Avec des chiffres, mais très classique. Un seul joueur.
Répétitions	Retrouver une séquence de chiffres de plus en plus longue.
Labyrinthix	Il vous faudra beaucoup de réflexion pour réussir.
Belote	Pour simuler une donne, avec un jeu de 32 cartes.
Un taquin de calcul	Un bon entraînement (2 joueurs).
Hanoï	Un conseil : commencez petit ! ! !
Plus grand, plus petit	Un jeu de déduction intéressant, pour un joueur.
Reines sur l'échiquier	Un vieux problème, pour vous aider à réfléchir.
Ordre	Un peu d'astuce pour remettre en ordre, le plus vite possible.
Idem	Un jeu de calcul mental, avec plusieurs niveaux, pour plusieurs joueurs.
Permutations	Un jeu de réflexion, pas toujours évident.
A travers	Un jeu de stratégie pour deux joueurs.
Le loup et les agneaux	Un jeu de réflexion pour 2 joueurs.
Le jeu de Marienbad	Un jeu de logique pour deux joueurs, dans une version étendue.
Pendu inversé	Cette fois, c'est l'ordinateur qui essaie de deviner le mot.
Espion	Dites-lui votre amour en code.
Décomaque	Pour vous éviter de sécher trop longtemps sur Nicomaque.
Horloge	Ne jetez pas votre vieille télévision : transformez-là en horloge.

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 lème 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 GW-Basic permet de plus le remplacement direct d'une sous-chaîne par une affectation du type MID\$ (-.) = "-".

Ainsi, on pourra écrire MID\$ (A\$, I, 1) = "-", ce qui permet de simplifier considérablement certaines écritures.

BOUCLE WHILE-WEND

Nous avons vu, au niveau 1, une première instruction permettant de définir des boucles : l'instruction FOR-NEXT. Mais cette instruction souffre d'un excès de rigidité, puisque le compteur de boucle (en général une variable entière I ou J) ne peut progresser que d'une quantité finie (de 1 en 1, ou de 2 en 2, etc.). La structure WHILE-WEND sera souvent plus souple d'utilisation : le test de fin est fait en début de boucle (par exemple WHILE a<>o) et il est possible de tester plusieurs variables (WHILE a<>o OR b=1)... De plus, la boucle ne sera pas parcourue si le test se révèle négatif dès le départ. Les boucles WHILE-WEND font partie des instructions des langages structurés.

FENETRE TEXTE : L'INSTRUCTION VIEW PRINT

Utilisable seulement en mode texte, 40 ou 80 colonnes, l'instruction VIEW PRINT a TO b permet de définir une fenêtre d'écran, allant de la ligne a à la ligne b (a et b sont entre 1 et 24). Les prochains affichages seront effectués dans la zone ainsi définie, qui devient la partie utile de l'écran. On peut alors l'effacer, changer sa couleur, y provoquer des défilements (scrolling) sans affecter le reste de l'écran. On reviendra en mode normal avec VIEW PRINT 1 TO 24, ou avec un SCREEN.

LES VARIABLES TIME\$ ET DATE\$

Le GW-Basic permet un accès très complet à l'horloge interne du PC, grâce aux 2 instructions TIME\$ et DATE\$. La première permet soit d'entrer, soit de lire l'heure, et la deuxième fonctionne de même pour la date.

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 bridgès 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 GW-BASIC :

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

La table B\$, dimensionnée en ligne 70, permettra d'afficher les symboles des couleurs et aussi d'améliorer la désignation de certaines cartes (10, valet, dame, roi et as).

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

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

La ligne 110 calcule la couleur de la carte tirée.

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

La ligne 160 s'occupe de l'impression complémentaire dans le cas où l'on tire un 10, un valet, une dame, un roi ou un as.

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

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

POUR JOUER :

On lance le programme par RUN.

Une première série de 13 cartes est affichée verticalement. Il suffira d'appuyer sur une touche quelconque 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 PC, par exemple), ou introduire des possibilités de jeu de la carte, y compris pour le PC.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM BRIDGE
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF
40 CLS:C$=CHR$(3)+CHR$(4)+CHR$(5)+CHR$(6)
 )+"0AADS"
50 R$="234567891VDRA":C(1)=4:C(2)=12:C(3)
  )=2:C(4)=1
60 R$=R$+R$+R$+R$
70 DIM B$(9):FOR I=1 TO 9:B$(I)=MID$(C$,
  I,1):NEXT I
80 FOR J=1 TO 4
90 FOR I=1 TO 13
100 X=INT(RND*LEN(R$)+1)
110 S=INT((X-1)/13)+1
120 IF MID$(R$,X,1)="0" THEN 100
130 T=X-13*INT(X/13)
140 T=T-13*(T=0)
150 LOCATE I+1,10*J-8:COLOR 0:PRINT MID$
  (R$,X,1);
160 IF T>=9 THEN PRINT B$(T-4)
170 COLOR C(S):LOCATE I+1,10*J-4:PRINT B
  $(S)
180 MID$(R$,X,1)="0":SOUND INT(RND*4000+
  500),1
190 NEXT I
200 LOCATE 19,1:COLOR 5:PRINT"APPUYEZ SU
  R UNE TOUCHE":T$=INPUT$(1)
210 LOCATE 19,1:PRINT SPACE$(30)
220 NEXT J

```

```

REM Bridge

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
  IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "BRIDGE"
WINDOW OPEN
RANDOMIZE TIME:f(1)=3:f(2)=13:f(3)=7:f(4)
  =2
r$="234567891VDRA":r$=r$+r$+r$+r$:DIM t(5
  2)
FOR i=1 TO 52:t(i)=0:NEXT i
b$(1)="TREFLE":b$(2)="PIQUE":b$(3)="COEUR
  ":b$(4)="CARREAU"
b$(5)="0":b$(6)="A":b$(7)="A":b$(8)="0":b
  $(9)="S"
SET POINTS(8) MODE(2) EFFECTS(1)
FOR j=1 TO 4
FOR i=1 TO 13
LABEL tirage
x=RND(52):IF t(x)=1 THEN GOTO tirage ELSE
  t(x)=1
s=INT((x-1)/13)+1:t=x-13*INT(x/13):t=t+(1
  3 AND t=0)
BOX 160*j-140;190-9*i,120,9 COLOR(f(s)) F
  ILL WITH 12 ROUNDED
PRINT AT (20*j-14;i+1) MID$(r$,x,1);
IF t>=9 THEN PRINT b$(t-4)
PRINT AT (20*j-10;i+1) b$(s)
FOR k=1 TO 600:NEXT k
NEXT i
IF j<>4 THEN PRINT AT (1;17) POINTS(10) "
  Appuyez une touche" ELSE GOTO fin
v$=INPUT$(1):PRINT AT (1;17) POINTS(10) M
  ODE(1) STRING$(30,32)
NEXT j
LABEL fin
v$=INPUT$(1):END

```

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 GW-BASIC :

Le mot à deviner est introduit en début de programme (ligne 30) ; des variables alphanumériques, en ligne 40, contiennent tous les éléments du dessin.

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

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

Les lignes 150-160 affichent le résultat et arrêtent le jeu.

La lettre essayée est entrée en ligne 170 et testée dans une boucle I (200-220) ; le sous-programme 260-280 remplace chaque tiret par la lettre correspondante.

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```
10 REM LE PENDU
20 SCREEN 0,1:WIDTH 40:COLOR 0,7:KEY OFF
:CLS
30 INPUT"VOTRE MOT :";R$
40 B$=CHR$(189)+" "+CHR$(211)+CHR$(186)+
" "+CHR$(186)+CHR$(186)+" "+CHR$(186)+CH
R$(219)+CHR$(219)+CHR$(219)+"***** "
+CHR$(239)+" "+CHR$(1)+" "+CHR$(179)+"
===="
50 P$="":K=1
60 FOR I=1 TO LEN(R$)
70 P$=P$+"-":IF MID$(R$,I,1)>="a" THEN M
ID$(R$,I,1)=CHR$(ASC(MID$(R$,I,1))-32)
80 NEXT I
90 P=12
100 CLS
110 LOCATE 4,1:PRINT P$
120 FOR I=12 TO P STEP -1
130 COLOR 1:LOCATE 13-P,23:PRINT MID$(B$
,1+3*(I-1),3):COLOR 4
140 NEXT I
150 IF P$=R$ THEN PRINT"BRAVO":PLAY"CEG"
:END
160 IF P=1 THEN PRINT"PERDU ; LE MOT ETA
IT :";R$:PLAY"CCC":END
170 COLOR 0:LOCATE 21,1:INPUT"VOTRE LETT
RE :";S$
180 T=0:IF S$>="a" THEN S$=CHR$(ASC(S$)-
32)
190 LOCATE 22,K:PRINT S$;K=K+1
200 FOR I=1 TO LEN(R$)
210 SOUND INT (RND*4000+500),1:IF MID$(R$
,I,1)=S$ THEN GOSUB 260
220 NEXT I
230 IF T<>0 THEN 110
240 P=P-1
250 GOTO 110
260 T=1
270 MID$(P$,I,1)=S$
280 RETURN
```

REM le pendu

```
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "LE JEU DU PENDU"
WINDOW OPEN
r$="":t$="":p$="":RANDOMIZE TIME
LABEL entrée
PRINT AT (1;3) "Votre mot : ";t$;
s$=INPUT$(1):IF s$=CHR$(13) GOTO début
r$=r$+s$:t$=t$+"#":IF s$="" THEN RUN EL
E GOTO entrée
LABEL début
r$=UPPER$(r$):FOR i=1 TO LEN(r$):p$=p$+"-
":NEXT i
p=10:CLS
LABEL dessin
PRINT AT (4;6) COLOR(4) POINTS(20) p$:RES
TORE
FOR i=10 TO p STEP -1
READ n:FOR j=1 TO n:READ a,b,c,d:LINE a+3
00;240-b,c+300;240-d:NEXT j
NEXT i
IF p=r$ THEN PRINT AT (1;18) COLOR(2) PO
INTS(20) "BRAVO":GOTO fin
IF p=1 THEN PRINT AT (1;18) "Perdu. Le mo
t était " r$:GOTO fin
PRINT AT (1;16) STRING$(30,32):INPUT AT (
1;16) "Votre lettre : ",s$
s$=UPPER$(s$):t=0
FOR i=1 TO LEN(r$)
IF MID$(r$,i,1)=s$ THEN t=1:MID$(p$,i,1)=
s$:PRINT CHR$(7)
NEXT i
IF t<>0 GOTO dessin
p=p-1:GOTO dessin
LABEL fin
PRINT CHR$(7);CHR$(7);CHR$(7);
r$=INPUT$(1):END
```

```
DATA 4,50,60,50,50,50,50,150,50,150,50,15
0,60,150,60,50,60
DATA 4,95,80,95,60,95,60,105,60,105,60,10
5,80,105,80,95,80
DATA 4,80,90,95,80,95,80,105,80,105,80,12
0,90,120,90,80,90
DATA 6,90,90,80,100,80,100,90,110,90,110,
110,110,110,110,120,100,120,100
DATA 110,90,110,90,90,90
DATA 4,95,120,95,110,95,110,105,110,105,1
10,105,120,105,120,95,120
DATA 10,50,135,60,120,60,120,140,120,140,
120,150,135,150,135,140,135
DATA 140,135,130,125,130,125,130,135,130,
135,70,135,70,135,70,125,70,125
DATA 60,135,60,135,50,135
DATA 4,70,135,130,135,130,135,130,155,130
,155,70,155,70,155,70,135
DATA 7,70,155,70,165,70,165,90,165,90,165
,90,155,90,155,110,155
DATA 110,155,110,165,110,165,130,165,130,
165,130,155
DATA 6,70,165,70,185,70,185,90,185,90,185
,90,165,110,165,110,185
DATA 110,185,130,185,130,185,130,165
DATA 8,50,185,50,200,50,200,90,200,90,200
,90,185,90,185,50,185
DATA 110,185,110,200,110,200,150,200,150,
200,150,185,150,185,110,185
```

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 PC 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 GW-BASIC :

Le choix du nombre de chiffres du nombre à découvrir est effectué aux lignes 40 et 50.

Le nombre est fabriqué à la ligne 60 (il pourra y avoir des chiffres répétés).

L'affichage est préparé par la ligne 90.

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

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

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

Les compteurs sont aux lignes 200 et 220.

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

POUR JOUER :

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

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

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

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

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

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

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic. On peut rejouer en répondant « O » à la question « UNE AUTRE PARTIE (O/N) ».

```
10 REM MASTERMIND
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF:CLS
40 INPUT"COMBIEN DE CHIFFRES :";N
50 IF N>6 THEN 40
60 E=1:F=1:X$=STR$(INT(RND*9^N+10^(N-1)))
70 X$=RIGHT$(X$,LEN(X$)-1)
80 CLS
90 PRINT"NO.          NOMBRE          BON
  PLACE"
100 COLOR 1:FOR J=1 TO 40:PRINT CHR$(220
);:NEXT J
110 COLOR 4:LOCATE 4+F,1:PRINT E;TAB(9);
:INPUT E$
120 X=0:Y=X
130 IF E$=X$ THEN PLAY"CED":PRINT"TROUVE
  EN ";E;" COUPS":END
140 FOR I=1 TO N
150 IF MID$(X$,I,1)=MID$(E$,I,1) THEN 22
  0
160 FOR J=1 TO N
170 IF J<>I AND MID$(X$,I,1)=MID$(E$,J,1
  ) THEN 200
180 NEXT J
190 GOTO 230
200 Y=Y+1
210 GOTO 230
220 X=X+1
230 NEXT I
240 LOCATE 4+F,24:PRINT X;TAB(33);Y:PLAY
"CDEFG"
250 E=E+1:F=F+1:IF F=18 THEN F=1
260 GOTO 110
```

```
REM Mastermind
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW TITLE "MASTERMIND"
WINDOW OPEN
RANDOMIZE TIME
PRINT AT (1;3) "Nombre de chiffres ( <9 )
  : ";
n$=INPUT$(1):n=VAL(n$):IF n>8 THEN RUN EL
SE CLS
e=1:g=1:f$=STRING$(8,32)
x$=STR$(INT(RND*9^n+10^(n-1))):x$=RIGHT$(
x$,LEN(x$)-1)
BOX 0;164,640,30 WIDTH 5 COLOR(8) FILL WI
TH 3
PRINT AT (2;3) COLOR(15) POINTS(16) MODE(
2) "ESSAI NO.  NOMBRE ESSAYE  BONS
  PLACES"
LINE 130;160,130;0 WIDTH 5:LINE 310;160,3
10;0 WIDTH 5
LINE 470;160,470;0 WIDTH 5
WHILE e$<>x$
PRINT AT (1;g+5) e AT (19;g+5) f$ AT (48;
g+5) f$ AT (66;g+5) f$
INPUT AT (19;g+5) "? ",e$
a=0:b=0
FOR i=1 TO n
IF MID$(e$,i,1)=MID$(x$,i,1) THEN a=a+1:G
OTO suivant
FOR j=1 TO n
IF j<>i AND MID$(e$,j,1)=MID$(x$,i,1) THE
N b=b+1:GOTO suivant
NEXT j
LABEL suivant
NEXT i
PRINT AT (48;g+5) COLOR(2) a AT (66;g+5)
COLOR(4) b
e=e+1:g=g+1:IF g=16 THEN g=1
WEND
PRINT:PRINT "Vous avez trouvé en " e " co
ups"
PRINT:PRINT "Une autre partie ( O/N )"
r$=INPUT$(1):IF UPPER$(r$)="O" THEN RUN
END
```

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 GW-BASIC :

La ligne 60 appelle un sous-programme (lignes 250 à 360) qui donne la règle du jeu.

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

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

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

La ligne 160 marque les points.

La ligne 170 contrôle le meilleur score.

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

POUR JOUER :

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

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

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

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

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```
10 REM REPETITIONS
20 SCREEN 0,1:WIDTH 40
30 RANDOMIZE TIMER:KEY OFF
40 M=0
50 COLOR 7,0:CLS:A$=""
60 GOSUB 250
70 H=0
80 X=INT(RND*10):A$=A$+RIGHT$(STR$(X),1)
90 CLS:COLOR 1,6:LOCATE 5,RND*40+1:PRINT
  X
100 FOR I=1 TO 500+RND*600:NEXT I
110 CLS:COLOR 4,6
120 LOCATE 11,1:INPUT"VOTRE SEQUENCE :";
  R$
130 IF R$<>A$ THEN 190
140 PRINT:PLAY"CDE":PRINT"EXACT . ATTEN
  TION"
150 FOR I=1 TO 1000:NEXT I
160 H=H+1
170 IF M<H THEN M=H
180 GOTO 80
190 COLOR 0,7:CLS
200 PRINT"VOTRE SCORE EST DE ";H
210 PRINT:PRINT"MEILLEUR SCORE ACTUEL "M
220 PRINT:PRINT"AUTRE PARTIE (O/N)":R$=I
  NPUT$(1)
230 IF R$="N" OR R$="n" THEN PLAY"CCC":E
  ND
240 GOTO 50
250 K=4:B$="VOUS DEVEZ REPETER TOUTE LA
  SERIE DES":GOSUB 310
260 B$="NOMBRES APPARUS DEPUIS LE DEBUT
  DU JEU":GOSUB 310
270 B$="IL Y EN A UN DE PLUS A CHAQUE FO
  IS":GOSUB 310
280 B$="APPUYEZ SUR UNE TOUCHE POUR COMM
  ENCER":GOSUB 310
290 R$=INPUT$(1)
300 RETURN
310 FOR I=1 TO LEN(B$)
320 LOCATE K,I:COLOR INT(RND*15+1):PRINT
  MID$(B$,I,1)
330 SOUND INT(RND*4000+500),1
340 NEXT I
350 K=K+3
360 RETURN
```

```
REM Répétitions

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
  IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "REPETITIONS"
WINDOW OPEN:GOTO début
RANDOMIZE TIME:m=0:BOX 0;0,640,200 COLOR(
  1) FILL WITH 8
a$="VOUS DEVEZ REPETER TOUTE LA SERIE":g=
  3:f=0:GOSUB lec
a$="DES NOMBRES APPARUS DEPUIS LE DEBUT":
  GOSUB lec
a$="DU JEU. IL Y EN A UN DE PLUS A CHAQUE
  ":GOSUB lec
a$="FOIS. APPUYEZ SUR UNE TOUCHE POUR":GO
  SUB lec
a$="COMMENCER":GOSUB lec:r$=INPUT$(1)
LABEL début
h=0:b$=""
LABEL partie
CLS:z=RND(10)-1:b$=b$+RIGHT$(STR$(z),1):r
  $=CHR$(48+z)
f=INT(RND*60+10):g=INT(RND*17+2):t=20:mo=
  2:ef=1:co=RND(13)+1
BOX 8*f-18;194-9*g,40,30 COLOR(15-co) FIL
  L WITH 8 ROUNDED:GOSUB aff
FOR i=1 TO 3000+RND(3000):NEXT i
CLS:INPUT AT (1;10) "Votre séquence : ",r
  $
IF r$<>b$ GOTO fin ELSE PRINT CHR$(7);CHR
  $(7)
a$="EXACT. ATTENTION...":g=15:f=0:GOSUB
  lec
FOR i=1 TO 2000:NEXT i:h=h+1:IF m<h THEN
  m=h
GOTO partie
LABEL fin
CLS:a$="VOTRE SCORE EST DE"+STR$(h):g=5:f
  =0:GOSUB lec
a$="MEILLEUR SCORE ACTUEL"+STR$(m):g=8:f=
  0:GOSUB lec
a$="AUTRE PARTIE ( O/N ) ?":g=11:f=0:GOSU
  B lec
r$=INPUT$(1):r$=UPPER$(r$):IF r$="O" GOTO
  début ELSE END

LABEL lec
FOR i=1 TO LEN(a$)
r$=MID$(a$,i,1):co=(i MOD 13)+2:a=LEN(a$)
:f=f+2
t=16:mo=2:ef=1:GOSUB aff
NEXT i
g=g+4:f=0:PRINT CHR$(7)
RETURN

LABEL aff
PRINT AT (f;g) COLOR(co) POINTS(t) MODE(m
  o) EFFECTS(ef) r$
RETURN
```

Labyrinthix

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez difficile.

LE JEU :

Vous êtes enfermé dans un labyrinthe sans issue, et il faut réussir à deviner, le plus vite possible, à quel emplacement vous vous trouvez. Dans la version 1, vous ne vous déplacez pas, quels que soient les essais que vous faites, alors que dans la version 2, si le déplacement est possible, il sera effectué.

LE PROGRAMME GW-BASIC :

Le type de jeu (1 ou 2) est choisi en ligne 30.

Le labyrinthe est enregistré dans une table A\$ en ligne 40 (on pourra modifier les données pour en dessiner un autre, aux lignes 60 à 170).

Le dessin du labyrinthe est effectué par une boucle I, en ligne 220.

La position du joueur est tirée par hasard et testée en ligne 200.

Le déplacement est entré en ligne 230.

La boucle I (250-310) contrôle si le déplacement est possible. Le résultat est affiché en ligne 320.

Les lignes 360 et 370 permettent d'entrer une solution, contrôlée en ligne 380 et 390.

POUR JOUER :

On lance le programme par RUN, puis on choisit la version (joueur fixe ou mobile) en tapant 1 ou 2.

Le labyrinthe est alors dessiné ; on entre le déplacement sous forme d'une suite de lettres, H pour haut, B pour bas, D pour droite, G pour gauche.

On tapera, par exemple HHHDD. Le nombre de cases libres avant la rencontre d'un mur est alors affiché. En version 2, le déplacement est effectué.

A la question "SOLUTION (O/N)", on répondra O si l'on pense avoir trouvé la position du joueur. On entre ensuite les lettres correspondant à la ligne et à la colonne, et un message vient donner confirmation. On tape CTRL et Break pour quitter le programme.

EXTENSIONS POSSIBLES :

On pourra facilement modifier le dessin du labyrinthe, faire apparaître le joueur, imaginer de le diriger vers une issue, etc.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM LABYRINTHIX
20 SCREEN 0,1:WIDTH 40:COLOR 1,7:RANDOMIZE TIMER:KEY OFF
30 PRINT:INPUT "VERSION (1) OU (2) :";VE
:IF VE<>1 AND VE<>2 THEN 20
40 CLS:DIM A$(13):B$(0)="" :B$(1)="S"
50 A$(0)=" ABCDEFGHIJKLMNOPQR"
60 A$(1)=" #####"
70 A$(2)="0# # # # # "
80 A$(3)="1# # # # # # # "
90 A$(4)="2# # # # # # "
100 A$(5)="3# # # # # # # "
110 A$(6)="4# # # # # # # "
120 A$(7)="5 # # # # # # "
130 A$(8)="6# # # # # # "
140 A$(9)="7# # # # # # # "
150 A$(10)="8# # # # # # # "
160 A$(11)="9# # # # # # "
170 A$(12)=" #####"
180 X=INT(RND*12+1)
190 Y=INT(RND*20+2)
200 IF MID$(A$(X),Y,1)="#" THEN 190
210 A=X:B=Y:T=1
220 CLS:COLOR 1:FOR I=0 TO 12:PRINT A$(I):NEXT I
230 LOCATE 15,1:COLOR 0:INPUT"VOTRE DEPLACEMENT :";R$
240 S=0:R$=R$+" "
250 FOR I=1 TO LEN(R$)
260 S$=CHR$(ASC(MID$(R$,I,1))+32*(MID$(R$,I,1)>"a"))
270 X0=X:X=X+(S$="H")-(S$="B")
280 Y0=Y:Y=Y+(S$="G")-(S$="D")
290 IF MID$(A$(X),Y,1)="#" OR S$=" " THEN X=X0:Y=Y0:GOTO 320
300 S=S+1:SOUND 500+200*S,2
310 NEXT I
320 COLOR 4:PRINT:PRINT"VOUS AVEZ ";S;" CASE";B$(-(S>1));" LIBRE";B$(-(S>1));" "
330 COLOR 0:PRINT:INPUT"SOLUTION (O/N)";R$:R$=R$+" ":R$=CHR$(ASC(R$)+32*(R$>"a"))
340 IF LEFT$(R$,1)<>"O" THEN T=T+1:LOCATE 15,1:PRINT SPACE$(30):IF VE=2 THEN 230
ELSE X=A:Y=B:GOTO 230
350 IF VE=2 THEN A=X:B=Y
360 INPUT"LIGNE :";X:X=X+2
370 INPUT"COLONNE :";R$:Y=ASC(R$)-62
380 IF X=A AND Y=B THEN PRINT"BRAVO...";T;" ESSAIS":END
390 PRINT"PERDU . C'ETAIT LIGNE ";A-2;" COLONNE ";CHR$(62+B)

```

REM Labyrinthix

```

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 FIXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "LABYRINTHIX"
WINDOW OPEN
RANDOMIZE TIME
PRINT AT (1;3) "Version 1 ou 2 ?":r$=INPUT$(1):ve=VAL(r$)
CLS:READ a:d=a:DIM l(a,a):y=11:h=10:c$=STRING$(40,32)
FOR i=1 TO a:PRINT AT (3*i+5;1) CHR$(64+i):NEXT i
FOR i=1 TO a:FOR j=1 TO a:l(i,j)=0:NEXT j,i
REPEAT
FOR i=1 TO a
READ b:x=24*b+24:l(1+((y-11)/h),b)=1
BOX x;184-y,24,h COLOR(2) FILL ONLY WITH 5
NEXT i
READ a:y=y+h-1:z=20:c=64+(y-11)/h:IF c<74 THEN PRINT AT (4;c-61) CHR$(c)
UNTIL a=0
REPEAT:x=RND(c-65):y=RND(d-1):UNTIL l(x,y)<>1
a=x:b=y:t=0
LABEL debut
PRINT AT (1;18) c:s=0:PRINT CHR$(7)
INPUT AT (1;18) "Déplacement (HHHGGHBB...)":r$=UPPER$(r$)
FOR i=1 TO LEN(r$)
x0=x:x=x+(MID$(r$,i,1)="H")-(MID$(r$,i,1)="B")
y0=y:y=y+(MID$(r$,i,1)="G")-(MID$(r$,i,1)="D")
IF l(x,y)=1 THEN x=x0:y=y0 ELSE s=s+1
NEXT i
PRINT AT (1;18) "Vous avez " s " cases libres ";STRING$(40,32)
PRINT AT (1;20) "Solution ( O/N ) ?":r$=INPUT$(1)
r$=UPPER$(r$):PRINT AT (1;20) c$
IF r$<>"O" THEN t=t+1:IF ve=1 THEN x=a:y=b:GOTO debut ELSE GOTO debut
IF ve=2 THEN a=x:b=y
PRINT AT (1;18) c$:INPUT AT (1;18) "Ligne : ";x$:u=ASC(UPPER$(x$))-64
PRINT AT (1;18) c$:INPUT AT (1;18) "Colonne : ";y$:u=ASC(UPPER$(x$))-64
IF a=u AND b=v THEN PRINT AT (1;20) "Bravo. Trouvé en " t+1 " essais"
IF a<>u OR b<>v THEN PRINT AT (1;20) "Perdu. C'était ligne " CHR$(64+a) " - colonne " CHR$(64+b)
r$=INPUT$(1):END

```

```

DATA 20,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
DATA 4,1,5,13,20,15,1,3,5,7,8,9,10,11,12,13,15,16,17,18,20
DATA 7,1,3,5,12,13,15,20,12,1,3,8,9,10,12,13,15,17,18,19,20
DATA 9,1,3,4,5,6,10,15,18,20,9,1,3,8,10,12,14,15,18,20
DATA 9,1,3,5,7,8,10,12,18,20,9,1,5,7,12,14,16,17,18,20
DATA 10,1,3,4,5,7,9,11,12,14,20,8,1,9,14,15,16,17,18,20
DATA 20,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,0

```


Belote

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de cartes.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME GW-BASIC :

En ligne 40, on crée un tableau pouvant contenir les 20 cartes.

La boucle I (110-170) assure le tirage aléatoire de 20 nombres, pris parmi 32 ; la boucle J (140-160) vérifie que l'on ne réalise pas deux fois le même tirage.

La deuxième boucle I (190-320) assure ensuite l'affichage, les noms des cartes étant affichés en clair, et la couleur étant décodée et affichée en clair, grâce aux lignes 200 à 300.

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 redéfinissant les caractères du PC ; 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).

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM BELOTE
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF:CLS
40 DIM A(20):C$(0)="VA":C$(1)="DA":C$(2)
  ="RO":C$(3)="AS"
50 K=1:F(0)=4:F(1)=2:F(2)=1:F(3)=6
60 A$=CHR$(200):B$=CHR$(201):C$=CHR$(205
  ):D$=CHR$(186):E$=CHR$(187):F$=CHR$(188)
70 FOR I=1 TO 21:G$=G$+C$:NEXT
80 LOCATE 11,10:PRINT B$+G$+E$
90 LOCATE 12,10:PRINT D$+" PATIENTEZ , J
  E TIRE "+D$
100 LOCATE 13,10:PRINT A$+G$+F$
110 FOR I=1 TO 20
120 A(I)=1+INT(RND*32)
130 IF I=1 THEN 170
140 FOR J=1 TO I-1
150 IF A(I)=A(J) THEN 120
160 NEXT J
170 NEXT I
180 CLS:G$=LEFT$(G$,6)
190 FOR I=1 TO 20
200 S$=CHR$(4):T=INT((A(I)-1)/8)
210 S=1+A(I)-8*INT(A(I)/8)
220 R$=STR$(S+6)+" ":R$=RIGHT$(R$,LEN(R
  $)-1)
230 R$=LEFT$(R$,2)
240 IF S>=5 THEN R$=C$(S-5)
250 IF A(I)<=24 THEN S$=CHR$(5)
260 IF A(I)<=16 THEN S$=CHR$(6)
270 IF A(I)<=8 THEN S$=CHR$(3)
280 COLOR 0:LOCATE 3*K,10*INT((I-1)/5)+1
  :PRINT B$+G$+E$
290 LOCATE 3*K+1,10*INT((I-1)/5)+1:PRINT
  D$+" "+R$;" "":COLOR F(T):PRINT S$;:COL
  OR 0:PRINT " "+D$
300 LOCATE 3*K+2,10*INT((I-1)/5)+1:PRINT
  A$+G$+F$
310 K=K+1:K=K+5*(K=6):SOUND INT(4000*RND
  +50 ),1
320 NEXT I

```

```

REM Belote

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "BELOTE"
WINDOW OPEN
RANDOMIZE TIME
s$(5)="VALET":s$(6)="DAME":s$(7)="ROI":s$
(8)="AS":f(1)=11:f(2)=2:f(3)=14:f(4)=4:t$
=" "
DIM a(20):k=1:s$(1)="TREFLE":s$(2)="COEUR
":s$(3)="CARREAU":s$(4)="PIQUE"
FOR i=1 TO 20
LABEL tirage
a(i)=RND(32)
PRINT AT (18;10) COLOR(RND(15)) POINTS(20
  ) "PATIENTEZ, JE TIRE"
FOR j=1 TO i-1
IF a(i)=a(j) GOTO tirage
NEXT j,i
CLS
FOR i=1 TO 20
t=-((a(i)<=8)+(a(i)<=16)+(a(i)<=24)+(a(i)
  <=32))
s=1+a(i)-8*INT(a(i)/8)
r$=STR$(s+6)+" ":r$=RIGHT$(r$,LEN(r$)-1)
r$=LEFT$(r$,2)
IF s>=5 THEN r$=s$(s)
l=13-LEN(r$)-LEN(s$(t))
PRINT AT (19*INT((i-1)/5)+2;2*k) COLOR(f(
  t)) MODE(4) " " r$ " " s$(t) LEFT$(t$,1)
k=k+1:PRINT CHR$(7);IF k=6 THEN k=1
NEXT i
r$=INPUT$(1)
END

```

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 PC) 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 GW-BASIC :

La ligne 40 permet d'obtenir l'entier de départ compris entre 5 et 20.

Une boucle I (90-180) permet d'entrer successivement les 5 opérations.

4 sous-programmes sont utilisés pour les calculs (aux lignes 260 à 370).

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 170).

La boucle I (200-230) affiche les 5 opérations utilisées, ainsi que les opérateurs correspondants.

POUR JOUER :

On frappe RUN pour lancer le programme.

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

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

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM UN TAQUIN DE CALCUL
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF:CLS
40 X=INT(RND*16+5):SOUND INT(4000*RND+500),2
50 PRINT"NOMBRE INITIAL :";X
60 PRINT"ENTRER 5 OPERATIONS"
70 PRINT"Sous LA FORME +5 , /2 , *4 ..."
80 F=X:DIM R$(5)
90 FOR I=1 TO 5
100 LOCATE 7,1:INPUT"VOTRE OPERATION :";
R$(I)
110 LOCATE 7,17:PRINT "          "
120 A=ASC(LEFT$(R$(I),1))-41
130 IF A<=0 OR A=3 OR A=5 OR A>6 THEN 100
140 IF A>3 THEN A=(A/2)+1
150 ON A GOSUB 260,280,300,320
160 LOCATE 11,15:PRINT "          ":SO
UND INT(RND*4000+500),2
170 LOCATE 11,1:PRINT"NOMBRE OBTENU :";F
180 NEXT I
190 PRINT:PRINT"OPERATIONS :";
200 FOR I=1 TO 5
210 X=INT(RND*5+1):IF T(X)=1 THEN 210 EL
SE T(X)=1
220 COLOR(INT(RND*7)):PRINT R$(X);" ";
230 NEXT I
240 COLOR 0:PRINT:PRINT:PRINT"BON COURAG
E...."
250 PLAY "CCCDEEDCEDDC":END
260 F=F*VAL(RIGHT$(R$(I),LEN(R$(I))-1))
270 RETURN
280 F=F+VAL(RIGHT$(R$(I),LEN(R$(I))-1))
290 RETURN
300 F=F-VAL(RIGHT$(R$(I),LEN(R$(I))-1))
310 RETURN
320 F=F/VAL(RIGHT$(R$(I),LEN(R$(I))-1))
330 RETURN

```

```

REM Un taquin de calcul

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "UN TAQUIN DE CALCUL"
WINDOW OPEN
SET FONT(1) POINTS(10)
RANDOMIZE TIME
x=RND(16)+4
PRINT AT (1;2) "Le nombre initial est : "
COLOR(10) POINTS(16) x
PRINT AT (1;5) "Entrez cinq opérations so
us la forme +5 , /2 , *4 , ...."
f=x
FOR i=1 TO 5
LABEL nouveau
PRINT AT (1;8) "Quelle opération : ";INP
UT "",r$(i)
PRINT AT (1;8) STRING$(40,32):t(i)=0
a=ASC(LEFT$(r$(i),1))-41
IF a<=0 OR a=3 OR a=5 OR a>6 THEN PRINT C
HR$(7):GOTO nouveau
IF a>3 THEN a=(a/2)+1
ON a GOSUB multi,addit,soust,divis
PRINT AT (1;12) POINTS(20) STRING$(40,32)
:f$=STR$(f)
PRINT AT (1;12) COLOR(4) POINTS(20) "Le n
ombre obtenu est : " LEFT$(f$,1);
f$=RIGHT$(f$,LEN(f$)-1)
FOR j=1 TO LEN(f$)
r$=MID$(f$,j,1):PRINT POINTS(20) CHR$(ASC
(r$));
NEXT j
NEXT i
PRINT AT (1;15) COLOR(14) "Les opérations
étaient : ";
FOR i=1 TO 5
LABEL tirage
x=RND(5):IF t(x)=1 THEN GOTO tirage ELSE
t(x)=1
PRINT COLOR(x+9) POINTS(16) r$(x);" ";
NEXT i
PRINT AT (1;18) "Bon courage..... "
r$=INPUT$(1):END

LABEL multi
f=f*VAL(RIGHT$(r$(i),LEN(r$(i))-1))
RETURN

LABEL addit
f=f+VAL(RIGHT$(r$(i),LEN(r$(i))-1))
RETURN

LABEL soust
f=f-VAL(RIGHT$(r$(i),LEN(r$(i))-1))
RETURN

LABEL divis
f=f/VAL(RIGHT$(r$(i),LEN(r$(i))-1))
RETURN

```

Hanoi

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

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

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

LE PROGRAMME GW-BASIC :

La variable E\$, en ligne 40, servira à dessiner les disques, et D\$ dessine les socles.

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

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

EXTENSIONS POSSIBLES :

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

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM HANOI
20 SCREEN 0,1:WIDTH 40:COLOR 7,0:KEY OFF
:CLS
30 DIM A$(3):A$(1)="9":A$(2)="9":A$(3)="
9":S=1
40 D$="":E$="":FOR I=1 TO 9:D$=D$+CHR$(2
19):E$=E$+CHR$(178):NEXT I
50 INPUT"NOMBRE DE TOURS ( < 9 ):";N
60 IF N>9 THEN 50
70 FOR I=0 TO N-1
80 A$(1)=A$(1)+RIGHT$(STR$(N-I),1)
90 NEXT I:F$=A$(1)
100 CLS:COLOR 7:LOCATE 1,15:PRINT "ESSAI
";S
110 LOCATE 15,1:COLOR 13:PRINT D$:LOCATE
15,13:PRINT D$:LOCATE 15,25:PRINT D$
120 LOCATE 15,5:COLOR 10,13:PRINT"1":LOC
ATE 15,17:PRINT"2":LOCATE 15,29:PRINT"3"
130 FOR I=0 TO N-1
140 FOR J=1 TO 3
150 B$=A$(J)
160 IF LEN(B$)>I+1 THEN L=VAL(MID$(B$,I+
2,1)):LOCATE 14-I,12*J-10:COLOR L,0:PRIN
T LEFT$(E$,L)
170 NEXT J
180 NEXT I
190 COLOR 4:LOCATE 17,1:INPUT"TOUR DE DE
PART :";D
200 IF D>3 OR D<1 THEN 190
210 IF A$(D)="9" THEN 190
220 PRINT:INPUT"TOUR D'ARRIVEE :";A
230 IF A=D OR A>3 OR A<1 THEN 190
240 B$=A$(D)
250 C$=A$(A)
260 IF MID$(B$,LEN(B$),1)>MID$(C$,LEN(C$
),1) THEN 190
270 A$(A)=C$+MID$(B$,LEN(B$),1)
280 A$(D)=LEFT$(B$,LEN(B$)-1)
290 S=S+1:PLAY"CDE":IF A$(3)<>F$ THEN 10
0
300 COLOR 7:LOCATE 22,1:PRINT "C'est fin
i...":PLAY "CGE"

```

REM Les tours de Hanoi

```

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "LES TOURS DE HANOI"
WINDOW OPEN
d$=STRING$(20,"="):e$=STRING$(18,"#"):f$=
STRING$(40,32)
a$(1)="9":a$(2)="9":a$(3)="9":u=26
LABEL début
PRINT AT (1;3) f$ AT (1;3) "Nombre de tou
rs ( moins de 8 ) : ";
INPUT "",n:n=VAL(n$)
IF n<>INT(n) OR n<0 OR n>8 GOTO début
FOR i=0 TO n-1:a$(1)=a$(1)+RIGHT$(STR$(n-
i),1):NEXT i
t$=a$(1)
LABEL dessin
PRINT AT (1;13)::PRINT CHR$(27)+"d"
PRINT COLOR(14) MODE(4) AT (2;13) d$ AT (
u+2;13) d$ AT (2*u+2;13) d$
PRINT AT (11;13) "1" AT (11+u;13) "2" AT
(11+2*u;13) "3"
FOR i=0 TO n-1
FOR j=3 TO 1 STEP -1
b$=a$(j):f$=VAL(MID$(b$,i+2,1))
IF LEN(b$)>i+1 THEN PRINT AT (26*j)-f-15;1
2-i) COLOR(f+8) LEFT$(e$,f+(f AND p<4))
NEXT j,i
IF a$(3)=t$ THEN PRINT AT (1;19) "Bravo..
..":r$=INPUT$(1):END
LABEL départ
PRINT AT (1;15) f$ AT (1;17) f$ AT (1;15)
"Tour de départ : ";
d$=INPUT$(1):d=VAL(d$):PRINT COLOR(2) d
IF d<>INT(d) OR d>3 OR d<1 OR a$(d)="9" G
OTO début
PRINT AT (1;17) "Tour d'arrivée : ";
a$=INPUT$(1):a=VAL(a$):PRINT COLOR(4) a
IF a<>INT(a) OR a=d OR a>3 OR a<1 GOTO dé
part
b$=a$(d):c$=a$(a)
IF MID$(b$,LEN(b$),1)>MID$(c$,LEN(c$),1)
THEN STOP:GOTO départ
a$(a)=c$+MID$(b$,LEN(b$),1):a$(d)=LEFT$(b
$,LEN(b$)-1)
PRINT CHR$(7);CHR$(7);CHR$(7);
GOTO dessin

```

Plus grand, plus petit

REFERENCES :

Difficulté de programmation : facile.

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

Difficulté du jeu : difficile.

LE JEU :

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

LE PROGRAMME GW-BASIC :

La ligne 50 interdit de demander des nombres de plus de 8 chiffres.

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

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

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

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

POUR JOUER :

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

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

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

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

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

Si l'on a trouvé le PC affichera VOUS AVEZ TROUVE... EN... ESSAIS. Puis AUTRE PARTIE (O/N) ; on fera O 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 PC qui tiendra alors le rôle de décodeur.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM PLUS GRAND , PLUS PETIT
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF
40 CLS
50 INPUT "NOMBRE DE CHIFFRES :";C
60 IF C>8 THEN 40
70 E=1:F=1
80 X%=STR$(INT(RND*9^C+10^(C-E)))
90 X%=RIGHT$(X%,LEN(X$)-1)
100 COLOR 0:LOCATE 4,1:PRINT"ESSAI NO.
NOMBRE ESSAYE      >      <      "
110 LOCATE 5,1:COLOR 1:FOR J=1 TO 40:PRI
NT CHR$(219);:NEXT J
120 LOCATE F+5,1:COLOR 6:PRINT TAB(4);E;
TAB(10):INPUT P$
130 IF P$=X$ THEN 220
140 A=0:B=0
150 FOR I=1 TO C
160 IF MID$(P$,I,1)>MID$(X$,I,1) THEN A=
A+1
170 IF MID$(P$,I,1)<MID$(X$,I,1) THEN B=
B+1
180 NEXT I
190 LOCATE F+5,28:PRINT A;TAB(34);B
200 E=E+1:F=F+1:IF F=18 THEN F=1
210 GOTO 120
220 PRINT:PRINT"VOUS AVEZ TROUVE ";X$;"
EN ";E;" ESSAIS":PLAY"CGE"
230 PRINT:PRINT"AUTRE PARTIE (O/N)":R$=I
NPUT$(1)
240 IF R$="O" OR R$="o" THEN 40

```

```

REM Plus grand, plus petit

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW TITLE "PLUS GRAND, PLUS PETIT"
WINDOW OPEN
RANDOMIZE TIME
PRINT AT (1;3) "Nombre de chiffres ( <9 )
: ";
c%=INPUT$(1):c=VAL(c%):IF c>8 THEN RUN EL
SE CLS
e=1:g=1:f%=STRING$(8,32)
x%=STR$(INT(RND*9^c+10^(c-e))):x%=RIGHT$(
x%,LEN(x$)-1)
BOX 0;164,640,30 WIDTH 5 COLOR(13) FILL W
ITH 4
PRINT AT (2;3) COLOR(10) POINTS(16) MODE(
2) "ESSAI NO. NOMBRE ESSAYE PLUS GRANDS
PLUS PETITS"
LINE 130;160,130;0 WIDTH 5:LINE 310;160,3
10;0 WIDTH 5
LINE 470;160,470;0 WIDTH 5
WHILE p$<>x$
PRINT AT (1;g+5) e AT (19;g+5) f$ AT (48;
g+5) f$ AT (66;g+5) f$
INPUT AT (19;g+5) "? ",p$
a=0:b=0
FOR i=1 TO c
IF MID$(p$,i,1)>MID$(x$,i,1) THEN a=a+1
IF MID$(p$,i,1)<MID$(x$,i,1) THEN b=b+1
NEXT i
PRINT AT (48;g+5) COLOR(2) a AT (66;g+5)
COLOR(4) b
e=e+1:g=g+1:IF g=16 THEN g=1
WEND
PRINT:PRINT "Vous avez trouvé en " e " es
sais"
PRINT:PRINT "Une autre partie ( O/N )"
r$=INPUT$(1):IF UPPER$(r$)="O" THEN RUN
END

```


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 GW-BASIC :

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

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

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

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

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

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

POUR JOUER :

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

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

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM REINES SUR L'ECHIQUEUR
20 SCREEN 0,1:WIDTH 40:COLOR 0,7:KEY OFF
:CLS
30 INPUT"TAILLE DE L'ECHIQUEUR :";S
40 DIM D(S)
50 I=1
60 D(I)=1
70 IF I=1 THEN 110
80 FOR J=1 TO I-1
90 IF D(I)=D(J) OR (ABS(D(I)-D(J))=I-J)
THEN 240
100 NEXT J
110 I=I+1:IF I<=S THEN 60
120 CLS
130 FOR F=1 TO S
140 COLOR 0:LOCATE 1,2*F+1:PRINT F
150 LOCATE 2*F+1,1:PRINT F
160 FOR K=1 TO S
170 LOCATE 2*F+1,2*K+2:PRINT "."
180 NEXT K
190 LOCATE 2*F+1,2*D(F)+2:COLOR 4:PRINT
"#"
200 LOCATE 22,F+1:COLOR 1:PRINT RIGHT$(S
TR$(D(F)),1)
210 NEXT F
220 COLOR 5:PLAY"CGE":LOCATE 23,1:PRINT"
APPUYEZ SUR UNE TOUCHE":R$=INPUT$(1)
230 I=I-1
240 D(I)=D(I)+1
250 IF D(I)<=S THEN 70
260 I=I-1
270 IF I<>0 THEN 240
280 PRINT "C'est fini..."

```

```

REM Reines sur l'échiquier
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW TITLE "REINES SUR L'ECHIQUEUR"
WINDOW OPEN
PRINT AT (1;3) "Vous voulez quelle taille
d'échiquier ( <10 ) : ";
s$=INPUT$(1):s=VAL(s$):PRINT s:IF s>9 THE
N RUN
i=1:k=1
LABEL début
d(i)=1
LABEL suite
FOR j=1 TO i-1
IF d(i)=d(j) OR ABS(d(i)-d(j))=i-j GOTO p
rogrès
NEXT j
i=i+1:IF i<=s GOTO début
CLS:BOX 16;178-18*s-10,32*s+26,18*s+20 CO
LOR(8) FILL WITH 8
FOR f=1 TO s
FOR j=1 TO s
BOX 32*f-4;178-18*j,32,18 COLOR(((f+j) MO
D 2)+6) FILL WITH 7
NEXT j,f
FOR f=1 TO s
PRINT AT (1;2*f+2) CHR$(64+f) AT (4*f+2;1
) CHR$(48+f)
PRINT AT (4*d(f)+2;2*f+2) POINTS(16) MODE
(2) " "
PRINT AT (50;3) "Solution " k
PRINT AT (50+f;5) RIGHT$(STR$(d(f)),1);
NEXT f
k=k+1
PRINT AT (50;10) "Appuyer sur une touche"
:r$=INPUT$(1)
i=i-1
LABEL progrès
d(i)=d(i)+1
IF d(i)<=s GOTO suite
i=i-1
IF i<>0 GOTO progrès
PRINT AT (50;15) "C'est fini...."
r$=INPUT$(1):END

```

Ordre

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de réflexion.

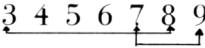
Difficulté du jeu : moyenne.

LE JEU :

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

B A D G H I J C F E

0 1 2 3 4 5 6 7 8 9 : permutation 83



B A D F H I J E G C

0 1 2 3 4 5 6 7 8 9 : résultat

LE PROGRAMME GW-BASIC :

Les boucles emboîtées I (40-100) et J (60-80) définissent une variable alphanumérique A\$, contenant les 10 premières lettres de l'alphabet dans le désordre.

La ligne 120 affiche A\$, et la ligne 130 affiche les numéros de lettres de l'alphabet pour la permutation.

On entre la permutation en ligne 170.

Les lignes 180 et 190 transforment la variable alphanumérique entrée en une variable numérique.

La ligne 200 appelle une première fois le sous-programme d'inversion.

La ligne 230 appelle une deuxième fois le sous-programme d'inversion.

La ligne 240 comptabilise les essais.

La ligne 160 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.

VERSION BASIC 2 AMSTRAD :

Si le principe du jeu reste le même, il est fait ici utilisation de la souris pour choisir les lettres à permuter. On devra cliquer sur le bouton de gauche pour choisir la lettre de départ (qui change de couleur) et sur celui de droite pour choisir celle d'arrivée (qui apparaît dans une autre couleur). On peut d'ailleurs recommencer ces deux opérations. On devra ensuite cliquer sur les deux boutons simultanément pour effectuer la permutation (si elle est possible).

```
10 REM ORDRE
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF:CLS:A$=""
40 FOR I=1 TO 10
50 R$=CHR$(INT(RND*10+65))
60 FOR J=1 TO I-1
70 IF R$=MID$(A$,J,1) THEN 50
80 NEXT J
90 A$=A$+R$
100 NEXT I
110 S=1
120 LOCATE 5,1:PRINT"RESULTAT ":";COLOR
3,4:PRINT A$:COLOR 1,7
130 PRINT TAB(11);"0123456789"
140 COLOR 0:LOCATE 10,1:PRINT"ESSAI NUME
RO ";S
150 LOCATE 13,34:PRINT"      "
160 IF A$="ABCDEFGH IJ" THEN PRINT"TROUVE
EN ";S;" COUPS":PLAY"CGE":END
170 LOCATE 13,1:INPUT"INVERSION (I ET J
SOUS FORME IJ )";R$
180 I=ASC(LEFT$(R$,1))-47
190 J=ASC(RIGHT$(R$,1))-47
200 GOSUB 260
210 IF LEFT$(R$,1)="0" OR LEFT$(R$,1)="9
" THEN 230
220 I=I-1:J=J+2
230 GOSUB 260
240 S=S+1
250 GOTO 120
260 B$=""
270 FOR T=1 TO LEN(A$)
280 IF T=I THEN N$=MID$(A$,T,1)
290 IF T=J THEN M$=MID$(A$,T,1)
300 NEXT T
310 FOR T=1 TO LEN(A$)
320 IF T=I THEN B$=B$+M$:GOTO 350
330 IF T=J THEN B$=B$+N$:GOTO 350
340 B$=B$+MID$(A$,T,1):SOUND INT(RND*400
0+500),1
350 NEXT T
360 A$=B$
370 RETURN
```

```
REM Ordre
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "ORDRE"
WINDOW OPEN
RANDOMIZE TIME:a$="":s=1:DIM t(10)
FOR i=1 TO 10
LABEL tirage
r$=CHR$(INT(RND*10+65))
FOR j=1 TO i-1
IF r$=MID$(a$,j,1) GOTO tirage
NEXT j
a$=a$+r$
NEXT i
LABEL début
CLS:FOR i=1 TO 10:t(i)=0:NEXT i
p$="R E S U L T A T":co=4:f=0:ta=16:u=2:v
=2:h=6:m=1:e=4:GOSUB affiche
BOX 40;148,27*LEN(a$),30 COLOR(1) WIDTH 4
ROUNDED
p$=a$:f=1:co=2:ta=20:u=3:v=5:h=5:m=2:e=1:
GOSUB affiche
p$="":FOR i=1 TO LEN(a$):p$=p$+CHR$(64+i)
+" ";NEXT i
f=0:co=1:ta=8:u=1:v=7:h=8:e=0:m=1:GOSUB a
ffiche
FOR i=1 TO LEN(a$):BOX 31+24*i;130,24,16:
NEXT i
p$="Essai numéro "+STR$(s):k=0:h=0
f=0:co=14:ta=10:u=1:v=11:h=0:GOSUB affich
e
IF a$="ABCDEFGH IJ" THEN PRINT AT (1;17) "
Trouvé en " s " essais":r$=INPUT$(1):END
PRINT AT (1;13) "Cliquez à gauche pour le
départ, à droite pour la fin"
LABEL clic
REPEAT:a=BUTTON(1):b=BUTTON(2):UNTIL a>-1
OR (b>-1 AND k<>0)
ch=0:x=XMOUSE:y=YMOUSE:IF a+b=2 AND k<>0
AND h<>0 GOTO permutte
FOR i=1 TO LEN(a$)
ch=-i*(x>31+24*i AND x<55+24*i AND y>117
AND y<130):IF ch<>0 GOTO suite
NEXT i:GOTO clic
LABEL suite
IF k=0 THEN k=i:t(k)=1 ELSE IF a<>-1 THEN
SWAP t(k),t(i)
IF t(i)<>1 THEN IF h=0 THEN h=i:t(h)=2 EL
SE SWAP t(h),t(i)
FOR i=1 TO LEN(a$)
BOX 33+24*i;131,20,13 MODE(2) COLOR(2*t(i
)) WIDTH 3
IF t(i)=1 THEN k=i
IF t(i)=2 THEN h=i
NEXT i:GOTO clic
```

```

LABEL permutte:PRINT AT (1;18) k;h
GOSUB change
IF k<>1 AND k<>LEN(a$) THEN k=k-1:h=k+2
GOSUB change
s=s+1:GOTO début

LABEL change
b$=""
FOR t=1 TO LEN(a$)
IF t=k THEN n$=MID$(a$,t,1)
IF t=h THEN m$=MID$(a$,t,1)
NEXT t
FOR t=1 TO LEN(a$)
IF t=k THEN b$=b$+m$:GOTO suivant
IF t=h THEN b$=b$+n$:GOTO suivant
b$=b$+MID$(a$,t,1)
LABEL suivant
NEXT t:a$=b$:PRINT CHR$(7):RETURN

LABEL affiche
FOR i=1 TO LEN(p$)
PRINT AT (u*i+h;v) COLOR(co+f*i) POINTS(t
a) MODE(m) EFFECTS(e) MID$(p$,i,1)
NEXT i
RETURN

```

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 PC 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 PC 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 GW-BASIC :

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

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

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

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

La ligne 130 affiche le numéro du joueur.

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

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

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

L'impression et la fin de partie sont effectuées en ligne 290. La ligne 300 s'occupe du changement de joueur.

POUR JOUER :

Lancer l'exécution du programme par RUN.

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

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

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```
10 REM IDEM
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF:CLS
40 H=1:DIM N(2)
50 PRINT:INPUT "NIVEAU (2-7) :";N
60 U=10:P=0
70 N(1)=INT((U^N-U^(N-1))*RND+U^(N-1))
80 N(2)=N(1)
90 A=INT(2*U^(N-1)*RND+U^(N-1))
100 IF H=1 THEN P=P+1
110 CLS
120 IF N(H)<U^(N-1) THEN N(H)=U^N-ABS(N(H))
130 COLOR 3:H=2:PRINT "JOUER ";H;" ES
SAI ";P
140 COLOR 0:LOCATE 7,1:PRINT "NOMBRE :":
LOCATE 7,16:N:COLOR 5:PRINT N(H)
150 COLOR 6:LOCATE 8,10:PRINT "7654321":
COLOR 0
160 LOCATE 11,1:INPUT "VOTRE JEU ( SOUS
FORME NC ) :";R$:IF LEN(R$)<>2 OR VAL(RI
GHT$(R$,1))>N THEN 160
170 N(H)=N(H)+INT(VAL(LEFT$(R$,1))*U^(VA
L(RIGHT$(R$,1))-1))
180 N(H)=N(H)-A
190 IF N(H)>U^N THEN N(H)=N(H)-U^N
200 N(H)=INT(N(H))
210 LOCATE 14,1:PRINT "RESULTAT :";:COLO
R 5:PRINT N(H)
220 PLAY "CDEFG"
230 FOR J=1 TO 2200:NEXT J
240 R$=STR$(N(H))
250 R$=RIGHT$(R$,LEN(R$)-1)
260 FOR I=1 TO LEN(R$)-1
270 IF MID$(R$,I,1)<>MID$(R$,I+1,1) THEN
300
280 NEXT I
290 COLOR 3:H=2:PRINT:PRINT "BRAVO, JOUE
UR ";H:PLAY "CGE":END
300 H=H+1:IF H>=3 THEN H=1
310 GOTO 100

REM Idem
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW TITLE "IDEM"
WINDOW OPEN
RANDOMIZE TIME
PRINT AT (1;3) "Niveau ( 2 à 7 ) : ";
n$=INPUT$(1):n=VAL(n$):IF n<>INT(n) OR n<
2 OR n>7 THEN RUN ELSE CLS
u=10:p=0:h=2
n1=INT((u^n-u^(n-1))*RND+u^(n-1)):n(1)=n1
:n(2)=n1
a=INT(2*u^(n-1)*RND+u^(n-1))
LABEL début
h=ABS(h-3):p=p-(h=1):r$="":CLS
IF n(h)<u^(n-1) THEN n(h)=u^n-ABS(n(h))
PRINT AT (1;3) COLOR(2*h) "Joueur " h "
- Essai "p
PRINT AT (1;6) "Nombre : " TAB(18-n) n(h)
PRINT AT (12;7) COLOR(14) "7654321"
BOX 70;130,80,30
LABEL entrée
ELLIPSE 270;113,40,0.3 FILL WITH 0
PRINT AT (1;10) "Votre jeu ( sous forme N
C ) : ";:INPUT "",r$
IF LEN(r$)<>2 OR VAL(RIGHT$(r$,1))>n GOTO
entrée
n(h)=n(h)+INT(VAL(LEFT$(r$,1))*u^(VAL(RIG
HT$(r$,1))-1))
n(h)=n(h)+a:IF n(h)>u^n THEN n(h)=n(h)-u^
n
n(h)=ABS(INT(n(h)))
PRINT AT (1;13) "Résultat : " n(h)
FOR j=1 TO 10000:NEXT j
r$=STR$(n(h)):r$=RIGHT$(r$,LEN(r$)-1)
IF LEN(r$)<>n GOTO début
FOR i=1 TO LEN(r$)-1
IF MID$(r$,i,1)<>MID$(r$,i+1,1) GOTO débu
t
NEXT i
PRINT AT (1;15) "Bravo, joueur " COLOR(2*
h) POINTS(16) h
r$=INPUT$(1):END
```

Permutations

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion.

Difficulté du jeu : moyenne.

LE JEU :

Le PC 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 GW-BASIC :

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 130 contrôle le nombre de lettres de B\$.

Les affichages sont effectués aux lignes 150, 160 et 170.

Le retournement est fait, lettre par lettre, dans la boucle I (190-210) ; le résultat est placé dans B\$ en ligne 220, affiché par la ligne 230 et comparé au résultat final en ligne 240.

Le nombre d'essais sera affiché en fin de partie, grâce à la ligne 250.

POUR JOUER :

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

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

VERSION BASIC 2 AMSTRAD :

On tape un nombre de lettres de 1 à 12. L'affichage suivant présente l'ordre initial (DEBUT) et l'ordre à venir (FIN). Les différentes permutations possibles sont affichées dans des carrés. On clique sur la permutation à réaliser à l'aide du bouton de gauche et on valide le choix définitif avec le bouton de droite. On obtient alors le résultat, et l'on recommence jusqu'à obtenir l'ordre correct. On répond « O » à la question « AUTRE PARTIE » si l'on veut rejouer.

```
10 REM PERMUTATIONS
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF:CLS
40 PRINT:INPUT"NOMBRE DE LETTRES ( DE 1
A 12 )":N
50 IF N>12 THEN 40
60 A$="ABCDEFGHIJKL":C$="          ":B
$=""
70 CLS
80 X=INT(RND*N+1)
90 IF MID$(A$,X,1)=" " THEN 80
100 B$=MID$(A$,X,1)+B$
110 MID$(C$,X,1)=MID$(A$,X,1)
120 MID$(A$,X,1)=" "
130 S=1:IF LEN(B$)<>N THEN 80
140 C$=LEFT$(C$,LEN(B$))
150 LOCATE 5,1:PRINT"DEBUT  ":;:COLOR 1:
PRINT B$:COLOR 0
160 LOCATE 7,1:PRINT"FIN    ":;:COLOR 4:
PRINT C$:COLOR 0
170 LOCATE 11,1:PRINT"PERMUTATION NUMERO
":S:PRINT:INPUT"VOTRE QUANTITE ":L:IF L
>N THEN 170
180 S=S+1:A$=""
190 FOR I=1 TO L
200 A$=A$+MID$(B$,L-I+1,1)
210 NEXT I
220 B$=A$+RIGHT$(B$,LEN(B$)-L)
230 PLAY"CDE":LOCATE 17,1:PRINT"RESULTAT
":;:COLOR 5:PRINT B$:COLOR 0
240 IF B$<>C$ THEN 170
250 PLAY"CGE":PRINT:PRINT"TROUVE EN ":S-
1;" ESSAIS"
260 PRINT:PRINT"AUTRE PARTIE ( O/N )":R$
=INPUT$(1)
270 IF R$="O" OR R$="o" THEN 40
```

REM Permutations

```
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "PERMUTATIONS"
WINDOW OPEN
RANDOMIZE TIME
INPUT AT (1;3) "Nombre de lettres ( 1 a 1
2 ) : ",n$
c=LET n=VAL(n$):IF c<>0 OR n<>INT(n) OR n
<1 OR n>12 THEN RUN
a$="ABCDEFGHIJKL":c$=STRING$(LEN(a$)," ")
:b$="":CLS
LABEL tirage
REPEAT:x=RND(n):UNTIL MID$(a$,x,1)<>" "
b$=MID$(a$,x,1)+b$:MID$(c$,x,1)=MID$(a$,x
,1)
MID$(a$,x,1)=" ":s=1:IF LEN(b$)<>n GOTO t
irage
c$=LEFT$(c$,LEN(b$))
PRINT AT (1;4) "DEBUT   : ":t=16:p$=b$:l=
9:h=4:GOSUB aff
PRINT AT (1;7) "FIN     : ":t=16:p$=c$:l=
9:h=7:GOSUB aff
LABEL début
PRINT AT (1;10) "Permutation numéro " s:k
=0
PRINT AT (1;12) "Clic gauche sur le nombr
e à permuter, droit pour valider"
LABEL clic
FOR i=1 TO n
BOX 24*i-3;70,24,17  COLOR(-(i=k)+1) FILL
WITH 8
PRINT AT (3*i+1;14) MID$(STR$(i),2)
NEXT i
REPEAT:a=BUTTON(1):b=BUTTON(2):UNTIL a>-1
OR (b>-1 AND k<>0)
ch=0:x=XMOUSE:y=YMOUSE:IF b<>-1 GOTO suite
e
FOR i=1 TO n
ch=-i*(x>24*i-3 AND x<24*i+21 AND y>56 AN
D y<75):IF ch<>0 THEN k=i:GOTO clic
NEXT i:GOTO clic
LABEL suite
s=s+1:a$="":PRINT AT (1;12) STRING$(60,32
)
FOR i=1 TO k:a$=a$+MID$(b$,k-i+1,1):NEXT
i
b$=a$+RIGHT$(b$,LEN(b$)-k):PRINT CHR$(7)
PRINT AT (1;17) "RESULTAT :":t=16:p$=b$:l
=9:h=17:GOSUB aff
IF b$<>c$ GOTO début
PRINT AT (1;19) "Trouvé en " s-1 " essais
"
PRINT AT (1;20) "Autre partie ( O/N )"
r$=INPUT$(1):r$=UPPER$(r$):IF r$="O" THEN
RUN ELSE END
LABEL aff
FOR i=1 TO LEN(p$)
PRINT AT (2*i+1;h) COLOR(i+3) EFFECTS(1)
POINTS(t) MID$(p$,i,1)
PRINT AT (2*i+1;h+1) POINTS(8) EFFECTS(1)
MID$(STR$(i),2)
NEXT i
RETURN
```

A travers

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : moyenne.

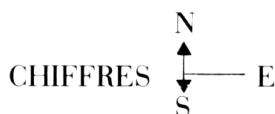
LE JEU :

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 GW-BASIC :

Les positions des chiffres et des lettres sont enregistrées dans une table N (5, 5) ; les lignes 50 à 90 définissent les valeurs de départ.

Les lignes 110 à 150 dessinent le terrain de jeu et les lignes 160 et 170 affichent le joueur qui a le trait.

Le changement de joueur est assuré par la ligne 330.

La boucle I (190-210) contrôle les réponses, et les boucles I et J (230 à 270) contrôlent le coup.

POUR JOUER :

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

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

On pourra ainsi faire 2E suivi de RETURN.

Si le coup est 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 PC.

VERSION BASIC 2 AMSTRAD :

La souris simplifie grandement les manipulations puisqu'il suffit, pour chaque joueur, de cliquer avec le bouton de gauche sur la case contenant le symbole (lettre ou chiffre) à déplacer, puis, avec le bouton de droite, on clique sur la case où il faut le déposer. Le programme contrôle automatiquement la validité des coups effectués. Un double clic permet le changement de joueur, si aucun coup n'est possible (sinon le changement est automatique après chaque coup). On devra taper CTRL et C pour arrêter le jeu.

```
10 REM A TRAVERS
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF
40 DIM N(5,5):K=INT(RND*2)
50 FOR I=1 TO 4
60 N(I,1)=ASC("0")+I
70 N(4,I)=ASC("?")+I
80 NEXT I
90 N(4,1)=0
100 CLS
110 FOR U=1 TO 4
120 FOR V=1 TO 4
130 LOCATE 4*V-1,4*U+1:IF N(V,U)=0 THEN
COLOR 5:PRINT ". " ELSE COLOR=3*(ASC(CHR$(
N(V,U)))>62)+1:PRINT CHR$(N(V,U))
140 NEXT V
150 NEXT U
160 COLOR 0:LOCATE 19,1:PRINT"JOUEUR AYA
NT LES ";
170 IF K=0 THEN COLOR 4:PRINT "LETTRES "
:PLAY"CDE" ELSE COLOR 1:PRINT"CHIFFRES":
PLAY"GEB"
180 COLOR 0:LOCATE 22,1:INPUT"VOTRE COUP
( SUIVI DE N,S,E,O ) :";R$
190 FOR I=1 TO LEN(R$)
200 IF MID$(R$,I,1)>="a" THEN MID$(R$,I,
1)=CHR$(ASC(MID$(R$,I,1))-32)
210 NEXT I
220 IF (K=1 AND RIGHT$(R$,1)="O") OR (K=
0 AND RIGHT$(R$,1)="S") THEN 160
230 FOR I=1 TO 4
240 FOR J=1 TO 4
250 IF N(I,J)=ASC(LEFT$(R$,1)) THEN 290
260 NEXT J
270 NEXT I
280 GOTO 160
290 A=I:B=J
300 I=I-(RIGHT$(R$,1)="S")+(RIGHT$(R$,1)
="N")
310 J=J-(RIGHT$(R$,1)="E")+(RIGHT$(R$,1)
="O")
320 IF (I=0 AND K=0) OR (J>4 AND K=1) TH
EN 350
330 IF N(I,J)<>0 THEN 160
340 N(I,J)=ASC(LEFT$(R$,1))
350 N(A,B)=0
360 K=ABS(K-1)
370 GOTO 100
```

REM A travers

```
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "A TRAVERS"
WINDOW OPEN
RANDOMIZE TIME:k=RND(2)-1:DIM n(5,5)
a$(1)="JEU DES CHIFFRES":a$(0)="JEU DES L
ETTES"
FOR i=1 TO 4:n(1,i)=ASC("0")+i-1:n(1,1)=A
SC("?")+i:NEXT i:n(1,1)=0
LABEL dessin
BOX 0;52,216,148 COLOR(8) FILL WITH 8
FOR u=1 TO 4
FOR v=1 TO 4
c=(n(u,v)=0)-2*(n(u,v)<ASC("?"))-4*(n(u,v)
)>ASC("?"))
BOX 54*u-54;200-37*v,54,37 WIDTH 6
IF n(u,v)=0 THEN s=46 ELSE s=n(u,v)
PRINT AT (7*v-5;4*u-1) COLOR(c) POINTS(20
) MODE(2) EFFECTS(1) CHR$(s)
NEXT v,u
LABEL joueur
PRINT AT (40;10) STRING$(40,32) AT (40;10
) COLOR(4-2*k) a$(k) " "
f=0:g=0:PRINT AT (1;21) "Clic à gauche po
ur choisir, à droite pour poser"
LABEL clic
REPEAT:a=BUTTON(1):b=BUTTON(2):UNTIL a<>-
1 OR b<>-1
x=XMOUSE:y=YMOUSE:uh=0:vh=0:IF a+b=2 THEN
k=ABS(1-k):GOTO joueur
FOR u=1 TO 5:vh=-u*(x>54*u-54 AND x<54*u)
:IF vh<>0 GOTO essai
NEXT u:GOTO clic
LABEL essai
FOR v=1 TO 5:uh=-v*(y<217-34*v AND y>180-
34*v):IF uh<>0 GOTO suite
NEXT v:GOTO clic
LABEL suite
t=(f=0 OR b=-1) AND ((n(uh,vh)>ASC("?") A
ND k=0) OR (n(uh,vh)<ASC("?") AND k=1)) A
ND n(uh,vh)<>0
IF t THEN IF f=0 THEN c=1:f=uh:g=vh:GOSUB
aff ELSE c=4-2*k:GOSUB aff:f=uh:g=vh:c=1
:GOSUB aff
s=(b<>-1 AND f<>0 AND n(uh,vh)=0)
s=s AND (((uh=f-1 OR uh=f+1) AND vh=g)
OR (vh=g+1 AND uh=f)) AND k=0 OR ((uh=f
+1 AND vh=g) OR (vh=g+1 OR vh=g-1) AND u
h=f) AND k=1)))
IF s=0 OR (uh=5 AND k=0) OR (vh=5 AND k=1
) GOTO clic
n(uh,vh)=n(f,g):n(f,g)=0:k=ABS(k-1)
BOX 54*g-50;204-37*f,46,29 COLOR(8) FILL
WITH 8
PRINT AT (7*g-5;4*f-1) COLOR(1) POINTS(20
) MODE(2) EFFECTS(1) CHR$(46)
IF uh>4 OR vh>4 THEN n(uh,vh)=0:GOTO jou
eur
BOX 54*vh-50;204-37*uh,46,29 COLOR(8) FIL
L WITH 8
PRINT AT (7*vh-5;4*uh-1) COLOR(2*k+2) POI
NTS(20) MODE(2) EFFECTS(1) CHR$(n(uh,vh))
GOTO joueur
LABEL aff
PRINT AT (7*g-5;4*f-1) COLOR(c) POINTS(20
) MODE(2) EFFECTS(1) CHR$(n(f,g))
RETURN
```

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 GW-BASIC :

Une variable A\$, en ligne 40 contient les positions de départ.

La boucle I (80-140) dessine le damier, et les lignes 160 à 180 placent le loup et les agneaux.

La ligne 190 gère le changement de joueur.

La ligne 210 affiche JEU DU LOUP si c'est le joueur qui manipule le loup qui a le trait, et la ligne 220 JEU DES AGNEAUX dans le cas contraire.

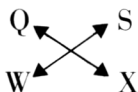
Les lignes 240 à 270 ou 290 à 310 analysent la réponse, en fonction du joueur concerné.

POUR JOUER :

RUN déclenche le tirage au sort du joueur qui commence. Après quoi, le damier s'affiche et la phrase JEU DU LOUP (A, S, Z, X) pour le loup et JEU DES AGNEAUX (ND et NG) pour les agneaux.

Les agneaux répondent par un numéro, suivi de D ou G (pour droite ou gauche).

Ainsi 2D provoque le déplacement de l'agneau numéro 2 qui monte en diagonale d'une case sur la droite. Le déplacement effectué est affiché, puis on obtient une question identique pour le loup qui répondra en tapant 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 augmentera 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 PC 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 PC).

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM LE LOUP ET LES AGNEAUX
20 SCREEN 0,1:WIDTH 40:COLOR 5,7:KEY OFF
:CLS
30 B$=CHR$(219):C$=CHR$(220):D$=CHR$(223)
)
40 A$="8183858714":C%=C%+C%+C%+C%:D%=D%+
D%+D%+D%
50 RANDOMIZE TIMER:S=INT(RND*2+1):CLS
60 PRINT "      A  B  C  D  E  F  G
H"
70 FOR I=1 TO 8:LOCATE 2*I+1,1:PRINT I:N
EXT I
80 FOR I=1 TO 8
90 FOR J=1 TO 8
100 COLOR 0:LOCATE 2*I,4*J:IF (I+J) MOD
2=1 THEN PRINT C$ ELSE IF I>1 THEN PRINT
D$
110 LOCATE 2*I+1,4*J:IF (I+J) MOD 2=1 TH
EN PRINT B$+B$+B$+B$ ELSE PRINT "
120 LOCATE 2*I+2,4*J:IF (I+J) MOD 2=1 TH
EN PRINT D$ ELSE IF I<8 THEN PRINT C$
130 NEXT J
140 NEXT I
150 L=2*VAL(MID$(A$,9,1))+1:K=4*VAL(MID$
(A$,10,1))+2:COLOR 4,0:LOCATE L,K:PRINT
"L"
160 FOR I=1 TO 4
170 L=2*VAL(MID$(A$,2*I-1,1))+1:K=4*VAL(
MID$(A$,2*I,1))+2:COLOR 1:LOCATE L,K:PRI
NT CHR$(48+I)
180 NEXT I
190 S=ABS(S-1):PLAY "CGE"
200 COLOR 0,7:LOCATE 21,1:PRINT"
"
210 LOCATE 21,1:IF S=0 THEN INPUT"JEU DU
LOUP (Q,S,W,X):":R$
220 IF S=1 THEN INPUT"JEU DES AGNEAUX (N
D OU NG):":R$:GOTO 280
230 IF R$>="a" THEN R$=CHR$(ASC(R$)-32)
240 C=10:LOCATE 2*VAL(MID$(A$,9,1))+1,4*
VAL(MID$(A$,10,1))+2:PRINT B$
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 320
280 IF RIGHT$(R$,1)>="a" THEN MID$(R$,2,
1)=CHR$(ASC(RIGHT$(R$,1))-32)

```

```

REM Le loup et les agneau:
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "LE LOUP ET LES AGNEAUX"
WINDOW OPEN
PRINT AT (1;3) "Qui commence : (L)oup ou
(A)gneau: ?"
R$=INPUT$(1):R$=UPPER$(R$):S=INSTR(1,"AL"
,R$)
IF S=0 THEN RUN ELSE S=S-1:CLS:A$="818385
8714"
FOR I=1 TO 8
PRINT AT (1;2*I+1) CHR$(48+I) AT (4*I+1;1
) CHR$(64+I)
FOR J=1 TO 8
GOSUB rectangle
NEXT J,I
LABEL dessin
F=4*VAL(MID$(A$,10,1))+1:G=2*VAL(MID$(A$,
9,1))+1:PRINT AT (F;G) "L"
F=4*VAL(MID$(A$,2,1))+1:G=2*VAL(MID$(A$,1
,1))+1:PRINT AT (F;G) "1"
F=4*VAL(MID$(A$,4,1))+1:G=2*VAL(MID$(A$,3
,1))+1:PRINT AT (F;G) "2"
F=4*VAL(MID$(A$,6,1))+1:G=2*VAL(MID$(A$,5
,1))+1:PRINT AT (F;G) "3"
F=4*VAL(MID$(A$,8,1))+1:G=2*VAL(MID$(A$,7
,1))+1:PRINT AT (F;G) "4"
S=ABS(S-1)
LABEL entrée
PRINT AT (1;19) STRING$(40,32)
IF S=1 THEN INPUT AT (1;19) "Jeu des agne
aux: ( ND ou NG ) : ",R$:R$=UPPER$(R$)
IF S=1 AND INSTR(1,"1D1G2D2G3D3G4D4G",R$)
=0 GOTO entrée ELSE IF S=1 GOTO agneau:
INPUT AT (1;19) "Jeu du loup ( Q,S,W,X )
: ",R$:R$=UPPER$(R$)
IF INSTR(1,"QSWX",R$)=0 GOTO entrée
C=10:A=VAL(RIGHT$(A$,2))+(C+1 AND R$="X")
+(C-1 AND R$="W")
A=A-(C+1 AND R$="Q")-(C-1 AND R$="S")
GOTO calcul
LABEL agneau:
C=2*VAL(MID$(R$,1,1))
A=VAL(MID$(A$,C-1,2))-(9 AND RIGHT$(R$,1)

```

```

290 C=2*VAL(MID$(R$,1,1)):L$=MID$(A$,C-1,1):K$=MID$(A$,C,1):A=VAL(L$+K$)
300 LOCATE 2*VAL(L$)+1,4*VAL(K$)+2:PRINT B$
310 A=A-(9 AND RIGHT$(R$,1)="D")-(11 AND RIGHT$(R$,1)="G")
320 A$=LEFT$(A$,C-2)+RIGHT$(STR$(A),2)+RIGHT$(A$,LEN(A$)-C)
330 GOTO 150

="D")-(11 AND RIGHT$(R$,1)="G")
LABEL calcul
b$=MID$(a$,c-1,2):j=VAL(LEFT$(b$,1)):i=VAL(RIGHT$(b$,1)):d$=STR$(a)
d$=RIGHT$(d$,2):MID$(a$,c-1,2)=d$:u$=LEFT$(d$,1):v$=RIGHT$(d$,1)
IF u$<"1" OR v$<"1" OR u$>"8" OR v$>"8" THEN MID$(a$,c-1,2)=b$:GOTO entrée
GOSUB rectangle:GOTO dessin

LABEL rectangle
BOX 32*i-14;188-18*j,32,18 COLOR(4*((i+j)MOD 2)+3) FILL WITH 8
RETURN

```


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 GW-BASIC :

La ligne 50 dimensionne une variable A (I) qui contiendra le nombre d'allumettes de chaque tas.

La boucle I (60-90) permet d'entrer ce nombre.

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

La ligne 170 assure l'affichage du joueur.

La ligne 190, ainsi que les lignes 200 et 210, contrôlent si le coup est autorisé.

La ligne 110 contrôle la fin de la partie (en ligne 280).

POUR JOUER :

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

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

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

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

EXTENSIONS POSSIBLES :

On peut désigner chaque joueur par son nom, faire un graphisme plus élaboré, écrire un programme permettant au PC de servir de partenaire, etc.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM LE JEU DE MARIENBAD
20 SCREEN 0,1:WIDTH 40:COLOR 7,0:KEY OFF
:CLS
30 INPUT "NOMBRE DE TAS ( <=7 ) :";N
40 IF N<>INT(N) OR N<2 OR N>8 THEN 20
50 S=0:DIM A(N)
60 FOR I=1 TO N
70 PRINT:PRINT "TAS ";I;:INPUT "NOMBRE D
'ALLUMETTES :";A(I)
80 S=S+A(I)
90 NEXT I
100 H=1
110 WHILE S<>0
120 CLS
130 FOR I=1 TO N
140 LOCATE 5,5*I+3:COLOR 4:PRINT A(I)
150 LOCATE 7,1:COLOR 7:PRINT "TAS":LOCAT
E 7,5*I+3:COLOR 1:PRINT I
160 NEXT I
170 LOCATE 11,1:COLOR 7:PRINT "JOUER ";
:COLOR H+5:PRINT H
180 SOUND INT(RND*4000+500),2
190 COLOR 7:PRINT:INPUT "DANS QUEL TAS :
";T
200 IF T>N OR T<>INT(T) OR T<1 THEN 120
210 IF A(T)=0 THEN 120
220 LOCATE 15,1:INPUT "QUANTITE A ENLEVE
R :";O
230 IF O>A(T) OR O=0 THEN 210
240 A(T)=A(T)-O
250 S=S-O
260 H=H+1:IF H=3 THEN H=1
270 WEND
280 PRINT:PRINT"LE JOUEUR ";:COLOR H+5:P
RINT H;:COLOR 7:PRINT "A GAGNE"
290 PLAY "CEG":END

```

```

REM Le jeu de Marienbad

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "LE JEU DE MARIENBAD"
WINDOW OPEN
REPEAT
PRINT AT (1;3) "Nombre de tas ( <=7 ) :
"
INPUT AT (25;3) "",n$:n=VAL(n$)
UNTIL n=INT(n) AND n<>0 AND n<8
FOR i=1 TO n
PRINT AT (1;2*i+3) EFFECTS(1) COLOR(i) "T
as " i "Nombre d'allumettes : ";
INPUT "",n$:a(i)=VAL(n$):s=s+a(i)
NEXT i
h=1
REPEAT
CLS
FOR i=1 TO n
PRINT AT (4*i+3;4) COLOR(i) POINTS(20) CH
R$(48+a(i))
PRINT AT (1;6) "Tas " AT (4*i+3;6) EFFECT
S(1) CHR$(48+i)
NEXT i
PRINT AT (1;10) "Joueur " COLOR(2*h) h:PR
INT CHR$(7)
REPEAT
PRINT AT (1;12) STRING$(20,32)
INPUT AT (1;12) "Dans quel tas : ",t$:t=V
AL(t$)
UNTIL t<=n AND t=INT(t) AND t<>0 AND a(t)
<>0
REPEAT
PRINT AT (1;14) STRING$(30,32)
INPUT AT (1;14) "Quantité à enlever : ",o
$:o=VAL(o$)
UNTIL o<=a(t) AND o<>0
a(t)=a(t)-o:s=s-o:h=h+1
IF h=3 THEN h=1
UNTIL s=0
PRINT AT (1;17) POINTS(16) "Le joueur " C
OLOR(2*h) h COLOR(1) " a gagné"
r$=INPUT$(1)
END

```

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 PC 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 GW-BASIC :

La ligne 40 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 60.

La boucle I (80-120) imprime des tirets aux emplacements des lettres à trouver.

La boucle en ligne 160 sert uniquement à simuler le temps de recherche d'une lettre.

La formule, placée en ligne 170, permet de modifier légèrement la probabilité de tirage, en faveur des voyelles et des consonnes usuelles.

La ligne 190 provoque un nouveau tirage, et la boucle I (230-260) réimprime les tirets et les lettres trouvées.

La ligne 270 incrémente le compteur d'essais.

La ligne 290 affiche le nombre d'essais réalisés, en fin de partie.

POUR JOUER :

Lancer le programme par RUN.

Entrer le mot à découvrir, suivi de RETURN.

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

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

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```
10 REM PENDU INVERSE
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF:CLS
40 D$="YAEIOUBCDFGLMNPSTVWHKJQXZ"
50 C=1:T$="":U=1:S=18
60 INPUT"ENTREZ VOTRE MOT :";R$
70 CLS
80 FOR I=1 TO LEN(R$)
90 IF MID$(R$,I,1)>="a" THEN MID$(R$,I,1)
   =CHR$(ASC(MID$(R$,I,1))-32)
100 LOCATE 6,I+1:PRINT "-"
110 T$=T$+"-"
120 NEXT I
130 LOCATE 16,1:PRINT "Lettres essayées
   sans succès"
140 WHILE T$<>R$
150 COLOR 1:LOCATE 10,1:PRINT"JE CHERCHE
   .... ESSAI ";C
160 FOR J=1 TO 1000:NEXT J
170 X=-6*(RND<.7)-13*(RND<.5)-7*(RND<.4)
180 X=INT(X#RND+1)
190 IF MID$(D$,X,1)="-" THEN 170
200 E$=MID$(D$,X,1)
210 MID$(D$,X,1)="-":K=0
220 COLOR 0:PRINT:PRINT"JE PRENDS
   : ";:COLOR 4:PRINT E$:PLAY"QSCDE"
230 FOR I=1 TO LEN(R$)
240 IF E$=MID$(R$,I,1) THEN K=1:MID$(T$,
   I,1)=E$:PLAY"DDG"
250 COLOR 0:LOCATE 6,I+1:PRINT MID$(T$,I
   ,1):COLOR 5
260 NEXT I
270 C=C+1:IF K=0 THEN LOCATE S,U:PRINT E
   $:U=U+2:IF U>=38 THEN U=1:S=S+1
280 WEND
290 COLOR 0:LOCATE 20,1:PRINT"TROUVE EN
   ";C-1;" COUPS":PLAY"CDEFG":END
```

```
REM Le pendu inversé

STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "LE PENDU INVERSE"
WINDOW OPEN
RANDOMIZE TIME
LABEL entree
CLS:r$="":t$="":p$="":u=1
REPEAT
r$=r$+s$
PRINT AT (1;3) "Votre mot : " COLOR(4) t$
;
s$=INPUT$(1):t$=t$+"#":IF s$="0" GOTO ent
ree
UNTIL s$=CHR$(13)
r$=UPPER$(r$)
FOR i=1 TO LEN(r$)
p$=p$+"-":PRINT AT (2*i;6) POINTS(20) COL
OR(2) "-"
NEXT i
c=1:d$="YAEIOUBCDFGLMNPSTVWHKJQXZ"
PRINT AT (1;15) "Lettres essayées sans su
ccès"
REPEAT
PRINT AT (1;9) EFFECTS(1) "Je cherche....
- Essai numéro " c
PRINT CHR$(7);:PRINT AT (11;12) POINTS(20
) " "
FOR j=1 TO RND(7500)+2000:NEXT j
REPEAT
x=(6 AND RND<.8)+(13 AND RND<.5)+(7 AND R
ND<.4):x=INT(RND*x+1)
UNTIL MID$(d$,x,1)<>"-"
e$=MID$(d$,x,1):MID$(d$,x,1)="-":k=0
PRINT AT (1;12) "Je prends " COLOR(14) PO
INTS(20) e$
FOR i=1 TO LEN(r$)
IF e$=MID$(r$,i,1) THEN k=1:MID$(p$,i,1)=
e$
PRINT AT (2*i;6) POINTS(20) COLOR(4) MID$
(p$,i,1)
NEXT i
c=c+1:IF k=0 THEN PRINT AT (u;17) COLOR(1
5) e$:u=u+1
UNTIL p$=r$
PRINT AT (1;19) "Trouvé en " c-1 " essais
"
r$=INPUT$(1)
END
```

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.

Le générateur aléatoire de l'appareil est initialisé à partir du code introduit par le joueur ; on devra bien veiller à utiliser toujours le même nombre (noyau) pour commencer le codage ou le décodage de deux messages identiques.

LE PROGRAMME GW-BASIC :

Le noyau initial (qui sera un entier quelconque) est entré grâce à une boucle WHILE-WEND (lignes 50 à 70) ; le message est entré à la ligne 90.

La ligne 100 permet de choisir entre les options de codage et de décodage.

La boucle I (120-200) est utilisée soit pour effectuer le codage, soit pour effectuer le décodage.

La ligne 140 appelle le générateur aléatoire.

La ligne 150 réalise la transposition des lettres.

Les lignes 160 et 170 vérifient que l'on reste bien dans les limites autorisées.

Les résultats seront imprimés par la ligne 210.

POUR JOUER :

Lancer le programme par RUN.

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

Puis taper le message, suivi de RETURN.

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

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.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM ESPION
20 SCREEN 0,1:WIDTH 40:COLOR 7,0:KEY OFF
:CLS
30 PRINT"DONNEZ VOTRE CODE SECRET : ";
40 S$="":R$=""
50 WHILE R$<>CHR$(13)
60 S$=S$+R$:R$=INPUT$(1):IF R$<>CHR$(13)
THEN PRINT "X";
70 WEND
80 RANDOMIZE VAL(S$)
90 PRINT:LINE INPUT"VOTRE MESSAGE :";A$
100 PRINT:PRINT"VOULEZ-VOUS (C)ODER OU (
D)ECODER :";:R$=INPUT$(1)
110 B$=""
120 FOR I=1 TO LEN(A$)
130 S=ASC(MID$(A$,I,1))
140 A=RND(26)
150 T=S-A*((R$="C" OR R$="c")-(R$="D" OR
R$="d"))
160 IF T>90 THEN T=T-26
170 IF T<65 THEN T=T+26
180 IF T>=65 AND T<=90 AND S>=65 THEN S$
=CHR$(T) ELSE S$=" "
190 B$=B$+S$
200 NEXT I
210 PRINT:PRINT"LE RESULTAT EST : ";B$

```

```

REM Espion
STREAM #1:CLS
WINDOW FULL ON
WINDOW TITLE "ESPION"
WINDOW OPEN
PRINT AT (1;4) EFFECTS(1) "Donnez votre c
ode secret : ";
s$="":r$=""
WHILE r$<>CHR$(13)
s$=s$+r$
r$=INPUT$(1):IF r$<>CHR$(13) THEN PRINT P
OINTS(16) COLOR(2) "x";
WEND
PRINT AT (1;6) EFFECTS(1) "Quel est votre
message : ";
LINE INPUT " ",a$:a$=UPPER$(a$)
PRINT:PRINT EFFECTS(1) "Voulez-vous (C)od
er ou (D)écoder : ";
r$=INPUT$(1):r$=UPPER$(r$)
b$="":RANDOMIZE (VAL(s$))
FOR i=1 TO LEN(a$)
s=ASC(MID$(a$,i,1))
a=RND(26)
t=s-a*((r$="C")-(r$="D"))
IF t>90 THEN t=t-26
IF t<65 THEN t=t+26
IF t>=65 AND t<=90 AND s>=65 THEN s$=CHR$
(t) ELSE s$=" "
b$=b$+s$
NEXT i
PRINT:PRINT:PRINT "Le résultat est : " CO
LOR(2) b$
r$=INPUT$(1)
END

```

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 PC calcule et affiche cet entier. Le procédé retenu peut d'ailleurs être généralisé (à vous de trouver comment).

LE PROGRAMME GW-BASIC :

Les lignes 30 à 50 affichent quelques explications en appelant le sous-programme 240-270.

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 DES DIVISIONS : 3, 5, 7" s'affiche.

On doit répondre 0, 1 ou 2 pour 3, puis de 0 à 4 pour 5 et de 0 à 6 pour 7.

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.

VERSION BASIC 2 AMSTRAD :

Les restes possibles étant affichés au fur et à mesure sur l'écran pour chaque division, on utilise la souris pour désigner, en appuyant sur le bouton de gauche, le reste choisi. La solution est affichée dès que l'on a choisi les trois restes. On répond « O » à la question « Voulez-vous faire un autre calcul » si l'on veut recommencer.

```

10 REM DECOMAQUE
20 SCREEN 0,1:WIDTH 40:COLOR 7,0:KEY OFF
:CLS
30 A$="VOUS DEVEZ ENTRER LES TROIS RESTE
S":L=2:GOSUB 240
40 A$="DANS LES DIVISIONS PAR 3 , 5 ET 7
":L=4:GOSUB 240
50 A$="D'UN NOMBRE ENTIER INFERIEUR A 10
0":L=6:GOSUB 240
60 I=3:S=1
70 COLOR 7:LOCATE 11,1:PRINT"RESTE DES D
IVISIONS PAR : 3 5 7"
80 PLAY"CDE":LOCATE 12,2*I+21:COLOR S:PR
INT CHR$(220)
90 A$=INPUT$(1):IF A$<"0" OR A$>CHR$(47+
I) THEN 90
100 A(S)=ASC(A$)-48
110 S=S+1:I=I+2
120 IF I<8 THEN 80 ELSE VIEW PRINT 14 TO
24:COLOR ,4:CLS
130 COLOR 0:LOCATE 15,1:PRINT"RESTES :
PAR 3 PAR 5 PAR 7"
140 FOR I=1 TO 3
150 FOR J=1 TO A(I)
160 IF A(I)<>0 THEN COLOR I:LOCATE 17,12
*I-2*J+5:PRINT CHR$(178)
170 NEXT J
180 NEXT I
190 T=A(1)*70+A(2)*21+A(3)*15
200 IF T>105 THEN T=T-105:GOTO 200
210 LOCATE 21,1:PRINT"LA SOLUTION EST ";
T:VIEW PRINT 1 TO 24
220 LOCATE 23,1:PRINT"VOULEZ-VOUS UN AUT
RE ESSAI (O/N)"
230 A$=INPUT$(1):IF A$="0" OR A$="o" THE
N RUN ELSE END
240 FOR I=1 TO LEN(A$)
250 COLOR INT(RND*15+1):LOCATE L,I:PRINT
MID$(A$,I,1):SOUND INT(RND*5000+400),1
260 NEXT I
270 RETURN

```

```

REM Decomaque

STREAM #1:SCREEN GRAPHICS 640 FIXED, 200
FIXED
USER SPACE 640, 200
WINDOW FULL ON
WINDOW TITLE "DECOMAQUE"
WINDOW OPEN
RANDOMIZE TIME:k=0:g=2
FOR j=1 TO 10:READ a$:GOSUB 1ec:NEXT j:r$
=INPUT$(1):CLS
LABEL debut
FOR i=1 TO 3
PRINT AT (1;5) COLOR(2*i+8) EFFECTS(1) PO
INTS(16) "Reste dans la division par " 2*
i+1
FOR j=0 TO 2*i
BOX 64*j+10;110,32,36 COLOR(2*i+8) WIDTH(
5) ROUNDED
PRINT AT (8*j+3;9) MODE(1) POINTS(20) CHR
$(48+j)
NEXT j
LABEL choix
bo=BUTTON(1):x=XMOUSE-12:y=YMOUSE:x=INT(x
/32):r=x/2
IF y>100 AND y<127 AND r=INT(r) AND r<2*i
+1 THEN PRINT AT (8*r+3;9) MODE(3) POINTS
(20) CHR$(48+r) AT (8*r+3;9) CHR$(48+r):I
F bo<>-1 GOTO suite
GOTO choix
LABEL suite
PRINT AT (60;5+2*i) COLOR(2*i+8) POINTS(1
6) "Par " 2*i+1 " : " r:a(i)=r
NEXT i
g=a(1)*70+a(2)*21+a(3)*15
WHILE g>105:g=g-105:WEND
PRINT AT (1;15) "La solution est " POINTS
(20) STR$(g)
PRINT AT (1;17) "Voulez-vous faire un aut
re calcul ( O/N ) ?"
r$=UPPER$(INPUT$(1)):IF r$="0" THEN CLS:G
OTO debut ELSE END

LABEL 1ec
a$a$+STRING$(40,32)
FOR i=1 TO 38
r$=MID$(a$,i,1):co=(i MOD 15)+1:k=k+2
PRINT AT (k;g) COLOR(co) POINTS(20) EFFEC
TS(1) MODE(4) r$:PRINT CHR$(7)
NEXT i:g=g+2:k=k=0
RETURN

DATA "",VOUS DEVEZ ENTRER AVEC LA SOURIS,
""
DATA LES TROIS RESTES DANS LES DIVISIONS,
""
DATA "PAR TROIS, PAR CINQ, ET PAR SEPT","
"
DATA D'UN NOMBRE ENTIER INFERIEUR A 100,"
"
DATA APPUYEZ SUR UNE TOUCHE POUR COMMENCE
R

```


Horloge

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : dessin.

Difficulté du jeu : aucune.

LE JEU :

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

LE PROGRAMME GW-BASIC :

La ligne 40 vous permettra d'entrer l'heure sous forme d'un entier de 4 chiffres (par exemple : 1024 signifiera 10 heures 24 minutes).

Les lignes 70-190 réalisent le dessin, en décalant l'impression au fur et à mesure (à l'aide de la variable K, incrémentée en ligne 180).

Les lignes 200-210 assurent la progression de l'heure.

La ligne 220 prépare l'affichage suivant en utilisant la variable TIME\$.

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'AMSTRAD disposant à l'origine d'une horloge alimentée par piles, il est proposé en début de programme un choix entre l'heure donnée par l'ordinateur, et une autre heure qui devra alors être entrée au clavier, sous forme de quatre chiffres consécutifs (on tape « 0 » pour garder l'heure AMSTRAD, et « 1 » pour donner une autre heure). Le programme se déroule ensuite automatiquement (on peut arrêter en tapant CTRL et C).

```

10 REM HORLOGE
20 SCREEN 0,1:WIDTH 40:COLOR 0,7:CLS
30 B$=CHR$(219):B$(1)=CHR$(178):B$(0)="
"
40 INPUT"HEURE (FORME XXXX )";A$
50 TIME$=LEFT$(A$,2)+":"+RIGHT$(A$,2)
60 CLS:COLOR 4
70 K=1
80 A=VAL(MID$(A$,K,1))
90 IF A <>1 AND A<>4 THEN LOCATE 7,6*K:P
RINT B$+B$+B$
100 FOR J=1 TO 5
110 IF A=0 OR (A>3 AND A<>7) THEN LOCATE
J+6,6*K:PRINT B$
120 IF A>5 AND A<>6 THEN LOCATE J+6,6*K
+2:PRINT B$
130 IF A=0 OR A=2 OR A=6 OR A=8 THEN LOC
ATE J+10,6*K:PRINT B$
140 IF A<>2 THEN LOCATE J+10,6*K+2:PRINT
B$
150 NEXT J
160 IF A>1 THEN LOCATE 11,6*K+1:PRINT B$
+B$
170 IF A<>1 AND A<>4 AND A<>7 THEN LOCAT
E 15,6*K:PRINT B$+B$+B$
180 K=K+1
190 IF K<5 THEN 80
200 F$=RIGHT$(TIME$,2):LOCATE 11,16:COLO
R 1:PRINT B$(L)
210 IF RIGHT$(TIME$,2)=F$ AND F$<>"59" T
HEN 200 ELSE IF F$<>"59" THEN L=ABS(L-1)
:GOTO 200
220 PLAY "CGE":A$=LEFT$(TIME$,2)+MID$(TI
ME$,4,2):GOTO 60

```

```

REM Horloge

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "HORLOGE"
WINDOW OPEN
LABEL heure
a$=CHR$(5):b$="":c$=CHR$(27)+a$
PRINT AT (1;3) "L'heure actuelle est " M
ID$(TIME$,1,8)
PRINT AT (1;5) "Voulez-vous utiliser cell
e-ci (0) ou la votre (1) ?"
r$=UPPER$(INKEY$):IF r$<>"0" AND r$<>"1"
GOTO heure
a$=LEFT$(TIME$,2)+MID$(TIME$,4,2)
IF r$="1" THEN GOSUB entree
SET COLOR(6) POINTS(20)
LABEL debut
CLS:BOX 0;0,640,200 COLOR(4) FILL WITH 8
k=1
LABEL affiche
a=VAL(MID$(a$,k,1)):b=(TIME MOD 360000) \
6000
IF a<>1 AND a<>4 THEN PRINT AT (16*k-4;6)
MODE(4) b$
FOR j=1 TO 5
IF a=0 OR (a>3 AND a<>7) THEN PRINT AT (1
6*k-4;j+5) c$
IF a<>5 AND a<>6 THEN PRINT AT (16*k+2;j+
5) c$
IF a=0 OR a=2 OR a=6 OR a=8 THEN PRINT AT
(16*k-4;j+9) c$
IF a<>2 THEN PRINT AT (16*k+2;j+9) c$
NEXT j
IF a>1 THEN PRINT AT (16*k-4;10) MODE(4)
b$
IF a<>1 AND a<>4 AND a<>7 THEN PRINT AT (
16*k-4;14) MODE(4) b$
k=k+1:IF k<5 GOTO affiche
IF r$="0" THEN a$=LEFT$(TIME$,2)+MID$(TIM
E$,4,2):GOTO battement
a=1+VAL(a$):IF a=2360 THEN a=0
a$="000"+RIGHT$(STR$(a),LEN(STR$(a))-1)
a$=MID$(a$,LEN(a$)-3,4)
IF RIGHT$(a$,2)="60" THEN a$=RIGHT$(STR$(
VAL(LEFT$(a$,2))+1),2)+"00"
LABEL battement
REPEAT
FOR i=1 TO 10
LOCATE 39;10:IF h=1 THEN PRINT MODE(4) ":
" ELSE PRINT COLOR(4) MODE(4) " "
NEXT i
h=ABS(h-1)
UNTIL b<>(TIME MOD 360000) \ 6000
PRINT CHR$(7):GOTO debut

LABEL entree
PRINT AT (1;7) STRING$(30,32)
INPUT AT (1;7) "Heure ( format xxxx ) : "
,a$

IF LEN(a$)<>4 OR VAL(LEFT$(a$,2))>24 OR V
AL(RIGHT$(a$,2))>59 GOTO entree
RETURN

```

Niveau 4

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Anomalie	Exercez votre perspicacité dans la reconnaissance des symboles.
Gouttes	Un véritable gag visuel. A exploiter.
Parcours du chevalier	Très spectaculaire et instructif à plus d'un titre.
Adresse	Jeu de réflexe pour 1 joueur, très visuel.
La bombe	Saurez-vous la découvrir à temps ? (1 joueur).
Calculatrice	Un programme de simulation très réaliste.
Anagrammes	Un bon jeu de réflexe ; pour deux joueurs.
Le jeu du 15	Un jeu de réflexion utilisant des chiffres.
Clavier	Apprenez à taper à la machine en vous amusant.
Quatre à quatre	Un jeu de réflexion un peu difficile, pour deux joueurs.
Tic Tac Toc	Un classique, mais pas sans intérêt, pour deux joueurs.
Smokey	Fumez sans risque, avec ce programme original pour un joueur.
Danger spatial	Serez-vous assez adroit pour éviter toutes les météorites.
Zombies	Un peu farfelu, mais très visuel (1 joueur).
Taquin de lettres	Un proverbe est caché dedans ; à vous de le découvrir.
Othello	Pour deux joueurs ; il retourne les pions à votre place.
Mur de briques	Un grand classique dans une version très complète.
Diago	Etes-vous vraiment sûr de vos réflexes ?
Mots	Une idée très simple, mais ce n'est pas si facile de trouver.
Le singe	Un jeu de lettres pour deux joueurs, où il faut de bons réflexes.

Les techniques d'animation élémentaire

SAISIE AU VOL D'UN CARACTERE FRAPPE AU CLAVIER : INKEY \$

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 GW-Basic, cette propriété est offerte, par la **fonction INKEY\$** qui effectue une scrutation des touches du clavier et enregistre le nom de la dernière touche frappée.

On pourra utiliser INKEY\$ de deux façons :

- soit dans une boucle (réalisation d'un mouvement). On saisit INKEY\$ et on le teste pour une éventuelle sortie de la boucle.
- soit dans une attente : tant que INKEY\$ est vide (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 GW-Basic 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.

LA RECONNAISSANCE D'UN CARACTERE SUR L'ECRAN : SCREEN (X,Y)

Dans la plupart des jeux d'action, la scène se déroule sur l'écran du moniteur, que l'on peut donc considérer comme le terrain de jeu. Il est nécessaire de détecter des rencontres ou des collisions entre les divers objets dessinés sur l'écran, ou encore de contrôler l'occupation de la case dans laquelle on désire jouer. Le GW-Basic permet de réaliser cela, grâce à l'instruction SCREEN (X,Y) qui renvoie le code du caractère présent en ligne X et colonne Y de l'écran.

MINUTAGE D'UNE ACTION : TIMER

Lorsqu'il est nécessaire de faire intervenir un temps de réponse, pour un jeu, on peut utiliser un ensemble d'instructions permettant de renvoyer, lorsqu'un laps de temps prédéfini s'est écoulé, dans un programme de contrôle. On prépare l'interruption avec une instruction du type : ON TIMER (n) GOSUB xxx. n est le temps autorisé, en secondes, avant le départ dans le sous-programme xxx. Puis le décompte est déclenché avec l'instruction TIMER ON. La remise à 0 du compteur sera faite, pour une nouvelle interruption, par TIMER OFF. Après quoi, TIMER ON peut déclencher un nouveau comptage, et ainsi de suite. Le temps peut être lu directement dans la variable TIMER.

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 GW-BASIC :

La boucle I (70-100) imprime les chiffres de repérage des grilles sur les deux écrans définis par J (50-110).

La boucle principale I (140-180) imprime les symboles des deux grilles, dans un ordre différent sur la grille non visible (130).

La ligne 190 choisit le caractère à imprimer.

Le sous-programme (270-310) s'occupe de faire l'impression à un emplacement libre (on l'appelle deux fois pour les deux grilles).

Les coordonnées du point essayé sont entrées aux lignes 210 et 220, et le contrôle est effectué par les lignes 230 et 240.

Les lignes 250 et 260 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 RETURN, et son numéro de colonne, suivi de RETURN.

Le programme répondra alors BRAVO ou PERDU en donnant dans ce cas la bonne réponse.

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.

VERSION BASIC 2 AMSTRAD :

Pour choisir la lettre à découvrir, il suffit de cliquer à l'aide du bouton de gauche de la souris, sur la case contenant la lettre choisie, dans la grille de droite. La ligne et la colonne sont affichées en permanence, ainsi que le temps. Le programme répondra BRAVO si l'on a juste, et affichera la lettre qu'il fallait découvrir si l'on s'est trompé.

```

10 REM ANOMALIE
20 SCREEN 0,1:WIDTH 40:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF
40 CLS:T=5
50 FOR J=0 TO 1
60 SCREEN ,,J,0
70 FOR I=1 TO 5
80 LOCATE 1,2*I+2:COLOR 1:PRINT I:LOCATE
  2*I+1,1:PRINT I
90 LOCATE 1,2*I+20:COLOR 4:PRINT I:LOCAT
  E 2*I+1,19:PRINT I
100 NEXT I
110 NEXT J
120 SCREEN ,,0,0:COLOR 7:LOCATE 17,9:PRI
  NT "ATTENDEZ UN INSTANT"
130 SCREEN ,,1,0:COLOR 12
140 FOR I=1 TO 22
150 D=INT(26*RND+1)
160 L=2:S=T:GOSUB 270
170 L=11:GOSUB 270
180 NEXT I
190 S=0:GOSUB 270
200 SCREEN ,,1,1
210 COLOR 7:LOCATE 17,1:INPUT"LIGNE.....
  ":";I
220 PRINT:INPUT"COLONNE....":T
230 U=2*I:V=2*T+20
240 C=SCREEN(U+1,V+1)
250 IF D+64=C THEN PRINT "BRAVO...":PLAY
  "CEG" ELSE PRINT"PERDU":PLAY"CCC"
260 PRINT:PRINT"LA LETTRE A TROUVER ETAI
  T ":";CHR$(D+64):END
270 U=2*INT(RND*T+1):V=2*INT(RND*S+L)
280 C=SCREEN(U+1,V+1)
290 IF C<>32 AND S<>0 THEN 270
300 LOCATE U+1,V+1:PRINT CHR$(D+64):SOUN
  D INT(RND*4000+500),1
310 RETURN

```

```

REM Anomalie
FOR i=1 TO 4:CLOSE WINDOW i:CLOSE #i:NEXT
  i
FOR i=1 TO 4:OPEN #i WINDOW i:NEXT
WINDOW #2 SIZE 37,13:WINDOW #4 SIZE 37,13
:WINDOW #3 SIZE 37,13
WINDOW #2 PLACE 0,100:WINDOW #2 TITLE "GR
  ILLE INITIALE"
WINDOW #4 PLACE 320,100:WINDOW #4 TITLE "
  GRILLE FINALE"
WINDOW #3 PLACE 320,100:WINDOW #3 TITLE "
  GRILLE FINALE"
WINDOW #1 SIZE 640,50:WINDOW #1 PLACE 0,0
:WINDOW #1 TITLE "REPONSE"
WINDOW #2 OPEN:WINDOW #4 OPEN:WINDOW #3 O
  PEN
RANDOMIZE TIME:DIM t(6,12)
FOR i=0 TO 5
  x=3*I+5:y=1:fe=2:co=2:ch=48+i:GOSUB aff:f
  e=4:co=4:GOSUB aff:fe=3:GOSUB aff
  x=2:y=2*I+3:fe=2:co=2:ch=48+i:GOSUB aff:f
  e=4:co=4:GOSUB aff:fe=3:GOSUB aff
NEXT i
FOR i=1 TO 30
  t=6:s=6:d=RND(26):l=2:GOSUB tir:l=4:GOSUB
    tir:PRINT CHR$(7)
NEXT i
s=0:GOSUB tir:t=0:CLOSE WINDOW 3:STREAM #
  1:WINDOW #1 OPEN
PRINT AT(1;1) "Cliquez sur la lettre à dé
  couvrir dans la grille finale"
REPEAT
t=t+1:PRINT AT(1;3) "Colonne :" AT (20;3)
  "Ligne :" AT (40;3) "Lettre :" AT (60;3)
  "Temps :":t
a=BUTTON(1):x=XMOUSE+12:y=216-YMOUSE:x=(x
  \24)-15:y=(y\16)-3
IF x<0 OR x>5 THEN x=-16
IF y<0 OR y>5 THEN y=-16
IF x<>-16 AND y<>-16 THEN IF t(y,x+6)<>0
  THEN d$=CHR$(64+t(y,x+6)) ELSE d$=" "
PRINT AT(12;3) CHR$(48+x) AT (28;3) CHR$(
  48+y) AT (49;3) d$
UNTIL a<>-1 AND d$<>" " AND d$<>" "
IF d$=CHR$(64+d) THEN PRINT AT(1;5) "BRAV
  O" ELSE PRINT AT(1;5) "La lettre à décou
  vrir était : ";CHR$(64+d)
PRINT CHR$(7)CHR$(7)CHR$(7):r$=INPUT$(1):
  END

```

```

LABEL aff
PRINT #fe;AT (x;y) COLOR(co) CHR$(ch)
RETURN

```

```

LABEL tir
f=INT(RND*t):g=INT(RND*s):u=2*f+3:v=3*g+5
:e=t(f,g-6*(1=4))
IF (e<>0 AND s<>0) OR (e+s=0) THEN GOTO t
ir ELSE t(f,g-6*(1=4))=d
x=v:y=u:ch=d+64+32*(d=0):co=1:fe=1:GOSUB
  aff
RETURN

```

Gouttes

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : dessin.

Difficulté du jeu : aucune.

LE JEU :

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

LE PROGRAMME GW-BASIC :

Les deux boucles emboîtées I (50-120) et J (60-110) dessinent le récipient et les obstacles.

La ligne 150 dessine la goutte et la ligne 190 envoie dans un sous-programme (330-350) qui permet de déterminer s'il y a ou non rencontre avec un obstacle.

Si c'est non, on va en 270, et la goutte continue de tomber.

Si c'est oui, la goutte part à droite ou à gauche (ligne 210) selon la valeur définie pour K en ligne 140.

On regarde de nouveau s'il y a ou non obstacle, ou si la goutte peut continuer à tomber (lignes 220, 270 ou 240).

Selon le résultat du test, on utilisera 290, 310, ou 250.

POUR JOUER :

Le RUN déclenche le dessin du récipient et des obstacles, la goutte apparaît, se met à tomber et finit par se trouver une place. Une autre goutte apparaît et l'on recommence.

Le programme s'arrête de lui-même lorsque le récipient déborde ; on peut l'arrêter à tout moment en tapant CTRL et Break.

EXTENSIONS POSSIBLES :

On peut, bien entendu, ajouter la couleur et le son et aussi envisager un jeu dans lequel un petit personnage devrait échapper aux gouttes ou à la noyade.

VERSION BASIC 2 AMSTRAD :

Le fonctionnement du programme reste le même que dans la version GW-Basic, mais il est en plus possible de choisir, à l'aide de la souris, les couleurs du récipient, des obstacles, et de la goutte qui tombe. On devra pour cela cliquer avec la touche sur la case contenant la couleur choisie.

Le programme débute ensuite automatiquement et s'arrête lorsque le récipient déborde.

```

10 REM GOUTTES
20 SCREEN 0,1:WIDTH 40:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF:CLS
40 K=-2*(RND<.5)-1
50 FOR I=12 TO 20
60 FOR J=1 TO 16
70 LOCATE I+1,1:COLOR 4:PRINT CHR$(219):
LOCATE I+1,18:PRINT CHR$(219)
80 LOCATE 21,J+1:PRINT CHR$(219)
90 IF I<>12 AND I<>20 THEN LOCATE I+1,J+
1:COLOR 1:PRINT CHR$(178)
100 IF I<>20 AND RND<.8 THEN LOCATE I+1,
INT(RND*15+2):PRINT " "
110 NEXT J
120 NEXT I
130 Y=INT(RND*7+6):X=6:CO=INT(RND*15+1)
140 K=-2*(RND<.5)-1:I=0
150 LOCATE X+1,Y+1,0:COLOR CO:PRINT CHR$(
2)
160 FOR J=0 TO 100:NEXT J
170 LOCATE X+1,Y+1,0:PRINT " "
180 U=X+1:V=Y
190 GOSUB 330
200 IF J=0 THEN 270
210 U=X:V=Y+K
220 GOSUB 330
230 IF J=0 THEN 290
240 IF I=0 THEN 310
250 LOCATE X+1,Y+1,0:COLOR 3:PRINT CHR$(
3)
260 GOTO 130
270 X=X+1
280 GOTO 140
290 Y=Y+K:I=1
300 GOTO 150
310 K=-K:I=1
320 GOTO 210
330 C=SCREEN(U+1,V+1):SOUND INT(RND*4000
+500),1
340 J=-(C<>32):IF U=24 THEN LOCATE 20,20
:PRINT"C'EST FINI...":PLAY"CEG":END
350 RETURN

```

```

REM Gouttes
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON:WINDOW MOUSE 3
WINDOW TITLE "GOUTTES"
WINDOW OPEN
RANDOMIZE TIME:DIM t(42,20):a$=CHR$(27)+C
HR$(5)
FOR i=1 TO 15
BOX 40*i-15;150,39,20 COLOR(i+i*(k=i OR h
=i)) FILL WITH 8 ROUNDED
NEXT i
PRINT AT (1;2) "Cliquez sur la couleur du
récipient"
k=0:h=0:GOSUB saisie:cr=x
PRINT AT (1;2) "Cliquez sur la couleur de
s obstacles"
k=x:GOSUB saisie:co=x
PRINT AT (1;2) "Cliquez sur la couleur de
la bille "
h=x:GOSUB saisie:cb=x
CLS:FOR i=6 TO 17:FOR j=9 TO 40:PRINT AT
(8;i) COLOR(cr) a$
PRINT AT (41;i) COLOR(cr) a$ AT (j;17) CO
LOR(cr) a$
IF i>7 AND i<>17 AND RND<.35 THEN PRINT A
T (j;i) COLOR(co) a$:t(j,i)=1
t(j,17)=1:NEXT j
t(8,i)=1:t(41,i)=1:NEXT i
LABEL début
x=INT(RND*15+1):y=1
LABEL tome
k=2*RND(2)-3:i=0
LABEL affiche
PRINT AT (x;y) COLOR(cb) a$:CHR$(7)
FOR j=0 TO 80:IF y>17 GOTO fin
NEXT j:PRINT AT (x;y) " ":u=y+1:v=x:j=t(v
,u):IF j=0 GOTO avance
LABEL suite:u=y:v=x+k:j=t(v,u):IF j=0 GOT
O cherche
IF i=0 GOTO change
PRINT AT (x;y) COLOR(cb) a$:t(x,y)=1:GOTO
début
LABEL avance:y=y+1:GOTO tome
LABEL cherche:x=x+k:i=1:GOTO affiche
LABEL change:k=-k:i=1:GOTO suite
LABEL fin
PRINT AT (5;18) "C'est fini.....":r$=INP
UT$(1):END

LABEL saisie
REPEAT
x=XMOUSE:y=YMOUSE:a=BUTTON(1):x=((x+15)\4
0)
UNTIL a<>-1 AND y<151 AND y>133 AND x>0 A
ND x<16
BOX 40*x-15;150,39,20 COLOR(0) FILL WITH
8 ROUNDED
BOX 40*x-15;130,39,20 COLOR(x) FILL WITH
8 ROUNDED
RETURN

```

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 GW-BASIC :

La ligne 50 définit les divers déplacements possibles, dans deux variables alphanumériques A\$ et B\$; la boucle I (60-90) dessine le terrain de jeu. On entre les coordonnées de la case de départ (ligne, puis colonne) aux lignes 100 et 110.

La ligne 130 imprime le numéro du coup.

La ligne 140 permet le tirage au hasard d'un déplacement du cavalier.

La boucle Z (150-200) 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 160 et 170 effectuent le déplacement.

La ligne 180 contrôle que l'on ne sorte pas du terrain de jeu.

La ligne 190 envoie en ligne 220 si le coup est possible.

La ligne 210 gère la fin de partie (cavalier bloqué) et affiche le score.

Les lignes 220, 230 et 240 réalisent le coup et incrémentent le score.

POUR JOUER :

On lance le programme par l'ordre RUN. Le terrain de jeu s'affiche ; on choisit alors la position de départ en tapant d'abord un numéro de ligne (entre 1 et 8), suivi de RETURN puis son numéro de colonne, suivi de RETURN. Le jeu commence immédiatement et il se déroule de façon entièrement automatique jusqu'à ce que le cavalier soit bloqué. On peut, bien sûr, arrêter à tout moment par CTRL et Break.

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

Il est de nouveau fait appel à la souris de l'AMSTRAD pour choisir la ligne et la colonne de départ : il suffit après avoir lancé l'exécution du programme par RUN, de cliquer avec la touche de gauche sur la case de départ choisie. Le cavalier va ensuite parcourir l'échiquier automatiquement, jusqu'au blocage par impossibilité de jouer en respectant les règles.

```
10 REM PARCOURS DU CAVALIER
20 SCREEN 0,1:WIDTH 40:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF
40 CLS:S=1:T=0
50 A$="56653223":B$="653223"+A$+"56":A$=
A$+A$
60 FOR I=1 TO 8
70 LOCATE 1,3*I+2:PRINT I
80 LOCATE 2*I+1,1:PRINT I;" . . . .
. . . ."
90 NEXT I
100 LOCATE 19,1:INPUT"LIGNE DE DEPART : "
:I
110 PRINT:INPUT"COLONNE DE DEPART :";J
120 IF S>9 THEN S=0:T=T+1
130 LOCATE 2*I+1,3*J+3,0:COLOR 4,6:PRINT
RIGHT$(STR$(S),1):IF T<>0 THEN LOCATE 2
*I+1,3*J+2,0:PRINT RIGHT$(STR$(T),1)
140 Z=INT(RND*8+1)
150 FOR Z=Z TO Z+8
160 A=I+VAL(MID$(A$,Z,1))-4
170 B=J+VAL(MID$(B$,Z,1))-4
180 IF A<1 OR B<1 OR A>8 OR B>8 THEN 200
190 IF SCREEN(2*A+1,3*B+3)=46 THEN 220
200 NEXT Z
210 LOCATE 19,26:PRINT"TOTAL :";T*10+S:P
LAY"CEG":END
220 I=A:J=B
230 S=S+1:SOUND INT(RND*4000+500),1
240 FOR K=1 TO 500:NEXT K
250 GOTO 120
```

```
REM Parcours du cavalier
STREAM #1:SCREEN GRAPHICS 640 FIXED, 200
FIXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "PARCOURS DU CAVALIER"
WINDOW OPEN
RANDOMIZE TIME:s=1:t=0:a$="56653223":b$=
"653223"+a$+"56"
a$=a$+a$: DIM t(8,8)
FOR i=1 TO 8
PRINT AT (1;2*i+1) CHR$(48+i) AT (6*i;1)
CHR$(64+i)
FOR j=1 TO 8:BOX 48*j-20;186-18*i,48,18 C
OLOR(7) FILL WITH 4:NEXT j
NEXT i
PRINT AT (1;19) "Cliquez sur la case de d
épart : Colonne      Ligne      "
REPEAT
a=BUTTON(1):x=XMOUSE+18:y=180-YMOUSE:x=IN
T(x/48):y=INT(y/16)
IF x>0 AND x<9 AND y>0 AND y<9 THEN c$=CH
R$(64+x):d$=CHR$(48+y) ELSE c$="      ":d$="
"
PRINT AT (42;19) COLOR(2) c$ AT (56;19) d
$
UNTIL a<>-1 AND x>0 AND x<9 AND y>0 AND y
<9
i=y:j=x:t(i,j)=1
LABEL boucle
IF s>9 THEN s=0:t=t+1
u=6*j:v=2*i+1:PRINT AT (u;v) COLOR(4) MOD
E(2) CHR$(s+48):PRINT CHR$(7)
IF t<>0 THEN u=6*j-1:PRINT AT (u;v) COLOR
(4) MODE(2) CHR$(t+48)
FOR l=1 TO 1500:NEXT l:z=RND(8)
FOR k=z TO z+8
a=i+VAL(MID$(a$,k,1))-4:b=j+VAL(MID$(b$,k
,1))-4
IF a<1 OR b<1 OR a>8 OR b>8 GOTO suivant
IF t(a,b)=0 THEN i=a:j=b:s=s+1:t(a,b)=1:G
OTO boucle
LABEL suivant
NEXT k
PRINT AT (1;21) "TOTAL : ";t*10+s;
PRINT CHR$(7)CHR$(7)CHR$(7):r$=INPUT$(1):
END
```

Adresse

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexe (pour 1 joueur).

Difficulté du jeu : difficile.

LE JEU :

Des chiffres, entre 0 et 9, vont apparaître au hasard, en divers emplacements de l'écran. Ils restent affichés pendant un temps très court. Le joueur doit réussir à appuyer sur la touche numérique correspondante, au moment où le chiffre est affiché.

Le chiffre tapé par le joueur sera alors affiché en grand format sur la gauche de l'écran ; et chaque réponse exacte rapporte 10 points. La durée de la partie est aléatoire et le score sera affiché en fin de partie.

LE PROGRAMME GW-BASIC :

Les lignes 50 et 60 contiennent, dans deux variables alphanumériques C\$ et B\$, les références qui seront utilisées pour dessiner les chiffres en grand format.

L'emplacement du chiffre tiré par le PC est choisi aux lignes 70 et 80, et le chiffre est tiré en ligne 90.

La ligne 100 réalise l'impression, et la ligne 110 détermine la durée d'affichage. La ligne 130 définit la durée du jeu, et la fin de partie est traitée par la ligne 150.

La boucle I (200-240) réalise le dessin grand format du chiffre frappé par le joueur.

Le contrôle de l'exactitude du résultat est effectué par la ligne 250.

POUR JOUER :

Le programme sera lancé par RUN, puis un premier chiffre apparaît, après une courte pause, vers le milieu de l'écran. Le joueur devra réagir suffisamment vite, tant qu'il reste à l'affichage, et le retaper. Le chiffre qu'il a tapé se dessinera alors en haut à gauche de l'écran, en grand format.

Si le résultat est exact, le joueur marquera 10 points.

Le temps de déroulement de la partie sera fonction des chiffres qui apparaîtront à l'affichage.

En fin de partie, le score sera affiché. Il sera possible de s'arrêter à tout moment en tapant CTRL et Break.

EXTENSIONS POSSIBLES :

On peut envisager de modifier le jeu pour faire participer plusieurs joueurs, en modifiant éventuellement l'affichage et en codant les touches du clavier mises à la disposition de chacun des joueurs.

On pourra aussi enlever des points à chaque résultat faux, modifier les règles de calcul du score et la durée du jeu, etc.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```
10 REM ADRESSE
20 SCREEN 0,1:WIDTH 40:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF
40 T=100:S=0:C$=CHR$(178):CLS
50 C$=C$+C$+C$+C$+" "+C$+C$+" "+C$
60 B$="144413333313121131314413312131121
41133331414114131"
70 X=INT(RND*15+1)
80 Y=INT(RND*15+15)
90 N=INT(RND*10)
100 LOCATE X+1,Y+1,0:COLOR INT(RND*15+1)
:PRINT N
110 FOR I=0 TO RND*600+200:A$=INKEY$:IF
A$="" THEN NEXT I
120 IF A$<>"" THEN 190
130 T=T-N
140 LOCATE X+1,Y+1:PRINT " "
150 IF T>=0 THEN 70
160 CLS
170 COLOR 7:PRINT:PRINT"VOTRE SCORE EST
DE ";S;" SUR 100"
180 PLAY"CDE":END
190 U=VAL(A$)
200 FOR I=1 TO 5
210 A=3*VAL(MID$(B$,5*U+I,1))-2
220 LOCATE 2*I+1,3:PRINT MID$(C$,A,3)
230 LOCATE 2*I+2,3:PRINT MID$(C$,A,3)
240 NEXT I
250 IF U=N THEN S=S+10
260 GOTO 130
```

```
REM Adresse
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "ADRESSE"
WINDOW OPEN
RANDOMIZE TIME
t=100;z=0:c$="111100001101":b1$=CHR$(27)+
CHR$(5)
b$="1444133333131211313144133121311214113
3331414114131"
LABEL tirage
x=RND(60)+10:y=RND(15)+2:n=RND(10)-1
PRINT AT (x;y) COLOR(n+1) POINTS(20) CHR$
(48+n)
p=RND(300)+250:i=0
REPEAT:a$=INKEY$:i=i+1:UNTIL a$<>"" OR i=
p
IF a$<>"" THEN GOSUB dessin
t=t-n:PRINT AT (x;y) POINTS(20) " "
IF t>=0 GOTO tirage
PRINT AT (1;19) "Votre score est de " s "
sur 100"
PRINT CHR$(7):r$=INPUT$(1):END

LABEL dessin
a=LET u=VAL(a$)
IF a<>0 THEN RETURN
FOR i=1 TO 5
a=3*VAL(MID$(b$,5*u+i,1))-2
FOR j=0 TO 2
c=VAL(MID$(c$,a+j,1)):PRINT AT (2+2*j;i+3
) COLOR(4*c) MODE(2-c) b1$;b1$
NEXT j,i
IF u=n THEN s=s+10
FOR i=1 TO 600:NEXT i
RETURN
```

La bombe

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

Vous êtes censé 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 GW-BASIC :

Une variable numérique R, non dimensionnée, contient les trois coordonnées définissant la position du joueur (ligne 230).

La position de la chambre contenant le colis piégé est définie par un tirage aléatoire de 3 variables A, B, C, en ligne 50.

La boucle I (60-100) dessine le détecteur.

Le signal de proximité est calculé aux lignes 110 et 120 et la ligne 130 permet l'affichage des coordonnées de la chambre à découvrir, dans le cas où l'on a perdu.

La ligne 140 permet l'affichage, à la fois du signal et du temps restant à courir.

La ligne 150 réalise le décompte du temps.

La ligne 160 contrôle si l'on a trouvé ou non.

La boucle I (200-240) permet d'entrer les nouvelles coordonnées, en utilisant le sous-programme placé aux lignes 260 à 300.

POUR JOUER :

Après le lancement du programme par RUN, le signal et le temps (en principe 200) sont affichés.

Le joueur est alors en position $x=0$, $y=0$, $z=0$, et on devra introduire successivement les trois coordonnées x, y, z de la chambre à explorer.

On fera, par exemple, 13 puis RETURN, 45 puis RETURN, etc.

Le nouveau signal est alors affiché, et le temps est 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.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```
10 REM LA BOMBE
20 SCREEN 0,1:WIDTH 40:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF:CLS
40 L=100:T=200:B$=CHR$(178)
50 A=INT(RND*L):B=INT(RND*L):C=INT(RND*L)
60 FOR I=1 TO 17
70 LOCATE 5,I+1:COLOR 5,0:PRINT B$
80 LOCATE 6,I+1:PRINT B$
90 LOCATE 7,I+1:PRINT B$
100 NEXT I
110 S=ABS((A/L+B*C/L)-(R(1)/L+R(2)+R(3)*L))
120 S$=LEFT$(STR$(S),7):S=VAL(S$)
130 IF T=0 THEN PRINT"PERDU . LA SOLUTION :";A;" ";B;" ";C:PLAY"EGC":END
140 COLOR 1:LOCATE 6,3:PRINT USING "####.###";S:COLOR 4:LOCATE 6,15:PRINT USING "####.###";T
150 PLAY"CDE":T=T-INT(SQR(L))
160 IF S=0 THEN LOCATE 22,1:COLOR 4:PRINT"BRAVO":PLAY"CDE":END
170 COLOR 3:LOCATE 11,1:PRINT" - X : "
180 LOCATE 13,1:PRINT" - Y : "
190 LOCATE 15,1:PRINT" - Z : "
200 FOR I=1 TO 3
210 R$=""
220 GOSUB 260
230 R(I)=VAL(R$)
240 NEXT I
250 GOTO 60
260 COLOR 7:LOCATE 2*I+9,8:PRINT "?";R$:COLOR 7:PRINT" ":BEEP
270 A$=INKEY$:IF A$="" THEN 270
280 IF ASC(A$)=13 THEN RETURN
290 R$=R$+A$
300 GOTO 260
```

```
REM La bombe
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 FIXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "LA BOMBE"
WINDOW OPEN
RANDOMIZE TIME:1=100:t=200
a=INT(RND*1):b=INT(RND*1):c=INT(RND*1)
BOX 0;0;640,200 COLOR(8) FILL WITH 8
a$="HOTEL":FOR i=1 TO LEN(a$):PRINT AT (3*i+8;4) COLOR(RND(6)) POINTS(20) MODE(2) MID$(a$,i,1):NEXT i
BOX 30;112,200,40 COLOR(2) FILL WITH 8
UNDEED
WHILE t<>0
s=ABS(((a/1)+b+(c*1))-((r(1)/1)+r(2)+(r(3)*1)))
PRINT AT (10;8) USING "####.###" s
PRINT AT (23;8) USING "####" t
PRINT CHR$(7):t=t-INT(SQR(1))
IF s=0 THEN PRINT AT (1;16) POINTS(20) COLOR(11) "BRAVO":r$=INPUT$(1):END
PRINT AT (1;12) " - X : "
PRINT AT (1;14) " - Y : "
PRINT AT (1;16) " - Z : "
FOR i=1 TO 3
r$="":GOSUB entrée:r(i)=VAL(r$)
NEXT i
WEND
PRINT AT (1;16) "PERDU. La solution était : " a;b;c
r$=INPUT$(1):END

LABEL entrée
PRINT AT (7;2*I+10) "?" r$:PRINT CHR$(7)
REPEAT:a$=INKEY$:UNTIL a$<>""
IF a$=CHR$(13) THEN RETURN
r$=r$+a$:IF LEN(r$)>2 THEN r$=""
GOTO entrée
```

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 GW-BASIC :

Les variables B\$ et C\$, des lignes 20 et 30, contiennent les informations codées qui permettront le dessin des chiffres en grand format.

Les lignes 60 à 150 s'occupent de la gestion des entrées.

La boucle I (160-190) réalise la représentation, en grand format, des chiffres sur l'écran, au fur et à mesure des entrées.

Le point décimal et le signe - sont gérés par les lignes 200 et 210.

Le nombre total des symboles entrés est contrôlé par la ligne 220.

Le résultat est inscrit grâce à un programme fonctionnant en automatique qui se trouve aux lignes 270 à 390 (l'automatisme est géré par la ligne 230, dans le programme principal).

Les sous-programmes placés aux lignes 400-410, 420-430, 440-450 et 460-470 effectuent l'opération indiquée, selon la valeur de Z (ligne 310).

Chaque résultat est mémorisé en A (F-1) (ligne 330).

POUR JOUER :

On devra lancer le programme par RUN.

Puis, on tape les chiffres du premier nombre à entrer qui vont s'afficher un par un, au fur et à mesure (on ne pourra entrer plus de 8 symboles, y compris le signe - 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 CTRL et Break, puis RUN. Taper CTRL et Break 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 PC), on peut aussi envisager une extension à plus de 8 chiffres, autoriser des opérations plus complexes (COS, SIN, ...), ajouter des mémoires, etc.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM CALCULATRICE
20 C$=CHR$(219):C$=C$+C$+C$+C$+C$ " +C$
+C$+ " +C$
30 B$="144413333313121131314413312131121
41133331414114131"
40 F=1:L=0:SCREEN 0,1:WIDTH 40:COLOR 4,2
:KEY OFF:CLS
50 N=0:T=0:K=0:R$=""
60 A$=INKEY$:IF A$="" THEN 60
70 IF K=1 THEN N=0:R$="":K=0
80 IF A$="" THEN 260
90 X=(1 AND A$="+")+(2 AND A$="-")+(3 AND
A$="*")+(4 AND A$="/")
100 R$=R$+A$
110 IF X<>0 THEN 260
120 IF R$="-" THEN A$=R$:GOTO 200
130 IF A$<"." OR A$>"9" OR A$="/" THEN 6
0
140 IF N=0 THEN CLS
150 IF A$="." OR A$="-" THEN 200
160 FOR I=1 TO 5
170 S=3*VAL(MID$(B$,I+5*VAL(A$),1))-2
180 LOCATE 2*I+4,N+2,0:PRINT MID$(C$,S,3
):LOCATE 2*I+5,N+2,0:PRINT MID$(C$,S,3)
190 NEXT I
200 IF A$="-" THEN CLS:LOCATE 10,2:PRINT
CHR$(220)
210 IF A$="." THEN LOCATE 14,N+2:PRINT C
HR$(254)
220 N=N-4*(N<28)
230 IF T<>0 THEN T=T+1:IF T<=LEN(R$) THE
N 380
240 IF K=1 AND L<>1 THEN 50
250 T=0:GOTO 60
260 IF R$="-" AND A(F)=0 THEN 120
270 N=0:K=1
280 IF (A$="*" OR A$="/") AND VAL(R$)=0 T
HEN A(F)=1: GOTO 300
290 A(F)=VAL(R$)
300 IF X<>0 THEN Z=X
310 ON Z GOSUB 400,420,440,460
320 IF A$<>"=" THEN F=F+1 ELSE F=2
330 A(F-1)=VAL(R$)
340 IF VAL(R$)=0 THEN A(F-1)=A(F)
350 R$=STR$(VAL(R$))
360 IF LEFT$(R$,1)=" " THEN R$=RIGHT$(R$
,LEN(R$)-1)
370 T=1
380 A$=MID$(R$,T,1)
390 GOTO 140
400 R$=STR$(A(F-1)+A(F)):A(F)=VAL(R$):L=
1
410 RETURN
420 IF F<>1 THEN R$=STR$(A(F-1)-A(F)):A(
F)=VAL(R$):L=1
430 RETURN
440 IF F<>1 THEN R$=STR$(A(F-1)*A(F)):A(
F)=VAL(R$):L=1
450 RETURN
460 IF F<>1 THEN R$=STR$(A(F-1)/A(F)):A(
F)=VAL(R$):L=1
470 RETURN

```

```

REM Calculatrice
STREAM #1:SCREEN GRAPHICS 640 FIXED, 200
FIXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "CALCULATRICE"
WINDOW OPEN:a$=CHR$(27)+CHR$(5)
DEF FN va(i$)=VAL(MID$(i$,1,LEN(i$)+(INST
R(1,"+-*/",d$)<>0)))
DEF FN ca$(i)=STR$(INT(i))+STRING$(-(FRAC
(i)<>0),".")+MID$(STR$(ROUND(FRAC(i),8)),
4)
c$="111100001101":b$="1444133333131211313
1441331213112141133331414114131"
f=1:l=0:GOSUB efface:FOR i=1 TO 9:a(i)=0:
NEXT
LABEL début
n=0:t=0:k=0:r$=""
LABEL intro
d$=INKEY$:IF d$="" GOTO intro ELSE IF d$=
" " THEN PRINT CHR$(7):END
IF k=1 THEN n=0:r$="":k=0
IF d$="" GOTO calcul
x=(1 AND d$="+")+(2 AND d$="-")+(3 AND d$
="*")+(4 AND d$="/")
r$=r$+d$:IF x<>0 GOTO calcul
LABEL sous
IF r$="-" THEN d$=r$:GOTO moins
IF d$<"." OR d$>"9" OR d$="/" GOTO intro
LABEL dessin
IF n=0 THEN GOSUB efface
IF d$="." OR d$="-" GOTO moins
FOR i=1 TO 5
s=3*VAL(MID$(b$,i+5*FN va(d$),1))-2:g=i+3
FOR j=0 TO 2:c=VAL(MID$(c$,s+j,1))+v:PRIN
T AT (2*n+j+3;i+4) COLOR(8-4*c) MODE(2-c)
a$:NEXT
NEXT i
LABEL moins
IF d$="-" THEN GOSUB efface:PRINT AT (4;6
) COLOR(4) a$:a$
IF d$="." THEN PRINT AT (2*n+4;9) COLOR(4
) POINTS(20) MODE(2) ", "
n=n-4*(n<28)
IF t<>0 THEN t=t+1:IF t<=LEN(r$) THEN d$=
MID$(r$,t,1):GOTO dessin
IF k=1 AND l<>1 GOTO début
t=0:GOTO intro
LABEL calcul
IF r$="-" AND a(f)=0 GOTO sous
n=0:k=1
IF (d$="*" OR d$="/") AND FN va(r$)=0 THE
N a(f)=1 ELSE a(f)=FN va(r$)
IF x<>0 THEN Z=x
ON Z GOSUB addit,soustrait,multi,divis
IF d$<>"=" THEN f=f+1 ELSE f=2
a(f-1)=FN va(r$):IF FN va(r$)=0 THEN a(f-
1)=a(f)
r$=FN ca$(FN va(r$))
IF LEFT$(r$,1)=" " THEN r$=RIGHT$(r$,LEN(

```

```

r$)-1)
t=1:d$=MID$(r$,t,1):GOTO dessin

LABEL addit
r$=FN ca$(a(f-1)+a(f)):a(f)=FN va(r$):l=1
RETURN

LABEL soustrait
IF f<>1 THEN r$=FN ca$(a(f-1)-a(f)):a(f)=
FN va(r$):l=1
RETURN

LABEL multi
IF f<>1 THEN r$=FN ca$(a(f-1)*a(f)):a(f)=
FN va(r$):l=1
RETURN

LABEL divis
IF f<>1 THEN r$=FN ca$(a(f-1)/a(f)):a(f)=
FN va(r$):l=1
RETURN

LABEL affiche
PRINT AT (2*h;g) COLOR(co) MODE(4) a$
RETURN

LABEL efface
BOX 0;0,640,200 COLOR(8) FILL WITH 8
RETURN

```

Anagrammes

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de réflexion (2 joueurs).

Difficulté du jeu : selon le niveau.

LE JEU :

Des lettres sont choisies au hasard par le PC, le nombre de lettres étant fonction du niveau de difficulté désiré. Puis les lettres du mot ainsi obtenu sont brouillées, et affichées au bas de l'écran ; un nouveau mot est alors confectionné, de façon aléatoire, à partir des mêmes lettres et il est affiché sous le premier, et ainsi de suite ; les deux joueurs (gauche et droit) disposent chacun d'une touche (1 ou 0) et doivent la presser dès qu'ils voient s'afficher au bas de l'écran un mot déjà présent sur l'écran ; leur score est alors évalué, et un message correspondant à leur résultat sera délivré.

LE PROGRAMME GW-BASIC :

Le niveau du jeu est entré à la ligne 40. La boucle I (70-90) réalise le choix des lettres.

Une deuxième boucle I (110-180) effectue le brouillage, et la ligne 130 permet de détecter la pression d'une touche.

La ligne 190 affiche un nouveau mot, et la ligne 200 fait monter d'une ligne sur l'écran les mots déjà obtenus.

La ligne 230 cherche quel est celui des deux joueurs qui a pressé une touche.

La boucle I (240-310) explore les mots présents sur l'écran ligne par ligne, alors que la boucle J (260-290) saisit un par un les caractères de ces mots.

On obtient ainsi un nouveau mot qui est comparé au mot initial en ligne 300 et le résultat BRAVO apparaît si c'est exact.

Sinon, la partie s'arrête sur l'affichage PERDU (ligne 320).

POUR JOUER :

On lance le programme par RUN.

On devra ensuite entrer le niveau de difficulté (en principe de 1 à 3) et faire RETURN. Les mots s'affichent en partant du bas de l'écran, et montent au fur et à mesure ; si l'un des joueurs croit reconnaître dans le mot qui apparaît un des mots déjà affichés, il devra immédiatement presser sa touche (1 ou 0), et son résultat sera affiché après un court instant.

La partie sera alors terminée.

EXTENSIONS POSSIBLES :

On pourra utiliser d'autres affichages que des mots, modifier les temps d'affichage, faire participer plus de deux joueurs, améliorer le son et la couleur, etc.

VERSION BASIC 2 AMSTRAD :

On fait appel ici à une fenêtre d'alerte pour choisir le niveau : on clique avec le bouton de gauche sur le niveau choisi (1, 2 ou 3) pour lancer le programme. On peut ensuite jouer soit en utilisant les boutons de la souris (chaque joueur dispose d'un bouton), soit en utilisant les touches ESC ou DEL du clavier.

```
10 REM ANAGRAMMES
20 SCREEN 0,1:WIDTH 40:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF:CLS
40 INPUT"NIVEAU ( DE 1 A 3 ) :";N
50 IF N<1 OR N>3 OR N<>INT(N) THEN 30
60 S$=""
70 FOR I=1 TO N+3
80 S$=S$+CHR$(INT(RND*26+65))
90 NEXT I
100 A$="":B$=S$
110 FOR I=1 TO LEN(S$)
120 N=INT(RND*LEN(S$)+1)
130 D$=INKEY$
140 IF D$="1" OR D$="2" OR D$="a" OR D$=
"0" THEN 230
150 IF MID$(B$,N,1)="-" THEN 120
160 A$=A$+MID$(B$,N,1)
170 MID$(B$,N,1)="-"
180 NEXT I
190 COLOR INT(RND*15+1):FOR I=1 TO LEN(A
$):LOCATE 23,I:PRINT MID$(A$,I,1):SOUND
INT(4000*RND+500),1:NEXT I
200 PRINT:FOR I=1 TO 80*LEN(B$):NEXT I
210 S$=A$
220 GOTO 100
230 IF D$="1" OR D$="2" THEN H$="JOUEUR
DE GAUCHE" ELSE H$="JOUEUR DE DROITE"
240 FOR I=0 TO 20
250 B$=""
260 FOR J=0 TO LEN(S$)-1
270 R$=CHR$(SCREEN(I+1,J+1))
280 B$=B$+R$
290 NEXT J
300 COLOR 7:IF B$=S$ THEN LOCATE I+1,LEN
(S$)+5:PRINT"BRAVO , ";H$:PLAY"CDE":END
310 NEXT I
320 LOCATE 24,LEN(S$)+5:PRINT"PERDU ";H$
;:PLAY"CCC"
```

REM Anagrammes

```
STREAM #1:SCREEN TEXT 80 FIXED, 22 FIXED
WINDOW FULL ON
WINDOW TITLE "ANAGRAMMES"
WINDOW OPEN
RANDOMIZE TIME:DIM t$(21):FOR i=0 TO 20:t
$(i)=" ";NEXT i
choix=ALERT 3 TEXT "", "Cliquez sur le niv
eau choisi " BUTTON "1","2","3"
CLS:c$="":FOR i=1 TO choix+3:c$=c$+CHR$(I
NT(RND*26+65)):NEXT
LABEL début
b$="":t$=c$
FOR i=1 TO LEN(c$)
REPEAT
n=RND(LEN(c$)):a=BUTTON(1):b=BUTTON(2):r=
INKEY
IF r=27 OR r=8 OR a<>-1 OR b<>-1 GOTO sui
te
UNTIL MID$(t$,n,1)<>"-"
b$=b$+MID$(t$,n,1):MID$(t$,n,1)="-"
NEXT i
co$=CHR$(INT(RND*8+33)):t$(19)=co$+b$
FOR i=19 TO 1 STEP-1
PRINT AT (1;i) COLOR(ASC(LEFT$(t$(i),1))-
25) MID$(t$(i),2)
NEXT i
FOR i=0 TO 19:t$(i)=t$(i+1):NEXT i
j=1:REPEAT:r=INKEY:a=BUTTON(1):b=BUTTON(2
):j=j+1:UNTIL j=7 OR r=8 OR r=27 OR a<>-1
OR b<>-1
IF j=7 GOTO début
LABEL suite
IF b<>-1 OR r=8 THEN h$="JOUEUR DE DROITE
" ELSE h$="JOUEUR DE GAUCHE"
FOR i=0 TO 17
IF b$=MID$(t$(i),2) GOTO fin
NEXT i
PRINT AT (8;19) "Perdu, " h$:r$=INPUT$(1)
:END
LABEL fin
PRINT AT (8;i+1) "BRAVO, " h$:r$=INPUT$(1
):END
```

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 GW-BASIC :

La boucle I (50-80) dessine le terrain de jeu, et les lignes 90 à 100 s'occupent de la gestion du tour des joueurs (numéro 1 ou 2 pour chaque joueur, et jeton O ou X correspondant).

La ligne 110 attend le coup joué.

La boucle I (120-140) vérifie la validité du coup.

La ligne 150 teste la fin de partie, en cas de match nul, et la ligne 160 refuse les coups non conformes (avec retour à la situation initiale, ligne 90).

Les lignes 170 et 180 s'occupent de l'affichage du coup, et la ligne 190 l'enregistre dans une variable R\$. La partie la plus intéressante du programme, qui pourrait servir à définir une stratégie machine, va de la ligne 210 à la ligne 300 : on utilise 3 boucles emboîtées pour vérifier si l'un des joueurs a obtenu, avec 3 pions, un total de 15. Le changement de joueur s'effectue en ligne 310.

POUR JOUER :

Lancer le jeu en tapant RUN.

Le terrain de jeu s'affiche, puis la question JOUEUR 1 : X, et le joueur qui a choisi les X devra donner le numéro de la case où il désire jouer si elle est libre, son coup y sera enregistré.

On obtient, ensuite, l'affichage JOUEUR 2 : O et le deuxième joueur procède de la même façon. Si l'un des joueurs gagne (total de 15 sur 3 coups), il obtient l'affichage BRAVO et la partie s'arrête.

En cas de match nul, la machine affiche C'EST FINI.

On fera une autre partie en tapant RUN.

EXTENSIONS POSSIBLES :

Le plus intéressant sera sans doute d'essayer de définir une stratégie machine (en essayant d'éviter que la machine ne joue toujours le 5 lorsqu'elle doit commencer).

On peut aussi, bien sûr, améliorer le graphisme et introduire des effets sonores pour les différents messages.

VERSION BASIC 2 AMSTRAD :

La souris est utilisée, après le lancement du programme, pour cliquer sur la case choisie, en appuyant sur le bouton de gauche. Le résultat (match nul ou joueur gagnant) est affiché en fin de partie.

```
10 REM LE JEU DU 15
20 SCREEN 0,1:WIDTH 40:COLOR 7,0:RANDOMIZE
30 F=INT(RND*2):R$="123456789":O$="":X$=""
40 A$=CHR$(200):B$=CHR$(201):C$=CHR$(205):D$=CHR$(186):E$=CHR$(187):F$=CHR$(188)
50 FOR I=0 TO 24 STEP 3
60 LOCATE 4,I+2:COLOR 2:PRINT B$+C$+E$:LOCATE 5,I+2:PRINT D$+" "+D$:LOCATE 6,I+2:PRINT A$+C$+F$
70 LOCATE 7,I+2:COLOR 3:PRINT (I+3)/3
80 NEXT I
90 COLOR 3:F+1:LOCATE 11,1:PRINT "JOUEUR ";F+1;
100 IF F=1 THEN PRINT" : 0" ELSE PRINT" : X"
110 A$=INKEY$:IF A$<"1" OR A$>"9" THEN 110
120 FOR I=1 TO LEN(R$)
130 IF A$=MID$(R$,I,1) THEN GOSUB 170
140 NEXT I
150 IF VAL(R$)=0 THEN LOCATE 21,1:PRINT "C'EST FINI":PLAY"CCC":END
160 GOTO 90
170 IF F=1 THEN O$=O$+A$:LOCATE 5,3*I:COLOR 4:PRINT"O"
180 IF F=0 THEN X$=X$+A$:LOCATE 5,3*I:COLOR 1:PRINT"X"
190 MID$(R$,I,1)="0"
200 IF F=1 THEN T$=O$ ELSE T$=X$
210 IF LEN(T$)<3 THEN 310
220 FOR I=1 TO 3
230 IF I=LEN(T$) THEN 310
240 FOR J=I+1 TO LEN(T$)
250 IF J=LEN(T$) THEN 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 THEN 330
280 NEXT K
290 NEXT J
300 NEXT I
310 F=ABS(F-1)
320 RETURN
330 LOCATE 11,1:COLOR 7
340 IF F=1 THEN PRINT"BRAVO , JOUEUR 0" ELSE PRINT"BRAVO , JOUEUR X"
350 PLAY"CEG"
```

```
REM Le jeu du 15
STREAM #1:SCREEN GRAPHICS 640 FIXED, 200
FIXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "LE JEU DU 15"
WINDOW OPEN:WINDOW MOUSE 2
RANDOMIZE TIME:h=RND(2)-1:r$="123456789":o$="":x$="":u=32
s$(0)="X":s$(1)="O":BOX 0;120,640,80 WIDTH 5 COLOR(4) FILL WITH 4 ROUNDED
FOR i=1 TO 9
BOX 64*i-30;158,60,30 WIDTH 3 COLOR(3) FILL WITH 4 ROUNDED
BOX 64*i-16;136,32,20 COLOR(0) FILL WITH 8
PRINT AT (8*i;7) POINTS(20) CHR$(48+i)
NEXT i
LABEL début
WINDOW MOUSE 3:PRINT AT (1;12) "Tour du joueur : " COLOR(2*h+2) POINTS(20) s$(h):PRINT CHR$(7)
PRINT AT (1;14) "Cliquez sur le numéro de la case choisie"
LABEL tirage
REPEAT:bo=BUTTON(1):x=XMOUSE+16:y=YMOUSE:x=INT(x/32):UNTIL bo<>-1
IF y<120 OR y>138 OR (x MOD 2)<>0 OR x<2 OR x>18 GOTO tirage
a$=RIGHT$(STR$(INT(x/2)),1):i=INSTR(r$,a$)
IF i<>0 THEN PRINT CHR$(7):GOSUB inscrit
IF VAL(r$)<>0 GOTO début
PRINT AT (1;17) "Match nul.... C'est fini....":r$=INPUT$(1):END
LABEL gagne
PRINT AT (1;17) "Bravo, joueur " POINTS(16) COLOR(2*h+2) s$(h)
PRINT CHR$(7)CHR$(7)CHR$(7):r$=INPUT$(1):END
LABEL inscrit
PRINT AT (8*i;4) POINTS(20) MODE(2) COLOR(2*h+2) s$(h)
IF h=0 THEN x$=x$+a$ ELSE o$=o$+a$
MID$(r$,i,1)="0":IF h=1 THEN t$=o$ ELSE t$=x$
IF LEN(t$)<3 GOTO change
FOR i=1 TO 3
IF i=LEN(t$) GOTO change
FOR j=i+1 TO LEN(t$)
IF j=LEN(t$) GOTO change
FOR k=j+1 TO LEN(t$)
IF VAL(MID$(t$,i,1))+VAL(MID$(t$,j,1))+VAL(MID$(t$,k,1))=15 GOTO gagne
NEXT k,j,i
LABEL change
h=ABS(h-1):PRINT CHR$(7):RETURN
```

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 GW-BASIC :

Une variable A\$ dimensionnée en ligne 60, contiendra la liste des touches concernées. Les valeurs des touches sont entrées de la ligne 70 à la ligne 120 par appel d'un sous-programme situé en 310.

Les boucles emboîtées J (130-170) et I (140-160) dessinent le clavier.

La ligne 190 définit la position du point imprimé par la ligne 190.

La touche frappée est testée aux lignes 200-240 et le point est effacé par la ligne 260.

Les affichages de fin sont réalisés aux lignes 280 à 300.

POUR JOUER :

On lance le programme par RUN. L'on dispose alors de quelques secondes pour frapper la touche qui est au-dessus de l'étoile, puis une autre étoile est affichée, etc.

La durée de la partie n'est pas limitée ; il suffit de taper 0 ou « espace » 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 lettre par une étoile. 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.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM CLAVIER
20 SCREEN 0,1:WIDTH 40:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF:CLS
40 PRINT:PRINT "(A)ZERTY ou (Q)WERTY ?":
R$=INPUT$(1):R$=CHR$(ASC(R$)+32*(R$>="a"
))
50 T=0:S=0:P=0
60 DIM A$(10,3):TIME$="00:00:00"
70 J=1:IF R$="A" THEN S$="AZERTYUIOP" EL
SE S$="QWERTYUIOP"
80 GOSUB 310
90 J=2:IF R$="A" THEN S$="QSDFGHJKLM" EL
SE S$="ASDFGHJKL;"
100 GOSUB 310
110 J=3:IF R$="A" THEN S$="WXCVRN,;:=" E
LSE S$="ZXCVRNM,./"
120 GOSUB 310
130 FOR J=1 TO 3
140 FOR I=1 TO 10
150 LOCATE S*J+1,3*I+J+1:COLOR 4,2:PRINT
A$(I,J)
160 NEXT I
170 NEXT J
180 X=INT(RND*10+1):Y=INT(RND*3+1):SOUND
INT(RND*4000+500),1
190 L=S*Y+1:C=3*X+Y+1:Z=SCREEN(L,C):COLO
R 1,12:LOCATE L,C:PRINT CHR$(Z)
200 FOR F=1 TO 300+RND*120:A$=INKEY$
210 T=T+1:IF A$="" THEN NEXT F
220 IF A$<>"" THEN A$=CHR$(ASC(A$)+32*(A
$>="a"))
230 IF A$=A$(X,Y) THEN S=S+1
240 IF A$="0" OR A$=" " THEN 280
250 P=P+1
260 LOCATE L,C:COLOR 4,2:PRINT CHR$(Z)
270 GOTO 180
280 COLOR 7,0:LOCATE 21,1:PRINT S;" TROU
VES JUSTES SUR ";P;" ESSAIS"
290 PRINT"VOTRE TEMPS :";INT(TIMER)
300 PLAY"CEG":END
310 FOR I=1 TO 10 : A$(I,J)=MID$(S$,I,1)
:NEXT I:RETURN

```

```

REM Clavier

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "CLAVIER"
WINDOW OPEN
RANDOMIZE TIME
DIM a$(10,3,2)
PRINT AT (1;5) "(A)ZERTY ou (Q)WERTY ?":r
$=INPUT$(1)
r$=UPPER$(r$):t=0:w=0:z=0:CLS
j=1:IF r$="A" THEN s$="AZERTYUIOP" ELSE s
$="QWERTYUIOP"
GOSUB tab1
j=2:IF r$="A" THEN s$="QSDFGHJKLM" ELSE s
$="ASDFGHJKL;"
GOSUB tab1
j=3:IF r$="A" THEN s$="WXCVRN,;:=" ELSE s
$="ZXCVRNM,./"
GOSUB tab1
FOR j=1 TO 3
FOR i=1 TO 10
c=13:GOSUB rec:PRINT AT (7*i+2*j-3;6*j) M
ODE(2) POINTS(16) EFFECTS(1) a$(i,j,1)
NEXT i,j
LABEL jeu
x=RND(10):y=RND(3)
i=x:j=y:c=2:GOSUB rec:PRINT AT (7*i+2*j-3
;6*j) MODE(2) POINTS(16) EFFECTS(1) a$(i
,j,1)
f=0:g=RND(130):REPEAT:f=f+1:t=t+1:r$=INKE
Y$:UNTIL r$<>"" OR f=100+g
r$=UPPER$(r$):IF r$=a$(i,j,1) THEN z=z+1:
PRINT CHR$(7)
IF r$=" " GOTO fin ELSE w=w+1:c=13:GOSUB
rec
PRINT AT (7*i+2*j-3;6*j) MODE(2) POINTS(1
6) EFFECTS(1) a$(i,j,1):GOTO jeu
LABEL fin
PRINT AT (1;21) z " justes sur " w "essai
s - TEMPS : " 0.01*t
r$=INPUT$(1):END

LABEL tab1
FOR i=1 TO 10
a$(i,j,1)=MID$(s$,i,1)
NEXT i
RETURN

LABEL rec
BOX 56*i+16*j-46;190-54*j,40,30 COLOR(c)
FILL WITH 12 WIDTH 4 ROUNDED
RETURN

```


Quatre à quatre

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion (un ou deux joueurs).

Difficulté du jeu : difficile.

LE JEU :

Il pourra être pratiqué à un ou plusieurs joueurs. Dans les deux cas, le but est d'arriver à échanger les pions X avec les pions 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 GW-BASIC :

La variable alphanumérique A\$, en ligne 30, contient la table des coups possibles (déplacement du cavalier).

La boucle I (50-90) dessine le terrain de jeu, et les positions de départ.

La ligne 110 permet d'afficher O ou X, selon le joueur qui a le trait.

La ligne 140 permet l'affichage du curseur clignotant qui indique sur quelle case on se trouve.

Les lignes 170 et 180 permettent le déplacement du curseur.

La ligne 190 regarde si l'on a appuyé sur une touche de 1 à 8, donnant la direction du déplacement.

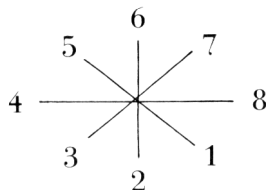
La ligne 200 réimprime la figure qui était sous le curseur. La ligne 230 transforme en nombre le symbole chiffre frappé. La ligne 220 réalise le contrôle de départ.

Les lignes 240 et 250 effectuent le déplacement. La ligne 260 réalise le contrôle d'arrivée. Le résultat est imprimé par la ligne 290, le changement de joueur est réalisé par la ligne 300 et un "." est imprimé à l'emplacement de départ. Le sous-programme 320-330 est utilisé pour les divers contrôles.

POUR JOUER :

On lance le programme par RUN. Le terrain de jeu est affiché, ainsi que le symbole permettant d'identifier le joueur qui a le trait. Le joueur peut choisir le pion qu'il désire déplacer en manœuvrant le curseur clignotant grâce aux touches de déplacement du curseur.

Pour déplacer son pion, il suffit, lorsque le curseur est sur ce pion, de taper la direction du déplacement, suivant le code :



Le coup sera refusé si la case de départ n'est pas conforme ou si la case d'arrivée est occupée. Sinon, le coup est réalisé, et le trait passe à l'autre joueur, qui déplace le curseur clignotant pour l'amener sur l'un des pions, et joue de la même façon.

Le joueur gagnant est celui qui réussit à amener le premier tous ses pions dans les cases initialement occupées par l'adversaire. On peut interrompre la partie à tout moment en tapant CTRL et Break.

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 PC de participer au jeu avec de bonnes chances de l'emporter.

VERSION BASIC 2 AMSTRAD :

Le joueur qui a le trait est affiché à l'écran. Il doit alors cliquer avec le bouton de droite de la souris sur la case de départ (contenant un de ses pions), puis, avec le même bouton, il clique sur la case d'arrivée (vide). Le trait passe alors au joueur suivant, qui procède de même en utilisant la souris. On termine la partie en cliquant simultanément sur les deux boutons.

```

10 REM QUATRE A QUATRE
20 SCREEN 0,1:WIDTH 40:COLOR 7,0:KEY OFF
30 H=1:A$="5665322356"
40 CLS:LOCATE 3,5:PRINT"1 2 3 4 5"
50 FOR I=1 TO 5
60 LOCATE 3*I+3,1:COLOR 7:PRINT RIGHT$(S
TR$(I),1):COLOR 3:LOCATE 3*I+3,5:PRINT".
. . . ."
70 IF I<3 THEN LOCATE 3*I+3,5:COLOR 4:PR
INT "O O"
80 IF I>3 THEN LOCATE 3*I+3,14:COLOR 1:P
RINT"X X"
90 NEXT I
100 X=1:Y=1
110 LOCATE 23,1:COLOR 1-3*(H=1):IF H=1 T
HEN PRINT"JOUEUR O" ELSE PRINT "JOUEUR X
"
120 GOSUB 320
130 I=J
140 LOCATE 3*Y+3,3*X+2,0:PRINT " "
150 A=X:B=Y
160 R$=INKEY$:R$=R$+R$:R$=MID$(R$,2,1)
170 Y=Y - (R$="P" AND Y<5) + (R$="H" AND
Y>1)
180 X=X - (R$="M" AND X<5) + (R$="K" AND
X>1)
190 IF R$<"9" AND R$>"0" THEN 220
200 LOCATE 3*B+3,3*A+2:COLOR -4*(J=79)-(
J=88)-7*(J=46):IF J=79 THEN PRINT "O" EL
SE IF J=88 THEN PRINT"X"ELSE PRINT","

```

```

REM Quatre à quatre
STREAM #1:SCREEN GRAPHICS 640 FIXED, 200
FIXED
USER SPACE 640, 200
WINDOW FULL ON
WINDOW TITLE "QUATRE A QUATRE"
WINDOW OPEN
RANDOMIZE TIME:BOX 4;4,334,180 COLOR(9) W
IDTH(11) FILL WITH 4 ROUNDED
FOR i=0 TO 5
FOR j=0 TO 5
t(i,j)=-47*(i<2 AND j<2)-56*(i>3 AND j>3)
+32
PRINT AT (6*i+5;3*j+5) POINTS(20) COLOR(2
-2*(t(i,j)=88)) CHR$(t(i,j))
NEXT j,i
k=RND(2)-1:PRINT AT (50;9) "Cliquez sur 1
a case"
PRINT AT (50;11) "de départ, puis sur " AT
(50;13) "celle d'arrivée"
LABEL début
s=0:PRINT CHR$(7)
PRINT AT (50;7) POINTS(16) "JOUEUR " COLO
R(2*k+2) CHR$(79+9*k)
LABEL boucle
REPEAT
a=BUTTON(1):b=BUTTON(2):x=XMOUSE:y=161-YM
OUSE:x=(INT(x/16)+1)/3:y=(INT(y/8)+3)/3
u=INT(x)-1:v=INT(y)-1:IF a+b=2 GOTO fin

```

```

210 GOTO 110
220 IF (I=46) OR (I=88 AND H=1) OR (I=79
    AND H=0) THEN 200
230 J=VAL(R$)
240 S=Y:Y=Y + VAL(MID$(A$,J,1))-4
250 T=X:X=X + VAL(MID$(A$,J+2,1))-4
260 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 200
270 GOSUB 320
280 IF J<>46 THEN A=T:B=S:J=88+9*(H=1):G
OTO 310
290 LOCATE 3*Y+3,3*X+2:COLOR 1-3*(H=1):I
F H=0 THEN PRINT "X" ELSE PRINT "O"
300 H=ABS(H-1):I=1:PLAY "CDE"
310 GOTO 200
320 J=SCREEN(3*Y+3,3*X+2)
330 RETURN

UNTIL a<>-1 AND x=INT(x) AND y<>INT(y) AN
D x>=1 AND x<7 AND y>=1 AND y<7
IF s=0 AND t(u,v)=79+9*k THEN PRINT AT (6
*u+5;3*v+5) POINTS(20) COLOR(8) CHR$(t(u,
v)):s=1:w=u:z=v:GOTO boucle
IF NOT((ABS(w-u)=1 AND ABS(z-v)=2) OR (AB
S(w-u)=2 AND ABS(z-v)=1)) GOTO boucle
IF s=1 AND t(u,v)=32 THEN PRINT AT (6*w+5
;3*z+5) POINTS(20) COLOR(2*k+2) CHR$(32)
AT (6*u+5;3*v+5) CHR$(79+9*k):t(u,v)=79+9
*k:k=1-k:t(w,z)=32:GOTO debut
GOTO boucle
LABEL fin
PRINT AT (50;17) "C'est fini...."
r$=INPUT$(1):END

```

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 GW-BASIC :

Le terrain de jeu est dessiné à l'aide des boucles I et J emboîtées (40-90). La ligne 110 s'occupe d'afficher le symbole cœur ou carreau correspondant au joueur qui a le trait.

Un symbole clignotant (ligne 140) 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 enregistre 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 Ø ou sur la barre d'espace, le coup sera refusé si la case n'est pas libre.

Si la case est libre, le symbole qui correspond à ce joueur est inscrit, et on change de joueur.

Lorsque la partie est finie, il convient de taper CTRL et Break pour sortir du programme. On pourra ensuite faire une autre partie avec RUN.

EXTENSIONS POSSIBLES :

On pourrait réaliser après chaque coup un contrôle pour vérifier si l'un des joueurs a gagné. On peut aussi réaliser un jeu machine mais il faudrait éviter qu'il soit par trop systématique (par exemple, la machine jouant toujours son premier coup sur la case centrale).

VERSION BASIC 2 AMSTRAD :

Chaque joueur devra à son tour cliquer avec le bouton de gauche de la souris sur la case où il désire poser son pion (ce n'est possible que si elle est libre). Le trait passe alors automatiquement au joueur suivant, qui procède de même. Le jeu se termine si toutes les cases sont occupées, ou si l'on clique simultanément sur les deux boutons.

```
10 REM TIC TAC TOE
20 SCREEN 0,1:WIDTH 40:COLOR 7,0:KEY OFF
:RANDOMIZE TIMER:CLS
30 A$=STRING$(7,CHR$(219)):B$(0)=CHR$(3)
:B$(1)=CHR$(4)
40 COLOR 3:FOR I=1 TO 3:FOR J=0 TO 2:LOC
ATE 2*I+J,4:PRINT A$:NEXT J,I
50 FOR I=1 TO 3
60 FOR J=1 TO 3
70 LOCATE 2*I+1,2*J+3:COLOR 0:PRINT CHR$(
219)
80 NEXT J
90 NEXT I
100 K=INT(RND*2)
110 LOCATE 19,1:COLOR 7:PRINT"JOUEUR ";:
COLOR 3*K+1:PRINT B$(K)
120 I=1:J=1
130 C=SCREEN(2*I+1,2*J+3):C=ABS(C):IF C=
219 THEN C=2
140 LOCATE 2*I+1,2*J+3,0:COLOR 5:PRINT"X
"
150 LOCATE 2*I+1,2*J+3,0:IF C<>2 THEN CO
LOR 3*(C-3)+1:PRINT CHR$(C) ELSE COLOR 0
:PRINT CHR$(219)
160 A$=INKEY$:A$=A$+A$:A$=MID$(A$,2,1)
170 J=J-((A$="6" OR A$="M") AND J<3)+((A
$="4" OR A$="K") AND J>1)
180 I=I-((A$="2" OR A$="P") AND I<3)+((A
$="8" OR A$="H") AND I>1)
190 IF C<>2 OR (A$<>"0" AND A$<>" ") THE
N 130
200 LOCATE 2*I+1,2*J+3:COLOR 3*K+1:PRINT
B$(K)
210 K=ABS(K-1)
220 IF K=1 THEN PLAY"CDE" ELSE PLAY"CEG"
230 GOTO 110
```

```
REM Tic tac toe
STREAM #1:SCREEN GRAPHICS 640 FIXED, 200
FIXED
USER SPACE 640, 200
WINDOW FULL ON
WINDOW TITLE "TIC TAC TOE"
WINDOW OPEN
RANDOMIZE TIME:k=RND(2)-1
PRINT AT (14;2) POINTS(16) "Cliquez sur 1
a case choisie"
FOR i=1 TO 4
LINE 64*i+80;151,64*i+80;26 WIDTH 32
BOX 144;170-36*i,182,18 FILL WITH 8
NEXT i
LABEL début
c=79+9*k:PRINT AT (50;9) POINTS(20) "JOUER
" COLOR(2*k+2) CHR$(c)
REPEAT
a=BUTTON(1):b=BUTTON(2):x=XMOUSE:y=YMOUSE
+8:x=INT(x/32)-3:y=9-INT(y/16)
u=x/2:v=y/2:IF a+b=2 OR s=9 GOTO fin
UNTIL (a<>-1 AND u>0 AND v>0 AND u<4 AND
v<4 AND x MOD 2=0 AND y MOD 2=0)
IF a<>-1 AND t(u,v)<>0 THEN GOTO début EL
SE s=s+1:t(u,v)=c:PRINT AT (8*u+14;4*v+5)
COLOR(2*k+2) POINTS(16) CHR$(c)
PRINT CHR$(7):k=ABS(k-1):GOTO début
LABEL fin
PRINT AT (50;15) "C'est fini..."
r$=INPUT$(1):END
```

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 une croix 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 GW-BASIC :

La ligne 80 fait apparaître la fumée au bas de l'écran ; la ligne 90 permet de changer (vers la droite ou vers la gauche) la direction de la colonne, et la ligne 110 fait monter la colonne.

Le joueur apparaît grâce à la ligne 130, et son déplacement est effectué par les lignes 140 à 190.

Les lignes 240 à 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'une croix colorée dans un coin de l'écran, va se déplacer sur elle à l'aide des 4 flèches. Dès qu'il arrive sur la fumée, il va automatiquement commencer à l'avalier, et devra essayer de suivre la colonne sans dévier (grâce aux flèches). Il gagne s'il arrive en bas de l'écran avant que la fumée n'arrive en haut.

L'affichage final indique alors BRAVO ou PERDU, ainsi que le score obtenu.

EXTENSIONS POSSIBLES :

On pourra, tout en conservant l'idée de départ, imposer d'autres règles : par exemple avaler toute la fumée, qui pourra aussi repartir d'un autre endroit si l'on dévie de la colonne initiale.

On pourra aussi améliorer les contrôles de fin de partie, et la qualité du graphisme.

VERSION BASIC 2 AMSTRAD :

Pour avaler la fumée, le joueur se déplace en cliquant sur les touches droite et gauche de la souris. Il peut également descendre en appuyant simultanément sur les deux touches. Le jeu sera plus facile à pratiquer si la souris est tenue en main, au lieu d'être posée sur la table.

```

10 REM SMOKEY
20 SCREEN 0,1:WIDTH 40:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF
40 E$=CHR$(3)
50 S=0:X=23:Y=INT(RND#24)
60 U=0:V=15
70 CLS
80 LOCATE X+1,Y+1:COLOR 4:PRINT E$
90 Y=Y-(RND<.5 AND Y<24) + (RND<.5 AND Y
>1)
100 T=1
110 IF RND<.7 THEN X=X-T
120 IF X=0 THEN 240
130 LOCATE U+1,V+1,0:COLOR 1:PRINT CHR$(
206)
140 IF T<>0 THEN FOR I=1 TO 300:A$=INKEY
$:A$=A$+A$:A$=MID$(A$,2,1):IF A$<>" " THE
N 160
150 IF T<>0 THEN NEXT I
160 LOCATE U+1,V+1:PRINT " "
170 U=U-((A$="2" OR A$="P") AND T=1 AND
X<24)
180 U=U-(T=0)
190 V=V-((A$="6" OR A$="M") AND V<24)+((
A$="4" OR A$="K") AND V>1)
200 C=SCREEN(U+2,V+1)
210 IF C<>3 AND U<22 THEN 80
220 T=0:S=S+1:SOUND INT(RND#4000+500),1
230 IF U<23 THEN 130
240 COLOR 7:IF S>=9 THEN PRINT"BRAVO":PL
AY"CDE":GOTO 260
250 PRINT"PERDU...":PLAY"CCE"
260 PRINT:PRINT"SCORE :";S

```

```

REM smokey

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "SMOKEY"
WINDOW OPEN
RANDOMIZE:DIM p(26,26):b$=CHR$(27)+CHR$(5
):c$=CHR$(178)
z=0:m=22:n=INT(RND#15+7):u=1:v=15
LABEL début
y=m:x=n:co=4:ta=10:mo=1:a$=b$:GOSUB aff
n=n-(RND<.5 AND n<25)+(RND<.5 AND n>1):p(
m,n)=1
t=1:m=m+t*(RND<.6)
REPEAT
x=v:y=u:co=2:mo=3:ta=16:a$=c$:GOSUB aff:I
F m=0 GOTO fin
IF t<>0 THEN i=1:REPEAT:i=i+1:a=BUTTON(1)
:b=BUTTON(2):UNTIL i=250 OR a<>-1 OR b<>-
1
GOSUB aff:u=u-(a+b=2 AND t=1 AND m<22)
u=u-(t=0):v=v-(b<>-1 AND v<25)+(a<>-1 AND
v>1)
et=p(u+1,v):IF et<>1 AND u<21 GOTO début
t=0:z=z+1:PRINT AT (v,u) COLOR(2) b$;CHR$
(7)
UNTIL u>=21
LABEL fin
IF t=0 THEN PRINT AT (40;10) "BRAVO" ELSE
PRINT AT (40;10) "PERDU..."
PRINT AT (40;15) "Votre score " z;CHR$(7)
r$=INPUT$(1):END

LABEL aff
PRINT AT (x;y) COLOR(co) POINTS(ta) MODE(
mo) a$
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 GW-BASIC :

La position de chaque météorite est déterminée par les lignes 50 et 70 ; elle est imprimée par les lignes 60 et 80.

L'avance des météorites est réalisée par la ligne 90.

Les lignes 90 et 100 vérifient si vous avez ou non heurté une météorite.

La ligne 120 dessine votre vaisseau spatial, et la ligne 140 permet de le déplacer.

Le compteur se trouve en ligne 150.

POUR JOUER :

Le jeu est lancé par l'ordre RUN.

Votre vaisseau apparaît vers le milieu de l'écran, et les météorites viennent à votre rencontre en partant du bas de l'écran. Pour vous déplacer, vous pouvez utiliser les touches de déplacement horizontal (→ et ←).

On peut arrêter la partie à tout moment en tapant CTRL et Break. Sinon, le jeu s'arrête lorsque vous heurtez une météorite, et le score est alors affiché.

EXTENSIONS POSSIBLES :

Il serait intéressant de rendre le jeu de plus en plus difficile en faisant apparaître de plus en plus de météorites au fur et à mesure du déroulement de la partie.

On peut aussi essayer de détruire les météorites en les pulvérisant au fur et à mesure de leur apparition.

VERSION BASIC 2 AMSTRAD :

Le déplacement du joueur vers la droite ou vers la gauche se fait en tenant la souris en main, et en cliquant sur les touches correspondantes. On peut arrêter à tout moment en cliquant simultanément sur les deux touches.


```

10 REM DANGER SPATIAL
20 SCREEN 0,1:WIDTH 40:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF:CLS
40 X=18:Z=0:Y=X
50 H=INT(RND*40)
60 IF RND<.6 THEN LOCATE 24,H+1:COLOR 1:
PRINT CHR$(127);
70 K=INT(RND*40)
80 IF RND<.5 THEN LOCATE 24,K+1:COLOR 3:
PRINT CHR$(127);
90 PRINT:B=SCREEN(13,X+1):C=SCREEN(13,X+
2)
100 IF B=127 OR C=127 THEN 170
110 LOCATE 12,Y:COLOR 0:PRINT"      ":LOCA
TE Y-1,10:PRINT"      "
120 LOCATE 13,X+1,0:COLOR 6:PRINT"###"
130 A$=INKEY$:A$=A$+A$:A$=MID$(A$,2,1)
140 Y=X:X=X-((A$="6" OR A$="M") AND X<39
)+(A$="4" OR A$="K") AND X>1)
150 Z=Z+1:SOUND INT(RND*3000+200),1
160 GOTO 50
170 COLOR 4:PRINT:PRINT"LE SCORE EST DE
";Z:PLAY"CEG"

```

```

REM Danger spatial
STREAM #1:SCREEN GRAPHICS 640 FIXED, 200
FIXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "DANGER SPATIAL"
WINDOW OPEN
RANDOMIZE TIME:w=35:z=0:c=75:DIM t(17,2)
LABEL début
h=RND(c):IF RND<.8 THEN PRINT AT (h;22) C
OLOR(2) CHR$(35);:t(1,1)=h
k=RND(c):IF RND<.8 THEN PRINT AT (k;22) C
OLOR(2) CHR$(35);:t(1,2)=k
FOR i=14 TO 1 STEP-1:t(i,1)=t(i-1,1):t(i,
2)=t(i-1,2):NEXT i
IF w=t(14,1) OR w=t(14,2) OR w+1=t(14,1)
OR w+1=t(14,2) GOTO fin
z=z+1:PRINT CHR$(7)
PRINT AT (w-1;11) POINTS(20) "      "
PRINT AT (w-1;12) COLOR(4) " /\ "
a=BUTTON(1):b=BUTTON(2):IF a+b=2 GOTO fin
w=w-(b<>-1 AND w>1)+(a<>-1 AND w<72)
GOTO début
LABEL fin
PRINT AT (1;3) "Score obtenu " z
r$=INPUT$(1):END

```

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), 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 GW-BASIC :

La position des ZOMBIES est définie par un tableau Z ligne 90 non dimensionné (c'est inutile en dessous de 10).

Les lignes 40 à 110 (boucle I) dessinent le cadre, les ZOMBIES et les trous noirs.

La ligne 120 définit la position initiale du joueur.

Une deuxième boucle I (140-230) réalise le déplacement du joueur.

Les lignes 260 à 320 contrôlent le déplacement du joueur.

POUR JOUER :

Après avoir lancé l'exécution par RUN, le joueur devra se déplacer en utilisant les 4 flèches et en veillant à ne pas sortir des limites du terrain.

Il devra manœuvrer de façon à faire venir les ZOMBIES sur les trous noirs, sans se faire lui-même coincer.

Pour arrêter la partie avant la fin, il suffira de taper CTRL et Break.

EXTENSIONS POSSIBLES :

Il serait intéressant d'agrandir le terrain de jeu, d'augmenter le nombre de ZOMBIES ou de trous noirs, d'améliorer les contrôles, de placer le joueur de façon plus aléatoire, éventuellement de modifier la règle.

On pourra aussi, bien sûr, améliorer la qualité du dessin et introduire la couleur.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM ZOMBIES
20 SCREEN 0,1:WIDTH 40:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF:CLS
40 FOR I=0 TO 21
50 COLOR 4,6:LOCATE 1,I+2:PRINT CHR$(127)
)
60 LOCATE I+1,1:PRINT CHR$(127):LOCATE I
+1,24:PRINT CHR$(127)
70 IF RND<.5 AND I<>21 THEN LOCATE INT(R
ND*20+2),I+1:COLOR 1:PRINT CHR$(178)
80 S=INT(RND*14+1)
90 Z(INT(I/3)+1)=5+S
100 LOCATE 22,I+2:COLOR 4,6:PRINT CHR$(1
27)
110 NEXT I
120 K=INT(RND*18+1)
130 P=0
140 FOR I=1 TO 4
150 P=P+Z(I)
160 IF Z(I)=0 THEN 230
170 LOCATE Z(I)+1,Z(I+4)+1:COLOR 2,0:PRI
NT" "
180 Z(I)=Z(I)-SGN(Z(I)-S)
190 Z(I+4)=Z(I+4)-SGN(Z(I+4)-K)
200 IF SCREEN(Z(I)+1,Z(I+4)+1)=178 THEN
Z(I)=0:GOTO 220
210 LOCATE Z(I)+1,Z(I+4)+1,0:COLOR 3,0:P
RINT"Z"
220 IF Z(I)=S AND Z(I+4)=K THEN PRINT"VO
US AVEZ PERDU..." :PLAY"CDC":END
230 NEXT I
240 IF P=0 THEN PRINT"BRAVO , VOUS AVEZ
DETUIT TOUS LES ZOMBIES":PLAY"CEG":END
250 LOCATE S+1,K+1,0:COLOR 5,12:PRINT"0"
260 FOR I=1 TO 700:A$=INKEY$:IF A$="" TH
EN NEXT I
270 LOCATE S+1,K+1:COLOR 7,0:PRINT" "
280 A$=A$+A$:A$=MID$(A$,2,1)
290 K=K-2*((A$="6" OR A$="M")-(A$="4" OR
A$="K"))
300 IF K<=1 OR K>=24 THEN PRINT"DOMMAGE.
..." :PLAY"CDE":END
310 S=S-2*((A$="2" OR A$="P")-(A$="8" OR
A$="H"))
320 IF S<=1 OR S>=22 THEN K=25:GOTO 300
330 GOTO 130

```

```

REM Zombies

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "ZOMBIES"
WINDOW OPEN
RANDOMIZE TIME:DIM t(40,20),z(10):a$=CHR$(
27)+CHR$(5)
BOX 4;14,320,180 COLOR(8) FILL WITH 8
FOR i=0 TO 39
PRINT AT (1;i+1+i*(i>20)) COLOR(4) MODE(2
) a$a$ AT (i+1;1) a$ AT (i+1;21) a$;
IF RND<.4 THEN j=RND(37)+2:k=RND(18)+1:PR
INT AT (j;k) COLOR(14) MODE(2) a$:t(j,k)=
1
s=RND(14)+2;z(INT(i/4)+1)=3+s
PRINT AT (40;i+1+i*(i>20)) COLOR(4) MODE(
2) a$a$;a$
NEXT i
k=RND(37)+2
LABEL debut
t=0
FOR i=1 TO 5
t=t+z(i):IF z(i)=0 GOTO conclus
PRINT AT (z(i);z(i+5)) COLOR(8) MODE(2) a
$
z(i)=z(i)-SGN(z(i)-k):z(i+5)=z(i+5)-SGN(z
(i+5)-s)
IF t(z(i),z(i+5))=1 THEN z(i)=0:t(z(i),z(
i+5))=0:GOTO conclus
PRINT AT (z(i);z(i+5)) MODE(4) "Z"
LABEL conclus
IF z(i)=k AND z(i+5)=s THEN PRINT AT (1;2
1) "Vous avez perdu":r$=INPUT$(1):END
NEXT i
IF t=0 THEN PRINT AT (1;21) "Bravo. Tous
les zombies sont morts":r$=INPUT$(1):END
PRINT AT (k;s) COLOR(2) MODE(2) "0"
j=0:REPEAT:m=INKEY:j=j+1:UNTIL m<>-1 OR j
=300
PRINT AT (k;s) COLOR(8) MODE(2) a$:PRINT
CHR$(7)
k=k-2*((m=333)-(m=331)):IF k<3 OR k>39 TH
EN PRINT "Dommage..." :r$=INPUT$(1):END
s=s-2*((m=336)-(m=328)):IF s<2 OR s>20 TH
EN PRINT "Dommage..." :r$=INPUT$(1):END
GOTO debut

```

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 colorée vous permet de déplacer les lettres, et un compteur affiche en permanence le nombre d'essais : arriverez-vous à retrouver le proverbe ?

LE PROGRAMME GW-BASIC :

Les lignes 40 et 50 impriment le terrain de jeu.

La phrase à reconstituer (qui pourra bien sûr être modifiée) est placée dans une variable R\$ en ligne 60.

Le mélange des lettres et l'impression sont faits par la boucle I (70-150).

La ligne 180 réalise l'affichage du carré coloré ; celui-ci est déplacé par les lignes 190 à 230 en utilisant les 4 flèches.

Les lignes 240 et 250 réimpriment la lettre concernée à l'emplacement du carré coloré.

La ligne 270 comptabilise le nombre d'essais qui est imprimé grâce à la ligne 170.

POUR JOUER :

On lance le programme par RUN.

La partie sera évidemment d'autant plus difficile que l'on ne connaît pas le proverbe à reconstituer. On peut déplacer les lettres et les faire passer à l'emplacement du carré coloré, à l'aide des 4 touches de direction (→, ←, ↓, ↑).

Le nombre d'essais est affiché en permanence, en dessous du puzzle.

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

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.

VERSION BASIC 2 AMSTRAD :

Le déplacement des lettres est différent de celui réalisé dans la version GW-Basic, où il n'était possible de déplacer qu'une seule lettre à la fois. Dans cette version, on peut, en effet, déplacer plusieurs lettres situées sur la même ligne ou sur la même colonne que la case libre. Il suffit pour cela de venir cliquer avec le bouton de gauche de la souris sur la case que l'on désire voir se libérer. Toutes les lettres de la ligne ou de la colonne correspondante vont alors glisser jusqu'à occuper l'ancienne case vide. On peut arrêter à tout moment en cliquant simultanément sur les deux touches.

```
10 REM TAQUIN DE LETTRES
20 SCREEN 0,1:WIDTH 40:COLOR 6,3
30 RANDOMIZE TIMER:KEY OFF
40 CLS:A$(0)=CHR$(201):A$(1)=CHR$(205):A
$(2)=CHR$(187):A$(3)=CHR$(186):A$(4)=CHR
$(200):A$(5)=CHR$(188)
50 FOR I=1 TO 6:FOR J=1 TO 6:LOCATE 3*I-
1,3*J-1:PRINT A$(0)+A$(1)+A$(2):LOCATE 3
*I,3*J-1:PRINT A$(3)+A$(4)+A$(5):NEXT
J,I
60 A$="":R$="ONASOUVENTBESIND'UNPLUSPET
ITQUESOI"
70 FOR I=1 TO 35
80 X=INT(RND*35+1)
90 IF MID$(R$,X,1)="-" THEN 80
100 A$=A$+MID$(R$,X,1)
110 MID$(R$,X,1)="-"
120 F=ASC(MID$(A$,I,1))
130 X=I-6*INT(I/6):Y=INT(I/6)
140 LOCATE 3*Y+3,3*X+3:COLOR 4:PRINT CHR
$(F)
150 NEXT I
160 X=0:Y=0:S=0:C=32
170 COLOR 8:LOCATE 21,1:PRINT"OMBRE D'E
SSAIS ":"S
180 COLOR 0:LOCATE 3*Y+3,3*X+3,0:PRINT C
HR$(219)
190 T$=INKEY$:T$=T$+T$:T$=MID$(T$,2,1)
200 IF T$="" THEN LOCATE 3*Y+3,3*X+3:PRI
NT" ":GOTO 180
210 I=X:J=Y
220 X=X - ((T$="6" OR T$="M") AND X<5) +
((T$="4" OR T$="K") AND X>0)
230 Y=Y - ((T$="2" OR T$="P") AND Y<5) +
((T$="8" OR T$="H") AND Y>0)
240 C=SCREEN(3*Y+3,3*X+3)
250 LOCATE 3*J+3,3*I+3:COLOR 4,3:PRINT C
HR$(C)
260 IF I=X AND J=Y THEN PLAY"CCC":GOTO 2
80
270 S=S+1:SOUND INT(RND*4000+500),1
280 GOTO 170
```

```
REM Taquin de lettres

STREAM #1:SCREEN GRAPHICS 640 FIXED, 200
FIXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "TAQUIN DE LETTRES"
WINDOW OPEN
RANDOMIZE TIME:s=1:a$=""
r$="ONASOUVENTBESIND'UNPLUSPETITQUESOI "
FOR i=0 TO 35
x=RND(LEN(r$)):a$=a$+MID$(r$,x,1):f$=LEFT
$(r$,x-1)
r$=f$+MID$(r$,x+1):t((i MOD 6)+1,INT(i/6)
+1)=ASC(MID$(a$,i+1,1))
IF ASC(MID$(a$,i+1,1))=32 THEN u=(i MOD 6
)+1:v=INT(i/6)+1
NEXT i
FOR i=1 TO 6:FOR j=1 TO 6:GOSUB aff:NEXT
j,i
LABEL boucle
REPEAT
a=BUTTON(1):b=BUTTON(2):x=XMOUSE+6:y=182-
YMOUSE:x=INT(x/32):y=INT(y/16)
PRINT AT (50;5) "ESSAI NUMERO " s:IF a+b=
2 GOTO fin
UNTIL a<>1 AND x>0 AND y>0 AND x<7 AND y
<7
IF x<>u AND y<>v THEN PRINT CHR$(7)CHR$(7
)CHR$(7):GOTO boucle
IF u=x AND v<y THEN FOR f=v TO y-1:t(u,f)
=t(u,f+1):i=u:j=f:GOSUB aff:NEXT f
IF u=x AND v<y THEN FOR f=v TO y+1 STEP-1
:t(u,f)=t(u,f-1):i=u:j=f:GOSUB aff:NEXT f
IF v=y AND u<x THEN FOR f=u TO x-1:t(f,v)
=t(f+1,v):i=f:j=v:GOSUB aff:NEXT f
IF v=y AND u<x THEN FOR f=u TO x+1 STEP-1
:t(f,v)=t(f-1,v):i=f:j=v:GOSUB aff:NEXT f
t(x,y)=32:u=x:v=y:i=x:j=y:GOSUB aff:s=s+1
:PRINT CHR$(7):GOTO boucle
LABEL fin
PRINT AT (50;13) "C'est fini.... "
r$=INPUT$(1):END

LABEL aff
BOX 32*i-6;188-18*j,32,18 COLOR(5) FILL W
ITH 3 WIDTH 5
PRINT AT (4*i+1;2*j+1) MODE(2) COLOR(2) P
OINTS(16) CHR$(t(i,j))
RETURN
```

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 GW-BASIC :

Les boucles emboîtées P (50-120) et Q (70-110) dessinent le terrain de jeu et la ligne 140 affiche un X ou un O, selon le joueur qui commence.

La boucle P (220-350) contrôle la validité du coup (la case est-elle libre : lignes 330 et 340), en utilisant une partie de programme qui se trouve aux lignes 390 à 400 (a-t-on pris un ou plusieurs ennemis en ligne 390 et retournement éventuel autorisé par la ligne 420 : on refait un balayage dans toutes les directions, si l'encadrement s'est révélé possible).

La ligne 220 définit les 8 directions d'exploration.

Les lignes 260 et 270 calculent les déplacements à effectuer pour explorer ces 8 directions.

La ligne 250 réalise les déplacements.

La ligne 370 assure le changement de joueur, s'il y a eu retournement (contrôlé en ligne 360).

POUR JOUER :

Après RUN, chaque joueur entrera à son tour les coordonnées (ligne, RETURN, puis colonne, RETURN) de la case où il désire jouer ; si le coup est accepté, les pions sont retournés et l'on change de joueur ; si un coup n'est pas possible, on entre 0 en première coordonnée pour le changement de joueur.

EXTENSIONS POSSIBLES :

Avant tout, bien sûr, améliorer l'affichage avec des couleurs, une grille plus aérée.

On pourra aussi imaginer un jeu machine pour le PC.

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

VERSION BASIC 2 AMSTRAD :

On commence par donner la taille du terrain de jeu, comme dans la version GW-Basic. Puis, le joueur désigné (x ou o) devra prendre la souris et cliquer avec la touche de gauche sur la case où il désire jouer. Le coup est réalisé s'il est possible, et le trait passe alors au joueur suivant. Si aucun coup n'est possible, le changement de joueur se fait en cliquant sur la touche de droite. On peut aussi arrêter la partie en appuyant simultanément sur les deux touches.

```

10 REM OTHELLO
20 CLS:INPUT"DIMENSION DE LA GRILLE :";A
30 IF A/2<>INT(A/2) THEN GOTO 20
40 SCREEN 0,1:WIDTH 40:COLOR 0,7:RANDOMIZE
   TIMER:KEY OFF:CLS
50 FOR P=1 TO A
60 LOCATE 1,2:P+1:PRINT RIGHT$(STR$(P),1)
70 FOR Q=1 TO A
80 LOCATE 2:P+1,1:COLOR 0:PRINT RIGHT$(STR$(P),1):LOCATE 2:P+1,2:Q+1:PRINT " "
90 IF Q=A/2+1 AND P=A/2 THEN LOCATE 2:P+1,2:Q-1:COLOR 1:PRINT "O ";:COLOR 4:PRINT "X"
100 IF Q=A/2+1 AND P=A/2+1 THEN LOCATE 2:P+1,2:Q-1:COLOR 4:PRINT"X ";:COLOR 1:PRINT"O"
110 NEXT Q
120 NEXT P
130 K=INT(RND*2):SO=2:SX=2
140 LOCATE 7,24:COLOR 0:PRINT"JOUEUR ";:COLOR 1-3*(K=1):IF K=1 THEN PRINT "X":PLAY"CDE" ELSE PRINT "O":PLAY"CEG"
150 LOCATE 9,32:COLOR 0:PRINT"      ":LOCATE 11,32:PRINT"      "
160 LOCATE 15,25:COLOR 5:PRINT "X ";:SX;
170 " ":LOCATE 15,33:PRINT "O ";:SO;
180 IF SX+SO=A*A THEN PLAY "CDEFG":END
190 COLOR 0:LOCATE 9,24:INPUT"LIGNE :";U
200 LOCATE 11,24:INPUT"COLONNE: ";V
210 H=1
220 FOR P=1 TO 8
230 S=0
240 W=0
250 T=2*U:R=2*V
260 F=(P>3 AND P<7) - (P<3 OR P>7)
270 G=(P>1 AND P<5) - (P)=6)
280 Q=46 - 42*(S=1) - 33*(S=2)
290 IF S=0 THEN 320
300 SO=SO+(C=88)-(C=79):SX=SX+(C=79)-(C=88)
310 LOCATE T+1,R+1:COLOR -1*(K=0)-4*(K=1):PRINT CHR$(Q)
320 T=T+2:F:R=R+2*G
330 C=SCREEN(T+1,R+1)
340 IF C=79 OR C=88 THEN 390
350 NEXT P
360 IF H<>0 THEN 140 ELSE SO=SO-(K=0):SX=SX-(K=1)
370 K=ABS(K-1)
380 GOTO 140
390 IF (C=79 AND K=1) OR (C=88 AND K=0)

```

```

REM Othello
STREAM #1:SCREEN GRAPHICS 640 FIXED, 350
FIXED
USER SPACE 640,350
WINDOW FULL ON
WINDOW TITLE "OTHELLO"
WINDOW OPEN
REPEAT
PRINT AT (1;1) "Dimension de la grille :
";
INPUT AT (27;1) "",a$:a=VAL(a$)
UNTIL a>1 AND a MOD 2=0 AND a<11
CLS:DIM t(a+2,a+2)
FOR i=1 TO a+1:LINE 28;360-32*i,28+32*a;360-32*i:LINE 32*i-4;328,32*i-4;328-32*a:NEXT i
FOR p=1 TO a
PRINT AT (4*p+2;1) CHR$(47+p) AT (1;2*p+1) CHR$(64+p)
FOR k=1 TO a:IF k<>(a/2)+1 GOTO fin
IF p=a/2 THEN PRINT AT (4*k-2;2*p+1) COLOR(2) CHR$(79) AT (4*k+2;2*p+1) COLOR(4) CHR$(88)
IF p=(a/2)+1 THEN PRINT AT (4*k-2;2*p+1) COLOR(4) CHR$(88) AT (4*k+2;2*p+1) COLOR(2) CHR$(79)
LABEL fin:t(p,k)=0
NEXT k,p:t(a/2,a/2)=79:t((a/2)+1,a/2)=88:t(a/2,(a/2)+1)=88:t((a/2)+1,(a/2)+1)=79
k=RND(2)-1:so=2:sx=2
LABEL début
PRINT AT (54;5) COLOR(2) "Pions O : ";so
AT (54;7) COLOR(4) "Pions X : ";sx
PRINT AT (54;9) "JOUEUR " COLOR(2*k+2) CHR$(79+9*k)
PRINT AT (54;11) "Cliquez sur la case"
REPEAT
c=BUTTON(1):b=BUTTON(2):x=XMOUSE+3:y=182-YMOUSE:u=2*INT(x/32):v=2*INT(y/16)
IF v/2>0 AND v/2<a+1 AND u/2>0 AND u/2<a+1 THEN PRINT AT (54;15) "Ligne " CHR$(64+v/2) AT (54;17) "Colonne " CHR$(47+u/2)
IF v/2<1 OR v/2>a OR u/2<1 OR u/2>a THEN PRINT AT (54;15) STRING$(20,32) AT (54;17) STRING$(20,32)
UNTIL (c<>-1 AND u/2>0 AND v/2>0 AND u/2<a+1 AND v/2<a+1) OR b<>-1 OR c+b=2
IF c<>-1 AND b=-1 AND t(u/2,v/2)<>0 THEN PRINT CHR$(7):GOTO début
IF c<>-1 AND b=-1 THEN GOSUB change:IF so+sx<>a*a GOTO début
IF b<>-1 AND c=-1 THEN k=ABS(k-1):GOTO début

```

```

THEN W=1
400 IF S<>0 THEN LOCATE T+1,R+1:COLOR -1
*(K=0)-4*(K=1):PRINT CHR$(Q)
410 IF S<>0 THEN SO=SO-(C=88)+(C=79):SX=
SX+(C=88)-(C=79)
420 IF W=1 AND ((C=79 AND K=1) OR (C=88
AND K=0)) THEN 320
430 IF W=0 OR S<>0 THEN 350
440 S=1-(K=0)
450 H=-H*(S=0)
460 GOTO 240

PRINT AT (54;19) "C'est fini..."
r$=INPUT$(1):END

LABEL change
h=1:p=1:s=0
LABEL essai
w=0:t=u:r=v:f=(p>3 AND p<7)-(p<3 OR p>7):
g=(p>1 AND p<5)-(p>=6)
q=32-56*(s=1)-47*(s=2):IF s=0 GOTO suite
y=r+2:x=2*t:c=q:PRINT AT (x+2;y-1) COLOR(
2-2*(q=88)) CHR$(q):t(t/2,r/2)=c
so=so+(co=88)-(co=79):sx=sx+(co=79)-(co=8
8)
LABEL suite
t=t+2*f:r=r+2*g:co=t(t/2,r/2):IF co=79 OR
co=88 GOTO tire
LABEL boucle
p=p+1:IF p<9 THEN s=0:GOTO essai ELSE IF
h<>0 THEN RETURN
so=so-(k=0):sx=sx-(k=1):k=ABS(k-1):RETURN
LABEL tire
IF (co=79 AND k=1) OR (co=88 AND k=0) THE
N w=1
IF s<>0 THEN y=r+2:x=2*t:c=q:PRINT AT (x+2
;y-1) COLOR(2-2*(q=88)) CHR$(q):t(t/2,r/2
)=c
IF s<>0 THEN so=so-(co=88)+(co=79):sx=sx-
(co=79)+(co=88)
IF w=1 AND ((co=79 AND k=1) OR (co=88 AND
k=0)) GOTO suite
IF w=0 OR s<>0 GOTO boucle
s=1-(k=0):h=-h*(s=0):GOTO essai

```


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 GW-BASIC :

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 à 280 et 300 à 320.

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, la partie reprend en tapant RUN.

On peut aussi terminer à tout moment par CTRL et Break.

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.

VERSION BASIC 2 AMSTRAD :

Dans cette version, le déplacement de la raquette sur l'écran sera effectué en prenant la souris en main (pour éviter de faire bouger la balle) et en appuyant sur les touches droite ou gauche pour déplacer la raquette. La partie s'arrête lorsque l'on rate la balle, ou bien si l'on appuie sur CTRL et C.

```
10 REM MUR DE BRIQUES
20 SCREEN 0,1:WIDTH 40:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF
40 CLS:A$=CHR$(178):B$=CHR$(219)
50 FOR I=1 TO 18
60 LOCATE 1,10+I:PRINT B$
70 LOCATE I+1,11:PRINT B$:LOCATE I+1,28:
PRINT B$
80 NEXT I
90 FOR I=1 TO 7
100 FOR J=11 TO 26
110 LOCATE I+1,J+1:COLOR I:PRINT A$
120 NEXT J
130 NEXT I
140 H=-1:K=-.5:J=18:I=17:Y=J
150 LOCATE 20,Y-1:COLOR 7:PRINT " ---- "
160 D$=INKEY$:D$=D$+D$:D$=MID$(D$,2,1)
170 Y=Y - ((D$="6" OR D$="M") AND Y<28)
+ ((D$="4" OR D$="K") AND Y>8)
180 IF T=1 THEN LOCATE I+1,J+1:PRINT " "
190 LOCATE 21,1:PRINT"SCORE":S
200 IF I>21 THEN PLAY"CCC":END
210 I=I+H:J=J+K
220 C=SCREEN(I+1,J+1)
230 IF C=32 THEN COLOR 2:LOCATE I+1,J+1:
PRINT "O":T=1:GOTO 150
240 IF C=219 THEN 280
250 IF C=178 THEN F=12-I:COLOR 2:LOCATE
I+1,J+1:PRINT"O":T=1:GOTO 300
260 H=-H
270 K=RND-RND
280 K=-K:T=0
290 SOUND 2000,1:GOTO 150
300 S=S+9+F
310 K=-(K-(2*(RND<.5)+1)*RND/3)
320 H=-H
330 SOUND 400,2:GOTO 150
340 RETURN
```

```
REM Mur de briques
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "MUR DE BRIQUES"
WINDOW OPEN
RANDOMIZE TIME:a$=CHR$(27)+CHR$(5):b$="
":DIM t(24,19)
PRINT AT (8;2) MODE(4) b$:FOR i=0 TO 15:P
RINT AT (6;i+2) a$a$ AT (31;i+2) a$a$:NEX
T
FOR j=1 TO 7
PRINT AT (8;j+2) COLOR(j) MODE(4) b$
FOR i=1 TO 23
BOX B#1+49;182-9*j,8,9 COLOR(0):t(i,j)=j
NEXT i,j
FOR j=8 TO 19:t(0,j)=35:t(24,j)=35:NEXT j
FOR i=1 TO 23:t(i,0)=0:FOR j=8 TO 19:t(i,
j)=32:NEXT j,i
z=0:t=0:h=-1:k=-.5:j=14:i=12:y=8*j
LABEL début
LINE y-20;40,y+10;40 WIDTH 3
a=BUTTON(1)+BUTTON(2)+2:a=a+(a=4):IF a=0
GOTO suite
y0=y:y=y-(16*(a=2) AND y<240)+(16*(a=1) A
ND y>56)
LINE y0-20;40,y0+10;40 COLOR(0) WIDTH(3)
LABEL suite
PRINT AT (j;i) " ":j0=j:i0=i:t=0
i=i+h:j=j+k:c=t(INT(j-7),INT(i-2)):IF i=1
9 OR c=0 GOTO retour
IF c=32 THEN PRINT AT (j;i) "O":GOTO debu
t
IF c=35 THEN t=1:PRINT AT (j0;i0) a$:GOTO
mur
IF c<8 THEN t(INT(j-7),INT(i-2))=32:t=0:G
OTO brique
LABEL retour:IF i>18 AND ABS(8*j-y)>30 G
OTO fin
h=-h:k=k+(RND-RND)*.5
LABEL mur:k=-k:PRINT CHR$(7):GOTO début
LABEL brique:k=-(k-(2*(RND<.5)+1)*RND/3):
z=z+8-c
PRINT CHR$(7):h=-h:GOTO début
LABEL fin
PRINT AT (50;11) "Votre score : " z
r$=INPUT$(1):END
```

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 10 secondes. Le jeu commence lorsque l'on arrive à 0 seconde. Des ronds apparaissent alors en bas et à gauche de l'écran, formant une ligne diagonale, et il vous faudra appuyer sur la touche 0 pour arrêter l'affichage des points dès que vous estimez que le nombre de points demandé au début est apparu à l'écran.

On obtient alors un affichage BRAVO ou PERDU selon le résultat obtenu.

LE PROGRAMME GW-BASIC :

Le nombre de points noirs sera choisi au hasard, en ligne 50.

Le temps est décompté par les lignes 60 à 100.

Les points sont imprimés et la ligne diagonale est fabriquée par la ligne 130.

La pression de la touche 0 ou de la barre d'espace est détectée en ligne 160, et la ligne 170 contrôle la fin de partie (aucun réflexe).

Les affichages de fin sont exécutés par les lignes 200 à 220.

POUR JOUER :

On lancera l'exécution par RUN.

On obtient alors l'affichage de la phrase LE BON REFLEXE EST A... qui indique le nombre de points gagnants.

Le temps sera ensuite décompté sous la forme ATTENTION : IL RESTE ... SECONDES qui indique le temps restant avant l'apparition de la diagonale.

À l'affichage de 0, la diagonale apparaît et commence à se déplacer sur l'écran.

Si l'on estime que le nombre de points réclamé a été atteint, on devra presser la touche 0 ou de la barre d'espace.

L'affichage indique alors BRAVO si l'on a gagné, ou PERDU dans le cas contraire ; puis la phrase AUTRE ESSAI (O/N) est affichée, répondre O pour faire une autre partie.

EXTENSIONS POSSIBLES :

On pourra envisager une adaptation du jeu à deux joueurs ou plus, en laissant la diagonale se former complètement et en testant la coïncidence lors de la pression de la touche affectée au joueur.

On peut aussi réaliser un graphisme plus soigné et modifier le principe du déplacement des objets sur l'écran (déplacement horizontal, vertical ou apparition au hasard...).

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic, mais seule la barre d'espace est autorisée pour jouer.

```

10 REM DIAGO                                REM Diago
20 SCREEN 0,1:WIDTH 40:COLOR 0,7           STREAM #1:SCREEN GRAPHICS 640 FIXED,5000
30 RANDOMIZE TIMER:KEY OFF                 FIXED
40 CLS:TIME$="00:00:00":A$(1)="S ":A$(0)   USER SPACE 640,200
=" "                                         WINDOW FULL ON
50 X=INT(RND*18+3):ON TIMER(10) GOSUB 11    WINDOW TITLE "DIAGO"
0                                           WINDOW OPEN
60 S=10-INT(TIMER):TIMER ON               RANDOMIZE TIME
70 PRINT:PRINT"LE BON REFLEXE EST A ";X     X=RND(15)+2:S=9:BOX 200;100,260,70 WIDTH
80 LOCATE 11,1,0:PRINT"ATTENTION . IL RE   6 ROUNDED
STE ";S;" SECONDE";:PRINT A$(-(S>1)):SOU   PRINT AT (28;6) "LE BON REFLEXE EST A" CO
ND INT(4000+500),1                        LOR(4) POINTS(16) STR$(X)
90 S=S-1                                    REPEAT
100 IF S<>10-INT(TIMER) THEN 100 ELSE 80    PRINT AT (30;9) "IL RESTE" COLOR(2) POINT
110 CLS                                     S(16) STR$(S) COLOR(1) POINTS(10) LEFT$("
120 U=0                                     SECONDES",9+(S=1))+ " "
130 LOCATE 24,U+1,0:COLOR 1:PRINT "O"      PRINT CHR$(7):S=S-1:FOR i=1 TO 6000:NEXT
140 FOR I=1 TO 30 : A$=INKEY$:IF A$="" T    i
HEN NEXT I                                UNTIL S=0
150 U=U+1                                  CLS:U=1
160 IF A$="O" OR A$=" " THEN 190            REPEAT
170 IF U>22 THEN COLOR 5:PRINT"PERDU":PL   PRINT AT (2*U;22) COLOR(2) "O"
AY"GEC":GOTO 210                           WINDOW SCROLL 0;30:U=U+1
180 GOTO 130                               FOR i=1 TO 10:A$=INKEY$:IF A$="" THEN NEX
190 IF U<>X THEN U=23:GOTO 170              T i
200 COLOR 4:PRINT"BRAVO":PLAY"CDE"         IF U>22 THEN PRINT "PERDU";CHR$(7);CHR$(7
210 COLOR 0:PRINT:PRINT"AUTRE ESSAI (O/N   );CHR$(7):GOTO suite
)"                                           UNTIL A$=CHR$(32)
220 A$=INPUT$(1):IF A$="O" OR A$="o" THE   IF U<>X THEN PRINT "C'est raté....";CHR
N 40                                         $(7):GOTO suite
                                           PRINT:PRINT COLOR(14) POINTS(20) "BRAVO"
                                           LABEL suite
                                           PRINT:PRINT "Autre essai ( O/N ) ?"
                                           R$=INPUT$(1):R$=UPPER$(R$):IF R$="O" THEN
                                           RUN ELSE END

```

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 GW-BASIC :

La ligne 40 permet de dimensionner la variable A\$(I) qui contiendra les mots.

La boucle I (50) saisit la liste des mots (inscrits en DATA aux lignes 280 à 320).

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

La ligne 160 affiche les lettres, et les lignes 180 à 210 attendent votre proposition, qui est testée en ligne 220.

Le nombre d'essais est incrémenté en ligne 230.

La fin de partie est contrôlée par la ligne 240, et le score est affiché en ligne 260.

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 (le temps est contrôlé).

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

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

EXTENSIONS POSSIBLES :

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

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM MOTS
20 SCREEN 0,1:WIDTH 40:COLOR 0,7
30 RANDOMIZE TIMER:KEY OFF
40 DIM A$(30):S=0:E=0:ON TIMER(10) GOSUB
  330
50 FOR I=1 TO 30:READ A$(I):NEXT I
60 CLS
70 Z=INT(RND*30+1)
80 IF A$(Z)="" THEN 70
90 R$=A$(Z):A$="":A$(Z)="":V$=R$
100 FOR I=1 TO LEN(R$)
110 Z=INT(RND*LEN(R$)+1)
120 IF MID$(R$,Z,1)="-" THEN 110
130 A$=A$+MID$(R$,Z,1)
140 MID$(R$,Z,1)="-"
150 NEXT I
160 COLOR 0:PRINT:PRINT "LES LETTRES SON
T :":COLOR 4:PRINT A$:PLAY"CEG":COLOR 0
170 TIMER ON:LOCATE 7,1:PRINT"VOTRE SOLU
TION :":S$="":V=17:COLOR 1
180 R$=INKEY$:IF R$="" THEN 180 ELSE IF
R$=CHR$(13) THEN TIMER OFF:GOTO 220
190 IF R$=CHR$(8) THEN R$="":S$=LEFT$(S$
,LEN(S$)+(LEN(S$)<>0))
200 IF R$>"a" THEN R$=CHR$(ASC(R$)-32)
210 S$=S$+R$:LOCATE 7,17:PRINT S$+" ":GO
TO 180
220 IF S$=V$ THEN S=S+1
230 E=E+1:IF S$<>V$ THEN PRINT:PRINT "LE
MOT ETAIT ":COLOR 5:PRINT V$
240 COLOR 0:LOCATE 15,1:PRINT"AUTRE ESSA
I (0/N) :":R$=INPUT$(1)
250 IF R$="0" OR R$="o" THEN 60 ELSE IF
R$<>"N" AND R$<>"n" THEN 240
260 COLOR 9:LOCATE 18,1:PRINT"VOTRE SCOR
E :":S; " SUR ";E; " ESSAIS"
270 PLAY"CGE":END
280 DATA JAMBON,ENERGUMENE,AFFILIE,ERUDI
T,INSIGNIFIANT,REALISTE
290 DATA IMPRESSIONNANT,ALLECHANT,EPIQUE
,INTREPIDE,COUARD,ORDINATEUR
300 DATA TREVE,BANDEROLLE,SENAT,ACTUEL,C
ONSACRE,DIEDRE
310 DATA LIVRE,TERMINE,SIGNATURE,MALODOR
ANT,AIMANT,ARBITRAIRE
320 DATA CHALEUR,OPPORTUN,PELURE,ALTRUIS
TE,RAMONEUR,NOVICE
330 TIMER OFF:CLS
340 LOCATE 11,1:PRINT"DESOLE, C'EST UN P
EU TARD"
350 GOTO 230

```

```

REM Mots
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "MOTS"
WINDOW OPEN
RANDOMIZE TIME:DIM a$(30):s=0:g=0:f$=""
FOR i=1 TO 30:READ a$(i):NEXT i
LABEL debut
CLS
REPEAT:z=RND(30):UNTIL a$(z)<>""
r$=a$(z):a$="":a$(z)="":v$=r$
FOR i=1 TO LEN(r$)
LABEL tirage
z=RND(LEN(r$)):IF MID$(r$,z,1)="-" GOTO t
irage
a$=a$+MID$(r$,z,1):MID$(r$,z,1)="-"
NEXT i
p$="LES LETTRES SONT":t=20:h=2:u=2:v=1:c=
4:m=1:GOSUB affiche
p=LEN(a$)*18+20:v=20-(LEN(p$)/2):BOX 8*v-
10;145,p,32 COLOR(6) FILL WITH 8
p$=a$:t=16:h=5:u=2:c=2:m=2:GOSUB affiche
INPUT AT (1;10) "Votre solution ( en maju
scules ) : ",s$:s$=UPPER$(s$)
IF s$=v$ THEN p$="BRAVO":t=20:h=15:u=2:c=
14:v=1:m=1:GOSUB affiche:s=s+1:PRINT CHR$
(7)
g=g+1:IF s$<>v$ THEN p$="C'ETAIT "+v$:t=1
6:h=15:u=2:v=1:c=11:m=1:GOSUB affiche:f$=
f$+v$+" "
p$="AUTRE ESSAI ( 0/N )":t=16:h=17:u=2:v=
1:c=12:m=1:GOSUB affiche
r$=INPUT$(1):r$=UPPER$(r$):IF r$="0" GOTO
debut
PRINT AT (1;19) "Votre score : " s " sur
" g " essais"
r$=INPUT$(1):END
LABEL affiche
FOR i=1 TO LEN(p$)
PRINT AT (u*i+v;h) COLOR(c) POINTS(t) MOD
E(m) MID$(p$,i,1)
NEXT i
RETURN
DATA JAMBON,ENERGUMENE,AFFILIE,ERUDIT,INS
IGNIFIANT,REALISTE
DATA IMPRESSIONNANT,ALLECHANT,EPIQUE,INTR
EPIDE,COUARD,ORDINATEUR
DATA TREVE,BANDEROLLE,SENAT,ACTUEL,CONSA
CRE,DIEDRE,LIVRE
DATA TERMINE,SIGNATURE,MALODORANT,AIMANT,
ARBITRAIRE
DATA CHALEUR,OPPORTUN,PELURE,ALTRUISTE,RA
MONEUR,NOVICE

```

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 significatif.

Le rôle du singe est, ici, tenu par le PC qui va afficher des groupes de 3 à 6 lettres pendant un temps relativement court.

Si l'un des joueurs estime qu'il peut réussir à faire un mot avec les lettres proposées, il doit taper rapidement sur sa touche de jeu (1 ou 0).

Le mot devra alors être accepté ou refusé par l'adversaire qui répondra JUSTE ou FAUX.

Chaque mot juste rapporte 1 point, chaque mot faux enlève 1 point.

LE PROGRAMME GW-BASIC :

La ligne 50 contient une liste VO\$ des voyelles courantes, et une liste CO\$ 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 VO\$ ou deux R dans CO\$, etc.).

La ligne 60 définit la longueur du mot (en principe, de 3 à 6 lettres, mais on peut également modifier cette longueur).

La boucle I (80-130) permet la confection d'un "mot" en prenant une lettre, soit dans l'alphabet complet, soit dans la liste CO\$ (ligne 100), soit dans la liste VO\$ (ligne 110).

La ligne 180 temporise l'apparition du résultat.

La ligne 190 gère la fin de partie, avec affichage du score.

La ligne 200 regarde si l'un des joueurs a réagi.

Enfin, les lignes 230 et 240 comptabilisent les points de chaque joueur.

POUR JOUER :

On lance le programme en tapant RUN.

La première série de lettres est alors affichée sur l'écran. Le joueur de gauche devra taper 1 s'il voit qu'il peut réaliser un mot, celui de droite devra taper 0.

Si rien n'a été frappé, une autre série de lettres sera affichée après peu de temps.

Si l'un des joueurs a frappé 0 ou 1, la question (J)USTE ou (F)AUX est alors affichée. On répond J si le mot est correct, F si le mot est incorrect, suivi de RETURN.

Les points sont alors calculés par PC, et la série de lettres suivante est alors affichée.

Pour sortir du programme, il faudra taper P au lieu de 0 ou 1 : le score final est alors affiché.

EXTENSIONS POSSIBLES :

On pourra modifier la longueur des mots, la fréquence d'apparition des lettres, le décompte des points, la règle du jeu, etc.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic, mais chaque joueur dispose d'une fenêtre, et devra taper le mot qu'il a découvert. L'autre joueur confirme en tapant J ou F, selon le cas. On termine en tapant la lettre P, lors d'un affichage.

```
10 REM LE SINGE
20 SCREEN 0,1:WIDTH 40:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF
40 A$=CHR$(200):B$=CHR$(201):C$=CHR$(205)
   D$=CHR$(186):E$=CHR$(187):F$=CHR$(188)
50 A=0:B=0:V0$="AEIOU":C0$="BCDGLMNPRSTV"
   G$=STRING$(40,C$)
60 X=INT(RND*5+3):M$="":ON TIMER(2+(X/3)
   ) GOSUB 250
70 CLS
80 FOR I=1 TO X
90 U$=CHR$(INT(RND*26+65))
100 IF RND<.4 THEN U$=MID$(C0$,INT(LEN(C
   0$)*RND+1),1)
110 IF RND<.4 THEN U$=MID$(V0$,INT(LEN(V
   0$)*RND+1),1)
120 M$=M$+U$
130 NEXT I
140 P=LEN(M$):LOCATE 1,18-P:PRINT B$+LEF
   T$(G$,2*P+1)+E$
150 LOCATE 2,18-P:PRINT D$:LOCATE 2,20+P
   :PRINT D$
160 LOCATE 3,18-P:PRINT A$+LEFT$(G$,2*P+
   1)+F$
170 LOCATE 2,20-P:FOR I=1 TO P:COLOR INT
   (RND*15+1):PRINT MID$(M$,I,1)+" ";:SOUND
   INT(RND*4000+500),1:NEXT I
180 S$=INKEY$:TIMER ON:IF S$="" THEN 180
190 IF S$="P" OR S$="p" THEN LOCATE 12,1
   :PRINT"SCORE DE A :";A;" ; SCORE DE B :";
   B:PLAY"CEG":END
200 IF S$<>"0" AND S$<>"1" AND S$<>"&" A
   ND S$<>"à" THEN 60
210 COLOR 7:LOCATE 6,1:PRINT"(J)USTE OU
   (F)AUX :":R$=INPUT$(1)
220 IF R$>="a" THEN R$=CHR$(ASC(R$)-32)
230 A=A-(R$="J" AND (S$="0" OR S$="à"))
   +(R$="F" AND (S$="0" OR S$="à"))
240 B=B-(R$="J" AND (S$="1" OR S$="&"))
   +(R$="F" AND (S$="1" OR S$="&"))
250 TIMER OFF:GOTO 60
```


REM Le singe

```
FOR i=1 TO 4:CLOSE WINDOW i:CLOSE #i:NEXT
i
FOR i=1 TO 4:OPEN #i WINDOW i:NEXT i
STREAM #1:SCREEN GRAPHICS
WINDOW #2 SIZE 37,10:WINDOW #4 SIZE 37,10
WINDOW SIZE XUSABLE,YUSABLE/3:WINDOW TITL
E "LE SINGE"
WINDOW PLACE 0,120:WINDOW #2 PLACE 0,10:W
INDOW #2 TITLE "JOUEUR 1"
WINDOW #4 PLACE 320,10:WINDOW #4 TITLE "J
OUEUR 2"
WINDOW #2 OPEN:WINDOW #4 OPEN:WINDOW OPEN
a=0:b=0:v$="AEIOU":c$="BCDGLMNPSTV":w=0
LABEL début
CLS:CLS #2:CLS #4:x=INT(RND*4+3):m$=""
FOR i=1 TO x
e$=CHR$(INT(RND*26+65))
IF RND<.4 THEN e$=MID$(c$,INT(LEN(c$)*RND
+1),1)
IF RND<.5 THEN e$=MID$(v$,INT(LEN(v$)*RND
+1),1)
m$=m$+e$
NEXT i
f=1:co=INT(RND*7+8):t=20:p$m$h=4:k=3:GO
SUB aff:PRINT CHR$(7)
i=0:REPEAT:r=INKEY:i=i+1:UNTIL r<>-1 OR i
=600
IF r=80 OR r=112 THEN w=1:GOTO fin
IF (r<>38 AND r<>49) AND (r<>133 AND r<>4
8) GOTO début
v=- (r=38 OR r=49)-2*(r=133 OR r=48)
p$="JOUEUR"+STR$(v):co=2*v:f=2*v:t=20:h=2
:k=3:GOSUB aff
p$="VOTRE MOT":co=1:f=2*v:t=10:h=4:k=3:GO
SUB aff
s$="":REPEAT:r$=INKEY$:s$=s$+r$:PRINT #f,
AT (2;6) UPPER$(s$):UNTIL r$=CHR$(13)
p$="(J)uste-(F)aux":co=6-2*v:f=6-2*v:t=8:
h=6:k=2:GOSUB aff
REPEAT:r$=INKEY$:UNTIL r$<>"":r$=UPPER$(r
$)
a=a-(r$="J" AND v=1)+(r$="F" AND v=1)
b=b-(r$="J" AND v=2)+(r$="F" AND v=2)
LABEL fin
PRINT #2,AT (1;8) "Score actuel " a
PRINT #4,AT (1;8) "Score actuel " b
FOR i=1 TO 2000:NEXT i
IF w<>1 GOTO début
r$=INPUT$(1):END
LABEL aff
FOR i=1 TO LEN(p$)
PRINT #f,AT (k*i;h) COLOR(co) POINTS(t) M
ID$(p$,i,1)
NEXT i
RETURN
```


Niveau 5

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Course de chevaux	Assez visuel : un jeu de hasard, pour plusieurs joueurs.
Chapeaux	Soyez très observateur et vous réussirez, pour deux joueurs.
Plusieurs en un	Un terrain de jeu et les pions, pour deux joueurs.
Œufs	Ne laissez pas faire la pondeuse diabolique (un joueur).
Combat aérien	Pas si facile de descendre un appareil qui bouge tout le temps.
Affichage publicitaire	Si vous ne savez pas quoi mettre dans votre vitrine.
Captures dans l'espace	Votre mission serait facile, sans ces damnées météorites.
Cibles	Un peu d'adresse, beaucoup de chance : très visuel.
Sous-marin	Un classique des jeux vidéo, pour un joueur.
Le jeu du missile	Une mission bien périlleuse, pour un seul joueur.
Obstacles	Beaucoup d'obstacles à éviter dans ce jeu d'adresse pour un joueur.
Course de voitures	Un jeu de réflexes pas trop difficile, pour un joueur.
Evasion	Il vous faudra beaucoup de chance pour réussir.
Mathieu	Vous ne savez pas dessiner ? Laissez faire votre ordinateur.
Ardoise magique	Mettez au point vos créations graphiques, sans papier ni crayon.
Cube	Jeu de dessin ; perspective sur un écran.
Chasse au sous-marin	Il vous faudra pas mal de réflexion, et beaucoup de chance.
Dames	Pour deux joueurs : classique.
Dessin de dés	Améliorer vos jeux de dés en incorporant ce programme.
Echecs	Un programme assez simplifié, pour deux joueurs.
Fais-moi un dessin	Composez vous-même votre paysage avec des images déjà définies.

Von Koch

Le lézard d'Alice

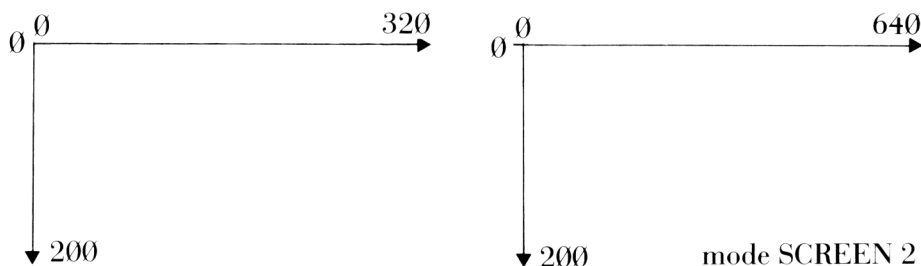
Une intéressante façon de déformer des figures.

Beaucoup de hasard, mais très spectaculaire.

Techniques particulières au PC

GRAPHIQUES HAUTE RESOLUTION

Le mode SCREEN 0, que nous avons utilisé jusqu'ici, est un mode texte. Il peut utiliser 40 ou 80 colonnes, grâce à l'instruction WIDTH. Pour les graphiques, on peut travailler en 40 colonnes avec SCREEN 1 (on dispose alors de 4 couleurs) ou en 80 colonnes avec SCREEN 2 (il n'y a plus que 2 couleurs). Les instructions graphiques sont alors accessibles, dans ces deux modes, si on dispose d'un écran graphique.



L'instruction PSET (x, y), C permet d'allumer un point en colonne x, ligne y sur l'écran. Le paramètre C, facultatif, indique la couleur. On peut effacer le point avec PRESET (x, y).

L'instruction LINE va permettre soit de tirer un trait entre deux points désignés par leurs coordonnées, dans la couleur précisée, soit, en ajoutant un paramètre B, de tracer un rectangle, et même un rectangle plein, en précisant BF.

L'instruction CIRCLE, très puissante ici, permet de tracer des cercles de centre et de rayon désigné, mais on peut aussi lui faire tracer des arcs de cercle, ou des ovales...

L'instruction PAINT permet de colorier l'intérieur d'une figure fermée.

L'instruction DRAW est une véritable macro-instruction graphique ; elle est en effet suivie d'une chaîne de caractères qui vont désigner l'orientation de la figure, son échelle, sa couleur, la position de départ du tracé, et le tracé lui-même.

On peut également reconnaître la couleur d'un point sur l'écran à l'aide de l'instruction POINT, utilisable dans les deux modes haute résolution.

ACCES DIRECT A LA MEMOIRE PEEK ET POKE

Il est possible, grâce à PEEK, de lire directement le contenu de n'importe quelle case de la mémoire de l'ordinateur (de 0 à 65535).

L'instruction POKE permet à l'inverse d'inscrire des valeurs dans les cases de la mémoire. Elle est à utiliser avec prudence, les résultats obtenus pouvant parfois être tout à fait étranges.

L'utilisation combinée de ces deux instructions permet si besoin est de remédier à certaines insuffisances du Basic ; dans cet ouvrage elles ont été utilisées pour définir des caractères graphiques (de code supérieur ou égal à 128).

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 9, et la course débutera immédiatement.

Il pourra y avoir des ex aequo, car l'on tiendra compte du fait que le PC 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 GW-BASIC :

Une première boucle non formelle (110-200) affiche les chevaux dans leur position de départ.

Une deuxième boucle I (130-210) les fera avancer ; la position de chaque cheval sera repérée par un chiffre ou une lettre dans une variable alpha-numérique A\$.

La ligne 150 fera avancer les chevaux.

La ligne 170 gère les arrivées.

La ligne 180 réalisera les impressions.

Le sous-programme 270-310 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 9, suivi de RETURN.

La course se déroulera ensuite automatiquement.

EXTENSIONS POSSIBLES :

On pourra dessiner les chevaux avec un graphisme plus soigné, et les différencier par des numéros.

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

Chapeaux

REFERENCES :

Difficulté de programmation : assez facile.

Catégorie de jeu : jeu de réflexion, 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 GW-BASIC :

La ligne 40 contient le dessin des deux chapeaux (dans deux variables alphanumériques B\$ et C\$).

La ligne 50 contient, en DATA, les symboles qui permettront de dessiner les personnages.

La ligne 70 permet, s'il y a lieu, l'échange des chapeaux.

La variable T de la ligne 80 sert à repérer s'il y a eu échange ou non.

Le dessin des deux personnages est réalisé par une boucle I (120-150), et leur déplacement est réalisé par la ligne 160.

Le test, permettant ou non le changement des couvre-chefs, est effectué en ligne 180.

Les lignes 220 à 250 déterminent si le premier joueur ayant appuyé sur une touche a gagné ou perdu.

POUR JOUER :

Lancer le programme par RUN.

Les personnages commencent immédiatement à se déplacer l'un vers l'autre.

Le joueur de gauche dispose des touches 1 et 2 et celui de droite, des touches 9 et 0.

S'il y a eu changement de chapeau, les touches à appuyer seront 1 (à gauche) et 0 (à droite).

S'il n'y a pas eu de changement, il faudra, pour gagner, être le premier à appuyer sur 2 (à gauche) et sur 9 (à droite).

L'affichage indiquera alors BRAVO GAUCHE ou BRAVO DROITE, ou, encore, PERDU GAUCHE ou PERDU DROITE.

EXTENSIONS POSSIBLES :

Il sera possible de prévoir un affichage couleur, avec une animation plus complexe 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).

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM CHAPEAUX
20 SCREEN 0,1:WIDTH 80:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF
40 B$=CHR$(178)+CHR$(179)+CHR$(178):C$=C
HR$(179)+CHR$(178)+CHR$(179)
50 A$=" "+CHR$(1)+" "+CHR$(186)+" "+CHR
$(220)+CHR$(219)+CHR$(220)+" "+CHR$(219)
+" "+CHR$(178)+CHR$(178)+CHR$(178)+CHR$(
178)+CHR$(178)+CHR$(178)+CHR$(186)+" "+C
HR$(186)+CHR$(186)+" "+CHR$(186)+CHR$(18
9)+" "+CHR$(211)
60 X=3:Y=76:T=1:K=1
70 SWAP B$,C$
80 T=ABS(T-1)
90 CLS
100 COLOR 3:LOCATE 2,X:PRINT C$
110 COLOR 2:LOCATE 2,Y:PRINT B$
120 FOR I=1 TO 9
130 COLOR 1:LOCATE I+2,X,0:PRINT MID$(A$
,3*I-2,3)
140 COLOR 4:LOCATE I+2,Y,0:PRINT MID$(A$
,3*I-2,3)
150 NEXT I
160 X=X+K:Y=Y-K:IF X>72 OR X<4 THEN K=-K
170 FOR I=1 TO 600:NEXT I:SOUND INT(RND*
4000+500),1
180 IF (RND<.5) AND ABS(X-Y)<3 THEN 70
190 R$=INKEY$:IF R$="" THEN 90
200 IF X<Y THEN 240
210 COLOR 5
220 IF (R$="1" AND T=1) OR (R$="2" AND T
=0) THEN PRINT "BRAVO GAUCHE":PLAY"CGE":
END
230 IF (R$="0" AND T=1) OR (R$="9" AND T
=0) THEN PRINT "BRAVO DROIT":PLAY"CDE":E
ND
240 IF R$="2" OR R$="1" THEN PRINT "PERD
U GAUCHE"
250 IF R$="9" OR R$="0" THEN PRINT "PERD
U DROIT"
260 PLAY"CCCCCCCC"

```

```

REM Chapeaux
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "CHAPEAUX"
WINDOW OPEN
RANDOMIZE TIME:t=1:e=10:f=12:k=1:DIM u(26
),v(26)
FOR i=1 TO 25 STEP 2:READ u(i),v(i),u(i+1
),v(i+1):NEXT i
FOR i=1 TO 5:READ c(i),d(i):NEXT i
x=10:b=90:u(0)=0:v(0)=0:u(26)=0:v(26)=0
IF RND<.5 THEN SWAP e,f:t=ABS(t-1)
LABEL dessin
CLS:a=620-x
FOR i=1 TO 26
LINE x+u(i-1);b+v(i-1),x+u(i);b+v(i):LINE
a+u(i-1);b+v(i-1),a+u(i);b+v(i)
NEXT i
SHAPE x+c(1);b+d(1),x+c(2);b+d(2),x+c(3);
b+d(3),x+c(4);b+d(4),x+c(5);b+d(5)COLOR(e
) FILL WITH e
SHAPE a+c(1);b+d(1),a+c(2);b+d(2),a+c(3);
b+d(3),a+c(4);b+d(4),a+c(5);b+d(5)COLOR(f
) MODE(2) FILL WITH f
x=x+1:12:IF x>600 THEN k=-k
PRINT CHR$(7);i=0:REPEAT:i=i+1:a$=INKEY$:
UNTIL a$<>"" OR i=300
IF ABS(x-y)<10 AND RND<.5 THEN t=ABS(t-1)
:SWAP e,f
IF a$<>"1" AND a$<>"2" AND a$<>"9" AND a$
<>"0" AND a$<>" " GOTO dessin
IF x<320 GOTO perdu
IF (a$="1" AND t=1) OR (a$="2" AND t=0) T
HEN PRINT AT (1;15) "BRAVO GAUCHE":GOTO f
in
IF (a$="0" AND t=1) OR (a$="9" AND t=0) T
HEN PRINT AT (1;15) "BRAVO DROIT":GOTO fi
n
LABEL perdu
IF a$="1" OR a$="2" THEN PRINT AT (1;15)
"PERDU GAUCHE"
IF a$="9" OR a$="0" THEN PRINT AT (1;15)
"PERDU DROIT"
LABEL fin
PRINT AT (1;17) "C'est fini...."
r$=INPUT$(1):END

```

```

DATA 0,0,0,7,7,7,13,55,10,65,3,40,0,40,10
,70,13,70,12,77,15,85
DATA 18,77,17,70,20,70,30,40,27,40,20,65,
17,55,23,7,30,7
DATA 30,0,20,0,17,40,13,40,10,0,0,0
DATA 0,85,5,100,25,100,30,85,15,85

```

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 de couleur, 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 GW-BASIC :

Le changement de joueur est effectué par la ligne 40.

Le joueur qui a le trait voit le symbole qu'il a choisi (x ou o) se placer en limite du cadre (ligne 60).

Les lignes 90 et 100 permettent aux deux joueurs de déplacer le symbole sur le terrain.

Les lignes 120 et 130 contrôlent si le déplacement est possible (case libre et non dépassement des limites).

La ligne 150 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 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 barre d'espace pour inscrire son propre symbole dans une case disponible.

Dès que cela a été fait, le symbole du second joueur apparaît dans la colonne de gauche, à peu près au milieu, et le joueur procédera de la même manière que son adversaire.

Pour sortir du programme, il suffira de taper CTRL et Break.

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

Le joueur qui a le trait est indiqué dans la barre de titre, en haut de l'écran.

La case où il veut jouer est alors désignée en cliquant le bouton de gauche

de la souris. Un clic sur le bouton de droite permet le changement de joueur, et un double clic (les deux boutons en même temps) provoque l'arrêt du programme.

```

10 REM PLUSIEURS EN UN
20 SCREEN 0,1:WIDTH 80:COLOR 7,0:CLS
30 RANDOMIZE TIMER:KEY OFF:J=INT(RND*2)
40 J=ABS(J-1)
50 X=12:Y=1
60 LOCATE X,Y,0:COLOR 1-3*(J=1):PRINT CH
R$(79-9*(J=1));
70 A$=INKEY$:A$=A$+A$:A$=MID$(A$,2,1)
80 A=X:B=Y
90 X=X-((A$="2" OR A$="P" ) AND X<22)+((
A$="8" OR A$="H") AND X>2)
100 Y=Y-((A$="6" OR A$="M") AND Y<78)+((
A$="4" OR A$="K") AND Y>2)
110 IF A$=" " AND Y<>1 THEN 150
120 C=SCREEN(X,Y):IF C<>32 AND A$<>" " TH
EN SOUND 2500,3
130 IF C<>32 THEN X=A:Y=B:GOTO 60 ELSE L
OCATE A,B,0:PRINT " "
140 GOTO 60
150 LOCATE X,Y,0:COLOR 1-3*(J=1):PRINT C
HR$(79-9*(J=1));
160 IF J=0 THEN PLAY"A" ELSE PLAY"G"
170 GOTO 40

REM Plusieurs en un
STREAM #1:SCREEN GRAPHICS 640 FIXED, 200
FIXED
USER SPACE 640, 200
WINDOW FULL ON
WINDOW OPEN:RANDOMIZE TIME:DIM t(25,10)
BOX 0;0,640,200 COLOR(11) WIDTH(28)
FOR i=1 TO 25
FOR j=1 TO 10
BOX 24*i-5;18*j-10,24,18
NEXT j,i
h=RND(2):WINDOW MOUSE 7
LABEL début
WINDOW TITLE "JOUEUR : "+CHR$(88+9*(h=1))
REPEAT
a=BUTTON(1):b=BUTTON(2):x=XMOUSE+3:y=184-
YMOUSE:x=3*INT(x/24)+1:y=2*INT(y/16)+1
IF b<>-1 AND a=-1 THEN h=ABS(3-h):PRINT C
HR$(7);:GOTO début
IF a<>-1 AND b=-1 AND t((x-1)/3,(y-1)/2)=
0 THEN PRINT AT (x;y) COLOR(2*h) CHR$(88+
9*(h=1));:t((x-1)/3,(y-1)/2)=h:h=ABS(3-h)
:PRINT CHR$(7);:GOTO début
UNTIL a+b=2
PRINT AT (15;1) "C'est fini..... ":WINDOW
MOUSE 0
r$=INPUT$(1):END

```

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 GW-BASIC :

L'oiseau est dessiné en deux parties (pour le mouvement) aux lignes 70 et 110 ; son déplacement aléatoire est géré par le sous-programme 230-320 (en particulier les lignes 280 à 310).

La ligne 150 permet de reboucler le programme sur la ligne 70.

Le déplacement du joueur est régi par les lignes 170 et 180, et le test permettant de détecter la présence d'un œuf est effectué en ligne 190.

La ligne 210 permet l'affichage du score.

La fin de jeu est détectée en ligne 250.

POUR JOUER :

On lancera le programme par RUN.

L'oiseau commence alors à se déplacer au hasard sur l'écran, et se met à pondre ses œufs. Lorsque le symbole qui représente le joueur (un carré de couleur) apparaît sur l'écran, celui-ci peut se déplacer en utilisant les 4 touches de déplacement. Il vous faut dès lors essayer de récupérer un maximum d'œufs avant la fin de la partie.

On pourra arrêter de jouer à tout moment en tapant CTRL et Break.

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM OEUFs
20 SCREEN 0,1:WIDTH 80:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF
40 A$(1)=CHR$(127):A$(2)=CHR$(170):A$(3)=CHR$(169):A$(4)=CHR$(95)
50 CLS:P=400:R=36:S=10:T=0
60 X=INT(RND*77):Y=INT(RND*21)
70 LOCATE Y+1,X+1,0:COLOR 6:PRINT A$(2)+A$(1)+A$(3)
80 FOR I=1 TO 50 : A$=INKEY$:A$=A$+A$:A$=MID$(A$,2,1)
90 IF A$="" THEN NEXT I ELSE GOSUB 160
100 GOSUB 230
110 LOCATE Y+1,X+1,0:COLOR 6:PRINT A$(4)+A$(1)+A$(3)
120 FOR I=1 TO 50 : A$=INKEY$:A$=A$+A$:A$=MID$(A$,2,1)
130 IF A$="" THEN NEXT I ELSE GOSUB 160
140 GOSUB 230
150 GOTO 70
160 LOCATE S+1,R+1:COLOR 0:PRINT " "
170 R=R-((A$="6" OR A$="M") AND R<77) + ((A$="4" OR A$="K") AND R>0)
180 S=S-((A$="2" OR A$="P") AND S<22) + ((A$="8" OR A$="H") AND S>0)
190 G=SCREEN(S+1,R+1):IF G=79 THEN FOR I=1 TO 6:SOUND 500+200*I,.2:NEXT
200 T=T - (G=79)
210 LOCATE 23,16:COLOR 2:PRINT"SCORE :";T
220 RETURN
230 A=X:B=Y
240 P=P-1
250 IF P=0 THEN PLAY"CDE":PRINT"C'EST FINI...":END
260 LOCATE S+1,R+1:COLOR 3:PRINT CHR$(1)
270 LOCATE B+1,A+1:COLOR 4:PRINT " "+"0"
+" "
280 X=ABS(X+INT(1+2*(RND<.5)))
290 Y=ABS(Y+INT(1+2*(RND<.5)))
300 IF X>77 THEN X=1
310 IF Y>22 THEN Y=1
320 RETURN

```

```

REM Deufs
STREAM #1:SCREEN GRAPHICS 640 FIXED, 200
FIXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "OEUFs"
WINDOW OPEN
RANDOMIZE TIME:p=400:u=INT(RND*30+5):v=INT(RND*13+6):DIM t(80,22)
a$=CHR$(27)+CHR$(5):b$=CHR$(92)+CHR$(234)+CHR$(47):c$=CHR$(95)+CHR$(234)+CHR$(95):
xo=37:yo=10
LABEL boucle
PRINT AT (u;v) COLOR(4) c$ AT (xo+1;yo+1)
COLOR(7) a$:a=xo+1:b=yo+1:r=INKEY
xo=xo+((r=331 OR r=52) AND xo>1)-((r=333 OR r=54) AND xo<78)
yo=yo+((r=328 OR r=56) AND yo>1)-((r=336 OR r=50) AND yo<20)
IF t(xo,yo)=1 THEN t=t+1:t(xo,yo)=0:PRINT CHR$(7) AT (xo+1;yo+1) CHR$(32) AT (1;20)
) "Score : " t;
PRINT AT (a;b) " ":GOSUB premd:PRINT AT (u;v) COLOR(4) b$:GOSUB premd:GOTO boucle
LABEL fin
r$=INPUT$(1):END
LABEL premd
a=u:b=v:p=p-1:IF p=0 GOTO fin
PRINT AT (u;v) " " AT (a+1;b) COLOR(2)
CHR$(42):t(a,b-1)=1
u=ABS(u+INT(1+2*(RND<.5))):v=ABS(v+INT(1+2*(RND<.5)))
u=u*(u<79)*(u>1)-(u>78)-78*(u<2):v=v*(v<20)*(v>4)-5*(v>19)-19*(v<5)
RETURN

```

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, S, ou la barre d'espace, et vous voyez vos projectiles qui convergent sur lui. Mais il continue à se déplacer, même pendant votre tir, et il n'est pas du tout évident que vous réussirez à l'avoir. La partie continuera à se dérouler jusqu'à ce que vous l'ayez touché (affichage de GAGNE).

LE PROGRAMME GW-BASIC :

La ligne 70 dessine l'appareil ennemi dont le déplacement aléatoire sera réalisé par les lignes 90 et 100.

Les lignes 110 à 130 vous permettent le pilotage de votre appareil.

La ligne 170 dessine le viseur de votre appareil.

Le sous-programme, gérant le tir, va de la ligne 200 à la ligne 230 et le test de fin de partie est effectué en ligne 240.

La ligne 180 regarde si vous avez tiré ou non.

POUR JOUER :

Dès que le programme a été lancé par l'ordre RUN, vous devez manœuvrer pour essayer de faire en sorte que l'appareil ennemi soit cadré dans votre viseur.

Pour cela, vous disposez des flèches de déplacement (touche ↑, ↓, →, ←) et vous pouvez tirer à tout moment en pressant la touche Ø, S, ou la barre d'espace.

L'appareil ennemi peut continuer à se déplacer de manière aléatoire, même lorsque vous avez tiré.

La partie se terminera avec l'affichage de GAGNE, si vous touchez l'avion ennemi.

On peut aussi arrêter à tout moment en frappant CTRL et Break.

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.

On peut aussi introduire plusieurs avions, compter les points ou le temps, tenir compte de sa propre réserve de carburant, etc.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic, mais le tir n'est possible qu'avec la barre d'espace.

```
10 REM COMBAT AERIEN
20 SCREEN 0,1:WIDTH 80:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF
40 CLS
50 A=5:B=36
60 N=21
70 LOCATE A,B,0:COLOR 4:PRINT CHR$(170)+
CHR$(249)+CHR$(169)
80 U=A:V=B
90 A=A+2*INT(RND*2)-1
100 B=B+2*INT(RND*2)-1
110 A$=INKEY$:A$=A$+A$:A$=MID$(A$,2,1)
120 B=B-2*(A$="M" OR A$="6")+2*(A$="K" O
R A$="4")
130 A=A-2*(A$="P" OR A$="2")+2*(A$="H" O
R A$="8")
140 IF B<1 OR B>80 THEN B=ABS(B-80)
150 IF A<3 OR A>22 THEN A=ABS(A-22)
160 LOCATE U,V:PRINT " "
170 LOCATE 10,38:COLOR 2:PRINT "* *":LOC
ATE 11,38:PRINT " X ":LOCATE 12,38:PRINT "
* *"
180 IF A$=" " OR A$="5" OR A$="0" OR N<2
1 THEN GOSUB 200
190 GOTO 70
200 COLOR 1:LOCATE N,49-N,0:PRINT "*":LO
CATE N,N+29:PRINT "*"
210 LOCATE N,49-N,0:PRINT " ":LOCATE N,N+
29:PRINT " "
220 N=N-1:SOUND 2000,1
230 IF N>9 THEN RETURN
240 IF (A=10 OR A=11 OR A=12) AND (B=38
OR B=39 OR B=40) THEN PRINT"GAGNE...":PL
AY"CDE":END
250 GOTO 60

REM Combat aérien
STREAM #1:SCREEN GRAPHICS 640 FIXED, 200
FIXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "COMBAT AERIEN"
WINDOW OPEN
RANDOMIZE TIME
a=5:b=15
LABEL entrée
n=21
LABEL début
PRINT AT (b;a) COLOR(2) CHR$(170)+CHR$(25
4)+CHR$(169)
u=a:v=b
a=a+2*INT(RND*2)-1:b=b+2*INT(RND*2)-1
t=INKEY
b=b-2*(t=333 OR t=54)+2*(t=331 OR t=52)
a=a+2*(t=328 OR t=56)-2*(t=336 OR t=50)
IF b<1 OR b>76 THEN b=ABS(b-76)
IF a<3 OR a>21 THEN a=ABS(a-21)
PRINT AT (v;u) " "
PRINT AT (33;9) "* *" AT (33;10) " x " AT
(33;11) "* *"
IF t=32 OR n<21 THEN GOSUB tir
GOTO début
LABEL fin
PRINT AT (2;2) "Bravo; c'est gagné....."
r$=INPUT$(1):GOTO fin

LABEL tir
PRINT AT (44-n;n) COLOR(4) "*" AT (n+24;n
) "*"
PRINT AT (44-n;n) " " AT (n+24;n) " "
n=n-1:IF n>9 THEN RETURN
IF (a=9 OR a=10 OR a=11) AND (b=33 OR b=3
4 OR b=35) GOTO fin
GOTO entrée
```


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 (80 caractères) ; 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 GW-BASIC :

Le texte sera introduit lettre par lettre aux lignes 60 à 90.

L'affichage sera fait à l'intérieur d'un cadre (lignes 150 à 170).

Le défilement vers la gauche est préparé à la ligne 190, et le défilement vers la droite à la ligne 200.

Le texte sera affiché en ligne 180, et la durée de l'affichage (vitesse de défilement) est choisie par la ligne 210.

La ligne 230 permettra de changer, si besoin est, les conditions du défilement.

POUR JOUER :

On lancera le programme par RUN, et à l'affichage de la question ENTREZ VOTRE TEXTE, on introduira le texte à faire défiler, suivi de RETURN.

On obtient ensuite l'affichage DEFILEMENT A DROITE (D) OU A GAUCHE (G) auquel on répondra en tapant, soit D, soit G.

Puis à la question VITESSE, on répondra en introduisant un grand nombre pour une vitesse rapide, et un petit pour une vitesse lente (par exemple, de 0 à 40 ou plus).

Le texte commencera à défiler dès que l'on aura frappé RETURN.

Si l'on désire changer les conditions du défilement, il suffit d'appuyer sur la touche 0 pour retrouver les questions SENS et VITESSE. On tapera CTRL et Break pour interrompre le programme.

EXTENSIONS POSSIBLES :

On pourra introduire de la couleur, réaliser un affichage ondulant, faire remonter les lignes sur l'écran, etc.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM AFFICHAGE PUBLICITAIRE
20 SCREEN 0,1:WIDTH 80:COLOR 0,7:KEY OFF
:CLS
30 B$="":C$=CHR$(200):D$=CHR$(201):E$=CHR$(205):F$=CHR$(186):G$=CHR$(187):H$=CHR$(188)
40 I$="":FOR I=1 TO 38:I$=I$+E$:NEXT I
50 PRINT"ENTREZ VOTRE TEXTE":LOCATE 4,1,1
60 A$=INKEY$:IF A$="" THEN 60
70 IF ASC(A$)=13 THEN 100
80 IF ASC(A$)=8 THEN B$=LEFT$(B$,LEN(B$)+(LEN(B$)<>0)) ELSE B$=B$+A$
90 LOCATE 4,1:PRINT B$;" ":LOCATE 4,1+LEN(B$),1:GOTO 60
100 WIDTH 40:RE$=" "+B$+" "
110 COLOR 1:PRINT:PRINT"DEFILEMENT A DROITE (D) OU A GAUCHE (G)":S$=INPUT$(1)
120 PRINT:IF S$>="a" THEN S$=CHR$(ASC(S$)-32)
130 INPUT"VITESSE (DE 0 A 50) :";V
140 V=51-V:B$=RE$:T=LEN(B$):CLS:S=T:IF T>38 THEN S=38
150 LOCATE 12,20-(S/2):COLOR 4:PRINT D$+LEFT$(I$,T)+G$
160 LOCATE 13,20-(S/2):PRINT F$:LOCATE 13,21-(S/2)+S:PRINT F$;
170 LOCATE 14,20-(S/2):PRINT C$+LEFT$(I$,T)+H$
180 COLOR 1:LOCATE 13,21-(S/2):PRINT MID$(B$,1,S):SOUND INT(RND*4000+500),1
190 IF LEFT$(S$,1)="G" THEN B$=RIGHT$(B$,T-1)+LEFT$(B$,1)
200 IF LEFT$(S$,1)="D" THEN B$=RIGHT$(B$,1)+LEFT$(B$,T-1)
210 FOR J=0 TO 10*V
220 R$=INKEY$
230 IF R$="0" THEN 110
240 NEXT J
250 GOTO 180

```

```

REM Affichage publicitaire

STREAM #1:SCREEN GRAPHICS
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "AFFICHAGE PUBLICITAIRE"
WINDOW OPEN
k=0:PRINT AT (1;2) "Entrez un texte court"
" AT (1;5) "-
LABEL texte
a$=INKEY$:IF a$=CHR$(13) THEN b$=" "+b$+" "
":GOTO début
IF a$=CHR$(8) AND a$<>" THEN b$=LEFT$(b$,LEN(b$)-1) ELSE b$=b$+a$
PRINT AT (1;4) b$ " ":LOCATE 1;5:PRINT TAB(LEN(b$)+1) "- ":GOTO texte
LABEL début
l=LEN(b$):d=8*1:IF k<>0 THEN CLS
PRINT AT (1;8) "Défilement (D)roite ou (G)auche : ";s$=INPUT$(1)
s$=UPPER$(s$):IF s$="" GOTO fin
LABEL vite
REPEAT
PRINT AT (1;11) "Vitesse ( de 0 à 50 ) : ";:INPUT v$
c=LET v=VAL(v$)
UNTIL c=0
IF v<0 OR v>50 GOTO vite ELSE v=51-v:r$=b$:l=LEN(r$)
CLS:BOX 20;160-d,1.4*d+40,d+20
BOX 60;180-d,1.4*d-50,d-20 COLOR(RND(16)-1) FILL WITH 8
LABEL boucle
LOCATE 7;4:PRINT r$
LOCATE 6;4+1/1.4:PRINT ANGLE(90) POINTS(8) r$
LOCATE 7+1;1-(1<8)-(1<15):PRINT ANGLE(180) r$
LOCATE 4+1.4*1;4:PRINT ANGLE(270) POINTS(8) r$
IF s$="G" THEN r$=RIGHT$(r$,l-1)+LEFT$(r$,1)
IF s$="D" THEN r$=RIGHT$(r$,1)+LEFT$(r$,l-1)
j=0:REPEAT:a$=INKEY$:j=j+1:UNTIL j=9*v OR a$<>" "
IF a$="0" THEN k=1:GOTO début ELSE GOTO boucle
LABEL fin
PRINT AT (50;8) "C'est fini...";CHR$(7)
r$=INPUT$(1):END

```

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 GW-BASIC :

La ligne 60 permet de faire apparaître les météorites.

La ligne 80 fera apparaître le minerai d'urane et assure le déplacement des météorites sur l'écran.

Les lignes 90 à 110 contrôlent le déroulement du jeu (a-t-on heurté une météorite ou a-t-on capturé un bloc d'urane ?).

La rencontre d'une météorite provoque la fin de partie.

Votre vaisseau spatial est imprimé par la ligne 130 et son déplacement est contrôlé par la ligne 150.

POUR JOUER :

On lance le programme par RUN et les météorites commencent à monter sur l'écran, ainsi que les blocs d'urane.

Les touches de déplacement horizontal (→ et ←) seront utilisées pour déplacer votre vaisseau et récupérer l'urane, mais le moindre heurt avec une météorite provoquera la fin de la partie et l'affichage de la quantité d'urane capturée.

On pourra aussi arrêter à tout moment en tapant CTRL et Break.

EXTENSIONS POSSIBLES :

On peut rendre le jeu de plus en plus difficile en augmentant progressivement le nombre de météorites, ou encore, en minutant la durée de la partie.

On peut aussi prévoir un affichage permanent du score, introduire un deuxième vaisseau spatial manœuvré par une puissance étrangère et qui cherche aussi à s'emparer de l'urane...

VERSION BASIC 2 AMSTRAD :

La souris devra être prise en main, de façon à utiliser les boutons droit et gauche pour déplacer le vaisseau spatial. On peut arrêter à tout moment en appuyant simultanément sur les deux boutons.

```

10 REM CAPTURES DANS L'ESPACE
20 SCREEN 0,1:WIDTH 80:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF
40 CLS:X=15:Z=0
50 S=INT(RND*80+1)
60 LOCATE 23,S:COLOR 1:PRINT "#";
70 S=INT(RND*80+1)
80 IF RND<.7 THEN LOCATE 23,S:COLOR 2:PR
INT"#";:COLOR 7:PRINT:PRINT
90 C=SCREEN(13,X+1)
100 IF C=42 THEN PRINT"SCORE OBTENU :";Z
:PLAY"CEG":END
110 IF C=35 THEN Z=Z+1:SOUND INT(RND*400
0+500),1
120 LOCATE 12,X:PRINT"  "
130 LOCATE 13,X:COLOR 4:PRINT" "+CHR$(2)
+" "
140 A$=INKEY$:A$=A$+A$:A$=MID$(A$,2,1)
150 X=X-((A$="6" OR A$="M") AND X<77) +
((A$="4" OR A$="K") AND X>1)
160 GOTO 50

```

```

REM Captures dans l'espace

STREAM #1:SCREEN GRAPHICS 640 FIXED, 200
FIXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "CAPTURES DANS L'ESPACE"
WINDOW OPEN
RANDOMIZE TIME:w=35:z=0:s=0:c=75:DIM t(17
,2)
LABEL début
h=RND(c):BOX 8*h;3,10,7 COLOR(2) FILL WIT
H 8:t(1,1)=10*h
k=RND(c):IF RND<.8 AND k<>h THEN PRINT AT
(k;22) COLOR(14) CHR$(35):t(1,2)=k
FOR i=14 TO 1 STEP-1:t(i,1)=t(i-1,1):t(i,
2)=t(i-1,2):NEXT i
IF w=t(14,1)/10 OR w+1=t(14,1)/10 THEN z=
z+1:PRINT CHR$(7)
IF w=t(14,2) OR w+1=t(14,2) GOTO fin
PRINT AT (w-1;11) POINTS(20) "  ":s=s+1
PRINT AT (w-1;12) COLOR(4) " /\ "
a=BUTTON(1):b=BUTTON(2):IF a+b=2 GOTO fin
w=w-(b<>-1 AND w>1)+(a<>-1 AND w<72)
GOTO début
LABEL fin
PRINT AT (1;3) "Score obtenu " z " sur "
s
r$=INPUT$(1):END

```

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 GW-BASIC :

Le nombre de cibles est entré en ligne 50. La boucle I (80-100) effectue l'impression des cibles.

La ligne 120 affichera le score en permanence.

La balle est affichée par la ligne 130.

La boucle I (150-200) réalise l'impression des chiffres contenus dans les cibles, et permet le déplacement de la boule (ligne 190).

La ligne 220 regarde si l'on a tiré, et la ligne 230 regarde si la boule est encore sur la ligne de départ.

La ligne 240 fait progresser la boule d'une case vers la droite, et la ligne 250 vérifie si l'on a ou non atteint les cibles ; puis les lignes 260 et 270 regardent ce que l'on a touché.

La ligne 280 incrémente le score.

Les lignes 290 et 300 s'occupent de la fin de partie.

POUR JOUER :

Lancer le programme par RUN, et taper le nombre de cibles désiré suivi de RETURN. Les cibles s'affichent, et la partie commence. Votre balle, représentée par un O, est affichée en bas à gauche ; vous pouvez la déplacer avec les flèches haut et bas (touches ↑ et ↓), et vous pouvez tirer à tout moment en tapant Ø ou la barre d'espace.

Le score sera modifié en fonction de votre résultat, et une nouvelle balle apparaît. Il est possible d'arrêter à tout moment en tapant CTRL et Break, sinon le jeu s'arrête au bout d'un certain nombre de tirs.

EXTENSIONS POSSIBLES :

On pourra sans difficulté augmenter le nombre de cibles, modifier la règle d'apparition des nombres dans les cibles pour la rendre moins hasardeuse, s'arranger pour que la balle elle-même soit représentée par un chiffre qui devrait coïncider à l'arrivée avec celui de la cible.

VERSION BASIC 2 AMSTRAD :

Le nombre de cibles est choisi avec la souris en cliquant sur les symboles + ou - dans une fenêtre d'alerte. On clique le nombre choisi, puis on choisit le niveau (facile ou difficile) en cliquant sur la case correspondante dans une deuxième fenêtre. Au niveau facile, il est possible de continuer à contrôler le mouvement de la boucle après l'avoir lancée. Ce n'est plus possible au niveau difficile. On fait ensuite monter ou descendre la balle en cliquant sur les boutons droit ou gauche de la souris, puis, on lance en appuyant simultanément sur les deux boutons. On arrête en cours de partie en tapant CTRL et C.

```
10 REM CIBLES
20 SCREEN 0,1:WIDTH 80:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF:CLS
40 A$(1)=CHR$(198):A$(2)=CHR$(205):A$(3)=CHR$(187):A$(4)=CHR$(186):A$(5)=CHR$(188)
50 PRINT:INPUT"NOMBRE DE CIBLES :";N:IF N>6 THEN CLS:GOTO 50
60 P=9:J=70:S=0:B=1
70 CLS
80 FOR I=1 TO N
90 LOCATE 3*I,J:PRINT A$(1);A$(2);A$(3):LOCATE 3*I+1,J:PRINT " ";A$(4):LOCATE 3*I+2,J:PRINT A$(1);A$(2);A$(5)
100 NEXT I
110 K=0:A=22
120 LOCATE 23,1:COLOR 4:PRINT"SCORE :";S;" - BALLE :";B;
130 LOCATE A,K+1,0:COLOR 2:PRINT CHR$(1)
140 D=A
150 FOR I=1 TO N
160 U=INT(RND*P)
170 LOCATE 3*I+1,J+1,0:COLOR 1:PRINT CHR$(U+48)
180 A$=INKEY$:A$=A$+A$:A$=MID$(A$,2,1)
190 A=A - (A$="2" OR A$="P") + (A$="B" OR A$="H")
200 NEXT I
210 LOCATE D,K+1,0:PRINT " "
220 IF A$="0" OR A$=" " THEN B=B+1:GOTO 240
230 IF K=0 THEN 130
240 K=K+1
250 IF K<J THEN 130
260 C=SCREEN(A,K+1)
270 IF C<48 OR C>57 THEN PRINT"PERDU..." :PLAY"GEC":END
280 S=S+C-48:SOUND INT(RND*4000+500),2
290 P=P-1
300 IF P=0 THEN PLAY"CDE":END
310 GOTO 110
```

```
REM Cibles
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 FIXED
USER SPACE 640,220
WINDOW FULL ON
WINDOW TITLE "CIBLES"
WINDOW OPEN
RANDOMIZE TIME:n$="3"
LABEL choix
nb=ALERT 1 TEXT " Nombre de cibles","", "Cliquez sur + ou -","Puis sur le nombre choisi","ou sur RETURN" BUTTON "+","-",RETURN n$
IF nb=1 AND VAL(n$)<8 THEN n$=CHR$(VAL(n$)+49)
IF nb=2 AND VAL(n$)>1 THEN n$=CHR$(VAL(n$)+47)
IF nb<>3 GOTO choix ELSE n=VAL(n$)
op=ALERT 2 TEXT "Facile","", "Difficile","", " Choix du niveau" BUTTON "F","D"
op=op-1:z=0:bn=10
FOR i=1 TO n
a(i)=9:LINE 590;195-20*i,620;195-20*i,616;195-20*i,616;215-20*i,590;215-20*i,620;215-20*i WIDTH 6
NEXT i
REPEAT
k=1:a=2:PRINT AT (5;1) "Score : " z " - balle numéro : " bn
LABEL début
PRINT AT (k;a) "o":d=a
FOR i=1 TO n:u(i)=INT(RND*a(i)):PRINT AT (75;2*i+2) CHR$(u(i)+48)
IF op=0 OR k=1 THEN c=BUTTON(1):b=BUTTON(2):a=a-(c<>-1 AND a<20)+(b<>-1 AND a>1)
NEXT i
PRINT AT (k;d) " "
IF c+b<>2 AND k=1 GOTO début ELSE k=k+1:IF k<76 GOTO début
IF (a MOD 2)<>0 OR a<3 OR a>2*n+2 GOTO fin
f=(a-2)/2:z=z+u(f):PRINT CHR$(7):a(f)=a(f)-(RND(a(f))-1):bn=bn-1
UNTIL bn=1
LABEL fin
PRINT AT (1;3) "C'est fini...."
r$=INPUT$(1):END
```

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 GW-BASIC :

La ligne 70 permet au PC de choisir la profondeur du sous-marin.

La ligne 40 dessine votre bâtiment.

La ligne 50 dessine la partie supérieure de l'écran.

La ligne 110 dessine le sous-marin.

Le tir est effectué par un programme spécifique placé aux lignes 190 à 230.

Le déplacement (très rapide) de votre navire est réalisé par les lignes 160 et 170.

Le test, permettant de savoir si l'on a touché ou non, est effectué en ligne 130.

POUR JOUER :

Taper RUN pour commencer la partie.

Le bâtiment sera déplacé grâce aux flèches droite et gauche.

La touche Ø ou la barre d'espace autorise le tir.

Le navire, se déplaçant trois fois plus vite que les sous-marins, il sera possible de toucher plusieurs fois le même bâtiment.

Pour terminer la partie, il conviendra de faire CTRL et Break 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...

VERSION BASIC 2 AMSTRAD :

Dans cette version, le chasseur de sous-marins est déplacé en surface, en appuyant sur les touches droite ou gauche de la souris. Le lancer de grenade est effectué en appuyant sur la barre d'espace. On peut l'arrêter à tout moment en tapant CTRL et C.

```

10 REM SOUS-MARIN
20 SCREEN 0,1:WIDTH 80:COLOR 7,1
30 RANDOMIZE TIMER:KEY OFF
40 P=0:C=35:S=6:T$=STRING$(80," "):E$=CHR$(219)
50 CLS:COLOR ,11:FOR I=1 TO 6:PRINT T$:N
EXT
60 LOCATE S,1:PRINT " "
70 T=77:S=INT(RND*14+7):P=P+.01
80 X=5
90 LOCATE 6,C+2:COLOR 4,11:PRINT " ";
CHR$(174);CHR$(205);CHR$(205);CHR$(202);
CHR$(175);" "
100 COLOR 0,1:LOCATE 19,1:PRINT"SCORE : "
;:PRINT USING "###.##";P
110 LOCATE S,T:COLOR 2,1:PRINT CHR$(17);
CHR$(216);CHR$(16);" "
120 IF X=5 THEN Y=C+4
130 IF S=X AND (T=Y OR T+1=Y OR T+2=Y) T
HEN P=P+1:LOCATE S,T:COLOR 6,1:PRINT"###
":PLAY"CCC"
140 T=T-1:IF P>99 THEN LOCATE 19,1:PRINT
"C'EST FINI...":PLAY"CDE":END
150 IF T<1 THEN 60
160 A$=INKEY$:A$=A$+A$:A$=MID$(A$,2,1)
170 C=C-4*((A$="6" OR A$="M") AND C<72)
-((A$="4" OR A$="K") AND C>2))
180 IF A$<>"0" AND A$<>" " AND X=5 THEN
90
190 LOCATE X,Y:IF X=5 THEN COLOR 0,11 EL
SE COLOR 0,1
200 PRINT " ":X=X+1:COLOR 0,1
210 IF X>5 THEN 80
220 LOCATE X,Y:PRINT"0"
230 GOTO 90

```

```

REM Sous-marin

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "SOUS-MARIN"
WINDOW OPEN
RANDOMIZE TIME:a$=CHR$(27)+CHR$(5):b$=CHR
$(92)+CHR$(95)+CHR$(95)+CHR$(226)
c$=CHR$(27)+CHR$(17)+CHR$(27)+CHR$(18):c=
30
BOX 0;0,640,164 COLOR(4) FILL WITH 8
BOX 0;164,640,36 COLOR(5) FILL WITH 8
LABEL tirage
t=74:s=INT(RND*7+6):g=g+0.01
LABEL début
x=5
LABEL dessin
PRINT AT (c;4) COLOR(5) a$;a$;a$ COLOR(2)
b$ COLOR(5) a$;a$;a$
PRINT AT (1;19) "Score ";:PRINT USING "##
.##" g
PRINT AT (t+2;s) COLOR(4) a$+a$ COLOR(6)
MODE(2) AT (t;s) c$
IF x=5 THEN y=c+2
IF s=x AND (t-1=y OR t=y OR t+1=y) THEN g
=g+1:GOSUB touché
t=t-1:IF p>99 THEN PRINT AT (20;19) "C'es
t fini":r$=INPUT$(1):END
IF t<1 THEN PRINT AT (1;s) COLOR(4) a$;a$
:GOTO tirage
j=0:REPEAT:r=INKEY:a=BUTTON(1):b=BUTTON(2)
:j=j+1:UNTIL a<>-1 OR b<>-1 OR r=32 OR j
=8
c=c-3*((b<>-1 AND c<68)-(a<>-1 AND c>3))
IF r<>32 AND x=5 GOTO dessin
PRINT AT (y;x) COLOR(7) a$
FOR i=1 TO 2:NEXT i:PRINT CHR$(7)
PRINT AT (y;x) COLOR(4) a$
x=x+1:IF x>s GOTO début ELSE GOTO dessin

LABEL touché
FOR i=1 TO 4
PRINT AT (t;s) COLOR(14) MODE(2) "11"
PRINT CHR$(7);CHR$(7):PRINT AT (t;s) COLO
R(4) a$;a$
NEXT i
RETURN

```


Le jeu du missile

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : contre le PC.

Difficulté du jeu : moyenne.

LE JEU :

Le PC 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 pavé agrandi), et de revenir sain et sauf sur votre base ; mais, dès que le PC 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 du pavé numérique ; réussirez-vous à remplir votre mission ?

LE PROGRAMME GW-BASIC :

La ligne 50 dessine la base ennemie à survoler (sous forme d'un pavé agrandi).

La ligne 60 dessine votre base de départ.

Les lignes 80 et 110 choisissent l'instant de départ du missile.

La ligne 120 dessine votre avion.

Les lignes 140 et 150 définissent son mouvement suivant 8 directions, en utilisant les sous-programmes en lignes 260, 270, 290 et 300.

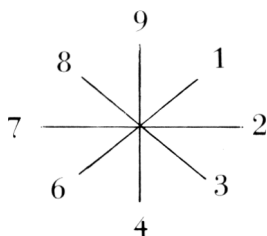
La ligne 160 assure le départ du missile qui est dessiné par la ligne 170 et dont le déplacement est géré par les lignes 180 et 190.

Les lignes 200 et 240 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 9, avec la convention suivante :



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

Si le missile vous détruit, vous obtenez un message TOUCHE, et la partie s'arrête.

Taper CTRL et Break pour arrêter la partie.

EXTENSIONS POSSIBLES :

Il sera sans doute nécessaire d'améliorer les divers contrôles du 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.

VERSION BASIC 2 AMSTRAD :

Dans cette version, le déplacement de l'avion est effectué en cliquant sur la direction choisie, à l'aide de la touche de gauche de la souris. Les directions sont symbolisées par des flèches, qui apparaissent dans des cases en bas et à droite de l'écran.

```
10 REM LE JEU DU MISSILE
20 SCREEN 0,1:WIDTH 80:COLOR 7,0
30 RANDOMIZE TIMER:KEY OFF
40 CLS:M=0:ON TIMER(3) GOSUB 320
50 LOCATE 6,38:COLOR 5:PRINT CHR$(177)+C
HR$(177):LOCATE 7,38:COLOR 3:PRINT CHR$(
219)+CHR$(219)
60 LOCATE 21,1:COLOR 2:PRINT CHR$(178)+C
HR$(178)+CHR$(178)
70 U=19:X=20
80 S=INT(RND*U+8)
90 Y=2
100 V=79
110 S=S-1
120 LOCATE X,Y,0:COLOR 4:PRINT CHR$(157)
130 A$=INKEY$:TIMER ON:IF A$="" THEN 130
140 F=-2*(VAL(A$)<5)-1:G=VAL(A$)+2*(F-1)
+2*(A$>"6" OR A$="3")-2*(A$="1" OR A$="6
")
150 ON G GOSUB 260,270,290,300
160 IF S<>0 THEN 110
170 LOCATE U,V:COLOR 1:PRINT CHR$(224)
180 U=U-SGN(U-X)
190 V=V-SGN(V-Y)
200 IF U=X AND V=Y THEN PRINT"TOUCHE..."
:PLAY"GEC":END
210 IF (U=6 OR U=7) AND (V=38 OR V=39) T
HEN 40
220 IF (X=6 OR X=7) AND (Y=38 OR Y=39) T
HEN 40
230 IF X<=6 AND Y>= 39 THEN M=1
240 IF M=1 AND X>=20 AND Y<=4 THEN PRINT
"MISSION ACCOMPLIE":PLAY"CEG":END
250 GOTO 120
260 Y=Y+F
270 X=X+G
280 RETURN
290 GOSUB 270
300 Y=Y-F
310 RETURN
320 TIMER OFF
330 U=U-SGN(U-X):V=V-SGN(V-Y):LOCATE U,V
:COLOR 1:PRINT CHR$(224)
340 RETURN
```

```

REM Le jeu du missile

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "LE JEU DU MISSILE"
WINDOW OPEN
RANDOMIZE TIME:m=0:a$=CHR$(27)+CHR$(3):b$
=CHR$(27)+CHR$(2)
PRINT AT (37;6) COLOR(2) POINTS(20) "H"
PRINT AT (2;18) COLOR(14) EFFECTS(64) MOD
E(4) " "
BOX 500;10,90,60 WIDTH 4:LINE 530;10,530;
69:LINE 560;69,560;10
LINE 500;30,590;30:LINE 590;50,500;50
LINE 504;14,526;26 START 1:LINE 504;40,52
6;40 START 1:LINE 504;66,526;54 START 1
LINE 545;14,545;26 START 1:LINE 545;66,54
5;54 START 1
LINE 586;14,564;26 START 1:LINE 586;40,56
4;40 START 1:LINE 586;66,564;54 START 1
PRINT AT (30;21) "Cliquez sur la directio
n"
u=74:v=9:x=a=2:ya=17:s=INT(RND*16+3)
LABEL debut:s=s-1
LABEL avance
PRINT AT (x;a;ya) COLOR(14) a$
REPEAT:ms=BUTTON(1)
GOSUB calcul:UNTIL ms<>-1 AND a<>0
f=SGN(6.5-a)+2*(a=4)
ON (a AND a<5)+((10-a) AND a>5) GOSUB mo1
,mo2,mo3,mo4
IF s<>0 GOTO debut
PRINT AT (u;v) b$
u=u-SGN(u-x):v=v-SGN(v-ya)
IF (u=x AND u=ya) THEN PRINT AT (1;18) "
TOUCHE....":r$=INPUT$(1):END
IF ((u=37 OR u=38) AND (v=5 OR v=6)) OR (
(x=37 OR x=38) AND (ya=5 OR ya=6)) THEN
RUN
IF x>=38 AND ya<=5 THEN m=1
IF m=1 AND x<=4 AND ya>=17 THEN PRINT AT
(1;18) "MISSION ACCOMPLIE":$=INPUT$(1):E
ND
GOTO avance

LABEL calcul
x=XMOUSE:y=YMOUSE:a=0
a=-((x>500 AND x<530 AND y>10 AND y<30)-2*
(x>530 AND x<560 AND y>10 AND y<30)
a=a-3*((x>560 AND x<590 AND y>10 AND y<30)
-4*((x>500 AND x<530 AND y>30 AND y<50)
a=a-6*((x>560 AND x<590 AND y>30 AND y<50)
-7*((x>500 AND x<530 AND y>50 AND y<70)
a=a-8*((x>530 AND x<560 AND y>50 AND y<70)
-9*((x>560 AND x<590 AND y>50 AND y<70)
RETURN

LABEL mo1:x=a-x-f
LABEL mo2:ya=ya+f:RETURN
LABEL mo3:GOSUB mo2
LABEL mo4:x=a+x+f:RETURN

```

Obstacles

REFERENCES :

Difficulté de programmation : assez difficile.

Catégorie de jeu : jeu de réflexion (pour un joueur).

Difficulté du jeu : moyenne.

LE JEU :

Des obstacles se déplacent sur l'écran pour venir à la rencontre d'une petite voiture qui dispose de la possibilité de se déplacer vers la gauche ou vers la droite pour les éviter et qui devra tenir le plus longtemps possible.

La distance parcourue sera affichée en permanence, et le jeu se termine lorsque la voiture rencontre, soit le bord du terrain, soit un obstacle.

LE PROGRAMME GW-BASIC :

Les lignes 40 à 60 définissent les caractères voitures et piétons.

La ligne 120 fait avancer les obstacles vers la voiture du joueur qui est imprimée grâce à la ligne 80.

Les lignes 90, 100, fabriquent la route (les lignes 100 et 110 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 130 affiche le score.

La ligne 150 permettra de déplacer le véhicule.

Les lignes 160 et 170 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 180.

POUR JOUER :

On lance le programme en tapant RUN.

La route et les obstacles apparaissent en bas de l'écran et montent vers la voiture. La voiture peut se déplacer à l'aide des flèches de déplacement horizontal (touches → et ←).

La rencontre avec un obstacle ou la bordure interrompt le déroulement du programme et provoque l'affichage du score.

On peut, cependant, sortir du programme à tout moment en utilisant les touches CTRL et Break.

EXTENSIONS POSSIBLES :

Il sera assez facile d'améliorer le dessin de la voiture, de la route, des obstacles. 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.

VERSION BASIC 2 AMSTRAD :

Les obstacles devront être évités, le plus longtemps possible, en cliquant sur les boutons de droite et de gauche de la souris, tenue à la main. L'arrêt est provoqué par la rencontre d'un obstacle, ou la frappe de CTRL et C.

```

10 REM OBSTACLES
20 SCREEN 1,1:WIDTH 40:COLOR 1
30 RANDOMIZE TIMER:KEY OFF:CLS
40 CLEAR ,49151!:DATA 93,127,85,20,93,12
7,93,8,8,20,8,62,42,8,28,54
50 DEF SEG=0:POKE 124,0:POKE 125,192:POK
E 126,PEEK(1296):POKE 127,PEEK(1297):DEF
SEG
60 FOR I=49152! TO 49167!:READ A:POKE I,
A:NEXT
70 S=0:Y=5
80 LOCATE 2,Y+1:PRINT CHR$(128)
90 LOCATE 23,1:PRINT CHR$(127)+"
"+CHR$(127);
100 IF RND<.7 THEN LOCATE 23,INT(RND*8+2
):COLOR ,2:PRINT CHR$(129);
110 IF RND <.8 THEN LOCATE 23,INT(RND*8+
2):COLOR ,3:PRINT CHR$(129);
120 LOCATE 23,1:PRINT:PRINT:LOCATE 6,16:
PRINT"
"
130 LOCATE 7,17:PRINT"SCORE :";S
140 A$=INKEY$:A$=A$+A$:A$=MID$(A$,2,1)
150 Y=Y - ((A$="6" OR A$="M") AND Y<9) +
((A$="4" OR A$="K") AND Y>0)
160 C=SCREEN(2,Y+1)
170 IF C=127 OR C=1 THEN PLAY"CDE":END
180 S=S+1
190 GOTO 80

```

```

REM Obstacles

STREAM #1:SCREEN GRAPHICS 640 FIXED, 200
FIXED
USER SPACE 640, 200
WINDOW FULL ON
WINDOW TITLE "OBSTACLES"
WINDOW OPEN
RANDOMIZE:t$=STRING$(184,"0")
u=22:z=0:c=7:d=c:a$=CHR$(27)+CHR$(5):b$=C
HR$(27)+CHR$(2)
REPEAT
PRINT AT (2;u) a$;a$ AT (12;u) a$;a$
IF RND<.7 THEN x=RND(8)+3:PRINT AT (x;u)
COLOR(4) b$;:MID$(t$,x-3,1)="1"
IF RND<.8 THEN x=RND(8)+3:PRINT AT (x;u)
COLOR(2) b$;:MID$(t$,x-3,1)="1"
t$="000000000"+LEFT$(t$,176)
PRINT AT (30;10) STRING$(30,32) AT (30;11
)"Score " z
PRINT AT (d;2) " " AT (c;3) "&"
a=BUTTON(1):b=BUTTON(2)
d=c:c=c-(b<>-1 AND c<12)+(a<>-1 AND c>2)
et$=MID$(t$,173+c,1):z=z+1
UNTIL et$="1" OR (c<=3 AND z>20) OR (c>=1
1 AND z>20)
PRINT AT (30;15) "C'est fini.....";CHR$(7
)
r$=INPUT$(1):END

```

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 GW-BASIC :

Les lignes 40 à 60 redéfinissent les caractères voiture et bordure.

L'impression de la route est assurée par la boucle I (80-100).

La ligne 110 fait avancer la route, et la ligne 120 imprime la voiture.

La ligne 150 contrôle son mouvement, et la ligne 160 détermine si la route fait ou non des virages.

La ligne 170 regarde si l'on a touché le bord de la route.

La ligne 180 incrémente le score.

La ligne 220 affiche le temps avant l'accident.

POUR JOUER :

Lancer l'exécution du programme par l'ordre RUN, le dessin de la route commence à évoluer, et votre véhicule, représenté par une voiture, apparaît.

Pour le déplacer vers la droite ou vers la gauche, vous devrez utiliser les touches → et ← (flèches de déplacement du curseur). Il vous faudra éviter le plus longtemps possible de toucher une bordure, ce qui provoquera l'arrêt du programme et l'affichage de la mention ACCIDENT, suivie de l'indication sur le score réalisé.

On pourra arrêter le déroulement de la partie à tout moment en pressant les touches CTRL et Break.

EXTENSIONS POSSIBLES :

On peut envisager de jouer à plusieurs, chacun son tour, en enregistrant le meilleur score.

On peut aussi envisager d'élargir la piste, et de faire courir plusieurs voitures.

On pourra aussi s'inspirer des jeux d'arcades qui sont particulièrement nombreux sur ce thème.

VERSION BASIC 2 AMSTRAD :

On utilise de nouveau les boutons droit et gauche de la souris pour déplacer la voiture, le jeu se terminant dès que l'on touche le bord de la route (on peut aussi taper CTRL et C).

```

10 REM COURSE DE VOITURES
20 SCREEN 1,1:WIDTH 40:COLOR 1
30 RANDOMIZE TIMER:KEY OFF:T=TIMER
40 CLEAR,49151':DATA 93,127,85,20,93,127
,93,8,255,255,255,255,255,255,255
50 DEF SEG=0:POKE 124,0:POKE 125,192:POK
E 126,PEEK(1296):POKE 127,PEEK(1297):DEF
SEG
60 FOR I=49152! TO 49167!:READ A:POKE I,
A:NEXT
70 CLS:X=22:Y=11:A=5:B=10:S=0
80 FOR I=0 TO 6
90 IF I<2 OR I>4 THEN LOCATE X+1,Y+I+1:P
RINT CHR$(129);
100 NEXT I
110 LOCATE 23,1:PRINT :PRINT
120 LOCATE A+1,B+1,0:PRINT CHR$(128):SOU
ND 600,1
130 FOR I=1 TO 10:A$=INKEY$:IF A$="" THE
N NEXT I
140 C=B:A$=A$+A$:A$=MID$(A$,2,1)
150 B=B+(A$="4" OR A$="K")-(A$="6" OR A$
="M")
160 Y=Y-(2*RND AND Y<25)+(2*RND AND Y>3)
170 IF POINT(8*B+3,8*A+11)=3 THEN 210
180 S=S+1
190 LOCATE A+1,C+1:PRINT" "
200 GOTO 80
210 PRINT"ACCIDENT :SCORE :";S
220 PRINT:PRINT"TEMPS :";INT(TIMER-T):PL
AY"CDE"

```

```

REM Course de voiture

STREAM #1:SCREEN GRAPHICS 640 FIXED, 200
FIXED
USER SPACE 640, 200
WINDOW FULL ON
WINDOW TITLE "COURSE DE VOITURE"
WINDOW OPEN
RANDOMIZE:t$=STRING$(16,CHR$(64+11))
u=22:z=-14:v=16:c=9:d=c:a$=CHR$(27)+CHR$(
5):b$=CHR$(178)
REPEAT
PRINT AT (v;u) COLOR(4) a$a$;" ";a$a$
PRINT AT (d+7;7) " " AT (c+7;8) COLOR(2)
b$
i=1:REPEAT:a=BUTTON(1):b=BUTTON(2):i=i+1:
UNTIL i=12 OR a<>-1 OR b<>-1
d=c:c=c-(b<>-1 AND c<26)+(a<>-1 AND c>0)
v=v-(-2*RND AND v<35)+(-2*RND AND v>13)
t$=CHR$(64+v)+LEFT$(t$,15)
et=ASC(RIGHT$(t$,1)):z=z+1:c$=CHR$(71+c)
UNTIL (c$=CHR$(et) OR c$=CHR$(et+1) OR c$
=CHR$(et+5) OR c$=CHR$(et+6)) AND z>0
PRINT AT (30;15) "C'est fini.....";CHR$(7
)
r$=INPUT$(1):END

```

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 PC 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 GW-BASIC :

La ligne 40 définit la position de départ du joueur.

La ligne 50 choisit la coordonnée horizontale de la case où PC va tirer (liée à votre position), et la ligne 60 choisit la coordonnée verticale (arbitraire) de cette case.

Le déplacement du joueur est contrôlé par les lignes 130 et 140. Sa position actuelle (sur l'écran, un symbole #) est imprimée par la ligne 150.

La ligne 160 regarde s'il est touché ou non.

La ligne 170 effectue le contrôle de fin de partie.

POUR JOUER :

On commence l'exécution du programme par RUN.

Le joueur apparaît, représenté par un symbole #.

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 CTRL et Break.

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.


```

10 REM EVASION
20 SCREEN 1,0:COLOR 1,2
30 RANDOMIZE TIMER:KEY OFF
40 CLS:X=0:Y=12
50 H=8*(X-1)+16*RND
60 K=190*RND+5
70 IF X<3 THEN 110
80 FOR J=1 TO 7*RND+3
90 PSET (H+6*RND-2,K+6*RND-2),1
100 NEXT J
110 SOUND INT(RND*4000+500),1:LOCATE Y+1
,X+1:PRINT "."
120 A$=INKEY$:A$=A$+A$:A$=MID$(A$,2,1)
130 X=X-(A$="M" OR A$="6")+(A$="K" OR A$
="4")
140 Y=Y-(A$="F" OR A$="2")+(A$="H" OR A$
="8")
150 LOCATE Y+1,X+1:PRINT "#"
160 IF X>3 AND (ABS(H-8*X)+ABS(K-8*(25-Y
))<=10) THEN PRINT"PERDU":PLAY"G6EECC":E
ND
170 IF X<39 THEN 50
180 PRINT"BRAVO":PLAY"CCEE66"

```

```

REM Evasion
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,220
WINDOW FULL ON
WINDOW TITLE "EVASION"
WINDOW OPEN
RANDOMIZE TIME
x=1:y=12
REPEAT
h=8*(x-1)+16*RND
k=180*RND+5
IF x<3 GOTO suite
FOR j=1 TO 7*RND+3
PLOT h+6*RND-2;k+6*RND-2 COLOR RND(6)
NEXT j
LABEL suite
PRINT AT (x;y) "."
a=INKEY
x=x+((a=331 OR a=52) AND x>1)-((a=333 OR
a=54) AND x<78)
y=y+((a=328 OR a=56) AND y>1)-((a=336 OR
a=50) AND y<20)
PRINT AT (x;y) "#"
IF x>3 AND (ABS(h-8*x)+ABS(k-8*(25-y))<=1
0) THEN PRINT "PERDU";CHR$(7):END
UNTIL x>=77
PRINT "BRAVO";CHR$(7);CHR$(7);CHR$(7):END

```

Mathieu

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : dessin aléatoire.

Difficulté du jeu : aucune.

LE JEU :

On utilise la définition d'une courbe en coordonnées polaires et la fonction PSET 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 GW-BASIC :

Les 3 premiers paramètres Z, K et T, introduits en ligne 40, définissent le rayon maximum et l'occupation de l'écran.

Les variables aléatoires A et B, choisies entre 0 et 3, de la ligne 50, définissent l'allure de la courbe ; il pourra être intéressant de les introduire manuellement (par exemple : 50 INPUT A, B).

La boucle H (60-110) réalise le tracé, point par point. Le rayon R est calculé en ligne 70, et le passage en coordonnées cartésiennes est effectué aux lignes 80 et 90.

L'affichage des points est réalisé en ligne 100.

La ligne 120 imprime un message de fin.

La ligne 150 permet de refaire un nouveau dessin, alors que la ligne 160 assure un nouveau tracé sans effacement du précédent.

POUR JOUER :

On lance le programme en tapant RUN.

Le dessin sera ensuite exécuté automatiquement.

Le message "c'est fini" signale la fin du tracé.

Si on veut un nouveau tracé, on tape 0.

Si on veut un tracé sans effacement du dessin précédent, on tape une touche quelconque.

Pour arrêter, faire CTRL et Break.

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).

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM MATHIEU
20 SCREEN 2
30 RANDOMIZE TIMER:KEY OFF
40 Z=9:k=10:L=INT(RND*5+1)+10:T=3:CLS
50 A=3*RND:B=3*RND
60 FOR H=0 TO 6.28 STEP 3.14/180
70 R=Z*SIN(H*T)
80 X=300+INT(L*R*COS(A*H))
90 Y=100+INT(K*R*SIN(B*H))
100 PSET (X,Y),1
110 NEXT H
120 LOCATE 22,1:PRINT"C'EST FINI..."
130 PLAY"CEG"
140 A$=INPUT$(1)
150 IF A$="0" OR A$="à" THEN 40
160 GOTO 50

```

```

REM Mathieu

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "MATHIEU"
WINDOW OPEN
RANDOMIZE TIME:OPTION RADIANS
d=15:z=9:k=10:l=RND(5)+10:u=300:v=100:t=3
:pa=0.6
LABEL début
c=RND(d):f=RND(d):IF c=f GOTO début
BOX 0;0,640,200 COLOR(f) FILL WITH 8
a=3*RND:b=3*RND
FOR h=0 TO 6.28 STEP pa/180
r=z*SIN(h*t)
x=u+INT(l*r*COS(a*h))
y=v+INT(k*r*SIN(b*h))
PLOT x;y COLOR(c)
NEXT h
PRINT AT (1;19) "C'est fini.... "
FOR i=1 TO 10:PRINT CHR$(7);:NEXT i
r$=INPUT$(1):r$=UPPER$(r$)
IF r$="0" GOTO début
END

```

Ardoise magique

REFERENCES :

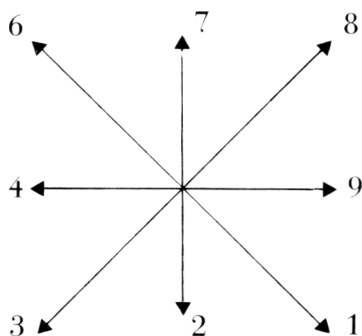
Difficulté de programmation : facile.

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 du pavé numérique suivant la convention :



On dispose aussi de la possibilité de déplacer le point sur l'écran sans laisser de trace en appuyant sur la touche 0 (ce qui permet l'effacement partiel) et de laisser de nouveau une trace en réappuyant sur 0.

On peut choisir la couleur du point en appuyant sur une touche de A à D ; et il est possible d'effacer tout l'écran en appuyant sur P.

LE PROGRAMME GW-BASIC :

La ligne 30 contient le choix des possibilités de couleur.

La ligne 50 imprime le point élémentaire.

Les lignes 60 à 100 contrôlent les entrées.

La ligne 140 appelle l'un des quatre sous-programmes effectuant le déplacement.

La ligne 150 vérifie que l'on ne sort pas des limites de l'écran.

La ligne 170 effectue l'effacement s'il y a lieu.

Le sous-programme 210-220 choisit la couleur ; 190-200 réalise ou non l'effacement.

POUR JOUER :

On lance le programme par RUN ; un point apparaît vers le milieu de l'écran. On devra taper 0 pour le déplacer sans afficher.

Si on frappe une lettre de A à D, la couleur du point change.

La touche P efface tout, et l'on recommence.

On peut sortir à tout moment, en tapant CTRL et Break.

EXTENSIONS POSSIBLES :

On pourra, bien sûr, envisager d'écrire un programme identique en utilisant des tracés de segments, prévoir un sous-programme qui permette la sauvegarde du dessin réalisé pour une utilisation ultérieure, etc.

VERSION BASIC 2 AMSTRAD :

On utilise ici les deux touches de la souris pour tracer des segments de droite (touche de gauche), des cercles (touche de droite) ou pour changer la couleur du fond (appui simultané des deux touches). La couleur de l'encre peut être choisie en appuyant sur la touche ESC, et la taille des cercles en appuyant sur la touche DEL.

```
10 REM ARDOISE MAGIQUE
20 SCREEN 1,K:WIDTH 40:COLOR 1-K,2:KEY 0
FF:CLS
30 C$="0123"
40 X=160:Y=100:C=2
50 PSET (X,Y),C
60 A$=INKEY$:IF A$="" THEN 60
70 IF A$>="A" AND A$<="D" THEN GOSUB 210
80 IF A$="0" THEN GOSUB 190
90 IF A$="P" THEN K=ABS(K-1):GOTO 20
100 IF VAL(A$)=0 THEN 60
110 F=-2*(VAL(A$)<5)-1
120 A=X:B=Y
130 U=VAL(A$)+2*(F-1)+2*(A$>"6" OR A$="3")-2*(A$="1" OR A$="6")
140 ON U GOSUB 230,240,260,270
150 D=-(X>0 AND X<319 AND Y>0 AND Y<199)
160 IF D=0 THEN GOSUB 290
170 IF S<>0 THEN PRESET (A,B)
180 GOTO 50
190 S=ABS(S-1)
200 RETURN
210 C=VAL(MID$(C$,ASC(A$)-64,1))
220 RETURN
230 X=X+F
240 Y=Y+F
250 RETURN
260 GOSUB 240
270 X=X-F
280 RETURN
290 X=A:Y=B
300 RETURN
```

```
REM Ardoise magique
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 FIXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "ARDOISE MAGIQUE"
WINDOW OPEN
WINDOW MOUSE(5):fo=0:co=1:t=2:x=320:y=100:x0=320:y0=100
REPEAT
PRINT AT (2;1) "ENCRE : " co " FOND : " fo " MODE : TAILLE : " t
LABEL boucle
IF z=1 THEN LINE x0;y0,x;y COLOR(fo) ELSE u=0
z=BUTTON(1)+BUTTON(2)+x=XMOUSE:y=YMOUSE+8:z=z+(z=4):PRINT AT (34;1) z
IF z=2 THEN ELLIPSE x;y,t,0.5 COLOR(co) FILL WITH 8:GOTO boucle
t1=-(z<>0)+u:IF t1=1 THEN x0=x:y0=y:u=-1
IF z=1 OR t1=-1 THEN LINE x0;y0,x;y COLOR(co)
IF z=3 THEN GOSUB fond
r=INKEY:IF r=27 THEN co=(co+1) MOD 16
IF r=8 THEN t=((t+1) MOD 20)+1
UNTIL r=32
WINDOW MOUSE(0)
r$=INPUT$(1):END

LABEL fond
PRINT AT (10;1) "? ":INPUT AT (12;1) " ",co
PRINT AT (22;1) "? ":INPUT AT (24;1) " ",fo
BOX 1;1,640,188 WIDTH 5 COLOR(fo) FILL WITH 8
RETURN
```

Cube

REFERENCES :

Difficulté de programmation : assez difficile.

Catégorie de jeu : dessin.

Difficulté du jeu : moyenne.

LE JEU :

Un cube est affiché en 3D, ainsi que 3 axes de coordonnées ox , oy et oz .

Il s'agit d'effectuer des rotations, d'un angle quelconque exprimé en degrés, autour de l'un des 3 axes.

Le cube tourne, puis on peut essayer une nouvelle rotation.

LE PROGRAMME GW-BASIC :

La ligne 30 prépare 3 variables indicées X, Y et Z, pour recevoir les coordonnées des sommets.

La boucle I (ligne 40) lit les DATA (ligne 120).

La ligne 50 calcule la position initiale.

La boucle WHILE-WEND (lignes 60 à 170) permet d'opérer tant que l'angle de rotation n'est pas égal à 0.

Les lignes 70 à 80 dessinent les axes.

La boucle I (ligne 100) dessine le cube.

Les entrées sont faites aux lignes 110 à 140, et gérées par les lignes 150 et 160, qui appellent les sous-programmes correspondants.

POUR JOUER :

On lance l'exécution par RUN. Un premier cube est affiché. On tape X, Y ou Z, selon l'axe de rotation choisi, puis on introduit un angle en degrés. La rotation est alors effectuée.

On arrête en tapant 0 à la question « ANGLE ».

EXTENSIONS POSSIBLES :

On pourra imaginer de colorer les faces, et faire disparaître les faces cachées.

On peut travailler avec d'autres objets, en modifiant les DATAs, et faire un véritable programme 3D en modifiant la position de l'œil de l'observateur (changement d'échelle).

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

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 GW-BASIC :

La boucle I (70-100) affiche le terrain.

La ligne 110 définit la position initiale du sous-marin ; la ligne 120 permet son déplacement, contrôlé par la ligne 130 (on reste dans les limites du terrain).

La ligne 140 affiche votre position.

Les lignes 150 à 170 contrôlent votre déplacement.

La distance est calculée en ligne 180, affichée en ligne 190 et testée en ligne 200.

Le test de fin de partie est effectué à la ligne 220.

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 les touches CTRL et Break.

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...

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.


```

10 REM CHASSE AU SOUS-MARIN
20 SCREEN 1,1:WIDTH 40:COLOR 5,0
30 RANDOMIZE TIMER:KEY OFF:CLS
40 LOCATE 1,1:INPUT "DIMENSION DE LA GRILLE : ",H
50 IF H<1 OR H<>INT(H) OR H>10 THEN 30 ELSE CLS
60 A=156-8*H:B=92-8*H:L=16*H:P=16*H:F=INT((H+1)/2):G=H MOD 2
70 FOR I=0 TO H*H-1
80 J=I MOD H:K=INT(I/H):LINE(A+16*J,B+16*K)-(A+16*J+16,B+16*K+16),,B
90 LINE(A+16*J+1,B+16*K+1)-(A+16*J+14,B+16*K+14),1,BF
100 NEXT I:I=1
110 V=INT(RND*H):U=INT(RND*H)
120 U=U+I:V=V+I:S=S+I
130 IF (U<0 OR U>H OR V<0 OR V>H) THEN I=-I
140 LOCATE 2*X+11+G,2*Y+19+G,1,0,7:R$=INKEY$:IF R$="" THEN 140
150 R$=R$+R$:R$=MID$(R$,2,1)
160 X=X+((R$="B" OR R$="H") AND X>-(F-1))-((R$="2" OR R$="P") AND X<F-1-(G=0))
170 Y=Y+((R$="4" OR R$="K") AND Y>-(F-1))-((R$="6" OR R$="M") AND Y<F-1-(G=0))
180 D=INT(SQR((ABS(X-(V-F)))^2+(ABS(Y-(U-F)))^2))
190 LOCATE 1,1,0,0,0:PRINT "DIST. ";STR$(D)+" "
200 IF D<>0 THEN IF D<3 THEN 120 ELSE 140
210 LOCATE 3,1:PRINT "PROF. ":D$=INPUT$(1)
220 D=VAL(D$):IF D<>INT(RND*3+2) THEN LOCATE 3,1:PRINT SPACE$(7):PLAY"CCC":GOTO 110
230 LOCATE 5,1:PRINT "BRAVO":PLAY"CDEF6":END

```

```

REM Chasse au sous-marin

STREAM #1:SCREEN GRAPHICS 320 FIXED,200 FIXED
USER SPACE 320,20:SCREEN #2 TEXT 38 FIXED,22 FIXED
WINDOW #2 SIZE 38,22:WINDOW #2 PLACE 328,20
WINDOW #2 OPEN:WINDOW #2 TITLE "JOUEUR":WINDOW FULL ON
WINDOW TITLE "CHASSE AU SOUS-MARIN"
WINDOW OPEN:RANDOMIZE TIME
INPUT AT (1;3) "Dimension de la grille : ",h$:h=VAL(h$)
IF h<>INT(h) OR h<2 OR h>10 THEN RUN
CLS:p=0:c=1
FOR i=1 TO h
PRINT AT (1;2*i+1) CHR$(47+i) AT (3*i+1;1) CHR$(47+i)
FOR j=1 TO h:GOSUB rectangle:NEXT j
NEXT i
k=1:x=1:y=1
LABEL tirplace
u=INT(RND*H):v=INT(RND*H)
LABEL fuite
u=u+k:v=v+k:IF u<0 OR u>h OR v<0 OR v>h THEN k=-k:PRINT CHR$(7)
LABEL déplace
i=:j=y:p=8:c=2:GOSUB rectangle
REPEAT:r=INKEY:UNTIL r<>-1
p=0:c=1:GOSUB rectangle
y=y-(r=336 AND y<h)+(r=328 AND y>1)
x=x-(r=333 AND x<h)+(r=331 AND x>1)
d=INT(SQR((x-u)*(x-u)+(y-v)*(y-v)))
PRINT #2,AT (1;3) "Distance " STR$(d)+" "
IF d=0 THEN PRINT CHR$(7);CHR$(7);CHR$(7):GOTO tirer
IF d<3 GOTO fuite ELSE GOTO déplace

LABEL tirer
PRINT #2,AT (1;5) COLOR(4) "Profondeur (2 à 4 )":d$=INPUT$(1):d=VAL(d$)
IF d<>INT(RND*3+2) THEN PRINT #2,AT (1;5) STRING$(30,32):GOTO tirplace
PRINT #2,AT (1;9) COLOR(14) "BRAVO...."
r$=INPUT$(1):END

LABEL rectangle
BOX 24*i-8;166-16*j,24,16 COLOR(c) FILL WITH p
RETURN

```

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 « 000 » pour pouvoir rejouer s'il y avait eu saut ou si le coup n'était pas possible.

LE PROGRAMME GW-BASIC :

Les boucles I (40-140) et J (60-130) 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 (160-220) et I (170-210).

La ligne 240 indique le joueur qui a le trait.

Le coup est entré en ligne 250.

La ligne 260 réalise le changement de joueur (on tape 000).

La ligne 280 regarde si le déplacement est possible.

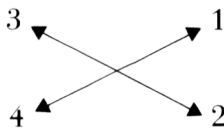
La ligne 310 traite le cas d'un déplacement et la ligne 320 celui d'un saut.

La ligne 360 assure le changement de joueur.

POUR JOUER :

Dès le lancement du programme par RUN, le damier est affiché avec ses pions, et un cercle de couleur apparaît à côté pour indiquer que le trait appartient au joueur correspondant.

On joue en indiquant par un mot de 3 lettres (pouvant commencer par un 0) la ligne et la colonne de la pièce à déplacer, puis la direction du déplacement (codée de 1 à 4) suivant la convention :



Par exemple 9D1 signifie que le pion situé en ligne 9, colonne D, se déplace dans la direction 1.

De même 0G3 signifiera pièce noire, ligne 0, colonne G, direction 3. Le changement de joueur étant effectué automatiquement il faudra taper « 000 » pour rejouer si le coup est refusé ou s'il y a eu prise. 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 PC de se mêler avec de bonnes chances de réussite à vos passionnantes parties.

VERSION BASIC 2 AMSTRAD :

Après l'affichage du damier, le joueur qui commence choisit le pion à déplacer en cliquant avec le bouton de gauche (le pion apparaît légèrement plus gros), puis il clique sur la flèche donnant la direction de déplacement avec le bouton de droite. Le déplacement est alors réalisé (s'il est possible). Dans le cas d'une prise, il pourra être nécessaire de cliquer sur les deux boutons pour imposer le changement de joueur. On quittera le programme en tapant CTRL et C.

```

10 REM DAMES
20 DIM R(10,10):DIM C(10,10):SCREEN 1,1:
COLOR 8:KEY OFF:CLS
30 PRINT"  A B C D E F G H I J
"
40 FOR I=1 TO 10
50 LOCATE 2*I+2,1:PRINT CHR$(47+I);
60 FOR J=1 TO 10
70 IF (I+J)/2=INT((I+J)/2) THEN 90
80 LINE (24*I-J-8,16*I+4)-(24*I+J+14,16*I+18
),3,BF
90 R(I,J)=0:C(I,J)=0
100 IF (I+J)/2<>INT((I+J)/2) THEN 130
110 IF J<5 THEN R(I,J)=1 :C(I,J)=2
120 IF J>6 THEN R(I,J)=1:C(I,J)=1
130 NEXT J
140 NEXT I
150 H=1
160 FOR J=1 TO 10
170 FOR I=1 TO 10
180 IF (I+J)/2<>INT((I+J)/2) THEN 210
190 X=3*I+1:Y=2*I
200 CIRCLE (8*X-4,8*Y+12),5,C(I,J):PAINT
(8*X-4,8*Y+12),C(I,J)
210 NEXT I
220 NEXT J
230 SOUND 800+400*H,2
240 CIRCLE (300,90),5,H+1:PAINT (300,90)
,H+1
250 D$=INPUT$(3)
260 IF D$="000" THEN H=ABS(H-1):GOTO 230
270 GOSUB 380
280 IF R(L,C)=0 OR (C(L,C)=1 AND H=1) OR
(C(L,C)=2 AND H=0) THEN 230
290 T=(1 AND (D=1 OR D=2)) - (1 AND (D=4
OR D=3))
300 S=(1 AND (D=3 OR D=2)) - (1 AND (D=1
OR D=4))
310 IF R(L+S,C+T)=0 THEN R(L,C)=0:R(L+S,
C+T)=1:C(L+S,C+T)=2+(H=0):C(L,C)=0:GOTO
360

```

REM Dames

```

STREAM #1:SCREEN GRAPHICS 640 FIXED, 200
FIXED
USER SPACE 640, 200
WINDOW FULL ON
WINDOW TITLE "DAMES"
WINDOW OPEN
RANDOMIZE TIME:DIM c(10,10):BOX 11;2,330,
188 WIDTH 5 COLOR(11) ROUNDED
BOX 350;0,280,200 COLOR(8) FILL WITH 4
FOR i=1 TO 10
PRINT AT (4*i+1;1) CHR$(64+i) AT (1;2*i+1
) CHR$(47+i)
FOR j=1 TO 10:GOSUB rec:c(i,j)=-(j<5)-5*(
j>6):h=-(j<5)
IF ((i+j) MOD 2)=1 AND c(i,j)<>0 THEN u=8
*i+1:v=2*j:r=10:fo=4:c=c(i,j):GOSUB cer
NEXT j,i
REPEAT:i=RND(10):j=RND(10):h=(i+j) MOD 2:
UNTIL c(i,j)<>0
BOX 440;20,100,50 COLOR(6) FILL WITH 8
LINE 450;25,530;65 START 1 END 1 WIDTH 3:
LINE 450;65,530;25 START 1 END 1 WIDTH 3
LABEL début
PRINT AT (50;5) MODE(2) POINTS(16) "JOUEU
R ":u=132:v=18:c=5-4*h:r=10:fo=4:GOSUB ce
r
PRINT AT (50;8) "CLIQUEZ LE PION (A GAUCH
E)" AT (50;10) "PUIS LA FLECHE (A DROTE)"
REPEAT
c=BUTTON(1):d=BUTTON(2):x=XMOUSE+15:y=181
-YMOUSE:i=INT(x/32):j=11-INT(y/16)
IF d<>-1 THEN h=ABS(h-1):PRINT CHR$(7):GO
TO début
UNTIL c<>-1 AND i>0 AND j>0 AND i<11 AND
j<11 AND ((i+j) MOD 2)<>0
IF c(i,j)=0 OR c(i,j)=1+4*h GOTO début
u=8*i+1:v=2*j:r=10:fo=8:GOSUB cer:PRINT C
HR$(7)
LABEL choix

```

```

320 IF C(L+S,C+T)=2+(H=1) AND R(L+S,C+T)
=1 AND R(L+2*S,C+2*T)=0 THEN C(L,C)=0:GO
TO 340
330 GOTO 230
340 R(L,C)=0:R(L+S,C+T)=0:R(L+2*S,C+2*T)
=1:C(L+S,C+T)=0:C(L+2*S,C+2*T)=2+(H=0)
350 GOTO 160
360 H=ABS(H-1)
370 GOTO 160
380 L=VAL(LEFT$(D$,1))+1:C=ASC(MID$(D$,2
,1))-64:D=VAL(RIGHT$(D$,1))
390 RETURN

REPEAT
c=BUTTON(1):d=BUTTON(2):x=XMOUSE:y=YMOUSE
:s=2*(x<490)+1:t=2*(y<40)+1
IF c+d=2 THEN PRINT CHR$(7):fo=4:GOSUB ce
r:h=ABS(h-1):GOTO début
UNTIL d<>-1 AND i+s>0 AND i+s<11 AND j+t>
0 AND j+t<11
IF x<350 OR y>80 GOTO choix
IF c(i+s,j+t)=0 THEN GOSUB eff:GOTO début
IF i+2*s<1 OR i+2*s>10 OR j+2*t<1 OR j+2*
t>10 GOTO choix
IF c(i+s,j+t)=1+4*h AND c(i+2*s,j+2*t)=0
THEN GOSUB dess:GOTO début
GOSUB rec:u=8*i+1:v=2*j:r=10:fo=4:GOSUB c
er:GOTO début

LABEL rec
a=32*i-16:b=18*j-12:BOX a;b,32,18 COLOR((
i+j) MOD 2) FILL WITH 8
RETURN

LABEL cer
ELLIPSE 4*u-4;9*v-3,r,0.7 COLOR(2*h+2) FI
LL WITH fo
RETURN

LABEL dess
c(i,j)=0:GOSUB rec:i=i+s:j=j+t:c(i,j)=0:G
OSUB rec:i=i+s:j=j+t
c(i,j)=5-4*h:u=8*i+1:v=2*j:r=10:fo=4:c=c(
i,j):GOSUB cer:RETURN

LABEL eff
c(i,j)=0:GOSUB rec:i=i+s:j=j+t:c(i,j)=5-4
*h:u=8*i+1:v=2*j:r=10:fo=4
c=c(i,j):GOSUB cer:h=ABS(h-1):RETURN

```

Dessin de dés

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : facile.

LE JEU :

Le programme proposé est surtout intéressant par son côté graphique puisqu'il permet de représenter, de façon très convenable, de 1 à 9 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 GW-BASIC :

La ligne 60 permet de contrôler s'il y a eu ou non action sur une touche de 1 à 9.

La boucle I (80-150) et les lignes 90 à 110 dessinent le cadre du dé.

La ligne 120 réalise le tirage, et la boucle J (130-150) dessine les points en relation avec le tirage réalisé (en appelant les variables A (J) et B (J), définies dans les sous-programmes 170 à 220).

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 9. Les dés demandés sont alors représentés, et la question NOMBRE DE DES est de nouveau posée. Il suffira de taper CTRL et Break pour sortir du programme.

EXTENSIONS POSSIBLES :

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

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM TIRAGE DE DES
20 SCREEN 2:CLS
30 RANDOMIZE TIMER:KEY OFF
40 LOCATE 1,1:PRINT"NOMBRE DE DES A LANCER : "
50 A$=INKEY$:IF A$="" THEN 50
60 IF A$<"1" OR A$>"9" THEN 40 ELSE N=VAL(A$)
70 CLS
80 FOR I=1 TO N
90 X1=70*I-66:Y1=80:X2=70*I-6:Y2=110:LINE (X1,Y1)-(X2,Y2),,B
100 LINE (X1,Y1)-(X1+8,Y1-8):LINE-(X2+8,Y1-8):LINE-(X2,Y1)
110 LINE (X2+8,Y1-8)-(X2+8,Y2-8):LINE-(X2,Y2)
120 X=INT(RND*6+1):ON X GOSUB 170,180,190,200,210,220
130 FOR J=1 TO X
140 CIRCLE A(J)+70*I-65,B(J)+65),5
150 NEXT J,I
160 PLAY"CEG":GOTO 40
170 A(1)=30:B(1)=30:RETURN
180 A(1)=10:A(2)=50:B(1)=20:B(2)=40:RETURN
190 GOSUB 180:A(3)=30:B(3)=30:RETURN
200 GOSUB 180:A(3)=10:B(3)=40:A(4)=50:B(4)=20:RETURN
210 GOSUB 200:A(5)=30:B(5)=30:RETURN
220 GOSUB 200:A(5)=10:B(5)=30:A(6)=50:B(6)=30:RETURN

```

```

REM Dessin de des

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 FIXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "DESSIN DE DES"
WINDOW OPEN
RANDOMIZE TIME
LABEL début
PRINT AT (1;2) "Nombre de dés ( 1 à 9 ) : "
:;r$=INPUT$(1):n=VAL(r$)
IF n<=0 OR n<>INT(n) OR n>9 THEN RUN
CLS
FOR i=1 TO n
x1=70*i-66:y1=140:x2=70*i-6:y2=110:c=RND(14)+1:c=c+(c=9)
SHAPE x1;y1,x2;y1,x2;y2,x1;y2 COLOR(c) FILL WITH B
SHAPE x1;y1,x1+8;y1+8,x2+8;y1+8,x2;y1 COLOR(c) FILL WITH 4
SHAPE x2+8;y1+8,x2+8;y2+8,x2;y2,x2;y1 COLOR(c) FILL WITH 4
x=RND(6):ON x GOSUB tir1,tir2,tir3,tir4,tir5,tir6
FOR j=1 TO x
PLOT a(j)+70*i-64;b(j)+94 MARKER 6 SIZE 5
NEXT j
PRINT CHR$(7)
NEXT i
GOTO début

LABEL tir1:a(1)=30:b(1)=30:RETURN
LABEL tir2:a(1)=10:a(2)=50:b(1)=20:b(2)=40:RETURN
LABEL tir3:GOSUB tir2:a(3)=30:b(3)=30:RETURN
LABEL tir4:GOSUB tir2:a(3)=10:b(3)=40:a(4)=50:b(4)=20:RETURN
LABEL tir5:GOSUB tir4:a(5)=30:b(5)=30:RETURN
LABEL tir6:GOSUB tir4:a(5)=10:b(5)=30:a(6)=50:b(6)=30:RETURN

```

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 instructions DRAW et PAINT. La couleur permet de distinguer chaque camp.

LE PROGRAMME GW-BASIC :

Les lignes 70 à 150 dessinent l'échiquier. Les lignes 120, 130 et 160, 170 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 (190-250) et I (200-240) dessinent les pièces.

La ligne 280 permet l'entrée de la ligne et de la colonne de départ, avec contrôle en ligne 310.

La ligne 330 permet d'entrer colonne et ligne d'arrivée.

La ligne 360 note la nouvelle position.

La ligne 370 assure le changement de joueur.

POUR JOUER :

On lance l'exécution par RUN.

L'échiquier est dessiné, puis les pièces, et enfin un texte indique le joueur qui a le trait ; on entre la colonne et la ligne de départ, sous la forme E 7 par exemple, un tiret s'affiche ensuite, et on donne la colonne et la ligne d'arrivée. Le déplacement est alors effectué, et c'est au tour de l'autre joueur.

Pour arrêter, il suffit de taper CTRL et Break.

EXTENSIONS POSSIBLES :

On pourra chercher à réaliser un graphisme plus précis, introduire le contrôle du mouvement des pièces. Et pourquoi pas, écrire un programme qui permette au PC de se comporter en partenaire honorable. Bon courage !

VERSION BASIC 2 AMSTRAD :

L'échiquier est dessiné, ainsi que les pièces. Puis le joueur actif est affiché et il choisit la pièce à déplacer en cliquant. La pièce apparaît alors en grisé, et le joueur choisit la case d'arrivée en cliquant de nouveau. Le déplacement est alors effectué et le trait passe au joueur suivant. On peut abandonner à tout moment en tapant CTRL et C.

```

10 REM ECHECS
20 DIM R(8,8):DIM C(8,8)
30 A$(0)="L3E2U2H1U1E1R2F1D1G1D2F2L3":A$(
1)="L4U2R3U3L3E3F1E1F3L3D3R3D2L4"
40 A$(2)="L4U1E1R2U3H2U1R1D1R2U1D1R2U1R1
D1G2D3R2F1D1L4":A$(3)="L4U2R3U2H3E2F1E1F
1E1F2G3D2R3D2L4"
50 A$(4)="L3E2U2H1U1E1R2F1D1G1D2F2L3":A$(
5)="L4U1E1R2U3L1G1H1E3R3F1G1D4R2F1D1L4"
60 SCREEN 1,1:COLOR 8:KEY OFF:RANDOMIZE
TIMER:CLS
70 PRINT:PRINT"      A  B  C  D  E  F  G
H"
80 FOR I=1 TO 8
90 LOCATE 2*I+3,1:PRINT I
100 FOR J=1 TO 8
110 IF (I+J) MOD 2<>0 THEN LINE(24*I+8,1
6*J+12)-(24*I+32,16*J+28),(I+J) MOD 2+2,
,BF
120 R(I,J)=6:IF I=2 OR I=7 THEN R(I,J)=4
130 C(I,J)=1:IF I=7 OR I=8 THEN C(I,J)=2
140 NEXT J
150 NEXT I
160 R(1,1)=2:R(1,2)=5:R(1,3)=0:R(1,4)=3:
R(1,5)=1:R(1,6)=0:R(1,7)=5:R(1,8)=2
170 FOR I=1 TO 8:R(8,I)=R(1,I):NEXT I
180 H=1
190 FOR J=1 TO 8
200 FOR I=1 TO 8
210 X=24*J+20:Y=16*I+24
220 IF R(I,J)=6 THEN LINE (24*J+12,16*I+
14)-(24*J+30,16*I+26),3*ABS((I+J) MOD 2)
,BF
230 IF R(I,J)<>6 THEN LINE(24*J+12,16*I+
14)-(24*J+30,16*I+26),3*ABS((I+J) MOD 2)
,BF:PSET (X,Y),C(I,J):DRAW "x a$(r(i,j));
":PAINT (X,Y-2),C(I,J)
240 NEXT I
250 NEXT J
260 LOCATE 7,29:PRINT"          ":LOCAT
E 23,20:PRINT"          "
270 LOCATE 22,1:IF H=1 THEN PRINT"JOUEUR
BLEU " ELSE PRINT"JOUEUR ROUGE"
280 LOCATE 23,1:INPUT"DEPART (C ET L) : "
;D$
290 LOCATE 7,31:PRINT D$:LOCATE 23,20:PR
INT "          "
300 GOSUB 390
310 IF R(L,C)=6 OR (C(L,C)=2 AND H=1) OR
(C(L,C)=1 AND H=0) THEN 260
320 T=R(L,C):S=C(L,C):R(L,C)=6
330 LOCATE 23,1:INPUT"ARRIVEE (C ET L) :
":D$
340 LOCATE 7,34:PRINT "- ";D$
350 GOSUB 390
360 R(L,C)=T:C(L,C)=S
370 H=ABS(H-1)
380 GOTO 190
390 L=VAL(RIGHT$(D$,1)):C=ASC(LEFT$(D$,1
))-64
400 RETURN

```



```

REM Echecs
STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "ECHECS"
WINDOW OPEN:DIM t(6,30)
FOR i=1 TO 6:j=1:REPEAT:READ t(i,j):j=j+1
:UNTIL j=28:NEXT i
a$(0)="ROUGE":a$(1)="BLEU ":g=0:e=0
BOX 45;38,334,151 WIDTH 5 COLOR(7) FILL W
ITH 4
FOR k=1 TO 8
PRINT AT (4;2*k+1) CHR$(48+k) AT (5*k+4;1
) CHR$(64+k)
FOR j=1 TO 8
BOX 40*k+10;186-18*j,40,18 COLOR((k+j) MO
D 2) FILL WITH 8
r(k,j)=-(j=2 OR j=7):c(k,j)=-2*(j>6)-4*(j
<3)
NEXT j
x=40*k+24:y=154:i=1:co=4:GOSUB dessin:y=6
4:co=2:GOSUB dessin
NEXT k
r(1,1)=2:r(2,1)=4:r(3,1)=3:r(4,1)=5:r(5,1
)=6:r(6,1)=3:r(7,1)=4:r(8,1)=2
FOR j=1 TO 8 STEP 7
FOR k=1 TO 8
r(k,8)=r(k,1):x=40*k+24:y=190-18*j:i=r(k,
j):co=c(k,j):GOSUB dessin
NEXT k,j
LABEL debut
PRINT AT (50;3) "JOUEUR " a$(g) AT (50;5)
"Cliquez la pièce"
PRINT AT (50;7) "Puis la case"
c=BUTTON(1)+BUTTON(2)+2:xo=INT((XMOUSE-8)
/40):yo=INT((181-YMOUSE)/16)
IF c=0 OR xo<1 OR yo<1 OR xo>8 OR yo>8 GO
TO debut
IF e=0 AND ((c(xo,yo)=2 AND g=1) OR (c(xo
,yo)=4 AND g=0) OR r(xo,yo)=0) GOTO debut
IF e=0 THEN x=40*xo+24:y=190-18*y:yo=i:r(xo
,yo):a=xo:b=yo:k=i:s=c(xo,yo):co=s+4:GOSU
B dessin:e=1:GOTO debut
IF e=1 THEN e=0:x=40*xo+24:y=190-18*y:yo=i=
k:co=s:BOX 40*a+10;186-18*b,40,18 COLOR((
a+b) MOD 2) FILL WITH 8:GOSUB dessin
r(xo,yo)=k:c(xo,yo)=s:g=1-g:PRINT CHR$(7)
:GOTO debut

LABEL dessin
a1=x+t(i,1):b1=y+t(i,2):a2=a1+t(i,3):b2=b
1+t(i,4):a3=a2+t(i,5):b3=b2+t(i,6):a4=a3+
t(i,7):b4=b3+t(i,8):a5=a4+t(i,9):b5=b4+t(
i,10)
a6=a5+t(i,11):b6=b5+t(i,12):a7=a6+t(i,13)
:b7=b6+t(i,14):a8=a7+t(i,15):b8=b7+t(i,16
):a9=a8+t(i,17):b9=b8+t(i,18):a10=a9+t(i,
19):b10=b9+t(i,20)
a11=a10+t(i,21):b11=b10+t(i,22):a12=a11+t
(i,23):b12=b11+t(i,24):a13=a12+t(i,25):b1
3=b12+t(i,26):a14=a13+t(i,27):b14=b13+t(i
,28)
SHAPE a1;b1,a2;b2,a3;b3,a4;b4,a5;b5,a6;b6
,a7;b7,a8;b8,a9;b9,a10;b10,a11;b11,a12;b1
2,a13;b13,a14;b14 COLOR(co) FILL WITH 8
RETURN

DATA 4,2,0,2,-2,2,3,2,3,-2,-2,0,-2,4,-
2,-10,0,0,0,0,0,0,0,0,0,0,0
DATA 5,3,0,2,-2,0,0,2,3,0,0,-2,2,0,0,2,3,
0,0,-2,-2,0,0,-2,5,-3,-14
DATA 5,2,0,2,-3,2,5,2,5,-2,-3,-2,0,-2,5,-
2,-14,0,0,0,0,0,0,0,0,0,0,0
DATA 5,2,0,3,-4,-2,-1,2,5,2,5,-2,-3,-1,0,
-2,5,-2,-12,0,0,0,0,0,0,0,0
DATA 5,3,-3,2,0,2,2,-2,2,2,2,-2,2,2,0,-2,
-3,-2,5,-3,-12,0,0,0,0,0,0,0
DATA 5,2,-3,3,2,2,-3,1,8,0,-3,-1,2,-2,-3,
-3,5,-2,-10,0,0,0,0,0,0,0,0

```

Fais-moi un dessin

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : création d'un dessin.

Difficulté du jeu : aucune.

LE JEU :

Le programme permet de confectionner son propre paysage, à partir d'icônes prédéfinies apparaissant dans des cases en haut de l'écran. Le programme initial propose 10 icônes.

LE PROGRAMME GW-BASIC :

Les lignes 40 à 130 définissent les 10 icônes. La boucle I (140-170) les dessine.

La ligne 190 gère la position du curseur et la pose du dessin.

Les lignes 210 et 220 contrôlent le déplacement du curseur.

La ligne 230 permet de choisir l'icône.

La ligne 250 permet de redessiner l'icône à l'emplacement choisi. La ligne 260 permet d'afficher la taille choisie.

POUR JOUER :

On lance le programme par RUN.

Les icônes sont affichées dans des carrés en haut à gauche : on déplace le curseur pour choisir l'icône et on tape ESPACE.

On choisit la taille en appuyant sur une touche de 1 à 9, puis on déplace l'icône, et on la dépose avec RETURN.

EXTENSIONS POSSIBLES :

On pourra augmenter le nombre d'icônes disponibles, permettre le choix des couleurs, l'agrandissement des icônes, etc.

On arrête le programme en tapant CTRL et Break.

VERSION BASIC 2 AMSTRAD :

La version Basic 2, outre l'utilisation pratique de la souris, permet en outre de choisir la largeur, la hauteur et la couleur des diverses icônes proposées.

Une icône d'effacement est également disponible. On choisit les icônes en cliquant avec la souris, puis on les déplace à la position désirée et on clique de nouveau pour les poser. On modifie les paramètres de dessin (couleur, largeur et hauteur) en cliquant sur les signes + et - disposés juste à côté des noms.

```

10 REM FAIS MOI UN DESSIN
20 SCREEN 2:KEY OFF:CLS
30 DIM K(600),A$(10),B$(10):T=1
40 B$(1)="" :A$(1)="U7E3R10U2R1D2R2F3D7L4
U3L6D3L8BM+2,-3U3R3D3L3"
50 B$(2)="BM+2,0":A$(2)="U3E2R7M+4,+3R3M
+1,+2L3G1L2H1L5G1L2H1L2BM+3,-2M+1,-2R3M+
3,+2L6"
60 B$(3)="BM+6,0":A$(3)="R1U5L1G1E1R1E1H
1E1F1G1F1R1F1H1L1D2L2R2D3R1"
70 B$(4)="BM+7,0":A$(4)="E1U8G1L2H1U2E2R
6F2D2G1L2H1D8F1L5"
80 B$(5)="BM+4,0":A$(5)="U1R1U2L1U1R1U1E
1F1D1R3U1E1F1D1R2D1L2D2R2D1L10"
90 B$(6)="BM+6,-6":A$(6)="E2H2F2R3U3F3M-
4,+3M+3,-3R4F1"
100 B$(7)="BM+6,0":A$(7)="U3E2M-2,-1M+2,
-1U2M+1,+3R2M-2,+1D2M-1,-2"
110 B$(8)="BM+2,0":A$(8)="U14R1D14R1U14R
1D14R1U14R1D14R1U14R1D14R1U14R1D14R1U14R
1D14R1U14R1D14R1U14R1D14R1U14R1D14"
120 B$(9)="BM+9,0":A$(9)="U14"
130 B$(10)="BM+4,-9":A$(10)="R10"
140 FOR I=1 TO 10
150 LINE (48*I-32,0)-(48*I-8,24),,B
160 N=48*I-30:DRAW "BM=N;,22"+B$(I)+A$(I
)
170 NEXT I
180 X=3:Y=1:F=1:CH=0
190 IF CH=0 THEN LOCATE X,Y,1 ELSE PUT(Y
,X),K,PSET
200 R$=INKEY$:R$=R$+R$:R$=MID$(R$,2,1)
210 A=X:X=X+F*((R$="B" OR R$="H") AND X
>1-26*(CH<>0))-((R$="2" OR R$="P") AND X
<23-152*(CH<>0)))
220 B=Y:Y=Y+F*((R$="4" OR R$="K") AND Y
>2)-((R$="6" OR R$="M") AND Y<79-530*(CH
<>0)))
230 IF R$=" " AND X<=3 AND ((INT(Y-4)/3)
MOD 2=0) AND Y<60 THEN CH=(1+INT(Y/3))/
2:LOCATE ,,0:GET(48*CH-31,3)-(48*CH-9,23
),K:X=26:Y=10:A=X:B=Y
240 IF R$=" " AND CH<>0 THEN F=ABS(9-F)
250 IF CH<>0 AND (A<>X OR B<>Y OR R$=CHR
$(13)) THEN PUT(B,A),K,XOR:IF R$=CHR$(13
) THEN U=4*T:I=Y+2:J=X+19:DRAW "S=U;BM=I
; ,=J;"+A$(CH):GOTO 180
260 LOCATE 2,64:PRINT "TAILLE ";T:IF VAL
(R$)<>0 THEN T=VAL(R$)
270 GOTO 190

```

```

REM Fais moi un dessin

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 F
IXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "FAIS MOI UN DESSIN"
WINDOW OPEN:DIM t(10,37):e=0:LINE 0;160,6
40;160
PRINT AT (74;2) MODE(4) "+" AT (76;2) "-"
AT (74;3) "+" AT (76;3) "-" AT (74;4) "+"
AT (76;4) "-"
FOR i=0 TO 10:j=1:REPEAT:READ t(i,j):j=j+
1:UNTIL j=35:NEXT i
FOR k=0 TO 10
m(1)=1:m(2)=1:x=40*k+24:y=172:BOX x-9;y-5
,30,20:i=k:m(3)=4:GOSUB dessin
NEXT k
LABEL début
PRINT AT (60;2) "Largeur * " m(1) AT (60;
3) "Hauteur * " m(2) AT (60;4) "Couleur :
" m(3)
c=BUTTON(1)+BUTTON(2)+2:xo=INT((XMOUSE-8)
/40):yo=INT((181-YMOUSE)/16)
IF c=2 AND xo=14 AND yo=1 THEN x=INT((XMO
USE-582)/10):y=INT((177-YMOUSE)/8):m(y)=A
BS(m(y)+(1 AND x=0)-(1 AND x=2)):IF y=3 T
HEN m(y)=m(y) MOD 16
IF c=0 OR (e=0 AND (xo<0 OR yo<1 OR xo>10
OR yo>1)) GOTO début
IF e=0 THEN x=40*xo+24:y=172:pl=x:i=xo:m(
3)=2:GOSUB dessin:e=1:GOTO début
IF e=1 AND c=4 THEN e=0:x=XMOUSE:y=190-18
*yo:GOSUB dessin:x=pl:y=172:m(3)=4:GOSUB
dessin
PRINT CHR$(7):GOTO début

LABEL dessin
m=m(1):p=m(2):co=m(3)
a1=x+m*t(i,1):b1=y+p*t(i,2):a2=a1+m*t(i,3
):b2=b1+p*t(i,4):a3=a2+m*t(i,5):b3=b2+p*t
(i,6):a4=a3+m*t(i,7):b4=b3+p*t(i,8):a5=a4
+m*t(i,9):b5=b4+p*t(i,10)
a6=a5+m*t(i,11):b6=b5+p*t(i,12):a7=a6+m*t
(i,13):b7=b6+p*t(i,14):a8=a7+m*t(i,15):b8
=b7+p*t(i,16):a9=a8+m*t(i,17):b9=b8+p*t(i
,18):a10=a9+m*t(i,19):b10=b9+p*t(i,20)
a11=a10+m*t(i,21):b11=b10+p*t(i,22):a12=a
11+m*t(i,23):b12=b11+p*t(i,24):a13=a12+m*
t(i,25):b13=b12+p*t(i,26):a14=a13+m*t(i,2
7):b14=b13+p*t(i,28)
a15=a14+m*t(i,29):b15=b14+p*t(i,30):a16=a
15+m*t(i,31):b16=b15+p*t(i,32):a17=a16+m*
t(i,33):b17=b16+p*t(i,34)
SHAPE a1;b1,a2;b2,a3;b3,a4;b4,a5;b5,a6;b6
,a7;b7,a8;b8,a9;b9,a10;b10,a11;b11,a12;b1
2,a13;b13,a14;b14,a15;b15,a16;b16,a17;b17
COLOR(co) FILL WITH 4
m(1)=1:m(2)=1:m(3)=4:RETURN

DATA 0,9,1,1,1,-1,0,-2,5,0,0,-2,-5,0,0,-3
,5,0,0,-2,-5,0,0,-1,-2,0,0,0,0,0,0,0
DATA 0,8,11,0,0,-4,4,0,0,-4,-2,0,-2,-2,-1
,0,-2,2,-2,0,-2,-2,-1,0,-2,2,-1,0,0,0,0
,0,0
DATA 1,3,-3,2,2,0,2,1,2,2,-3,0,3,2,1,2,1,
-4,3,-1,-3,-1,-3,-3,-2,-3,2,0,-3,0,0,0,0
,0
DATA 3,2,-2,3,3,-2,3,2,-1,-3,2,-2,-3,0,-2
,-2,0,2,-2,0,0,0,0,0,0,0,0,0,0,0,0,0
DATA 0,2,14,0,0,-2,-14,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
DATA 0,10,3,0,0,-10,-3,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

DATA 2,1,0,5,-3,-3,-1,1,4,4,-2,3,8,0,-2,-
3,4,-4,-1,-1,-3,3,0,-5,2,-1,-3,0,-1,3,-1,
-3,-3,0
DATA 2,3,0,4,-3,-2,2,4,-2,-2,3,3,3,-3,-2,
2,2,-4,-3,2,0,-4,2,-3,-4,0,0,0,0,0,0,0,0
,0
DATA 4,4,3,-3,1,1,1,-1,3,3,3,-3,-3,2,-3,-
3,1,-1,-5,0,1,1,-3,3,-3,-3,0,0,0,0,0,0,0
,0
DATA 0,6,2,2,5,0,0,2,2,0,0,-2,2,0,2,-2,0,
-6,-3,0,0,4,-3,0,0,-4,-7,0,0,0,0,0,0,0,0
DATA 3,7,6,0,2,-3,3,-1,3,-2,-3,0,-2,-2,-2
,2,-5,0,-2,-2,-2,2,-2,0,0,0,0,0,0,0,0,0
,0

```

Von Koch

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : dessin.

Difficulté du jeu : facile.

LE JEU :

Il s'inspire d'une définition de courbe fractale, proposée par le mathématicien Von Koch : un segment de droite est divisé en 3 parties égales ; sur la partie du milieu, on construit une pointe formée de 2 segments de même longueur que les précédents.

On peut ensuite recommencer l'opération sur chacun des 4 segments obtenus, et ceci indéfiniment. On obtient ainsi une courbe très déchiquetée, dite courbe fractale, dont la longueur peut être aussi grande que l'on veut.

LE PROGRAMME GW-BASIC :

En ligne 30, 3 tables X, Y et Z sont prévues pour enregistrer les données de chaque dessin, la table V devant contenir les emplacements des points de chaque segment après subdivision. La courbe à fractaliser est tirée au hasard.

Le niveau de fractalisation est entré en ligne 50. La ligne 60 gère la fin de partie.

Les données sont lues aux lignes 80 à 110.

La boucle I (120 à 230) gère les dessins des segments, la fractalisation étant effectuée par la boucle J (140-190) et la représentation graphique étant effectuée par une autre boucle J (200-220).

Les lignes 240 à 260 gèrent la fractalisation suivante.

POUR JOUER :

On lance le programme par RUN. Puis on choisit le niveau de fractalisation (de 0 à 5). La courbe est alors dessinée (ce peut être assez long avec un niveau de fractalisation élevé).

On tape F pour sortir du programme.

EXTENSIONS POSSIBLES :

On pourra définir d'autres courbes, d'autres méthodes de fractalisation. On peut améliorer la précision du tracé (due aux imperfections de DRAW en GW-Basic).

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM VON KOCH
20 SCREEN 2:KEY OFF:RANDOMIZE TIMER:CLS
30 DIM V(2000),X(30),Y(30),Z(30):X=INT(R
ND*3+1)
40 LOCATE 1,1:PRINT "NIVEAU DE FRACTALIS
ATION ( F pour Fin ) : "
50 R$=INPUT$(1):F=VAL(CHR$(ASC(R$)+32*(R
$>"a")))
60 LOCATE 1,43:PRINT F:IF R$="F" THEN PL
AY "CEG":END
70 RESTORE 270:IF X=2 THEN RESTORE 280 E
LSE IF X=3 THEN RESTORE 300
80 READ N,A,B,C:DRAW "BM=A; ,=B;"
90 FOR I=1 TO N
100 READ X(I),Y(I)
110 NEXT I
120 FOR I=1 TO N
130 Z(I)=Y(I)/(3^F):V(1)=X(I):V(2)=X(I)-
60:V(3)=X(I)+60:V(4)=X(I)
140 FOR J=5 TO 4^F
150 IF J MOD 4=1 THEN V(J)=V(INT(J/4)+1)
160 IF J MOD 4=2 THEN V(J)=V(J-1)-60
170 IF J MOD 4=3 THEN V(J)=V(J-2)+60
180 IF J MOD 4=0 THEN V(J)=V(J-3)
190 NEXT J
200 FOR J=1 TO 4^F
210 A=(V(J)+C) MOD 360:B=Z(I):DRAW "TA=A
;U=B;"
220 NEXT J
230 NEXT I
240 LOCATE 2,1:PRINT "APPUYEZ UNE TOUCHE
":R$=INPUT$(1)
250 R$=CHR$(ASC(R$)+32*(R$>"a"))
260 IF R$<>"F" THEN CLS:GOTO 40 ELSE PLA
Y "CEG":END
270 DATA 3,400,170,0,0,150,120,150,240,1
50
280 DATA 18,120,70,-90,45,10,135,10,0,20
,30,20,135,20,45,5,315,25,30,50
290 DATA 330,50,290,50,260,60,170,50,200
,25,250,25,170,40,80,55,100,45,180,20
300 DATA 14,120,70,-90,45,25,0,75,90,15,
0,15,270,15,0,50,315,25,270,100
310 DATA 180,75,90,45,180,25,270,45,180,
75,90,100

```

```

REM Von Koch

STREAM #1:SCREEN GRAPHICS
WINDOW FULL ON
WINDOW TITLE "VON KOCH"
WINDOW OPEN
GRAPHICS CURSOR 3:RANDOMIZE TIME:OPTION D
EGREES:DIM v(2000),x(30),y(30),z(30):x=RN
D(3)
LABEL debut
PRINT AT (1;1) ANGLE(0) "Niveau de fracta
lisation (F pour fin) : "
f=VAL(INPUT$(1)):PRINT AT (43;1) ANGLE(0)
RESTORE triangle:IF x=2 THEN RESTORE fran
ce ELSE IF x=3 THEN RESTORE maison
PRINT AT (1;1);:READ n,a,b:POINT a:MOVE F
D b
IF UPPER$(f)="F" THEN END
FOR i=1 TO n
READ x(i),y(i)
NEXT i
FOR i=1 TO n
z(i)=y(i)/(3^f):v(1)=x(i):v(2)=x(i)-60:v(
3)=x(i)+60:v(4)=x(i)
FOR j=5 TO 4^f
IF j MOD 4=1 THEN v(j)=v(INT(j/4)+1)
IF j MOD 4=2 THEN v(j)=v(j-1)-60
IF j MOD 4=3 THEN v(j)=v(j-2)+60
IF j MOD 4=0 THEN v(j)=v(j-3)
NEXT j
FOR j=1 TO 4^f
POINT v(j) MOD 360:FD z(i)
NEXT j
NEXT i
PRINT AT (1;2) ANGLE(0) "Appuyez sur une
touche":r$=UPPER$(INPUT$(1))
IF r$<>"F" THEN CLS RESET:GOTO debut ELSE
END

LABEL triangle
DATA 3,305,4000,0,3000,120,3000,240,3000

LABEL france
DATA 18,320,3000,45,200,135,200,0,400,30,
400,135,400,45,100,315,500,30,1000
DATA 330,1000,290,1000,260,1200,170,1000,
200,500,250,500,170,800,80,1100,100,900,1
80,400

LABEL maison
DATA 14,330,2600,45,500,0,1500,90,300,0,3
00,270,300,0,1000,315,500,270,2000
DATA 180,1500,90,900,180,500,270,900,180,
1500,90,2000

```

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 GW-BASIC :

Les lignes 30 à 50 définissent le caractère plein.

La boucle I (70-100) dessine les verres.

Les deux boucles emboîtées I (190-230) et J (200-220) les remplissent avec la quantité indiquée en ligne 180.

La variation de taille du lézard est définie par la ligne 270, avec contrôle par les lignes 280 et 290.

La boucle I (300-320) dessine le corps du lézard, les lignes 330 à 380 dessinent l'œil et les pattes.

Un message gagnant sera affiché par la ligne 400 si l'on a trouvé la bonne taille.

POUR JOUER :

On lance le programme par RUN : les deux verres se dessinent, puis apparaît la question QUEL COTE à laquelle on répond en tapant G ou D ; on obtient l'affichage QUELLE QUANTITE ; on entre alors une quantité (de 0 à 130) et on frappe RETURN.

Le verre désigné se remplit sous vos yeux, puis, après quelques instants, le lézard est affiché. On devra alors frapper une touche quelconque ; si le lézard n'avait pas la bonne taille, aucun message n'est affiché et l'on revient au dessin des verres.

EXTENSIONS POSSIBLES :

Puisqu'il s'agit d'un jeu essentiellement graphique, il sera intéressant d'améliorer les dessins et 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.

VERSION BASIC 2 AMSTRAD :

L'utilisation se fait de la même manière qu'en GW-Basic.

```

10 REM LE LEZARD D'ALICE
20 SCREEN 1,1:COLOR 0:RANDOMIZE TIMER:KEY OFF:CLEAR,49151!
30 DEF SEG=0:POKE 124,0:POKE 125,192:POKE 126,PEEK(1296):POKE 127,PEEK(1297):DEF SEG
40 FOR I=49152! TO 49159!:POKE I,255:NEXT I
50 CLS:C$=CHR$(128)
60 N=18
70 FOR I=2 TO 10
80 LOCATE I+4,4:PRINT C$:LOCATE I+4,11:PRINT C$
90 LOCATE I+4,14:PRINT C$:LOCATE I+4,21:PRINT C$
100 NEXT I
110 FOR I=4 TO 11:LOCATE 15,I:PRINT C$:NEXT I
120 FOR I=14 TO 21:LOCATE 15,I:PRINT C$:NEXT I
130 LOCATE 16,5:PRINT"GAUCHE":LOCATE 16,15:PRINT"DROIT"
140 LOCATE 18,1:PRINT "QUEL COTE : "
150 A$=INKEY$+" ":A$=CHR$(ASC(A$)+32*(A$>"a")):IF A$<>"D" AND A$<>"G" THEN 150
160 U=2-20*(A$="D")
170 LOCATE 18,13:IF A$="D" THEN PRINT"DR OIT" ELSE PRINT"GAUCHE"
180 PRINT:INPUT"QUELLE QUANTITE :":Q
190 FOR I=1 TO 4*SQR(Q)
200 FOR J=0 TO 47
210 PSET (4*U+J+24,112-I),1:SOUND 24*I+INT(J/2)+40,..1
220 NEXT J
230 NEXT I:PLAY"CDE"
240 A$=INKEY$:IF A$="" THEN 240
250 CLS
260 H=2*(RND<.5)-1
270 N=N+H*INT(SQR(Q))
280 IF N<12 THEN N=12
290 IF N>60 THEN N=60
300 FOR I=0 TO 4*(N+3)
310 PSET(I,100),1:PSET(I,99),1:PSET(I,98),1:PSET(I,97),1:SOUND 300+8*I,..5
320 NEXT I
330 T=4*(N-INT((SQR(N)/2)))
340 PSET(T,88),2:PSET(T-1,89),2:PSET(T-2,90),2:PSET(T+1,89),2:PSET(T+2,90),2
350 T=4*SQR(N):PSET(T,102),2:PSET(T-1,103),2:PSET(T-2,104),2
360 T=T+2:PSET(T,102),2:PSET(T+1,103),2:PSET(T+2,104),2
370 T=4*(N-SQR(N)):PSET(T,102),2:PSET(T-1,103),2:PSET(T-2,104),2
380 T=T+2:PSET(T,102),2:PSET(T+1,103),2:PSET(T+2,104),2
390 A$=INKEY$:IF A$="" THEN 390
400 IF N=28 THEN PRINT"BRAVO":PLAY"CEG":END
410 GOTO 50

```

```

REM Le lézard d'Alice

STREAM #1:SCREEN GRAPHICS 640 FIXED,200 FIXED
USER SPACE 640,200
WINDOW FULL ON
WINDOW TITLE "LE LEZARD D'ALICE"
WINDOW OPEN
RANDOMIZE TIME:k=2:n=18
LABEL début:CLS
x1=24:y1=80:x2=32:y2=140:GOSUB rectangle
x1=24+50*k:y1=80:x2=32+50*k:y2=140:GOSUB rectangle
x1=64+50*k:y1=80:x2=72+50*k:y2=140:GOSUB rectangle
x1=64+100*k:y1=80:x2=72+100*k:y2=140:GOSUB rectangle
B rectangle
x1=24:y1=72:x2=32+50*k:y2=80:GOSUB rectangle
x1=64+50*k:y1=72:x2=72+100*k:y2=80:GOSUB rectangle
PRINT AT (8;16) "GAUCHE" AT (25;16) "DROIT"
PRINT AT (1;1) "Quel côté : ";
LABEL choix
a$=UPPER$(INKEY$):IF a$<>"D" AND a$<>"G" GOTO choix
u=8-(140*(a$="D"))
IF a$="D" THEN PRINT "Droit" ELSE PRINT "Gauche"
PRINT AT (1;3) "Quelle quantité : ";a$=" "
LABEL quantité
r$=UPPER$(INKEY$):IF r$<>CHR$(13) THEN a$=a$+r$:PRINT r$;GOTO quantité
FOR i=1 TO 4*SQR(VAL(a$))
LINE u+24;79+i,u+115;79+i COLOR(4):PRINT CHR$(7);
NEXT i
PRINT AT (20;19) "Appuyez sur une touche"
:r$=INPUT$(1)
CLS:BOX 0;0,640,200 COLOR(6) FILL WITH 8
h=-2*(RND<.5)-1:n=n+h*INT(SQR(q))
IF n<12 OR n>60 THEN n=-12*(n<12)-60*(n>60)
FOR i=0 TO 8*(n+3)
PLOT i;100,i;99,i;98,i;97 COLOR(2)
NEXT i
t=8*(n-INT((SQR(n)/2))):PLOT t-2;105,t-1;106,t;107,t+1;106,t+2;105
t=8*SQR(n):PLOT t;95,t-1;94,t-2;93
t=t+2:PLOT t;95,t+1;94,t+2;93
t=8*(n-SQR(n)):PLOT t;95,t-1;94,t-2;93
t=t+2:PLOT t;95,t+1;94,t+2;93
PRINT CHR$(7);CHR$(7);CHR$(7);
PRINT AT (1;19) "Appuyez sur une touche"
r$=INPUT$(1):IF n<>28 GOTO début
PRINT AT (1;19) POINTS(20) "BRAVO....."
...."
r$=INPUT$(1):END

LABEL rectangle
BOX x1;y1,x2-x1,y2-y1 COLOR(7)FILL WITH 4
RETURN

```


Conseils de lecture

POUR MAITRISER LE BASIC MICROSOFT

- **Le Basic Microsoft** - Ken Knecht.
Ouvrage de référence de la version 5.0. Un livre très utile aux débutants complets.
- **Techniques de Basic sur IBM/PC et compatibles** - John P. Grillo.
Ouvrage de perfectionnement traitant, notamment, des fichiers à accès direct ou séquentiel.
- **Logic Basic, tomes 1 et 2** - Patrick Sénicourt et Michel Massiou.
Tome 1 : une méthode de programmation structurée, illustrée d'exemples concrets.
Tome 2 : une étude très poussée de la programmation structurée des fichiers en Basic Microsoft.
- **Dictionnaire du Basic IBM** - David A. Lien.
Chaque mot Basic, illustré par un programme testé, est présenté dans son contexte, dans l'ordre le plus logique, l'ordre alphabétique.
- **Le Basic et ses fichiers, tomes 1 et 2** - Jacques Boisgontier.
Tome 1 : la programmation à applications utilisant des fichiers sur disquettes ou sur disque dur, en Basic Microsoft 5.0.
Tome 2 : étude poussée des méthodes d'accès par clé sur IBM/PC et compatibles.
- **La gestion de données et de fichiers sur IBM/PC et compatibles** - John P. Grillo et J.D. Robertson.
Techniques de manipulation de fichiers avec explications sur les pointeurs, les listes chaînées, les arbres, etc.

POUR MIEUX UTILISER VOTRE PC :

- **Au cœur de l'IBM/PC, logique et fonctionnement interne** - Jeanine Thièle.
Pour les utilisateurs avertis, une présentation détaillée des différents éléments constituant l'IBM/PC, et de leurs interruptions.
- **Systèmes PC-DOS et MS-DOS, versions 2.1 à 4** - Jacques Boyer, Michel Treillet, Jean-Pierre Lamoitier.
Introduction et utilisation avancée aux systèmes MS-DOS et PC-DOS (dernières versions), et aux réseaux.
- **MS-DOS pas à pas, versions 2 à 4** - Alain Pinaud.
Après la définition du système sont vus les commandes essentielles, les nouveautés des versions 3.2 à 4 et les logiciels d'accompagnement.
- **Clefs pour MS-DOS, versions 2 à 4** - Yvon Dargery
Mémento de référence du système MS-DOS : le livre de chevet de l'utilisateur d'un PC.

- **Le livre du MS/PC-DOS pour PC et AT** - François Piette.
Tout sur le fonctionnement interne des ordinateurs IBM/PC et compatibles pour les utilisateurs avertis ; comment tirer le maximum des PC, AT, AMSTRAD-PC, et de leur système d'exploitation.

 Achevé d'imprimer par l'imprimerie Tardy Quercy S.A. - Cahors
Dépôt légal : Janvier 1989 - N° d'Imprimeur : 90254F - N° d'Édition : 86595-458-5

ISBN : 2-86595-458-7



A VOTRE SERVICE

DISQUETTE D'ACCOMPAGNEMENT

Certains ouvrages font l'objet d'une disquette d'accompagnement reprenant les programmes contenus dans le livre ou les applications associés. Pour les obtenir, reportez-vous à la page "disquette d'accompagnement" insérée au début de chaque ouvrage possédant une disquette.

EN COMPOSANT LE 3615 CODE OI * LIV NOTRE SERVICE MINITEL VOUS PROPOSE

- de vous renseigner sur notre catalogue et toutes nos nouveautés
- de vous indiquer le plus proche point de vente
- de répondre à vos questions techniques concernant nos ouvrages grâce à la messagerie P.S.I.

CATALOGUES ET "LIVRES MICRO"

Vous pouvez recevoir chez-vous les catalogues complets de nos ouvrages et être abonné gratuitement à la revue "Livres Micro". Pour ce faire, envoyez le coupon ci-dessous à :

Editions P.S.I. 6-10 boulevard Jourdan - 75014 PARIS



VOTRE AVIS NOUS INTERESSE

Je désire recevoir gratuitement : ☐ vos catalogues ☐ la revue "Livres Micro"

Pour nous permettre de faire de meilleurs livres, adressez-nous vos critiques et vos suggestions sur le présent ouvrage.

Titre de l'ouvrage : _____

- Ce livre vous donne t-il toute satisfaction ?

- Y a t-il un aspect du problème que vous auriez aimé voir aborder ?

Nom _____ Prénom _____ Age _____

Adresse _____

Profession _____

Centre d'intérêt : ☐ PC

☐ Macintosh

☐ Autre



Apprendre en se distrayant, tel est l'objectif de ce livre. Ces 102 programmes de jeux vous guideront progressivement dans l'exploration de GW-Basic et Basic2 sur PC, Amstrad PC et compatibles.

Niveau par niveau vous assimilerez de nouvelles instructions pour arriver à une plus grande maîtrise du Basic. Chaque niveau commence par une présentation pédagogique du jeu d'instructions utilisé.

Tous les jeux sont décrits et les programmes sont analysés afin d'en faciliter la modification. Pour chaque jeu, vous disposez du programme en GW-basic et en Basic-2.

**DES LIVRES
POUR VOTRE
ORDINATEUR**

PSi.

ISBN: 2-86595-458-7
135 FF 500 140



Couverture GIBRAT

