102 PROGRAMMES

POUR

COMMODORE 64



P.S.I.

PAR JACQUES DECONCHAT

P.S.I.

PROGRAMMES

102
PROGRAMMES POUR
Commodore 64

Autres ouvrages relatifs au Commodore 64

— La découverte du Commodore 64 — Daniel-Jean David
— La pratique du Commodore 64 — Daniel-Jean David
— L'assembleur du Commodore 64 — Daniel-Jean David
— Jeux, trucs et comptes pour Commodore 64 — Michel Bénelfoul et Cyril Cambien
— Commodore 64 en famille — Jean-François Séhan
— Le Commodore 64 à l'affiche — Jean-François Séhan
— Commodore 64 pour tous — Jacques Boisgontier, Sophie Brébion et Gérard Foucault
— Commodore 64 : méthodes pratiques — Jacques Boisgontier
— Clefs pour le Commodore 64 — Daniel-Jean David

A paraître :

— Destination aventure sur Commodore 64 — Delton T. Horn

Ouvrages d'initiation

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

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

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

© Editions du P.S.I., B.P. 86, 77402 Lagny Cedex, 1985
ISBN : 2-86595-115-4

JACQUES
DECONCHAT

102
PROGRAMMES POUR
Commodore 64

EDITIONS DU P.S.I.
1984

Sommaire

	Pages
Avant-propos	7
Organisation de l'ouvrage	9
NIVEAU 1	11
Les instructions élémentaires	13
Le nombre mystérieux	18
Cube	20
Chaud, froid	22
Les allumettes	24
Allumettes à plusieurs	26
Calendrier perpétuel	29
Bataille navale	30
Golf	32
Solfège	34
Le jeu du 21	36
Pair-impair	38
Devine un chiffre	40
Le jeu de Witthof	42
Rebonds	44
Jackpot	46
Roulette	48
Couloirs	50
Echiquier	52
Le jeu de la fève	54
Jeu de l'oie aléatoire	56
NIVEAU 2	61
Dimensionnement d'un tableau - Recherche d'un élément dans un tableau et une chaîne - L'instruction DATA	60
Dé truqué	62
Visions	64
421	66
L'électricien fou	68
La course au 20	70
Dessin	72
Musique	74
Poker	76
Le jeu de saute-mouton	78
Lettres	80
Machinamot	82
Espion	84
Accrochez les wagons	86
Tirage d'un dé truqué	88
Tour de cartes	90
Le jeu de Marienbad	92
Compose	94
Jour de la semaine	96
Le jeu de la vie (manuel)	98
Le jeu de la vie (automatique)	100
NIVEAU 3	
Manipulation des chaînes de caractères	103
Un dé	104
Plus grand, plus petit	106
Ordre	108
Le pendu	110
Belote	112
Un taquin de calcul	114

Pendu inversé	116
Nicomaque	118
Course de chevaux	120
A travers	122
Mastermind	124
Idem	126
Horloge 1	128
Reines sur l'échiquier	130
Répétitions	132
Mots	134
Permutations	136
Bridge	138
Le loup et les agneaux	140
Hanoï	142
NIVEAU 4	145
Les techniques d'animation élémentaire	146
Tirage de dés	148
Jeu de loto	150
La bombe	152
Ardoise magique	154
Affichage publicitaire	156
Evasion	158
Chapeaux	160
Combat aérien	162
Dessin de dés	164
Le jeu du missile	166
Le singe	168
Le jeu du 15	170
Adresse	172
Sous-marin	174
Zombies	176
Diago	178
Decomaque	180
Chasse au sous-marin	182
Clavier	184
Calculatrice	186
NIVEAU 5	189
Les manipulations directes des mémoires	191
Smokey	192
Course de voiture	194
Echecs	196
Danger spacial	198
Tic tac toe	200
Horloge 2	202
Dames	204
Anagrammes	206
Oeufs	208
Affichage tournant (vertical)	210
Quatre à quatre	212
Taquin de lettres	214
Cibles	216
Gouttes	218
Obstacles	220
Captures dans l'espace	222
Anomalie	224
Plusieurs en un	226
Parcours du cavalier	228
Mur de briques	230
Othello	232
Mathieu	234
Imprime	236
Annexe	239

AVANT-PROPOS

Vous trouverez dans ce livre plus de 100 programmes, rédigés en Basic Microsoft, pour l'ordinateur individuel Commodore 64; la plupart des programmes seront d'ailleurs utilisables tels quels sur les autres ordinateurs de la gamme Commodore, comme la série des PET/CBM, ou le VIC 20.

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

La brièveté des programmes proposés permettra de les taper facilement, sans risque d'erreur; vous pourrez, bien sûr, les utiliser comme base pour la réalisation de programmes plus performants; les programmes sont abondamment commentés pour vous aider à comprendre leur fonctionnement, et les pages de présentation vous permettront de retrouver rapidement les caractéristiques propres des instructions Basic qu'ils utilisent.

Les programmes n'utilisent pas toutes les particularités d'un appareil comme le Commodore 64, par ailleurs très bien détaillées dans le manuel de base. En effet, le son n'a pas été abordé, ni la possibilité tout à fait remarquable de création de SPRITES, dont dispose cet appareil. La maîtrise de ces différentes techniques exige en effet du lecteur une certaine connaissance de la programmation, qui dépasse le cadre de cet ouvrage.

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

Organisation de l'ouvrage

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

- **Le niveau 1** permet d'étudier les instructions élémentaires du langage Basic : **LET** (optionnel ici) **PRINT, INPUT, IF THEN , FOR...NEXT, GOTO, GOSUB, ON GOTO - ,ON - GOSUB**

- **Au niveau 2**, on utilisera les instructions de confection et de manipulation des éléments d'un tableau, et de recherche d'un élément dans une chaîne : **DIM, R(I), MID$, READ, DATA.**

- **Au niveau 3**, on utilise la panoplie complète des instructions de traitement des chaînes de caractères : **MID$, RIGHT$, LEFT$.**

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

- Enfin, **le niveau 5** nous fait découvrir les instructions frontières du Basic, qui permettent la manipulation directe dans les mémoires de l'appareil : **PEEK** et **POKE.**

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

Le programme, fonctionnant sur le Commodore 64, est ensuite proposé, suivi d'un exemple d'exécution. Les listings ont été obtenus directement d'après les originaux de l'auteur, sur une imprimante matricielle, ce qui devrait garantir, contre tout risque de disfonctionnement, le programme ayant servi à l'impression, très élémentaire (il n'accepte pas certains caractères) est fourni en fin d'ouvrage. Des explications sur les codes écran très particuliers qui apparaissent dans les PRINT, lors d'un listing, sont fournies en ANNEXE.

NIVEAU 1

N°	Nom du programme	Thème et particularités
1	**Le nombre mystérieux**	Jeu de déduction, pour un joueur. Plusieurs niveaux
2	**Cube**	Jeu de dessin : perspective sur un écran
3	**Chaud, froid**	Logique et hasard, pour un joueur. Plusieurs niveaux.
4	**Les allumettes**	Jeu de reflexion, contre l'ordinateur.
5	**Allumettes à plusieurs**	Jeu de reflexion et de stratégie, pour plusieurs joueurs.
6	**Calendrier perpétuel**	Confection d'un calendrier, mois par mois.
7	**Bataille navale**	Un seul navire à trouver, mais quelques particularités.
8	**Golf**	Un parcours de 18 trous, pour un seul joueur.
9	**Solfège**	Apprenez le solfège à l'aide de votre ordinateur.
10	**Le jeu du 21**	Jeu de hasard, l'ordinateur tient la banque.
11	**Pair - Impair**	Un jeu de casino en trois temps (1 ou plusieurs joueurs).
12	**Devine un chiffre**	Un jeu très facile, pour un ou plusieurs joueurs.
13	**Le jeu de Wytthof**	Un jeu de reflexion assez simple, pour deux joueurs.
14	**Rebonds**	Un jeu d'adresse très visuel, pour un joueur.
15	**Jackpot**	Jeu de hasard, pour un joueur.
16	**Roulette**	Une simulation élégante du jeu très connu.
17	**Couloirs**	Un jeu de logique, à découvrir.
18	**Echiquier**	Une base graphique pour vos propres créations.
19	**Le jeu de la fève**	Un jeu de reflexion, pour deux joueurs.
20	**Jeu de l'oie aléatoire**	Un très vieux jeu, revu et corrigé par l'ordinateur, 2 joueurs.

Instructions élémentaires

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

- ***RND(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 I, et présentes dans pratiquement tous les programmes sont :

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

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

L'INSTRUCTION LET,
OU INSTRUCTION D'AFFECTATION

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

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

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

Sur le Commodore 64, comme dans la plupart des Basic, le LET est devenu optionnel, aucune ambiguité n'étant possible, on écrira donc : A = 12 ; NOM$ = "TOTO" ; etc.

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

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

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

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

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

Le Commodore 64 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 Commodore 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'introduire des commandes de mouvement du curseur.

On pourra, par exemple, écrire :

```
INPUT "VOTRE NOM, S'IL VOUS PLAIT" ; A$
```

L'INSTRUCTION GOTO
OU INSTRUCTION DE RENVOI INCONDITIONNEL

Une des particularités du langage Basic, qui le rend facilemant accessible, est que ***chaque ligne de programme doit être précédée d'un numéro***, ces lignes étant ensuite exécutées par l'ordinateur individuel en séquences, dans le sens des numéros croissants. Pour diverses raisons, il pourra être nécessaire au cours du déroulement d'un programme, de ne pas suivre cet ordre séquentiel très rigide ; on utilisera alors ***l'instruction GOTO, qui permet de détourner le fonctionnement du programme de son cours normal*** quitte d'ailleurs à le reprendre ensuite par un autre GOTO. Le renvoi doit, par ailleurs, impérativement se faire sur un numéro de ligne existant dans le programme (GOTO 2Ø, par exemple).

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

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

L'INSTRUCTION IF ___ THEN ___
OU INSTRUCTION DE TEST

On peut l'utiliser de deux façons :

- soit en rupture de séquence conditionnelle, sous la forme **IF** ⌊**condition = vraie**⌋ **THEN** ⌊**numéro de ligne.**⌋
C'est l'usage le plus courant de ce test, qui permet de renvoyer à une autre partie de programme, si par exemple X = Ø, ou si A = B, etc.

Les **opérateurs logiques NOT, OR** ou **AND** peuvent d'ailleurs être utilisés dans l'écriture de la condition.
On pourra aussi utiliser le test sous la forme :
IF ⌊**condition**⌋ **THEN** ⌊**instruction**⌋ ; l'instruction présenté à la suite de THEN peut être une affectation A = 12, une impression PRINT A, une entrée INPUT B $, ou même un autre test IF ___ THEN ___ .

On peut par exemple écrire :

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

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

On est très souvent conduit, en programmation, à vouloir exécuter un nombre entier de fois une certaine suite d'instructions (par exemple, faire avancer un mobile de la case 6 à la case 2Ø, etc.). Plutôt que d'utiliser un test et un renvoi, il sera préférable de faire appel à une ***boucle***, la variable de boucle prenant ses valeurs dans les entiers ou les décimaux, et le ***pas*** de cette variable (c'est-à-dire la quantité dont elle augmente ou diminue à chaque passage) pouvant aussi prendre ses valeurs dans les entiers ou les décimaux.

Ainsi, si l'on écrit :

```
FOR I = Ø TO 2Ø STEP 2
        Instructions
NEXT I
```

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

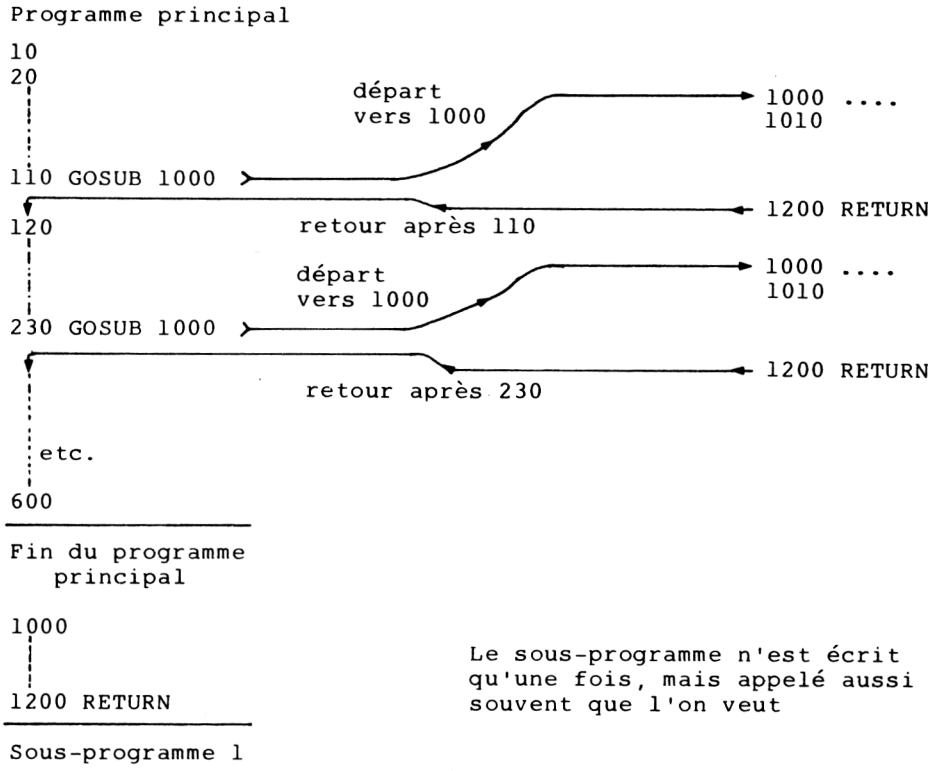
Si ***le pas de la boucle (STEP)*** n'est pas précisé, il sera automatiquement pris égal à 1.

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

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

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

Voici le schéma théorique de fonctionnement :



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 possibles, en recevant les seuls renseignements TROP GRAND ou TROP PETIT.

LE PROGRAMME : La ligne 4Ø permet de fabriquer le nombre à deviner.
Les entrées sont faites par les lignes 6Ø à 7Ø
La ligne 8Ø teste si le résultat est exact.
La ligne 9Ø signale un essai trop petit, et la ligne 1ØØ un essai trop grand.
Le compteur d'essais est en ligne 11Ø.
Les lignes 2ØØ à 23Ø s'occupent de l'affichage du résultat, et de la gestion de la partie suivante.

POUR JOUER : On lancera le programme par RUN.
On obtient un premier affichage NOMBRE DE CHIFFRES ? auquel on répond par un entier entre 1 et 7, suivi de 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 PRINT"□"
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"
100 IF M > X THEN PRINT"TROP GRAND"
110 J = J + 1
120 PRINT"□"
130 GOTO 60
200 PRINT"□"
210 PRINT"VOUS AVEZ TROUVE";X;" EN";J;" ESSAIS"
220 INPUT"AUTRE PARTIE ( O/N )";R$
230 IF R$ = "O" THEN RUN
```

```
NOMBRE DE CHIFFRES :? 3

ESSAI NO. 1
VOTRE NOMBRE :? 268
TROP PETIT
ESSAI NO. 2
VOTRE NOMBRE :? 579
TROP PETIT
ESSAI NO. 3
VOTRE NOMBRE :? 891
TROP GRAND
ESSAI NO. 4
```

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 (2Ø-15Ø), et la boucle I (16Ø-21Ø).
Le nom de l'objet représenté est affiché en ligne 23Ø (ici : CUBE).
La ligne 24Ø provoque l'arrêt du programme.

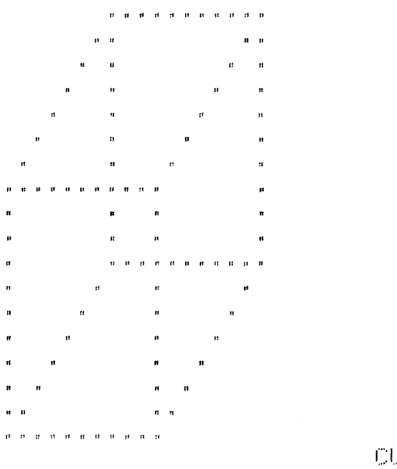
POUR JOUER : Il suffit de lancer l'exécution par RUN, et de regarder.

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.

```
READY.

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

READY.
```



CUBE

Chaud, froid

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

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

LE PROGRAMME : Les lignes 4Ø et 5Ø définissent la position de l'objet.
La ligne 9Ø calcule la distance, par rapport aux deux coordonnées que vous avez entrées aux lignes 6Ø-7Ø.
La ligne 1ØØ regarde si vous avez gagné.
Les lignes 12Ø-13Ø normalisent la distance (qui doit avoir un sens, quelles que soient les dimensions de la grille initiale).
La ligne 14Ø 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 : Si l'on dispose de la couleur, il sera intéressant de trouver un moyen plus visuel de symboliser la distance, par exemple des pavés allant du rouge au bleu, aux emplacements essayés.
On peut aussi travailler dans un espace à trois dimensions, cacher plusieurs objets, etc.

```
10 REM CHAUD,FROID
20 PRINT"■"
30 INPUT"DIMENSION DE LA GRILLE :";D
40 X = INT ( RND (1) * D + 1 )
50 Y = INT ( RND (1) * D + 1 )
60 INPUT"DONNEZ X :";A
70 INPUT"DONNEZ Y :";B
80 S = S + 1
90 T = ABS ( X - A ) + ABS ( Y - B )
100 IF T = 0 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 1000,2000,3000,4000,5000
150 PRINT"COMPLETEMENT GELE"
160 GOTO 60

1000 PRINT"A PEINE TIEDE"
1010 GOTO 60
2000 PRINT"TIEDE"
2010 GOTO 60
3000 PRINT"CHAUD"
3010 GOTO 60
4000 PRINT"TRES CHAUD"
4010 GOTO60
5000 PRINT"BRULANT"
5010 GOTO 60
6000 PRINT"■"
6010 PRINT"VOUS AVEZ TROUVE EN";S;" COUPS"
```

```
DIMENSION DE LA GRILLE :? 6
DONNEZ X :? 3
DONNEZ Y :? 4
VOUS ETES CHAUD
DONNEZ X :? 5
DONNEZ Y :? 6
VOUS ETES COMPLETEMENT GELE
DONNEZ X :? 2
DONNEZ Y :? 1
```

Les allumettes

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

LE PROGRAMME : Le jeu du joueur commence en ligne 7Ø.
Celui du Commodore commence en ligne 11Ø.
La ligne 8Ø contrôle la légalité du coup du joueur.
La ligne 9Ø gère le changement de joueur.
Le coup du Commodore est calculé dans une boucle non formelle, de la ligne 12Ø à la ligne 15Ø. Le résultat est affiché par la ligne 1Ø2Ø.
Si la partie est terminée, la ligne 1Ø3Ø le détecte et envoie en ligne 2ØØØ, pour le message de fin.

POUR JOUER : On commence l'exécution par RUN, et, à l'affichage du message NOMBRE TOTAL, on entre le nombre d'allumettes, suivi de RETURN. Puis on entre la prise maximum autorisée, suivie de RETURN. Si le joueur veut commencer, il frappe ensuite Ø ; sinon il frappe 1, puis RETURN.
Après chaque coup, le nombre d'allumettes restantes est affiché, et le Commodore affiche VOUS RETIREZ lorsque c'est au joueur de prendre.
Celui qui prend la dernière allumette a perdu.

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

```
10 REM LES ALLUMETTES
20 PRINT"□"
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 INPUT"VOUS RETIREZ :";P
80 IF P <>INT(P) OR P < 1 OR P > I OR P > N THEN PRINT"□":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
1000 PRINTP
1010 N = N - P
1020 PRINT"IL RESTE";N
1030 IF N = 0 THEN 2000
1040 GOTO 60
2000 PRINT"□"
2010 IF D = 1 THEN PRINT"J'AI GAGNE"
2020 IF D <>1 THEN PRINT"BRAVO"
```

```
NOMBRE TOTAL :? 36
PRISE MAXIMUM :? 4
QUI COMMENCE (VOUS:0 / MOI:1 )? 1
 1
IL RESTE 35
VOUS RETIREZ :? 4
 4
IL RESTE 31
 1
IL RESTE 30
VOUS RETIREZ :? 2
 2
IL RESTE 28
 2
IL RESTE 26
VOUS RETIREZ :? 2
 2
```

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

LE PROGRAMME : Les lignes 1Ø à 4Ø permettent les différentes entrées.
Les lignes 7Ø à 1ØØ réalisent les impressions.
La ligne 12Ø contrôle le coup de chaque joueur (par rapport au coup précédent).
La ligne 2Ø2 gère les tours (remise à 1 du numéro du joueur).
La ligne 17Ø réalise le test de fin, et la ligne 21Ø affiche le numéro du joueur gagnant (avec la règle prévue initialement).

POUR JOUER : Après avoir lancé le programme par RUN, on entre le nombre d'allumettes initial, suivi de RETURN puis le nombre de joueurs (et RETURN).
Le premier joueur ne pourra enlever qu'une allumette, le deuxième pourra en enlever 1 ou 2 ; le suivant 1, 2, ou 1, 2, 3 ou 4 selon le coup précédent, et ainsi de suite ; le retrait maximum autorisé est à chaque fois affiché par le Commodore.
Lorsque la dernière allumette a été prise, on obtient l'affichage LE JOUEUR... A GAGNE, et la partie est terminée.

EXTENSIONS POSSIBLES : On peut convenir que celui qui prend la dernière allumette est éliminé du jeu, et recommencer la partie entre les joueurs restants qui seront ainsi éliminés les uns après les autres.
On peut aussi imaginer une représentation graphique des tas d'allumettes, tirer au hasard le nombre à enlever en début de partie, etc.

```
READY.

10 REM ALLUMETTES A PLUSIEURS
20 PRINT"□"
30 INPUT"NOMBRE TOTAL D'ALLUMETTES :";N
40 INPUT"NOMBRE DE JOUEURS :";J
50 F = 1
60 K = 1
70 PRINT"□"
80 PRINT"TOUR DU JOUEUR :";K
90 PRINT"TOTAL RESTANT :";N
100 PRINT"MAXIMUM A ENLEVER :";F
110 INPUT"VOTRE COUP :";C
120 IF C = 0 OR C >F THEN 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 K = K + 1
170 IF K > J THEN K = 1
180 GOTO 70
200 PRINT"□"
210 PRINT"LE JOUEUR NUMERO";K;" A GAGNE"

READY.


TOUR DU JOUEUR : 2
TOTAL RESTANT : 30
MAXIMUM A ENLEVER : 10
VOTRE COUP :?5
```

Calendrier perpétuel

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

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

LE PROGRAMME : Les lignes 3Ø et 4Ø vous permettront d'entrer le mois et l'année.
Les lignes 5Ø, 6Ø et 7Ø calculent le numéro du premier jour du mois.
La ligne 8Ø calcule le nombre de jours du mois (il n'est pas tenu compte ici des 29 février, pour simplifier).
La boucle A (12Ø-15Ø) réalise l'impression du calendrier proprement dit.

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

```
READY.


10 REM CALENDRIER PERPETUEL
20 PRINT"■"
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
120 FOR A = 1 TO M
130 PRINTTAB(4*(D+A-1));A;
140 IF D+A-1=6 THENPRINT:D = D - 7
150 NEXT A


READY.


POUR QUELLE ANNEE :? 1984
POUR QUEL MOIS    :? 1

DIM LUN MAR MER JEU VEN SAM

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

Bataille navale

REFERENCES : Difficulté de programmation : facile.
Catégorie de jeu : reflexion.
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 Commodore répond en affichant la distance séparant le navire invisible de la case visée. Mais, si l'on tire trop près, le navire risque de se recacher au hasard.

LE PROGRAMME : La boucle I (3Ø-5Ø) dessine le terrain de jeu.
Les lignes 6Ø-7Ø définissent l'emplacement du navire.
Les coordonnées du tir sont entrées aux lignes 8Ø et 1ØØ, et la ligne 15Ø affiche le coup sur l'écran. La distance est calculée en ligne 16Ø, et la ligne 17Ø regarde si l'on a ou non fait mouche.
La ligne 22Ø intervient pour déplacer le navire si les coups sont trop rapprochés.
La ligne 23Ø permet l'affichage de la distance.

POUR JOUER : Le terrain de jeu est affiché après avoir frappé RUN. A l'affichage de LIGNE, on répond en donnant le numéro de la ligne horizontale sur laquelle on désire tirer. On tape ensuite 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 Commodore, mettre plusieurs navires, des sous-marins, etc.

```
READY.


10 REM BATAILLE NAVALE
20 PRINT"□";: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";H$:H=VAL(H$)+1
90 PRINT"□";"■■■■■■■■■■■■";
100 INPUT"COLONNE";V$:V=VAL(V$)
110 PRINT"■";
120 FOR J = 1 TO H
130 PRINT:PRINT
140 NEXT J
150 PRINTTAB(3*V+3)S
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"□                              PFFFF...
230 PRINTTAB(25)"□DISTANCE"D
    GOTO 240
240 PRINT"□□":GOTO 80
250 PRINT"□"
260 PRINT"TROUVE EN"S;" COUPS"

READY.
```

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

 0 -.  .  .  .  .  .  .  .  .  .

 1 -.  .  .  .  .  .  .  .  .  .

 2 -.  .  .  .  .  .  .  2  .  .

 3 -.  .  .  .  .  .  .  .  .  .

 4 -.  .  .  .  .  .  .  .  .  .

 5 -.  .  .  .  .  1  .  .  .  .

 6 -.  .  .  .  .  .  .  .  .  .

 7 -.  .  .  .  .  .  .  .  .  .

 8 -.  .  .  .  .  .  .  3  .  .

 9 -.  .  .  .  .  .  .  .  .  .

LIGNE? 8    COLONNE? 7   DISTANCE 6
```

Golf

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

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

LE PROGRAMME : La ligne 4Ø permet de définir la position du trou, et la ligne 5Ø dessine le terrain de jeu.
Le déplacement à réaliser est calculé en ligne 8Ø.
La boucle I (1ØØ-13Ø) réalise le déplacement de la balle.
La ligne 14Ø regarde si le coup est, ou non, gagnant ; s'il l'est, on recommence (ligne 22Ø) jusqu'à ce que l'on ait réalisé 18 trous (boucle S (2Ø-22Ø). ; fin de programme en 24Ø, avec affichage de fin.

POUR JOUER : On obtient l'affichage du terrain en frappant RUN ainsi que le dessin de la balle sur sa position de départ, et le texte : TROU 1. FORCE; on doit alors répondre en donnant la force désirée (un nombre entre 1 et 3ØØ...). Dès que l'on aura tapé 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...

```
READY.


10 REM GOLF
20 FOR S = 1 TO 18
30 PRINT"♥";
35 X = 1
40 T = INT(29*RND(1)+9)
50 PRINT"SQQQQ——————————————————————————————————————"
55 PRINT"SQQQQ"TAB(T);" ":PRINT"SQQQ"TAB(X);"●"
60 PRINT"SQQQQQQQ"TAB(15);"                    "
65 PRINT"SQQQQQQQTROU"S;:INPUT"    FORCE";R$
70 F = VAL(R$)
80 F = INT(F*0.16-(ABS(F)*(SQR(RND(1))+0.1)↑2)/1000)
90 IF F = 0 THEN PRINT"♥":GOTO 65
100 FOR I = SGN(F) TO F STEP SGN(F)
110 X = X + SGN(F)
120 IFX<39ANDX>=1THENPRINT"SQQQ"TAB(X-SGN(F));" ";:
    PRINT"SQQQ"TAB(X)"●"
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)"♥ ":PRINTTAB(X)"●"
210 FOR P = 1 TO 300: NEXT P
220 NEXT S
230 PRINT"♥"
240 PRINT"18 TROUS EN";E;" COUPS"

READY.
```



```
TROU 1     FORCE? 50
```

Solfège

REFERENCES : Difficulté de programmation : moyenne.
Catégorie de jeu : reflexion.
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.

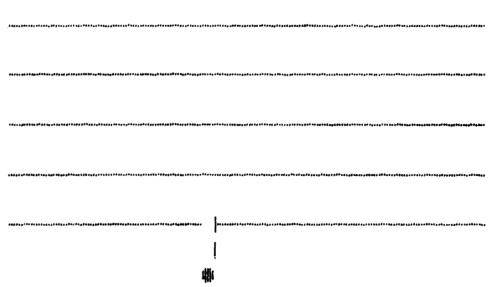
POUR JOUER : Après avoir lancé le programme par RUN, la portée sera affichée, ainsi que la première note.
Vous répondrez à la question NOTE ? en donnant le nom en clair de la note (do, ré, mi, fa, sol, la ou si), suivi de RETURN.
Le score vous indiquera si c'est exact ou non, et une nouvelle note sera 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.

```
READY.

10 REM SOLFEGE
20 S = 0 : P = 0
30 PRINT"▯▯▯▯"
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)"▩ |"
110 PRINTTAB(20)"▩ |"
120 PRINTTAB(20)"▩●"
130 PRINT"▩▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯SCORE"S;" SUR"P;" ESSAIS"
140 INPUT"▯CLE DE SOL ; NOTE ";R$
150 IF R$ = "0" THEN END
160 Y = ( 1 AND R$="FA" ) + (2 AND R$="MI" ) + ( 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

READY.
```



```
SCORE 3  SUR 4  ESSAIS

CLE DE SOL ; NOTE ? DO
```

Le jeu du 21

REFERENCES : Difficulté de programmation : facile.
Catégorie de jeu : jeu contre le Commodore.
Difficulté du jeu : facile.

LE JEU : On devra entrer une mise au début du jeu, et on pourra lancer les dés en tapant 1 ; à chaque lancer, on obtiendra un nouveau total, qui augmentera peu à peu, et se rapprochera de 21. On devra essayer de se rapprocher le plus possible de 21, sans dépasser ce nombre. Lorsque l'on décidera de s'arrêter, il suffira de taper Ø pour laisser jouer le Commodore qui essaiera alors d'avoir un meilleur score. A la fin de chaque partie, le gain sera affiché ainsi que le total des gains (ou des pertes).

LE PROGRAMME : Les lignes 5Ø, 6Ø et 7Ø réalisent le contrôle de celui qui doit jouer (joueur ou Commodore)
Le sous-programme (1ØØØ-1Ø1Ø) lance les dés.
Les lignes 9Ø et 1ØØ calculent le total fait par le joueur et l'affichent.
Le total du Commodore est calculé en ligne 18Ø et affiché par la ligne 19Ø.
La ligne 21Ø contrôle la fin du jeu pour le Commodore.
Les lignes 25Ø-26Ø affichent les résultats, et la ligne 27Ø cherche quel est le gagnant.
Le gain (ou la perte) sera affiché par la ligne 3ØØ, et le total par la ligne 32Ø.
La ligne 34Ø contrôle la fin de partie.

POUR JOUER : Taper RUN pour lancer le jeu.
A la question VOTRE MISE, entrer la mise initiale, suivie de 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 Ø et RETURN pour laisser jouer le Commodore (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.

```
READY.

10 REM JEU DU 21
15 G = 0
20 PRINT"■"
30 T = 0
40 INPUT"VOTRE MISE";M
50 PRINT"■■■■FAIRE 1 POUR TIRER , 0 POUR FINIR"
60 INPUT R
70 IF R = 0 THEN 150
80 GOSUB 1000
90 T = T + X
100 PRINT"■■■■■■VOTRE TOTAL EST DE"T
110 GOTO 50
150 PRINT"■LA BANQUE JOUE"
160 S = 0
170 GOSUB 1000
180 S = S + X
190 PRINT"■■■■■TOTAL DE LA BANQUE"S
200 FOR I = 1 TO 600:NEXT I
210 IF S > T OR S > 17 THEN 250
220 GOTO 170
250 PRINT"■■■■VOUS AVEZ"T
260 PRINT"■■LA BANQUE A"S
270 IF S > 21 OR ( S<T AND T<=21 ) THEN 300
280 M = - M
300 PRINT"■■GAIN"M
310 G = G + M
320 PRINT"■■VOTRE TOTAL ACTUEL"G
330 INPUT"■■■ON CONTINUE ( O/N )";R$
340 IF R$ = "N" THEN END
350 GOTO 20
1000 X = INT( RND(1)*6 + 1)
1010 RETURN

READY.


VOTRE MISE? 100

FAIRE 1 POUR TIRER , 0 POUR FINIR
? 0
VOTRE TOTAL EST DE 19
```

Pair - impair

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

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

LE PROGRAMME : Les dés sont tirés par un sous-programme placé aux lignes 1ØØØ et 1ØlØ.
La ligne 6Ø tient le compte du nombre de parties, et les lignes 15Ø, 2ØØ et 23Ø comptabilisent les points marqués.
La parité du total est déterminée par les lignes lØØ et llØ (la ligne llØ compare avec la réponse du joueur).
La ligne 41Ø réalise l'affichage de fin de partie : nombre de points, nombre de parties jouées ;
La ligne 43Ø provoque l'arrêt si l'on répond N à la question ON CONTINUE.

POUR JOUER : On lance le programme par l'instruction RUN. Il suffit ensuite de répondre aux diverses questions posées, en suivant scrupuleusement les indications données et en tapant 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 lØ.
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.)

```
READY.

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 PRINT"■"
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"■■■■■FAUX "
130 GOTO 160
140 PRINT"■■■■■EXACT"
150 T = T + 1
160 INPUT"■■TOTAL PRESUME";N
170 IF N = S THEN 200
180 PRINT"■■PERDU                "
185 FOR I = 1 TO 400:NEXT I
190 GOTO 400
200 T = T + 2
210 INPUT"■■■■■■■■■■■■■■BRAVO ; DONNEZ UN DE";N
220 IF N <> A AND N<>S-A THEN 180
230 T = T + 3
240 PRINT"■■BRAVO":GOTO 185
400 PRINT"■"
410 PRINT"VOUS AVEZ UN TOTAL DE"T;" SUR"R
420 INPUT"■■ON CONTINUE ( O/N )";R$
430 IF R$ ="N" THEN END
440 GOTO 30
1000 X = INT(RND(1)*6 + 1)
1010 RETURN

READY.


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


EXACT

TOTAL PRESUME? 7


BRAVO ; DONNE UN DE? 3

BRAVO
```

Devine un chiffre

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

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

LE PROGRAMME : Le tirage des dés est réalisé par un sous-programme placé en lignes 1ØØØ et 1Ø1Ø.
L'affichage est réalisé par les lignes 12Ø et 13Ø.
Les lignes 15Ø à 18Ø calculent le gain dans les diverses configurations possibles.
La totalisation est effectuée en ligne 2ØØ pour les gains, et en ligne 51Ø pour les pertes.
La variable R$, testée en ligne 53Ø détermine la poursuite éventuelle du jeu.

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

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

```
READY.

10 REM DEVINE UN CHIFFRE
20 G = 0
30 PRINT"■"
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"■■ON CONTINUE( O/N )";R$
530 IF R$ ="N" THEN END
540 GOTO 30
1000 X = INT(RND(1)*6 + 1)
1010 RETURN

READY.



VOTRE MISE? 100


DONNEZ UN CHIFFRE DE 1 A 6? 4

    A          B          C

    4          6          4


VOUS GAGNEZ 200

TOTAL ACTUEL 0

ON CONTINUE( O/N )? N
```

Le jeu de Wytthof

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

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

LE PROGRAMME : Les lignes 1Ø, 2Ø et 3Ø permettent de définir une position de départ, en dehors de la diagonale.
La ligne 4Ø et la boucle I (5Ø-7Ø) dessinent le terrain de jeu.
La ligne 9Ø imprime la position du point mobile.
Le test de fin est effectué en ligne 1ØØ.
Le changement de joueur est assuré par la ligne 14Ø.
Les directions de déplacement disponibles, 1, 2 et 3, sont décodées par trois sous-programmes qui commencent aux lignes 1ØØØ, 2ØØØ et 3ØØØ.

POUR JOUER : Taper RUN pour lancer l'exécution du programme
Le terrain de jeu s'affiche, ainsi que la position de départ.
Puis on obtient l'affichage JOUEUR 1 et en-dessous la question DIRECTION (1,2,3), à laquelle on répond en tapant l'un de ces trois nombres, suivi de RETURN.
C'est ensuite au tour du joueur 2.
Le jeu se termine lorsque l'un des joueurs arrive en case (9,9).

EXTENSION 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 Commodore de jouer, etc.

```
READY.


10 X = 3*INT(RND(1)*5)+4
20 Y = INT(RND(1)*5)
30 IF X = Y THEN 20
40 PRINT"■    0  1  2  3  4  5  6  7  8  9"
50 FOR I = 0 TO 9
60 PRINT"■";I;" .  .  .  .  .  .  .  .  .  ."
70 NEXTI
75 H = 1
80 GOSUB 500
90 PRINTTAB(X)"+"
100 IF X = 31 AND Y = 10 THEN END
110 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■JOUEUR"H;:
    INPUT"■■DIRECTION (1,2,3)";D$
120 D = VAL(D$)
130 ON D GOSUB 1000,2000,3000
140 H = 2 - ABS(H-1)
150 GOTO 80
500 PRINT"■";
510 FOR I = 1 TO Y
520 PRINT:PRINT
530 NEXT I
540 RETURN
1000 X = X + 3
1010 RETURN
2000 GOSUB 1000
3000 Y = Y + 1
3010 RETURN


READY.


    0  1  2  3  4  5  6  7  8  9

 0  .  +  +  +  .  .  .  .  .  .

 1  .  .  .  +  .  .  .  .  .  .

 2  .  .  .  .  +  .  .  .  .  .

 3  .  .  .  .  .  +  .  .  .  .

 4  .  .  .  .  .  +  +  .  .  .

 5  .  .  .  .  .  .  .  +  .  .

 6  .  .  .  .  .  .  .  +  .  .

 7  .  .  .  .  .  .  .  .  .  .

 8  .  .  .  .  .  .  .  .  .  .

 9  .  .  .  .  .  .  .  .  .  .

JOUEUR 2   DIRECTION (1,2,3)? 0
```

Rebonds

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

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

LE PROGRAMME : La ligne 2Ø imprime les chiffres et les lettres en haut de l'écran.
Les lignes 3Ø et 4Ø définissent les positions initiales de la balle et de la cible.
Les lignes 7Ø et 1ØØ effectuent leur impression.
Le symbole correspondant à la direction choisie est entré en ligne 11Ø.
La boucle J (14Ø-19Ø) réalise le déplacement de la balle.
La ligne 18Ø calcule la nouvelle position.
Le changement de direction, lors de la rencontre d'un bord, est effectué par la ligne 16Ø., par renvoi dans une partie du programme indépendante, située aux lignes 25Ø à 27Ø.
La ligne 2ØØ effectue l'impression de fin de partie.
Le nombre de rebonds est compté en ligne 26Ø.

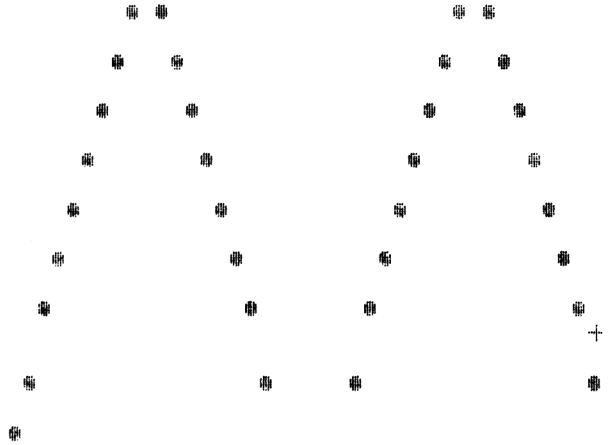
POUR JOUER : On lance le programme par RUN.
Le terrain de jeu est affiché, ainsi que les positions initiales de la balle et de la cible.
On choisit l'endroit où l'on désire faire rebondir la balle (de 1 à Z), et on l'entre suivi de RETURN.
La trajectoire est alors affichée, suivie s'il y a lieu de la 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, sur un appareil disposant d'un graphisme plus fin, de dessiner un véritable billard, avec son tapis vert, et d'indiquer la direction par un segment, symbolisant la queue de billard, se déplaçant autour de la bille.
On peut aussi ajouter des trous, mettre plusieurs billes, sonoriser les rebonds, ou les heurts avec d'autres balles, les comptabiliser pour faire une partie à plusieurs joueurs, etc.

```
10 REM REBONDS
20 PRINT"■ 0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"
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"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■VOTRE COUP";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"))
160 IF T >=22 OR T <=0 THEN 250
170 Z = T : GOSUB 500
180 PRINTTAB(J)"●"
190 NEXT J
200 IF ABS(T-Y) < 3 THEN PRINT"■■BRAVO :"S;" REBONDS"
210 PRINT"■■■APPUYER SUR - SHIFT -"
220 WAIT653,1
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


READY.


 0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
BRAVO : 3  REBONDS
APPUYER SUR - SHIFT -
```



```
●
VOTRE COUP? 9
```

Jackpot

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

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

LE PROGRAMME : La boucle I (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.

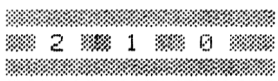
POUR JOUER : Lancer le programme par RUN ; à l'affichage de la question POT DE DEBUT, on répond en entrant un nombre suffisamment élevé, sur lequels vos mises seront décomptées.
Puis on tape RETURN, et on entre une première mise (différente de 0), suivie de RETURN.
Les trois chiffres tirés sont alors affichés un bref instant, ainsi que la mise faite, et le résultat obtenu (gain ou perte).
On peut ensuite miser de nouveau,faire RETURN et ainsi de suite.
Pour terminer, il suffit de répondre 0 à la question VOTRE MISE.
Le restant de votre pot initial sera alors affiché.

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

```
READY.

5 REM JACKPOT
7 PRINT"■"
10 INPUT"POT DE DEBUT";P
30 PRINT"■■■■■"
40 FOR I = 1 TO 3
50 PRINTTAB(12)"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
60 NEXT I
70 PRINT"■■■■■■■";TAB(14);"   ";TAB(19);"   ";TAB(24);"   "
80 INPUT "■■■■■■■■■■■VOTRE MISE";M
90 IF M = 0 THEN 2000
100 GOSUB 1000
110 A = X
120 GOSUB 1000
130 B = X
140 GOSUB 1000
150 PRINT"■■■■■■■";TAB(14);A;TAB(19);B;TAB(24);C
160 IF A = B AND A = X THEN 300
170 N = - M
180 PRINT"■■■■■■■■■■■MISE   "M;"    RESULTAT "N;"   "
190 PRINT"■■APPUYER SUR - SHIFT -":WAIT 653,1
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)
1010 RETURN
2000 PRINT"■"
2010 PRINT"VOUS AVEZ"P
2020 PRINT"A BIENTOT"

READY.
```




```
MISE    10?    RESULTAT -10

APPUYER SUR - SHIFT -
```

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 rond noir, se déplace sur un terrain de jeu comportant des cases numérotées de 1 à 36, ainsi que trois cases supplémentaires, le ØØ, une case P (comme pair) et une case I (comme impair).
La boule parcourt au moins une fois chaque case, puis s'arrête au deuxième tour ; on peut ensuite la relancer pour une autre partie.

LE PROGRAMME : La boucle K (2Ø-7Ø) dessine le terrain de jeu, auquel la ligne 8Ø rajoute les trois symboles P, ØØ et I.
La boucle K (95-16Ø) réalise le déplacement de la boule, l'arrêt étant contrôlé par les lignes 115 et 125.
La ligne 13Ø réalise un effacement particulier pour le cas où la boule passe sur les symboles P, ØØ et I.
La ligne 15Ø fait descendre de trois lignes.
La ligne 17Ø tire la position arrêt.

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

```
READY.


10 REM ROULETTE
15 PRINT"■"
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"■■■■■■■■■■■■■■■■P":PRINT:PRINT:PRINT:PRINT"00":PRINT:
   PRINT:PRINT:PRINT"I"
90 PRINT"■■■■■■■■■■"
95 FOR K = 1 TO 4
100 FOR I = 1 TO 10
110 PRINTTAB(4*I+1)"●";
115 X=X-1
120 GOSUB 200
125 IF X = 1 THEN WAIT653,1:RUN
130 PRINT"■■■  ";
140 NEXT I
150 PRINT"■■■■";
160 NEXT K
170 X = INT(38*RND(1))
190 PRINT"■■■■■■■■":GOTO 90
200 FOR J = 1 TO 200:NEXT J
210 RETURN

READY.


     2   4   6   8   10  12  14  16  18

P

     20  22  24  26  28  30  32  34  36

00

     1   3   5   7   9   11  13  15  17

I

     19  21  23  25  27  29  31  33  35
              ●
```

Couloirs

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

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

LE PROGRAMME : Les lignes 3Ø à 7Ø dessinent les couloirs (et les trois tourniquets).
La ligne 1ØØ attend votre choix.
Le décodage du chemin est fait par le Commodore en utilisant trois variables logiques I, J et K, représentant les états des trois tourniquets (lignes 13Ø à 27Ø).

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

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.).

```
READY.

10 REM COULOIRS
20 PRINT"■"
30 PRINT"■    ──────────────────────────────"
40 PRINT"■    B       ──────────      ×   2"
50 PRINT"■    ──────────  ×   ──────────"
60 PRINT"■    A       ──────────      ×   1"
70 PRINT"■    ──────────────────────────────"
80 I = 0 : J = 0 : K = 0
90 R$ = "" : T$= ""
100 INPUT"■■■■■■■■■■■■■■■■■■■■ENTREE 1 OU 2 :";C$
110 IF C$ ="" OR C$ >= "3" OR LEN(C$) <> 1 THEN 100
120 R$ = R$ + C$
130 PRINT"■";R$
140 IF C$ ="1" THEN 190
150 J = NOT(J)
160 IF J <>0 THEN 250
170 S$ = "B"
180 GOTO 220
190 I = NOT(I)
200 IF I <>0 THEN 250
210 S$ = "A"
220 T$ = T$ + S$
230 PRINT"■■■■■■■■■■■■■■■■■■■■■■SORTIE :";S$:PRINT"■";T$
240 GOTO 100
250 K = NOT(K)
260 IF K <>0 THEN 170
270 GOTO 210

READY.
```

```
    ──────────────────────────────
    B       ──────────      ×   2
    ──────────  ×   ──────────
    A       ──────────      ×   1
    ──────────────────────────────

ENTREE 1 OU 2 :? 1

122211211111

SORTIE A

BABBAABABAA
```

Echiquier

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

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

LE PROGRAMME : Les lignes 3Ø, 4Ø et 5Ø définissent trois variables A$, B$ et C$ qui contiennent les éléments de dessin. La ligne 6Ø imprime les lettres horizontales et la première ligne du dessin.
La boucle I (8Ø-16Ø) permet l'impression, le motif à imprimer étant affiché ou non selon la valeur de (I+J)/2 ; la ligne 11Ø réalise l'impression.

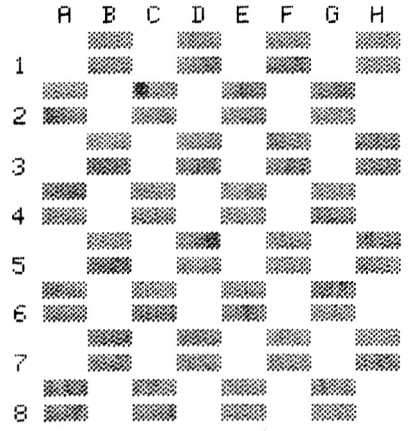
POUR JOUER : L'ordre RUN provoque l'affichage du damier sur l'écran.
Pour utiliser le programme, il conviendra de tenir compte du repérage des cases utilisables qui sont les centres des carrés noirs ou gris. Les cases sont obtenues en largeur par une tabulation de la forme 3*J + 1, J variant de 1 à 8, et en hauteur par 2*I + 1.

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.).

```
READY.

10 REM ECHIQUIER
20 PRINT"■"
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 PRINTTAB(3*J);A$:PRINTI;TAB(3*J);B$:PRINTTAB(3*J);C$;
130 PRINT"■■■"
140 NEXT J
150 PRINT"■"
160 NEXT I

READY.
```



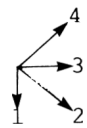
Le jeu de la fève

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

LE JEU : Présentant une certaine analogie avec le jeu de WYTTHOF, cette version offre en fait 4 possibilités de déplacement ; la position de départ est ici fixée en (Ø, Ø), et pourrait éventuellement être modifiée.
Les directions de déplacement disponibles sont : droite, bas, diagonale bas et droite, diagonale haut et droite.
On joue en principe avec la règle : le premier arrivé en (9,9) a perdu.

LE PROGRAMME : La ligne 2Ø et la boucle I (3Ø-5Ø) dessinent le terrain de jeu.
La ligne 1ØØ réalise le test de fin de partie.
La ligne 12Ø permet d'exécuter l'un des sous-programmes de déplacement.
Les lignes 13Ø et 14Ø assurent le changement de joueur.

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



Le joueur 1 entre son déplacement, et tape 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 piéges (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.

```
READY.


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


READY.


     0  1  2  3  4  5  6  7  8  9
 0   *  .  .  .  .  .  .  .  .  .
 1   *  .  .  .  .  .  .  .  .  .
 2   .  *  .  *  .  .  .  .  .  .
 3   .  .  *  .  *  *  .  .  .  .
 4   .  .  .  .  .  *  .  .  .  .
 5   .  .  .  .  .  *  .  *  .  .
 6   .  .  .  .  .  .  *  .  .  .
 7   .  .  .  .  .  .  .  .  .  .
 8   .  .  .  .  .  .  .  .  .  .
 9   .  .  .  .  .  .  .  .  .  .
JOUEUR 1  DIRECTION (1,2,3,4)? 0
```

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

LE PROGRAMME : La ligne 6Ø permet d'appeler le sous-programme 1ØØØ à 11ØØ) qui réalise l'affichage du terrain de jeu, le contrôle de la position des joueurs et l'affichage de cette position. Cet affichage sera d'ailleurs refait en cours de partie, pour tenir compte du déplacement réellement effectué (qui n'est pas nécessairement celui indiqué par le dé).
La ligne 7Ø s'occupe du changement de joueur, et la ligne 8Ø effectue le lancer de dé.
La ligne 9Ø traite le cas où l'un des joueurs termine.
La ligne 13Ø tire un nombre aléatoire entre Ø et 3, qui servira s'il y a lieu pour déterminer le vrai déplacement du joueur, en envoyant dans l'un des sous-programmes de "chance".
Les lignes 1ØØØ et 1Ø1Ø font avancer les pions, et les lignes 1Ø15 à 1Ø8Ø redessinent à chaque affichage le terrain de jeu en entier.

POUR JOUER : On lancera le jeu avec RUN. Le terrain de jeu est alors dessiné, puis on voit apparaître l'affichage JOUEUR 1 : ..
On devra alors appuyer sur une touche quelconque pour que le coup du joueur numéro 1 soit pris en compte ; ne pas s'inquiéter du temps de réaction, qui pourra sembler un peu long.
Le déplacement indiqué par les dés sera exécuté, puis au bout de 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 RESTORE.

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.

```
READY.

10 REM JEU DE L'OIE ALEATOIRE
15 PRINT"■"
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"■■■■■■!■■■BRAVO":END
100 PRINT"■■■■■■■■JOUEUR";J+1;"   :";I
110 PRINT"APPUYER SUR - SHIFT -":WAIT 653,1
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 PRINT"■■■■■0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"
1020 FOR I = 0 TO 35
1030 PRINTTAB(I)"■■"
1040 PRINTTAB(I)"■■"
1050 PRINT"III"
1060 NEXT I
1070 PRINTTAB(X)"1"
1080 PRINTTAB(Y)"2"
1090 FOR I = 1 TO 1000 : NEXT I
1100 RETURN

READY.
```

```
0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒1▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒2▒▒▒▒▒▒▒▒▒▒▒▒▒

JOUEUR 1    : 5
APPUYER SUR - SHIFT -
```

NIVEAU 2

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

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

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

Les structures de tableaux sont très couramment utilisées en programmation, se révélant même indispensables dans un grand nombre de cas. Si vous voulez, par exemple, entrer vos recettes et dépenses pour tous les jours du mois (on supposera pour simplifier des mois de trente jours) vous pourriez définir 30 variables appelées PREMIER JOUR, DEUXIEME JOUR, etc. Il est plus facile, et plus rationnel d'utiliser une boucle I, I variant de 1 à 3Ø, et une variable R (I) pour les recettes, et D (I) pour les dépenses ; ce faisant, vous avez, sans le savoir, utilisé une structure de tableau à une dimension ; ceci ne pourra être admis par le Commodore que si vous l'avez auparavant prévenu que vous alliez utiliser un tel tableau, en donnant une instruction DIM R (3Ø) pour les recettes, et DIM D (3Ø) pour les dépenses.

On pourrait aussi imaginer de conserver dépenses et recettes dans un même tableau, en lui donnant cette fois deux dimensions : par exemple DIM RD (3Ø,2) (ou DIM RD (2,3Ø)). Ainsi, avec DIM RD (3Ø,2) on pourra convenir que RD (I,1) représente les recettes, et RD (I,2) les dépenses. Des tableaux de plus de 2 dimensions sont également possibles.

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

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

Sur le Commodore 64, il n'est pas obligatoire de dimensionner un tableau qui contient moins de 10 éléments.

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

Si l'on a un tableau à une dimension, numérique ou alphanumérique, défini par une instruction comme DIM R (3Ø) ou DIM N$ (2Ø), le Ième élément de ce tableau est repéré par R (I) (ou N$ (I)) ; il sera alors manipulé exactement comme une variable ordinaire (affectation, calcul, tests, etc.).

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

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

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

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

LE TRAITEMENT DE DONNEES MULTIPLES
INSTRUCTIONS READ, DATA, RESTORE

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

Ainsi, on pourra écrire :

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

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

Dé truqué

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

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

LE PROGRAMME : Il utilise le générateur aléatoire du Commodore supposé de qualité convenable (on pourra aisément utiliser tout autre générateur, si besoin était). La boucle I (4Ø-1ØØ) s'occupe de l'entrée de chacune des probabilités, avec vérification à chaque entrée et sur le total (ligne 7Ø) ; chaque probabilité est conservée dans une variable indicée E (I) : ligne 5Ø du programme.
Le lancer du dé truqué est effectué à l'aide d'une autre boucle I (14Ø-16Ø), la ligne permettant le trucage se trouvant en 15Ø.
Un nouveau tirage pourra être effectué sur pression de - SHIFT - (ligne 18Ø-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 Ø et 1. Il conviendra de ne pas oublier que le total de toutes les probabilités entrées ne devra en aucun cas dépasser 1.1 (tolérance de Ø.1).
Des messages d'erreurs pourront être affichés et le programme sera éventuellement relancé.
Lorsque toutes les probabilités auront été définies, le premier tirage sera automatiquement effectué, et il suffira de presser la touche SHIFT pour effectuer les tirages suivants. On arrête le programme en tapant RUN/STOP et RESTORE.

EXTENSIONS POSSIBLES : Il sera bien 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 correspondants à une série de lancers de dés sous une forme spectaculaire, qui mette le trucage en évidence...

```
READY.

10 REM DE TRUQUE
15 PRINT"■"
20 P = 0
30 DIM E(6)
40 FOR I = 1 TO 6
45 PRINT"■■■■";TAB(17);"              "
50 PRINT"■■■■PROBABILITE DU DE"I;:INPUTE(I)
60 P = P + E(I)
70 IF E(I) < 0 OR E(I) > 1 OR P > 1.1 THEN 200
80 PRINT"■■■■■■■■";TAB(6*I-6);"DE"I;
90 PRINT"■■■■■■■■■■";TAB(6*I-6);E(I)
100 NEXT I
110 IF P <0.9 THEN RUN
120 FOR I = 1 TO 500 : NEXT I
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"■■■■■■■■■■■■TIRAGE :"N;"      "
175 FOR J = 1 TO 2000 : NEXT J
178 PRINT"■"TAB(8);"      "
180 PRINT"■FAIRE - SHIFT - POUR UN AUTRE TIRAGE"
185 WAIT 653,1
190 GOTO 130
200 PRINT"IMPOSSIBLE : 0 <= P <= 1"
210 P = P - E(I)
220 FOR J = 1 TO 2000: NEXT J
230 PRINT"■                                    "
240 GOTO 50

READY.
```

```
PROBABILITE DU DE 6 ? 0.15

DE 1   DE 2   DE 3   DE 4   DE 5   DE 6

 .2     .1     0      .35    .15    .15

TIRAGE : 1

FAIRE - SHIFT - POUR UN AUTRE TIRAGE
```

Visions

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

LE JEU : Le Commodore 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 Commodore dès que l'on aura frappé RETURN.
Pour recommencer, il suffira de frapper une touche quelconque, et RUN/STOP et RESTORE pour finir.
Je vous laisse savourer le dessin qu'a réalisé Commodore lorsqu'il a connu le prénom de l'auteur de ce livre.

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

```
READY.

10 REM VISIONS
20 PRINT"■"
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)
90 IF S <> 0 THEN 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 WAIT653,1
150 GOTO 20

READY.



QUEL DESSIN? JACQUES

                  ▒▒▒   ▒ ▒
                ▒ ▒ ▒▒▒▒▒
                ▒▒▒▒      ▒
                ▒ ▒       ▒▒
                ▒   ▒ ▒   ▒▒
                ▒   ▒▒▒     ▒
                ▒▒▒▒ ▒    ▒▒

APPUYER SUR - SHIFT -
```

421

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

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

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

POUR JOUER : On lance l'exécution du programme par RUN, ce qui conduit à l'affichage NOMBRE DE JOUEURS.
On entre le nombre de joueurs, et l'on tape 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 procèderont de la même façon.

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

```
10 REM 421
20 INPUT"NOMBRE DE JOUEURS";N
25 P = 1
30 I = 1
40 K = 0
50 PRINT"■"
60 R$="ABC"
70 GOTO 170
80 K = K + 1
90 PRINT"S■■■■■■■■■■■■■■■■421"
100 PRINT"                  ■■■■"
110 PRINT"■■■■JOUEUR NO."I;" ESSAI"K;" PARTIE"P
120 PRINT"■■■■■      "A;"          "B;"          ";C
130 PRINT"■■■       A            B            C"
135 R$="":PRINT"■■■             "
140 INPUT"■ABC";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 )
4010 RETURN
READY.
```

```
              421
              000


JOUEUR NO. 1  ESSAI 1  PARTIE 1


       1          3          5


       A          B          C


ABC? BC
```

L'électricien fou

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

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

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

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

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

```
READY.

10 REM L'ELECTRICIEN FOU
20 M = 0
30 DIM A(10)
40 PRINT"■"
50 FOR I = 1 TO 7
60 PRINTTAB(5)"■■                              "
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);"■■ "
140 IF A(I)=0 THEN PRINT"■■■■";TAB(2*I+5);"▒"
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"

READY.


                ▒   ▒   ▒ ▒

       0 1 2 3 4 5 6 7 8 9


INTERRUPTEUR NUMERO? 5
```

La course au 20

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

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

LE PROGRAMME : La boucle I (3Ø-6Ø) dessine 4 rangées de 20 étoiles.
La variable F, dimensionnée en ligne 8Ø, servira à enregistrer le nombre d'étoiles prises dans chaque rangée.
La variable T contiendra le nombre total d'étoiles enlevées.
La ligne 12Ø va contrôler qu'il reste suffisamment d'étoiles sur la ligne indiquée.
Les boucles I (15Ø-2ØØ) et J (16Ø-18Ø) suppriment les étoiles ; la ligne 23Ø 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 graphique et des contrôles, de nombreuses variantes sont envisageables, en modifiant si besoin est les règles de départ : on peut autoriser l'enlèvement de seulement un nombre impair d'étoiles, ou alterner pair et impair, ou tirer au hasard le nombre à enlever, ou encore jouer à plusieurs...

```
READY.

10 REM LA COURSE AU 20
15 PRINT"■"
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 PRINT"■■■■■■■■■■■■■■■JOUEUR"B
110 INPUT"■LIGNE";L:PRINT"                "
115 INPUT"■CASES";C
120 IF C + F(L) > 20 THEN 100
130 F(L) = F(L) + C
140 T = T + C
145 PRINT"■■■"
150 FOR I = 1 TO 4
160 FOR J = 0 TO F(I)
170 PRINTTAB(J + 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"■■■■■■LE JOUEUR";B;" GAGNE"

READY.



 1        **********

 2       **********

 3    **********

 4          **********


JOUEUR 2

LIGNE? 4
CASES? 5
```

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$ (I) selon le code retenu (voir plus loin).
Les lignes 40 à 120 contiennent le dessin codé.
La boucle I (130-200) assure le passage d'un 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.

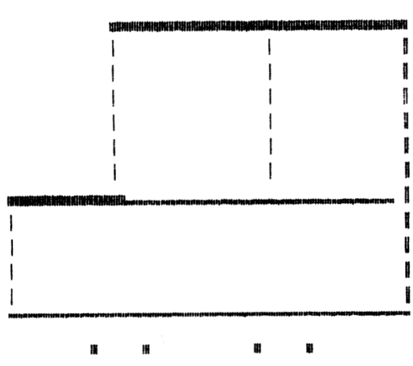
POUR JOUER : Le dessin devra auparavant être défini dans des variables A$ (I), en utilisant les caractères graphiques, et en donnant la position d'impression de chaque caractère, par rapport au début de la ligne, avec un chiffre de 1 à 9, ou une lettre (A : 10 ; B : 11 ; C : 12 ; etc.).
Le passage à 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 devrait essayer de coder un dessin donné le plus rapidement possible, etc.

```
READY.

10 REM DESSIN
20 PRINT"□"
30 DIM A$(25)
40 A$(10)="A▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬"
50 A$(11)="A▌K IT ▌"
60 A$(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$(18)="3▌T ▌"
90 A$(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"▒"

READY.
```



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 (3Ø-6Ø) dessine la portée.
Une deuxième boucle I (7Ø-11Ø) affiche les numéros des lignes.
La ligne 12Ø définit le nom des notes, et la ligne 13Ø réalise le tirage aléatoire d'une note.
Le nom de la note est calculé à la ligne 14Ø et il est affiché par la ligne 16Ø, suivi de la mention GRAVE ou AIGUE, selon le cas.
La réponse sera entrée en ligne 19Ø.
La ligne 2ØØ s'occupe du décodage de la réponse.
La ligne 22Ø affiche la note.
Les lignes 25Ø et 26Ø regardent si l'on a gagné ou perdu et affichent le message correspondant.
Les lignes 27Ø à 29Ø permettent de faire un nouvel essai.

POUR JOUER : On lance le programme par RUN.
On obtient alors l'affichage de la portée, ainsi que le nom de la note à placer et la question 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 introduire de la couleur, des sons, etc.

```
10 REM MUSIQUE
20 PRINT"▯"
30 FOR I = 1TO 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 "
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);" |":PRINTTAB(14);" |":PRINTTAB(14);"●"
230 PRINT"▯"
240 Z = 20 : GOSUB 500
250 IF L = X THEN PRINTTAB(25);"BRAVO"
260 IF L <>X THEN PRINTTAB(25);"PERDU"
270 PRINT"▮FAIRE - SHIFT - POUR CONTINUER"
280 WAIT653,1
290 RUN
500 PRINT"▮";
510 FOR P = 0 TO Z
520 PRINT
530 NEXT P
540 RETURN


READY.
```

```
 1
 2
 3
 4  ────────────────────────────────
 5
 6  ────────────────────────────────
 7
 8  ──────────────┬─────────────────
 9               |
 A  ─────────────●──────────────────
 B
 C  ────────────────────────────────
 D
 E

NOTE SOL GRAVE

NUMERO DE LA LIGNE? A     BRAVO

FAIRE - SHIFT - POUR CONTINUER
```

Poker

REFERENCES : Difficulté de programmation : difficile.
Catégorie de jeu : jeu contre le Commodore.
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 Commodore.
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 Commodore de jouer.

LE PROGRAMME : Les lignes 3Ø et 4Ø définissent les cartes.
Une variable N est dimensionnée à 15, en ligne 5Ø, pour recevoir 10 cartes joueur et 5 cartes Commodore.
Le tirage sera effectué par une boucle I (6Ø-11Ø) et le contrôle par une boucle J (8Ø-1ØØ).
L'affichage des cartes est réalisé par le sous-programme 5ØØ-6ØØ.
Une autre boucle I (16Ø-18Ø) réalise le deuxième tirage du joueur, la ligne 2ØØ permettant le deuxième affichage, et la boucle I (5ØØ-57Ø) 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 Commodore, ce dernier n'ayant pas, dans ce programme, la possibilité de reprendre des cartes.
Il suffira ensuite de faire RETURN 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 Commodore, l'introduction des règles sur les paris et même pourquoi pas, des possibilités de bluff.

```
10 REM POKER
20 PRINT"■"
30 V$="7 8 9 10VADAROAS"
40 C$="TRCACOPI"
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"■■■■■■■■■■■■■■■■■■■■■■■■CARTES REFUSEES"
140 GOSUB 500
150 IF R$ ="0" THEN END
160 FOR I = 1 TO LEN(R$)
170 N(VAL(MID$(R$,I,1))) = N(I+5)
180 NEXT I
190 PRINT"■"
195 P = 1
200 GOSUB 500
210 K = K + 10
500 FOR I = 1 TO 5
510 J=2*INT(N(I+K)/8)+1
520 F=2*(N(I+K)-8*INT(N(I+K)/8))+1
525 R$ = MID$(V$,F,2)
530 PRINT"■":FOR S = 1 TO K : PRINT : NEXT S
540 PRINTTAB(7*I-4);"■■";I:PRINT:PRINT
550 PRINTTAB(7*I-5);"■";R$;"■:";MID$(C$,J,2)
560 PRINTTAB(7*I-7);"  ▔▔▔▔▔"
570 NEXT I
575 IF P = 1 THEN P = 0 : RETURN
580 IF K = 10 THEN PRINT"■■FAIRE - SHIFT - POUR CONTINUER"
    :WAIT653,1 : RUN
590 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■";TAB(16);:INPUTR$
600 RETURN

READY.
```

```
    1       2       3       4       5

  VA:CO   10:PI   DA:CA   10:CA   DA:CO



    1       2       3       4       5

  DA:TR   RO:CA   10:CO   RO:PI   9 :CO


 FAIRE - SHIFT - POUR CONTINUER
```

Le jeu de saute-mouton

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

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

LE PROGRAMME : La boucle I (5Ø-8Ø) initialise une table A(I), qui contient les positions des moutons.
La boucle I (1ØØ-15Ø) dessine le terrain de jeu et les moutons.
Les lignes 16Ø et 17Ø s'occupent du texte et des entrées.
La ligne 18Ø contrôle la validité du coup.
Le changement de joueur est réalisé en ligne 21Ø.

POUR JOUER : On commence l'exécution du programme par RUN. On obtient alors un affichage JOUEUR 1, qui indique le numéro du joueur, suivi de la question DEBUT, FIN ?
On répond en indiquant le numéro du pion à déplacer suivi de RETURN, 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.

```
READY.

10 REM LE JEU DE SAUTE-MOUTON
20 PRINT"■"
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 PRINT"■■■■■■■■"TAB(4*I);I
120 PRINT"■■■■■■■"TAB(4*I+1);" "
130 IF A(I) = 1 THEN PRINT"■■■■■■"TAB(4*I+1);"■■ "
140 IF A(I) = -1 THEN PRINT"■■■■■■"TAB(4*I+1);"■■ "
145 PRINT"■"
150 NEXT I
155 J = 1 - (H = -1)
160 PRINT"■■■■■■■■■■■■JOUEUR"J
170 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

READY.
```

```
     ■           ■   ■   ■   ■   ■

     1   2   3   4   5   6   7   8

JOUEUR 2

DEBUT , FIN? 7
?? 3
```

Lettres

REFERENCES : Difficulté de programmation : facile.
Catégorie de jeu : réflexion (deux 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 Commodore choisit au hasard un certain nombre de symboles (voyelles et consonnes) dans l'alphabet latin, et chacun des joueurs devra indiquer la longueur du mot qu'il a réussi à obtenir (l'appareil ne vérifiera pas si le mot est correct !).
Le score total de chaque joueur est affiché ensuite.

LE PROGRAMME : La boucle I (6Ø-9Ø) choisit dans l'alphabet entre 7 et 12 lettres, en fonction de la variable aléatoire L, tirée en ligne 3Ø.
Pour éviter d'avoir trop de consonnes, on utilise en ligne 8Ø une petite astuce, qui consiste à refaire au hasard un tirage dans la table A$, contenant des voyelles (on peut en avoir plus ou moins en utilisant d'autres RND).
La boucle 1ØØ-14Ø demande à chacun des joueurs la longueur de son mot.
Les lignes 15Ø et 16Ø comparent les résultats, et totalisent le score de chaque joueur, affiché par la ligne 17Ø.

POUR JOUER : On lance l'exécution du jeu avec RUN.
Les lettres sont alors affichées, puis la question JOUEUR 1 LONGUEUR ? à laquelle le premier joueur répond en donnant la longueur du mot qu'il a obtenu (la correction du mot devra toutefois être contrôlée par l'autre joueur) ; puis le joueur 2 fera de même et le score de chaque joueur sera alors affiché.
Pour continuer la partie, il suffira d'appuyer sur la touche -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 Commodore, 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.

```
READY.

10 REM LETTRES
15 PRINT"■"
20 S = 0 : J = 0
25 PRINT"■"
30 L = INT(RND(1)*6 +7)
40 A$ = "AEIOUEA"
50 PRINT"■■■■■LETTRES"
60 FOR I = 1 TO L
70 PRINT"■■■■■"TAB(7+2*I);CHR$(INT(RND(1)*26+65))
80 IF RND(1) < 0.4 THEN PRINT"■■■■■"TAB(7+2*I);MID$(A$,INT
   (RND(1)*7+1),1)
90 NEXT I
100 I = 1
110 PRINT"■■■■■■■■JOUEUR"I;:INPUT";LONGUEUR";L
115 PRINT"■"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 PRINT"■■■■■■■■■■SCORES : JOUEUR 1 :"S;";JOUEUR 2 :"J
180 PRINT"■POUR CONTINUER FAIRE - SHIFT -"
190 WAIT653,1
200 GOTO 25

READY.
```

```
LETTRES  L M O I E E I

JOUEUR 2 ;LONGUEUR

SCORES : JOUEUR 1 : 4 ;JOUEUR 2 : 0

POUR CONTINUER FAIRE - SHIFT -
```

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 Commodore, 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 I (80-100) lit les lignes DATA (180-190) concernant les suffixes.
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 alors affiché à la suite du précédent.
Taper RUN/STOP pour sortir du programme.

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

```
READY.

10 REM MACHINAMOT
20 PRINT"■";
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 PRINTC$
145 FOR S = 1 TO 1500 : NEXT S
150 GOTO 110
160 DATA ANDRO,COSMO,MICRO,HELIO,PEDO,PORNO,PALEO
170 DATA MACRO,HYDRO,STENO,PATHO,OLEO,MYTHO,HOMEO,OCTO
180 DATA CRATE,PHILE,PHOBE,TROPE,GYNE,MANE,NAUTE,PHAGE,FUGE
190 DATA PATHE,CARDE,DERME,SCOPE,VORE,PHONE

READY.

PATHOPHILE
PALEOPATHE
HOMEOPHONE
OCTOPHAGE
MACROVORE
OCTOPHAGE
PATHOTROPE
COSMOCRATE
PEDOGYNE
COSMOMANE
PALEOPHAGE
OLEONAUTE
OCTOPHAGE
MYTHOPATHE
MACRONAUTE
HELIOPHOBE
HELIOPATHE
```

Espion

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

LE JEU : Le programme peut servir simultanément de codeur et de décodeur de message secrets, suivant une méthode très facile à mettre en oeuvre sur ce type d'appareil.
Un petit générateur aléatoire situé en fin de programme est utilisé pour modifier de façon arbitraire chaque lettre du message ; on devra bien sur, veiller à utiliser toujours le même nombre (noyau) pour commencer le codage ou le décodage de deux messages identiques.

LE PROGRAMME : Le noyau initial (qui sera un entier quelconque) est entré en ligne 3Ø ; le message est entré à la ligne 5Ø.
La ligne 6Ø permet de choisir entre les options de codage et de décodage.
La boucle I (7Ø-14Ø) est utilisée soit pour effectuer le codage, soit pour effectuer le décodage.
La ligne 9Ø appelle le générateur aléatoire.
La ligne 1ØØ réalise la transposition des lettres.
Les lignes 11Ø et 12Ø vérifient que l'on reste bien dans les limites autorisées.
Les résultats seront imprimés par la ligne 15Ø.
Les lignes 1ØØØ à 1Ø3Ø contiennent le générateur.

POUR JOUER : Lancer le programme par RUN.
Entrer ensuite un entier N qui servira de noyau pour le générateur (il faudra le retenir pour le décodage).
Puis taper le message, suivi de 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.

```
READY.

10 REM ESPION
20 PRINT"■"
30 INPUT"VOTRE CODE";N
40 A = 1/N
50 INPUT"■VOTRE MESSAGE";A$
60 INPUT"■VOULEZ-VOUS (C)ODER OU (D)ECODER";R$
65 B$=""
70 FOR I = 1 TO LEN(A$)
80 T = ASC(MID$(A$,I,1))
90 GOSUB 1000
100 T = T - A*((R$="C")-(R$="D"))
110 IF T >90 THEN T = T - 26
120 IF T < 65 THEN T = T + 26
130 B$ = B$ + CHR$(T)
140 NEXT I
150 PRINT"■■■■■■■■■■RESULTAT :";B$ : END
1000 A = (A + π)↑8
1010 A = A - INT(A)
1020 A = INT(26*A + 1)
1030 RETURN

READY.


VOTRE CODE? 1234

VOTRE MESSAGE? BONJOUR MONSIEUR

VOULEZ-VOUS (C)ODER OU (D)ECODER? C

RESULTAT :FHOTCVBHNYBTSSVB




VOTRE CODE? 1234

VOTRE MESSAGE? FHOTCVBHNYBTSSVB

VOULEZ-VOUS (C)ODER OU (D)ECODER? D

RESULTAT :BONJOURTMONSIEUR
```

Accrochez les wagons

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

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

LE PROGRAMME : Les lignes 2Ø et 3Ø dessinent la locomotive, la la ligne 4Ø le wagon, et les roues sont en ligne 5Ø.
La position initiale de la locomotive est définie par les lignes 7Ø et 8Ø, celle du wagon par les lignes 9Ø et 1ØØ.
La ligne 11Ø évite qu'ils soient initialement trop proches l'un de l'autre.
Les lignes 14Ø et 15Ø réalisent l'affichage du wagon et les lignes 17Ø et 18Ø celui de la locomotive.
Le test de fin est en ligne 2ØØ.
Les mouvements à effectuer sont entrés dans une variable alphanumérique R$, en ligne 22Ø, et sont décodés dans une boucle I (25Ø-28Ø), pour être exécutés en ligne 29Ø.

POUR JOUER : RUN permet d'afficher les positions de départ de la locomotive et du wagon.
On tape alors la série des déplacements à effectuer, sous la forme DDDHHHBHGG, suivi de RETURN.
La locomotive sera alors affichée dans sa nouvelle position, et l'on recommence, jusqu'à ce qu'elle soit accrochée au wagon.
On obtiendra alors l'affichage du score.

EXTENSIONS POSSIBLES : Il sera intéressant de prévoir la perte de la partie si, par accident, au cours du déplacement, la locomotive vient en collision avec le wagon.
On pourra aussi prévoir plusieurs wagons, et un ordre d'accrochage, jouer à plusieurs avec plusieurs locomotives, dessiner les voies ferrées, introduire des obstacles à contourner obligatoirement d'un certain côté, etc.

```
10 REM ACCROCHEZ LES WAGONS
20 D$=" ▄▄ ▗"
30 A$="▄▩  ▀"
40 B$="└──┘"
50 C$="O O"
60 N = 0
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 PRINT"▒";
130 Z = WX : GOSUB 500
140 PRINTTAB(WY);"▒";B$
150 PRINTTAB(WY);C$
160 Z = LX - 1 : GOSUB 500
170 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 INPUT"CHEMIN A SUIVRE (DGGHBB...)";R$
230 X = 0 : Y = 0
240 N = N + 1
250 FOR I = 1 TO LEN(R$)
260 X = X - (MID$(R$,I,1)="B" AND LX < 22) + (MID$(R$,I,1)=
    "H" AND LX > 0)
270 Y = Y - (MID$(R$,I,1)="D" AND LY < 37) + (MID$(R$,I,1)=
    "G" AND LY > 0)
280 NEXT I
290 LX = LX + X : LY = LY + Y
295 GOTO 120
300 Z = 22 : GOSUB 500
310 PRINT"BRAVO ; TROUVE EN"N;" COUPS"
320 END
500 PRINT"▒";
510 FOR I = 0 TO Z
520 PRINT
530 NEXT I
540 RETURN

READY.
```

```
                                   ▄▄ ▗
                                   ↑▄
                                    OO
```

```
                 └──┘
                 OO
```

```
CHEMIN A SUIVRE (DGGHBB...)? BBDD
```

Tirage d'un dé truqué

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

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

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

POUR JOUER : Commencer l'exécution en frappant RUN, puis, à la question PROBABILITE DU DE 1, introduire un nombre compris entre Ø et 1.
Faire de même pour les dés 2, 3, etc., sans oublier que le total des probabilités introduites devra être très proche de 1.
Ensuite, à l'affichage de la question NOMBRE DE TIRAGES on introduira un entier suffisamment grand, suivi de 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, qui apparaissent en caractères inversés : ainsi 32.3 apparaissant sous le 1 inversé signifie que le tirage de 1, auquel on avait affecté une probabilité de 0.3 a été réalisé 32 fois.

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

```
10 REM DE TRUQUE
15 PRINT"■"
20 P = 0
30 DIM E(6)
40 FOR I = 1 TO 6
45 PRINT"■■■■";TAB(17);"              "
50 PRINT"■■■■PROBABILITE DU DE"I;:INPUTE(I)
60 P = P + E(I)
70 IF E(I) < 0 OR E(I) >= 1 OR P > 1.1 THEN 200
80 PRINT"■■■■■■■";TAB(6*I-6);"DE"I;
90 PRINT"■■■■■■■■■■";TAB(6*I-6);E(I)
100 NEXT I
110 IF P <0.9 THEN RUN
115 INPUT"■NOMBRE DE TIRAGES";P
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
175 PRINT"■■■■■■■■■■■■■■■■"TAB(6*N-4);N
178 PRINT"■■■■■■■■■■■■■■■■■■"TAB(6*N-5);E(N)
180 NEXT J
190 END
200 PRINT"IMPOSSIBLE : 0 <= P <= 1"
210 P = P - E(I)
220 FOR J = 1 TO 2000: NEXT J
230 PRINT"■                                  "
240 GOTO 50

READY.
```

```
 PROBABILITE DU DE 6 ? 0.05

 DE 1  DE 2  DE 3  DE 4  DE 5  DE 6

  .1    .3    .2    .15   .15   .05

 NOMBRE DE TIRAGES? 100

    1     2     3     4     5     6

   11.1  28.3  25.2  16.15 15.15 5.05
```

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

LE PROGRAMME : La boucle I (3Ø-9Ø) permet de choisir, au hasard, 21 cartes sur 52, et la boucle emboîtée J (5Ø-7Ø) permet de s'assurer que l'on ne réalise pas deux fois le même tirage.
La ligne 11Ø contient les couleurs des cartes, ainsi que les noms des cartes figures (VA, DA, RO, AS), dans une variable alphanumérique B$.
La boucle I (12Ø-21Ø) réalise l'affichage du jeu.
La ligne 13Ø décode la couleur de la carte, la ligne 14Ø choisit la position d'affichage, la ligne 15Ø décode la hauteur.
L'affichage proprement dit est effectué de la ligne 18Ø à la ligne 2ØØ.
La ligne 23Ø permet, au joueur, d'entrer le numéro de la colonne où se trouve la carte.
La boucle J (27Ø-29Ø) réalise le brouillage, et les boucles emboîtées J (32Ø-36Ø) et I (33Ø-35Ø) fabriquent la nouvelle variable A$ à utiliser pour le prochain affichage.

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 2ème affichage et l'on fait de même ; puis un 3ème affichage et l'on fait de même.
Au 4ème affichage, la carte choisie est auto-

matiquement 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 PRINT"■"
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$ = "TRCACOPIVADAROAS"
120 FOR I = 1 TO LEN(A$)*P - 11*(P=0)
130 S = INT((ASC(MID$(A$,I,1))-38)/13)
140 J = 7*INT((I-1)/7)
150 T = ASC(MID$(A$,I,1))-37-13*S
160 IF P = 0 THEN PRINT"■VOTRE CARTE EST :";"       ":PRINT"■"; :GOTO 180
170 PRINT"■":FOR L = 0 TO (I-J)*P :PRINT:NEXT L
180 IF T <= 9 THEN PRINTTAB(J*P+INT(J/2)-16*(P=0));STR$(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)
210 NEXT I
220 IF P = 0 THEN 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)
350 NEXT I
360 NEXT J
370 GOTO 110

READY.
```

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

QUELLE COLONNE? 2
```

Le jeu de Marienbad

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

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

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

```
READY.

10 REM LE JEU DE MARIENBAD
15 PRINT"■"
20 INPUT"NOMBRE DE TAS ( <=7 )";N
30 IF N >=8 THEN RUN
40 S = 0
50 DIM A(N)
60 FOR I = 1 TO N
70 PRINT"■TAS"I;:INPUT";NOMBRE D'ALLUMETTES";A(I)
80 S = S + A(I)
90 NEXT I
95 H = 1
100 PRINT"■"
110 FOR I = 1 TO N
120 PRINT"■■■■■"TAB(5*I+2);A(I)
130 PRINT"■■TAS"TAB(5*I+2);I
140 NEXT I
150 PRINT"■■JOUEUR"H
160 INPUT"■■DANS QUEL TAS";T
170 IF T > N THEN 100
180 IF A(T) = 0 THEN 100
190 INPUT"■■■■■■■■■■■■■■■■■■QUANTITE A ENLEVER";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"■■LE JOUEUR"H;" A GAGNE" : END
260 GOTO 100

READY.
```

```
        5    4    6    4    7    4

TAS     1    2    3    4    5    6

JOUEUR 1

DANS QUEL TAS? 4

QUANTITE A ENLEVER? 1
```

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 (3Ø-6Ø) dessine la portée.
La variable N$, en ligne 7Ø, contient les noms des notes.
La deuxième boucle I (8Ø-16Ø) permet de faire le choix d'une note de façon aléatoire, à la ligne 9Ø ; son nom est calculé en ligne 1ØØ et la note est dessinée par la ligne 12Ø.
Les lignes 18Ø et 19Ø permettent de recommencer en tapant SHIFT.

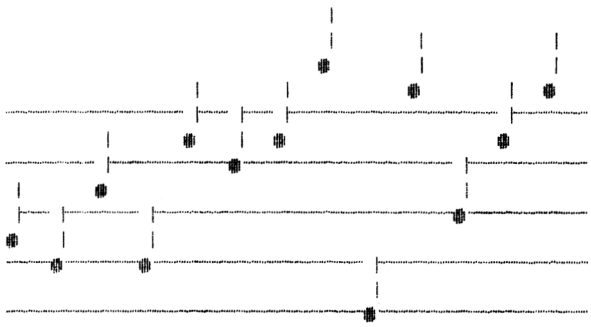
POUR JOUER : Lancer l'exécution du programme en tapant RUN. La portée sera alors très rapidement affichée, puis les notes sont représentées, avec leur nom affiché en bas de l'écran.
Une pression sur la touche SHIFT provoquera l'affichage d'une nouvelle série de notes, et une pression sur la touche RUN/STOP provoque la fin de la partie.

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

```
READY.

10 REM COMPOSE
20 PRINT"■"
30 FOR I = 1 TO 5
40 Z = 3 + 2*I : GOSUB 500
50 PRINT"■──────────────────────────────────────────────────────────■"
60 NEXT I
70 N$ = "SI LA SOLFA MI RE DO "
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);" ■":PRINTTAB(3*I-3);" ■":PRINTTAB(3*I-3);"●"
130 PRINT"■"
140 Z = 18 : GOSUB 500
150 PRINTTAB(3*I-3);D$
160 NEXT I
170 Z = 21 : GOSUB 500
180 PRINT"FAIRE - SHIFT - POUR CONTINUER"
190 WAIT653,1
200 RUN
500 PRINT"■"
510 FOR P = 0 TO Z
520 PRINT
530 NEXT P
540 RETURN

READY.
```



```
LA SOLDO SOLMI RE MI LA MI SOLSI MI SOL

FAIRE - SHIFT - POUR CONTINUER
```

Jour de la semaine

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

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

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

POUR JOUER : Lancer le programme 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 et le programme revient sur la question JOUR.
Pour continuer, il suffit de retaper un jour, un mois, une année;
Pour terminer, il faut répondre Ø à la question JOUR.

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

```
READY.


10 REM JOUR DE LA SEMAINE
20 PRINT"■"
30 INPUT"JOUR.....:";J
40 IF J = 0 THEN 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$="JANVFEVRMARSAVRIMAI JUINJUILAOUTSEPTOCTONOVEDECE"
110 M$ = MID$(M$,4*M-3,4)
120 IF M < 3 THEN GOSUB 200
130 S=J-1+INT(13*(M+1)/5)+INT(5*A/4)-S
140 S=S-7*INT(S/7)
150 J$="DIMANCHELUNDI   MARDI   MERCREDIJEUDI   VENDREDISAMEDI  "
160 J$=MID$(J$,8*S+1,8)
170 PRINT"■■■LE";J;"■ ";M$;B;"■ ETAIT UN ";J$
180 PRINT"■■■FAIRE - SHIFT - POUR CONTINUER":WAIT 653,1
190 GOTO 20
200 M = M + 12
210 A = A - 1
220 RETURN

READY.


JOUR.....:? 14

MOIS ( 1-12 )..:? 7

ANNEE.....:? 1983

LE 14 JUIL 1983 ETAIT UN JEUDI

FAIRE - SHIFT - POUR CONTINUER
```

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

LE PROGRAMME : Deux tables (B et C) sont utilisées aux lignes 40 et 50, pour enregistrer l'état de la population.
Les boucles F (230-310) et G (240-300) sont utilisées en début de jeu pour définir la population initiale (lignes 250 et 260).
La 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.

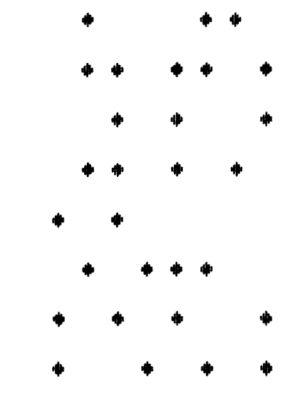
POUR JOUER : Dans la version automatique, il suffit de lancer le programme par RUN et de regarder.
Pour arrêter, il faudra taper RUN/STOP et RESTORE.
Dans la version manuelle,la configuration de départ sera entrée, après le lancement du programme par RUN, en tapant 0 si l'on ne veut rien mettre sur la 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 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.

```
READY.

10 REM LE JEU DE LA VIE (AUTOMATIQUE)
20 PRINT"□"
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"□"
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"■":FOR T = 0 TO 2*F:PRINT:NEXT T
280 IF C(F,G) = 1 THEN PRINTTAB(2*G);"♦"
290 B(F,G) = C(F,G)
300 NEXT G
310 NEXT F
320 GOTO 80

READY.
```



```
10 REM LE JEU DE LA VIE (MANUEL)
20 PRINT"□"
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"□"
230 FOR F = 1 TO A
240 FOR G = 1 TO A
250 IF I <>0 THEN 270
255 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■LIGNE";F;" COLONNE";G;"
    ENTRER 1 OU 0"
260 INPUT"■■■■■■■■■■■■■■■■■■■■■■■■";C(F,G)
270 PRINT"■":FOR T = 0 TO 2*F:PRINT:NEXT T
280 IF C(F,G) = 1 THEN PRINTTAB(2*G);"♦"
290 B(F,G) = C(F,G)
300 NEXT G
310 NEXT F
320 GOTO 80

READY.
```

```
      ♦ ♦ ♦ ♦ ♦     ♦

      ♦ ♦       ♦ ♦ ♦

  ♦ ♦       ♦ ♦ ♦ ♦

    ♦   ♦   ♦   ♦ ♦ ♦

    ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦

  ♦ ♦             ♦ ♦

    ♦ ♦ ♦       ♦ ♦ ♦

  ♦ ♦       ♦ ♦ ♦
```

```
LIGNE 9  COLONNE 10 ENTRER 1 OU 0
```

NIVEAU 3

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

N°	Nom du programme	Thème et particularités
18	**Bridge**	Mettez au point vos propres problèmes de bridge.
19	**Le loup et les agneaux**	Un jeu de réflexion, pour 2 joueurs.
20	**Hanoï**	Un conseil : commencez petit !!!

Manipulation des chaînes

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

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

Ainsi si,

A$ = "MONSIEUR"

on aura :

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

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

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

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

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

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

Il est plus difficile de se passer de la fonction RIGHT$, mais il serait possible de faire MID$ (A$, LEN (A$)-3, 3) pour obtenir le même résultat.

Le Basic du Commodore 64 ne permet pas le remplacement direct d'une sous-chaîne par une affectation du type MID$ (-.-) = " ". Il sera nécessaire, pour remplacer la Ième lettre de la chaîne A$ d'écrire :

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

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

Un dé

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

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

LE PROGRAMME : La ligne 3Ø utilise le générateur aléatoire du Commodore pour tirer un nombre de 1 à 6.
La boucle J(5Ø-7Ø) dessine les bords du dé, et les lignes 4Ø et 8Ø dessinent le haut et le bas.
Les lignes 1ØØ, 11Ø, 13Ø, 14Ø, 16Ø et 17Ø sont utilisées pour le dessin des points.
Les lignes 18Ø et 19Ø permettront le cas échéant de refaire un tirage.

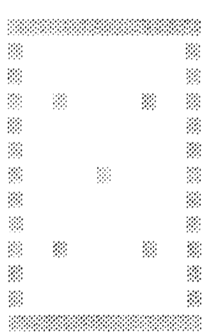
POUR JOUER : On lance le programme par RUN. Un dé grand format est affiché sur l'écran, ainsi que la question AUTRE TIRAGE (O/N). On répondra O pour relancer le programme (suivi de RETURN).

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

```
READY.

10 REM UN DE
20 PRINT"♥"
30 A = INT(RND(1)*6 + 1)
40 PRINT"QQQQ]]]]]]]]]]]]▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
50 FOR J = 0 TO 10
60 PRINT"]]]]]]]]]]]]▒         ▒"
70 NEXT J
80 PRINT"]]]]]]]]]]]]▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
90 PRINT"SQQQQQQQQQQ";
100 IF A <> 1 THEN PRINTTAB(13)"▒";
110 IF A >= 4 THEN PRINTTAB(19)"▒"
120 PRINT"SQQQQQQQQQQQQQQQ";
130 IF A = 6 THEN PRINTTAB(13)"▒     ▒";
140 IF A/2 <> INT(A/2) THEN PRINTTAB(16)"▒";
150 PRINT"SQQQQQQQQQQQQQQQQQQQ";
160 IF A >= 4 THEN PRINTTAB(13)"▒";
170 IF A <> 1 THEN PRINTTAB(19)"▒"
180 INPUT"SQQQQQQQQQQQQQQQQQQQQQQQQQQQQQAUTRE TIRAGE ( O/N )";R$
190 IF LEFT$(R$,1)="O" THEN RUN

READY.
```



AUTRE TIRAGE (O/N)? N

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 4Ø interdit de demander des nombres de plus de 8 chiffres.
Le nombre à deviner est placé, en ligne 6Ø, dans une variable alphanumérique X$.
Les comparaisons avec le nombre essayé sont faites à la ligne 11Ø (qui teste si l'on a trouvé juste), et dans la boucle I (13Ø-17Ø), où l'on fait la comparaison "lettre à lettre".
L'affichage du résultat est réalisé grâce à la ligne 18Ø.
Le compteur du nombre d'essais est en ligne 19Ø.

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 Commodore affichera VOUS AVEZ TROUVE...EN...ESSAIS. Puis AUTRE PARTIE (O/N) ? ; on fera O suivi de RETURN 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 Commodore qui tiendra alors le rôle de décodeur.

```
READY.

10 REM PLUS GRAND , PLUS PETIT
20 PRINT"]"
30 INPUT"NOMBRE DE CHIFFRES";C
40 IF C > 8 THEN RUN
50 E = 1
60 X$ = STR$(INT(RND(1)*9↑C + 10↑(C - E)))
70 X$ = RIGHT$(X$,LEN(X$)-1)
80 PRINT"SQQQQESSAI NO.  NOMBRE ESSAYE   >    <"
90 PRINT"SQQQQQ────────────────────────────────────────"
100 PRINTTAB(4)E;:INPUT"    ";P$:PRINT"]";
110 IF P$ = X$ THEN 200
120 A = 0 : B = 0
130 FOR I = 1 TO C
140 IF MID$(P$,I,1)>MID$(X$,I,1) THEN A = A + 1
150 IF MID$(P$,I,1)<MID$(X$,I,1) THEN B = B + 1
170 NEXT I
180 PRINTTAB(27)A;TAB(32)B
190 E = E + 1
195 GOTO 100
200 PRINT"S"
210 PRINT"VOUS AVEZ TROUVE ";X$;" EN"E;" ESSAIS"
220 PRINT"                                         "
230 PRINT"]]":INPUT"AUTRE PARTIE ( O/N )";R$
240 IF LEFT$(R$,1)="O" THEN RUN

READY.


NOMBRE DE CHIFFRES? 4

ESSAI NO.  NOMBRE ESSAYE   >    <
─────────────────────────────────────────
    1      ? 1289          2    2
    2      ? 3456          3    1
    3      ? 7832          2    2
    4      ? 3333          0    3
    5      ? 9399          3    0
    6      ? 4321
```

Ordre

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

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

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

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

LE PROGRAMME : Les boucles emboîtées I (4Ø-1ØØ) et J (6Ø-8Ø) définissent une variable alphanumérique A$, contenant les 10 premières lettres de l'alphabet dans le désordre.
La ligne 12Ø affiche A$, et la ligne 13Ø affiche les numéros de lettres à utiliser pour la permutation.
On entre la permutation en ligne 15Ø.
Les lignes 16Ø et 17Ø transforment la variable alphanumérique entrée en une variable numérique.
La ligne 18Ø appelle une première fois le sous-programme d'inversion.
La ligne 23Ø appelle une deuxième fois le sous-programme d'inversion.
La ligne 25Ø comptabilise les essais.
La ligne 26Ø contrôle les résultats.

POUR JOUER : Lancer le programme par RUN.
Le mot initial est affiché, chaque lettre étant repérée par un numéro de 1 à 9.
A la question : INVERSION (I/J), on répond en entrant un nombre de deux chiffres xy ; x est le numéro de la première lettre, et y le numéro de la seconde. Les deux inversions sont alors successivement effectuées, et l'on recommence.
Le nombre de coups sera affiché en fin de partie.

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

```
READY.

10 REM ORDRE
20 PRINT"■"
30 A$ =""
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 S = 1
120 PRINT"■■■";A$;"■■"
130 PRINT"■■0123456789"
135 PRINT"■■ESSAI NUMERO"S
140 PRINT"■■■■■■■"TAB(33);"     ";
150 INPUT"■■■■■■■INVERSION (I ET J SOUS FORME IJ )";R$
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 IF A$="ABCDEFGHIJ" THEN PRINT"TROUVE EN"S;"COUPS":END
270 GOTO 120
500 B$ = ""
510 FOR T = 1 TO LEN(A$)
520 IF T = I THEN N$ = MID$(A$,T,1)
530 IF T = J THEN M$ = MID$(A$,T,1)
540 NEXT T
550 FOR T = 1 TO LEN(A$)
560 IF T = I THEN B$ = B$ + M$ : GOTO 600
570 IF T = J THEN B$ = B$ + N$ : GOTO 600
580 B$ = B$ + MID$(A$,T,1)
600 NEXT T
610 A$ = B$
620 RETURN

READY.

FGBADHCJIE
0123456789

INVERSION (I ET J SOUS FORME IJ )?
```

Le pendu

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

LE JEU : Il serait sans doute assez difficile de retrouver le nom de l'inventeur de ce jeu très à la mode, mais il est certain qu'il existe depuis des temps immémoriaux, quelquefois pratiqué sous des formes assez différentes (par exemple, reconstituer un petit cochonnet en lançant des dés).
L'apparition des jeux sur micro-ordinateur et la magie des écrans de télévision lui ont fait connaître un renouveau certain : dans la version retenue, l'un des joueurs doit entrer un mot de longueur quelconque. L'ordinateur affichera alors des tirets à l'emplacement des lettres de ce mot et demande, au deuxième joueur, de proposer une lettre.
Chaque réponse correcte fait apparaître la lettre à son emplacement exact dans le mot, et chaque réponse fausse fait progresser le dessin de la potence.
Pour gagner, il suffit de réussir à trouver le mot inconnu avant d'être pendu.

LE PROGRAMME : Le mot à deviner est introduit en début de programme (ligne 3Ø) ; une variable B$, en ligne 4Ø, contient tous les éléments du dessin.
La boucle I (6Ø-8Ø) fabrique un mot P$ qui contient des tirets à l'emplacement de chacune des lettres du mot initial.
La boucle I (12Ø-14Ø) s'occupe du dessin (plus ou moins complet, selon la valeur de P).
Les lignes 15Ø-16Ø affichent le résultat et arrêtent le jeu.
La lettre essayée est entrée en ligne 17Ø et testée dans une boucle I (19Ø-21Ø) ; le sous-programme 3ØØ-32Ø remplace chaque tiret par la lettre correspondante.

POUR JOUER : Après le lancement du programme par RUN, l'un des joueurs doit entrer son mot et taper 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 : On pourra prévoir une liste de mots à trouver, inscrits dans le programme. Ajouter du son, améliorer le dessin du pendu, etc...

```
READY.


10 REM LE PENDU
20 PRINT"□"
30 INPUT"VOTRE MOT";R$
40 B$ = "┘ └▌▌▒▒▒▚▚▚▚▚▚ |  ♠  .  .  ▀▀▀   "
50 P$=""
60 FOR I =1 TO LEN(R$)
70 P$ = P$ + "-"
80 NEXT I
90 P = 11
100 PRINT"□"
110 PRINT"▒▒▒▒  ";P$
120 FOR I = 11 TO P STEP -1
130 PRINT"▒";TAB(22);MID$(B$,1+3*(I-1),3);"□"
140 NEXTI
150 IF P$=R$ THEN PRINT"▒▒BRAVO":END
160 IF P = 1 THEN PRINT"▒▒PERDU ; LE MOT ETAIT :";R$:END
170 INPUT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒VOTRE LETTRE :";S$
180 T = 0
190 FOR I = 1 TO LEN(R$)
200 IF MID$(R$,I,1)=S$ THEN GOSUB 300
210 NEXT I
220 IF T <>0 THEN 100
230 P = P - 1
240 GOTO 100
300 T = 1
310 P$ = LEFT$(P$,I-1) + S$ + RIGHT$(P$,LEN(P$)-I)
320 RETURN

READY.
```

```
 E---TE
```



```
VOTRE LETTRE :? Q
```

Belote

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

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

LE PROGRAMME : En ligne 2Ø, on crée un tableau pouvant contenir les 20 cartes.
La boucle I (5Ø-1ØØ) assure le tirage aléatoire de 20 nombres, pris parmi 32 ; la boucle J (7Ø-9Ø) vérifie que l'on ne réalise pas deux fois le même tirage.
La deuxième boucle I (11Ø-3ØØ) assure ensuite l'affichage, les noms des cartes étant affichés en clair, et la couleur étant décodée et affichée en clair, grâce aux lignes 16Ø, 17Ø, 18Ø et 19Ø.

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

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

```
READY.

10 REM BELOTE
20 DIM A(20)
30 K = 1
40 PRINT"■"
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$="CA"
130 S = 1 + A(I) - 8*INT(A(I)/8)
140 R$=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$="TR"
210 IF A(I) <= 16 THEN S$="PI"
220 IF A(I) <= 8 THEN S$="CO"
230 PRINT"■"
240 FOR P = 1 TO 2*K
250 PRINT
260 NEXT P
270 PRINTTAB(10*INT((I-1)/5));R$;" ";S$
280 K = K + 1
290 IF K = 6 THEN K = 1
300 NEXT I

READY.
```

```
DA PI     10 TR     8  PI     7  TR

VA TR     DA TR     AS CO     7  PI

10 PI     RO CO     7  CA     8  TR

VA CA     9  CO     RO TR     9  PI

7  CO     DA CA     DA CO     9  CA
```

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

LE PROGRAMME : La ligne 3Ø permet d'obtenir l'entier de départ compris entre 5 et 20.
Une boucle I (9Ø-17Ø) permet d'entrer successivement les 5 opérations.
4 sous-programmes sont utilisés pour les calculs (aux lignes 1ØØØ et 4ØØØ).
Après chaque calcul, le résultat est affiché (seul, le codeur devra regarder l'écran pendant cette opération) et le résultat définitif restera affiché pour le décodeur à la fin des calculs (ligne 16Ø).
La ligne 18Ø affiche les 5 opérations utilisées, ainsi que les opérateurs correspondants.

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

```
READY.


10 REM UN TAQUIN DE CALCUL
20 PRINT"■"
30 X = INT(RND(1)*16+5)
40 PRINT"NOMBRE INITIAL ";X
50 PRINT"ENTRER 5 OPERATIONS"
60 PRINT"SOUS LA FORME +5 , /2 , *4 ..."
70 F = X
80 DIM R$(5)
90 FOR I = 1 TO 5
100 PRINT"■■■■■■■                              "
110 INPUT"■■■■■■■VOTRE OPERATION";R$(I)
120 A = ASC(LEFT$(R$(I),1))-41
130 IF A <= 0 OR A = 3 OR A = 5 OR A > 6 THEN 100
140 IF A > 3 THEN A = (A/2)+1
150 ON A GOSUB 1000,2000,3000,4000
155 PRINT"■■■■■■■■■■■"TAB(13);"      "
160 PRINT"■■■■■■■■■■■NOMBRE OBTENU";F
170 NEXT I
180 PRINT"■OPERATIONS :";R$(4);" ";R$(1);" ";R$(5);" ";
    R$(3);" ";R$(2)
190 PRINT"■BON COURAGE"
200 END
1000 F = F*VAL(RIGHT$(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


READY.



NOMBRE INITIAL  16
ENTRER 5 OPERATIONS
SOUS LA FORME +5 , /2 , *4 ...


VOTRE OPERATION? +30


NOMBRE OBTENU 100

OPERATIONS :*35 +14 +30 -4 /5

BON COURAGE
```

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 Commodore 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 2Ø permet d'enregistrer, dans une variable alphanumérique D$, les lettres de l'alphabet, dans l'ordre : voyelles, consonnes usuelles, autres consonnes. Le mot à deviner est introduit en ligne 4Ø (en fait, il suffirait d'introduire sa longueur).
La boucle I (5Ø-8Ø) imprime des tirets aux emplacements des lettres à trouver.
La boucle en ligne 1ØØ sert uniquement à simuler le temps de recherche d'une lettre.
La formule, placée à la ligne 11Ø, permet de modifier légèrement la probabilité de tirage, en faveur des voyelles et des consonnes usuelles.
La ligne 13Ø provoque un nouveau tirage, et la boucle I (17Ø-2ØØ) réimprime les tirets et les lettres trouvées.
La ligne 21Ø incrémente le compteur d'essais.
La ligne 22Ø affiche le nombre d'essais réalisés, en fin de partie.

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

EXTENSIONS POSSIBLES : Il sera sans doute intéressant, pour de jeunes enfants, de prévoir une version non

automatique où le joueur répondrait OUI ou NON pour chaque lettre proposée par le Commodore ; on pourra aussi imaginer d'ajouter un affichage graphique, par exemple une potence ou une voiture qui avance vers un précipice, etc.

```
READY.

10 REM PENDU INVERSE
15 PRINT"■"
20 D$ = "YAEIOUBCDFGLMNPRSTVWHKQXZ"
30 C = 0 : T$ = ""
40 INPUT"ENTRER VOTRE MOT";R$
45 PRINT"■"
50 FOR I = 1 TO LEN(R$)
60 PRINT"■■■■■■";TAB(I);"-"
70 T$ = T$ + "-"
80 NEXT I
90 PRINT"■■■■■■■■■■■■JE CHERCHE....; ESSAI";C
100 FOR J = 1 TO 1000 : NEXT J
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 D$ = LEFT$(D$,X-1)+"-"+RIGHT$(D$,26-X)
160 PRINT"■■JE PRENDS : ";E$
170 FOR I = 1 TO LEN(R$)
180 IF E$ = MID$(R$,I,1) THEN T$ = LEFT$(T$,I-1)+E$+RIGHT$
    (T$,LEN(T$)-I)
190 PRINT"■■■■■■";TAB(I);MID$(T$,I,1)
200 NEXT I
210 C = C + 1
220 IF T$ = R$ THEN PRINT"■■■■■■■■■■■■■■■■TROUVE EN";C-1;
    " COUPS" : END
230 GOTO 90

READY.
```

```
 COMMODORE


JE CHERCHE....; ESSAI 20

JE PRENDS : M

TROUVE EN 20  COUPS
```

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 identifiées 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 X 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 3Ø permet de confectionner l'entier X.
La boucle I (4Ø-9Ø) calcule les trois restes successifs, et la boucle emboîtée J (6Ø-8Ø) permet de les afficher sous une forme assez visuelle, aux bons emplacements.
Les lignes 35 et 38 sont là pour indiquer de quelle division il s'agit.
Le nombre proposé par le joueur est entré à la ligne 1ØØ ; il est affiché, et testé aux lignes 11Ø et 12Ø (affichage de fin de partie : BRAVO ou PERDU).
Les lignes 13Ø et 14Ø s'occupent de gérer la prochaine partie, selon la réponse du joueur.

POUR JOUER : On lance le programme en tapant RUN.
Les trois restes seront alors affichés sur la même ligne, symbolisés par des coeurs (l'absence de coeurs 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 AUTRE PARTIE (O/N)?
Pour faire un nouvel essai, on répondra 0, suivi de RETURN.

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

```
10 REM NICOMAQUE
20 PRINT"■"
30 X = INT(RND(1)*100 + 1)
35 PRINT"     RESTE DANS LA DIVISION PAR"
38 PRINT"■                3       5       7"
40 FOR I = 1 TO 3
50 A = X - (2*I+1)*INT(X/(2*I+1))
60 FOR J = 1 TO A
70 IF A<>0 THEN PRINT"■■■■■■■";TAB(J + 8*I + 6);"*"
80 NEXT J
90 NEXT I
100 INPUT"■■■VOTRE ESSAI";N
110 IF N = X THEN PRINT"■■BRAVO"
120 IF N <>X THEN PRINT"■■PERDU"
130 INPUT"■■■UNE AUTRE PARTIE ( O/N )";R$
140 IF LEFT$(R$,1) = "O" THEN RUN

READY.
```

```
     RESTE DANS LA DIVISION PAR

               3       5       7

               *       ***     *


VOTRE ESSAI? 35

PERDU


UNE AUTRE PARTIE ( O/N )? N
```

Course de chevaux

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

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

LE PROGRAMME : Une première boucle I (70-160) affiche les chevaux dans leur position de départ.
Une deuxième boucle I (100-160), utilisant le même NEXT I, les fera avancer ; la position de chaque cheval sera repérée par un chiffre ou une lettre dans une variable alphanumérique A$.
La ligne 120 fera avancer les chevaux.
La ligne 130 gère les arriyées.
La ligne 140 réalisera les impressions.
Le sous-programme 200-240 fait avancer le compteur des arrivées, et imprime les résultats.

POUR JOUER : On lance le programme par RUN.
A la question NOMBRE DE CHEVAUX : on répond par un entier de 1 à 10, suivi de RETURN.
La course se déroulera ensuite automatiquement.

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

```
READY.

10 REM COURSE DE CHEVAUX
20 PRINT"■";
30 INPUT"NOMBRE DE CHEVAUX";N
35 A$ = ""
40 P = 0
50 T = 1
60 A = P
65 PRINT"■■■■■"
70 FOR I = T TO N
80 A$ = A$ + "0"
90 GOTO 140
95 PRINT"■■■■■"
100 FOR I = 1 TO N
110 IF MID$(A$,I,1) = "K" THEN PRINT"■":GOTO160
120 B$ = CHR$(ASC(MID$(A$,I,1))+INT(RND(1)*4 + 1))
125 A$ = LEFT$(A$,I-1) + B$ + RIGHT$(A$,LEN(A$) - I)
130 IF B$ >= "K" THEN GOSUB 200
140 PRINT"■"TAB(ASC(MID$(A$,I,1))-46);"    ";"■";I
150 FOR K = 1 TO 600 : NEXT K
160 NEXT I
170 P = P + T
180 T = 0
190 IF A < N THEN 95
195 END
200 A$ = LEFT$(A$,I-1) + "K" + RIGHT$(A$,LEN(A$)-I  )
210 PRINT"■■■";P;"■■"
220 T = 1
230 A = A + T
240 RETURN

READY.

NOMBRE DE CHEVAUX? 5


          1

         2

             3

          4

        5
```

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 doite, en haut ou en bas.
Le gagnant est le premier qui arrive à faire sortir tous ses symboles (les lettres sortent en haut, et les chiffres à droite).
Il n'y a pas de prise, et les sauts ne sont pas autorisés ; on ne peut pas non plus se placer sur une case déjà occupée.



LE PROGRAMME : Les positions des chiffres et des lettres sont enregistrées dans une table N (4,4) ; les lignes 1Ø à 8Ø définissent les valeurs de départ.
Les lignes 1ØØ à 16Ø dessinent le terrain de jeu et la ligne 18Ø affiche le joueur qui a le trait.
La ligne 21Ø contrôle le coup joué, de même que les lignes 28Ø à 34Ø, et le changement de joueur est assuré par la ligne 35Ø.

POUR JOUER : Le terrain de jeu sera affiché après la frappe de RUN et l'ordinateur attend le coup du joueur désigné.
On entrera d'abord le chiffre ou la lettre représentant le pion à déplacer, suivi d'une indication de direction sous la forme N (pour Nord), S (pour Sud), E (pour Est), ou O (pour Ouest).
On pourra ainsi faire 2E suivi de 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 et de lettres et d'améliorer les contrôles. On peut aussi définir une stratégie pour le Commodore.

```
10 REM A TRAVERS
20 K = 1
30 DIM N(4,4)
40 FOR I = 1 TO 4
50 N(I,1) = ASC("0") + I
60 N(4,I) = ASC("?")+ I
70 NEXT I
80 N(4,1) = 0
90 PRINT"■"
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 PRINTTAB(2*U)"."
140 IF N(V,U) <>0 THEN PRINTTAB(2*U);CHR$(N(V,U))
150 NEXT V
160 NEXT U
170 Z = 15 : GOSUB 1000
180 PRINT"JOUEUR AYANT LES ";:IF K = 0 THEN PRINT"LETTRES"
190 IF K = 1 THEN PRINT"CHIFFRES"
200 INPUT"■VOTRE COUP ( N , S , E , O )";R$
210 IF ( K = 1 AND RIGHT$(R$,1)="O" ) 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)="O")
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

READY.

  1 . . .

  . 2 . .

  3 . . .

  . A B C



JOUEUR AYANT LES LETTRES

VOTRE COUP ( N , S , E , O )? CN
```

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 touver 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 Commodore 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.

LE PROGRAMME : 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...).

```
READY.

10 REM MASTERMIND
20 PRINT"■"
30 INPUT"■COMBIEN DE CHIFFRES";N
40 IF N > 6 THEN RUN
50 E = 1
60 X$ = STR$(INT(RND(1)*9↑N+10↑(N-1)))
70 X$ = RIGHT$(X$,LEN(X$)-1)
80 PRINT"■";
90 PRINT" NO.       NOMBRE        BON     PLACE"
95 PRINT"■"
100 FOR I = 0 TO 1 + E
110 PRINT
120 NEXT I
130 PRINTE;TAB(8);:INPUTE$:PRINT"■"
140 X = 0 : Y = X
150 IF E$ = X$ THEN PRINT"■TROUVE EN"E;"COUPS":END
160 FOR I = 1 TO N
170 IF MID$(X$,I,1)=MID$(E$,I,1) THEN 250
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 PRINTTAB(23);X;TAB(30);Y
280 E = E + 1
290 PRINT"■":GOTO 100

READY.

 NO.       NOMBRE        BON     PLACE


 1       ? 3456          1       1
 2       ? 7891          0       0
 3       ? 2035          2       2
 4       ? 4063          0       0
 5       ? 2255
TROUVE EN 5 COUPS
```

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 Commodore 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 Commodore en début de jeu ; vous devez, à chaque coup, choisir une colonne (les dizaines, les centaines, etc.) et ajouter dans cette colonne un certain entier ; le but est d'arriver à obtenir à un moment l'affichage d'un nombre ayant tous ses chiffres identiques.
Chaque résultat sera affiché un court instant, après quoi la main passera au joueur suivant, qui repartira de son précédent résultat.

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

POUR JOUER : Lancer l'exécution du programme par RUN.
Le nombre initial sera affiché pour le joueur n° 1, qui joue en tapant xy, suivi de 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 PRINT"■"
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"■"
120 IF N(H)<U↑(N-1) THEN N(H)=U↑N-ABS(N(H))
130 PRINT"■■■JOUEUR"H;" ; ESSAI"P
140 PRINT"■■■■■NOMBRE";TAB(15-N);N(H)
150 PRINTTAB(9);"■7654321■"
160 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$,1))*U↑(VAL(RIGHT$(R$,1))-1)
180 N(H) = N(H) - A
185 IF N(H) > U↑N THEN N(H) = N(H) - U↑N
187 N(H) = INT(N(H))
190 PRINT"■■RESULTAT"N(H)
200 FOR J = 1 TO1200 : NEXT J
210 R$ = STR$(N(H))
215 R$ = RIGHT$(R$,LEN(R$)-1)
220 FOR I = 1 TO LEN(R$)-1
230 IF MID$(R$,I,1) <>MID$(R$,I+1,1) THEN 260
240 NEXT I
250 PRINT"■■■BRAVO , JOUEUR"H:END
260 H = H + 1
270 IF H >= 3 THEN H = 1
280 GOTO 100

READY.


JOUEUR 1  ; ESSAI 1

NOMBRE      9482
          7654321

VOTRE JEU ( N ET C SOUS FORME NC )? 32


RESULTAT 8654
```

Horloge 1

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

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

LE PROGRAMME : La ligne 3Ø vous permettra d'entrer l'heure sous forme d'un entier de 6 chiffres (par exemple : 102436 signifiera 10 heures 24 minutes 36 secondes).
La ligne 4Ø utilise la variable temps TI$ du Commodore.
Les lignes 8Ø-22Ø réalisent le dessin, en décalant l'impression au fur et à mesure (à l'aide de la variable K, incrémentée en ligne 23Ø).
Les lignes 26Ø à 28Ø assurent l'affichage des secondes et le changement d'heure.

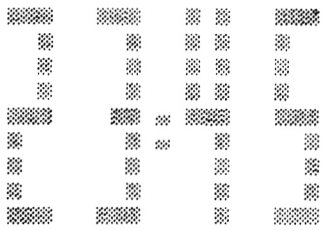
POUR JOUER : Dès le lancement par RUN, le programme se place en attente de l'heure ; on tape au prochain top sonore et on attend le top pour appuyer sur RETURN. L'heure s'affiche alors assez rapidement et l'horloge est en route. Pour l'arrêter, il faudra appuyer sur la touche RUN/STOP.
L'heure sera entrée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

```
READY.

10 REM HORLOGE 1
20 PRINT"■"
30 INPUT"HEURE ( FORME XXXXXX )";A$
40 TI$ = A$
50 PRINT"■"
60 K = 1
70 A = VAL (MID$(TI$,K,1))
80 Z = 1 : GOSUB 500
90 IF A <> 1 AND A <>4 THEN PRINTTAB(6*K-5);"■■■"
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);"■";
130 IF A <>5 AND A <>6 THEN PRINTTAB(6*K-3);"■"
140 Z = J + 4 : GOSUB 500
150 IF A = 0 OR A = 2 OR A = 6 OR A = 8 THEN PRINTTAB(6*K-5);"■";
160 IF A <>2 THEN PRINTTAB(6*K-3);"■"
170 NEXT J
180 Z = 5 : GOSUB 500
190 IF A > 1 THEN PRINTTAB(6*K-4);"■■";
200 IF K = 2 THEN PRINTTAB(11);"■":PRINTTAB(11);"■"
210 Z = 9 : GOSUB 500
220 IF A <>1 AND A <>4 AND A <>7 THEN PRINTTAB(6*K-5);"■■■"
230 K = K + 1
240 IF K <5 THEN 70
250 Z = 5 : GOSUB 500
260 PRINTTAB(28);RIGHT$(TI$,2)
270 IF RIGHT$(TI$,2) <>"00" THEN 250
280 PRINTTI$:GOTO 50
500 PRINT"■"
510 FOR L = 1 TO Z
520 PRINT:
530 NEXT L
540 RETURN

READY.
```



43

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

LE PROGRAMME : La ligne 3Ø vous permet de choisir la taille de l'échiquier.
Une variable D sera dimensionnée en conséquence, en ligne 4Ø.
Une boucle d'exploration principale I (5Ø-1ØØ) contient une sous-boucle J (7Ø-9Ø) qui réalise le contrôle de non-alignement (ligne 8Ø).
Si, au cours de l'exploration, on rencontre un alignement, on revient en arrière, en utilisant les lignes 25Ø à 29Ø et on recommence.
La boucle F (12Ø-22Ø) assure l'impression de l'échiquier, et la boucle K (17Ø-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.
Les lignes 23Ø-24Ø permettent un arrêt après l'affichage de chacune des solutions.

POUR JOUER : On lance le programme par RUN, puis on tape la dimension de l'échiquier désiré, et on fait 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 avant l'exploration de toutes les solutions, il faudra taper RUN/STOP.

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

```
10 REM REINES SUR L'ECHIQUIER
20 PRINT"■"
30 INPUT"TAILLE DE L'ECHIQUIER";S
40 DIM D(S)
50 I = 1
60 D(I) = 1
65 IF I = 1 THEN 100
70 FOR J = 1 TO I - 1
80 IF (D(I) = D(J)) OR (ABS(D(I) - D(J)) = I - J) THEN 260
90 NEXT J
100 I = I + 1 : IF I <= S THEN 60
110 PRINT"■"
120 FOR F = 1 TO S
130 Z = 0 : GOSUB 1000
140 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 PRINTTAB(2*K+1);"."
195 NEXT K
200 Z = 2*F : GOSUB 1000
203 PRINTTAB(2*D(F)+1);"■ ■"
205 Z = 20 : GOSUB 1000
210 PRINTTAB(F);RIGHT$(STR$(D(F)),1)
220 NEXT F
230 PRINT"■APPUYER SUR - SHIFT - POUR CONTINUER"
240 WAIT 653,1
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"■";
1010 FOR T = 0 TO Z
1020 PRINT
1030 NEXT T
1040 RETURN
```

```
READY.

  1 2 3 4 5 6 7

1   . . . . . .

2 . .   . . . .

3 . . . .   . .

4 . . . . . .

5 .   . . . . .

6 . . .   . . .

7 . . . . .   .

1357246
```

```
  1 2 3 4 5 6 7 8

1 ■ . . . . . .

2 . . . . ■ . . .

3 . . . . . . . ■

4 . . . . . ■ . .

5 . . ■ . . . . .

6 . . . . . . ■ .

7 . ■ . . . . . .

8 . . . ■ . . . .

15863724
```

```
APPUYER SUR - SHIFT - POUR CONTINUER
```

Répétitions

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

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

LE PROGRAMME : La ligne 3Ø appelle un sous-programme (lignes 1ØØØ à 1Ø6Ø) qui donne la règle du jeu.
La ligne 5Ø fabrique les chiffres, qui sont ensuite stockés dans une variable alphanumérique A$ (ligne 6Ø).
La ligne 7Ø permet l'impression du dernier chiffre au hasard sur la quatrième ligne de l'écran, et la durée de l'affichage est déterminée par la ligne 75.
La ligne 85 attend la réponse du joueur, qui est comparée à la réponse correcte en ligne 9Ø.
La ligne 11Ø marque les points.
La ligne 12Ø contrôle le meilleur score.
Les affichages de fin de partie sont effectués de la ligne 2ØØ à la ligne 25Ø.

POUR JOUER : Le lancement du programme par RUN provoque l'affichage de la règle du jeu, puis d'un chiffre qui disparaît. L'appareil pose alors la question VOTRE SEQUENCE ?
On lui répond en redonnant le chiffre, suivi de 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 0 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...

```
READY.


10 REM REPETITIONS
20 M = 0
25 A$=""
30 GOSUB 1000
40 H = 0
50 X = INT(RND(1)*10)
60 A$ = A$ + RIGHT$(STR$(X),1)
70 PRINT"■■■■■";TAB(RND(1)*40)X
75 FOR I = 1 TO 200 + RND(1)*300:NEXT I
80 PRINT"■"
85 INPUT "■■■■■■■■■■■■VOTRE SEQUENCE";R$
90 IF R$ <> A$ THEN 200
100 PRINT"■■EXACT . ATTENTION"
105 FOR I = 1 TO 400:NEXT I
110 H = H + 1
120 IF M <H THEN M = H
130 GOTO 50
200 PRINT"■"
210 PRINT"VOTRE SCORE EST DE"H
220 PRINT"MEILLEUR SCORE ACTUEL"M
230 INPUT"■■AUTRE PARTIE ( O/N )";R$
240 IF R$="N" THEN END
250 GOTO 25
1000 PRINT"■"
1010 PRINT"VOUS DEVEZ REPETER TOUTE LA SERIE DES"
1020 PRINT"NOMBRES APPARUS DEPUIS LE DEBUT DU JEU"
1030 PRINT:PRINT"IL Y EN A UN DE PLUS A CHAQUE FOIS"
1040 PRINT:PRINT"APPUYER SUR - SHIFT - POUR COMMENCER"
1050 WAIT653,1
1060 RETURN


READY.



VOUS DEVE REPETER TOUTE LA SERIE DES
NOMBRES APPARUS DEPUIS LE DEBUT DU JEU

IL Y EN A UN DE PLUS A CHAQUE FOIS

APPUYER SUR - SHIFT - POUR COMMENCER
```

Mots

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

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

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

POUR JOUER : On lance le programme par RUN. Une série de lettres est affichée dans le désordre, et il faut essayer de retrouver le mot correspondant.
On tape le mot supposé, puis RETURN, et une nouvelle série de lettres sera affichée, si l'on répond "0" 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 JOKER, jouer à plusieurs, tenir le compte des points, etc.

```
READY.


10 REM MOTS
20 PRINT"■"
30 DIM A$(30) : S = 0 : E = 0
40 FOR I = 1 TO 30
50 READ A$(I)
60 NEXT I
65 PRINT"■"
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 R$ = LEFT$(R$,Z-1) + "-" + RIGHT$(R$,LEN(R$)-Z)
150 NEXT I
160 PRINT"LES LETTRES SONT :";A$
170 INPUT"■■■■■■■VOTRE SOLUTION :";S$
180 IF S$ = V$ THEN S = S+1
190 E = E+1
200 INPUT"■■■■■■■■■■AUTRE ESSAI ( O/N )";R$
210 IF LEFT$(R$,1) = "O" THEN 65
220 PRINT"■■VOTRE SCORE :"S;" SUR "E;" ESSAIS"
230 END
240 DATA JAMBON,ENERGUMENE,AFFILIE,ERUDIT,INSIGNIFIANT,REALISTE
250 DATA IMPRESSIONNANT,ALLECHANT,EPIQUE,INTREPIDE,COUARD,
    ORDINATEUR
260 DATA TREVE,BANDEROLLE,SENAT,ACTUEL,CONSACRE,DIEDRE
270 DATA LIVRE,TERMINE,SIGNATURE,MALODORANT,AIMANT,ARBITRAIRE
280 DATA CHALEUR,OPPORTUN,PELURE,ALTRUISTE,RAMONEUR,NOVICE


READY.


LES LETTRES SONT :UITDER


VOTRE SOLUTION :? REDUIT


AUTRE ESSAI ( O/N )? N

VOTRE SCORE : 0  SUR  1  ESSAIS
```

Permutations

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

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

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

POUR JOUER : Après avoir tapé RUN, on obtient l'affichage de la question NOMBRE DE LETTRES (DE 1 A 12) ; on répond en frappant N, suivi de 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 AEGBDFC.
On poursuivra jusqu'à l'obtention de l'ordre désiré.
Le score sera affiché en fin de partie ; ainsi qu'une question AUTRE PARTIE (O/N) ; répondre 0 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.).

```
READY.

10 REM PERMUTATIONS
20 PRINT"■"
30 INPUT"NOMBRE DE LETTRES ( DE 1 A 12 )";N
40 IF N > 12 THEN RUN
50 A$ = "ABCDEFGHIJKL"
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 C$ = LEFT$(C$,X-1) + MID$(A$,X,1) + RIGHT$(C$,12-X)
120 A$ = LEFT$(A$,X-1) + " " + RIGHT$(A$,12-X)
130 S = 1
140 IF LEN(B$) <> N THEN 80
150 C$ = LEFT$(C$,LEN(B$))
160 PRINT"■■■■DEBUT  :";B$
170 PRINT"■FIN    :";C$
175 PRINT"■■■■■■■■■■                    "
180 PRINT"■■■■■■■■PERMUTATION NUMERO";S:INPUT"■VOTRE 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$)-L)
250 PRINT"■■■■■■■■■■■■■■RESULTAT :";B$
260 IF B$ <>C$ THEN 175
270 PRINT"■■TROUVE EN"S-1;" ESSAIS"
280 INPUT"■AUTRE PARTIE ( O/N )";R$
290 IF LEFT$(R$,1) = "O" THEN RUN

READY.


NOMBRE DE LETTRES ( DE 1 A 12 )? 7

DEBUT  :CEBGDAF

FIN    :ABCDEFG

PERMUTATION NUMERO 1

VOTRE QUANTITE? 3
```

Bridge

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

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

LE PROGRAMME : Les lignes 2Ø et 3Ø définissent un jeu de 52 cartes.
La variable B$, en ligne 4Ø permettra de donner en abrégé le nom des couleurs et aussi d'améliorer la désignation de certaines cartes (1Ø, valet, dame, roi et as).
La boucle J (5Ø-2ØØ) réalise le tirage de quatre jeux de 13 cartes.
La boucle I (6Ø-17Ø) réalise tirage et impression. La carte est choisie parmi les 52 disponibles, en ligne 8Ø.
La ligne 9Ø calcule la couleur de la carte tirée.
La ligne 1ØØ vérifie que cette carte n'a pas déjà été tirée.
Les lignes 11Ø et 12Ø s'occupent de l'impression complémentaire dans le cas où l'on tire un 10, un valet, une dame, un roi ou un as.
L'impression est réalisée par les lignes 13Ø, 14Ø et 15Ø.
La ligne 16Ø permet d'éliminer les cartes déjà tirées, et la ligne 18Ø réalise un arrêt entre chaque affichage.

POUR JOUER : On lance le programme par RUN.
Une première série de 13 cartes est affichée verticalement. Il suffira d'appuyer sur la touche SHIFT pour obtenir la deuxième série de 13 cartes. On procèdera de même pour la troisième et la quatrième.

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

```
READY.

10 REM BRIDGE
15 PRINT"■"
20 R$="234567891VDRA"
30 R$=R$+R$+R$+R$
40 B$="TRCACOPI0AAOS"
50 FOR J = 1 TO 4
60 FOR I = 1 TO 13
70 F = I - 13*INT((I-1)/13)
80 X = INT(RND(1)*LEN(R$)+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 PRINTTAB(10*J-10);MID$(R$,X,1);
140 IF T >= 9 THEN PRINTMID$(B$,T,1);
150 PRINTTAB(10*J-6);MID$(B$,S,2)
160 R$ = LEFT$(R$,X-1) + "0" + RIGHT$(R$,LEN(R$)-X)
170 NEXT I
180 PRINT"■■FAIRE - SHIFT - POUR CONTINUER":WAIT 653,1
190 PRINT"■"
200 NEXT J

READY.

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

FAIRE - SHIFT - POUR CONTINUER
```

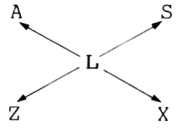
Le loup et les agneaux

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 "JEUX DES AGNEAUX" dans le cas contraire.
Les lignes 335 à 340 analysent la réponse, en fonction du joueur concerné.

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



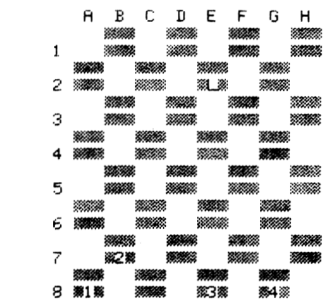
On devra veiller à ne pas sortir de l'échiquier et à ne pas venir sur l'adversaire, aucun contrôle n'ayant été prévu.

EXTENSIONS POSSIBLES : On pourra prévoir un repérage des cases, un graphisme plus soigné, des contrôles de non dépassement des limites du terrain, de non venue sur une case déjà occupée...
On pourra aussi envisager de jouer sur un damier de 10 cases sur 10 cases, ce qui augmente légèrement la difficulté du jeu pour les agneaux. Mais l'extension la plus intéressante à réaliser sera certainement la réalisation d'un programme permettant au Commodore 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 Commodore).

```
READY.

10 REM LE LOUP ET LES AGNEAUX
15 INPUT"□AGNEAUX (0) OU LOUP (1)";S
17 A$="8183858714"
20 PRINT"□"
30 D$ = "■■■■"
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 PRINTTAB(3*J);D$:PRINTI;TAB(3*J);B$:PRINTTAB(3*J);C$;
130 PRINT"□□□"
140 NEXT J
150 PRINT"■"
160 NEXT I
170 Z = 2*VAL(MID$(A$,9,1))  :GOSUB 500
180 PRINTTAB(3*VAL(MID$(A$,10,1))+1);"■■■□"
190 PRINT"■■"
200 Z = 2*VAL(MID$(A$,1,1))  :GOSUB 500
210 PRINTTAB(3*VAL(MID$(A$,2,1))+1);"1"
220 Z = 2*VAL(MID$(A$,3,1)):GOSUB 500
230 PRINTTAB(3*VAL(MID$(A$,4,1))+1);"2"
240 Z = 2*VAL(MID$(A$,5,1))  :GOSUB 500
250 PRINTTAB(3*VAL(MID$(A$,6,1))+1);"3"
260 Z = 2*VAL(MID$(A$,7,1))  :GOSUB 500
270 PRINTTAB(3*VAL(MID$(A$,8,1))+1);"4"
280 S = ABS(S-1)
290 Z = 20 : GOSUB 500
295 PRINT"□                                    □"
300 IF S = 0 THEN INPUT"JEU DU LOUP (A,S,Z,X)";R$
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 20
500 PRINT"■"
510 FOR K = 1 TO Z
520 PRINT
530 NEXT K
540 RETURN

READY.
```




```
JEU DES AGNEAUX (ND OU NG)?
```

Hanoï

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

LE JEU : Il est présenté, ici, dans une version pour un joueur. On dispose d'un certain nombre de disques (8 au plus) et l'on doit les transférer de la tour numéro 1 à la tour numéro 3, en respectant une seule règle : à aucun moment, un disque ne peut être recouvert par un disque d'un diamètre supérieur.
Le nombre de coups n'est pas comptabilisé par ce programme mais, par contre, les coups illicites seront refusés.

LE PROGRAMME : La variable DØ, en ligne 5Ø, servira à dessiner les disques.
Le nombre de disques est entré en ligne 6Ø et refusé s'il y a plus de 8 disques.
La boucle I (8Ø-1ØØ) permet de fabriquer une variable A (I) qui contient l'état de la pile sous forme numérique (en ligne 3Ø, trois variables sont prévues pour conserver l'état des trois piles).
Les socles sont dessinés par la ligne 13Ø et les trois piles par les boucles emboîtées I (15Ø-21Ø) et J (16Ø-2ØØ).
Le contrôle du coup est effectué par la ligne 27Ø et l'état des piles est modifié par les lignes 28Ø et 29Ø.

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

```
10 REM HANOI
20 PRINT"■"
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$;TAB(12);C$;TAB(24);C$
140 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(12*J-11);LEFT$(D$,VAL
    (MID$(B$,I+2,1)))
195 PRINT"■■■"
200 NEXT J
210 NEXT I
220 INPUT"■■■■■■■■■■■■■■■■■■■■■■TOUR DE DEPART";D
230 IF A(D) = 9 THEN 220
240 INPUT"■TOUR D'ARRIVEE";A:IF A=D THEN 220
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$),1) THEN 220
280 A(A) = VAL(C$+MID$(B$,LEN(B$),1))
290 A(D) = VAL(LEFT$(B$,LEN(B$)-1))
300 GOTO 110

READY.
```



```
    1           2           3

TOUR DE DEPART? 2

TOUR D'ARRIVEE? 1
```

NIVEAU 4

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

Animation

SAISIE AU VOL D'UN OU PLUSIEURS CARACTERES FRAPPES AU CLAVIER : GET

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

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

Cette possibilité est disponible par **la fonction 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 :

< (on frappe en réalité ,) pour les déplacements vers la gauche.

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

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

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

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

DEPLACEMENT D'OBJETS DE BAS EN HAUT SUR L'ECRAN : SIMULATION DU SCROLL

Le Commodore 64 réalisant automatiquement un "Scrolling" vers le haut lorsque l'on écrit sur la dernière ligne en bas de l'écran, cette propriété a été mise à profit dans de nombreux jeux : cela permet, en effet, de déplacer d'une ligne vers le haut les divers objets (textes ou graphiques) présents sur l'écran à un moment donné. La ligne du haut est 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.

Pour réaliser un **SCROLL**, il suffit dès lors d'incorporer dans le programme une ligne :

PRINT " [S] [Q] [Q] ..... [Q]

←——— 24 fois ———→

[S] correspond à HOME (renvoi du curseur en haut de l'écran, sans effacer).

[Q] correspond à CRSR bas (touche CRSR ↓).

Tirage des dés

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

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

LE PROGRAMME : La ligne 4Ø saisit le caractère frappé au clavier et 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.
Une boucle J (7Ø-1ØØ) est utilisée pour dessiner un fond gris, et une autre boucle I (6Ø-14Ø) sert au tirage (tirage en ligne 11Ø et impression en ligne 13Ø).
La ligne 16Ø offre la possibilité de refaire un tirage.

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

EXTENSIONS POSSIBLES : On pourra, bien sûr, utiliser le canevas fourni par ce programme pour la réalisation de tout jeu se servant de dés (421, YAM'S, CRAPS, etc.).

```
READY.

10 REM TIRAGE DE DES
20 PRINT"■"
30 PRINT"■NOMBRE DE DES A LANCER ";
40 GETA$ : IF A$ < "1" OR A$ > "9" THEN 30
45 N = VAL(A$)
50 PRINT"■"
60 FOR I = 1 TO N
70 FOR J = 1 TO 3
80 Z = 8 + J : GOSUB 500
90 PRINTTAB(3*I-1);"■■ ";TAB(3*I);" ";TAB(3*I+1);" "
100 NEXT J
110 D(I) = INT(RND(1)*6+1)
120 Z = 10 : GOSUB 500
130 PRINTTAB(3*I);"■";RIGHT$(STR$(D(I)),1)
140 NEXT I
150 PRINT"■■"
160 GOTO 30
500 PRINT"■"
510 FOR T =0 TO Z
520 PRINT
530 NEXT T
540 RETURN

READY.

NOMBRE DE DES A LANCER

 ■■■■■■■■■■■■■■■■■■■■
 ■3■■5■■6■■4■■5■
 ■■■■■■■■■■■■■■■■■■■■
```

Jeu de loto

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

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

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

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

EXTENSIONS POSSIBLES : On peut imaginer un programme qui pourrait tenir compte, soit de ses propres résultats, soit 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.

```
READY.

10 REM JEU DE LOTO
20 PRINT"■"
30 DIM N(7)
40 FOR I = 1 TO 7
50 N(I) = INT(RND(1)*49 + 1)
60 IF I = 1 THEN 100
70 FOR J = 1 TO I - 1
80 IF N(I) = N(J) THEN 50
90 NEXT J
100 NEXT I
110 PRINT"■■■";TAB(7);"TIRAGE DU LOTO"
115 PRINT:PRINT
120 FOR I = 1 TO 6
130 PRINTTAB(5*I+3);N(I);
140 NEXT I
145 PRINT
150 PRINT"■NUMERO COMPLEMENTAIRE";N(7)
160 PRINT"■■■VOULEZ-VOUS UN AUTRE TIRAGE ( O/N )"
170 GET A$ : IF A$ = "" THEN 170
180 IF A$ = "O" THEN RUN

READY.
```

```
       TIRAGE DU LOTO

         13   32   15   29   7    38

NUMERO COMPLEMENTAIRE 5


VOULEZ-VOUS UN AUTRE TIRAGE ( O/N )
```

La bombe

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

LE JEU : Vous êtes sensé trouver, à l'aide d'un détecteur spécialisé, un colis radioactif piégé disposé par un voyageur mal intentionné, dans une chambre d'hôtel. Seul problème, mais de taille : l'hôtel comporte un million de chambres (100*100*100).
Votre détecteur affiche, en permanence, un signal qui vous renseigne sur la proximité de la bombe, et le temps qui vous reste est affiché en permanence, décompté de 10 en 10.
Pourrez-vous découvrir la bombe avant l'explosion ?

LE PROGRAMME : Une variable numérique R, non dimensionnée, contient les trois coordonnées définissant la position du joueur.
La position de la chambre contenant le colis piégé est définie par un tirage aléatoire de 3 variables A, B, C, aux lignes 30, 40, 50.
La boucle I (70-140) dessine le détecteur.
Le signal de proximité est calculé en ligne 150 et la ligne 160 permet l'affichage des coordonnées de la chambre à découvrir, dans le cas où l'on a perdu.
La ligne 190 permet l'affichage, à la fois du signal et du temps restant à courir.
La ligne 200 réalise le décompte du temps.
La ligne 210 contrôle si l'on a trouvé ou non.
La boucle I (280-320) permet d'entrer les nouvelles coordonnées, en utilisant le sous-programme placé aux lignes 400 à 440.

POUR JOUER : Après le lancement du programme par RUN, le signal et le temps (en principe 200) sont affichés.
Le joueur est alors en position x=0, y=0, z=0, et on devra introduire successivement les trois coordonnées x, y, z de la chambre à explorer.
On fera, par exemple, 13, puis 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), ou jusqu'à l'apparition de 3 nombres (x, y, z) sans sortie du programme, ce qui signifie alors que l'on a perdu.

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

```
10 REM LA BOMBE
20 L = 100 : T = 200
30 A = INT(RND(1)*L)
40 B = INT(RND(1)*L)
50 C = INT(RND(1)*L)
60 PRINT"■"
70 FOR I = 0 TO 17
80 Z = 4 : GOSUB 500
90 PRINTTAB(I);"■■ "
100 Z = 5 : GOSUB 500
110 PRINTTAB(I);"■■ "
120 Z = 6 : GOSUB 500
130 PRINTTAB(I);"■■ "
140 NEXT I
150 S = ABS((A/L+B+C*L)-(R(1)/L+R(2)+R(3)*L))
155 S$ = LEFT$(STR$(S),7) : S = VAL(S$)
160 IF T = 0 THEN PRINT"■■PERDU : SOLUTION :"A;"  ";B;"  ";C
165 PRINT"■■"
170 Z = 5 : GOSUB 500
190 PRINTTAB(2);S;TAB(12);T
200 T = T - INT(SQR(L))
210 IF S = 0 THEN Z = 21 : GOSUB 500 : PRINT"BRAVO":END
220 PRINT"■■" : Z = 10 : GOSUB 500
230 PRINT" - X  :       "
240 Z = 12 : GOSUB 500
250 PRINT" - Y  :       "
260 Z = 14 : GOSUB 500
270 PRINT" - Z  :       "
280 FOR I = 1 TO 3
290 R$ = ""
300 GOSUB 400
310 R(I) = VAL (R$)
320 NEXT I
330 GOTO 70
400 Z = 8 + 2*I : GOSUB 500
410 PRINTTAB(7);"? ";R$;"    "
420 GET A$ : IF A$ = "" THEN 420
425 IF ASC(A$) = 13 THEN RETURN
430 R$ = R$ + A$
440 GOTO 400
500 PRINT"■"
510 FOR P = 0 TO Z
520 PRINT
530 NEXT P
540 RETURN

READY.
■■■■■■■■■■■■■■■■■■■■■■
■■ 9539.5■■■ 200■■
■■■■■■■■■■■■■■■■■■■■■■


 - X  :? 24

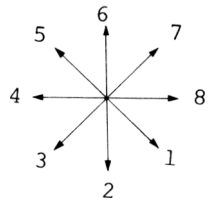
 - Y  :? 56

 - Z  :? 37
```

Ardoise magique

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

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



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

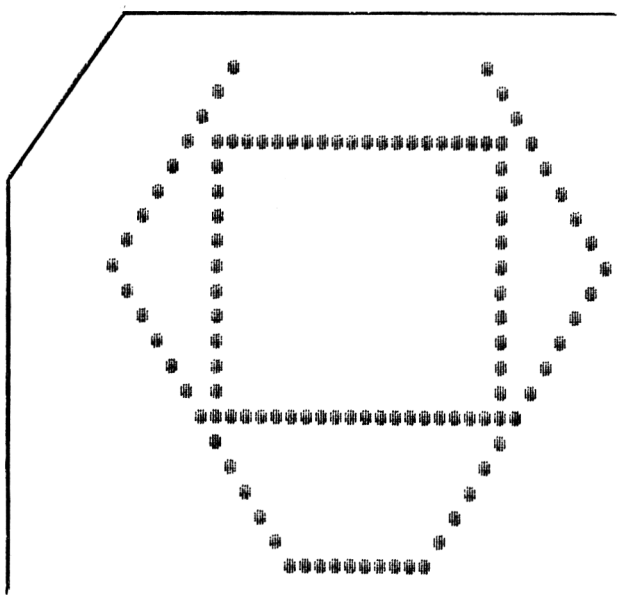
LE PROGRAMME : La ligne 3Ø contient le choix des possibilités de couleur (obtenues avec CTRL, 1 à 8 et Commodore, 1 à 8).
Les lignes 6Ø et 7Ø impriment le point élémentaire.
Les lignes 8Ø à 1Ø5 contrôlent les entrées.
La ligne 14Ø appelle l'un des quatre sous-programmes effectuant le déplacement.
La ligne 15Ø vérifie que l'on ne sort pas des limites de l'écran.
La ligne 18Ø effectue l'effacement s'il y a lieu.
Le sous-programme 3ØØ-31Ø choisit la couleur ; 4ØØ-41Ø réalise ou non l'effacement.

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

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

```
10 REM ARDOISE MAGIQUE
20 PRINT"■";
30 C$ = "■■■■■■■■■■■■■■■■■■■■■■■■■"
40 X = 10 : Y = 20 : G$ = "■"
50 S = 0
60 Z = X : GOSUB 500
70 PRINTG$;TAB(Y);"●"
80 GETA$ : IF A$ = "" THEN 80
90 IF A$ >="A" AND A$ <="P" 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<23 AND Y>0 AND Y<39)
160 IF C = 0 THEN GOSUB 5000
170 Z = A : GOSUB 500
180 IF S <> 0 THEN PRINTTAB(B);" "
190 GOTO 60
300 G$ = MID$(C$,ASC(A$)-64,1)
310 RETURN
400 S = ABS(S-1)
410 RETURN
500 PRINT"■";
510 FOR P = 0 TO Z : PRINT : NEXT P
520 RETURN
1000 Y = Y + F
2000 X = X + F
2010 RETURN
3000 GOSUB 2000
4000 Y = Y - F
4010 RETURN
5000 X = A : Y = B
5010 RETURN

READY.
```



Affichage publicitaire

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

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

LE PROGRAMME : Le texte sera introduit lettre par lettre aux lignes 5Ø à 7Ø.
L'affichage sera tronqué à 38 caractères au plus, grâce à la ligne 14Ø.
Le défilement vers la gauche est préparé à la ligne 15Ø, et le défilement vers la droite à la ligne 16Ø.
Le texte sera affiché en ligne 14Ø, et la durée de l'affichage (vitesse de défilement) est choisie par la ligne 17Ø.
La ligne 19Ø permettra de changer, si besoin est, les conditions du défilement.

POUR JOUER : On lancera le programme par RUN et, à l'affichage de la question ENTRER VOTRE TEXTE, on introduira le texte à faire défiler, suivi de RETURN.
On obtient ensuite l'affichage DEFILEMENT A DROITE (D) OU A GAUCHE (G) auquel on répondra en tapant, soit D, soit G, suivi de RETURN.
Puis à la question VITESSE, on répondra en introduisant un petit nombre pour une vitesse rapide, et un grand pour une vitesse lente (par exemple, de 0 à 40 ou plus).
Le texte commencera à défiler dès que l'on aura frappé 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 pour interrompre le programme.

EXTENSIONS POSSIBLES : On pourra introduire de la couleur, réaliser un affichage ondulant, faire remonter les lignes sur l'écran, etc.

```
READY.

10 REM AFFICHAGE PUBLICITAIRE
20 PRINT"■"
25 B$ = ""
30 PRINT"ENTRER VOTRE TEXTE"
40 PRINT
50 GET A$ : IF A$ = "" THEN 50
55 IF ASC(A$) = 13 THEN 80
60 B$ = B$ + A$
65 PRINTA$;
70 GOTO 50
80 PRINT : PRINT"DEFILEMENT A DROITE (D) OU A GAUCHE (G)": INPUT S$
90 PRINT
100 INPUT"VITESSE ( DE 0 A 50 )";V
110 V = 51 - V
120 PRINT"■"
130 T = LEN(B$)
135 PRINT"■■■■                                                    "
140 PRINT"■■■■■";" "+MID$(B$,1,38)+" "
145 PRINT"■■■■■■                                                    "
150 IF LEFT$(S$,1) = "G" THEN B$ = RIGHT$(B$,T-1)+LEFT$(B$,1)
160 IF LEFT$(S$,1) = "D" THEN B$ = RIGHT$(B$,1) + LEFT$(B$,T-1)
170 FOR J = 0 TO 10*V
180 GET C$
190 IF C$ ="0" THEN 80
200 NEXT J
210 GOTO 135

READY.


ENTRER VOTRE TEXTE

BONJOUR MONSIEUR LE PRESIDENT
DEFILEMENT A DROITE (D) OU A GAUCHE (G)
? D

VITESSE ( DE 0 A 50)? 40
```

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 Commodore vous ait auparavant touché par l'un des nombreux obus explosifs qu'il va tirer sur vous.
Vous pouvez vous déplacer dans 4 directions (à l'aide des 4 flèches), et vous devez réussir à sortir du terrain par le bord droit, sans, bien sûr, avoir été touché une seule fois.

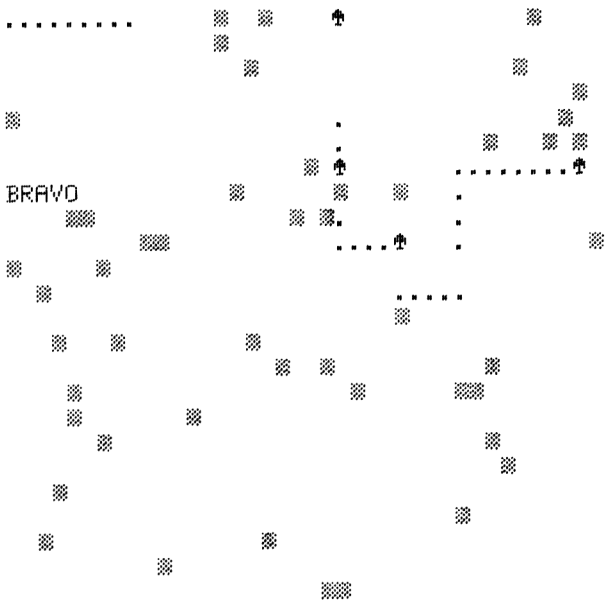
LE PROGRAMME : La ligne 3Ø définit la position de départ du joueur.
La ligne 4Ø choisit la coordonnée horizontale de la case où Commodore va tirer (liée à votre position), et la ligne 5Ø choisit la coordonnée verticale (arbitraire) de cette case.
Le déplacement du joueur est contrôlé par les lignes 11Ø et 12Ø ; sa trajectoire est tracée par la ligne 9Ø, et sa position actuelle (sur l'écran, un pique) est imprimé par la ligne 14Ø.
La ligne 15Ø regarde s'il est touché ou non.
La ligne 16Ø 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 pique.
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.

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 PRINT"■"
30 X = 10 : Y = 0
40 H = INT(Y+5-5*RND(1))
45 IF H < 0 THEN 40
50 K = INT(22*RND(1)+1)
60 Z = K : GOSUB 500
70 PRINT"■"TAB(H);"■";"■"
80 Z = X : GOSUB 500
90 PRINTTAB(Y);"."
100 GET A$
110 X = X - (A$="■") + (A$="■")
120 Y = Y - (A$=".") + (A$=",")
130 Z = X : GOSUB 500
140 PRINT"■";TAB(Y);"■"
150 IF (ABS(H-Y)+ABS(K-X)<=2) AND (Y>3) THEN PRINT"PERDU":END
160 IF Y < 38 THEN 40
170 PRINT"BRAVO"
180 END
500 PRINT"■";
510 FOR F = 0 TO Z : PRINT : NEXT F
520 RETURN

READY.
```



Chapeaux

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

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

LE PROGRAMME : Les lignes 2Ø et 3Ø contiennent le dessin des deux chapeaux (dans deux variables alphanumériques B$ et C$).
La ligne 4Ø contient, dans une autre variable A$, les symboles qui permettront de dessiner les personnages.
Les lignes 6Ø à 8Ø permettront, s'il y a lieu, l'échange des chapeaux.
La variable T de la ligne 9Ø sert à repérer s'il y a eu échange ou non.
Le dessin des deux personnages est réalisé par une boucle I (12Ø-14Ø), et leur déplacement est réalisé par les lignes 15Ø et 16Ø.
Le test, permettant ou non le changement des couvre-chefs, est effectué en ligne 18Ø.
Les lignes 2ØØ à 23Ø déterminent si le premier joueur ayant appuyé sur une touche a gagné ou perdu.

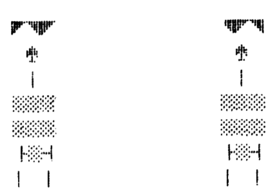
LE PROGRAMME : Lancer le programme par RUN.
Les personnages commencent immédiatement à se déplacer l'un vers l'autre.
Le joueur de gauche dispose des touches 1 et 2 et celui de droite, des touches 9 et 0.
S'il y a eu changement de chapeau, les touches à appuyer seront 1 (à gauche) et 0 (à droite).
S'il 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 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).

```
READY.


10 REM CHAPEAUX
20 B$ = "▛▼▜"
30 C$ = "▼▛▜"
40 A$ = " ♠  |  ▒▒▒▒▒▒▒▒▒ ▒▒|  |┘  └"
50 X = 0 : Y = 30 : T = 1
60 D$ = C$
70 C$ = B$
80 B$ = D$
90 T = ABS(T-1)
100 PRINT"▯"
110 PRINTTAB(X);C$
115 PRINT"▯"TAB(Y);B$
120 FOR I = 1 TO 7
130 PRINT"▥" TAB(X);MID$(A$,3*I-2,3)
135 PRINT"▩▯"TAB(Y);MID$(A$,3*I-2,3)
140 NEXT I
145 PRINT"▯"
150 X = X+1
160 Y = Y-1
170 FOR I = 1 TO 300 : NEXT I
180 IF X=Y AND RND(1) < .5 THEN 60
190 GET R$ : IF R$ ="" THEN 100
195 IF X <= Y THEN 220
200 IF (R$="1"AND T=1)OR(R$="2"AND T=0)THEN PRINT"▩▩BRAVO GAUCHE"
    :END
210 IF (R$="0"AND T=1)OR(R$="9"AND T=0)THEN PRINT"▩▩BRAVO DROITE"
    :END
220 IF R$="2" OR R$="1" THEN PRINT"▩▩PERDU GAUCHE"
230 IF R$="9" OR R$="0" THENPRINT"▩▩PERDU DROITE"


READY.
```



Combat aérien

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

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

LE PROGRAMME : La ligne 4Ø dessine l'appareil ennemi dont le déplacement aléatoire sera réalisé par les lignes 6Ø et 7Ø.
Les lignes 9Ø et 1ØØ vous permettent le pilotage de votre appareil.
La ligne 14Ø dessine le viseur de votre appareil.
Le sous-programme, gérant le tir, va de la ligne 21Ø à la ligne 27Ø, et le test de fin de partie est effectué en ligne 25Ø.
La ligne 15Ø regarde si vous avez tiré ou non.

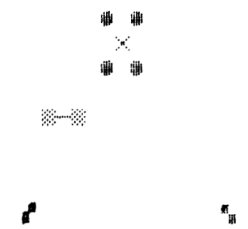
POUR JOUER : Dès que le programme a été lancé par l'ordre RUN, vous devez manoeuvrer pour essayer de faire en sorte que l'appareil ennemi soit cadré dans votre viseur.
Pour cela, vous disposez des flèches de déplacement (touches >, <, ↑↓, ⇄) et vous pouvez tirer à tout moment en pressant la touche Ø.
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, ou exceptionnellement, par le fait que l'appareil ennemi sort des limites de l'écran.
On peut aussi arrêter à tout moment en frappant RUN/STOP.

EXTENSIONS POSSIBLES : Il sera assez facile de faire un affichage au graphisme plus soigné, utilisant la couleur. On pourrait aussi envisager de stabiliser la position de l'avion ennemi pendant le tir (il suffit de modifier la ligne 15Ø).
On peut aussi introduire plusieurs avions, compter les points ou le temps, tenir compte de sa propre réserve de carburant, etc.

```
READY.


10 REM COMBAT AERIEN
20 PRINT"■"
25 A = 5 : B = 15
27 N = 21
30 Z = A : GOSUB 500
40 PRINT"■"TAB(B);"▒▬▒"
45 PRINT"■";
50 U = A : V = B
60 A = A + 2*INT(RND(1)*2) - 1
70 B = B + 2*INT(RND(1)*2) - 1
80 GET A$
90 A = A - 2*(A$="■") + 2*(A$="■")
100 B = B - 2*(A$="■") + 2*(A$="■")
110 Z = U : GOSUB 500
120 PRINTTAB(V);"   "
130 Z = 9 : GOSUB 500
140 PRINTTAB(18);"● ●":PRINTTAB(18);" ✕ ":PRINTTAB(18);"● ●"
150 IF A$ = "0" OR N <21 THEN GOSUB 200
160 GOTO 30
200 Z = N : GOSUB 500
210 PRINTTAB(29-N);"■■";TAB(N+9);"■"
220 PRINT"                                        ";
230 N = N - 1
240 IF N > 9 THEN RETURN
250 IF (A=9 OR A=10 OR A=11) AND (B=18 OR B=19 OR B=20)
    THEN PRINT"GAGNE...":END
270 GOTO 27
500 PRINT"■";
510 FOR F = 0 TO Z : PRINT : NEXT F
520 RETURN

READY.
```



Dessin de dés

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

LE JEU : Le programme proposé est surtout intéressant par son côté graphique puisqu'il permet de représenter, de façon très convenable, de 1 à 5 dés sur l'écran, en utilisant une technique de dessin un peu particulière.
Chaque réponse à la question NOMBRE DE DES provoque immédiatement un nouveau lancer du nombre de dés indiqué.

LE PROGRAMME : La ligne 3Ø permet de contrôler s'il y a eu ou non action sur une touche de 1 à 5.
Les lignes 6Ø et 7Ø initialisent les deux variables de dessin des dés.
La boucle I (8Ø-19Ø) et les lignes 9Ø et 1ØØ dessinent le cadre du dé.
La ligne 11Ø réalise le tirage, et la boucle J (12Ø-18Ø) dessine les points en relation avec le tirage réalisé (en appelant les variables A$ et B$).

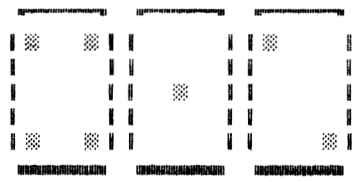
POUR JOUER : On lance le programme par l'ordre RUN, et, à la question NOMBRE DE DES, on répond en frappant une touche de 1 à 5. Les dés demandés sont alors représentés, et la question NOMBRE DE DES est de nouveau posée. Il suffira de taper RUN/STOP pour sortir du programme.

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

```
READY.


10 REM DESSIN DE DES
15 PRINT"■"
20 PRINT"■■■NOMBRE DE DES"
30 GET N$ : IF N$ = "" THEN 30
40 N = VAL(N$) : IF N<1 OR N>5 THEN RUN
50 PRINT"■"
60 A$ ="        ▒  ▒         ▒▒▒    ▒"
70 B$ = "1213143245155255555"
80 FOR I = 1 TO N
90 PRINT"■■■"TAB(8*I-8);"■▄▄▄▄▄▄▄▄▄"
100 PRINT"■■■■■■■■■■"TAB(8*I-8);"■▀▀▀▀▀▀▀▀▀▀"
110 X = INT(RND(1)*6+1)
120 FOR J = 1 TO 5
130 IF J <= 3 THEN S = 5*VAL(MID$(B$,3*(X-1)+J,1)) - 4
140 PRINT"■":FOR Z = 0 TO 2*J-1 : PRINT :NEXT Z
150 IF J <= 3 THEN PRINTTAB(8*I-7);MID$(A$,S,5)
160 PRINT"■":FOR Z = 0 TO J : PRINT : NEXT Z
170 PRINTTAB(8*I-8);"▌";TAB(8*I-2);"■▐"
180 NEXT J
190 NEXT I
200 GOTO 20

READY.
```



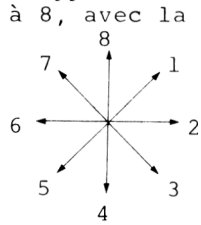
Le jeu du missile

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

LE JEU : Le Commodore est en alerte : vous êtes aux commandes d'un avion chargé d'une mission d'espionnage sur son territoire ; votre but est de survoler une base ennemie (représentée par un H inversé), et de revenir sain et sauf sur votre base ; mais, dès que le Commodore 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 5Ø dessine la base ennemie à survoler (sous forme d'un H inversé).
La ligne 7Ø dessine votre base de départ.
Les lignes 9Ø et 11Ø choisissent l'instant de départ du missile.
La ligne 14Ø dessine votre avion.
Les lignes 16Ø et 17Ø définissent son mouvement suivant 8 directions, en utilisant les sous-programmes en lignes 1ØØØ, 2ØØØ, 3ØØØ et 4ØØØ.
La ligne 18Ø assure le départ du missile qui est dessiné par la ligne 2ØØ et dont le déplacement est géré par les lignes 21Ø et 22Ø.
Les lignes 23Ø et 24Ø 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 :



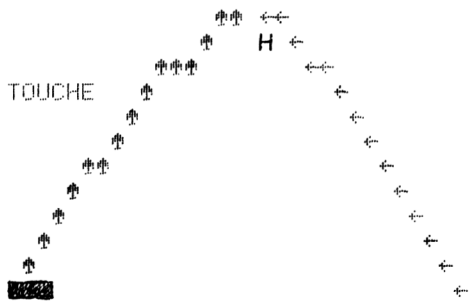
Vous devez lui faire contourner le territoire ennemi (si vous, ou le missile, touchez ce territoire, une deuxième partie commence).
Si le missile vous détruit, vous obtenez un message TOUCHE, et la partie s'arrête.
Taper RUN/STOP pour arrêter la partie.

EXTENSIONS POSSIBLES : Il sera sans doute nécessaire de rajouter divers contrôles dans le programme (l'avion a-t-il survolé la base ennemie, est-il revenu à sa propre base...).
On peut aussi prévoir divers niveaux de jeu

en faisant, par exemple, varier la vitesse du missile, prévoir l'utilisation de leurres par l'avion, l'explosion du missile au bout d'un certain temps, etc.

```
10 REM LE JEU DU MISSILE
20 PRINT"■"
30 M = 0
40 Z = 5 : GOSUB 500
50 PRINTTAB(17);"■H"
60 Z = 15 : GOSUB 500
70 PRINT"■   "
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 PRINTTAB(Y);"↑"
150 GET A$ : IF A$ = "" THEN 150
160 F = -2*(VAL(A$)<5)-1
170 ON VAL(A$)+2*(F-1) GOSUB 1000,2000,3000,4000
180 IF S <>0 THEN 110
190 Z = U : GOSUB 500
200 PRINTTAB(V);"←"
210 U = U - SGN(U-X)
220 V = V - SGN(V-Y)
230 IF U = X AND V = Y THEN PRINT"■■TOUCHE" : END
240 IF U = 5 AND V = 17 THEN RUN
250 IF X <= 5 AND Y >= 17 THEN M = 1
260 IF M = 1 AND X = 15 AND Y <= 3 THEN PRINT"■■MISSION ACCOMPLIE"
    :END
270 GOTO 130
500 PRINT"■"
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

READY.
```



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 Commodore qui va afficher des groupes de 3 à 6 lettres pendant un temps relativement court. Si l'un des joueurs estime qu'il peut réussir à faire un mot avec les lettres proposées, il doit taper rapidement sur sa touche de jeu (1 ou 0).
Le mot devra alors être accepté ou refusé par l'adversaire qui répondra JUSTE ou FAUX.
Chaque mot juste rapporte 1 point, chaque mot faux enlève 1 point.

LE PROGRAMME : La ligne 3Ø contient une liste V$ des voyelles courantes, et la ligne 4Ø une liste C$ des consonnes courantes.
Il sera possible de modifier ces deux listes pour augmenter, par exemple, la fréquence d'apparition de certaines lettres (on pourra ainsi mettre deux E dans V$ ou deux R dans C$, etc).
La ligne 5Ø définit la longueur du mot (en principe, de 3 à 6 lettres, mais on peut également modifier cette longueur).
La boucle I (9Ø-14Ø) permet la confection d'un "mot" en prenant une lettre, soit dans l'alphabet complet, soit dans la liste C$ (ligne 11Ø), soit dans la liste V$ (ligne 12Ø).
La ligne 16Ø temporise l'apparition du résultat.
La ligne 17Ø gère la fin de partie, avec affichage du score.
La ligne 18Ø regarde si l'un des deux joueurs a réagi.
Enfin, les lignes 21Ø et 22Ø comptabilisent les points de chaque joueur.

POUR JOUER : On lance le programme en tapant RUN.
La première série de lettres est alors affichée sur l'écran. Le joueur de gauche devra taper 1 s'il voit qu'il peut réaliser un mot, celui de droite devra taper Ø.
Si rien n'a été frappé, une autre série de lettres sera affichée après peu de temps.

Si l'un des joueurs a frappé 0 ou 1, la question (J)USTE ou (F)AUX est alors affichée. On répond J si le mot est correct, F si le mot est incorrect, suivi de RETURN. Les points sont alors calculés par Commodore, 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.

```
READY.

10 REM LE SINGE
20 A = 0 : B = 0
30 V$ = "AEIOU"
40 C$ = "BCDGLMNPRSTV"
50 X = INT(RND(1)*4 + 3)
60 M$ = ""
70 PRINT"■"
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$(C$,INT(LEN(C$)*RND(1) + 1),1)
120 IF RND(1) < 0.5 THEN E$ = MID$(V$,INT(LEN(V$)*RND(1) +1),1)
130 M$ = M$ + E$
140 NEXT I
150 PRINT"■■■■■■";M$
160 FOR J = 1 TO 500 : NEXT J
165 GET A$
170 IF A$ = "P" THEN PRINT"■■SCORE DE A ";A;" SCORE DE B ";B : END
180 IF A$ <>"0" AND A$ <>"1" THEN 50
190 INPUT"■■ (J)USTE OU (F)AUX ";R$
200 PRINT"■                                   "
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

READY.



EOAUVU

 (J)USTE OU (F)AUX ? F
```

Le jeu du 15

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

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

LE PROGRAMME : La boucle I (5Ø-8Ø) dessine le terrain de jeu, et les lignes 9Ø à 11Ø s'occupent de la gestion du tour des joueurs (numéro 1 ou 2 pour chaque joueur, et jeton O ou X correspondant).
La ligne 12Ø attend le coup joué.
La boucle I (13Ø-15Ø) vérifie la validité du coup.
La ligne 16Ø teste la fin de partie, en cas de match nul, et la ligne 17Ø refuse les coups non conformes (avec retour à la situation initiale, ligne 9Ø).
Les lignes 22Ø et 23Ø s'occupent de l'affichage du coup, et la ligne 24Ø 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 27Ø à la ligne 34Ø : 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 35Ø.

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.
On pourra modifier la longueur des mots, la fréquence d'apparition des lettres, le décompte des points, la règle du jeu, etc.

```
READY.


10 REM LE JEU DU 15
15 PRINT"■"
20 F = 0
30 R$ = "123456789"
40 O$ = "" : X$ = ""
50 FOR I = 0 TO 24 STEP 3
60 PRINT"■■";TAB(I+1);"┌──┐":PRINTTAB(I+1);"|  |":
   PRINTTAB(I+1);"└──┘"
70 PRINTTAB(I+1);(I+3)/3
80 NEXT I
90 PRINT"■■■■■■■■■JOUEUR";F+1;
100 IF F = 1 THEN PRINT" : O"
110 IF F = 0 THEN PRINT" : X"
120 GET A$ : IF A$ < "1" OR A$ > "9" THEN 120
130 FOR I = 1 TO LEN(R$)
140 IF A$ = MID$(R$,I,1) THEN GOSUB 200
150 NEXT I
160 IF VAL (R$) = 0 THEN PRINT"■■■■■■■■■■■■■■■■■C'EST FINI" : END
170 GOTO 90
200 IF F = 1 THEN O$ = O$ + A$
210 IF F = 0 THEN X$ = X$ + A$
220 IF F = 1 THEN PRINT"■■■";TAB(3*I-1);"O"
230 IF F = 0 THEN PRINT"■■■";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 PRINT"■■■■■■■■■■BRAVO , JOUEUR O"
410 IF F = 0 THEN PRINT"■■■■■■■■■■BRAVO , JOUEUR X"


READY.
```

```
┌──┬──┬──┬──┬──┬──┬──┬──┬──┐
|  | X|  | X|  |  |  | O| O|  |
└──┴──┴──┴──┴──┴──┴──┴──┴──┘
  1  2  3  4  5  6  7  8  9

JOUEUR 2  : O
```

Adresse

REFERENCES : Difficulté de programmation : difficile.
Catégorie du jeu : réflexe (pour un 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 4Ø et 5Ø 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 Commodore est choisi aux lignes 6Ø et 7Ø, et le chiffre est tiré en ligne 8Ø.
La ligne 1ØØ réalise l'impression, et la ligne 11Ø détermine la durée d'affichage.
La ligne 13Ø définit la durée du jeu, et la fin de partie est traitée par la ligne 16Ø.
La boucle I (21Ø-25Ø) réalise le dessin grand format du chiffre frappé par le joueur.
Le contrôle de l'exactitude du résultat est effectué par la ligne 26Ø.

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 suffisament vite, tant qu'il reste à l'affichage, et le retaper.
Le chiffre qu'il a tapé se dessinera alors en haut et 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.

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.

```
READY.


10 REM ADRESSE
15 PRINT"■"
20 T = 100
30 S = 0
40 C$ = "▒▒▒▒▒    ▒▒▒ ▒"
50 B$ = "144413333313121113131441331213112141133331414114131"
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 PRINTTAB(Y);N
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"■"
180 PRINT"SCORE"S;" / 100"
190 END
200 U = VAL(A$)
210 FOR I = 1 TO 5
220 A = 3*VAL(MID$(B$,5*U+I,1)) - 2
230 Z = I : GOSUB 500
240 PRINTTAB(2);MID$(C$,A,3)
250 NEXT I
260 IF U = N THEN S = S + 10
270 GOTO 130
500 PRINT"▒"
510 FOR P = 0 TO Z : PRINT : NEXT P
520 RETURN

READY.
```

```
▒▒▒
▒
▒▒▒
  ▒
▒▒▒
```

Sous-marin

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

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

LE PROGRAMME : La ligne 7Ø permet au Commodore de choisir la profondeur du sous-marin.
La ligne 1ØØ dessine votre bâtiment.
Les lignes 12Ø et 13Ø affichent le score et dessinent la surface de l'eau.
La ligne 15Ø dessine le sous-marin.
Le tir est effectué par un programme spécifique placé aux lignes 24Ø à 3ØØ.
Le déplacement (très rapide) de votre navire est réalisé par les lignes 2ØØ et 21Ø.
Le test, permettant de savoir si l'on a touché ou non, est effectué en ligne 17Ø.

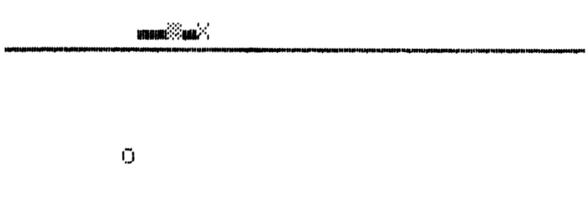
POUR JOUER : Taper RUN pour commencer la partie.
Le bâtiment sera déplacé grâce aux flèches droite et gauche (CRSR > ou CRSR <).
La touche Ø autorise le tir.
Le navire, se déplaçant trois fois plus vite que les sous-marins, il sera possible de toucher plusieurs fois le même bâtiment.
Pour terminer la partie, il conviendra de faire RUN/STOP.

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...

```
READY.


10 REM SOUS-MARIN
20 P = 0 : C = 15
30 PRINT"□"
40 Z = S : GOSUB 500
50 PRINT"       "
60 T = 27
70 S = INT(RND(1)*6 + 4)
80 P = P + 0.01
90 X = 1
100 PRINT"■■■"TAB(C);"    ■▀▀ ▀X■□    "
110 Z = 18 : GOSUB 500
120 PRINT"SCORE :"P
130 PRINT"■■■■■□▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔□"
140 Z = S : GOSUB 500
150 PRINT"■";TAB(T);"▄▄▄ □"
160 IF X = 1 THEN Y = C + 2*X
170 IF S = X AND T+2 = Y THEN P=P+1:Z=S:GOSUB500:
    PRINTTAB(T);"***"
180 T = T-1 : IF P = 99 THEN Z = 18 : GOSUB 500
    : PRINT"C'EST FINI":END
190 IF T < 0 THEN 40
200 GET A$
210 C = C - 3*((A$="." AND C < 27)-(A$="," AND C>0))
230 IF A$ <> "■" AND X = 1 THEN 100
240 Z = X : GOSUB 500
250 PRINTTAB(Y);" "
260 X = X + 1
270 IF X >S THEN 90
280 Z = X : GOSUB 500
290 PRINTTAB(Y);"O"
300 GOTO 100
500 PRINT"■"
510 FOR L = 0 TO Z : PRINT : NEXT L
520 RETURN

READY.
```



SCORE : 1.06

Zombies

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

LE JEU : Les ZOMBIES sont des êtres malfaisants qui vont essayer de vous détruire, mais vous allez vous aussi, essayer de les utiliser à votre profit, pour débarrasser votre espace vital des trous noirs qui l'obscurcissent. Sur l'écran, est tracé un cadre qui définit votre espace vital. Dans ce cadre, vont apparaître de façon aléatoire, les ZOMBIES (ce sont des Z inversés), les trous noirs (des carrés noirs) et le joueur (un 0). Les ZOMBIES convergent vers le joueur pour essayer de le capturer, mais celui-ci se déplace plus rapidement qu'eux, et il va essayer de leur faire avaler tous les trous noirs, sans se faire prendre. La partie se termine, soit avec la disparition du joueur, soit par l'élimination de tous les trous noirs.

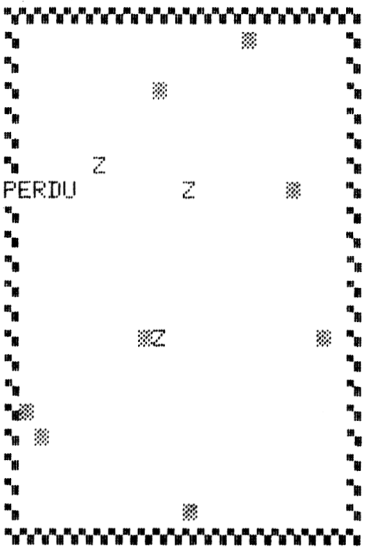
LE PROGRAMME : La position des ZOMBIES est définie par un tableau Z non dimensionné (c'est inutile en-dessous de 10).
Les lignes 3Ø à 14Ø (boucle I) dessinent le cadre, les ZOMBIES et les trous noirs.
La ligne 15Ø définit la position initiale du joueur.
Une deuxième boucle I (16Ø-24Ø) réalîse le déplacement des Z.
Les lignes 3ØØ à 34Ø contrôlent le déplacement du joueur.

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

EXTENSIONS POSSIBLES : Il serait intéressant d'agrandir le terrain de jeu, d'augmenter le nombre de ZOMBIES ou de trous noirs, d'améliorer les contrôles (pratiquement inexistants), de placer le joueur de façon plus aléatoire, éventuellement de modifier la règle (par exemple, faire disparaître les ZOMBIES dans les trous noirs).
On pourra aussi, bien sûr, améliorer la qualité du dessin et introduire la couleur.

```
10 REM ZOMBIES
20 PRINT"■"
30 FOR I = 0 TO 21
40 Z = 0 : GOSUB 500
50 PRINT"■"TAB(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 PRINTTAB(I);"■■"
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)
160 FOR I = 1 TO 4
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)
210 Z = Z(I) : GOSUB 500
220 PRINT"■";TAB(Z(I+4));"Z"
230 IF Z(I) = S AND Z(I+4) = K THEN PRINT"PERDU":END
240 NEXT I
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$=".") - (A$=","))
320 IF K <= 0 OR K >= 23 THEN PRINT"PERDU" : END
330 S = S - 2*((A$="■") - (A$="■"))
340 IF S <= 0 OR S >= 21 THEN PRINT"PERDU" : END
350 GOTO 160
500 PRINT"■"
510 FOR F = 0 TO Z : PRINT : NEXT F
520 RETURN
```

```
READY.
```



Diago

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

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

LE PROGRAMME : Le nombre de points noirs sera choisi au hasard, en ligne 3Ø.
Le temps est décompté par les lignes 4Ø à 8Ø.
Les points sont imprimés et la ligne diagonale est fabriquée par la ligne 1Ø5.
La pression de la touche 0 est détectée en ligne 13Ø, et la ligne 15Ø contrôle la fin de partie (aucun réflexe).
Les affichages de fin sont exécutés par les lignes 17Ø à 2ØØ.

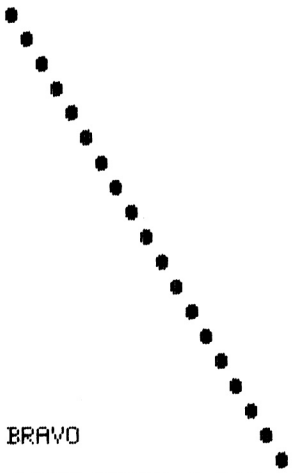
POUR JOUER : On lancera l'exécution par RUN.
On obtient alors l'affichage de la phrase LE BON REFLEXE EST A ... qui indique le nombre de points gagnants.
Le temps sera ensuite décompté sous la forme ATTENTION : IL RESTE ... SECONDES qui indique le temps restant avant l'apparition de la diagonale.
A l'affichage de 0, la diagonale apparaît et commence à se déplacer sur l'écran.
Si l'on estime que le nombre de points réclamés a été atteint, on devra presser la touche 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...).

```
READY.

10 REM DIAGO
20 PRINT"■"
30 X = INT(RND(1)*18 + 3)
40 TI$ = "000000"
50 PRINT"LE BON REFLEXE EST A"X
60 A = 10 - VAL(TI$):A$ = STR$(A)
70 PRINT"■■■■■■■ATTENTION : IL RESTE";A$;" SECONDES  "
80 IF A <>0 THEN 60
90 PRINT"■"
100 U = 0
105 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■";TAB(U);"■■■"
110 FOR I = 1 TO 30 : NEXT I
120 U = U + 1
130 GET A$
140 IF A$ = "■" THEN 170
150 IF U > 22 THEN PRINT"■■PERDU" : END
160 GOTO 105
170 IF U <>X THEN PRINT"■■PERDU" : END
180 PRINT"■■BRAVO"
190 INPUT"■AUTRE ESSAI ( O/N )";R$
200 IF LEFT$(R$,1) = "O" THEN RUN

READY.
```



```
BRAVO

AUTRE ESSAI ( O/N )? N
```

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 successsivement dans les divisions par 3, 5 et 7 d'un entier compris entre 1 et 100, et le Commodore calcule et affiche cet entier. Le procédé retenu peut d'ailleurs être généralisé (à vous de trouver comment).

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

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

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

```
READY.


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


READY.


VOUS DEVEZ ENTRER LES TROIS RESTES

DANS LES DIVISIONS PAR 3 , 5 ET 7

D'UN ENTIER INFERIEUR A 100




RESTE DANS LA DIVISION PAR  7


RESTE PAR 3: 2 . PAR 5: 1 . PAR 7: 4

LA SOLUTION EST : 11

VOULEZ-VOUS UN AUTRE ESSAI ( O/N ) ?
```

Chasse au sous-marin

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

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

LE PROGRAMME : La boucle I (4Ø-12Ø) affiche le terrain.
La ligne 14Ø définit la position initiale du sous-marin ; la ligne 15Ø permet son déplacement, contrôlé par la ligne 16Ø (on reste dans les limites du terrain).
La ligne 18Ø affiche votre position.
Les lignes 23Ø à 25Ø contrôlent votre déplacement.
La distance est calculée en ligne 26Ø, affichée en ligne 28Ø et testée en lignes 29Ø et 3ØØ.
Le test de fin de partie est effectuée à la ligne 42Ø.

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

EXTENSIONS POSSIBLES : Il est possible de prévoir une amélioration du graphique, d'améliorer les possibilités de déplacement du chasseur (un peu lent, dans cette version), de prévoir des parcours plus sophistiqués pour le sous-marin, de mettre un compteur de temps...

```
10 REM CHASSE AU SOUS-MARIN
20 PRINT"□"
30 INPUT"DIMENSION DE LA GRILLE";H
35 PRINT"□";
40 FOR I = 0 TO H
50 Z = 0 : GOSUB 500
60 PRINTTAB(2*I+3);I
70 Z = 2*I + 1 : GOSUB 500
80 PRINTI;
90 FOR J = 0 TO H
100 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
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 PRINTTAB(2*Y+4);"■■ ■□";
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$="," AND Y<H) + (A$="," AND Y>0)
250 X = X - (A$="■" AND X <H) + (A$="■" AND X>0)
260 D = INT(SQR(ABS(X-V)↑2 + ABS(Y-U)↑2))
270 Z = 10 : GOSUB 500
280 PRINTTAB(29)"DIST. ";STR$(D)+" ";
290 IF D = 0 THEN 400
300 IF D < 3 THEN 150
310 GOTO 170
400 Z = 12 : GOSUB 500
410 INPUT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■PROF.";D
420 IF D = INT(RND(1)*3 + 2) THEN PRINT"BRAVO" : END
430 GOTO 140
500 PRINT"■";
510 FOR P = 0 TO Z : PRINT : NEXT P
520 RETURN
```

```
READY.
    0 1 2 3 4 5 6 7 8
 0  . . . . . . . . .
 1  . . . . . . . . .
 2  . . . . . . . . .
 3  . . . . . . . . .
 4  . . . . . . . . .
                        DIST.  2
 5  . . . . . . . . .
 6  . . . . . . . . .
 7  . . . . . . . . .
 8  . . . . . ■ . . .
```

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, et la colonne dans laquelle se trouve le point indique le caractère correspondant, 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 30, contiendra la liste des touches concernées.
Les valeurs des touches sont entrées de la ligne 40 à la ligne 90, par appel d'un sous-programme situé en 400.
Les boucles emboîtées J (100-150) et I (110-140) dessinent le clavier.
La ligne 160 définit la position de la balle, imprimée par la ligne 180.
La touche frappée est testée aux lignes 200, 210 et 220.
Le nombre d'essais est totalisé en ligne 230, et la balle est effacée par la ligne 240.
Les affichages de fin sont réalisés aux lignes 300 à 340.

POUR JOUER : On lance le programme par RUN. L'on dispose alors de quelques secondes pour frapper la touche qui occupe la colonne au-dessus de la balle, puis une autre colonne est affichée, etc.
La durée de la partie n'est pas limitée ; il suffit de taper 0 pour obtenir l'affichage du score, et du temps total.
Faire RUN pour une autre partie.

EXTENSIONS POSSIBLES : On pourra envisager de faire clignoter les lettres, ou de les afficher en couleur, au lieu d'indiquer la colonne par une balle. On pourra afficher les lettres devinées 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 PRINT"■"
25 S = 0 : TI$="000000" : P = 0
30 DIM A$(10,3)
40 S$ = "QWERTYUIOP" : J = 1
50 GOSUB 400
60 S$ = "ASDFGHJKL:" : J = 2
70 GOSUB 400
80 S$ = "ZXCVBNM,./" : J = 3
90 GOSUB 400
100 FOR J = 1 TO 3
110 FOR I = 1 TO 10
120 Z = 3*J : GOSUB 500
130 PRINTTAB(3*I+J);A$(I,J)
140 NEXT I
150 NEXT J
160 X = INT(RND(1)*10+1) : Y = INT(RND(1)*3+1)
170 Z = 11 : GOSUB 500
180 PRINT"■■"TAB(3*X+Y);"●"
190 FOR F = 1 TO 50+70*RND(1) : GET A$
200 IF A$ = "" THEN NEXT F
210 IF A$ = A$(X,Y) THEN S = S + 1
220 IF A$ = "0" THEN 300
230 P = P + 1
240 PRINT"■"TAB(3*X+Y);" "
250 GOTO 160
300 Z = 17 : GOSUB 500
310 PRINTS;" TROUVES JUSTES SUR";P;" ESSAIS ."
320 PRINT"■TEMPS TOTAL ";LEFT$(TI$,2);" HEURES , ";MID$(TI$,3,2);
    " MN. , ";
330 PRINTRIGHT$(TI$,2);" SEC."
340 END
400 FOR I = 1 TO 10 : A$(I,J)=MID$(S$,I,1):NEXT I
410 RETURN
500 PRINT"■";
510 FOR P = 0 TO Z : PRINT : NEXT P
520 RETURN

READY.
```

```
    Q  W  E  R  T  Y  U  I  O  P

     A  S  D  F  G  H  J  K  L  :

      Z  X  C  V  B  N  M  ,  .  /

                ●


 0  TROUVES JUSTES SUR 18  ESSAIS .
TEMPS TOTAL 00 HEURES , 00 MN. , 48 SEC.
```

Calculatrice

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

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

LE PROGRAMME : Les variables B$ et C$, des lignes 2Ø et 3Ø, contiennent les informations codées qui permettront le dessin des chiffres en grand format.
Les lignes 6Ø à 125 s'occupent de la représentation, en grand format, des chiffres sur l'écran, au fur et à mesure des entrées.
Le point décimal et le signe - sont gérés par les lignes 17Ø et 18Ø.
Le nombre total de symboles entrés est contrôlé par la ligne 19Ø.
Le résultat est inscrit grâce à un programme fonctionnant en automatique qui se trouve aux lignes 22Ø à 27Ø (l'automatisme est géré par la ligne 2ØØ, dans le programme principal).
Les sous-programmes placés aux lignes 3ØØ-31Ø 35Ø-36Ø, 4ØØ-41Ø et 45Ø-46Ø effectuent l'opération indiquée, selon la valeur de Z (ligne 227).
Chaque résultat est mémorisé en A (F-1) (ligne 229).

POUR JOUER : On devra lancer le programme par RUN.
Puis, on tape les chiffres du premier nombre à entrer qui vont s'afficher un par un, au fur et à mesure (on ne pourra entrer plus de 8 symboles, y compris le signe - et le point décimal). La frappe d'un signe opératoire laisse l'affichage intact, mais l'introduction du prochain chiffre ou symbole provoque l'effacement immédiat et l'affichage du nouveau symbole. Il est à remarquer que les chiffres s'introduisent de gauche à droite, sens de l'écriture usuelle.
L'appareil accepte, comme signes opératoires les symboles +, -, *, / ; le signe = provoquera l'affichage du résultat. Les opérations sont enchaînées automatiquement, le résultat précédent étant pris en compte comme opérateur du calcul suivant.
Pour faire une autre opération, il faudra taper RUN/STOP, puis RUN.
Taper RUN/STOP pour sortir du programme.

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

```
10 REM CALCULATRICE
15 PRINT"□"
20 C$ = "▒▒▒▒▒    ▒▒▒ ▒"
30 B$ = "144413333313121131314413312131121411333314141141 31"
40 F = 1 : L = 0
50 N = 0 : T = 0 : K = 0 :R$=""
60 GET A$ : IF A$ = "" THEN 60
65 IF K = 1 THEN N = 0 : R$ = "" : K = 0
70 IF A$ = "=" THEN 220
75 X = (1 AND A$ ="+")+(2 AND A$="-")+(3 AND A$="*")+(4 AND A$="/")
80 R$ = R$ + A$
90 IF X <>0 THEN 220
95 IF R$ = "-" THEN A$ = R$ : GOTO 170
100 IF A$ < "." OR A$ > "9" OR A$ = "/" THEN 60
110 IF N = 0 THEN PRINT"□"
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$)=0 THEN A(F) = 1 : GOTO 225
224 A(F) = VAL(R$)
225 IF X <>0 THEN Z = X
227 ON Z GOSUB 300,350,400,450
228 IF A$ <>"=" THEN F = F + 1
229 A(F-1) = VAL(R$)
230 IF VAL(R$) = 0 THEN A(F-1) = A(F)
233 R$ = STR$(VAL(R$))
235 IF LEFT$(R$,1) = " " THEN R$ = RIGHT$(R$ , LEN(R$)-1)
250 FOR T = 1 TO LEN(R$)
260 A$ = MID$(R$,T,1)
270 GOTO 110
300 R$ = STR$(A(F-1) + A(F)) : A(F) = VAL(R$) : L = 1
310 RETURN
350 IF F <>1 THEN R$ = STR$(A(F-1) - A(F)) : A(F) = VAL(R$) : L = 1
360 RETURN
400 IF F <> 1 THEN R$ = STR$(A(F-1) * A(F)) : A(F) = VAL(R$) : L =
410 RETURN
450 IF F <>1 THEN R$ = STR$(A(F-1) / A(F)) : A(F) = VAL(R$) :
460 RETURN

READY.
```



NIVEAU 5

N°	Nom du programme	Thème et particularités
1	**Smokey**	Fumez sans risque, avec ce programme original pour un joueur.
2	**Course de voiture**	Un jeu de réflexe pas trop difficile, pour un joueur.
3	**Echecs**	Un programme assez simplifié, pour deux joueurs.
4	**Danger spatial**	Serez-vous assez adroit pour éviter toutes les météorites.
5	**Tic Tac Toe**	Un classique, mais pas sans intérêt (deux joueurs).
6	**Horloge 2**	Utilisation du générateur de caractères de l'ordinateur.
7	**Dames**	Pour deux joueurs : classique.
8	**Anagrammes**	Un bon jeu de réflexes , pour deux joueurs.
9	**Oeufs**	Ne laissez pas faire la pondeuse diabolique (un joueur).
10	**Affichage tournant (vertical)**	Une utilisation intéressante de votre appareil.
11	**Quatre à quatre**	Un jeu de réflexion un peu difficile, pour deux joueurs.
12	**Taquin de lettres**	Un proverbe est caché dedans : à vous de le découvrir.
13	**Cibles**	Un peu d'adresse, beaucoup de chance : très visuel.
14	**Gouttes**	Un véritable gag visuel. A exploiter.
15	**Obstacles**	Beaucoup d'obstacles à éviter dans ce jeu d'adresse pour un joueur.
16	**Captures dans l'espace**	Votre mission serait facile, sans ces damnés météorites.
17	**Anomalie**	Exercer votre perspicacité dans la reconnaissance des symboles.
18	**Plusieurs en un**	Un terrain de jeu et les pions, pour deux joueurs.

19	**Parcours du cavalier**	Très spectaculaire, et instructif à plus d'un titre.
20	**Mur de briques**	Un grand classique dans une version très complète.
21	**Othello**	Pour deux joueurs ; il retourne les pions à votre place.
22	**Mathieu**	Haute résolution et dessin au hasard.
23	**Imprime**	Si vous avez une imprimante...

Inscription directe dans les mémoires

LES INSTRUCTIONS D'EXPLORATION DES CASES MEMOIRES : PEEK et POKE

Sur le Basic Microsoft fourni à l'origine sur le Commodore 64, on ne dispose pas d'un certain nombre d'instructions pratiques (VTAB, COLOR, SOUND, etc.). Mais il est tout à fait possible (et rentable, pour ce qui est de la vitesse d'exécution) de simuler un certain nombre des fonctions absentes en agissant directement sur les cases mémoires concernées.

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

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 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 colorer en vert (couleur 5) le caractère présent en ligne X, colonne Y.

Smokey

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

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

LE PROGRAMME : Les lignes 7Ø et 8Ø font apparaître la fumée au bas de l'écran ; la ligne 9Ø permet de changer (vers la droite ou vers la gauche) la direction de la colonne, et la ligne 11Ø fait monter la colonne.
Le joueur apparaît grâce aux lignes 13Ø et 14Ø, et son déplacement est effectué par les lignes 18Ø et 19Ø.
Les lignes 27Ø et 28Ø se chargent de l'affichage du résultat.

POUR JOUER : On doit lancer l'exécution du programme par RUN. La fumée commence à s'élever, tandis que le joueur, apparu sous forme d'un rond coloré dans un coin de l'écran, va se déplacer sur elle à l'aide des 4 flèches. Dès qu'il arrive sur la fumée, il va automatiquement commencer à l'avaler, et devra essayer de suivre la colonne sans dévier (grâce aux flèches).
Il gagne s'il arrive en bas de l'écran avant que la fumée n'arrive en haut.
L'affichage final indique alors BRAVO ou PERDU, ainsi que le score obtenu.

EXTENSIONS POSSIBLES : On pourra, tout en conservant l'idée de départ, imposer d'autres règles : par exemple avaler toute la fumée, qui pourra aussi repartir d'un autre endroit si l'on dévie de la colonne initiale.
On pourra aussi améliorer les contrôles de fin de partie, et la qualité du graphisme.

```
READY.

10 REM SMOKEY
20 POKE 53281,9
30 POKE 53280,8
40 S = 0 : X = 24 : Y = INT(RND(1)*24)
50 U = 0 : V = 15
60 PRINT"□"
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 250
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$="■" AND T AND X < 24) + (A$="■" AND T AND X > 0)
190 U = U - ( T = 0)
200 V = V - (A$="." AND V < 24) + (A$="," AND V > 0)
210 C = PEEK(1024 + V + 40*U)
220 IF C <>83 THEN 70
230 T = 0 : S = S + 1
240 IF U <> 24 THEN 130
250 POKE 53281 , 6
260 PRINT"□"
270 IF T = 0 THEN PRINT"BRAVO" : GOTO 290
280 PRINT"PERDU"
290 PRINT"SCORE ";S

READY.
```



Course de voitures

REFERENCES : Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe (un 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 assuré par la boucle I (6Ø-8Ø) et les lignes 9Ø à 11Ø impriment la couleur.
La ligne 12Ø fait avancer la route, et les lignes 13Ø et 14Ø impriment la voiture.
La ligne 16Ø contrôle son mouvement, et la ligne 17Ø détermine si la route fait ou non des virages.
La ligne 19Ø regarde si l'on a touché le bord de la route.
La ligne 2ØØ incrémente le score, et la ligne 21Ø efface l'ancienne position de la voiture (une trace claire apparaît à l'emplacement de sa trajectoire).
La ligne 24Ø affiche le temps avant l'accident.

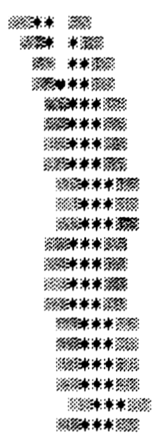
POUR JOUER : Lancer l'exécution du programme par l'ordre RUN, le dessin de la route commence à évoluer, et votre véhicule, représenté par un coeur apparaît.
Pour le déplacer vers la droite ou vers la gauche, vous devrez utiliser les touches —> et <— (flèches de déplacement du curseur).
Il vous faudra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

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

```
READY.


10 REM COURSE DE VOITURES
15 TI$ ="000000"
20 POKE 53280,2
30 POKE 53281,11
35 PRINT"♥"
40 X = 24 : Y = 11
45 A = 5 : B = 10
50 S = 0
60 FOR I = 0 TO 6
65 POKE 1024 + Y + 40*X +I , 42
70 IF I <2 OR I>4 THEN POKE 1024 + Y + 40*X + I , 102
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: POKE 55298 + Y + 40*X,8:POKE55299+Y+40*X,8
110 POKE 55300 + Y + 40*X , 8
120 PRINT"SQQQQQQQQQQQQQQQQQQQQQQQ"
130 POKE 1024 + B + 40*A , 83
140 POKE 55296 + B + 40*A , 3
145 FOR I = 1 TO 60 : GET A$ : IF A$ <>""THEN 155
150 NEXT I
155 C = B
160 B = B - (A$=".") + (A$=",")
170 Y = Y - (2*RND(1) AND Y < 25) +(2*RND(1) AND Y > 3)
190 IF PEEK(1024 + B + 40*(A+1)) = 102 THEN 230
200 S = S + 1
210 POKE 1024 + C + 40*A , 32
220 GOTO 60
230 PRINT"ACCIDENT ; SCORE :";S
240 PRINT"TEMPS ";MID$(TI$,3,2);" MINUTES ET ";RIGHT$(TI$,2);" SECONDES"

READY.
```



```
ACCIDENT ; SCORE : 38
TEMPS 00 MINUTES ET 28 SECONDES
```

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 6Ø à 16Ø dessinent l'échiquier.
Les lignes 17Ø à 24Ø 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 (26Ø-32Ø) et I (27Ø-31Ø) dessinent les pièces.
La ligne 35Ø permet l'entrée de la ligne et de la colonne de départ, avec contrôle en ligne 37Ø.
La ligne 39Ø permet d'entrer colonne et ligne d'arrivée.
La ligne 41Ø note la nouvelle position.
La ligne 42Ø assure le changement de joueur.

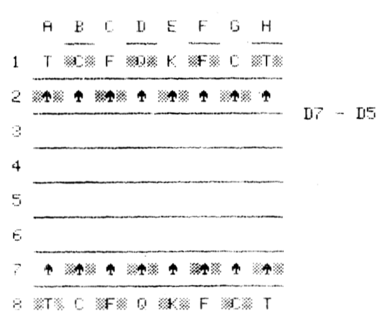
POUR JOUER : On lance l'exécution par RUN.
L'échiquier est dessiné, puis les pièces, et enfin l'inscription JOUEUR JAUNE, suivie de DEPART (C et L) ; on entre la colonne et la ligne de départ, sous la forme E7 par exemple, on obtient alors l'affichage de ARRIVEE (C et L) et on répond de même.
Le déplacement 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!

```
READY.

10 REM ECHECS
15 DIM R(8,8) : DIM C(8,8)
20 PRINT"□"
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 PRINTTAB(3*J);A$:PRINTI;TAB(3*J);B$:PRINTTAB(3*J);C$;
130 PRINT"□□□"
140 NEXT J
150 PRINT"Ø"
160 NEXT I
170 FOR I = 1 TO 8
180 FOR J = 1 TO 8
190 IF I = 2 OR I = 7 THEN R(I,J)=65
200 IF I <>2 AND I <>7 THEN R(I,J)=96
205 C(I,J) = 5
210 IF I = 7 OR I = 8 THEN C(I,J)=7
220 NEXT J
230 NEXT I
240 R(1,1)=20:R(1,2)=3:R(1,3)=6:R(1,4)=17:R(1,5)=11:R(1,6)=6:R(1,7)=3:R(1,8)=20
250 FOR I = 1 TO 8 : R(8,I)=R(1,I):NEXTI
255 H = 1
260 FOR J = 1 TO 8
270 FOR I = 1 TO 8
280'X = 1 + 3*J : Y = 1 + 2*I
290 POKE 1024 + X + 40*Y , R(I,J)
300 POKE 55296 + X + 40*Y , C(I,J)
310 NEXT I
320 NEXT J
330 PRINT"ØØØØØØØØ";TAB(28);"          "
335 PRINT"ØØØØØØØØØØØØØØØØØØØØØØØØØJOUEUR VERT     "
340 IF H = 1 THEN PRINT"ØØØØØØØØØØØØØØØØØØØØØØØØØJOUEUR JAUNE "
345 PRINTTAB(16);"            "
350 INPUT"□DEPART (C ET L) ";D$
352 PRINT"□";TAB(16);"          "
355 PRINT"ØØØØØØØ▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶";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"ØØØØØØØØØØØØØØØØØØØØØØØØØARRIVEE (C ET L)";D$
395 PRINT"ØØØØØØØ▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶- ";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

READY.
```




```
JOUEUR JAUNE
ARRIVEE (C ET L)? D5
```

Danger spatial

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

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

LE PROGRAMME : La position de chaque météorite est déterminée par les lignes 3Ø et 45 ; elles sont imprimées par les lignes 4Ø et 5Ø.
L'avance des météorites est réalisé par la ligne 6Ø.
Les lignes 7Ø et 8Ø vérifient si vous avez ou non heurté une météorite.
Les lignes 9Ø à 11Ø dessinent votre vaisseau spatial, et la ligne 13Ø permet de le déplacer.
Le compteur se trouve en ligne 14Ø.

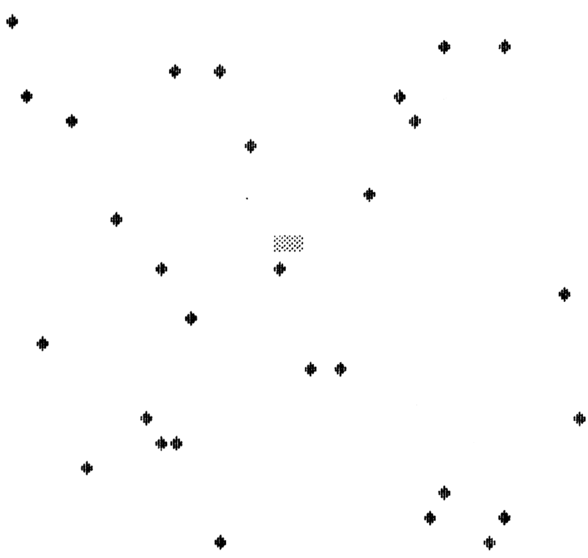
POUR JOUER : Le jeu est lancé par l'ordre RUN.
Votre vaisseau apparaît vers le milieu de l'écran, et les météorites viennent à votre rencontre en partant du bas de l'écran.
Pour vous déplacer, vous pouvez utiliser les touches de déplacement horizontal (→ et ←).
On peut arrêter la partie à tout moment en tapant RUN/STOP. Sinon, le jeu s'arrête lorsque vous heurtez une météorite, et le score est alors affiché.

EXTENSIONS POSSIBLES : Il serait intéressant de rendre le jeu de plus en plus difficile en faisant apparaître de plus en plus de météorites au fur et à mesure du déroulement de la partie.
On peut aussi essayer de détruire les météorites en les pulvérisant au fur et à mesure de leur apparition.

```
READY.

10 REM DANGER SPATIAL
15 PRINT"♥"
20 X = 18 : Z = 0
25 POKE 53281 , 9
30 H = INT(RND(1)*40)
40 IF RND(1) < 0.7THEN POKE 1984 + H , 90 : POKE 56256 + H,3
45 K = INT(RND(1)*40)
50 IF RND(1) < 0.6 THEN POKE 1984 +K , 90 : POKE 56256 +K , 3
60 PRINT"SQQQQQQQQQQQQQQQQQQQQQQQQ"
70 B = PEEK(1464 + X) : C = PEEK(1464 + X + 1)
80 IF B = 90 OR C = 90 THEN 200
90 POKE 1424 + Y , 32 : POKE 1424 + Y + 1 , 32
100 POKE 1464 + X , 102 : POKE 1464 + X + 1 , 102
110 POKE 55736 + X , 8 : POKE 55736 + X + 1 , 8
120 GET A$
125 Y = X
130 X = X - (A$="." AND X < 38) + (A$="," AND X > 0)
140 Z = Z + 1
150 GOTO 30
200 PRINT"LE SCORE EST DE :"Z

READY.
```



```
LE SCORE EST DE : 23
```

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 symbole (pique ou coeur) dans une case libre de la grille de jeu, et le premier qui réussit un alignement de 3 symboles identiques a gagné (l'alignement peut être horizontal, vertical, ou suivant l'une des deux diagonales).

LE PROGRAMME : Le terrain de jeu est dessiné à l'aide d'une boucle I (40-70).
Les lignes 100 et 110 s'occupent d'afficher le symbole coeur ou pique correspondant au joueur qui a le trait.
Un symbole clignotant (ligne 140) est affiché en haut et à gauche du terrain, même si la case est déjà occupée ; les lignes 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 affiche le coup, et la ligne 210 réalise le changement de joueur.

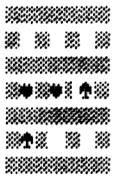
POUR JOUER : Le terrain de jeu sera affiché dès la frappe de RUN.
Un symbole clignotant apparaît alors en haut et à gauche de l'écran ; le joueur peut le déplacer sur le terrain de jeu à l'aide des 4 flèches (touches ← → et ↑ ↓).
Lorsqu'il arrive sur la case où il désire jouer, il suffit qu'il appuie sur la touche 0, le coup sera refusé si la case n'est pas libre.
Si la case était libre, le symbole qui correspond à ce joueur sera inscrit, et on change de joueur.
Lorsque la partie est finie, il convient de taper RUN/STOP pour sortir du programme.
On pourra ensuite faire une autre partie avec RUN.

EXTENSIONS POSSIBLES : On pourrait réaliser après chaque coup un contrôle pour vérifier si l'un des joueurs a gagné. On peut aussi réaliser un jeu machine mais il faudrait éviter qu'il soit par trop systématique (par exemple, la machine jouant toujours son premier coup sur la case centrale).

```
READY.


10 REM TIC TAC TOE
20 PRINT"■"
30 POKE 53281 , 0
40 FOR I = 1 TO 7
50 IF I/2 <>INT(I/2) THEN PRINTTAB(8);"■■        "
60 IFI/2=INT(I/2) THENPRINTTAB(8);"■ ";TAB(10);"■ ";TAB(12);
   "■ ";TAB(14);"■ "
70 NEXT I
80 PRINT"■■"
90 K = 0
100 PRINT"■■■■■■■■■■■■■■JOUEUR ♠"
110 IF K = 0 THEN PRINT"■JOUEUR ♥"
120 I = 0 : J = 0
130 C = PEEK(1113 + 2*I + 80*J):D=PEEK(55385+2*I+80*J)
140 POKE 1113+2*I+80*J,102:POKE 55385+2*I+80*J,1
150 POKE 1113+2*I+80*J,C:POKE55385+2*I+80*J,D
160 GET A$
170 J = J - (A$="■" AND J<2) + (A$="■" AND J>0)
180 I = I - (A$="." AND I<2) + (A$="," AND I>0)
190 IF C <>32 OR A$ <>"0" THEN 130
200 POKE 1113+2*I+80*J,83-K*18:POKE55385+2*I+80*J,3+K*5
210 K = ABS(K-1)
220 GOTO 100


READY.
```



```
JOUEUR ♥
```

Horloge 2

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

LE JEU : C'est un autre programme de simulation d'une horloge digitale grand format, qui présente la particularité intéressante d'utiliser le générateur de caractères du Commodore, transféré à une adresse convenable. Ce sont donc les caractères d'origine qui seront reproduits notablement agrandis, sur l'écran.

LE PROGRAMME : La ligne 5Ø protège la zone mémoire qui contient le générateur.
La partie du générateur de caractères qui contient les chiffres de 0 à 9 est recopiée en mémoire vive par la boucle I (6Ø-8Ø).
Les lignes 9Ø et 1ØØ assurent le retour aux conditions initiales.
L'heure est entrée en ligne 11Ø, sous forme d'un entier de 4 chiffres.
La variable temps TI$ du Commodore est utilisée en ligne 12Ø.
Les lignes 15Ø à 17Ø affichent les secondes, et contrôlent le changement d'affichage.
La boucle K (3ØØ-42Ø) dessine les 4 chiffres.
Le générateur de caractères est appelé en ligne 32Ø.
L'affichage de chaque ligne est effectué par la boucle T (34Ø-4ØØ).

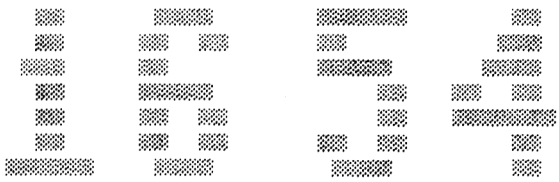
POUR JOUER : Faire RUN, puis à l'affichage de la question "ENTREZ L'HEURE" (il faudra attendre quelques instants), on répond en donnant un entier de 4 chiffres : les deux premiers représentent l'heure, et les deux suivants les minutes. 0416 signifiera ainsi 4 heures et 16 minutes. On attendra le quatrième top pour appuyer sur RETURN, et l'affichage commence.
On peut arrêter à tout moment en tapant RUN/STOP.

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

```
READY.

10 REM HORLOGE 2
20 PRINT"■"
30 POKE 56334,PEEK(56334) AND 254
40 POKE 1,PEEK(1) AND 251
50 POKE 52,48:POKE 56,48:CLR
60 FOR I = 0 TO 80
70 POKE 12288+I,PEEK(53632+I)
80 NEXT I
90 POKE 1,PEEK(1) OR 4
100 POKE 56334,PEEK(56334) OR 1
110 INPUT"ENTRER L'HEURE (FORME XXXX) :";A$
120 TI$ = A$ + "00"
130 PRINT"■"
140 GOSUB 300
150 J = 19 : GOSUB 500
160 PRINTTAB(19);RIGHT$(TI$,2)
170 IF RIGHT$(TI$,2) <>"00" THEN 150
180 GOTO 130
300 FOR K = 0 TO 3
310 FOR I = 0 TO 7
320 R =PEEK(12288 + 8*(ASC(MID$(TI$,K+1,1))-48)+I)
330 S = 128
340 FOR T = 0 TO 7
350 Z = INT(R/S)
360 R = R - Z*S
370 S = S/2
380 F = 6 + I : GOSUB 500
390 IF Z <>0 THEN PRINTTAB(9*K+T+1-3*(K>1));"■■ "
400 NEXT T
410 NEXT I
420 NEXT K
430 RETURN
500 PRINT"■"
510 FOR J = 0 TO F : PRINT : NEXT J
520 RETURN

READY.
```



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 de pièces sont entrés en une seule manoeuvre (position de départ et direction du déplacement) et les prises sont réalisées automatiquement, si elles sont possibles. Par contre le changement de joueur étant réalisé à chaque coup, il faudra éventuellement taper Ø pour pouvoir rejouer s'il y avait eu saut ou si le coup n'était pas possible.

LE PROGRAMME : Les boucles I (8Ø-16Ø) et J (9Ø-14Ø) dessinent le damier.
Les boucles I (17Ø-23Ø) et J (18Ø-22Ø) définissent les positions initiales des pions noirs et des pions blancs. Les positions sont dans une table R (X, Y) et la couleur dans une autre table C (X, Y).
L'impression des pions proprement dite est effectuée par les boucles emboîtées J (26Ø-32Ø) et I (27Ø-31Ø).
Les lignes 33Ø et 34Ø indiquent le joueur qui a le trait.
Le coup est entré en ligne 35Ø.
La ligne 355 réalise le changement de joueur (on tape Ø).
La ligne 37Ø regarde si le déplacement est possible.
La ligne 4ØØ traite le cas d'un déplacement et la ligne 41Ø celui d'un saut.
La ligne 45Ø assure le changement de joueur.

POUR JOUER : Dès le lancement du programme par RUN, le damier est affiché avec ses pions, et un caractère O inversé apparaît en-dessous pour indiquer que le trait appartient au joueur qui a choisi les blancs.
On joue en indiquant par un mot de 3 lettres (pouvant commencer par un Ø) la ligne et la colonne de la pièce à déplacer, puis la direction du déplacement (codée de 1 à 4) suivant la convention :

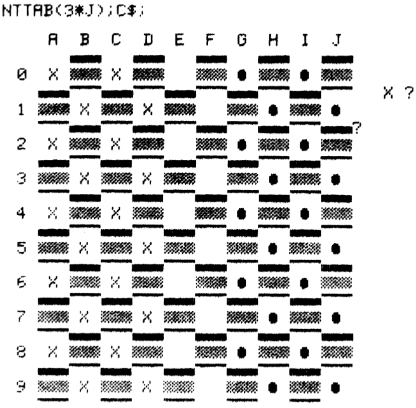


Par exemple 9C1 signifie que le O situé en ligne 9, colonne C, se déplace dans la direction 1 (ne pas oublier de faire RETURN).
De même, ØG3 signifiera pièce noir, ligne Ø, colonne G, direction 3.
Le changement de joueur étant effectué automatiquement il faudra taper Ø pour rejouer si le coup est refusé ou s'il y a eu prise (toujours terminer par RETURN). 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 Commodore 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 PRINT"▯"
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$;
130 PRINT"▯TT▯"
140 NEXT J
150 PRINT"▩"
160 NEXT I
170 FOR I = 1 TO 10
180 FOR J = 1 TO 10
185 R(I,J)=32
190 IF (I+J)/2 <> INT((I+J)/2) THEN 220
200 IF J < 5 THEN R(I,J) = 81 : C(I,J) = 0
210 IF J > 6 THEN R(I,J) = 81 : C(I,J) = 1
220 NEXT J
230 NEXT I
255 H = 1
260 FOR J = 1 TO 10
270 FOR I = 1 TO 10
275 IF (I+J)/2 <>INT((I+J)/2) THEN 310
280 X = 1 + 3*J : Y = 1 + 2*I
290 POKE 1024 + X + 40*Y , R(I,J)
300 POKE 55296 + X + 40*Y , C(I,J)
310 NEXT I
320 NEXT J
330 PRINT"▩▩▩▩▩▩"TAB(36);"■● ▯?"
340 IF H = 0 THEN PRINT"▯"TAB(36);"▩● ▯?"
345 PRINTTAB(33);"▩        "
350 INPUT"▯▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶I";D$
355 IF D$ ="0" 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)=32THENR(L,C)=32:R(L+S,C+T)=81:C(L+S,C+T)=-(H=0):GOTO450
410 IFC(L+S,C+T)=-(H=1)ANDR(L+S,C+T)=81ANDR(L+2*S,C+2*T)=32THEN430
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

READY.
```



Anagrammes

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

LE JEU : Des lettres sont choisies au hasard par le Commodore, le nombre de lettres étant fonction du niveau de difficulté désiré. Puis les lettres du mot ainsi obtenu sont brouillées, et affichées au bas de l'écran ; un nouveau mot est alors confectionné, de façon aléatoire, à partir des mêmes lettres, et il est affiché sous le premier, et ainsi de suite ; les deux joueurs (gauche et droit) disposent chacun d'une touche (1 ou Ø) et doivent l'appuyer dès qu'ils voient s'afficher au bas de l'écran un mot déjà présent sur l'écran ; leur score est alors évalué, et un message correspondant à leur résultat sera délivré.

LE PROGRAMME : Le niveau du jeu est entré à la ligne 3Ø.
La boucle I (6Ø-8Ø) réalise le choix des lettres.
Une deuxième boucle I (1ØØ-17Ø) effectue le brouillage, et la ligne 12Ø permet de détecter la pression d'une touche.
La ligne 18Ø affiche un nouveau mot, et fait monter d'une ligne sur l'écran les mots déjà obtenus.
Les lignes 23Ø et 24Ø cherchent quel est celui des deux joueurs qui a pressé une touche.
La boucle I (25Ø-32Ø) explore les mots présents sur l'écran ligne par ligne, alors que la boucle J (27Ø-3ØØ) saisit un par un les caractères de ces mots. On obtient ainsi un nouveau mot qui est comparé au mot initial en ligne 31Ø et le résultat BRAVO apparaît si c'est exact.
Sinon, la partie s'arrête sur l'affichage PERDU (ligne 33Ø).

POUR JOUER : On lance le programme par RUN.
On devra ensuite entrer le niveau de difficulté (en principe de 1 à 3) et faire 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 Ø), et son résultat sera affiché après un court instant. La partie sera alors terminée.

EXTENSIONS POSSIBLES : On pourra utiliser d'autres affichages que des mots, modifier les temps d'affichage, faire participer plus de deux joueurs, introduire le son et la couleur, etc.

```
10 REM ANAGRAMMES
20 PRINT"♥"
30 INPUT"NIVEAU ( DE 1 A 3 ) :";N
40 IF N<1 OR N>3 THEN RUN
50 S$ = ""
60 FOR I = 1 TO N+3
70 S$ = S$ + CHR$(INT(RND(1)*26+65))
80 NEXT I
90 A$ ="" : 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 B$ = LEFT$(B$,N-1) + "-" + RIGHT$(B$,LEN(B$)-N)
170 NEXT I
180 PRINT"SQQQQQQQQQQQQQQQQQQQQQQQQ";A$
190 FOR I = 1 TO 20*LEN(B$):NEXT I
210 S$ = A$
220 GOTO 90
230 H$ = "JOUEUR DE DROITE"
240 IF D$="1" THEN H$ = "JOUEUR DE GAUCHE"
245 PRINT"S"
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$)+4);"TTBRAVO , ";H$ :
    PRINT"S" : END
320 NEXT I
330 PRINTTAB(LEN(S$)+4);"TTPERDU , ";H$


READY.

OVEB
BVEO
EVBO
OEBV
OEVB
VBEO
OVBE
BOVE
EVBO
EBVO
BEOV
VBEO
BEVO
         PERDU , JOUEUR DE DROITE
```

Œufs

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

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

LE PROGRAMME : L'oiseau est dessiné en deux parties (pour le mouvement) aux lignes 7Ø, 8Ø et 12Ø ; son déplacement aléatoire est géré par le sous-programme 3ØØ-41Ø (en particulier les lignes 37Ø à 4ØØ).
La ligne 14Ø permet de reboucler le programme sur la ligne 7Ø.
Le déplacement du joueur est régi par les lignes 16Ø et 17Ø, et le test permettant de détecter la présence d'un oeuf est effectué en ligne 18Ø.
La ligne 2ØØ permet l'affichage du score.
La fin de jeu est détectée en ligne 32Ø.

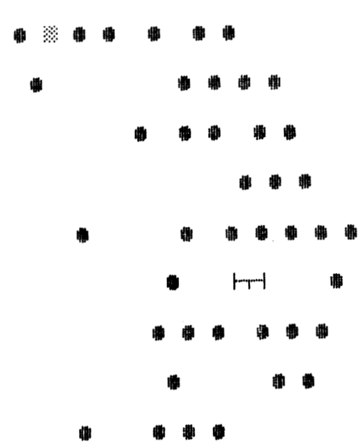
POUR JOUER : On lancera le programme par RUN.
L'oiseau commence alors à se déplacer au hasard sur l'écran, et se met à pondre ses oeufs.
Lorsque le symbole qui représente le joueur (un carré gris) apparaît sur l'écran, celui-ci peut se déplacer en utilisant les 4 touches de déplacement. Il vous faut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

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

```
READY.


10 REM OEUFS
20 PRINT"■"
30 POKE 53280,2:POKE 53281,5
40 P = 200
50 X = INT(RND(1)*36) : Y = INT(RND(1)*21)
60 R = 18 : S = 10 : T = 0
70 POKE1024+X+40*Y,115:POKE1025+X+40*Y,113:POKE1026+X+40*Y,107
80 POKE 55296+X+40*Y,0:POKE55297+X+40*Y,0:POKE55298+X+40*Y,0
90 FOR I = 1 TO 30 : GET A$
95 IF A$="" THEN NEXT I
100 IF A$ <>""THEN GOSUB 150
110 GOSUB 300
120 POKE1024+X+40*Y,107:POKE1025+X+40*Y,114:POKE1026+X+40*Y,115
130 GOSUB 300
140 GOTO 70
150 POKE1024+R+40*S,32
160 R = R - (A$="." AND R < 36) + (A$="," AND R >0)
170 S = S - (A$="■Q" AND S < 21) + (A$="■]" AND S >0)
180 G = PEEK(1024+R+40*S)
190 T = T - (G = 81)
200 PRINT"■SQQQQQQQQQQQQQQQQQQQQQQQQ";TAB(15);"SCORE :";T
210 RETURN
300 A = X : B = Y
310 P = P - 1
320 IF P = 0 THEN POKE 53281,6:PRINT"C'EST FINI":END
330 POKE1024+R+40*S,102:POKE55296+R+40*S,2
360 POKE1024+A+40*B,32:POKE1025+A+40*B,81:POKE1026+A+40*B,32:
    POKE55297+A+40*B,7
370 X = ABS(X+INT(1+2*(RND(1)<0.5)))
380 Y = ABS(Y+INT(2+4*(RND(1)<0.5)))
390 IF X > 36 THEN X = 1
400 IF Y > 21 THEN Y = 1
410 RETURN

READY.
```



```
              SCORE : 18
C'EST FINI
```

Affichage tournant (vertical)

REFERENCES : Difficulté de programmation : difficile.
Catégorie de jeu : affichage publicitaire.
Difficulté du jeu : facile.

LE JEU : Le programme proposé réalise le défilement d'un texte, affiché en gros caractères, sur l'écran. L'utilisation du générateur de caractères a permis de simplifier considérablement le programme.

LE PROGRAMME : La ligne 5Ø permet de protéger la zone mémoire qui contient le générateur.
La boucle I (6Ø-8Ø) recopie en mémoire vive, à partir de l'adresse 12288, les codes du générateur de caractères (on ne recopie pas tout le générateur).
Les lignes 9Ø et 1ØØ permettent de revenir dans les conditions initiales.
Le texte est introduit en ligne 11Ø, et la boucle A (12Ø-16Ø) calcule le code générateur de chaque lettre (ligne 13Ø) avant d'envoyer dans le sous-programme d'affichage (ligne 14Ø).
La ligne 15Ø fait monter l'écran d'une ligne.
Le générateur (recopié en mémoire vive) est appelé en ligne 21Ø.
La boucle T (23Ø-28Ø) réalise l'impression d'une ligne du caractère, et la boucle Y (2ØØ-3ØØ) appelle les lignes suivantes.

POUR JOUER : On doit d'abord lancer le programme par l'instruction RUN, puis on entre le texte, on tape ensuite RETURN, et on regarde.
Le programme boucle sur lui-même et le défilement ne pourra être interrompu que par un appui sur la touche RUN/STOP.

EXTENSIONS POSSIBLES : Il suffira d'avoir un peu d'imagination pour adapter ce programme à toutes sortes d'affichages de types publicitaires (il s'agit alors d'attirer l'attention, d'une part, et d'informer, d'autre part). On pourra à cet effet utiliser la couleur, personnaliser les messages, utiliser plusieurs lignes, etc.

```
READY.

10 REM AFFICHAGE PUBLICITAIRE
20 PRINT"{CLR}"
30 POKE 56334,PEEK(56334) AND 254
40 POKE 1,PEEK(1) AND 251
50 POKE 52,48:POKE 56,48:CLR
60 FOR I = 0 TO 504
70 POKE 12288+I,PEEK(53256+I)
80 NEXT I
90 POKE 1,PEEK(1) OR 4
100 POKE 56334,PEEK(56334) OR 1
110 INPUT"ENTRER VOTRE MESSAGE ";A$
120 FOR A = 1 TO LEN(A$)
130 X = ASC(MID$(A$,A,1))+65*(MID$(A$,A,1)>="A")
140 GOSUB 200
150 PRINT"{control characters}"
160 NEXT A
170 PRINT:PRINT
180 GOTO 120
200 FOR Y = 0 TO 7
210 R = PEEK(12288+8*X+Y)
220 S = 128
230 FOR T = 15 TO 22
240 Z = INT(R/S)
250 R = R - Z*S
260 S = S/2
270 IF Z<>0 THEN PRINT"{control characters}"TAB(T);"{control characters} ";
280 NEXT T
290 PRINT"{CLR}"
300 NEXT Y
310 RETURN

READY.
```



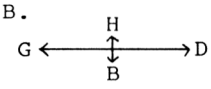
Quatre à quatre

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

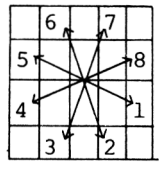
LE JEU : Il pourra être pratiqué à un ou plusieurs joueurs. Dans les deux cas, le but est d'arriver à échanger les pions X avec les pions O ; si l'on joue en solitaire, cet échange devra être réalisé avec le moins de coups possibles. Les pions sont déplacés alternativement en respectant les règles de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

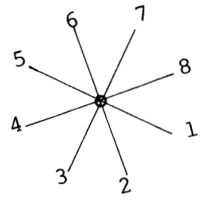
LE PROGRAMME : La variable alphanumérique A$, en ligne 40, contient la table des coups possibles (déplacement du cavalier).
Les boucles emboîtées I (50-110) et J (60-100) dessinent le terrain de jeu, et les positions de départ.
Les lignes 130 et 140 permettent d'afficher O ou X, selon le joueur qui a le trait.
La ligne 170 permet l'affichage du curseur clignotant (ici un rond 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 du départ. Les lignes 270 et 280 effectuent le déplacement. La ligne 300 réalise le contrôle d'arrivée.
Le résultat est imprimé par la ligne 310, le changement de joueur est réalisé par la ligne 320 et un . est imprimé à l'emplacement de départ (ligne 330). Le sous-programme 350-360 est utilisé pour les divers contrôles.

POUR JOUER : On lance le programme par RUN. Le terrain de jeu est affiché, ainsi que le symbole permettant d'identifier le joueur qui a le trait.
Le joueur peut choisir le pion qu'il désire déplacer en manoeuvrant le curseur clignotant grâce aux touches D, G, H, B.



Pour déplacer son pion, il suffit, lorsque le curseur est sur ce pion, de taper la direction du déplacement, suivant le code :





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

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

```
10 REM QUATRE A QUATRE
20 PRINT"□"
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 O"
140 IF H = 0 THEN PRINT"□JOUEUR X"
150 GOSUB 350
160 I = J
170 POKE1024+2*X+80*Y,81:POKE55296+2*X+80*Y,10
180 A = X : B = Y
190 GET R$
200 Y = Y - (R$="B" AND Y<5) + (R$="H" AND Y>1)
210 X = X - (R$="G" AND X<5) + (R$="D" AND X>1)
220 IF R$<"9" AND R$>"0" THEN 250
230 POKE1024+2*A+80*B,I: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 230
260 J = VAL(R$)
270 Y = Y + VAL(MID$(A$,J,1)) - 4
280 X = X + VAL(MID$(A$,J+2,1)) - 4
290 GOSUB 350
300 IF J <> 46 THEN 230
310 POKE1024+2*X+80*Y,15-9*(H=0)
320 H = ABS(H-1)
330 I = 46
340 GOTO 230
350 J = PEEK(1024+2*X+80*Y)
360 RETURN
```

```
READY.
  1 2 3 4 5

1 O . . . .

2 O O . . .

3 . X O . .

4 . . . . X

5 . . . X X


JOUEUR O
```

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, pour permettre de reconstituer un proverbe très connu.
Une case grise vous permet de déplacer les lettres, et un compteur affiche en permanence le nombre d'essais : arriverez-vous à retrouver le proverbe ?

LE PROGRAMME : La phrase à reconstituer (qui pourra bien sûr être modifiée) est placée dans une variable R$ en ligne 7Ø.
Le mélange des lettres et l'impression sont faits par la boucle I (8Ø-15Ø).
La ligne 18Ø réalise l'affichage du carré gris ; celui-ci est déplacé par les lignes 22Ø et 23Ø en utilisant les 4 flèches.
Les lignes 24Ø et 25Ø réimpriment la lettre concernée à l'emplacement du carré gris.
La ligne 26Ø comptabilise le nombre d'essais, qui est imprimé grâce à la ligne 17Ø.

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

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

```
READY.

10 REM TAQUIN DE LETTRES
20 PRINT"■"
30 FOR I = 1 TO 10
40 PRINT"■■                ■"
50 NEXT I
60 A$ = ""
70 R$ ="ONASOUVENTBESOIND'UNPLUSPETITQUESOI"
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 R$ = LEFT$(R$,X-1) + "-" + RIGHT$(R$,LEN(R$)-X)
130 F = ASC(MID$(A$,I,1))+64*(MID$(A$,I,1)<>"'")
140 POKE1146+(I-6*INT(I/6))+40*INT(I/6),F
145 POKE55418+(I-6*INT(I/6))+40*INT(I/6),1
150 NEXT I
160 X = 0 : Y = 0 : S = 0 : C = 32
170 PRINT"■■■■■■■■■■"TAB(15);"NOMBRE D'ESSAIS :"S
180 POKE1146+Y+40*X,102:POKE55418+Y+40*X,7
190 GET T$
200 IF T$="" THENPOKE1146+Y+40*X,C:GOTO180
210 I = X : J = Y
220 X = X - (T$="■" AND X<5) + (T$="■" AND X>0)
230 Y = Y - (T$="." AND Y<5) + (T$="," AND Y>0)
240 C = PEEK(1146+Y+40*X)
250 POKE1146+J+40*I,C
260 S = S + 1
270 GOTO 170

READY.
```

```
■IVSUQ
PPSUSO
NNEAST
IE'LIB
UNEOTT
DONEOU        NOMBRE D'ESSAIS : 5
```

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

POUR JOUER : Lancer le programme par RUN, et taper le nombre de cibles désirées 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, 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.

```
READY.

10 REM CIBLES
20 PRINT"♥"
30 INPUT"NOMBRE DE CIBLES ";N
40 P = 10 : J = 30
50 S = 0 : B = 1
60 PRINT"♥"
65 FOR I = 1 TO N
70 PRINTTAB(J)"‾‾┐":PRINTTAB(J+1)" |":PRINTTAB(J)"▁▁┘"
75 NEXT I
80 K = 0 : A = 18
90 PRINT"SQQQQQQQQQQQQQQQQQQQQQQQ]]]]]SCORE :";S;" - BALLES :";B
100 POKE1024+K+40*A,81:POKE55296+K+40*A,5
110 D = A
120 FOR I= 1 TO N
130 U = INT(RND(1)*P)
140 POKE1024+J+(3*I-1)*40,U+48:POKE55296+J+(3*I-1)*40,8
150 GET A$
160 A = A - (A$="B") + (A$="H")
170 NEXT I
180 POKE1024+K+40*D,32
190 IF A$ = "0" THEN B = B+1 : GOTO 210
200 IF K = 0 THEN 100
210 K = K + 1
220 IF K < J THEN 100
230 C = PEEK(1024+K+40*A)
240 IF C <48 OR C > 57 THEN PRINT"PERDU" : END
250 S = S + C - 48
260 P = P - 1
270 IF P = 0 THEN END
280 GOTO 80

READY.
```

```
                                              ‾‾┐
                                              6 |
                                              ▁▁┘
                                              ‾‾┐
                                         ●    0 |
                                              ▁▁┘
                                              ‾‾┐
                                              8 |
                                              ▁▁┘
                                              ‾‾┐
                                              0 |
                                              ▁▁┘




     SCORE : 0  - BALLES : 1
```

Gouttes

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

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

LE PROGRAMME : Les deux boucles emboîtées I (2Ø-9Ø) et J (3Ø-8Ø) dessinent le récipient et les obstacles.
La ligne 13Ø dessine la goutte.
La ligne 15Ø efface la goutte précédente et la ligne 17Ø envoie dans un sous-programme (4ØØ-42Ø) qui permet de déterminer s'il y a ou non rencontre avec un obstacle.
Si c'est non, on va en 3ØØ et la goutte continue de tomber.
Si c'est oui, la goutte part à droite ou à gauche (ligne 19Ø) selon la valeur définie pour K en ligne 11Ø.
On regarde de nouveau s'il y a ou non obstacle, ou si la goutte peut continuer à tomber (lignes 2ØØ, 21Ø ou 22Ø).
Selon le résultat du test, on utilisera 23Ø, 3ØØ, 32Ø ou 35Ø.

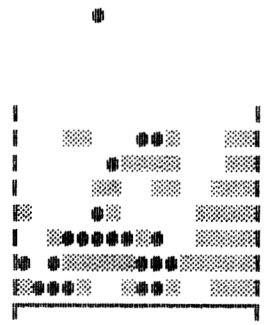
POUR JOUER : Le RUN déclenche le dessin du récipient et des obstacles, la goutte apparaît, se met à tomber et finit par se trouver une place. Une autre goutte apparaît et l'on recommence.
Le programme ne s'ârrête pas de lui-même lorsque le récipient déborde ; on peut l'arrêter à tout moment en tapant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

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

```
READY.


10 REM GOUTTES
15 PRINT"■"
20 FOR I = 12 TO 20
30 FOR J = 1 TO 16
40 POKE1024+40*I,118:POKE55296+40*I,8:POKE1041+40*I,117:
   POKE55296+17+40*I,8
50 POKE1824+J,120:POKE56096+J,8
60 IF I<>12 AND I<>20 THENPOKE1024+J+40*I,102:POKE55296+J+40*I,5
70 IF I <> 20 AND RND(1) < 0.8 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 FOR J = 0 TO 100 : NEXT J
150 POKE 1024+Y+40*X,32
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+V+40*U)
410 J = -(C <> 32)
420 RETURN
500 S = INT(RND(1)*13)
510 POKE1025+S+40*I,32:POKE1026+S+40*I,32
520 RETURN


READY.
```



Obstacles

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

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

LE PROGRAMME : La ligne 4Ø fait avancer les obstacles vers la voiture du joueur qui est imprimée grâce à la ligne 35.
Les lignes 5Ø, 6Ø et 7Ø fabriquent la route (les lignes 6Ø et 7Ø, bien qu'identiques, sont toutes deux indispensables, car la valeur de RND ne sera, en principe, pas la même sur ces deux lignes).
La ligne 8Ø affiche le score.
La ligne 1ØØ permettra de déplacer le véhicule.
Les lignes 11Ø et 12Ø permettent de contrôler s'il y a ou non un obstacle et arrêtent le programme s'il y a lieu.
Le score est incrémenté par la ligne 13Ø.

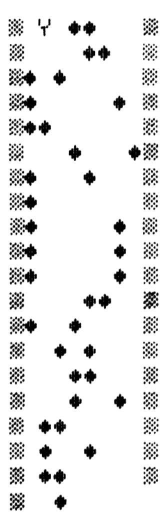
POUR JOUER : On lance le programme en tapant RUN.
La route et les obstacles apparaissent en bas de l'écran et montent vers la voiture représentée par un Y inversé.
La voiture peut se déplacer à l'aide des flèches de déplacement horizontal (touches ← et →).
La rencontre avec un obstacle interrompt le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

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

```
READY.

10 REM OBSTACLES
20 PRINT"♥"
30 S = 0 : Y = 5
35 PRINT"▒Q"TAB(Y);"▒Y▒"
40 PRINT"▒QQQQQQQQQQQQQQQQQQQQQQQQ"
50 PRINT"]▒▒          ▒▒▒"
60 PRINT"]"TAB(INT(RND(1)*8+1));"♦"
70 PRINT"]"TAB(INT(RND(1)*8+1));"♦"
75 PRINT"▒QQQ"TAB(15);"          "
80 PRINT"▒QQQQ"TAB(15);"SCORE"S
90 GET A$
100 Y = Y -(A$="." AND Y<9) + (A$="," AND Y>0)
110 C = PEEK(1064 + Y)
120 IF C = 90 OR C = 102 THEN END
130 S = S + 1
140 GOTO 35

READY.
```



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 manoeuvrer pour recueillir l'urane, sans vous faire heurter par une météorite.

LE PROGRAMME : La ligne 5Ø permet de faire apparaître les météorites.
La ligne 7Ø fera apparaître le minerai d'urane.
Leur déplacement, sur l'écran, est assuré par la ligne 8Ø.
Les lignes 1ØØ à 12Ø contrôlent le déroulement du jeu (a-t-on heurté une météorite ou a-t-on capturé un bloc d'urane).
La rencontre d'une météorite provoque la fin de partie.
Votre vaisseau spatial est imprimé par les lignes 13Ø et 14Ø et son déplacement est contrôlé par la ligne 16Ø.

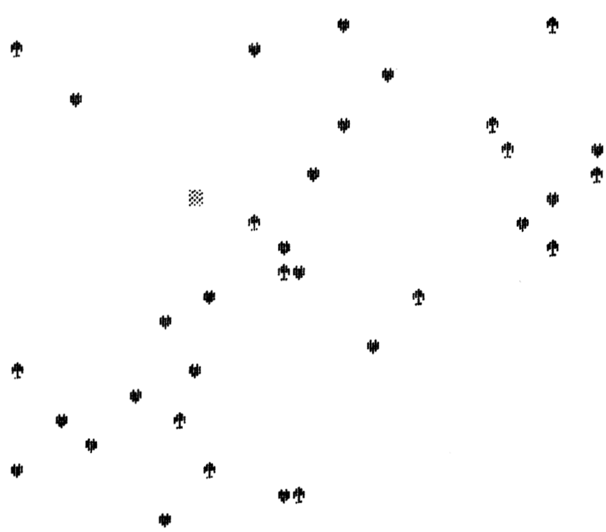
POUR JOUER : On lance le programme par RUN et les météorites commencent à monter sur l'écran, ainsi que les blocs d'urane.
Les touches de déplacement horizontal (← et →) seront utilisées pour déplacer votre vaisseau et récupérer l'urane, mais le moindre heurt avec une météorite provoquera la fin de la partie et l'affichage de la quantité d'urane capturée.
On pourra aussi arrêter à tout moment en tapant RUN/STOP.

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

```
READY.

10 REM CAPTURES DANS L'ESPACE
20 POKE 53281,9
25 PRINT"J"
30 X = 15 : Z = 0
40 S = INT(RND(1)*40)
50 POKE1984+S,83:POKE56256+S,14
60 S = INT(RND(1)*40)
70 IF RND(1)<0.5 THEN POKE1984+S,65:POKE56256+S,5
80 PRINT"SQQQQQQQQQQQQQQQQQQQQQQQQ"
90 FOR I = 1 TO 40 : NEXT I
100 C = PEEK(1424+X)
110 IF C = 83 THEN GOTO 200
120 IF C = 65 THEN Z = Z + 1
130 POKE1383+X,32:POKE1384+X,32:POKE1385+X,32
140 POKE1424+X,102:POKE55696+X,1
150 GET A$
160 X = X - (A$="." AND X<40) + (A$="," AND X>0)
170 GOTO 40
200 PRINT"SCORE OBTENU :"Z

READY.
```



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 à 270 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.

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

```
READY.


10 REM ANOMALIE
20 PRINT"■"
30 T = 5 :TI$="000000"
40 FOR I = 1 TO T
50 POKE 1024+2*I,48+I:POKE55296+2*I,7
60 POKE 1040+2*I,48+I:POKE55312+2*I,3
70 POKE 1024+80*I,48+I:POKE55296+80*I,7
80 POKE 1040+80*I,48+I:POKE55312+80*I,3
90 NEXT I
100 FOR I = 1 TO 22
110 D = INT(RND(1)*26+1)
120 L = 1 : S = T
130 GOSUB 1000
140 L = 9
150 GOSUB 1000
160 NEXT I
170 S = 0
180 GOSUB 1000
190 INPUT"■■■■■■■■■■■■■■■■■■■■LIGNE :";I
200 INPUT"COLONNE :";T
210 U = 2*I : V = 2*T + 16
220 GOSUB 1010
230 PRINT"BRAVO..."
240 IF D <> C THEN PRINT"■PERDU..."
250 PRINT"C'ETAIT LA LETTRE ";CHR$(D+64)
260 PRINT"VOTRE TEMPS :";MID$(TI$,3,2);" MN. , ";RIGHT$(TI$,2);
    " SEC."
270 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,1
1040 RETURN


READY.
```

```
  1 2 3 4 5        1 2 3 4 5

1   I B Z        1 H Z J I I

2 N J Z S V      2 F I B B T

3 J T I J S      3 B S   B J

4 F I L   D      4 D A V   J

5 H V A I B      5 L N Z   I
```

```
LIGNE :? 3
COLONNE :? 4
PERDU...
C'ETAIT LA LETTRE B
VOTRE TEMPS :00 MN. , 24 SEC.
```

Plusieurs en un

REFERENCES : Difficulté de programmation : difficile.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : variable, selon l'utilisation.

LE JEU : Plusieurs jeux sont en réalité possibles avec ce programme qui va se comporter comme un crayon électronique vous permettant d'inscrire des symboles O ou X, en n'importe quel emplacement de votre feuille-écran de télévision. On pourra, par exemple, l'utiliser pour un jeu de morpions (le premier à réaliser un alignement de 5 pions identiques gagne) ou pour un GO (on réalise des encadrements des pions adverses) ou tout autre jeu, selon son imagination.

LE PROGRAMME : Le terrain de jeu est délimité par 4 symboles mis en place par les lignes 3Ø et 4Ø.
Le changement de joueur est effectué par la ligne 6Ø.
Le joueur qui a le trait voit le symbole qu'il a choisi (O ou X) se placer en limite du cadre (ligne 8Ø).
Les lignes 11Ø et 12Ø permettent, aux deux joueurs, de déplacer le symbole sur le terrain.
Les lignes 13Ø à 16Ø contrôlent si le déplacement est possible (case libre et non dépassement des limites).
La ligne 22Ø permet d'écrire sur une case libre.

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

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

```
READY.


10 REM PLUSIEURS EN UN
20 PRINT"■"
30 POKE1024,79:POKE55296,1:POKE1054,80:POKE55326,1
40 POKE1784,76:POKE56056,1:POKE1814,122:POKE56086,1
50 J = 0
60 J = ABS(J-1)
70 X = 12 : Y = 0
80 POKE1024+Y+40*X,15-9*(J=1):POKE55296+Y+40*X,1-7*(J=1)
90 GET A$ : IF A$ = "" THEN 90
100 A = X : B = Y
110 X = X - (A$="B" AND X <19) + (A$="H" AND X>1)
120 Y = Y -(A$="D" AND Y <29) + (A$="G" AND Y>1)
130 POKE1024+B+40*A,32
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
200 FOR I = 1 TO 30 : 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,1-7*(J=1)
230 GOTO 60


READY.
```



Parcours du cavalier

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

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

LE PROGRAMME : Les lignes 4Ø et 5Ø définissent les divers déplacements possibles, dans deux variables alphanumériques A$ et B$; les deux boucles emboîtées I (6Ø-11Ø) et J (7Ø-1ØØ) dessinent le terrain de jeu.
On entre les coordonnées de la case de départ (ligne, puis colonne) aux lignes 12Ø et 13Ø.
La ligne 16Ø imprime le numéro du coup.
La ligne 18Ø permet le tirage au hasard d'un déplacement du cavalier.
La boucle Z (19Ø-24Ø) 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 2ØØ et 21Ø effectuent le déplacement.
La ligne 22Ø contrôle que l'on ne sorte pas du terrain de jeu.
La ligne 23Ø envoie en ligne 3ØØ si le coup est possible.
La ligne 25Ø gère la fin de partie (cavalier bloqué) et affiche le score.
Les lignes 3ØØ, 31Ø et 32Ø réalisent le coup et 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.

EXTENSIONS POSSIBLES : On peut imaginer de nombreuses variantes sur ce thème, ne serait-ce qu'en commençant par faire varier la taille des échiquiers on peut aussi améliorer considérablement le graphisme, utiliser la couleur. Mais le plus intéressant est d'arriver à définir un al-

gorithme donnant la ou les solutions du problème initial...

```
10 REM PARCOURS DU CAVALIER
20 PRINT"■"
30 S = 1 : T = 0
40 A$ = "56653223"
50 B$ = "653223" + A$ + "65"
60 FOR I = 1 TO LEN(A$)
70 FOR J = 1 TO LEN(A$)
80 POKE1024+3*I,48+I:POKE55296+3*I,8:POKE1024+80*I,48+I:
   POKE55296+80*I,8
90 POKE1024+3*J+80*I,46:POKE55296+3*J+80*I,1
100 NEXT J
110 NEXT I
120 INPUT"■■■■■■■■■■■■■■■■■■■■■■■■■■LIGNE DE DEPART ";I
130 INPUT"COLONNE DE DEPART ";J
140 A$ = A$ + A$
150 IF S>9 THEN S=0 : T = T + 1
160 POKE1024+3*J+80*I,S+48:POKE1023+3*J+80*I,T+48
170 POKE55296+3*J+80*I,5:IF T <>0 THEN POKE55295+3*J+80*I,5
180 Z = INT(RND(1)*8+1)
190 FOR Z = Z TO Z + 8
200 A = I + VAL(MID$(A$,Z,1)) - 4
210 B = J + VAL(MID$(B$,Z,1)) - 4
220 IF A<1 OR B<1 OR A>8 OR B>8 THEN 240
230 IF PEEK(1024+3*B+80*A)=46 THEN 300
240 NEXT Z
250 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■"TAB(25);"TOTAL"T*10+S : END
300 I = A : J = B
310 S = S + 1
320 GOTO 150

READY.

   1  2  3  4  5  6  7  8

1  .  . 24 19  .  . 14  .

2  . 20 17  . 23  .  .  .

3  .  . 22 25 18 15  . 13

4 21  .  . 16 27 12  .  .

5  .  . 26  . 04  . 28 07

6  .  .  .  . 11 06 01  .

7  .  . 10 05  . 03 08 29

8  .  .  .  . 09 30  . 02


LIGNE DE DEPART ? 6
COLONNE DE DEPART ? 7    TOTAL 30
```

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.

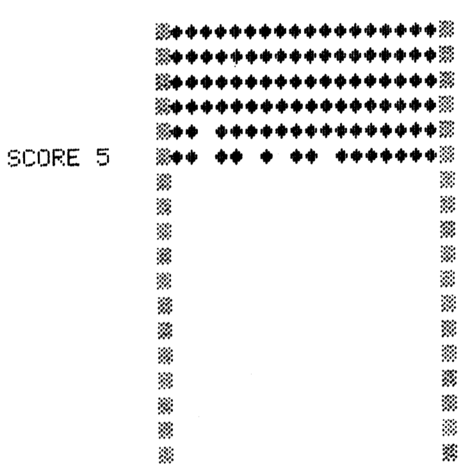
LE PROGRAMME : La boucle I (4Ø-6Ø) dessine le terrain de jeu et les briques sont mises en place par les boucles emboîtées I (7Ø-11Ø) et J (8Ø-1ØØ).
La ligne 13Ø dessine la raquette dont la position est contrôlée par la ligne 15Ø.
Les lignes 2ØØ à 23Ø regardent si l'on a touché le mur, les briques ou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

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

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

```
10 REM MUR DE BRIQUES
15 POKE 53281,14: POKE 53280,9
20 PRINT"♥"
30 PRINTTAB(10);"▒▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬"
40 FOR I = 1 TO 18
50 PRINTTAB(10);"▒                  ▒"
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"▒QQQQQQQQQQQQQQQQQQQQQQQQ"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"▒QQQQQQQQSCORE"S
180 IF I > 22 THEN 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:POKE55296+J+40*I,0:T=1:
    GOTO 130
220 IF C = 102 THEN 260
230 IF C = 90 THEN POKE1024+J+40*I,81:POKE55296+J+40*I,0:T=1
    :GOTO 280
240 H = - H
250 K = RND(1)-RND(1)
260 K = -K : T = 0
270 GOTO 130
280 S = S + 1
290 K = -(K - (2*(RND(1)<0.5)+1)*RND(1)/3)
300 H = -H
310 GOTO 130


READY.
```



Othello

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

LE JEU : Devenu un grand classique des jeux sur micro-ordinateur, le jeu d'OTHELLO ou REVERSI gagne même si comme ici il est pratiqué à deux joueurs, à être géré sur ordinateur, avec visualisation ; en effet, le jeu consistant essentiellement en des retournements des pions de l'adversaire, ceci peut se révéler, à la longue, très fastidieux, surtout si vous êtes un joueur assidu. Le principe est que le joueur, possédant les X, doit 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 procèdera de la même façon, et ceci jusqu'à ce que le damier soit rempli ou que personne ne puisse plus jouer (si un joueur ne peut retourner des pions, il doit passer son tour).

LE PROGRAMME : Les boucles emboîtées P (4Ø-9Ø) et Q (6Ø-8Ø) dessinent le terrain de jeu et les lignes 11Ø et 12Ø affichent un X ou un O, selon le joueur qui commence.
La boucle P (19Ø-3ØØ) contrôle la validité du coup (la case est-elle libre ? : lignes 28Ø et 29Ø), en utilisant une partie de programme qui se trouve aux lignes 35Ø à 41Ø (a-t-on pris un ou plusieurs ennemis en ligne 35Ø et retournement éventuel, autorisé par la ligne 37Ø : on refait un balayage dans toutes les directions, si l'encadrement s'est révélé possible).
La ligne 19Ø définit les 8 directions d'exploration.
Les lignes 23Ø et 24Ø calculent les déplacements à effectuer pour explorer ces 8 directions.
La ligne 22Ø réalise les déplacements.
La ligne 32Ø assure le changement de joueur, s'il y a eu retournement (contrôlé en ligne 31Ø).

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 Ø en première coordonnée

pour le changement de joueur.

EXTENSIONS POSSIBLES : Avant tout, bien sûr, améliorer l'affichage avec des couleurs, une grille plus aérée.
On pourra aussi imaginer un jeu machine pour le Commodore.

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

```
READY.

10 REM OTHELLO
20 PRINT"■"
30 INPUT"DIMENSION DE LA GRILLE ";A
32 IF A/2 <>INT(A/2) THEN RUN
35 PRINT"■"
40 FOR P = 1 TO A
50 POKE 1024+2*P,48+P:POKE 55296+2*P,8
60 FOR Q = 1 TO A
70 POKE1024+80*P,48+P:POKE55296+80*P,8:POKE1024+2*Q+80*P,46:POKE55296+2*Q+80*P,1
75 IF Q = A/2 + 1 AND P = A/2 THENPOKE1022+2*Q+80*P,24:POKE1024+2*Q+80*P,15
77 IF Q=A/2 + 1 AND P=A/2 + 1 THENPOKE1022+2*Q+80*P,15:POKE1024+2*Q+80*P,24
80 NEXT Q
90 NEXT P
100 K = 1
110 PRINT"■■■■";TAB(23)"JOUEUR X"
120 IF K = 0 THEN PRINT"■"TAB(23);"JOUEUR O"
130 PRINT"■"TAB(23);"                "
140 INPUT"■■■■■■■■■■■■■■■■■■■■■■■■■LIGNE";U
150 IF U = 0 THEN 320
160 INPUT"■■■■■■■■■■■■■■■■■■■■■■■■■COLONNE";V
170 PRINT"■"TAB(23);"                "
180 H = 1
190 FOR P = 1 TO 8
200 S = 0
210 W = 0
220 T = 2*U : R = 2*V
230 F = (P>3 AND P<7) - (P<3 OR P>7)
240 G = (P>1 AND P<5) - (P>=6)
250 Q = 46 + 22*(S=1) + 31*(S=2)
255 IF S = 0 THEN 270
260 POKE 1024+R+40*T,Q
270 T = T + 2*F : R = R + 2*G
280 C = PEEK(1024+R+40*T)
290 IF C = 15 OR C = 24 THEN 350
300 NEXT P
310 IF H <> 0 THEN 110
320 K = ABS(K-1)
330 GOTO 110
350 IF (C=15 AND K=1) OR (C=24 AND K=0) THEN W=1
360 IF S <> 0THEN POKE1024+R+40*T,Q
370 IF W=1 AND ((C=15 AND K=1) OR (C=24 AND K=0)) THEN 270
380 IF W=0 OR S<>0 THEN 300
390 S = 1 - (K=0)
400 H = -H*(S=0)
410 GOTO 210

READY.
```

```
  1 2 3 4 5 6

1 . . . . . O          JOUEUR X

2 . . . X O X          LIGNE? 3

3 . . X O . .          COLONNE? 5

4 . . O X . .

5 . . . . . .

6 . . . . . .
```

Mathieu

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

LE JEU : On utilise la définition d'une courbe en coordonnées polaires et la haute résolution pour obtenir sur l'écran des dessins aléatoires, de type fleur, plus ou moins élégants. La méthode utilisée pour obtenir le passage en haute résolution pourra, bien entendu, être reprise avec profit pour d'autres programmes.

LE PROGRAMME : La ligne 3Ø permet de choisir l'origine de la mémoire d'affichage, et la ligne 4Ø fixe la plus haute adresse mémoire disponible pour Basic.
La ligne 5Ø remplit la mémoire de couleur HR avec des 8.
La ligne 6Ø fait un effacement de la mémoire d'affichage HR (qui va de 8192 à 16191, soit 8000 octets).
La ligne 65 met le Commodore en mode HR.
La boucle M (8Ø-14Ø) définit la courbe en coordonnées polaires, et les lignes 1ØØ et 11Ø assurent la conversion en coordonnées cartésiennes.
Les lignes 12Ø et 13Ø appellent les deux sous-programmes de calcul de la position d'un point.
La ligne 15Ø contrôle la fin de partie.

POUR JOUER : On lance le programme par RUN et l'on attend patiemment que l'initialisation soit terminée. Le dessin est alors effectué automatiquement. On pourra modifier les couleurs en changeant les nombres contenus dans les lignes 5Ø et 6Ø. Pour faire un autre dessin, il suffit de taper RUN et d'attendre.

EXTENSIONS POSSIBLES : L'accès à la haute résolution ouvre la voie à d'immenses possibilités : il sera prudent au début de commencer l'exploration de ce domaine en faisant tracer au Commodore des courbes mathématiques plus ou moins complexes, en s'inspirant de la méthode donnée ci-contre.

Remarque : *L'exemple d'exécution n'est pas fourni pour ce programme, la haute résolution n'étant pas possible sur l'imprimante avec le programme de copie d'écran dont disposait l'auteur.*

```
READY.


10 REM MATHIEU
20 PRINT"□"
30 POKE 53272,PEEK(53272) OR 8
40 POKE 52,32 : POKE 56,32 : CLR
50 FOR I = 1024 TO 2023 : POKE I,8 : NEXT
60 FOR I = 8192 TO 16191 : POKE I,0 : NEXT I

65 POKE 53265,PEEK(53265) OR 32
70 A = 3*RND(1) : B = 3*RND(1)
80 FOR H = 0 TO π*2 STEP π/180
90 R = 7*SIN(3*H)
100 X = 150+INT(8*R*COS(A*H))
110 Y = 100+INT(8*R*SIN(B*H))
120 GOSUB 200
130 GOSUB 300
140 NEXT H
150 IF A$ = "" THEN 150
160 END
200 U = INT(X/8):V=INT(Y/8)
210 P = 8192+320*V+8*U
220 RETURN
300 K = X AND 7 : S = Y AND 7
310 Q = P + S
320 POKE Q,PEEK(Q) OR 2↑(7-K)
330 RETURN


READY.
```

Imprime

REFERENCES : Difficulté de programmation : facile.
Catégorie de jeu : programme de reproduction d'écran.
Difficulté du jeu : aucune.

LE JEU : Il ne s'agit pas ici d'un jeu,mais d'un utilitaire qui pourra rendre service à ceux qui ne disposent pas de la carte d'extension Commodore, permettant la copie d'écran. Le programme se contente d'explorer la mémoire d'écran, ligne par ligne, et de transformer les codes écrans obtenus en code imprimante (la transformation n'est pas réalisée pour tous les codes écran possibles).

LE PROGRAMME : Les lignes 3Ø1ØØ et 3Ø2ØØ dirigent la sortie sur l'imprimante.
La boucle I (3Ø3ØØ-3Ø98Ø) permet l'exploration ligne par ligne, et la boucle emboîtée J (3Ø4ØØ-3Ø9ØØ) permet l'exploration des colonnes.
La ligne 3Ø6ØØ lit le contenu de chaque position.
La ligne 3Ø7ØØ transforme les codes écran en codes imprimante, et la ligne 3Ø71Ø élimine les cas douteux.
La ligne 3Ø8ØØ réalise l'impression.
La ligne 3Ø99Ø provoque une erreur, qui retire la main au périphérique imprimante.

POUR JOUER : On affichera le programme sur l'écran par LIST, puis on effectuera le chargement du programme à exécuter.
Le programme d'impression sera alors incorporé au programme à exécuter en reprenant les lignes 3ØØØØ à 3Ø98Ø, présentes sur l'écran, à l'aide des flèches "curseur haut" et de la touche RETURN.
Les numéros de lignes (3ØØØØ à 3Ø98Ø) sont choisis de telle sorte qu'il n'y ait pas de risque de suppression de lignes dans le programme principal.
On utilisera le programme soit en prévoyant une ligne GOTO 3ØØØØ dans le programme principal, soit en arrêtant le programme (RUN/STOP) et en écrivant GOTO 3ØØØØ directement en bas d'écran.

EXTENSIONS POSSIBLES : On pourra améliorer le programme en évitant le problème des espaces sur l'imprimante, en affinant le contrôle des codes écrans, etc.

```
READY.

30000 REM IMPRIME
30100 OPEN4,4
30200 CMD4
30300 FOR I = 1024 TO 2023 STEP 40
30400 FOR J = 0 TO 39
30500 S = I + J
30600 P = PEEK(S)
30700 P = P-64*(P<32)-32*(P>63)-32*(P>97)
30750 IF P <0 OR P > 255 THEN P = 32
30800 PRINTCHR$(P);
30900 NEXT J
30950 PRINT
30980 NEXT I
30990 GO

READY.
```

Annexe

```
READY.


10 REM COMMENTAIRES SUR L'AFFICHAGE
20 REM
30 PRINT"        L'AFFICHAGE DES CARACTERES SPECIAUX"
40 PRINT"        ──────────────────────────────────────"
50 REM
60 PRINT"DEPLACEMENTS DU CURSEUR"
70 PRINT"────────────────────────"
80 REM
90 PRINT"  - CURSEUR A DROITE S'IMPRIME : ▌ "
100 REM
110 PRINT" - CURSEUR A GAUCHE S'IMPRIME : ▌ "
120 REM
130 PRINT" - CURSEUR EN HAUT S'IMPRIME : ▗ "
140 REM
150 PRINT" - CURSEUR EN BAS S'IMPRIME : ▚ "
160 REM
170 PRINT" SHIFT - CLR/HOME S'IMPRIME : ▝ "
180 REM
190 PRINT" CLR/HOME S'IMPRIME          : ▞ "
200 REM
210 PRINT"LES TOUCHES UTILISEES POUR DEPLACER LE CURSEUR LORS DES JEUX SONT :",
220 PRINT" , - . - ▚ - ▌ "
230 REM
240 PRINT"POUR LES COULEURS"
250 PRINT"─────────────────"
260 REM
270 PRINT"ON PEUT UTILISER LA TOUCHE - CTRL - SUIVI D'UN NUMERO DE 1 A 0 "
280 REM
290 PRINT"CTRL 1 DONNE : ■ (SOIT NOIR)"
300 PRINT"CTRL 2 DONNE : ▗ (SOIT BLANC)"
310 PRINT"CTRL 3 DONNE : ▚ (SOIT ROUGE)"
320 PRINT"CTRL 4 DONNE : ▙ (SOIT TURQUOISE)"
330 PRINT"CTRL 5 DONNE : ▞ (SOIT VIOLET)"
340 PRINT"CTRL 6 DONNE : ▛ (SOIT VERT)"
350 PRINT"CTRL 7 DONNE : ▜ (SOIT BLEU)"
360 PRINT"CTRL 8 DONNE : ▝ (SOIT JAUNE)"
370 PRINT"DE PLUS CTRL 9 , SOIT ▘ , PERMET LE PASSAGE EN CONTRASTE INVERSE"
380 PRINT"ET CTRL 0 , SOIT ▀ , PERMET DE REVENIR EN CONTRASTE NORMAL"
390 PRINT"ON OBTIENT LES AUTRES COULEURS EN ACTIONNANT LA TOUCHE COMMODORE "
400 PRINT"AINSI COMM 1 DONNE ▖ (SOIT ORANGE)"
410 PRINT"COMM 2 DONNE ▞ (SOIT MARRON)"
420 PRINT"COMM 3 DONNE ▚ (SOIT ROUGE CLAIR)"
430 PRINT"COMM 4 DONNE ▌ (SOIT GRIS 1)"
440 PRINT"COMM 5 DONNE ▟ (SOIT GRIS 2)"
450 PRINT"COMM 6 DONNE ▐ (SOIT VERT CLAIR)"
460 PRINT"COMM 7 DONNE ▗ (SOIT BLEU CLAIR)"
470 PRINT"COMM 8 DONNE ▟ (SOIT GRIS 3)"
480 PRINT"LES INSTRUCTIONS PRINT PEUVENT CONTENIR TOUTES CES COMMANDES"
500 END

READY.
```

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

N° d'impression : 412014
N° d'édition : 86595-115-4
ISBN : 2-86595-115-4



Apprendre en se distrayant, tel est l'objectif de ce livre. Au fil de ces 102 programmes de jeux il vous guidera dans l'exploration du Basic Commodore 64. Les programmes sont classés par niveau, chacun d'eux faisant appel à de nouvelles connaissances et à une plus grande maîtrise du Basic. Chaque niveau commence par une présentation concise de nouvelles instructions utilisées. Tous les jeux sont décrits, et les programmes abondamment commentés; un exemple d'exécution est fourni pour chaque version.

DES LIVRES POUR VOTRE ORDINATEUR

P.S.I.

ISBN : 2.86595.115.4

****

120 FF

Couverture GIBRAT


102 PROGRAMMES POUR COMMODORE 64
PAR JACQUES DECONCHAT
PSI