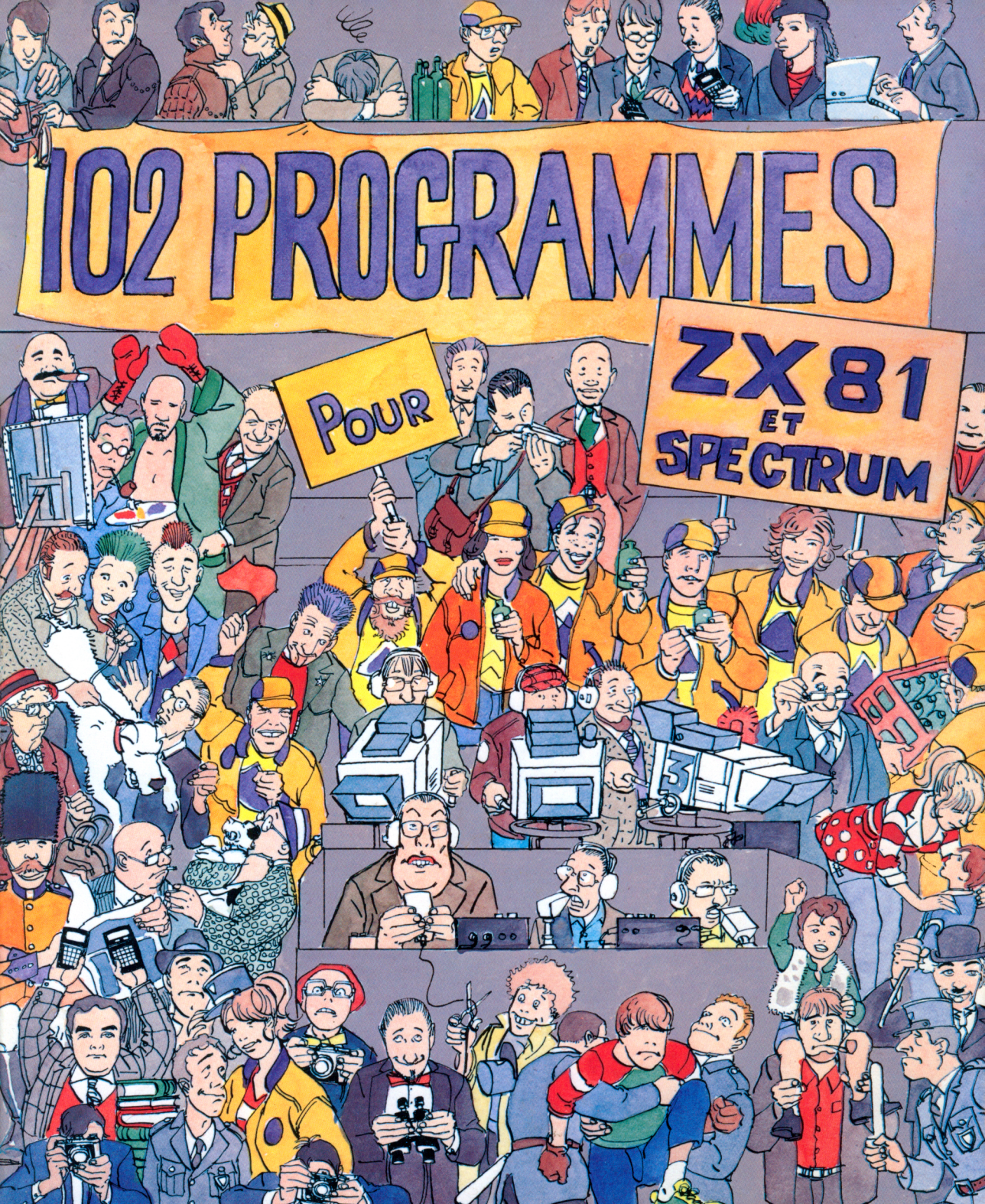




PAR
JACQUES
DECONCHAT



102 PROGRAMMES POUR
ZX-81 et
SPECTRUM

Autres ouvrages relatifs au matériel Sinclair

- Le petit livre du ZX 81, Trévor Toms.
- La pratique du ZX 81, tome 1, Xavier Linant de Bellefonds.
- La pratique du ZX 81, tome 2, Marcel Henrot.
- Etudes pour ZX 81, tomes 1 et 2, Jean-François Séhan.
- Clefs pour le ZX 81, Jean-François Séhan.
- Le petit livre du Spectrum, Trévor Toms.
- Spectrum pour tous, Jacques Boigontier et Marcel Henrot.
- La pratique du ZX Spectrum, tome 1, Xavier Linant de Bellefonds.
- La pratique du ZX Spectrum, tome 2, Marcel Henrot.
- Le Spectrum à l'affiche, Jean-François Séhan.
- Exercices pour ZX Spectrum, Julien Lévy.
- Clefs pour le ZX Spectrum, Jean-François Séhan.

Ouvrages d'initiation

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

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

© Editions du P.S.I. B.P. 86, 77402 LAGNY CEDEX, 1984

ISBN : 2-86595-095-6

**JACQUES
DECONCHAT**

**102 PROGRAMMES POUR
ZX-81 et
SPECTRUM**



EDITIONS DU P.S.I.

1984

SOMMAIRE

Avant-propos	8
Organisation de l'ouvrage	9

Niveau 1

Instructions élémentaires	15
Le nombre mystérieux	20
Cube	22
Chaud, froid	24
Les allumettes	26
Allumettes à plusieurs	28
Calendrier perpétuel	30
Bataille navale	32
Golf	34
Solfège	36
Le jeu du 21	38
Pair - Impair	40
Devine un chiffre	42
Le jeu de Witthof	44
Rebonds	46
Jackpot	48
Roulette	50
421	52
Simon	54
Jeu de la fève	56
Jeu de l'oie aléatoire	58

Niveau 2

Dimensionnement d'un tableau - Recherche d'un élément dans un tableau et dans une chaîne	62
Un dé	64
Visions	66
Plus grand, plus petit	68
L'électricien fou	70
La course au 20	72
Dessin	74
Dé truqué	76

Le jeu de saute-mouton	78
Lettres	80
Espion	82
Accrochez les wagons	84
Tirage d'un dé truqué	86
Ordre	88
Course de chevaux	90
Le jeu de Marienbad	92
Idem	94
Mastermind	96
A travers	98
Reines sur l'échiquier	100
Le jeu de la vie	102

Niveau 3

Manipulation des chaînes et graphiques haute résolution	106
Losanges	108
Echiquier	110
Couloirs	112
Jour de la semaine	114
Le pendu	116
Bridge	118
Horloge 1	120
Le loup et les agneaux	122
Belote	124
Un taquin de calcul	126
Poker	128
Hanoï	130
Musique	132
Pendu inverse	134
Mathieu	136
Machinamot	138
Compose	140
Nicomaque	142
Permutation	144
Tour de cartes	146

Niveau 4

Les techniques d'animation élémentaire	150
Tirage de dés	152
Jeu de loto	154
La bombe	156
Ardoise magique	158
Affichage publicitaire	160
Evasion	162

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

Niveau 5

Inscription directe dans les mémoires	194
Dames	195
Smokey	198
Course de voitures	200
Echecs	202
Danger spatial	204
Tic Tac Toe	206
Horloge 2	208
Anagrammes	210
Œufs	212
Affichage tournant (vertical)	214
Taquin de lettres	216
Cibles	218
Gouttes	220
Obstacles	222
Captures dans l'espace	224
Affichage publicitaire (2)	226
Anomalie	228
Plusieurs en un	230
Parcours du cavalier	232
Mur de briques	234
Othello	236
Quatre à quatre	238
 ANNEXE	 241

Avant-propos

Vous trouverez dans ce livre plus de 100 programmes, rédigés en BASIC SINCLAIR, écrits et présentés à chaque fois dans deux versions : l'une pour le ZX-81 1K octet (utilisable la plupart du temps sur ZX-80 New Rom, sans modification), et l'autre pour le ZX-Spectrum (16K dans la version de base).

Le fait de ne présenter que des programmes très courts (moins de 1K octet dans la version ZX-81, pour tous les programmes) correspond ici à un choix : d'une part le désir de guider le lecteur dans un apprentissage progressif du langage, en utilisant pour cela des jeux courts, mais néanmoins assez complets et performants, d'autre part la volonté de ne pas tomber dans le piège classique de programmes trop longs, très difficiles à entrer en machine sans erreur, et qui sont dès lors plus à l'aise sur des cassettes préenregistrées que l'on trouvera par ailleurs de plus en plus couramment dans le commerce.

Chaque niveau de programme (il y en a 5) est précédé d'une courte étude des instructions Basic mises en œuvre dans les jeux qui suivent. Les différences entre ZX-81 et ZX-Spectrum, ainsi que les différences avec les autres Basic, sont mises en évidence.

De plus les programmes proposés sont "ouverts", ce qui signifie qu'ils ont été conçus, dans la mesure du possible, pour être facilement utilisés et modifiés, si la taille mémoire dont vous disposez est supérieure : par exemple un jeu comme Othello, qui ne peut être pratiqué que sur un damier 6×6 dans la version 1K, pourra sans aucune modification être pratiqué sur damier 8×8 ou 10×10 sur une version 16K ou plus (tous les jeux sont utilisables en 16K, sans modification).

Des indications sont fournies à la fin de chaque programme, pour vous suggérer des améliorations possibles, ou des modifications qui pourraient rendre le jeu 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, PRINT, INPUT, IF...THEN..., FOR...NEXT, GOTO, COSUB.**
- **Au niveau 2**, on utilisera les instructions de confection et de manipulation des éléments d'un tableau : **DIM, R(I).**
- **Au niveau 3**, on fait appel à la puissante instruction de traitement des chaînes de caractères : **R\$(TO)** et aux instructions graphiques : **PLOT, UNPLOT.**
- **Au niveau 4**, on rencontre des instructions propres aux jeux interactifs : **INKEYS** et **SCROLL.**
- Enfin, **le niveau 5** nous fait découvrir les instructions frontières du Basic, qui permettent la manipulation directe dans les mémoires de l'appareil : **PEEK** et **POKE.**

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

Les programmes ZX-81 et ZX-Spectrum sont donnés en vis-à-vis, et le numérotage des lignes est dans la mesure du possible le même dans les deux versions ; l'intérêt de cette formule est multiple : ceci devrait permettre aux anciens utilisateurs de ZX-81, désireux de passer au ZX-Spectrum, de très vite cerner les similitudes et les différences (minimes) entre les deux appareils ; cela permettra d'autre part de lever tout doute d'interprétation dans la lecture d'un programme (impression un peu floue, par exemple) et de mieux comprendre la signification des codes employés sur la version ZX-81, pour diminuer l'occupation mémoire.

Pour augmenter la facilité de transposition des programmes du ZX-81 au ZX-Spectrum, ce dernier a été utilisé systématiquement en position C A P S L O C K (blocage des touches en position majuscule) ; on se prive ainsi volontairement des minuscules, présentes sur le Spectrum ; il sera facile de modifier les programmes pour les utiliser.

Il existe par ailleurs quelques différences marquantes entre le Basic du ZX-81, et celui du ZX-Spectrum. Les différences seront généralement faciles à contourner, et la lecture comparée des programmes permettra de trouver aisément les diverses méthodes retenues pour remédier à cela.

DIFFERENCES LES PLUS NOTABLES

- Présence de deux modes, FAST et SLOW, sur le ZX-81, et rien sur le Spectrum. Aucun inconvénient pratique, si ce n'est la nécessité de rajouter dans quelques programmes une temporisation pour ralentir le Spectrum, qui fonctionne suivant le mode SLOW du 81, mais à la vitesse du mode FAST.
- Présence d'une instruction SCROLL sur le ZX-81. On y remédie, sur le Spectrum, en annulant le SCROLL ? qui apparaît automatiquement si l'écran est plein, et en provoquant l'impression d'un caractère retour chariot (CHR\$(13)) sur la 21^e ligne : on écrira PRINT AT 20,0:POKE 23692,0:PRINT CHR\$(13).
- Possibilité d'écrire plusieurs instructions par ligne sur le ZX-Spectrum, séparées par 2 points.
- Possibilité d'imprimer un texte à l'aide d'une instruction INPUT, et d'attendre plusieurs variables, sur Spectrum.
- Possibilité de lire directement le caractère contenu à l'emplacement X, Y de l'écran, sur le Spectrum ; l'instruction SCREEN\$(X,Y) remplacera donc les deux lignes :

```
PRINT AT X, Y ;  
LET C$=CHR$ PEEK(PEEK 16398+256×PEEK 16399)
```

du ZX-81.

Pour la couleur, il faudra utiliser l'instruction **ATTR**, d'un maniement plus délicat.

- Changement des codes des différents caractères, le ZX-Spectrum utilisant le code ASCII, ce qui n'était pas le cas du ZX-81.
- Certains caractères graphiques du ZX-81 ne sont plus disponibles sur le Spectrum (on pourra les remplacer par de la couleur).
- Les instructions **READ**, **RESTORE** et **DATA**, qui n'existaient pas sur le ZX-81, sont présentes sur le ZX-Spectrum (elles ne sont que très rarement utilisées dans cet ouvrage, pour respecter l'analogie avec le ZX-81).
- L'instruction **PLOT** du ZX-Spectrum permet d'adresser 256×176 points, au lieu de 64×44 sur le 81, ce qui permet un graphique haute résolution, mais complique la traduction de quelques programmes.
- L'instruction **UNPLOT** est remplacée sur le Spectrum par **PLOT OVER1**.

Il existe beaucoup d'autres différences, mais que nous n'aurons pratiquement pas l'occasion de rencontrer au cours de notre étude. Il semble important de bien préciser ici que les analogies restent beaucoup plus nombreuses que les différences, et que les programmes écrits pour le ZX-81 resteront, sauf cas très particulier, très facilement adaptables pour le ZX-Spectrum ; par contre, la vitesse de chargement et de sauvegarde des programmes sur cassette n'est pas la même, et les programmes cassette du ZX-81 ne seront pas directement utilisables sur le Spectrum.

Un certain nombre de techniques ont été utilisées pour limiter l'occupation de la mémoire, notamment sur la version ZX-81. Lorsque ces techniques font appel aux propriétés particulières du Basic Sinclair (et c'est souvent le cas), elles sont directement utilisables sur le ZX-Spectrum, et vous les découvrirez avec intérêt en étudiant les programmes proposées (utilisation de fonctions logiques pour supprimer des tests, utilisation de GOTO et GOSUB calculés, etc.). Par contre, dans certains cas, le procédé utilisé, s'il économise de la mémoire, rend le programme plus difficile à interpréter ; dans ce cas là, le codage n'a pas été retenu pour la version Spectrum, ce qui permettra de comprendre le sens des symboles employés, en jetant un simple coup d'œil sur la version ZX-Spectrum. Ainsi, pour le ZX-81, 0 est remplacé par NOT PI, 1 par SGN PI et 3 par INT PI ; cette technique, bien que rentable aussi sur le Spectrum, n'a pas été retenu pour les raisons indiquées plus haut. De même une grande quantité de nombres sont remplacés par les codes caractères correspondants, sur le ZX-81. Ainsi GOTO 200 est fréquemment écrit GOTO CODE "COS", PRINT AT 16, 15 sera écrit PRINT AT CODE "(" , CODE "?" ; etc.

Niveau 1

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

Instructions élémentaires

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

- *CLS pour nettoyer l'écran.*
- *PAUSE pour arrêter momentanément le déroulement du programme.*
- *RND pour obtenir un nombre aléatoire entre 0 et 1.*
- *INT pour prendre la partie entière du résultat.*

etc.

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

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

soit seulement sept instructions de base 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.

Dans la plupart des Basic, l'instruction LET peut être omise, ce qui facilite l'écriture : A = 12 ; B\$ = "TOTO", etc.

L'affectation une fois réalisée, le micro-ordinateur saura, chaque fois qu'on lui parle de A1, qu'il s'agit du nombre 12 ; si on parle de NOM\$, du mot RENOIR, 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 le micro-ordinateur pour faire connaître ses résultats. Ainsi, PRINT A1 provoque l'affichage de 12 (si l'on a fait auparavant LET A1 = 12), PRINT NOM\$ provoque l'affichage de RENOIR (si l'on a fait LET NOM\$ = "RENOIR"). Utilisé après un effacement d'écran (CLS) le PRINT provoque l'affichage de la 1^{re} ligne, en haut, à gauche.

Divers paramètres seront utilisés pour indiquer au ZX où il doit afficher ses résultats ou ses symboles (y compris les dessins) :

- **PRINT TAB(N)** provoque sur la ligne d'impression courante un affichage commençant à la Nième colonne à partir de la gauche ($0 \leq N \leq 31$).
- **PRINT AT (X, Y)** provoque un affichage débutant ligne X (à partir du haut, avec $0 \leq x \leq 19$) et colonne Y (à partir de la gauche avec $0 \leq Y \leq 31$).
- Un ; (**point virgule**) en fin de PRINT permet soit d'enchaîner une autre impression sans répéter l'instruction PRINT, soit d'indiquer que la prochaine impression sera, en l'absence d'indication contraire, effectuée exactement à la suite de l'impression précédente.
- La , (**virgule**) provoque une tabulation horizontale automatique pré-définie.

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

Cette instruction permettra d'entrer et d'effectuer directement, dans la variable désignée par le INPUT, un nombre ou un texte et ceci en mode dit conversationnel, c'est-à-dire au fur et à mesure des besoins ; en effet, lorsque le ZX rencontre une instruction INPUT, il s'arrête et vous rend la main ; on peut alors entrer une information de nature convenable, et lui signaler que cette entrée est terminée, à l'aide de la touche **NEW-LINE**, qui rendra la main au ZX.

Par exemple, si, dans le programme, le ZX trouve INPUT B, il s'arrête, et vous devez taper un nombre B quelconque, suivi de NEWLINE (si vous tapez un caractère autre qu'un chiffre ou un point décimal, vous aurez un message d'erreur : il est prudent de prévoir l'affichage d'un texte avant l'arrêt, vous précisant ce que vous devez entrer).

Sur d'autres appareils, ainsi que sur le Spectrum, il existe d'autres moyens d'entrer des informations (LET, DATA, etc.), mais ils n'ont pas la souplesse de INPUT, puisque l'information est alors figée dans le programme, et n'est que difficilement modifiable.

Sur ZX-81, l'instruction INPUT est très simplifiée : on ne peut mentionner par un texte le type de l'information, ni entrer plusieurs informations les unes à la suite des autres ; sur le ZX-Spectrum, l'instruction INPUT est similaire à l'INPUT des autres Basic, qui permet d'afficher un texte, et d'entrer successivement plusieurs variables si nécessaire.

L'INSTRUCTION GOTO OU INSTRUCTION DE RENVOI INCONDITIONNEL

Une des particularités du langage Basic, qui le rend facilement accessible, est que *chaque ligne de programme doit être précédée d'un numéro*, ces lignes étant ensuite exécutées par le micro-ordinateur 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 Basic, et c'est le cas du Basic Sinclair, n'autorise en général que les renvois à un numéro de ligne connu (GOTO 20, par exemple, ou GOTO 200, etc.).

Mais une des particularités les plus remarquables du Basic Sinclair est qu'il permet aussi le GOTO calculé, ce qui signifie que, derrière l'instruction GOTO, on pourra mettre une variable, ou même une expression arithmétique ou logique à calculer ; ce qui fait de cette instruction une instruction de renvoi conditionnel très puissante, analogue au CASE OF du Pascal ; on pourra écrire,

Par exemple :

GOTO(100 AND I = 1)+(200 AND I = 2)+(300 AND I = 3)
ou GOTO I × 100, qui envoie aux lignes 100, 200 ou 300, selon la valeur de I.

L'INSTRUCTION IF___ THEN___ OU INSTRUCTION DE TEST

L'instruction IF___ THEN___, très puissante dans le Basic Sinclair, pourra être utilisée de plusieurs façons :

– soit en rupture de séquence conditionnelle, sous la forme :

IF condition = vraie THEN GOTO nnn numéro de ligne.

C'est l'usage le plus courant de ce test, qui pourra toutefois être amélioré ici par le fait que la condition peut être une expression arithmétique ou logique complexe, évaluée logiquement, et que le GOTO peut lui-même être calculé.

– mais on pourra également utiliser le test sous la forme :

IF condition THEN instruction,

l'instruction présente à la suite de THEN pouvant être une affectation LET, une impression PRINT, une entrée INPUT, ou même un autre test IF___ THEN___ ; on peut également écrire par exemple :

IF A THEN IF B THEN PRINT A + B.

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

On sera très souvent amené, en programmation, à vouloir exécuter un nombre entier de fois une certaine suite d'instructions (par exemple, faire avancer un mobile de la case 0 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 sur une quantité quelconque.

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

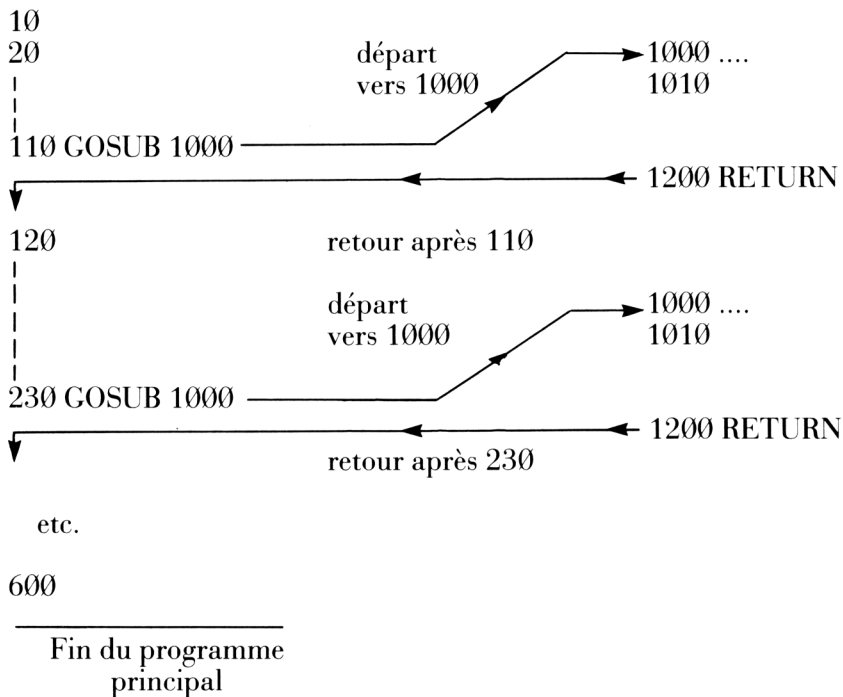
Le Basic ZX permet d'emboîter un nombre indéfini de boucles (limité par la mémoire disponible).

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

Lorsque la même suite d'instructions devra 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



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 sans autre limitation que la taille mémoire, dans le Basic Sinclair.

Il est important de préciser que, dans le Basic Sinclair, comme pour le `GOTO`, le `GOSUB` peut être aussi calculé, ce qui permettra une très grande concision et une très grande souplesse de programmation.

Par exemple : on pourra écrire `GOSUB A × 1000`, et, si $A = 1$, on exécutera le sous-programme 1000, si $A = 2$, le sous-programme 2000, etc.

Les Basic usuels réalisent ceci par l'instruction `ON A GOSUB 1000, 2000, -`, qui est loin de présenter la même souplesse.

Le nombre mystérieux

REFERENCES :

Mode FAST ou SLOW.

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 PLUS GRAND ou PLUS PETIT.

LE PROGRAMME :

La ligne 40 permet de fabriquer le nombre à deviner.

Les entrées sont faites par les lignes 60 à 90.

La ligne 100 teste si le résultat est exact.

La ligne 110 signale un essai trop petit, et la ligne 120 un essai trop grand.

Le compteur d'essais est en ligne 130.

Les lignes 200 à 240 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 NBRE DE CHIFFRES ? auquel on répond par un entier entre 1 et 7, suivi de NEWLINE.

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

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

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 ou N) ? ; on répond O pour faire une autre partie (suivi de NEWLINE).

On peut arrêter à tout moment en entrant une lettre en réponse à la question VOTRE NOMBRE.

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...).

```

5 CLS
10 PRINT AT 3,1;"NBRE DE CHIFF
RES :";
20 INPUT N
25 IF N>7 THEN RUN
30 PRINT N
40 LET X=INT (RND*10**N+1)
50 LET J=1
60 PRINT AT 6,1;"ESSAI NO ";J
70 PRINT AT 8,1;"VOTRE NOMBRE
";
80 INPUT M
90 PRINT M
100 IF M=X THEN GOTO 200
110 PRINT AT 10,1;"TROP PETIT"
120 IF M>X THEN PRINT AT 10,1;"
TROP GRAND"
130 LET J=J+1
140 PAUSE 120
150 PRINT AT 8,14;"
160 PRINT AT 10,1;"
170 GOTO 60
200 CLS
210 PRINT AT 8,0;"VOUS AVEZ TRO
UVE ";X;" EN ";J;" ESSAIS"
220 PRINT AT 12,1;"AUTRE PARTIE
(O OU N)?"
230 INPUT R$
240 IF R$="O" THEN RUN

```

NBRE DE CHIFFRES : 3

ESSAI NO 4

VOTRE NOMBRE 75

TROP PETIT

```

5 CLS
10 PRINT AT 3,1;"NBRE DE CHIFFRES :";
20 INPUT N
25 IF N>7 THEN RUN
30 PRINT N
40 LET X=INT (RND*10^N+1)
50 LET J=1
60 PRINT AT 6,1;"ESSAI NO ";J
70 PRINT AT 8,1;"VOTRE NOMBRE :";
80 INPUT M
90 PRINT M
100 IF M=X THEN GO TO 200
110 IF M<X THEN PRINT PAPER 2; INK 5;AT
10,1;"TROP PETIT": BEEP 1.3,-10
120 IF M>X THEN PRINT PAPER 5; INK 2;AT
10,1;"TROP GRAND": BEEP 1.3,10
130 LET J=J+1
140 PAUSE 120
150 PRINT AT 8,14;"
160 PRINT AT 10,1;"
170 GO TO 60
200 CLS
210 PRINT AT 8,1;"VOUS AVEZ TROUVE ";X;"
EN ";J;" ESSAIS"
220 PRINT FLASH 1; INVERSE 1; INK 4;AT 1
2,1;"AUTRE PARTIE (O OU N)?"
230 INPUT R$
240 IF R$="O" THEN RUN

```

NBRE DE CHIFFRES : 3

ESSAI NO 3

VOTRE NOMBRE : 350

TROP PETIT

Cube

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : aucune.

Catégorie de jeu : dessin.

Difficulté du jeu : aucune.

LE JEU :

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

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

LE PROGRAMME :

Le dessin du cube est réalisé par l'impression de points, en utilisant deux boucles : la boucle I (10-50), et la boucle I (60-90).

Le nom de l'objet représenté est affiché en ligne 100 (ici : CUBE).

La ligne 110 est en pratique inutile, puisque le programme s'arrête lorsqu'il n'y a plus d'instructions à exécuter.

POUR JOUER :

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

On pourra taper n'importe quelle touche pour sortir du programme.

EXTENSIONS POSSIBLES :

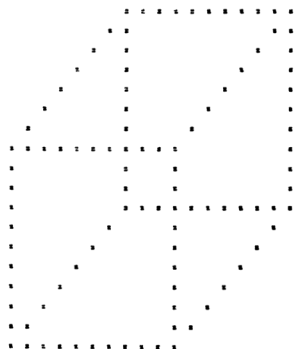
L'instruction PLOT permettra d'avoir un tracé plus précis.

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

```

5 LET L=7
7 LET O=NOT PI
10 FOR I=0 TO 9
20 PRINT AT 0,L+I;". ";AT L,I;".
";AT 10,L+I;". ";AT 17,I;". "
30 PRINT AT L+I,0;". ";TAB 10;".
"
40 PRINT AT I,L;". ";TAB 17;". "
50 NEXT I
60 FOR I=0 TO L
70 PRINT AT L-I,I;". ";TAB 10+I
";".
80 PRINT AT 17-I,I;". ";TAB 10+
I;". "
90 NEXT I
100 PRINT AT 21,0;"CUBE"
110 PAUSE 4E4

```

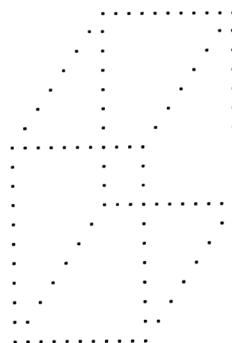


CUBE

```

7 LET O=0
10 FOR I=0 TO 9
20 PRINT AT 0,L+I;". ";AT L,I;". ";AT 10,
L+I;". ";AT 17,I;". "
30 PRINT AT L+I,0;". ";AT L+I,10;". "
40 PRINT AT I,L;". ";AT I,17;". "
50 NEXT I
60 FOR I=0 TO L
70 PRINT AT L-I,I;". ";AT L-I,10+I;". "
80 PRINT AT 17-I,I;". ";AT 17-I,10+I;". "
90 NEXT I
100 PRINT AT 21,0;"CUBE"
110 PAUSE 4E4

```



CUBE

Chaud, froid

REFERENCES :

Mode FAST.

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

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

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

LE PROGRAMME :

Les lignes 7 et 8 définissent la position de l'objet.

La ligne 110 cacule la distance par rapport aux deux coordonnées que vous avez entrées aux lignes 70 et 90.

La ligne 120 regarde si vous avez gagné.

Les lignes 140 et 150 normalisent la distance (qui doit avoir un sens, quelles que soient les dimensions de la grille initiale).

La ligne 160 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 NEWLINE.

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

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

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 donner une réponse alphanumérique à l'une des questions DONNEZ... (tapez A et NEWLINE, par exemple).

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.

ZX-81

```
1 LET F=7
2 LET G=11
3 LET S=NOT PI
4 PRINT "DIM. DE GRILLE "
5 INPUT D
6 PRINT D
7 LET X=INT (RND*D+SGN PI)
8 LET Y=INT (RND*D+SGN PI)
11 PAUSE 120
15 CLS
20 PRINT AT INT PI,SGN PI;"DON
NEZ X "
70 INPUT A
80 PRINT A;AT 5,SGN PI;"DONNEZ
Y "
90 INPUT B
100 PRINT B
105 LET S=S+SGN PI
110 LET T=ABS (X-A)+ABS (Y-B)
120 IF NOT T THEN GOTO 6000
130 PRINT AT F,SGN PI;"VOUS ETE
S "
140 LET N=INT (SQD D)/T+SGN PI
150 IF N>5 THEN LET N=5
160 IF T<=(SQD D)+2 THEN GOTO N
*1000
170 PRINT AT F,G;"COMPLETEMENT
GELE"
180 GOTO G
1000 PRINT AT F,G;"A PEINE TIEDE
"
1010 GOTO G
2000 PRINT AT F,G;"TIEDE"
2010 GOTO G
3000 PRINT AT F,G;"CHAUD"
3010 GOTO G
4000 PRINT AT F,G;"TRES CHAUD"
4010 GOTO G
5000 PRINT AT F,G;"BRULANT"
5010 GOTO G
6000 CLS
6010 PRINT "VOUS AVEZ TROUVE EN
";S;" COUPS"
```

DONNEZ X 3

DONNEZ Y 4

VOUS ETES TIEDE

ZX-SPECTRUM

```
1 LET F=7
2 LET G=11
3 LET S=0
4 PRINT "DIM. DE GRILLE "
5 INPUT D
6 PRINT D
7 LET X=INT (RND*D+1)
8 LET Y=INT (RND*D+1)
11 PAUSE 120
15 CLS
20 PRINT AT 3,1;"DONNEZ X "
70 INPUT A
80 PRINT A;AT 5,1;"DONNEZ Y "
90 INPUT B
100 PRINT B
105 LET S=S+1
110 LET T=ABS (X-A)+ABS (Y-B)
120 IF NOT T THEN GO TO 6000
130 PRINT AT F,1;"VOUS ETES "
140 LET N=INT (SQD D)/T+1
150 IF N>5 THEN LET N=5
160 IF T<=(SQD D)+2 THEN GO TO N*1000
170 PRINT PAPER 1; BRIGHT 1;AT F,G;"COMP
LETEMENT GELE"
180 GO TO G
1000 PRINT PAPER 5;AT F,G;"A PEINE TIEDE"
1010 GO TO G
2000 PRINT PAPER 4;AT F,G;"TIEDE"
2010 GO TO G
3000 PRINT PAPER 6;AT F,G;"CHAUD"
3010 GO TO G
4000 PRINT PAPER 3; BRIGHT 1;AT F,G;"TRES
CHAUD"
4010 GO TO G
5000 PRINT PAPER 2;AT F,G;"BRULANT"
5010 GO TO G
6000 CLS : BEEP 0.1,1: BEEP 0.1,1: BEEP 0
.1,1: BEEP 0.1,1: BEEP 0.1,1
6010 PRINT "VOUS AVEZ TROUVE EN ";S;" COU
PS"
```

DONNEZ X 3

DONNEZ Y 2

VOUS ETES TIEDE

Les allumettes

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : facile.

Catégorie de jeu : contre le ZX.

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 ZX. Chacun prend à tour de rôle autant d'allumettes qu'il le désire, dans la limite du nombre maximum autorisé. Celui qui prend la dernière allumette est considéré comme perdant.

LE PROGRAMME :

Le jeu du joueur commence en ligne 110.

Celui du ZX commence en ligne 200.

La ligne 130 contrôle la légalité du coup du joueur.

La ligne 140 gère le changement de joueur.

Le coup du ZX est calculé dans une boucle non formelle, de la ligne 210 à la ligne 240.

Le résultat est affiché par la ligne 1020.

Si la partie est terminée, la ligne 1040 le détecte et envoie en ligne 2000, pour le message de fin.

POUR JOUER :

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

Après chaque coup, le nombre d'allumettes restantes est affiché, et le ZX affiche A VOUS lorsque c'est au joueur de prendre.

Celui qui prend la dernière allumette a perdu.

EXTENSIONS POSSIBLES :

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

```

10 PRINT "TOTAL ";
20 INPUT N
30 PRINT N
40 PRINT "PRISE MAX ";
50 INPUT I
60 PRINT I
70 PRINT "1ER:VOUS(0),MOI(1)"
75 INPUT D
100 IF D THEN GOTO 200
110 PRINT "A VOUS ";
120 INPUT P
130 IF P<>INT P OR P<1 OR P>I OR P>N THEN GOTO 110
140 LET D=SGN PI
150 GOTO 1000
200 LET S=I+D
210 LET T=(N-S)/(I+D)
220 IF T=INT T THEN GOTO 250
230 LET S=S-D
240 GOTO 210
250 LET P=ABS (S-D)
260 IF NOT P THEN LET P=D
270 LET D=NOT PI
1000 PRINT P
1010 LET N=N-P
1020 PRINT "RESTE ";N
1030 PAUSE 120
1040 IF NOT N THEN GOTO 2000
1050 GOTO 100
2000 CLS
2005 PRINT "BRAVO"
2010 IF D THEN PRINT "POUR MOI"

```

```

TOTAL 49
PRISE MAX 7
1ER:VOUS(0),MOI(1)
1
RESTE 48
A VOUS 6
RESTE 42
1
RESTE 41
A VOUS 6
RESTE 35
2
RESTE 33
A VOUS 5
RESTE 28
3
RESTE 25
A VOUS 4
RESTE 21
4
RESTE 17
A VOUS

```

```

10 PRINT PAPER 6;"TOTAL ";; BEEP .2,10
20 INPUT N
30 PRINT N
40 PRINT PAPER 4;"PRISE MAXIMUM ";; BEE
P .2,10
50 INPUT I
60 PRINT I
70 PRINT PAPER 5;"QUI JOUE 1ER : VOUS (
0);MOI (1)"; BEEP .2,10
75 INPUT D
100 IF D THEN GO TO 200
110 PRINT INK 5; FLASH 1;"A VOUS "
120 INPUT P
130 IF P<>INT P OR P<1 OR P>I OR P>N THE
N GO TO 110
140 LET D=1
150 GO TO 1000
200 LET S=I+D
210 LET T=(N-S)/(I+D)
220 IF T=INT T THEN GO TO 250
230 LET S=S-D
240 GO TO 210
250 LET P=ABS (S-D)
260 IF NOT P THEN LET P=D
270 LET D=0
1000 PRINT P
1010 LET N=N-P
1020 PRINT BRIGHT 1; PAPER 3;"RESTE ";N
1030 PAUSE 120
1040 IF NOT N THEN GO TO 2000
1050 GO TO 100
2000 CLS
2005 PRINT PAPER 5;"BRAVO"
2010 IF D THEN PRINT PAPER 5;"POUR MOI"

```

```

TOTAL 35
PRISE MAXIMUM 5
QUI JOUE 1ER : VOUS (0);MOI (1)
4
RESTE 31
A VOUS
1
RESTE 30
5
RESTE 25
A VOUS
1
RESTE 24
5
RESTE 19
A VOUS
3
RESTE 16
3
RESTE 13
A VOUS

```


Allumettes à plusieurs

REFERENCES :

Mode FAST ou SLOW.

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 ZX le nombre de joueurs.

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

LE PROGRAMME :

Les lignes 10 à 50 permettent les différentes entrées.

Les lignes 100 à 130 réalisent les impressions.

La ligne 150 contrôle le coup de chaque joueur (par rapport au coup précédent).

La ligne 202 gère les tours (remise à 1 du numéro du joueur).

La ligne 200 réalise le test de fin, et la ligne 220 affiche le numéro du joueur gagnant (avec la règle prévue initialement).

POUR JOUER :

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

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 ZX.

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

EXTENSIONS POSSIBLES :

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

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

```

10 PRINT "NBRE D ALLUMETTES ";
20 INPUT N
30 PRINT N
40 PRINT "NBRE DE JOUEURS "
50 INPUT J
60 LET F=SGN PI
70 LET K=J
80 CLS
100 PRINT AT 4,SGN PI;"JOUEUR N
0. ";K
110 PRINT AT 6,SGN PI;"NBRE. MA
X A OTER ";F
120 PRINT AT 8,SGN PI;"NBRE. TO
TAL ";N
130 PRINT AT 12,SGN PI;"VOTRE C
OUP ";
140 INPUT C
150 IF C=NOT PI OR C>F THEN GOT
O 130
160 LET N=N-C
190 LET F=C+C
200 IF NOT N THEN GOTO 220
202 LET K=K-SGN PI
205 IF NOT K THEN LET K=J
210 GOTO 80
220 PRINT AT 12,SGN PI;"LE JOUE
UR ";K;" A GAGNE"

```

JOUEUR NO. 3

NBRE. MAX A OTER 8

NBRE. TOTAL 16

VOTRE COUP

```

10 PRINT "NBRE D'ALLUMETTES ";
20 INPUT N
30 PRINT N
40 PRINT "NBRE DE JOUEURS "
50 INPUT J
60 LET F=1
70 LET K=J
80 CLS
100 PRINT AT 4,1;"JOUEUR NO. ";K
110 PRINT AT 6,1;"NBRE. MAX A OTER ";F
120 PRINT AT 8,1;"NBRE. TOTAL ";N
130 PRINT AT 12,1;"VOTRE COUP ";
140 INPUT C
150 IF C=0 OR C>F OR C>N THEN GO TO 130
180 LET N=N-C
190 LET F=C*2
200 IF NOT N THEN GO TO 220
202 IF NOT (K-1) THEN LET K=J
205 LET K=K-1
210 GO TO 80
220 PRINT AT 12,1;"LE JOUEUR ";K;" A GAG
NE": BEEP .8,10

```

JOUEUR NO. 2

NBRE. MAX A OTER 8

NBRE. TOTAL 20

VOTRE COUP

Calendrier perpétuel

REFERENCES :

Mode FAST.

Difficulté de programmation : moyenne.

Catégorie de jeu : gadget publicitaire.

Difficulté du jeu : aucune.

LE JEU :

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

LE PROGRAMME :

Les lignes 20 et 40 vous permettront d'entrer les mois et l'année.

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

La ligne 70 calcule le nombre de jours, du mois (il n'est pas tenu compte ici des 29 février, pour simplifier).

La boucle A (90-110) réalise l'impression du calendrier proprement dit.

POUR JOUER :

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

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 FÉVRIER, etc.).

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

Pour en garder une trace, si l'on dispose bien sûr d'une imprimante, il suffira de taper COPY en fin d'affichage.

EXTENSIONS POSSIBLES :

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

```

10 PRINT "ANNEE ";
20 INPUT A;
25 PRINT A;
30 PRINT "      MOIS ";
40 INPUT M
45 PRINT M
50 LET D=INT (365.25*(A-(M<INT
PI))) +INT (30.6*(M+SGN PI+(M<IN
T PI)*12)) -INT ((INT ((A-(M<INT
PI))/100)-7)*.75)+SGN PI
60 LET D=D-7*INT (D/7)
70 LET M=31-(M=4 OR M=6 OR M=9
OR M=11)-(M=2)*INT PI
75 PRINT
80 PRINT "DIM LUN MAR MER JEU
VEN SAM"
85 PRINT
90 FOR A=SGN PI TO M
100 PRINT TAB 4*(D+A-SGN PI);A;
105 IF D+A-SGN PI=6 THEN LET D=D-
D-7
110 NEXT A

```

ANNEE 1984 MOIS 9

DIM LUN MAR MER JEU VEN SAM

						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

```

10 PRINT "      ANNEE ";
20 INPUT A
25 PRINT A;
30 PRINT "      MOIS ";
40 INPUT M
45 PRINT M
50 LET D=INT (365.25*(A-(M<3)))+INT (30
.6*(M+1+(M<3)*12))-INT ((INT ((A-(M<3))/1
00)-7)*.75)+1
60 LET D=D-7*INT (D/7)
70 LET M=31-(M=4 OR M=6 OR M=9 OR M=11)
-(M=2)*3
75 PRINT : PRINT : PRINT
80 PRINT INK 2; INVERSE 1;" DIM LUN MA
R MER JEU VEN SAM "
85 PRINT
90 FOR A=SGN PI TO M
100 PRINT PAPER 6; INK 2; BRIGHT 1;TAB 4
*(D+A-1)+2;A;
105 IF D+A-1=6 THEN LET D=D-7
110 NEXT A

```

ANNEE 1984 MOIS 12

DIM LUN MAR MER JEU VEN SAM

						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Bataille navale

REFERENCES :

Mode SLOW ou FAST.

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 ZX 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 :

Les boucles I (20-60) et J (30-50) dessinent le terrain de jeu.

Les lignes 70 et 80 définissent l'emplacement du navire.

Les coordonnées du tir sont entrées aux lignes 100 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 160 intervient pour déplacer le navire si les coups sont rapprochés.

La ligne 170 permet l'affichage de la distance.

POUR JOUER :

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

Si on a tiré trop près, on obtient l'affichage PFTT... 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 ZX, mettre plusieurs navires, des sous-marins, etc.

```

10 LET A=NOT PI
20 FOR I=A TO INT (PI*PI)
30 FOR J=A TO INT (PI*PI)
40 PRINT AT 2,U+J+INT PI;J;AT
INT PI+I,A;I;AT INT PI+I,U+J+INT
PI;" "
50 NEXT J
60 NEXT I
65 LET S=SGN PI
70 LET X=INT (RND*10)
80 LET Y=INT (RND*10)
90 PRINT AT CODE "?" ,A;"H"
100 INPUT H
110 PRINT AT CODE " (" ,A;"Q"
120 INPUT V
130 PRINT AT H+INT PI,V+V+INT P
I;S
140 LET D=INT SQR ((X-H)*(X-H)+(Y-V)*(Y-V))
150 IF NOT D THEN GOTO CODE "CO
S "
155 LET S=S+SGN PI
157 IF D=SGN PI THEN PRINT AT C
ODE ")",A;"PFTTT..."
160 IF D=SGN PI THEN GOTO 70
170 PRINT AT CODE ")",A;"DIST :
";D;" "
180 GOTO 90
200 CLS
210 PRINT S;" COUPS"

```

	0	1	2	3	4	5	6	7	8	9
0	.	.	.	5	.	3	.	7	.	.
1	.	.	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.	.	.
3	.	.	.	.	.	.	.	.	.	.
4	.	8	.	.	4	.	.	.	.	.
5	2	.	.	.	.	.	.	.	.	.
6	.	.	.	.	.	.	.	.	.	.
7	.	.	.	.	.	.	.	.	.	.
8	.	.	.	.	.	.	.	.	1	.
9	.	.	.	.	6	.	.	.	.	.

H
V
DIST 7.

```

10 LET A=0
20 FOR I=A TO 9
30 FOR J=A TO 9
40 PRINT AT 0,3*J+3;J;AT 2+2*I,A;I: CIR
CLE INK 2;5+256*(3*J+3)/32,180-(176*(3+2*
I)/22),8
50 NEXT J
60 NEXT I
65 LET S=1
70 LET X=INT (RND*10)
80 LET Y=INT (RND*10)
90 PRINT AT 21,A;"H "
100 INPUT H
110 PRINT AT 21,A;"V "
120 INPUT V
130 PRINT AT 2*H+2,3*V+3;S
140 LET D=INT SQR ((X-H)*(X-H)+(Y-V)*(Y-
V))
150 IF NOT D THEN GO TO 200
155 LET S=S+1
157 IF D=1 THEN PRINT AT 21,A;"PFTT... "
: PAUSE 20
160 IF D=1 THEN GO TO 70
170 PRINT AT 21,A;"DIST :";D: BEEP 0.5,1
0: PAUSE 150
180 GO TO 90
200 CLS
210 PRINT S;" COUPS"

```

	0	1	2	3	4	5	6	7	8	9
0	(	(	(	(	(	(	(	(	(	(
1	(	(	(	(	(	(	(	(	(	(
2	(	(	(	(	(	(	(	(	(	(
3	(	(	(	(	(	(	(	(	(	(
4	(	(	(	(	(	(	(	(	(	(
5	(	(	(	(	(	(	(	(	(	(
6	(	(	(	(	(	(	(	(	(	(
7	(	(	(	(	(	(	(	(	(	(
8	(	(	(	(	(	(	(	(	(	(
9	(	(	(	(	(	(	(	(	(	(

DIST : 6

Golf

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez facile.

LE JEU :

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

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

LE PROGRAMME :

La ligne 25 permet de définir la position du trou, et la ligne 30 dessine le terrain de jeu.

Le déplacement à réaliser est calculé en ligne 80.

La boucle I (90-120) réalise le déplacement de la balle.

La ligne 130 regarde si le coup est, ou non, gagnant ; s'il l'est, on recommence (ligne 220) jusqu'à ce que l'on ait réalisé 18 trous (boucle S (15-220) ; fin de boucle en 240, 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 NEWLINE, la balle commence à se déplacer ; elle peut alors soit s'arrêter avant le trou (auquel cas on recommence), soit tomber dedans (on passe alors au trou suivant), soit le dépasser : on devra alors frapper avec une force négative.

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

EXTENSIONS POSSIBLES :

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

```

10 LET E=NOT PI
15 FOR S=SGN PI TO 18
20 LET X=SGN PI
25 LET T=INT (21*RND+9)
30 PRINT AT 21,SGN PI;"
T;" ";AT 20,X;"0"
35 PRINT AT 5,SGN PI;"TROU ";S
;" FORCE
60 INPUT F
70 PRINT AT 5,15;F
80 LET F=INT (F*.16-(ABS (F)*(
SQR (RND)+.1)**2)/1000)
85 IF NOT F THEN GOTO 35
90 FOR I=SGN F TO F STEP SGN F
100 LET X=X+SGN F
110 IF X<31 AND X>=SGN PI THEN
PRINT AT 20,X-SGN F;" ";AT 20,X,
"0"
120 NEXT I
130 IF X=T THEN GOTO 200
140 LET E=E+SGN PI
150 GOTO 35
200 PRINT AT 20,X;" ";AT 21,X;"
0"
210 PAUSE 200
220 NEXT S
230 CLS
240 PRINT "18 TROUS EN ";E;" CO
UPS"

```

TROU 3 FORCE -10

```

10 LET E=0
15 FOR S=1 TO 18
20 LET X=1
25 LET T=INT (21*RND+9)
30 PRINT INK 3;AT 21,1;"
"; INK 0;AT 21,T;" ";AT 2
0,X;"0"
35 PRINT AT 5,1;"TROU ";S;" FORCE
"
60 INPUT F
70 PRINT AT 5,15;F
80 LET F=INT (F*.16-((ABS F)*(SQR (RND)
+.1)^2)/1000)
85 IF NOT F THEN GO TO 35
90 FOR I=SGN F TO F STEP SGN F
100 LET X=X+SGN F: PAUSE 15
110 IF X<31 AND X>=1 THEN PRINT AT 20,X-
SGN F;" ";AT 20,X;"0": BEEP 0.03,0
120 NEXT I
130 IF X=T THEN GO TO 200
140 LET E=E+1
150 GO TO 35
200 BEEP 1,10: PRINT AT 20,X;" ";AT 21,X
;"0"
210 PAUSE 200
220 NEXT S
230 CLS
240 PRINT "18 TROUS EN ";E;" COUPS"

```

TROU 1 FORCE 63

Solfège

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion.

Difficulté du jeu : facile.

LE JEU :

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

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

LE PROGRAMME :

La boucle I (15-30) dessine une portée.

La note est choisie en ligne 40 et affichée en ligne 50.

La ligne 60 affiche le score.

La ligne 70 affiche le nom de la clé (ici clé de sol) et vous interroge.

La ligne 85 s'occupe de la fin de la partie.

Les lignes 90 et 100 effectuent le décodage de la note entrée.

La ligne 120 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 NEWLINE.

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 NEWLINE, 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.

```

5 LET S=NOT PI
7 LET P=S
10 CLS
15 FOR I=SGN PI TO 5
20 PRINT AT I+I,NOT PI:
30 NEXT I
40 LET X=INT (12*RND)
50 PRINT AT X-SGN PI,9;" "AT
X,9;" "AT X+SGN PI,9;" "
60 PRINT AT 13,NOT PI:"SCORE :
";S;" SUR ";P;" ESSAIS"
70 PRINT AT 15,NOT PI:"CLE DE
SOL; NOTE ?"
80 INPUT R$
85 IF R$="0" THEN STOP
90 LET Y=(SGN PI AND R$="FA")+
(2 AND R$="MI")+(INT PI AND R$="
RE")
100 LET Y=Y+(4 AND R$="DO")+(5
AND R$="SI")+(6 AND R$="LA")
110 LET P=P+SGN PI
120 IF Y=X OR Y=X-7 THEN LET S=
S+SGN PI
130 GOTO 10

```

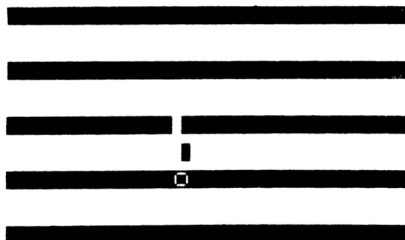


SCORE :3 SUR 6 ESSAIS
CLE DE SOL; NOTE ?

```

5 LET S=0
7 LET P=S
10 PAPER 6: CLS
15 FOR I=1 TO 5
20 PRINT INK 5;AT I+I,0;"
30 NEXT I
40 LET X=INT (12*RND)
50 PRINT INK 2: INVERSE 1;AT X-1,9;" ";
AT X,9;" ";AT X+1,9;" "
60 PRINT AT 13,0:"SCORE :";S;" SUR ";P;
" ESSAIS"
70 PRINT AT 15,0:"CLE DE SOL; NOTE :";
80 INPUT R$: PRINT R$
85 IF R$="0" THEN STOP
90 LET Y=(1 AND R$="FA")+(2 AND R$="MI"
)+(3 AND R$="RE")
100 LET Y=Y+(4 AND R$="DO")+(5 AND R$="S
I")+(6 AND R$="LA")
110 LET P=P+1
120 IF Y=X OR Y=X-7 THEN LET S=S+1: BEEP
1,20-4*X: PAUSE 100
130 GO TO 10

```



SCORE :3 SUR 5 ESSAIS
CLE DE SOL; NOTE :SOL

Le jeu du 21

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : facile.

Catégorie de jeu : jeu contre le ZX.

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 ZX qui essaiera alors d'avoir un meilleur score.

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

LE PROGRAMME :

Les lignes 60, 70 et 80 réalisent le contrôle de celui qui doit jouer (joueur ou ZX).

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

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

Le total du ZX est calculé en ligne 170 et affiché par la ligne 175.

La ligne 180 contrôle la fin du jeu pour le ZX.

Les lignes 200 à 210 affichent les résultats, et la ligne 220 cherche quel est le gagnant.

Le gain (ou la perte) sera affiché par la ligne 260, et le total par la ligne 500.

La ligne 520 contrôle la fin de partie.

POUR JOUER :

Taper RUN pour lancer le jeu.

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

Dès que l'on a terminé, on tape 0 et NEWLINE pour laisser jouer le ZX (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 SUITE (O ou 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.

ZX-81

```
10 LET G=NOT PI
20 CLS
30 PRINT "VOTRE MISE ";
40 INPUT M
50 PRINT M
55 LET T=NOT PI
60 PRINT AT 5,SGN PI;"1 :TIRER
0 :FINIR"
70 INPUT R
80 IF NOT R THEN GOTO 130
100 GOSUB 1000
110 LET T=T+X
120 PRINT AT 9,SGN PI;"TOTAL :";
T
125 GOTO 70
130 CLS
140 PRINT "LA BANQUE JOUE"
150 LET S=NOT PI
160 GOSUB 1000
170 LET S=S+X
175 PRINT AT 5,SGN PI;"TOTAL :";
S
177 PAUSE 120
180 IF S>T OR S>=CODE "0" THEN
GOTO CODE "CODE "
190 GOTO 150
195 CLS
200 PRINT AT 9,SGN PI;"VOUS AVE
Z ";T
210 PRINT AT 11,SGN PI;"LA BANQ
UE A ";S
220 IF S>CODE "+" OR (S<T AND T
<=CODE "+") THEN GOTO 260
240 LET M=-M
250 PRINT AT 14,SGN PI;"GAIN ";
M
270 LET G=G+M
500 PRINT AT 18,SGN PI;"TOTAL "
;G
505 PRINT "SUITE (O OU N)?"
510 INPUT R$
520 IF R$="N" THEN STOP
530 GOTO CODE "="
1000 LET X=INT (RND*6+1)
1010 RETURN
```

VOUS AVEZ 22
LA BANQUE A 19

GAIN -10

TOTAL -10
SUITE (O OU N)?

ZX-SPECTRUM

```
10 LET G=0
20 CLS
30 PRINT "VOTRE MISE ";
40 INPUT M
50 PRINT M
55 LET T=0
60 PRINT AT 5,1;"1 :TIRER ; 0 :FINIR"
70 INPUT R
80 IF NOT R THEN GO TO 130
100 GO SUB 1000
110 LET T=T+X
120 PRINT AT 9,1;"TOTAL :";T
125 GO TO 70
130 CLS
140 PRINT "LA BANQUE JOUE"
150 LET S=0
160 GO SUB 1000
170 LET S=S+X
175 PRINT AT 5,1;"TOTAL :";S
177 PAUSE 120
180 IF S>T OR S>=17 THEN GO TO 196
190 GO TO 160
196 CLS
200 PRINT AT 9,1;"VOUS AVEZ ";T: BEEP 0,
05,0
210 PRINT AT 11,1;"LA BANQUE A ";S: BEEP
0,05,0
220 IF S>21 OR (S<T AND T<=21) THEN GO T
O 260
240 LET M=-M
260 PRINT AT 14,1;"GAIN ";M
270 LET G=G+M
500 PRINT AT 18,1;"TOTAL ";G
505 PRINT "SUITE (O OU N)?"
510 INPUT R$
520 IF R$="N" THEN STOP
530 GO TO 20
1000 LET X=INT (RND*6+1): BEEP 1,7*X-22
1010 RETURN
```

VOTRE MISE 100

1 :TIRER ; 0 :FINIR

TOTAL :21

Pair-impair

REFERENCES :

Mode FAST.

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 ZX lance simultanément deux dés, non visibles et pose trois questions : quelle est la parité du total, quel est le total, et enfin donner la valeur de l'un des dés.

On peut ensuite soit poursuivre le jeu, soit l'interrompre à tout moment.

LE PROGRAMME :

Les dés sont tirés par un sous-programme placé aux lignes 1000 et 1010.

La ligne 50 tient le compte du nombre de parties, et les lignes 150, 210 et 250 comptabilisent les points marqués.

La parité du total est déterminée par les lignes 97 et 100 (la ligne 100 compare avec la réponse du joueur).

La ligne 400 réalise l'affichage de fin de partie : total, nombre de points, nombre de parties jouées.

La ligne 420 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 NEWLINE 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 ou N) et l'on devra répondre N suivi de NEWLINE pour arrêter.

EXTENSIONS POSSIBLES :

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

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

```

10 LET T=NOT PI
15 LET R=T
20 GOSUB 1000
30 LET A=X
40 GOSUB 1000
50 LET R=R+SGN PI
60 LET S=A+X
70 CLS
80 PRINT "TOTAL PAIR (0) OU IM
PAIR (1) :";
90 INPUT N
95 PRINT N
97 LET P=S/2-INT (S/2)
100 IF N=P OR N-.5=P THEN GOTO
140
110 PRINT AT 5,SGN PI;"FAUX"
120 GOTO 150
140 PRINT AT 5,SGN PI;"EXACT"
150 LET T=T+SGN PI
160 PRINT AT 8,SGN PI;"VOTRE TO
TAL :";
170 INPUT N
175 PRINT N
180 IF N=S THEN GOTO 210
190 PRINT AT 12,SGN PI;"PERDU"

195 PAUSE 130
200 GOTO 400
210 LET T=T+2
220 PRINT AT 12,SGN PI;"BRAVO;
DONNEZ UN DE ?"
230 INPUT N
240 IF N<>R AND N<>S-A THEN GOT
O 190
250 LET T=T+INT PI
260 GOTO 195
400 CLS
405 PRINT "TOTAL :";T;" / ";R;" ; O
N CONTINUE (0 OU N) "
410 INPUT R$
420 IF R$="N" THEN STOP
430 GOTO 20
1000 LET X=INT (RND*6+SGN PI)
1010 RETURN

```

TOTAL PAIR (0) OU IMPAIR (1) : 1

EXACT

VOTRE TOTAL : 7

PERDU

```

10 LET T=0
15 LET R=T
20 GO SUB 1000
30 LET A=X
40 GO SUB 1000
50 LET R=R+1
60 LET S=A+X: BEEP 1,3*S-20
70 CLS
80 PRINT "TOTAL PAIR (0) OU IMPAIR (1)
:";
90 INPUT N
95 PRINT N
97 LET P=S/2-INT (S/2)
100 IF N=P OR N-.5=P THEN GO TO 140
110 PRINT AT 5,1;"FAUX"
120 GO TO 160
140 PRINT AT 5,1;"EXACT"
150 LET T=T+1
160 PRINT AT 8,1;"VOTRE TOTAL :";
170 INPUT N
175 PRINT N
180 IF N=S THEN GO TO 210
190 PRINT AT 12,1;"PERDU"
"
195 PAUSE 130
200 GO TO 400
210 LET T=T+2
220 PRINT AT 12,1;"BRAVO; DONNEZ UN DE ?"
"
230 INPUT N
240 IF N<>A AND N<>S-A THEN GO TO 190
250 LET T=T+3: PRINT AT 16,1;"BRAVO": GO
TO 195
400 CLS
405 PRINT "TOTAL :";T;" / ";R;" ; ON CONTI
NUE (0 OU N) "
410 INPUT R$
420 IF R$="N" THEN STOP
430 GO TO 20
1000 LET X=INT (RND*6+1)
1010 RETURN

```

TOTAL PAIR (0) OU IMPAIR (1) : 1

EXACT

VOTRE TOTAL : 7

BRAVO; DONNEZ UN DE ?

BRAVO

Devine un chiffre

REFERENCES :

Mode FAST.

Difficulté de programmation : facile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

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

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

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

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

La variable R\$, testée en ligne 620, 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 NEWLINE. On obtient alors un deuxième affichage UN CHIFFRE DE 1 A 6 ? ; on entre un chiffre suivi de NEWLINE ; 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 ou N) ? ; taper O pour continuer, N pour arrêter (suivi de NEWLINE).

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 ZX.

```

5 LET G=NOT PI
7 CLS
10 PRINT "VOTRE MISE :";
20 INPUT N
30 PRINT N
50 PRINT AT 5,SGN PI;"UN CHIFFRE DE 1 A 6 :";
60 INPUT D
80 PRINT D
90 GOSUB 1000
100 LET A=X
110 GOSUB 1000
120 LET B=X
130 GOSUB 1000
140 LET C=X
150 PRINT AT 9,4;A;TAB 8;B;TAB 12;C
160 PRINT AT 11,4;"A";TAB 8;"B";TAB 12;"C"
170 LET T=NOT PI
180 IF D=A THEN LET T=T+N
190 IF D=B THEN LET T=T+N
200 IF D=C THEN LET T=T+N
210 IF T=NOT PI THEN GOTO 500
220 PRINT AT 15,SGN PI;"VOUS GAGNEZ ";T
230 LET G=G+T
240 GOTO 600
500 PRINT AT 15,SGN PI;"PERDU"
510 LET G=G-N
600 PRINT AT 17,NOT PI;"TOTAL ACTUEL :";G;" ON CONTINUE (O OU N)?"
610 INPUT R$
620 IF R$="N" THEN STOP
630 GOTO 7
1000 LET X=INT (RND*6+SGN PI)
1010 RETURN

```

VOTRE MISE : 10

UN CHIFFRE DE 1 A 6 : 3

```

5   6   4
A   B   C

```

PERDU

TOTAL ACTUEL : 30; ON CONTINUE (O OU N)?

```

5 LET G=0
7 CLS
10 PRINT "VOTRE MISE :";
20 INPUT N
30 PRINT N
50 PRINT AT 5,1;"UN CHIFFRE DE 1 A 6 :";
60 INPUT D
80 PRINT D
90 GO SUB 1000
100 LET A=X
110 GO SUB 1000
120 LET B=X
130 GO SUB 1000
140 LET C=X
150 PRINT INK 2;AT 9,4;A: BEEP 1,7*A-22;
PRINT INK 4;AT 9,8;B: BEEP 1,7*B-22; PRINT INK 5;AT 9,12;C: BEEP 1,7*C-22
160 PRINT FLASH 1;AT 11,4;"A";AT 11,8;"B";AT 11,12;"C"
170 LET T=0
180 IF D=A THEN LET T=T+N
190 IF D=B THEN LET T=T+N
200 IF D=C THEN LET T=T+N
210 IF T=0 THEN GO TO 500
220 PRINT AT 15,1;"VOUS GAGNEZ ";T
230 LET G=G+T
240 GO TO 600
500 PRINT AT 15,1;"PERDU"
510 LET G=G-N
600 PRINT AT 17,1;"TOTAL ACTUEL :";G;" ON CONTINUE (O OU N)?"
610 INPUT R$
620 IF R$="N" THEN STOP
630 GO TO 7
1000 LET X=INT (RND*6+1)
1010 RETURN

```

VOTRE MISE : 100

UN CHIFFRE DE 1 A 6 : 3

```

6   6   6
A   B   C

```

PERDU

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

Le jeu de Wytthof

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion (deux joueurs).

Difficulté du jeu : moyenne.

LE JEU :

On dispose d'une grille carrée de 10 cases sur 10 cases. Un point apparaît, en dehors de la diagonale et les joueurs devront le déplacer à chaque tour, suivant seulement trois directions : droite, bas et diagonale bas et droite.

Le premier joueur qui arrive en (9,9) sera considéré comme gagnant, ou perdant, selon la convention faite au début.

LE PROGRAMME :

Les lignes 10, 20 et 30 permettent de définir une position de départ, en dehors de la diagonale.

Les deux boucles emboîtées I (50-90) et J (60-80) dessinent le terrain de jeu.

La ligne 110 imprime la position du point mobile.

Le test de fin est effectué en ligne 115.

Le changement de joueur est assuré par la ligne 130.

Les directions de déplacement disponibles, 1, 2 et 3, sont décodées par trois sous-programmes qui commencent aux lignes 1000, 2000 et 3000.

POUR JOUER :

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

Puis on obtient l'affichage JR 1 et en dessous la question DIR 1, 2, 3 ? à laquelle on répond en tapant l'un de ces trois nombres, suivi de NEW-LINE.

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 ZX de jouer, etc.

```

10 LET X=INT (RND*5)
20 LET Y=INT (RND*5)
30 IF X=Y THEN GOTO 20
40 LET A=NOT PI
50 FOR I=A TO 9
60 FOR J=A TO 9
70 PRINT AT INT PI,U+U+INT PI;
  U;AT 4+I,A;I;AT 4+I,U+U+INT PI;"
80 NEXT J
90 NEXT I
105 LET H=SGN PI
110 PRINT AT Y+4,X+X+INT PI CHR
$ 128
115 IF X+Y=CODE ">" THEN STOP
120 PRINT AT CODE(">"),SGN PI;"J
R 2"
130 IF NOT (H=SGN PI) THEN PRIN
T AT CODE(">"),SGN PI;"JR 1"
140 PRINT "DIR 1,2,3 ?"
150 INPUT D
160 GOSUB D*1000
170 LET H=-H
190 GOTO 110
1000 LET X=X+SGN PI
1010 RETURN
2000 GOSUB 1000
3000 LET Y=Y+SGN PI
3010 RETURN

```

	0	1	2	3	4	5	6	7	8	9
0	.	.	.	.	.	.	.	.	.	.
1	.	.	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.	.	.
3	.	.	.	.	.	.	.	.	.	.
4	.	.	.	.	.	.	.	.	.	.
5	.	.	.	.	.	.	.	.	.	.
6	.	.	.	.	.	.	.	.	.	.
7	.	.	.	.	.	.	.	.	.	.
8	.	.	.	.	.	.	.	.	.	.
9	.	.	.	.	.	.	.	.	.	.

JR 1
DIR 1,2,3 ?

```

10 LET X=INT (RND*5)
20 LET Y=INT (RND*5)
30 IF X=Y THEN GO TO 20
40 PAPER 5: LET A=0
50 FOR I=0 TO 9
60 FOR J=0 TO 9
70 PRINT AT 3,2*J+3;J;AT 4+I,0;I; INVER
SE 1;AT 4+I,2*J+3;".
80 NEXT J
90 NEXT I
105 LET H=1
110 PRINT FLASH 1; INK 1;AT Y+4,X*2+3;".
": BEEP 1,0
115 IF X+Y=18 THEN STOP
120 PRINT AT 17,1;"JOUEUR 2"
130 IF NOT (H=1) THEN PRINT AT 17,1;"JOU
EUR 1"
140 PRINT "DIRECTION 1,2,3 ?"
150 INPUT D
160 GO SUB D*1000
170 LET H=-H
190 GO TO 110
1000 LET X=X+1
1010 RETURN
2000 GO SUB 1000
3000 LET Y=Y+1
3010 RETURN

```

	0	1	2	3	4	5	6	7	8	9
0	.	.	.	.	.	.	.	.	.	.
1	.	.	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.	.	.
3	.	.	.	.	.	.	.	.	.	.
4	.	.	.	.	.	.	.	.	.	.
5	.	.	.	.	.	.	.	.	.	.
6	.	.	.	.	.	.	.	.	.	.
7	.	.	.	.	.	.	.	.	.	.
8	.	.	.	.	.	.	.	.	.	.
9	.	.	.	.	.	.	.	.	.	.

JOUEUR 1
DIRECTION 1,2,3 ?

Rebonds

REFERENCES :

Mode SLOW.

Difficulté de programmation : facile.

Catégorie de jeu : jeu d'adresse.

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

LE PROGRAMME :

La boucle I (10-30) imprime les chiffres et les lettres en haut de l'écran.

Les lignes 40 et 50 définissent les positions initiales de la balle et de la cible.

La ligne 60 effectue leur impression.

Le symbole correspondant à la direction choisie est entré en ligne 90.

La boucle I (100-130) réalise le déplacement de la balle.

La ligne 120 calcule la nouvelle position.

Le changement de direction, lors de la rencontre d'un bord, est effectué par la ligne 115, par renvoi dans une partie de programme indépendante, située aux lignes 150 à 160.

La ligne 140 effectue l'impression de fin de partie.

Le nombre de rebonds est compté en ligne 155.

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 U), et on l'entre, suivi de NEWLINE.

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

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

EXTENSIONS POSSIBLES :

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

ZX-81

```

5 CLS
10 FOR I=NOT PI TO CODE "3"
20 PRINT CHR$(I+CODE "0")
30 NEXT I
40 LET X=INT (CODE "2"*RND+INT
PI)
50 LET Y=INT (CODE "2"*RND+INT
PI)
60 PRINT AT X,NOT PI;"0" AT Y,
CODE "3";"■"
90 INPUT R$
95 LET S=NOT PI
97 LET T=X
100 FOR I=SGN PI TO CODE "3"
110 LET T=T-X/(CODE R$-CODE "0"
)
115 IF T>=CODE "=" OR T<=NOT PI
THEN GOTO 150
120 PRINT AT T,I;"0"
130 NEXT I
140 IF ABS (T-Y)<INT PI THEN PR
INT "BRAVO :";S;" REBONDS"
145 PAUSE 4E4
147 RUN
150 LET X=-X
155 LET S=S+SGN PI
160 GOTO 130

```

0123456789ABCDEF0GHIJKLMN0PQRSTU0

```

      0 0          0 0          0
BRAVO :4 REBONDS

      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 0

```

ZX-SPECTRUM

```

5 CLS
10 FOR I=0 TO 31
20 PRINT INVERSE 1; INK 1;CHR$(I+CODE
"0") AND I<=9;CHR$(I+CODE "7") AND I>9;
30 NEXT I
40 LET X=INT (15*RND+3)
50 LET Y=INT (15*RND+3)
60 PRINT AT X,0;"0" AT Y,31;"■"
90 INPUT R$
95 LET S=0
97 LET T=X
100 FOR I=1 TO 31
110 LET T=T-X/(CODE R$-CODE "0"-7*(R$>="
A"))
115 IF T>=20 OR T<=0 THEN GO TO 150
120 PRINT AT T,I;"0"
130 NEXT I
140 IF ABS (T-Y)<3 THEN PRINT INVERSE 1;
INK 2;"BRAVO :";S;" REBONDS": BEEP .4,18
145 PAUSE 4E4
147 RUN
150 LET X=-X
155 LET S=S+1
160 GO TO 130

```

012304056789AB00E0GHIJKLH000PQRSTU0

```

      0 0      0 0      0 0      0
BRAVO :6 REBONDS

      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 0
      0 0      0 0      0 0

```

Jackpot

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : facile.

Catégorie de jeu : hasard (un joueur).

Difficulté du jeu : facile.

LE JEU :

C'est une variante simplifiée d'un jeu de hasard classique, dans lequel le joueur doit faire une mise et tirer sur une poignée pour lancer le jeu. S'il obtient certaines combinaisons de symboles, il gagne (plus ou moins, selon sa mise et la combinaison obtenue).

Ici, les symboles sont remplacés par des chiffres, et le fait d'avoir trois chiffres identiques permet de gagner. Il existe de plus une combinaison particulière qui permet de remporter le Jackpot.

LE PROGRAMME :

La boucle I (40-60) permet le dessin de la machine, la ligne 74 affichant les emplacements (vides) des chiffres.

La mise sera entrée en ligne 87, et testée en ligne 95 : si on mise 0, la partie est terminée (départ vers la ligne 2000).

Les lignes 130, 150 et 170 permettent le tirage des 3 chiffres, en appelant un sous-programme qui se trouve en ligne 1000.

Le résultat est affiché par la ligne 180, puis testé en ligne 190, et en ligne 310, si l'on a gagné.

POUR JOUER :

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

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

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 NEWLINE et ainsi de suite.

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

Le restant de votre pot initial sera alors affiché.

EXTENSIONS POSSIBLES :

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

```

10 PRINT "POT DE DEBUT"
20 INPUT P
30 CLS
40 FOR I=SGN PI TO INT PI
50 PRINT AT I+INT PI,VAL "5;"
60 NEXT I
74 PRINT AT VAL "5",VAL "7;"
";TAB VAL "10";" ";TAB VAL "13";
";AT VAL "9",SGN PI)"
85 PRINT AT VAL "11",SGN PI;"V
OTRE MISE"
87 INPUT M
95 IF NOT M OR P-M<NOT PI THEN
GOTO VAL "2000"
130 GOSUB VAL "1000"
140 LET A=X
150 GOSUB VAL "1000"
160 LET B=X
170 GOSUB VAL "1000"
180 PRINT AT VAL "5",VAL "7";A;
TAB VAL "10";B;TAB VAL "13";X
190 IF A=B AND A=X THEN GOTO VA
L "300"
195 LET N=-M
200 PRINT AT VAL "9",SGN PI;"MI
SE ";M;" RESULTAT ";N
204 PAUSE VAL "160"
205 LET P=P+N
210 GOTO VAL "40"
300 LET N=M*50
310 IF A=7 THEN LET N=M*500
320 GOTO CODE "COS "
1000 LET X=INT (VAL "9"*RND+SGN
PI)
1010 RETURN
2000 CLS
2010 PRINT "VOUS AVEZ ";P;";"P; B
IENTOT"

```

```

5 6 7

```

```

MISE 10  RESULTAT -10
VOTRE MISE

```

```

10 PRINT "POT DE DEBUT"
20 INPUT P
30 CLS
40 FOR I=1 TO 3
50 PRINT INK 2;AT I+3,6;" "
60 NEXT I
74 PRINT INK 5;AT 5,7;" ";AT 5,10;" ";A
T 5,13;" "
85 PRINT AT 11,1;"VOTRE MISE"
87 INPUT M
95 IF NOT M THEN GO TO 2000
130 GO SUB 1000
140 LET A=X
150 GO SUB 1000
160 LET B=X
170 GO SUB 1000
180 PRINT AT 5,7;A: BEEP 1,5*A-20: PRINT
AT 5,10;B: BEEP 1,5*B-20: PRINT AT 5,13;
X: BEEP 1,5*X-20
190 IF A=B AND A=X THEN GO TO 300
195 LET N=-M
200 PRINT AT 9,1;"MISE ";M;"  RESULTAT "
;N
204 PAUSE 160
205 LET P=P+N
210 GO TO 40
300 LET N=M*50
310 IF A=7 THEN LET N=M*500
320 GO TO 200
1000 LET X=INT (9*RND+1)
1010 RETURN
2000 CLS
2010 PRINT "VOUS AVEZ ";P;","A BIENTOT"

```

```

5 6 7

```

```

MISE 10  RESULTAT -10
VOTRE MISE

```

Roulette

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : moyenne.

Catégorie de jeu : hasard (plusieurs joueurs).

Difficulté du jeu : facile.

LE JEU :

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

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

LE PROGRAMME :

La boule I (10-30) dessine le terrain de jeu, auquel la ligne 40 rajoute les trois symboles P, 00 et I.

La boucle I (110-130) réalise le déplacement de la boule, l'arrêt étant contrôlé par les lignes 125 et 128.

La ligne 135 réalise un effacement particulier pour le cas où la boule passe sur les symboles P, 00 et I.

La ligne 140 fait descendre d'une ligne.

La ligne 160 teste l'arrivée en fin de terrain.

La ligne 170 tire la position d'arrêt.

POUR JOUER :

Il suffit de faire RUN, et de regarder.

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

EXTENSIONS POSSIBLES :

On pourra essayer de se rapprocher davantage d'une véritable roulette, de différentes manières : mieux dessiner le terrain de jeu, introduire des couleurs, simuler le ralentissement de la boule, introduire les 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.

```

5 LET A=INT PI
7 LET B=A-SGN PI
10 FOR I=SGN PI TO A+A
20 PRINT AT A,A*I+A;B*I;AT A+A
,A*I+A;B*I+18;AT A+A,A*I+A;B*I-S
GN PI;AT 12,A*I+A;B*I+17
30 NEXT I
32 LET X=NOT PI
35 LET L=A+SGN PI
40 PRINT AT L,SGN PI;"P";AT L+
A,SGN PI;"00";AT 10,SGN PI;"I"
110 FOR I=SGN PI TO 10
120 PRINT AT L,A*I;" ■" AND L
*I<>VAL "130"
122 PAUSE CODE "="
123 LET X=X-SGN PI
125 IF NOT X THEN PAUSE 4E4
129 PRINT AT L,A*I;"
130 NEXT I
140 LET L=L+A
150 IF L<>CODE "(" THEN GOTO 11
0
170 LET X=INT (RND*CODE "PI")
180 GOTO CODE "7"

```

```

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

```

```

5 BORDER 3: LET A=3
7 LET B=A-1
10 FOR I=1 TO 9
20 PRINT PAPER 2;AT A,A*I+A;B*I; PAPER
4;AT A+A,A*I+A;B*I+18; PAPER 2;AT 9,A*I+A
;B*I-I; PAPER 4;AT 12,A*I+A;B*I+17
30 NEXT I
32 LET X=0
35 LET L=A+1
40 PRINT AT L,1;"P";AT L+A,1;"00";AT 10
,1;"I"
110 FOR I=1 TO 10
120 PRINT INK 5;AT L,A*I;" ■"
122 BEEP 0.03,0: PAUSE 10
125 LET X=X-1
128 IF NOT X THEN BEEP 0.2,10: BEEP 0.2,
10: BEEP 0.2,10: PAUSE 4E4
130 NEXT I
135 PRINT AT 5,1;" ";AT 8,1;" ";AT 11,1;
" ";AT 14,1;" "
140 LET L=L+A
160 IF L<>16 THEN GO TO 110
170 LET X=INT (RND*66)
180 GO TO 35

```

```

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

```


421

REFERENCES :

Mode FAST.

Difficulté de programmation : facile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

Le nombre de joueurs est tout d'abord entré en ligne 20.

La ligne 90 affiche le numéro de joueur et le nombre d'essais déjà effectués.

Un sous-programme placé en ligne 4000, utilisé 3 fois dans la boucle J (140-160), réalise le lancer de dés.

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

La variable R\$ gère les 2^e et 3^e tirages. L'impression du résultat est réalisée dans tous les cas par les lignes 100 et 110.

Les tests pour le nombre d'essais et pour le changement de joueur sont effectués aux lignes 170 et 190.

POUR JOUER :

On lance l'exécution du programme par RUN ce qui conduit à l'affichage NBRE DE JOUEURS. On entre le nombre de joueurs, et l'on tape NEWLINE. L'écran s'efface et affiche JR N° 1. ESSAI A, 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 NEWLINE. 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 NEWLINE : par exemple, pour retirer A et C on tapera AC suivi de NEWLINE.

Les joueurs suivants procéderont de la même façon.

EXTENSIONS POSSIBLES :

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

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

```

10 PRINT "NBRE DE JOUEURS"
20 INPUT N
30 LET S=SGN PI
40 LET I=S
50 LET K=NOT PI
60 CLS
65 LET R$="ABC"
70 GOTO 140
80 LET K=K+S
85 PRINT AT 5,12;"421"
87 PRINT AT 2,12;" "
90 PRINT AT 4,5;"JR. NO. ";I;"
  ESSAI ";K
100 PRINT AT 8,4;A;TAB 8;B;TAB
12;C
110 PRINT AT 10,4;"A";TAB 8;"B"
  TAB 12;"C"
120 PRINT AT 15,S;"ABC ?"
130 INPUT R$
133 IF R$="" THEN GOTO 180
137 LET R$=R$+"DDD"
140 FOR J=S TO INT PI
145 GOSUB 4000
150 GOSUB (CODE (R$(J))-37)*100
0
160 NEXT J
170 IF K<INT PI THEN GOTO 80
180 LET I=I+S
190 IF I<=N THEN GOTO 50
200 GOTO 40
1000 LET A=D
1010 RETURN
2000 LET B=D
2010 RETURN
3000 LET C=D
3010 RETURN
4000 LET D=INT (RND*5+S)
4010 RETURN

```

421

JR. NO. 2; ESSAI 1

3 3 5

A B C

ABC ?

```

10 PRINT "NOMBRE DE JOUEURS"
20 INPUT N
30 LET S=1
40 LET I=S
50 LET K=0
60 CLS
65 LET R$="ABC"
70 GO TO 140
80 LET K=K+S
85 PRINT FLASH 1;AT 5,12;"421"
87 PRINT INVERSE 1;AT 2,12;" "
90 PRINT AT 4,5;"JR. NO. ";I;" ;ESSAI "
  ;K
100 PRINT AT 8,4;A; BEEP 1,5*A-20; PRINT
  AT 8,8;B; BEEP 1,5*B-20; PRINT AT 8,12;C
  : BEEP 1,5*C-20
110 PRINT INK 4;AT 10,4;"A";AT 10,8;"B";
  AT 10,12;"C"
120 PRINT AT 15,S;"ABC ?"
130 INPUT R$
133 IF R$="" THEN GO TO 180
137 LET R$=R$+"DDD"
140 FOR J=S TO 3
145 GO SUB 4000
150 GO SUB (CODE R$(J)-64)*1000
160 NEXT J
170 IF K<3 THEN GO TO 80
180 LET I=I+S
190 IF I<=N THEN GO TO 50
200 GO TO 40
1000 LET A=D
1010 RETURN
2000 LET B=D
2010 RETURN
3000 LET C=D
3010 RETURN
4000 LET D=INT (RND*5+S)
4010 RETURN

```

421

JR. NO. 2 ;ESSAI 1

2 1 4

A B C

ABC ?

Simon

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : facile.

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

Difficulté du jeu : moyenne.

LE JEU :

Des nombres vont apparaître un par un.

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

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

LE PROGRAMME :

La ligne 15 appelle un sous-programme (lignes 900 à 1010) qui donne la règle du jeu.

La ligne 20 fabrique les chiffres, qui sont ensuite stockés dans une variable numérique A\$ (ligne 30).

La ligne 40 permet l'impression du dernier chiffre au hasard sur la sixième ligne de l'écran, et la durée de l'affichage est déterminée par la ligne 50.

Les lignes 80 et 90 attendent la réponse du joueur, qui est comparée à la réponse correcte en ligne 100.

La ligne 140 marque les points.

La ligne 150 contrôle le meilleur score.

Les affichages de fin de partie sont effectués de la ligne 200 à la ligne 250.

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 NEWLINE. 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 deux chiffres déjà obtenus, suivis de NEWLINE.

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 ou N) ? ; on tape O pour continuer.

On peut quitter le programme à tout moment, en tapant BREAK avant l'affichage d'un chiffre, ou en tapant une lettre, suivie de NEWLINE.

EXTENSIONS POSSIBLES :

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

```

10 LET M=NOT PI
12 LET A$=""
15 GOSUB 900
17 LET H=NOT PI
18 PAUSE 90
20 LET X=INT (RND*10)
30 LET A$=A$+STR$ X
40 PRINT AT 6,RND*33;X
50 PAUSE 20+20*RND
70 GOSUB 900
80 PRINT AT 8,SGN PI;"VOTRE SE
QUENCE ?"
90 INPUT R$
100 IF R$<>A$ THEN GOTO 200
120 PRINT AT 10,SGN PI;"EXACT .
ATTENTION"
140 LET H=H+SGN PI
150 IF M<H THEN LET M=H
160 GOTO 18
200 CLS
210 PRINT "VOTRE SCORE ";H
220 PRINT "MEILLEUR SCORE ";M
230 PRINT "AUTRE PARTIE (O OU N
)?"
240 INPUT R$
250 IF R$="N" THEN STOP
260 GOTO 12
900 CLS
1000 PRINT AT PI,SGN PI;"VOUS DE
VEZ REPETER TOUS LES NOMBRES (UN
DE PLUS A CHAQUE ESSAI)"
1010 RETURN

```

VOUS DEVEZ REPETER ~~TOUS~~ LES NOM
BRES (UN DE PLUS A CHAQUE ESSAI)

5

VOTRE SEQUENCE ?
EXACT . ATTENTION

```

10 LET M=0
12 LET A$=""
15 GO SUB 900
17 LET H=0
18 PAUSE 90
20 LET X=INT (RND*10)
30 LET A$=A$+STR$ X
40 PRINT AT 6,RND*33;X: BEEP 1,5*X-20
50 PAUSE 20+20*RND
70 GO SUB 900
80 PRINT AT 8,1;"VOTRE SEQUENCE ?"
90 INPUT R$
100 IF R$<>A$ THEN GO TO 200
120 PRINT AT 10,1;"EXACT . ATTENTION"
140 LET H=H+1
150 IF M<H THEN LET M=H
160 GO TO 18
200 CLS
210 PRINT "VOTRE SCORE ";H
220 PRINT "MEILLEUR SCORE ";M
230 PRINT "AUTRE PARTIE (O OU N)?"
240 INPUT R$
250 IF R$="N". THEN STOP
260 GO TO 12
900 CLS
1000 PRINT AT 3,1;"VOUS DEVEZ REPETER TOU
S LES NOMBRES (UN DE PLUS A CHAQUE FOIS)"
1010 RETURN

```

VOUS DEVEZ REPETER TOUS LES NOM
BRES (UN DE PLUS A CHAQUE FOIS)

VOTRE SEQUENCE ?
EXACT . ATTENTION

Le jeu de la fève

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : facile.

Catégorie de jeu : réflexion (deux joueurs).

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

LE PROGRAMME :

Les boucles emboîtées I (20-80) et J (30-70) dessinent le terrain de jeu.

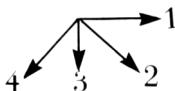
La ligne 93 réalise le test de fin de partie.

La ligne 125 permet d'exécuter l'un des sous-programmes de déplacement.

Les lignes 130 et 132 assurent le changement de joueur.

POUR JOUER :

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



Le joueur 1 entre son déplacement, et tape NEWLINE.

Le mouvement correspondant est réalisé, puis on obtient l'affichage JR 2 et DIR.

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

EXTENSIONS POSSIBLES :

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

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

```

10 LET A=NOT PI
20 FOR J=A TO 9
30 FOR I=A TO 9
40 PRINT AT INT PI,I+I+INT PI:
I;AT J+4,SGN PI;J;AT J+4,I+I+INT
PI;"*"
50 NEXT I
60 NEXT J
80 LET X=4
84 LET Y=INT PI
90 LET J=SGN PI
96 PRINT AT X,Y;CHR$ 128
98 IF X=13 AND Y=21 THEN STOP
100 PRINT AT VAL "16",SGN PI;"J
R."
110 PRINT "DIR. 1,2,3,4 ?"
120 INPUT I
125 GOSUB I*1000
130 LET J=J+SGN PI
132 IF J=INT PI THEN GOTO 90
140 GOTO 96
1000 LET Y=Y+2
1010 RETURN
2000 GOSUB 1000
3000 LET X=X+SGN PI
3010 RETURN
4000 GOSUB 3000
4010 LET Y=Y-2
4020 RETURN

```

```

0 1 2 3 4 5 6 7 8 9
0 * * * * *
1 * * * * *
2 * * * * *
3 * * * * *
4 * * * * *
5 * * * * *
6 * * * * *
7 * * * * *
8 * * * * *
9 * * * * *

```

```

DIR. 1
DIR. 1,2,3,4 ?

```

```

10 LET A=0
20 FOR J=0 TO 9
30 FOR I=0 TO 9
40 PRINT AT 3,2*I+3;I;AT J+4,1;J; INK 3
;AT J+4,2*I+3;"■"
50 NEXT I
60 NEXT J
82 LET X=4
84 LET Y=3
90 LET J=1
96 PRINT INVERSE 1;AT X,Y;"*"
98 IF X=13 AND Y=21 THEN STOP
100 PRINT AT 16,1;"JOUEUR ";J
110 PRINT "DIRECTION 1,2,3,4 ?"
120 INPUT I
125 GO SUB I*1000
130 LET J=J+1
132 IF J=3 THEN GO TO 90
140 GO TO 96
1000 LET Y=Y+2
1010 RETURN
2000 GO SUB 1000
3000 LET X=X+1
3010 RETURN
4000 GO SUB 3000
4010 LET Y=Y-2
4020 RETURN

```

```

0 1 2 3 4 5 6 7 8 9
0 * * * * *
1 * * * * *
2 * * * * *
3 * * * * *
4 * * * * *
5 * * * * *
6 * * * * *
7 * * * * *
8 * * * * *
9 * * * * *

```

```

JOUEUR 2
DIRECTION 1,2,3,4 ?

```

Jeu de l'oie aléatoire

REFERENCES :

Mode SLOW (FAST en rajoutant une pause).

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

LE PROGRAMME :

La ligne 50 permet d'appeler le sous-programme (1000 à 1060) 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 55 s'occupe du changement de joueur, et la ligne 60 effectue le lancer de dé.

La ligne 75 traite le cas où l'un des joueurs termine.

La ligne 100 tire un nombre aléatoire entre 0 et 3, qui servira s'il y a lieu pour déterminer le vrai déplacement du joueur, en envoyant dans l'un des sous-programmes de "chance".

Les lignes 1000 et 1005 font avancer les pions, et les lignes 1020 à 1040 redessinent à chaque affichage le terrain de jeu en entier.

POUR JOUER :

On lancera le jeu avec RUN. Le terrain de jeu est alors dessiné, puis on voit apparaître l'affichage JR 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 JR 2 :...

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

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

On peut aussi s'arrêter à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :

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

ZX-81

```

10 LET X=NOT PI
30 LET Y=X
40 LET J=SGN PI
45 LET I=X
50 GOSUB 1000
55 LET J=ABS (J-SGN PI)
60 LET I=INT (6*RND+SGN PI)
75 IF X=CODE "3" OR Y=CODE "3"
THEN PRINT "BRAVO ";J
80 PRINT "JR ";J+SGN PI;" :";I
85 PAUSE 4E4
90 GOSUB 1000
100 LET I=INT (4*RND)
110 GOSUB I*(RND<.5)*50+150
120 GOTO CODE "M"
150 RETURN
200 LET I=-I
210 RETURN
250 LET I=I+I
260 RETURN
300 IF NOT J THEN LET X=INT (30
*RND)
310 IF J THEN LET Y=INT (30*RND
)
320 RETURN
1000 IF NOT J THEN LET X=X+I*(X+
I<=CODE "3")
1005 LET Y=Y+J*I*(Y+I<=CODE "3")
1010 FOR I=NOT PI TO CODE "3"
1020 PRINT AT INT PI,I;CHR$ (COD
E "0"+I)
1030 PRINT TAB I;CHR$ (CODE "■"+
CODE "1"*(I=X))
1040 PRINT TAB I;CHR$ (CODE "■"+
CODE "2"*(I=Y))
1050 NEXT I
1060 RETURN

```

0123456789ABCDEF GHIJKLMNOPQRSTUVWXYZ

1

JR 2 : 6

ZX-SPECTRUM

```

10 LET X=0
30 LET Y=X
40 LET J=1
45 LET I=X
50 GO SUB 1000
55 LET J=ABS (J-1)
60 LET I=INT (6*RND+1)
75 IF X=31 OR Y=31 THEN PRINT "BRAVO ";
W
80 PRINT "JOUER ";J+1;" :";I: BEEP 0.
2,10+10*J
85 PAUSE 4E4
90 GO SUB 1000
100 LET I=INT (4*RND)
110 GO SUB I*(RND<.5)*50+150
120 GO TO 50
150 RETURN
200 LET I=-I
210 RETURN
250 LET I=I+I
260 RETURN
300 IF NOT J THEN LET X=INT (30*RND)
310 IF J THEN LET Y=INT (30*RND)
320 RETURN
1000 IF NOT J THEN LET X=X+I*(X+I<=31)
1005 LET Y=Y+J*I*(Y+I<=31)
1010 FOR I=0 TO 31
1020 PRINT AT 3,I;I AND I<=9: PRINT AT 3,
I;CHR$ (55+I) AND I>9
1030 PRINT INK 1;AT 4,I;CHR$ (CODE "■"-94
*(I=X))
1040 PRINT INK 2;AT 5,I;CHR$ (CODE "■"-93
*(I=Y))
1050 NEXT I
1060 RETURN

```

0123456789ABCDEF GHIJKLMNOPQRSTUVWXYZ

1

2

JOUER 1 : 1

Niveau 2

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Un dé	Tirage d'un dé, avec graphisme.
Visions	Un jeu sur les mots, pour l'ordinateur !
Plus grand, plus petit	Un jeu de déduction intéressant, pour un joueur.
L'électricien fou	Logique et difficile, pour un seul joueur.
La course au 20	Jeu de réflexion, pour deux joueurs ou plus.
Dessin	Une méthode de codage originale.
Dé truqué	A utiliser, si vous êtes malhonnête, dans d'autres programmes.
Le jeu de saute-mouton	Un jeu de réflexion, pour deux joueurs.
Lettres	Un jeu sur les mots, assez simple (2 joueurs).
Espion	Offrez un ZX à votre fiancée, et dites-lui votre amour en code.
Accrochez les wagons	Pour les plus jeunes ; très visuel.
Tirage d'un dé truqué	Révissez vos connaissances en statistiques.
Ordre	Un peu d'astuce pour remettre en ordre, le plus vite possible...
Course de chevaux	Assez visuel ; un jeu de hasard pour plusieurs joueurs.
Le jeu de Marienbad	Un jeu de logique pour deux joueurs, dans une version étendue.
Idem	Un jeu de calcul mental, pour 2 joueurs (plusieurs niveaux).
Mastermind	Avec des chiffres, mais très classique, un seul joueur.
A travers	Un jeu de stratégie élémentaire, pour 2 joueurs.
Reines sur l'échiquier	Un vieux problème, pour vous aider à réfléchir.
Le jeu de la vie	Un grand classique : deux versions, manuelle et automatique.

Dimensionnement d'un tableau Recherche d'un élément dans un tableau et dans une chaîne

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

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

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

On peut aussi concevoir des tableaux dont les éléments sont des chaînes de caractères, et non plus des nombres, mais il devient alors indispensable de préciser la longueur maximum des chaînes autorisées (c'est-à-dire le nombre maximum de caractères de chaque mot).

Ainsi DIM NOM\$ (5,10) réserve 5 emplacements en mémoire, pour des mots de 10 lettres au maximum.

LA RECHERCHE D'UN ELEMENT DANS UN TABLEAU : R (1)

Si l'on a un tableau à une dimension, défini par une instruction comme DIM R (30), le lème élément numérique de ce tableau est repéré par R (1) ; il sera alors manipulé exactement comme une variable numérique 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 même tableau R, etc.

LA RECHERCHE D'UN ELEMENT DANS UNE CHAÎNE : R\$ (1)

Dans le Basic Sinclair, une chaîne de caractères est traitée un peu comme un tableau, qu'il ne serait pas utile de déclarer, ayant pour dimension le nombre de caractères de la chaîne.

Cette façon de faire permet une extraction facile du lème élément de la chaîne R\$, qui s'appellera simplement R\$ (1) ; elle présente aussi l'avantage de permettre le remplacement facile d'un élément de la chaîne R\$, par une simple affectation.

Par exemple : soit l'affectation LET R\$="BONJOUR".

PRINT R\$ (1) affichera B, PRINT R\$ (3) affichera N. Si on écrit LET R\$ (3)="U", on aura alors PRINT R\$ qui affichera BONJOUR.

Les Basic usuels ne permettent pas une telle simplicité dans le traitement des chaînes, l'instruction correspondante à R\$ (3) étant MID\$(R\$,3,1) ; on trouvera, en revanche, plus de souplesse dans les tableaux de chaînes (difficiles à manipuler sur le ZX).

Un dé

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : facile.

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

Difficulté du jeu : aucune.

LE JEU :

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

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

LE PROGRAMME :

La ligne 10 utilise le générateur aléatoire du ZX pour tirer un nombre de 1 à 6.

La boucle J (40-60) dessine les bords du dé, et les lignes 30 et 65 dessinent le haut et le bas.

Les 3 lignes 70, 80 et 90 sont utilisées pour le dessin des points.

La ligne 120 permet le cas échéant de refaire un tirage.

POUR JOUER :

On lance le dé par RUN ; le résultat est affiché sous forme symbolique, comme sur un véritable dé.

La question AUTRE TIRAGE (O ou N) est ensuite affichée. Si l'on répond O, un autre lancer sera effectué.

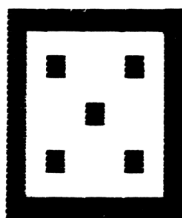
EXTENSIONS POSSIBLES :

On pourra utiliser ce programme en tant que sous-programme chaque fois que l'on a besoin de simuler un lancer de dé.

```

5 CLS
10 LET A=INT (RND*6+1)
30 PRINT AT 5,8;"████████"
40 FOR J=6 TO 12
50 PRINT AT J,8;"■"
60 NEXT J
65 PRINT AT 13,8;"████████"
70 PRINT AT 7,10;"■" AND A<>1;
TAB 14;"■" AND A=4
80 PRINT AT 9,10;"■" AND A
=6;TAB 12;"■" AND A/2<>INT (A/2)
90 PRINT AT 11,10;"■" AND A=4
;TAB 14;"■" AND A<>1
100 PRINT AT 17,1;"AUTRE TIRAGE
(O/N)"
110 INPUT R$
120 IF R$(1)="O" THEN RUN

```

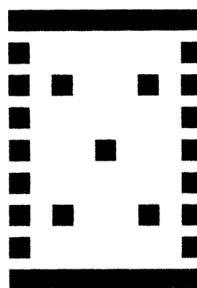


AUTRE TIRAGE (O/N)

```

5 BORDER 6: PAPER 4: CLS
10 LET A=INT (RND*6+1): BEEP .05,5: BEE
P .05,5: BEEP .05,5: BEEP .05,5
,5
30 PRINT INK 1;AT 5,8;"████████"
40 FOR J=6 TO 12
50 PRINT INK 1;AT J,8;"■"; INK 5;"████
■"; INK 1;"■"
60 NEXT J
65 PRINT INK 1;AT 13,8;"████████"
70 PRINT INK 2;AT 7,10;"■" AND A<>1;AT
7,14;"■" AND A=4
80 PRINT INK 2;AT 9,10;"■" AND A=6;
AT 9,12;"■" AND A/2<>INT (A/2)
90 PRINT INK 2;AT 11,10;"■" AND A=4;AT
11,14;"■" AND A<>1
100 PRINT FLASH 1;AT 17,1;"AUTRE TIRAGE
(O OU N) ?"
110 PAUSE 4E4
120 IF INKEY$="O" THEN RUN

```



AUTRE TIRAGE (O OU N) ?

Visions

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : facile.

Catégorie de jeu : dessin.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

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

La ligne 50 permet de transformer en nombre chaque lettre du mot entré au début (ligne 20). Ce nombre sera ensuite traduit en base 2 par une boucle J (lignes 70 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 ZX dès que l'on aura frappé NEW-LINE.

Pour recommencer, il suffira de frapper une touche quelconque et BREAK pour finir. Je vous laisse savourer le dessin qu'à réalisé ZX lorsqu'il a connu le prénom de l'auteur de ce livre.

EXTENSIONS POSSIBLES :

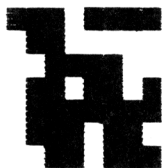
La même idée peut être reprise en introduisant des éléments de symétrie dans les dessins, en ajoutant la couleur. On peut aussi imaginer de coder par cette méthode un certain nombre de dessins, que l'on introduirait au fur et à mesure dans la machine et l'on pourra alors créer un jeu dans lequel ZX deviendra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

```

10 PRINT AT SGN PI,NOT PI;"QUE
L DESSIN :";
20 INPUT R$
30 PRINT R$
40 FOR I=SGN PI TO LEN R$
50 LET N=9*(CODE R$(I))-340
60 LET J=SGN PI
70 LET S=(N/2)-INT (N/2)
80 LET N=INT N/2
90 IF S THEN PRINT AT I+5,J+12
"■"
100 LET J=J+SGN PI
110 IF J=9 THEN GOTO 70
120 NEXT I
130 PAUSE 4E4
150 CLS
160 GOTO 10

```

QUEL DESSIN : JACQUES



```

10 PRINT AT 1,0;"QUEL DESSIN :";
20 INPUT R$
30 PAPER SQR ((LEN (R$))*RND+1): CLS :
PRINT R$
40 FOR I=1 TO LEN R$
50 LET N=9*(CODE R$(I))-27)-340
60 LET J=1
70 LET S=(N/2)-INT (N/2)
80 LET N=INT N/2: BEEP 0.05,5*SQR ABS N
90 IF S THEN PRINT INK 9;AT I+5,J+12;"■"
"
100 LET J=J+1
110 IF J=9 THEN GO TO 70
120 NEXT I
130 PAUSE 4E4
150 BORDER 7*RND: CLS
160 GO TO 10

```

JACQUES



Plus grand, plus petit

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : facile.

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

Difficulté du jeu : difficile.

LE JEU :

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

LE PROGRAMME :

La ligne 30 interdit de demander des nombres de plus de 8 chiffres.

Le nombre à deviner est placé, en ligne 50, dans une variable alphanumérique X\$.

Les comparaisons avec le nombre essayé sont faites à la ligne 95 (qui teste si l'on a trouvé juste), et dans la boucle I (120-170) où l'on fait la comparaison "lettre à lettre".

L'affichage du résultat est réalisé grâce aux lignes 180 et 190.

Le compteur du nombre d'essais est en ligne 195.

POUR JOUER :

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

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

Le numéro de l'essai s'affiche, ainsi que la question NBRE PRESUME ; on répond en entrant un nombre, suivi de NEWLINE.

Le résultat est alors affiché sous la forme ...CHIFFRES PLUS GRANDS, et ...CHIFFRES PLUS PETITS.

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 ZX affichera TROUVE... EN..., ESSAIS. Puis AUTRE PARTIE (O ou N) ? ; on fera O suivi de NEWLINE 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 ZX qui tiendra alors le rôle de décodeur.

```

5 CLS
10 PRINT "NBRE DE CHIFFRES ";
20 INPUT C
30 IF C>8 THEN RUN
40 PRINT C
45 LET E=SGN PI
50 LET X$=STR$ (INT (RND*9+*C+
10** (C-E)))
60 PRINT AT 5,SGN PI;"ESSAI NO
";E
70 PRINT AT 8,SGN PI;"NBRE PRE
SUME ";
80 INPUT P$
90 PRINT P$
95 IF P$=X$ THEN GOTO 200
100 LET A=NOT PI
110 LET B=A
120 FOR I=SGN PI TO C
140 IF P$(I)>X$(I) THEN LET A=A
+SGN PI
150 IF P$(I)<X$(I) THEN LET B=B
+SGN PI
170 NEXT I
180 PRINT AT 13,SGN PI;A;" CHIF
FRES PLUS GRANDS"
190 PRINT AT 15,SGN PI;B;" CHIF
FRES PLUS PETITS"
195 LET E=E+SGN PI
197 GOTO 60
200 CLS
210 PRINT AT 5,SGN PI;"TROUVE "
;X$;" EN ";E;" ESSAIS"
220 PRINT AT 8,SGN PI;"AUTRE PA
RTIE (O/N)?"
230 INPUT R$
240 IF R$(1)="O" THEN RUN

```

NBRE DE CHIFFRES 4

ESSAI NO 2

NBRE PRESUME 1234

1 CHIFFRES PLUS GRANDS

2 CHIFFRES PLUS PETITS

```

5 CLS
10 PRINT "NOMBRE DE CHIFFRES ";
20 INPUT C
30 IF C>8 THEN RUN
40 PRINT C
45 LET E=1
50 LET X$=STR$ (INT (RND*9+C+10^(C-E)))
60 PRINT INVERSE 1;AT 5,1;"ESSAI NO ";
FLASH 1;E
70 PRINT AT 8,1;"NOMBRE PRESUME ";
80 INPUT P$
90 PRINT P$
95 IF P$=X$ THEN GO TO 200
100 LET A=0
110 LET B=A
120 FOR I=1 TO C
140 IF P$(I)>X$(I) THEN LET A=A+1
150 IF P$(I)<X$(I) THEN LET B=B+1
170 NEXT I
180 PRINT PAPER 4;AT 13,1;A;" CHIFFRES P
LUS GRANDS"
190 PRINT PAPER 5;AT 15,1;B;" CHIFFRES P
LUS PETITS"
195 LET E=E+1: BEEP 0.2,10
197 GO TO 60
200 CLS
210 PRINT AT 5,1;"TROUVE ";X$;" EN ";E;"
ESSAIS"
220 PRINT AT 8,1;"AUTRE PARTIE (O OU N)
?"
230 INPUT R$
240 IF R$="O" THEN RUN

```

NOMBRE DE CHIFFRES 4

ESSAI NO 3

NOMBRE PRESUME 4490

2 CHIFFRES PLUS GRANDS

2 CHIFFRES PLUS PETITS

L'électricien fou

REFERENCES :

Mode FAST.

Difficulté de programmation : difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez difficile.

LE JEU :

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

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

LE PROGRAMME :

Une table A, en ligne 15, contiendra l'état (allumé ou éteint) des 10 fenêtres.

La boucle I (20-50) et l'autre boucle I (62-70) dessinent l'immeuble dans son état initial.

La ligne 85 regarde si tout est éteint, auquel cas l'on ira en ligne 229 afficher le nombre d'essais.

La ligne 100 attend l'indication de l'interrupteur manœuvré.

Les lignes 110 à 180 définissent le résultat de la manœuvre.

POUR JOUER :

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

A la question INTER, on répond en indiquant le numéro de l'interrupteur à manœuvrer, suivi de NEWLINE.

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

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

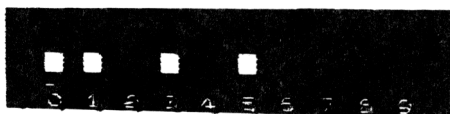
EXTENSIONS POSSIBLES :

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

```

10 LET S=SGN PI
12 LET M=NOT PI
15 DIM A(CODE "####")
20 FOR I=S TO CODE "F"
30 PRINT AT NOT PI,I;" "AT S,
I;" "AT S+S,I;" "AT INT PI,I;"
"AT INT PI+S,I;" "
50 NEXT I
60 LET T=NOT PI
62 FOR I=S TO CODE "###"
64 LET T=T+A(I)
65 PRINT AT S+S,2*I+S;" "
70 PRINT AT S+S,2*I+S;" " AND
A(I)=NOT PI;AT INT PI+S,2*I+S;CH
R$(155+I)
80 NEXT I
85 IF T=CODE "###" THEN GOTO COD
E "FAST"
90 PRINT AT CODE "2",S;"INTER
"
100 INPUT N
110 LET N=N+S
120 LET M=M+S
130 IF A(N)=S THEN GOTO CODE "C
OS "
140 LET A(N)=S
150 LET R=TAN (RND+N/RND-N)-3IN
(RND/N)+336*SIN (CODE "###" *N)
160 LET N=INT (CODE "###" *(R-INT
R))
165 IF NOT N THEN LET N=INT (S+
CODE "###" *RND)
170 IF A(N)=S THEN GOTO CODE "C
OS "
180 LET A(N)=S
190 GOTO CODE "U"
200 LET A(N)=NOT PI
210 GOTO CODE " "
220 PRINT AT CODE "2",S;M;" ESS
AIS"

```

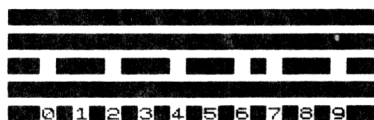


INTER

```

10 LET S=1
12 LET M=0
15 DIM A(10)
20 FOR I=S TO 23
30 PRINT INK 5;AT 0,I;" ";AT S,I;" ";AT
S+S,I;" ";AT 3,I;" ";AT 4,I;" "
50 NEXT I
60 LET T=0
62 FOR I=S TO 10
64 LET T=T+A(I)
65 PRINT INK 5;AT S+S,2*I+S;" "
70 PRINT PAPER 2;AT S+S,2*I+S;" " AND A
(I)=0;AT 4,2*I+S; PAPER 3; BRIGHT 1;I-1
80 NEXT I
85 IF T=10 THEN GO TO 229
90 PRINT AT 12,S;"INTERRUPTEUR NUMERO :
"
100 INPUT N: BEEP 1,10: PRINT FLASH 1; I
NK 3;AT 12,22;N
110 LET N=N+S
120 LET M=M+S
130 IF A(N)=S THEN GO TO 200
140 LET A(N)=S
150 LET R=TAN (RND+N/RND-N)-SIN (RND/N)+
336*SIN (8*N)
160 LET N=INT (10*(R-INT R))
165 IF NOT N THEN LET N=INT (S+10*RND)
170 IF A(N)=S THEN GO TO 200
180 LET A(N)=S
190 GO TO 60
200 LET A(N)=0
210 GO TO 150
229 PRINT FLASH 1;AT 15,S;"TROUVE EN ";M
;" ESSAIS"

```



INTERRUPTEUR NUMERO :

La course au 20

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : facile.

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

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

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

La variable F, dimensionnée en ligne 60, servira à enregistrer le nombre d'étoiles prises dans chaque rangée.

La variable T contiendra le nombre total d'étoiles enlevées.

La boucle I (70-90) imprime les numéros de ligne.

La ligne 125 va contrôler qu'il reste suffisamment d'étoiles sur la ligne indiquée.

Les boucles I (170-210) et J (180-200) suppriment les étoiles ; la ligne 240 teste la fin de partie.

POUR JOUER :

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

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

EXTENSIONS POSSIBLES :

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

```

10 LET A=SGN PI
30 FOR I=A TO 20
40 PRINT AT INT PI,I+INT PI;"*
"AT 5,I+INT PI;"*";AT 7,I+INT P
I;"*";AT 9,I+INT PI;"*"
50 NEXT I
55 LET T=NOT PI
60 DIM F(4)
70 FOR I=A TO 4
75 PRINT AT I+I+A,A,I
80 LET F(I)=NOT PI
90 NEXT I
95 LET B=A
100 PRINT AT 12,A;"JOUR. ";B
110 PRINT AT 14,A;"LGN. ";CASES?
120 INPUT L
125 INPUT C
125 IF C+F(L)>CODE "=" THEN GOT
O 110
165 LET F(L)=F(L)+C
167 LET T=T+C
170 FOR I=A TO 4
180 FOR J=NOT PI TO F(I)
190 PRINT AT I+I+A,J+INT PI;" "
200 NEXT J
210 NEXT I
220 LET B=B+A
230 IF B=INT PI THEN LET B=A
240 IF T<80 THEN GOTO 100
300 CLS
310 PRINT "LE ";B;" GAGNE"

```

```

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

```

JR. 2
LGN. ,CASES?

```

10 LET A=1
30 FOR I=A TO 20
40 PRINT INK 2;AT 3,I+3;"■"; INK 3;AT 5
,I+3;"■"; INK 4;AT 7,I+3;"■"; INK 5;AT 9,
I+3;"■"
50 NEXT I
55 LET T=0
60 DIM F(4)
70 FOR I=A TO 4
75 PRINT AT 2*I+A,A;I
80 LET F(I)=0
90 NEXT I
95 LET B=A
100 PRINT AT 12,A;"JOUEUR ";B
110 PRINT AT 14,A;"LIGNE,CASES?"
120 INPUT L: IF L>4 THEN GO TO 110
123 INPUT C
125 IF C+F(L)>20 THEN GO TO 110
165 LET F(L)=F(L)+C
167 LET T=T+C
170 FOR I=1 TO 4
180 FOR J=1 TO F(I)
190 PRINT BRIGHT 1; PAPER I+1;AT 2*I+A,J
+3;"*"
200 NEXT J
210 NEXT I
220 BEEP 1,10: LET B=B+A
230 IF B=3 THEN LET B=A
240 IF T<80 THEN GO TO 100
300 CLS
310 PRINT "LE JOUEUR ";B;" GAGNE": BEEP
1,0

```

```

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

```

JOUEUR 1
LIGNE ,CASES?

Dessin

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : facile.

Catégorie de jeu : dessin.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

Le codage du dessin est, dans le programme retenu, figé dans une variable A\$, définie en ligne 10.

On pourra éventuellement essayer d'autres dessins en entrant directement leurs codes : il suffit de modifier la ligne concernée par `10 INPUT A$`.

La boucle N (20-50) réalise le décodage de A\$.

La ligne 40 détermine les positions de la tête d'écriture.

La ligne 45 détecte le caractère 0, qui sert à indiquer le retour à la ligne.

POUR JOUER :

Le dessin devra auparavant être défini par une variable A\$, en utilisant les caractères graphiques, et en donnant la position d'impression de chaque caractère, par rapport au début de la ligne, avec un chiffre de 1 à 9, ou une lettre (A : 10 ; B : 11 ; C : 12 ; etc.).

Le passage à la ligne suivante sera indiquée par le caractère 0.

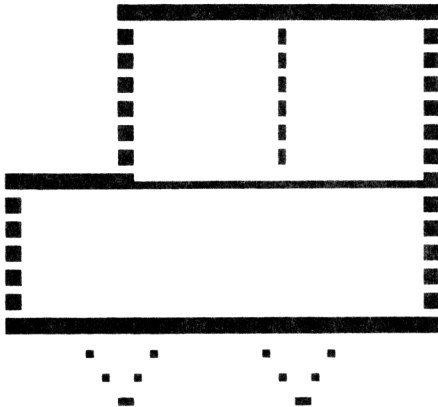
EXTENSIONS POSSIBLES :

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

```

10 LET A$="A
0A K T 0A K T 0A K T 0A K T 0A
K T 0A K T 03
03 T 03 T 03 T 03 T 03
T 03
S,C,J,N,09,B,K,M,0A,L,0"
20 FOR N=SGN PI TO LEN A$
30 PRINT A$(N) AND (CODE A$(N)
<=10 OR CODE A$(N)>=128);
40 IF CODE A$(N)>=29 AND CODE
A$(N)<=63 THEN PRINT TAB (CODE A
$(N)-28);
45 IF CODE A$(N)=28 THEN PRINT
""
50 NEXT N

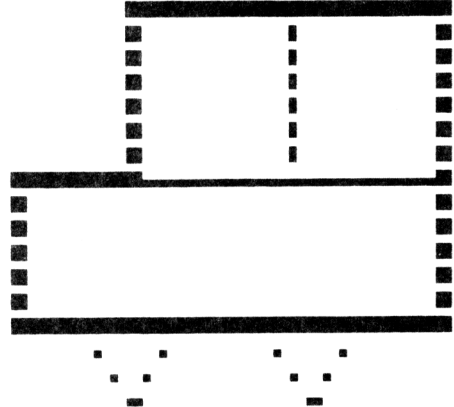
```



```

10 LET A$="A
0A K T 0A K T 0A K T 0A K T 0A
K T 0A K T 03
03 T 03 T 03 T 03 T 03
T 03 T 03
S,C,J,N,09,B,K,M,0A,L,0"
20 FOR N=1 TO LEN A$
30 PRINT INK 2;A$(N) AND (CODE A$(N)>=1
28 AND CODE A$(N)<=143);
40 IF (CODE A$(N)>=49 AND CODE A$(N)<=5
7) OR (CODE A$(N)>=65 AND CODE A$(N)<=90)
THEN PRINT INK 2;TAB (CODE A$(N)-48-7*(C
ODE A$(N)>=65));
45 IF CODE A$(N)=48 THEN PRINT ""
50 NEXT N

```



Dé truqué

REFERENCES :

Mode FAST.

Difficulté de programmation : facile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

Il utilise le générateur aléatoire du SINCLAIR ZX supposé de qualité convenable (on pourra aisément utiliser tout autre générateur, si besoin est). La boucle I (20-95) s'occupe de l'entrée de chacune des probabilités, avec vérification à chaque entrée et sur le total (ligne 60) ; chaque probabilité est conservée dans une variable indicée E (I) : ligne 40 du programme.

Le lancer du dé truqué est effectué à l'aide d'une autre boucle I (120-140), la ligne permettant le trucage se trouvant en 130. Le résultat est affiché grâce à la ligne 150. Un nouveau tirage pourra être effectué sur simple pression d'une touche (lignes 160 et 170).

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 (sauf BREAK) pour effectuer les tirages suivants. On arrête le programme en tapant BREAK.

EXTENSIONS POSSIBLES :

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

```

10 LET P=NOT PI
15 DIM E(6)
20 FOR I=SGN PI TO 6
30 PRINT AT INT PI,NOT PI;"PRO
BABILITE DU DE ";I
40 INPUT E(I)
50 LET P=P+E(I)
60 IF E(I)<NOT PI OR E(I)>SGN
PI OR P>1.1 THEN GOTO 200
70 PRINT AT NOT PI,5*I-5;"DE "
;I;AT SGN PI,5*I-5;E(I)
80 NEXT I
90 IF P<.9 THEN RUN
95 PAUSE 60
100 RAND
110 LET N=INT (RND*6+SGN PI)
120 FOR I=SGN PI TO 6
130 IF N=I AND RND>E(I) THEN GO
TO 110
140 NEXT I
150 PRINT AT INT PI,NOT PI;"TIR
AGE :";N;"
160 PAUSE 4E4
170 GOTO 110
200 PRINT "IMPOSSIBLE : 0 <= P <=
1"
210 LET P=P-E(I)
215 PAUSE 50
220 CLS
230 GOTO 30

```

DE 1 DE 2 DE 3 DE 4 DE 5 DE 6
0.3 0.2 0.1 .05 .05 0.3

TIRAGE : 6

```

10 LET P=0
15 DIM E(6)
20 FOR I=1 TO 6
30 PRINT FLASH 1;AT 3,0;"PROBABILITE DU
DE ";I
40 INPUT E(I)
50 LET P=P+E(I)
60 IF E(I)<0 OR E(I)>=1 OR P>1.1 THEN G
O TO 200
70 PRINT PAPER 5;AT 0,5*I-5;"DE ";I;AT
1,5*I-5;E(I)
80 NEXT I
90 IF P<.9 THEN RUN
95 PAUSE 60
100 RANDOMIZE
110 LET N=INT (RND*6+1)
120 FOR I=1 TO 6
130 IF N=I AND RND>E(I) THEN GO TO 110
140 NEXT I
150 PRINT PAPER 2; BRIGHT 1; INK 4;AT 3,
0;"TIRAGE "; INVERSE 1; INK 1;" "; F
LASH 1; INK 7;" ";N;" "; INK 1; FLASH 0
;" "; BEEP 1,7*N-24
160 PAUSE 4E4
170 GO TO 110
200 PRINT FLASH 1;"IMPOSSIBLE : 0 <= P <
= 1": BEEP .3,0: BEEP .3,0: BEEP .3,0
210 LET P=P-E(I)
215 PAUSE 50
220 PRINT AT 4,0;"
"
230 GO TO 30

```

DE 1 DE 2 DE 3 DE 4 DE 5 DE 6
.05 .05 0.1 0.3 0.3 0.2

TIRAGE : 3

Le jeu de saute-mouton

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : facile.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

La boucle I (50-80) initialise une table A (I), qui contient les positions des moutons. La boucle I (100-140) dessine le terrain de jeu et les moutons.

Les lignes 150 à 190 s'occupent du texte et des entrées.

La ligne 195 contrôle la validité du coup. Le changement de joueur est réalisé en ligne 220.

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

On commencera par rajouter un contrôle du joueur gagnant.

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

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

```

10 LET A=SGN PI
20 DIM A(8)
30 LET A(SGN PI)=NOT A
40 LET A(8)=NOT A
50 FOR I=A TO INT PI
60 LET A(A+I)=A
70 LET A(4+I)=-A
80 NEXT I
90 LET H=A
100 FOR I=A TO 8
110 PRINT AT 7,INT PI+I+A;I
115 PRINT AT 5,INT PI+I+A;" "
120 IF A(I)=A THEN PRINT AT 5,I
    NT PI+I+A;CHR$ 8
130 IF A(I)=-A THEN PRINT AT 5,
INT PI+I+A;CHR$ 128
140 NEXT I
150 PRINT AT 11,SGN PI;"JR 2"
160 IF NOT (H-A) THEN PRINT AT
11,SGN PI;"JR 1"
170 PRINT "DEBUT,FIN ?"
180 INPUT D
190 INPUT F
195 IF A(D)<>H OR A(F)<>NOT A T
HEN GOTO 150
200 LET A(D)=NOT A
210 LET A(F)=A*H
220 LET H=-H
230 GOTO 100

```

■	■	■	■	■	■	■	■
1	2	3	4	5	6	7	8

JR 1
DEBUT,FIN ?

```

10 LET A=1: PAPER 7: CLS
20 DIM A(8)
30 LET A(1)=0
40 LET A(8)=0
50 FOR I=A TO 3
60 LET A(A+I)=A
70 LET A(4+I)=-A
80 NEXT I
90 LET H=A
100 FOR I=A TO 8
110 PRINT AT 7,3*I+A;I
115 PRINT AT 5,3*I+A;" "
120 IF A(I)=A THEN PRINT INK 2;AT 5,3*I+
A;"■"
130 IF A(I)=-A THEN PRINT INK 4;AT 5,3*I
+A;"■"
140 NEXT I
150 PRINT PAPER 4: FLASH 1;AT 11,1;"JQUE
UR 2"
160 IF NOT (H-A) THEN PRINT PAPER 2: FLA
SH 1;AT 11,1;"JQUEUR 1"
170 PRINT "DEBUT , FIN ?": BEEP 1,12
180 INPUT D
190 INPUT F
195 IF A(D)<>H OR A(F)<>NOT A THEN GO TO
150
200 LET A(D)=NOT A
210 LET A(F)=A*H
220 LET H=-H
230 GO TO 100

```

■	■	■	■	■	■	■	■
1	2	3	4	5	6	7	8

JOUEUR 1
DEBUT , FIN ?

Lettres

REFERENCES :

Mode SLOW ou FAST.

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 ZX choisit au hasard un certain nombre de symboles (voyelles et consonnes) dans l'alphabet latin, et chacun des joueurs devra indiquer la longueur du mot qu'il a réussi à obtenir (l'appareil ne vérifiera pas si le mot est correct !).

Le score total de chaque joueur est affiché ensuite.

LE PROGRAMME :

La boucle I (20-40) choisit dans l'alphabet entre 7 et 12 lettres, en fonction de la variable aléatoire L, tirée en ligne 10. Pour éviter d'avoir trop de consonnes, on utilise en ligne 30 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 50-100 demande à chacun des joueurs la longueur de son mot. Les lignes 110 et 120 comparent les résultats et totalisent le score de chaque joueur, affiché par la ligne 130.

POUR JOUER :

On lance l'exécution du jeu avec RUN.

Les lettres sont alors affichées, puis la question JR 1 LONG. 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 (sauf la touche BREAK qui permettra de 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 ZX, 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.

```

5 LET S=NOT PI
7 LET J=S
10 LET L=INT (6*RND+7)
15 LET A$="AEIOUE"
20 FOR I=SGN PI TO L
30 PRINT AT 2,NOT PI;"LETTRES
";AT 2,7+2*I;CHR$(INT (26*RND+38
));AT 2,7+2*I;A$(INT (6*RND+SGN P
I)) AND RND<.4
40 NEXT I
50 LET I=SGN PI
60 PRINT AT 5,NOT PI;"JR ";I;"
.LONG"
70 INPUT L
80 IF I=SGN PI THEN LET P=L
90 LET I=I+SGN PI
100 IF I<INT PI THEN GOTO 60
110 LET S=S+(P AND P>L)
120 LET J=J+(L AND L>P)
130 PRINT "JR 1 :";S;" JR 2
";J
140 PAUSE 4E4
150 CLS
160 GOTO 10

```

LETTRES I J A D L U S A E X

JR 2:LONG
JR 1:6 JR 2:16

```

5 LET S=0
7 LET J=S
10 LET L=INT (6*RND+7)
15 LET A$="AEIOUE"
20 FOR I=1 TO L
30 PRINT AT 2,0;"LETTRES :";AT 2,7+2*I;
CHR$(INT (26*RND+65));AT 2,7+2*I;A$(INT (6
*RND+1)) AND RND<.4
40 NEXT I
50 LET I=1
60 PRINT AT 5,0;"JOUER ";I;" LONGUEUR
?"
70 INPUT L
80 IF I=1 THEN LET P=L
90 LET I=I+1
100 IF I<3 THEN GO TO 60
110 LET S=S+(P AND P>L)
120 LET J=J+(L AND L>P)
130 PRINT "JOUER 1: ";S;" JOUEUR 2: ";J
: BEEP 1,12
140 PAUSE 4E4
150 CLS
160 GO TO 10

```

LETTRES :U U Q M U O E E

JOUER 2 LONGUEUR ?
JOUER 1:3 JOUEUR 2:5

Espion

REFERENCES :

Mode FAST ou SLOW.

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 messages secrets, suivant une méthode très facile à mettre en œuvre sur ce type d'appareil.

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

LE PROGRAMME :

Le noyau initial (qui sera un entier quelconque) est entré en ligne 20 ; le message est entré à la ligne 40.

Les lignes 50 et 60 permettent de choisir entre les options de codage ou de décodage.

La boucle I (90-140) est utilisée soit pour effectuer le codage, soit pour effectuer le décodage.

La ligne 110 appelle le générateur aléatoire.

La ligne 120 réalise la transposition des lettres.

Les lignes 123 et 125 vérifient que l'on reste bien dans les limites autorisées.

Les résultats seront imprimés par la ligne 150.

Les lignes 1000 à 1030 contiennent le générateur.

POUR JOUER :

Lancer le programme par RUN.

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

Puis taper le message, suivi de NEWLINE.

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

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

EXTENSIONS POSSIBLES :

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

```

10 PRINT "CODE ?"
20 INPUT N
25 LET A=SGN PI/N
30 PRINT "MESSAGE"
40 INPUT A$
50 PRINT "(C)ODER OU (D)ECODER"
60 INPUT R$
90 FOR I=SGN PI TO LEN A$
95 LET T=CODE A$(I)
110 GOSUB 1000
120 LET T=T+A*((R$="C")-(R$="D"))
123 IF T<63 THEN LET T=T-26
125 IF T<68 THEN LET T=T+26
130 LET A$(I)=CHR$(T)
140 NEXT I
150 PRINT "RESULTAT " (A$)
1000 LET A=(A+PI)**8
1010 LET A=A-INT A
1020 LET A=INT (26*A+SGN PI)
1030 RETURN

```

```

CODE ?
MESSAGE
(C)ODER OU (D)ECODER
RESULTAT FGOSCVACDQSS7APWJUNIS

```

```

CODE ?
MESSAGE
(C)ODER OU (D)ECODER
RESULTAT BONJOUROCHEROMONSIEUR

```

```

10 PRINT "CODE :";
20 INPUT N: PRINT N
25 LET A=1/N
30 PRINT "MESSAGE :";
40 INPUT A$: PRINT INK 5;A$
50 PRINT "(C)ODER OU (D)ECODER :";
60 INPUT R$: PRINT R$
90 FOR I=1 TO LEN A$
95 LET T=CODE A$(I)
110 GO SUB 1000
120 LET T=T+A*((R$="C")-(R$="D"))
123 IF T>90 THEN LET T=T-26
125 IF T<65 THEN LET T=T+26
130 LET A$(I)=CHR$(T)
140 NEXT I
150 PRINT "RESULTAT :"; INK 2;A$: BEEP 1
,10: BEEP 1,4: STOP
1000 LET A=(A+PI)^8
1010 LET A=A-INT A
1020 LET A=INT (26*A+1)
1030 RETURN

```

```

CODE : 1234
MESSAGE : SALUT LES COPAINS
(C)ODER OU (D)ECODER : C
RESULTAT : USMDH;USTCQPYQJWG

```

```

CODE : 1234
MESSAGE : USMDH;USTCQPYQJWG
(C)ODER OU (D)ECODER : D
RESULTAT : SALUTTLESTCOPAINS

```


Accrochez les wagons

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : facile.

Catégorie de jeu : solitaire.

Difficulté du jeu : facile.

LE JEU :

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

On tapera, par exemple : BBGDHHH.

LE PROGRAMME :

Les lignes 5 et 10 dessinent la locomotive, la ligne 20 le wagon, et les roues sont en ligne 30.

La position initiale de la locomotive est définie par les lignes 40 et 50, celle du wagon par les lignes 60 et 70.

La ligne 80 évite qu'ils soient initialement trop proches l'un de l'autre.

La ligne 100 réalise l'affichage du wagon et la ligne 110 celui de la locomotive.

Le test de fin est en ligne 115.

Les mouvements à effectuer sont entrés dans une variable alphanumérique R\$, en ligne 130, et sont décodés dans une boucle I (140-170) pour être exécutés en lignes 180 et 190.

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 NEWLINE.

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.

```

5 LET D$=""
10 LET A$=""
20 LET B$=""
30 LET C$="0 0"
35 LET N=NOT PI
40 LET LY=INT (CODE "I")*RND)
50 LET LX=INT (CODE "I")*RND)+5
GN PI
60 LET WY=INT (CODE "I")*RND)
70 LET WX=INT (CODE "I")*RND)+5
GN PI
80 IF ABS (LX-WX)<4 THEN GOTO
CODE "C"
90 CLS
100 PRINT AT WX,WY;B$;AT WX+SGN
PI,WY;C$
110 PRINT AT LX-SGN PI,LY;D$;AT
LX,LY;A$;AT LX+SGN PI,LY+SGN PI
;C$
115 IF LX=WX AND WY=LY-INT PI T
HEN GOTO CODE "ABS "
120 PRINT AT CODE "=";SGN PI;"C
HEMIN (DGGDH..)"
130 INPUT R$
132 LET X=NOT PI
134 LET Y=X
136 LET N=N+SGN PI
140 FOR I=SGN PI TO LEN R$
150 LET X=X+(R$(I)="B")-(R$(I)=
"H")
160 LET Y=Y+(R$(I)="D")-(R$(I)=
"G")
170 NEXT I
180 LET LX=LX+X
190 LET LY=LY+Y
200 GOTO 90
210 PRINT AT CODE "=";SGN PI;"B
RAVO :":N;" COUPS"

```



CHEMIN (DGGDH..)

```

6 LET D$=""
10 LET A$=""
20 LET B$=""
30 LET C$="0 0"
35 LET N=0
40 LET LY=INT (25*RND)
50 LET LX=INT (17*RND)+1
60 LET WY=INT (25*RND)
70 LET WX=INT (17*RND)+1
80 IF ABS (LX-WX)<4 THEN GO TO 40
90 BORDER 6: PAPER 4: CLS
100 PRINT INK 2;AT WX,WY;B$; INK 0;AT WX
+1,WY;C$
110 PRINT INK 3;AT LX-1,LY;D$;AT LX,LY;A
$; INK 0;AT LX+1,LY+1;C$
115 IF LX=WX AND WY=LY-3 THEN GO TO 210
120 PRINT AT 21,1;"CHEMIN A SUIVRE (DGGH
...)"
130 INPUT R$
132 LET X=0
134 LET Y=X
136 LET N=N+1
140 FOR I=1 TO LEN R$
150 LET X=X+(R$(I)="B" AND LX<20)-(R$(I)
="H" AND LX>0): BEEP 0.3,18
160 LET Y=Y+(R$(I)="D" AND LY<28)-(R$(I)
="G" AND LY>0)
170 NEXT I
180 LET LX=LX+X
190 LET LY=LY+Y
200 GO TO 90
210 PRINT FLASH 1;AT 21,1;"BRAVO : TROUV
E EN "; BRIGHT 1;N; BRIGHT 0;" COUPS": BE
EP 1,12

```



CHEMIN A SUIVRE (DGGH...)

Tirage d'un dé truqué

REFERENCES :

Mode FAST.

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

Les probabilités désirées pour chaque tirage sont introduites par une boucle I (20-80) avec toutes les vérifications de validité nécessaires. Le nombre de tirages demandés est introduit en ligne 96.

La boucle J (105-160) comptabilise les tirages.

La boucle I (120-140) assure la conformité de chaque tirage avec les probabilités introduites en début de jeu.

Les résultats sont comptabilisés en ligne 145, et affichés grâce à la ligne 150.

POUR JOUER :

Commencer l'exécution en frappant RUN, puis, à la question PROBA 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 NBRE DE TIRAGES, on introduira un entier suffisamment grand, suivi de NEWLINE. L'affichage des résultats mettra d'autant plus de temps à apparaître que l'entier introduit sera grand.

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

EXTENSIONS POSSIBLES :

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

```

10 LET P=NOT PI
15 DIM E(6)
20 FOR I=SGN PI TO 6
30 PRINT AT INT PI,NOT PI;"PRO
BA DU DE ";I
40 INPUT E(I)
50 LET P=P+E(I)
60 IF E(I)<NOT PI OR E(I)>=SGN
PI OR P>1.1 THEN GOTO 200
70 PRINT AT NOT PI,5*I-5;"DE "
;I;AT SGN PI,5*I-5;E(I)
80 NEXT I
90 IF P<.9 THEN RUN
95 PRINT AT INT PI,NOT PI;"NBR
E DE TIRAGES"
96 INPUT P
100 RAND
105 FOR J=SGN PI TO P
110 LET N=INT (RND*6+SGN PI)
120 FOR I=SGN PI TO 6
130 IF N=I AND RND>E(I)-INT E(I)
) THEN GOTO 110
140 NEXT I
145 LET E(N)=E(N)+SGN PI
150 PRINT AT PI*PI,5*N-5;E(N);A
T 8,5*N-4;CHR$(156+N)
160 NEXT J
170 STOP
200 PRINT "IMPOSSIBLE :0<= P <=
1"
210 LET P=P-E(I)
215 PAUSE 50
220 CLS
230 GOTO 30

```

DE 1 DE 2 DE 3 DE 4 DE 5 DE 6
0.2 0.3 .05 .05 0.2 0.2

NBRE DE TIRAGES

1	2	3	4	5	6
19.2	27.3	4.05	5.05	18.2	27.2

```

10 LET P=0
15 DIM E(6)
20 FOR I=1 TO 6
30 PRINT AT 3,0;"PROBABILITE DU DE ";I
40 INPUT E(I)
50 LET P=P+E(I)
60 IF E(I)<0 OR E(I)>=1 OR P>1.1 THEN G
0 TO 200
70 PRINT INK 3;AT 0,5*I-5;"DE ";I;AT 1,
5*I-5;E(I)
80 NEXT I
90 IF P<.9 THEN RUN
95 PRINT INVERSE 1;AT 3,0;"NOMBRE DE TI
RAGES "; INVERSE 0;" "
96 INPUT P: PRINT INK 4;AT 3,19;P
100 RANDOMIZE
105 FOR J=1 TO P
110 LET N=INT (RND*6+1)
120 FOR I=1 TO 6
130 IF N=I AND RND>E(I)-INT E(I) THEN GO
TO 110
140 NEXT I
145 LET E(N)=E(N)+1
150 PRINT AT 10,5*N-5;INT E(N); INK 2; I
NVERSE 1;AT 8,5*N-4;N
160 NEXT J
170 STOP
200 PRINT FLASH 1;"IMPOSSIBLE :0<= P <=1
"
210 LET P=P-E(I)
215 PAUSE 50
220 CLS
230 GO TO 30

```

DE 1 DE 2 DE 3 DE 4 DE 5 DE 6
.05 .05 0.1 0.3 0.4 0.1

NOMBRE DE TIRAGES 200

1	2	3	4	5	6
12	12	16	53	81	26

Ordre

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : moyenne.

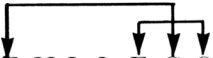
Catégorie de jeu : jeu de réflexion.

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 J I C F E
0 1 2 3 4 5 6 7 8 9 : permutation 83


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

LE PROGRAMME :

Les boucles emboîtées I (10-70) et J (30-50) définissent une variable alphanumérique A\$, contenant les 10 premières lettres de l'alphabet dans le désordre.

La ligne 90 affiche A\$, et la ligne 100 affiche les numéros de lettres à utiliser pour la permutation.

On entre la permutation en ligne 120.

Les lignes 132 et 135 transforment la variable alphanumérique entrée en une variable numérique.

Les lignes 150 à 170 effectuent la première inversion.

Les lignes 190 à 210 effectuent la deuxième.

La ligne 220 comptabilise les essais.

La ligne 230 contrôle les résultats.

POUR JOUER :

Lancer le programme par RUN.

Le mot initial est affiché, chaque lettre étant repérée par un numéro de 0 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.

ZX-81

```

5 LET A$=""
10 FOR I=SGN PI TO 10
20 LET R$=CHR$ (INT (10*RND+38
))
30 FOR J=SGN PI TO I-SGN PI
40 IF R$=A$(J) THEN GOTO 20
50 NEXT J
60 LET A$=A$+R$
70 NEXT I
80 LET S=NOT PI
90 PRINT AT PI,NOT PI;R$
100 PRINT "ABCD EFGHIJ"
110 PRINT AT PI+PI,NOT PI;"INVE
RSION (I/J) NO. ";S
120 INPUT R$
130 PRINT R$
132 LET I=CODE R$(SGN PI)-27
135 LET J=CODE R$(2)-27
150 LET N$=A$(J)
160 LET A$(J)=A$(I)
170 LET A$(I)=N$
180 IF R$(SGN PI)="0" OR R$(SGN
PI)="9" THEN GOTO 220
190 LET N$=A$(I-SGN PI)
200 LET A$(I-SGN PI)=A$(I+SGN P
I)
210 LET A$(I+SGN PI)=N$
220 LET S=S+SGN PI
230 IF A$="ABCDEFGHJIJ" THEN PRI
NT "TROUVE EN ";S;" COUPS:";A$(J)
240 GOTO 90

```

ABCD EFGHIJ
0123456789

INVERSION (I/J) NO. 7
98
TROUVE EN 8 COUPS: ABCDEFGHIJ

ZX-SPECTRUM

```

5 LET A$=""
10 FOR I=1 TO 10
20 LET R$=CHR$ (INT (10*RND+65))
30 FOR J=1 TO I-1
40 IF R$=A$(J) THEN GO TO 20
50 NEXT J
60 LET A$=A$+R$
70 NEXT I
80 LET S=0
90 PRINT INVERSE 1; INK 2;AT 3,0;A$
100 PRINT PAPER 4;"0123456789"
110 PRINT AT 6,0;"INVERSION (I/J) ; ESSA
I ";S
120 INPUT R$
130 PRINT "I = ";R$(1);" ; J = ";R$(2):
BEEP 1,10: BEEP 1,4
132 LET I=CODE R$(1)-47
135 LET J=CODE R$(2)-47
150 LET N$=A$(J)
160 LET A$(J)=A$(I)
170 LET A$(I)=N$
180 IF R$(1)="0" OR R$(1)="9" THEN GO TO
220
190 LET N$=A$(I-1)
200 LET A$(I-1)=A$(I+1)
210 LET A$(I+1)=N$
220 LET S=S+1
230 IF A$="ABCDEFGHJIJ" THEN PRINT FLASH
1; PAPER 4;"TROUVE EN ";S;" COUPS :"; FLA
SH 0; PAPER 5;A$: BEEP 1,10: STOP
240 GO TO 90

```

ABCD EFGHIJ
0123456789

INVERSION (I/J) ; ESSAI 10
I = 9 ; J = 6
TROUVE EN 11 COUPS : ABCDEFGHIJ

Course de chevaux

REFERENCES :

Mode SLOW.

Difficulté de programmation : moyenne.

Catégorie de jeu : hasard (plusieurs joueurs).

Difficulté du jeu : aucune.

LE JEU :

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

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

LE PROGRAMME :

Une première boucle I (60-150) affiche les chevaux dans leur position de départ.

Une deuxième boucle I (90-150) utilisant le même NEXT I les fera avancer ; la position de chaque cheval sera repérée par un chiffre ou une lettre dans une variable alphanumérique A\$.

La ligne 110 fera avancer les chevaux.

La ligne 120 gère les arrivées.

La ligne 140 réalisera les impressions.

Le sous-programme 200-230 fait avancer le compteur des arrivées, et imprime les résultats.

POUR JOUER :

On lance le programme par RUN.

A la question NBRE DE CHEVAUX : on répond par un entier de 1 à 10, suivi de NEWLINE.

La course se déroulera ensuite automatiquement.

EXTENSIONS POSSIBLES :

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

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

```

10 LET A$=""
20 PRINT "NBRE. DE CHEVAUX ";
30 INPUT N
35 PRINT N
40 LET P=NOT PI
50 LET T=SGN PI
55 LET A=P
60 FOR I=T TO N
70 LET A$=A$+"0"
80 GOTO 140
90 FOR I=SGN PI TO N
100 IF A$(I)="R" THEN GOTO 150
110 LET A$(I)=CHR$(CODE A$(I)+
INT (RND*4)+SGN PI)
120 IF A$(I)>="R" THEN GOSUB 20
0
140 PRINT AT I+I, CODE A$(I)-28;
"CHR$(156+I)
150 NEXT I
160 LET P=P+T
170 LET T=NOT PI
180 IF A<N THEN GOTO 90
190 STOP
200 LET A$(I)="R"
210 PRINT AT I+I, NOT PI; P
220 LET T=SGN PI
230 LET A=A+T
240 RETURN

```

NBRE. DE CHEVAUX : 6

3
2
3
3
3
1

1
2
3
4
5
6

```

10 LET A$=""
20 PRINT "NOMBRE DE CHEVAUX : ";
30 INPUT N
35 PRINT N
40 LET P=0
50 LET T=1
55 LET A=P
60 FOR I=T TO N
70 LET A$=A$+"0"
80 GO TO 140
90 FOR I=1 TO N
100 IF A$(I)="K" THEN GO TO 150
110 LET A$(I)=CHR$(CODE A$(I)+INT (RND*
4)+1)
120 IF A$(I)>="K" THEN GO SUB 200
140 PRINT AT I+I, CODE A$(I)-48; " "; I
NVERSE 1; INK I-7*INT (I/7); CHR$(96+I);
BEEP 0.2,15; PAUSE 10
150 NEXT I
160 LET P=P+T
170 LET T=0; PAUSE 20
180 IF A<N THEN GO TO 90
190 BEEP 2,0; STOP
200 LET A$(I)="K"
210 PRINT AT I+I,0; FLASH 1; PAPER 5;P
220 LET T=1
230 LET A=A+T
240 RETURN

```

NOMBRE DE CHEVAUX : 7

2
1
2

2

3
5
6
7
8
9

Le jeu de Marienbad

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : moyenne.

Catégorie de jeu : logique (2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

LE PROGRAMME :

La ligne 40 dimensionne une variable A (I) qui contiendra le nombre d'allumettes de chaque tas.

La boucle I (50-56) permet d'entrer ce nombre.

La boucle I (59-80) imprime les tas.

La ligne 105 assure le changement de joueur.

La ligne 130, ainsi que la ligne 160, contrôlent si le coup est autorisé.

La ligne 215 contrôle la fin de la partie.

POUR JOUER :

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

On entrera ensuite le nombre d'allumettes dans chaque tas (faire à chaque entrée N suivi de NEWLINE). On obtiendra alors l'affichage des différents tas, avec le numéro du joueur concerné (1 ou 2), ainsi que le message NO DU TAS ; on répond en tapant le numéro du tas dans lequel on désire faire un retrait suivi de NEWLINE. On obtient ensuite la question NBRE A OTER, 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 ZX de servir de partenaire, etc.

```

5 CLS
10 PRINT "NBRE DE TAS"
20 INPUT N
30 LET S=NOT PI
35 IF N>=8 THEN RUN
40 DIM A(N)
50 FOR I=1 TO N
52 PRINT AT SGN PI,SGN PI:"ALL
TAS "
53 INPUT A(I)
55 LET S=S+A(I)
56 NEXT I
57 LET H=SGN PI
58 CLS
59 FOR I=1 TO N
60 PRINT AT 4,4*I+2:A(I)
65 PRINT AT 6,SGN PI:"TAS"
70 PRINT AT 6,4*I+2:I
80 NEXT I
100 PRINT AT 9,4:"JOUEUR "H
105 IF H=SGN PI THEN LET H=NOT
PI
110 PRINT AT 11,SGN PI:"NO. DU
TAS "
120 INPUT T
130 IF T>N THEN GOTO 110
132 IF A(T)=NOT PI THEN GOTO 11
0
135 PRINT T
140 PRINT AT 13,SGN PI:"NBRE A
OTER"
150 INPUT O
160 IF O>A(T) OR O=NOT PI THEN
GOTO 140
170 LET A(T)=A(T)-O
180 LET S=S-O
210 LET H=H+SGN PI
215 IF NOT S THEN GOTO 300
220 GOTO 58
300 PRINT AT 15,SGN PI:"UR. "H
" GAGNE"

```

	1	3	5	7	9	11	8
TAS	1	2	3	4	5	6	7

JOUEUR 2
 NO. DU TAS 6
 NBRE A OTER

```

5 CLS
10 PRINT "NOMBRE DE TAS ";
20 INPUT N: PRINT N
32 LET S=0
35 IF N>=8 THEN RUN
40 DIM A(N)
50 FOR I=1 TO N
52 PRINT AT 1,1:"TAS ";I;"; NOMBRE D'AL
LUMETTES ?"
53 INPUT A(I)
55 LET S=S+A(I)
56 NEXT I
57 LET H=1
58 CLS
59 FOR I=1 TO N
60 PRINT PAPER 5; BRIGHT 1;AT 4,4*I+1;"
";A(I);" "
65 PRINT AT 6,1:"TAS"
70 PRINT FLASH 1; INK 1;AT 6,4*I+2;I
80 NEXT I
100 PRINT AT 9,4:"JOUEUR ";H
105 IF H=1 THEN LET H=0
110 PRINT AT 11,1:"PRISE DANS LE TAS :";
: BEEP 1,12
120 INPUT T: IF T>N THEN GO TO 110
130 IF A(T)=0 THEN GO TO 110
135 PRINT T
140 PRINT AT 13,1:"NOMBRE A OTER : "
150 INPUT O: PRINT AT 13,16;O: PAUSE 18
160 IF O>A(T) OR O=0 THEN GO TO 140
170 LET A(T)=A(T)-O
180 LET S=S-O
210 LET H=H+1
215 IF NOT S THEN GO TO 300
220 GO TO 58
300 PRINT PAPER 4; INVERSE 1;AT 15,1:"LE
JOUEUR ";H;" GAGNE": BEEP 2,5

```

	1	3	5	7	9
TAS	1	2	3	4	5

JOUEUR 1
 PRISE DANS LE TAS : 2
 NOMBRE A OTER : 2

Idem

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : moyenne.

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

Difficulté du jeu : selon le niveau choisi.

LE JEU :

Le ZX 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 ZX en début de jeu ; vous devez, à chaque coup, choisir une colonne (les dizaines, les centaines, etc.) et ajouter dans cette colonne un certain entier ; le but est d'arriver à obtenir à un moment l'affichage d'un nombre ayant tous ses chiffres identiques.

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

LE PROGRAMME :

Une variable numérique N est dimensionnée à 2 en ligne 12, pour contenir les résultats successifs des deux joueurs.

La ligne 20 permettra d'entrer le niveau de difficulté choisi.

Le nombre initial est confectionné par la ligne 25. La quantité qui interviendra dans les décrémentation est obtenue en ligne 30.

La ligne 40 teste si le résultat n'est pas trop petit.

La ligne 45 affiche le numéro du joueur.

Le nombre est affiché en ligne 50 et le jeu du joueur est entré en ligne 70 (nombre de deux chiffres).

La ligne 80 incrémente la colonne choisie, et la ligne 85 décrémenté le nombre de la quantité choisie au départ.

Le résultat sera affiché un court instant (lignes 90-100). La boucle I (120-135) permet de vérifier si tous les chiffres sont identiques.

L'impression et la fin de partie sont effectuées en ligne 140. Les lignes 150 et 160 s'occupent du changement de joueur.

POUR JOUER :

Lancer l'exécution du programme par RUN.

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

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

EXTENSIONS POSSIBLES :

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

ZX-81

```

10 LET H=SGN PI
12 DIM N(H+H)
15 PRINT "NIVEAU (2-7)"
20 INPUT N
22 LET U=VAL "10"
25 LET N(SGN PI)=INT ((U**N-U*
*(N-SGN PI))*RND+U**N-SGN PI))
27 LET N(H+H)=N(SGN PI)
30 LET A=INT (2*U**N-SGN PI)*
RND+U**N-SGN PI))
35 CLS
40 IF N(H) < U**N-SGN PI THEN
LET N(H)=U**N-ABS N(H)
45 PRINT AT INT PI,NOT PI;"VOU
EUR "H
50 PRINT AT PI+PI,NOT PI;"NOMB
RE "TAB 15-N;N(H)
55 PRINT TAB 8;"7654321"
60 PRINT AT PI*PI,NOT PI;"VOTR
E JEU (N,C) ?"
70 INPUT R$
80 LET N(H)=N(H)+VAL R$(SGN PI
)+U**N-SGN PI)
85 LET N(H)=N(H)-R
90 PRINT AT CODE " ("NOT PI;"R
ESULTAT "N(H)
100 PAUSE CODE " FAST "
110 LET R$=STR$ N(H)
120 FOR I=SGN PI TO LEN R$-SGN
PI
130 IF R$(I) <> R$(I+SGN PI) THEN
GOTO 150
135 NEXT I
140 PRINT "BRAVO, JOUEUR "H;U
150 LET H=H+SGN PI
160 IF H>=INT PI THEN LET H=SGN
PI
170 GOTO CODE "7"

```

JOUEUR 2

NOMBRE 768
7654321

VOTRE JEU (N,C) ?

RESULTAT 634

ZX-SPECTRUM

```

10 LET H=1
12 DIM N(2)
15 PRINT "NIVEAU (2-7)"
20 INPUT N
22 LET U=10: LET P=0
25 LET N(1)=INT ((U^N-U^(N-1))*RND+U^(N
-1))
27 LET N(2)=N(1)
30 LET A=INT (2*U^(N-1)*RND+U^(N-1))
35 CLS : LET P=P+(H=1)
40 IF N(H) < U^N-AB
S N(H)
45 PRINT INVERSE 1; INK H;AT 3,0;"JOUER
R "H;AT 3,12;"ESSAI "P
50 PRINT AT 6,0;"NOMBRE "TAB 15-N; INV
ERSE 1; INK H;N(H)
55 PRINT TAB 8; PAPER 5;"7654321"
60 PRINT AT 9,0;"VOTRE JEU (N/C) ?"
70 INPUT R$
80 LET N(H)=N(H)+VAL R$(1)*U^(VAL R$(2)
-1)
85 LET N(H)=N(H)-A
90 PRINT AT 15,0;"RESULTAT "; INVERSE 1
; INK H;N(H)
100 BEEP 1,10: PAUSE 250
110 LET R$=STR$ N(H)
120 FOR I=1 TO LEN R$-1
130 IF R$(I) <> R$(I+1) THEN GO TO 150
135 NEXT I
140 PRINT FLASH 1; PAPER 4;"BRAVO , JOUE
UR "H; BEEP .3,10: BEEP .3,18: BEEP .3,8
: PRINT " "P;" ESSAIS": STOP
150 LET H=H+1
160 IF H>=3 THEN LET H=1
170 GO TO 35

```

JOUEUR 1

ESSAI 2

NOMBRE 925
7654321

VOTRE JEU (N/C) ?

RESULTAT 756

Mastermind

REFERENCES :

Mode FAST ou SLOW.

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

LE PROGRAMME :

Le choix du nombre de chiffres du nombre à découvrir est effectué de la ligne 10 à la ligne 30.

Le nombre est fabriqué à la ligne 50 (il pourra y avoir des chiffres répétés).

L'affichage est préparé par la ligne 60.

La ligne 95 s'occupe de la fin de partie.

Une première boucle I (100-124) teste les chiffres qui sont à la bonne place.

Une deuxième boucle J (108-114) teste les chiffres qui sont présents et mal placés.

Les compteurs sont aux lignes 118 et 122.

La ligne 128 affiche le résultat, et la ligne 130 comptabilise les essais.

POUR JOUER :

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

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

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.

Si l'écran est plein (message d'erreur 5), il faudra faire CONT pour effacer l'écran et poursuivre.

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

On peut sortir du programme à tout moment en entrant une lettre, suivie de NEWLINE, à la place du nombre essayé.

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...).

ZX-81

```

5 CLS
10 PRINT "NBRE DE CHIFFRES "
20 INPUT N
25 IF N>6 THEN RUN
30 PRINT N
35 LET X$=""
40 LET E=SGN PI
50 LET X$=STR$ INT (RND*9**N+1
0*+(N-SGN PI))
60 PRINT AT INT PI,SGN PI;"NO.
NOMBRE      BON      PLACE"
70 INPUT E$
80 LET X=NOT PI
90 LET Y=X
95 IF E$=X$ THEN PRINT "TROUVE
EN ";E;" COUPS"
100 FOR I=SGN PI TO N
105 IF X$(I)=E$(I) THEN GOTO 12
2
108 FOR J=SGN PI TO N
112 IF J<>I AND X$(I)=E$(J) THE
N GOTO 118
114 NEXT J
116 GOTO 124
118 LET Y=Y+SGN PI
120 GOTO 124
122 LET X=X+SGN PI
124 NEXT I
128 PRINT AT 4+E,SGN PI;E;TAB 7
E$;TAB 18;X;TAB 25;Y
130 LET E=E+SGN PI
140 GOTO 60

```

NBRE DE CHIFFRES 4

NO.	NOMBRE	BON	PLACE
1	1234	1	0
2	9878	0	1
3	0090	0	1
4	0107	0	0
5	0170	0	4
6	1007	1	1
7	1097	0	0
8	1900	0	0

ZX-SPECTRUM

```

5 CLS
10 PRINT "NOMBRE DE CHIFFRES "
20 INPUT N
25 IF N>6 THEN RUN
30 PRINT N
40 LET E=1
50 LET X$=STR$ (INT (RND*9**N+10*(N-1)))
60 PRINT FLASH 1; PAPER 4;AT 3,1;"NO.
NOMBRE      BON      PLACE"
70 INPUT E$
80 LET X=0
90 LET Y=X
95 IF E$=X$ THEN PRINT FLASH 1; PAPER 2
; INK 1;AT 21,0;"TROUVE EN ";E;" COUPS":
BEEP 1,12; STOP
100 FOR I=1 TO N
105 IF X$(I)=E$(I) THEN GO TO 122
108 FOR J=1 TO N
112 IF J<>I AND X$(I)=E$(J) THEN GO TO 1
18
114 NEXT J
116 GO TO 124
118 LET Y=Y+1
120 GO TO 124
122 LET X=X+1
124 NEXT I
128 PRINT INK 1;AT 4+E,1;E;TAB 7;E$;TAB
18;X;TAB 25;Y
130 LET E=E+1
140 GO TO 60

```

NOMBRE DE CHIFFRES 4

NO.	NOMBRE	BON	PLACE
1	1234	1	1
2	1567	0	1
3	5890	1	0
4	5243	1	1
5	9292	3	0
6	9212	2	1
7	9291	2	1
8	9212	2	1
9	1111	0	0
10	9293	2	1
11	4292	3	0
12	5292	3	0

TROUVE 6292 EN 13 COUPS

A travers

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

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



LE PROGRAMME :

Les positions des chiffres et des lettres sont enregistrées dans une table N (4, 4) ; les lignes 5 à 80 définissent les valeurs de départ.

Les lignes 110 à 150 dessinent le terrain de jeu et la ligne 160 affiche le joueur qui a le trait (sous la forme simplifiée 1 pour les chiffres, et A pour les lettres).

La ligne 190 contrôle le coup joué, de même que les boucles I (210-226) et J (220-224).

Le jeu proprement dit est géré par les lignes 229 à 400, et le changement de joueur est assuré par la ligne 410.

POUR JOUER :

Le terrain de jeu sera affiché après la frappe de RUN et l'ordinateur attend le coup du joueur désigné par un 1 ou un A.

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 NEWLINE.

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 ZX.

ZX-81

```

5 LET K=SGN PI
7 LET F=K+K
10 LET S=F+F
30 DIM N(S,S)
40 FOR I=K TO S
50 LET N(I,K)=CODE "0"+I
50 LET N(S,I)=CODE "8"+I
70 NEXT I
80 LET N(S,K)=NOT PI
100 CLS
110 FOR J=SGN PI TO S
120 FOR I=SGN PI TO S
130 PRINT AT F*I,F*J;". " AND
I,J)=NOT PI)CHR$(N(I,J)) AND N(
J)<>NOT PI
140 NEXT I
150 NEXT J
160 PRINT AT S*S,SGN PI;"UR "
HR$(CODE "B"-K)
180 INPUT R$
190 IF (K<S AND R$(F)="0") OR
K>S AND R$(F)="8") THEN GOTO C
E "A"
210 FOR I=SGN PI TO S
220 FOR J=SGN PI TO S
222 IF N(I,J)=CODE R$(SGN PI)
HEN GOTO CODE "FAST"
224 NEXT J
226 NEXT I
227 GOTO CODE "A"
229 LET A=I
230 LET B=J
320 LET I=I+(R$(F)="S")-(R$(F)=
"N")
330 LET J=J+(R$(F)="E")-(R$(F)=
"0")
340 IF (NOT I AND K>S) OR (J>S
AND K<S) THEN GOTO CODE "COS" *F
350 IF N(I,J)<>NOT PI THEN GOTO
CODE "A"
360 LET N(I,J)=CODE R$(SGN PI)
400 LET N(A,B)=NOT PI
410 LET K=ABS (CODE "$"-F-K)
420 GOTO CODE "COS" *F

```

```

1 . . .
. 2 . C
3 . . .
. A B .

```

UR 1

ZX-SPECTRUM

```

5 LET K=1
7 LET F=2
10 LET S=4
30 DIM N(S,S)
40 FOR I=K TO S
50 LET N(I,K)=CODE "0"+I
60 LET N(S,I)=CODE "?" +I
70 NEXT I
80 LET N(S,K)=0
100 BORDER 1: CLS
110 FOR J=1 TO S
120 FOR I=1 TO S
130 PRINT AT F*I,F*J;". " AND N(I,J)=0; P
APER (2 AND N(I,J)<60)+(4 AND N(I,J)>60);
CHR$(N(I,J)) AND N(I,J)<>0
140 NEXT I
150 NEXT J
160 PRINT AT S*S,1;"JOUER AYANT LES ";"
CHIFFRES" AND K;"LETTRES" AND NOT K
180 INPUT R$
190 IF (K AND R$(F)="0") OR (NOT K AND R
$(F)="S") THEN GO TO 160
210 FOR I=1 TO S
220 FOR J=1 TO S
222 IF N(I,J)=CODE R$(1) THEN GO TO 229
224 NEXT J
226 NEXT I
227 GO TO 160
229 LET A=I
230 LET B=J
320 LET I=I+(R$(F)="S")-(R$(F)="N")
330 LET J=J+(R$(F)="E")-(R$(F)="0")
340 IF (NOT I AND NOT K) OR (J>S AND K)
THEN GO TO 400
350 IF N(I,J)<>0 THEN GO TO 160
360 LET N(I,J)=CODE R$(1)
400 LET N(A,B)=0
410 LET K=ABS (K-1): BEEP 1,20*K-10
420 GO TO 100

```

```

. 1 . .
. 2 . C
3 . . .
. A B .

```

JOUER AYANT LES CHIFFRES

Reines sur l'échiquier

REFERENCES :

Mode FAST.

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 retenu est très souple, puisque vous disposez même de la possibilité de choisir la taille de votre échiquier, et les résultats seront affichés à la fois sous forme graphique et sous forme numérique.

LE PROGRAMME :

La ligne 10 vous permet de choisir la taille de l'échiquier.

Une variable D sera dimensionnée en conséquence, en ligne 20.

Une boucle d'exploration principale I (30-80) contient une sous-boucle J (50-70) qui réalise le contrôle de non-alignement (ligne 60).

Si, au cours de l'exploration, on rencontre un alignement, on revient en arrière, en utilisant les lignes 170 à 210 et on recommence.

La boucle F (100-140) assure l'impression de l'échiquier, et la boucle T (115-125), 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 160 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 NEWLINE.

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 (sauf BREAK) pour le faire repartir. Pour terminer l'exploration sans avoir toutes les solutions, il faudra taper BREAK.

EXTENSIONS POSSIBLES :

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

```

10 INPUT S
20 DIM D(S)
30 FOR I=SGN PI TO S
40 LET D(I)=SGN PI
50 FOR J=SGN PI TO I-SGN PI
60 IF D(I)=D(J) OR ABS (D(I)-D
(J))=I-J THEN GOTO CODE "0"
70 NEXT J
80 NEXT I
90 CLS
100 FOR F=SGN PI TO S
110 PRINT AT NOT PI,F+F;F;AT F+
F,NOT PI;F
115 FOR T=SGN PI TO S
120 PRINT AT F+F,T+T;" "
125 NEXT T
130 PRINT AT F+F,D(F)+D(F);"■"
135 PRINT AT CODE "=";F;D(F)
140 NEXT F
150 PAUSE 4E4
170 LET I=I-SGN PI
180 LET D(I)=D(I)+SGN PI
190 IF D(I)<=S THEN GOTO CODE "
M"
200 LET I=I-SGN PI
210 IF I THEN GOTO CODE "0"

```

	1	2	3	4
1	.	.	■	.
2	■	.	.	.
3	.	.	.	■
4	.	■	.	.

3142

```

10 INPUT S
20 DIM D(S)
30 FOR I=1 TO S
40 LET D(I)=1
50 FOR J=1 TO I-1
60 IF D(I)=D(J) OR ABS (D(I)-D(J))=I-J
THEN GO TO 180
70 NEXT J
80 NEXT I
90 CLS
100 FOR F=1 TO S
110 PRINT AT 0,F+F;F;AT F+F,0;F
115 FOR T=1 TO S
120 PRINT INK 4;AT F+F,T+T;"*"
125 NEXT T
130 PRINT PAPER 1;AT F+F,D(F)+D(F);" ":
BEEP .3,15
135 PRINT INK 2;AT 21,F;D(F)
140 NEXT F
160 PAUSE 4E4
170 LET I=I-1
180 LET D(I)=D(I)+1
190 IF D(I)<=S THEN GO TO 50
200 LET I=I-1
210 IF I THEN GO TO 180

```

	1	2	3	4	5	6	7	8
1		*	*	*	*	*	*	*
2	*	*	*	*	*		*	*
3	*	*	*	*	*	*	*	
4	*	*		*	*	*	*	*
5	*	*	*	*	*	*		*
6	*	*	*		*	*	*	*
7	*		*	*	*	*	*	*
8	*	*	*	*		*	*	*

16837425

Le jeu de la vie

REFERENCES :

Mode FAST ou SLOW.

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 ZX (version entièrement automatique) soit par vous-même (version manuelle). Cette population va évoluer sous vos yeux, en respectant certaines règles, très élémentaires de naissance, de vie et de mort. L'évolution se poursuivra jusqu'à ce que l'on décide d'arrêter le programme.

LE PROGRAMME :

Deux tables (B et C) sont utilisées aux lignes 30 et 35, pour enregistrer l'état de la population. Les boucles F (210-250) et G (220-240) sont utilisées en début de jeu pour définir la population initiale (lignes 222 et 227). Le calcul de la nouvelle population est ensuite réalisé à l'aide d'une quadruple boucle : 60-195, 110-190, 120-160, 130-150. La ligne 135 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 (210-250) et G (220-240).

La ligne 270 permet de recommencer.

POUR JOUER :

Dans la version automatique, il suffit de lancer le programme par RUN et de regarder. Pour arrêter, il faudra taper BREAK. Dans la version manuelle, la configuration de départ sera entrée, après le lancement du programme par RUN, en tapant 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 des PLOT, etc.

```

10 LET A=INT PI+SGN PI
20 LET B=SGN PI
30 DIM B(A,A)
35 DIM C(A,A)
40 LET I=NOT PI
50 GOTO CODE "COS"
60 FOR F=B TO A
110 FOR G=B TO A
125 LET S=NOT PI
120 FOR I=-B TO B
130 FOR J=-B TO B
135 IF F+J=NOT PI OR G+I=NOT PI
OR F+J=A+B OR G+I=A+B THEN GOTO
CODE "■"
140 LET S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 LET S=S+B(F,G)
175 LET C(F,G)=NOT PI
180 IF S>=A+B AND S<=A+B THEN
LET C(F,G)=B
190 NEXT G
195 NEXT F
200 CLS
210 FOR F=B TO A
220 FOR G=B TO A
222 IF I THEN GOTO CODE "FAST"
227 LET C(F,G)=B AND RND<.5
229 PRINT AT F+F,G+G;"*" AND C(
F,G)=B
230 LET B(F,G)=C(F,G)
240 NEXT G
250 NEXT F
260 PAUSE A+A
270 GOTO CODE "U"

```

```

*      *
      *
*  *

```

```

10 PRINT "DIMENSION": INPUT A
20 LET B=1
30 DIM B(A,A)
35 DIM C(A,A)
40 LET I=0
50 GO TO 200
60 FOR F=B TO A
110 FOR G=B TO A
115 LET S=0
120 FOR I=-B TO B
130 FOR J=-B TO B
135 IF F+J=0 OR G+I=0 OR F+J=A+B OR G+I=
A+B THEN GO TO 150
140 LET S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 LET S=2*S-B(F,G)
175 LET C(F,G)=0
180 IF S>=5 AND S<=7 THEN LET C(F,G)=B
190 NEXT G
195 NEXT F
200 CLS
210 FOR F=B TO A
220 FOR G=B TO A
222 IF I THEN GO TO 229
227 LET C(F,G)=B AND RND<.5
229 PRINT INK 2;AT F+F,G+G;"■" AND C(F,G
)=B
230 LET B(F,G)=C(F,G)
240 NEXT G
250 NEXT F
260 BEEP 1,0
270 GO TO 60

```

```

■ ■ ■ ■
■ ■ ■ ■
      ■
      ■ ■ ■

```

```

10 LET A=INT PI+SGN PI
20 LET B=SGN PI
30 DIM B(A,A)
35 DIM C(A,A)
40 LET I=NOT PI
50 GOTO CODE "COS "
60 FOR F=B TO A
110 FOR G=B TO A
115 LET S=NOT PI
120 FOR I=-B TO B
130 FOR J=-B TO B
135 IF F+J=NOT PI OR G+I=NOT PI
OR F+J=A+B OR G+I=A+B THEN GOTO
CODE "■"
140 LET S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 LET S=S+B(F,G)
175 LET C(F,G)=NOT PI
180 IF S=A+B AND S=A+A-B THEN
LET C(F,G)=B
190 NEXT G
195 NEXT F
200 CLS
210 FOR F=B TO A
220 FOR G=B TO A
222 IF I THEN GOTO CODE "FAST"
"
225 PRINT AT PI*A,B;F;"G
226 PAUSE A**A
227 LET C(F,G)=VAL INKEY$
229 PRINT AT F+F,G+G;"+" AND C(
F,G)=B
230 LET B(F,G)=C(F,G)
240 NEXT G
250 NEXT F
260 PAUSE A**A
270 GOTO CODE "U"

```

```

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

```

4,4

```

10 PRINT "DIMENSION": INPUT A
20 LET B=1
30 DIM B(A,A)
35 DIM C(A,A)
40 LET I=0
50 GO TO 200
60 FOR F=B TO A
110 FOR G=B TO A
115 LET S=0
120 FOR I=-B TO B
130 FOR J=-B TO B
135 IF F+J=0 OR G+I=0 OR F+J=A+B OR G+I=
A+B THEN GO TO 150
140 LET S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 LET S=2*S-B(F,G)
175 LET C(F,G)=0
180 IF S>5 AND S<=7 THEN LET C(F,G)=B
190 NEXT G
195 NEXT F
200 CLS
210 FOR F=B TO A
220 FOR G=B TO A
222 IF I THEN GO TO 229
225 PRINT AT 3*A,B;F;"G
226 PAUSE A^A
227 LET C(F,G)=VAL INKEY$
229 PRINT INK 2;AT F+F,G+G;"■" AND C(F,G
)=B
230 LET B(F,G)=C(F,G)
240 NEXT G
250 NEXT F
260 BEEP 1,0
270 GO TO 60

```

```

■ ■ ■
  ■ ■ ■
■ ■ ■
  ■ ■ ■
■ ■ ■
  ■ ■

```

6,6

Niveau 3

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Losanges	Pour vous aider à vous endormir ...
Echiquier	Des bases graphiques, pour vos propres créations.
Couloirs	Un jeu de logique, à découvrir.
Jour de la semaine	Profitez-en pour faire le point sur les anniversaires.
Le pendu	Un jeu de lettres très classique (2 joueurs).
Bridge	Mettez au point vos propres problèmes de bridge.
Horloge 1	Ne jetez pas votre vieille télé : transformez-la en horloge !
Le loup et les agneaux	Un jeu de réflexion, pour 2 joueurs.
Belote	Vous n'aimez pas le bridge ? Alors, utilisez Belote !
Un taquin de calcul	Un bon entraînement au calcul (2 joueurs).
Poker	Une base de départ, pour mettre au point votre programme.
Hanoï	Un conseil : commencez petit ! ! !
Musique	Révisez vos notions de solfège, grâce à ZX.
Pendu inversé	Cette fois, c'est ZX qui essaie de deviner votre mot !
Mathieu	Vous ne savez pas dessiner ? Laisser faire ZX ...
Machinamot	Peut-être ZX trouvera-t-il des mots qui n'existent pas ?
Compose	Pour voir, pour apprendre, et pour entendre ...
Nicomaque	Entraînement au calcul mental (1 ou plusieurs joueurs).
Permutations	Un jeu de réflexion, pas toujours évident.
Tour de cartes	Et, pour finir, un peu de prestidigitation, par ZX.

Manipulation des chaînes et graphiques haute résolution

TRAITEMENT DES CHAINES DE CARACTERES : R\$ (/ TO)

Nous avons rencontré, au niveau 2, une première instruction permettant d'extraire le lème élément de la chaîne de caractères R\$; il s'agit de l'instruction R\$ (I).

L'instruction que nous allons manipuler à ce niveau donne toute sa puissance au Basic Sinclair, en permettant une extrême souplesse dans la manipulation des chaînes de caractères.

Soit, en effet, une chaîne R\$="MONSIEUR" ; on dispose de la possibilité d'extraire une sous-chaîne commençant au caractère I et finissant au caractère J, en écrivant simplement R\$ (I TO J).

Ainsi, R\$ (1 TO 3) = "MON"
R\$ (4 TO 6) = "SIE"

Si I est omis, on prend tous les caractères de la chaîne, jusqu'au Jème (instruction LEFT\$ (R\$,J) des Basic usuels).

Ainsi, R\$ (TO 3) = "MON".

Si J est omis, on prend tous les caractères de la chaîne depuis le lème (instruction RIGHT\$ (R\$,I) des Basic usuels).

Ainsi, R\$ (4 TO) = "SIEUR".

Le signe + permet la concaténation de deux chaînes :

R\$ (TO 3) + R\$ (4 TO) = "MONSIEUR"

De plus, il est possible de remplacer toute une sous-chaîne par une autre, avec une affectation de chaînes de longueur convenable.

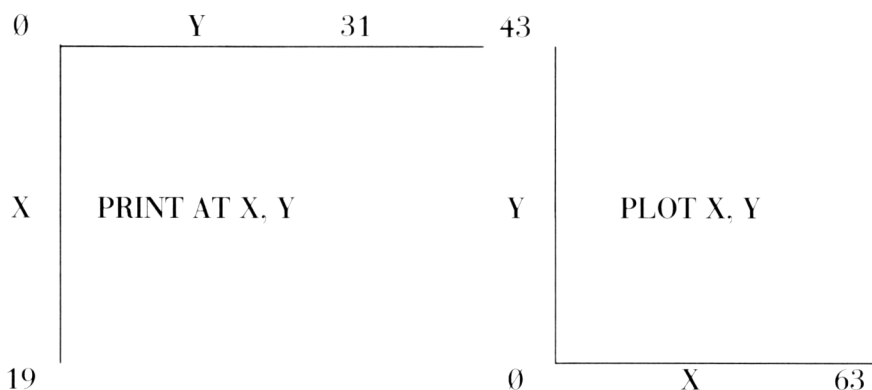
Ainsi, si R\$ = "MONSIEUR", et si l'on écrit LET R\$ (3 TO 5) = "PAT", on aura obtenu : PRINT R\$ = "MOPATEUR".

Les variables I et J peuvent bien sûr être calculées.

LES GRAPHIQUES “HAUTE RESOLUTION” : INSTRUCTION PLOT

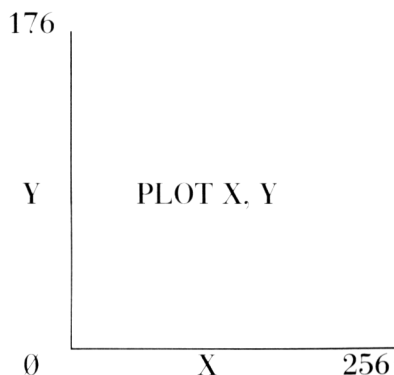
Le ZX-81 dispose de la possibilité (un peu limitée, il est vrai) d’allumer ou d’effacer des points sur l’écran, avec une résolution de 64×44 points.

Le repérage des axes n’est toutefois plus le même que lors de l’utilisation de PRINT AT.



Un point allumé par PLOT 18,23 sera, sur le ZX-81, effacé par UNPLOT 18,23.

Sur le ZX Spectrum, le principe reste le même, mais la résolution est de 256×176 .



Un point allumé par PLOT 150,79 sera, sur le ZX-Spectrum, effacé par PLOT OVER 1, 150,79.

Losanges

REFERENCES :

Mode SLOW.

Difficulté de programmation : facile.

Catégorie de jeu : interlude.

Difficulté du jeu : aucune.

LE JEU :

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

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

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

LE PROGRAMME :

Une première boucle K (10-60) définit les dimensions des tracés successifs.

La deuxième boucle I (20-50) trace des bandes de points, en utilisant les instructions PLOT (lignes 25 à 45).

Si la variable S (ligne 70) est à 1, alors il y aura effacement par utilisation des instructions, UNPLOT (lignes 90 à 105).

Pour obtenir un effacement aléatoire (deuxième programme), il suffira de modifier la ligne 22 qui contient normalement : 22 IF S THEN GOTO 90, en écrivant 22 IF S THEN GOTO 90 + 5 × INT (5 × RND).

On obtient, alors, un effacement aléatoire.

POUR JOUER :

On commence l'exécution du programme avec RUN. Il suffit dès lors d'attendre et de regarder.

On pourra arrêter à tout moment le déroulement, en tapant BREAK.

EXTENSIONS POSSIBLES :

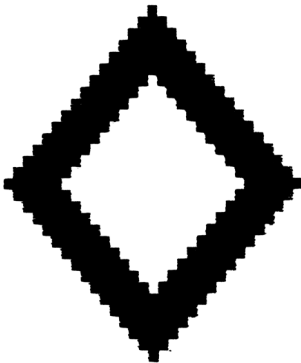
On pourra immédiatement, bien sûr, songer à introduire la couleur et, éventuellement, les sons.

On peut, à partir de là, imaginer toute une série de programmes fonctionnant sur le même principe (kaléidoscope, tapisseries, etc.).

```

5 LET S=NOT PI
10 FOR K=NOT PI TO 15
20 FOR I=NOT PI TO K
22 IF S THEN GOTO 90+5*INT (5*
RND)
25 PLOT 31-K+I,21-I
30 PLOT 31-K+I,21+I
35 PLOT 31+K-I,21+I
40 PLOT 31+K-I,21-I
50 NEXT I
60 NEXT K
70 LET S=NOT S
80 GOTO 10
90 UNPLOT 31-K+I,21-I
95 UNPLOT 31-K+I,21+I
100 UNPLOT 31+K-I,21+I
105 UNPLOT 31+K-I,21-I
110 GOTO 50

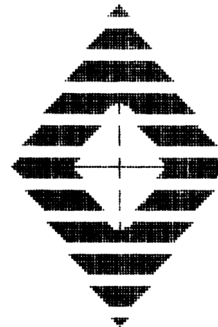
```



```

5 LET S=0: INK 2: PAPER 5: CLS
10 FOR K=0 TO 40
20 FOR I=0 TO K
22 IF S THEN GO TO 90
25 PLOT 124-K+I,84-I
30 PLOT 124-K+I,84+I
35 PLOT 124+K-I,84+I
40 PLOT 124+K-I,84-I
50 NEXT I
60 NEXT K
70 LET S=NOT S
80 GO TO 10
90 PLOT OVER 1;124-K+I,84-I
95 PLOT OVER 1;124-K+I,84+I
100 PLOT OVER 1;124+K-I,84+I
105 PLOT OVER 1;124+K-I,84-I
110 GO TO 50

```

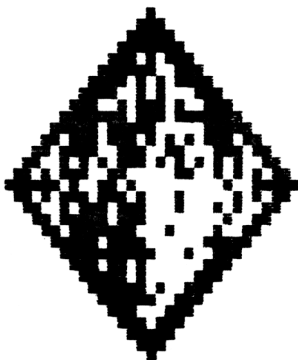


losanges aléatoires

```

5 LET S=NOT PI
10 FOR K=NOT PI TO 15
20 FOR I=NOT PI TO K
22 IF S THEN GOTO 90
25 PLOT 31-K+I,21-I
30 PLOT 31-K+I,21+I
35 PLOT 31+K-I,21+I
40 PLOT 31+K-I,21-I
50 NEXT I
60 NEXT K
70 LET S=NOT S
80 GOTO 10
90 UNPLOT 31-K+I,21-I
95 UNPLOT 31-K+I,21+I
100 UNPLOT 31+K-I,21+I
105 UNPLOT 31+K-I,21-I
110 GOTO 50

```



```

5 LET S=0: INK 2: PAPER 5: CLS
10 FOR K=0 TO 40
20 FOR I=0 TO K
22 IF S THEN GO TO 90+5*INT (5*RND)
25 PLOT 124-K+I,84-I
30 PLOT 124-K+I,84+I
35 PLOT 124+K-I,84+I
40 PLOT 124+K-I,84-I
50 NEXT I
60 NEXT K
70 LET S=NOT S
80 GO TO 10
90 PLOT OVER 1;124-K+I,84-I
95 PLOT OVER 1;124-K+I,84+I
100 PLOT OVER 1;124+K-I,84+I
105 PLOT OVER 1;124+K-I,84-I
110 GO TO 50

```



Echiquier

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : difficile.

Catégorie de jeu : graphisme.

Difficulté du jeu : variable, selon utilisation.

LE JEU :

Le programme proposé vient en appoint du programme Echecs pour être utilisé avec une éventuelle extension de mémoire sur 8 cases ; il donne, en effet, la possibilité de réaliser un affichage graphique de bonne qualité pour des jeux utilisant un damier de 8 cases sur 8 cases (on pourra facilement l'étendre à 10 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 :

Les lignes 10 et 20 définissent deux variables A\$ et C\$ qui contiennent les éléments de dessin.

La ligne 30 imprime les lettres horizontales et la première ligne du dessin.

La boucle I (40-70) permet l'impression, le motif à imprimer étant choisi dans A\$ ou C\$ par une variable S (ligne 50) qui prend les valeurs 1 ou 4, selon la valeur de I ; la ligne 60 réalise l'impression.

POUR JOUER :

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

Pour utiliser le programme, il conviendra de tenir compte du repérage des cases utilisables qui sont les centres des carrés noirs ou gris. Les cases sont obtenues en largeur par une tabulation de la forme $3 * I + 1$, I variant de 1 à 8, et en hauteur par $2 * I + 1$.

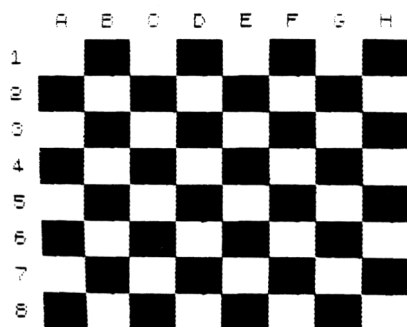
EXTENSIONS POSSIBLES :

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

```

5 LET S=VAL "2"
1000 LET A$=" "
2000 LET B$=" "
3000 LET C$=" "
40 FOR I=SGN PI TO VAL "8"
45 PRINT AT NOT PI,INT PI+I+SG
N PI,CHR$ (CODE "9"+I)
50 FOR J=SGN PI TO VAL "8"
60 IF (I+J)/S=INT ((I+J)/S) TH
EN GOTO 80
70 PRINT AT S*I-SGN PI,INT PI+
J;A$;TAB SGN PI,I;AT S*I,INT PI+
J;B$;AT S*I+SGN PI,INT PI+J;C$
80 NEXT J
90 NEXT I

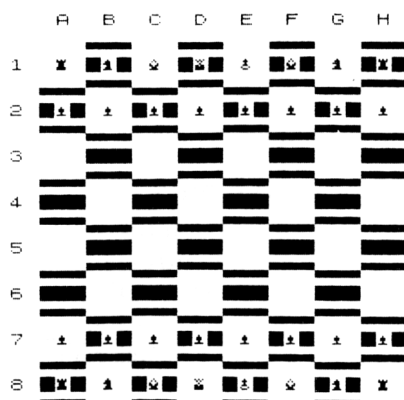
```



```

2 BORDER 2: CLS
5 LET S=2
10 LET A$=" "
20 LET B$=" "
30 LET C$=" "
40 FOR I=1 TO 8
45 PRINT AT 0,3*I+1;CHR$ (CODE "a"+I)
50 FOR J=1 TO 8
60 IF (I+J)/S=INT ((I+J)/S) THEN GO TO
80
70 PRINT AT S*I,1;I; INK 1;AT S*I-1,3*J
;A$;AT S*I,3*J;B$;AT S*I+1,3*J;C$
80 NEXT J
90 NEXT I
100 GO SUB 900
110 FOR I=1 TO 8
120 PRINT INK 2;AT 4,3*I+1;CHR$ 159
130 PRINT INK 4;AT 14,3*I+1;CHR$ 159
140 NEXT I
145 LET H=2: LET P=2
150 PRINT INK P;AT H,4;CHR$ 163;AT H,7;C
HR$ 154;AT H,10;CHR$ 149;AT H,13;CHR$ 160
;AT H,16;CHR$ 161;AT H,19;CHR$ 149
160 PRINT INK P;AT H,22;CHR$ 154;AT H,25
;CHR$ 163
170 LET H=H+14: LET P=P+2
180 IF H>20 THEN STOP
200 GO TO 150
900 REM DESSIN DES PIECES
910 FOR i=1 TO 6: READ p$
920 FOR j=0 TO 7: READ a: POKE USR p$+j,
a
930 NEXT j
940 NEXT i
1000 DATA "f",0,16,40,68,108,56,124,0
1100 DATA "r",0,16,56,16,56,68,56,0
1200 DATA "t",0,84,124,56,56,124,124,0
1300 DATA "q",0,84,40,16,108,124,124,0
1400 DATA "p",0,0,16,56,56,16,124,0
1500 DATA "k",0,16,56,120,24,56,124,0
1600 RETURN

```



Couloirs

REFERENCES :

Mode SLOW ou FAST.

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 :

La boucle I (10-70) dessine les couloirs (on ne voit pas les trois tourniquets) et les lignes 80 et 90 impriment les noms des entrées (1 et 2) et des sorties (A et B).

Les lignes 100 et 110 attendent votre choix.

Le décodage du chemin est fait par le ZX en utilisant trois variables logiques I, J et K, représentant les états des trois tourniquets (lignes 130 à 240).

POUR JOUER :

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

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 ZX.

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.).

```

3" 10 FOR I=INT (PI+PI) TO CODE "
20 PLOT NOT PI,I
30 PLOT CODE "?" ,I
35 IF I<CODE "1" OR I>CODE "1"
THEN GOTO CODE "0"
40 PLOT INT PI,I
50 PLOT CODE "E" ,I
60 IF I<CODE "E" OR I>CODE "E"
THEN PLOT PI+PI+SGN PI,I
70 NEXT I
75 LET I=NOT PI
76 LET J=I
77 LET K=I
80 PRINT AT 5,SGN PI;"A";TAB P
I+PI;"B"
90 PRINT AT CODE "E",SGN PI;"1"
";TAB PI+PI;"2"
95 LET R$=""
96 LET T$=""
100 PRINT AT NOT PI,NOT PI;"1 0
U 2 : "
110 INPUT C$
120 LET R$=R$+C$
130 PRINT C$;TAB CODE "$";R$
140 IF C$="1" THEN GOTO CODE "A
BS"
145 LET J=NOT J
150 IF J THEN GOTO CODE " FAST
"
200 LET S$="B"
205 GOTO CODE " THEN "
210 LET I=NOT I
215 IF I THEN GOTO CODE " FAST
"
220 LET S$="A"
222 LET T$=T$+S$
224 PRINT AT INT PI,NOT PI;"SOR
TIE :";S$";";T$
226 GOTO 100
229 LET K=NOT K
235 IF K THEN GOTO CODE "COS "
240 GOTO CODE ">="

```

1 00 2 12 1112222211212

SORT IE :E SAAAAABABABABAB

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

12 13

```

10 FOR I=12 TO 140
20 PLOT 5,I
30 PLOT 65,I
35 IF I<45 OR I>100 THEN GO TO 60
40 PLOT 20,I
50 PLOT 50,I
60 IF I<40 OR I>90 THEN PLOT 35,I
70 NEXT I
72 PRINT PAPER 5;AT 18,2;"T";AT 18,6;"T
";AT 12,4;"T"
75 LET I=0
76 LET J=I
77 LET K=I
80 PRINT INK 2; FLASH 1;AT 5,2;"A";AT 5
,6;"B"
90 PRINT INK 1; FLASH 1;AT 21,2;"1";AT
21,6;"2"
95 LET R$=""
96 LET T$=""
100 PRINT AT 0,0;"ENTREE 1,2 :";
110 INPUT C$: IF C$="" OR C$>="3" OR LEN
C$<>1 THEN GO TO 100
120 LET R$=R$+C$
130 PRINT C$;TAB 12;R$
140 IF C$="1" THEN GO TO 210
145 LET J=NOT J
150 IF J THEN GO TO 229
200 LET S$="B"
205 GO TO 222
210 LET I=NOT I
215 IF I THEN GO TO 229
220 LET S$="A"
222 LET T$=T$+S$
224 PRINT AT 3,0;"SORTIE :";S$;TAB 12;T$
: .BEEP 0.3,12
226 GO TO 100
229 LET K=NOT K
235 IF K THEN GO TO 200
240 GO TO 220

```

ENTREE 1,2 : 1

12211211122221

SORTIE : A BABABABABABABABA

A	B
	T
T	T

1 2

Jour de la semaine

REFERENCES :

Mode FAST ou SLOW.

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 (sous forme un peu abrégée : MAR pour mardi, DIM pour dimanche ..).

LE PROGRAMME :

Les lignes 10 à 75 contrôlent les entrées. La ligne 30 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 105 à 110 permettront l'affichage en clair du mois dans la réponse (sur 4 lettres).

La formule utilisée apparaît aux lignes 130 et 135.

Les lignes 140 et 150 permettront l'affichage en clair du jour dans la réponse (sur 3 lettres). La ligne 160 imprime le résultat.

Le sous-programme 200-220 intervient pour tenir compte des mois de janvier et février.

POUR JOUER :

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

Le résultat est très rapidement affiché et le programme revient sur la question JOUR.

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

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

EXTENSIONS POSSIBLES :

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

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

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

```

10 PRINT AT NOT PI,NOT PI;"JOU
R
20 INPUT J
30 IF NOT J THEN STOP
35 PRINT J
40 PRINT "MOIS(1-12):";
50 INPUT M
55 PRINT M
60 PRINT "ANNEE      ";
65 INPUT A
70 PRINT A
75 LET B=A
80 LET S=INT (A/100)-INT (A/40
0)
90 IF A<=VAL "1582" THEN LET S
=VAL "2"
105 LET M$="JANVF EVMARS AVRIMAI
JUINJUILAOUTSEPTOCTONOVEDECE"
110 LET M$=M$(M+VAL "4"-INT PI
TO M*VAL "4")
120 IF M<INT PI THEN GOSUB CODE
"005 "
130 LET S=J-SGN PI+INT (VAL "13
"*(M+SGN PI)/VAL "5")+INT (VAL "
5"*A/VAL "4")-S
135 LET S=S-VAL "7"*INT (S/VAL
"7")
140 LET J$="DIMLUNMARMERJEU VENS
AM"
150 LET J$=J$(INT PI*S+SGN PI T
O INT PI*S+INT PI)
160 PRINT AT VAL "7",NOT PI:"LE
";J;" ";M$;" ";B;" ETAIT UN ";
J$
170 GOTO VAL "10"
200 LET M=M+VAL "12"
210 LET A=A-SGN PI
220 RETURN

```

```

JOUR      :14
MOIS(1-12):7
ANNEE     :1984

```

LE 14 JUIL 1984 ETAIT UN :SAM

```

10 PRINT AT 0,0;"JOUR      ";
20 INPUT J
30 IF NOT J THEN STOP
35 PRINT INK 2; FLASH 1;J
40 PRINT "MOIS(1-12):";
50 INPUT M
55 PRINT INK 2; FLASH 1;M
60 PRINT "ANNEE      ";
65 INPUT A
70 PRINT INK 2; FLASH 1;A
75 LET B=A
80 LET S=INT (A/100)-INT (A/400)
90 IF A<=1582 THEN LET S=2
105 LET M$="JANVF EVMARS AVRIMAI JUINJUIL
AOUTSEPTOCTONOVEDECE"
110 LET M$=M$(M+4-3 TO M*4)
120 IF M<3 THEN GO SUB 200
130 LET S=J+INT (13*(M+1)/5)+INT (5*A/
4)-S
135 LET S=S*INT (S/7)
140 LET J$="DIMANCHELUNDI  MARDI  MERC
REDIJEUDI  VENDREDISAMEDI  "
150 LET J$=J$(8*S+1 TO 8*S+8)
160 PRINT INVERSE 1;AT 7,0;"LE ";J;" ";M
$;" ";B;" ETAIT UN "; FLASH 1;J$: BEEP 1,
12
170 GO TO 10
200 LET M=M+12
210 LET A=A-1
220 RETURN

```

```

JOUR      :25
MOIS(1-12):12
ANNEE     :1984

```

LE 25 DECE 1984 ETAIT UN MARDI

Le pendu

REFERENCES :

Mode SLOW ou FAST.

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 affiche alors des tirets à l'emplacement des lettres de ce mot et demande au deuxième joueur de proposer une lettre. Chaque réponse correcte fait apparaître la lettre à son emplacement exact dans le mot, et chaque réponse fausse fait progresser le dessin de la potence.

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

LE PROGRAMME :

Le mot à deviner est introduit en début de programme (lignes 10 et 20) ; une variable B\$, en ligne 25, contient tous les éléments du dessin.

La boucle I (40-60) fabrique un mot P\$ qui contient des tirets à l'emplacement de chacune des lettres du mot initial.

La boucle J (84 et 88) s'occupe du dessin (plus ou moins complet, selon la valeur de P).

Les lignes 90 et 92 affichent le résultat et arrêtent le jeu (grâce à une variable non définie W).

La lettre ess: yée est entrée en lignes 95 et 100 et testée dans une boucle I (110-130) ; le sous-programme 200-220 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 NEWLINE. Des tirets sont affichés à l'emplacement de chacune des lettres du mot ; et la question LETTRE apparaît sur l'écran ; l'autre joueur répond alors en proposant une lettre suivie de NEWLINE.

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 ZX pourra piocher au hasard ; on pourra aussi améliorer le dessin et modifier le nombre d'essais conduisant à la pendaison, en fonction de la difficulté du mot.

ZX-81

```

10 PRINT "VOTRE MOT"
20 INPUT R$
25 LET B$=" "
30 LET P$=""
40 FOR I=SGN PI TO LEN R$
50 LET P$=P$+"-"
60 NEXT I
65 LET P=11
70 CLS
80 PRINT AT NOT PI,NOT PI:P$
84 FOR I=10 TO P STEP -SGN PI
86 PRINT TAB 14;B$(SGN PI+INT
PI*(I-SGN PI) TO I*INT PI)
88 NEXT I
90 IF P$=R$ THEN PRINT "BRAVO"
92 IF P=SGN PI THEN PRINT "PER
DU"
95 PRINT AT PI,NOT PI;"LETTRE"
100 INPUT S$
105 LET T=NOT PI
110 FOR I=SGN PI TO LEN R$
120 IF R$(I)=S$ THEN GOSUB 200
130 NEXT I
170 IF T THEN GOTO 70
180 LET P=P-SGN PI
190 GOTO 70
200 LET T=SGN PI
210 LET P$(I)=S$
220 RETURN

```

-0-



PERDU : BOF

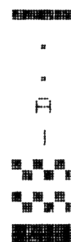
ZX-SPECTRUM

```

10 PRINT "VOTRE MOT"
20 INPUT R$
25 LET B$=" "
30 LET P$=""
40 FOR I=1 TO LEN R$
50 LET P$=P$+"-"
60 NEXT I
65 LET P=11
70 CLS
80 PRINT AT 0,0;P$
84 FOR I=10 TO P STEP -1
86 PRINT TAB 14;INK 2;B$(1+3*(I-1) TO
I*3)
98 NEXT I
90 IF P$=R$ THEN PRINT FLASH 1; PAPER 4
;"BRAVO": BEEP .2,10: BEEP .2,10: BEEP .2
,10: BEEP .2,10: STOP
92 IF P=1 THEN PRINT FLASH 1;"PERDU ;LE
MOT ETAIT ";R$: BEEP .2,-10: BEEP .2,-10
: BEEP .2,-10: BEEP .2,-10: STOP
95 PRINT AT 15,0; PAPER 5; INK 1; BRIGH
T 1;"VOTRE LETTRE"
100 INPUT S$: BEEP 1,10
105 LET T=0
110 FOR I=1 TO LEN R$
120 IF R$(I)=S$ THEN GO SUB 200
130 NEXT I
170 IF T THEN GO TO 70
180 LET P=P-1
190 GO TO 70
200 LET T=1
210 LET P$(I)=S$
220 RETURN

```

FABRICATION



BRAVO

Bridge

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : variable, selon le problème posé.

LE JEU :

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

LE PROGRAMME :

Les lignes 10 et 15 définissent un jeu de 52 cartes.

La variable B\$, en ligne 30, permettra de donner en abrégé le nom des couleurs et aussi d'améliorer la désignation de certaines cartes (10, valet, dame, roi et as).

La boucle J (40-140) réalise le tirage de trois jeux de 13 cartes (il suffit de mettre 4 au lieu de INT PI en ligne 40 pour tirer les 52 cartes).

La ligne 50 sert uniquement à replacer la tête d'écriture en haut de l'écran.

La boucle I (60-110) réalise tirage et impression. La carte est choisie parmi les 52 disponibles, en ligne 66.

La ligne 75 calcule la couleur de la carte tirée.

La ligne 80 vérifie que cette carte n'a pas déjà été tirée.

Les lignes 81 et 82 s'occupent de l'impression complémentaire dans le cas où l'on tire un 10, un valet, une dame, un roi ou un as.

L'impression est réalisée par la ligne 88.

La ligne 100 permet d'éliminer les cartes déjà tirées, et la ligne 120 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 (sauf BREAK) pour obtenir la deuxième série de 13 cartes. On procédera de même pour la troisième.

EXTENSIONS POSSIBLES :

Sur un matériel disposant, comme c'est le cas du ZX-Spectrum, d'un affichage plus performant, 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 ZX, par exemple), ou introduire des possibilités de jeu de la carte, y compris pour ZX.

```

10 LET R$="234567891VDRA"
15 LET R$=R$+R$+R$+R$
20 LET U=INT (PI*PI)
30 LET B$="TRCACOP10AADS"
35 LET V=LEN B$
40 FOR J=SGN PI TO INT PI
50 PRINT AT NOT PI,V
60 FOR I=SGN PI TO V
65 LET X=INT (LEN R$*RND+SGN P
I)
75 LET S=(SGN PI+SGN PI)*INT (
X/(U+SGN PI))+SGN PI
80 IF R$(X)="0" THEN GOTO CODE
"PI"
81 LET T=X-U*INT (X/U)
82 LET T=T+V*NOT T
83 PRINT TAB U*J-U;R$(X)+(B$(T
) AND T>=U);TAB U*J-PI-PI;B$(S T
O S+SGN PI)
100 LET R$(X)="0"
110 NEXT I
120 PAUSE PI**U
140 NEXT J

```

```

8 CA 2 TR VA CO
0A PI 3 TR RO CO
10 PI 7 TR 8 CO
10 TR 0A CA 4 CO
03 PI 4 PI 5 CO
6 PI 5 CO 00 CO
2 CO 7 PI 00 CO
4 TR 3 TR RO TR
0A TR VA CA 50 TR
AS TR 2 CO 10 CA
AS CO 3 CO 00 CO
VA PI 4 CO 00 CO
RO CA 10 CO 7 CO

```

```

10 LET R$="234567891VDRA"
15 LET R$=R$+R$+R$+R$
20 LET U=9: PAPER 7: CLS
30 LET B$="TRCACOP10AADS"
35 LET V=LEN B$
40 FOR J=1 TO 4
60 FOR I=1 TO V
65 LET F=I-13*INT ((I-1)/13)
66 LET X=INT (LEN R$*RND+1)
75 LET S=2*INT ((X-1)/V)+1
80 IF R$(X)="0" THEN GO TO 66
81 LET T=X-V*INT (X/V)
82 LET T=T+V*NOT T
88 PRINT AT F,U*J-U;R$(X)+(B$(T) AND T>
=U);AT F,U*J-6; PAPER ((S-1)/2)+3;B$(S TO
S+1)
100 LET R$(X)="0"
110 NEXT I
120 BEEP 1,12: PAUSE 4E4
140 NEXT J

```

```

AS CA 9 CA 3 CO AS TR
VA CO 10 PI 10 CO 5 TR
7 CO RO TR RO CA 6 TR
4 CA 8 CO AS CO VA PI
5 PI DA CA 3 CA 2 CO
2 TR 5 CA 4 TR 7 TR
9 CO 6 CO 3 PI 8 CA
2 PI VA CA 8 PI DA CO
6 CA 9 TR 2 CA 7 CA
7 PI 10 CA 6 PI 10 TR
9 PI RO PI RO CO 4 PI
4 CO DA PI 3 TR 5 CO
AS PI DA TR VA TR 8 TR

```

Horloge 1

REFERENCES :

Mode FAST.

Difficulté de programmation : difficile.

Catégorie de jeu : dessin.

Difficulté du jeu : aucune.

LE JEU :

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

LE PROGRAMME :

La ligne 20 vous permettra d'entrer l'heure sous forme d'un entier de 4 chiffres (par exemple : 1024 signifiera 10 heures 24 minutes).

La ligne 50 effectue le décodage de chaque chiffre.

Les lignes 60 à 130 réalisent le dessin, en décalant l'impression au fur et à mesure (à l'aide de la variable K, incrémentée en ligne 150).

La ligne 175 assure le passage de 23 heures 59 minutes à 00 heure, 00 minute.

La ligne 178 permet d'ajouter une heure toutes les 60 minutes.

La ligne 190 assure l'affichage (on peut ajuster la durée de la pause pour régler l'horloge).

POUR JOUER :

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

L'heure s'affiche alors assez rapidement et l'horloge est en route. Pour l'arrêter, il faudra appuyer sur la touche BREAK.

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

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, utiliser la possibilité du Spectrum, de saisir le temps pour avoir une plus grande précision, etc.

```

20 INPUT A$
25 LET F=INT PI+INT PI
26 LET S=SGN PI
27 LET G=F-S
30 CLS
40 LET K=S
50 LET A=VAL A$(K)
60 PRINT AT S,F*K-G;"■■■■" AND
A<>S AND A<>G-S
70 FOR J=S TO G
80 PRINT AT J,F*K-G;"■" AND ((
A=NOT PI) OR (A>INT PI AND A<>F+
S));TAB F*K-INT PI;"■" AND A<>G
AND A<>F
85 PRINT AT G+J-S,F*K-G;"■" AN
D (A=NOT PI OR A=S+S OR A=F OR A
=F+S+S);TAB F*K-INT PI;"■" AND A
<>S+S
90 NEXT J
100 PRINT AT G,F*K-G+S;"■■■■" AND
A<>S;TAB F+G;"■■■■"
130 PRINT AT F+INT PI,F*K-G;"■■■■
■" AND A<>S AND A<>G-S AND A<>F+
S
150 LET K=K+S
170 IF K<G THEN GOTO CODE "M"
172 LET A=S+VAL A$
175 IF A=VAL "2360" THEN LET A=
NOT PI
176 LET A$="000"+STR$ A
177 LET A$=A$(LEN A$-INT PI TO
)
178 IF A$(INT PI TO )="60" THEN
LET A$=STR$ (VAL A$( TO S+S)+S)
+"00"
190 PAUSE VAL "3000"
210 GOTO CODE "2"

```

12-35

```

10 INPUT A$
25 LET F=6
26 LET S=1
27 LET G=F-S
30 BORDER 2: PAPER 6: CLS
40 LET K=S
50 LET A=VAL A$(K)
60 PRINT INK 1;AT S,F*K-G;"■■■■" AND A<>
S AND A<>G-S
70 FOR J=S TO G
80 PRINT INK 1;AT J,F*K-G;"■" AND ((A=0
) OR (A>3 AND A<>F+S));AT J,F*K-3;"■" AND
A<>G AND A<>F
85 PRINT INK 1;AT G+J-S,F*K-G;"■" AND (
A=0 OR A=S+S OR A=F OR A=F+S+S);AT G+J-S,
F*K-3;"■" AND A<>S+S
90 NEXT J
100 PRINT INK 1;AT G,F*K-G+S;"■■■■" AND A<>
S
130 PRINT INK 1;AT F+3,F*K-G;"■■■■" AND A
<>S AND A<>G-S AND A<>F+S
150 LET K=K+S
170 IF K<G THEN GO TO 50
172 LET A=S+VAL A$
175 IF A=2360 THEN LET A=0
176 LET A$="000"+STR$ A
177 LET A$=A$(LEN A$-3 TO )
178 IF A$(3 TO )="60" THEN LET A$=STR$ (
VAL A$( TO S+S)+S)+"00"
179 LET H=S
180 FOR J=S TO 115
185 FOR I=S TO 16
190 IF H THEN PRINT INK 1;AT G,F+G;"■■■■";A
T G+S,F+G;"■"
192 IF NOT H THEN PRINT AT G,F+G;" ";AT
G+S,F+G;" "
195 NEXT I
200 LET H=NOT H
205 NEXT J
210 BEEP .7,10: GO TO 30

```

12-35

Le loup et les agneaux

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

C'est un jeu assez connu qui peut être pratiqué indifféremment sur un damier 8 sur 8, ou 10 sur 10. Il est présenté, ici, dans une version pour un damier de 8 cases ; l'un des joueurs choisit le loup qui se déplace suivant les diagonales, en restant sur les cases noires, d'une case à chaque coup, et qui peut avancer ou reculer. Les agneaux se déplacent également sur les cases noires, en diagonale, mais n'ont pas le droit de reculer. Les cases déjà occupées sont interdites (pas de prises), et il n'y a pas de saut.

Le loup gagne s'il réussit à arriver sur la ligne du bas avant que tous les agneaux n'occupent celle du haut ; les agneaux gagnent dans ce dernier cas, ou s'ils arrivent à bloquer le loup.

LE PROGRAMME :

Une variable A\$, en ligne 2 contient les positions de départ.

La boucle I (10-40) dessine le damier, et les lignes 45 et 47 placent le loup et les agneaux.

La ligne 50 gère le changement de joueur.

La ligne 60 affiche L si c'est le joueur qui manipule le loup qui a le trait, et A si c'est son adversaire.

La ligne 120 attend le coup du joueur.

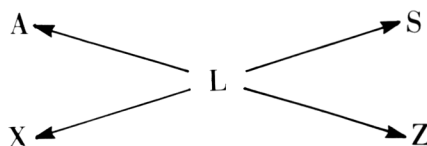
La ligne 130 choisira le programme analysant la réponse, en fonction du joueur concerné.

POUR JOUER :

RUN déclenche l'arrêt sur l'attente d'une entrée : 0 et NEWLINE si l'on veut que les agneaux commencent, et 1 pour le loup. Après quoi, le damier s'affiche et la phrase COUP A pour les agneaux et COUP L pour le jeu du loup.

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

Ainsi 2D provoque le déplacement de l'agneau numéro 2 qui monte en diagonale d'une case sur la droite. Le déplacement effectué est affiché, puis on obtient une question identique pour le loup qui répondra en tapant A, S, Z ou X, suivant la direction choisie :



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

EXTENSIONS POSSIBLES :

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

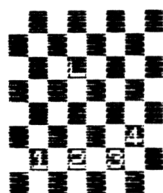
On pourra aussi envisager de jouer sur un damier de 10 cases sur 10 cases, ce qui augmente légèrement la difficulté du jeu pour les agneaux. Mais l'extension la plus intéressante à réaliser sera certainement la réalisation d'un programme permettant au ZX de se comporter en adversaire honorable, soit avec le loup, soit avec les agneaux (et, dans ce cas, il sera possible de définir une stratégie gagnante pour le ZX).

ZX-81

```

2 LET A$="8183858714"
5 INPUT J
7 CLS
10 FOR I=SGN PI TO VAL "4"
30 PRINT AT I+I,SGN PI;"
";AT I+I-SGN PI,SGN PI;"
40 NEXT I
45 PRINT AT VAL A$(9),VAL A$(10):"
0)";
47 PRINT AT VAL A$(SGN PI),VAL A$(2):"
";AT VAL A$(INT PI),VAL A$(4):"
";AT VAL A$(5),VAL A$(6):"
";AT VAL A$(7),VAL A$(8):"
50 LET J=ABS (J-SGN PI)
60 PRINT AT PI*PI,NOT PI;"COUP
";"L" AND NOT J;"A" AND J
120 INPUT R$
130 IF J THEN GOTO CODE "COS "
135 LET C=VAL "10"
140 LET A=VAL A$(C-SGN PI TO )+(C+SGN PI AND R$="X")+ (C-SGN PI AND R$="Z")-(C+SGN PI AND R$="A")-(C-SGN PI AND R$="S")
150 GOTO 207
200 LET C=VAL "2"*VAL A$(SGN PI)
205 LET A=VAL A$(C-SGN PI TO C)-(VAL "9" AND R$(2)="D")-(VAL "1" AND R$(2)="G")
207 LET A$(C-SGN PI TO C)=STR$ A
210 GOTO VAL "7"

```



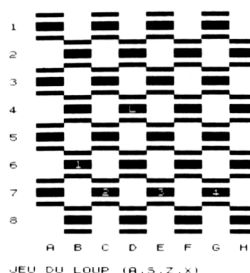
COUP A

ZX-SPECTRUM

```

2 LET A$="8183858714"
5 PRINT "AGNEAUX (0) OU LOUP (1) ?": I
NPUT J
7 PAPER 6: CLS
8 PRINT AT 0,3;"
";AT 4*1-2,3;"
10 FOR I=1 TO 4
30 PRINT AT 4*I-3,0;2*I-1;"
";AT 4*I-1,0;2*I;"
";AT 4*I,3;"
40 NEXT I
43 PRINT AT 16,3;"
";AT 17,3;" A B C D E F G H "
45 PRINT AT 2*VAL A$(9)-1,3*VAL A$(10)+4; INVERSE 1;"L"
47 PRINT INVERSE 1;AT 2*VAL A$(1)-1,3*VAL A$(2)+4;"1";AT 2*VAL A$(3)-1,3*VAL A$(4)+4;"2";AT 2*VAL A$(5)-1,3*VAL A$(6)+4;"3";AT 2*VAL A$(7)-1,3*VAL A$(8)+4;"4"
50 LET J=ABS (J-1): BEEP 0.5,-15
60 PRINT AT 21,0;"JEU "; "DU LOUP (A,S,Z,X)" AND NOT J;"DES AGNEAUX (ND OU NG)" A
ND J
120 INPUT R$
130 IF J THEN GO TO 200
135 LET C=10
140 LET A=VAL A$(C-1 TO )+(C+1 AND R$="X")+(C-1 AND R$="Z")-(C+1 AND R$="A")-(C-1 AND R$="S")
150 GO TO 207
200 LET C=2*VAL R$(1)
205 LET A=VAL A$(C-1 TO C)-(9 AND R$(2)="D")-(11 AND R$(2)="G")
207 LET A$(C-1 TO C)=STR$ A
210 GO TO 7

```



JEU DU LOUP (A,S,Z,X)

Belote

REFERENCES :

Mode FAST.

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 données de 20 cartes sur 32.

LE PROGRAMME :

En ligne 10, on crée un tableau pouvant contenir les 20 cartes.

La boucle I (20-70) assure le tirage aléatoire de 20 nombres, pris parmi 32 ; la boucle J (40-80) vérifie que l'on ne réalise pas deux fois le même tirage.

La deuxième boucle I (90-180) assure ensuite l'affichage, les noms des cartes étant affichés en clair, et la couleur étant décodée et affichée en clair, grâce aux lignes 117, 118, 120 et 123.

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 (notamment sur le Spectrum) ; on pourra également se servir de ce programme comme d'un noyau pour la mise au point de programmes de jeu de cartes sur l'ordinateur (qui pourra, si nécessaire, servir de partenaire : voir l'exemple du Poker, dans le même ouvrage).

```

10 DIM A(CODE "=")
15 LET K=SGN PI
20 FOR I=K TO CODE "="
30 LET A(I)=K+INT (CODE "4"*RN
D)
40 FOR J=K TO I-K
50 IF A(I)=A(J) THEN GOTO CODE
"2"
60 NEXT J
70 NEXT I
90 FOR I=SGN PI TO CODE "="
110 LET S$="CA"
113 LET S=SGN PI+A(I)-CODE "█" *
INT (A(I)/CODE "█")
114 LET R$=STR$ (S+6)+" "
115 LET R$=R$( TO CODE " ")
117 IF S=CODE "█" THEN LET R$="
VA"
118 IF S=CODE "█" THEN LET R$="
DA"
120 IF S=CODE "█" THEN LET R$="
RO"
123 IF S=CODE "█" THEN LET R$="
AS"
125 IF A(I)<CODE "/" THEN LET S
$="TR"
127 IF A(I)<CODE "(" THEN LET S
$="PI"
130 IF A(I)<CODE "█" THEN LET S
$="CO"
150 PRINT AT K, CODE "█" * INT ((I
-SGN PI)/CODE "█"); R$; S$
160 LET K=K+SGN PI
170 IF K=CODE "█" THEN LET K=SG
N PI
180 NEXT I

```

```

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

```

```

10 DIM A(20)
15 LET K=1
20 FOR I=K TO 20
30 LET A(I)=K+INT (32*RND)
40 FOR J=K TO I-K
50 IF A(I)=A(J) THEN GO TO 30
60 NEXT J
70 NEXT I
90 FOR I=1 TO 20
110 LET S$="CA"
113 LET S=1+A(I)-8*INT (A(I)/8)
114 LET R$=STR$ (S+6)+" "
115 LET R$=R$( TO 2)
117 IF S=5 THEN LET R$="VA"
118 IF S=6 THEN LET R$="DA"
120 IF S=7 THEN LET R$="RO"
123 IF S=8 THEN LET R$="AS"
125 IF A(I)<=24 THEN LET S$="TR"
127 IF A(I)<=16 THEN LET S$="PI"
130 IF A(I)<=8 THEN LET S$="CO"
150 PRINT AT (K+K,8*INT ((I-1)/5);R$;" ";
PAPER INT ((A(I)-1)/8)+3; BRIGHT 1;S$; B
EEP 0.1,15
160 LET K=K+1
170 IF K=6 THEN LET K=1

```

```

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

```

Un taquin de calcul

REFERENCES :

Mode SLOW ou FAST.

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 ZX) et doit le transformer à l'aide de 5 opérations qu'il choisit lui-même, ainsi d'ailleurs que les opérateurs.

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

LE PROGRAMME :

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

Une boucle 1 (50-100) permet d'entrer successivement les 5 opérations.

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

Après chaque calcul, le résultat est affiché (seul, le codeur devra regarder l'écran pendant cette opération) et le résultat définitif restera affiché pour le décodeur à la fin des calculs (ligne 95).

La ligne 140 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 NEWLINE.

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 ZX participer davantage en lui confiant le rôle du codage, etc.

```

3 CLS
5 LET X=INT (RND*16+5)
10 PRINT "NBRE INITIAL ";X
20 LET L=SGN PI
30 PRINT "ENTREZ 5 OPERATIONS"
35 LET F=X
40 PRINT "FORME +6 , /5 ..."
45 DIM R$(5,INT PI)
50 FOR I=L TO 5
60 INPUT R$(I)
70 LET A=CODE R$(I,L)-20
80 IF A<NOT PI OR A>4 THEN GO
TO 60
90 GOSUB A*1000
95 PRINT AT 4,L;"NBRE OBTENU "
;F;" "
100 NEXT I
130 PRINT "OPER."
140 PRINT R$(4);TAB 5;R$(1);TAB
10;R$(5);TAB 15;R$(3);TAB 20;R$(
2)
150 PRINT "BON COURAGE"
160 PAUSE 4E4
170 RUN
1000 LET F=F+VAL R$(I,2 TO )
1010 RETURN
2000 LET F=F-VAL R$(I,2 TO )
2010 RETURN
3000 LET F=F*VAL R$(I,2 TO )
3010 RETURN
4000 LET F=F/VAL R$(I,2 TO )
4010 RETURN

```

NBRE INITIAL 14
ENTREZ 5 OPERATIONS
FORME +6 , /5 ...

NBRE OBTENU 10
OPER.
+5 +6 -90 +9 /2
BON COURAGE

```

3 CLS
5 LET X=INT (RND*16+5)
10 PRINT INVERSE 1;"NOMBRE INITIAL "; I
NVERSE 0;" "; FLASH 1;X
20 LET L=1
30 PRINT "ENTREZ 5 OPERATIONS"
35 LET F=X
40 PRINT "FORME +6 , /5 , ..."
45 DIM R$(5,3)
50 FOR I=L TO 5
60 INPUT R$(I)
70 LET A=CODE R$(I,L)-41
80 IF A<0 OR A=3 OR A=5 OR A>6 THEN GO
TO 60
85 IF A>3 THEN LET A=(A/2)+1
90 GO SUB A*1000
95 PRINT INVERSE 1;AT 4,L;"NOMBRE OBTEN
U "; INVERSE 0;" "; FLASH 1; PAPER 5;F;
FLASH 0; PAPER 7;" "
100 NEXT I
130 PRINT : PRINT "OPERATIONS"
140 PRINT PAPER 4;R$(4);TAB 5;R$(1);TAB
10;R$(5);TAB 15;R$(3);TAB 20;R$(2)
150 PRINT : PRINT "BON COURAGE": BEEP 1,
20
160 PAUSE 4E4
170 RUN
1000 LET F=F+VAL R$(I,2 TO )
1010 RETURN
2000 LET F=F+VAL R$(I,2 TO )
2010 RETURN
3000 LET F=F-VAL R$(I,2 TO )
3010 RETURN
4000 LET F=F/VAL R$(I,2 TO )
4010 RETURN

```

NOMBRE INITIAL 6
ENTREZ 5 OPERATIONS
FORME +6 , /5 , ...

NOMBRE OBTENU 35

OPERATIONS
/3 +4 *7 -45 *6
BON COURAGE

NOMBRE INITIAL 12
ENTREZ 5 OPERATIONS
FORME +6 , /5 , ...

NOMBRE OBTENU 10
OPERATIONS
-5 +8 /2 /4
BON COURAGE

Poker

REFERENCES :

Mode FAST.

Difficulté de programmation : difficile.

Catégorie de jeu : jeu contre le ZX.

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 ZX.

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 ZX de jouer.

LE PROGRAMME :

Les lignes 10 et 20 définissent les cartes (avec 1 au lieu de 10 et A au lieu de A\$). Une variable N est dimensionnée à 15, en ligne 30, pour recevoir 10 cartes joueur et 5 cartes ZX.

Le tirage sera effectué par une boucle I (40-90) et le contrôle par une boucle J (60-80).

L'affichage des cartes est réalisé par le sous-programme 229-1060.

Une autre boucle I (150-170) réalise le deuxième tirage du joueur, la ligne 180 permettant le deuxième affichage, et la boucle I (229-1050) 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 REFUS, le joueur peut répondre 0 suivi de NEWLINE s'il veut arrêter la partie. Pour jouer, il répondra simplement NEWLINE s'il s'estime servi, ou il entrera les numéros des cartes qu'il refuse suivis de NEWLINE.

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

Il suffira ensuite de faire NEWLINE 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 ZX, l'introduction des règles sur les paris et même, pourquoi pas, des possibilités de bluff.

```

1 CLS
5 LET S=VAL "6"
10 LET V$="7891UDRA"
20 LET C$="TRCACOP1"
30 DIM N(CODE "?")
40 FOR I=SGN PI TO CODE "?"
50 LET N(I)=INT (RND*CODE "4")
60 FOR J=SGN PI TO I-SGN PI
70 IF N(I)=N(J) THEN GOTO CODE
"4"
80 NEXT J
90 NEXT I
100 LET K=NOT PI
105 PRINT AT S*PI,NOT PI;"REFUS
"
110 GOSUB CODE " FAST "
140 IF R$="0" THEN STOP
150 FOR I=SGN PI TO LEN R$
160 LET N(VAL R$(I))=N(I+S-SGN
PI)
170 NEXT I
175 CLS
180 GOSUB CODE " FAST "
200 LET K=VAL "10"
229 FOR I=SGN PI TO S-SGN PI
1010 LET J=INT (N(I+K)/8)+SGN PI
1020 LET R$=V$(N(I+K)-8*INT (N(I
+K)/8)+SGN PI)
1030 PRINT AT K,S*I-S;CHR$(CODE
"0"+I);" "+R$ AND R$<"1";"10"
AND R$="1";C$(J+J-SGN PI TO J+J)
1050 NEXT I
1055 INPUT R$
1058 IF K THEN RUN
1060 RETURN

```

1 7TR 2 RTR 3 VTR 4 RCO 5 7CA

11000 2 800 3 7PI 4 700 5 API

```

1 CLS
5 LET S=6
10 LET V$="7 8 9 10VADAR0AS"
20 LET C$="TRCACOP1"
30 DIM N(15)
40 FOR I=1 TO 15
50 LET N(I)=INT (RND*32)
60 FOR J=1 TO I-1
70 IF N(I)=N(J) THEN GO TO 50
80 NEXT J
90 NEXT I
100 LET K=0
105 PRINT AT 21,0;"CARTES REFUSEES ?"
110 GO SUB 229
140 IF R$="0" THEN STOP
150 FOR I=1 TO LEN R$
160 LET N(VAL R$(I))=N(I+S-1)
170 NEXT I
175 CLS
180 GO SUB 229
200 LET K=10
229 FOR I=1 TO 5
1010 LET J=2*INT (N(I+K)/8)+1
1015 LET F=2*(N(I+K)-8*INT (N(I+K)/8))+1
1020 LET R$=V$(F TO F+1)
1030 PRINT AT K,S*I-4; INVERSE 1;I; INVER
SE 0;AT K+2,S*I-S;R$;";"; PAPER (J+1)/2+1
; BRIGHT 1;C$(J TO J+1); BEEP .1,0; PRINT
AT K+3,S*I-S;" "
1050 NEXT I
1055 INPUT R$
1058 IF K THEN RUN
1060 RETURN

```

1 2 3 4 5

RO:TR AS:TR AS:CO DA:TR AS:CA

1 2 3 4 5

10:TR 8 :CO 9 :TR 10:CO RO:PI

Hanoi

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

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

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

LE PROGRAMME :

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

Le nombre de disques est entré en ligne 60 et refusé s'il y a plus de 8 disques.

La boucle I (70-90) permet de fabriquer une variable A(I) qui contient l'état de la première pile sous forme numérique (en ligne 5, 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 (120-160) et J (125-150).

Le contrôle du coup est effectué par la ligne 230 et l'état des piles est modifié par les lignes 240 et 250.

POUR JOUER :

Après un RUN qui lance le programme, on entre le nombre de disques demandés, suivi de NEWLINE. On obtient le dessin de l'état initial et, à la question DEP., 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 NEWLINE et, à la question ARR., on répond par le numéro de la tour d'arrivée. Si le coup n'est pas possible, on sera obligé de recommencer à DEP. 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.

```

5 DIM A(INT PI)
7 LET U=INT (PI*PI)
10 LET A(SGN PI)=U
20 LET A(SGN PI+SGN PI)=1)
30 LET A(INT PI)=U
40 LET D$=" "
50 PRINT "NBRE"
60 INPUT N
65 IF N>=U-SGN PI THEN RUN
70 FOR I=NOT PI TO N-SGN PI
80 LET A(SGN PI)=VAL (STR$ A(S
GN PI)+STR$ (N-I))
90 NEXT I
100 CLS
105 LET C$=" "
110 PRINT AT PI*PI,NOT PI)C$:"1
":C$;"2":C$;"3"
120 FOR I=NOT PI TO N-SGN PI
125 FOR J=SGN PI TO INT PI
127 LET B$=STR$ A(J)
130 IF LEN B$>I+SGN PI THEN PRI
NT AT U-I,(U+SGN PI)*J-U;D$( TO
VAL B$(I+SGN PI+SGN PI))
150 NEXT J
160 NEXT I
170 PRINT AT CODE "$",NOT PI;"D
EP"
180 INPUT D
185 IF A(D)=U THEN GOTO CODE "$
"
190 PRINT "ARR"
200 INPUT A
210 LET B$=STR$ A(D)
220 LET C$=STR$ A(A)
230 IF (D=A) OR (B$(LEN B$)>C$(
LEN C$)) THEN GOTO CODE "$"
240 LET A(A)=VAL (C$+B$(LEN B$)
)
250 LET A(D)=VAL B$( TO LEN (B$
)-SGN PI)
260 GOTO 100

```



DEP
ARR

```

5 DIM A(3)
7 LET U=9
10 LET A(1)=U
20 LET A(2)=U
30 LET A(3)=U
40 LET D$=" "
50 PRINT "NOMBRE DE TOURS (<8) :";
60 INPUT N: PRINT N
65 IF N>=8 THEN RUN
70 FOR I=0 TO N-1
80 LET A(1)=VAL (STR$ A(1)+STR$ (N-I))
90 NEXT I
100 CLS
105 LET C$=" "
110 PRINT AT 10,0; INK 1;C$;TAB 10;C$;TA
B 20;C$
115 PRINT AT 12,4; INK 1;"1";TAB 14;"2";
TAB 24;"3"
120 FOR I=0 TO N-1
125 FOR J=1 TO 3
127 LET B$=STR$ A(J)
130 IF LEN B$>I+1 THEN PRINT AT U-I,(U+1
)*J-U; INK 2;D$( TO VAL B$(I+2))
150 NEXT J
160 NEXT I
170 PRINT AT 14,0;"TOUR DE DEPART : ";
180 INPUT D: PRINT AT 14,17;D
185 IF A(D)=U THEN BEEP 1,0: GO TO 170
190 PRINT AT 15,0;"TOUR D ARRIVEE : ";

200 INPUT A: PRINT AT 15,17;A: IF A=D TH
EN GO TO 190
210 LET B$=STR$ A(D)
220 LET C$=STR$ A(A)
230 IF B$(LEN B$)>C$(LEN C$) THEN BEEP 1
,0: GO TO 170
240 LET A(A)=VAL (C$+B$(LEN B$))
250 LET A(D)=VAL B$( TO LEN (B$)-1)
260 BEEP .2,10: GO TO 100

```



1 2 3

TOUR DE DEPART : 3
TOUR D ARRIVEE : 1

Musique

REFERENCES :

Mode SLOW.

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu éducatif.

Difficulté du jeu : moyenne.

LE JEU :

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

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

LE PROGRAMME :

La boucle I (10-30) dessine la portée. Une deuxième boucle I (40-60) affiche les numéros des lignes.

La ligne 70 définit le nom des notes, et la ligne 80 réalise le tirage aléatoire d'une note.

Le nom de la note est calculé à la ligne 85 et il est affiché par la ligne 90, suivi de la mention GRAVE ou AIGUE, selon le cas.

La réponse sera entrée en ligne 110.

La ligne 115 s'occupe du décodage de la réponse.

La ligne 120 affiche la note.

La ligne 130 regarde si l'on a gagné ou perdu et affiche le message correspondant.

Les lignes 140 et 150 permettent 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 LIGNÉ.

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

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

On peut taper une touche quelconque pour continuer, et s'arrêter en tapant BREAK.

EXTENSIONS POSSIBLES :

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

```

5 CLS
10 FOR I=SGN PI TO 5
20 PRINT AT 4+I+I,INT PI;"
30 NEXT I
40 FOR I=SGN PI TO 14
50 PRINT AT I+2,NOT PI;CHR$ (2
8+I)
60 NEXT I
70 LET N$="SI LA SOLFA MI RE D
0 "
80 LET X=INT (RND*14+SGN PI)
85 LET T=X-SGN PI-7*INT ((X-SG
N PI)/7)
90 PRINT AT CODE ">"NOT PI;"N
OTE ";N$(INT PI+T+SGN PI TO INT
PI+T+INT PI);"AIGUE" AND X<=7;"G
RAVE" AND X>7
100 PRINT "LIGNE"
110 INPUT L$
115 LET L=CODE L$-28
120 PRINT AT X,PI*PI;" "AT X+5
GN PI*PI;" "AT X+2,PI*PI;" "
130 PRINT AT CODE "="NOT PI;"B
RAVO" AND L=X;"PERDU" AND L<>X
140 PAUSE 4E4
150 RUN

```

1
2
3
4
5
6
7
8
9
A
B
C
D
E



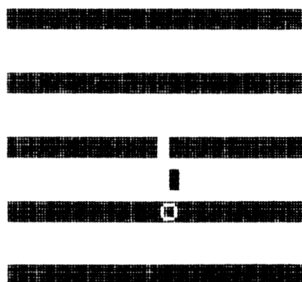
NOTE MI GRAVE
LIGNE
BRAVO

```

5 CLS
10 FOR I=1 TO 5
20 PRINT INK 1;AT 4+I+I,3;"
30 NEXT I
40 FOR I=1 TO 14
50 PRINT AT I+2,0;I AND I<=9;AT I+2,0;C
HR$ (55+I) AND I>9
60 NEXT I
70 LET N$="SI LA SOLFA MI RE DO "
80 LET X=INT (RND*14+1)
85 LET T=X-1-7*INT ((X-1)/7)
90 PRINT AT 18,0;"NOTE ";N$(3*T+1 TO 3*
T+3);" AIGUE" AND X<=7;" GRAVE" AND X>7
100 PRINT "LIGNE :";
110 INPUT L$: PRINT L$
115 LET L=(CODE L$-48)*(L$<"")+ (CODE L$
-55)*(CODE L$>64)
120 PRINT INVERSE 1; INK 2;;AT X,10;" ";
AT X+1,10;" ";AT X+2,10;"O": BEEP 1,20-X
130 PRINT AT 20,0;"BRAVO" AND L=X;"PERDU
" AND L<>X
140 PAUSE 4E4
150 RUN

```

1
2
3
4
5
6
7
8
9
A
B
C
D
E



NOTE SOL GRAVE
LIGNE : A
BRAVO

Pendu inversé

REFERENCES :

Mode SLOW.

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 ZX qui va chercher à deviner votre mot, mais il joue quelque peu au hasard et ne possède pas votre intuition ; il lui faudra donc, en règle générale, beaucoup d'essais pour découvrir la solution. Pour augmenter ses chances, on a fait appel à une formule qui lui permet de tirer davantage dans les voyelles puis, dans les consonnes courantes, et, enfin, dans les consonnes plus rares.

LE PROGRAMME :

La ligne 10 permet d'enregistrer, dans une variable alphanumérique B\$, les lettres de l'alphabet, dans l'ordre : voyelles, consonnes usuelles, autres consonnes. Le mot à deviner est introduit en ligne 50 (en fait, il suffirait d'introduire sa longueur).

La boucle I (60-90) imprime des tirets aux emplacements des lettres à trouver.

La pause en ligne 105 sert uniquement à simuler le temps de recherche d'une lettre.

La formule, placée en ligne 110, permet de modifier légèrement la probabilité de tirage, en faveur des voyelles et des consonnes usuelles.

La ligne 130 provoque un nouveau tirage, si la lettre avait déjà été essayée.

La ligne 160 affiche la lettre choisie, et la boucle I (170-200) réimprime les tirets et les lettres trouvées.

La ligne 205 incrémente le compteur d'essais.

La ligne 210 affiche le nombre d'essais réalisés, en fin de partie.

POUR JOUER :

Lancer le programme par RUN.

Entrer le mot à découvrir, suivi de NEWLINE.

Des tirets sont alors affichés (un par lettre) et le ZX 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 BREAK.

EXTENSIONS POSSIBLES :

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

Avec une minidisquette, on pourra faire retenir, au ZX, les mots les plus couramment utilisés par les joueurs, etc.

ZX-81

```
10 LET D$="YAEIOUBCDFGJLMNPRST
VWHKQXZ"
20 LET C=NOT PI
30 LET T$=""
40 PRINT "VOTRE MOT"
50 INPUT R$
60 FOR I=SGN PI TO LEN R$
70 PRINT AT 5,I;"-"
80 LET T$=T$+"-"
90 NEXT I
100 PRINT AT PI*PI,NOT PI;"JE C
HERCHE... ESSAI ";C
105 PAUSE VAL "140"
110 LET X=6*(RND<.7)+13*(RND<.5
)+7*(RND<.4)
120 LET X=INT (X*RND+SGN PI)
130 IF D$(X)="-" THEN GOTO 110
140 LET E$=D$(X)
150 LET D$(X)="-"
160 PRINT AT CODE "$",NOT PI;"J
E PRENDS ";E$
170 FOR I=SGN PI TO LEN R$
180 IF E$=R$(I) THEN LET T$(I)=
E$
190 PRINT AT 5,I;T$(I)
200 NEXT I
205 LET C=C+SGN PI
210 IF T$=R$ THEN PRINT AT CODE
"=",NOT PI;"TROUVE EN ";C;" COU
PS";C)
230 GOTO 100
```

VOTRE MOT

BONJOUR

JE CHERCHE... ESSAI 21

JE PRENDS B

TROUVE EN 22 COUPS

ZX-SPECTRUM

```
10 LET D$="YAEIOUBCDFGJLMNPRSTVWHKQXZ"
20 LET C=0
30 LET T$=""
40 PRINT "VOTRE MOT"
50 INPUT R$
60 FOR I=1 TO LEN R$
70 PRINT AT 5,I;"-"
80 LET T$=T$+"-"
90 NEXT I
100 PRINT INVERSE 1; INK 1;AT 10,0;"JE C
HERCHE...";" ESSAI ";C
105 PAUSE 150
110 LET X=6*(RND<.7)+13*(RND<.5)+7*(RND<
.4)
120 LET X=INT (X*RND+1)
130 IF D$(X)="-" THEN GO TO 110
140 LET E$=D$(X)
150 LET D$(X)="-"
160 PRINT AT 15,0; INVERSE 1; INK 2;"JE
PRENDS "; FLASH 1;E$: BEEP 1,10
170 FOR I=1 TO LEN R$
180 IF E$=R$(I) THEN LET T$(I)=E$
190 PRINT AT 5,I;T$(I)
200 NEXT I
205 LET C=C+1
210 IF T$=R$ THEN PRINT INVERSE 1; BRIGH
T 1; INK 4;AT 20,0;"TROUVE EN ";C-1;" COU
PS": BEEP 1,12; BEEP 1,18; BEEP 1,8; STOP
230 GO TO 100
```

VOTRE MOT

HYSTERIQUE

JE CHERCHE... ESSAI 23

JE PRENDS :H

TROUVE EN 23 COUPS

Mathieu

REFERENCES :

Mode SLOW ou FAST.

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 PLOT pour obtenir sur l'écran des dessins aléatoires (de type fleur), plus ou moins élégants (le PLOT du ZX-Spectrum est, bien entendu, beaucoup plus précis que celui du ZX-81).

Les deux paramètres principaux, A et B, pourront être, si nécessaire, entrés à la main (instruction INPUT) pour faire une étude plus systématique des différentes courbes possibles.

LE PROGRAMME :

La ligne 10 permet de définir le rayon maximum et la ligne 20 corrige ce paramètre pour utiliser, au mieux, la surface de l'écran.

On pourra, évidemment, modifier ces deux paramètres en veillant à rester dans les limites de l'écran.

Les lignes 30 et 40 définissent deux variables aléatoires A et B, choisies ici entre 0 et 3 ; l'allure des courbes dépendant de ces variables, on pourra modifier leur intervalle de valeurs, ou même les entrer à la main, en écrivant 30 INPUT A et 40 INPUT B.

La boucle I (100-145) réalise le tracé, point par point ; le rayon R est calculé à la ligne 110, et le passage en coordonnées cartésiennes (indispensable pour PLOT) est effectué aux lignes 120 et 130.

La ligne 140 réalise l'impression des points.

La ligne 150 affiche le message FINI.

Les lignes 160 et 170 permettent l'exécution d'un autre dessin, dès que l'on effleure une touche du clavier.

POUR JOUER :

On lance le programme en tapant RUN.

Le dessin sera ensuite exécuté automatiquement.

Le message FINI (en bas, à gauche) signale la fin du tracé.

Pour exécuter un autre dessin, taper une touche quelconque.

Dans la version Spectrum, il faut taper 0 si l'on veut effacer le dessin précédent (sinon, le nouveau dessin se superpose à l'ancien).

Taper BREAK pour sortir du programme.

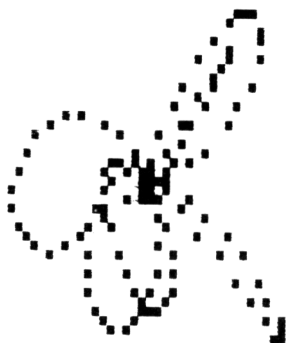
EXTENSIONS POSSIBLES :

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

```

5 CLS
10 LET H=3
20 LET K=6
30 LET A=3*RND
40 LET B=3*RND
100 FOR I=0 TO PI+PI STEP 3*PI/
180
110 LET R=H*SIN (3*I)
120 LET X=30+INT (K*R*COS (A*I)
)
130 LET Y=24+INT (K*R*SIN (B*I)
)
140 PLOT X,Y
145 NEXT I
150 PRINT AT 20,0;"FINI"
160 PAUSE 4E4
170 RUN

```



FINI

```

5 BORDER 5: BRIGHT 1: CLS
10 LET Z=7
20 LET K=8
30 LET T=3
40 LET A=3*RND: LET B=3*RND
50 INK INT (4*RND+1)
100 FOR H=0 TO PI*2 STEP PI/180
110 LET R=Z*SIN (H*T)
120 LET X=120+INT (K*R*COS (A*H))
130 LET Y=90+INT (K*R*SIN (B*H))
150 PLOT X,Y
160 NEXT H
165 PRINT AT 20,0;"FINI"
170 BEEP .6,10: PAUSE 4E4
175 PRINT AT 20,0;" "
180 IF INKEY$="0" THEN RUN
190 GO TO 40

```



FINI

Machinamot

REFERENCES :

Mode SLOW ou FAST.

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 ZX 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 :

La ligne 10 permet d'entrer la liste des préfixes dans une variable A\$ (il sera bien sûr possible d'ajouter d'autres préfixes, en ne dépassant pas 5 lettres et en ajoutant, si nécessaire, des blancs).

La ligne 20 contiendra, de la même façon, la liste des suffixes qui sont placés dans une autre variable alphanumérique B\$.

La ligne 30 effectue le choix d'un préfixe au hasard.

La ligne 40 effectue le choix d'un suffixe au hasard.

Le mot obtenu est affiché par la ligne 50, et la ligne 70 permettra de recommencer un tirage.

POUR JOUER :

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

Pour obtenir un nouveau mot, il convient d'appuyer sur une touche quelconque (sauf BREAK) ; un nouveau mot sera alors affiché à la suite du précédent.

Taper BREAK pour sortir du programme.

EXTENSIONS POSSIBLES :

On pourra très facilement augmenter la liste des mots disponibles (sur ZX SPECTRUM, il sera préférable d'utiliser des instructions DATA qui permettront de ne pas avoir de contrainte sur la longueur des mots).

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 ZX.

```

10 LET A$="ANDROCOSMOMICROHELIO
0 PEDOPORNOPALEOMACROHYDRO"
20 LET B$="CRATEPHILEPHOBETROP
EGYNE MANE NAUTEPHAGEFUGE "
30 LET I=5*INT ((LEN A$/5)*RAND
)+SGN PI
40 LET J=5*INT ((LEN B$/5)*RAND
)+SGN PI
50 PRINT A$(I TO I+4)+B$(J TO
J+4)
60 PAUSE 4E4
70 GOTO 30

```

ANDRONAUTE
 MICROMANE
 MACROPHILE
 PALEONAUTE
 COSMOPHAGE
 PALEOCRATE
 MACROFUGE
 HYDROCRATE
 HYDROPHAGE
 HYDROPHAGE
 PALEOTROPE
 MICROPHOBE
 HELIOMANE
 HYDROPHAGE

```

10 LET A$="ANDROCOSMOMICROHELIO PEDOPOR
NOPALEOMACROHYDROSTENOPATHO OLEOMYTHOHOM
0 OCTO"
20 LET B$="CRATEPHILEPHOBETROPEGYNE MAN
E NAUTEPHAGEFUGE PATHECARDERMSCOPEVORE
PHONE"
30 LET I=5*INT ((LEN A$/5)*RAND)+1
40 LET J=5*INT ((LEN B$/5)*RAND)+1
50 PRINT INVERSE 1; INK INT (5*RND+1);A
$(I TO I+4)+B$(J TO J+4)
60 BEEP .3,18: PAUSE 4E4
70 GO TO 30

```

MICROCRATE
 STENOCARDE
 PALEOFUGE
 HYDRONAUTE
 OLEOFUGE
 MYTHOGYNE
 PORNOPHOBE
 HYDROPHOBE
 ANDROPATHE
 PORNQGYNE
 STENOFUGE
 STENOFUGE
 HELIOPHONE
 HELIODERME
 MACROTROPE
 STENOCARDE
 OLEOFUGE
 OCTOCARDE
 OCTODERME
 MICROPHONE
 MICRONAUTE

Compose

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : facile.

Catégorie de jeu : composition musicale.

Difficulté du jeu : aucune, a priori.

LE JEU :

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

LE PROGRAMME :

La boucle I (10-30) dessine la portée. La variable N\$, en ligne 40, contient les noms des notes.

La deuxième boucle I (50-100) permet de faire le choix d'une note de façon aléatoire, à la ligne 60 ; son nom est calculé en ligne 70 et la note est dessinée par la ligne 80.

La ligne 110 permet de recommencer.

POUR JOUER :

Lancer l'exécution du programme en tapant RUN.

La portée sera alors très 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 la touche BREAK provoquera la fin de la partie.

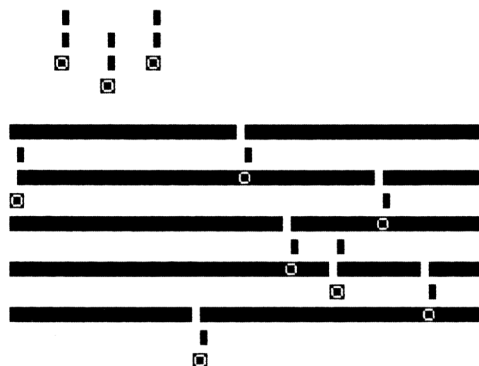
EXTENSIONS POSSIBLES :

On pourra affiner les règles de composition, afficher davantage de notes, introduire des dièses et des bémols, améliorer le graphisme (notamment dans la version ZX-Spectrum), pour faire jouer la mélodie affichée (ceci est réalisé dans la version Spectrum, mais les notes ne correspondent pas aux notes réellement affichées : il faut recalculer les intervalles).

```

5 CLS
10 FOR I=SGN PI TO 5
20 PRINT AT INT PI+I+I,NOT PI;
"
30 NEXT I
40 LET N$="SI LA SOLFA MI RE D
0 "
50 FOR I=SGN PI TO 10
60 LET X=INT (14*RND+SGN PI)
70 LET D$=N$(INT PI*(X-7*(X>7)
)-2 TO INT PI*(X-7*(X>7)))
80 PRINT AT X-SGN PI,INT PI*I-
INT PI;"AT X,INT PI*I-INT PI;
"AT X+SGN PI,INT PI*I-INT PI;
"
90 PRINT AT 20,INT PI*I-INT PI
D$
100 NEXT I
110 PAUSE 4E4
120 RUN

```

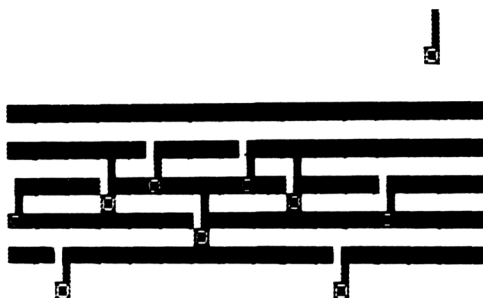


DO SI LA SI DO RE SOLFA SI MI

```

5 CLS
10 FOR I=1 TO 5
20 PRINT INK 5;AT 3+I+I,0;"
"
30 NEXT I
40 LET N$="SI LA SOLFA MI RE DO "
50 FOR I=1 TO 10
60 LET X=INT (14*RND+1)
70 LET D$=N$(3*(X-7*(X>7))-2 TO 3*(X-7*
(X>7)))
80 PRINT INVERSE 1; INK 2;AT X-1,3*I-3;
" ";AT X,3*I-3;" ";AT X+1,3*I-3;"0"
90 PRINT AT 20,3*I-3;D$: BEEP .4,20-X
100 NEXT I
110 PAUSE 4E4
120 RUN

```



SOLDO LA SI FA SI LA DO SOLSI

Nicomaque

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : moyenne.

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

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

LE PROGRAMME :

La ligne 10 permet de confectionner l'entier N.

La boucle I (20-70) calcule les trois restes successifs, et la boucle emboîtée J (40-60) permet de les afficher sous une forme assez visuelle, aux bons emplacements.

La ligne 80 est là pour indiquer de quelle division il s'agit.

Le nombre proposé par le joueur est entré à la ligne 100 ; il est affiché par la ligne 105 et testé dans une variable logique R\$ qui permettra, ensuite, l'affichage de fin de partie (BRAVO ou PERDU).

La ligne 120 réalise l'affichage de fin.

Les lignes 130 et 150 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 points (l'absence de points signifiant : reste 0).

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

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

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

EXTENSIONS POSSIBLES :

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

```

5 CLS
10 LET X=INT (100*RND+SGN PI)
20 FOR I=SGN PI TO INT PI
30 LET A=X-(I+I+SGN PI)*INT (X
/(I+I+SGN PI))
40 FOR J=SGN PI TO A
50 PLOT 15*I+2*J+4,26
60 NEXT J
70 NEXT I
80 PRINT AT 10,NOT PI;"RESTE";
TAB 9;"PAR 3";TAB 17;"PAR 5";TAB
25;"PAR 7"
90 PRINT AT 15,NOT PI;"VOTRE E
SSAI :";
100 INPUT N
105 PRINT N
110 LET R$=("BRAVO" AND N=X)+("
PERDU" AND N<>X)
120 PRINT R$
130 PRINT "AUTRE PARTIE (O/N) ?
"
140 INPUT R$
150 IF R$(1)="O" THEN RUN

```

```

  **      ****      **
RESTE      PAR 3      PAR 5      PAR 7

```

```

VOTRE ESSAI :44
BRAVO
AUTRE PARTIE (O/N) ?

```

```

5 CLS : BORDER 4
10 LET X=INT (100*RND+1)
20 FOR I=1 TO 3
30 LET A=X-(I+I+1)*INT (X/(I+I+1))
40 FOR J=1 TO A
50 INK 1: CIRCLE 60*I+8*J+15,80,3
60 NEXT J
70 NEXT I
80 PRINT INVERSE 1; INK 2;A1 10,0;"REST
E";TAB 9;"PAR 3";TAB 17;"PAR 5";TAB 25;"P
AR 7"
90 PRINT AT 15,0;"VOTRE ESSAI :";
100 INPUT N
105 PRINT N
110 LET R$=("BRAVO" AND N=X)+("PERDU" AN
D N<>X)
120 PRINT FLASH 1; BRIGHT 1;R$: BEEP 1,1
0
130 PRINT "AUTRE PARTIE (O/N) ?"
140 INPUT R$
150 IF R$="O" THEN RUN

```

```

RESTE      PAR 3      PAR 5      PAR 7
          O          OO          OO

```

```

VOTRE ESSAI :37
BRAVO
AUTRE PARTIE (O/N) ?

```

Permutations

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion.

Difficulté du jeu : moyenne.

LE JEU :

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

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

LE PROGRAMME :

La ligne 30 permet de choisir le nombre de lettres, et la ligne 40 contrôle que l'on n'en demande pas plus de 12.

La ligne 60 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 70), supprimées dans A\$ et replacées en C\$, dans le bon ordre (lignes 75 et 80).

La ligne 90 contrôle le nombre de lettres de B\$.

Les affichages sont effectués aux lignes 100 et 115.

Le retournement est fait, lettre par lettre, dans la boucle I (140-160) ; le résultat est placé dans B\$ en ligne 170, affiché par la ligne 180 et comparé au résultat final en ligne 190.

Le nombre d'essais sera affiché en fin de partie, grâce à la ligne 200.

POUR JOUER :

Après avoir tapé RUN, on obtient l'affichage de la question NBRE DE LETTRES (DE 1 A 12) ; on répond en frappant N, suivi de NEWLINE. 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 NEWLINE, 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 ou 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.).

ZX-81

```

5 CLS
10 LET A$="ABCDEFGHJKLM"
20 PRINT "NBRE. DE LETTRES <=1
2
30 INPUT N
40 IF N>CODE "Z" THEN RUN
50 LET C$=""
55 LET B$=""
60 LET X=INT (RND*N+SGN PI)
65 IF A$(X)=" " THEN GOTO CODE
"U"
70 LET B$=A$(X)+B$
75 LET C$=C$( TO X-SGN PI)+A$(
X)+C$(X+SGN PI TO )
80 LET A$=A$( TO X-SGN PI)+" "
+A$(X+SGN PI TO )
85 LET S=NOT PI
90 IF LEN B$<>N THEN GOTO CODE
"U"
95 LET C$=C$( TO LEN B$)
100 PRINT "DEBUT :";B$;" FIN :
";C$
115 PRINT AT 7,SGN PI;"PERM. ";
S
120 INPUT L
125 LET S=S+SGN PI
130 LET A$=""
140 FOR I=SGN PI TO L
150 LET A$=A$+B$(L+SGN PI-I)
160 NEXT I
170 LET B$=A$+B$(L+SGN PI TO )
180 PRINT "RESULTAT ";B$
190 IF B$<>C$ THEN GOTO 112
200 PRINT "TROUVE EN ";S;" ESSA
IS"
210 PRINT "AUTRE PARTIE (O OU N
)"
220 INPUT A$
230 IF A$="O" THEN RUN

```

NBRE. DE LETTRES <=12
DEBUT :DABEC FIN :ABCD

PERM. 5
RESULTAT ABCD
TROUVE EN 6 ESSAIS
AUTRE PARTIE (O OU N)

ZX-SPECTRUM

```

5 CLS
10 LET A$="ABCDEFGHJKLM"
20 PRINT "NOMBRE DE LETTRES (DE 1 A 12)
";
30 INPUT N: PRINT N
40 IF N>12 THEN RUN
50 LET C$=""
55 LET B$=""
60 LET X=INT (RND*N+1)
65 IF A$(X)=" " THEN GO TO 60
70 LET B$=A$(X)+B$
75 LET C$(X)=A$(X)
80 LET A$(X)=" "
85 LET S=1
90 IF LEN B$<>N THEN GO TO 60
95 LET C$=C$( TO LEN B$)
100 PRINT ;"DEBUT :"; PAPER 4;B$: PRINT
"FIN :"; PAPER 5;C$
105 PRINT AT 4,0;"RESULTAT ";S-1;" :";B$
112 BEEP 1,10: PAUSE 5: BEEP 1,10: PAUSE
5: BEEP 1,10: PRINT AT 12,10;"
"
115 PRINT AT 7,1;"PERMUTATION :"; FLASH
1; INK 2;S
120 INPUT L
125 LET S=S+1
130 LET A$=""
140 FOR I=1 TO L
150 LET A$=A$+B$(L-I+1)
160 NEXT I
170 LET B$=A$+B$(L+1 TO )
180 PRINT INK 2;AT 12,0;"RESULTAT :"; PA
PER 7; INK 1; FLASH 1;B$
190 IF B$<>C$ THEN GO TO 105
200 PRINT "TROUVE EN ";S-1;" ESSAIS"
210 PRINT "AUTRE PARTIE (O OU N)?"
220 INPUT A$
230 IF A$="O" THEN RUN

```

NOMBRE DE LETTRES (DE 1 A 12) 5
DEBUT :DABEC
FIN :ABCDE

RESULTAT 5 :EDCBA

PERMUTATION : 6

RESULTAT :ABCDE
TROUVE EN 6 ESSAIS
AUTRE PARTIE (O OU N)?

Tour de cartes

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de prestidigitation.

Difficulté du jeu : facile.

LE JEU :

C'est un tour assez classique mais, ici, les cartes sont manipulées sous vos yeux par le ZX 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 ZX 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 ZX est en mesure de deviner la carte choisie (sur la version ZX-81, à cause du manque de place mémoire, le ZX ne désigne pas la carte, ce qui pourra vous permettre d'épater facilement vos amis, si vous reprenez que la carte se trouve à la 4^e place de la 2^e colonne).

LE PROGRAMME :

La boucle I (20-80) permet de choisir, au hasard, 21 cartes sur 52, et la boucle emboîtée J (40-60) permet de s'assurer que l'on ne réalise pas deux fois le même tirage. La ligne 100 contient les couleurs des cartes ainsi que les noms des cartes figures (V, D, R, A), dans une variable alphanumérique B\$.

La boucle I (110-160) réalise l'affichage du jeu.

La ligne 120 décode la couleur de la carte.

La ligne 130 choisit la position d'affichage.

La ligne 140 décode la hauteur.

L'affichage proprement dit est effectué à la ligne 150.

La ligne 170 permet au joueur d'entrer le numéro de la colonne où se trouve sa carte. La boucle J (185-195) réalise le brouillage, et les boucles emboîtées J (200-240) et I (210-230) 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 NEWLINE.

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 se trouve en 4^e position dans la 2^e colonne (le ZX-81 ne permet pas, dans sa version 1K, de la nommer, mais ceci est fait dans la version Spectrum, plus complète).

On sortira du programme en tapant STOP, suivi de NEWLINE, à la place du numéro de la colonne.

EXTENSIONS POSSIBLES :

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

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


```

10 LET A$=""
12 LET R=SGN PI+SGN PI
15 LET Q=R+INT PI
20 FOR I=SGN PI TO INT PI*Q
30 LET B$=CHR$ INT (CODE "2"+C
ODE "0"*RND)
40 FOR J=SGN PI TO LEN A$
50 IF B$=A$(J) THEN GOTO CODE
"2"
60 NEXT J
70 LET A$=A$+B$
80 NEXT I
100 LET B$="TRCACOPIVADRA"
110 FOR I=SGN PI TO LEN A$
120 LET S=INT ((CODE A$(I)-CODE
"2")/CODE "$")
130 LET J=Q*INT ((I-SGN PI)/Q)
140 LET T=SGN PI+CODE A$(I)-COD
E "2"-CODE "$"+S
150 PRINT AT I-J,J:(STR$(T+SGN
PI) AND T=(Q+R)+(B$(T) AND T>Q+
R);TAB J+INT PI;B$(R+S+SGN PI TO
R+S+R)
160 NEXT I
170 INPUT N
175 CLS
180 LET R$="20120"
182 LET B$=""
185 FOR J=NOT PI TO R
190 LET B$=B$+A$(Q*VAL R$(N+J)+
SGN PI TO Q*VAL R$(N+J)+Q)
195 NEXT J
197 LET A$=""
200 FOR J=NOT PI TO R
210 FOR I=SGN PI TO Q
220 LET A$=A$+B$(INT PI*I-R+J)
230 NEXT I
240 NEXT J
250 GOTO CODE "COS" /R

```

```

J 00 00 00 00 00 00
0 00 00 00 00 00 00
00 00 00 00 00 00
00 00 00 00 00 00
00 00 00 00 00 00
00 00 00 00 00 00
00 00 00 00 00 00
00 00 00 00 00 00
00 00 00 00 00 00
00 00 00 00 00 00

```

```

10 LET A$=""
12 LET R=2
15 LET Q=7
20 FOR I=1 TO 21
30 LET B$=CHR$ INT (38+52*RND)
40 FOR J=1 TO LEN A$
50 IF B$=A$(J) THEN GO TO 30
60 NEXT J
70 LET A$=A$+B$
80 NEXT I
90 LET H=3
95 LET P=1
100 LET B$="TRCACOPIVADAROAS"
110 FOR I=1 TO LEN A$*P+11*NOT P
120 LET S=INT ((CODE A$(I)-38)/13)
130 LET J=Q*INT ((I-1)/Q)
140 LET T=CODE A$(I)-37-13*S
145 IF NOT P THEN PRINT INVERSE 1;AT 0,0
;"VOTRE CARTE EST :"; INVERSE 0; PAPER 2;
" "
150 PRINT AT (I-J)*P,J*P+18*NOT P;(STR$(
(T+1) AND T=(Q+R)+(B$(ABS (2*T-11) TO ABS
(2*T-10)) AND T>Q+R);AT (I-J)*P,(J+3)*P+
21*NOT P;B$(R*S+1 TO R*S+R)
160 NEXT I
165 BEEP 0.2,20: IF NOT P THEN BEEP 0.5,
10: BEEP 0.5,30: BEEP 0.5,20: STOP
170 INPUT "QUELLE COLONNE :";N
175 CLS
177 LET H=H-1
180 LET R$="20120"
182 LET B$=""
185 FOR J=0 TO R
190 LET B$=B$+A$(Q*VAL R$(N+J)+1 TO Q*VA
L R$(N+J)+Q)
195 NEXT J
196 IF NOT H THEN LET A$=B$: LET P=0: GO
TO 100
197 LET A$=""
200 FOR J=0 TO R
210 FOR I=1 TO Q
220 LET A$=A$+B$(3*I-R+J)
230 NEXT I
240 NEXT J
250 GO TO 100

```

VOTRE CARTE EST : AS CO

Niveau 4

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Tirage de dés	Suffisamment souple pour être utilisé partout.
Jeu de loto	Gagnez sans problème, au prochain tirage.
La bombe	Saurez-vous la découvrir à temps ? (1 joueur).
Ardoise magique	Mettez au point vos créations graphiques, sans papier ni crayon.
Affichage publicitaire	Employez les possibilités de traitements de chaîne du ZX.
Evasion	Il vous faudra beaucoup de chance pour réussir.
Chapeaux	Soyez très observateur, et vous réussirez (2 joueurs).
Combat aérien	Pas si facile de descendre un appareil qui bouge tout le temps !
Dessin de dés	Améliorez vos jeux de dés en incorporant ce programme.
Le jeu du missile	Une mission bien périlleuse (un seul joueur).
Le singe	Un jeu de lettres pour 2 joueurs, où il faut de bons réflexes.
Le jeu du 15	Un jeu de réflexion utilisant des chiffres (2 joueurs).
Adresse	Jeu de réflexe pour 1 joueur, très visuel.
Sous-marin	Un classique des jeux vidéos (1 joueur).
Zombies	Un peu farfelu, mais très visuel (1 joueur).
Diago	Etes-vous vraiment sûr de vos réflexes ? (1 joueur).
Topologie	Une direction à explorer, pour d'autres jeux (1 ou 2 joueurs).
Chasse au sous-marin	Il vous faudra pas mal de réflexion, et beaucoup de chance.
Le lézard d'Alice	Beaucoup de hasard, mais très spectaculaire (1 joueur).
Calculatrice	Un programme de simulation, qui pourra facilement être amélioré.

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 Basic Sinclair, cette propriété est offerte, par la **fonction INKEY\$** qui effectue une structation de toutes les 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, au contraire : tant que INKEY\$ est vide (on n'a rien frappé) on attend, et l'on sort dès que l'on a frappé quelque chose.

Des problèmes de rebonds pourront éventuellement se poser (notamment sur ZX-81) lors du décodage du caractère saisi par INKEY\$.

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 Basic du ZX-81 dispose d'une instruction qui permet de remédier à cela et que l'on pourra utiliser pour de nombreux jeux : il s'agit de l'**instruction SCROLL** qui fait simplement monter d'une ligne, vers le haut, les divers objets (textes ou graphiques) présents sur l'écran au moment de son exécution ; la ligne du haut est, bien évidemment, perdue. Cela permet

de simuler la montée vers le haut de l'écran d'un ensemble d'objets, avec effacement automatique de la position précédente. Bien entendu, un objet, qui devrait rester immobile lors d'un SCROLL, devra être redessiné en permanence à sa bonne position (et effacé à la mauvaise, sauf s'il est en haut de l'écran).

Le ZX-Spectrum ne dispose pas d'une instruction SCROLL, mais réalise un SCROLL sur commande, lorsque l'écran est plein ; on pourra mettre cette propriété à profit en remplaçant le SCROLL du ZX-81 par une séquence adaptée au jeu considéré.

Tirage de dés

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : aucune.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : aucune.

LE JEU :

Le programme proposé ne réalise pas un jeu en lui-même, mais pourra être utilisé en sous-programme pour de nombreux jeux.

Il offre simplement la possibilité de lancer de 1 à 9 dés, par simple pression d'une touche. Les dés sont ensuite affichés avec un graphisme très simplifié.

LE PROGRAMME :

La ligne 20 saisit le caractère frappé au clavier et le transforme en nombre (si ce n'est pas possible, le programme s'arrête avec un message d'erreur). Une variable D est dimensionnée en ligne 35 pour recevoir la valeur indiquée par chaque dé.

Une boucle J (43-45) est utilisée pour dessiner un fond gris, et une autre boucle I (40-80) sert au tirage (tirage en ligne 50 et impression en ligne 70).

Les lignes 90 à 110 offrent la possibilité de refaire un tirage ; elles ne sont pas indispensables.

POUR JOUER :

On lance le programme par l'ordre RUN et on obtient l'affichage NOMBRE DE DES A LANCER auquel on répond en frappant un chiffre.

Le résultat est alors automatiquement affiché ; il suffit de presser une touche quelconque pour recommencer le programme.

Pour arrêter, il est nécessaire de taper BREAK.

EXTENSIONS POSSIBLES :

On pourra, bien sûr, utiliser le canevas fourni par ce programme pour la réalisation de tout jeu se servant de dés (421, YAM'S, CRAPS, etc.).

```

10 PRINT "NOMBRE DE DES A LANCER "
15 PAUSE 4E4
20 LET N=VAL INKEY$
30 CLS
35 DIM D(N)
40 FOR I=1 TO N
43 FOR J=1 TO 3
44 PRINT AT 8+J,1;TAB 3*I-1;" "
";TAB 3*I;" ";TAB 3*I+1;" "
45 NEXT J
50 LET D(I)=INT (6*RND+1)
70 PRINT AT 10,1;TAB 3*I;D(I)
80 NEXT I
90 PAUSE 4E4
100 CLS
110 GOTO 10

```

```

5 4 4

```

```

10 PRINT AT 0,0;"NOMBRE DE DES A LANCER "
: "
15 PAUSE 4E4
20 LET N=VAL INKEY$
30 PRINT AT 0,25;N
35 DIM D(N)
40 FOR I=1 TO N
43 FOR J=1 TO 3
44 PRINT INK 2;AT 8+J,3*I-1;" ";AT 8+J,
3*I;" ";AT 8+J,3*I+1;" "
45 NEXT J
50 LET D(I)=INT (RND*6)+1
70 PRINT INK 2; INVERSE 1;AT 10,3*I;D(I)
)
80 NEXT I
90 BEEP .3,10: BEEP .3,18: BEEP .3,8
100 PAUSE 4E4: PAPER 5: CLS
110 GO TO 10

```

NOMBRE DE DES A LANCER : 5

```

4 5 4 4 6

```

Jeu de loto

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : facile.

Catégorie de jeu : hasard (plusieurs joueurs).

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

On utilisera un tableau N(7), dimensionné en ligne 10, pour enregistrer les résultats.

Le tirage des nombres est effectué dans une boucle I (20-70), une deuxième boucle J (40-60) étant utilisée pour contrôler qu'il n'y ait pas deux tirages identiques.

Le tirage (ou le retraitage) est effectué en ligne 50.

La boucle I (90-110) affiche le résultat, et la ligne 120 affiche le numéro complémentaire.

Les lignes 130 à 150 s'occupent du cas où l'on voudrait refaire une partie.

POUR JOUER :

Il suffit de taper RUN pour lancer le programme et obtenir l'affichage d'un premier loto.

La question VOULEZ-VOUS UN AUTRE TIRAGE (O ou 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.

```

5 CLS
10 DIM N(7)
20 FOR I=1 TO 7
30 LET N(I)=INT (RND*49+1)
40 FOR J=1 TO I-1
50 IF N(I)=N(J) THEN GOTO 30
60 NEXT J
70 NEXT I
80 PRINT AT 3,8;"TIRAGE DU LOT
0"
90 FOR I=1 TO 6
100 PRINT AT 5,4*I+1;N(I)
110 NEXT I
120 PRINT AT 8,1;"NUMERO COMPLE
MENTAIRE ";N(7)
130 PRINT AT 12,1;"VOULEZ-VOUS
UN AUTRE TIRAGE (O/N) ?"
140 PAUSE 4E4
150 IF INKEY$="0" THEN RUN

```

TIRAGE DU LOT0

18 9 37 48 29 43

NUMERO COMPLEMENTAIRE 13

2X81

VOULEZ-VOUS UN AUTRE TIRAGE (O/N) ?

```

5 CLS
10 DIM N(7)
20 FOR I=1 TO 7
30 LET N(I)=INT (RND*49+1)
40 FOR J=1 TO I-1
50 IF N(I)=N(J) THEN GOTO 30
60 NEXT J
70 NEXT I
80 PRINT INK 2; FLASH 1;AT 3,7;"TIRAGE
DU LOT0"
90 FOR I=1 TO 6
100 PRINT PAPER 6; INK 1; BRIGHT 1;AT 8,
4*I-1;N(I)
110 NEXT I
120 PRINT AT 10,1;"NUMERO COMPLEMENTAIRE
: "; INK 3; FLASH 1;N(7); BEEP .4,10; B
EEP .4,15; BEEP .4,10
130 PRINT INVERSE 1;AT 15,0;"VOULEZ-VOUS
UN AUTRE TIRAGE(O/N)"
140 PAUSE 4E4
150 IF INKEY$="0" THEN RUN

```

TIRAGE DU LOT0

28 21 30 34 49 33

NUMERO COMPLEMENTAIRE : 17

VOULEZ-VOUS UN AUTRE TIRAGE(O/N)

La bombe

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

Vous êtes sensé trouver, à l'aide d'un détecteur spécialisé, un colis radioactif piégé disposé par un voyageur mal intentionné, dans une chambre d'hôtel. Seul problème, mais de taille : l'hôtel comporte un million de chambres (100*100*100).

Votre détecteur affiche, en permanence, un signal qui vous renseigne sur la proximité de la bombe, et le temps qui vous reste est affiché en permanence, décompté de 10 en 10.

Pourrez-vous découvrir la bombe avant l'explosion ?

LE PROGRAMME :

Une variable numérique R est dimensionnée à 3 en ligne 30, pour contenir les trois coordonnées définissant la position du joueur.

La position de la chambre contenant le colis piégé est définie par un tirage aléatoire de 3 variables A, B, C, aux lignes 40, 50, 52.

La boucle I (55-62) dessine le détecteur.

Le signal de proximité est calculé en ligne 63 et la ligne 90 permet l'affichage des coordonnées de la chambre à découvrir, dans le cas où l'on a perdu.

La ligne 100 permet l'affichage, à la fois du signal et du temps restant à courir.

La ligne 110 réalise le décompte du temps.

La ligne 120 contrôle si l'on a trouvé ou non.

La boucle I (140-180) permet d'entrer les nouvelles coordonnées, en utilisant le sous-programme placé aux lignes 200 à 210.

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 NEWLINE, 45 puis NEWLINE, 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'explosion, d'afficher symboliquement la position dans l'hôtel, etc.

ZX-81

```

20 LET L=VAL "100"
25 LET T=L+L
30 DIM R( INT PI)
40 LET A=INT (RND*L)
50 LET B=INT (RND*L)
60 LET C=INT (RND*L)
70 FOR I=NOT PI TO CODE "1"
80 PRINT AT 4,I;"■";AT 5,I;"■"
;AT 6,I;"■"
82 NEXT I
83 LET S=ABS ((A/L+B+C*L)-(R(5
GN PI)/L+R(2)+R(INT PI)*L))
90 IF NOT T THEN PRINT A,B,C
95 PRINT AT 5,15;" "
100 PRINT AT 5,INT PI;S;TAB 13;
T
110 LET T=T-INT SQR L
120 IF NOT S THEN STOP
130 PRINT AT SQR L,INT PI;"- X
;AT 12,INT PI;"- Y";AT 14,INT
PI;"- Z"
140 FOR I=SGN PI TO INT PI
150 LET R$=""
170 GOSUB CODE "COS "
175 LET R(I)=VAL R$
180 NEXT I
190 GOTO CODE "Z"
200 PRINT AT 8+I+I,7;"?"R$;"
"
205 PAUSE 4E4
210 IF CODE INKEY$=VAL "118" TH
EN RETURN
220 LET R$=R$+INKEY$
240 GOTO CODE "COS "

```

```

757.388 190

```

```

- X ? 117
- Y 70
- Z 790

```

ZX-SPECTRUM

```

20 LET L=100
25 LET T=L+L
30 DIM R(3)
40 LET A=INT (RND*L)
50 LET B=INT (RND*L)
52 LET C=INT (RND*L)
55 FOR I=0 TO 17
60 PRINT INK 5;AT 4,I;"■";AT 5,I;"■";AT
6,I;"■"
62 NEXT I
63 LET S=ABS ((A/L+B+C*L)-(R(1)/L+R(2)+
R(3)*L))
90 IF NOT T THEN PRINT INK 4; FLASH 1;A
T 20,0;"PERDU :";A;" ";B;" ";C;" "; BE
EP .3,0; BEEP .3,0; BEEP .3,0
95 PRINT AT 5,15;" "
100 PRINT AT 5,3;S;AT 5,13; FLASH 1;T
110 LET T=T-INT SQR L; BEEP 1,15
120 IF NOT S THEN PRINT AT 21,0; PAPER 2
; FLASH 1;"BRAVO"
130 PRINT INK 2;AT SQR L,3;"- X : ";
AT 12,3;"- Y : ";AT 14,3;"- Z : "
140 FOR I=1 TO 3
160 LET R$=""
170 GO SUB 200
175 LET R(I)=VAL R$
180 NEXT I
190 GO TO 63
200 PRINT AT 8+2*I,8; INK 2; FLASH 1;"?"
; INK 0; FLASH 0;" ";R$;" "
205 PAUSE 4E4
210 IF CODE INKEY$=13 THEN BEEP .3,10; R
ETURN
220 LET R$=R$+INKEY$
240 GO TO 200

```

```

3396.92 180

```

```

- X :? 30
- Y :? 60
- Z :? 70

```

Ardoise magique

REFERENCES :

Mode SLOW ou FAST.

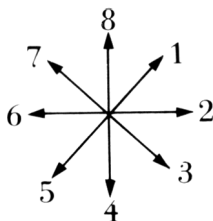
Difficulté de programmation : facile.

Catégorie de jeu : dessin.

Difficulté du jeu : facile.

LE JEU :

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



On dispose aussi de la possibilité de déplacer le point sur l'écran sans laisser de trace en appuyant sur la touche 0 (ceci permettra aussi l'effacement partiel) et l'on pourra réinscrire en tapant 9.

Pour effacer complètement un dessin, on devra frapper la touche P et, pour garder une trace sur l'imprimante, on sortira du programme par BREAK avant de faire COPY.

LE PROGRAMME :

La ligne 100 réalise l'affichage d'un point sur l'écran.

Les lignes 160 à 170 saisissent au vol la touche frappée (de 0 à 9, ainsi que P).

La ligne 250 appelle l'un des sous-programmes effectuant le déplacement.

Les lignes 300 et 350 contrôlent que l'on ne sorte pas de l'écran.

La ligne 370 permet de tout effacer si l'on appuie sur la touche P.

POUR JOUER :

On lancera le programme en tapant RUN, puis on pourra déplacer le point qui apparaît en bas à gauche de l'écran, en utilisant les 8 touches de déplacement.

La touche 0 permet de déplacer le point sans laisser de trace (il y aura effacement si l'on repasse sur un ancien trait).

La touche 9 permet d'inscrire le déplacement.

La touche P efface tout, et l'on recommence.

Pour sortir du programme, il faudra taper BREAK.

EXTENSIONS POSSIBLES :

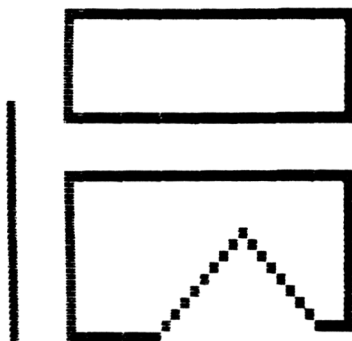
On pourra envisager des possibilités particulières pour tracer des droites, des cercles, dessiner avec plusieurs couleurs, améliorer la finesse.

On pourra ajouter, si besoin est, un programme qui sauve le dessin obtenu pour une utilisation ultérieure.

```

5 CLS
10 LET X=10
20 LET Y=X
80 LET S=NOT PI
100 PLOT X,Y
150 PAUSE 4E4
160 IF INKEY$="0" THEN GOTO 500
165 IF INKEY$="9" THEN GOTO 80
166 IF INKEY$="P" THEN RUN
170 LET F=2*((VAL INKEY$)<5)-50
N PI
190 LET A=X
200 LET B=Y
250 GOSUB 1000*((VAL INKEY$)+2+
(F-SGN PI))
300 LET C=X>NOT PI AND X<40 AND
Y>NOT PI AND Y<30
350 IF NOT C THEN GOSUB 5000
370 IF S THEN UNPLOT A,B
400 GOTO 100
500 LET S=SGN PI
510 GOTO 100
1000 LET Y=Y+F
2000 LET X=X+F
2010 RETURN
3000 GOSUB 2000
4000 LET Y=Y-F
4010 RETURN
5000 LET X=A
5010 LET Y=B
5020 RETURN

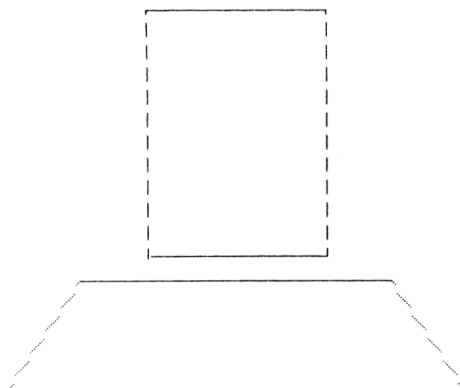
```



```

5 CLS
10 LET X=10
20 LET Y=X
80 LET S=0
100 PLOT X,Y
150 PAUSE 4E4
160 IF INKEY$="0" THEN GO TO 500
165 IF INKEY$="9" THEN GO TO 80
166 IF INKEY$="P" THEN RUN
168 LET R=VAL INKEY$
170 LET F=2*(R<5)-1
190 LET A=X
200 LET B=Y
250 GO SUB 1000*(R+2*(F-1))
300 LET C=X>0 AND X<255 AND Y>0 AND Y<17
5
350 IF NOT C THEN GO SUB 5000
370 IF S THEN PLOT OVER 1;A,B
400 GO TO 100
500 LET S=1
510 GO TO 100
1000 LET Y=Y+F
2000 LET X=X+F
2010 RETURN
3000 GO SUB 2000
4000 LET Y=Y-F
4010 RETURN
5000 LET X=A
5010 LET Y=B
5020 RETURN

```



Affichage publicitaire

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : moyenne.

Catégorie de jeu : affichage publicitaire.

Difficulté du jeu : aucune.

LE JEU :

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

LE PROGRAMME :

Le texte sera introduit dans une variable P\$, à la ligne 30.

L'affichage sera tronqué à 32 caractères au plus, grâce à la ligne 130.

Le défilement vers la gauche est préparé à la ligne 150, et le défilement vers la droite à la ligne 160.

Le texte sera affiché en ligne 140, et la durée de l'affichage (vitesse de défilement) est choisie par la ligne 170.

La ligne 175 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 VOTRE TEXTE, on introduira le texte à faire défiler, suivi de NEWLINE.

On obtient ensuite l'affichage SENS D ou G auquel on répondra en tapant, soit D, soit G, suivi de NEWLINE.

Puis à la question VITESSE, on répondra en introduisant un petit nombre pour une vitesse rapide, et un grand nombre pour une vitesse lente (par exemple, de 0 à 40 ou plus).

Le texte commencera à défiler dès que l'on aura frappé NEWLINE.

Si l'on désire changer les conditions du défilement, il suffit d'appuyer sur la touche O pour retrouver les questions SENS et VITESSE. On tapera BREAK pour interrompre le programme.

EXTENSIONS POSSIBLES :

On pourra introduire de la couleur, réaliser un affichage ondulant, faire remonter les lignes sur l'écran, etc.

```

10 PRINT "ENTREZ VOTRE TEXTE"
20 PRINT
30 INPUT P$
35 LET P$=P$+" "
40 PRINT "DEFILEMENT D OU G : "
50 INPUT S$
60 PRINT "DROITE" AND S$(SGN P
I)="D"; "GAUCHE" AND S$(SGN PI)="
G"
70 PRINT
80 PRINT "VITESSE DE 0 A 50"
90 INPUT V
105 CLS
110 LET V=51-V
120 LET T=LEN P$
130 IF T>CODE "2" THEN LET T=CO
DE "2"
140 PRINT AT INT PI,NOT PI;" "+
P$(SGN PI TO T)+" "
150 IF S$(SGN PI)="G" THEN LET
P$=P$(2 TO LEN P$)+P$(SGN PI)
160 IF S$(SGN PI)="D" THEN LET
P$=P$(LEN P$)+P$(SGN PI TO LEN P
$-SGN PI)
170 PAUSE V*PI
175 IF INKEY$="0" THEN GOTO 40
180 GOTO 140

```

ENTREZ VOTRE TEXTE

DEFILEMENT D OU G :DROITE

VITESSE DE 0 A 50

```

10 PRINT "ENTREZ VOTRE TEXTE"
20 PRINT
30 INPUT P$: LET P$=" "+P$+" ": LET A$=
P$
40 PRINT "DEFILEMENT D OU G :";
50 INPUT S$
60 PRINT "DROITE" AND S$(1)="D"; "GAUCHE
" AND S$(1)="G"
70 PRINT
80 PRINT "VITESSE DE 0 A 50"
90 INPUT V: LET V=51-V
105 BORDER 4: PAPER 6: INK 2: CLS
120 LET T=LEN A$: LET P$=A$
130 IF T>30 THEN LET T=30
140 PRINT INK 2;AT 3,0;" "+P$(1 TO T)+"
"
150 IF S$(1)="G" THEN LET P$=P$(2 TO LEN
P$)+P$(1)
160 IF S$(1)="D" THEN LET P$=P$(LEN P$-1
)+P$(1 TO LEN P$-1)
170 PAUSE V*PI
175 IF INKEY$="0" THEN GO TO 40
180 GO TO 140

```

ENTREZ VOTRE TEXTE

DEFILEMENT D OU G :DROITE

VITESSE DE 0 A 50

Evasion

REFERENCES :

Mode SLOW.

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 ZX 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 :

Les lignes 5 et 7 définissent la position de départ du joueur.

La ligne 10 choisit la coordonnée horizontale de la case où ZX va tirer (liée à votre position), et la ligne 20 choisit la coordonnée verticale (arbitraire) de cette case.

Le déplacement du joueur est contrôlé par les lignes 70 et 80 ; sa trajectoire est tracée par la ligne 60, et sa position actuelle (sur l'écran, un H inversé) est imprimée par la ligne 90.

La ligne 100 regarde s'il est touché ou non.

La ligne 110 effectue le contrôle de fin de partie.

POUR JOUER :

On commence l'exécution du programme par RUN.

Un premier tir a lieu, et ce n'est qu'après que le joueur apparaît, représenté par un H inversé.

Les 4 touches de direction (5, 6, 7, 8) servent à déplacer le joueur.

S'il est touché, on obtient le message PERDU et la partie s'arrête.

S'il réussit à sortir par la droite de l'écran sans être touché, on obtient le message BRAVO, et la partie est également terminée.

On peut terminer à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :

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

```

5 LET X=10
7 LET Y=NOT PI
10 LET H=2*(Y-1)+6*RND
20 LET K=25*RND+5
30 FOR J=SGN PI TO 7*RND+PI
40 PLOT H+4*RND-2,K+4*RND-2
50 NEXT J
60 PRINT AT X,Y;"-"
70 LET X=X+(INKEY$="6")-(INKEY$="7")
80 LET Y=Y+(INKEY$="8")-(INKEY$="5")
90 PRINT AT X,Y;"■"
100 IF ABS (H-Y-Y)+ABS (K-2*(21-X))<PI THEN PRINT "PERDU"
110 IF Y<30 THEN GOTO 10
120 PRINT "BRAVO"

```



PERDU

```

5 LET X=10
7 LET Y=0
10 LET H=8*(Y-1)+16*RND
20 LET K=150*RND+5
25 IF Y<3 THEN LET K=0: GO TO 60
30 FOR J=1 TO 7*RND+3
40 INK 2: PLOT H+6*RND-2,K+6*RND-2
50 NEXT J
60 INK 4: BEEP .3,-20: PRINT AT X,Y;"."
70 LET X=X+(INKEY$="6")-(INKEY$="7")
80 LET Y=Y+(INKEY$="8")-(INKEY$="5")
90 PRINT FLASH 1: INK 5: AT X,Y;"H"
100 IF ABS (H-8*Y)+ABS (K-8*(21-X))<=12
THEN PRINT INK 0: FLASH 1;"PERDU": BEEP .
3,10: BEEP .3,18: BEEP .3,8: STOP
110 IF Y<30 THEN GO TO 10
120 PRINT INK 1: FLASH 1;"BRAVO": STOP

```



PERDU

Chapeaux

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : assez facile.

Catégorie de jeu : jeu de réflexe, pour deux joueurs.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

Les lignes 10 et 20 contiennent le dessin des deux chapeaux (dans deux variables alphanumériques B\$ et C\$).

La ligne 30 contient, dans une autre variable A\$, les symboles qui permettront de dessiner les personnages.

Les lignes 60 à 80 permettent, s'il y a lieu, l'échange des chapeaux.

La variable T de la ligne 90 sert à repérer s'il y a eu échange ou non.

Le dessin des deux personnages est réalisé par une boucle I (130-160), et leur déplacement est réalisé par les lignes 170 et 180.

Le test, permettant ou non le changement des couvre-chefs, est effectué en ligne 110.

Les lignes 190 à 210 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 :

Sur un appareil comme le Spectrum, il sera possible de prévoir un affichage couleur, avec une animation des personnages.

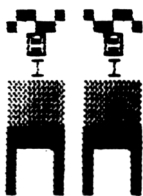
On pourra aussi compliquer le jeu à plaisir avec plusieurs objets échangés de façon aléatoire (il faudrait alors en trouver la liste complète).

ZX-81

```

10 LET B$="██"
20 LET C$="██"
30 LET A$=" A I ██████████ █"
40 LET X=NOT PI
45 LET T=SGN PI
50 LET Y=CODE "-"
60 LET D$=C$
70 LET C$=B$
80 LET B$=D$
90 LET T=ABS (T-SGN PI)
125 CLS
130 FOR I=SGN PI TO 7
140 PRINT AT NOT PI,X;C$;AT NOT
PI,Y;B$
150 PRINT AT I,X;A$(INT PI+I-2
TO INT PI+I);AT I,Y;A$(INT PI+I-
2 TO INT PI+I)
160 NEXT I
170 LET X=X+SGN PI
180 LET Y=Y-SGN PI
182 PAUSE 30
185 IF X=Y AND RND<.5 THEN GOTO
60
187 IF INKEY$="" THEN GOTO 125
190 IF (INKEY$="1" AND T) OR (I
NKEY$="2" AND NOT T) THEN PRINT
"BRAVO GAUCHE";J
200 IF (INKEY$="0" AND T) OR (I
NKEY$="9" AND NOT T) THEN PRINT
"BRAVO DROITE";J
210 PRINT "PERDU GAUCHE" AND (I
NKEY$="2" OR INKEY$="1");"PERDU
DROITE" AND (INKEY$="9" OR INKEY
$="0")

```



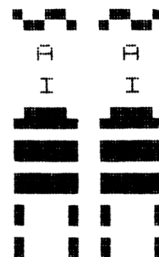
BRAVO DROITE

ZX-SPECTRUM

```

10 LET B$="██"
20 LET C$="██"
30 LET A$=" A I ██████████ █"
40 LET X=0
45 LET T=1
50 LET Y=22
60 LET D$=C$
70 LET C$=B$
80 LET B$=D$
90 LET T=ABS (T-1)
125 PAPER 6: CLS
130 FOR I=1 TO 7
140 PRINT AT 0,X;C$;AT 0,Y;B$
150 PRINT INK 2+(I>2 AND I<5);AT I,X;A$(
3*I-2 TO 3*I); PRINT OVER 1; INK 5-4*(I>2
AND I<5);AT I,Y;A$(3*I-2 TO 3*I)
160 NEXT I
170 LET X=X+1
180 LET Y=Y-1
182 BEEP 0.2,30: PAUSE 30: IF X=Y THEN B
EEP 1,10
185 IF X=Y AND RND<.5 THEN GO TO 60
187 IF INKEY$="" THEN GO TO 125
190 IF (INKEY$="1" AND T) OR (INKEY$="2"
AND NOT T) THEN PRINT FLASH 1;"BRAVO GAU
CHE": STOP
200 IF (INKEY$="0" AND T) OR (INKEY$="9"
AND NOT T) THEN PRINT FLASH 1;"BRAVO DRO
ITE": STOP
210 PRINT "PERDU GAUCHE" AND (INKEY$="2"
OR INKEY$="1");"PERDU DROITE" AND (INKEY
$="9" OR INKEY$="0")

```



BRAVO DROITE

Combat aérien

REFERENCES :

Mode SLOW.

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexe (1 joueur).

Difficulté du jeu : difficile.

LE JEU :

Au centre de l'écran, se trouve le viseur de votre appareil, et l'avion ennemi se déplace devant vous, d'une façon qui pourra vous sembler au début incohérente.

Vous pouvez vous déplacer dans toutes les directions, grâce aux flèches de mouvement, et vous allez essayer d'amener l'appareil ennemi au centre de votre viseur.

Vous tirerez sur lui en frappant Ø, et vous voyez vos projectiles qui convergent sur lui. Mais, il continue à se déplacer, même pendant votre tir, et il n'est pas du tout évident que vous réussirez à l'avoir.

La partie continuera à se dérouler jusqu'à ce que vous l'ayez touché (affichage de GAGNE) ou qu'il soit sorti de votre champ de vision.

LE PROGRAMME :

La ligne 40 dessine l'appareil ennemi dont le déplacement aléatoire sera réalisé par les lignes 50 et 60.

Les lignes 70 et 80 vous permettent le pilotage de votre appareil.

La ligne 100 dessine le viseur de votre appareil.

Le sous-programme, gérant le tir, va de la ligne 210 à la ligne 280 et le test de fin de partie est effectué en ligne 270.

La ligne 110 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 (touches 5, 6, 7, 8) et vous pouvez tirer à tout moment en pressant la touche O.

L'appareil ennemi peut continuer à se déplacer de manière aléatoire, même lorsque vous avez tiré.

La partie se terminera avec l'affichage de GAGNE, si vous touchez l'avion ennemi ou, exceptionnellement, par le fait que l'appareil ennemi sort des limites de l'écran.

On peut aussi arrêter à tout moment en frappant BREAK.

EXTENSIONS POSSIBLES :

Il sera assez facile de faire un affichage au graphisme plus soigné, utilisant la couleur. On pourrait envisager de stabiliser la position de l'avion ennemi pendant le tir (il suffit de modifier la ligne 160).

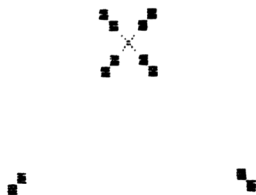
On peut aussi introduire plusieurs avions, compter les points ou le temps, tenir compte de sa propre réserve de carburant, etc.

ZX-81

```

20 LET A=INT PI
30 LET B=A+A
32 LET S=SGN PI+SGN PI
35 LET N=S+S
40 PRINT AT A,B;"■-■"
45 PRINT AT A,B;"X"
50 LET A=A+S*INT (S*RND)-SGN P
I 60 LET B=B+S*INT (S*RND)-SGN P
I 70 LET A=A+(INKEY$="B")-(INKEY$="7")
80 LET B=B+(INKEY$="8")-(INKEY$="5")
100 PRINT AT PI+PI,PI+PI;"X"
AT PI+PI+SGN PI,PI+PI;"X"
*PI+S*PI+PI;"X"
110 IF INKEY$="0" OR N>S+S THEN
GOSUB CODE "00"
120 GOTO CODE "0"
210 PLOT S+N,N
220 PLOT CODE "F"-N,N
230 UNPLOT N,N-S
240 UNPLOT CODE "H"-N,N-S
250 LET N=N+SGN PI
260 IF N<CODE "+" THEN RETURN
270 IF A=11 AND B=10 THEN PRINT
"GAGNE"
280 GOTO CODE "7"

```



ZX-SPECTRUM

```

20 LET A=3
30 LET B=A+A
32 LET S=2
35 LET N=S+S
40 PRINT INK 2;AT A,B;"■-■"
45 LET U=A: LET V=B
50 LET A=A+S*INT (S*RND)-1
60 LET B=B+S*INT (S*RND)-1
70 LET A=A+S*((INKEY$="6")-(INKEY$="7"))
)
80 LET B=B+S*((INKEY$="8")-(INKEY$="5"))
)
84 IF B<0 OR B>29 THEN LET B=ABS (B-29)
86 IF A<1 OR A>20 THEN LET A=ABS (A-20)
90 PRINT AT U,V;" "
100 PRINT INK 1;AT 9,9;"X";AT 10,10;"X"
";AT 11,9;"X"
110 IF INKEY$="0" OR N>S+S THEN GO SUB 2
00
120 GO TO 40
210 PLOT 4*(S+N-3),4*N
220 PLOT 172-4*N,4*N
230 PLOT OVER 1,4*(N-3),4*(N-S)
240 PLOT OVER 1,180-4*N,4*(N-S)
250 LET N=N+1
260 IF N<21 THEN RETURN
270 IF (A=11 OR A=10 OR A=12) AND (B=10
OR B=9 OR B=11) THEN PRINT FLASH 1; PAPER
4;"GAGNE...": BEEP .3,10: BEEP .3,18: BE
EP .3,7: STOP
280 GO TO 35

```



Dessin de dés

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : difficile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : facile.

LE JEU :

Le programme proposé est surtout intéressant par son côté graphique puisqu'il permet de représenter, de façon très convenable, de 1 à 4 dés sur l'écran, en utilisant une technique de dessin un peu particulière.

Chaque réponse à la question NBRE DE DES provoque immédiatement un nouveau lancer du nombre de dés indiqué.

LE PROGRAMME :

La ligne 30 permet de contrôler s'il y a eu ou non action sur une touche de 1 à 4.

Les lignes 50 et 60 initialisent les deux variables de dessin des dés.

La boucle I (70-140) et les lignes 80 et 100 dessinent le cadre du dé.

La ligne 85 réalise le tirage, et la boucle J (90-130) dessine les points en relation avec le tirage réalisé (en appelant les variables A\$ et B\$).

POUR JOUER :

On lance le programme par l'ordre RUN, et, à la question NBRE DE DES, on répond en frappant une touche de 1 à 4. Les dés demandés sont alors représentés, et la question NBRE DE DES est de nouveau posée. Il suffira de taper BREAK pour sortir du programme.

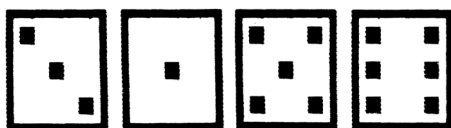
EXTENSIONS POSSIBLES :

On pourra conserver la partie dessin utilisée dans ce programme pour réaliser toutes sortes de jeux de dés, avec une représentation graphique de qualité convenable et facile à modifier si besoin était.

```

5 LET U=8
10 PRINT AT U+U,NOT PI;"NBRE D
E DES"
20 PAUSE 4E4
30 IF INKEY$<"1" OR INKEY$>"4"
THEN RUN
35 CLS
40 LET N=VAL INKEY$
50 LET A$=""
60 LET B$="121314324515525555"
70 FOR I=SGN PI TO N
80 PRINT AT NOT PI,U*I-U;"
";AT PI+PI,U*I-U;"
85 LET X=INT (6*RND+SGN PI)
90 FOR J=SGN PI TO 5
94 IF J<=PI THEN LET S=5*VAL B
$(3*(X-SGN PI)+J)-4
96 IF J<=PI THEN PRINT AT 2*J-
SGN PI,U*I-7;A$(S TO S+4)
100 PRINT AT J,U*I-U;" ";TAB U*
I-2;" "
130 NEXT J
140 NEXT I
150 RUN

```



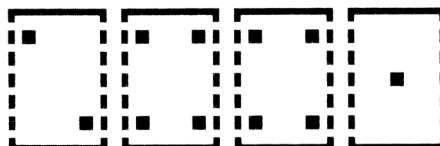
NBRE DE DES

```

5 LET U=8
10 PRINT AT U+U,0;"NOMBRE DE DES"
20 PAUSE 4E4
30 IF INKEY$<"1" OR INKEY$>"4" THEN RUN

35 CLS : BORDER 5
40 LET N=VAL INKEY$
50 LET A$=""
60 LET B$="121314324515525555"
70 FOR I=1 TO N
80 PRINT AT 0,U*I-U;" ";AT 6,U*I-
U;" "
85 LET X=INT (6*RND+1)
90 FOR J=1 TO 5
94 IF J<=3 THEN LET S=5*VAL B$(3*(X-1)+
J)-4
96 IF J<=3 THEN PRINT INK 4;AT 2*J-1,U*
I-7;A$(S TO S+4)
100 PRINT AT J,U*I-U;" ";AT J,U*I-2;" "
130 NEXT J
140 NEXT I
150 BEEP 0.5,12: GO TO 5

```



NOMBRE DE DES

Le jeu du missile

REFERENCES :

Mode FAST.

Difficulté de programmation : moyenne.

Catégorie de jeu : contre le ZX.

Difficulté du jeu : moyenne.

LE JEU :

Le ZX est en alerte : vous êtes aux commandes d'un avion chargé d'une mission d'espionnage sur son territoire ; votre but est de survoler une base ennemie (représenté par un H inversé), et de revenir sain et sauf sur votre base ; mais, dès que le ZX vous aura repéré, il va envoyer à votre poursuite un missile très perfectionné qui n'aura de cesse de vous détruire.

Votre avion peut se déplacer dans toutes les directions, grâce aux touches de 1 à 8 ; réussirez-vous à remplir votre mission ?

LE PROGRAMME :

La ligne 30 dessine la base ennemie à survoler (sous forme d'un H inversé).

La ligne 40 dessine votre base de départ.

Les lignes 55 et 65 choisissent l'instant de départ du missile.

La ligne 70 dessine votre avion.

Les lignes 100 et 120 définissent son mouvement suivant 8 directions, en utilisant les sous-programmes en lignes 1000, 2000, 3000 et 4000.

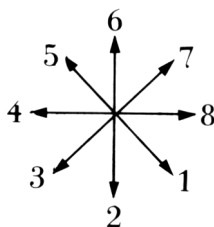
La ligne 125 assure le départ du missile qui est dessiné par la ligne 130 et dont le déplacement est géré par les lignes 135 et 140.

Les lignes 145 à 150 contrôlent la fin de partie.

POUR JOUER :

Lancer le programme par RUN.

Le terrain de jeu s'affiche et vous pouvez déplacer votre appareil en utilisant les touches de 1 à 8, avec la convention suivante :



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

Si le missile vous détruit, vous obtenez un message TOUCHE, et la partie s'arrête.

Taper BREAK pour arrêter la partie.

EXTENSIONS POSSIBLES :

Il sera sans doute nécessaire de rajouter divers contrôles dans le programme (l'avion a-t-il survolé la base ennemie, est-il revenu à sa propre base...).

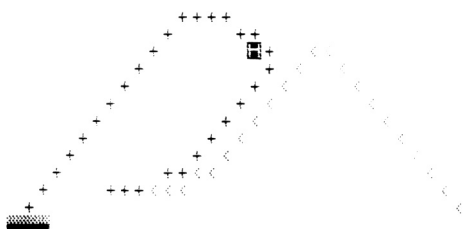
On peut aussi prévoir divers niveaux de jeux en faisant, par exemple, varier la vitesse du missile, prévoir l'utilisation de leurrex par l'avion, l'explosion du missile au bout d'un certain temps, etc.

ZX-81

```

5 CLS
10 LET A=SGN PI+SGN PI
20 LET B=A+INT PI
30 PRINT AT B, CODE "1"
40 PRINT AT CODE "2", NOT PI
50 LET X=CODE "1"
60 LET U=X
70 LET V=CODE "3"
80 LET S=INT (RND*U)
90 LET Y=SGN PI
100 LET S=S-SGN PI
110 PRINT AT X,Y;"+"
120 PAUSE 4E4
130 LET F=A*(VAL INKEY$(B))-SGN PI
140 GOSUB 1000*(VAL INKEY$(A)+(F-SGN PI))
150 IF S THEN GOTO CODE "INKEY$
160 PRINT AT U,V;"<"
170 IF U=X AND V=Y THEN STOP
180 LET U=U-SGN (U-X)
190 LET V=V-SGN (V-Y)
200 IF U=B AND V=CODE "3" THEN
RUN
210 GOTO 70
1000 LET X=X-F
2000 LET Y=Y+F
3010 RETURN
3000 GOSUB 2000
4000 LET X=X+F
4010 RETURN

```



ZX-SPECTRUM

```

5 PAPER 7: CLS
10 LET A=2
15 LET M=0
20 LET B=5
30 PRINT AT B,17;"H"
40 PRINT INK 4;AT 15,0;"███"
47 LET U=15
50 LET X=14
55 LET S=INT (RND*U)
60 LET Y=1
65 LET S=S-1
67 LET V=30
70 PRINT AT X,Y;"^"
90 PAUSE 4E4
100 LET F=A*(VAL INKEY$(B))-1
120 GO SUB 1000*(VAL INKEY$(A)+(F-1))
125 IF S THEN GO TO 65
130 PRINT PAPER 5;AT U,V;"<="
135 LET U=U-SGN (U-X)
140 LET V=V-SGN (V-Y)
145 IF U=X AND V=Y THEN PRINT FLASH 1;"T
DUCE": BEEP 1,0: STOP
150 IF U=B AND V=17 THEN RUN
160 IF X<=B AND Y>=17 THEN LET M=1
170 IF M AND X=15 AND Y<=3 THEN PRINT IN
K 0; FLASH 1;"MISSION ACCOMPLIE": BEEP 1,
10: STOP
200 GO TO 70
1000 LET X=X-F
2000 LET Y=Y+F
2010 RETURN
3000 GO SUB 2000
4000 LET X=X+F
4010 RETURN

```



Le singe

REFERENCES :

Mode SLOW ou FAST.

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 ZX qui va afficher des groupes de 3 à 6 lettres pendant un temps relativement court.

Si l'un des joueurs estime qu'il peut réussir à faire un mot avec les lettres proposées, il doit taper rapidement sur sa touche de jeu (1 ou 0).

Le mot devra alors être accepté ou refusé par l'adversaire qui répondra JUSTE ou FAUX.

Chaque mot juste rapporte 1 point, chaque mot faux enlève 1 point.

LE PROGRAMME :

La ligne 30 contient une liste V\$ des voyelles courantes, et la ligne 40, une liste C\$ des consonnes courantes.

Il sera possible de modifier ces deux listes pour augmenter, par exemple, la fréquence d'apparition de certaines lettres (on pourra ainsi mettre deux E dans V\$ ou deux R dans C\$, etc.).

La ligne 50 définit la longueur du mot (en principe, de 3 à 6 lettres, mais on peut également modifier cette longueur).

La boucle I (80-130) permet la confection d'un "mot" en prenant une lettre, soit dans l'alphabet complet, soit dans la liste C\$ (ligne 100), soit dans la liste V\$ (ligne 110).

La ligne 150 temporise l'apparition du résultat.

La ligne 160 gère la fin de partie, avec affichage du score.

La ligne 170 regarde si l'un des joueurs a réagi.

Enfin, les lignes 220 et 230 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 très peu de temps.

Si l'un des joueurs a frappé 0 ou 1, la question J ou F est alors affichée. On répond J si le mot est correct, F si le mot est incorrect, suivi de NEW-LINE.

Les points sont alors calculés par ZX, 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.

ZX-81

```

10 LET A=NOT PI
20 LET B=A
30 LET V$="AEIOU"
40 LET C$="BCDGLMNPQRSTV"
50 LET X=INT (4*RND+INT PI)
60 LET M$=""
70 PRINT AT 5,NOT PI;" "
80 FOR I=SGN PI TO X
90 LET E$=CHR$ (INT (26*RND+65)
)
100 IF RND<.4 THEN LET E$=C$(IN
T (LEN C$*RND+SGN PI))
110 IF RND<.5 THEN LET E$=V$(IN
T (LEN V$*RND+SGN PI))
120 LET M$=M$+E$
130 NEXT I
140 PRINT AT 5,NOT PI;M$
150 PAUSE 60
160 IF INKEY$="P" THEN PRINT AT
7,NOT PI;"SCORE A :";A;" SCORE
B :";B;J
170 IF INKEY$<>"1" AND INKEY$<>
"0" THEN GOTO 50
180 LET T$=INKEY$
190 PRINT AT 7,NOT PI;"J OU F"
200 INPUT R$
210 PRINT AT 7,NOT PI;" "
220 LET A=A+(R$="J" AND T$="0")
230 LET B=B+(R$="J" AND T$="1")
240 GOTO 50

```

UEJES

J OU F

ZX-SPECTRUM

```

10 LET A=0
20 LET B=A
30 LET V$="AEIOU"
40 LET C$="BCDGLMNPQRSTV"
50 LET X=INT (4*RND+3)
60 LET M$=""
70 PRINT AT 5,0;" "
80 FOR I=1 TO X
90 LET E$=CHR$ INT (26*RND+65)
100 IF RND<.4 THEN LET E$=C$(INT (LEN C$
*RND+1))
110 IF RND<.5 THEN LET E$=V$(INT (LEN V$
*RND+1))
120 LET M$=M$+E$
130 NEXT I
140 PRINT PAPER 5; FLASH 1;AT 5,0;M$
150 PAUSE 60
160 IF INKEY$="F" THEN PRINT FLASH 1;AT
7,0;"SCORE DE A :";A;" SCORE DE B :";B;
BEEP 1,10; STOP
170 IF INKEY$<>"1" AND INKEY$<>"0" THEN
GO TO 50
180 LET T$=INKEY$
190 PRINT AT 7,0; FLASH 1;"(J)"; FLASH 0
;"(F)"; FLASH 1;"(F)"; FLASH 0;"AUX
"
200 INPUT R$
210 PRINT AT 7,0;" "
220 LET A=A+(R$="J" AND T$="0")-(R$="F"
AND T$="0")
230 LET B=B+(R$="J" AND T$="1")-(R$="F"
AND T$="1")
240 GO TO 50

```

IRU

SCORE DE A : 3 SCORE DE B : 2

Le jeu du 15

REFERENCES :

Mode SLOW.

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

C'est un jeu pour deux joueurs, assez ancien, qui était autrefois pratiqué dans les foires. On dispose d'un tapis de jeu de 9 cases, et chacun mise à son tour un jeton de sa couleur, sur une case disponible.

Le premier des deux joueurs qui, avec exactement 3 jetons, arrive à faire un total de 15, a gagné.

LE PROGRAMME :

La boucle I (40-65) dessine le terrain de jeu, et la ligne 66 s'occupe de la gestion du tour des joueurs (numéro 1 ou 2 pour chaque joueur, et jeton O ou X correspondant).

La ligne 90 attend le coup joué (on pourrait utiliser le programme en mode FAST en remplaçant cette ligne par une pause).

La boucle I (100-120) vérifie la validité du coup.

La ligne 122 teste la fin de partie, en cas de match nul, et la ligne 130 refuse les coups non conformes (avec retour à la situation initiale, ligne 66).

Les lignes 140 et 143 enregistrent les coups respectifs de chaque joueur dans des variables alphanumériques O\$ et X\$.

La ligne 145 s'occupe de l'affichage du coup, et la ligne 150 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 157 à la ligne 180 : 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 190.

POUR JOUER :

Lancer le jeu en tapant RUN.

Le terrain de jeu s'affiche, puis la question JR 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 JR 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 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.

ZX-81

```

10 LET F=NOT PI
20 LET R$="123456789"
30 LET O$=""
35 LET X$=""
40 FOR I=F TO CODE "?" STEP IN
T PI
50 PRINT AT INT PI,I," "
60 PRINT AT INT PI+SGN PI,I+SG
N PI,(I+INT PI)/INT PI
65 NEXT I
66 PRINT AT CODE "2",NOT PI;J
R,"F+SGN PI;" " " "0" AND F;"X"
AND NOT F
70 NEXT I
90 PAUSE 4E4
95 IF INKEY$="0" THEN RUN
100 FOR I=SGN PI TO LEN R$
115 IF INKEY$=R$(I) THEN GOSUB
140
120 NEXT I
122 IF NOT VAL R$ THEN PRINT "F
INI";J
130 GOTO CODE "PI"
140 LET O$=O$+(INKEY$ AND F)
143 LET X$=X$+(INKEY$ AND NOT F)
145 PRINT AT INT PI-SGN PI,INT
PI+I-2;"X" AND NOT F;"0" AND F
150 LET R$(I)="0"
157 LET T$=(O$ AND F)+(X$ AND N
OT F)
158 IF LEN T$<INT PI THEN GOTO
190
160 FOR I=SGN PI TO INT PI
162 FOR J=I+SGN PI TO LEN T$
164 FOR K=J+SGN PI TO LEN T$
166 IF VAL T$(I)+VAL T$(J)+VAL
T$(K)=CODE "?" THEN PRINT "BRAVO
";J
170 NEXT K
175 NEXT J
180 NEXT I
190 LET F=ABS (F-SGN PI)
200 RETURN

```

```

      0      X  X  0  0  X
BRAVO 1 2 3 4 5 6 7 8 9

```

JR, 2 : 0

ZX-SPECTRUM

```

10 LET F=0
20 LET R$="123456789"
30 LET O$=""
35 LET X$=""
40 FOR I=F TO 24 STEP 3
50 PRINT INK 5;AT 1,I;" ";AT 2,I;" |
";AT 3,I;" "
60 PRINT AT 4,I+1;(I+3)/3
65 NEXT I
66 PRINT AT 12,0;"JOUER ";F+1;" :0" AN
D F;" :X" AND NOT F
90 IF INKEY$<"1" OR INKEY$>"9" THEN GO
TO 90
100 FOR I=1 TO LEN R$
115 IF INKEY$=R$(I) THEN GO SUB 140
120 NEXT I
122 IF NOT VAL R$ THEN PRINT FLASH 1; IN
K 2;AT 18,0;"C'EST FINI": STOP
130 GO TO 66
140 LET O$=O$+(INKEY$ AND F)
143 LET X$=X$+(INKEY$ AND NOT F)
145 PRINT AT 2,3*I-2;"X" AND NOT F;"0" A
ND F: BEEP .3,10
150 LET R$(I)="0"
157 LET T$=(O$ AND F)+(X$ AND NOT F)
158 IF LEN T$<3 THEN GO TO 190
160 FOR I=1 TO 3
162 FOR J=I+1 TO LEN T$
164 FOR K=J+1 TO LEN T$
166 IF VAL T$(I)+VAL T$(J)+VAL T$(K)=15
THEN PRINT FLASH 1; INK 2;AT 18,0;"BRAVO
, JOUEUR "; "0" AND F;"X" AND NOT F: STOP
170 NEXT K
175 NEXT J
180 NEXT I
190 LET F=ABS (F-1)
200 RETURN

```

```

      0  X  X  X  0  0
      0  X  X  X  0  0
      0  X  X  X  0  0
      0  X  X  X  0  0
      0  X  X  X  0  0
      0  X  X  X  0  0
      0  X  X  X  0  0
      0  X  X  X  0  0
      0  X  X  X  0  0

```

JOUER 2 : 0

BRAVO , JOUEUR 0

Adresse

REFERENCES :

Mode FAST.

Difficulté de programmation : difficile.

Catégorie de jeu : réflexe (pour 1 joueur).

Difficulté du jeu : difficile.

LE JEU :

Des chiffres, entre 0 et 9, vont apparaître au hasard, en divers emplacements de l'écran. Ils restent affichés pendant un temps très court. Le joueur doit réussir à appuyer sur la touche numérique correspondante, au moment où le chiffre est affiché.

Le chiffre tapé par le joueur sera alors affiché en grand format sur la gauche de l'écran ; et chaque réponse exacte rapporte 10 points. La durée de la partie est aléatoire et le score sera affiché en fin de partie.

LE PROGRAMME :

Les lignes 40 et 50 contiennent, dans deux variables alphanumériques C\$ et B\$, les références qui seront utilisées pour dessiner les chiffres en grand format.

L'emplacement du chiffre tiré par le ZX est choisi aux lignes 60 et 70, et le chiffre est tiré en ligne 80.

La ligne 90 réalise l'impression, et la ligne 100 détermine la durée d'affichage. La ligne 120 définit la durée du jeu, et la fin de partie est traitée par la ligne 130.

La boucle I (210-240) réalise le dessin grand format du chiffre frappé par le joueur.

Le contrôle de l'exactitude du résultat est effectué par la ligne 250.

POUR JOUER :

Le programme sera lancé par RUN, puis un premier chiffre apparaît, après une courte pause, vers le milieu de l'écran. Le joueur devra réagir suffisamment vite, tant qu'il reste à l'affichage, et le retaper. Le chiffre qu'il a tapé se dessinera alors en haut à gauche de l'écran, en grand format.

Si le résultat est exact, le joueur marquera 10 points.

Le temps de déroulement de la partie sera fonction des chiffres qui apparaîtront à l'affichage.

En fin de partie, le score sera affiché. Il sera possible de s'arrêter à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :

On peut envisager de modifier le jeu pour faire participer plusieurs joueurs, en modifiant éventuellement l'affichage et en codant les touches du clavier mises à la disposition de chacun des joueurs.

On pourra aussi enlever des points à chaque résultat faux, modifier les règles de calcul du score et la durée du jeu, etc.

```

5 LET M=10
10 LET T=M*M
20 PAUSE T
30 LET S=NOT PI
40 LET C$=""
50 LET B$="1444113333331312111313
14113312131121411133331414141141312"
60 LET X=INT (M*RND)
70 LET Y=INT (M*RND+M)
80 LET N=INT (M*RND)
90 PRINT AT X,Y;N
100 PAUSE INT (40*RND+M+M+M)
110 IF INKEY$((">")) THEN GOTO COD
E "COS "
120 LET T=T-N
125 PRINT AT X,Y;" "
130 IF T>=0 THEN GOTO 80
140 CLS
150 PRINT "SCORE ";S;"%"/100"
160 STOP
200 LET U=VAL INKEY$
210 FOR I=SGN PI TO 5
220 LET R=2*VAL B$(5+U+I)-SGN P
I
230 PRINT AT I,NOT PI)C$(R TO A
+SGN PI)
240 NEXT I
250 IF U=N THEN LET S=S+M
260 GOTO 120

```



```

5 LET M=10
10 LET T=M*M
20 PAUSE T: BORDER 6: CLS
30 LET S=0
40 LET C$="■■■■ ■■"
50 LET B$="1444133333131211313144133121
3112141133331414114131"
60 LET X=INT (M*RND)
70 LET Y=INT (M*RND+M)
80 LET N=INT (M*RND)
90 PRINT AT X,Y:N
100 PAUSE INT (40*RND+M+M+M)
110 IF INKEY$<>" " THEN GO TO 200
120 LET T=T-N
125 PRINT AT X,Y;" "
130 IF T>=0 THEN GO TO 60
140 CLS : BORDER .3,10: BEEP .3,18: BEEP .
3.5
150 PRINT "SCORE ";S;"/100"
160 PAUSE 4E4: RUN
200 LET U=VAL INKEY$
210 FOR I=1 TO 5
220 LET A=2*VAL B$(5*U+I)-1
230 PRINT INK INT (4*RND+1);AT I,0;C$(A
TO A+1)
240 NEXT I
250 IF U=N THEN LET S=S+M: BEEP .1,15
260 GO TO 120

```



Sous-marin

REFERENCES :

Mode SLOW.

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexes (1 seul joueur).

Difficulté du jeu : facile.

LE JEU :

Vous manœuvrez en surface un chasseur de sous-marins, et des sous-marins passent très régulièrement en dessous de votre unité, à différentes profondeurs.

Vous pouvez, à tout moment, larguer des grenades sous-marines et vous pouvez même envisager de toucher plusieurs fois le même sous-marin.

Votre score sera affiché en permanence, sur la gauche de l'écran (rapport nombre de coups au but sur nombre de sous-marins apparus).

LE PROGRAMME :

La ligne 20 permet au ZX de choisir la profondeur du sous-marin.

La ligne 50 dessine votre bâtiment.

La ligne 60 affiche le score et dessine la surface de l'eau.

La ligne 80 dessine le sous-marin.

Le tir est effectué par un programme spécifique placé aux lignes 200 à 225.

Le déplacement (très rapide) de votre navire est réalisé par la ligne 110.

Le test, permettant de savoir si l'on a touché ou non, est effectué en ligne 85.

POUR JOUER :

Taper RUN pour commencer la partie.

Le bâtiment sera déplacé grâce aux flèches droite et gauche (5 et 8).

La touche O autorise le tir.

Le navire, se déplaçant trois fois plus vite que les sous-marins, il sera possible de toucher plusieurs fois le même bâtiment.

Pour terminer la partie, il conviendra de faire BREAK.

EXTENSIONS POSSIBLES :

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

```

6 LET S=NOT PI
12 LET C=CODE "?"
14 LET P=S
15 PRINT AT S,NOT PI:" "
17 LET T=CODE "."
20 LET S=INT (6*RND+PI)
25 LET P=P+.01
35 LET X=SGN PI
50 PRINT AT SGN PI,0:" "
60 PRINT P:" "
80 PRINT AT S,T:" "
82 IF X=SGN PI THEN LET Y=C+X+
X
85 IF S=X AND T+SGN PI=Y THEN
LET P=P+SGN PI
90 LET T=T-SGN PI
100 IF T<NOT PI THEN GOTO CODE
"?"
110 LET C=C+INT PI*(INKEY$="8"
AND C<CODE "+")-INT PI*(INKEY$="
5" AND C>NOT PI)
120 IF INKEY$<>"0" AND X=SGN PI
THEN GOTO CODE "M"
200 PRINT AT X,Y:" "
210 LET X=X+SGN PI
215 IF X>S THEN GOTO CODE "?"
220 PRINT AT X,Y:"0"
225 GOTO CODE "M"

```

3.05

0

```

6 LET S=0
12 LET C=15
14 LET P=S
15 PRINT AT S,0:" "
17 LET T=27
20 LET S=INT (6*RND+4)
25 LET P=P+.01
35 LET X=1
50 PRINT INK 2:AT 1,C:" "
60 PRINT PAPER 5; FLASH 1:AT 16,1;"SCOR
E : ";P; FLASH 0; INK 1:AT 2,0:" "
80 PRINT INK 6:AT S,T:" "
82 IF X=1 THEN LET Y=C+X+X
85 IF S=X AND (T+2=Y OR T+1=Y) THEN LET
P=P+1; PRINT INK 3; FLASH 1:AT S,T;"**"
: BEEP 1,-15
90 LET T=T-1; IF P=99 THEN PRINT INVERS
E 1; INK 4:AT 12,0;"C'EST FINI"; BEEP .3,
10; BEEP .3,18; BEEP .3,5
100 IF T<0 THEN GO TO 15
110 LET C=C+3*((INKEY$="8" AND C<21)-(IN
KEY$="5" AND C>0))
120 IF INKEY$<>"0" AND X=1 THEN GO TO 50
200 PRINT AT X,Y:" "
210 LET X=X+1
215 IF X>S THEN GO TO 35
220 PRINT AT X,Y;"0"
225 GO TO 50

```

SCORE = 11.12

Zombies

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : assez difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez facile.

LE JEU :

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

La partie se termine, soit avec la disparition du joueur, soit par l'élimination de tous les trous noirs.

LE PROGRAMME :

La position des ZOMBIES est définie par un tableau Z, en ligne 10.

Les lignes 40 à 80 (boucle 1) dessinent le cadre, les ZOMBIES et les trous noirs.

La ligne 92 définit la position initiale du joueur.

Une deuxième boucle 1 (95-100) réalise le déplacement des Z.

Les lignes 120 à 130 contrôlent le déplacement du joueur.

POUR JOUER :

Après avoir lancé l'exécution par RUN, le joueur devra se déplacer en utilisant les 4 flèches et en veillant à ne pas sortir des limites du terrain.

Il devra manœuvrer de façon à faire venir les ZOMBIES sur les trous noirs, sans se faire lui-même coïncider.

Pour arrêter la partie, il suffira de taper BREAK.

EXTENSIONS POSSIBLES :

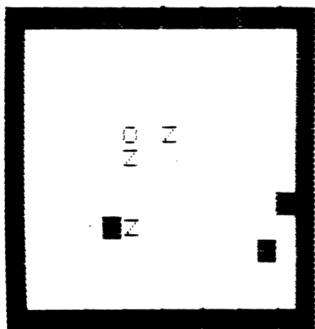
Il serait intéressant d'agrandir le terrain de jeu, d'augmenter le nombre de ZOMBIES ou de trous noirs, d'améliorer les contrôles (pratiquement inexistantes), de placer le joueur de façon plus aléatoire, éventuellement de modifier la règle (par exemple, faire disparaître les ZOMBIES dans les trous noirs).

On pourra aussi, bien sûr, améliorer la qualité du dessin et introduire la couleur.

```

10 DIM Z(INT PI+INT PI)
20 LET L=NOT PI
40 FOR I=L TO CODE "$"
500 PRINT AT L,I+SGN PI;"■"
50 PRINT AT I,L;"■" AT I,CODE
"7"■
65 IF RND<.4 THEN PRINT AT INT
(RND*CODE "$"),I;"■"
67 LET S=RND*VAL "10"
69 LET Z(INT ((I+SGN PI+SGN PI
)/INT PI)+SGN PI)=INT S+INT PI
"5" PRINT AT CODE "$",I+SGN PI;
"■"
80 NEXT I
85 LET L=SGN PI+SGN PI
91 LET K=S
92 PRINT AT S,K;"0"
95 FOR I=SGN PI TO INT PI
96 PRINT AT Z(I),Z(I+INT PI);"
"
97 LET Z(I)=Z(I)-SGN (Z(I)-S)
98 LET Z(I+INT PI)=Z(I+INT PI)
-SGN (Z(I+INT PI)-K)
99 PRINT AT Z(I),Z(I+INT PI);"
Z"
100 NEXT I
110 PAUSE I+I
115 PRINT AT S,K;" "
120 LET K=K+L*(INKEY$="B")-L*(I
NKEY$="S")
130 LET S=S+L*(INKEY$="6")-L*(I
NKEY$="7")
190 GOTO VAL "92"

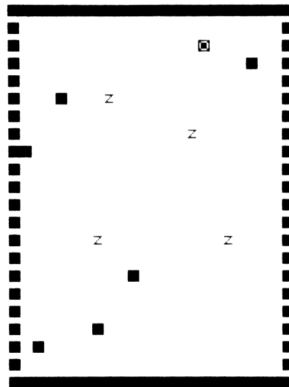
```



```

10 DIM Z(8)
20 LET L=0
40 FOR I=L TO 21
50 PRINT INK 3;AT L,I+1;"■"
60 PRINT INK 3;AT I,L;"■";AT I,23;"■"
65 IF RND<.4 THEN PRINT AT INT (RND*20)
+1,I;"■"
67 LET S=INT (RND*14+1)
69 LET Z(INT (I/3)+1)=5+S
75 PRINT INK 3;AT 21,I+1;"■"
80 NEXT I
85 LET L=2
91 LET K=INT (RND*18+1)
95 FOR I=1 TO 4
96 PRINT AT Z(I),Z(I+4);" "
97 LET Z(I)=Z(I)-SGN (Z(I)-S)
98 LET Z(I+4)=Z(I+4)-SGN (Z(I+4)-K)
99 PRINT INK 2; FLASH 1;AT Z(I),Z(I+4);
"Z": IF Z(I)=S AND Z(I+4)=K THEN BEEP .3,
18: BEEP .3,18: BEEP .3,18: PRINT FLASH 1
;"PERDU": STOP
100 NEXT I
105 PRINT PAPER 5; INVERSE 1;AT S,K;"0"
110 PAUSE I+I
115 PRINT AT S,K;" "
120 LET K=K+L*((INKEY$="B")-(INKEY$="S"))
): IF K<=0 OR K>=23 THEN PRINT "PERDU": B
EEP 3,-10: STOP
130 LET S=S+L*((INKEY$="6")-(INKEY$="7"))
): IF S<=0 OR S>=21 THEN PRINT "PERDU": B
EEP 3,-10: STOP
190 GO TO 92

```



Diago

REFERENCES :

Mode SLOW.

Difficulté de programmation : facile.

Catégorie de jeu : réflexes (1 seul joueur).

Difficulté du jeu : difficile.

LE JEU :

On lance le programme par RUN et on obtient un premier affichage qui vous renseigne sur la performance à réaliser (par exemple, 5 points noirs ou 17 points noirs...), puis le temps est affiché et décompté, à partir de 10 secondes. Le jeu commence lorsque l'on arrive à 0 seconde. Des points noirs 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 :

Le nombre de points noirs sera choisi au hasard, en ligne 10.

Le temps est décompté par les lignes 40 à 90.

Les points sont imprimés par la ligne 120, et la ligne diagonale est fabriquée par la ligne 130.

La pression de la touche 0 est détectée en ligne 150, et la ligne 160 contrôle la fin de partie (aucun réflexe).

Les affichages de fin sont exécutés par les lignes 190 à 230.

POUR JOUER :

On lancera l'exécution par RUN.

On obtient alors l'affichage de la phrase LE BON REFLEXE EST A... qui indique le nombre de points gagnants.

Le temps sera ensuite décompté sous la forme ATTENTION ... SECONDES qui indique le temps restant avant l'apparition de la diagonale. A l'affichage de 0, la diagonale apparaît et commence à se déplacer sur l'écran.

Si l'on estime que le nombre de points réclamés a été atteint, on devra presser la touche 0.

L'affichage indique alors BRAVO si l'on a gagné, ou PERDU dans le cas contraire ; puis la phrase AUTRE ESSAI (O ou 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...).

ZX-81

```
5 CLS
10 LET X=INT (RND*18+INT PI)
20 PRINT "LE BON REFLEXE EST A ";X
"X
30 LET S=10
40 PRINT "ATTENTION (");S;" SECO
ONDES"
50 LET S=S-SGN PI
70 PAUSE 30
75 CLS
80 IF NOT S THEN GOTO 100
90 GOTO 40
100 LET U=NOT PI
110 LET V=U
120 PLOT U,V
130 SCROLL
140 LET U=U+SGN PI
150 IF INKEY$="0" THEN GOTO 180
160 IF U>21 THEN GOTO 220
170 GOTO 120
180 IF U<>X THEN GOTO 220
190 PRINT "BRAVO "
195 PRINT "AUTRE ESSAI (O/N)? "
200 INPUT R$
210 IF R$="O" THEN RUN
215 STOP
220 PRINT "PERDU "
230 GOTO 195
```



BRAVO AUTRE ESSAI (O/N)?

ZX-SPECTRUM

```
5 CLS
10 LET X=INT (RND*18+3)
20 PRINT "LE BON REFLEXE EST A ";X
30 LET S=10
40 PRINT FLASH 1;"ATTENTION (");S;" SECO
NDES"
60 LET S=S-1
70 BEEP 1,10: PAUSE 10
75 CLS
80 IF NOT S THEN GO TO 100
90 GO TO 40
100 LET U=0
110 LET V=U
120 CIRCLE 6*U+5,V+5,2
130 PRINT AT 21,0: POKE 23692,0: PRINT C
HR$ 13
140 LET U=U+1
150 IF INKEY$="0" THEN GO TO 180
160 IF U>22 THEN GO TO 220
170 GO TO 120
180 IF U<>X THEN GO TO 220
190 PRINT "BRAVO;"
195 PRINT " AUTRE ESSAI (O OU N)?"
200 INPUT R$
210 IF R$="O" THEN RUN
215 STOP
220 PRINT "PERDU;"
230 GO TO 195
```



BRAVO: AUTRE ESSAI (O OU N)?

Topologie

REFERENCES :

Mode FAST et SLOW imposé par le programme.

Difficulté de programmation : moyenne.

Catégorie de jeu : graphisme.

Difficulté du jeu : moyenne.

LE JEU :

Un dessin, très élémentaire, à base de traits, apparaît sur l'écran (il peut aussi être entré par un partenaire).

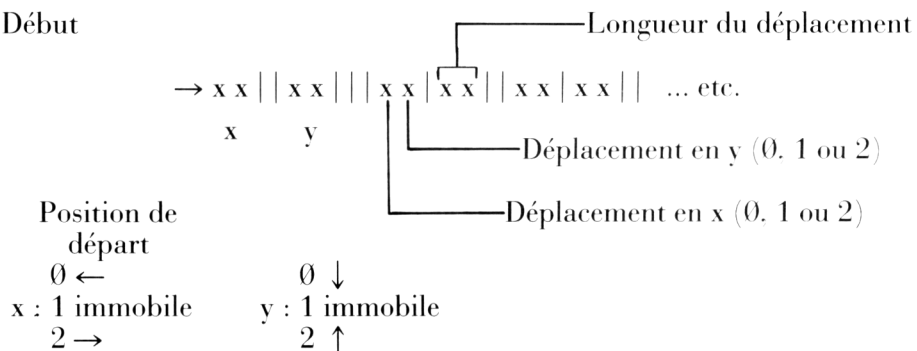
Le joueur dispose d'un plot mobile symbolisant une gomme et devra essayer d'effacer le dessin précédent en repassant, une fois et une seule, sur chacun des traits.

LE PROGRAMME :

La ligne 20 contient un dessin préprogrammé (variable B\$). On pourra, bien entendu, soit définir un autre dessin, soit entrer le dessin en cours de jeu en tapant 20 INPUT B\$.

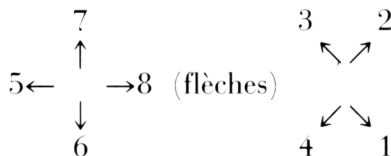
Le codage du dessin est réalisé de la façon suivante :

Début



Les lignes 100 et 110 calculent la position de départ.

La bouclé 1 (130-170) assure le décodage, l'impression étant assurée par le sous-programme 200-230. La partie effacement commence en ligne 173. Le décodage du déplacement du joueur est assuré par les lignes 180 et 182 (8 directions).



POUR JOUER :

On commence avec l'instruction RUN et, si le dessin n'a pas été préprogrammé, on le rentre en suivant les instructions données précédemment.

On tape ensuite NEWLINE.

A l'affichage du dessin, un point lumineux clignotant apparaît en bas et à gauche de l'écran. On peut le déplacer en utilisant les 4 touches de déplacement d'une part, et en utilisant les touches de 1 à 4 pour les déplacements en diagonale d'autre part.

On devra essayer d'effacer le dessin initial, sans lever le crayon et sans repasser deux fois sur le même trait.

EXTENSIONS POSSIBLES :

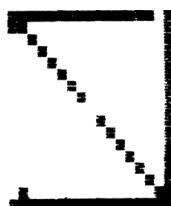
On pourra introduire la couleur, définir une table de dessins topologiquement intéressants, ajouter des contrôles sur le fait que l'on ne quitte pas le dessin initial, etc.

ZX-81

```

10 FAST
20 LET B$="1212211612160115101
6221610160216"
100 LET S=VAL B$(SGN PI TO 2)
110 LET T=VAL B$(INT PI TO 4)
130 FOR I=5 TO LEN B$ STEP 4
140 LET U=VAL B$(I)-SGN PI
150 LET V=VAL B$(I+SGN PI)-SGN
PI
160 GOSUB CODE "COS "
170 NEXT I
171 LET X=NOT PI
172 LET Y=X
173 SLOW
174 PLOT X,Y
175 UNPLOT X,Y
176 IF INKEY$="" THEN GOTO 174
180 LET X=X+(INKEY$="8" OR INKE
Y$="1" OR INKEY$="2")-(INKEY$="5
" OR INKEY$="3" OR INKEY$="4")
182 LET Y=Y+(INKEY$="7" OR INKE
Y$="1" OR INKEY$="4")-(INKEY$="6
" OR INKEY$="2" OR INKEY$="3")
184 GOTO 174
200 FOR K=SGN PI TO VAL B$(I+2
TO I+INT PI)
201 PLOT S,T
202 LET S=S+U
204 LET T=T+V
220 NEXT K
230 RETURN

```

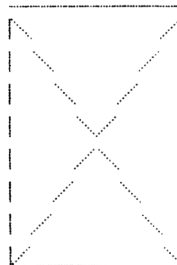


ZX-SPECTRUM

```

20 LET B$="3232216412640164106422641064
0264"
100 LET S=VAL B$(1 TO 2)
110 LET T=VAL B$(3 TO 4)
130 FOR I=5 TO LEN B$ STEP 4
140 LET U=VAL B$(I)-1
150 LET V=VAL B$(I+1)-1
160 GO SUB 200
170 NEXT I
171 LET X=150
172 LET Y=120
174 PLOT X,Y: PLOT X+1,Y: PLOT X,Y+1: PL
OT X+1,Y+1
176 PLOT OVER 1;X,Y: PLOT OVER 1;X+1,Y:
PLOT OVER 1;X,Y+1: PLOT OVER 1;X+1,Y+1
178 IF INKEY$="" THEN GO TO 174
180 LET X=X+(INKEY$="8" OR INKEY$="1" OR
INKEY$="2")-(INKEY$="5" OR INKEY$="3" OR
INKEY$="4")
182 LET Y=Y+(INKEY$="7" OR INKEY$="1" OR
INKEY$="4")-(INKEY$="6" OR INKEY$="2" OR
INKEY$="3")
184 GO TO 174
200 FOR K=1 TO VAL B$(I+2 TO I+3)
201 PLOT S,T
202 LET S=S+U
204 LET T=T+V
220 NEXT K
230 RETURN

```



Chasse au sous-marin

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de réflexion.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

La boucle I (20-40) affiche le terrain.

Les lignes 97 et 98 définissent la position initiale du sous-marin ; les lignes 100 et 105 permettent son déplacement, contrôlé par la ligne 107 (on reste dans les limites du terrain).

La ligne 110 affiche votre position.

Les lignes 117 à 120 contrôlent votre déplacement.

La distance est calculée en ligne 130, affichée en ligne 140 et testée en ligne 150.

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 touches 5, 6, 7, 8.

Après chaque déplacement, la nouvelle distance du sous-marin s'affiche ; à chaque coup le sous-marin se déplace. Si vous réussissez à vous placer à sa verticale, la distance 0 est affichée et il ne bouge plus.

Vous tirez en donnant une profondeur entière (entre 2 et 4, bornes comprises). Si vous le ratez, il se replace n'importe où, et on recommence ; si vous le coulez, vous obtenez l'affichage BRAVO et la partie s'arrête.

On peut arrêter à tout moment le programme en utilisant la touche BREAK.

EXTENSIONS POSSIBLES :

Il est possible de prévoir une amélioration du graphique, d'améliorer les possibilités de déplacement du chasseur (un peu lent, dans cette version), de prévoir des parcours plus sophistiqués pour le sous-marin, de mettre un compteur de temps...

```

10 LET A=NOT PI
15 INPUT H
17 LET G=SGN PI+SGN PI
20 FOR I=A TO H
30 PRINT AT I+G,A;I;AT A,G*I+I
  NT PI;I
40 NEXT I
45 LET I=SGN PI
75 LET X=A
78 LET Y=A
97 LET U=INT (RND*H)
98 LET V=INT (RND*H)
100 LET U=U+I
105 LET V=V+I
107 IF (U<A OR U>H OR V<A OR V>H) THEN LET I=-I
110 PRINT AT X+G,G*Y+SGN PI;" ■ ";AT X+SGN PI,G*Y+INT PI;" "
  AT X+INT PI,G*Y+INT PI;" "
115 PAUSE PI**H
117 LET Y=Y+(INKEY$="8" AND Y<H)-(INKEY$="5" AND Y>A)
120 LET X=X+(INKEY$="6" AND X<H)-(INKEY$="7" AND X>A)
130 LET D=INT SQR (ABS (X-U)**G+ABS (Y-V)**G)
140 PRINT AT H+H,A;"D ";D;" "
150 IF NOT D THEN GOTO CODE "005"
160 IF D<INT PI THEN GOTO VAL "100"
190 GOTO VAL "110"
200 PRINT AT H+H,A;"PROF"
210 INPUT D
220 IF D=INT (PI*RND)+G THEN PR
  INT "BRAVO";D
230 GOTO VAL "95"

```

0 1 2 3 4 5 6 7 8 9

0
1
2
3
4
5
6
7
8
9

D 3

```

10 LET A=0
15 INPUT "DIMENSION :";H
17 LET G=2
20 FOR I=A TO H
30 PRINT PAPER 5;AT I+G,A;I;AT A,G*I+3;
  I
40 NEXT I
45 LET I=1
75 LET X=A
78 LET Y=A
97 LET V=INT (RND*H)
98 LET U=INT (RND*H)
100 LET U=U+I
105 LET V=V+I
107 IF (U<A OR U>H OR V<A OR V>H) THEN LET I=-I
110 PRINT INK 2;AT X+G,G*Y+1;" ■ ";AT X+1,G*Y+3;" ";AT X+3,G*Y+3;" "
115 PAUSE 4E4
117 LET Y=Y+(INKEY$="8" AND Y<H)-(INKEY$="5" AND Y>A)
120 LET X=X+(INKEY$="6" AND X<H)-(INKEY$="7" AND X>A)
130 LET D=INT SQR (ABS (X-V)^G+ABS (Y-U)^G)
140 PRINT AT H+H,A;"DISTANCE :";D;" "
150 IF NOT D THEN GO TO 200
180 IF D<3 THEN BEEP 1,10: GO TO 100
190 GO TO 110
200 PRINT AT H+H,A;"PROFONDEUR :";
210 INPUT D: PRINT D
220 IF D=INT (3*RND)+G THEN PRINT FLASH
  1;"BRAVO": BEEP .3,10: BEEP .3,18: BEEP .
  3,8: STOP
230 GO TO 95

```

0 1 2 3 4 5 6

0
1
2
3
4
5
6

PROFONDEUR : 3
BRAVO

Le lézard d'Alice

REFERENCES :

Mode SLOW.

Difficulté de programmation : facile.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez difficile.

LE JEU :

Il s'inspire très vaguement d'une idée de LEWIS CARROLL, dans "Alice au Pays des Merveilles". Dans l'histoire originale, Alice grandissait ou diminuait, alors que dans le jeu, pour des raisons d'encombrement de la mémoire, c'est un lézard qui change de taille.

Deux verres apparaissent sur l'écran : l'un contient le liquide qui fait grandir, l'autre le liquide qui fait rapetisser (malheureusement ce n'est pas toujours le même verre). Le joueur devra indiquer la quantité de liquide qu'il désire boire et le lézard changera de taille en fonction de cette quantité. Le but du jeu est de parvenir à lui rendre sa taille initiale.

LE PROGRAMME :

La boucle I (5-35) dessine les verres.

Les deux boucles emboîtées I (55-65) et J (57-62) les remplissent avec la quantité indiquée en ligne 50.

La variation de taille du lézard est définie par la ligne 75, avec contrôle par les lignes 80 et 90.

La boucle I (100-300) dessine le corps du lézard, les lignes 500 à 800 dessinent l'œil et les pattes.

Un message gagnant sera affiché par la ligne 810 si l'on a trouvé la bonne taille.

POUR JOUER :

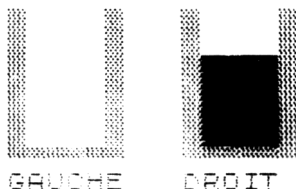
On lance le programme par RUN : les deux verres se dessinent, puis apparaît la question VERRE à laquelle on répond en tapant G ou D ; on obtient alors l'affichage COMBIEN ; on entre alors une quantité (de 0 à 130) et on frappe NEWLINE.

Le verre désigné se remplit sous vos yeux, puis, après quelques instants, le lézard est affiché. On devra alors frapper une touche quelconque (sauf BREAK qui arrêterait la partie) ; si le lézard n'avait pas la bonne taille, aucun message n'est affiché et l'on revient au dessin des verres.

EXTENSIONS POSSIBLES :

Puisqu'il s'agit d'un jeu essentiellement graphique, il sera intéressant d'améliorer les dessins, d'introduire la couleur, de modifier le caractère aléatoire du choix des verres, etc.

Pourquoi ne pas relire Alice au Pays des Merveilles, les idées n'y manquent pas.

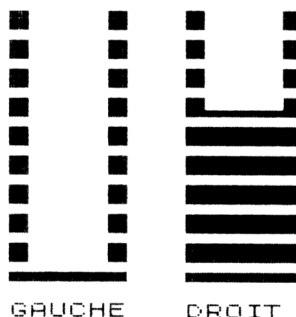
[illegible]

VERRE
COMBIEN

```

2 INK 0: PAPER 7: CLS
3 LET N=18
4 LET S=2
5 FOR I=1 TO 9
10 PRINT INK 4;"      ■      ■      ■      ■"
20 NEXT I
30 PRINT INK 4;"      _____"
35 PRINT "    GAUCHE    DROIT"
38 PRINT AT 15,0:"QUELLE COTE    ":";
40 IF INKEY$<>"D" AND INKEY$<>"G" THEN
GO TO 40
45 LET U=S+18*(INKEY$="D")
47 PRINT "DROIT" AND INKEY$="D";"GAUCHE
" AND INKEY$="G": PRINT "QUELLE QUANTITE
 ":";
50 INPUT Q: PRINT Q
53 INK 5
55 FOR I=1 TO 4*SQR Q
57 FOR J=0 TO 31
60 PLOT 4*U+J+24,I+103
62 NEXT J
65 BEEP 0.2,50: NEXT I
67 PAUSE 4E4
68 BORDER 6: INK 2: CLS
70 LET H=2*(RND<.5)-1
75 LET N=N+H*INT SQR Q
80 IF N<12 THEN LET N=12
90 IF N>60 THEN LET N=60
100 FOR I=0 TO 4*(N+3)
200 PLOT 1,4*S+3: PLOT 1,4*S+2: PLOT 1,4
*S+1: PLOT 1,4*S: BEEP 0.01,20
300 NEXT I
400 LET T=4*(N-INT ((SQR N)/S)): PLOT T,
9*S: PLOT T-1,9*S-1: PLOT T-2,9*S-1: PLOT
T+1,9*S: PLOT T+2,9*S-1: CIRCLE T,8*S-2,
1.5
500 LET T=4*SQR N: PLOT T,3*S: PLOT T-1,
3*S-1: PLOT T-2,3*S-2
600 LET T=T+S: PLOT T,3*S: PLOT T+1,3*S-
1: PLOT T+2,3*S-2
700 LET T=4*(N-SQR N): PLOT T,3*S: PLOT
T-1,3*S-1: PLOT T-2,3*S-2
800 LET T=T+S: PLOT T,3*S: PLOT T+1,3*S-
1: PLOT T+2,3*S-2
805 PAUSE 4E4
810 IF N=28 THEN PRINT "BRAVO": STOP
820 GO TO 2

```



QUELLE COTE : DROIT
QUELLE QUANTITE : 120

Calculatrice

REFERENCES :

Mode FAST ou SLOW.

Difficulté de programmation : très difficile.

Catégorie de jeu : simulation d'une calculatrice.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

Les variables B\$ et C\$, des lignes 20 et 30, contiennent les informations codées qui permettront le dessin des chiffres en grand format.

Les lignes 129 à 150 s'occupent de la gestion des entrées. Les lignes 160 à 180 s'occupent de la représentation, en grand format, des chiffres sur l'écran, au fur et à mesure des entrées.

Le point décimal et le signe moins sont gérés par la ligne 185.

Le nombre total des symboles entrés est contrôlé par la ligne 190.

Le résultat est inscrit grâce à un programme fonctionnant en automatique qui se trouve aux lignes 229 à 270 (l'automatisme est géré par la ligne 195, dans le programme principal).

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 7 symboles, y compris le signe moins et le point décimal). La frappe d'un signe opératoire laisse l'affichage intact, mais l'introduction du prochain chiffre ou symbole provoque l'effacement immédiat et l'affichage du nouveau symbole. Il est à remarquer que les chiffres s'introduisent de gauche à droite, sens de l'écriture usuelle.

L'appareil accepte, comme signes opératoires les symboles +, -, *, / : le signe = provoquera l'affichage du résultat. Les opérations sont enchaînées automatiquement, le résultat précédent étant pris en compte comme opérateur du calcul suivant.

Pour faire une autre opération, il faudra taper BREAK, puis RUN.

Taper BREAK pour sortir du programme.

EXTENSIONS POSSIBLES :

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

ZX-81

```

10 LET U=SGN PI+SGN PI
20 LET C$=" "
30 LET B$="1444133333131211313
1441331213112141133331414114131"
40 LET R$=""
50 LET T=NOT PI
60 LET N=NOT PI
128 PAUSE 4E4
129 IF INKEY$="" THEN GOTO 200
E "FAST"
130 LET R$=R$+INKEY$
131 IF R$="-" THEN GOTO CODE " "
132 IF INKEY$<"." OR INKEY$>"9"
THEN GOTO CODE "W"
133 LET P$=INKEY$
143 IF NOT N THEN CLS
150 IF P$="." OR P$="-" THEN GO
TO CODE " "
160 FOR I=SGN PI TO INT PI+U
165 LET S=U*VAL B$(I+INT PI+U)
*VAL P$)-SGN PI
170 PRINT AT I,N;C$(S TO S+SGN
PI)
180 NEXT I
185 PLOT N+N+INT PI,CODE "4"
190 LET N=N+(U+U)*(N<CODE " ")
195 IF T THEN NEXT T
200 GOTO CODE " "
229 LET N=NOT PI
230 LET R$=STR$ VAL R$
250 FOR T=SGN PI TO LEN R$
260 LET P$=R$(T)
270 GOTO CODE "5"

```

ZX-SPECTRUM

```

10 LET U=2
20 LET C$=" "
30 LET B$="1444133333131211313144133121
3112141133331414114131"
40 LET R$=""
50 LET T=0
60 LET N=0
70 LET H=0
128 PAUSE 4E4
129 IF INKEY$="" THEN GO TO 229
130 LET R$=R$+INKEY$
131 IF R$="-" THEN LET P$=R$: GO TO 185
132 IF INKEY$<"." OR INKEY$>"9" OR INKEY
$="/" THEN GO TO 60
135 LET P$=INKEY$
145 IF NOT N THEN CLS
150 IF P$="." OR P$="-" THEN GO TO 185
160 FOR I=1 TO 5
165 LET S=U*VAL B$(I+5*VAL P$)-1
170 PRINT INK 2-H;AT I,N;C$(S TO S+1)
180 NEXT I
185 IF P$="-" THEN PRINT INK 2-H;AT 3,2;
" "
187 IF P$="." THEN PRINT INK 2-H;AT 3,N;
" "
188 BEEP .3,20
190 LET N=N+4*(N<24)
195 IF T THEN NEXT T
200 GO TO 128
229 LET N=0
230 LET R$=STR$ VAL R$
232 LET H=NOT H
240 BEEP 1,-5
250 FOR T=1 TO LEN R$
260 LET P$=R$(T)
270 GO TO 145

```

456.783

1333.75

Niveau 5

Nom du programme

Thèmes et particularités

Dames	Pour deux joueurs : classique.
Smokey	Fumez sans risque, avec ce programme original pour un joueur.
Course de voitures	Un jeu de réflexe pas trop difficile pour un joueur.
Echecs	Un programme très simplifié, pour deux joueurs.
Danger spatial	Serez-vous assez adroit pour éviter toutes les météorites ?
Tic Tac Toe	Un classique, mais pas sans intérêt (un joueur)
Horloge 2	Utilisation du générateur de caractères incorporé.
Anagrammes	Un bon jeu de réflexes, pour 2 joueurs.
Oeufs	Ne laissez pas faire la pondeuse diabolique ! (1 joueur).
Affichage tournant (vertical)	Une possibilité à exploiter de votre ZX.
Taquin de lettres	Un proverbe est caché dedans ; à vous de le découvrir.
Cibles	Un peu d'adresse, beaucoup de chance. Très visuel.
Gouttes	Plutôt un gag visuel et sonore. A exploiter.
Obstacles	Beaucoup d'obstacles à éviter dans ce jeu d'adresse pour 1 joueur.
Captures dans l'espace	Votre mission serait facile, sans ces damnées météorites !
Affichage publicitaire (2)	Inclinez votre téléviseur de 90°, et tout se passera bien !
Anomalie	Exercez votre perspicacité dans la reconnaissance des symboles.
Plusieurs en un	Un terrain de jeu, et des pions, pour 2 joueurs ! Choisissez !
Parcours du cavalier	Très spectaculaire, et instructif à plus d'un titre.
Mur de briques	Un grand classique, dans une version très complète.
Othello	Pour deux joueurs ; il retournera les pions à votre place.
Quatre à quatre	Un jeu de réflexion un peu difficile, pour 2 joueurs.

Inscription directe dans les mémoires

L'INSTRUCTION D'EXPLORATION DES CASES MEMOIRES : PEEK

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

Pour ce qui nous intéresse, nous avons utilisé le **POKE** sur le ZX-Spectrum, pour inhiber l'affichage **SCROLL ?** qui apparaît automatiquement en bas de l'écran, lorsque celui-ci est plein. L'adresse mémoire correspondante étant 23692, il suffira d'écrire **POKE 23692,0** pour éviter cet affichage intempestif.

Sur le ZX-81, la fonction **PEEK** est surtout utilisée pour retrouver une adresse mémoire dans la zone d'affichage : en effet les cases mémoires 16398 et 16399 contiennent sur 2 octets l'adresse de la prochaine position d'impression (gestion du **PRINT**) ; on utilise cette propriété pour regarder ce qui se trouve à un emplacement désigné de l'écran.

On écrit **PRINT AT X, Y** : ceci prépare le ZX à écrire en ligne X, colonne Y,

On lit alors l'adresse correspondant à cette position par **PEEK 16398+256×PEEK 16399** ; pour savoir ce qui se trouve sur l'écran, il suffit dès lors de faire un **PEEK** de cette adresse : on trouve un nombre de 0 à 255, qui correspond au code du caractère présent sur la mémoire d'écran.

On peut procéder de la même façon sur le ZX-Spectrum, mais il sera plus simple d'utiliser l'une des deux fonctions préprogrammées **SCREEN\$** (lecture d'un caractère sur l'écran) et **ATTR(X,Y)** (lecture des attributs de couleur du point X,Y de l'écran).

On utilisera également la fonction **PEEK** pour lire les informations contenues dans la mémoire morte, par exemple le générateur de caractère, situé en ROM à partir de l'adresse 7680 sur le ZX-81, et 15360 sur le ZX-Spectrum.

Dames

REFERENCES :

Mode SLOW ou FAST.

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 (1K oblige) puisqu'il ne prévoit pas la possibilité de faire des dames. La règle du jeu reste classique, mais les contrôles ont été un peu simplifiés. Les mouvements des pièces sont entrés en une seule manœuvre (position de départ et direction du déplacement) et les prises sont réalisées automatiquement, si elles sont possibles.

Par contre, le changement de joueur étant réalisé à chaque coup, il faudra éventuellement taper Ø pour pouvoir rejouer s'il y avait eu saut ou si le coup n'était pas possible.

LE PROGRAMME :

La boucle I (20-40) dessine le damier (10 cases sur 10 cases) et les pions (X ou O).

La ligne 50 indique qui a le trait.

Le changement de joueur est traité par la ligne 75 (entrée d'un O).

Les lignes 80 et 90 calculent le déplacement, qui sera réalisé lorsqu'il est possible, par la boucle I (120-210).

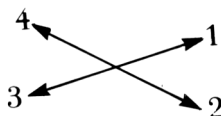
La ligne 130 calcule la case d'arrivée (on tient compte du cas où l'on arriverait en ligne 0, dans la ligne 140).

Les lignes 150 et 160 testent les cases à parcourir. La ligne 180 traite le cas d'un déplacement, et 190 celui d'un saut. La ligne 220 effectue les éventuelles modifications, et la ligne 229 réalise le changement de joueur.

POUR JOUER :

Dès le lancement du programme par RUN, le damier est affiché avec ses pions, et un caractère O inversé apparaît en dessous pour indiquer que le trait appartient au joueur qui a choisi les O.

On joue en indiquant par un nombre de 3 chiffres (pouvant commencer par un Ø) la ligne et la colonne de la pièce à déplacer, puis la direction du déplacement (codée de 1 à 4) suivant la convention :



Par exemple 931 signifie que le O situé en ligne 9, colonne 3, se déplace dans la direction 1 (ne pas oublier de faire NEWLINE).

De même 063 signifiera pion noir, ligne 0, colonne 6, direction 3.

Le changement de joueur étant effectué automatiquement, il faudra taper 0 pour rejouer si le coup est refusé ou s'il y a eu prise (toujours terminer par NEWLINE). Le programme ne contrôle pas l'arrivée à dame, et n'interdit pas les déplacements en arrière.

EXTENSIONS POSSIBLES :

Il sera assez facile de commencer par améliorer les contrôles (changement de joueur, coup en arrière qui ne correspondrait pas à une prise, sortie de terrain). On pourra envisager la création de dames, leur déplacement, et éventuellement écrire un programme permettant à ZX de se mêler avec de bonnes chances de réussite à vos passionnantes parties.

ZX-81

```

10 LET H=SGN PI
20 FOR I=0 TO PI*PI
30 PRINT AT NOT PI,I+H;I;AT I+
H,NOT PI;I;TAB H+((I/(H+H))<>INT
(I/(H+H))) ; "O O O O O"
40 NEXT I
45 LET U=H+H
50 LET A$=("O" AND H)+("X" AND
NOT H)
60 PRINT AT CODE "I",U;A$
70 INPUT R$
75 IF NOT VAL R$ THEN GOTO 000
E "FAST"
80 LET D=VAL R$(INT PI)
90 LET D=INT (PI*PI)*((D=INT P
I)-(D=SGN PI))+INT (PI*PI+U)*((D
=U)-(D=U+U))
100 LET B$=R$
105 LET T=VAL R$( TO U)
110 LET R=U
120 FOR I=SGN PI TO INT PI
130 LET R$=STR$ ABS (T+(I-SGN P
I)*D)
140 IF LEN R$<U THEN LET R$="O"
+R$
150 PRINT AT VAL R$(SGN PI)+SGN
PI,VAL R$(U)+SGN PI;
160 IF R=U THEN LET B$(I)=CHR$
PEEK (PEEK 16398+256*PEEK 16399)
180 IF B$( TO U)=A$+"O" THEN LE
T B$( TO U)="O"+A$
190 IF B$(SGN PI)=A$ AND B$(U)<
>A$ AND B$(INT PI)="O" THEN LET
B$="O"+A$
200 PRINT B$(I)
210 NEXT I
220 LET R=R-SGN PI
225 IF R THEN GOTO 120
229 LET H=NOT H
240 GOTO CODE "M"

```

ZX-SPECTRUM

```

10 LET H=1
20 FOR I=0 TO 9
30 PRINT AT 0,2*I+3;I;AT 2*I+2,0;I; INV
ERSE 1;AT 2*I+2,2+2*((I/2)<>INT (I/2));"
O O O O X X X X"
35 PRINT AT 2*I+3,2;"
" AND I/2=INT (I/2);"
" AND I/2<>INT (I/2) AND I<9
40 NEXT I
42 PRINT AT 1,2;"
T 21,2;"
45 LET U=2
50 LET A$=("O" AND H)+("X" AND NOT H)
60 PRINT AT 12,24;"JOUEUR";AT 14,26;A$;
BEEP .03,10
70 INPUT "VOTRE COUP (LCD) :";R$
75 IF NOT VAL R$ THEN GO TO 229
80 LET D=VAL R$(3)
90 LET D=9*((D=3)-(D=1))+11*((D=U)-(D=U
+U))
100 LET B$=R$
105 LET T=VAL R$( TO U)
110 LET R=U
115 LET S=0
120 FOR I=1 TO 3
130 LET R$=STR$ ABS (T+(I-1)*D)
140 IF LEN R$<U THEN LET R$="O"+R$
150 LET X=2*VAL R$(1)+2: LET Y=2*VAL R$(
U)+3
160 IF R=U THEN LET B$(I)=SCREEN$ (X,Y)
170 IF I=1 AND R=U AND SCREEN$ (X,Y)=" "
THEN LET S=4
180 IF B$( TO U)=A$+" " THEN LET B$( TO
U)=" "+A$: LET S=3
190 IF B$(1)=A$ AND B$(U)<>A$ AND B$(3)=
" " THEN LET B$=" "+A$: LET S=0
200 IF S<4 THEN PRINT INVERSE 1;AT X,Y;B
$(I): LET S=S+1
210 NEXT I
220 LET R=R-1
225 IF R THEN GO TO 115
229 LET H=NOT H
240 GO TO 50

```

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	X	X				
1				0	0		X	X		
2	0	0				X	X			
3	0									
4	0	0				X	X			
5		0			0					X
6	0	0				X	X			
7		0	0			X				
8	0	0			X	X				
9		0	0		X	X				

	0	1	2	3	4	5	6	7	8	9
0	0		0		0		X		X	
1		0						X		X
2	0		0		0		X		X	
3		0						X		X
4	0								X	
5		0		0				X		X
6	0		0				X		X	
7		0		0				X		X
8	0		0						X	
9		0		0		X		X		X

JOUEUR

0

Smokey

REFERENCES :

Mode SLOW.

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

La ligne 30 fait apparaître la fumée au bas de l'écran ; la ligne 35 permet de changer (vers la droite ou vers la gauche) la direction de la colonne, et la ligne 50 fait monter la colonne.

Le joueur apparaît grâce à la ligne 66, et son déplacement est effectué par les lignes 90 et 100.

Les lignes 120 et 190 se chargent de l'affichage du résultat.

POUR JOUER :

On doit lancer l'exécution du programme par RUN.

La fumée commence à s'élever, tandis que le joueur, apparu sous forme d'un carré noir 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 8 et 5). 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.

EXTENSIONS POSSIBLES :

On pourra, tout en conservant l'idée de départ, imposer d'autres règles : par exemple avaler toute la fumée, qui pourra aussi repartir d'un autre endroit si l'on dévie de la colonne initiale.

On pourra aussi améliorer les contrôles de fin de partie, et la qualité du graphisme.

```

10 LET X=CODE "="
15 LET U=NOT PI
20 LET V=CODE "2"
25 LET Y=INT (CODE " ") *RND)
30 PRINT AT X,Y;" "
35 LET Y=Y+(RND<.5 AND Y<CODE
")")-(RND<.5 AND Y>NOT PI)
40 LET T=SGN PI
50 LET X=X-T
60 IF NOT X THEN GOTO CODE "2"
66 PRINT AT U,V;" "
70 IF T THEN PAUSE CODE "PI"
80 PRINT AT U,V;" "
90 LET U=U+(INKEY$="6" AND T A
ND X<CODE "=")-(INKEY$="7" AND T
AND X>NOT PI)
95 LET U=U+NOT T
100 LET V=V+(INKEY$="8" AND V<C
ODE " ")-(INKEY$="5" AND V>NOT P
I)
110 PRINT AT U,V;
120 LET C=PEEK (PEEK 16396+256*
PEEK 16399)
130 IF CHR$ C<>" " THEN GOTO CO
DE "2"
140 LET T=NOT PI
150 IF U<>CODE "=" THEN GOTO CO
DE "PI"
160 CLS
190 PRINT "BRAVO" AND NOT T;"PE
RDU" AND T

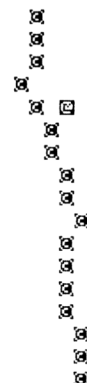
```



```

5 BORDER 6: LET S=0
10 LET X=20
15 LET U=0
20 LET V=12
25 LET Y=INT (17*RND)
30 PRINT INVERSE 1; INK 4;AT X,Y;CHR$ 1
27
35 LET Y=Y+(RND<.5 AND Y<17)-(RND<.5 AN
D Y>0)
40 LET T=1
50 LET X=X-T
60 IF NOT X THEN GO TO 180
66 PRINT FLASH 1; INVERSE 1; INK 2;AT U
,V;CHR$ 64
70 IF T THEN PAUSE 66
80 PRINT AT U,V;" "
90 LET U=U+(INKEY$="6" AND T AND X<20)-
(INKEY$="7" AND T AND X>0)
95 LET U=U+NOT T
100 LET V=V+(INKEY$="8" AND V<17)-(INKEY
$="5" AND V>0)
120 LET C$=SCREEN$ (U,V)
130 IF CODE C$<>127 THEN GO TO 30
140 LET T=0: BEEP .05,0: LET S=S+1
160 IF U<>20 THEN GO TO 66
180 CLS
190 PRINT "BRAVO" AND NOT T;"PERDU" AND
T;" SCORE :"; FLASH 1;S: BEEP .3,10: BEEP
.3,18: BEEP .3,8

```



Course de voitures

REFERENCES :

Mode SLOW.

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de réflexe (1 joueur).

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

L'impression de la route est assurée par la ligne 50.

L'instruction SCROLL, en ligne 55 fait avancer la route, et la ligne 60 imprime la voiture.

La ligne 80 contrôle son mouvement, et la ligne 90 détermine si la route fait ou non des virages.

Les lignes 100 et 110 regardent si l'on a touché le bord de la route.

La ligne 120 incrémente le score, et la ligne 130 efface l'ancienne position de la voiture (une trace claire apparaît à l'emplacement de sa trajectoire).

POUR JOUER :

Lancer l'exécution du programme par l'ordre RUN, le dessin de la route commence à évoluer, et votre véhicule, représenté par un Y inversé, apparaît.

Pour le déplacer vers la droite ou vers la gauche, vous devrez utiliser les touches 5 et 8 (flèches de déplacement du curseur). Il vous faudra éviter le plus longtemps possible de toucher une bordure noire, ce qui provoquera l'arrêt du programme et l'affichage de la mention ACCIDENT, suivie de l'indication sur le score réalisé.

On pourra arrêter le déroulement de la partie à tout moment en pressant la touche BREAK.

EXTENSIONS POSSIBLES :

On peut envisager de jouer à plusieurs, chacun son tour, en enregistrant le meilleur score.

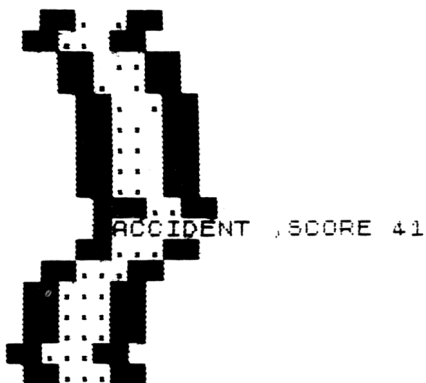
On peut aussi envisager d'élargir la piste, et de faire courir plusieurs voitures.

On pourra aussi s'inspirer des jeux d'arcades qui sont particulièrement nombreux sur ce thème.

```

10 LET A=INT PI*INT PI
20 LET B=A
30 LET X=A+A
40 LET Y=A-INT PI
45 LET S=NOT PI
50 PRINT AT X,Y;"■■■■■■■■■■"
55 SCROLL
60 PRINT AT A,B;"■"
65 LET C=B
80 LET B=B+(INKEY$="8")-(INKEY
$="5")
90 LET Y=Y+(2*RND AND Y<CODE "
")-(2*RND AND Y>INT PI)
100 PRINT AT A+SGN PI,B
110 IF PEEK (PEEK 16398+256*PEE
K 16399)=CODE "■" THEN PRINT "AC
CIDENT SCORE ";S;W
120 LET S=S+SGN PI
130 PRINT AT A,C;" "
140 GOTO CODE "M"

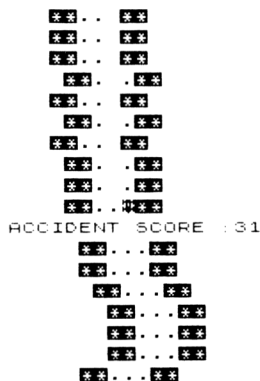
```



```

10 FOR I=0 TO 7: READ A: POKE USR "A"+I
,A: NEXT I
20 DATA 93,127,85,20,93,127,93,8
30 LET A=9: LET B=A: LET X=A+A
40 LET Y=A-3
45 LET S=0
50 PRINT INK 2;AT X,Y;"*****"
55 PRINT AT 20,0: POKE 23692,0: PRINT C
HR$ 13
60 PRINT INK 1;AT A,B;"■": REM GRAPHICS
A
65 LET C=B
80 LET B=B+(INKEY$="8")-(INKEY$="5")
90 LET Y=Y+(2*RND AND Y<16)-(2*RND AND
Y>3)
100 PAUSE 6
110 IF SCREEN$ (A+1,B)="*" THEN BEEP .2,
10: BEEP .2,5: BEEP .2,10: PRINT INK 2; F
LASH 1;"ACCIDENT"; INK 0; FLASH 0;" SCORE
";S;W
120 LET S=S+1
130 PRINT AT A,C;" "
140 GO TO 50

```



Echecs

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : difficile.

Catégorie de jeu : deux joueurs.

Difficulté du jeu : difficile.

LE JEU :

Il n'était pas question, sur 1K-Octet, et en utilisant le Basic, de réaliser un programme permettant au ZX de jouer aux échecs.

Le programme proposé permettra seulement à 2 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 validité des coups ; mais seulement gestion de la position des pièces sur l'échiquier. Les pièces sont représentées par leur initiale, les pions par P, les tours par T, les fous par F, les cavaliers par C, les dames par D et les rois par R. La couleur du fond permet de distinguer pièces noires et pièces blanches.

LE PROGRAMME :

La ligne 10 réalise l'impression de l'échiquier.

La ligne 20 indique le joueur qui a le trait (noir ou blanc).

Les lignes 50 et 60 détectent le symbole présent sur la case de départ.

La ligne 70 vérifie que la case désignée n'est pas vide, et que la pièce manipulée est à la bonne couleur.

La ligne 80 cherche si la pièce à déplacer occupait une case blanche ou une case noire, et la ligne 90 enlève la pièce en réimprimant la couleur de la case.

Les lignes 100 et 120 demandent la case d'arrivée.

La ligne 130 réalise l'impression.

La ligne 150 assure le changement de joueur.

POUR JOUER :

On lance l'exécution par RUN.

L'échiquier est dessiné, et l'on voit apparaître la question JR BLANC DEPART (LC) ; on introduit la ligne, puis la colonne de la pièce à manœuvrer, sous la forme, par exemple : 24 (pions blanc de la ligne 2 et de la colonne 4) suivi de NEWLINE. La pièce disparaît de l'échiquier, et on obtient l'affichage P : 24 (rappel de la pièce déplacée) JR BLANC ARRIV. (LC) ; on tape alors la ligne et la colonne de la case d'arrivée, sous la forme, par exemple : 44 suivi de NEWLINE.

EXTENSIONS POSSIBLES :

On pourra chercher à réaliser un graphisme plus précis, (un exemple possible est d'ailleurs fourni dans ce même ouvrage). On pourra aussi, sans trop de difficultés, introduire le contrôle du mouvement des pièces. Il ne restera alors plus qu'à écrire le programme de jeu du ZX ! Bon courage !

```

5 LET H=SGN PI
7 LET U=H+H
10 PRINT " 12345678" ; "1T C F R D F
CT" ; "2 P P P P P P P P" ; "3 " ; "4
" ; "5 " ; "6 " ; "7 " ; "8 "
20 PRINT AT CODE "2" NOT PI: "J
R " ; "BLANC" AND H: "NOIR" AND NO
T H
30 PRINT "DEPART (LC)"
40 INPUT D$
50 PRINT AT VAL D$(SGN PI) VAL
D$(U)
60 LET C=PEEK (PEEK 16398+256+
PEEK 16399)
70 IF C=CODE " " OR (CHR$ C>="
" AND H) OR (CHR$ C<=" " AND NO
T H) THEN GOTO CODE "="
80 LET S=(VAL D$(SGN PI)+VAL D
$(U))/2:U=INT ((VAL D$(SGN PI)+VAL
D$(U))/2)
90 PRINT " " AND NOT S: " " AND
S
100 PRINT AT PI*PI,NOT PI:CHR$
C: " " ; D$ ; AT CODE "3" NOT PI: "ARR
IV" (LC)
120 INPUT D$
130 PRINT AT VAL D$(SGN PI) VAL
D$(U):CHR$ C: AT PI*PI NOT PI: "
150 LET H=ABS (H-SGN PI)
160 GOTO CODE "="

```

```

      12345678
1T C F R D F
2P P P P P P P
3C " " " " "
4 " P " " "
5 " " F " "
6 " C " " "
7P P P P P P P
8T C F R D F

```

JR NOIR
DEPART (LC)

```

2 PAPER 6: CLS
5 LET H=1
7 LET U=2
10 FOR I=1 TO 8
12 PRINT AT 0,3*I+1;I;AT I+1,0;I
14 PRINT AT I+1,3+3*(I/2=INT (I/2));"
"
16 PRINT AT I+1+1,3+3*(I/2=INT (I/2));"
" ; OVER 1;AT I+I-
1,3+3*(I/2=INT (I/2));" "
18 NEXT I
20 FOR I=1 TO 8
22 PRINT INK 2;AT 2,1+3*I;"T" AND I=1;"
C" AND I=2;"F" AND I=3;"R" AND I=4;"D" AN
D I=5;"F" AND I=6;"C" AND I=7;"T" AND I=8
;AT 4,1+3*I;"P"
23 PRINT INK 1;AT 16,1+3*I;"t" AND I=1;
"c" AND I=2;"f" AND I=3;"r" AND I=4;"d" A
ND I=5;"f" AND I=6;"c" AND I=7;"t" AND I=
8;AT 14,1+3*I;"p"
25 NEXT I
29 PRINT AT 20,0;"JOUER " ; "BLEU " ; AND
NOT H;"ROUGE;" AND H;" "
40 BEEP .03,10: INPUT "DEPART (LC) :";D
$
50 PRINT AT 2*VAL D$(1),3*VAL D$(U)+1;
60 LET C$=SCREEN$ (2*VAL D$(1),3*VAL D$
(U)+1)
70 IF C$=" " OR C$=" " OR (C$>"Z" AND H
) OR (C$<"a" AND NOT H) THEN GO TO 29
80 LET S=(VAL D$(1)+VAL D$(U))/2:U=INT ((
VAL D$(1)+VAL D$(U))/2)
90 PRINT " " AND NOT S: " " AND S
100 PRINT AT 20,13;"JEU " ; "C$;" - ";D$
120 BEEP .03,15: INPUT "ARRIVEE (LC) :";
D$
130 PRINT AT 2*VAL D$(1),3*VAL D$(U)+1;
INK H+1;C$
150 LET H=ABS (H-1)
160 GO TO 29

```

```

      1 2 3 4 5 6 7 8
1 T C F R D F C T
2 P P P P P P P
3 " " " " "
4 " P " " "
5 " " F " "
6 " C " " "
7 P P P P P P P
8 t c f r d f c t

```

JOUER BLEU ;

Danger spatial

REFERENCES :

Mode SLOW.

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de réflexe, pour un joueur.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

La position de chaque météorite est déterminée par la ligne 10 ; elles sont imprimées par la ligne 20 (graphic shift A).

L'avance des météorites est réalisée par deux instructions SCROLL, aux lignes 30 et 40.

Les lignes 45, 46 et 47 vérifient si vous avez ou non heurté une météorite.

La ligne 60 dessine votre vaisseau spatial, et la ligne 70 permet de le déplacer.

Le compteur se trouve en ligne 80.

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 (5 et 8).

On peut arrêter la partie à tout moment en tapant BREAK. Sinon, le jeu s'arrête lorsque vous heurtez une météorite, et le score est alors affiché.

EXTENSIONS POSSIBLES :

Il serait intéressant de rendre le jeu de plus en plus difficile en faisant apparaître de plus en plus de météorites au fur et à mesure du déroulement de la partie.

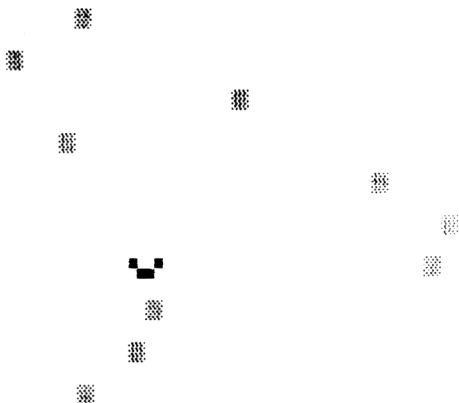
On peut aussi essayer de détruire les météorites en les pulvérisant au fur et à mesure de leur apparition.

ZX-81

```

5 LET X=12
7 LET Z=NOT PI
10 LET H=INT (RND*30)
20 PRINT AT 20,H;"██" AND RND<.
8
30 SCROLL
40 SCROLL
45 PRINT AT 12,X;
46 LET C=PEEK (PEEK 16398+255*
PEEK 16399)
47 IF CHR$ C="██" THEN GOTO 200
50 PRINT AT 10,X-2;" "
60 PRINT AT 12,X;"██"
70 LET X=X+(INKEY$="8" AND X<30
0)-(INKEY$="5" AND X>NOT PI)
80 LET Z=Z+SGN PI
90 GOTO 10
200 CLS
210 PRINT "SCORE ":Z

```

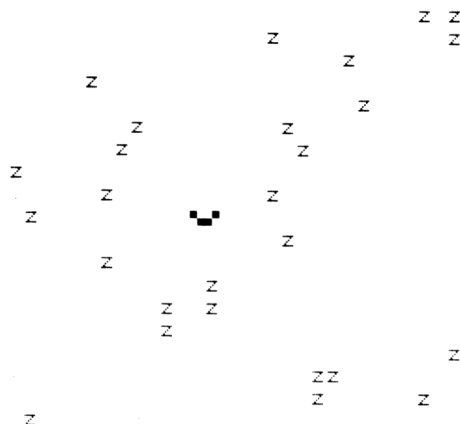


ZX-SPECTRUM

```

5 LET X=12
7 LET Z=0
10 LET H=INT (RND*30)
20 PRINT PAPER 2;AT 20,H;"Z" AND RND<.7
30 PRINT PAPER 6;AT 20,INT (RND*30);"Z"
AND RND<.6
40 POKE 23692,0: PRINT CHR$ 13
45 LET B$=SCREEN$ (12,X+1)
46 LET C$=SCREEN$ (12,X)
47 IF C$="Z" OR B$="Z" THEN GO TO 200
50 PRINT AT 10,X-2;" "
60 PRINT INK 5;AT 11,X;"██"
70 LET X=X+(INKEY$="8" AND X<30)-(INKEY
$="5" AND X>0)
80 LET Z=Z+1
90 GO TO 10
200 CLS
210 PRINT FLASH 1;"SCORE :"; FLASH 0;Z:
BEEP .3,10: BEEP .3,18: BEEP .3,8

```



Tic tac toe

REFERENCES :

Mode SLOW.

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion (deux joueurs).

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

Le terrain de jeu est dessiné à l'aide des boucles I (10-25 et 27-40).

La ligne 60 s'occupe d'afficher le symbole X ou Ø correspondant au joueur qui a le trait.

Un symbole clignotant (ligne 110) est affiché en haut et à gauche du terrain, même si la case est déjà occupée ; les lignes 123 et 125 permettent de déplacer le symbole sur le terrain de jeu.

La ligne 140 vérifie si la case où l'on veut jouer est libre, et si l'on désire y jouer.

La ligne 150 affiche le coup, et la ligne 160 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 5, 6, 7, 8).

Lorsqu'il arrive sur la case où il désire jouer, il suffit qu'il appuie sur la touche Ø, le coup sera refusé si la case n'est pas libre.

Si la case est libre, le symbole qui correspond à ce joueur est inscrit, et on change de joueur.

Lorsque la partie est finie, il convient de taper BREAK pour sortir du programme. On pourra ensuite faire une autre partie avec RUN.

EXTENSIONS POSSIBLES :

On pourrait réaliser après chaque coup un contrôle pour vérifier si l'un des joueurs a gagné. On peut aussi réaliser un jeu machine mais il faudrait éviter qu'il soit par trop systématique (par exemple, la machine jouant toujours son premier coup sur la case centrale).

```

10 FOR I=2 TO 6
20 PRINT AT INT PI,I;" " AT 5,
I;" " AT I,INT PI;" " AT I,5;" "
25 NEXT I
27 FOR I=SGN PI TO 7
30 PRINT AT SGN PI,I;" " AT 7,
I;" " AT I,SGN PI;" " AT I,7;" "
40 NEXT I
50 LET K=NOT PI
60 PRINT AT 12,NOT PI:JOUVEUR
";"0" AND K;"X" AND NOT K
70 LET I=SGN PI
80 LET J=I
90 PRINT AT I+I,J+J;
100 LET C=PEEK (PEEK 16398+256+
PEEK 16399)
110 PRINT "*"
120 PRINT AT I+I,J+J;CHR$ C
123 LET I=I+(INKEY$="8" AND I<I
NT PI)-(INKEY$="7" AND I>SGN PI)
125 LET J=J+(INKEY$="8" AND J<I
NT PI)-(INKEY$="5" AND J>SGN PI)
140 IF CHR$ C<>" " OR INKEY$<>"
0" THEN GOTO 90
150 PRINT AT I+I,J+J;"0" AND K;
"X" AND NOT K
160 LET K=ABS (K-SGN PI)
170 GOTO 60

```

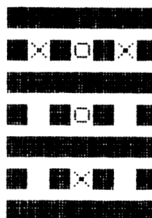


JOUVEUR 0

```

10 FOR I=2 TO 6
20 PRINT INK 5;AT 3,I;" ";AT 5,I;" ";AT
I,3;" ";AT I,5;" "
25 NEXT I
27 FOR I=1 TO 7
30 PRINT INK 1;AT 1,I;" ";AT 7,I;" ";AT
I,1;" ";AT I,7;" "
40 NEXT I
50 LET K=0
60 PRINT PAPER 4; INVERSE 1;AT 12,0;"JO
UEUR ";"0" AND K;"X" AND NOT K
70 LET I=1
80 LET J=1
90 PRINT AT I+I,J+J;
100 LET C=CODE SCREEN$ (I+I,J+J)
110 PRINT "*"
120 PRINT AT I+I,J+J;CHR$ C
123 LET I=I+(INKEY$="6" AND I<3)-(INKEY$
="7" AND I>1)
125 LET J=J+(INKEY$="8" AND J<3)-(INKEY$
="5" AND J>1)
140 IF CHR$ C<>" " OR INKEY$<>"0" THEN G
O TO 90
150 PRINT AT I+I,J+J;"0" AND K;"X" AND N
OT K; BEEP .3,10+8*K
160 LET K=ABS (K-1)
170 GO TO 60

```



JOUVEUR 0

Horloge 2

REFERENCES :

Mode FAST.

Difficulté de programmation : difficile.

Catégorie de jeu : dessin.

Difficulté du jeu : aucune.

LE JEU :

Ce deuxième programme de simulation d'une horloge digitale grand format présente la particularité intéressante d'utiliser le générateur de caractères du ZX.

Ce sont donc les caractères d'origine qui seront reproduits agrandis.

LE PROGRAMME :

La ligne 110 permet l'introduction de l'heure, sous la forme d'un entier de 4 chiffres.

La ligne 130 fait l'appel du générateur, qui dessine les caractères agrandis (sous-programme classique, de la ligne 9000 à la ligne 9150).

Les caractères sont implantés à partir de l'adresse 7680 sur le ZX-81.

La ligne 220 se charge de la remise à zéro au bout de 24 heures.

Les lignes 170, 173 et 175 calibrent le nombre introduit pour réaliser un format d'affichage standard.

Le réglage du temps est fait en ligne 140.

POUR JOUER :

Faire RUN, puis introduire un entier de 4 chiffres (les deux premiers représentent l'heure, les deux suivants les minutes : 0416 signifiera 4 heures 16 minutes).

Attendre le quatrième top pour taper NEWLINE, et l'affichage commence. On pourra essayer de régler l'horloge en jouant sur la durée de la pause, en ligne 140.

EXTENSIONS POSSIBLES :

On peut faire clignoter les deux points, ajouter une possibilité de réveil, prévoir un agenda automatique, faire apparaître un coucou chantant au moment des heures, etc.

```

100 INPUT A$
110 CLS
115 LET U=CODE "2"
120 LET V=CODE "7"
130 GOSUB VAL "9000"
140 PAUSE VAL "2640"
150 LET I=SGN PI+VAL A$(4 TO 7)
155 LET J=VAL A$(1 TO 3)SGN PI
160 IF I=CODE "U" THEN GOSUB CODE "COS"
170 LET A$="0"+STR$(U)
173 LET A$=A$(LEN A$-SGN PI TO )+"0"+STR$(I)
175 LET A$=A$(1 TO INT PI)+A$(LEN A$-SGN PI TO )
180 GOTO VAL "110"
200 LET I=NOT PI
210 LET J=J+SGN PI
220 IF J=CODE "/" THEN LET J=NOT PI
230 RETURN
9000 LET Z$="■"AND4(■)■"
9010 FOR A=SGN PI TO LEN A$
9020 LET X=CODE A$(A)
9030 GOSUB VAL "9070"
9040 LET U=U+8
9050 NEXT A
9060 RETURN
9070 FOR I=U TO U-7 STEP -SGN PI
9080 LET X$=CHR$(PEEK(7680+8*X+U-I))
9090 FOR J=SGN PI TO 8
9100 IF X$(Z$(J)) THEN GOTO 9130
9110 PLOT U+J-SGN PI,I
9120 LET X$=CHR$(CODE X$-CODE Z$(J))
9130 NEXT J
9140 NEXT I
9150 RETURN

```

1536

```

100 INPUT "HEURE (HHMM) :";A$
110 PAPER 6: BORDER 6: CLS
115 LET U=12
120 LET V=26
130 GO SUB 9000
140 PAUSE 2640
150 LET I=1+VAL A$(3 TO 7)
155 LET J=VAL A$(1 TO 2)
160 IF I=60 THEN GO SUB 200
170 LET A$="0"+STR$(J)
173 LET A$=A$(LEN A$-1 TO )+"0"+STR$(I)
175 LET A$=A$(1 TO 2)+A$(LEN A$-1 TO 7)
180 GO TO 110
200 LET I=0
210 LET J=J+1
220 IF J=24 THEN LET J=0
230 RETURN
9000 FOR K=0 TO 3
9010 LET X=15360+8*CODE A$(K+1)
9020 FOR I=0 TO 7
9030 LET Y=PEEK(X+I)
9040 FOR J=0 TO 7
9070 IF Y/2<>INT(Y/2) THEN PRINT AT 7+I, 8*(K+1)-J-2*(K<2); INK 1+(K<2);"■"
9110 LET Y=INT(Y/2)
9120 NEXT J
9130 NEXT I
9140 NEXT K
9150 RETURN

```

20 35

Anagrammes

REFERENCES :

Mode SLOW ou FAST (imposé par le programme).

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 ZX, 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 l'appuyer dès qu'ils voient s'afficher au bas de l'écran un mot déjà présent sur l'écran ; leur score est alors évalué, et un message correspondant à leur résultat sera délivré.

LE PROGRAMME :

Le niveau du jeu est entré à la ligne 20. La boucle I (30-50) réalise le choix des lettres.

Une deuxième boucle I (80-135) effectue le brouillage, et la ligne 100 permet de détecter la pression d'une touche.

La ligne 140 affiche un nouveau mot, et la ligne 155 fait monter d'une ligne sur l'écran les mots déjà obtenus.

La ligne 180 cherche quel est celui des deux joueurs qui a pressé une touche.

La boucle I (190-260) explore les mots présents sur l'écran ligne par ligne, alors que la boucle J (210-240) saisit un par un les caractères de ces mots. On obtient ainsi un nouveau mot qui est comparé au mot initial en ligne 250 et le résultat BRAVO apparaît si c'est exact.

Sinon, la partie s'arrête sur l'affichage PERDU (ligne 270).

POUR JOUER :

On lance le programme par RUN.

On devra ensuite entrer le niveau de difficulté (en principe de 1 à 3) et faire NEWLINE. 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, introduire le son et la couleur, etc.

```

10 PRINT "NIVEAU (1/3)"
15 SLOW
20 INPUT N
25 LET S$=""
30 FOR I=SGN PI TO INT PI+N
40 LET S$=S$+CHR$ INT (35+26*R
ND)
50 NEXT I
60 LET A$=""
70 LET B$=S$
80 FOR I=SGN PI TO LEN S$
90 LET N=INT (LEN S$*RND+SGN P
I)
100 IF INKEY$="0" OR INKEY$="1"
THEN GOTO 170
110 IF B$(N)="-" THEN GOTO 90
120 LET A$=A$+B$(N)
130 LET B$(N)="-"
135 NEXT I
140 PRINT AT 20,NOT PI;A$
150 LET S$=A$
155 SCROLL
160 GOTO 60
170 FAST
180 LET H$=("DROIT" AND INKEY$=
"0")+("GAUCHE" AND INKEY$="1")
190 FOR I=SGN PI TO 13
200 LET B$=""
210 FOR J=SGN PI TO LEN S$
220 PRINT AT I,J-1;
230 LET B$=B$+CHR$ PEEK (PEEK 1
6398+256*PEEK 16399)
240 NEXT J
250 IF B$=S$ THEN PRINT TAB LEN
S$+INT PI;"BRAVO ";H$;J
255 NEXT I
270 PRINT TAB LEN S$+INT PI;"PE
RDU ";H$

```

```

KPZY
YKZP
ZYKP      BRAVO DROIT
YZPK
YPZK
PYZK
YZPK
YKFP
PKZY
ZYKP

```

```

10 PRINT "NIVEAU (1/3)"
20 INPUT N
25 LET S$=""
30 FOR I=1 TO N+3
40 LET S$=S$+CHR$ INT (65+26*RND)
50 NEXT I
60 LET A$=""
70 LET B$=S$
80 FOR I=1 TO LEN S$
90 LET N=INT (LEN S$*RND+1)
100 IF INKEY$="0" OR INKEY$="1" THEN GO
TO 170
110 IF B$(N)="-" THEN GO TO 90
120 LET A$=A$+B$(N)
130 LET B$(N)="-"
135 NEXT I
140 PRINT INVERSE 1; INK INT (5*RND+1);
BRIGHT 1;AT 20,0;A$
145 PAUSE 8*LEN A$
150 LET S$=A$
155 POKE 23692,0: PRINT CHR$ (13)
160 GO TO 60
180 LET H$=("DROIT" AND INKEY$="0")+("GA
UCHE" AND INKEY$="1")
190 FOR I=1 TO 18
200 LET B$=""
210 FOR J=1 TO LEN S$
220 LET R$=SCREEN$ (I,J-1)
230 LET B$=B$+R$
240 NEXT J
250 IF B$=S$ THEN PRINT FLASH 1; INK 2;A
T I,LEN S$+3;"BRAVO ";H$; BEEP .3,10; BEE
P .3,18; BEEP .3,9; STOP
260 NEXT I
270 PRINT TAB LEN S$+3; INVERSE 1;"PERDU
";H$; BEEP .8,-8; BEEP .3,-5; BEEP .5,-1
0

```

```

RXUM      BRAVO GAUCHE
XRUM
URMX
XURM
RUMX
RXMU
MRXU
URXM
MRUX
RXUM
XRUM
MURX
XURM
XUMR
XMUR
RXUM

```


Œufs

REFERENCES :

Mode FAST ou SLOW.

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 + inversé) devez essayer de récupérer. Votre score (nombre d'œufs récupérés) sera affiché en permanence, et la partie se déroule en un laps de temps défini.

LE PROGRAMME :

L'oiseau est dessiné en deux parties (pour le mouvement) aux lignes 50 et 70 ; son déplacement aléatoire est géré par le sous-programme 200-270 (en particulier les lignes 230 à 260).

La ligne 100 permet de reboucler le programme sur la ligne 50.

Le déplacement du joueur est régi par les lignes 160 et 170, et le test permettant de détecter la présence d'un œuf est effectué aux lignes 175, 178 et 181.

La ligne 186 permet l'affichage du score.

La fin de jeu est détectée en ligne 214.

POUR JOUER :

On lancera le programme avec 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 + inversé) 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 BREAK.

EXTENSIONS POSSIBLES :

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

```

10 CLS
15 LET P=CODE "■"
20 LET X=CODE "f"
30 LET Y=X
40 LET R=NOT PI
45 LET S=R
47 LET T=R
50 PRINT AT X,Y;"■"
57 IF INKEY$<>" " THEN GOSUB 15

0 60 GOSUB CODE "C05."
70 PRINT AT X,Y;"■"
90 GOSUB CODE "C05."
100 GOTO CODE "M"
150 PRINT AT R,S;" "
160 LET R=R+(INKEY$="5" AND R<0
ODE "$")-(INKEY$="7" AND R>NOT P
I)
170 LET S=S+(INKEY$="8" AND S<0
ODE "$")-(INKEY$="5" AND S>NOT P
I)
175 PRINT AT R,S)
178 LET A=PEEK 16398+256*PEEK 1
6399
181 LET A=PEEK A=CODE "0"
184 LET T=T+A
186 PRINT AT CODE "(C",CODE "2")

T 187 RETURN
200 LET A=X
210 LET B=Y
212 LET P=P-SGN PI
214 IF NOT P THEN STOP
215 PRINT AT R,S;"■"
220 PAUSE CODE "PI"
225 PRINT AT R,B;" 0"
230 LET X=ABS (X+2*RND-SGN PI)
240 LET Y=ABS (Y+4*RND-SGN PI-S
GN PI)
250 IF X>CODE "f" THEN LET X=SG
N PI
260 IF Y>CODE "=" THEN LET Y=SG
N PI
270 RETURN

```

0

23

```

10 PAPER 6: INK 0: BORDER 1: CLS
15 LET P=200
20 LET X=INT (RND*20)
30 LET Y=INT (RND*20)
40 LET R=0
45 LET S=R
47 LET T=R
50 PRINT INK 5;AT X,Y;"■"
57 IF INKEY$<>" " THEN GO SUB 150
60 GO SUB 200
70 PRINT INK 5;AT X,Y;"■"
90 GO SUB 200
100 GO TO 50
150 PRINT AT R,S;" "
160 LET R=R+(INKEY$="6" AND R<20)-(INKEY$="7" AND R>0)
170 LET S=S+(INKEY$="8" AND S<30)-(INKEY$="5" AND S>0)
178 LET A$=SCREEN$(R,S)
181 LET A=A$+"0"
184 LET T=T+A
186 PRINT AT 21,12;"SCORE :";T
187 RETURN
200 LET A=X
210 LET B=Y
212 LET P=P-1: BEEP 0.1,10
214 IF NOT P THEN BEEP .3,10: BEEP .3,18
: BEEP .3,8: PRINT FLASH 1;"C'EST FINI..."
": STOP
215 PRINT PAPER 2;AT R,S;"#"
220 PAUSE 20
225 PRINT AT A,B;" "; PAPER 4;"0"
230 LET X=ABS (X+2*RND-1)
240 LET Y=ABS (Y+4*RND-2)
250 IF X>19 THEN LET X=1
260 IF Y>29 THEN LET Y=1
270 RETURN

```

#

□ □

SCORE : 19

Affichage tournant (vertical)

REFERENCES :

Mode SLOW.

Difficulté de programmation : difficile.

Catégorie de jeu : affichage publicitaire.

Difficulté du jeu : facile.

LE JEU :

Le programme proposé réalise le défilement d'un texte, affiché en gros caractères, sur l'écran. L'utilisation du générateur de caractères a permis de simplifier considérablement le programme, ainsi que l'utilisation de l'instruction SCROLL, propre au ZX, qui assure à elle seule le défilement.

LE PROGRAMME :

La ligne 20 permet l'introduction du texte, qui sera rangé dans une variable alphanumérique A\$.

La boucle A (30-70) réalise l'affichage lettre après lettre, en appelant le sous-programme 200-310, qui fait appel au générateur de caractères (ligne 210), avec une technique d'impression très simplifiée (boucle T (230-280) pour assurer l'impression de chaque ligne).

Les divers SCROLL (60, 80, 85, 290) permettent le défilement vertical.

POUR JOUER :

On doit d'abord lancer le programme par l'instruction RUN, puis on entre le texte, on tape ensuite NEWLINE, et on regarde.

Le programme boucle sur lui-même et le défilement ne pourra être interrompu que par un appui sur la touche BREAK.

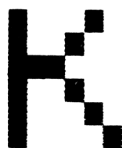
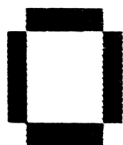
EXTENSIONS POSSIBLES :

Il suffira d'avoir un peu d'imagination (et de disposer de préférence d'une capacité mémoire plus étendue) pour adapter ce programme à toute sorte d'affichage de type publicitaire (il s'agit alors d'attirer l'attention, d'une part, et d'informer, d'autre part). On pourra à cet effet utiliser la couleur, personnaliser les messages, utiliser plusieurs lignes, etc.

```

10 CLS
20 INPUT A$
30 FOR A=SGN PI TO LEN A$
40 LET X=CODE A$(A)
50 GOSUB 200
60 SCROLL
70 NEXT A
80 SCROLL
85 SCROLL
90 GOTO 30
200 FOR Y=NOT PI TO 8
210 LET R=PEEK (7680+8*X+Y)
220 LET S=128
230 FOR T=15 TO 22
240 LET Z=INT (R/S)
250 LET R=R-Z*S
260 LET S=S/2
270 IF Z THEN PRINT AT 20,T;"■"
280 NEXT T
290 SCROLL
300 NEXT Y
310 RETURN

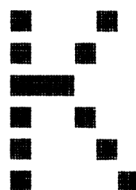
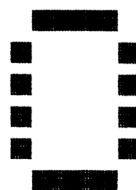
```



```

10 CLS
20 INPUT A$
30 FOR A=1 TO LEN A$
40 LET X=CODE A$(A)
50 GO SUB 200
60 PRINT AT 20,0: POKE 23692,0: PRINT C
HR$ 13
70 NEXT A
80 PRINT AT 20,0: POKE 23692,0: PRINT C
HR$ 13
90 GO TO 30
200 FOR Y=0 TO 8
205 IF Y>=1 AND Y<=6 THEN INK Y
210 LET R=PEEK (PEEK 23606+256*PEEK 2360
7+8*X+Y)
220 LET S=128
230 FOR T=15 TO 22
240 LET Z=INT (R/S)
250 LET R=R-Z*S
260 LET S=S/2
270 IF Z THEN PRINT AT 20,T;"■"
280 NEXT T
290 PRINT AT 20,0: POKE 23692,0: PRINT C
HR$ 13
300 NEXT Y
310 RETURN

```



Taquin de lettres

REFERENCES :

Mode SLOW ou FAST.

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 grise vous permet de déplacer les lettres, et un compteur affiche en permanence le nombre d'essais : arriverez-vous à retrouver le proverbe ?

LE PROGRAMME :

La phrase à reconstituer (qui pourra bien sûr être modifiée) est placée dans une variable B\$ en ligne 25.

Le mélange des lettres et l'impression sont faits par la boucle I (40-120).

La ligne 170 réalise l'affichage du carré gris ; celui-ci est déplacé par les lignes 210 et 220 en utilisant les 4 flèches.

Les lignes 240 et 250 réimpriment la lettre concernée à l'emplacement du carré gris.

La ligne 260 comptabilise le nombre d'essais, qui est imprimé grâce à la ligne 160.

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 gris, à l'aide des 4 touches de direction (5, 6, 7, 8).

Le nombre d'essais est affiché en permanence, à droite du puzzle.

On peut arrêter à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :

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

```

5 LET A$=""
25 LET R$="ONASOUVENTBESOIND.U
NPLUSPETITQUESOI"
35 RAND
40 FOR I=SGN PI TO 35
50 LET X=INT (RND*35+1)
60 IF R$(X)="-" THEN GOTO 50
70 LET A$=A$+R$(X)
80 LET R$(X)="-"
110 PRINT AT INT (I/6),I-6*INT
(I/6);A$(I)
120 NEXT I
130 LET X=NOT PI
140 LET Y=X
150 LET S=X
160 PRINT AT NOT PI,12;S
170 PRINT AT X,Y;"■"
180 PAUSE 4E4
190 LET I=X
200 LET J=Y
210 LET X=X+(INKEY$="6" AND X<5
)-(INKEY$="7" AND X>NOT PI)
220 LET Y=Y+(INKEY$="8" AND Y<5
)-(INKEY$="5" AND Y>NOT PI)
230 PRINT AT X,Y;
240 LET C=PEEK (PEEK 16396+256*
PEEK 16399)
250 PRINT AT I,J;CHR$ C
260 LET S=S+SGN PI
270 GOTO 160

```

```

SUTBTS
SSTONN
.DIELN
UOOUVO
UIAUN
PPIEEE

```

18

```

5 LET A$=""
10 BORDER 6: PAPER 7: CLS
25 LET R$="101N2A3S303U3V3E3N3T4B4E4S40
414N5D5'3U3N1P1L1U1S2P2E2T2I2T3Q3U3E4S404
I"
35 RANDOMIZE
40 FOR I=1 TO 35
50 LET X=INT (RND*35+1)
60 IF R$(2*X-1 TO 2*X)="--" THEN GO TO
50
70 LET A$=A$+R$(2*X-1 TO 2*X)
80 LET R$(2*X-1 TO 2*X)="--"
110 PRINT INVERSE 1: BRIGHT 1: INK VAL A
$(2*I-1);AT 2+INT (I/6),2+I-6*INT (I/6);A
$(2*I)
120 NEXT I
130 LET X=0
140 LET Y=X
150 LET S=X
160 PRINT INVERSE 1;AT 0,12;"NOMBRE D'ES
SAIS :";S
170 PRINT FLASH 1;AT X+2,Y+2;"■"
180 PAUSE 4E4
190 LET I=X
200 LET J=Y
210 LET X=X+(INKEY$="6" AND X<5)-(INKEY$
="7" AND X>0)
220 LET Y=Y+(INKEY$="8" AND Y<5)-(INKEY$
="5" AND Y>0)
240 LET C$=SCREEN$(X+2,Y+2): LET B=ATTR
(X+2,Y+2)
245 IF B>125 OR B<121 THEN LET B=127: BE
EP .3,18
250 PRINT INVERSE 1: BRIGHT 1: INK (B-12
0);AT I+2,J+2;C$: BEEP .03,8
260 LET S=S+1
270 GO TO 160

```

NOMBRE D'ESSAIS : 10

```

10E15D
11S 05
50 TEN
0NU1AN
EUNUTE
PLABOU

```

Cibles

REFERENCES :

Mode SLOW.

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 3) sera choisi en début de partie, et le score réalisé sera affiché en permanence. Si l'on touche le bord d'une cible, la partie s'arrête. On marquera d'autant plus de points que le nombre sur lequel on tombe sera plus grand.

LE PROGRAMME :

Le nombre de cibles est entré en ligne 5. La boucle I (30-50) effectue l'impression des cibles.

La ligne 75 affichera le score en permanence.

La boule est affichée par la ligne 80.

La boucle I (100-140) réalise l'impression des chiffres contenus dans les cibles, et permet le déplacement de la boule (ligne 124).

La ligne 150 regarde si l'on a tiré, et la ligne 160 regarde si la boule est encore sur la ligne de départ.

La ligne 190 fait progresser la boule d'une case vers la droite, et la ligne 200 vérifie si l'on a ou non atteint les cibles ; puis les lignes 210 et 220 regardent ce que l'on a touché.

La ligne 240 incrémente le score ; si l'on désire que le nombre de points marqués corresponde au chiffre affiché, il faudra écrire : 240 LET S = S + VAL CHR\$ C.

Les lignes 255 et 260 s'occupent de la fin de partie.

POUR JOUER :

Lancer le programme par RUN, et taper le nombre de cibles désirées suivi de NEWLINE. 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 6 et 7), et vous pouvez tirer à tout moment en tapant Ø.

Le score sera modifié en fonction de votre résultat, et une nouvelle balle apparaît. Il est possible d'arrêter à tout moment en tapant BREAK, sinon le jeu s'arrête au bout d'un certain nombre de tirs.

EXTENSIONS POSSIBLES :

On pourra sans difficulté augmenter le nombre de cibles, modifier la règle d'apparition des nombres dans les cibles pour la rendre moins hasardeuse, s'arranger pour que la balle elle-même soit représentée par un chiffre qui devrait coïncider à l'arrivée avec celui de la cible.

```

5 INPUT N
7 LET P=CODE "■"
8 LET J=P+P
10 LET S=NOT PI
30 FOR I=SGN PI TO N
40 PRINT AT INT PI*I-PI,J;" "
"AT INT PI*I-SGN PI-SGN PI,J;"
"AT INT PI*I-SGN PI,J;"
50 NEXT I
60 LET K=NOT PI
70 LET A=J
75 PRINT AT A,N;"SCORE ";S
80 PRINT AT A,K;"0"
90 LET D=A
100 FOR I=SGN PI TO N
105 LET U=INT (P*RND)
110 PRINT AT I+INT PI-SGN PI-SG
N PI,U+SGN PI,U
124 LET A=A+(INKEY$="5")-(INKEY
$="7")
140 NEXT I
145 PRINT AT D,K;" ";
150 IF INKEY$="0" THEN GOTO VAL
"190"
150 IF NOT K THEN GOTO VAL "30"
190 LET K=K+SGN PI
200 IF K=J THEN GOTO VAL "80"
210 LET C=PEEK (PEEK 16398+256*
PEEK 16399)
220 IF C=CODE "■" OR C=CODE "■"
THEN PRINT "PERDU";W
240 LET S=S+VAL CHR$ C
255 LET P=P-SGN PI
260 IF NOT P THEN STOP
270 GOTO CODE "U"

```

SCORE 8

1
2
3

```

5 INPUT "NOMBRE DE CIBLES :";N
7 LET P=10
8 LET J=P+P
10 LET S=0: LET B=1
30 FOR I=1 TO N
40 PRINT PAPER 4;AT 4*I-4,J;" ■";AT 4*
I-3,J;" ■";AT 4*I-2,J;" ■";AT 4*I-1,J;"
■";AT 4*I,J;" ■"
50 NEXT I
60 LET K=0
70 LET A=J
75 PRINT INVERSE 1; INK 1;AT A,N;"SCORE
:";S; INK 3;" BALLES ";B
80 PRINT AT A,K;"0"
90 LET D=A
100 FOR I=1 TO N
105 LET U=INT (P*RND)
110 PRINT INVERSE 1; INK 2;AT 4*I-2,J+1;
U
124 LET A=A+(INKEY$="6")-(INKEY$="7")
140 NEXT I
145 PRINT AT D,K;" ";
150 IF INKEY$="0" THEN BEEP .03,10: LET
B=B+1: GO TO 190
160 IF NOT K THEN GO TO 80
190 LET K=K+1
200 IF K=J THEN GO TO 80
210 LET C$=SCREEN$ (A,K)
220 IF C$="■" OR K>J+1 THEN PRINT "PERDU
";W
240 LET S=S+VAL C$: BEEP .3,10: BEEP .3,
18: BEEP .3,8
255 LET P=P-1
260 IF NOT P THEN STOP
270 GO TO 60

```

SCORE : 11 BALLES 8

1
2
3
4
5
6
7
8
9

Gouttes

REFERENCES :

Mode FAST ou SLOW.

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 carré gris, apparaît au-dessus du récipient et tombe en remplissant le récipient au fur et à mesure. Sa trajectoire, lorsqu'elle rencontre un obstacle, est entièrement déterminée par le hasard : elle part à droite ou à gauche, jusqu'à ce qu'il n'y ait plus aucune possibilité. Alors, une autre goutte tombe, et ainsi de suite, jusqu'au débordement final.

LE PROGRAMME :

Les deux boucles emboîtées I (10-50) et J (20-40) dessinent le récipient et les obstacles.

La ligne 100 dessine la goutte. La ligne 120 efface la goutte précédente et la ligne 140 envoie dans un sous-programme (228-230) qui permet de déterminer s'il y a ou non rencontre avec un obstacle.

Si c'est non, on va en 212, et la goutte continue de tomber.

Si c'est oui, la goutte part à droite ou à gauche (ligne 160) selon la valeur définie pour K en ligne 90.

On regarde de nouveau s'il y a ou non obstacle, ou si la goutte peut continuer à tomber (lignes 170, 180, 190).

Selon le résultat du test, on utilisera 200, 212, 218 ou 223.

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 aussi l'arrêter à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :

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

```

10 FOR I=CODE ":" TO CODE "="
20 FOR J=NOT PI TO CODE "("
30 PRINT AT CODE "$",NOT PI;"■"
)TAB CODE "(";"■"AT I,J;"■"
35 PRINT AT I,SGN PI+(RND*CODE
"$");" " AND I<>CODE "=" AND R
ND<.8
40 NEXT J
50 NEXT I
60 LET Y=INT PI*INT PI
70 LET X=Y
90 LET K=2*(RND<.5)-SGN PI
95 LET I=NOT PI
100 PRINT AT X,Y;"■"
110 PAUSE CODE "PI"
120 PRINT AT X,Y;" "
130 PRINT TAB Y;
140 GOSUB CODE "SLOW"
150 IF NOT J THEN GOTO CODE "US
R"
160 PRINT AT X,Y+K;
170 GOSUB CODE "SLOW"
180 IF NOT J THEN GOTO CODE "A
ND"
190 IF NOT I THEN GOTO CODE "T
O"
200 PRINT AT X,Y;"■"
210 GOTO CODE "U"
212 LET X=X+SGN PI
213 GOTO 90
218 LET Y=Y+K
219 LET I=SGN PI
220 GOTO 100
223 LET K=-K
224 LET I=SGN PI
225 GOTO CODE "■"
228 LET J=PEEK .16398+256*PEEK 1
6399
229 LET J=PEEK J=CODE "■" OR PE
EK J=CODE "■"
230 RETURN

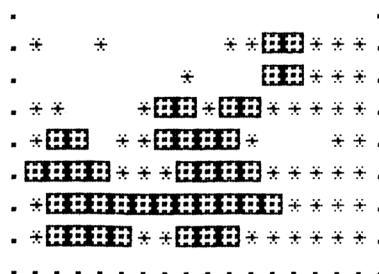
```



```

10 FOR I=12 TO 20
20 FOR J=1 TO 16
30 PRINT PAPER 1;AT I,0;" ";AT I,17;" "
;AT 20,J;" "; PAPER 4;AT I,J;"*" AND I<>1
2 AND I<>20
35 PRINT AT I,1+(RND*13);" " AND I<>20
AND RND<.8
40 NEXT J
50 NEXT I
60 LET Y=INT (7*RND+6)
70 LET X=6
90 LET K=2*(RND<.5)-1
95 LET I=0
100 PRINT INVERSE 1; INK 2;AT X,Y;"#"
110 PAUSE 10: BEEP .02,0
120 PRINT AT X,Y;" "
130 LET U=X+1: LET V=Y
140 GO SUB 228
150 IF NOT J THEN GO TO 212
160 LET U=X: LET V=Y+K
170 GO SUB 228
180 IF NOT J THEN GO TO 218
190 IF NOT I THEN GO TO 223
200 PRINT INVERSE 1; INK 2;AT X,Y;"#"
210 GO TO 60
212 LET X=X+1
213 GO TO 90
218 LET Y=Y+K
219 LET I=1
220 GO TO 100
223 LET K=-K
224 LET I=1
225 GO TO 160
228 LET C#=SCREEN$ (U,V)
229 LET J=C#<>" "
230 RETURN

```



Obstacles

REFERENCES :

Mode SLOW.

Difficulté de programmation : assez difficile.

Catégorie de jeu : jeu de réflexion (pour un joueur).

Difficulté du jeu : moyenne.

LE JEU :

Des piétons se déplacent sur l'écran pour venir à la rencontre d'une petite voiture, symbolisée par un Y inversé, qui dispose de la possibilité de se déplacer vers la gauche ou vers la droite pour les éviter et qui devra tenir le plus longtemps possible.

La distance parcourue sera affichée en permanence, et le jeu se termine lorsque la voiture rencontre, soit le bord du terrain, soit un obstacle.

LE PROGRAMME :

La ligne 30 (instruction SCROLL) fait avancer les piétons vers la voiture du joueur qui est imprimée grâce à la ligne 40.

Les lignes 60, 70, et 80 fabriquent la route (les lignes 70 et 80 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 90 affiche la route et le score.

La ligne 100 permettra de déplacer le véhicule.

Les lignes 110, 120 et 130 permettent de contrôler s'il y a ou non un obstacle et arrêtent le programme s'il y a lieu.

Le score est incrémenté par la ligne 150.

POUR JOUER :

On lance le programme en tapant RUN.

La route et les piétons apparaissent en bas de l'écran et montent vers la voiture représentée par un Y inversé. La voiture peut se déplacer à l'aide des flèches de déplacement horizontal (touches 5 et 8).

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 la touche BREAK.

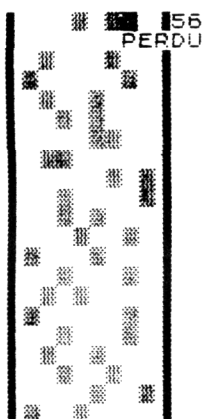
EXTENSIONS POSSIBLES :

Il sera assez facile, notamment si l'on dispose du ZX-Spectrum, d'améliorer le dessin de la voiture, de la route, des obstacles. On pourra aussi introduire la couleur, admettre un certain nombre d'accidents, enregistrer le meilleur score (pour plusieurs joueurs), éventuellement dessiner plusieurs voitures, etc.

```

10 LET Y=5
20 LET S=NOT PI
30 SCROLL
40 PRINT AT NOT PI,Y;"■"
50 LET A$=""
60 LET A$(7*RND+2)="■"
70 LET A$(7*RND+2)="■"
80 LET A$(7*RND+2)="■"
90 PRINT AT 20,NOT PI:A$;AT NO
T PI,PI*PI;S
100 LET Y=Y+(INKEY$="8")-(INKEY
$="5")
110 PRINT AT SGN PI,Y;
120 LET C=PEEK (PEEK 16398+256*
PEEK 16399)
130 IF (C<>NOT PI) AND (C<>118)
THEN PRINT "PERDU";0
150 LET S=S+SGN PI
160 GOTO 30

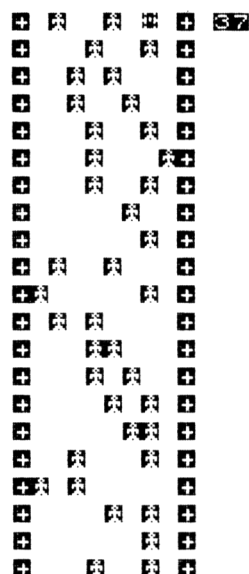
```



```

5 FOR I=0 TO 15: READ A: POKE USR "A"+
I,A: NEXT I
10 DATA 93,127,85,20,93,127,93,0,8,20,8
,62,42,8,28,54
20 PAPER 6: CLS : LET Y=5: LET S=0
30 PRINT AT 20,0: POKE 23692,0: PRINT C
HR$ 13
40 PRINT INK 2;AT 0,Y;"■": REM GRAPHICS
A
60 LET A$="+■■■■■+"
70 LET A$(7*RND+2)="■": REM GRAPHICS B
80 LET A$(7*RND+2)="■": REM GRAPHICS B
90 PRINT INK 1: INVERSE 1;AT 20,0:A$;AT
0,11;S
100 LET Y=Y+(INKEY$="8")-(INKEY$="5")
120 LET C$=SCREEN$ (1,Y): LET D=POINT (B
*Y+1,164)
130 IF D=1 OR C$="+" THEN STOP
150 LET S=S+1
160 GO TO 30

```



Captures dans l'espace

REFERENCES :

Mode SLOW.

Difficulté de programmation : difficile.

Catégorie de jeu : réflexe (1 seul joueur).

Difficulté du jeu : moyenne.

LE JEU :

Votre vaisseau spatial navigue dans un dangereux champ de météorites et votre mission est de vous emparer de la plus grande quantité possible de minéral d'urane, un métal indispensable que l'on ne peut trouver que dans ces lieux particulièrement dangereux.

Vous devez arriver à manœuvrer pour recueillir l'urane, sans vous faire heurter par une météorite.

LE PROGRAMME :

La ligne 20 permet de faire apparaître les météorites.

La ligne 30 fera apparaître le minéral d'urane.

Leur déplacement sur l'écran est assuré par des instructions SCROLL, aux lignes 30 et 40.

Les lignes 45 à 48 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 60 et son déplacement est contrôlé par la ligne 70.

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 (5 et 8) seront utilisées pour déplacer votre vaisseau et récupérer l'urane, mais le moindre heurt avec une météorite provoquera la fin de la partie et l'affichage de la quantité d'urane capturée.

On pourra aussi arrêter à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :

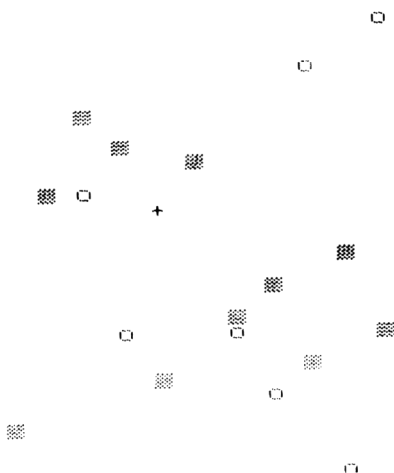
On peut rendre le jeu de plus en plus difficile en augmentant progressivement le nombre de météorites, ou encore, en minuant la durée de la partie.

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...

```

5 LET X=12
7 LET Z=NOT PI
20 PRINT AT 20,INT (RND*30);"█"
"
25 PRINT AT 20,INT (RND*30);"O"
" AND RND<.5
30 SCROLL
40 SCROLL
45 PRINT AT 12,X;
46 LET C=PEEK (PEEK 16398+255*
PEEK 16399)
47 IF CHR$ C="█" THEN GOTO 200
48 IF CHR$ C="O" THEN LET Z=Z+
SGN PI
50 PRINT AT 10,X-2;" "
60 PRINT AT 12,X;"+"
70 LET X=X+(INKEY$="B" AND X<0
0)-(INKEY$="5" AND X>NOT PI)
90 GOTO 10
200 CLS
210 PRINT "SCORE ";Z

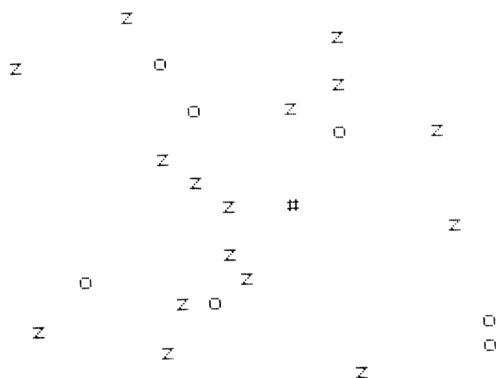
```



```

5 LET X=12
7 LET Z=0
20 PRINT PAPER 3;AT 20,INT (RND*30);"Z"
25 PRINT INK 4;AT 20,INT (RND*30);"O" A
ND RND<.5
30 POKE 23692,0: PRINT CHR$ 13
40 PAUSE 2
46 LET C$=SCREEN$ (12,X)
47 IF C$="Z" THEN GO TO 200
48 IF C$="O" THEN LET Z=Z+1: BEEP .2,18
50 PRINT AT 11,X-1;" "
60 PRINT FLASH 1; PAPER 5;AT 12,X;"#"
70 LET X=X+(INKEY$="B" AND X<30)-(INKEY
$="5" AND X>0)
90 GO TO 10
200 CLS
210 PRINT FLASH 1;"SCORE :"; FLASH 0;Z;
BEEP .3,10: BEEP .3,18: BEEP .3,8

```



Affichage publicitaire

(2)

REFERENCES :

Mode SLOW.

Difficulté de programmation : difficile.

Catégorie de jeu : graphique.

Difficulté du jeu : aucune.

LE JEU :

Ce programme va permettre l'affichage tournant d'un texte en grand format ; il utilise les instructions SCROLL et le générateur de caractères, mais possède la particularité de faire défiler les caractères en les couchant à l'horizontale, ce qui implique d'incliner l'écran de visualisation dans le sens de la hauteur pour obtenir un affichage géant avec défilement horizontal de la droite vers la gauche.

LE PROGRAMME :

La ligne 20 permettra l'introduction du texte.

La boucle A (30-70) réalise l'affichage lettre par lettre.

Le sous-programme, qui va de la ligne 200 à la ligne 320, affiche chaque lettre en grand, en appelant le générateur de caractère (ligne 210) et en réalisant le décodage de chaque caractère verticalement et non horizontalement (boucle I, de 200 à 220).

Une autre boucle I (250-280) réalise l'affichage.

POUR JOUER :

On devra d'abord lancer le programme par RUN, puis on entrera le texte à afficher, suivi de NEWLINE.

Le texte commence à défiler sur l'écran d'une façon curieuse, et il vous faudra incliner le téléviseur sur un côté pour arriver à le lire.

La largeur des caractères dessinés ne permet pas d'en avoir plus de 3 simultanément sur l'écran.

On peut arrêter le programme à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :

Il sera possible de reprendre cette idée (sans nécessairement incliner le téléviseur !) pour réaliser toute sorte d'affichages fantaisie (couleurs, caractères spéciaux, etc.).

On pourra aussi, sans trop toucher au programme, modifier la ligne 160 de telle sorte que l'affichage d'un P grand format soit réalisé avec des P, et de même pour les autres lettres...

```

10 CLS
15 DIM N(INT.PI+INT PI)
20 INPUT A$
30 FOR A=SGN PI TO LEN A$
40 LET X=CODE A$(A)
50 GOSUB CODE "COS "
60 SCROLL
70 NEXT A
80 SCROLL
85 SCROLL
90 GOTO CODE "2"
200 FOR I=SGN PI TO INT PI+INT
PI
210 LET N(7-I)=PEEK (7680+8*X+I
)
220 NEXT I
230 LET S=CODE "RND"
240 LET Y=INT PI
250 FOR I=SGN PI TO INT PI+INT
PI
260 IF N(I)-S>=NOT PI THEN PRIN
T AT CODE "=",Y;"■"
265 LET N(I)=N(I)-(S AND N(I)-S
)>=NOT PI)
270 LET Y=Y+SGN PI
280 NEXT I
290 LET S=S/2
300 SCROLL
310 IF S>SGN PI THEN GOTO 240
320 RETURN

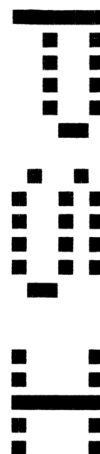
```



```

10 PAPER 6: CLS
15 DIM N(6)
20 INPUT "VOTRE TEXTE :";A$
30 FOR A=1 TO LEN A$
40 LET X=CODE A$(A)
50 GO SUB 200
60 PRINT AT 20,0: POKE 23692,0: PRINT C
HR$ 13
70 NEXT A
80 PRINT AT 20,0: POKE 23692,0: PRINT C
HR$ 13
90 GO TO 30
200 FOR I=1 TO 6
210 LET N(7-I)=PEEK (15360+8*X+I)
220 NEXT I
230 LET S=64
240 LET Y=3
250 FOR I=1 TO 6
260 IF N(I)-S>=0 THEN PRINT INK 2; BRIGH
T 1;AT 20,Y;"■"
265 LET N(I)=N(I)-(S AND N(I)-S>=0)
270 LET Y=Y+1
280 NEXT I
290 LET S=S/2
300 PRINT AT 20,0: POKE 23692,0: PRINT C
HR$ 13
310 IF S>1 THEN GO TO 240
320 RETURN

```



Anomalie

REFERENCES :

Mode FAST.

Difficulté de programmation : difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

La boucle I (20-42) imprime les chiffres de repérage des grilles.

La boucle principale I (47-100) imprime les symboles des deux grilles, dans un ordre différent.

La ligne 50 choisit le caractère à imprimer.

Le sous-programme (1000-1040) s'occupe de faire l'impression à un emplacement libre (on l'appelle deux fois pour les deux grilles).

Les lignes 120 et 130 changent, de façon aléatoire, l'un des caractères.

Les coordonnées du point essayé sont entrées aux lignes 140 et 160, et le contrôle est effectué par les lignes 170 et 180.

Les lignes 185 à 210 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 NEWLINE, et son numéro de colonne, suivi de NEWLINE.

Le programme répondra alors BRAVO ou PERDU.

EXTENSIONS POSSIBLES :

Si l'on dispose de la couleur, 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.

```

15 LET T=5
20 FOR I=SGN PI TO T
30 PRINT AT NOT PI,I+I;I;TAB I
+I+16;I
40 PRINT AT I+I,NOT PI;I;TAB 1
6;I
42 NEXT I
47 FOR I=SGN PI TO 22
50 LET L%=CHR$ (11*RND+128*INT
(RND*2))
60 LET L=SGN PI
65 LET S=T
70 GOSUB 1000
80 LET L=9
85 LET S=T+SGN PI
90 GOSUB 1000
100 NEXT I
110 LET S=NOT PI
120 GOSUB 1000
130 LET L=CODE L$
140 INPUT I
150 INPUT T
170 PRINT AT I+I,T+T+16;
180 GOSUB 1010
185 CLS
190 PRINT "BRAVO" AND L=C;"PERD
U" AND L<>C
210 STOP
1000 PRINT AT 2*INT (T*RND+SGN P
I),2*INT (S*RND+L);
1010 LET C=PEEK (PEEK 16396+236*
PEEK 16399)
1020 IF C<>CODE "" AND S THEN GO
TO 1000
1030 PRINT L$
1040 RETURN

```

	1	2	3	4	5		1	2	3	4	5
1	...	.	■	■	■	1	■	■	...	■	
2	■	■	■	■	.	2	■	.		■	.
3	.	■	■	■	■	3	■	"	■	■	■
4		■	.	■	■	4	■	.	■	■	■
5	■		■	"	■	5	■	.	■	■	■

```

15 LET T=5
20 FOR I=1 TO T
30 PRINT AT 0,I+I;I;AT 0,I+I+16;I
40 PRINT AT I+I,0;I;TAB 16;I
42 NEXT I
47 FOR I=1 TO 22
50 LET L%=CHR$ (26*RND+65)
60 LET L=1
65 LET S=T
70 GO SUB 1000
80 LET L=9
90 GO SUB 1000
100 NEXT I
110 LET S=0
120 GO SUB 1000
130 BEEP .3,10: BEEP .3,18: BEEP .3,8
140 INPUT "LIGNE :";I
160 INPUT "COLONNE :";T
170 PRINT AT I+I,T+T+16;
180 GO SUB 1010
185 CLS
190 PRINT "BRAVO" AND L=C$;"PERDU" AND
L$<>C$
210 STOP
1000 LET U=2*INT (T*RND+1): LET V=2*INT (
S*RND+L)
1010 LET C$=SCREEN$ (U,V)
1020 IF C$<>" " AND S THEN GO TO 1000
1030 PRINT INVERSE 1; INK INT (4*RND+1);A
T U,V;L$: BEEP .02,0
1040 RETURN

```

	1	2	3	4	5		1	2	3	4	5
1	■	■	■	■	■	1	■	■	■	■	■
2		■	■	■	■	2	■		■	■	
3	■	■	■	■	■	3	■	■	■	■	■
4	■	■	■	■	■	4	■	■	■	■	■
5	■	■	■	■	■	5	■	■	■	■	■

Plusieurs en un

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : difficile.

Catégorie de jeu : deux joueurs.

Difficulté du jeu : variable, selon l'utilisation.

LE JEU :

Plusieurs jeux sont en réalité possibles avec ce programme qui va se comporter comme un crayon électronique vous permettant d'inscrire des symboles O ou X, en n'importe quel emplacement de votre feuille-écran de télévision. On pourra, par exemple, l'utiliser pour un jeu de morpion (le premier à réaliser un alignement de 5 pions identiques gagne) ou pour un GO (on réalise des encadrements des pions adverses) ou tout autre jeu, selon son imagination.

LE PROGRAMME :

Le terrain de jeu est délimité par 4 symboles mis en place par la ligne 10.

Le changement de joueur est effectué par la ligne 35.

Le joueur qui a le trait voit le symbole qu'il a choisi (O ou X) se placer en limite du cadre (lignes 60 et 70).

Les lignes 100 et 110 permettent aux deux joueurs de déplacer le symbole sur le terrain.

Les lignes 120 à 140 contrôlent si le déplacement est possible (case libre et non dépassement des limites).

La ligne 205 permet d'écrire sur une case libre.

POUR JOUER :

RUN provoque l'affichage des limites du terrain et du numéro du joueur, ainsi que des symboles correspondants (O ou X) que l'on devra déplacer sur l'écran à l'aide des 4 touches de direction (5, 6, 7, 8).

Il n'est pas possible de venir sur une case déjà occupée et il faudra appuyer sur la touche 0 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 BREAK.

EXTENSIONS POSSIBLES :

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

```

10 PRINT "F";TAB CODE "<";"I";
AT CODE "=";NOT PI;"L";TAB CODE
">";"J"
30 LET J=NOT PI
35 LET J=ABS (J-SGN PI)
60 LET X=CODE "<"
70 LET Y=NOT PI
80 PRINT AT X,Y;"O" AND NOT J;
"X" AND J
85 PAUSE 4E4
90 LET A=X
95 LET B=Y
100 LET X=X+(INKEY$="6" AND X<0
ODE "<")-(INKEY$="7" AND X>SGN P
I)
110 LET Y=Y+(INKEY$="8" AND Y<0
ODE ">")-(INKEY$="5" AND Y>SGN P
I)
115 PRINT AT A,B;" "
120 PRINT AT X,Y;
130 LET C=PEEK (PEEK 16398+256*
PEEK 16399)
140 IF C<>CODE "O" AND C<>CODE
"X" THEN GOTO CODE "OO"
150 PRINT CHR$ C
160 LET X=A
170 LET Y=B
200 IF INKEY$<>"O" THEN GOTO UR
L "80"
205 PRINT "O" AND NOT J;"X" AND
J
210 GOTO CODE "7"

```

```

      O
     XO
    XO X
   O XX
  O

```

```

10 PRINT "F";TAB 20;"I";AT 20,0;"L";TAB
20;"J"
30 LET J=0
35 LET J=ABS (J-1)
60 LET X=12
70 LET Y=0
80 PRINT AT X,Y;"O" AND NOT J;"X" AND J
85 PAUSE 4E4
90 LET A=X
95 LET B=Y
100 LET X=X+(INKEY$="6" AND X<19)-(INKEY
$="7" AND X>1)
110 LET Y=Y+(INKEY$="8" AND Y<19)-(INKEY
$="5" AND Y>1)
115 PRINT AT A,B;" "
120 PRINT AT X,Y;
130 LET C$=SCREEN$ (X,Y)
140 IF C$<>"O" AND C$<>"X" THEN GO TO 20
0
150 PRINT INVERSE 1; BRIGHT 1; INK 1+(C$
="X");C$
160 LET X=A
170 LET Y=B
200 IF INKEY$<>"O" OR Y=0 THEN GO TO 80
205 PRINT INVERSE 1; BRIGHT 1; INK 1+J;"
O" AND NOT J;"X" AND J: BEEP .03,10
210 GO TO 35

```

```

      O
     XO
    XO X
   O XX
  O

```

Parcours du cavalier

REFERENCES :

Mode SLOW.

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

Il se présente, ici, comme un jeu de hasard total, mais provient, en réalité, d'un problème assez difficile à résoudre : comment déplacer un cavalier sur un échiquier de façon qu'il passe une fois, et une seule, sur chacune des 64 cases du damier.

Le cavalier se déplace suivant les règles communément admises aux échecs, et sa position de départ est définie par le joueur ; par contre, ses déplacements sont effectués au hasard. Arrivera-t-il à parcourir les 64 cases ?

LE PROGRAMME :

Les lignes 15 et 20 définissent les divers déplacements possibles, dans deux variables alphanumériques A\$ et B\$; les deux boucles emboîtées I (25-45) et J (30-40) dessinent le terrain de jeu.

On entre les coordonnées de la case de départ (ligne, puis colonne) aux lignes 50 et 55.

La ligne 66 imprime le numéro du coup.

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

La boucle Z (145-170) cherche si ce déplacement est possible ; si ce n'est pas le cas, on essaie tous les autres (toujours dans le même ordre).

Les lignes 146 et 147 effectuent le déplacement.

La ligne 148 contrôle que l'on ne sorte pas du terrain de jeu.

Les lignes 150 et 160 envoient en ligne 220 si le coup est possible.

La ligne 180 gère la fin de partie (cavalier bloqué) et affiche le score.

Les lignes 200, 210 et 220 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 NEWLINE puis son numéro de colonne, suivi de NEWLINE. Le jeu commence immédiatement et il se déroule de façon entièrement automatique jusqu'à ce que le cavalier soit bloqué. On peut, bien sûr, arrêter à tout moment par BREAK.

EXTENSIONS POSSIBLES :

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

```

10 LET S=SGN PI
15 LET A$="56653223"
20 LET B$="653223"+A$+"56"
25 FOR I=5 TO LEN A$
30 FOR J=5 TO LEN A$
35 PRINT AT I+I,INT PI*J;";"
40 NEXT J
45 NEXT I
50 INPUT I
55 INPUT J
60 LET A$=A$+A$
65 PRINT AT I+I,INT PI*J;S
140 LET Z=INT (CODE "██" *RND+SGN
PI)
145 FOR Z=Z TO Z+CODE "██"
146 LET A=I+VAL A$(Z)-CODE "██"
147 LET B=J+VAL B$(Z)-CODE "██"
148 IF A<SGN PI OR B<SGN PI OR
A>CODE "██" OR B>CODE "██" THEN NE
XT Z
150 PRINT AT A+A,INT PI*B;
160 IF PEEK (PEEK 16398+256*PEE
K 16399)=CODE "." THEN GOTO CODE
"COS"
170 NEXT Z
180 PRINT AT CODE "=",NOT PI;S;
)
200 LET I=A
210 LET J=B
220 LET S=S+SGN PI
230 GOTO CODE "PI"

```

```

. 40 3 26 . 28 5 .
2 23 . 37 4 25 . 29
39 . 1 24 27 36 15 6
22 . 38 . 14 . 30 .
. . . . 35 . 7 16
. 21 34 13 . 9 . 31
. . 19 . 11 32 17 8
20 . 12 33 18 . 10 .

```

40

```

10 LET S=1
15 LET A$="56653223"
20 LET B$="653223"+A$+"56"
25 FOR I=5 TO LEN A$
30 FOR J=5 TO LEN A$
35 PRINT PAPER 5;AT 0,3*I;I;AT I+I,0;I;
PAPER 2;AT I+I,3*J;";"
40 NEXT J
45 NEXT I
50 INPUT "LIGNE DE DEPART :";I
55 INPUT "COLONNE DE DEPART :";J
60 LET A$=A$+A$
66 PRINT INVERSE 1; INK 4;AT I+I,3*J;S
140 LET Z=INT (8*RND+1)
145 FOR Z=Z TO Z+8
146 LET A=I+VAL A$(Z)-4
147 LET B=J+VAL B$(Z)-4
148 IF A<1 OR B<1 OR A>8 OR B>8 THEN NEX
T Z
160 IF SCREEN$ (A+A,3*B)="" THEN GO TO
200
170 NEXT Z
180 PRINT FLASH 1;AT 20,0;"TOTAL :";S; B
EEP .3,10; BEEP .3,18; BEEP .3,8; STOP
200 LET I=A
210 LET J=B
220 LET S=S+1; BEEP 0.1,13
230 GO TO 66

```

	1	2	3	4	5	6	7	8
1	.	13	.	.	2	.	.	.
2	.	.	1	14	.	20	3	.
3	12	.	.	19	4	.	.	21
4	.	28	15	.	.	.	3	.
5	16	11	18	27	.	.	22	.
6	29	26	.	.	.	.	.	8
7	10	17	30	25	8	.	.	23
8	31	.	9	.	.	24	7	.

TOTAL : 31

Mur de briques

REFERENCES :

Mode SLOW.

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 1K, mais offre néanmoins de bonnes performances : affichage du score en fin de partie, rebonds aléatoires, etc. La règle en est très élémentaire : vous disposez d'une raquette qui peut être déplacée horizontalement pour renvoyer une balle (matérialisée par un point) qui va démolir les briques dessinées en haut de l'écran.

Votre problème est d'en démolir le plus grand nombre possible, sans rater la balle, ce qui provoquerait la fin de partie et l'affichage du score.

LE PROGRAMME :

La boucle I (30-50) dessine le terrain de jeu et les briques.

La ligne 55 dessine la raquette dont la position est contrôlée par la ligne 60.

Les lignes 85, 90 et 100 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 132 et 138 (le grisé employé correspond, ici, à graphic shift H).

La ligne 64 affichera le score en fin de partie.

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 (5 et 8).

Si, à un moment quelconque, sa balle ne touche pas la raquette, le score sera affiché et la partie s'arrête.

On peut aussi terminer à tout moment par BREAK.

EXTENSIONS POSSIBLES :

On peut envisager d'agrandir le terrain de jeu, d'augmenter le nombre de briques, de prévoir plusieurs balles, de dessiner chaque rangée de briques d'une couleur différente, d'introduire des effets sonores.

On peut aussi, en s'inspirant des jeux d'arcades, introduire de petits ouvriers qui viendront reconstruire votre mur de briques, au fur et à mesure que vous le démolissez.

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 ZX.

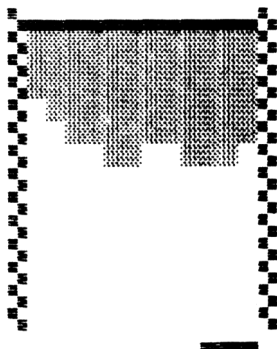
Après avoir lancé le programme par RUN, il convient d'entrer la dimension de la grille (3, NEWLINE, ou 4,).

ZX-81

```

5 LET S=NOT PI
10 LET Y=PI+PI
20 LET J=Y
30 FOR I=NOT PI TO CODE "$"
40 PRINT TAB SGN PI;"
" AND NOT I;"
" AND I AND I<Y;"
" AND I>Y
50 NEXT I
52 LET H=-SGN PI
54 LET K=H
55 PRINT AT CODE "Y;"
60 LET Y=Y+(INKEY$="8" AND Y<C
ODE "E")-(INKEY$="5" AND Y>SGN P
I)
62 PRINT AT I,J;"
64 IF I>CODE " THEN PRINT S;
66 LET I=I+H
80 LET J=J+K
85 PRINT AT I,J;
90 LET C=PEEK (PEEK 16398+256*
PEEK 16399)
100 IF C THEN GOTO C
118 PRINT "
120 GOTO CODE "R"
131 LET H=-H
132 LET K=RND-RND
134 LET K=-K
135 GOTO CODE "PI"
136 LET S=S+SGN PI
138 LET K=-K+(2*(RND<.5)-SGN P
I)*RND/INT PI)
140 LET H=-H
142 GOTO CODE "R"

```



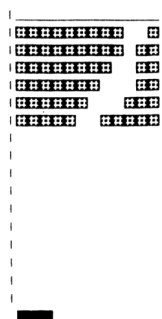
10

ZX-SPECTRUM

```

5 LET S=0
10 LET Y=6
20 LET J=Y
30 FOR I=0 TO 16
40 PRINT TAB 1; PAPER 2;"(
" AND NOT I; PAPER 2;"I" AND I AND I<=Y;
INVERSE 1; INK 3+INT (I/2.5);"#####
#" AND I AND I<=Y; INVERSE 0; INK 0; PAPE
R 2;"I" AND I AND I<=Y;"I" AND I>Y; PAPER
7;"
" AND I>Y; PAPER 2;"I" AN
D I>Y
50 NEXT I
52 LET H=-1
54 LET K=H
55 PRINT INVERSE 1; INK 1;AT 17,Y;"_
"
60 LET Y=Y+(INKEY$="8" AND Y<12)-(INKEY
$="5" AND Y>1)
62 PRINT AT I,J;"
64 PRINT AT 20,17;"SCORE :";S: IF I>20
THEN BEEP 1,0: PAUSE 4E4: RUN
66 LET I=I+H
80 LET J=J+K
85 PRINT AT I,J;
90 LET C$=SCREEN$ (I,J)
100 IF C$<>" " THEN BEEP .03,0: GO TO 13
1+3*(C$="I")+5*(C$="#")
118 PRINT "0"
120 GO TO 55
131 LET H=-H
132 LET K=RND-RND
134 LET K=-K
135 GO TO 66
136 LET S=S+1
138 LET K=-K+(2*(RND<.5)-1)*RND/3)
140 LET H=-H
142 GO TO 55

```



SCORE : 13

Othello

REFERENCES :

Mode SLOW ou FAST.

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de 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 :

Comme il est difficile de faire tenir le programme sur 1 Koctet, il a fallu sacrifier légèrement sur la présentation.

La boucle P (11-26) dessine le terrain de jeu et affiche un X ou un O, selon le joueur qui commence.

La boucle P (41-212) contrôle la validité du coup (la case est-elle libre ? : lignes 203 à 209), en utilisant une partie de programme qui se trouve aux lignes 224 à 242 (a-t-on pris un ou plusieurs ennemis en ligne 224, et retournement éventuel, autorisé par la ligne 230 : on refait un balayage dans toutes les directions, si l'encadrement s'est révélé possible).

La ligne 41 définit les 8 directions d'exploration.

Les lignes 56 et 59 calculent les déplacements à effectuer pour explorer ces 8 directions.

Les lignes 66 et 200 réalisent les déplacements.

La ligne 218 assure le changement de joueur, s'il y a eu retournement (contrôlé en ligne 215).

POUR JOUER :

Après RUN, chaque joueur entrera à son tour les coordonnées (ligne, NEWLINE, puis colonne, NEWLINE) de la case où il désire jouer ; si le coup est accepté, les pions sont retournés et l'on change de joueur ; si aucun coup n'est possible, on entre 0 en première coordonnée pour le changement de joueur.

```

5 INPUT A
8 LET B=SGN PI
11 FOR P=B TO A+A
14 PRINT AT NOT PI,P;P;AT P,ND
T PI,P
17 NEXT P
20 PRINT AT A,R;"XO"(AT A+B,A)
"OX"
23 LET K=B
26 PRINT AT A*A,B;"X" AND K;"O
" AND NOT K
29 INPUT U
32 IF NOT U THEN GOTO CODE " A
ND "
35 INPUT U
38 LET H=PI
41 FOR P=B TO CODE " "
44 LET S=NOT PI
47 LET W=NOT PI
50 LET T=U
53 LET R=V
56 LET F=(P<INT PI OR P>PI+PI+
B)-(P>=PI AND P<=PI+PI)
59 LET G=(P>=INT PI+INT PI)-(P
<=INT PI AND P<=PI+B)
62 LET R$=("X" AND S=B)+("O" A
ND S=B+B)
64 PRINT AT T,R;R$
66 LET T=T+F
200 LET R=R+G
203 PRINT AT T,R;
206 LET C=PEEK (PEEK 16398+256*
PEEK 16399)
209 IF C=CODE "O" OR C=CODE "X"
THEN GOTO CODE " STEP "
212 NEXT P
215 IF H THEN GOTO CODE " "
218 LET K=ABS (K-B)
221 GOTO CODE " "
224 IF (C=CODE "O" AND K) OR (C
=CODE "X" AND NOT K) THEN LET W=
B
227 PRINT R$
230 IF W AND ((C=CODE "O" AND K
) OR (C=CODE "X" AND NOT K)) THE
N GOTO CODE "PI"
233 IF NOT W OR S THEN GOTO COD
E "USR "
236 LET S=B+NOT K
239 LET H=H+NOT S
242 GOTO CODE "J"

```

12345678

```

1
2 0 XO
3 0 XXO
4 000X O
5 0X
6 X X
7 X
8

```

X

```

5 INPUT "DIMENSIONS :";A
8 LET B=1
11 FOR P=B TO A+A
14 PRINT AT O,P+P;P;AT P+P,O;P
17 NEXT P
20 PRINT INVERSE 1; INK 2;AT A+A,A+A;"X
";AT A+A,A+A+B+B; INK 5;"O";AT A+A+B+B,A+
A;"O"; INK 2;AT A+A+B+B,A+A+B+B;"X"
23 LET K=B
26 PRINT INVERSE 1; FLASH 1; INK 2+3*NO
T K;AT 21,0;"X" AND K;"O" AND NOT K
29 INPUT "LIGNE :";U
32 IF NOT U THEN GO TO 218
35 INPUT "COLONNE :";V
38 LET H=PI
41 FOR P=B TO 8
44 LET S=0
47 LET W=0
50 LET T=U+U
53 LET R=V+V
56 LET F=(P<3 OR P>7)-(P>3 AND P<7)
59 LET G=(P>=6)-(P>=8 AND P<=5)
62 LET R$=("X" AND S=B)+("O" AND S=B+B)
64 PRINT INVERSE 1; INK 2+3*NOT K;AT T,
R;R$
66 LET T=T+F+F
200 LET R=R+G+G
203 PRINT AT T,R;
206 LET C=CODE SCREEN$ (T,R)
209 IF C=CODE "O" OR C=CODE "X" THEN GO
TO 224
212 NEXT P
215 IF H THEN GO TO 26
218 LET K=ABS (K-B); BEEP .4,10
221 GO TO 26
224 IF (C=CODE "O" AND K) OR (C=CODE "X"
AND NOT K) THEN LET W=B
227 PRINT INVERSE 1; INK 2+3*NOT K;R$
230 IF W AND ((C=CODE "O" AND K) OR (C=C
ODE "X" AND NOT K)) THEN GO TO 66
233 IF NOT W OR S THEN GO TO 212
236 LET S=B+NOT K
239 LET H=H+NOT S
242 GO TO 47

```

1 2 3 4 5 6 7 8

```

1
2
3
4
5
6
7
8

```

Quatre à quatre

REFERENCES :

Mode SLOW.

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion (un ou deux joueurs).

Difficulté du jeu : difficile.

LE JEU :

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

LE PROGRAMME :

La variable alphanumérique A\$, en ligne 15, contient la table des coups possibles (déplacement du cavalier).

Les boucles emboîtées I (25-50) et J (30-47) dessinent le terrain de jeu, et les positions de départ.

La ligne 66 permet d'afficher O ou X, selon le joueur qui a le trait.

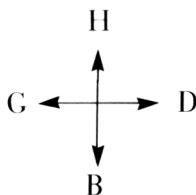
La ligne 75 permet l'affichage du curseur clignotant (ici un +) qui indique sur quelle case on se trouve.

Les lignes 90 et 95 permettent le déplacement du curseur.

La ligne 100 regarde si l'on a appuyé sur une touche de 1 à 8, donnant la direction du déplacement.

La ligne 180 réimprime la figure qui était sous le curseur. La ligne 187 transforme en nombre le symbole chiffre frappé. La ligne 190 réalise le contrôle de départ.

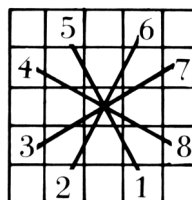
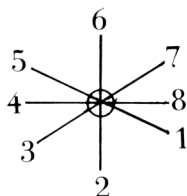
Les lignes 195 et 200 effectuent le déplacement. La ligne 210 réalise le contrôle d'arrivée. Le résultat est imprimé par la ligne 201, et un . est imprimé à l'emplacement de départ (ligne 223). Le sous-programme 229-240 est utilisé pour les divers contrôles.



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 D, G, H, B.

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 **BREAK**.

Si la manœuvre de la touche **INKEY\$** conduisait à un moment à sortir du programme, il faudrait repartir en tapant **GOTO 66**.

EXTENSIONS POSSIBLES :

On peut utiliser la couleur, améliorer les contrôles, prévoir un test de fin, utiliser un terrain de jeu différent ; on pourra aussi envisager l'écriture d'un programme permettant au **ZX** de participer au jeu avec de bonnes chances de l'emporter.

```

10 LET H=SGN PI
15 LET A$="5665322356"
25 FOR I=H TO CODE "I"
30 FOR J=H TO CODE "I"
35 PRINT AT I+I,J+J;"."
40 IF I<INT PI AND J<INT PI TH
EN PRINT AT I+I,J+J;"0"
45 IF I>INT PI AND J>INT PI TH
EN PRINT AT I+I,J+J;"X"
47 NEXT J
50 NEXT I
55 LET X=H
60 LET Y=X
66 PRINT AT CODE "$",NOT PI;"0"
" AND H;"X" AND NOT H
70 GOSUB CODE " FAST "
72 LET I=J
75 PRINT "+"
80 LET A=X
85 LET B=Y
90 LET X=X+(INKEY$="B" AND X<C
ODE "I")-(INKEY$="H" AND X>SGN P
I)
95 LET Y=Y+(INKEY$="D" AND Y<C
ODE "I")-(INKEY$="G" AND Y>SGN P
I)
100 IF INKEY$<"9" AND INKEY$>"0"
" THEN GOTO CODE "0"
180 PRINT AT A+A,B+B;CHR$ I
185 GOTO CODE "PI"
187 LET J=VAL INKEY$
190 IF J=PI*PI OR (I<>CODE "0"
AND H) OR (I<>CODE "X" AND NOT
H) THEN GOTO CODE "0"
195 LET X=X+VAL A$(J)-CODE "I"
200 LET Y=Y+VAL A$(J+2)-CODE "I"
205 GOSUB CODE " FAST "
210 IF J<>CODE "." THEN GOTO CO
DE "0"
215 PRINT "0" AND H;"X" AND NOT
H
220 LET H=ABS (H-SGN PI)
223 LET I=CODE "I"
225 GOTO CODE "0"
229 PRINT AT X+X,Y+Y;
235 LET J=PEEK (PEEK 16398+256*
PEEK 16399)
240 RETURN

```

```

0 0 . . .
. . X . .
. . 0 . X
. . 0 . X
. . . . X
0

```

```

10 LET H=1
15 LET A$="5665322356"
25 FOR I=H TO 5
30 FOR J=H TO 5
35 PRINT AT 0,I+I;I;AT I+I,0;I;AT I+I,J
+J;"."
40 IF I<3 AND J<3 THEN PRINT AT I+I,J+J
;"0"
45 IF I>3 AND J>3 THEN PRINT AT I+I,J+J
;"X"
47 NEXT J
50 NEXT I
55 LET X=H
60 LET Y=X
66 PRINT AT 13,0;"JOUER "; "0" AND H;"X"
" AND NOT H
70 GO SUB 229
72 LET I=J
75 PRINT "+"
80 LET A=X
85 LET B=Y
90 LET X=X+(INKEY$="B" AND X<5)-(INKEY$
="H" AND X>1)
95 LET Y=Y+(INKEY$="D" AND Y<5)-(INKEY$
="G" AND Y>1)
100 IF INKEY$<"9" AND INKEY$>"0" THEN GO
TO 187
180 PRINT AT A+A,B+B;CHR$ I
185 GO TO 66
187 LET J=VAL INKEY$
190 IF J>9 OR (I<>CODE "0" AND H) OR (I<
>CODE "X" AND NOT H) THEN GO TO 180
195 LET X=X+VAL A$(J)-4
200 LET Y=Y+VAL A$(J+2)-4
205 GO SUB 229
210 IF J<>CODE "." THEN GO TO 180
215 PRINT "0" AND H;"X" AND NOT H
220 LET H=ABS (H-1)
223 LET I=CODE "I"
225 GO TO 180
229 PRINT AT X+X,Y+Y;
235 LET J=CODE SCREEN$ (X+X,Y+Y)
240 RETURN

```

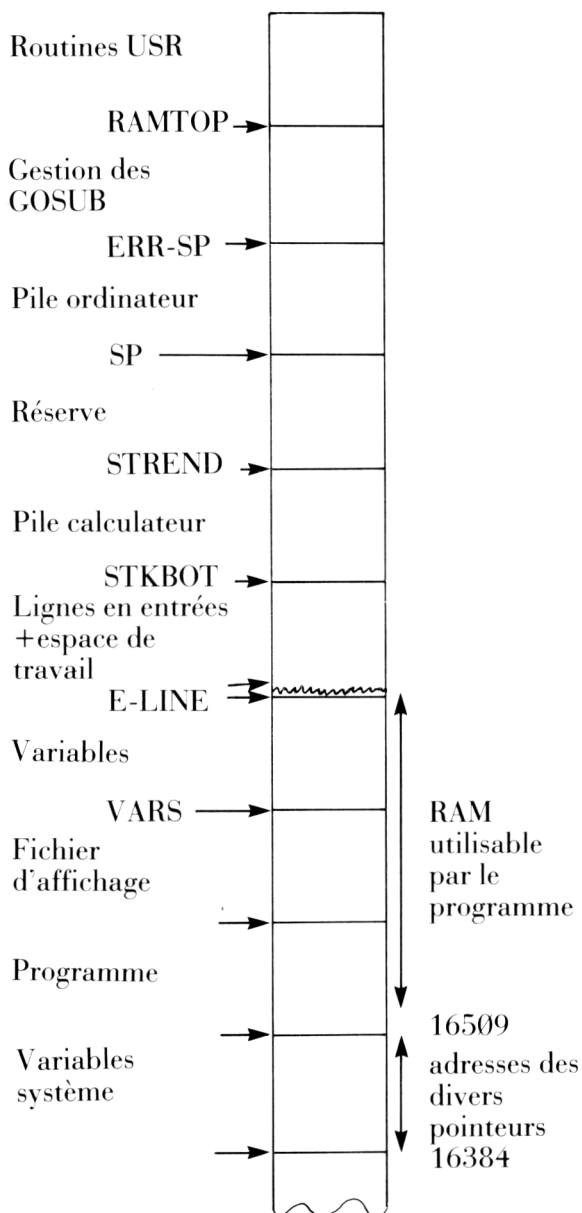
```

1 2 3 4 5
1 0 0 . . .
2 . . X . .
3 . . X . .
4 . 0 0 . .
5 . . . X X

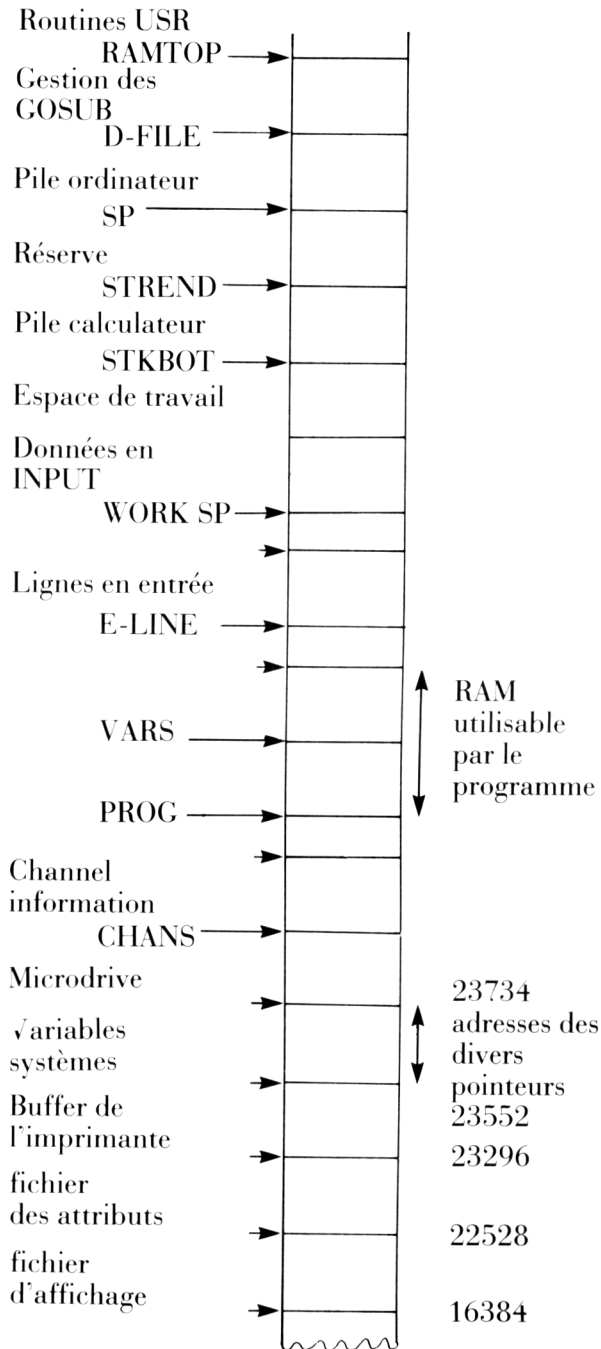
```

Annexe

L'organisation de la mémoire sur ZX-81



L'organisation de la mémoire sur ZX-Spectrum



Achevé d'imprimer en novembre 1984
sur les presses de l'imprimerie Laballery et C°
58500 Clamecy
Dépôt légal : novembre 1984

N° d'impression : 410057
N° d'édition : 86595-95-3
ISBN : 2-86595-095-6



Apprendre en se distrayant, tel est l'objectif de ce livre.

Au fil de ces 102 programmes de jeux il vous guidera dans l'exploration du Basic Sinclair. Les programmes sont classés par niveaux, chaque niveau faisant appel à de nouvelles connaissances et à une plus grande maîtrise du Basic. Des instructions correspondant à chaque niveau sont présentées et commentées avec des remarques concernant les points spécifiques du ZX81 et du ZX Spectrum.

Tous les jeux sont décrits, et les programmes abondamment commentés; un exemple d'exécution est fourni pour chaque version (ZX81 et ZX Spectrum). Écrits pour le ZX81 dans sa version de base (1K octet), pour le ZX spectrum et pour les Timex 1000, 1500 et 2000, la plupart des programmes sont également utilisables pour le ZX80 version New Rom.

**DES LIVRES
POUR VOTRE
ORDINATEUR**



ISBN : 2.86595.095.6

110FF

Couverture GIBRAT

PROGRAMMES POUR ZX81 ET ZX SPECTRUM PAR JACQUES DECOURCHAI

