



102 PROGRAMMES



PAR
JACQUES
DECONCHAT



102 PROGRAMMES POUR
Commodore
128

Connaissez-vous toute la collection Commodore 128 chez P.S.I. ?

Système

- Clefs pour Commodore 128 : 1. Mode 64 — Daniel-Jean David
- CP/M Plus pour Commodore 128 — Yvon Dargery

Maîtrise du BASIC

- Basic Commodore 128 — Méthodes pratiques — Jacques Boisgontier

A paraître :

- Clefs pour Commodore 128 : 2. Mode 128 — Jean-François Sehan

Pour tout problème rencontré dans les ouvrages P.S.I.
vous pouvez nous contacter au numéro ci-dessous :

Numéro Vert/Appel Gratuit en France

05 21 22 01

(Composer tous les chiffres, même en région parisienne)

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, 1986
ISBN : 2-86595-322-X

102 PROGRAMMES POUR C 128

**JACQUES
DECONCHAT**

**102 PROGRAMMES POUR
Commodore
128**



EDITIONS DU P.S.I.

1986

SOMMAIRE

Avant-propos	8
Organisation de l'ouvrage	9

Niveau 1

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

Niveau 2

Dimensionnement d'un tableau	
Recherche d'un élément dans un tableau et dans une chaîne	
Les instructions DATA, READ et RESTORE	60
Dé truqué	62
Visions	64
421	66
L'électricien fou	68
La course au 20	70
Dessin	72
Poker	74
Le jeu de saute-mouton	76
Lettres	78
Machinamot	80
Espion	82
Accrochez les wagons	84

Tirage d'un dé truqué	86
Le jeu de Marienbad	88
Compose	90
Jour de la semaine	92
Tour de cartes	94
Le jeu de la vie	96

Niveau 3

Manipulation des chaînes, attente d'une touche et instructions du Basic structuré	101
Plus grand, plus petit	104
Ordre	106
Le pendu	108
Belote	110
Un taquin de calcul	112
Pendu inversé	114
Nicomaque	116
Mastermind	118
Idem	120
Horloge	122
Reines sur l'échiquier	124
Mots	126
Permutations	128
Bridge	130
Hanoi	132
Répétitions	134
Jeu de loto	136
Roulette	138
Musique	140
A travers	142
Le jeu du 15	144

Niveau 4

Les techniques d'animation élémentaires	148
Le loup et les agneaux	151
Un dé	154
La bombe	156
Diago	158
Décomaque	160
Le jeu du missile	162
Ardoise magique	164
Affichage publicitaire	166
Le singe	168
Mathieu	170
Adresse	172
Sous-marin	174

Zombies	176
Calculatrice	178
Clavier	180
Chasse au sous-marin	182
Le lézard d'Alice	184
Dessin de dés	186
Évasion	188
Chapeaux	190
Combat aérien	192
L'indésirable	194
Course de chevaux	196
CREASPRITE	198

Niveau 5

Inscription directe dans les mémoires :	
déplacement d'objets de bas en haut	
sur l'écran : SCROLL	203
Dames	205
Tic tac toe	208
Smokey	210
Course de voitures	212
Danger spatial	214
Anagrammes	216
Œufs	218
Taquin de lettres	220
Cibles	222
Gouttes	224
Obstacles	226
Captures dans l'espace	228
Anomalie	230
Plusieurs en un	232
Parcours du cavalier	234
Mur de briques	236
Othello	238
Échecs	240
Festival	242
Quatre à quatre	244
Annexe	247

Avant-propos

Vous trouverez dans ce livre plus de 100 programmes (essentiellement des jeux), rédigés en Basic Microsoft pour l'ordinateur Commodore C128.

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

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

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

Le niveau 5 donne une idée sur l'utilisation de certaines techniques un peu spécifiques, comme la lecture et l'écriture directe dans la mémoire. Ces techniques pourront être reprises avec profit pour améliorer les programmes précédents.

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

Organisation de l'ouvrage

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

- **Le niveau 1** permet l'étude des instructions élémentaires du langage Basic : **LET** (optionnel ici), **PRINT**, **INPUT**, **IF...THEN**, **FOR...NEXT**, **GOTO**, **GOSUB**, **ON...GOTO**, **ON...GOSUB**.
- **Au niveau 2**, on utilisera les instructions de confection et de manipulation des éléments d'un tableau et de recherche d'un élément dans une chaîne : **DIM**, **R(I)**, **MID\$**, **READ**, **DATA**.
- **Au niveau 3**, on utilise la panoplie complète des instructions de traitement des chaînes de caractères : **MID\$**, **RIGHT\$**, **LEFT\$**, le **GET-KEY** pour attendre une touche et les instructions du Basic structuré **DO/LOOP BEGIN/BEND**.
- **Le niveau 4** fait appel à une instruction très particulière, **GET**, qui facilite la réalisation de jeux interactifs et nous fait découvrir la haute résolution et les **SPRITE**.
- Enfin, **le niveau 5** nous fait découvrir des instructions plus spécifiques : déplacement d'objets sur l'écran, lecture et écriture dans la mémoire.

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

Le programme, écrit en Basic C128, est ensuite reproduit. Les listings ont été obtenus directement d'après les originaux de l'auteur, sur une imprimante matricielle, ce qui devrait garantir contre tout risque de dysfonctionnement. Les minuscules ne sont pas utilisées.

Des explications sur les codes écran très particuliers qui apparaissent dans les **PRINT**, lors d'un listing, sont fournies en annexe. Il sera prudent, avant de lancer un nouveau programme, de remettre l'appareil dans son état d'origine en appuyant sur les touches **RUN/STOP** et **RESTORE**.

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.
Tirage de dés	Tirage de dés, avec graphisme simplifié.
Le jeu du 21	Jeu de hasard, l'ordinateur tient la banque.
Pair - impair	Un jeu de casino en trois temps (un ou plusieurs joueurs).
Devine un chiffre	Un jeu très facile, pour un ou plusieurs joueurs.
Le jeu de Wytthof	Un jeu de réflexion assez simple, pour deux joueurs.
Rebonds	Un jeu d'adresse très visuel, pour un joueur.
Jackpot	Jeu de hasard, pour un joueur.
Couloirs	Un jeu de logique, à découvrir.
Echiquier	Une base graphique pour vos propres créations.
Le jeu de la fève	Un jeu de réflexion, pour deux joueurs.
Jeu de l'oie aléatoire	Un très vieux jeu, revu et corrigé par l'ordinateur, deux joueurs.

Instructions élémentaires

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

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

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

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

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

L'INSTRUCTION LET OU INSTRUCTION D'AFFECTION

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.

Pour le Basic du C128, comme dans la plupart des Basic, le LET est devenu optionnel, aucune ambiguïté n'étant possible, on écrira donc :
A = 12 ; NOM\$ = "TOTO" ; etc.

L'INSTRUCTION PRINT OU INSTRUCTION DE SORTIE (ECRAN)

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

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

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

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

Le Commodore 128 possède en outre quelques traits supplémentaires concernant l'instruction PRINT, qui peut être utilisée pour commander l'effacement de l'écran, les mouvements du curseur ou la couleur (voir l'annexe à ce sujet).

Pour remédier au manque de tabulation verticale (il n'y en a pas sur la version de base), on pourra, soit utiliser des symboles "curseur bas" dans un PRINT, soit faire appel à un petit sous-programme (voir plus loin) très élémentaire.

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

Cette instruction, qui provoque un arrêt et une attente de l'appareil, *permettra d'entrer un nombre ou un texte, qui sera affecté à la variable suivant l'ordre INPUT*. Ceci est fait en mode conversationnel, c'est-à-dire au fur et à mesure des besoins ; en effet, lorsque le C128 rencontre

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

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

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

On pourra, par exemple, écrire :

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

L'INSTRUCTION GOTO OU INSTRUCTION DE RENVOI INCONDITIONNEL

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

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

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

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

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

IF condition = vraie THEN numéro de ligne,

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

On pourra aussi utiliser le test sous la forme :

IF condition THEN instruction ;

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

On peut par exemple écrire :

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

Le Basic 7.0 du C128 dispose en outre d'une clause ELSE : on peut alors l'interpréter comme l'équivalent français de Si ____ alors ____ sinon ____.

On écrit :

IF ____ THEN ____ : ELSE ____

Le ELSE doit nécessairement être précédé de deux points.

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

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

Ainsi, si l'on écrit :

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

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

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

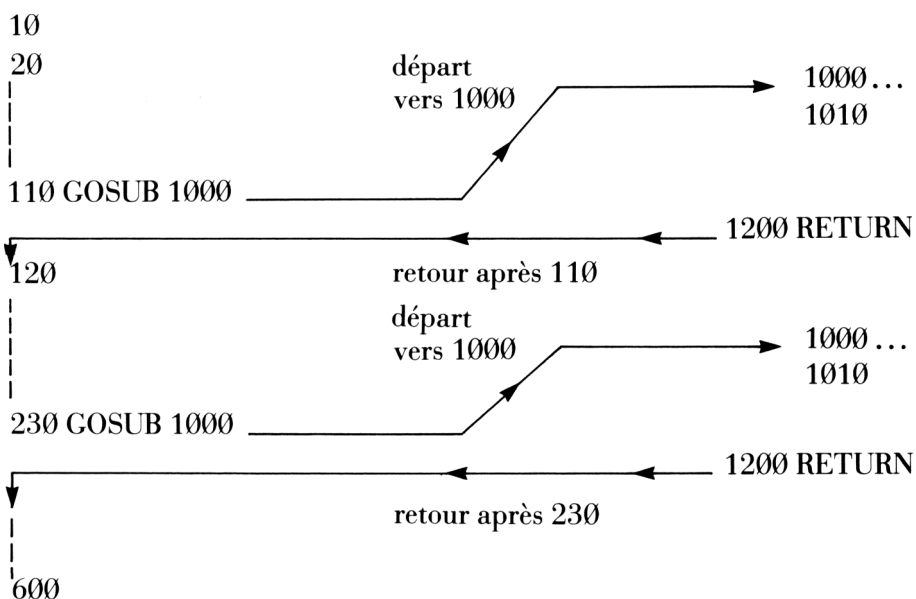
Les instructions contenues dans la boucle seront exécutées pour $I = 0$, pour $I = 2$, ... et ceci jusqu'à $I = 20$, soit 11 fois. La variable I peut être utilisée pour faire un calcul dans la boucle, mais ne devra pas être modifiée dans la boucle. On peut, bien sûr, sortir de la boucle, avant $I = 20$,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

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

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

Voici le schéma théorique de fonctionnement :

Programme principal



Fin du programme principal

1000
1200 RETURN

Sous-programme 1

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

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

On peut aussi appeler un sous-programme de façon conditionnelle, en utilisant l'instruction `ON___ GOSUB ___`.

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

Le nombre mystérieux

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu de déduction pour un joueur.

Difficulté du jeu : facile.

LE JEU :

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

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

LE PROGRAMME :

La ligne 40 permet de fabriquer le nombre à deviner.

Les entrées sont faites par les lignes 60 et 70.

La ligne 80 teste si le résultat est exact.

La ligne 90 signale un essai trop petit, et la ligne 100 un essai trop grand.

Le compteur d'essais est en ligne 110.

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

POUR JOUER :

On lancera le programme par RUN.

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

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

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

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

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

On peut arrêter à tout moment en tapant RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

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

```

10 REM LE NOMBRE MYSTERIEUX
15 COLOR 0,6:COLOR 5,3:COLOR 4,1:GRAPHIC
  0,1
20 INPUT"NOMBRE DE CHIFFRES :";N
30 IF N > 7 THEN RUN
40 X = INT ( RND ( 1 ) * 10 ^ N + 1 )
50 J = 1
60 PRINT:PRINT"ESSAI NO.";J
70 INPUT"VOTRE NOMBRE :";M
80 IF M = X THEN 200
90 IF M < X THEN PRINT"TROP PETIT":PLAY"
05T0U5CDE"
100 IF M > X THEN PRINT"TROP GRAND":PLAY
"01T5U5AGF"
110 J = J + 1
120 PRINT"IT"
130 GOTO 60
200 PRINT"J"
210 PRINT"VOUS AVEZ TROUVE";X;" EN";J;"
ESSAIS":PLAY"03T8U5CDEFGAB04C"
220 INPUT"AUTRE PARTIE ( O/N )";R$
230 IF R$ = "O" THEN RUN

```

Cube

REFERENCES :

Difficulté de programmation : aucune.

Catégorie de jeu : dessin.

Difficulté du jeu : aucune.

LE JEU :

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

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

LE PROGRAMME :

Le dessin du cube est réalisé par l'impression de points, en utilisant deux boucles : la boucle I (20-150), et la boucle I (160-210).

Le nom de l'objet représenté est affiché en ligne 230 (ici : CUBE).

POUR JOUER :

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

EXTENSIONS POSSIBLES :

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

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


```

10 REM CUBE
15 COLOR 0,2:COLOR 4,2:COLOR 5,7:GRAPHIC
  0,1
20 FOR I = 0 TO 9
30 Z = 0 : GOSUB 500
40 PRINTTAB(17+I);"."
50 Z = 7 : GOSUB 500
60 PRINTTAB(10+I);"."
70 Z = 10 : GOSUB 500
80 PRINTTAB(17+I);"."
90 Z = 17 : GOSUB 500
100 PRINTTAB(10+I);"."
110 Z = 7 + I : GOSUB 500
120 PRINTTAB(10);".";TAB(20);"."
130 Z = I : GOSUB 500
140 PRINTTAB(17);".";TAB(27);"."
150 NEXT I
160 FOR I = 0 TO 7
170 Z = 17-I : GOSUB 500
180 PRINTTAB(I+10);".";TAB(20+I);"."
190 Z = 7-I : GOSUB 500
200 PRINTTAB(I+10);".";TAB(20+I);"."
210 NEXT I
220 Z = 18 : GOSUB 500
230 COLOR 5,8:PRINTTAB(25);"CUBE"
240 PLAY"04T4U5CEG":END
500 PRINT"8";
510 FOR J = 0 TO 2
520 PRINT
530 NEXT J
540 RETURN

```

Chaud, froid

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

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

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

LE PROGRAMME :

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

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

La ligne 100 regarde si vous avez gagné.

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

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

POUR JOUER :

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

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

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

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

Pour sortir du programme avant la fin de la partie, il suffit de taper RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

Il sera intéressant de trouver un moyen plus visuel pour symboliser la distance, par exemple des pavés allant du rouge au bleu, aux emplacements essayés.

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

```

10 REM CHAUD,FROID
20 COLOR 0,1:COLOR 4,1:COLOR 5,2:GRAPHIC
  0,1
30 INPUT"DIMENSION DE LA GRILLE :";D
40 X = INT ( RND (1) * D + 1 )
50 Y = INT ( RND (1) * D + 1 )
60 COLOR 5,2:PLAY"CEG":INPUT"DONNEZ X : "
  ;A
70 PLAY"DFA":INPUT"DONNEZ Y : ";B
80 S = S + 1
90 T = ABS ( X - A ) + ABS ( Y - B )
100 IF T = 0 THEN 6000
110 PRINT"VOUS ETES ";
120 N = INT ( SQR (D) / T ) + 1
130 IF N > 5 THEN N = 5
140 IF T <= SQR (D) + 2 THEN ON N GOTO 1
000,2000,3000,4000,5000
150 COLOR 5,7:PRINT"COMPLETEMENT GELE"
160 GOTO 60
1000 COLOR 5,14:PRINT"A PEINE TIEDE"
1010 GOTO 60
2000 COLOR 5,6:PRINT"TIEDE"
2010 GOTO 60
3000 COLOR 5,9:PRINT"CHAUD"
3010 GOTO 60
4000 COLOR 5,11:PRINT"TRES CHAUD"
4010 GOTO 60
5000 COLOR 5,3:PRINT"BRULANT"
5010 GOTO 60
6000 PRINT"J"
6010 PRINT"VOUS AVEZ TROUVE EN";S;" COUP
S":PLAY"05T4CEGB"

```

Les allumettes

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : contre le C128.

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 C128. Chacun prend à son tour autant d'allumettes qu'il le désire, dans la limite du nombre maximal autorisé. Celui qui prend la dernière allumette est considéré comme perdant.

LE PROGRAMME :

Le jeu du joueur commence en ligne 70.

Celui du C128 commence en ligne 110.

La ligne 80 contrôle la légalité du coup du joueur.

La ligne 90 gère le changement de joueur.

Le coup du C128 est calculé dans une boucle non formelle, de la ligne 120 à la ligne 150.

Le résultat est affiché par la ligne 1020.

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

POUR JOUER :

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

Après chaque coup, le nombre d'allumettes restant est affiché, et le C128 affiche VOUS RETIREZ lorsque c'est au joueur de prendre.

Celui qui prend la dernière allumette a perdu.

EXTENSIONS POSSIBLES :

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

```

10 REM LES ALLUMETTES
20 COLOR 0,7:COLOR 4,7:COLOR 5,2:GRAPHIC
  0,1
30 INPUT"NOMBRE TOTAL :";N
40 INPUT"PRISE MAXIMUM :";I
50 INPUT"QUI COMMENCE (VOUS:0 / MOI:1 )"
  ;D
60 IF D = 1 THEN 110
70 COLOR 5,2:INPUT"VOUS RETIREZ :";P
80 IF P <>INT(P) OR P < 1 OR P > I OR P
> N THEN PRINT".IT":GOTO 70
90 D = 1
100 GOTO 1000
110 S = I + D
120 T = (N - S)/(I + D)
130 IF T = INT(T) THEN 200
140 S = S - D
150 GOTO 120
200 P = ABS (S - D)
210 IF P = 0 THEN P = D
220 D = 0:COLOR 5,1
1000 PRINTP
1010 N = N - P
1020 PRINT"IL RESTE";N
1030 IF N = 0 THEN 2000
1040 PLAY"ACE":GOTO 60
2000 PRINT:COLOR 5,8
2010 IF D = 1 THEN PRINT"J'AI GAGNE":PLA
Y"CDEFG"
2020 IF D <>1 THEN PRINT"BRAVO":PLAY"AAA
"
```

Allumettes à plusieurs

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : logique (plusieurs joueurs).

Difficulté du jeu : assez difficile.

LE JEU :

Dans ce jeu pour plusieurs joueurs, le meneur de jeu devra, en début de partie, choisir un nombre d'allumettes, et indiquer au C128 le nombre de joueurs.

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

LE PROGRAMME :

Les lignes 10 à 40 permettent les différentes entrées.

Les lignes 70 à 100 réalisent les impressions.

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

La ligne 160 permet le retour au numéro du joueur de départ.

La ligne 150 réalise le test de fin, et la ligne 210 affiche le numéro du joueur gagnant (avec la règle prévue initialement).

POUR JOUER :

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

Le premier joueur ne pourra enlever qu'une allumette, le deuxième pourra en enlever 1 ou 2 ; le suivant 1, 2 ou 1, 2, 3 ou 4 selon le coup précédent, et ainsi de suite ; le retrait maximal autorisé est à chaque fois affiché par le C128.

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 restant qui seront ainsi éliminés les uns après les autres.

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

```

10 REM ALLUMETTES A PLUSIEURS
20 COLOR 0,7:COLOR 4,7:COLOR 5,2:GRAPHIC
  0,1
30 INPUT"NOMBRE TOTAL D'ALLUMETTES :";N
40 INPUT"NOMBRE DE JOUEURS :";J
50 F = 1
60 K = J
70 GRAPHIC 0,1
80 COLOR 5,K+2:PLAY"AC":PRINT"TOUR DU JO
UEUR :";K
90 PRINT"TOTAL RESTANT :";N
100 PRINT"MAXIMUM A ENLEVER :";F
110 INPUT"VOTRE COUP :";C
120 IF C = 0 OR C >F THEN PRINT"!!":GOTO
  110
130 N = N - C
135 IF N<0 THEN N=N+C:PRINT"!!":GOTO 110
140 F = C + C
150 IF N = 0 THEN 200
160 IF K = 1 THEN K = J+1
170 K = K - 1
180 GOTO 70
200 PRINT:COLOR 5,8
210 PRINT"LE JOUEUR NUMERO";K;" A GAGNE"
:PLAY"CDEFG"

```

Calendrier perpétuel

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : gadget publicitaire.

Difficulté du jeu : aucune.

LE JEU :

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

LE PROGRAMME :

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

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

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

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

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

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


```

10 REM CALENDRIER PERPETUEL
20 COLOR 0,7:COLOR 4,7:COLOR 5,8:GRAPHIC
  0,1
30 INPUT"POUR QUELLE ANNEE :";A
40 INPUT"POUR QUEL MOIS   :";M
50 D=INT(365.25*(A+(M<3)))+1
60 D=D+INT(30.6*(M+1-(M<3)*12))-INT((INT
  ((A+(M<3))/100)-7)*0.75)
70 D=D-7*INT(D/7)
80 M=31+(M=4 OR M=6 OR M=9 OR M=11)+(M=2
  )*3
90 PRINT
100 PRINT"DIM LUN MAR MER JEU VEN SAM"
110 PRINT:COLOR 5,9
120 FOR A = 1 TO M
130 PRINTTAB(4*(D+A-1));A;:TEMPO 24:PLAY
  "05T0U5C"
140 IF D+A-1=6 THENPRINT:D = D - 7
150 NEXT A

```

Bataille navale

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : réflexion.

Difficulté du jeu : facile.

LE JEU :

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

A chaque coup, on devra tirer en donnant les coordonnées H (horizontale) et V (verticale) de la case visée, et le C128 répond en affichant la distance séparant le navire invisible de la case visée. Mais, si l'on tire trop près, le navire risque de se recacher au hasard.

LE PROGRAMME :

La boucle I (30-50) dessine le terrain de jeu.

Les lignes 60-70 définissent l'emplacement du navire.

Les coordonnées du tir sont entrées aux lignes 80 et 100, et la ligne 150 affiche le coup sur l'écran. La distance est calculée en ligne 160, et la ligne 170 regarde si l'on a ou non fait mouche.

La ligne 220 intervient pour déplacer le navire si les coups sont rapprochés.

La ligne 230 permet l'affichage de la distance.

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

Elles sont innombrables, puisqu'aussi bien la bataille navale est l'un des jeux les plus répandus qui soit. On peut introduire deux joueurs, jouer contre le C128, mettre plusieurs navires, des sous-marins, etc.

```

10 REM BATAILLE NAVALE
20 COLOR 0,7:COLOR 4,7:COLOR 5,2:GRAPHIC
  0,1
25 PRINT"    0  1  2  3  4  5  6  7  8
9"
30 FOR I = 0 TO 9
40 PRINT:PRINTI;" . . . . .
. . ."
50 NEXT I
55 S = 1
60 X=INT(RND(1)*10)
70 Y=INT(RND(1)*10)
75 PRINT
80 INPUT"LIGNE### IIII";H$:H=VAL(H$)+1
90 PRINT"J";"#####";
100 INPUT"COLONNE### IIII";V$:V=VAL(V$)
110 PRINT"J";
120 FOR J = 1 TO H
130 PRINT"J"
140 NEXT J
150 PRINTTAB(3*V+3)S:PLAY"OST0U5D"
160 D=INT(SQR((X-H)*(X-H)+(Y-V)*(Y-V)))
170 IF D=0 THEN 250
180 S = S + 1
190 FOR J = H TO 10
200 PRINT:PRINT
210 NEXT J
220 IF D = 1 THEN PRINT"J
      PFFFF...      ":GOTO 60
230 PRINTTAB(25)"DISTANCE"D
240 PRINT"JJ":GOTO 80
250 PRINT"J"
260 PRINT"TROUVE EN"S;" COUPS":PLAY"CDEF
G"

```

Golf

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez facile.

LE JEU :

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

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

LE PROGRAMME :

La ligne 40 permet de définir la position du trou, et la ligne 50 dessine le terrain de jeu.

Le déplacement à réaliser est calculé en ligne 80.

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

La ligne 140 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 (20-220) ; 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 RETURN, la balle recommence à se déplacer ; elle peut alors soit s'arrêter avant le trou (auquel cas on recommence), soit tomber dedans (on passe alors au trou suivant), soit le dépasser : on devra alors frapper avec une force négative.

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

EXTENSIONS POSSIBLES :

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

```

10 REM GOLF
20 FOR S = 1 TO 18
30 COLOR 0,2:COLOR 4,2:COLOR 5,7:GRAPHIC
  0,1
35 X = 1
40 T = INT(29*RND(1)+9)
50 COLOR 5,6:PRINT"XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX"
55 COLOR 5,3:PRINT"XXXXXXXXTAB(T);" ":PRIN
T"XXXXXXXXTAB(X);" "●"
60 PRINT"XXXXXXXXXXTAB(15);" "
65 COLOR 5,1:PRINT"XXXXXXXXXXTROUS";INPUT"
  FORCE";R$
70 F = VAL(R$)
80 F = INT(F*0.16-(ABS(F)*(SQR(RND(1))+0
.1)*12)/1000)
90 IF F = 0 THEN PRINT"J":GOTO 65
100 FOR I = SGN(F) TO F STEP SGN(F)
110 X = X + SGN(F)
120 IF X<39 AND X>=1 THEN BEGIN:PRINT"XXX
XXTAB(X-SGN(F));" ";
122 PRINT"XXXXXXTAB(X)"●":TEMPO 32:PLAY"O
3T0U50D":BEND
125 IF X >=38 THEN X = 38
127 IF X<=1 THEN X = 1
130 NEXT I
140 IF X = T THEN 200
150 E = E + 1
160 GOTO 50
200 PRINTTAB(X)"J ":PRINTTAB(X)"●"
210 SLEEP 4
220 NEXT S
230 PRINT"J"
240 PRINT"18 TROUS EN";E;" COUPS":TEMPO
8:PLAY"T6U5CEG"

```

Solfège

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion.

Difficulté du jeu : facile.

LE JEU :

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

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

LE PROGRAMME :

La boucle I (40-70) dessine une portée.

La note est choisie en ligne 80 et affichée en lignes 100-110-120.

La ligne 130 affiche le score.

La ligne 140 affiche le nom de la clé (ici clé de sol) et vous interroge.

La ligne 150 s'occupe de la fin de la partie.

Les lignes 160 et 170 effectuent le décodage de la note entrée.

La ligne 190 contrôle le résultat.

POUR JOUER :

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

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

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

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

EXTENSIONS POSSIBLES :

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

```

10 REM SOLFEGE
20 S = 0 : P = 0
30 COLOR 0,2:COLOR 4,2:COLOR 5,3:PRINT"
000"
40 FOR I = 1 TO 5
50 PRINT"-----"
      "
60 PRINT
70 NEXT I
80 X = INT(RND(1)*12)
90 GOSUB 500
100 PRINTTAB(20)" I"
110 PRINTTAB(20)" I"
120 PRINTTAB(20)" "
130 PRINT"-----SCORE"S;" S
UR"P;" ESSAIS"
140 INPUT"CLE DE SOL ; NOTE ";R$
150 IF R$ = "0" THEN PLAY"CEG":END
160 Y = ( 1 AND R$="FA" ) + ( 2 AND R$="M
I" ) + ( 3 AND R$="RE" )
170 Y = Y + ( 4 AND R$="DO" ) + ( 5 AND R
$="SI" ) + ( 6 AND R$="LA" )
180 P = P + 1
190 IF Y = X OR Y = X - 7 THEN S=S + 1
200 GOTO 30
500 PRINT" ";
510 FOR I = 0 TO X
520 PRINT
530 NEXT I
540 RETURN

```

Tirage de dés

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : facile.

LE JEU :

Le programme proposé est surtout intéressant par son côté graphique puisqu'il permet de représenter, de façon très convenable, de 1 à 6 dés sur l'écran, en utilisant une technique de dessin un peu particulière.

Chaque réponse à la question NOMBRE DE DES provoque immédiatement un nouveau lancer du nombre de dés indiqué.

LE PROGRAMME :

La ligne 40 permet d'introduire le nombre de dés à tirer.

Les variables B\$ et C\$, en ligne 50, contiennent le dessin du fond.

La boucle I (60-100) et la ligne 70 dessinent le cadre du dé.

La ligne 80 réalise le tirage et la ligne 90 affiche les points en relation avec le tirage réalisé.

POUR JOUER :

On lance le programme par l'ordre RUN, et, à la question NOMBRE DE DES, on répond en frappant une touche de 1 à 6, puis RETURN. Les dés demandés sont alors représentés, et la question NOMBRE DE DES est de nouveau posée. Il suffira de taper RUN/STOP et RESTORE pour sortir du programme.

EXTENSIONS POSSIBLES :

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


```

10 REM TIRAGE DE DES
20 COLOR 0,15:COLOR 4,13:COLOR 5,1:GRAPH
IC 0,1
30 PRINT"Nombre de des a lancer ";
40 INPUT A$:IF A$<"1" OR A$>"6" THEN 20
50 SCNCLR 0:N = VAL(A$):B$="":C$=
""
60 FOR I = 1 TO N
70 COLOR 5,I+1:PRINT "TAB(6*I-4)
B$:PRINTTAB(6*I-4)C$:PRINTTAB(6*I-4)B$
80 D(I) = INT(RND(1)*6+1)
90 COLOR 5,1:PRINT "TAB(6*I-3)
D(I):SOUND 1,700*D(I)+1000,5
100 NEXT I
110 GOTO 30

```

Le jeu du 21

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu contre le C128.

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 C128 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 et 70 réalisent le contrôle de celui qui doit jouer (joueur ou C128).

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

Les lignes 90-100 calculent le total fait par le joueur et l'affichent.

Le total du C128 est calculé en ligne 180 et affiché par la ligne 190.

La ligne 210 contrôle la fin du jeu pour le C128.

Les lignes 250-260 affichent les résultats, et la ligne 270 cherche quel est le gagnant.

Le gain (ou la perte) sera affiché par la ligne 300, et le total par la ligne 320.

La ligne 340 contrôle la fin de partie.

POUR JOUER :

Taper RUN pour lancer le jeu.

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

Dès que l'on a terminé, on tape 0 et RETURN pour laisser jouer le C128 (qui tient ici le rôle de la banque).

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

EXTENSIONS POSSIBLES :

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

```

10 REM JEU DU 21
15 G = 0
20 COLOR 0,7:COLOR 4,7:COLOR 5,6:GRAPHIC
  0,1
30 T = 0
40 INPUT"VOTRE MISE ";M
50 PRINT"SAUFFAIRE 1 POUR TIRER , 0 POUR
  FINIR ";
60 INPUT R:PRINT"#####  "
70 IF R = 0 THEN 150
80 GOSUB 1000
90 T = T + X
100 PRINT"SAUFVOTRE TOTAL EST DE "T
110 GOTO 50
150 PRINT"LA BANQUE JOUE"
160 S = 0
170 GOSUB 1000
180 S = S + X
190 COLOR 5,1:PRINT"SAUFTOTAL DE LA BAN
QUE "S
200 SLEEP 1
210 IF S > T OR S > 17 THEN 250
220 GOTO 170
250 COLOR 5,2:PRINT"SAUFVOUS AVEZ "T
260 PRINT"LA BANQUE A "S:PLAY"05T5U5CE
  G"
270 IF S>21 OR (S<T AND T<=21) THEN 300
280 M = - M
300 PRINT"SAUFGAIN "M
310 G = G + M
320 PRINT"SAUFVOTRE TOTAL ACTUEL "G
330 INPUT"SAUFON CONTINUE ( O/N )";R$
340 IF R$ = "N" THEN PLAY"03T7CDEFG":END
350 GOTO 20
1000 X=INT( RND(1)*6 + 1):PLAY"05T9U5C"
1010 RETURN

```

Pair-impair

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : facile.

LE JEU :

C'est un jeu de hasard, pour un ou plusieurs joueurs. Le C128 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 60 tient le compte du nombre de parties, et les lignes 150, 200 et 230 comptabilisent les points marqués.

La parité du total est déterminée par les lignes 100 et 110 (la ligne 110 compare avec la réponse du joueur).

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

La ligne 430 provoque l'arrêt si l'on répond N à la question ON CONTINUE.

POUR JOUER :

On lance le programme par l'instruction RUN.

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

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

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

EXTENSIONS POSSIBLES :

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

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

```

10 REM PAIR IMPAIR
20 T = 0 : R = 0
30 GOSUB 1000
40 A = X
50 GOSUB 1000
60 R = R + 1
70 S = A + X
80 COLOR 0,2:COLOR 4,3:COLOR 5,7:GRAPHIC
   0,1
90 INPUT"TOTAL PAIR ( 0 ) OU IMPAIR ( 1
   )";N
100 P = S/2 - INT(S/2)
110 IF N = P OR N - 0.5 = P THEN 140
120 PRINT"FAUXFAUX ":PLAY"CCC"
130 GOTO 160
140 PRINT"EXACTEXACT":PLAY"CEG"
150 T = T + 1
160 INPUT"TOTAL PRESUMETOTAL PRESUME";N
170 IF N = S THEN 200
180 PRINT"PERDUPERDU                ":PLAY"GE
   C"
185 SLEEP 2
190 GOTO 400
200 T = T + 2:PLAY"IFA"
210 INPUT"BRAVOBRAVO ; DONNEZ UN
   DE";N
220 IF N <> A AND N<>S-A THEN 180
230 T = T + 3
240 PRINT"BRAVOBRAVO":PLAY"CEG":GOTO 185
400 PRINT"J"
410 PRINT"VOUS AVEZ UN TOTAL DE"T;" SUR"
   R
420 INPUT"ON CONTINUE ( O/N )";R$
430 IF R$ ="N" THEN PLAY"CEGB":END
440 GOTO 30
1000 X = INT(RND(1)*6 + 1)
1010 RETURN

```

Devine un chiffre

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

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

L'affichage est réalisé par les lignes 120 et 130.

Les lignes 150 à 180 calculent le gain dans les diverses configurations possibles.

La totalisation est effectuée en ligne 200 pour les gains, et en ligne 510 pour les pertes.

La variable R\$, testée en ligne 530, détermine la poursuite éventuelle du jeu.

POUR JOUER :

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

EXTENSIONS POSSIBLES :

On pourra facilement introduire des contrôles (mise minimale, 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 C128.

```

10 REM DEVINE UN CHIFFRE
20 G = 0
30 COLOR 0,9:COLOR 4,9:COLOR 5,13:GRAPHI
C 0,1
40 INPUT"VOTRE MISE";N
50 INPUT"DONNEZ UN CHIFFRE DE 1 A 6";
D
60 GOSUB 1000
70 A = X
80 GOSUB 1000
90 B = X
100 GOSUB 1000
110 C = X
120 PRINT"          A          B
C"
130 PRINT"    "A;"          ";B;"          "
;C
140 T = 0
150 IF D = A THEN T = T + N
160 IF D = B THEN T = T + N
170 IF D = C THEN T = T + N
180 IF T = 0 THEN 500
190 PRINT"VOUS GAGNEZ" T
200 G = G + T
210 GOTO 520
500 PRINT"PERDU          "
510 G = G - N
520 PRINT"TOTAL ACTUEL"G:INPUT"CON
TINUE( O/N )";R$
530 IF R$ ="N" THEN PLAY"CDEFG":END
540 GOTO 30
1000 X = INT(RND(1)*6 + 1):SOUND 1,3000*
X,20
1010 RETURN

```

Le jeu de Wytthof

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion (deux joueurs).

Difficulté du jeu : moyenne.

LE JEU :

On dispose d'une grille carrée de 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 20, 30 et 40 permettent de définir une position de départ, en dehors de la diagonale.

La ligne 60 et la boucle I (70-90) dessinent le terrain de jeu.

La ligne 120 imprime la position du point mobile.

Le test de fin est effectué en ligne 130.

Le changement de joueur est assuré par la ligne 180.

Les directions de déplacement disponibles, 1, 2 et 3, sont décodées par trois sous-programmes qui commencent aux lignes 250, 270 et 280.

POUR JOUER :

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

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

C'est ensuite au tour du joueur 2.

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

EXTENSIONS POSSIBLES :

On peut tout d'abord améliorer les contrôles (la sortie du terrain n'est pas interdite, par exemple), améliorer ensuite l'affichage (graphisme, couleur), étendre le jeu à plusieurs joueurs avec un terrain agrandi, introduire un monstre qui va essayer de capturer un des joueurs, prévoir un programme qui permettrait au C128 de jouer, etc.


```

10 REM LE JEU DE WYTHOF
20 X = 3*INT(RND(1)*5)+4
30 Y = INT(RND(1)*5)
40 IF X = Y THEN 30
50 COLOR 0,2:COLOR 4,1:COLOR 5,7:GRAPHIC
   0,1
60 PRINT"      0  1  2  3  4  5  6  7  8
9"
70 FOR I = 0 TO 9
80 PRINT"0";I;" . . . . .
. ."
90 NEXT I
100 H = 1
110 GOSUB 200
120 COLOR 5,11:PRINTTAB(X)CHR$(113)
130 IF X = 31 AND Y = 10 THEN PLAY"U5T8C
DEFG":END
140 COLOR 5,7:PRINT"XXXXXXXXXXXXXXXXXXXXX
XXXXXXXXX JOUEUR "H;
150 INPUT"Direction (1,2,3) ";D$:PRINT
"XXXXXXXXXX "
160 D = VAL(D$)
170 ON D GOSUB 250,270,280
180 H = 2 - ABS(H-1)
190 GOTO 110
200 PRINT"X";
210 FOR I = 1 TO Y
220 PRINT:PRINT
230 NEXT I
240 RETURN
250 X = X + 3:PLAY"U3T5C"
260 RETURN
270 GOSUB 250
280 Y = Y + 1:PLAY"U3T5G"
290 RETURN

```

Rebonds

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu d'adresse.

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

LE PROGRAMME :

La ligne 20 imprime les chiffres et les lettres en haut de l'écran.

Les lignes 30 et 40 définissent les positions initiales de la balle et de la cible.

Les lignes 70 et 100 effectuent leur impression.

Le symbole correspondant à la direction choisie est entré en ligne 110.

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

La ligne 180 imprime la nouvelle position.

Le changement de direction, lors de la rencontre d'un bord, est effectué par la ligne 160, par renvoi dans une partie du programme indépendante, située aux lignes 250 à 270.

La ligne 200 effectue l'impression de fin de partie.

Le nombre de rebonds est compté en ligne 260.

POUR JOUER :

On lance le programme par RUN.

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

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

On peut lancer une autre partie en tapant une touche quelconque, et s'arrêter à tout moment en tapant RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

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

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

```

10 REM REBONDS
15 COLOR 0,11:COLOR 4,11:COLOR 5,2:GRAPH
10 0,1
20 PRINT" 0123456789ABCDEFGHIJKLMNPOQRST
UVWXYZ"
30 X = INT(18*RND(1)+2)
40 Y = INT(18*RND(1)+2)
50 Z = X
60 GOSUB 500
70 PRINT"●"
80 Z = Y
90 GOSUB 500
100 PRINTTAB(39)"+"
110 INPUT"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/VOTRE CO
UP";R$
115 PRINT"§"
120 S = 0
130 T = X
140 FOR J = 1 TO 39
150 T = T - X / (ASC(R$)-ASC("0")+7*(R$="
A")+0.1)
160 IF T >=22 OR T <=0 THEN 250
170 Z = T : GOSUB 500
180 PRINTTAB(J)"●":SOUND 1,10000,8:FOR I
=1 TO 40:NEXT I
190 NEXT J
200 IF ABS(T-Y) < 3 THEN PRINT"SOBRAYO :
"S;" REBONDS":PLAY"CEG"
210 PRINT"§"APPUYER SUR - SHIFT -"
220 WAIT211,1:WAIT211,1,1
230 RUN
250 X = - X
260 S = S + 1
270 GOTO 190
500 PRINT"§"
510 FOR I = 1 TO Z
520 PRINT
530 NEXT I
540 RETURN

```

Jackpot

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : hasard (un joueur).

Difficulté du jeu : facile.

LE JEU :

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

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

LE PROGRAMME :

La boucle I (40-60) permet le dessin de la machine, la ligne 70 affichant les emplacements (vides) des chiffres.

La mise sera entrée en ligne 80, et testée en ligne 90 : si on mise 0, la partie est terminée (départ vers la ligne 2000).

Les lignes 100, 120 et 140 permettent le tirage des 3 chiffres, en appelant un sous-programme qui se trouve en ligne 1000.

Le résultat est affiché par la ligne 150, puis testé en ligne 160, et en ligne 310, si l'on a gagné.

POUR JOUER :

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

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

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 SHIFT et ainsi de suite.

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

Le reste 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.

```

5 REM JACKPOT
7 COLOR 0,2:COLOR 4,2:COLOR 5,7:GRAPHIC
0,1
10 INPUT"POT DE DEBUT";P
30 PRINT"20000"
40 FOR I = 1 TO 3
50 PRINTTAB(12)"████████████████████"
60 NEXT I
70 PRINT"20000";TAB(14);"  ";TAB(19);
"  ";TAB(24);"  "
80 INPUT "20000VOTRE MISE";M
90 IF M = 0 OR M >= P THEN 2000
100 GOSUB 1000
110 A = X
120 GOSUB 1000
130 B = X
140 GOSUB 1000
150 PRINT"20000";TAB(14);A;TAB(19);B;T
AB(24);C
160 IF A = B AND A = X THEN 300
170 N = - M
180 PRINT"20000VOTRE MISE ";M;"
RESULTAT "N
190 PRINT"20000APPUYER SUR - SHIFT -":WAIT
211,1:WAIT 211,1,1
200 P = P + N
210 GOTO 30
300 N = M * 50
310 IF A = 7 THEN N = M * 500
320 GOTO 180
1000 X=INT(RND(1)*9+1):SOUND 1,400*X,20
1010 RETURN
2000 PRINT"J":PLAY"CDEFG"
2010 PRINT"VOUS AVEZ ";P
2020 PRINT"A BIENTOT"

```

Couloirs

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

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

LE PROGRAMME :

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

La ligne 100 attend votre choix.

Le décodage du chemin est fait par le C128 en utilisant trois variables logiques I, J et K, représentant les états des trois tourniquets (lignes 130 à 270).

POUR JOUER :

Après la frappe de RUN, le couloir est dessiné sur l'écran, les tourniquets étant visibles, et l'on devra taper 1 ou 2 en réponse à la question du C128. 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 C128.

EXTENSIONS POSSIBLES :

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

```

10 REM COULOIRS
20 COLOR 0,13:COLOR 4,13:COLOR 5,3:GRAPH
IC 0,1
30 PRINT"0"      "
40 PRINT"0"      B      "      X
2"
50 PRINT"0"      "      X      "
60 PRINT"0"      A      "      X
1"
70 PRINT"0"      "
80 I = 0 : J = 0 : K = 0
90 R$ = "" : T$= ""
100 COLOR 5,1:INPUT"XXXXXXXXXXXXXXXXXENTREE
1 OU 2 :###" ;C$
110 IF C$ ="" OR C$ >= "3" OR C$ < "1" O
R LEN(C$) <> 1 THEN 100
120 R$ = R$ + C$
130 PRINT"0";R$
140 IF C$ ="1" THEN 190
150 J = NOT(J)
160 IF J <>0 THEN 250
170 S$ = "B"
180 GOTO 220
190 I = NOT(I)
200 IF I <>0 THEN 250
210 S$ = "A"
220 T$ = T$ + S$
230 PRINT"XXXXXXXXXXXXXXXXXSORTIE :";S$
:PRINT"0";T$:PLAY S$
240 GOTO 100
250 K = NOT(K)
260 IF K <>0 THEN 170
270 GOTO 210

```

Echiquier

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : graphisme.

Difficulté du jeu : variable, selon utilisation.

LE JEU :

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

LE PROGRAMME :

Les lignes 30, 40 et 50 définissent trois variables A\$, B\$ et C\$ qui contiennent les éléments de dessin. La ligne 60 imprime les lettres horizontales et la première ligne du dessin.

La boucle I (80-160) permet l'impression, le motif à imprimer étant affiché ou non selon la valeur de $(I + J)/2$; les lignes 110 et 120 réalisent l'impression.

POUR JOUER :

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

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

EXTENSIONS POSSIBLES :

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


```

10 REM ECHIQUIER
20 COLOR 0,16:COLOR 4,16:COLOR 5,2:GRAPH
IC 0,1
30 A$ ="      "
40 B$ ="  3  "
50 C$ ="  3  "
60 PRINT"      A  B  C  D  E  F  G  H"
80 FOR I = 1 TO 8
90 FOR J = 1 TO 8
100 IF (I+J)/2 = INT((I+J)/2) THEN 140
110 COLOR 5,3:PRINTTAB(3*J);A$:COLOR 5,1
:PRINTI;
120 COLOR 5,3:PRINTTAB(3*J);B$:PRINTTAB(
3*J);C$;
130 PRINT"TTT"
140 NEXT J
150 PRINT"  "
160 NEXT I:COLOR 5,7

```

Le jeu de la fève

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : réflexion (deux joueurs).

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

LE PROGRAMME :

La ligne 20 et la boucle I (30-50) dessinent le terrain de jeu.

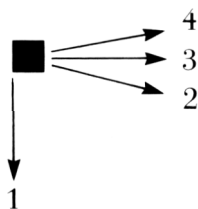
La ligne 100 réalise le test de fin de partie.

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

Les lignes 130 et 140 assurent le changement de joueur.

POUR JOUER :

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



Le joueur 1 entre son déplacement, et tape RETURN.

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

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

EXTENSIONS POSSIBLES :

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

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

```

10 REM LE JEU DE LA FEVE
15 COLOR 0,16:COLOR 4,16:COLOR 5,7:GRAPH
IC 0,1
20 PRINT"      0  1  2  3  4  5  6  7  8
9"
30 FOR I = 0 TO 9
40 COLOR 5,7:PRINT"0";I;:COLOR 5,2:PRINT
" . . . . . ."
50 NEXT I
60 X = 5 : Y = 1
70 J = 1
80 GOSUB 500
90 COLOR 5,3:PRINTTAB(X)"*":COLOR 5,7
100 IF X = 32 AND Y = 10 THEN 200
110 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXJOUVEUR"
J;:INPUT" DIRECTION (1,2,3,4)###: ";I$
115 I = VAL (I$):SOUND 1,3000*I,10
120 ON I GOSUB 1000,2000,3000,4000
130 J = J + 1
140 IF J = 3 THEN 70
150 GOTO 80
200 PRINT"X LE JOUEUR"J;" A GAGNE
"

210 PLAY"CEG":END
500 PRINT"X";
510 FOR I = 1 TO Y
520 PRINT:PRINT
530 NEXT I
540 RETURN
1000 Y = Y + 1
1010 RETURN
2000 GOSUB 1000
3000 X = X + 3
3010 RETURN
4000 GOSUB 3000
4010 Y = Y - 1
4020 RETURN

```

Jeu de l'oie aléatoire

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : hasard (deux joueurs).

Difficulté du jeu : facile.

LE JEU :

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

La règle du jeu est sans grand intérêt, puisque les joueurs n'interviennent à aucun moment dans le déroulement de la partie, qui est entièrement gérée par le C128 ; il suffira d'appuyer sur une touche après chaque lancer de dé, et de regarder ce qui se passe.

LE PROGRAMME :

La ligne 60 permet d'appeler le sous-programme 1000 à 1100 qui réalise l'affichage du terrain de jeu, le contrôle de la position des joueurs et l'affichage de cette position. Cet affichage sera d'ailleurs refait en cours de partie, pour tenir compte du déplacement réellement effectué (qui n'est pas nécessairement celui indiqué par le dé).

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

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

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

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

POUR JOUER :

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

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

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

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

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

On peut aussi s'arrêter à tout moment en tapant RUN/STOP et RE-STORE.

EXTENSIONS POSSIBLES :

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

```

10 REM JEU DE L'OEIE ALEATOIRE
15 COLOR 0,2:COLOR 4,2:COLOR 5,14:GRAPHI
C 0,1
20 X = 0
30 Y = 0
40 J = 1
50 I = X
60 GOSUB 1000
70 J = ABS(J - 1)
80 I = INT(RND(1)*6 + 1)
90 IF X = 35 OR Y = 35 THEN PRINT"XXXXXXXX
XXXXXXXX BRAVO " :PLAY"CEG":END
100 SOUND 1,5000*J+5000,8:PRINT"XXXXXXXX
JOUEUR";J+1;"  ";I
110 PRINT"APPUYER SUR - SHIFT -":WAIT 21
1,1:WAIT 211,1,1
120 GOSUB 1000
130 I = INT(RND(1)*4)
140 F = INT(RND(1)*4 + 1)
150 ON F GOSUB 200,300,400,500
160 GOTO 60
200 RETURN
300 I = - I
310 RETURN
400 I = I + I
410 RETURN
500 IF J=0 THEN X=INT(RND(1)*30):RETURN
510 Y = INT(RND(1)*30):RETURN
1000 IF J=0 THEN X=X-I*(X+I<=35)
1010 Y = Y - J*I*(Y + I <= 35)
1015 COLOR 5,1:PRINT"XXXXXXXX123456789ABCDE
FGHIJKLMNOPQRSTUVWXYZ"
1020 FOR I = 0 TO 35
1030 COLOR 5,11:PRINTTAB(I)"X "
1040 COLOR 5,6:PRINTTAB(I)"X ■"
1050 PRINT"III"
1060 NEXT I
1070 COLOR 5,1:PRINTTAB(X)"1"
1080 PRINTTAB(Y)"2"
1090 SLEEP 2
1100 RETURN

```


Niveau 2

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

Dimensionnement d'un tableau Recherche d'un élément dans un tableau et dans une chaîne Les instructions DATA, READ et RESTORE

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 pourrez définir 30 variables appelées PREMIER JOUR, DEUXIEME JOUR, etc. Il est plus facile et plus rationnel d'utiliser une boucle I, I variant de 1 à 30, et une variable R (I) pour les recettes, et D (I) pour les dépenses ; ce faisant, vous avez, sans le savoir, utilisé une structure de tableau à une dimension ; ceci ne pourra être admis en Basic C128 que si vous avez auparavant prévenu que vous alliez utiliser un tel tableau, en donnant une instruction DIM R (30) pour les recettes, et DIM D (30) pour les dépenses.

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

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

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

Dans le Basic du C128, il n'est pas obligatoire de dimensionner un tableau qui contient moins de 10 éléments.

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

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

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

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

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

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

LE TRAITEMENT DE DONNEES MULTIPLES INSTRUCTIONS DATA, READ ET RESTORE

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

Ainsi, on pourra écrire :

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

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

Dé truqué

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

Il utilise le générateur aléatoire du C128 supposé de qualité convenable (on pourra aisément utiliser tout autre générateur, si besoin est). La boucle I (40-100) s'occupe de l'entrée de chacune des probabilités, avec vérification à chaque entrée et sur le total (ligne 70) ; chaque probabilité est conservée dans une variable indicée E (I) : ligne 50 du programme.

Le lancer de dé truqué est effectué à l'aide d'une autre boucle I (140-160), la ligne permettant le truchage se trouvant en 150. Un nouveau tirage pourra être effectué sur pression de la touche SHIFT (ligne 185).

POUR JOUER :

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

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

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 truchage en évidence...

```

10 REM DE TRUQUE
15 COLOR 0,2:COLOR 4,2:COLOR 5,7:GRAPHIC
  0,1
20 P = 0
30 DIM E(6)
40 FOR I = 1 TO 6
45 PRINT"*****";TAB(17);"      "
50 PRINT"*****PROBABILITE DU DE"I;:INPUT E
(I)
60 P = P + E(I)
70 IF E(I) < 0 OR E(I) > 1 OR P > 1.1 TH
EN 200
80 PRINT"*****";TAB(6*I-6);"DE"I;
90 PRINT"*****";TAB(6*I-6);E(I)
100 NEXT I
110 IF P < 0.9 THEN PLAY"G":RUN
120 SLEEP 2
130 N = INT(RND(1)*6 + 1)
140 FOR I = 1 TO 6
150 IF N = I AND RND(1) > E(I) THEN 130
160 NEXT I
170 PRINT"*****LE TIRAGE : "N;"      "
PLAY"CGE"
175 SLEEP 5
178 PRINT"]"TAB(8);"      "
180 PRINT"*****FAIRE - SHIFT - POUR UN AUTRE
TIRAGE"
185 WAIT 211,1:WAIT 211,1,1
190 GOTO 130
200 PRINT"IMPOSSIBLE : 0 <= P <= 1":PLAY
"CCC"
210 P = P - E(I)
220 SLEEP 5
230 PRINT"]
"
240 GOTO 50

```

Visions

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : dessin.

Difficulté du jeu : facile.

LE JEU :

Le C128 vous permet, pour la première fois, d'entrer dans son intimité en vous laissant entrevoir sa propre représentation du monde ! Vous allez, en effet, lui proposer des mots, n'importe lesquels, et il va vous répondre en vous proposant une représentation graphique de ce que représente ce mot pour lui. Le jeu sera beaucoup plus amusant si vous le pratiquez à plusieurs, et vous pourrez quelquefois avoir des surprises.

LE PROGRAMME :

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

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

POUR JOUER :

On débute l'exécution du programme par RUN, ce qui amène l'affichage de la question QUEL DESSIN. On répond en entrant un mot quelconque qui sera transformé en dessin par le C128 dès que l'on aura frappé RETURN.

Pour recommencer, il suffira de frapper la touche SHIFT, et RUN/STOP et RESTORE pour finir.

EXTENSIONS POSSIBLES :

La même idée peut être reprise en introduisant des éléments de symétrie dans les dessins. 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 C128 deviendra de plus en plus savant. On peut aussi réaliser un jeu à deux joueurs dans lequel C128 vous demande de trouver le mot qui correspond à un certain dessin, etc.

```

10 REM VISIONS
20 COLOR 0,16:COLOR 4,16:COLOR 5,2:GRAPH
   IC 0,1
30 INPUT"QUEL DESSIN";R$
35 PRINT:PRINT
40 FOR I = 1 TO LEN(R$)
50 N = 9*ASC(MID$(R$,I,1)) - 340
60 J = 1
70 S = N/2 - INT(N/2)
80 N = INT(N/2):SOUND 1,50*N+2000,5
90 IF S <> 0 THEN COLOR 5,J+2:PRINTTAB(J
   + 15)"▣ ▢";
100 J = J + 1
110 IF J < 10 THEN 70
115 PRINT
120 NEXT I
125 PRINT:PRINT
130 PRINT"APPUYER SUR - SHIFT -"
140 WAIT 211,1:WAIT 211,1,1
150 GOTO 20

```

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

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

La ligne 110 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 (170-210), 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 120-130.

Les tests pour le nombre d'essais et pour le changement de joueur sont effectués aux lignes 220 et 240.

POUR JOUER :

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

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

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

Les joueurs suivants procéderont de la même façon.

EXTENSIONS POSSIBLES :

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

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

```

10 REM 421
15 COLOR 0,14:COLOR 4,14:COLOR 5,1:GRAPH
IC 0,1
20 INPUT"NOMBRE DE JOUEURS";N
25 P = 1
30 I = 1
40 K = 0
50 PRINT"3"
60 R$="ABC"
70 GOTO 170
80 K = K + 1
90 PRINT"XXXXXXXXXXXXXXXXXXXX421"
100 PRINT"_____"
110 PRINT"XXXXJOUEUR NO."I;" ESSAI"K;" PA
RTIE"P
120 PRINT"XXXX      "A;"      "B;"
      ";C
130 PRINT"XX      A      B
      C"
135 R$="":PRINT"XXXX      "
140 INPUT"3ABC";R$
150 IF R$ = "" THEN 230
160 R$=R$ + "DDD"
170 FOR J = 1 TO 3
180 GOSUB 4000
185 X$ = MID$(R$,J,1)
190 X = ASC(X$)-64
200 ON X GOSUB 1000,2000,3000,4000
210 NEXT J
220 IF K < 3 THEN 80
230 I = I + 1
240 IF I <= N THEN 40
245 P = P + 1
250 GOTO 30
1000 A = D
1010 RETURN
2000 B = D
2010 RETURN
3000 C = D
3010 RETURN
4000 D=INT(RND(1)*6+1):SOUND 1,2000*D,8
4010 RETURN

```

L'électricien fou

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez difficile.

LE JEU :

Un immeuble comportant 10 fenêtres, numérotées de 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 30, contiendra l'état (allumé ou éteint) des 10 fenêtres.

La boucle I (50-70) et l'autre boucle I (100-160) dessinent l'immeuble dans son état initial.

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

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

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

POUR JOUER :

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

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

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

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

EXTENSIONS POSSIBLES :

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


```

10 REM L'ELECTRICIEN FOU
20 M = 0
30 DIM A(10)
40 COLOR 0,10:COLOR 4,12:COLOR 5,5:GRAPH
   IC 0,1
50 FOR I = 1 TO 7
60 PRINTTAB(5)CHR$(30)"  "
   "
70 NEXT I
80 PRINT"  "
90 T = 0
100 FOR I = 1 TO 10
110 T = T + A(I)
120 PRINT"  ";TAB(2*I+5);CHR$(30)"  "
140 IF A(I)=0 THEN COLOR 5,8:PRINT"  "
   ;TAB(2*I+5);"  ":SOUND 1,1500*I,8
150 PRINT"  ";TAB(2*I+5);RIGHT$(STR$(I
   -1),1)
160 NEXT I
170 IF T = 10 THEN 310
180 INPUT"  INTERRUPTEUR NUMERO
   ";N
190 N = N + 1
200 M = M + 1
210 IF A(N) = 1 THEN 290
220 A(N) = 1
230 R = TAN(RND(1) + N/RND(1) -N) - SIN(
   RND(1)/N) + 336*SIN(8*N)
240 N = INT(10*(R-INT(R)))
250 IF N = 0 THEN N = INT(RND(1)*10 + 1)
260 IF A(N)= 1 THEN 290
270 A(N) = 1
280 GOTO 90
290 A(N) = 0
300 GOTO 230
310 PRINT"  TROUVE EN"M);" ESSAIS":PLAY
   "CDEFG"

```

La course au 20

REFERENCES :

Difficulté de programmation : facile.

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

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

La boucle I (30-60) dessine 4 rangées de 20 étoiles.

La variable F, dimensionnée en ligne 80, servira à enregistrer le nombre d'étoiles prises dans chaque rangée.

La variable T contiendra le nombre total d'étoiles enlevées (ligne 70).

Les lignes 117 et 120 vont contrôler qu'il reste suffisamment d'étoiles sur la ligne indiquée.

La boucle J (160-180) supprime les étoiles ; la ligne 230 teste la fin de partie.

POUR JOUER :

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

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

EXTENSIONS POSSIBLES :

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

```

10 REM LA COURSE AU 20
15 COLOR 0,8:COLOR 4,8:COLOR 5,3:GRAPHIC
  0,1
20 PRINT:PRINT
30 FOR I = 1 TO 4
40 PRINTI;" *****"
50 PRINT
60 NEXT I
70 T = 0
80 DIM F(4)
90 B = 1
100 COLOR 5,1:PRINT"XXXXXXXXXXXXJOUEUR"
B:SOUND 1,5000*B,8
110 INPUT"X"LINE"X"X"X"X"X";L:PRINT"
  "
115 INPUT"O"CASES";C
117 IF L>4 THEN PLAY"CCC":GOTO 100
120 IF C+F(L)>20 THEN PLAY"CCC":GOTO 100
130 F(L) = F(L) + C
140 T = T + C
145 PRINT"X"
150 FOR I = 1 TO 4
160 FOR J = F(I)-C TO F(I)
170 IF I=L THEN PRINTTAB(J + 4);" ";:SOU
ND 1,15000,4
180 NEXT J
190 PRINT:PRINT
200 NEXT I
210 B = B + 1
220 IF B = 3 THEN B = 1
230 IF T < 80 THEN 100
240 PRINT"XXXXXXXXXLE JOUEUR";B;" GAGNE":PL
AY"CDEFG"

```

Dessin

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : dessin.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

Une table A\$ est dimensionnée à 25 en ligne 30, pour contenir chaque ligne de dessin (il y a 25 lignes d'affichage). Le codage du dessin est effectué dans les lignes concernées, en remplissant chacune des variables A\$ (2) selon le code retenu (voir plus loin).

Les lignes 40 à 120 contiennent le dessin codé.

La boucle I (130-200) assure le passage d'une ligne à la suivante.

La boucle N (140-180) réalise le décodage de A\$ (I).

La ligne 160 détermine les positions de la tête d'écriture.

La ligne 170 réalise l'impression.

POUR JOUER :

Le dessin devra auparavant être défini dans des variables A\$ (I), en utilisant les 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 est fait en incrémentant I d'une unité.

EXTENSIONS POSSIBLES :

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

```

10 REM DESSIN
20 COLOR 0,14:COLOR 4,14:COLOR 5,3:GRAPH
IC 0,1
30 DIM A$(25)
40 A$(10)="A"
50 A$(11)="A K I T I"
60 A$(12)=A$(11):A$(13)=A$(11):A$(14)=A$(
11):A$(15)=A$(11):A$(16)=A$(11)
70 A$(17)="3"
80 A$(18)="3 T I"
90 A$(19)=A$(18):A$(20)=A$(18):A$(21)=A$(
18)
100 A$(22)="3"
110 A$(23)="8 C J N"
120 A$(24)="9 B K M"
130 FOR I = 1 TO 25
140 FOR N = 1 TO LEN(A$(I))
145 IF A$(I)="" THEN 200
150 P = ASC(MID$(A$(I),N,1))
160 IF P < 91 THEN PRINTTAB(P-48+7*(P>64
)): GOTO 180
170 PRINTMID$(A$(I),N,1);
180 NEXT N
190 PRINT
200 NEXT I
210 PRINT"§":PLAY"CDEFG"

```

Poker

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu contre le C128.

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 C128. 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 C128 de jouer.

LE PROGRAMME :

Les lignes 30 et 40 définissent les cartes. Une variable N est dimensionnée à 15, en ligne 50, pour recevoir 10 cartes joueur et 5 cartes C128.

Le tirage sera effectué par une boucle I (60-110) et le contrôle par une boucle J (80-100).

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

Une autre boucle I (160-180) réalise le deuxième tirage du joueur, la ligne 200 permettant le deuxième affichage, et la boucle I (500-570) où l'on entre directement après, réalisera l'affichage du jeu machine.

POUR JOUER :

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

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

Par exemple, s'il tape 235 et RETURN, son nouveau jeu conservera les cartes 1 et 4. Son jeu sera alors affiché ainsi que celui du C128, ce dernier n'ayant pas, dans ce programme, la possibilité de reprendre des cartes.

Il suffira ensuite de taper une touche quelconque pour démarrer une autre partie.

EXTENSIONS POSSIBLES :

Elles seront, bien sûr, très nombreuses, depuis un affichage graphique en couleur des cartes jusqu'à l'amélioration du jeu du C128, l'introduction des règles sur les paris et même, pourquoi pas, des possibilités de bluff.

```

10 REM POKER
20 COLOR 0,8:COLOR 4,8:COLOR 5,12:GRAPHI
C 0,1
30 V$="7 8 9 10VADAR0AS"
40 C$="TRCAC0PI"
50 DIM N(15)
60 FOR I = 1 TO 15
70 N(I)=INT(RND(1)*32)
75 IF I = 1 THEN 110
80 FOR J = 1 TO I-1
90 IF N(I) = N(J) THEN 70
100 NEXT J
110 NEXT I
120 K = 0 : P = 0
130 PRINT"XXXXXXXXXXXXXXXXXXXXCARTES REF
USES"
140 GOSUB 500
150 IF R$ ="0" THEN END
160 FOR I = 1 TO LEN(R$)
170 N(VAL(MID$(R$,I,1))) = N(I+5)
180 NEXT I
190 PRINT"3"
195 P = 1
200 GOSUB 500
210 K = K + 10
500 FOR I = 1 TO 5
510 J=2*INT(N(I+K)/8)+1
520 F=2*(N(I+K)-8*INT(N(I+K)/8))+1
525 R$ = MID$(V$,F,2)
530 PRINT"0":FOR S=1 TO K:PRINT : NEXT S
540 PRINTTAB(7*I-4);"33"I:PRINT:PRINT
550 PRINTTAB(7*I-5);"■";R$;"0";MID$(C$,
J,2):SOUND 1,400*N(I+K),8
560 PRINTTAB(7*I-7);" ————"
570 NEXT I
575 IF P = 1 THEN P = 0 : RETURN
580 IF K = 10 THEN PRINT"NOFAIRE - SHIFT
- POUR CONTINUER":WAIT211,1 : RUN
590 PRINT"XXXXXXXXXXXXXXXXXXXX";TAB(16);
:INPUTR$
600 RETURN

```

Le jeu de saute-mouton

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

La boucle I (50-80) initialise une table A (I), qui contient les positions des moutons. La boucle I (100-150) dessine le terrain de jeu et les moutons.

Les lignes 160 et 170 s'occupent du texte et des entrées.

La ligne 180 contrôle la validité du coup. Le changement de joueur est réalisé en ligne 210.

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

On commencera par rajouter un contrôle du joueur gagnant.

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

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


```

10 REM LE JEU DE SAUTE-MOUTON
20 COLOR 0,16:COLOR 4,16:COLOR 5,1:GRAPH
   IC 0,1
30 DIM A(8)
40 A(1) = 0 : A(8) = 0
50 FOR I = 1 TO 3
60 A(1 + I) = 1
70 A(4 + I) = -1
80 NEXT I
90 H = 1
100 FOR I = 1 TO 8
110 COLOR 5,9:PRINT"#####TAB(4*I);I
120 PRINT"#####TAB(4*I+1);" "
130 IF A(I)=1 THEN COLOR 5,7:PRINT"#####
   #TAB(4*I+1);"# "
140 IF A(I)=-1 THEN COLOR 5,3:PRINT"#####
   #TAB(4*I+1);"# "
145 PRINT"#"
150 NEXT I
155 J = 1 - (H = -1)
160 COLOR 5,3-4*(J=1):PRINT"#####J
   QUEUR"J:SOUND 1,5000*J,8
170 COLOR 5,2:INPUT"#DEBUT , FIN      ###
   ###";D,F
180 IF A(D) <> H OR A(F) <> 0 THEN 160
190 A(D) = 0
200 A(F) = H
210 H = -H
220 GOTO 100

```

Lettres

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

La règle est fortement inspirée d'un jeu de lettres très connu en France, pratiqué notamment dans une émission télévisée (les chiffres et les lettres).

Le C128 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 (60-90) choisit dans l'alphabet entre 7 et 12 lettres, en fonction de la variable aléatoire L, tirée en ligne 30. Pour éviter d'avoir trop de consonnes, on utilise en ligne 80 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 100-140 demande à chacun des joueurs la longueur de son mot. Les lignes 150 et 160 comparent les résultats et totalisent le score de chaque joueur, affiché par la ligne 170.

POUR JOUER :

On lance l'exécution du jeu avec RUN.

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

Pour continuer la partie, il suffira d'appuyer sur la touche SHIFT.

On devra faire RUN/STOP et RESTORE pour finir.

EXTENSIONS POSSIBLES :

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

On pourra aussi entrer non point la longueur du mot, mais le mot lui-même, l'adversaire devant alors répondre à la question CORRECT OU NON ? On peut aussi étendre le jeu à plus de deux joueurs. Et pourquoi ne pas essayer de définir un jeu contre le C128, auquel on aurait auparavant entré une liste de mots assez longue, dont il afficherait les lettres dans le désordre, avec éventuellement quelques lettres de complément choisies au hasard ?

```

10 REM LETTRES
15 COLOR 0,9:COLOR 4,9:COLOR 5,7:GRAPHIC
  0,1
20 S = 0 : J = 0
25 PRINT"J"
30 L = INT(RND(1)*6 +7)
40 A$ = "AEIOUEA"
50 PRINT"SCORELETTRES"
60 FOR I = 1 TO L
70 COLOR 5,2:PRINT"SCORE"TAB(7+2*I);CHR$(
  INT(RND(1)*26+65))
75 SOUND 1,1000*INT(RND(1)*9+1),5
80 IF RND(1) < 0.4 THEN PRINT"SCORE"TAB(
  7+2*I);MID$(A$,INT(RND(1)*7+1),1)
90 NEXT I
100 I = 1:COLOR 5,7
110 PRINT"SCOREJOUEUR"1;:INPUT";LONGUE
  UR";L
115 PRINT"J"TAB(18);"      "
120 IF I= 1 THEN P = L
130 I = I + 1
140 IF I < 3 THEN 110
150 S = S + (P AND P > L)
160 J = J + (L AND L > P)
170 COLOR 5,3:PRINT"SCORESCORES : J
  QUEUR 1 :\"S\";JOUEUR 2 :\"J
180 PRINT"SCOREPOUR CONTINUER FAIRE - SHIFT
  _"
190 WAIT211,1
200 GOTO 25

```

Machinamot

REFERENCES :

Difficulté de programmation : facile.

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

Difficulté du jeu : aucune.

LE JEU :

Ce très court programme permet au C128 de faire la démonstration de ses capacités à manipuler des chaînes de caractères : on part simplement d'un préfixe et d'un suffixe d'origine grecque, que l'on va assembler au hasard, on tombera bien sûr, de temps à autre, sur des mots absurdes ou même légèrement équivoques.

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

LE PROGRAMME :

Deux variables A\$ et B\$ sont dimensionnées à 15 en ligne 30 et 40 : l'une contiendra les préfixes, l'autre les suffixes (si l'on désire rajouter des possibilités, il suffira de modifier cette limitation à 15).

La boucle I (50-70) lit les lignes DATA (160-170) concernant les préfixes et la boucle J (80-100) lit les lignes des suffixes (180-190).

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

La ligne 120 effectue le choix d'un suffixe au hasard.

Le mot obtenu en ligne 130 est affiché par la ligne 140, puis on recommence (ligne 150).

POUR JOUER :

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

Taper RUN/STOP et RESTORE pour sortir du programme.

EXTENSIONS POSSIBLES :

On pourra très facilement augmenter la liste des mots disponibles.

On pourra aussi utiliser un procédé analogue pour faire confectionner des phrases à l'ordinateur, lui faire inventer des poèmes, etc. On pourra aussi transformer le programme initial en un jeu de société qui consisterait, par exemple, à prendre des paris sur le prochain mot qui va apparaître. Et confier, éventuellement, la gestion des paris au C128...

```

10 REM MACHINAMOT
20 COLOR 0,15:COLOR 4,15:COLOR 5,1:GRAPH
   IC 0,1
30 DIM A$(15)
40 DIM B$(15)
50 FOR I = 1 TO 15
60 READ A$(I)
70 NEXT I
80 FOR I = 1 TO 15
90 READ B$(I)
100 NEXT I
110 I = INT(RND(1)*15 + 1)
120 J = INT(RND(1)*15 + 1)
130 C$ = A$(I) + B$(J)
140 COLOR 5,INT(RND(1)*14+1):PRINTC$:SOU
ND 1,1000*INT(RND(1)*9+1),5
145 SLEEP 3
150 GOTO 110
160 DATA ANDRO,COSMO,MICRO,HELIO,PEDO,PO
RNO,PALEO
170 DATA MACRO,HYDRO,STENO,PATHO,OLEO,MY
THO,HOMEQ,OCTO
180 DATA CRATE,PHILE,PHOBE,TROPE,GYNE,MA
NE,NAUTE,PHAGE,FUGE
190 DATA PATHE,CARIE,DERME,SCOPE,VORE,PH
ONE

```

Espion

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu de réflexion.

Difficulté du jeu : facile.

LE JEU :

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

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

LE PROGRAMME :

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

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

La boucle I (70-140) est utilisée soit pour effectuer le codage, soit pour effectuer le décodage.

La ligne 90 appelle le générateur aléatoire.

La ligne 100 réalise la transposition des lettres.

Les lignes 110 et 120 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 RETURN.

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

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

EXTENSIONS POSSIBLES :

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

```

10 REM ESPION
20 COLOR 0,2:COLOR 4,2:COLOR 5,7:GRAPHIC
  0,1
30 INPUT"VOTRE CODE";N
40 A = 1/N
50 INPUT"VOTRE MESSAGE";A$
60 INPUT"VOULEZ-VOUS (C)ODER OU (D)ECOD
ER";R$
65 B$=""
70 FOR I = 1 TO LEN(A$)
80 T = ASC(MID$(A$,I,1))
90 GOSUB 1000
100 T = T - A*((R$="C")-(R$="D"))
110 IF T >90 THEN T = T - 26
120 IF T < 65 THEN T = T + 26
130 B$ = B$ + CHR$(T)
140 NEXT I
150 PRINT"RESULTAT :";B$
160 PLAY"CDEFG":END
1000 A = (A +  $\pi$ ) $\uparrow$ 8
1010 A = A - INT(A)
1020 A=INT(26*A+1):SOUND 1,500*A+500,4
1030 RETURN

```

Accrochez les wagons

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : solitaire.

Difficulté du jeu : facile.

LE JEU :

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

On tapera, par exemple : BBGDHHH.

LE PROGRAMME :

Les lignes 20 à 50 dessinent la locomotive et le wagon.

La position initiale de la locomotive est définie par les lignes 70 et 80, celle du wagon par les lignes 90 et 100.

La ligne 110 évite qu'ils soient initialement trop proches l'un de l'autre.

Les lignes 130 à 195 réalisent l'affichage du wagon et de la locomotive.

Le test de fin est en ligne 200.

Les mouvements à effectuer sont entrés dans une variable alphanumérique R\$, en ligne 220, et sont décodés dans une boucle I (250-280) pour être exécutés en ligne 290.

POUR JOUER :

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

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

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

On obtiendra alors l'affichage du score.

EXTENSIONS POSSIBLES :

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

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


```

10 REM ACCROCHEZ LES WAGONS
20 D$="  "
30 A$="  "
40 B$="  "
50 C$="O O"
60 N = 0:TRAP 210
70 LY=INT(RND(1)*33)
80 LX=INT(RND(1)*20) + 1
90 WY=INT(RND(1)*33)
100 WX=INT(RND(1)*20) + 1
110 IF ABS(LX - WX) < 4 THEN 70
120 COLOR 0,2:COLOR 4,9:COLOR 5,7:GRAPHIC 0,1
130 Z = WX : GOSUB 500
140 COLOR 5,3:PRINTTAB(WY);"  ";B$
150 PRINTTAB(WY);C$
160 Z = LX - 1 : GOSUB 500
170 COLOR 5,8:PRINTTAB(LY);"  ";D$
180 PRINTTAB(LY);"  ";A$
190 PRINTTAB(LY+1);C$
195 PRINT"  "
200 IF LX = WX AND WY = LY - 3 THEN 300
210 Z = 22 : GOSUB 500
220 COLOR 5,1:INPUT"CHEMIN A SUIVRE (DGG HBB...)" ;R$
230 X = 0 : Y = 0
240 N = N + 1
250 FOR I = 1 TO LEN(R$)
260 X = X - (MID$(R$,I,1)="B" AND LX < 20) + (MID$(R$,I,1)="H" AND LX > 0)
270 Y = Y - (MID$(R$,I,1)="D" AND LY < 35) + (MID$(R$,I,1)="G" AND LY > 0)
280 SOUND 1,2000,4:NEXT I
290 LX = LX + X : LY = LY + Y
295 GOTO 120
300 Z = 22 : GOSUB 500
310 COLOR 5,12:PRINT"BRAVO ; TROUVE EN"N ;" COUPS"
320 PLAY"CEG":END
500 PRINT"  ";
510 FOR I = 0 TO Z
520 PRINT
530 NEXT I
540 RETURN

```

Tirage d'un dé truqué

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

Les probabilités désirées pour chaque tirage sont introduites par une boucle I (40-100) avec toutes les vérifications de validité nécessaires. Le nombre de tirages demandés est introduit en ligne 115.

La boucle J (120-180) comptabilise les tirages.

La boucle I (140-160) assure la conformité de chaque tirage avec les probabilités introduites en début de jeu.

Les résultats sont comptabilisés en ligne 170, et affichés grâce aux lignes 175 et 178.

POUR JOUER :

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

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

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

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

EXTENSIONS POSSIBLES :

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

```

10 REM TIRAGE DE TRUQUE
15 COLOR 0,9:COLOR 4,1:COLOR 5,7:GRAPHIC
  0,1
20 P = 0
30 DIM E(6)
40 FOR I = 1 TO 6
45 PRINT"S0000";TAB(17);"          "
50 SOUND 1,10000,4:PRINT"S0000PROBABILITE
  DU DE"I::INPUTE(I)
60 P = P + E(I)
70 IF E(I) < 0 OR E(I) > 1 OR P > 1.1 TH
EN 200
80 PRINT"S0000000";TAB(6*I-6);"DE"I;
90 PRINT"S0000000";TAB(6*I-6);E(I)
100 NEXT I
110 IF P < 0.9 THEN RUN
115 PLAY"CEG":INPUT"NOMBRE DE TIRAGES";
P
117 WINDOW 0,11,39,16,1
118 FOR I=1TO7:COLOR 5,12:PRINT"2
          ";:NEXT
120 FOR J = 1 TO P
130 N = INT(RND(1)*6 + 1)
140 FOR I = 1 TO 6
150 IF N = I AND RND(1) > E(I)-INT(E(I))
  THEN 130
160 NEXT I
170 E(N) = E(N) + 1:SOUND 1,500*N+4000,2
175 COLOR 5,1:PRINT"S0"TAB(6*N-4);N
178 PRINT"S0000"TAB(6*N-5);E(N)
180 NEXT J
190 PLAY"CDEFG":WINDOW 0,0,39,24:END
200 PRINT"IMPOSSIBLE : 0 <= P <= 1"
210 P = P - E(I):PLAY"CCC"
220 SLEEP 6
230 PRINT"7
  "
240 GOTO 50

```

Le jeu de Marienbad

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : logique (2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

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

En effet, il est possible, en début de partie, de choisir le nombre 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 50 dimensionne une variable A (I) qui contiendra le nombre d'allumettes de chaque tas.

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

La boucle I (110-140) imprime les tas.

La ligne 150 assure l'affichage du joueur.

La ligne 170, ainsi que la ligne 180, contrôlent si le coup est autorisé.

La ligne 250 contrôle la fin de la partie.

POUR JOUER :

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

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

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

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

EXTENSIONS POSSIBLES :

On peut désigner chaque joueur par son nom, faire un graphisme plus élaboré, écrire un programme permettant au C128 de servir de partenaire, etc.

```

10 REM LE JEU DE MARIENBAD
15 COLOR 0,4:COLOR 4,5:COLOR 5,9:GRAPHIC
  0,1
20 INPUT"NOMBRE DE TAS ( <=7 )":N
30 IF N >=8 THEN RUN
40 S = 0
50 DIM A(N)
60 FOR I = 1 TO N
70 SOUND 1,10000,4:PRINT"00TAS" I):INPUT";
NOMBRE D'ALLUMETTES":A(I)
80 S = S + A(I)
90 NEXT I
95 H = 1
100 SCNCLR 0
110 FOR I = 1 TO N
120 COLOR 5,7:PRINT"00000000"TAB(5*I+2):A(
I):SOUND 1,500*INT(RND(1)*10+1)+1000,6
130 COLOR 5,1:PRINT"00TAS"TAB(5*I+2):I
140 NEXT I
150 PRINT"00JOUEUR"H:SOUND 1,4000*H,6
160 INPUT"00DANS QUEL TAS":T
170 IF T > N THEN 100
180 IF A(T) = 0 THEN 100
190 INPUT"00000000000000000000QUANTITE A ENL
EVER":O
200 IF O > A(T) OR O <= 0 THEN 190
210 A(T) = A(T) - O
220 S = S - O
230 H = H + 1
240 IF H = 3 THEN H = 1
250 IF S = 0 THEN PRINT"00LE JOUEUR"H):
A GAGNE" : PLAY"CDEFG":END
260 GOTO 100

```

Compose

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : composition musicale.

Difficulté du jeu : aucune, a priori.

LE JEU :

Il s'agit, en 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 (30-60) dessine la portée. La variable N\$, en ligne 70, contient les noms des notes et P\$ contient les notes sous forme codée.

La deuxième boucle I (80-160) permet de faire le choix d'une note de façon aléatoire, à la ligne 90 ; son nom est calculé en ligne 100 et la note est dessinée par la ligne 120.

La ligne 180 permet de recommencer en tapant SHIFT.

POUR JOUER :

Lancer l'exécution du programme en tapant RUN.

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

Une pression sur la touche SHIFT provoquera l'affichage d'une nouvelle série de notes, et une pression sur les touches RUN/STOP et RESTORE provoque la fin de la partie.

EXTENSIONS POSSIBLES :

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

```

10 REM COMPOSE
20 COLOR 0,4:COLOR 4,12:COLOR 5,7:GRAPHI
C 0,1
30 FOR I = 1 TO 5
40 Z = 3 + 2*I : GOSUB 500
50 PRINT"8"
5"
60 NEXT I
70 N$ = "SI LA SOLFA MI RE DO ":P$="BAGF
EDC"
80 FOR I = 1 TO 13
90 X = INT(RND(1)*14 + 1)
100 D$ = MID$(N$,3*(X+7*(X>7))-2,3)
110 Z = X-1 : GOSUB 500
120 PRINT"▯";TAB(3*I-3);" I":PRINTTAB(3*I
-3);" I":PRINTTAB(3*I-3);"●"
130 PRINT"▯"
140 Z = 18 : GOSUB 500
150 PRINTTAB(3*I-3);D$:IF X<=7 THEN PLAY
"04U5"+MID$(P$,X+7*(X>7),1):GOTO 157
155 PLAY"03U5"+MID$(P$,X+7*(X>7),1)
157 SLEEP 1
160 NEXT I
170 Z = 21 : GOSUB 500
180 PRINT"FAIRE - SHIFT - POUR CONTINUER
"
190 WAIT211,1
200 RUN
500 PRINT"8"
510 FOR P = 0 TO Z
520 PRINT
530 NEXT P
540 RETURN

```

Jour de la semaine

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : utilitaire.

Difficulté du jeu : facile.

LE JEU :

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

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

LE PROGRAMME :

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

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

La formule utilisée apparaît aux lignes 130 et 140.

Les lignes 150 et 160 permettront l'affichage en clair du jour dans la réponse. La ligne 170 imprime les résultats.

POUR JOUER :

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

Le résultat est très rapidement affiché.

On tape la touche SHIFT et le programme revient sur la question JOUR.

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

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

EXTENSIONS POSSIBLES :

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

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

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


```

10 REM JOUR DE LA SEMAINE
20 COLOR 0,1:COLOR 4,1:COLOR 5,3:GRAPHIC
  0,1
30 INPUT"JOUR.....:";J
40 IF J = 0 THEN PLAY"CEG":END
50 INPUT"MOIS ( 1-12 )..:";M
60 INPUT"ANNEE.....:";A
70 B = A
80 S = INT(A/100)-INT(A/400)
90 IF A <= 1582 THEN S = 2
100 M$="JANVFEVRMARSAVRIMAI JUINJUILAOUT
SEPTOCTONOVEDECE"
110 M$ = MID$(M$,4*M-3,4)
120 IF M < 3 THEN GOSUB 200
130 S=J-1+INT(13*(M+1)/5)+INT(5*A/4)-S
140 S=S-7*INT(S/7)
150 J$="DIMANCHELUNDI   MARDI   MERCREDI
JEUDI   VENDREDISAMEDI   "
160 J$=MID$(J$,8*S+1,8)
170 PRINT"0000LE";J;"   ";M$;B;"   ETAIT U
N ";J$:PLAY"CDEFG"
180 PRINT"0000FAIRE - SHIFT - POUR CONTI
NUER":WAIT 211,1
190 GOTO 20
200 M = M + 12
210 A = A - 1
220 RETURN

```

Tour de cartes

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de prestidigitation.

Difficulté du jeu : facile.

LE JEU :

C'est un tour assez facile mais, ici, les cartes sont manipulées sous vos yeux par le C128 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 C128 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 C128 est en mesure de deviner la carte choisie.

LE PROGRAMME :

La boucle I (30-90) permet de choisir, au hasard, 21 cartes sur 52, et la boucle emboîtée J (50-70) permet de s'assurer que l'on ne réalise pas deux fois le même tirage. La ligne 110 contient les couleurs des cartes ainsi que les noms des cartes figures (VA, DA, RO, AS), dans une variable alphanumérique B\$.

La boucle I (120-210) réalise l'affichage du jeu.

La ligne 130 décode la couleur de la carte.

La ligne 140 choisit la position d'affichage.

La ligne 150 décode la hauteur.

L'affichage proprement dit est effectué de la ligne 160 à la ligne 200.

La ligne 230 permet au joueur d'entrer le numéro de la colonne où se trouve sa carte. La boucle J (270-290) réalise le brouillage, et les boucles emboîtées J (320-360) et I (330-350) fabriquent la nouvelle variable A\$ à utiliser pour le prochain affichage.

POUR JOUER :

On lance le programme par RUN.

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

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

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

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

On sortira du programme en tapant RUN/STOP et RESTORE à la place du numéro de la colonne.

EXTENSIONS POSSIBLES :

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

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

```

10 REM TOUR DE CARTES
15 COLOR 0,2:COLOR 4,9:COLOR 5,6:GRAPHIC
  0,1
20 A$=""
30 FOR I = 1 TO 21
40 B$ = CHR$(INT(RND(1)*52+38))
50 FOR J = 1 TO LEN(A$)
60 IF B$ = MID$(A$,J,1) THEN 40
70 NEXT J
80 A$ = A$ + B$
90 NEXT I
100 H = 3 : P = 1
110 B$ = "TRCACOPIVADARORAS"
120 FOR I = 1 TO LEN(A$)*P - 11*(P=0)
130 S = INT((ASC(MID$(A$,I,1))-38)/13)
140 J = 7*INT((I-1)/7)
150 T = ASC(MID$(A$,I,1))-37-13*S
160 IF P = 0 THEN PRINT"§VOTRE CARTE EST
  :"; " " :PRINT"§" :GOTO 180
170 PRINT"§":FOR L = 0 TO (I-J)*P :PRINT
  :NEXT L
180 IF T <= 9 THEN PRINTTAB(J*P+INT(J/2)
-16*(P=0));STR$(T+1);
190 IF T > 9 THEN PRINTTAB(J*P+INT(J/2)-
16*(P=0));MID$(B$,ABS(2*T-11),2);
200 PRINTTAB((J+3)*P+1+INT(J/2)-21*(P=0)
);MID$(B$,2*S+1,2):SOUND 1,10000,2
210 NEXT I
220 IF P = 0 THEN PLAY"CDEFG":END
230 INPUT"§§QUELLE COLONNE";N
240 PRINT"§"
250 H = H - 1
260 R$ = "20120" : B$ = ""
270 FOR J = 0 TO 2
280 B$ = B$ + MID$(A$,7*VAL(MID$(R$,N+J,
1))+1,7)
290 NEXT J
300 IF H=0 THEN A$=B$:P=0:GOTO 110
310 A$ = ""
320 FOR J = 0 TO 2
330 FOR I = 1 TO 7
340 A$ = A$ + MID$(B$,3*I-2+J,1):SOUND 1
,400*I+500*J,4
350 NEXT I
360 NEXT J
370 GOTO 110

```

Le jeu de la vie

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu solitaire (2 versions).

Difficulté du jeu : facile.

LE JEU :

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

Une certaine population est définie au début du jeu soit par le C128 (version automatique) soit par vous-même (version manuelle). Cette population va évoluer sous vos yeux, en respectant certaines règles, très élémentaires de naissance, de vie et de mort. L'évolution se poursuivra jusqu'à ce que l'on décide d'arrêter le programme.

LE PROGRAMME :

Deux tables (B et C) sont utilisées aux lignes 40 et 50, pour enregistrer l'état de la population. Les boucles F (230-310) et G (240-300) sont utilisées en début de jeu pour définir la population initiale (ligne 260). Le calcul de la nouvelle population est ensuite réalisé à l'aide d'une quadruple boucle : 80-210, 90-200, 110-160, 120-150. La ligne 130 réalise la gestion du nouvel état suivant les règles.

L'affichage de ce nouvel état est obtenu en réutilisant la double boucle d'affichage F (230-310) et G (240-300).

La ligne 320 permet de recommencer.

POUR JOUER :

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

EXTENSIONS POSSIBLES :

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

```

10 REM LE JEU DE LA VIE (AUTOMATIQUE)
20 COLOR 0,13:COLOR 4,13:COLOR 5,3:GRAPH
   IC 0,1
30 INPUT"DIMENSION":A
40 DIM B(A,A)
50 DIM C(A,A)
60 I = 0
70 GOTO 220
80 FOR F = 1 TO A
90 FOR G = 1 TO A
100 S = 0
110 FOR I = -1 TO 1
120 FOR J = -1 TO 1
130 IF F + J = 0 OR G + I = 0 OR F + J =
   A + 1 OR G + I = A + 1 THEN 150
140 S = S + B(F + J , G + I)
150 NEXT J
160 NEXT I
170 S = 2*S - B(F,G)
180 C(F,G) = 0
190 IF S >= 5 AND S <= 7 THEN C(F,G) = 1
200 NEXT G
210 NEXT F
220 PRINT"J"
230 FOR F = 1 TO A
240 FOR G = 1 TO A
250 IF I <> 0 THEN 270
260 IF RND(1) < 0.5 THEN C(F,G) = 1
270 PRINT"0":FOR T=0 TO 2*I:PRINT:NEXT T
280 IF C(F,G) = 1 THEN PRINTTAB(2*I);"+"
   :SOUND 1,12000,2
290 B(F,G) = C(F,G)
300 NEXT G
310 NEXT F
320 GOTO 80

```

```

10 REM LE JEU DE LA VIE (MANUEL)
20 COLOR 0,13:COLOR 4,13:COLOR 5,3:GRAPH
   IC 0,1
30 INPUT"DIMENSION";A
40 DIM B(A,A)
50 DIM C(A,A)
60 I = 0
70 GOTO 220
80 FOR F = 1 TO A
90 FOR G = 1 TO A
100 S = 0
110 FOR I = -1 TO 1
120 FOR J = -1 TO 1
130 IF F + J = 0 OR G + I = 0 OR F + J =
   A + 1 OR G + I = A + 1 THEN 150
140 S = S + B(F + J , G + I)
150 NEXT J
160 NEXT I
170 S = 2*S - B(F,G)
180 C(F,G) = 0
190 IF S >= 5 AND S <= 7 THEN C(F,G) = 1
200 NEXT G
210 NEXT F
220 PRINT"J"
230 FOR F = 1 TO A
240 FOR G = 1 TO A
250 IF I <> 0 THEN 270
255 PRINT"#####LIGNE";F
   ;" COLONNE";G;" TAPER 1 OU 0":PRINT" "
260 INPUT"#####";C(F,G)
   )
270 PRINT"0":FOR T=0 TO 2*F:PRINT:NEXT T
280 IF C(F,G) = 1 THEN PRINTTAB(2*G);"♦"
   :SOUND 1,10000,2
290 B(F,G) = C(F,G)
300 NEXT G
310 NEXT F
320 GOTO 80

```

Niveau 3

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Plus grand, plus petit	Un jeu de déduction intéressant, pour un joueur.
Ordre	Un peu d'astuce pour remettre en ordre, le plus vite possible.
Le pendu	Un jeu de lettres très classique (2 joueurs).
Belote	Pour simuler une donne, avec un jeu de 32 cartes.
Un taquin de calcul	Un bon entraînement (2 joueurs).
Pendu inversé	Cette fois, c'est l'ordinateur qui essaie de deviner le mot.
Nicomaque	Pour se perfectionner en calcul mental (1 ou plusieurs joueurs).
Mastermind	Avec des chiffres, mais très classique. Un seul joueur.
Idem	Un jeu de calcul mental, avec plusieurs niveaux, pour plusieurs joueurs.
Horloge	Ne jetez pas votre vieille télévision : transformez-la en horloge.
Reines sur l'échiquier	Un vieux problème, pour vous aider à réfléchir.
Mots	Une idée très simple, mais ce n'est pas si facile de trouver !
Permutations	Un jeu de réflexion, pas toujours évident.
Bridge	Mettez au point vos propres problèmes de bridge.
Hanoi	Un conseil : commencez petit ! ! !
Répétitions	Retrouvez une séquence de chiffres de plus en plus longue.
Jeu de loto	Gagnez sans problème, au prochain tirage.
Roulette	Une simulation élégante du jeu très connu.
Musique	Révissez vos notions de solfège, grâce à l'ordinateur.
A travers	Un jeu de stratégie pour 2 joueurs.
Le jeu du quinze	Un jeu de réflexion pour 2 joueurs.

Manipulation des chaînes, attente d'une touche et instructions du Basic structuré

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

Nous avons eu l'occasion de rencontrer, dès le niveau 2, la fonction essentielle utilisée en Basic pour manipuler les chaînes : il s'agit de MID\$ (A\$, I, J) qui permet d'extraire, dans la chaîne A\$, une sous-chaîne de longueur J, commençant au I^{ème} caractère.

Ainsi si,

A\$ = "MONSIEUR"

on aura :

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

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

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

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

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

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

On peut aussi écrire MID\$ (A\$, 6) = "EUR".

Le Basic du C128 permet de plus le remplacement direct d'une sous-chaîne par une affectation du type MID\$ (-.) = "-".

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

ATTENTE D'UNE TOUCHE

Nous avons déjà eu l'occasion d'utiliser l'instruction `WAIT`, qui permettait d'attendre l'appui de la touche `SHIFT`. Le Basic du C128 dispose d'une instruction plus générale, `GET KEY`, qui permet d'attendre l'appui d'une touche quelconque.

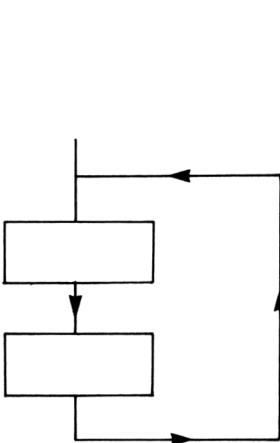
On l'utilise sous la forme `GET KEY R$`, la touche pressée est enregistrée dans la variable alphanumérique `R$`, et l'exécution est immédiatement reprise sans appuyer sur `RETURN`.

INSTRUCTIONS DU BASIC STRUCTURE

DO ... LOOP

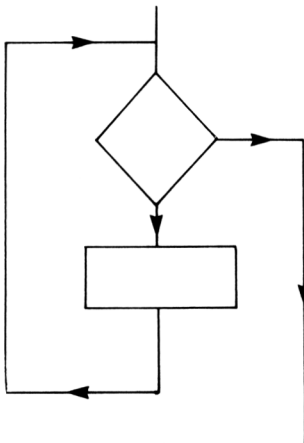
La boucle `DO ... LOOP` permet des répétitions sous une forme beaucoup plus souple que le classique `FOR ... NEXT`. On peut en effet l'utiliser en association avec deux instructions de test originales : `UNTIL` et `WHILE`, et quitter la boucle sans contrainte avec `EXIT`.

Boucle infinie



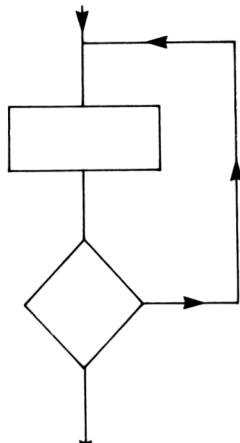
DO
.
.
.
LOOP
On sort avec
RUN/STOP
et RESTORE

Boucle avec sortie
en tête



condition
DO UNTIL . . .
.
.
.
LOOP

Boucle avec sortie
en queue

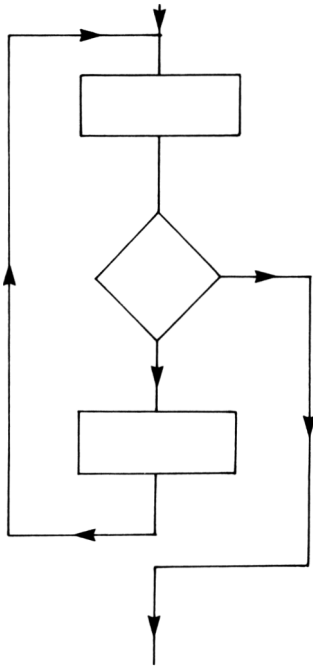


DO
.
.
.
LOOP
WHILE
condition

Pour une sortie à n'importe quel endroit d'une boucle, on pourra utiliser EXIT dans un test.

```
DO
.
.
IF ... THEN EXIT
.
.
LOOP
```

Schéma



BEGIN ... BEND

Dans le Basic du C128, le couple BEGIN ... BEND permet une intéressante extension des possibilités de IF ... THEN ... ou IF ... THEN ... : ELSE. En effet, le nombre de caractères par ligne est limité à 160, ce qui ne permet pas beaucoup de possibilités derrière la clause THEN ou ELSE. On peut alors utiliser BEGIN :, ce qui permet d'introduire de nouvelles lignes associées au THEN (ou au ELSE) ; la fin de ces lignes est alors impérativement signalée par BEND.

Plus grand, plus petit

REFERENCES :

Difficulté de programmation : facile.

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

Difficulté du jeu : difficile.

LE JEU :

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

LE PROGRAMME :

La ligne 40 interdit de demander des nombres de plus de 8 chiffres.

Le nombre à deviner est placé en ligne 60, dans une variable alphanumérique X\$.

Les comparaisons avec le nombre essayé sont faites à la ligne 110 (qui teste si l'on a trouvé juste), et dans la boucle I (130-170) où l'on fait la comparaison "lettre à lettre".

L'affichage du résultat est réalisé grâce à la ligne 180.

Le compteur du nombre d'essais est en ligne 190.

POUR JOUER :

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

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

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

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

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

Si l'on a trouvé le C128 affichera VOUS AVEZ TROUVE... EN... ESSAIS. Puis AUTRE PARTIE (O/N) ; on fera O pour commencer une nouvelle partie.

EXTENSIONS POSSIBLES :

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

On peut aussi prévoir de laisser jouer le C128 qui tiendra alors le rôle de décodeur.

```

10 REM PLUS GRAND, PLUS PETIT
20 COLOR 0,11:COLOR 4,15:COLOR 5,4:GRAPH
IC 0,1
30 INPUT"NOMBRE DE CHIFFRES";C
40 IF C > 8 THEN PLAY"CCC":RUN
50 E = 1
60 X$=STR$(INT(RND(1)*9+C+10*(C-E)))
70 X$ = RIGHT$(X$,LEN(X$)-1)
80 PRINT"SUITEESSAI NO.  NOMBRE ESSAYE
>    <"
90 PRINT"SUITE"


---


100 COLOR 5,9:PRINTTAB(4>E);INPUT"    ";P
$:PRINT"J";
110 IF P$ = X$ THEN 200
120 A = 0 : B = 0
130 FOR I = 1 TO C
140 IF MID$(P$,I,1)>MID$(X$,I,1) THEN A
= A + 1:SOUND 1,5000,4
150 IF MID$(P$,I,1)<MID$(X$,I,1) THEN B
= B + 1:SOUND 1,7000,4
170 NEXT I
180 COLOR 5,7:PRINTTAB(27>A):COLOR 5,3:P
RINTTAB(32>B)
190 E = E + 1
195 GOTO 100
200 PRINT"J"
210 COLOR 5,2:PRINT"VOUS AVEZ TROUVE ";X
$;" EN"E;" ESSAIS":PLAY"CEG"
220 PRINT"
"
230 PRINT"J":PRINT"AUTRE PARTIE ( O/N )
":GETKEY R$
240 IF R$="O" THEN RUN:ELSE PLAY"CDEFG"

```

Ordre

REFERENCES :

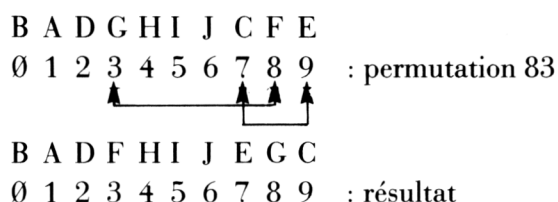
Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de réflexion.

Difficulté du jeu : moyenne.

LE JEU :

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



LE PROGRAMME :

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

La ligne 110 contrôle les résultats.

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

On entre la permutation en ligne 150.

Les lignes 160 et 170 transforment la variable alphanumérique entrée en une variable numérique.

La ligne 180 appelle une première fois le sous-programme d'inversion.

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

La ligne 250 comptabilise les essais.

POUR JOUER :

Lancer le programme par RUN.

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

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

Le nombre de coups sera affiché en fin de partie.

EXTENSIONS POSSIBLES :

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

```

10 REM ORDRE
20 COLOR 0,11:COLOR 4,15:COLOR 5,7:GRAPH
IC 0,1
30 A$="":S=1
40 FOR I = 1 TO 10
50 R$ = CHR$(INT(RND(1)*10 + 65))
60 FOR J = 1 TO I - 1
70 IF R$ = MID$(A$,J,1) THEN 50
80 NEXT J
90 A$ = A$ + R$
100 NEXT I
110 DO UNTIL A$="ABCDEFGHIJ"
120 COLOR 5,7:PRINT"#####";A$;"■"
130 COLOR 5,1:PRINT"#####0123456789"
135 COLOR 5,2:PRINT"#####DESSAI NUMERO"S
140 PRINT"#####TAB(33);"   ";
150 COLOR 5,8:INPUT"#####INVERSION (I
ET J SOUS FORME IJ )";R$
160 I = ASC(LEFT$(R$,1)) - 47
170 J = ASC(RIGHT$(R$,1)) - 47
180 GOSUB 500
210 IF LEFT$(R$,1)="0" OR LEFT$(R$,1)="9"
" THEN 250
220 I = I-1 : J = I+2
230 GOSUB 500
250 S = S + 1
260 LOOP
270 COLOR 5,3:PRINT"TROUVE EN"S;"COUPS":
PLAY"CDEFG":END
500 B$ = ""
510 FOR T = 1 TO LEN(A$)
520 IF T = I THEN N$ = MID$(A$,T,1)
530 IF T = J THEN M$ = MID$(A$,T,1)
540 NEXT T
550 FOR T = 1 TO LEN(A$)
560 IF T = I THEN B$ = B$ + M$:GOTO 600
570 IF T = J THEN B$ = B$ + N$:GOTO 600
580 B$ = B$ + MID$(A$,T,1):SOUND 1,3000+
400*INT(RND(1)*9),5
600 NEXT T
610 A$ = B$
620 RETURN

```

Le pendu

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de lettres (version 2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

LE PROGRAMME :

Le mot à deviner est introduit en début de programme (ligne 30) ; une variable alphanumérique, en ligne 40, contient tous les éléments du dessin. La boucle I (60-80) fabrique un mot P\$ qui contient des tirets à l'emplacement de chacune des lettres du mot initial.

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

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

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

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

Il sera intéressant de constituer une liste de mots dans lesquels le C128 pourra piocher au hasard ; on pourra aussi améliorer le dessin (en redéfinissant les caractères, par exemple) et modifier le nombre d'essais permis selon la difficulté du mot.

Belote

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de cartes.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

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

La boucle I (50-100) assure le tirage aléatoire de 20 nombres, pris parmi 32 ; la boucle J (70-90) vérifie que l'on ne réalise pas deux fois le même tirage.

La deuxième boucle I (110-300) assure ensuite l'affichage, les noms des cartes étant affichés en clair, et la couleur étant décodée et affichée en clair, grâce aux lignes 160 à 270.

POUR JOUER :

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

Un nouveau RUN permet d'obtenir une nouvelle distribution.

EXTENSIONS POSSIBLES :

On pourra facilement mettre au point un affichage plus graphique en redéfinissant les caractères du C128, ou en utilisant les caractères prédéfinis ; on pourra également se servir de ce programme comme d'un noyau pour la mise au point de programmes de jeux de cartes sur l'ordinateur (qui pourra, si nécessaire, servir de partenaire : voir l'exemple du Poker, dans le même ouvrage).

```

10 REM BELOTE
20 DIM A(20)
30 K = 1
40 COLOR 0,13:COLOR 4,13:COLOR 5,2:GRAPH
   IC 0,1
45 PRINT"PATIENTEZ , JE TIRE"
50 FOR I = 1 TO 20
60 A(I) = 1 + INT(RND(1)*32)
65 IF I = 1 THEN 100
70 FOR J = 1 TO I-1
80 IF A(I) = A(J) THEN 60
90 NEXT J
100 NEXT I
105 PRINT"□"
110 FOR I = 1 TO 20
120 S$=CHR$(28)+CHR$(122)
130 S = 1 + A(I) - 8*INT(A(I)/8)
140 R$=STR$(S+6)+"   ":R$=RIGHT$(R$,LEN(R
   $)-1)
150 R$=LEFT$(R$,2)
160 IF S = 5 THEN R$="VA"
170 IF S = 6 THEN R$="DA"
180 IF S = 7 THEN R$="RO"
190 IF S = 8 THEN R$="AS"
200 IF A(I) <= 24 THEN S$=CHR$(144)+CHR$
   (120)
210 IF A(I) <= 16 THEN S$=CHR$(144)+CHR$
   (97)
220 IF A(I) <= 8 THEN S$=CHR$(28)+CHR$(1
   5)
230 PRINT"Ⓢ"
240 FOR P = 1 TO 2*K
250 PRINT
260 NEXT P
270 COLOR 5,1:PRINTTAB(10*INT((I-1)/5));
   R$;" ";S$:SOUND 1,1000+A(I)*300,4
280 K = K + 1
290 IF K = 6 THEN K = 1
300 NEXT I:PLAY"CDEFG"

```

Un taquin de calcul

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : difficile.

LE JEU :

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

Le codeur part d'un nombre (proposé par le C128) 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 30 permet d'obtenir l'entier de départ compris entre 5 et 20.

Une boucle I (90-170) permet d'entrer successivement les 5 opérations.

4 sous-programmes sont utilisés pour les calculs (aux lignes 1000 et 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 160).

La ligne 180 affiche les 5 opérations utilisées, ainsi que les opérateurs correspondants.

POUR JOUER :

On frappe RUN pour lancer le programme.

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

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

EXTENSIONS POSSIBLES :

On pourra, bien sûr, songer à améliorer la partie décodage du programme en introduisant des contrôles sur le résultat ; on peut aussi envisager de laisser le C128 participer davantage en lui confiant le rôle du codage, etc.

```

10 REM UN TAQUIN DE CALCUL
20 COLOR 0,13:COLOR 4,13:COLOR 5,7:GRAPH
IC 0,1
30 X = INT(RND(1)*16+5)
40 PRINT"OMBRE INITIAL ";X
50 COLOR 5,3:PRINT:PRINT"ENTRER 5 OPERAT
IONS"
60 PRINT"SOUS LA FORME +5 , /2 , *4 ..."
70 F = X
80 DIM R$(5)
90 FOR I = 1 TO 5
100 PRINT"XXXXXXXXXX"
"
110 PLAY"CDEFG":INPUT"XXXXXXXXXXOTRE OPERATI
ON";R$(I)
120 A = ASC(LEFT$(R$(I),1))-41
130 IF A<=0 OR A=3 OR A=5 OR A>6 THEN100
140 IF A > 3 THEN A = (A/2)+1
150 ON A GOSUB 1000,2000,3000,4000
160 PRINT"XXXXXXXXXXOMBRE OBTENU :
";PRINT"XXXXXXXXXX";F;"
170 NEXT I
175 PRINT"XXXXXXXXXX"
"
180 COLOR 5,1:PRINT"XXXXOPERATIONS : "R$(
4)" "R$(1)" "R$(5)" "R$(3)" "R$(2)
190 COLOR 5,9:PRINT"XXXXBON COURAGE":PLA
Y"CDEFG"
200 END
1000 F=F*VAL(RIGHT$(R$(I),LEN(R$(I))-1))
1010 RETURN
2000 F=F+VAL(RIGHT$(R$(I),LEN(R$(I))-1))
2010 RETURN
3000 F=F-VAL(RIGHT$(R$(I),LEN(R$(I))-1))
3010 RETURN
4000 F=F/VAL(RIGHT$(R$(I),LEN(R$(I))-1))
4010 RETURN

```

Pendu inversé

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu sur les mots.

Difficulté du jeu : facile.

LE JEU :

Le Pendu est un jeu de lettres très connu, présenté par ailleurs dans ce recueil, dans lequel il s'agit de deviner un mot caché par l'ordinateur et où chaque lettre non découverte vous rapproche un peu plus de la potence. Ici, c'est le C128 qui va chercher à deviner votre mot, mais il joue quelque peu au hasard et ne possède pas votre intuition ; il faudra donc, en règle générale, beaucoup d'essais pour découvrir la solution. Pour augmenter ses chances, on a fait appel à une formule qui lui permet de tirer davantage dans les voyelles puis, dans les consonnes courantes, et, enfin, dans les consonnes plus rares.

LE PROGRAMME :

La ligne 20 permet d'enregistrer, dans une variable alphanumérique D\$, les lettres de l'alphabet, dans l'ordre : voyelles, consonnes usuelles, autres consonnes. Le mot à deviner est introduit en ligne 40.

La boucle I (50-80) imprime des tirets aux emplacements des lettres à trouver.

La boucle en ligne 100 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, et la boucle I (170-200) réimprime les tirets et les lettres trouvées.

La ligne 210 incrémente le compteur d'essais.

La ligne 230 affiche le nombre d'essais réalisés, en fin de partie.

POUR JOUER :

Lancer le programme par RUN.

Entrer le mot à découvrir, suivi de RETURN.

Des tirets sont alors affichés (un par lettre) et le C128 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 RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

Il sera sans doute intéressant, pour de jeunes enfants, de prévoir une version non automatique où le joueur répondrait OUI ou NON pour chaque lettre proposée par le C128 ; on pourra aussi imaginer d'ajouter un affichage graphique, par exemple une potence ou une voiture qui avance vers un précipice, etc.

```

10 REM PENDU INVERSE
15 COLOR 0,13:COLOR 4,13:COLOR 5,7:GRAPH
   IC 0,1
20 D$ = "YAEIOUBCDFGJLMNPRSTVWHKQXZ"
30 C = 1 : T$ = ""
40 PLAY"C6G":INPUT"ENTRER VOTRE MOT";R$
45 PRINT"J"
50 FOR I = 1 TO LEN(R$)
60 COLOR 5,2:PRINT"XXXXXXXXXX";TAB(I);"-"
70 T$ = T$ + "-"
80 NEXT I
90 COLOR 5,3:PRINT"XXXXXXXXXXXXJE CHERCHE.
...; ESSAI";:COLOR 5,1:PRINTC
100 SLEEP 3
110 X = -6*(RND(1)<.7)-13*(RND(1)<.5)-7*
(RND(1)<.4)
120 X = INT(X*RND(1)+1)
130 IF MID$(D$,X,1) = "-" THEN 110
140 E$ = MID$(D$,X,1)
150 MID$(D$,X,1)="-"
160 COLOR 5,9:PRINT"XXXXJE PRENDS : ";:COL
OR 5,2:PRINT E$:SOUND 1,300*X+500,4
170 FOR I = 1 TO LEN(R$)
180 IF E$ = MID$(R$,I,1) THEN MID$(T$,I,
1)=E$
190 COLOR 5,2:PRINT"XXXXXXXXXX";TAB(I);MID$(
T$,I,1)
200 NEXT I
210 C = C + 1
220 IF T$<>R$ THEN 90
230 COLOR 5,7:PRINT"XXXXXXXXXXXXXXXXXXXXTROUV
E EN";C-1;" COUPS" :PLAY"CDEFG":END

```

Nicomaque

REFERENCES :

Difficulté de programmation : moyenne.

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

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

LE PROGRAMME :

La ligne 30 permet de confectionner l'entier.

La boucle I (40-90) calcule les trois restes successifs, et la boucle DO/LOOP (60-80) permet de les afficher sous une forme assez visuelle, aux bons emplacements.

Les lignes 35 et 38 sont là pour indiquer de quelle division il s'agit.

Le nombre proposé par le joueur est entré à la ligne 100 ; il est affiché et testé aux lignes 110 et 120 (affichage de fin de partie : BRAVO ou PERDU).

Les lignes 130 à 160 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 cœurs (l'absence de cœur signifiant : reste 0).

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

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

Pour faire un nouvel essai, on répondra O, ou M.

EXTENSIONS POSSIBLES :

On peut envisager de réaliser le programme décodeur (on donne les restes et le C128 devra vous donner N) et chercher si la propriété peut être étendue (en effectuant davantage de divisions, par exemple).


```

10 REM NICOMAUQUE
20 COLOR 0,13:COLOR 4,13:COLOR 5,7:GRAPH
IC 0,1
30 X = INT(RND(1)*100 + 1):S=1
35 PRINT"      RESTE DANS LA DIVISION PAR
"
38 PRINT"0          3          5
7"
40 FOR I = 1 TO 3
50 A = X - (2*I+1)*INT(X/(2*I+1))
60 DO UNTIL A=0
70 COLOR 5,3:PRINT"5000000";TAB(A + 8*I +
6);"♥":A=A-1
80 LOOP
90 NEXT I
100 PLAY"CEG":COLOR 5,2:INPUT"000000VOTR
E ESSAI";N
110 IF N = X THEN PRINT"00BRAVO. TROUVE
EN ";S;" ESSAIS":PLAY"CDEFG":END
120 PRINT"00PERDU":S=S+1:PLAY"CCC"
130 COLOR 5,7:PRINT"00UNE AUTRE PARTIE
OU LA MEME ( O/N/M )":GETKEY R$
140 IF R$ = "O" THEN RUN
150 IF R$ = "M" THEN WINDOW 0,9,39,22,1:
GOTO 100: ELSE PLAY"CDEFG"
160 PRINT"00LA SOLUTION ETAIT ";X

```

Mastermind

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : logique (un seul joueur).

Difficulté du jeu : difficile.

LE JEU :

Devenu un grand classique des jeux sur micro-ordinateur, ce jeu développe le raisonnement déductif ; il faut réussir à trouver le plus vite possible un nombre de n chiffres (on pourra prendre n compris entre 1 et 5), en disposant à chaque essai de renseignements.

En effet, le C128 répondra en indiquant d'une part le nombre de chiffres qui sont, dans le nombre proposé, exactement à la même place que dans le nombre à découvrir, et, d'autre part, le nombre de chiffres qui figurent dans le nombre à découvrir sans être exactement au bon emplacement.

LE PROGRAMME :

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

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

L'affichage est préparé par la ligne 90.

La ligne 150 s'occupe de la fin de partie.

Une première boucle I (160-260) teste les chiffres qui sont à la bonne place.

Une deuxième boucle J (180-200) teste les chiffres qui sont présents et mal placés.

Les compteurs sont aux lignes 220 et 250.

La ligne 270 affiche le résultat, et la ligne 280 comptabilise les essais.

POUR JOUER :

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

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

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

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

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

On peut sortir du programme à tout moment en tapant RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

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

```

10 REM MASTERMIND
20 COLOR 0,15:COLOR 4,12:COLOR 5,2:GRAPH
   IC 0,1
30 INPUT"COMBIEN DE CHIFFRES";N
40 IF N > 6 THEN RUN
50 E = 1:F=1
60 X$ = STR$(INT(RND(1)*9↑N+10↑(N-1)))
70 X$ = RIGHT$(X$,LEN(X$)-1)
80 PRINT"J";
90 COLOR 5,9:PRINT" NO.      NOMBRE
   BON      PLACE"
95 PRINT"J"
100 FOR I = 0 TO 1 + F
110 PRINT
120 NEXT I
130 COLOR 5,1:PRINT;TAB(8);:INPUTE$:PRI
   NT TAB(23);:PRINT"J"
140 X = 0 : Y = X
150 IF E$ = X$ THEN COLOR 5,1:PRINT"TRO
   UVE EN"E;"COUPS":PLAY"CDEFG":END
160 FOR I = 1 TO N
170 IF MID$(X$,I,1)=MID$(E$,I,1) THEN250
180 FOR J = 1 TO N
190 IF J <>I AND MID$(X$,I,1) = MID$(E$,
   J,1) THEN 220
200 NEXT J
210 GOTO 260
220 Y = Y + 1
230 GOTO 260
250 X = X + 1
260 NEXT I
270 COLOR 5,7:PRINTTAB(23);X;TAB(30);:CO
   LOR 5,3:PRINT Y:SOUND 1,10000,4
280 E=E+1:F=F+1:IF F=18 THEN F=1:PRINT "
   "
290 PRINT"J":GOTO 100

```

Idem

REFERENCES :

Difficulté de programmation : moyenne.

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

Difficulté du jeu : selon le niveau choisi.

LE JEU :

Le C128 choisit un nombre de n chiffres (selon le niveau de difficulté indiqué au début). Ce nombre (le même au début pour les 2 joueurs) sera régulièrement décrémenté à chaque tour, d'une quantité également choisie au hasard par le C128 en début de jeu ; vous devez, à chaque coup, choisir une colonne (les dizaines, les centaines, etc.) et ajouter dans cette colonne un certain entier ; le but est d'arriver à obtenir à un moment l'affichage d'un nombre ayant tous ses chiffres identiques.

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

LE PROGRAMME :

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

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

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

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

La ligne 130 affiche le numéro du joueur.

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

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

Le résultat sera affiché un court instant (ligne 190). La boucle I (220-240) permet de vérifier si tous les chiffres sont identiques.

L'impression et la fin de partie sont effectuées en ligne 250. Les lignes 260 et 270 s'occupent du changement de joueur.

POUR JOUER :

Lancer l'exécution du programme par RUN.

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

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

EXTENSIONS POSSIBLES :

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

```

10 REM IDEM
20 COLOR 0,2:COLOR 4,13:COLOR 5,7:GRAPHI
C 0,1
30 H = 1
40 DIM N(2)
50 INPUT"NIVEAU ( 2-7 )";N
60 U = 10 : P = 0
70 N(1) = INT((U↑N-U↑(N-1))*RND(1)+U↑(N-
1))
80 N(2) = N(1)
90 A = INT(2*U↑(N-1)*RND(1)+U↑(N-1))
100 IF H = 1 THEN P = P + 1
110 PRINT"J"
120 IF N(H)<U↑(N-1) THEN N(H)=U↑N-ABS(N(
H))
130 COLOR 5,5:SOUND 1,4000*H+2000,4:PRIN
T"JOUVEUR"H;" ; ESSAI"P
140 COLOR 5,12:PRINT"NOMBRE";TAB(15
-N);N(H)
150 COLOR 5,1:PRINTTAB(9);"7654321"
160 COLOR 5,9:INPUT"VOTRE JEU (
N ET C SOUS FORME NC )";R$
165 IF LEN(R$)<>2 OR VAL(RIGHT$(R$,1))>N
THEN 160
170 N(H) = N(H) + VAL(LEFT$(R$,1))*U↑(VA
L(RIGHT$(R$,1))-1)
180 N(H) = N(H) - A
185 IF N(H) > U↑N THEN N(H) = N(H) - U↑N
187 N(H) = INT(N(H))
190 COLOR 5,9:PRINT"RESULTAT"N(H)
200 SLEEP 4
210 R$ = STR$(N(H))
215 R$ = RIGHT$(R$,LEN(R$)-1)
220 FOR I = 1 TO LEN(R$)-1
230 IF MID$(R$,I,1) <>MID$(R$,I+1,1) THE
N 260
240 NEXT I
250 PRINT"BRAVO , JOUEUR"H:PLAY"CDEFG"
:END
260 H = H + 1
270 IF H >= 3 THEN H = 1
280 GOTO 100

```

Horloge

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : dessin.

Difficulté du jeu : aucune.

LE JEU :

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

LE PROGRAMME :

La ligne 30 vous permettra d'entrer l'heure sous forme d'un entier de 6 chiffres (par exemple : 102400 signifiera 10 heures 24 minutes 00 secondes).

Les lignes 60-240 réalisent le dessin, en décalant l'impression au fur et à mesure (à l'aide de la variable K, incrémentée en ligne 230).

Les lignes 250 à 270 assurent la progression de l'heure et effectuent l'affichage des secondes.

POUR JOUER :

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

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

L'heure sera entrée sous forme d'un entier ayant impérativement 6 chiffres, pour 5 heures 6 minutes, on veillera à taper 050600 suivi de RETURN.

EXTENSIONS POSSIBLES :

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

```

10 REM HORLOGE
20 COLOR 0,15:COLOR 4,13:COLOR 5,7:GRAPH
IC 0,1
30 INPUT"HEURE ( FORME XXXXXX )";A$
40 TI$ = A$:B$=CHR$(18)
50 PRINT"J"
60 K = 1
70 A = VAL (MID$(TI$,K,1)):IF K>2 THEN C
OLOR 5,9
80 Z = 1 : GOSUB 500
90 IF A <> 1 AND A <>4 THEN PRINTTAB(6*K
-5);B$;" ";
100 FOR J = 1 TO 5
110 Z = J : GOSUB 500
120 IF A = 0 OR (A > 3 AND A <> 7) THEN
PRINTTAB(6*K-5);B$;" ";
130 IF A <>5 AND A <>6 THEN PRINTTAB(6*K
-3);B$;" ";
140 Z = J + 4 : GOSUB 500
150 IF A = 0 OR A = 2 OR A = 6 OR A = 8
THEN PRINTTAB(6*K-5);B$;" ";
160 IF A <>2 THEN PRINTTAB(6*K-3)B$" ";
170 NEXT J
180 Z = 5 : GOSUB 500
190 IF A>1 THEN PRINTTAB(6*K-4);B$;" ";
200 IF K = 2 THEN PRINTTAB(11);CHR$(162)
:PRINTTAB(11);CHR$(162)
210 Z = 9 : GOSUB 500
220 IF A <>1 AND A <>4 AND A <>7 THEN PR
INTTAB(6*K-5);B$;" ";
230 K = K + 1
240 IF K <5 THEN 70
250 Z = 5 : GOSUB 500 : COLOR 5,12
260 T$=RIGHT$(TI$,2):PRINTTAB(28);T$:IF
T$<>S$ THEN SOUND 1,8000,1:S$=T$
270 IF RIGHT$(TI$,2) <>"00" THEN 250
280 PRINTTI$:GOTO 50
500 PRINT"J"
510 FOR L = 1 TO Z
520 PRINT
530 NEXT L
540 RETURN

```

Reines sur l'échiquier

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

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

LE PROGRAMME :

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

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

Une boucle d'exploration principale I (50-100) contient une sous-boucle J (70-90) qui réalise le contrôle de non-alignement (ligne 80).

Si, au cours de l'exploration, on rencontre un alignement, on revient en arrière, en utilisant les lignes 250 à 290 et on recommence.

La boucle F (120-220) assure l'impression de l'échiquier, et la boucle K (170-195), 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 230 permet un arrêt après l'affichage de chacune des solutions.

POUR JOUER :

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

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

EXTENSIONS POSSIBLES :

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


```

10 REM REINES SUR L'ECHIQUEUR
20 COLOR 0,15:COLOR 4,11:COLOR 5,7:GRAPH
   IC 0,1
30 INPUT"TAILLE DE L'ECHIQUEUR";S
40 DIM D(S):FAST
50 I = 1
60 D(I) = 1
65 IF I = 1 THEN 100
70 FOR J = 1 TO I - 1
80 IF (D(I) = D(J)) OR (ABS(D(I) - D(J))
   = I - J) THEN 260
90 NEXT J
100 I = I + 1 : IF I <= S THEN 60
110 PRINT"J":SLOW
120 FOR F = 1 TO S
130 Z = 0 : GOSUB 1000
140 COLOR 5,9:PRINTTAB(2*F);F
150 Z = 2*F : GOSUB 1000
160 PRINTF
170 FOR K = 1 TO S
180 Z = 2*F : GOSUB 1000
190 COLOR 5,2:PRINTTAB(2*K+1);"."
195 NEXT K
200 Z = 2*F : GOSUB 1000
203 COLOR 5,7:PRINTTAB(2*D(F)+1);CHR$(18
)+CHR$(32):SOUND 1,10000,4
205 Z = 20 : GOSUB 1000
210 COLOR 5,1:PRINTTAB(F);RIGHT$(STR$(D
(F)),1)
220 NEXT F
230 PRINT"APPUYER SUR - SHIFT - POUR CO
NTINUER"
240 WAIT 211,1
250 I = I-1
260 D(I) = D(I) + 1
270 IF D(I) <= S THEN 65
280 I = I - 1
290 IF I <>0 THEN 260
300 END
1000 PRINT"0";
1010 FOR T = 0 TO Z
1020 PRINT
1030 NEXT T
1040 RETURN

```

Mots

REFERENCES :

Difficulté de programmation : facile.

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

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

La ligne 30 permet de dimensionner la variable A\$(I) qui contiendra les mots.

La boucle I (40-60) saisit la liste des mots (inscrits en DATA aux lignes 240 à 280).

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

La ligne 165 affiche les lettres, et la ligne 170 attend votre proposition, qui est testée en ligne 180.

Le nombre d'essais est incrémenté en ligne 190.

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

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

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

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

```

10 REM MOTS
20 COLOR 0,11:COLOR 4,13:COLOR 5,7:GRAPH
IC 0,1
30 DIM A$(30) : S = 0 : E = 0
40 FOR I = 1 TO 30
50 READ A$(I)
60 NEXT I
65 PRINT"J"
70 Z = INT(RND(1)*30+1)
80 IF A$(Z) = "" THEN 70
90 R$ = A$(Z) : A$ = "" : A$(Z)="" : V$=R$
100 FOR I = 1 TO LEN(R$)
110 Z = INT(RND(1)*LEN(R$)+1)
120 IF MID$(R$,Z,1) = "-" THEN 110
130 A$ = A$ + MID$(R$,Z,1)
140 MID$(R$,Z,1)="-"
150 NEXT I
160 PRINT"LES LETTRES SONT :";COLOR 5,3
165 FOR I=1 TO LEN(A$):PRINT " ";MID$(A$,I,1);SOUND 1,12000,6:NEXT I
170 COLOR 5,2:INPUT"XXXXXXXXXXVOTRE SOLUTION
:";S$
180 IF S$ = V$ THEN S = S+1:PLAY"C6G"
190 E = E+1
200 COLOR 5,5:PRINT"XXXXXXXXXXAUTRE ESSA
I ( O/N )":GETKEY R$
210 IF R = "0" THEN 65
220 COLOR 5,1:PRINT"XXXXVOTRE SCORE : "S;"
SUR "E;" ESSAIS"
230 PLAY"CDEFG":END
240 DATA JAMBON,ENERGUMENE,AFFILIE,ERUDI
T,INSIGNIFIANT,REALISTE
250 DATA IMPRESSIONNANT,ALLECHANT,EPIQUE
,INTREPIDE,COUARD,ORDINATEUR
260 DATA TREVE,BANDEROLLE,SENAT,ACTUEL,C
ONSACRE,DIEDRE
270 DATA LIVRE,TERMINE,SIGNATURE,MALODOR
ANT,AIMANT,ARBITRAIRE
280 DATA CHALEUR,OPPORTUN,PELURE,ALTRUIS
TE,RAMONEUR,NOVICE

```

Permutations

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion.

Difficulté du jeu : moyenne.

LE JEU :

Le C128 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 80 réalise le brouillage, en tirant au hasard la nouvelle position de chaque lettre. Les lettres sont ensuite ajoutées au fur et à mesure de leur tirage, dans une variable B\$ (ligne 100), supprimées dans A\$ et re-placées en C\$, dans le bon ordre (lignes 110-120).

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

Les affichages sont effectués aux lignes 155 à 180.

Le retournement est fait, lettre par lettre, dans la boucle I (210-230) ; le résultat est placé dans B\$ en ligne 240, affiché par la boucle I (250-255) comparé au résultat final en ligne 260.

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

POUR JOUER :

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

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

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

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

EXTENSIONS POSSIBLES :

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

```

10 REM PERMUTATIONS
20 COLOR 0,11:COLOR 4,13:COLOR 5,7:GRAPH
IC 0,1
30 INPUT"OMBRE DE LETTRES ( DE 3 A 12 )
";N
40 IF N < 3 OR N > 12 THEN RUN
50 A$ = "ABCDEFGHJKLM"
60 C$ = " "
70 B$ = ""
80 X = INT(RND(1)*N + 1)
90 IF MID$(A$,X,1) = " " THEN 80
100 B$ = MID$(A$,X,1) + B$
110 MID$(C$,X,1)=MID$(A$,X,1)
120 MID$(A$,X,1)=" "
130 S = 1
140 IF LEN(B$) <> N THEN 80
150 C$ = LEFT$(C$,LEN(B$)):IF C$=B$ THEN
60
155 COLOR 5,7:PRINT"XXXXXXXXXXDEBUT  ";:COLOR
5,3
160 FOR I=1 TO LEN(B$):PRINT" ";MID$(B$,
I,1);:SOUND 1,400*I+800,6:NEXT I
165 PRINT:COLOR 5,2:PRINT"XXXXXXXXXXFIN  ";:CO
LOR 5,4
170 FOR I=1 TO LEN(C$):PRINT" ";MID$(C$,
I,1);:SOUND1,400*(LEN(C$)-I)+800,6:NEXTI
175 PRINT"XXXXXXXXXX
";:COLOR 5,1:PLAY"CEG"
180 PRINT"XXXXXXXXXXPERMUTATION NUMERO";S:
INPUT"XXXXXXXXXXVOTRE QUANTITE";L
190 S = S + 1
200 A$ = ""
210 FOR I = 1 TO L
220 A$ = A$ + MID$(B$,L-I+1,1)
230 NEXT I
240 B$ = A$ + RIGHT$(B$,LEN(B$)-L)
245 COLOR 5,7:PRINT"XXXXXXXXXXRESULTAT
";:COLOR 5,9
250 FOR I=1 TO LEN(B$)
253 PRINT" ";MID$(B$,I,1);:SOUND 1,ASC(M
ID$(B$,I,1))*600-30000,8
255 NEXT I
260 IF B$ <>C$ THEN 175
270 COLOR 5,7:PLAY"CDEFG":PRINT:PRINT"XXXX
TROUVE EN"S-1;" ESSAIS"
280 PRINT"XXXXXXXXXXAUTRE PARTIE ( O/N )":GETKEY
R$
290 IF R$ = "O" THEN RUN

```

Bridge

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : variable, selon le problème posé.

LE JEU :

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

LE PROGRAMME :

Les lignes 20 et 30 définissent un jeu de 52 cartes.

La variable B\$ (ligne 40) permettra d'afficher les symboles des couleurs et aussi d'améliorer la désignation de certaines cartes (10, valet, dame, roi et as).

La boucle J (50-190) réalise le tirage de quatre jeux de 13 cartes.

La boucle I (60-170) réalise tirage et impression. La carte est choisie parmi les 52 disponibles, en ligne 80.

La ligne 90 calcule la couleur de la carte tirée.

La ligne 100 vérifie que cette carte n'a pas déjà été tirée.

Les lignes 110 et 120 s'occupent de l'impression complémentaire dans le cas où l'on tire un 10, un valet, une dame, un roi ou un as.

L'impression est réalisée par les lignes 130, 140 et 150.

La ligne 160 permet d'éliminer les cartes déjà tirées, et la ligne 180 réalise un arrêt entre chaque affichage.

POUR JOUER :

On lance le programme par RUN.

Une première série de 13 cartes est affichée verticalement. Il suffira d'appuyer sur la touche SHIFT pour obtenir la deuxième série de 13 cartes.

On procédera de même pour la troisième et la quatrième.

EXTENSIONS POSSIBLES :

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

On peut aussi ajouter la possibilité de faire des annonces (et de les comparer à celles du C128, par exemple), ou introduire des possibilités de jeu de la carte, y compris pour le C128.

```

10 REM BRIDGE
15 COLOR 0,11:COLOR 4,13:COLOR 5,7:GRAPH
IC 0,1
20 R$="234567891VDRA"
30 R$=R$+R$+R$+R$
40 B$="CACOTRPI0AAOS"
50 FOR J = 1 TO 4
55 PRINT " ";
60 FOR I = 1 TO 13
70 F = I - 13*INT((I-1)/13)
80 X = INT(RND(1)*LEN(R$)+1)
90 S = 2*INT((X-1)/13)+1
100 IF MID$(R$,X,1)="0" THEN 80
110 T = X - 13*INT(X/13)
120 T = T - 13*(T=0)
130 COLOR 5,2-7*(T>=9):PRINTTAB(10*J-10)
;MID$(R$,X,1);
140 IF T >= 9 THEN PRINTMID$(B$,T,1);
150 COLOR 5,3+2*(S>3):PRINTTAB(10*J-6);M
ID$(B$,S,2)
160 MID$(R$,X,1)="0":SOUND 1,400*X+800,4
170 NEXT I
180 PLAY"CEG":PRINT"MAIRE - SHIFT - PO
UR CONTINUER":WAIT 211,1
190 NEXT J
200 PLAY"CDEFG"

```

Hanoi

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

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

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

LE PROGRAMME :

La variable D\$, en ligne 50, servira à dessiner les disques, et C\$, en ligne 120, dessine les socles.

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

La boucle I (80-100) permet de fabriquer une variable A (1) qui contient l'état de la première pile sous forme numérique (en ligne 40, trois variables sont prévues pour conserver l'état des trois piles).

Les socles sont dessinés par la ligne 120 et les trois piles par les boucles emboîtées I (150-210) et J (160-200).

Le contrôle du coup est effectué par les lignes 230, 240 et 270 et l'état des piles est modifié par les lignes 280 et 290.

POUR JOUER :

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

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

EXTENSIONS POSSIBLES :

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

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


```

10 REM HANOI
20 COLOR 0,13:COLOR 4,11:COLOR 5,1:GRAPH
   IC 0,1
30 DIM A(3)
40 A(1) = 9 : A(2) = 9 : A(3) = 9
50 D$ = " "
60 INPUT "NOMBRE DE TOURS ( MOINS DE 8 )"
   ;N
70 IF N <= 8 THEN RUN
80 FOR I = 0 TO N-1
90 A(1) = VAL(STR$(A(1))+STR$(N-I))
100 NEXT I
110 PRINT " "
120 C$ = " "
130 PRINT " " ; C$ ; TAB(12) ; C$ ; TA
   B(24) ; C$
140 COLOR 5,1:PRINT " " ; TAB(4) ; "1" ; TAB(16)
   ; "2" ; TAB(28) ; "3"
150 FOR I = 0 TO N-1
160 FOR J = 1 TO 3
170 B$ = RIGHT$(STR$(A(J)),LEN(STR$(A(J)
   ))-1)
180 PRINT " " ; FOR K = 0 TO 10-I : PRINT :
   NEXT K
190 IF LEN(B$) > I+1 THEN PRINT " " ; TAB(1
   2*J-11) ; LEFT$(D$,VAL(MID$(B$,I+2,1)))
195 PRINT " "
200 NEXT J
210 NEXT I
215 WINDOW 0,18,39,24,1:COLOR 5,9
220 SOUND 1,6000,8:INPUT "TOUR DE DEPART"
   ;D
230 IF A(D) = 9 THEN PLAY "CCC":GOTO 215
240 SOUND 1,8000,8:INPUT " " ; "TOUR D'ARRIVEE
   " ; A: IF A=D THEN PLAY "CCC":GOTO 215
250 B$ = RIGHT$(STR$(A(D)),LEN(STR$(A(D)
   ))-1)
260 C$ = RIGHT$(STR$(A(A)),LEN(STR$(A(A)
   ))-1)
270 IF MID$(B$,LEN(B$),1) > MID$(C$,LEN(
   C$),1) THEN PLAY "CCC":GOTO 215
280 A(A) = VAL(C$+MID$(B$,LEN(B$),1))
290 A(D) = VAL(LEFT$(B$,LEN(B$)-1))
300 WINDOW 0,0,39,24:GOTO 110

```

Répétitions

REFERENCES :

Difficulté de programmation : facile.

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

Difficulté du jeu : moyenne.

LE JEU :

Des nombres vont apparaître un par un.

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

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

LE PROGRAMME :

La ligne 30 appelle un sous-programme (lignes 1000 à 1060) qui donne la règle du jeu.

La ligne 50 fabrique les chiffres, qui sont ensuite stockés dans une variable alphanumérique A\$ (ligne 60).

La ligne 70 permet l'impression du dernier chiffre au hasard sur la quatrième ligne de l'écran, et la durée de l'affichage est déterminée par la ligne 75.

La ligne 85 attend la réponse du joueur, qui est comparée à la réponse correcte en lignes 90 et 100.

La ligne 110 marque les points.

La ligne 120 contrôle le meilleur score.

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

POUR JOUER :

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

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

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

On peut aussi quitter le programme à tout moment, en tapant RUN/STOP et RESTORE avant l'affichage d'un chiffre.

EXTENSIONS POSSIBLES :

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

```

10 REM REPETITIONS
20 M = 0
25 A$="":R$="":H=0
30 GOSUB 1000
40 DO
50 X = INT(RND(1)*10)
60 A$ = A$ + RIGHT$(STR$(X),1)
70 COLOR 5,9:PRINT"10000";TAB(RND(1)*40)
X:SOUND 1,X*400+1000,3
75 FOR I = 1 TO 200 + RND(1)*300:NEXT I
80 PRINT"J"
85 COLOR 5,7:INPUT "XXXXXXXXXXXXVOTRE SEQ
UENCE";R$
90 SOUND 1,10000,5:IF R$<>A$ THEN EXIT
100 IF R$=A$ THEN COLOR 5,2:PRINT"EXAC
T . ATTENTION"
105 SLEEP 2
110 H = H + 1
120 IF M < H THEN M = H
130 LOOP
200 PRINT"J"
210 COLOR 5,1:PRINT"VOTRE SCORE EST DE"H
220 PRINT"MEILLEUR SCORE ACTUEL"M:PLAY"C
EG"
230 INPUT"MAUTRE PARTIE ( O/N )";R$
240 IF R$="N" THEN PLAY"CDEFG":END
250 GOTO 25
1000 COLOR 0,14:COLOR 4,12:COLOR 5,7:GRA
PHIC 0,1
1010 PRINT"VOUS DEVEZ REPETER TOUTE LA S
ERIE DES"
1020 PRINT"NOMBRES APPARUS DEPUIS LE DEB
UT DU JEU"
1030 PRINT:PRINT"IL Y EN A UN DE PLUS A
CHAQUE FOIS"
1040 PRINT:PRINT"APPUYER SUR - SHIFT - P
OUR COMMENCER"
1050 WAIT211,1
1060 RETURN

```

Jeu de loto

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : hasard (plusieurs joueurs).

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

On utilisera un tableau N(8), dimensionné en ligne 30, pour enregistrer les résultats.

Le tirage des nombres est effectué dans une boucle I (40-100), une deuxième boucle J (60-90) étant utilisée pour contrôler qu'il n'y a pas deux tirages identiques.

Le tirage (ou le retraitage) est effectué en ligne 50.

La boucle I (120-140) affiche le résultat, et la ligne 150 affiche le numéro complémentaire.

Les lignes 160 à 180 s'occupent du cas où l'on voudrait refaire une autre partie.

POUR JOUER :

Il suffit de taper RUN pour lancer le programme et obtenir l'affichage d'un premier loto.

La question VOULEZ-VOUS UN AUTRE TIRAGE (O/N) s'affiche en bas de l'écran ; on répondra O si l'on désire continuer.

EXTENSIONS POSSIBLES :

On peut imaginer un programme qui pourrait tenir compte, soit de ses propres résultats, soit de résultats entrés par un joueur, pour améliorer le tirage, en tenant compte des probabilités de sortir un numéro donné.

On peut aussi envisager un programme qui réaliserait des cartons pour le jeu de loto plus usuel, tel qu'il est encore couramment pratiqué dans le Midi de la France.

```

10 REM JEU DE LOTO
20 COLOR 0,15:COLOR 4,13:COLOR 5,9:GRAPH
IC 0,1
30 DIM N(8)
40 I=1
50 N(I)=INT(RND(1)*49)+1
60 IF I=1 THEN 100 : ELSE J=1
70 DO WHILE J<I
80 IF N(I) = N(J) THEN 50
90 J=J+1:LOOP
100 I=I+1:IF I<8 THEN 50
110 PRINT"###";TAB(8);"TIRAGE DU LOTO"
115 PRINT:PRINT
120 FOR I = 1 TO 6
130 COLOR 5,2:PRINTTAB(5*I-2);N(I);:SOUN
D 1,400*N(I)+500,5
140 NEXT I
145 PRINT
150 COLOR 5,12:PRINT"NUMERO COMPLEMENTA
IRE":COLOR 5,14:PRINTN(7):PLAY"CEG"
160 COLOR 5,4:PRINT"###VOULEZ-VOUS UN AU
TRE TIRAGE ( O/N )"
170 GETKEY A$
180 IF A$="O" THEN RUN:ELSE PLAY"CDEFG"

```

Roulette

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : hasard (plusieurs joueurs).

Difficulté du jeu : facile.

LE JEU :

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

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

LE PROGRAMME :

La boucle K (20-70) dessine le terrain de jeu, auquel la ligne 80 rajoute les trois symboles P, 00 et I.

La boucle K (95-160) réalise le déplacement de la boule, l'arrêt étant contrôlé par la ligne 125.

La boucle J (ligne 200) réalise une légère temporisation.

La ligne 170 tire la position d'arrêt.

La ligne 125 gère les essais suivants.

POUR JOUER :

Il suffit de faire RUN, et de regarder.

Une fois la boule arrêtée, il suffira de frapper une touche quelconque pour la relancer. Taper RUN/STOP et RESTORE pour sortir du programme.

EXTENSIONS POSSIBLES :

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

```

10 REM ROULETTE
15 COLOR 0,6:COLOR 4,6:COLOR 5,2:GRAPHIC
  0,1
20 FOR K = 1 TO 4
30 FOR I = 1 TO 9
35 S = I + I
40 PRINTTAB(4*I)(SAND (K=1))+(S+18AND(K=
2))+(S-1AND(K=3))+(S+17AND(K=4));
50 NEXT I
60 PRINT:PRINT:PRINT
70 NEXT K
80 PRINT"TTTTTTTTTTTTT P":PRINT:PRINT:PR
INT:PRINT"00":PRINT:PRINT:PRINT:PRINT"I"
90 PRINT"TTTTTTTTT"
95 FOR K = 1 TO 4
100 FOR I = 1 TO 10
105 IF K*I=40 THEN PRINT:GOTO 130
110 COLOR 5,1:PRINTTAB(4*I+1)"●":SOUND
  1,10000,10
115 X=X-1:T=T+1
120 GOSUB 200
125 IF X=1 THEN PLAY"CEG":GETKEY A$:RUN
130 PRINT"III ";
140 NEXT I
150 PRINT"000";
160 NEXT K
170 X = INT(38*RND(1)):T=100-X
190 PRINT"TTTTTTTTT":GOTO 90
200 FOR J=1 TO 3*T:NEXT J
210 RETURN

```

Musique

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu éducatif.

Difficulté du jeu : moyenne.

LE JEU :

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

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

LE PROGRAMME :

La boucle I (30-60) dessine la portée. Une deuxième boucle I (70-110) affiche les numéros des lignes.

La ligne 120 définit le nom des notes, et la ligne 130 réalise le tirage aléatoire d'une note.

Le nom de la note est calculé à la ligne 140 et il est affiché par les lignes 160, 170 et 180, suivi de la mention GRAVE ou AIGUE, selon le cas.

La réponse sera entrée en ligne 190.

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

La ligne 220 affiche la note.

Les lignes 250 et 260 regardent si l'on a gagné ou perdu et affichent le message correspondant.

La ligne 270 permet de faire un nouvel essai, en appuyant sur SHIFT.

POUR JOUER :

On lance le programme par RUN.

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

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

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

On doit taper la touche SHIFT pour continuer, et s'arrêter en tapant RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

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


```

10 REM MUSIQUE
20 COLOR 0,2:COLOR 4,2:COLOR 5,7:GRAPHIC
  0,1
30 FOR I = 1 TO 5
40 Z = 2*I + 4 : GOSUB 500
50 PRINTTAB(4);"■"-----
  _"
60 NEXT I
65 PRINT"□"
70 FOR I = 1 TO 14
80 Z = I+2 : GOSUB 500
90 IF I <= 9 THEN PRINTI
100 IF I > 9 THEN PRINT" ";CHR$(I+55)
110 NEXT I
120 N$ = "SI LA SOLFA MI RE DO ":P$="BAG
FEDC"
130 X = INT(RND(1)*14+1)
140 T = X - 1 - 7*INT((X-1)/7)
150 Z = 18 : GOSUB 500
160 PRINT"NOTE ";MID$(N$,3*T+1,3);
170 IF X <= 7 THEN PRINT" AIGUE"
180 IF X > 7 THEN PRINT" GRAVE"
190 INPUT"NUMERO DE LA LIGNE";L$
200 L = - (ASC(L$)-48)*(L$ < ":") - (ASC
(L$)-55)*(L$ >= "A")
210 Z = X : GOSUB 500
220 PRINT"■";TAB(14);" I":PRINTTAB(14);" I
":PRINTTAB(14);"●"
230 PRINT"□"
240 Z = 20 : GOSUB 500
250 IF L <> X THEN PRINTTAB(25);"PERDU":G
OTO 270
260 PRINTTAB(25);"BRAVO":IF X>7 THEN BEG
IN:PLAY"03"+MID$(P$,T+1,1)
265 ELSE PLAY"04"+MID$(P$,T+1,1):BEND
270 PRINT"FAIRE - SHIFT - POUR CONTINUE
R"
280 WAIT211,1:WAIT 211,1,1
290 RUN
500 PRINT"§";
510 FOR P = 0 TO Z
520 PRINT
530 NEXT P
540 RETURN

```

A travers

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

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



LE PROGRAMME :

Les positions des chiffres et des lettres sont enregistrées dans une table S (5, 5) ; les lignes 10 à 80 définissent les valeurs de départ.

Les lignes 100 à 160 dessinent le terrain de jeu et les lignes 180 et 190 affichent le joueur qui a le trait.

La ligne 210 contrôle le coup joué, de même que les boucles I et J (200 à 240), et le changement de joueur est assuré par la ligne 330.

POUR JOUER :

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

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

On pourra ainsi faire 2E suivi de RETURN.

Si le coup est 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 C128.

```

10 REM A TRAVERS
20 K = 1
30 DIM S(5,5)
40 FOR I = 1 TO 4
50 N(I,1) = ASC("0") + I
60 N(4,I) = ASC("?")+ I
70 NEXT I
80 N(4,1) = 0
90 COLOR 0,13:COLOR 4,15:COLOR 5,7:GRAPH
IC 0,1
100 FOR U = 1 TO 4
110 FOR V = 1 TO 4
120 Z = 2*V : GOSUB 1000
130 IF N(V,U) = 0 THEN COLOR 5,1:PRINTTA
B(2*U);"."
140 IF N(V,U) <>0 THEN COLOR 5,2:PRINTTA
B(2*U);CHR$(N(V,U))
150 NEXT V
160 NEXT U
170 Z = 15 : GOSUB 1000
180 COLOR 5,7:PRINT"JOUER AYANT LES ";
IF K = 0 THEN PRINT"LETTRES":PLAY"CE"
190 IF K=1 THEN PRINT"CHIFFRES":PLAY"DE"
200 COLOR 5,9:INPUT"VOTRE COUP ( N , S
, E , 0 )";R$
210 IF ( K = 1 AND RIGHT$(R$,1)="0" ) OR
( K = 0 AND RIGHT$(R$,1)="S") THEN 170
220 FOR I = 1 TO 4
230 FOR J = 1 TO 4
240 IF N(I,J)=ASC(LEFT$(R$,1)) THEN 280
250 NEXT J
260 NEXT I
270 GOTO 170
280 A = I : B = J
290 I = I - ( RIGHT$(R$,1)="S") + (RIGHT
$(R$,1)="N")
300 J = J - ( RIGHT$(R$,1)="E") + (RIGHT
$(R$,1)="0")
310 IF ( I=0 AND K=0 ) OR ( J>4 AND K=1
) THEN 340
320 IF N(I,J) <>0 THEN 170
330 N(I,J) = ASC(LEFT$(R$,1))
340 N(A,B) = 0
350 K = ABS(K-1)
360 GOTO 90
1000 PRINT" ";
1010 FOR F = 0 TO Z
1020 PRINT
1030 NEXT F
1040 RETURN

```

Le jeu du 15

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

C'est un jeu pour deux joueurs, assez ancien, qui était autrefois pratiqué dans les foires. On dispose d'un tapis de jeu de 9 cases, et chacun mise à son tour un jeton de sa couleur, sur une case disponible.

Le premier des deux joueurs qui, avec exactement 3 jetons, arrive à faire un total de 15, a gagné.

LE PROGRAMME :

La boucle I (50-80) dessine le terrain de jeu, et les lignes 90 à 110 s'occupent de la gestion du tour des joueurs (numéro 1 ou 2 pour chaque joueur, et jeton O ou X correspondant).

La ligne 120 attend le coup joué.

La boucle I (130-150) vérifie la validité du coup.

La ligne 160 teste la fin de partie, en cas de match nul, et la ligne 170 refuse les coups non conformes (avec retour à la situation initiale, ligne 90).

Les lignes 200 à 230 s'occupent de l'affichage du coup, et la ligne 240 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 270 à la ligne 340 : 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 350.

POUR JOUER :

Lancer le jeu en tapant RUN.

Le terrain de jeu s'affiche, puis la question JOUEUR 1 : X, et le joueur qui a choisi les X devra donner le numéro de la case où il désire jouer si elle est libre, son coup y sera enregistré.

On obtient, ensuite, l'affichage JOUEUR 2 : O et le deuxième joueur procède de la même façon. Si l'un des joueurs gagne (total de 15 sur 3 coups), il obtient l'affichage BRAVO et la partie s'arrête.

En cas de match nul, la machine affiche C'EST FINI.

On fera une autre partie en tapant RUN.

EXTENSIONS POSSIBLES :

Le plus intéressant sera sans doute d'essayer de définir une stratégie machine (en essayant d'éviter que la machine ne joue toujours le 5 lorsqu'elle doit commencer).

On peut aussi, bien sûr, améliorer le graphisme et introduire des effets sonores pour les différents messages.

```

10 REM LE JEU DU 15
15 COLOR 0,13:COLOR 4,15:COLOR 5,1:GRAPH
IC 0,1
20 F = 0
30 R$ = "123456789"
40 O$ = "" : X$ = ""
50 FOR I = 0 TO 24 STEP 3
60 PRINT"000";TAB(I+1);"□":PRINTTAB(I+1)
);"|"":PRINTTAB(I+1);"□"
70 PRINTTAB(I+1);(I+3)/3
80 NEXT I
90 COLOR 5,9:PRINT"00000000 JOUEUR ";F+1;
100 IF F=1 THEN COLOR 5,3:PRINT" : O":PL
AY"C"
110 IF F=0 THEN COLOR 5,7:PRINT" : X":PL
AY"G"
120 GETKEY A$ : IF A$ < "1" OR A$ > "9"
THEN SOUND 1,2000,3:GOTO 120
130 FOR I = 1 TO LEN(R$)
140 IF A$ = MID$(R$,I,1) THEN GOSUB 200
150 NEXT I
160 IF VAL(R$)=0 THEN COLOR 5,2:PRINT"000
0000000000000000 C'EST FINI":PLAY"EEE":END
170 GOTO 90
200 IF F = 1 THEN O$ = O$ + A$
210 IF F = 0 THEN X$ = X$ + A$
220 IF F=1 THEN PRINT"000"TAB(3*I-1)"O"
230 IF F=0 THEN PRINT"000"TAB(3*I-1)"X"
240 R$ = LEFT$(R$,I-1) + "0" + RIGHT$(R$
,LEN(R$)-I)
250 IF F = 1 THEN T$ =O$
260 IF F = 0 THEN T$ = X$
270 IF LEN (T$) < 3 THEN 350
280 FOR I = 1 TO 3
285 IF I = LEN(T$) THEN 350
290 FOR J = I + 1 TO LEN(T$)
295 IF J = LEN(T$) THEN 350
300 FOR K = J + 1 TO LEN(T$)
310 IF VAL(MID$(T$,I,1))+VAL(MID$(T$,J,1)
)+VAL(MID$(T$,K,1))=15 THEN 400
320 NEXT K
330 NEXT J
340 NEXT I
350 F = ABS(F-1)
360 RETURN
400 IF F = 1 THEN COLOR 5,3:PRINT"00000000
000BRAVO , JOUEUR O"
410 IF F = 0 THEN COLOR 5,7:PRINT"00000000
000BRAVO , JOUEUR X"
420 PLAY"CDEFG"

```


Niveau 4

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Le loup et les agneaux	Un jeu de réflexion pour 2 joueurs.
Un dé	Tirage d'un dé, avec graphisme.
La bombe	Saurez-vous la découvrir à temps (1 joueur) ?
Diago	Etes-vous vraiment sûr de vos réflexes ?
Décomaque	Pour vous éviter de sécher trop longtemps sur Nicomaque.
Le jeu du missile	Une mission bien périlleuse (un seul joueur).
Ardoise magique	Mettez au point vos créations graphiques, sans papier ni crayon.
Affichage publicitaire	Si vous ne savez pas quoi mettre en vitrine.
Le singe	Un jeu de lettres pour 2 joueurs, où il faut de bons réflexes.
Mathieu	Vous ne savez pas dessiner ? Laissez faire votre ordinateur.
Adresse	Jeu de réflexe pour un joueur, très visuel.
Sous-marin	Un classique des jeux vidéos (1 joueur).
Zombies	Un peu farfelu, mais très visuel (1 joueur).
Calculatrice	Un programme de simulation très réaliste.
Clavier	Apprenez à taper à la machine en vous amusant.
Chasse au sous-marin	Il faudra un peu de réflexion, et beaucoup de chance.
Le lézard d'Alice	Un pays des merveilles où le hasard semble roi.
Dessin de dés	Suffisamment souple pour être utilisé partout.
Evasion	Heureusement, ce n'est qu'un jeu.
Chapeaux	Soyez très observateurs et vous réussirez (2 joueurs).
Combat aérien	Pas si facile de descendre un appareil qui bouge tout le temps.
L'indésirable	Arriverez-vous à mettre la main sur ce locataire indésirable.
Course de chevaux	Un jeu très visuel, pour plusieurs joueurs.
CREASPRITE	Pour définir des lutins en DATA.

Les techniques d'animation élémentaire

SAISIE AU VOL D'UN CARACTERE FRAPPE AU CLAVIER : GET

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.

Cette possibilité est disponible par la fonction GET, suivie d'un nom de variable alphanumérique, GETA\$ par exemple, qui enregistre le nom de la dernière touche frappée ; on pourra d'ailleurs enregistrer dans A\$ une suite de caractères frappés, qui seront restitués à chaque lecture de A\$, dans l'ordre de la frappe (il est nécessaire qu'il y ait frappe : on ne devra pas laisser le doigt appuyé sur la touche concernée).

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

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

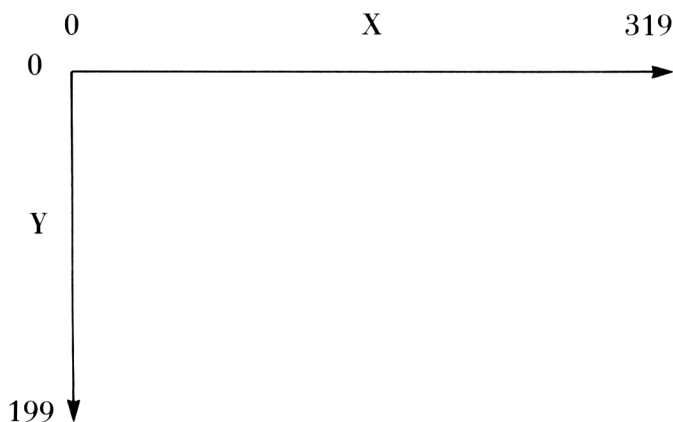
LE GRAPHISME HAUTE RESOLUTION

En plus du mode texte, obtenu avec GRAPHIC 0, le Commodore 128 offre 5 modes graphiques, numérotés de GRAPHIC 1 à GRAPHIC 5. Le dernier, qui n'est utilisable qu'avec le moniteur 80 colonnes, n'a pas été employé ici.

GRAPHIC 1 est le mode graphique standard, sans texte.

GRAPHIC 2 permet de partager l'écran entre une partie graphique et une partie texte.

GRAPHIC 3 est un mode permettant l'utilisation de lutins multicolores. GRAPHIC 4 permet de réserver une partie texte dans ce même mode. L'écran disponible en mode graphique est repéré comme suit :



Les coordonnées sont toujours entrées dans l'ordre : colonne (X), puis ligne (Y).

L'instruction DRAW permet de placer des points sur l'écran graphique, ou de tracer des lignes.

L'instruction BOX permet de tracer des rectangles, dans tous les sens (on peut les faire tourner) qui peuvent être peints ou non.

L'instruction CIRCLE autorise le tracé de cercles, d'ellipses, d'arcs de cercles, etc.

L'instruction PAINT permet de peindre une figure fermée.

L'instruction CHAR rend possible l'affichage de texte sur l'écran graphique.

L'instruction SSHAPE permet de "photographier" une zone de l'écran graphique, enregistrée dans une variable alphanumérique.

L'instruction GSHAPE permet de restituer à un endroit quelconque de l'écran la zone photographiée par SSHAPE.

OBJETS GRAPHIQUES AUTONOMES : LES SPRITES OU LUTINS

Les SPRITES sont des objets graphiques qui sont manipulés par le processeur vidéo, ce qui signifie qu'une fois affichés ou mis en mouvement, ils disposent d'une existence autonome, le programme pouvant continuer à se dérouler sans qu'il soit nécessaire de s'en occuper.

Il est néanmoins possible de les contrôler, grâce à certaines instructions spécifiques, de maniement un peu délicat. Le C128 autorise 8 SPRITES simultanés.

La création de SPRITES est facilitée, sur le C128, par l'instruction SPRDEF, qui appelle un générateur de SPRITES situé en mémoire morte. D'autres possibilités existent : soit grâce à SSHAPE, soit avec des méthodes spécifiques (voir le programme CREASPRITE, en fin de chapitre).

L'instruction SPRITE permet d'allumer ou d'éteindre un SPRITE.

L'instruction MOVSPR le positionne sur l'écran, ou provoque son déplacement automatique, avec une vitesse et dans une direction données.

L'instruction RSPPOS renvoie des indications sur la position et la vitesse d'un SPRITE.

L'instruction COLLISION permet l'exécution d'un sous-programme en cas de rencontre entre 2 SPRITES, ou entre un SPRITE et un objet graphique.

L'instruction BUMP renvoie des informations sur la nature de la collision.

Le loup et les agneaux

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

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

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

LE PROGRAMME :

Une variable A\$, en ligne 17 contient les positions de départ.

La boucle I (80-160) dessine le damier, et les lignes 170 à 270 placent le loup et les agneaux.

La ligne 280 gère le changement de joueur.

La ligne 300 affiche JEU DU LOUP si c'est le joueur qui manipule le loup qui a le trait, et la ligne 310 JEU DES AGNEAUX dans le cas contraire.

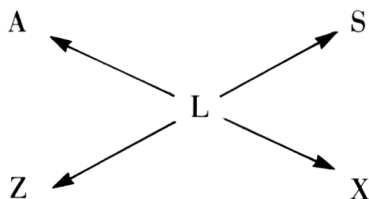
Les lignes 330 à 340 ou 350 à 390 analysent la réponse, en fonction du joueur concerné.

POUR JOUER :

RUN déclenche l'arrêt sur l'attente d'une entrée : Ø et RETURN si l'on veut que les agneaux commencent, et 1 pour le loup. Après quoi, le damier s'affiche et la phrase JEU DU LOUP (A, S, Z, X) pour le loup et JEU DES AGNEAUX (ND et NG) pour les agneaux.

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

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



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

EXTENSIONS POSSIBLES :

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

On pourra aussi envisager de jouer sur un damier de 10 cases sur 10 cases, ce qui augmente légèrement la difficulté du jeu pour les agneaux. Mais l'extension la plus intéressante à réaliser sera certainement la réalisation d'un programme permettant au C128 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 C128).

```
10 REM LE LOUP ET LES AGNEAUX
12 COLOR 0,12:COLOR 4,11:COLOR 5,7:GRAPH
IC 0,1
15 INPUT"AGNEAUX (0) OU LOUP (1)";S
17 A$="8183858714"
20 FOR I=1 TO 21:READ A,B,C:F$=F$+CHR$(A
)+CHR$(B)+CHR$(C):NEXT I
22 FOR I=1 TO 21:READ A,B,C:E$=E$+CHR$(A
)+CHR$(B)+CHR$(C):NEXT I
25 SPRSAV F$,1:SPRSAV F$,2:SPRSAV F$,3:S
PRSAV F$,4:SPRSAV E$,5
27 COLOR 1,4:COLOR 4,9:GRAPHIC 4,1,20
30 D$ = "      "
40 B$ = "      "
50 C$ = "      "
60 CHAR 1,0,0,"      A    B    C    D    E
   F    G    H"
80 FOR I = 1 TO 8
90 FOR J = 1 TO 8
100 IF (I+J)/2 = INT((I+J)/2) THEN 140
110 CHAR 3,4*J,2*I-1,D$,1:CHAR 1,0,2*I,C
HR$(48+I)
120 CHAR 3,4*J,2*I,B$,1:CHAR 3,4*J,2*I+1
,C$,1
140 NEXT J
160 NEXT I
170 F=4*VAL(MID$(A$,10,1)):G = 2*VAL(MID
$(A$,9,1))
180 CHAR 1,F,G,"L":SPRITE 5,1,1,0,0,0,0:
MOVSPR 5,8*F+32,8*G+46
200 F=4*VAL(MID$(A$,2,1)):G = 2*VAL(MID$
(A$,1,1))
210 CHAR 1,F,G,"1":SPRITE 1,1,6,0,0,0,0:
MOVSPR 1,8*F+34,8*G+46
220 F=4*VAL(MID$(A$,4,1)):G = 2*VAL(MID$
(A$,3,1))
```

```

230 CHAR 1,F,G,"2":SPRITE 2,1,6,0,0,0,0:
MOVSPR 2,8*F+34,8*G+46
240 F=4*VAL(MID$(A$,6,1)):G = 2*VAL(MID$(
A$,5,1))
250 CHAR 1,F,G,"3":SPRITE 3,1,6,0,0,0,0:
MOVSPR 3,8*F+34,8*G+46
260 F=4*VAL(MID$(A$,8,1)):G = 2*VAL(MID$(
A$,7,1))
270 CHAR 1,F,G,"4":SPRITE 4,1,6,0,0,0,0:
MOVSPR 4,8*F+34,8*G+46
280 S = ABS(S-1):SOUND 1,4000*S+4000,6
300 COLOR 5,1:IF S = 0 THEN INPUT"JEU DU
LOUP (A,S,Z,X)":R$
310 IF S = 1 THEN INPUT"JEU DES AGNEAUX
(ND OU NG)":R$:GOTO 350
330 C = 10
335 A = VAL(RIGHT$(A$,2))+(C+1 AND R$="X
")+(C-1 AND R$="Z")
337 A = A - (C+1 AND R$="A") - (C-1 AND
R$="S")
340 GOTO 380
350 C = 2*VAL(MID$(R$,1,1))
360 A = VAL(MID$(A$,C-1,2))-(9 AND RIGHT
$(R$,1)="D")-(11 AND RIGHT$(R$,1)="G")
380 A$ = LEFT$(A$,C-2)+RIGHT$(STR$(A),2)
+RIGHT$(A$,LEN(A$)-C)
390 GOTO 80
1000 DATA 0,0,0,0,0,0,0,3,128,0,7,64,15,
143,240,63,216,240,255,240,96,63,240
1005 DATA 0,63,224,0,48,96,0,40,144,0,72
,136,0
1010 DATA 72,132,0,136,130,0,144,132,0,1
44,132,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1020 DATA 0,0,0,0,0,0,0,0,0,0,128,0,3,0,
0,7,0,0,11,62,0,29,255,0,24,255,224
1025 DATA 1,255,240,1,255,248,0,255,228,
1,128,98
1030 DATA 2,128,161,4,129,33,8,66,33,8,6
6,32,0,0,0,0,0,0,0,0,0,0,0,0,0

```

Un dé

REFERENCES :

Difficulté de programmation : facile.

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

Difficulté du jeu : aucune.

LE JEU :

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

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

LE PROGRAMME :

La ligne 30 utilise le générateur aléatoire du C128 pour tirer un nombre de 1 à 6.

La ligne 40 dessine les bords du dé.

Les lignes 50 à 110 sont utilisées pour le dessin des points.

Les lignes 120 et 130 permettront le cas échéant de refaire un tirage.

POUR JOUER :

Lancer le programme par RUN.

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

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

EXTENSIONS POSSIBLES :

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

```

10 REM UN DE
20 COLOR 0,1:COLOR 4,1:COLOR 1,3:GRAPHIC
  1,1
30 A=INT(RND(1)*6+1)
40 BOX 1,100,50,198,150,0,1
50 COLOR 1,14:IF A<>1 THEN CHAR 1,15,8,"
  ",1
60 IF A>=4 THEN CHAR 1,21,8," ",1
70 IF A=6 THEN CHAR 1,15,12," ",1:CHAR 1
  ,21,12," ",1
80 IF A/2<>INT(A/2) THEN CHAR 1,18,12,"
  ",1
90 IF A>=4 THEN CHAR 1,15,16," ",1
100 IF A<>1 THEN CHAR 1,21,16," ",1
110 PLAY"CEG":CHAR 1,0,22,"AUTRE TIRAGE
  ( O/N )"
120 GETKEY R$
130 IF R$="O" THEN RUN
140 PLAY"CDEFG":GRAPHIC 0,1

```

La bombe

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

Vous êtes censé trouver, à l'aide d'un détecteur spécialisé, un colis radioactif piégé disposé par un voyageur mal intentionné, dans une chambre d'hôtel. Seul problème, mais de taille : l'hôtel comporte un million de chambres (100*100*100).

Votre détecteur affiche, en permanence, un signal qui vous renseigne sur la proximité de la bombe, et le temps qui vous reste est affiché en permanence, décompté de 10 en 10.

Pourrez-vous découvrir la bombe avant l'explosion ?

LE PROGRAMME :

Une variable numérique R, non dimensionnée, contient les trois coordonnées définissant la position du joueur.

La position de la chambre contenant le colis piégé est définie par un tirage aléatoire de 3 variables A, B, C, aux lignes 30 à 50.

Les lignes 70 à 90 dessinent le détecteur.

Le signal de proximité est calculé aux lignes 100 et 110 et la ligne 120 permet l'affichage des coordonnées de la chambre à découvrir, dans le cas où l'on a perdu.

La ligne 150 permet l'affichage, à la fois du signal et du temps restant à courir.

La ligne 160 réalise le décompte du temps.

La ligne 170 contrôle si l'on a trouvé ou non.

La boucle I (240-280) permet d'entrer les nouvelles coordonnées, en utilisant le sous-programme placé aux lignes 300 à 350.

POUR JOUER :

Après le lancement du programme par RUN, le signal et le temps (en principe 200) sont affichés.

Le joueur est alors en position $x=0$, $y=0$, $z=0$, et on devra introduire successivement les trois coordonnées x, y, z de la chambre à explorer.

On fera, par exemple, 13 puis RETURN, 45 puis RETURN, etc.

Le nouveau signal est alors affiché, et le temps est décrémenté de 10.

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

EXTENSIONS POSSIBLES :

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


```

10 REM LA BOMBE
20 L = 100 : T = 200
30 A = INT(RND(1)*L)
40 B = INT(RND(1)*L)
50 C = INT(RND(1)*L)
60 COLOR 0,15:COLOR 4,12:COLOR 5,7:GRAPH
IC 0,1:D$=" "
70 Z = 4 : GOSUB 360:COLOR 5,7:PRINT D$
80 Z = 5 : GOSUB 360 : PRINT D$
90 Z = 6 : GOSUB 360 : PRINT D$
100 S = ABS((A/L+B+C*L)-(R(1)/L+R(2)+R(3)
)*L))
110 S$ = LEFT$(STR$(S),7) : S = VAL(S$)
120 IF T=0 THENCOLOR5,4:PRINT"PERDU :
SOLUTION : "A;" " ;B;" " ;C:PLAY"CCC":END
130 COLOR 5,2
140 Z = 5 : GOSUB 360
150 PRINTTAB(2);S;TAB(12);T:PLAY"A"
160 T = T - INT(SQR(L))
170 IF S = 0 THEN Z = 21 : GOSUB 360 : C
OLOR 5,1:PRINT"BRAVO":PLAY"CDEFG":END
180 COLOR 5,9 : Z = 10 : GOSUB 360
190 PRINT" - X : "
200 Z = 12 : GOSUB 360
210 PRINT" - Y : "
220 Z = 14 : GOSUB 360
230 PRINT" - Z : "
240 FOR I = 1 TO 3
250 R$ = ""
260 GOSUB 300
270 R(I) = VAL (R$)
280 NEXT I
290 GOTO 70
300 Z = 8 + 2*I : GOSUB 360
310 COLOR 5,1:PRINTTAB(7);"? ";R$;" " :
SOUND 1,3000+1000*I,6
320 GET A$ : IF A$ = "" THEN 320
330 IF ASC(A$) = 13 THEN RETURN
340 R$ = R$ + A$
350 GOTO 300
360 PRINT" "
370 FOR P = 0 TO Z
380 PRINT
390 NEXT P
400 RETURN

```

Diago

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : réflexe (1 seul joueur).

Difficulté du jeu : difficile.

LE JEU :

On lance le programme par RUN et on obtient un premier affichage qui vous renseigne sur la performance à réaliser (par exemple, 5 points noirs ou 17 points noirs...), puis le temps est affiché et décompté, à partir de 10 secondes. Le jeu commence lorsque l'on arrive à 0 seconde. Des ronds apparaissent alors en bas et à gauche de l'écran, formant une ligne diagonale, et il vous faudra appuyer sur la touche 0 pour arrêter l'affichage des points dès que vous estimez que le nombre de points demandé au début est apparu à l'écran.

On obtient alors un affichage BRAVO ou PERDU selon le résultat obtenu.

LE PROGRAMME :

Le nombre de points noirs sera choisi au hasard, en ligne 30.

Le temps est décompté par les lignes 40 à 80.

Les points sont imprimés et la ligne diagonale est fabriquée par la ligne 105.

La pression de la touche 0 est détectée en ligne 140, et la ligne 150 contrôle la fin de partie (aucun réflexe).

Les affichages de fin sont exécutés par les lignes 170 à 200.

POUR JOUER :

On lancera l'exécution par RUN.

On obtient alors l'affichage de la phrase LE BON REFLEXE EST A... qui indique le nombre de points gagnants.

Le temps sera ensuite décompté sous la forme ATTENTION : IL RESTE ... SECONDES qui indique le temps restant avant l'apparition de la diagonale.

À l'affichage de 0, la diagonale apparaît et commence à se déplacer sur l'écran.

Si l'on estime que le nombre de points réclamé a été atteint, on devra presser la touche 0.

L'affichage indique alors BRAVO si l'on a gagné, ou PERDU dans le cas contraire ; puis la phrase AUTRE ESSAI (O/N) est affichée, répondre O pour faire une autre partie.

EXTENSIONS POSSIBLES :

On pourra envisager une adaptation du jeu à deux joueurs ou plus, en laissant la diagonale se former complètement et en testant la coïncidence lors de la pression de la touche affectée au joueur.

On peut aussi réaliser un graphisme plus soigné et modifier le principe du déplacement des objets sur l'écran (déplacement horizontal, vertical ou apparition au hasard...).

```

10 REM DIAGO
20 COLOR 0,13:COLOR 4,15:COLOR 5,2:GRAPH
IC 0,1
30 X = INT(RND(1)*18 + 3)
40 TI$ = "000000":B=10
50 PRINT"LE BON REFLEXE EST A "X
60 A=10-VAL(TI$):A$=STR$(A):IF A<>B THEN
  SOUND 1,12000,1:B=A
70 PRINT"ATTENTION : IL RESTE ";A$
;" SECONDES "
80 IF A <> 0 THEN 60
90 PRINT"3":PLAY"V1"
100 U = 0
105 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXXXX";TAB
(U);"1003"
110 FOR I = 1 TO 30 : NEXT I
120 U = U + 1
130 GET A$:COLOR 5,3
140 IF A$ = "0" THEN 170
150 IF U > 22 THEN PRINT"3PERDU":PLAY"C
CC":GOTO 190
160 GOTO 105
170 IF U <> X THEN PRINT"3PERDU":PLAY"GE
C":GOTO 190
180 PRINT"3BRAVO":PLAY"CEG"
190 PRINT"MAUTRE ESSAI (O/N)":GETKEY R$
200 IF R$="O" THEN RUN:ELSE PLAY"CDEFG"

```

Décomaque

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : solution du programme Nicomaque.

Difficulté du jeu : aucune.

LE JEU :

C'est une réplique au programme Nicomaque, proposé dans les pages précédentes ; vous entrez les restes successivement dans les divisions par 3, 5 et 7 d'un entier compris entre 1 et 100, et le C128 calcule et affiche cet entier. Le procédé retenu peut d'ailleurs être généralisé (à vous de trouver comment).

LE PROGRAMME :

Les lignes 30 à 50 affichent quelques explications.

La ligne 90 attend que vous frappiez un reste, qui sera refusé s'il est incompatible avec la division par le nombre affiché.

La ligne 100 emmagasine les restes dans une variable non dimensionnée A (S).

La ligne 130 réaffiche les restes entrés.

Les boucles I (140-180) et J (150-170) dessinent des petits carrés selon les restes introduits.

Les lignes 210 à 240 gèrent la fin de partie.

POUR JOUER :

On lance le programme par RUN. Un texte de présentation est affiché, puis le message "RESTE DES DIVISIONS : 3, 5, 7" s'affiche.

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

La solution est affichée dès que l'on a donné les 3 restes.

Pour une autre partie, taper O à la question VOULEZ-VOUS UN AUTRE ESSAI (O/N).

Tapez N pour finir.

EXTENSIONS POSSIBLES :

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

```

10 REM DECOMAQUE
20 COLOR 0,13:COLOR 4,15:COLOR 5,2:GRAPH
IC 0,1
30 PRINT"VOUS DEVEZ ENTRER LES TROIS RES
TES"
40 PRINT"XXDANS LES DIVISIONS PAR 3 , 5 E
T 7"
50 PRINT"XXUN ENTIER INFERIEUR A 105"
60 I=3:S=1:GRAPHIC 1,1
70 COLOR 1,1:CHAR 1,0,0,"RESTE DES DIVIS
IONS PAR 3, 5, 7"
80 COLOR 1,I:BOX 1,146+16*I,12,166+16*I,
17,,1:SOUND 1,1000*I+3000,4
90 GET A$: IF A$ = "" OR A$ < "0" OR A$
> CHR$(47+I) THEN 90
100 A(S) = ASC(A$)-48
110 S = S + 1 : I = I + 2
120 IF I < 8 THEN 70
130 COLOR 1,2:CHAR 1,0,9,"RESTE PAR 3
PAR 5 PAR 7 "
140 FOR I=1 TO 3
150 FOR J=1 TO A(I)
160 IF A(I)<>0 THEN COLOR 1,2*I+1:BOX 1,
14+76*I-12*J,100,22+76*I-12*J,108,45,1
170 NEXT J
180 NEXT I
190 T=A(1)*70+A(2)*21+A(3)*15:PLAY"C EG"
200 IF T>105 THEN T=T-105:GOTO 200
210 COLOR 1,9:CHAR 1,0,16,"LA SOLUTION E
ST :"+STR$(T)
220 CHAR 1,0,18,"VOULEZ-VOUS UN AUTRE ES
SAI ( O/N ) ?"
230 GET A$: IF A$ = "" THEN 230
240 IF A$ ="O" THEN RUN:ELSE PLAY"CDEFG"

```

Le jeu du missile

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : contre le C128.

Difficulté du jeu : moyenne.

LE JEU :

Le C128 est en alerte : vous êtes aux commandes d'un avion chargé d'une mission d'espionnage sur son territoire ; votre but est de survoler une base ennemie (représentée par un pavé agrandi), et de revenir sain et sauf sur votre base ; mais, dès que le C128 vous aura repéré, il va envoyer à votre poursuite, un missile très perfectionné qui n'aura de cesse de vous détruire. Votre avion peut se déplacer dans toutes les directions, grâce aux touches de 1 à 8 ; réussirez-vous à remplir votre mission ?

LE PROGRAMME :

La ligne 50 dessine la base ennemie à survoler (sous forme d'un inversé).

La ligne 70 dessine votre base de départ.

Les lignes 90 et 110 choisissent l'instant de départ du missile.

La ligne 140 dessine votre avion.

Les lignes 160 et 170 définissent son mouvement suivant 8 directions, en utilisant les sous-programmes en lignes 1000, 2000, 3000 et 4000.

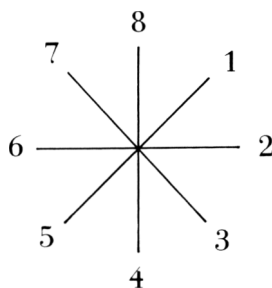
La ligne 180 assure le départ du missile qui est dessiné par la ligne 200 et dont le déplacement est géré par les lignes 210 et 220.

Les lignes 230 et 260 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 RUN/STOP et RESTORE pour arrêter la partie.

EXTENSIONS POSSIBLES :

Il sera sans doute nécessaire d'améliorer les divers contrôles du programme (l'avion a-t-il survolé la base ennemie ? Est-il revenu à sa propre base ?).

On peut aussi prévoir divers niveaux de jeu en faisant, par exemple, varier la vitesse du missile, prévoir l'utilisation de leurres par l'avion, l'explosion du missile au bout d'un certain temps, etc.

```
10 REM LE JEU DU MISSILE
20 COLOR 0,15:COLOR 4,13:COLOR 5,9:GRAPH
   IC 0,1
30 M = 0
40 Z = 5 : GOSUB 500
50 PRINTTAB(17);"3H"
60 Z = 15 : GOSUB 500
70 PRINT"3  "
80 U = 15 : X = 14
90 S = INT(RND(1)*U)
100 Y = 1
110 S = S - 1
120 V = 30
130 Z = X : GOSUB 500
140 COLOR 5,2:PRINTTAB(Y);"↑"
150 GET A$: IF A$ = "" THEN 150
160 F = -2*(VAL(A$)<5)-1:SOUND 2,VAL(A$)
   *1000+4000,3
170 ON VAL(A$)+2*(F-1) GOSUB 1000,2000,3
   000,4000
180 IF S <>0 THEN 110
190 Z = U : GOSUB 500
200 COLOR 5,3:PRINTTAB(V);"←":COLOR 5,9
210 U = U - SGN(U-X)
220 V = V - SGN(V-Y)
230 IF U=X AND V=Y THEN PRINT"00TOUCHE"
   : SOUND 3,40000,240,1,0,100,0:END
240 IF U = 5 AND V = 17 THEN RUN
250 IF X <= 6 AND Y >= 18 THEN M = 1
260 IF M=1 AND X=15 AND Y<=3 THEN PRINT"
   00MISSION ACCOMPLIE":PLAY"CEG":END
270 GOTO 130
500 PRINT"3"
510 FOR P = 0 TO Z
520 PRINT
530 NEXT P
540 RETURN
1000 X = X - F
2000 Y = Y + F
2010 RETURN
3000 GOSUB 2000
4000 X = X + F
4010 RETURN
```

Ardoise magique

REFERENCES :

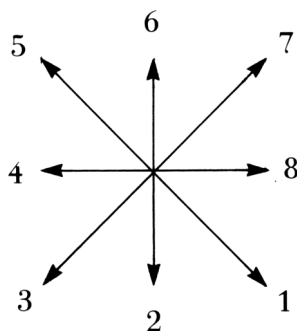
Difficulté de programmation : facile.

Catégorie de jeu : dessin.

Difficulté du jeu : facile.

LE JEU :

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



On dispose aussi de la possibilité de déplacer le point sur l'écran sans laisser de trace en appuyant sur la touche 0 (ce qui permet l'effacement partiel) et de laisser de nouveau une trace en réappuyant sur 0.

On peut choisir la couleur du point en appuyant sur une touche de A à I ; et il est possible d'effacer tout l'écran en appuyant sur P.

LE PROGRAMME :

La ligne 30 contient le choix des possibilités de couleur.

La ligne 60 imprime le point élémentaire.

Les lignes 80 à 105 contrôlent les entrées.

La ligne 140 appelle l'un des quatre sous-programmes effectuant le déplacement.

La ligne 150 vérifie que l'on ne sort pas des limites de l'écran.

La ligne 180 effectue l'effacement s'il y a lieu.

Le sous-programme 300-310 choisit la couleur ; réalise ou non l'effacement, 500-530 propose deux modes de fonctionnement (fin ou pavés).

POUR JOUER :

On lance le programme par RUN ; un point apparaît vers le milieu de l'écran. On devra taper 0 pour le déplacer sans afficher.

Si on frappe une lettre de A à I, la couleur du point change.

La touche P efface tout, et l'on recommence.

On peut sortir à tout moment, en tapant RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

On pourra, bien sûr, envisager d'écrire un programme identique en utilisant des tracés de segments, prévoir un sous-programme qui permette la sauvegarde du dessin réalisé pour une utilisation ultérieure, etc.

```
10 REM ARDOISE MAGIQUE
20 GOSUB 500:COLOR 0,15:COLOR 1,12:COLOR
  4,14:COLOR 5,11:GRAPHIC 1,1
30 C$="0123456789"
40 X=160:Y=96:C=1
50 S = 1
60 IF K=8 THEN BOX 1,X,Y,X+7,Y+7,,1:ELSE
  DRAW 1,X,Y
70 SOUND 1,10000,4
80 GETA$ : IF A$ = "" THEN 80
90 IF A$>="A" AND A$<="I" THEN GOSUB 300
100 IF A$ = "0" THEN GOSUB 400
105 IF A$ = "P" THEN RUN
110 F = -2*(VAL(A$)<5)-1
120 A = X : B = Y
130 U = VAL(A$)+2*(F-1)
140 ON U GOSUB 1000,2000,3000,4000
150 C=-(X>0 AND X<319 AND Y>0 AND Y<199)
160 IF C = 0 THEN GOSUB 5000
170 IF S=1 THEN 190
180 IF K=8 THEN BOX S,A,B,A+7,B+7,,1:ELS
E DRAW S,A,B
190 GOTO 60
300 C = VAL(MID$(C$,ASC(A$)-63,1)):COLOR
  1,C
310 RETURN
400 S = ABS(S-1)
410 RETURN
500 COLOR 0,14:COLOR 4,14:COLOR 5,1:GRAP
HIC 0,1
510 PRINT"MODE 1 (FIN) OU 0 (PAVES)"
520 GETKEY R$:IF R$="1" THEN K=1:ELSEK=8
530 RETURN
1000 Y = Y + K*F
2000 X = X + K*F
2010 RETURN
3000 GOSUB 2000
4000 Y = Y - K*F
4010 RETURN
5000 X = A : Y = B
5010 RETURN
```

Affichage publicitaire

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : affichage publicitaire.

Difficulté du jeu : aucune.

LE JEU :

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

LE PROGRAMME :

Le texte sera introduit lettre par lettre aux lignes 60 à 110.

La ligne 80 permet de corriger les erreurs.

Le défilement vers la gauche est préparé à la ligne 210, et le défilement vers la droite à la ligne 220.

Le texte sera affiché en ligne 190, et la durée de l'affichage (vitesse de défilement) est choisie par la ligne 230.

La ligne 250 permettra de changer, si besoin est, les conditions du défilement.

POUR JOUER :

On lancera le programme par RUN, et à l'affichage de la question ENTREZ VOTRE TEXTE, on introduira le texte à faire défiler, suivi de RETURN.

On peut corriger les erreurs avec la flèche gauche : ←.

On obtient ensuite l'affichage DEFILEMENT A DROITE (D) OU A GAUCHE (G) auquel on répondra en tapant, soit D, soit G.

Puis à la question VITESSE, on répondra en introduisant un grand nombre pour une vitesse rapide, et un petit pour une vitesse lente (par exemple, de 0 à 40 ou plus).

Le texte commencera à défiler dès que l'on aura frappé RETURN.

Si l'on désire changer les conditions du défilement, il suffit d'appuyer sur la touche 0 pour retrouver les questions SENS et VITESSE. On tapera RUN/STOP et RESTORE pour interrompre le programme.

EXTENSIONS POSSIBLES :

On pourra introduire de la couleur, réaliser un affichage ondulant, faire remonter les lignes sur l'écran, etc.

```

10 REM AFFICHAGE PUBLICITAIRE
20 COLOR 0,13:COLOR 4,15:COLOR 5,7:GRAPH
   IC 0,1
30 B$ = "":C$=CHR$(157)+" "+CHR$(157)
40 PRINT"ENTRER VOTRE TEXTE"
50 PRINT
60 GET A$ : IF A$ = "" THEN 60
70 IF ASC(A$)=13 THEN B$=B$+" ":GOTO 120
80 IF ASC(A$)=20 AND B$<>"" THEN PRINTC$
   :B$=LEFT$(B$,LEN(B$)-1):GOTO 60
90 B$ = B$ + A$:SOUND 1,2000,2
100 COLOR 5,9:PRINTA$:
110 GOTO 60
120 COLOR 5,1:PRINT : PRINT"DEFILEMENT A
   DROITE (D) OU A GAUCHE (G)": INPUT S$
130 PRINT:COLOR 5,2
140 INPUT"VITESSE ( DE 0 A 50 )":V
150 V = 51 - V
160 PRINT"J"
170 T = LEN(B$)
180 PRINT"XXXXXXXX"
   "
190 COLOR 5,9:PRINT"XXXXXXXX"; " "+MID$(B$,1
   ,38)+" ":SOUND 1,12000,1
200 PRINT"XXXXXXXX"
   "
210 IF LEFT$(S$,1) = "G" THEN B$ = RIGHT
   $(B$,T-1)+LEFT$(B$,1)
220 IF LEFT$(S$,1) = "D" THEN B$ = RIGHT
   $(B$,1) + LEFT$(B$,T-1)
230 FOR J = 0 TO 3*V
240 GET C$
250 IF C$ ="0" THEN PLAY"C":GOTO 120
260 NEXT J
270 GOTO 180

```

Le singe

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : deux joueurs.

Difficulté du jeu : difficile.

LE JEU :

Il s'inspire de l'idée déjà ancienne du singe qui, frappant au hasard sur une machine à écrire, a des chances non nulles de finir par écrire un texte signifiant.

Le rôle du singe est, ici, tenu par le C128 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 (90-140) permet la confection d'un "mot" en prenant une lettre, soit dans l'alphabet complet, soit dans la liste C\$ (ligne 110), soit dans la liste V\$ (ligne 120).

La ligne 160 temporise l'apparition du résultat.

La ligne 170 gère la fin de partie, avec affichage du score.

La ligne 180 regarde si l'un des joueurs a réagi.

Enfin, les lignes 210 et 220 comptabilisent les points de chaque joueur.

POUR JOUER :

On lance le programme en tapant RUN.

La première série de lettres est alors affichée sur l'écran. Le joueur de gauche devra taper 1 s'il voit qu'il peut réaliser un mot, celui de droite devra taper 0.

Si rien n'a été frappé, une autre série de lettres sera affichée après peu de temps.

Si l'un des joueurs a frappé 0 ou 1, la question (J)USTE ou (F)AUX est alors affichée. On répond J si le mot est correct, F si le mot est incorrect, suivi de RETURN.

Les points sont alors calculés par C128, et la série de lettres suivante est alors affichée.

Pour sortir du programme, il faudra taper P au lieu de 0 ou 1 ; le score final est alors affiché.

EXTENSIONS POSSIBLES :

On pourra modifier la longueur des mots, la fréquence d'apparition des lettres, le décompte des points, la règle du jeu, etc.

```
10 REM LE SINGE
20 A = 0 : B = 0
30 V$ = "AEIOU"
40 C$ = "BCDGLMNPQRSTV"
50 X = INT(RND(1)*4 + 3)
60 M$ = ""
70 COLOR 0,13:COLOR 4,15:COLOR 5,1:GRAPH
IC 0,1
80 PRINT"#####"
90 FOR I = 1 TO X
100 E$ = CHR$(INT(RND(1)*26 + 65))
110 IF RND(1) < 0.4 THEN E$ = MID$(C$,IN
T(LEN(C$)*RND(1) + 1),1)
120 IF RND(1) < 0.5 THEN E$ = MID$(V$,IN
T(LEN(V$)*RND(1) + 1),1)
130 M$ = M$ + E$
140 NEXT I
145 PRINT"#####";
150 FOR I=1 TO LEN(M$):T$=MID$(M$,I,1):S
OUND 1,300*ASC(T$),4:PRINT T$;" ";:NEXT I
160 FOR J = 1 TO 500 : NEXT J:PRINT
165 GET A$
170 IF A$="P" THEN PRINT"#####SCORE DE A : "
;A;"- SCORE DE B : ";B:PLAY"CDEFG":END
180 IF A$ <>"0" AND A$ <>"1" THEN 50
190 COLOR 5,7:PRINT"#####(J)USTE OU (F)AUX
";:GETKEY R$
200 PRINT"#####";
" :SOUND 1,3000,4
210 A = A - (R$="J" AND A$="0" ) + (R$="
F" AND A$="0" )
220 B = B - (R$="J" AND A$="1" ) + (R$="
F" AND A$="1" )
230 GOTO 50
```

Mathieu

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : dessin aléatoire.

Difficulté du jeu : aucune.

LE JEU :

On utilise la définition d'une courbe en coordonnées polaires et la fonction DRAW pour obtenir sur l'écran des dessins aléatoires (de type fleur), plus ou moins élégants.

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

LE PROGRAMME :

Les 3 premiers paramètres Z, K et T, introduits en ligne 30, définissent le rayon maximal et l'occupation de l'écran.

Les variables aléatoires A et B, choisies entre 0 et 3, de la ligne 50, définissent l'allure de la courbe ; il pourra être intéressant de les introduire manuellement (par exemple : 50 INPUT A, B).

La boucle H (60-110) réalise le tracé, point par point. Le rayon R est calculé en ligne 70, et le passage en coordonnées cartésiennes est effectué aux lignes 80 et 90.

L'affichage des points est réalisé en ligne 100.

La ligne 120 imprime un message de fin.

La ligne 150 permet de refaire un nouveau dessin.

POUR JOUER :

On lance le programme en tapant RUN.

Le dessin sera ensuite exécuté automatiquement.

Le message "c'est fini" signale la fin du tracé.

Si on veut un nouveau tracé, on tape 0.

Pour effacer le dessin précédent, on tape une touche quelconque.

Pour arrêter, faire RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

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

```

10 REM MATHIEU
20 COLOR 4,1:COLOR 5,2
30 Z=7:K=9:T=3
40 C=INT(RND(1)*16+1):F=17-C:COLOR 0,C:C
   OLOR 1,F:GRAPHIC 1,1
50 A=3*RND(1):B=3*RND(1)
60 FOR H=0 TO 6.28 STEP 3.14/180
70 R=Z*SIN(H*T)
80 X=150+INT(K*R*COS(A*H))
90 Y=100+INT(K*R*SIN(B*H))
100 DRAW 1,X,Y
110 NEXT H
120 CHAR 1,0,22,"C'EST FINI..."
130 PLAY"CEG"
140 GETKEY A$
150 IF A$="0" THEN 30
160 GOTO 50

```

Adresse

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexe (pour 1 joueur).

Difficulté du jeu : difficile.

LE JEU :

Des chiffres, entre 0 et 9, vont apparaître au hasard, en divers emplacements de l'écran. Ils restent affichés pendant un temps très court. Le joueur doit réussir à appuyer sur la touche numérique correspondante, au moment où le chiffre est affiché.

Le chiffre tapé par le joueur sera alors affiché en grand format sur la gauche de l'écran ; et chaque réponse exacte rapporte 10 points. La durée de la partie est aléatoire et le score sera affiché en fin de partie.

LE PROGRAMME :

Les lignes 40 et 50 contiennent, dans deux variables alphanumériques C\$ et B\$, les références qui seront utilisées pour dessiner les chiffres en grand format.

L'emplacement du chiffre tiré par le C128 est choisi aux lignes 60 et 70, et le chiffre est tiré en ligne 80.

La ligne 100 réalise l'impression, et la ligne 110 détermine la durée d'affichage. La ligne 130 définit la durée du jeu, et la fin de partie est traitée par la ligne 160.

La boucle I (210-250) réalise le dessin grand format du chiffre frappé par le joueur.

Le contrôle de l'exactitude du résultat est effectué par la ligne 260.

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 RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

On peut envisager de modifier le jeu pour faire participer plusieurs joueurs, en modifiant éventuellement l'affichage et en codant les touches du clavier mises à la disposition de chacun des joueurs.

On pourra aussi enlever des points à chaque résultat faux, modifier les règles de calcul du score et la durée du jeu, etc.


```

10 REM ADRESSE
15 COLOR 0,13:COLOR 4,15:COLOR 5,9:GRAPH
IC 0,1
20 T = 100
30 S = 0
40 C$ = "#####  ## #"
50 B$ = "1444133333131211313144133121311
2141133331414114131"
60 X = INT(RND(1)*15)
70 Y = INT(RND(1)*15 + 15)
80 N = INT(RND(1)*10)
90 Z = X : GOSUB 500
100 COLOR 5,2:PRINTTAB(Y);N:A$=""
110 FOR I=0 TO RND(1)*300+150 : NEXT I
120 GET A$ : IF A$ <>"" THEN 200
130 T = T - N
140 Z = X : GOSUB 500
150 PRINTTAB(Y);" "
160 IF T >= 0 THEN 60
170 PRINT"J"
180 COLOR 5,1:PRINT"SCORE "S;" / 100"
190 PLAY"CDEFG":END
200 U = VAL(A$)
210 FOR I = 1 TO 5
220 A = 3*VAL(MID$(B$,5*U+I,1)) - 2
230 Z = I : GOSUB 500
240 COLOR 5,3:PRINTTAB(2);MID$(C$,A,3)
250 NEXT I
260 IF U = N THEN S = S + 10
270 GOTO 120
500 PRINT"@"
510 FOR P = 0 TO Z : PRINT : NEXT P
520 RETURN

```

Sous-marin

REFERENCES :

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

LE JEU :

Vous manœuvrez en surface un chasseur de sous-marins, et des sous-marins passent très régulièrement en dessous de votre unité, à différentes profondeurs.

Vous pouvez, à tout moment, larguer des grenades sous-marines et vous pouvez même envisager de toucher plusieurs fois le même sous-marin.

Votre score sera affiché en permanence, sur la gauche de l'écran (rapport nombre de coups au but sur nombre de sous-marins apparus).

LE PROGRAMME :

La ligne 70 permet au C128 de choisir la profondeur du sous-marin.

La ligne 100 dessine votre bâtiment.

La ligne 130 dessine la surface de l'eau.

La ligne 150 dessine le sous-marin.

Le tir est effectué par un programme spécifique placé aux lignes 240 à 300.

Le déplacement (très rapide) de votre navire est réalisé par les lignes 200 et 210.

Le test, permettant de savoir si l'on a touché ou non, est effectué en ligne 170.

POUR JOUER :

Taper RUN pour commencer la partie.

Le bâtiment sera déplacé grâce aux flèches droite et gauche.

La touche Ø autorise le tir.

Le navire, se déplaçant trois fois plus vite que les sous-marins, il sera possible de toucher plusieurs fois le même bâtiment.

Pour terminer la partie, il conviendra de faire RUN/STOP et RESTORE ou d'attendre 99 sous-marins.

EXTENSIONS POSSIBLES :

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

```

10 REM SOUS-MARIN
20 P = 0 : C = 15
30 COLOR 0,13:COLOR 4,15:COLOR 5,9:GRAPH
IC 0,1
40 Z = S : GOSUB 500
50 PRINT"      "
60 T = 27
70 S = INT(RND(1)*6 + 4)
90 P = P + 0.01
90 X = 1
100 COLOR 5,3:PRINT"8000TAB(C);"
X= "
110 Z = 18 : GOSUB 500
120 COLOR 5,2:PRINT"SCORE : "P
130 COLOR 5,7:PRINT"8000"


---


140 Z = S : GOSUB 500
150 COLOR 5,4:PRINTTAB(T);"       "
160 IF X = 1 THEN Y = C + 2*X
170 IF S=X AND (T+1=Y OR T+2=Y OR T+3=Y)
  THEN BEGIN:P=P+1:Z=S:GOSUB500
175 COLOR 5,3:PRINT TAB(T)"***":SOUND 1,
12000,3:BEND
180 T=T-1:IF P>=99 THEN Z=18:GOSUB 500:P
RINT"C'EST FINI":PLAY"CDEFG":END
190 IF T < 0 THEN 40
200 GET A$
210 C = C - 3*((A$=CHR$(29) AND C < 27)-
(A$=CHR$(157) AND C>0))
230 IF A$<>CHR$(32) AND X=1 THEN 100
240 Z = X : GOSUB 500
250 PRINTTAB(Y);" ":SOUND 1,20000,3,1,0,
1000,1,0
260 X = X + 1
270 IF X >S THEN 90
280 Z = X : GOSUB 500
290 COLOR 5,10:PRINTTAB(Y);"o"
300 GOTO 100
500 PRINT"8"
510 FOR L = 0 TO Z : PRINT : NEXT L
520 RETURN

```

Zombies

REFERENCES :

Difficulté de programmation : assez difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez facile.

LE JEU :

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

La partie se termine, soit avec la disparition du joueur, soit par l'élimination de tous les ZOMBIES.

LE PROGRAMME :

La position des ZOMBIES est définie par un tableau Z non dimensionné (c'est inutile en dessous de 10).

Les lignes 30 à 140 (boucle I) dessinent le cadre, les ZOMBIES et les trous noirs.

La ligne 150 définit la position initiale du joueur.

Une deuxième boucle I (160-240) réalise le déplacement du joueur.

Les lignes 300 à 350 contrôlent le déplacement du joueur.

POUR JOUER :

Après avoir lancé l'exécution par RUN, le joueur devra se déplacer en utilisant les 4 flèches et en veillant à ne pas sortir des limites du terrain.

Il devra manœuvrer de façon à faire venir les ZOMBIES sur les trous noirs, sans se faire lui-même coincer.

Pour arrêter la partie avant la fin, il suffira de taper RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

Il serait intéressant d'agrandir le terrain de jeu, d'augmenter le nombre de ZOMBIES ou de trous noirs, d'améliorer les contrôles, de placer le joueur de façon plus aléatoire, éventuellement de modifier la règle.

On pourra aussi, bien sûr, améliorer la qualité du dessin et introduire la couleur.

```

10 REM ZOMBIES
20 COLOR 0,13:COLOR 4,14:COLOR 5,3:GRAPH
IC 0,1:DIM T(23,21)
30 FOR I = 0 TO 21
40 Z = 0 : GOSUB 500
50 PRINTTAB(I+1);"■"
60 Z = I : GOSUB 500
70 PRINT"■";TAB(23);"■"
80 Z = INT(RND(1)*20 + 1) : GOSUB 500
90 IF RND(1)<0.4 THEN COLOR 5,1:PRINTTAB
(I)"■":COLOR 5,3:T(I,Z)=1
100 S = INT(RND(1)*14 + 1)
110 Z(INT(I/3)+1) = 5 + S
120 Z = 21 : GOSUB 500
130 PRINTTAB(I+1);"■"
140 NEXT I
150 K = INT(RND(1)*18 + 1)
155 P=0
160 FOR I = 1 TO 4
165 P=P+Z(I):IF Z(I)=0 THEN 240
170 Z = Z(I) : GOSUB 500
180 PRINTTAB(Z(I+4));" "
190 Z(I) = Z(I) - SGN(Z(I)-S)
200 Z(I+4) = Z(I+4) - SGN(Z(I+4)-K)
205 IF T(Z(I+4),Z(I))=1 THEN Z(I)=0:SOUN
D 2,20000,10,1,0,400,1,0:GOTO 230
210 Z = Z(I) : GOSUB 500
220 COLOR 5,7:PRINT"0":COLOR 5,4:PRINT
TAB(Z(I+4));"Z"
230 IF Z(I) = S AND Z(I+4) = K THEN COLO
R 5,2:PRINT"PERDU":PLAY"V104U5CCC":END
240 NEXT I
245 IF P=0 THEN BEGIN:COLOR 5,2:PRINT"BR
AVD, VOUS AVEZ DETRUIT TOUS LES ZOMBIES"
247 PLAY"V104U5CDEFG":END:BEND
250 Z = S : GOSUB 500
260 PRINT"■":TAB(K);"●"
270 FOR I = 1 TO 1200 : NEXT I
280 Z = S : GOSUB 500
290 PRINTTAB(K);" "
300 GET A$
310 K=K-2*((A$=CHR$(29))-(A$=CHR$(157)))
320 IF K <= 0 OR K >= 23 THEN PRINT"PERD
U" :PLAY"V104U5GFEDC":END
330 S=S-2*((A$=CHR$(17))-(A$=CHR$(145)))
340 IF S <= 0 OR S >= 21 THEN COLOR 5,1:
PRINT"DOMMAGE" :PLAY"V104U5GFEDC":END
350 GOTO 155
500 PRINT"0"
510 FOR F = 0 TO Z : PRINT : NEXT F
520 RETURN

```

Calculatrice

REFERENCES :

Difficulté de programmation : très difficile.

Catégorie de jeu : simulation d'une calculatrice.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

Les variables B\$ et C\$, des lignes 20 et 30, contiennent les informations codées qui permettront le dessin des chiffres en grand format.

Les lignes 60 à 120 s'occupent de la gestion des entrées.

La boucle I (130-160) réalise la représentation, en grand format, des chiffres sur l'écran, au fur et à mesure des entrées.

Le point décimal et le signe - sont gérés par les lignes 170 et 180.

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 222 à 270 (l'automatisme est géré par la ligne 200, dans le programme principal).

Les sous-programmes placés aux lignes 300-310, 350-360, 400-410 et 450-460 effectuent l'opération indiquée, selon la valeur de Z (ligne 227).

Chaque résultat est mémorisé en A (F-1) (ligne 229).

POUR JOUER :

On devra lancer le programme par RUN.

Puis, on tape les chiffres du premier nombre à entrer qui vont s'afficher un par un, au fur et à mesure (on ne pourra entrer plus de 8 symboles, y compris le signe - et le point décimal). La frappe d'un signe opératoire laisse l'affichage intact, mais l'introduction du prochain chiffre ou symbole provoque l'effacement immédiat et l'affichage du nouveau symbole. Il est à remarquer que les chiffres s'introduisent de gauche à droite, sens de l'écriture usuelle.

L'appareil accepte, comme signes opératoires les symboles +, -, *, / ; le signe = provoquera l'affichage du résultat. Les opérations sont enchaînées automatiquement, le résultat précédent étant pris en compte comme opérateur du calcul suivant.

Pour faire une autre opération, il faudra taper RUN/STOP et RESTORE, puis RUN.

Taper RUN/STOP et RESTORE pour sortir du programme.

EXTENSIONS POSSIBLES :

On devra, tout d'abord, améliorer l'affichage du signe - (représenté, ici, par un trait), puis on pourra améliorer le dessin des chiffres (en utilisant, par exemple, le générateur du C128), on peut aussi envisager une extension à plus de 8 chiffres, autoriser des opérations plus complexes (COS, SIN, ...), ajouter des mémoires, etc.

```

10 REM CALCULATRICE
15 COLOR 0,13:COLOR 4,11:COLOR 5,3:GRAPH
IC 0,1:DIM A(30)
20 C$ = "#####  ##  ##"
30 B$ = "1444133333131211313144133121311
2141133331414114131"
40 F = 1 : L = 0
50 N = 0 : T = 0 : K = 0 : R$=""
60 GET A$:IF A$="" THEN 60:ELSE PLAY"C"
65 IF K = 1 THEN N = 0 : R$ = "" : K = 0
70 IF A$ = "=" THEN 220
75 X = (1 AND A$ = "+")+(2 AND A$ = "-")+(3
AND A$ = "*")+(4 AND A$ = "/")
80 R$ = R$ + A$
90 IF X <> 0 THEN 220
95 IF R$ = "-" THEN A$ = R$ : GOTO 170
100 IF A$ < "." OR A$ > "9" OR A$ = "/" THEN 60
110 IF N = 0 THEN PRINT "J"
120 IF A$ = "." OR A$ = "-" THEN 170
125 PRINT " "
130 FOR I = 1 TO 5
140 S = 3*VAL(MID$(B$,I+5*VAL(A$),1))-2
150 PRINTTAB(N);MID$(C$,S,3)
160 NEXT I
170 IF A$ = "-" THEN PRINT "##### "
180 IF A$ = "." THEN PRINT "##### "TAB(N+1
);"% "
190 N = N - 4*(N<28)
200 IF T <> 0 THEN NEXT T
205 IF K = 1 AND L <> 1 THEN 50
207 T = 0
210 GOTO 60
220 IF R$ = "-" AND A(F) = 0 THEN 95
222 N = 0 : K = 1
223 IF(A$="*" OR A$="/")AND VAL(R$)=0 T
HEN A(F) = 1 : GOTO 225
224 A(F) = VAL(R$)
225 IF X <> 0 THEN Z = X
227 ON Z GOSUB 300,350,400,450
228 IF A$ <> "=" THEN F = F + 1
229 A(F-1) = VAL(R$)
230 IF VAL(R$) = 0 THEN A(F-1) = A(F)
233 R$ = STR$(VAL(R$))
235 IF LEFT$(R$,1)=" " THEN R$=RIGHT$(R$
,LEN(R$)-1)
250 FOR T = 1 TO LEN(R$)
260 A$ = MID$(R$,T,1)
270 GOTO 110
300 R$ = STR$(A(F-1) + A(F)) : A(F) = VA
L(R$) : L = 1
310 RETURN
350 IF F <> 1 THEN R$ = STR$(A(F-1) - A(F
)) : A(F) = VAL(R$) : L = 1
360 RETURN
400 IF F <> 1 THEN R$ = STR$(A(F-1) * A(
F)) : A(F) = VAL(R$) : L = 1
410 RETURN
450 IF F <> 1 THEN R$ = STR$(A(F-1) / A(F
)) : A(F) = VAL(R$) : L = 1
460 RETURN

```

Clavier

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de réflexe (apprentissage du clavier).

Difficulté du jeu : difficile.

LE JEU :

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

LE PROGRAMME :

Une variable A\$ dimensionnée en ligne 50, contiendra la liste des touches concernées. Les valeurs des touches sont entrées de la ligne 60 à la ligne 110 par appel d'un sous-programme situé en 330.

Les boucles emboîtées J (120-170) et I (130-160) dessinent le clavier.

La ligne 180 définit la position du point imprimé par la ligne 200.

La touche frappée est testée aux lignes 230-240 et le point est effacé par la ligne 260.

Les affichages de fin sont réalisés aux lignes 280 à 310.

POUR JOUER :

On lance le programme par RUN. L'on dispose alors de quelques secondes pour frapper la touche qui est au-dessus de l'étoile, puis une autre étoile est affichée, etc.

La durée de la partie n'est pas limitée ; il suffit de taper 0 pour obtenir l'affichage du score, et du temps total.

Faire RUN pour une autre partie.

EXTENSIONS POSSIBLES :

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


```

10 REM CLAVIER
20 COLOR 0,13:COLOR 4,15:COLOR 5,2:GRAPH
IC 0,1
30 PRINT:INPUT"(A)ZERTY OU (Q)WERTY ";R$
40 S = 0 : TI$="000000" : P = 0
50 DIM A$(10,3):GRAPHIC 0,1
60 J=1:IF R$="Q" THEN S$="QWERTYUIOP":EL
SE S$="AZERTYUIOP"
70 GOSUB 330
80 J=2:IF R$="Q" THEN S$="ASDFGHJKL":EL
SE S$="QSDFGHJKLM"
90 GOSUB 330
100 J=3:IF R$="Q" THEN S$="ZXCVBNM,./":EL
SE S$="WXCVBN,;:=
110 GOSUB 330
120 FOR J = 1 TO 3
130 FOR I = 1 TO 10
140 Z = 5*J : GOSUB 350
150 PRINTTAB(3*I+J-4);A$(I,J)
160 NEXT I
170 NEXT J
180 X=INT(RND(1)*10+1):Y=INT(RND(1)*3+1)
190 Z = 2+5*Y : GOSUB 350
200 COLOR 5,3:PRINTTAB(3*X+Y-4);"●":SOUN
D 1,10000,2
210 FOR F = 1 TO 50+70*RND(1) : GET A$
220 IF A$ = "" THEN NEXT F
230 IF A$ = A$(X,Y) THEN S = S + 1
240 IF A$ = "0" THEN 280
250 P = P + 1 : Z=2+5*Y:GOSUB 350
260 PRINTTAB(3*X+Y-4);" "
270 GOTO 180
280 Z = 19 : GOSUB 350
290 PRINTS;" TROUVES JUSTES SUR";P;" ESS
AIS ."
300 PRINT"TEMPS TOTAL ";LEFT$(TI$,2);"
HEURES , ";MID$(TI$,3,2);" MN. , ";
310 PRINTRIGHT$(TI$,2);" SEC."
320 PLAY"CDEFG":END
330 FOR I = 1 TO 10 : A$(I,J)=MID$(S$,I,
1):NEXT I
340 RETURN
350 PRINT"0";
360 FOR P = 0 TO Z : PRINT : NEXT P
370 RETURN

```

Chasse au sous-marin

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de réflexion.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

La boucle I (40-120) affiche le terrain.

La ligne 140 définit la position initiale du sous-marin ; la ligne 150 permet son déplacement, contrôlé par la ligne 160 (on reste dans les limites du terrain).

La ligne 180 affiche votre position.

Les lignes 230 à 250 contrôlent votre déplacement.

La distance est calculée en ligne 260, affichée en ligne 280 et testée en lignes 240 et 300.

Le test de fin de partie est effectué à la ligne 420.

POUR JOUER :

On lance le programme par RUN et on entre la dimension ; la grille de jeu s'affiche. Vous êtes au départ en position (0, 0) et vous vous déplacez avec les flèches.

Après chaque déplacement, la nouvelle distance du sous-marin s'affiche ; à chaque coup le sous-marin se déplace. Si vous réussissez à vous placer à sa verticale, la distance 0 est affichée et il ne bouge plus.

Vous tirez en donnant une profondeur entière (entre 2 et 4, bornes comprises). Si vous le ratez, il se replace n'importe où, et on recommence ; si vous le coulez, vous obtenez l'affichage BRAVO et la partie s'arrête.

On peut arrêter à tout moment le programme en utilisant les touches RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

Il est possible de prévoir une amélioration du graphisme, d'améliorer les possibilités de déplacement du chasseur (un peu lent, dans cette version), de prévoir des parcours plus sophistiqués pour le sous-marin, de mettre un compteur de temps...

```

10 REM CHASSE AU SOUS-MARIN
20 COLOR 0,13:COLOR 4,15:COLOR 5,2:GRAPH
   IC 0,1
30 INPUT"DIMENSION DE LA GRILLE ";H
35 PRINT"J";H=H-1
40 FOR I = 0 TO H
50 Z = 0 : GOSUB 500
60 COLOR 5,3:PRINTTAB(2*I+3);I
70 Z = 2*I + 1 : GOSUB 500
80 PRINTI;
90 FOR J = 0 TO H
100 COLOR 5,2:PRINTTAB(2*J + 4);".";
110 NEXT J
120 NEXT I
130 I = 1 : X = 0 : Y = 0
140 V=INT(RND(1)*H):U=INT(RND(1)*H)
150 U = U + I : V = V + I:COLOR 5,2
160 IF (U<0 OR U>H OR V<0 OR V>H) THEN I
   = -I
170 Z = 2*X + 1 : GOSUB 500
175 IF Y <>0 THEN PRINTTAB(2*Y+2);".";
180 COLOR 5,4:PRINTTAB(2*Y+4);"■";COL
   OR 5,2
185 IF Y <>H THEN PRINTTAB(2*Y+6);"."
190 Z = 2*X -1 : GOSUB 500
200 IF X <>0 THEN PRINTTAB(2*Y+4);"."
210 Z = 2*X + 3 : GOSUB 500
220 IF X <>H THEN PRINTTAB(2*Y+4);"."
230 GET A$: IF A$ = "" THEN 230
240 Y=Y-(A$=CHR$(29) AND Y<H) + (A$=CHR$
   (157) AND Y>0)
250 X = X - (A$=CHR$(17) AND X<H) + (A$
   =CHR$(145) AND X>0)
260 D=INT(SQR(ABS(X-V)^2 + ABS(Y-U)^2))
270 Z = 10 : GOSUB 500
280 COLOR 5,2:PRINTTAB(29)"DIST. ";STR$(
   D)+"";PLAY"C"
290 IF D = 0 THEN 400
300 IF D < 3 THEN 150
310 GOTO 170
400 Z = 12 : GOSUB 500:PLAY"IFA"
410 WINDOW 29,13,39,18,1:INPUT"PROF.";D
420 IF D = INT(RND(1)*3 + 2) THEN PRINT"
   BRAVO" :PLAY"CDEFG":WINDOW 0,0,39,24:END
430 WINDOW 29,13,39,18,1:WINDOW 0,0,39,2
   4:GOTO 140
500 PRINT"0";
510 FOR P = 0 TO Z : PRINT : NEXT P
520 RETURN

```

Le lézard d'Alice

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez difficile.

LE JEU :

Il s'inspire très vaguement d'une idée de LEWIS CARROLL, dans *Alice au Pays des Merveilles*. Dans l'histoire originale, Alice grandissait ou diminuait, alors que dans le jeu, pour des raisons d'encombrement de la mémoire, c'est un lézard qui change de taille.

Deux verres apparaissent sur l'écran : l'un contient le liquide qui fait grandir, l'autre le liquide qui fait rapetisser (malheureusement ce n'est pas toujours le même verre). Le joueur devra indiquer la quantité de liquide qu'il désire boire et le lézard changera de taille en fonction de cette quantité. Le but du jeu est de parvenir à lui rendre sa taille initiale.

LE PROGRAMME :

La boucle I (60-90) dessine les verres.

La boucle I (200-230) les remplit avec la quantité indiquée en ligne 170.

La variation de taille du lézard est définie par la ligne 270, avec contrôle par les lignes 280 et 290.

La boucle I (300-320) dessine le corps du lézard, les lignes 330 à 380 dessinent l'œil et les pattes.

Un message gagnant sera affiché par la ligne 400 si l'on a trouvé la bonne taille.

POUR JOUER :

On lance le programme par RUN : les deux verres se dessinent, puis apparaît la question QUEL COTE à laquelle on répond en tapant G ou D ; on obtient l'affichage QUELLE QUANTITE ; on entre alors une quantité (de 0 à 130) et on frappe RETURN.

Le verre désigné se remplit sous vos yeux, puis, après quelques instants, le lézard est affiché. On devra alors frapper une touche quelconque ; si le lézard n'avait pas la bonne taille, aucun message n'est affiché et l'on revient au dessin des verres.

EXTENSIONS POSSIBLES :

Puisqu'il s'agit d'un jeu essentiellement graphique, il sera intéressant d'améliorer les dessins et la couleur, de modifier le caractère aléatoire du choix des verres, etc.

Pourquoi ne pas relire *Alice au Pays des Merveilles*, les idées n'y manquent pas.

```

10 REM LE LEZARD D'ALICE
20 COLOR 4,15:GRAPHIC 1,1
30 PLAY"CDEFGFEDC"
40 B$=" ":COLOR 0,13:COLOR 1,1:SCNCLR
50 N=18
60 FOR I=1 TO 9
70 CHAR 1,3,I+4,B$,1:CHAR 1,10,I+4,B$,1
80 CHAR 1,13,I+4,B$,1:CHAR 1,20,I+4,B$,1
90 NEXT I
100 FOR I=3 TO 10:CHAR 1,I,14,B$,1:NEXT
110 FOR I=13 TO 20:CHAR 1,I,14,B$,1:NEXT
120 COLOR 1,2:CHAR 1,4,15,"GAUCHE":CHAR
1,14,15,"DROIT"
130 COLOR 1,9:CHAR 1,1,17,"QUEL COTE :"
140 GETA$:IF A$<>"D"AND A$<>"G" THEN 140
150 U=2-20*(A$="D")
160 COLOR 1,5:IF A$="D" THEN CHAR 1,13,1
7,"DROIT":ELSE CHAR 1,12,17,"GAUCHE"
170 A$="":COLOR 1,9:CHAR 1,1,19,"QUELLE
QUANTITE":COLOR 1,5
180 GETKEY R$:IF ASC(R$)<>13 THEN A$=A$+
R$:CHAR 1,17,19,A$:GOTO 180
190 COLOR 1,7:Q=VAL(A$)
200 FOR I=1 TO 4*SQR(Q)
210 DRAW 1,4*U+24,112-I TO 4*U+70,112-I
220 SOUND 1,400*I+500,2
230 NEXT I
240 GETKEY A$
250 COLOR 0,1:COLOR 1,3:SCNCLR 1
260 H=-2*(RND(1)<.5)-1:WIDTH 1
270 N=N+H*INT(SQR(Q))
280 IF N<12 THEN N=12
290 IF N>60 THEN N=60
300 FOR I=0 TO 4*(N+3)
310 DRAW 1,I,100:DRAW 1,I,99:DRAW 1,I,98
:DRAW 1,I,97
320 NEXT I
330 T=4*(N-INT((SQR(N)/2))):WIDTH 2
340 DRAW 1,T,88:DRAW 1,T-1,89:DRAW 1,T-2
,90:DRAW 1,T+1,89:DRAW 1,T+2,90
350 T=4*SQR(N):DRAW 1,T,102:DRAW 1,T-1,1
03:DRAW 1,T-2,104
360 T=T+2:DRAW 1,T,102:DRAW 1,T+1,103:DR
AW 1,T+2,104
370 T=4*(N-SQR(N)):DRAW 1,T,102:DRAW 1,T
-1,103:DRAW 1,T-2,104
380 T=T+2:DRAW 1,T,102:DRAW 1,T+1,103:DR
AW 1,T+2,104
390 GETKEY A$
400 COLOR 1,2:IF N=28 THEN CHAR 1,1,20,"
BRAVO":PLAY"CDEFG":END
410 GOTO 40

```

Dessin de dés

REFERENCES :

Difficulté de programmation : aucune.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : aucune.

LE JEU :

Le programme proposé ne réalise pas un jeu en lui-même, mais pourra être utilisé en sous-programme pour de nombreux jeux.

Il offre simplement la possibilité de lancer de 1 à 8 dés, par simple pression d'une touche. Les dés sont ensuite affichés avec un graphisme très simplifié.

LE PROGRAMME :

La ligne 130 saisit le caractère frappé au clavier et le transforme en nombre (si ce n'est pas possible, le programme refuse l'entrée).

Une variable D non dimensionnée reçoit la valeur indiquée par chaque dé (ligne 200).

Une boucle J (170-190) est utilisée pour dessiner un fond, et une autre boucle I (160-220) sert au tirage (tirage en ligne 200 et impression en ligne 210).

La ligne 230 offre la possibilité de refaire un tirage ; elle n'est pas indispensable.

POUR JOUER :

On lance le programme par l'ordre RUN et on obtient l'affichage NOMBRE DE DES A LANCER auquel on répond en frappant un chiffre.

Le résultat est alors automatiquement affiché ; il suffit de presser une touche quelconque pour recommencer le programme.

Pour arrêter, il est nécessaire de taper RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

On pourra, bien sûr, utiliser le canevas fourni par ce programme pour la réalisation de tout jeu se servant de dés (421, YAM'S, CRAPS, etc.).

```

10 REM DESSIN DE DES
20 DATA 0,0,0,24,24,0,0,0,96,96,0,0,0,0,
6,6,96,96,0,24,24,0,6,6,102,102,0,0,0,0
30 DATA 102,102,102,102,0,24,24,0,102,10
2,102,102,0,102,102,0,102,102
40 FOR I=1 TO 21:B$=B$+CHR$(0):NEXT I
50 FOR I=1 TO 6
60 A$(I)=B$
70 FOR J=0 TO 7
80 READ A:A$(I)=A$(I)+CHR$(0)+CHR$(A)+CH
R$(0)
90 NEXT J
100 A$(I)=A$(I)+B$
110 NEXT I
115 COLOR 0,15:COLOR 4,9:COLOR 5,12:GRAP
HIC 1,1
120 COLOR 1,9:CHAR 1,0,0,"NOMBRE DE DES
A LANCER : "
130 GET A$:IF A$="" THEN 130
140 IF A$<"1" OR A$>"8" THEN 110
150 N=VAL(A$):COLOR 1,13:GRAPHIC 1,1
160 FOR I=1 TO 8
170 FOR J=1 TO 4
180 IF I<=N THEN CHAR 1,4*I-4+16*(I>4),8
+J-5*(I>4)," ",1
190 NEXT J
200 D(I)=INT(RND(1)*6+1):SPRSAY A$(D(I))
,I
210 SPRITE I,-(I<=N),I,0,1,1:MOVSPR I,32
*I-20+128*(I>4),116-40*(I>4)
220 NEXT I
230 PLAY"CEG":GOTO 120

```

Evasion

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : solitaire.

Difficulté du jeu : facile.

LE JEU :

Votre seul problème : réussir à traverser l'écran, sans que le C128 vous ait auparavant touché par l'un des nombreux obus explosifs qu'il va tirer sur vous.

Vous pouvez vous déplacer dans 4 directions (à l'aide des 4 flèches), et vous devez réussir à sortir du terrain par le bord droit, sans, bien sûr, avoir été touché une seule fois.

LE PROGRAMME :

La ligne 50 définit la position de départ du joueur.

La ligne 60 choisit la coordonnée horizontale de la case où C128 va tirer (liée à votre position), et la ligne 70 choisit la coordonnée verticale (arbitraire) de cette case.

Le déplacement du joueur est contrôlé par les lignes 140 et 150. Sa position actuelle (sur l'écran, un SPRITE) est imprimée par la ligne 160.

La ligne 170 regarde s'il est touché ou non.

La ligne 190 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 symbole graphique.

Les 4 touches de direction (←, ↓, ↑, →) servent à déplacer le joueur.

S'il est touché, on obtient, sur l'écran, le message PERDU et la partie s'arrête.

S'il réussit à sortir par la droite de l'écran sans être touché, on obtient le message BRAVO, et la partie est également terminée.

On peut terminer à tout moment en tapant RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

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


```

10 REM EVASION
20 COLOR 0,13:COLOR 1,9:COLOR 4,15:COLOR
  5,7:GRAPHIC 1,1
30 DATA 0,0,0,0,0,0,0,8,20,8,62,42,8,28,
  54,0,0,0,0,0,0,0
40 FOR I=0 TO 21:READ A:A$=A$+CHR$(0)+CH
  R$(A)+CHR$(0):NEXT:SPRSAY A$,1
50 X=2:Y=17
60 H=8*(X-1)+16*RND(1)
70 K=180*RND(1)+5
80 IF X<5 THEN 120
90 FOR J=1 TO 7*RND(1)+3
100 DRAW 1,H+6*RND(1)-2,K+6*RND(1)-2
110 NEXT J
120 SOUND 1,12000,1
130 GET A$
140 Y = Y + (A$=CHR$(145) AND Y>6) - (A$
  =CHR$(17) AND Y<29)
150 X = X - (A$=CHR$(29) AND X<39) + (A$
  =CHR$(157) AND X>2)
160 SPRITE 1,1,7:MOVSPR 1,8*X,8*Y
170 IF X>3 AND (ABS(H-8*X)+ABS(K-8*(25-Y
  ))<=10) THEN BEGIN
180 COLOR 1,2:CHAR 1,0,22,"PERDU":PLAY"G
  CC":END:BEND
190 IF X<38 THEN 60
200 COLOR 1,5:CHAR 1,0,22,"BRAVO":PLAY"C
  DEFG"

```

Chapeaux

REFERENCES :

Difficulté de programmation : assez facile.

Catégorie de jeu : jeu de réflexion, pour deux joueurs.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

Les lignes 20 et 30 contiennent le dessin des deux chapeaux (dans deux variables alphanumériques B\$ et C\$).

Les lignes 40, 50, 60 contiennent, en DATA, les symboles qui permettront de dessiner les personnages.

La ligne 170 permet, s'il y a lieu, l'échange des chapeaux.

La variable T de la ligne 80 sert à repérer s'il y a eu échange ou non.

Le dessin des deux personnages est réalisé par les lignes 120 et 130, et leur déplacement est réalisé par les lignes 140 et 150.

Le test, permettant ou non le changement des couvre-chefs, est effectué en ligne 170.

Les lignes 200 à 230 déterminent si le premier joueur ayant appuyé sur une touche a gagné ou perdu.

POUR JOUER :

Lancer le programme par RUN.

Les personnages commencent immédiatement à se déplacer l'un vers l'autre.

Le joueur de gauche dispose des touches 1 et 2 et celui de droite, des touches 9 et 0.

S'il y a eu changement de chapeau, les touches à appuyer seront 1 (à gauche) et 0 (à droite).

S'il n'y a pas eu de changement, il faudra, pour gagner, être le premier à appuyer sur 2 (à gauche) et sur 9 (à droite).

L'affichage indiquera alors BRAVO GAUCHE ou BRAVO DROITE, ou, encore, PERDU GAUCHE ou PERDU DROITE.

EXTENSIONS POSSIBLES :

Il sera possible de prévoir un affichage couleur, avec une animation plus complexe des personnages.

On pourra aussi compliquer le jeu à plaisir avec plusieurs objets échangés de façon aléatoire (il faudrait alors en trouver la liste complète).

```

10 REM CHAPEAUX
20 B$ = "  ▲  "
30 C$ = "  ▼  "
40 DATA 0,56,0,0,68,0,0,90,0,0,74,0,0,67
,0,0,36,0,0,24,0,0,24,0,0,126,0,0
50 DATA 189,0,0,189,0,0,189,0,0,189,0
60 DATA 0,189,0,0,60,0,0,60,0,0,36,0,0,3
6,0,0,36,0,0,102,0,0,231,0
70 FOR I=1 TO 21:READ A,B,C:A$=A$+CHR$(A
)+CHR$(B)+CHR$(C):NEXT I
80 SPRSAV A$,1:SPRSAV A$,2:X=2:Y=28:T=1
90 COLOR 0,15:COLOR 1,9:COLOR 4,13:COLOR
5,2:GRAPHIC 2,1,15
100 CHAR 1,X,1,C$
110 CHAR 1,Y,1,B$
120 SPRITE 1,1,3,0,1,1,0:MOVSPR 1,8*X+16
,70
130 SPRITE 2,1,7,0,1,1,0:MOVSPR 2,8*Y+24
,70
140 X = X+1
150 Y = Y-1
160 FOR I=1 TO 300:NEXT:SOUND 1,10000,4
170 IF X=Y AND RND(1) < .5 THEN D$=B$:B$
=C$:C$=D$:T=ABS(T-1)
180 GET R$ : IF R$ ="" THEN 100
190 IF X <= Y THEN 220
200 IF (R$="1"AND T=1)OR(R$="2"AND T=0)T
HEN PRINT"  BRAVO GAUCHE":PLAY"CEG":END
210 IF (R$="0"AND T=1)OR(R$="9"AND T=0)T
HEN PRINT"  BRAVO DROITE":PLAY"DFA":END
220 IF R$="2" OR R$="1" THEN PRINT"  PER
DU GAUCHE":PLAY"CCC"
230 IF R$="9" OR R$="0" THENPRINT"  PERD
U DROITE":PLAY"GGG"

```

Combat aérien

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexe (un joueur).

Difficulté du jeu : difficile.

LE JEU :

Au centre de l'écran, se trouve le viseur de votre appareil, et l'avion ennemi se déplace devant vous, d'une façon qui pourra vous sembler au début incohérente.

Vous pouvez vous déplacer dans toutes les directions, grâce aux flèches de mouvement, et vous allez essayer d'amener l'appareil ennemi au centre de votre viseur.

Vous tirez sur lui en frappant Ø, et vous voyez vos projectiles qui convergent sur lui. Mais il continue à se déplacer, même pendant votre tir, et il n'est pas du tout évident que vous réussirez à l'avoir.

La partie continuera à se dérouler jusqu'à ce que vous l'ayez touché (affichage de GAGNE).

LE PROGRAMME :

La ligne 40 dessine l'appareil ennemi dont le déplacement aléatoire sera réalisé par les lignes 60 et 70.

Les lignes 80 à 100 vous permettent le pilotage de votre appareil.

La ligne 140 dessine le viseur de votre appareil.

La ligne 150 regarde si vous avez tiré ou non.

Le sous-programme, gérant le tir, va de la ligne 200 à la ligne 280 et le test de fin de partie est effectué en ligne 250.

POUR JOUER :

Dès que le programme a été lancé par l'ordre RUN, vous devez manœuvrer pour essayer de faire en sorte que l'appareil ennemi soit cadré dans votre viseur.

Pour cela, vous disposez des flèches de déplacement (touche ↑, ↓, →, ←) et vous pouvez tirer à tout moment en pressant la touche Ø.

L'appareil ennemi peut continuer à se déplacer de manière aléatoire, même lorsque vous avez tiré.

La partie se terminera avec l'affichage de GAGNE, si vous touchez l'avion ennemi.

On peut aussi arrêter à tout moment en frappant RUN/STOP et RE-STORE.

EXTENSIONS POSSIBLES :

Il sera assez facile de faire un affichage au graphisme plus soigné, utilisant la couleur. On pourrait envisager de stabiliser la position de l'avion ennemi pendant le tir.

On peut aussi introduire plusieurs avions, compter les points ou le temps, tenir compte de sa propre réserve de carburant, etc.

```

10 REM COMBAT AERIEN
20 A$="":B$="":GOSUB 500
25 A = 5 : B = 15
27 N = 27
30 COLOR 0,15:COLOR 1,7:COLOR 4,13:COLOR
  5,9:GRAPHIC 1,1
40 SPRITE 1,1,3,1,1,1:MOVSPR 1,8*B,8*A
60 A = A + 2*INT(RND(1)*2) - 1
70 B = B + 2*INT(RND(1)*2) - 1
80 GET R$
90 A=A-2*(R$=CHR$(17))+2*(R$=CHR$(145))
100 B=B-2*(R$=CHR$(29))+2*(R$=CHR$(157))
140 CHAR 1,17,8,"● ●":CHAR 1,17,9," × ":
CHAR 1,17,10,"● ●"
150 IF R$ = "0" OR N <27 THEN GOSUB 200
160 GOTO 40
200 SPRITE 2,1,1,1,1:SPRITE 3,1,1,1,1:MOVSPR
  2,8*(34-N),8*N:MOVSPR 3,8*(N+6),8*N
210 N=N-1:SOUND 1,5000,1,1,0,100,1,0
220 IF N>14 THEN RETURN
250 IF (A=9 OR A=10 OR A=11) AND (B=17 O
  R B=18 OR B=19) THEN BEGIN
260 MOVSPR 1,8*B,8*A+8:PLAY"V2T5U6CDEFG"
270 CHAR 1,0,20,"GAGNE...",1:END:BEND
280 GOTO 27
500 REM DEFINITION DE LA CIBLE
510 FOR I=0 TO 21:READ A:A$=A$+CHR$(0)+C
  HR$(A)+CHR$(0):NEXT
520 FOR I=0 TO 21:READ A:B$=B$+CHR$(0)+C
  HR$(A)+CHR$(0):NEXT
530 SPRSAV A$,1:SPRSAV B$,2:SPRSAV B$,3
540 RETURN
550 DATA 0,0,0,0,0,0,0,0,24,24,60,255,255,
  60,66,0,0,0,0,0,0,0,0
560 DATA 0,0,0,0,0,0,0,0,8,8,8,28,62,28,
  0,0,0,0,0,0,0,0

```

L'indésirable

REFERENCES :

Difficulté de programmation : difficile.

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

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

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

La boucle J (60-110) dessine le terrain de jeu.

La position initiale du locataire est tirée au hasard en ligne 120.

Le numéro de la case à visiter est donné en ligne 130.

Les lignes 140 à 220 contrôlent les diverses réponses possibles.

La boucle I (185-200) dessine les cases visitées.

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

Les lignes 270 et 280 assurent son déplacement.

POUR JOUER :

On lance l'exécution par RUN.

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

EXTENSIONS POSSIBLES :

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

```

10 REM L'INDESIRABLE
15 DATA 0,0,0,0,0,0,0,24,60,126,24,36,36
,24,24,60,127,189,189,188,188,188
20 FOR I=0 TO 21:READ A:A#=A#+CHR$(0)+CH
R$(A)+CHR$(0):NEXT I:SPRSAY A#,1
25 COLOR 0,13:COLOR 4,15:COLOR 1,13:GRAP
HIC 1,1
30 BOX 1,1,10,30,25:DRAW 1,1,10 TO 10,1
TO 20,1 TO 30,10
40 GSHAPE B#,0,0,37,37:SPRITE 1,0,1
50 COLOR 1,2:GRAPHIC 2,1,21:DIM B(16)
60 FOR J=1 TO 4
70 FOR I=1 TO 4
80 COLOR 1,2:GSHAPE B#,64*I-28,40*J-30,0
90 COLOR 1,7:CHAR 1,8*I-2,5*J-2,CHR$(60+
I+4*J)
100 NEXT I
110 NEXT J
120 X=INT(RND(1)*16+1):S=1:K=0
130 COLOR 5,4:INPUT"CASE VISITEE " :V$:C
=ASC(V$)-64
140 IF B(C)=1 THEN PRINT"DEJA VISITEE":P
LAY"CCC":GOSUB 300:GOTO 130
150 IF C=X THEN PRINT"TROUVE EN ";S;" CO
UPS. BRAVO":PLAY"CDEFG":K=1:GOTO 230
160 IF X-5=CORX-4=CORX-3=CORX-1=CORX+1=C
ORX+3=CORX+4=5ORX+5=C THEN 230
170 PRINT"IL N'EST PAS LA":PLAY"CEG"
180 S=S+1:B(C)=1
185 FOR I=1 TO 4
190 FOR J=1 TO 4
195 IF B(I+4*J-4)=1 THEN COLOR 1,3:PAINT
1,64*I-20,40*J-25,1
200 NEXT J,I
210 IF S=9 THEN PRINT"PERDU. IL ETAIT EN
";CHR$(X+64):K=1:GOTO 230
220 FOR I=1 TO 150:NEXT:GOTO 130
230 SPRITE 1,1,7:MOVSPR 1,64*(X-1-4*INT(
(X-1)/4))+65,10*(4*INT((X-1)/4))+40
240 FOR I=1 TO 700:NEXT
250 IF K=1 THEN PLAY"GEC":END
260 SPRITE 1,0,5
270 Z=INT(10*RND(1)+1)-5:IF ABS(Z)=2 THE
N 270
280 X=X+Z:IF X>16 OR X<1 THEN X=X-Z:GOTO
270
290 GOTO 180
300 I=C-4*INT(C/4):J=INT(C/4)+1:COLOR 1,
8:PAINT 1,64*I-20,40*J-10,1:RETURN

```

Course de chevaux

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : hasard (plusieurs joueurs).

Difficulté du jeu : aucune.

LE JEU :

On entrera un nombre de chevaux compris entre 1 et 9, et la course débutera immédiatement.

Il pourra y avoir des ex aequo, car l'on tiendra compte du fait que le C128 s'occupe de faire avancer les chevaux les uns après les autres, en commençant par le numéro 1 ; le classement sera affiché au fur et à mesure des arrivées (et en tenant compte des ex aequo).

LE PROGRAMME :

Une première boucle non formelle (70-157) affiche les chevaux dans leur position de départ.

Une deuxième boucle I (100-160) les fera avancer ; la position de chaque cheval sera repérée par un chiffre ou une lettre dans une variable alphanumérique A\$.

La ligne 120 fera avancer les chevaux.

La ligne 130 gère les arrivées.

La ligne 140 réalisera les impressions.

Le sous-programme 200-240 fait avancer le compteur des arrivées, et imprime les résultats.

POUR JOUER :

On lance le programme par RUN.

A la question NOMBRE DE CHEVAUX : on répond par un entier de 1 à 8, suivi de RETURN.

La course se déroulera ensuite automatiquement.

EXTENSIONS POSSIBLES :

On pourra dessiner les chevaux avec un graphisme plus soigné, et les différencier par des numéros.

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


```

10 REM COURSE DE CHEVAUX
20 COLOR 0,13:COLOR 4,15:COLOR 5,3:COLOR
  1,2:GRAPHIC 0,1
30 INPUT"NOMBRE DE CHEVAUX";N
40 IF N>8 THEN 20
50 GOSUB 300:GRAPHIC 1,1
60 C$="":P=0:A=0:T=1:K=0
70 J=T
80 C$ = C$ + "0":I=J
90 GOTO 140
100 FOR I = 1 TO N
110 IF MID$(C$,I,1) = "V" THEN 150
120 B$ = CHR$(ASC(MID$(C$,I,1))+INT(RND(
  1)*4 + 1))
125 MID$(C$,I,1)=B$
130 IF B$ >= "V" THEN GOSUB 200
140 SPRITE I,1,I,0,0,0,0:MOVSPR I,8*(ASC
  (MID$(C$,I,1))-46),24*I+30
150 SOUND 1,400*I+900,4:SLEEP 1
155 IF K=0 THEN J=J+1
157 IF K=0 AND J<=N THEN 80:ELSE IF J>N
  AND K=0 THEN K=1:GOTO 100
160 NEXT I
170 P = P + T
180 T = 0
190 IF A < N THEN 100
195 PLAY"CEG":END
200 MID$(C$,I,1)="V"
210 CHAR 1,0,3*I-2,CHR$(48+P),1
220 T = 1:PLAY"CCC"
230 A = A + T
240 RETURN
300 REM LECTURE D'UN SPRITE EN DATA
310 A$=""
320 FOR I=0 TO 20
330 READ A,B,C
340 A$=A$+CHR$(A)+CHR$(B)+CHR$(C)
350 NEXT I
360 FOR I=1 TO N:SPRSAY A$,I:NEXT I
370 RETURN
420 DATA 0,0,128,0,1,128,0,3,128,0,7,64,
  15,143,240,63,216,240,255,240,96,63,240
430 DATA 0,63,224,0,48,96,0,40,144,0,72,
  136,0,72,132,0
440 DATA 136,130,0,144,132,0,144,132,0,1
  44,132,0,0,0,0,0,0,0,0,0,0,0,0,0

```

Création de SPRITE

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : utilitaire.

Difficulté du jeu : difficile.

LE JEU :

Le Commodore 128 possède d'origine une très puissante instruction, DEFSPRITE, qui permet de créer soi-même des SPRITES multicolores.

Malheureusement, les SPRITES ainsi créés ne peuvent apparaître dans les listings que sous forme de chaîne de caractères, difficile à recopier.

Le programme permet d'une part la création d'un SPRITE, puis sa sauvegarde dans des lignes de DATA, que l'on pourra facilement incorporer dans ses propres programmes.

LE PROGRAMME :

Lignes 60 à 170 : lecture ; 190 à 370 création.

Les DATA constituent le dessin du SPRITE, et une fois convenablement remplies, sont lues par la ligne 80, et enregistrées dans une variable A\$ (ligne 90). Puis on crée le SPRITE (ligne 110) et on fait ensuite une petite animation (lignes 130 et 140).

Lors de la création, les données, introduites sous forme d'étoiles aux lignes 380 à 590, sont lues en ligne 190, et décodées par une double boucle J (200-260) et K (220-240). La ligne 250 permet un premier affichage (symbolique). Les lignes 280 et 290 permettront d'afficher des lignes de DATA directement utilisables sur l'écran.

POUR JOUER :

On commence par faire une liste du programme à l'écran, grâce à l'ordre LIST. S'il y a lieu, il conviendra d'effacer les lignes 600 et 610, qui pourraient rester d'une précédente manipulation, et de refaire un LIST.

L'écran affiche alors les lignes de DATA de 380 à 590, contenant des points. On se déplace sur l'écran à l'aide des flèches du curseur et on marque des étoiles aux emplacements désirés, jusqu'à obtenir le dessin du SPRITE.

On devra alors enregistrer les lignes DATA, en renvoyant le curseur en haut de l'écran, et en passant d'une ligne à la suivante en tapant RETURN.

On pourra alors passer en "création". On lance le programme par RUN, et on tape C. Les éléments du dessin sont traduits par des chiffres, l'ensemble étant récapitulé au moyen de deux lignes DATA, 600 et 610, qui apparaissent en bas de l'écran. On les enregistre, en venant dessus avec le curseur, et en tapant RETURN (on peut aussi les garder sur l'écran pour charger un autre programme).

On passe alors en "lecture". On lance le programme par RUN et on tape L. Un exemple d'utilisation des SPRITES est mis en route.

EXTENSIONS POSSIBLES :

SPRITES multicolores, extension du même programme à 8 SPRITES.

```

10 REM LECTURE D'UN SPRITE EN DATA
20 COLOR 0,15:COLOR 4,13:COLOR 5,4:GRAPH
   IC 0,1
30 PRINT"(C)REATION OU (L)ECTURE :";GET
   KEY C$
40 A$=""
50 RESTORE380:IF C$="L" THEN RESTORE600
60 FOR I=0 TO 21
70 IF C$="C" THEN 180
80 READ A,B,C
90 A$=A$+CHR$(A)+CHR$(B)+CHR$(C)
100 NEXT I
110 SPRSAV A$,1
120 COLOR 0,13:COLOR 4,13:COLOR 1,3:GRAP
   HIC 1,1
130 SPRITE 1,1,7,0,0,0,0
140 MOVSPR 1,240,70:MOVSPR 1,180 #6
150 SOUND1,45000,200,2,300,3000,2:SLEEP7
160 PLAY"CDEFG":CHAR 1,0,0,"GARDEZ LES L
   IGNEES 600 ET 610 SUR L'ECRAN"
170 CHAR 1,0,3,"PUIS FAITES NEW OU BIEN
   CHARGEZ VOTRE PROGRAMME":END
180 REM LECTURE ET TRAD DATA
190 PRINT:READ B$:PRINT"DATA";
200 FOR J=0 TO 2
210 A=0
220 FOR K=0 TO 7
230 IF MID$(B$,8*J+K+1,1)="*" THEN A=A+2
   ↑(7-K)
240 NEXT K
250 PRINTTAB(10*J+5);PRINT USING "###";
   A;
260 NEXT J
270 NEXT I
280 RESTORE 380:PRINT:PRINT"600 DATA";G
   =0:F=10:GOSUB 310
290 PRINT:PRINT"610 DATA";G=11:F=21:GOS
   UB 310
300 END
310 FOR I=G TO F:READB$:FOR J=0 TO 2:A=0
320 FOR K=0 TO 7
330 IF MID$(B$,8*J+K+1,1)="*" THEN A=A+2
   ↑(7-K)
340 NEXT K
350 PRINT MID$(STR$(A),2);IF I+J=F+2 TH
   EN 370 : ELSE :PRINT",";
360 NEXT J,I
370 RETURN
380 DATA .....
390 DATA .....

```

400	DATA	
410	DATA	
420	DATA	
430	DATA	
440	DATA	
450	DATA	
460	DATA	
470	DATA	
480	DATA	
490	DATA	
500	DATA	
510	DATA	
520	DATA	
530	DATA	
540	DATA	
550	DATA	
560	DATA	
570	DATA	
580	DATA	
590	DATA	

Niveau 5

Nom du programme

Thèmes et particularités

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

Inscription directe dans les mémoires : déplacement d'objets de bas en haut sur l'écran : SCROLL

LES INSTRUCTIONS D'EXPLORATION DES CASES MÉMOIRE : PEEK ET POKE

La fonction PEEK permet la lecture directe du contenu de l'une des cases mémoire adressables par le microprocesseur.

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

On pourra par exemple, colorier la bordure de l'écran en tapant POKE 53280, A (avec A choisi de 0 à 15, ce qui donne 16 couleurs différentes. De même POKE 53281, A permet de colorier le fond. La mémoire d'écran commençant à l'adresse 1024 et finissant en 2023, on pourra lire le caractère présent sur l'écran en ligne X et en colonne Y en faisant : PEEK (1024 + Y + 40 * X), qui donne un nombre de 0 à 255, représentant le code écran (et non ASCII) du caractère présent ligne X, colonne Y.

De même, les couleurs de chaque caractère (ou graphique) présent sur l'écran sont définies dans une mémoire de couleur allant de l'adresse 55296 à 56295. On fera POKE 55296 + Y + 40 * X, 5 pour colorier en vert (couleur 5) le caractère présent en ligne X, colonne Y.

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 excessif).

Le C128 réalise automatiquement un "**Scrolling**" (c'est-à-dire un déplacement de l'ensemble de l'image) vers le haut, lorsque l'on écrit sur la dernière ligne en bas de l'écran.

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

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

Dames

REFERENCES :

Difficulté de programmation : très difficile.

Catégorie de jeu : deux joueurs.

Difficulté du jeu : difficile.

LE JEU :

C'est le jeu classique de dames à la française, mais un peu incomplet puisqu'il ne prévoit pas la possibilité de faire des dames. La règle du jeu reste classique, mais les contrôles ont été un peu simplifiés. Les mouvements des pièces sont entrés en une seule manœuvre (position de départ et direction du déplacement) et les prises sont réalisées automatiquement, si elles sont possibles.

Par contre le changement de joueur étant réalisé à chaque coup, il faudra éventuellement taper 0 pour pouvoir rejouer s'il y avait eu saut ou si le coup n'était pas possible.

LE PROGRAMME :

Les boucles I (80-160) et J (90-140) dessinent le damier et définissent les positions initiales des pions noirs et des pions blancs. Les positions sont dans une table R (X, Y) et la couleur dans une autre table C (X, Y).

L'impression des pions proprement dite est effectuée par les boucles emboîtées I (170-230) et J (180-220).

Les lignes 330 et 340 indiquent le joueur qui a le trait.

Le coup est entré en ligne 350.

La ligne 355 réalise le changement de joueur (on tape 000).

La ligne 370 regarde si le déplacement est possible.

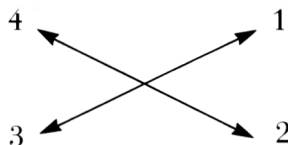
La ligne 400 traite le cas d'un déplacement et la ligne 410 celui d'un saut.

La ligne 450 assure le changement de joueur.

POUR JOUER :

Dès le lancement du programme par RUN, le damier est affiché avec ses pions, et un symbole de couleur apparaît à côté pour indiquer que le trait appartient au joueur correspondant.

On joue en indiquant par un mot de 3 lettres (pouvant commencer par un 0) la ligne et la colonne de la pièce à déplacer, puis la direction du déplacement (codée de 1 à 4) suivant la convention :



Par exemple 9D1 signifie que le O situé en ligne 9, colonne D, se déplace dans la direction 1.

De même 0G3 signifiera pièce en ligne 0, colonne G, 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. Le programme ne contrôle pas l'arrivée à dame, et n'interdit pas les déplacements en arrière.

EXTENSIONS POSSIBLES :

Il sera assez facile de commencer par améliorer les contrôles (changement de joueur, coup en arrière qui ne correspondrait pas à une prise, sortie de terrain). On pourra envisager la création de dames, leur déplacement, et éventuellement écrire un programme permettant au C128 de se mêler avec de bonnes chances de réussite à vos passionnantes parties.

```

10 REM DAMES
15 DIM R(10,10) : DIM C(10,10)
20 COLOR 0,12:COLOR 4,12:GRAPHIC 0,1
30 A$ = "  "
40 B$ = "  "
50 C$ = "  "
60 PRINT"      A  B  C  D  E  F  G  H  I
J"
80 FOR I = 1 TO 10
90 FOR J = 1 TO 10
100 IF (I+J)/2 = INT((I+J)/2) THEN 140
110 PRINTTAB(3*J);A$;PRINTI-1;TAB(3*J);B
$:PRINTTAB(3*J);C$;
130 PRINT"TTT"
140 NEXT J
150 PRINT""
160 NEXT I
170 FOR I = 1 TO 10
180 FOR J = 1 TO 10
185 R(I,J)=32
190 IF (I+J)/2 <> INT((I+J)/2) THEN 220
200 IF J<5 THEN R(I,J)=81:C(I,J)=0
210 IF J>6 THEN R(I,J)=81:C(I,J)=1
220 NEXT J
230 NEXT I
255 H = 1
260 FOR J = 1 TO 10
270 FOR I = 1 TO 10
275 IF (I+J)/2 <> INT((I+J)/2) THEN 310
280 X = 1 + 3*I : Y = 2*I
290 POKE 1024 + X + 40*Y , R(I,J)
300 POKE 55296 + X + 40*Y , C(I,J)
310 NEXT I
320 NEXT J
330 PRINT"TTTTT"TAB(36);""
340 IF H=0 THEN PRINT"TTT"TAB(36);""
345 PRINTTAB(33);""

```

```

350 SOUND 1,3000*H+4000,4:INPUT"?????????
????????????????????";D$
355 IF D$ ="0" OR D$="" THEN 450
360 GOSUB 500
370 IF R(L,C)=32 OR (C(L,C)=1 AND H = 1)
OR (C(L,C)=0 AND H=0) THEN 330
380 T = (1 AND (D=1 OR D=2)) - (1 AND (D
=4 OR D=3))
390 S = (1 AND (D=3 OR D=2)) - (1 AND (D
=1 OR D=4))
400 IFR(L+S,C+T)=32 THEN R(L,C)=32:R(L+S,C
+T)=81:C(L+S,C+T)=-(H=0):GOTO 450
410 IFC(L+S,C+T)=-(H=1) AND R(L+S,C+T)=81 A
NDR(L+2*S,C+2*T)=32 THEN 430
420 GOTO 330
430 R(L,C)=32:R(L+S,C+T)=32:R(L+2*S,C+2*
T)=81:C(L+2*S,C+2*T)=-(H=0)
440 GOTO 260
450 H = ABS(H-1)
460 GOTO 260
500 L = VAL(LEFT$(D$,1))+1 : C = ASC(MID
$(D$,2,1))-64 : D = VAL(RIGHT$(D$,1))
510 RETURN

```

Tic tac toe

REFERENCES :

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

LE JEU :

Un grand classique des jeux d'écoliers, proposé ici dans une version ne comportant pas le jeu machine ; chaque joueur devra placer à son tour un O ou un X 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 de la boucle I (40-70).

Les lignes 100 et 110 s'occupent d'afficher le symbole X ou O correspondant au joueur qui a le trait.

Un symbole clignotant (lignes 140 et 150) est affiché en haut et à gauche du terrain, même si la case est déjà occupée ; les lignes 170 et 180 permettent de déplacer le symbole sur le terrain de jeu.

La ligne 190 vérifie si la case où l'on veut jouer est libre, et si l'on désire jouer.

La ligne 200 enregistre le coup, et la ligne 210 réalise le changement de joueur.

POUR JOUER :

Le terrain de jeu sera affiché dès la frappe de RUN.

Un symbole clignotant apparaît alors en haut et à gauche de l'écran ; le joueur peut le déplacer sur le terrain de jeu à l'aide des 4 flèches (touches ←, → et ↑, ↓).

Lorsqu'il arrive sur la case où il désire jouer, il suffit qu'il appuie sur la touche Ø, 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.

La partie se termine automatiquement lorsque les neuf symboles O ou X ont été placés. On pourra ensuite faire une autre partie avec RUN.

EXTENSIONS POSSIBLES :

On pourrait réaliser après chaque coup un contrôle pour vérifier si l'un des joueurs a gagné. On peut aussi réaliser un jeu machine mais il faudrait éviter qu'il soit par trop systématique (par exemple, la machine jouant toujours son premier coup sur la case centrale).

```

10 REM TIC TAC TOE
20 COLOR 0,15:COLOR 4,13:COLOR 5,9:GOSUB
  500
40 FOR I = 1 TO 7
50 IF I/2<>INT(I/2)THENC$=""
  ":CHAR 1,8,2*I-1,C$,1:CHAR 1,8,2*I,C$,1
60 IF I/2=INT(I/2) THEN BEGIN:D$=""  ":CHA
R1,8,2*I-1,D$,1:CHAR1,12,2*I-1,D$,1
62 CHAR1,16,2*I-1,D$,1:CHAR1,20,2*I-1,D$
,1:CHAR1,8,2*I,D$,1:CHAR1,12,2*I,D$,1
65 CHAR1,16,2*I,D$,1:CHAR1,20,2*I,D$,1:B
END
70 NEXT I
90 K = 0 : S = 1
100 COLOR 1,K+1:CHAR 1,0,18,"JOUEUR X"
110 IF K=1 THEN CHAR 1,0,18,"JOUEUR O"
120 I=0:J=0:SOUND 1,4000*K+3000,4
130 C = PEEK(1113 + 2*I + 80*J):D=3+(C=6
5)+2*(C=83)
140 COLOR 1,D:POKE 1113+2*I+80*J,102:GSH
APE A$(3),32*I+80,32*J+24,1
150 POKE 1113+2*I+80*J,C:COLOR 1,D:GSHAP
E A$(D),32*I+80,32*J+24
160 GET A$
170 J=J-(A$=CHR$(17) AND J<2)+(A$=CHR$(1
45) AND J>0)
180 I=I-(A$=CHR$(29) AND I<2)-(A$=CHR$(
157) AND I>0)
190 IF A$<>"0" OR C=65 OR C=83 THEN 130
200 POKE 1113+2*I+80*J,83-K*18:COLOR 1,K
+1:GSHAPE A$(K+1),32*I+80,32*J+24
210 K = ABS(K-1) : S = S + 1
220 IF S <10 THEN 100
230 CHAR 1,0,21,"C'EST FINI"
240 PLAY"CDEFG":END
500 REM DESSIN DES FORMES
510 COLOR 1,9:GRAPHIC 1,1:WIDTH 2
520 SSHAPE A$(3),0,0,15,15
530 CIRCLE 1,45,45,7:SSHAPE A$(2),38,38,
53,53
540 DRAW 1,80,80TO95,95:DRAW1,80,95TO95,
80:SSHAPE A$(1),80,80,95,95
550 GRAPHIC 1,1:RETURN

```

Smokey

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

Une colonne de fumée bleue apparaît sur l'écran. Le joueur, représenté par un rond rouge, 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 70 fait apparaître la fumée au bas de l'écran ; la ligne 90 permet de changer (vers la droite ou vers la gauche) la direction de la colonne, et la ligne 110 fait monter la colonne.

Le joueur apparaît grâce à la ligne 130, et son déplacement est effectué par les lignes 180 et 200.

Les lignes 270 et 290 se chargent de l'affichage du résultat.

POUR JOUER :

On doit lancer l'exécution du programme par RUN.

La fumée commence à s'élever, tandis que le joueur, apparu sous forme d'un rond rouge dans un coin de l'écran, va se déplacer sur elle à l'aide des 4 flèches. Dès qu'il arrive sur la fumée, il va automatiquement commencer à l'avalier, et devra essayer de suivre la colonne sans dévier (grâce aux flèches). Il gagne s'il arrive en bas de l'écran avant que la fumée n'arrive en haut.

L'affichage final indique alors BRAVO ou PERDU, ainsi que le score obtenu.

EXTENSIONS POSSIBLES :

On pourra, tout en conservant l'idée de départ, imposer d'autres règles : par exemple avaler toute la fumée, qui pourra aussi repartir d'un autre endroit si l'on dévie de la colonne initiale.

On pourra aussi améliorer les contrôles de fin de partie, et la qualité du graphisme.

```

10 REM SMOKEY
40 S = 0 : X = 24 : Y = INT(RND(1)*24)
50 U = 0 : V = 15
60 PRINT"J"
70 POKE 1024 + Y + 40*X , 83
80 POKE 55296 + Y + 40*X , 3
90 Y = Y - (RND(1)<0.5 AND Y < 24) + (RND(1)<0.5 AND Y > 0)
100 T = 1
110 X = X - T
120 IF X = 0 THEN 260
130 POKE 1024 + V + 40*U , 81
140 POKE 55296 + V + 40*U , 10
150 IF T <> 0 THEN FOR I = 1 TO 400 : GET
  A$ : IF A$ <> "" THEN 170
160 IF T <> 0 THEN NEXT I
170 POKE 1024 + V + 40*U , 32
180 U = U - (A$=CHR$(145) AND T AND X <
24) + (A$=CHR$(17) AND T AND X > 0)
190 U = U - ( T = 0)
200 V = V - (A$=CHR$(29) AND V < 24) + (
  A$=CHR$(157) AND V > 0)
210 C = PEEK(1024 + V + 40*U)
220 IF C <> 83 THEN 70
230 T=0:S=S+1:SOUND 1,2000,3
240 IF U <> 24 THEN 130
260 PRINT"J"
270 IF S>8 THEN PRINT"BRAVO" : GOTO 290
280 PRINT"PERDU"
290 PRINT"SCORE ";S:PLAY"CDEFG"

```

Course de voitures

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de réflexe (1 joueur).

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

L'impression de la route est assurée par la boucle I (60-80).

La ligne 120 fait avancer la route, et la ligne 130 imprime la voiture.

La ligne 160 contrôle son mouvement, et la ligne 170 détermine si la route fait ou non des virages.

La ligne 190 regarde si l'on a touché le bord de la route.

La ligne 200 incrémente le score.

La ligne 260 affiche le temps avant l'accident.

POUR JOUER :

Lancer l'exécution du programme par l'ordre RUN, le dessin de la route commence à évoluer, et votre véhicule, représenté par une voiture, apparaît.

Pour le déplacer vers la droite ou vers la gauche, vous devrez utiliser les touches → et ← (flèches de déplacement du curseur). Il vous faudra éviter le plus longtemps possible de toucher une bordure noire, ce qui provoquera l'arrêt du programme et l'affichage de la mention ACCIDENT, suivie de l'indication sur le score réalisé.

On pourra arrêter le déroulement de la partie à tout moment en pressant les touches RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

On peut envisager de jouer à plusieurs, chacun son tour, en enregistrant le meilleur score.

On peut aussi envisager d'élargir la piste, et de faire courir plusieurs voitures.

On pourra aussi s'inspirer des jeux d'arcades qui sont particulièrement nombreux sur ce thème.


```

10 REM COURSE DE VOITURES
15 TI$ ="000000"
20 COLOR 0,14:COLOR 4,10:COLOR 5,6:GRAPH
IC 0,1
25 POKE2604,(PEEK(2604) AND NOT 14) OR 8
30 DATA93,127,85,20,93,127,93,8,255,255,
255,255,255,255,255,255,0,0,0,0,0,0,0
35 FOR I=0TO23:READ A:POKE8192+I,A:NEXT
40 X = 23 : Y = 11
45 A = 5 : B = 10
50 S = 0:POKE 2595,0:POKE 2592,1
60 FOR I = 0 TO 6
65 POKE 1024 + Y + 40*X +I , 2
70 IF I <2 OR I>4 THEN POKE 1024 + Y + 4
0*X + I , 1
80 NEXT I
90 POKE 55296 + Y + 40*X , 5:POKE 55297
+ Y + 40*X , 5 :POKE55301 + Y + 40*X,5
100 POKE 55302 + Y + 40*X , 5
120 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXXXXBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"
130 POKE 1024+B+40*A,0
140 POKE 55296+B+40*A,2
145 FOR I = 1 TO 15 : GET A$ : IF A$ <>"
"THEN 155
150 NEXT I
155 C = B:SOUND 1,12000,2
160 B=B-(A$=CHR$(29))+(A$=CHR$(157))
170 Y = Y - (2*RND(1) AND Y > 3) +(2*RND
(1) AND Y < 25)
190 IF PEEK(1024+B+40*(A+1))=1 THEN 230
200 S = S + 1
210 POKE 1024 + C + 40*A , 2
220 GOTO 60
230 SOUND 1,15000,200,2,2000,1000,1
240 POKE2604,(PEEK(2604)ANDNOT 14)OR 4
250 COLOR 5,9:PRINT"ACCIDENT; SCORE :";S
260 PRINT"TEMPS ";MID$(TI$,3,2);" MINUTE
S ET ";RIGHT$(TI$,2);" SECONDES"

```

Danger spatial

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de réflexe, pour un joueur.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

La position de chaque météorite est déterminée par les lignes 80 et 100 ; elle est imprimée par les lignes 90 et 110.

L'avance des météorites est réalisée par la ligne 130.

Les lignes 140 et 150 vérifient si vous avez ou non heurté une météorite.

Les lignes 160 et 170 dessinent votre vaisseau spatial, et la ligne 210 permet de le déplacer.

Le compteur se trouve en ligne 220.

POUR JOUER :

Le jeu est lancé par l'ordre RUN.

Votre vaisseau apparaît vers le milieu de l'écran, et les météorites viennent à votre rencontre en partant du bas de l'écran. Pour vous déplacer, vous pouvez utiliser les touches de déplacement horizontal (→ et ←).

On peut arrêter la partie à tout moment en tapant RUN/STOP et RESTORE. Sinon, le jeu s'arrête lorsque vous heurtez une météorite, et le score est alors affiché.

EXTENSIONS POSSIBLES :

Il serait intéressant de rendre le jeu de plus en plus difficile en faisant apparaître de plus en plus de météorites au fur et à mesure du déroulement de la partie.

On peut aussi essayer de détruire les météorites en les pulvérisant au fur et à mesure de leur apparition.

```

10 REM DANGER SPATIAL
20 COLOR 0,2:COLOR 4,13:COLOR 5,7:GRAPHI
C 0,1
30 POKE2604,(PEEK(2604)AND NOT 14)OR 8
40 DATA 0,3,7,14,124,62,6,8,0,192,224,24
0,62,252,96,16,0,73,28,54,119
50 DATA 54,28,73,0,0,0,0,0,0,0
60 FOR I=0TO31:READ A:POKE8192+I,A:NEXT
70 X=18:Z=0:FORI=1024TO1983:POKEI,3:NEXT
80 H=INT(RND(1)*40)
90 IF RND(1)<.7 THEN POKE 1944+H,2:POKE
56216+H,3
100 K=INT(RND(1)*40)
110 IF RND(1) < 0.6 THEN POKE 1944 +K ,
2 : POKE 56216 +K , 3
130 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXXXXCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC"
140 B=PEEK(1464+X):C=PEEK(1464+X+1)
150 IF B = 2 OR C = 2 THEN 240
160 POKE 1424+Y,3:POKE 1424+Y+1,3
170 POKE 1464+X,0:POKE 1464+X+1,1
180 POKE 55736+X,8:POKE 55736+X+1,8
190 GET A$
200 Y = X
210 X = X - (A$=CHR$(29) AND X < 38) + (
A$=CHR$(157) AND X > 0)
220 Z = Z + 1:SOUND 1,12000,1
230 GOTO 80
240 SOUND 1,24000,240,2,300,300,1
250 POKE2604,(PEEK(2604)AND NOT 14)OR 4
260 GRAPHIC 0,1:COLOR 5,9:PLAY"CDEFG"
270 PRINT"LE SCORE EST DE : "Z

```

Anagrammes

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de réflexion (2 joueurs).

Difficulté du jeu : selon le niveau.

LE JEU :

Des lettres sont choisies au hasard par le C128, le nombre de lettres étant fonction du niveau de difficulté désiré. Puis les lettres du mot ainsi obtenu sont brouillées, et affichées au bas de l'écran ; un nouveau mot est alors confectionné, de façon aléatoire, à partir des mêmes lettres et il est affiché sous le premier, et ainsi de suite ; les deux joueurs (gauche et droit) disposent chacun d'une touche (1 ou 0) et doivent la presser dès qu'ils voient s'afficher au bas de l'écran un mot déjà présent sur l'écran ; leur score est alors évalué, et un message correspondant à leur résultat sera délivré.

LE PROGRAMME :

Le niveau du jeu est entré à la ligne 30. La boucle I (60-80) réalise le choix des lettres.

Une deuxième boucle I (100-170) effectue le brouillage, et la ligne 120 permet de détecter la pression d'une touche.

La ligne 180 affiche un nouveau mot, et fait monter d'une ligne sur l'écran les mots déjà obtenus.

La ligne 240 cherche quel est celui des deux joueurs qui a pressé une touche.

La boucle I (250-320) explore les mots présents sur l'écran ligne par ligne, alors que la boucle J (270-300) saisit un par un les caractères de ces mots. On obtient ainsi un nouveau mot qui est comparé au mot initial en ligne 310 et le résultat BRAVO apparaît si c'est exact.

Sinon, la partie s'arrête sur l'affichage PERDU (ligne 330).

POUR JOUER :

On lance le programme par RUN.

On devra ensuite entrer le niveau de difficulté (en principe de 1 à 3) et faire RETURN. Les mots s'affichent en partant du bas de l'écran, et montent au fur et à mesure ; si l'un des joueurs croit reconnaître dans le mot qui apparaît un des mots déjà affichés, il devra immédiatement presser sa touche (1 ou 0), et son résultat sera affiché après un court instant.

La partie sera alors terminée.

EXTENSIONS POSSIBLES :

On pourra utiliser d'autres affichages que des mots, modifier les temps d'affichage, faire participer plus de deux joueurs, améliorer le son et la couleur, etc.

```

10 REM ANAGRAMMES
20 COLOR 0,1:COLOR 4,12:COLOR 5,2:GRAPHI
C 0,1
30 INPUT"NIVEAU ( DE 1 A 3 ) :";N
40 IF N<1 OR N>3 THEN PLAY"CCC":RUN
50 S$ = ""
60 FOR I = 1 TO N+3
70 S$ = S$ + CHR$(INT(RND(1)*26+65))
80 NEXT I
90 A$ = "" : B$ = S$
100 FOR I = 1 TO LEN(S$)
110 N = INT(RND(1)*LEN(S$)+1)
120 GET D$
130 IF D$="1" OR D$="0" THEN 230
140 IF MID$(B$,N,1)="-" THEN 110
150 A$ = A$ + MID$(B$,N,1)
160 MID$(B$,N,1)="-"
170 NEXT I
180 X=INT(RND(1)*9+2):COLOR 5,X:PRINT"
XXXXXXXXXXXXXXXXXXXX";A$
190 SOUND 1,1000*X+3000,4
200 FOR I = 1 TO 20*LEN(B$):NEXT I
210 S$ = A$
220 GOTO 90
230 H$ = "JOUER DE DROITE"
240 IF D$="1" THEN H$="JOUER DE GAUCHE"
245 PRINT"
":COLOR 5,2
250 FOR I = 0 TO 24
260 B$ = "" :PRINT
270 FOR J = 0 TO LEN(S$)-1
280 R$ = CHR$(PEEK(1024+40*I+J)+64)
290 B$ = B$ + R$
300 NEXT J
310 IF B$=S$ THEN PRINTTAB(LEN(S$)+4);"
TBRAVO , ";H$ : PRINT"
":PLAY"CDEFG":END
320 NEXT I
330 PRINTTAB(LEN(S$)+4);"
TPERDU , ";H$:
PLAY"CCC"

```

Œufs

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

L'oiseau est dessiné en deux parties (pour le mouvement) aux lignes 140 et 210 ; son déplacement aléatoire est géré par le sous-programme 310-400 (en particulier les lignes 360 à 390).

La ligne 230 permet de reboucler le programme sur la ligne 140.

Le déplacement du joueur est régi par les lignes 250 et 260, et le test permettant de détecter la présence d'un œuf est effectué en ligne 270.

La ligne 290 permet l'affichage du score.

La fin de jeu est détectée en ligne 330.

POUR JOUER :

On lancera le programme par RUN.

L'oiseau commence alors à se déplacer au hasard sur l'écran, et se met à pondre ses œufs. Lorsque le symbole qui représente le joueur (un carré de couleur) apparaît sur l'écran, celui-ci peut se déplacer en utilisant les 4 touches de déplacement. Il vous faut dès lors essayer de récupérer un maximum d'œufs avant la fin de la partie. Certains œufs ne pourront être récupérés facilement.

On pourra arrêter de jouer à tout moment en tapant RUN/STOP et RE-STORE.

EXTENSIONS POSSIBLES :

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

```

10 REM OEUF5
20 COLOR 0,1:COLOR 4,15:COLOR 5,9:COLOR
1,7:GRAPHIC 2,1,20
30 DATA 0,0,0,24,126,153,36,36,66,36,36,
24,24,24,36,36,0,73,28,54,119,54,28,73
50 FOR J=1 TO 3
60 FOR I=0 TO 21
70 IF I<8 OR I>15 THEN A$(J)=A$(J)+CHR$(
0)+CHR$(0)+CHR$(0):GOTO 90
80 READ A:A$(J)=A$(J)+CHR$(0)+CHR$(A)+CH
R$(0)
90 NEXT I
100 SPRSAV A$(J),J
110 NEXT J
120 P=200:R=18:S=10:T=0:POKE 2592,1
130 X=INT(RND(1)*36):Y=INT(RND(1)*21)
140 SPRITE 2,0,6:SPRITE 1,1,6:MOVSPR 1,8
*X+16,8*Y+40
170 FOR I = 1 TO 30 : GET A$
180 IF A$="" THEN NEXT I
190 IF A$ <>"" THEN GOSUB 240
200 GOSUB 310
210 SPRITE 1,0,6:SPRITE 2,1,6:MOVSPR 2,8
*X+16,8*Y+40
220 GOSUB 310
230 GOTO 140
240 SPRITE 3,1,3
250 R = R - (A$=CHR$(29) AND R < 36) + (
A$=CHR$(157) AND R >0)
260 S = S - (A$=CHR$(17) AND S < 21) + (
A$=CHR$(145) AND S >0)
270 MOVSPR 3,8*R+16,8*S+40:G = BUMP(2)
280 T=T-(G=7):CHAR 0,R,S," "
285 SOUND 1,8000,2:IF G=7 THEN SOUND 1,8
000,20,2,1000,400,2
290 PRINT"#####"TAB(15
);"SCORE :";T
300 RETURN
310 A = X : B = Y
320 P = P - 1
330 IF P=0 THEN PRINT"C'EST FINI":PLAY"C
DEFG":END
350 CHAR 1,A,B,"0"
360 X = ABS(X+INT(1+2*(RND(1)<0.5)))
370 Y = ABS(Y+INT(1+2*(RND(1)<0.5)))
380 IF X > 36 THEN X = 2
390 IF Y > 20 THEN Y = 5
400 RETURN

```

Taquin de lettres

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

Qui d'entre nous n'a pas, un jour ou l'autre, eu dans les mains un de ces petits puzzles où l'on essaie de déplacer d'une case à une autre des petits carrés de plastique colorés, jusqu'à l'obtention d'un résultat bien précis ? En son temps le 14-15 de SAM LOYD, a eu autant de succès que notre célèbre RUBIK'S.

Le taquin qui vous est proposé ici utilise 35 lettres, disposées au hasard dans un carré de 6 cases sur 6 cases, et ces lettres peuvent, replacées dans leur ordre initial, vous permettre de reconstituer un proverbe très connu. Une case colorée vous permet de déplacer les lettres, et un compteur affiche en permanence le nombre d'essais : arriverez-vous à retrouver le proverbe ?

LE PROGRAMME :

La phrase à reconstituer (qui pourra bien sûr être modifiée) est placée dans une variable R\$ en ligne 70.

Le mélange des lettres et l'impression sont faits par la boucle I (80-150).

La ligne 180 réalise l'affichage du carré coloré ; celui-ci est déplacé par les lignes 220 et 230 en utilisant les 4 flèches.

Les lignes 240 et 250 réimpriment la lettre concernée à l'emplacement du carré coloré.

La ligne 260 comptabilise le nombre d'essais qui est imprimé grâce à la ligne 170.

POUR JOUER :

On lance le programme par RUN.

La partie sera évidemment d'autant plus difficile que l'on ne connaît pas le proverbe à reconstituer. On peut déplacer les lettres et les faire passer à l'emplacement du carré coloré, à l'aide des 4 touches de direction (→, ←, ↓, ↑).

Le nombre d'essais est affiché en permanence, à côté du puzzle.

On peut arrêter à tout moment en tapant RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

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


```

10 REM TAQUIN DE LETTRES
20 COLOR 0,11:COLOR 4,11:COLOR 5,7:GRAPH
IC 0,1
30 FOR I = 1 TO 12
40 PRINT"  "
50 NEXT I
60 A$ = ""
70 R$ ="ONASOUVENTBESOIND'UNPLUSPETITQUE
SOI"
80 FOR I = 1 TO 35
90 X = INT(RND(1)*35+1)
100 IF MID$(R$,X,1)="-" THEN 90
110 A$ = A$ + MID$(R$,X,1)
120 MID$(R$,X,1)="-"
130 F = ASC(MID$(A$,1,1))+64*(MID$(A$,1,
1)◇"/"):SOUND 1,12000,1
140 POKE1146+(I-6*INT(I/6))+40*INT(I/6),
F
145 POKE55418+(I-6*INT(I/6))+40*INT(I/6)
,9
150 NEXT I
160 X = 0 : Y = 0 : S = 0 : C = 32
170 PRINT"#####"TAB(15);"NOMBRE D'ES
SAIS : "S
180 POKE1146+Y+40*X,102:POKE55418+Y+40*X
,9
190 GET T$
200 IF T$="" THENPOKE1146+Y+40*X,C:GOTO1
80
210 I = X : J = Y
220 X = X - (T$=CHR$(17) AND X<5) + (T$=
CHR$(145) AND X>0)
230 Y = Y - (T$=CHR$(29) AND Y<5) + (T$=
CHR$(157) AND Y>0)
240 C = PEEK(1146+Y+40*X)
250 POKE1146+J+40*I,C
260 S = S + 1:SOUND 1,12000,2
270 GOTO 170

```

Cibles

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : hasard.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

Le nombre de cibles est entré en ligne 30. La boucle I (65-75) effectue l'impression des cibles.

La ligne 90 affichera le score en permanence.

La balle est affichée par la ligne 100.

La boucle I (120-170) réalise l'impression des chiffres contenus dans les cibles, et permet le déplacement de la boule (ligne 160).

La ligne 190 regarde si l'on a tiré, et la ligne 200 regarde si la boule est encore sur la ligne de départ.

La ligne 210 fait progresser la boule d'une case vers la droite, et la ligne 220 vérifie si l'on a ou non atteint les cibles ; puis les lignes 230 et 240 regardent ce que l'on a touché.

La ligne 250 incrémente le score.

Les lignes 260 et 270 s'occupent de la fin de partie.

POUR JOUER :

Lancer le programme par RUN, et taper le nombre de cibles désiré suivi de RETURN. Les cibles s'affichent, et la partie commence. Votre balle, représentée par un O est affichée en bas à gauche ; vous pouvez la déplacer avec les flèches haut et bas (touches ↑ et ↓), et vous pouvez tirer à tout moment en tapant Ø.

Le score sera modifié en fonction de votre résultat, et une nouvelle balle apparaît. Il est possible d'arrêter à tout moment en tapant RUN/STOP et RESTORE, 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.

Gouttes

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : dessin.

Difficulté du jeu : aucune.

LE JEU :

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

LE PROGRAMME :

Les deux boucles emboîtées I (20-90) et J (30-80) dessinent le récipient et les obstacles.

La ligne 130 dessine la goutte et la ligne 170 envoie dans un sous-programme (400-420) qui permet de déterminer s'il y a ou non rencontre avec un obstacle.

Si c'est non, on va en 300, et la goutte continue de tomber.

Si c'est oui, la goutte part à droite ou à gauche (ligne 190) selon la valeur définie pour K en ligne 110.

On regarde de nouveau s'il y a ou non obstacle, ou si la goutte peut continuer à tomber (lignes 200, 210 ou 220).

Selon le résultat du test, on utilisera 230, 300, 320 ou 350.

POUR JOUER :

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

Le programme ne s'arrête pas de lui-même lorsque le récipient déborde ; on peut l'arrêter à tout moment en tapant RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

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

```

10 REM GOUTTES
15 COLOR 0,12:COLOR 4,12:COLOR 5,7:GRAPH
IC 0,1
20 FOR I = 12 TO 20
30 FOR J = 1 TO 16
40 POKE1024+40*I,118:POKE55296+40*I,6:PO
KE1041+40*I,117:POKE55296+17+40*I,6
50 POKE1824+J,120:POKE56096+J,6
60 IF I<>12 AND I<>20 THENPOKE1024+J+40*
I,102:POKE55296+J+40*I,4
70 IF I<>20 AND RND(1)<.8 THEN GOSUB 500
80 NEXT J
90 NEXT I
100 Y = INT(RND(1)*7+6):X=6
110 K = -2*(RND(1)<0.5) - 1
120 I = 0
130 POKE1024+Y+40*X,81:POKE55296+Y+40*X,
0
140 SLEEP 1
150 POKE 1024+Y+40*X,32:SOUND 1,12000,2
160 U = X + 1 : V = Y
170 GOSUB 400
180 IF J = 0 THEN 300
190 U = X : V = Y + K
200 GOSUB 400
210 IF J = 0 THEN 320
220 IF I = 0 THEN 350
230 POKE1024+Y+40*X,81:POKE55296+Y+40*X,
0
240 GOTO 100
300 X = X + 1
310 GOTO 110
320 Y = Y + K
330 I = 1
340 GOTO 130
350 K = -K
360 I = 1
370 GOTO 190
400 C = PEEK(1024+Y+40*U)
410 J = -(C <> 32)
420 RETURN
500 S = INT(RND(1)*13)
510 POKE1025+S+40*I,32:POKE1026+S+40*I,3
2
520 RETURN

```

Obstacles

REFERENCES :

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 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 fin de partie, et le jeu se termine lorsque la voiture rencontre, soit le bord du terrain, soit un obstacle.

LE PROGRAMME :

La ligne 130 fait avancer les piétons vers la voiture du joueur qui est imprimée grâce à la ligne 80.

La ligne 100 imprime la route.

Les lignes 110 et 120 affichent les piétons.

La ligne 230 affiche le score.

La ligne 160 permettra de déplacer le véhicule.

Les lignes 170 et 180 permettent de contrôler s'il y a ou non un obstacle et arrêtent le programme s'il y a lieu.

Le score est incrémenté en ligne 190.

POUR JOUER :

On lance le programme en tapant RUN.

La route et les obstacles apparaissent en bas de l'écran et montent vers la voiture. La voiture peut se déplacer à l'aide des flèches de déplacement horizontal (touches → et ←).

La rencontre avec un obstacle ou la bordure interrompt le déroulement du programme et provoque l'affichage du score.

On peut, cependant, sortir du programme à tout moment en utilisant les touches RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

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

```

10 REM OBSTACLES
20 DATA 93,127,85,20,93,127,93,8,8,20,8,
62,42,8,28,54,255,255,255,255,255
30 DATA 255,255,255,0,0,0,0,0,0,0,0
40 POKE2604,(PEEK(2604)ANDNOT14)OR 8
50 FOR I=0TO31:READ A:POKE 8192+I,A:NEXT
60 COLOR 0,12:COLOR 4,12:COLOR 5,7:GRAPH
IC 0,1
70 S = 0 : Y = 5
80 POKE 1024+Y,0:POKE 55296+Y,7
100 POKE 1944,2:POKE 1953,2:POKE 56216,1
0:POKE 56225,10
110 Z=INT(RND(1)*8+1):POKE 1944+Z,1:POKE
56216+Z,3
120 Z=INT(RND(1)*8+1):POKE 1944+Z,1:POKE
56216+Z,9
130 PRINT"#####CCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC"
150 GET A$
160 Y = Y -(A$=CHR$(29) AND Y<9) + (A$=C
HR$(157) AND Y>0)
170 C = PEEK(1064 + Y)
180 IF C = 2 OR C = 1 THEN 210
190 S = S + 1
200 GOTO 80
210 POKE2604,(PEEK(2604)AND NOT 14)OR 4
220 SOUND 1,20000,300,2,3000,500,1
230 COLOR 5,9:PRINT"ACCIDENT- SCORE :";S

```

Captures dans l'espace

REFERENCES :

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

LE JEU :

Votre vaisseau spatial navigue dans un dangereux champ de météorites et votre mission est de vous emparer de la plus grande quantité possible de minerai d'urane, un métal indispensable que l'on ne peut trouver que dans ces lieux particulièrement dangereux.

Vous devez arriver à manœuvrer pour recueillir l'urane, sans vous faire heurter par une météorite.

LE PROGRAMME :

La ligne 90 permet de faire apparaître les météorites.

La ligne 110 fera apparaître le minerai d'urane et la ligne 120 assure le déplacement des météorites sur l'écran.

Les lignes 140 à 160 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 (lignes 220 à 240).

Votre vaisseau spatial est imprimé par la ligne 180 et son déplacement est contrôlé par la ligne 200.

POUR JOUER :

On lance le programme par RUN et les météorites commencent à monter sur l'écran, ainsi que les blocs d'urane.

Les touches de déplacement horizontal (→ et ←) seront utilisées pour déplacer votre vaisseau et récupérer l'urane, mais le moindre heurt avec une météorite provoquera la fin de la partie et l'affichage de la quantité d'urane capturée.

On pourra aussi arrêter à tout moment en tapant RUN/STOP et RE-STORE.

EXTENSIONS POSSIBLES :

On peut rendre le jeu de plus en plus difficile en augmentant progressivement le nombre de météorites, ou encore, en minutant la durée de la partie.

On peut aussi prévoir un affichage permanent du score, introduire un deuxième vaisseau spatial manœuvré par une puissance étrangère et qui cherche aussi à s'emparer de l'urane...


```

10 REM CAPTURES DANS L'ESPACE
20 DATA 0,73,28,54,119,54,28,73,0,28,28,
127,127,127,28,28,0,127,65,85,73,85
30 DATA 65,127,0,0,0,0,0,0,0,0
40 COLOR 0,14:COLOR 4,10:COLOR 5,6:GRAPH
IC 0,1
50 POKE2604,(PEEK(2604)AND NOT14)OR 8
60 FOR I=0TO31:READ A:POKE8192+I,A:NEXT
70 X = 15 : Z = 0
80 S = INT(RND(1)*40)
90 POKE1944+S,0:POKE56216+S,4
100 S = INT(RND(1)*40)
110 IF RND(1)<0.5 THEN POKE1944+S,1:POKE
56216+S,5
120 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC"
130 FOR I = 1 TO 10 : NEXT I
140 C = PEEK(1424+X)
150 IF C = 1 THEN GOTO 220
160 IF C=0 THEN Z=Z+1:SOUND 1,16000,20,2
,200,150,2
170 POKE1383+X,3:POKE1384+X,3:POKE1385+X
,3
180 POKE1424+X,2:POKE55696+X,1
190 GET A$
200 X = X - (A$=CHR$(29) AND X<40) + (A$
=CHR$(157) AND X>0)
210 GOTO 80
220 POKE2604,(PEEK(2604)AND NOT 14)OR 4
230 PLAY"CDEFG"
240 COLOR 5,2:PRINT"SCORE OBTENU : "Z

```

Anomalie

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

La boucle I (40-90) imprime les chiffres de repérage des grilles.

La boucle principale I (100-160) imprime les symboles des deux grilles, dans un ordre différent.

La ligne 110 choisit le caractère à imprimer.

Le sous-programme (1000-1040) s'occupe de faire l'impression à un emplacement libre (on l'appelle deux fois pour les deux grilles).

Les lignes 170 et 180 changent, de façon aléatoire, l'un des caractères.

Les coordonnées du point essayé sont entrées aux lignes 190 et 200, et le contrôle est effectué par les lignes 210 et 220.

Les lignes 230, 240, 250 et 260 s'occupent de l'affichage du résultat.

POUR JOUER :

Après avoir lancé le programme par l'instruction RUN, on obtient l'affichage des deux grilles. On devra alors chercher, par comparaison, quel est le caractère qui a été modifié (sur la grille de droite).

On entre alors son numéro de ligne, suivi de RETURN, et son numéro de colonne, suivi de RETURN.

Le programme répondra alors BRAVO ou PERDU et indiquera la lettre à découvrir et le temps total.

EXTENSIONS POSSIBLES :

Il sera possible d'utiliser des symboles graphiques de diverses couleurs, ce qui pourra simplifier la recherche.

On peut aussi envisager d'utiliser des dessins plus sophistiqués, améliorer le contrôle du résultat, admettre la bonne réponse sur la grille de droite ou celle de gauche, introduire un minutage, etc.

```

10 REM ANOMALIE
20 COLOR 0,5:COLOR 4,13:COLOR 5,5:GRAPHI
C 0,1
30 T = 5 :TI$="000000"
40 FOR I = 1 TO T
50 POKE 1024+2*I,48+I:POKE55296+2*I,7
60 POKE 1040+2*I,48+I:POKE55312+2*I,3
70 POKE 1024+80*I,48+I:POKE55296+80*I,7
80 POKE 1040+80*I,48+I:POKE55312+80*I,3
90 NEXT I
100 FOR I = 1 TO 22
110 D = INT(RND(1)*26+1)
120 L = 1 : S = T
130 GOSUB 1000
140 L = 9
150 GOSUB 1000
160 NEXT I
170 S = 0
180 GOSUB 1000:COLOR 0,11:COLOR 5,2
190 INPUT"XXXXXXXXXXXXXXXXLIGNE :";I
200 PRINT:INPUT"COLONNE :";T
210 U = 2*I : V = 2*T + 16
220 GOSUB 1010
230 PRINT"BRAVO..."
240 IF D <> C THEN PRINT"PERDU..."
250 PRINT"C'ETAIT LA LETTRE ";CHR$(D+64)
260 PRINT"VOTRE TEMPS :";MID$(TI$,3,2);"
  MN. , ";RIGHT$(TI$,2);" SEC."
270 PLAY"CDEFG":END
1000 U = 2*INT(RND(1)*T+1):V=2*INT(RND(1)
)*S+L)
1010 C = PEEK(1024+V+40*U)
1020 IF C <>32 AND S <>0 THEN 1000
1030 POKE1024+V+40*U,D:POKE55296+V+40*U,
4:SOUND 1,12000,1
1040 RETURN

```

Plusieurs en un

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : deux joueurs.

Difficulté du jeu : variable, selon l'utilisation.

LE JEU :

Plusieurs jeux sont en réalité possibles avec ce programme qui va se comporter comme un crayon électronique vous permettant d'inscrire des symboles de couleur, en n'importe quel emplacement de votre feuille-écran de télévision. On pourra, par exemple, l'utiliser pour un jeu de morpion (le premier à réaliser un alignement de 5 pions identiques gagne) ou pour un GO (on réalise des encadrements des pions adverses) ou tout autre jeu, selon son imagination.

LE PROGRAMME :

Le terrain de jeu est délimité par un cadre mis en place par les lignes 30 et 40.

Le changement de joueur est effectué par la ligne 60.

Le joueur qui a le trait voit le symbole qu'il a choisi (lettre X ou O) se placer en limite du cadre (ligne 80).

Les lignes 110 et 120 permettent aux deux joueurs de déplacer le symbole sur le terrain.

Les lignes 130, 140 et 150 contrôlent si le déplacement est possible (case libre et non dépassement des limites).

La ligne 220 permet d'écrire sur une case libre.

POUR JOUER :

RUN provoque l'affichage des limites du terrain et du numéro du joueur, ainsi que des symboles correspondants que l'on devra déplacer sur l'écran à l'aide des 4 touches de direction (→, ←, ↓, ↑).

Il n'est pas possible de venir sur une case déjà occupée et il faudra appuyer sur la 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 RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

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

```

10 REM PLUSIEURS EN UN
20 COLOR 0,11:COLOR 4,10:COLOR 5,2:GRAPH
IC 0,1
30 POKE1024,79:POKE55296,1:POKE1054,80:P
OKE55326,1
40 POKE1784,76:POKE56056,1:POKE1814,122:
POKE56086 1
50 J = 0
60 J = ABS(J-1):SOUND 1,4000*J+3000,8
70 X = 12 : Y = 0
80 POKE1024+Y+40*X,15-9*(J=1):POKE55296+
Y+40*X,2-4*(J=1)
90 GET A$ : IF A$ = "" THEN 90
100 A = X : B = Y
110 X = X - (A$=CHR$(17) AND X < 19) + (A
$=CHR$(145) AND X > 1)
120 Y = Y - (A$=CHR$(29) AND Y < 29) + (A$
=CHR$(157) AND Y > 1)
130 POKE1024+B+40*A,32:SOUND 1,14000,1
140 C = PEEK(1024+Y+40*X):D=PEEK(55296+Y
+40*X)
150 IF C <>15 AND C <>24 THEN 200
160 POKE1024+Y+40*X,C:POKE55296+Y+40*X,D
170 X = A : Y = B :PLAY"CG"
200 FOR I = 1 TO 10 : GET A$
205 IF A$ = "" THEN NEXT I
210 IF A$ <> "0" OR Y = 0 THEN 80
220 POKE1024+Y+40*X,15-9*(J=1):POKE55296
+Y+40*X,2-4*(J=1)
230 GOTO 60

```

Parcours du cavalier

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

Il se présente, ici, comme un jeu de hasard total, mais provient, en réalité, d'un problème assez difficile à résoudre : comment déplacer un cavalier sur un échiquier de façon qu'il passe une fois, et une seule, sur chacune des 64 cases du damier.

Le cavalier se déplace suivant les règles communément admises aux échecs, et sa position de départ est définie par le joueur ; par contre, ses déplacements sont effectués au hasard. Arrivera-t-il à parcourir les 64 cases ?

LE PROGRAMME :

Les lignes 40 et 50 définissent les divers déplacements possibles, dans deux variables alphanumériques A\$ et B\$; les boucles emboîtées I (60-110) et J (70-100) dessinent le terrain de jeu.

On entre les coordonnées de la case de départ (ligne, puis colonne) aux lignes 120 et 130.

La ligne 160 imprime le numéro du coup.

La ligne 180 permet le tirage au hasard d'un déplacement du cavalier.

La boucle Z (190-240) cherche si le déplacement est possible ; si ce n'est pas le cas, on essaie tous les autres (toujours dans le même ordre).

Les lignes 200 et 210 effectuent le déplacement.

La ligne 220 contrôle que l'on ne sorte pas du terrain de jeu.

La ligne 230 envoie en ligne 300 si le coup est possible.

La ligne 250 gère la fin de partie (cavalier bloqué) et affiche le score.

Les lignes 300 et 310 réalisent le coup et incrémentent le score.

POUR JOUER :

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

EXTENSIONS POSSIBLES :

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

Mur de briques

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexe (pour un joueur).

Difficulté du jeu : difficile.

LE JEU :

S'il est un jeu que l'on peut présenter comme un grand classique des jeux d'arcades pour micro-ordinateur, c'est bien le mur de briques (en anglais BREAKOUT).

Il est offert, ici, dans une version simplifiée, mais offre néanmoins de bonnes performances : affichage du score en fin de partie, rebonds aléatoires, etc.

La règle en est très élémentaire : vous disposez d'une raquette qui peut être déplacée horizontalement pour renvoyer une balle (matérialisée par un point) qui va démolir les briques dessinées en haut de l'écran.

Votre problème est d'en démolir le plus grand nombre possible, sans rater la balle, ce qui provoquerait la fin de partie et l'affichage du score.

LE PROGRAMME :

La boucle I (40-60) dessine le terrain de jeu et les briques sont mises en place par les boucles emboîtées I (70-110) et J (80-100).

La ligne 130 dessine la raquette dont la position est contrôlée par la ligne 150.

Les lignes 200 à 230 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 240 à 260 et 280 à 300.

La ligne 170 affiche le score en permanence.

POUR JOUER :

Le programme sera lancé par l'instruction RUN, ce qui provoque l'affichage du terrain de jeu puis, très rapidement, l'envoi de la balle. Le joueur devra alors essayer de déplacer sa raquette à l'aide des touches de déplacement horizontal (→ et ←).

Si, à un moment quelconque, sa balle ne touche pas la raquette, la partie s'arrête. On la relance en tapant RUN.

On peut aussi terminer à tout moment par RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

On peut envisager d'agrandir le terrain de jeu, d'augmenter le nombre de briques, de prévoir plusieurs balles, d'introduire des effets sonores.

On peut aussi, en s'inspirant des jeux d'arcades, introduire de petits ouvriers qui viendront reconstruire votre mur de briques, au fur et à mesure que vous le démolissez.


```

10 REM MUR DE BRIQUES
15 COLOR 0,15:COLOR 4,10:GRAPHIC 0,1
20 PRINT"J"
30 COLOR 5,1:PRINTTAB(9);"
"
40 FOR I = 1 TO 18
50 PRINTTAB(9);"
60 NEXT I
70 FOR I = 2 TO 7
80 FOR J = 11 TO 28
90 POKE1024+J+40*I,90:POKE55296+J+40*I,I
+2
100 NEXT J
110 NEXT I
120 H = -1 : K = -0.5 : J=18 : I = 17 :
Y = J
130 PRINT"
"TAB(Y-2)
;"
140 GET A$
150 Y = Y - (A$="." AND Y <28) + (A$=","
AND Y >8)
160 IF T=1 THEN A=1024+J+40*I:POKE A,32
170 PRINT"SCORE"S
180 IF I > 22 THEN PLAY"CDEFG":END
190 I = I + H : J = J + K
200 C = PEEK(1024+J+40*I)
210 IF C = 32 THEN POKE1024+J+40*I,81:PO
KE55296+J+40*I,0:T=1:GOTO 130
220 IF C = 102 THEN 260
230 IF C = 90 THEN POKE1024+J+40*I,81:PO
KE55296+J+40*I,0:T=1:GOTO 280
240 H = - H
250 K = RND(1)-RND(1)
260 K = -K : T = 0 :SOUND 1,12000,2
270 GOTO 130
280 S = S + 1:SOUND 1,4000,5
290 K=-(K-(2*(RND(1)<0.5)+1)*RND(1)/3)
300 H = -H
310 GOTO 130

```

Othello

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : difficile.

LE JEU :

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

Le joueur ayant les O procédera de la même façon, et ceci jusqu'à ce que le damier soit rempli ou que la personne ne puisse plus jouer (si un joueur ne peut retourner des pions, il doit passer son tour).

LE PROGRAMME :

Les boucles emboîtées P (60-130) et Q (80-120) dessinent le terrain de jeu et les lignes 150 et 160 affichent le joueur.

La boucle P (230-350) contrôle la validité du coup (la case est-elle libre : lignes 330 et 340), en utilisant une partie de programme qui se trouve aux lignes 390 à 450 (a-t-on pris un ou plusieurs ennemis en ligne 390 et retournement éventuel autorisé par la ligne 410 : on refait un balayage dans toutes les directions, si l'encadrement s'est révélé possible).

La ligne 230 définit les 8 directions d'exploration.

Les lignes 270 et 280 calculent les déplacements à effectuer pour explorer ces 8 directions.

La ligne 260 réalise les déplacements.

La ligne 370 assure le changement de joueur, s'il y a eu retournement (contrôlé en ligne 360).

POUR JOUER :

Après RUN, chaque joueur entrera à son tour les coordonnées (ligne, RETURN, puis colonne, RETURN) de la case où il désire jouer ; si le coup est accepté, les pions sont retournés et l'on change de joueur ; si un coup n'est pas possible, on entre 0 en première coordonnée pour le changement de joueur.

EXTENSIONS POSSIBLES :

Avant tout, bien sûr, améliorer l'affichage avec des couleurs, une grille plus aérée.

On pourra aussi imaginer un jeu machine pour le C128.

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

```

10 REM OTHELLO
20 COLOR 0,11:COLOR 4,11:COLOR 5,2:GRAPH
IC 0,1
30 INPUT"DIMENSION DE LA GRILLE ":A
40 IF A/2 <>INT(A/2) THEN RUN
50 COLOR 5,9:PRINT"J"
60 FOR P = 1 TO A
70 CHAR 1,2*P,0,MID$(STR$(P),2)
80 FOR Q = 1 TO A
90 CHAR 1,0,2*P,MID$(STR$(P),2):POKE1024
+2*Q+80*P,46:POKE55296+2*Q+80*P,1
100 IF Q=A/2+1 AND P=A/2 THENBEGIN:POKE1
022+2*Q+80*P,24:POKE1024+2*Q+80*P,15
105 POKE 55294+2*Q+80*P,2:POKE 55296+2*Q
+80*P,6:BEND
110 IF Q=A/2+1 AND P=A/2+1 THENBEGIN:POK
E1022+2*Q+80*P,15:POKE1024+2*Q+80*P,24
115 POKE 55294+2*Q+80*P,6:POKE 55296+2*Q
+80*P,2:BEND
120 NEXT Q
130 NEXT P
140 K = 1
150 COLOR 5,-3*(K=1)-7*(K=0):PRINT"X000";
TAB(23)"JOUEUR X":SOUND 1,3000*K+4000,1
160 IF K = 0 THEN PRINT"J"TAB(23);"JOUEU
R 0"
170 PRINT"0"TAB(23);" "
180 COLOR 5,2:INPUT">XXXXXXXXXXXXXXXXXXXXX
XXXXLIGNE";U
190 IF U = 0 THEN 370
200 INPUT">XXXXXXXXXXXXXXXXXXXXXCOLONN
E";V
210 PRINT"J"TAB(23);" "
220 H = 1
230 FOR P = 1 TO 8
240 S = 0
250 W = 0
260 T = 2*U : R = 2*V
270 F = (P>3 AND P<7) - (P<3 OR P>7)
280 G = (P>1 AND P<5) - (P>=6)
290 Q = 46 + 22*(S=1) + 31*(S=2)
300 IF S = 0 THEN 320
310 POKE 1024+R+40*T,Q:POKE 55296+R+40*T
,-2*(S=1)-6*(S=2)
320 T = T + 2*F : R = R + 2*G
330 C = PEEK(1024+R+40*T)
340 IF C = 15 OR C = 24 THEN 390
350 NEXT P
360 IF H <> 0 THEN 150
370 K = ABS(K-1)
380 GOTO 150
390 IF (C=15 AND K=1) OR (C=24 AND K=0)
THEN W=1
400 IF S <> 0 THEN POKE1024+R+40*T,Q:POKE
55296+R+40*T,-2*(S=1)-6*(S=2)
410 IF W=1 AND ((C=15 AND K=1) OR (C=24
AND K=0)) THEN 320
420 IF W=0 OR S<>0 THEN 350
430 S = 1 - (K=0)
440 H = -H*(S=0)
450 GOTO 250

```

Echecs

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de réflexion, pour deux joueurs.

Difficulté du jeu : difficile.

LE JEU :

Le programme propose seulement à deux joueurs de disputer une partie en se servant de l'écran du téléviseur comme plateau de jeu. Il n'y a pas de contrôle de la validité des coups, mais seulement gestion de la position des pièces sur l'échiquier.

Les pions sont représentés par des "piques" et les autres pièces par leurs initiales (T pour tour, F pour fou, C pour cavalier, Q pour reine, et K pour roi). La couleur (vert ou jaune) permet de distinguer chaque coup.

LE PROGRAMME :

Les lignes 60 à 160 dessinent l'échiquier.

Les lignes 170 à 240 définissent les conditions initiales, à l'aide de deux tableaux : R (I, J) pour les codes des pièces, et C (I, J) pour la couleur.

Les boucles emboîtées J (260-320) et I (270-310) dessinent les pièces.

La ligne 350 permet l'entrée de la ligne et de la colonne de départ, avec contrôle en ligne 370.

La ligne 390 permet d'entrer colonne et ligne d'arrivée.

La ligne 410 note la nouvelle position.

La ligne 420 assure le changement de joueur.

POUR JOUER :

On lance l'exécution par RUN.

L'échiquier est dessiné, puis les pièces, et enfin l'inscription JOUEUR BLEU, suivie de DEPART (C et L) ; on entre la colonne et la ligne de départ, sous la forme E7 par exemple, on obtient alors l'affichage de ARRIVEE (C et L) et on répond de même.

Le déplacement est alors effectué, et c'est au tour de l'autre joueur.

Pour arrêter, il suffit de taper RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

On pourra chercher à réaliser un graphisme plus précis, introduire le contrôle du mouvement des pièces. Et pourquoi pas, écrire un programme qui permette au Commodore de se comporter en partenaire honorable. Bon courage !

```

10 REM ECHECS
15 DIM R(8,8) : DIM C(8,8)
20 COLOR 0,8:COLOR 4,12:COLOR 5,2:GRAPH1
C 0,1
30 A$ = "      "
40 B$ = "  ■  "
50 C$ = "  ■  "
60 PRINT"      A  B  C  D  E  F  G  H"
80 FOR I = 1 TO 8
90 FOR J = 1 TO 8
100 IF (I+J)/2 = INT((I+J)/2) THEN 140
110 COLOR5,1:PRINTTAB(3*J);A$:PRINTI;TAB
(3*J);B$:PRINTTAB(3*J);C$;
130 PRINT"TTT"
140 NEXT J
150 PRINT"0"
160 NEXT I
170 FOR I = 1 TO 8
180 FOR J = 1 TO 8
190 IF I = 2 OR I = 7 THEN R(I,J)=65
200 IF I <>2 AND I <>7 THEN R(I,J)=96
205 C(I,J) = 2
210 IF I = 7 OR I = 8 THEN C(I,J)=6
220 NEXT J
230 NEXT I
240 R(1,1)=20:R(1,2)=3:R(1,3)=6:R(1,4)=1
7:R(1,5)=11:R(1,6)=6:R(1,7)=3:R(1,8)=20
250 FOR I = 1 TO 8 : R(8,I)=R(1,I):NEXTI
255 H = 1
260 FOR J = 1 TO 8
270 FOR I = 1 TO 8
280 X = 1 + 3*I : Y = 2*I
290 POKE 1024 + X + 40*I , R(I,J)
300 POKE 55296 + X + 40*I , C(I,J)
310 NEXT I
320 NEXT J
330 PRINT"#####";TAB(28);"      "
335 COLOR5,4*H+3:PRINT"#####
#####JOUVEUR ROUGE " :SOUND 1,4000*H+3000,4
340 IF H = 1 THEN PRINT"#####
#####JOUVEUR BLEU "
345 PRINTTAB(16);"      "
350 INPUT"DEPART (C ET L) ";D$
352 PRINT" ";TAB(16);"      "
355 PRINT"#####
#####";D$
360 GOSUB 500
370 IF R(L,C)=32 OR (C(L,C)=5 AND H = 1)
OR (C(L,C)=7 AND H=0) THEN 335
380 T=R(L,C):S=C(L,C):R(L,C)=96
390 INPUT"#####ARRIVEE
(C ET L)";D$
395 PRINT"#####
#####- ";D$
400 GOSUB 500
410 R(L,C)=T:C(L,C)=S
420 H = ABS(H-1)
430 GOTO 260
500 L = VAL(RIGHT$(D$,1)) : C = ASC(LEFT
$(D$,1))-64
510 RETURN

```

Festival

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu d'adresse.

Difficulté du jeu : facile.

LE JEU :

Des envahisseurs plutôt souriants, semblables à des ballons d'enfant de diverses couleurs, se déplacent en tous sens sur l'écran.

Le joueur dispose d'une batterie de canons et de 100 munitions pour réussir à les détruire. Mais il ne peut tirer (en appuyant sur la barre d'espace) que s'il est exactement derrière un canon. Et il n'est pas certain que son projectile fasse éclater le ballon à tous les coups.

LE PROGRAMME :

Les lignes 20 à 150 contiennent les différentes formes utilisées au cours du jeu ; les boucles J (75-120) et I (80-110) définissent avec des DATA les formes du personnage et des projectiles. Les autres formes sont définies par des instructions SSHAPE (ligne 40 et 60).

Le déplacement des ballons est réglé par la ligne 170.

Le programme proprement dit est contenu dans une boucle qui va de la ligne 180 à la ligne 240. Les contrôles de tir sont effectués dans les lignes 210 à 220, avec appel d'un sous-programme en ligne 300.

Le déplacement du personnage est réalisé par la ligne 230.

Les lignes 260 et 270 gèrent la fin de partie.

POUR JOUER :

L'exécution est lancée par RUN. On attend quelques instants, puis le décor est mis en place, et les ballons commencent à se déplacer. Les flèches droite et gauche permettent de déplacer le personnage. On tire en appuyant sur la barre d'espace, le tir n'étant possible que si l'on est exactement sous un canon.

Si l'on a touché un ballon, celui-ci disparaît, mais ce n'est pas toujours le cas. Le jeu s'arrête lorsque les 6 ballons sont détruits ou si l'on n'a plus de munitions.

On peut terminer à tout moment en tapant RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

On pourra contrôler davantage la trajectoire des ballons, les faire rebondir les uns sur les autres. On pourra aussi obliger le joueur à éviter les ballons qui viennent sur lui, etc.

```

10 REM FESTIVAL
20 COLOR 0,15:COLOR 1,15:COLOR 4,13:GRAP
HIC 1,1
30 CIRCLE 1,12,12,9:CIRCLE 1,10,8,2:CIRC
LE 1,14,8,2:CIRCLE 1,12,9,7,7,140,220
40 SSHAPE A$,2,2,22,22:FOR I=1 TO 6:SPRS
AV A$,I:NEXT
50 BOX 1,49,0,51,12:BOX 1,44,8,46,16:BOX
1,54,8,56,16:BOX 1,48,12,52,18,0,1
60 SSHAPE B$,44,0,56,18
70 DATA 0,8,8,8,28,62,28,0,8,20,8,62,42,
8,28,54
75 FOR J=1 TO 2
80 FOR I=0 TO 21
90 IF I<8 OR I>15 THEN A=0:ELSE READ A
100 C$(J)=C$(J)+CHR$(0)+CHR$(A)+CHR$(0)
110 NEXT I
120 NEXT J
130 SPRSAV C$(1),7:SPRSAV C$(2),8
140 COLOR 1,4:BOX 1,0,180,319,199,0,1
150 COLOR 1,3:FOR I=1 TO 7:GSHAPE B$,48*
I-48,160:NEXT
160 P=0:X=16:T=100:POKE2594,0:POKE2592,1
170 FOR I=1 TO 6:SPRITE 1,1,1,1:MOVSPR I
,RND(1)*360#INT(RND(1)*9+2):NEXT
180 SPRITE8,1,7,0,1,1:MOVSPR8,8*X+8,216
200 GET A$:IF A#<>" " OR X<>6*INT(X/6) T
HEN 230
210 SPRITE7,1,1:MOVSPR7,8*X+20,190:MOVSP
R7,0#15:A$="":T=T-1
212 SOUND 1,3000,10,2,300,300,2:IF T<0 O
R P=6 THEN GOTO 250
215 Z=RSPPOS(7,1):Y=BUMP(1)
217 IF Y<128 AND Y>63 THEN GOSUB 300
220 IF Z>50 THEN 215:ELSE SPRITE 7,0,1
230 X=X-(A#=CHR$(29))+(A#=CHR$(157))
240 GOTO 180
250 REM FIN
260 CHAR 1,12,12,"C'EST FINI":PLAY"CDEFG
":CHAR 1,10,14,"VOTRE SCORE "
270 CHAR 1,22,14,STR$(S):END
300 REM CONTROLE DU RESULTAT
320 S=S+1:SOUND 1,4000,50,2,100,100,1:Y=
Y-64
330 FOR I=6 TO 1 STEP -1
340 IF Y=2*(I-1) THEN SPRITE 1,0,I:P=P+1
350 NEXT I
360 RETURN

```

Quatre à quatre

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion (un ou deux joueurs).

Difficulté du jeu : difficile.

LE JEU :

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

LE PROGRAMME :

La variable alphanumérique A\$, en ligne 40, contient la table des coups possibles (déplacement du cavalier).

Les boucles emboîtées I (50-110) et J (60-100) dessinent le terrain de jeu, et les positions de départ.

Les lignes 130 et 140 permettent d'afficher O ou X, selon le joueur qui a le trait.

La ligne 170 permet l'affichage du curseur clignotant (ici un pavé de couleur) qui indique sur quelle case on se trouve.

Les lignes 200 et 210 permettent le déplacement du curseur.

La ligne 220 regarde si l'on a appuyé sur une touche de 1 à 8, donnant la direction du déplacement.

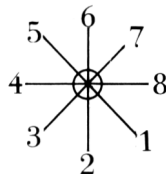
La ligne 230 réimprime la figure qui était sous le curseur. La ligne 260 transforme en nombre le symbole chiffre frappé. La ligne 250 réalise le contrôle de départ.

Les lignes 270 et 280 effectuent le déplacement. La ligne 300 réalise le contrôle d'arrivée. Le résultat est imprimé par la ligne 310, le changement de joueur est réalisé par la ligne 320 et un "." est imprimé à l'emplacement de départ. Le sous-programme 350-360 est utilisé pour les divers contrôles.

POUR JOUER :

On lance le programme par RUN. Le terrain de jeu est affiché, ainsi que le symbole permettant d'identifier le joueur qui a le trait. Le joueur peut choisir le pion qu'il désire déplacer en manœuvrant le curseur clignotant grâce aux touches de déplacement du curseur.

Pour déplacer son pion, il suffit, lorsque le curseur est sur ce pion, de taper la direction du déplacement, suivant le code :



Le coup sera refusé si la case de départ n'est pas conforme ou si la case d'arrivée est occupée. Sinon, le coup est réalisé, et le trait passe à l'autre joueur, qui déplace le curseur clignotant pour l'amener sur l'un des pions, et joue de la même façon.

Le joueur gagnant est celui qui réussit à amener le premier tous ses pions dans les cases initialement occupées par l'adversaire. On peut interrompre la partie à tout moment en tapant RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES :

On peut améliorer les contrôles, prévoir un test de fin, utiliser un terrain de jeu différent ; on pourra aussi envisager l'écriture d'un programme permettant au C128 de participer au jeu avec de bonnes chances de l'emporter.

```
10 REM QUATRE A QUATRE
20 COLOR 0,12:COLOR 4,12:COLOR 5,1:GRAPH
   IC 0,1
30 H = 1
40 A$ = "5665322356"
50 FOR I = 1 TO 5
60 FOR J = 1 TO 5
70 POKE1024+80*I,48+I:POKE1024+2*I,48+I:
   POKE1024+2*I+80*J,46
75 POKE55296+2*I+80*J,1:POKE55296+80*I,5
   :POKE55296+2*I,8
80 IF I<3 AND J<3 THEN POKE 1024+2*I+80*
   J,15
90 IF I>3 AND J>3 THEN POKE 1024+2*I+80*
   J,24
100 NEXT J
110 NEXT I
120 X = 1 : Y = 1
130 PRINT "#####JOUEUR 0"
140 IF H = 0 THEN PRINT "JOUEUR X"
150 GOSUB 350
160 I = J
170 POKE1024+2*X+80*Y,81:POKE55296+2*X+8
   0*Y,10
180 A = X : B = Y
190 GET R$
200 Y = Y - (R$="B" AND Y<5) + (R$="H" A
   ND Y>1)
210 X = X - (R$="D" AND X<5) + (R$="G" A
   ND X>1)
220 IF R$<"9" AND R$>"0" THEN 250
230 POKE1024+2*A+80*B,1:POKE55296+2*A+80
   *B,1
240 GOTO 130
```

```

250 IF (I=46) OR (I=15 AND H=0) OR (I=24
    AND H=1) THEN PLAY"CCC":GOTO 230
260 J = VAL(R$)
270 Y = Y + VAL(MID$(R$,J,1)) - 4
280 X = X + VAL(MID$(R$,J+2,1)) - 4
290 GOSUB 350
300 IF J <> 46 THEN PLAY"CCC":GOTO 230
310 POKE1024+2*X+80*Y,15-9*(H=0)
320 H = ABS(H-1):SOUND 1,3000*H+3000,4
330 I = 46
340 GOTO 230
350 J = PEEK(1024+2*X+80*Y)
360 RETURN

```

Annexe

COMMENTAIRES SUR L'AFFICHAGE

L'AFFICHAGE DES CARACTERES SPECIAUX
DEPLACEMENTS DU CURSEUR

- CURSEUR A DROITE S'IMPRIME : "▀"
- CURSEUR A GAUCHE S'IMPRIME : "▄"
- CURSEUR EN HAUT S'IMPRIME : "▴"
- CURSEUR EN BAS S'IMPRIME : "▾"
- SHIFT - CLR/HOME S'IMPRIME : "⌂"
- CLR/HOME S'IMPRIME : "⌂"

TOUCHES UTILISEES POUR DEPLACER LE
CURSEUR LORS DES JEUX : ",",

, - . - 0 - J OU LES QUATRE FLECHES
POUR LES COULEURS

ON UTILISE LA TOUCHE - CTRL - SUIVIE
D'UN NUMERO DE 1 A 0

- CTRL 1 DONNE : "■" (SOIT NOIR)
- CTRL 2 DONNE : "□" (SOIT BLANC)
- CTRL 3 DONNE : "■" (SOIT ROUGE)
- CTRL 4 DONNE : "■" (SOIT TURQUOISE)
- CTRL 5 DONNE : "■" (SOIT VIOLET)
- CTRL 6 DONNE : "■" (SOIT VERT)
- CTRL 7 DONNE : "■" (SOIT BLEU)
- CTRL 8 DONNE : "■" (SOIT JAUNE)

DE PLUS CTRL 9 , SOIT "■", PERMET LE
CONTRASTE INVERSE

ET CTRL 0 , SOIT "■", PERMET DE REVEN
IR EN CONTRASTE NORMAL

ON TROUVE LES AUTRES COULEURS AVEC
LA TOUCHE COMMODE

- AINSI COMM 1 DONNE "■" (SOIT ORANGE)
- COMM 2 DONNE "■" (SOIT MARRON)
- COMM 3 DONNE "■" (SOIT ROUGE CLAIR)
- COMM 4 DONNE "■" (SOIT GRIS 1)
- COMM 5 DONNE "■" (SOIT GRIS 2)
- COMM 6 DONNE "■" (SOIT VERT CLAIR)
- COMM 7 DONNE "■" (SOIT BLEU CLAIR)
- COMM 8 DONNE "■" (SOIT GRIS 3)

LES INSTRUCTIONS PRINT PEUVENT CONTE
NIR TOUTES CES COMMANDES

Conseils de lecture

Pour approfondir vos connaissances en Basic 7.0 et mieux connaître le système du C 128, P.S.I. vous propose une palette d'ouvrages utiles.

POUR MAITRISER LE BASIC DU C 128

- **Basic Commodore 128 – Méthodes pratiques**, Jacques Boisgontier (Éditions du P.S.I.).
Pour ceux qui ont déjà pratiqué un Basic, voici un ouvrage de perfectionnement au Basic 7.0. Un chapitre sur le CP/M Plus donne les principales commandes système.

POUR MIEUX CONNAITRE LE SYSTÈME DU COMMODORE 128

- **Clefs pour Commodore 128 –1– Mode 64**, Daniel-Jean David (Éditions du P.S.I.).
Mémento présentant synthétiquement le jeu d'instructions du 6510, les points d'entrée des routines système, les connecteurs, les messages d'erreurs, etc. Le livre de chevet du programmeur sur Commodore 128.
- **CP/M Plus sur Commodore 128**, Yvon Dargery (Éditions du P.S.I.).
Toutes les commandes CP/M et CP/M Plus pour maîtriser le système du Commodore 128 : un ouvrage de référence illustré par de nombreux programmes.

A PARAÎTRE :

- **Clefs pour Commodore 128 –2– Mode 128**, Jean-François Sehan (Éditions du P.S.I.).
Ce deuxième tome du Clefs pour C 128 donne les adresses Basic du 7.0, l'architecture du CP/M Plus et toutes les informations indispensables au programmeur sur Commodore 128.

Votre avis nous intéresse

- Pour nous permettre de faire de meilleurs livres, adressez-nous vos critiques sur le présent livre.
- Si vous souhaitez des éclaircissements techniques, écrivez-nous, nous adresserons votre demande à l'auteur qui ne manquera pas de vous répondre directement.

- Ce livre vous donne-t-il toute satisfaction?

- Y a-t-il un aspect du problème que vous auriez aimé voir abordé?

Comment avez-vous eu connaissance de ce livre?

- | | |
|---|-------------------------------------|
| ☐ publicité | ☐ cadeau |
| ☐ catalogue | ☐ librairie |
| ☐ boutique micro | ☐ exposition |
| ☐ autres | |

Avez-vous déjà acquis des livres PSI?

lesquels? _____

qu'en pensez-vous? _____

Nom _____ Prénom _____ Age _____

Adresse _____

Profession _____

Centre d'intérêt _____

CATALOGUE GRATUIT

Vous pouvez obtenir un catalogue complet des ouvrages PSI, sur simple demande, ou en retournant cette page remplie à votre libraire, à votre boutique micro ou aux

Editions du PSI
BP 86
77402 Lagny-sur-Marne Cedex

Achévé d'imprimer en février 1986
sur les presses de l'imprimerie Laballery et C^{ie}
58500 Clamecy
Dépôt légal : février 1986
N° d'impression : 602026
N° d'édition : 86595-322-1
ISBN : 2-86595-322-X



Apprendre en se distrayant, tel est l'objectif de ce livre.
Ces 102 programmes de jeu vous guident progressivement
dans l'exploration du Basic 7.0. Niveau par niveau,
vous assimilez de nouvelles instructions pour arriver
à la maîtrise de votre Commodore 128.

Les modes 64 et 128 sont étudiés en détail.

Chaque niveau commence par une présentation pédagogique
du jeu d'instructions utilisé.

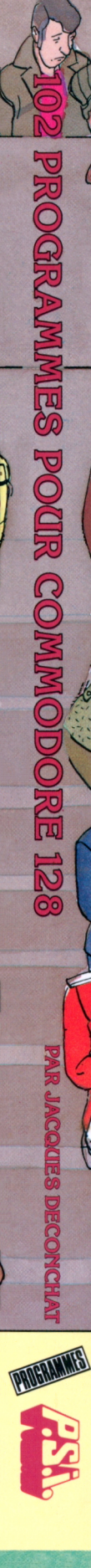
Tous les jeux sont commentés et les programmes analysés
ligne par ligne afin d'en faciliter la modification.

**DES LIVRES
POUR VOTRE
ORDINATEUR**



ISBN: 2.86595. 322. X
120FF

Couverture GIBRAT



102 PROGRAMMES POUR COORDONNER VOS ACTIVITÉS

PAR JACQUES DECOURCHAT

PROGRAMME

102