



102 PROGRAMMES

POUR

Ti.99/4A



PAR
JACQUES
DECONCHAT



PROGRAMMES

102
PROGRAMMES POUR
TI-99/4A

La collection « **PROGRAMMES** » est constituée actuellement de 18 titres. Certains disposent d'une disquette pour Apple II (*).

- (*) Bibliothèque scientifique en Pascal — Hervé Haut
- (*) Mathématiques et statistiques — Hervé Haut
- (*) Modèles pratiques de décision — tomes 1 et 2 — Jean-Pierre Blanger
- Jeux, trucs et comptes pour PET/CBM — Michel Benelfoul
- Jeux, trucs et comptes pour TRS-80 — Michel Benelfoul et Denis Chauvin
- Jeux, trucs et comptes pour T07 — Michel Benelfoul
- Jeux, trucs et comptes pour TI-99/4A — Michel Benelfoul
- Jeux, trucs et comptes pour Commodore 64 — Michel Benelfoul et Cyril Cambien
- Etudes pour ZX-81 — tomes 1 et 2 — Jean-François Séhan
- Variations pour PC-1211 — Jean-François Séhan
- Récréations pour TI-57 — tomes 1 et 2 — Jacques Deconchat
- Collège, pochettes et maths — Jacques Deconchat
- « 102 Programmes pour Sinclair et Timex » — Jacques Deconchat
- 102 programmes pour Commodore 64 — Jacques Deconchat
- 102 programmes pour TI-99/4A — Jacques Deconchat
- Le TI-99/4A à l'affiche — Jean-François Séhan
- Le vic à l'affiche — Jean-François Séhan
- Suites pour PC-1500 — Jean-François Séhan

D'autres ouvrages relatifs au TI-99/4A :

- La découverte du TI-99/4A — Frédéric Lévy et Dominique Schraen
Collection « **MATERIELS** » — série verte
- Exercices pour TI-99/4A — Frédéric Lévy
Collection « **MATERIELS** » — série verte
- Le TI-99/4A à l'affiche — Jean-François Séhan
Collection « **PROGRAMMES** » — série bleue
- Jeux, trucs et comptes pour TI-99/4A — Michel Benelfoul
Collection « **PROGRAMMES** » — série verte

RAPPELS

Les séries :

En fait, il faudrait parler de niveaux, puisque la couleur attachée à chaque ouvrage permet de situer la « force » de celui-ci selon le code suivant :

Série VERTE : ouvrage d'initiation ne nécessitant que des connaissances de base.

Série BLEUE : suppose une connaissance élémentaire du sujet traité.

Série ROUGE : ouvrage d'approfondissement, niveau de complexité moyen.

Série NOIRE : ouvrage d'approfondissement, niveau de complexité élevé.

Les collections :

Les ouvrages d'Édition du PSI, actuellement au nombre de 100, sont répartis en collections :

« **LANGAGES** », « **MATERIELS** », « **PROGRAMMES** », « **GUIDES PRATIQUES** », « **MEMENTOS** », « **UTILISATIONS DE L'ORDINATEUR** », « **LOGI GUIDES** » et pour l'initiation, outre quelques livres hors collection, « ... **POUR TOUS** ».

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

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

© Editions du P.S.I. Parc industriel nord, Bâtiment 9, 77200 Torcy Marne-la-Vallée 1984

ISBN : 2-86595-114-6

**JACQUES
DECONCHAT**

**102
PROGRAMMES POUR
TI-99/4A**

**ADAPTE PAR
GERARD SANTRAILLE**



EDITIONS DU P.S.I.

1984

PRÉSENTATION DE L'AUTEUR

Jacques DECONCHAT, ancien élève de l'ENSET, est actuellement professeur de mathématiques au Lycée Turgot à Paris.

Il suit depuis plusieurs années le développement de la micro-informatique, et notamment ses applications dans le domaine de l'éducation. Après s'être penché dans ses précédents ouvrages sur l'utilisation des calculatrices programmables, il s'intéresse maintenant aux nouveaux systèmes de grande diffusion, et c'est le fruit de ses premières expériences que vous trouverez dans cet ouvrage.

Trois ouvrages, du même auteur, concernant l'utilisation des calculatrices programmables sont parus dans la collection "Programmes" :

- Récréations pour TI-57 - Tome 1
- Récréations pour TI-57 - Tome 2
- Collège, poquettes et maths.
- 102 Programmes pour Sinclair ZX et Timex.

SOMMAIRE

	pages
Avant-propos	7
Organisation de l'ouvrage	9
NIVEAU 1	11
Instructions élémentaires	12
Le nombre mystérieux	18
Cube	20
Chaud, froid	22
Les allumettes	24
Les allumettes à plusieurs	26
Calendrier perpétuel	28
Roulette	30
Bataille navale	32
Golf	34
Solfège	36
Le jeu du 21	38
Pair - Impair	40
Simon	42
Devine un chiffre	44
Le jeu du Wytthof	46
4 2 1	48
Rebonds	50
Jackpot	52
Le jeu de la fève	54
Jeu de l'oie aléatoire	56
NIVEAU 2	59
Dimensionnement d'un tableau - Recherche d'un élément dans un tableau et dans une chaîne	60
Un dé	62
Visions	64
Plus grand, plus petit	66
L'électricien fou	68
La course au 20	70
Dessin	72
Dé truqué	74
Le jeu de saute-mouton	76
Lettres	78
Espion	80
Accrochez les wagons	82
Tirage d'un dé truqué	84
Ordre	86
Course de chevaux	88
Le jeu de Marienbad	90
Idem	92
Mastermind	94
A travers	96
Reines sur l'échiquier	98
Le jeu de la vie	100
NIVEAU 3	103
Manipulations des chaînes graphiques	104
Losanges	106
Echiquier	108
Couloirs	110
Jour de la semaine	112
Le pendu	114

102 PROGRAMMES POUR TI-99/4A

Bridge	116
Horloge 1	118
Le loup et les agneaux	120
Belote	122
Un taquin de calcul	124
Poker	126
Hanoi	128
Musique	130
Pendu inversé	132
Mathieu	134
Machinamot	136
Composé	138
Nicomaque	140
Permutation	142
Tour de cartes	144
Parcours du cavalier	146
Mur de briques	148
NIVEAU 4	151
Animation	153
Tirage de dés	154
Jeu du loto	156
La bombe	158
Ardoise magique	160
Affichage publicitaire (1)	162
Evasion	164
Chapeaux	166
Combat aérien	168
Dessin de dés	170
Le jeu du missile	172
Le singe	174
Le jeu du 15	176
Adresse	178
Sous-marin	180
Zombies	182
Diago	184
Topologie	186
Chasse au sous-marin	188
Le lézard d'Alice	190
Calculatrice	192
Smokey	194
Course de voitures	196
Echecs	198
Danger spatial	200
Tic Tac Toe	202
Gouttes	204
Obstacles	206
Captures dans l'espace	208
Plusieurs en un	210
NIVEAU 5	213
Présentation de certaines fonctions du Basic Etendu	215
Horloge 2	216
Dames	218
Anagrammes	220
Oeufs	222
Affichage tournant (vertical)	224
Quatre à quatre	226
Taquin de lettres	228
Cibles	230
Affichage publicitaire (2)	232
Anomalies	234
Othello	236

AVANT-PROPOS

Vous trouverez dans ce livre plus de 100 programmes, rédigés en Basic Texas Instrument (TI Basic).

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

Chaque niveau de programme (il y en a 5) est précédé d'une courte étude des instructions Basic mises en oeuvre dans les jeux qui suivent. Les différences avec les autres Basic, sont mises en évidence.

De plus les programmes proposés sont "ouverts", ce qui signifie qu'ils ont été conçus, dans la mesure du possible, pour être facilement utilisés et modifiés.

Des indications sont fournies à la fin de chaque programme, pour vous suggérer des améliorations possibles, ou des modifications qui pourraient rendre le jeu plus attrayant, plus performant, ou plus complet.

Les programmes du 5ème chapitre, une dizaine au total, de cet ouvrage, sont écrits en Basic Etendu et nécessitent donc, pour être exécutés, que le module correspondant soit connecté à la console.

Organisation de l'ouvrage

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

- **Le niveau 1** permet d'étudier les instructions élémentaires du langage Basic : **LET**, **PRINT**, **INPUT**, **IF** ☐ **THEN** ☐, **FOR...NEXT**, **GOTO**, **GOSUB**, **ON-GOTO**, **ON-GOSUB**, **HCHAR**.
- **Au niveau 2**, on utilisera les instructions de confection et de manipulation des éléments d'un tableau et d'une chaîne : **DIM** et **SEG\$**.
- **Au niveau 3**, on fait appel à la puissante instruction de traitement des chaînes de caractères : **SEG\$** et aux instructions graphiques : **HCHAR**, **VCHAR** et **GCHAR**.
- **Au niveau 4**, on rencontre des instructions propres aux jeux interactifs : **KEY**.
- **Enfin, le niveau 5** nous familiarisera avec certaines instructions du Basic Etendu comme la manipulation des **SPRITES** (petit lutins animés) et avec l'affichage sur écran d'une chaîne à un endroit désiré par l'instruction **DISPALAY AT**.

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.

NIVEAU 1

<i>N°</i>	<i>Nom du programme</i>	<i>Thème et particularités</i>
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 réflexion, contre le TI-99/4A.
5	Allumettes à plusieurs	Jeu de réflexion et de stratégie pour plusieurs joueurs.
6	Calendrier perpétuel	Confection d'un calendrier, suivant mois et année.
7	Roulette	Jeu de hasard, pour plusieurs joueurs, simulation.
8	Bataille navale	Un seul navire à trouver, mais quelques particularités.
9	Golf	Un parcours en 18 trous, pour un seul joueur.
10	Solfège	Apprenez le solfège, avec l'aide de votre TI-99.
11	Le jeu du 21	Jeu de hasard, contre le TI-99.
12	Pair - Impair	Un jeu de casino en trois temps (un ou plusieurs joueurs).
13	Simon	Répétez une séquence de chiffres de plus en plus longue.
14	Devine un chiffre	Jeu de casino, pour un ou plusieurs joueurs.
15	Le jeu de Wytthof	Un jeu de réflexion très simple pour deux joueurs.
16	421	Une simulation simplifiée, pour plusieurs joueurs.
17	Rebonds	Un jeu d'adresse très visuel, pour un joueur.
18	Jackpot	Jeu de hasard, très classique (un joueur).
19	Le jeu de la fève	Un jeu de réflexion, pour deux joueurs.
20	Jeu de l'oie aléatoire	Un vieux jeu, à la sauce TI-99, pour deux joueurs.

Instructions élémentaires

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

- *CALL CLEAR pour nettoyer l'écran,*
 - *RND pour obtenir un nombre aléatoire entre 0 et 1, associé à RANDOMIZE,*
 - *INT pour prendre la partie entière du résultat,*
- etc.

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

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

soit seulement huit instructions de base, que nous allons maintenant examiner en détail.

L'INSTRUCTION LET, OU INSTRUCTION D'AFFECTION

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 l2, 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 numérique NOM\$ prend la valeur RENOIR.

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

L'affectation une fois réalisée, le micro-ordinateur saura, chaque fois qu'on lui parle de A1, qu'il s'agit du nombre 12 ; si on parle de NOM\$, du mot RENOIR, etc.

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

C'est une instruction de dialogue qui permet à la machine d'afficher sur l'écran un texte, le résultat d'un calcul, ou une variable ; c'est le seul moyen que possède le micro-ordinateur de faire connaître ses résultats. Ainsi PRINT A1 provoque l'affichage de 12 (si on a fait auparavant LET A1 = 12) et PRINT NOM\$ l'affichage de RENOIR (si on a fait LET NOM\$ = "RENOIR") ; utilisé après effacement d'écran (CALL CLEAR) le PRINT provoque l'affichage de la 1ère ligne en haut, à gauche.

Divers paramètres seront utilisés pour indiquer au TI-99 où il doit afficher ses résultats ou ses symboles (y compris les dessins).

- PRINT TAB(N) provoque sur la ligne d'impression courante un affichage commençant à la Nième colonne à partir de la gauche ($1 \leq N \leq 28$).
- 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 automatique prédéfinie.
- Le : (deux points) permet de revenir à la ligne.

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

Cette instruction permettra d'entrer et d'affecter directement dans la variable désignée par le INPUT un nombre, ou un texte et ceci en mode dit conversationnel, c'est-à-dire au fur et à mesure des besoins ; en effet, lorsque le TI-99 rencontre une instruction INPUT, il s'arrête et vous rend la main. Vous pourrez alors entrer une information de nature convenable, et lui signaler que cette entrée est terminée, à l'aide de la touche ENTER, qui rendra la main au TI-99.

Par exemple, si dans le programme, le TI-99 trouve INPUT B, il s'arrête, et vous devez taper un nombre B quelconque, suivi de ENTER (si vous tapez un caractère autre qu'un chiffre ou un

point décimal, vous aurez un message d'erreur : il est prudent de prévoir l'affichage d'un texte avant l'arrêt, vous préciserez ce que vous devez entrer).

Par exemple : INPUT "TEXTE A AFFICHER" : B

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

En TI-Basic, l'instruction INPUT est similaire à l'INPUT des autres Basic, qui permet d'afficher un texte, et d'entrer successivement plusieurs variables si nécessaire.

L'INSTRUCTION GOTO, OU INSTRUCTION DE RENVOI INCONDITIONNEL

Une des particularités du langage Basic, qui le rend facilement accessible, est que **chaque ligne de programme doit être précédée d'un numéro**, ces lignes étant ensuite exécutées par le micro-ordinateur en séquences, dans le sens des numéros croissants. Pour diverses raisons, il pourra être nécessaire, au cours du déroulement d'un programme, de ne pas suivre cet ordre séquentiel très rigide ; on utilisera alors **l'instruction GOTO, qui permet de détourner le déroulement du programme de son cours normal**, quitte d'ailleurs à le reprendre ensuite par un autre GOTO. Le Basic, et c'est le cas du Basic Texas, n'autorise en général que les renvois à un numéro de ligne connu (GOTO 20, par exemple, ou GOTO 200, etc.).

Mais une des particularités les plus remarquables du Basic Texas est qu'il permet aussi le GOTO calculé.

En effet, une expression du type **ON** expression numérique, **GOTO** ligne, ligne, ligne, s'exécute en deux temps. L'ordinateur évalue l'expression numérique, ou la variable, située entre **ON** et le **GOTO** qui est arrondie à un entier. Il se dirige ensuite vers le numéro de la ligne à droite du **GOTO** situé à la position pointée par la variable précédente.

Par exemple : ON A GOTO 1000, 2000, 500 branchera le programme en 1000 si A = 1, en 2000 si A = 2 et en 500 si A = 3. Notons qu'il n'y a pas de limite au nombre de numéros de lignes situés à droite du GOTO.

L'INSTRUCTION IF THEN OU INSTRUCTION DE TEST

L'instruction **IF THEN** est très puissante. On l'emploie en rupture de séquence conditionnelle, sous la forme **IF condition = vraie THEN nnn numéro de ligne**. C'est l'usage le plus courant de ce test, qui pourra toutefois être amélioré ici par le fait que la condition peut être une expression arithmétique ou logique complexe, évaluée logiquement.

Remarque : Le Basic Texas supporte la clause **ELSE** qui branche le programme à un autre numéro de ligne si la condition est fausse.

Par exemple : IF A = 22*B THEN 320 ELSE 840

L'INSTRUCTION FOR NEXT, OU INSTRUCTION DE BOUCLES

On sera très souvent amené, en programmation, à vouloir exécuter un nombre entier de fois une certaine suite d'instructions (par exemple, faire avancer un mobile depuis la case 0 jusqu'à la case 20, etc.). Plutôt que d'utiliser un test, et un renvoi, il sera préférable de faire appel à une **boucle**, la variable de boucle prenant ses valeurs dans les entiers ou les décimaux, et le pas de cette variable (quantité dont elle augmente ou diminue à chaque passage) pouvant aussi prendre ses valeurs dans les entiers ou les décimaux.

Ainsi, si l'on écrit :

```
FOR I = 0 TO 20 STEP 2
  |
  | Instructions
  |
NEXT I
```

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

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

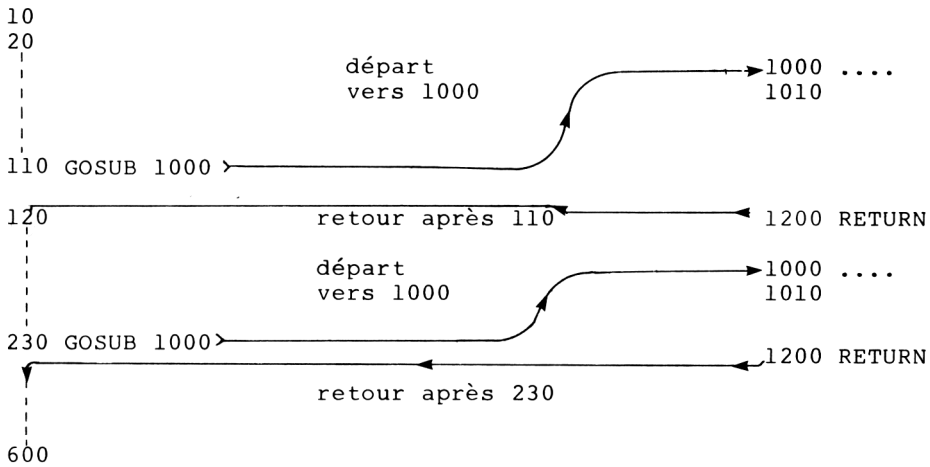
Le Basic permet d'emboîter un nombre indéfini de boucles (limité par la mémoire disponible).

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

Lorsque la même suite d'instructions devra être exécutée plusieurs fois, et appelée à partir de plusieurs endroits du programme, plutôt que de la réécrire plusieurs fois, on essaiera de la rédiger une fois pour toutes, et on la placera en dehors du programme proprement dit (on lui donnera, par exemple un numéro de ligne de départ supérieur au numéro de la dernière ligne du programme) ; indépendamment des difficultés que l'on pourra rencontrer au niveau des noms 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 appelée, 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.

Ci-après, le schéma théorique de fonctionnement.

Programme principal



Fin du programme principal

1000
1200 RETURN

Sous-programme 1

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

Un sous-programme pourra lui-même éventuellement appeler un autre sous-programme, et ainsi de suite, sans autre limitation que la taille mémoire, dans le Basic Texas.

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

Par exemple : ON A GOSUB 830,5410 provoquera l'exécution du sous-programme 830 si A = 1 et du sous-programme 5410 si A = 2.

L'INSTRUCTION HCHAR

Cette instruction, largement employée dans cet ouvrage, permet d'afficher un caractère en un point quelconque de l'écran. D'une utilisation moins souple que PRINT elle est néanmoins très intéressante et très usitée parce que l'écran "ne monte pas" lors de son utilisation. Sa syntaxe est ;

CALL HCHAR (n° de ligne, n° de colonne, code ASCII , répétitions).

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 PLUS GRAND ou PLUS PETIT.
- LE PROGRAMME** : La ligne 50 permet de fabriquer le nombre à deviner.
Les entrées sont faites par les lignes 70 et 80.
La ligne 90 teste si le résultat est exact.
La ligne 110 signale un essai trop petit, et la ligne 130 un essai trop grand.
Le compteur d'essais est en ligne ?
Les lignes 160 à 190 s'occupent de l'affichage du résultat, et de la gestion de la partie suivante.
- POUR JOUER** : On lancera le programme par RUN.
On obtient un premier affichage NBRE. DE CHIFFRES ? auquel on répond par un entier entre 1 et 7, suivi de ENTER.
On obtient alors l'affichage ESSAI N° 1, suivi de la question VOTRE NOMBRE.
On entre un nombre, et l'on presse ENTER.
On obtient alors une réponse TROP PETIT, ou TROP GRAND, ou encore, si l'on a trouvé juste un affichage de VOUS AVEZ TROUVE ... EN ... ESSAIS. Si le résultat n'est pas exact, on revient à ESSAI N° 2, et VOTRE NOMBRE.
En fin de partie, on obtient la question AUTRE PARTIE (O OU N) ? ; on répond O pour faire une autre partie, suivi de ENTER.
On peut arrêter à tout moment en entrant une lettre en réponse à la question VOTRE NOMBRE.
- EXTENSIONS POSSIBLES** : On pourra imaginer de faire deviner au joueur autre chose qu'un chiffre (par exemple, un mot, en utilisant l'ordre alphabétique, ou encore autre chose...).

```

10 CALL CLEAR
20 INPUT "NOMBRE DE CHIFFRES
   : "N
30 IF N>7 THEN 10
40 RANDOMIZE
50 X=INT(RND*10^N+1)
60 J=1
70 PRINT : : "ESSAI NO";J: :
80 INPUT "VOTRE NOMBRE : "M
90 IF M=X THEN 160
100 IF M>X THEN 130
110 PRINT : "TROP PETIT"
120 GOTO 140
130 PRINT : "TROP GRAND"
140 J=J+1
150 GOTO 70
160 CALL CLEAR
170 PRINT TAB(6); "VOUS AVEZ
TROUVE";X;TAB(6); "EN";J; "COU
PS": :
180 INPUT "   UNE AUTRE PART
IE O/N ? "R$
190 IF R$="O" THEN 10

```


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 HCHAR 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 une boucle (100-150) pour l'impression des arêtes fuyantes. Le nom de l'objet représenté est affiché en ligne 160 (ici : CUBE).
- POUR JOUER** : Il suffit de lancer l'exécution par RUN, et de regarder.
Sortie du programme par CLEAR en fin d'exécution.
- EXTENSIONS POSSIBLES** : On pourra imaginer un jeu utilisant des parallélépipèdes rectangles de différentes tailles, avec affichage par exemple des 3 côtés, et il faudrait trouver leur volume, etc.

```

10 CALL CLEAR
20 CALL HCHAR(10,7,46,11)
30 CALL HCHAR(20,7,46,11)
40 CALL VCHAR(11,7,46,9)
50 CALL VCHAR(11,17,46,9)
60 CALL HCHAR(3,14,46,11)
70 CALL HCHAR(13,14,46,11)
80 CALL VCHAR(4,14,46,9)
90 CALL VCHAR(4,24,46,9)
100 FOR I=1 TO 6
110 CALL HCHAR(10-I,7+I,46)
120 CALL HCHAR(10-I,17+I,46)
130 CALL HCHAR(20-I,7+I,46)
140 CALL HCHAR(20-I,17+I,46)
150 NEXT I
160 PRINT TAB(5);"CUBE"
170 GOTO 170

```

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 40 et 50 définissent la position de l'objet.
La ligne 100 calcule la distance, par rapport aux deux coordonnées que vous avez entrées à la ligne 80.
La ligne 110 regarde si vous avez gagné.
Les lignes 130 et 140 normalisent la distance (qui doit avoir un sens, quelles que soient les dimensions de la grille initiale).
La ligne 170 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 ENTER.
Dès l'affichage suivant (DONNEZ K, Y) on entre la coordonnée X supposée, la coordonnée Y supposée suivie de ENTER.
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.
Sortie du programme avant la fin par CLEAR.
- 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 S=0
20 INPUT "DIM. DE LA GRILLE
: ":D
30 RANDOMIZE
40 X=INT(RND*D+1)
50 Y=INT(RND*D+1)
60 CALL CLEAR
70 PRINT : :
80 INPUT "DONNEZ X,Y : ":A,B
90 S=S+1
100 T=ABS(X-A)+ABS(Y-B)
110 IF T=0 THEN 300
120 PRINT : : "VOUS ETES ";
130 N=INT(SQR(D))/T+1
140 IF N<=5 THEN 160
150 N=5
160 IF T>SQR(D)+2 THEN 180
170 ON N GOTO 200,220,240,26
0,280
180 PRINT "COMPLETEMENT GELE
"
190 GOTO 70
200 PRINT "A PEINE TIEDE"
210 GOTO 70
220 PRINT "TIEDE"
230 GOTO 70
240 PRINT "CHAUD"
250 GOTO 70
260 PRINT "TRES CHAUD"
270 GOTO 70
280 PRINT "BRULANT"
290 GOTO 70
300 CALL CLEAR
310 CALL SOUND(100,700,0)
320 CALL SOUND(100,200,0)
330 CALL SOUND(100,400,0)
340 PRINT "VOUS AVEZ TROUVE
EN";S;"COUPS"

```

Les allumettes

- REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : contre le TI-99.
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 TI-99. 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 60.
Celui du TI-99 commence en ligne 100.
La ligne 70 contrôle la légalité du coup du joueur.
La ligne 80 gère le changement de joueur.
Le coup du TI-99 est calculé dans une boucle non formelle, de la ligne 110 à la ligne 140.
Le résultat est affiché par la ligne 210.
Si la partie est terminée, la ligne 220 le détecte et envoie en ligne 240, pour le message de fin.
- POUR JOUER** : On commence l'exécution par RUN, et, à l'affichage du message TOTAL, on entre le nombre d'allumettes, suivi de ENTER. Puis on entre la prise maximum autorisée, suivie de ENTER.
Si le joueur veut commencer, il frappe ensuite 0 ; sinon, il frappe 1, puis ENTER.
Après chaque coup, le nombre d'allumettes restantes est affiché, et le TI-99 affiche A VOUS.
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 CALL CLEAR
20 INPUT "TOTAL ";N
30 INPUT "PRISE MAX ";I
40 INPUT "1ER:VOUS(0),MOI(1)
   ":D
50 IF D THEN 100
60 INPUT "A VOUS ";P
70 IF (P<>INT(P))+(P<1)+(P>I
   )+(P>N) THEN 60
80 D=1
90 GOTO 190
100 S=I+D
110 T=(N-S)/(I+D)
120 IF T=INT(T) THEN 150
130 S=S-D
140 GOTO 110
150 P=ABS(S-D)
160 IF P THEN 180
170 P=D
180 D=0
190 PRINT P
200 N=N-P
210 PRINT "RESTE ";N
220 IF N=0 THEN 240
230 GOTO 50
240 CALL CLEAR
250 PRINT "BRAVO"
260 IF D=0 THEN 280
270 PRINT "POUR MOI"
280 END

```

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 TI-99 le nombre de joueurs.
A chaque coup, le joueur qui a le trait est autorisé à retirer du tas d'allumettes restantes, jusqu'au double de ce que vient de retirer le précédent joueur ; celui qui retire la dernière allumette a gagné (ou perdu, si l'on modifie légèrement les règles).
- LE PROGRAMME** : Les lignes 10 à 40 permettent les différentes entrées.
La ligne 80 réalise les impressions.
La ligne 100 contrôle le coup de chaque joueur (par rapport au coup précédent).
La ligne 140 gère les tours (remise à 1 du numéro du joueur).
La ligne 130 réalise le test de fin, et la ligne 180 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 ENTER, puis le nombre de joueurs (et ENTER). 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 TI-99.
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 éliminés les uns après les autres.
On peut aussi imaginer une représentation graphique des tas d'allumettes, tirer au hasard le nombre à enlever en début de partie, etc.

```
10 CALL CLEAR
20 INPUT "NBRE D'ALLUMETTES
":N
30 PRINT
40 INPUT "NBRE DE JOUEURS ":
J
50 F=1
60 K=J
70 CALL CLEAR
80 PRINT "JOUEUR NO";K: : "NB
RE MAX A OTER";F: : "NBRE TOT
AL";N: : : "VOTRE COUP"
90 INPUT C
100 IF (C=0)+(C>F) THEN 90
110 N=N-C
120 F=C*2
130 IF N=0 THEN 180
140 IF K=1 THEN 160
150 K=J+1
160 K=K-1
170 GOTO 70
180 PRINT : : "LE JOUEUR";K;"
A GAGNE"
```


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 20 et 30 vous permettront d'entrer le mois et l'année.
Les lignes 40 et 50 calculent le numéro du premier jour du mois.
La ligne 60 calcule le nombre de jours du mois (il n'est pas tenu compte ici des 29 février, pour simplifier).
La boucle A (80-120) 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 ENTER, 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 ENTER et le calendrier est alors affiché sur l'écran.
Pour en garder une trace, si l'on dispose bien sûr d'une imprimante, il suffira de modifier les PRINT en conséquence.
- EXTENSIONS POSSIBLES** : On pourra sans difficulté prévoir l'affichage des 29 février, prévoir une option permettant l'affichage pour une année complète, prévoir l'introduction du nom des mois en clair, etc.

```

10 CALL CLEAR
20 INPUT "ANNEE ":A
30 INPUT "MOIS ":M
40 D=INT(365.25*(A-M-3))+INT
   (30.6*(13*M+37))-INT((INT((A
   -M-3)/100)-7)*.75)+1
50 D=D-7*INT(D/7)
60 M=31+((M=4)+(M=6)+(M=9)+(
   M=11))+(M=2)*3
70 PRINT : ":" DIM LUN MAR ME
   R JEU VEN SAM": :
80 FOR A=1 TO M
90 PRINT TAB(4*(D+A-1))+2;ST
   R*(A);
100 IF D+A-7 THEN 120
110 D=D-7
120 NEXT A
130 PRINT : : : :

```

Roulette

- REFERENCES** : Difficulté de programmation : moyenne
Catégorie de jeu : hasard (plusieurs joueurs)
Difficulté du jeu : facile.
- LE JEU** : Il est surtout spectaculaire par son aspect graphique : une boule représentée par un carré noir se déplace sur un terrain de jeu comportant des cases numérotées de 1 à 36, ainsi que trois cases supplémentaires, le 00, une case P (comme pair) et une case I (comme impair). La boule parcourt au moins une fois chaque case, puis s'arrête au deuxième tour ; on peut ensuite la relancer pour une autre partie.
- LE PROGRAMME** : Les lignes (40-80) dessinent le terrain de jeu, auquel les lignes 90-110 rajoutent les trois symboles P, 00 et I.
Les boucles I (170-370) réalise le déplacement de la boule, l'arrêt étant contrôlé par les lignes 210-220.
La ligne 330 réalise un effacement particulier pour le cas où la boule passe sur les symboles P, 00 et I.
La ligne 120 tire la position d'arrêt.
- POUR JOUER** : Il suffit de faire RUN, et de regarder.
Une fois la boule arrêtée, il suffira de frapper une touche quelconque (sauf CLEAR) pour la relancer. Taper CLEAR pour sortir du programme.
- EXTENSIONS POSSIBLES** : On pourra essayer de se rapprocher davantage d'une véritable roulette, de différentes manières : mieux dessiner le terrain de jeu, introduire des couleurs, simuler le ralentissement de la boule, introduire les différentes possibilités de pari, et assurer la gestion des gains correspondants, tenir les comptes de plusieurs joueurs, et annoncer les résultats sous une forme plus usuelle.

```

10 CALL CLEAR
20 RANDOMIZE
30 W=0
40 PRINT "  2  4  6  8 10
12 14 16 1": : : " 20 22 24
26 28 30 32 34 3": : : "
   1  3  5  7  9 11 13 15 1
": : : " 19 21 23 25 27 29
31 33 3": : : : :
50 CALL HCHAR(9,31,56)
60 CALL HCHAR(12,31,54)
70 CALL HCHAR(15,31,55)
80 CALL HCHAR(18,31,53)
90 CALL HCHAR(10,2,80)
100 CALL HCHAR(13,2,48,2)
110 CALL HCHAR(16,2,73)
120 X=37+INT(36*RND)
130 IF X>54 THEN 160
140 LL=11
150 GOTO 170
160 LL=17
170 FOR L=10 TO 19 STEP 3
180 FOR C=6 TO 30 STEP 3
190 CALL HCHAR(L,C,30)
200 CALL SOUND(50,-1,0)
210 X=X-1
220 IF X THEN 350
230 CALL HCHAR(LL,2,30)
240 W=W+1
250 W$=STR$(W)
260 IF W>9 THEN 280
270 W$="0"&W$
280 CALL HCHAR(13,2,ASC(W$))
290 CALL HCHAR(13,3,ASC(SEG$(W$,2,1)))
300 CALL KEY(5,K,S)
310 IF S=0 THEN 300
320 CALL HCHAR(LL,2,32)
330 CALL HCHAR(L,C,32)
340 GOTO 120
350 CALL HCHAR(L,C,32)
360 NEXT C
370 NEXT L
380 GOTO 170

```

Bataille navale

- REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : réflexion.
Difficulté de programmation : 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 TI-99 répond en affichant la distance séparant le navire invisible de la case visée. Mais, si l'on tire trop près le navire risque de se recacher au hasard.
- LE PROGRAMME** : Les boucles I (30-90) et J (40-80) dessinent le terrain de jeu.
Les lignes 110 et 120 définissent l'emplacement du navire.
Les coordonnées du tir sont entrées aux lignes 180 et 230, et la ligne 250 affiche le coup sur l'écran. La distance est calculée en ligne 260, et la ligne 270 regarde si l'on a ou non fait mouche.
La ligne 310 intervient pour déplacer le navire si les coups sont trop rapprochés.
La ligne 410 permet l'affichage de la distance.
- POUR JOUER** : Le terrain de jeu est affiché après avoir taper RUN. A l'affichage de H, on répond en donnant le numéro de la ligne horizontale sur laquelle on désire tirer, et, à la question V, on répond en donnant le numéro de la colonne verticale.
Si l'on a tiré trop près, on obtient l'affichage PFTT... ce 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 TI-99, mettre plusieurs navires, des sous-marins, etc.

```

10 CALL CLEAR
20 A=0
30 FOR I=A TO 9
40 FOR J=A TO 9
50 CALL HCHAR(4,9+2*J,48+J)
60 CALL HCHAR(5+I,7+A,48+I)
70 CALL HCHAR(5+I,2*J+9,46)
80 NEXT J
90 NEXT I
100 S=1
110 X=INT(10*RND)
120 Y=INT(10*RND)
130 CALL HCHAR(19,1,32,32)
140 A=19
150 B=8
160 T$="H="
170 GOSUB 510
180 GOSUB 560
190 H=K-48
200 B=18
210 T$="V="
220 GOSUB 510
230 GOSUB 560
240 V=K-48
250 CALL HCHAR(5+H,9+2*V,48+
S)
260 D=INT(SQR((X-H)^2+(Y-V)^
2))
270 IF D=0 THEN 480
280 S=S+1
290 IF S=10 THEN 32767
300 S=0
310 IF D<>1 THEN 400

```

```

320 T$="PFTT...."
330 A=22
340 B=13
350 GOSUB 510
360 FOR I=1 TO 1000
370 NEXT I
380 CALL HCHAR(A,1,32,32)
390 GOTO 110
400 CALL HCHAR(21,1,32,32)
410 T$="DISTANCE="&STR$(D)
420 A=21
430 B=7
440 GOSUB 510
450 FOR I=1 TO 500
460 NEXT I
470 GOTO 130
480 CALL CLEAR
490 PRINT ,S;"COUPS": : : :
: : : : :
500 END
510 REM ROUTINE D'AFFICHAGE
520 FOR I=1 TO LEN(T$)
530 CALL HCHAR(A,B+I,ASC(SEG
$(T$,I,1)))
540 NEXT I
550 RETURN
560 REM ROUTINE DE SAISIE
570 CALL SOUND(100,-1,0)
580 CALL KEY(S,K,L)
590 IF L=0 THEN 580
600 CALL HCHAR(A,B+I,K)
610 FOR P=1 TO 400
620 NEXT P
630 RETURN

```

Golf

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté de programmation : 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 60 permet de définir la position du trou, et la ligne 80 dessine le terrain de jeu. Le déplacement à réaliser est calculé en ligne 130.
La boucle I (150-200) réalise le déplacement de la balle.
La ligne 210 regarde si le coup est, ou non gagnant ; s'il l'est, on recommence (ligne 260) jusqu'à ce que l'on ait réalisé 18 trous (boucle S (40-260) ; fin de boucle en 280, avec affichage de fin).
- POUR JOUER** : On obtient l'affichage du terrain en frappant RUN ainsi que le dessin de la balle sur sa position de départ, et le texte : TROU 1. FORCE ; on doit alors répondre en donnant la force désirée (un nombre entre 1 et 300...). Dès que l'on aura tapé ENTER, la balle commence à se déplacer ; elle peut alors soit s'arrêter avant le trou (auquel cas on recommence), soit tomber dedans (on passe alors au trou suivant), soit le dépasser : on devra alors frapper avec une force négative. Le jeu se termine avec l'affichage 18 TROUS EN ... COUPS.
- EXTENSIONS POSSIBLES** : Il sera possible d'étendre le jeu pour faire participer plusieurs joueurs, d'améliorer la qualité graphique, d'utiliser une représentation spatiale du terrain (se référer à certains jeux d'arcades), d'avoir un terrain plus accidenté, des obstacles, de changer de club...

```

10 RANDOMIZE
20 CALL CHAR(128,"FFFFFFFFF
")
30 E=0
40 FOR S=1 TO 18
50 X=2
60 T=INT(22*RND+9)
70 CALL CLEAR
80 CALL HCHAR(16,1,128,32)
90 CALL HCHAR(16,T,32)
100 CALL HCHAR(15,X,48)
110 A$=" TROU "&STR$(S)&" F
ORCE "
120 INPUT A$:F
130 F=INT(F*.16-(ABS(F)*(SQR
(RND)+.1)^2)/1000)
140 IF F=0 THEN 70
150 FOR I=SGN(F) TO F STEP S6
N(F)
160 X=X+SGN(F)
170 IF (X>=32)+(X<2) THEN 200
180 CALL HCHAR(14,X-SGN(F),3
2)
190 CALL HCHAR(14,X,48)
200 NEXT I
210 IF X=T THEN 240
220 E=E+1
230 GOTO 70
240 CALL HCHAR(14,X,32)
250 CALL HCHAR(15,X,48)
260 NEXT S
270 CALL CLEAR
280 PRINT "18 TROUS EN";E;"C
OLUPS": : : : :

```


Solfège

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : facile.
- LE JEU** : Ce programme vous propose une initiation élémentaire au solfège, en vous permettant d'appréhender 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 (80-100) dessine la portée.
La note est choisie en ligne 110 et affichée en lignes 120, 130 et 140.
La ligne 150 affiche le score.
La ligne 160 affiche le nom de la clé (ici clé de sol) et vous interroge.
La ligne 170 s'occupe de la fin de partie.
La ligne 180 effectue le décodage de la note entrée.
La ligne 200 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, si), suivi de ENTER.
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 ENTER, à la question NOTE ?
- EXTENSIONS POSSIBLES** : On pourra tout d'abord facilement modifier le programme pour lui permettre de vous faire travailler les différentes clés.
On pourra aussi améliorer le graphisme, introduire la couleur, le son (lorsque l'on a répondu juste, on entendra la note, par exemple)
On pourra aussi réaliser suivant le même principe un jeu vous permettant de reconnaître des accords, etc.

```

10 RANDOMIZE
20 CALL CHAR(128,"FFFFFFFFF
FFFFF")
30 CALL CHAR(129,"0C0C0C0C0C
0C0C0C")
40 CALL CHAR(130,"0C0C3C7C7C
7C3C1C")
50 S=0
60 P=S
70 CALL CLEAR
80 FOR I=1 TO 5
90 CALL HCHAR(10+2*I,1,128,3
2)
100 NEXT I
110 X=INT(12*RND)
120 CALL HCHAR(9+X,9,129)
130 CALL HCHAR(10+X,9,129)
140 CALL HCHAR(11+X,9,130)
150 PRINT "SCORE";S;"SUR";P;
"ESSAIS": : :
160 INPUT "CLE DE SOL , NOTE
? ":R$
170 IF R$="0" THEN 230
180 Y=-1*((R$="FA")+2*(R$="M
I")+3*(R$="RE")+4*(R$="DO")+
5*(R$="SI")+6*(R$="LA"))
190 P=P+1
200 IF (Y-X)*(Y-X+7) THEN 220
210 S=S+1
220 GOTO 70
230 END

```

Le jeu du 21

REFERENCES

: Difficulté de programmation : facile.
Catégorie de jeu : jeu contre le TI-99.
Difficulté du jeu : facile.

LE JEU

: On devra entrer une mise au début du jeu, et on pourra lancer les dés en tapant 1 ; à chaque lancer, on obtiendra un nouveau total, qui augmentera peu à peu, et se rapprochera de 21. On devra essayer de se rapprocher le plus possible de 21, sans dépasser ce nombre. Lorsque l'on décidera de s'arrêter, il suffira de taper 0 pour laisser jouer le TI-99, qui essaiera alors d'avoir un meilleur score. A la fin de chaque partie, le gain sera affiché, ainsi que le total des gains (ou des pertes).

LE PROGRAMME

: Les lignes 60, 70 et 80 réalisent le contrôle de celui qui doit jouer (joueur ou TI-99). Le sous-programme 340-350 lance les dés. Les lignes 100 et 110 calculent le total fait par le joueur et l'affichent. Le total du TI-99 est calculé en ligne 170 et affiché par la ligne 180. La ligne 210 contrôle la fin du jeu pour le TI-99 et la ligne 250 cherche quel est le gagnant. Le gain (ou la perte) sera affiché par la ligne 270, et le total par la ligne 290. La ligne 310 contrôle la fin de partie.

POUR JOUER

: Taper RUN pour lancer le jeu.
A la question VOTRE MISE, entrer la mise initiale, suivie de ENTER, et taper 1 pour lancer le dé. Le premier tirage est affiché sous la forme VOUS AVEZ... et on devra refaire 1 suivi de ENTER, tant que l'on désire recommencer à tirer.
Dès que l'on a terminé, on tape 0 et ENTER pour laisser jouer le TI-99 (qui tient ici le rôle de la banque).
Le gain (ou la perte, s'il y a un signe -) sera affiché en fin de jeu, ainsi que le total des gains (ou pertes) réalisé depuis le début. Si l'on désire arrêter de jouer, on devra taper N en réponse à la question SUITE (O ou N) ?

EXTENSIONS POSSIBLES : Ce jeu sera évidemment plus intéressant s'il est pratiqué à plusieurs, et il sera aisé de modifier le programme en conséquence ; on pourra aussi dessiner le dé, ou mieux, utiliser des cartes, ce qui permettra alors de faire un jeu de BLACK JACK.

```

10 RANDOMIZE
20 G=0
30 CALL CLEAR
40 INPUT "VOTRE MISE ":M
50 T=0
60 PRINT : : "1 pour TIRER 2
   pour FINIR": :
70 INPUT R
80 IF R=0 THEN 130
90 GOSUB 340
100 T=T+X
110 PRINT : : "TOTAL :";T
120 GOTO 70
130 CALL CLEAR
140 PRINT "LA BANQUE JOUE"
150 S=0
160 GOSUB 340
170 S=S+X
180 PRINT : : "TOTAL :";S
190 FOR I=1 TO 500
200 NEXT I
210 IF (S>T)+(S>=17) THEN 230
220 GOTO 160
230 CALL CLEAR
240 PRINT : : "VOUS AVEZ";T:
   : "LA BANQUE A";S
250 IF (S>21)+(S<T)*(T<=21) T
   HEN 270
260 M=-M
270 PRINT : : "GAIN";M
280 G=G+M
290 PRINT : : : "TOTAL";G
300 INPUT "SUITE (O OU N)? "
   :R$
310 IF R$="N" THEN 330
320 GOTO 30
330 END
340 X=INT(RND*6+1)
350 RETURN

```

Pair - impair

- REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.
- LE JEU** : C'est un jeu de hasard, pour un ou plusieurs joueurs. Le TI-99 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 360 et 370.
La ligne 70 tient le compte du nombre de parties, et les lignes 160, 240 et 280 comptabilisent les points marqués.
La parité du total est déterminée par les lignes 110 et 120 (la ligne 120 compare avec la réponse du joueur).
La ligne 310 réalise l'affichage de fin de partie : total, nombre de points, nombre de parties jouées.
La ligne 330 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 ENTER 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, 6/10.
On obtient alors l'affichage de la question ON CONTINUE (O ou N) et l'on devra répondre N suivi de ENTER pour arrêter.
- EXTENSIONS POSSIBLES** : Si l'on désire jouer à plusieurs, il sera intéressant de prévoir la possibilité de miser, et d'ajouter une gestion du score réalisé par chacun des joueurs.
On pourra aussi prévoir une amélioration sur le plan de la présentation (utilisation de la couleur, etc.).

```

10 RANDOMIZE
20 T=0
30 R=T
40 GOSUB 360
50 A=X
60 GOSUB 360
70 R=R+1
80 S=A+X
90 CALL CLEAR
100 INPUT "TOTAL PAIR (0) OU
IMPAIR (1) ":N
110 P=S/2-INT(S/2)
120 IF (N=P)+(N-.5=P) THEN 15
0
130 PRINT : : "FAUX"
140 GOTO 170
150 PRINT : : "VRAI"
160 T=T+1
170 PRINT : :
180 INPUT "VOTRE TOTAL ":N
190 IF N=S THEN 240
200 PRINT : : "PERDU"
210 FOR P=1 TO 300
220 NEXT P
230 GOTO 300
240 T=T+2
250 PRINT : :
260 INPUT "BRAVO, DONNEZ UN
DE ? ":N
270 IF (N<>A)+(N<>5-A) THEN 2
00
280 T=T+3
290 GOTO 210
300 CALL CLEAR
310 PRINT "TOTAL :";T;"/";R:
:
320 INPUT "ON CONTINUE (0 OU
N) ":R$
330 IF R$="N" THEN 350
340 GOTO 40
350 END
360 X=INT(RND*6+1)
370 RETURN

```

Simon

- 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 aura un de plus à taper à chaque fois).
Il devra essayer de tenir le plus longtemps possible. Le meilleur score est mémorisé après chaque partie, ce qui permettra éventuellement de jouer à plusieurs.
- LE PROGRAMME** : La ligne 30 appelle un sous-programme (lignes 270 à 300) qui donne la règle du jeu.
La ligne 80 fabrique les chiffres, qui sont ensuite stockés dans une variable alphanumérique A\$ (ligne 90).
La ligne 100 permet l'impression du dernier chiffre au hasard sur la sixième ligne de l'écran, et la durée de l'affichage est déterminée par la ligne 120.
Les lignes 150 et 160 attendent la réponse du joueur, qui est comparée à la réponse correcte en ligne 170.
La ligne 190 marque les points.
La ligne 200 contrôle le meilleur score.
Les affichages de fin de partie sont effectués de la ligne 230 à la ligne 260.
- 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 ENTER. Si c'est exact, on a un court instant l'affichage de EXACT. ATTENTION, puis un nouveau chiffre est affiché, qui s'efface et de nouveau la question VOTRE SEQUENCE ?
On retape alors deux chiffres déjà obtenus, suivis de ENTER.
On continue ainsi (3 chiffres, puis 4, 5, etc.) jusqu'à ce que l'on fasse une erreur, ce qui provoque l'affichage du score (et du meilleur score obtenu jusqu'à présent), suivi de la question AUTRE PARTIE (O ou N)? ; on tape O pour continuer.
On peut aussi quitter le programme à tout moment, en tapant CLEAR.

102 PROGRAMMES POUR TI-99/4A

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

```
10 M=0
20 A$=""
30 GOSUB 270
40 H=0
50 FOR I=1 TO 300
60 NEXT I
70 RANDOMIZE
80 X=INT(10*RND)
90 A$=A$&STR$(X)
100 PRINT TAB(28*RND+4);X
110 CALL SOUND(100,700,0)
120 FOR I=1 TO 300
130 NEXT I
140 GOSUB 270
150 PRINT "VOTRE SEQUENCE ?"
160 INPUT R$
170 IF R$<>A$ THEN 230
180 PRINT "EXACT . ATTENTION"
"
190 H=H+1
200 IF M>=H THEN 50
210 M=H
220 GOTO 50
230 CALL CLEAR
240 PRINT "VOTRE SCORE";H;"M
EILLEUR SCORE";M
250 INPUT "AUTRE PARTIE (O/N
) ? ";R$
260 IF R$="N" THEN 300 ELSE
20
270 CALL CLEAR
280 PRINT "VOUS DEVEZ REPETE
R TOUS LES NOMBRES(UN DE PLU
S A CHAQUE FOIS)"
290 RETURN
300 END
```


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 ; 3 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 260-270.
L'affichage est réalisé par la ligne 120.
La ligne 140 calcule le gain dans les diverses configurations possibles.
La totalisation est effectuée en ligne 170 pour les gains, et en ligne 200 pour les pertes.
La variable R\$, testée en ligne 230 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 ENTER. On obtient alors un deuxième affichage UN CHIFFRE DE 1 A 6 ? ; on entre un chiffre suivi de ENTER ; les 3 dés, repérés par les lettres A, B, C s'affichent alors, ainsi qu'un message indiquant votre gain ou votre perte. Puis une indication sur votre total actuel apparaît, ainsi que la question ON CONTINUE (O ou N) ? ; taper O pour continuer, N pour arrêter (suivi de ENTER).

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 TI-99,...

```

10 G=0
20 CALL CLEAR
30 INPUT "VOTRE MISE : ":N
40 PRINT : :
50 INPUT "UN CHIFFRE DE 1 A
6 ?":D
60 GOSUB 260
70 A=X
80 GOSUB 260
90 B=X
100 GOSUB 260
110 C=X
120 PRINT : : :TAB(4);A;TAB(
8);B;TAB(12);C: :TAB(5);"A";
TAB(9);"B";TAB(13);"C"
130 T=0
140 T=T-N*((D=A)+(D=B)+(D=C)
)
150 IF T=0 THEN 190
160 PRINT : : "VOUS GAGNEZ";T
170 G=G+T
180 GOTO 210
190 PRINT : : "PERDU"
200 G=G-N
210 PRINT : : "TOTAL ACTUEL :
";G
220 INPUT "ON CONTINUE (O OU
N) ?":R$
230 IF R$="N" THEN 250
240 GOTO 20
250 END
260 X=INT(RND*6+1)
270 RETURN

```

Le jeu de Wytthof

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : moyenne.
- LE JEU** : On dispose d'une grille carrée de 10 cases sur 10 cases. Un point apparaît, en dehors de la diagonale et les joueurs devront le déplacer à chaque tour, suivant seulement trois directions : droite, bas et diagonale bas et droite. Le premier joueur qui arrive en (9, 9) sera considéré comme gagnant, ou perdant, selon la convention faite au début.
- LE PROGRAMME** : Les lignes 30, 40 et 50 permettent de définir une position de départ, en dehors de la diagonale.
La boucle I (70-110) dessine le terrain de jeu.
La ligne 170 imprime la position du point mobile.
Le test de fin est effectué en ligne 115.
Le changement de joueur est assuré par les lignes 190 et 210 selon la valeur de H.
Les directions de déplacement disponibles, 1 et 2, sont décodées par trois sous-programmes qui commencent aux lignes 290, 310 et 320.
- 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 N° 1, 2, 3 ?, à laquelle on répond en tapant l'un de ces trois nombres.
C'est ensuite au tour du joueur 2.
Le jeu se termine lorsque l'un des joueurs arrive en case (9,9).
- EXTENSIONS POSSIBLES** : On peut tout d'abord améliorer les contrôles (la sortie du terrain n'est pas interdite, par exemple), améliorer ensuite l'affichage (graphisme, couleur) étendre le jeu à plusieurs joueurs avec un terrain agrandi, introduire un monstre qui va essayer de capturer l'un des joueurs, prévoir un programme qui permettrait au TI-99 de jouer, etc.

```

10 CALL CLEAR
20 RANDOMIZE
30 X=INT(5*RND)
40 Y=INT(5*RND)
50 IF X=Y THEN 40
60 A=0
70 FOR I=0 TO 9
80 CALL HCHAR(3,13+I,48+I)
90 CALL HCHAR(4+I,12,48+I)
100 CALL HCHAR(4+I,13,46,10)
110 NEXT I
120 H=1
130 T$="JUEUR NO      1,2,3
    ?"
140 FOR I=1 TO 22
150 CALL HCHAR(20,6+I,ASC(SE
G$(T$,I,1)))
160 NEXT I
170 CALL HCHAR(4+X,13+Y,30)
180 IF X+Y=18 THEN 280
190 CALL HCHAR(20,17,50)
200 IF H=1 THEN 220
210 CALL HCHAR(20,17,49)
220 CALL KEY(5,K,S)
230 IF (S=0)+(S=-1) THEN 220
240 CALL SOUND(70,-1,0)
250 ON K-48 GOSUB 290,310,32
0
260 H=-H
270 GOTO 170
280 STOP
290 X=X+1
300 RETURN
310 GOSUB 290
320 Y=Y+1
330 RETURN

```

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 joueur est tout d'abord entré en ligne 20.
La ligne 100 affiche le numéro de joueur, et le nombre d'essais déjà effectués.
Un sous-programme placé en ligne 280, utilisé 3 fois dans la boucle J (140-170) réalise le lancer de dés.
Les retirages sont effectués par les lignes 220, 240, 260.
La variable R\$ gère les 2èmes et 3èmes tirages.
L'impression du résultat est réalisée dans tous les cas par la ligne 100.
Les tests pour le nombre d'essais et pour le changement de joueur sont effectués aux lignes 180 et 200.
- POUR JOUER** : On lance l'exécution du programme par RUN, ce qui conduit à l'affichage NBR. DE JOUEURS.
On entre le nombre de joueurs, et l'on tape ENTER. L'écran s'efface et affiche JR N° 1.
ESSAI A, ainsi que le tirage réalisé, et les noms des dés (A, B ou C).
Si l'on ne désire pas refaire de tirage, il suffit de frapper ENTER. 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 ENTER : par exemple, pour retirer A et C on tapera AC suivi de ENTER.
Les joueurs suivants procéderont de la même façon.
- EXTENSIONS POSSIBLES** : La plus évidente consiste à assurer la gestion des résultats des joueurs, en se référant aux règles de gain usuelles du jeu du 421.
On pourra envisager l'affichage après chaque tour d'un tableau récapitulant les résultats des joueurs.
On pourra aussi améliorer l'affichage des dés (se référer aux divers programmes d'affichage de dés se trouvant dans le même ouvrage).

```

10 RANDOMIZE
20 INPUT "NBR. DE JOUEURS ":
N
30 S=1
40 I=S

50 K=0
60 CALL CLEAR
70 R$="ABC"
80 GOTO 140
90 K=K+S
100 PRINT , "421": , "----": : "J
R. NO. ": I ; "ESSAI"; K : : TAB(
4) ; A ; TAB(8) ; B ; TAB(12) ; C :
: TAB(5) ; "A" ; TAB(9) ; "B" ; TAB(
13) ; "C" : : :
110 INPUT "ABC ? ": R$
120 IF R$="" THEN 190
130 R$=R$&"DDD"
140 FOR J=S TO 3
150 GOSUB 280
160 ON ASC(SEG$(R$,J,1))-64
GOSUB 220,240,260,280
170 NEXT J
180 IF K<3 THEN 90
190 I=I+S
200 IF I<=N THEN 50
210 GOTO 40
220 A=D
230 RETURN
240 B=D
250 RETURN
260 C=D
270 RETURN
280 D=INT(RND*5+S)
290 RETURN

```

Rebonds

- REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : jeu d'adresse.
Difficulté du jeu : moyenne.
- LE JEU** : Il utilise l'idée de la simulation d'un billard à bandes.
De part et d'autre de l'écran se trouvent la balle et le but visé ; une série de chiffres et de lettres, en haut de l'écran, permettent de choisir l'endroit où l'on désire rebondir. La balle va alors partir vers le haut de l'écran (vers le bas si l'on introduit un signe moins), y rebondir, revenir éventuellement frapper le bas, rebondir, et ainsi de suite jusqu'à atteindre la ligne de but ; si l'on est suffisamment près de la cible on a gagné, et le nombre de rebonds sera alors affiché.
- LE PROGRAMME** : Les boucles Z (30-50) et (60-80) impriment les chiffres et les lettres en haut de l'écran. Les lignes 90 et 100 définissent les positions initiales de la balle et de la cible. Les lignes (110-120) effectuent leur impression. Le symbole correspondant à la direction choisie est entré en ligne 130. La boucle I (160-200) réalise le déplacement de la balle. La ligne 190 affiche la nouvelle position. Le changement de direction, lors de la rencontre d'un bord est effectué à la ligne 180, par renvoi dans une partie de programme indépendante, située aux lignes 250 à 270. La ligne 220 effectue l'impression de fin de partie. Le nombre de rebonds est compté en ligne 260.
- POUR JOUER** : On lance le programme par RUN.
Le terrain de jeu est affiché, ainsi que les positions initiales de la balle et de la cible. On choisit l'endroit où l'on désire faire rebondir la balle (de 1 à U), et on l'entre, suivi de ENTER.
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 CLEAR.
- EXTENSIONS POSSIBLES** : Il sera très intéressant, sur un appareil disposant d'un graphisme plus fin, de dessiner un véritable billard par un segment, symbolisant la queue de billard, se déplaçant

autour de la balle.

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

```

10 CALL CLEAR
20 RANDOMIZE
30 FOR I=2 TO 6
40 CALL HCHAR(5,I,51+I)
50 NEXT I
60 FOR I=7 TO 32
70 CALL HCHAR(5,I,58+I)
80 NEXT I
90 X=INT(15*RND+6)
100 Y=INT(15*RND+6)
110 CALL HCHAR(X,2,48)
120 CALL HCHAR(Y,32,30)
130 INPUT R$
140 S=0
150 T=X
160 FOR I=2 TO 32
170 T=T-X/(ASC(R$)-48+7*(R$>
="A"))
180 IF (T>=22)+(T<=5) THEN 25
0
190 CALL HCHAR(T,I,48)
200 NEXT I
210 IF ABS(T-Y)>=3 THEN 230
220 PRINT "BRAVO :";S;"REBON
DS"
230 CALL KEY(5,K,F)
240 IF F=0 THEN 230 ELSE 10
250 X=-X
260 S=S+1
270 GOTO 200

```


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 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, les lignes 70, 80 et 90 affichent les emplacements (vides) des chiffres.
La mise sera entrée en ligne 100, et testée en ligne 110 : si on mise 0, la partie est terminée (départ vers la ligne 360).
Les lignes 120, 140 et 160 permettent le tirage des 3 chiffres, en appelant un sous-programme qui se trouve en ligne 340.
Le résultat est affiché par les lignes 170, 190 et 210, puis testé en ligne 230, et en ligne 310, si l'on a gagné.
- POUR JOUER** : Lancer le programme par RUN ; à l'affichage de la question POT DE DEBUT, on répond en entrant un nombre suffisamment élevé, sur lequel vos mises seront décomptées.
Puis on tape ENTER, et on entre une première mise (différente de 0), suivie de ENTER.
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 ENTER et ainsi de suite.
Pour terminer, il suffit de répondre 0 à la question VOTRE MISE.
Le restant de votre pot initial sera alors affiché.
- EXTENSIONS POSSIBLES** : On pourra envisager d'améliorer sérieusement les contrôles, utiliser la couleur et des possibilités graphiques plus élaborées pour simuler davantage un véritable JACKPOT.

```

10 INPUT "POT DE DEBUT ";P
20 CALL CHAR(128,"FFFFFFFFF
  FFFFFF")
30 CALL CLEAR
40 FOR I=1 TO 3
50 CALL HCHAR(10+I,11,128,9)
60 NEXT I
70 CALL HCHAR(12,12,32)
80 CALL HCHAR(12,15,32)
90 CALL HCHAR(12,18,32)
100 INPUT "VOTRE MISE ":M
110 IF M=0 THEN 360
120 GOSUB 340
130 A=X
140 GOSUB 340
150 B=X
160 GOSUB 340
170 CALL HCHAR(11,12,48+A)
180 CALL SOUND(20,-1,0)
190 CALL HCHAR(11,15,48+B)
200 CALL SOUND(20,-1,0)
210 CALL HCHAR(11,18,48+X)
220 CALL SOUND(20,-1,0)
230 IF (A=B)*(A=X) THEN 300
240 N=-M
250 PRINT "MISE";M;" RESULTA
  T";N
260 FOR L=1 TO 500
270 NEXT L
280 P=P+N
290 GOTO 30
300 N=M*50
310 IF A<>7 THEN 330
320 N=M*500
330 GOTO 250
340 X=INT(9*RND+1)
350 RETURN
360 CALL CLEAR
370 PRINT "VOUS AVEZ";P,"A B
  IENTOT"

```

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 WYTHOF, cette version offre en fait 4 possibilités de déplacement ; la position de départ est ici fixée en (0, 0), et pourrait éventuellement être modifiée.
Les directions de déplacement disponibles sont : droite, bas, diagonale bas et droite, diagonale bas et gauche.
On joue en principe avec la règle : le premier arrivé en (9, 9) a perdu.
- LE PROGRAMME** : Les boucles emboîtées I (30-90) et J (40-80) dessinent le terrain de jeu.
La ligne 140 réalise le test de fin de partie.
La ligne 220 permet d'exécuter l'un des sous-programmes de déplacement.
Les lignes 230 et 240 assurent le changement de joueur.
- POUR JOUER** : On lance le programme par RUN, et le terrain de jeu s'affiche, ainsi que la question DIR 1, 2, 3, 4 ? ; les 4 directions proposées correspondent aux déplacements suivants :



- Le joueur 1 entre son déplacement.
Le mouvement correspondant est réalisé, puis on obtient l'affichage JR 2, et DIR...
Le joueur 2 procède de même, et ainsi de suite jusqu'à ce que l'un des joueurs arrive en (9, 9) auquel cas le jeu se termine.
- EXTENSIONS POSSIBLES** : On peut imaginer d'introduire en début de partie des cases pièges (chacun essayant alors de faire venir l'autre sur l'une de ces cases).
On peut aussi jouer avec plusieurs symboles, si possible de couleurs différentes, déplacés simultanément, envisager des terrains de jeu triangulaires, aux dimensions arbitraires, etc.

```

10 CALL CLEAR
20 A=0
30 FOR J=0 TO 9
40 FOR I=0 TO 9
50 CALL HCHAR(S,2*I+6,48+I)
60 CALL HCHAR(J+7,4,48+J)
70 CALL HCHAR(J+7,2*I+6,30)
80 NEXT I
90 NEXT J
100 X=7
110 Y=6
120 J=1
130 CALL HCHAR(X,Y,42)
140 IF (X=16)*(Y=24) THEN 250
150 T$="JOUEUR NO "&STR$(J)&
"   DIR 1,2,3,4 ?"
160 FOR I=1 TO 27
170 CALL HCHAR(22,3+I,ASC(SEG$(T$,I,1)))
180 NEXT I
190 CALL SOUND(70,-1,0)
200 CALL KEY(S,K,S)
210 IF S=0 THEN 200
220 ON K-48 GOSUB 260,280,29
Q,310
230 J=J+1
240 IF J=3 THEN 120 ELSE 130
250 END
260 Y=Y+2
270 RETURN
280 GOSUB 260
290 X=X+1
300 RETURN
310 GOSUB 290
320 Y=Y-2
330 RETURN

```

Jeu de l'oie aléatoire

- REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : hasard (deux joueurs).
Difficulté du jeu : facile.
- LE JEU** : Le jeu reprend la très ancienne idée du jeu de l'oie, mais en y ajoutant la possibilité offerte par l'ordinateur de choisir au hasard les cases "à problème".
La règle du jeu est sans grand intérêt, puisque les joueurs n'interviennent à aucun moment dans le déroulement de la partie, qui est entièrement gérée par le TI-99 ; il suffira d'appuyer sur une touche après chaque lancer de dé, et de regarder ce qui se passe.
- LE PROGRAMME** : La ligne 70 permet d'appeler le sous-programme (300-470) 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 80 s'occupe du changement de joueur, et la ligne 90 effectue le lancer de dé.
La ligne 100 traite le cas où l'un des joueurs termine.
La ligne 160 tire un nombre aléatoire entre 0 et 3, qui servira s'il y a lieu pour déterminer le vrai déplacement du joueur, en envoyant dans l'un des sous-programmes de "chance".
Les lignes 300 à 360 font avancer les pions, et les lignes 370 à 460 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 CLEAR.

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

```

10 RANDOMIZE
20 CALL CHAR(130,"FFFFFFFFF
FFFFF")
30 X=2
40 Y=X
50 J=1
60 I=X
70 GOSUB 300
80 J=ABS(J-1)
90 I=INT(6*RND+1)
100 IF (X-32)*(Y-32) THEN 120
110 PRINT "BRAVO ";2-J
120 PRINT "JR";J+1;" ";I
130 CALL KEY(5,KK,SS)
140 IF SS=0 THEN 130
150 GOSUB 300
160 I=INT(4*RND)
170 IF RND>=.5 THEN 70
180 ON 1+I GOSUB 200,210,230
,250
190 GOTO 70
200 RETURN
210 I=-I
220 RETURN
230 I=I*2
240 RETURN
250 IF J THEN 280
260 X=INT(2+31*RND)
270 GOTO 290
280 Y=INT(2+31*RND)
290 RETURN
300 IF J THEN 340
310 X=X+I*(X-(I<=32))
320 IF X<=32 THEN 340
330 X=32
340 Y=Y+J*I*(Y-(I<=32))
350 IF Y<=32 THEN 370
360 Y=32
370 CALL CLEAR
380 FOR I=2 TO 6
390 CALL HCHAR(13,I,51+I)
400 NEXT I
410 FOR I=7 TO 32
420 CALL HCHAR(13,I,58+I)
430 NEXT I
440 CALL HCHAR(14,1,130,64)
450 CALL HCHAR(14,X,49)
460 CALL HCHAR(15,Y,50)
470 RETURN

```


NIVEAU 2

<i>N°</i>	<i>Nom du programme</i>	<i>Thème et particularités</i>
1	Un dé	Tirage d'un dé, avec graphisme
2	Visions	Un jeu sur les mots, pour l'ordinateur !
3	Plus grand, plus petit	Un jeu de déduction intéressant, pour un joueur.
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	Dé truqué	A utiliser, si vous êtes malhonnête, dans d'autres programmes.
8	Le jeu de saute-mouton	Un jeu de réflexion, pour 2 joueurs.
9	Lettres	Un jeu de mots, assez simple (2 joueurs).
10	Espion	Offrez un TI-99 à votre fiancée, et dites-lui votre amour en code...
11	Accrochez les wagons	Pour les plus jeunes, très visuel.
12	Tirage d'un dé truqué	Révissez vos connaissances en statistique...
13	Ordre	Un peu d'astuce pour remettre en ordre, le plus vite possible...
14	Course de chevaux	Assez visuel ; un jeu de hasard pour plusieurs joueurs.
15	Le jeu de Marienbad	Un jeu de logique pour 2 joueurs dans une version étendue.
16	Idem	Un jeu de calcul mental, pour 2 joueurs (plusieurs niveaux).
17	Mastermind	Avec des chiffres, mais très classique ; un seul joueur.
18	A travers	Un jeu de stratégie élémentaire, pour 2 joueurs.
19	Reines sur l'échiquier	Un vieux problème, pour vous aider à réfléchir.
20	Le jeu de la vie	Un grand classique : deux versions, manuelle et automatique.

Dimensionnement d'un tableau

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

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

Les structures de tableaux sont très couramment utilisées en programmation, se révélant même indispensable dans un grand nombre de cas. Si vous voulez, par exemple, entrer vos recettes et vos dépenses pour tous les jours du mois (on supposera pour simplifier des mois de trente jours) vous pourriez définir 30 variables appelées PREMIER JOUR, DEUXIEME JOUR, etc. Il est plus facile, et plus rationnel d'utiliser une boucle I, I variant de 1 à 30, et une variable R (I) pour les recettes, et D (I) pour les dépenses ; ce faisant, vous avez, sans le savoir utilisé une structure de tableau à une dimension ; ceci ne pourra être admis par le TI-99 que si vous l'avez auparavant prévenu que vous alliez utiliser un tel tableau, en donnant une instruction DIM R (30) pour les recettes, et DIM D (30) pour les dépenses.

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

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

Ainsi DIM NOM\$ (5) réserve 5 emplacements en mémoire, pour des mots.

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

Si l'on a un tableau à une dimension, défini par une instruction comme DIM R (30), le Ième élément (numérique) de ce tableau est repéré par R (I) ; il sera alors manipulé exactement comme une variable numérique ordinaire (affectation, calculs, 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 d'afficher le 5ème élément du même tableau R ; etc.

EXTRACTION D'UN ELEMENT DANS UNE CHAINE

Dans le Basic Texas les instructions classiques **MID\$**, **RIGHT\$**, et **LEFT\$** sont absentes. Elles sont néanmoins remplaçables par l'unique instruction **SEG\$** (chaîne, début, longueur). En effet :

MID\$ (R\$, I, J) = SEG\$ (R\$, I, J)

RIGHT\$ (R\$, R) = SEG\$ (R\$, LEN (R\$) - R + I, R)

LEFT\$ (R\$, L) = SEG\$ (R\$, 1, L)

Rappelons rapidement la syntaxe de l'instruction **SEG\$**.

SEG\$ (R\$, P, L) extrait de la chaîne **R\$** le segment débutant à la position **P** et de la longueur **L**.

Ainsi **A\$ = SEG\$ ("EXEMPLE", 3, 4)** affecte à la variable **A\$** la valeur "EMPL".

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 être facilement utilisé en sous-programme pour tout type de jeu faisant appel au lancer d'un ou plusieurs dés.
- LE PROGRAMME** : La ligne 30 utilise le générateur aléatoire du TI-99 pour tirer un entier de 1 à 6.
Les lignes 60 et 70 dessinent les bords du dé et les lignes 50 et 80 dessinent le haut et le bas.
Les 7 lignes 100, 120, 140, 150, 170, 190 et 210 sont utilisées pour le dessin des points.
La ligne 230 permet le cas échéant de refaire un tirage.
- POUR JOUER** : On lance le dé par RUN, le résultat est affiché sous forme symbolique, comme sur un véritable dé.
La question AUTRE TIRAGE (O ou N) est ensuite affichée. Si l'on répond O, un autre lancer sera effectué.
- EXTENSIONS POSSIBLES** : On pourra utiliser ce programme en tant que sous-programme chaque fois que l'on a besoin de simuler un lancer de dé.

```

10 CALL CLEAR
20 RANDOMIZE
30 A=INT(1+6*RND)
40 CALL CHAR(128,"FFFFFFFFF
FFFFF")
50 CALL HCHAR(5,8,128,9)
60 CALL VCHAR(6,8,128,7)
70 CALL VCHAR(6,16,128,7)
80 CALL HCHAR(13,8,128,9)
90 IF A=1 THEN 110
100 CALL HCHAR(7,10,128)
110 IF A<4 THEN 130
120 CALL HCHAR(7,14,128)
130 IF A<>6 THEN 160
140 CALL HCHAR(9,10,128)
150 CALL HCHAR(9,14,128)
160 IF A/2=INT(A/2) THEN 180
170 CALL HCHAR(9,12,128)
180 IF A<4 THEN 200
190 CALL HCHAR(11,10,128)
200 IF A=1 THEN 220
210 CALL HCHAR(11,14,128)
220 INPUT "AUTRE TIRAGE (O O
U N) ":R$
230 IF R$="O" THEN 10

```

Visions

- REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : dessin.
Difficulté du jeu : facile.
- LE JEU** : Le TI-99 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 (30-120) se charge de réaliser les dessins.
La ligne 40 permet de transformer en nombre chaque lettre du mot entré au début (ligne 20) Ce nombre sera ensuite traduit en base 2 par une boucle J (lignes 60 à 110) ; selon le résultat de cette transformation, un point sera imprimé, ou non (ligne 80).
- 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 TI-99 dès que l'on aura frappé ENTER.
Pour recommencer, il suffira de frapper une touche quelconque, et CLEAR.
Je vous laisse savourer le dessin qu'a réalisé TI-99 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 le TI-99 deviendra de plus en plus savant. On peut aussi réaliser un jeu à deux joueurs, dans lequel TI-99 vous demande de trouver le mot qui correspond à un certain dessin, etc.

```
10 CALL CLEAR
20 INPUT "QUEL DESSIN : ":R$
30 FOR I=1 TO LEN(R$)
40 N=9*(ASC(SEG$(R$,I,1))-27
)-340
50 J=1
60 S=(N/2)-INT(N/2)
70 N=INT(N/2)
80 IF S=0 THEN 100
90 CALL HCHAR(I+5,J+12,128)
100 J=J+1
110 IF J=9 THEN 60
120 NEXT I
130 CALL KEY(S,K,L)
140 IF L THEN 10 ELSE 130
```

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 les 4 et les 2), et 1 plus petit (c'est le 1).
- LE PROGRAMME** : La ligne 40 interdit de demander des nombres de plus de 8 chiffres.
Le nombre à deviner est placé, en ligne 60, dans une variable alphanumérique X\$.
Les comparaisons avec le nombre essayé sont faites à la ligne 90 (qui teste si l'on a trouvé juste), et dans la boucle I (120-150) où l'on fait la comparaison "lettre à lettre".
L'affichage du résultat est réalisé grâce à la ligne 160.
Le compteur du nombre d'essais est en ligne 170.
- POUR JOUER** : Le lancement du programme par RUN provoque l'affichage NBRE DE CHIFFRES.
On entre un nombre inférieur à 8 et on appuie sur la touche ENTER.
Le numéro de l'essai s'affiche, ainsi que la question NBRE PRESUME ; on répond en entrant un nombre, suivi de ENTER.
Le résultat est alors affiché sous la forme ...CHIFFRES PLUS GRANDS, et ...CHIFFRES PLUS PETITS.
S'il n'est pas exact, on recommence (attention à entrer uniquement des nombres ayant autant de chiffres que le nombre à deviner).
Si l'on a trouvé, le TI-99 affichera TROUVE...
EN...ESSAIS. Puis AUTRE PARTIE (O ou N) ? ;
on fera O suivi de ENTER, pour commencer une autre 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 TI-99, qui tiendra alors le rôle de décodeur.

```

10 CALL CLEAR
20 RANDOMIZE
30 INPUT "NOMBRE DE CHIFFRES
   ":C
40 IF C>8 THEN 10
50 E=1
60 X$=STR$(INT(9*RND^C+10^(C
-E)))
70 PRINT : : "ESSAI NO";E: :
   : :
80 INPUT "NBRE PRESUME ":P$
90 IF P$=X$ THEN 190
100 A=0
110 B=A
120 FOR I=1 TO C
130 A=A-(SEG$(P$,I,1)>SEG$(X
$,I,1))
140 B=B-(SEG$(P$,I,1)<SEG$(X
$,I,1))
150 NEXT I
160 PRINT : : : : A;"CHIFFR
ES PLUS GRANDS": : B;"CHIFFRE
S PLUS PETITS"
170 E=E+1
180 GOTO 70
190 CALL CLEAR
200 PRINT "TROUVE ";X$;" EN"
   ;E;"ESSAIS": : :
210 INPUT "AUTRE PARTIE (O O
U N) ":R$
220 IF R$="O" THEN 10

```


L'électricien fou

- REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez difficile.
- LE JEU** : Un immeuble comportant 10 fenêtres, numérotées de 0 à 9, apparaît sur l'écran. Le joueur devra à chaque essai manoeuvrer l'un des interrupteurs, en indiquant son numéro (de 0 à 9).
En principe, la fenêtre correspondante devrait s'éteindre, mais il est aussi possible qu'une autre s'allume (ou s'éteigne). Il s'agit, bien sûr, d'essayer d'obtenir l'extinction des 10 fenêtres, avec le moins d'essais possibles.
- LE PROGRAMME** : Une table A, contiendra l'état (allumé ou éteint) des 10 fenêtres.
La boucle I (60-120) et l'autre boucle I (180-240) dessinent l'immeuble dans son état initial.
La ligne 250 regarde si tout est éteint, auquel cas l'on ira en ligne 460 afficher le nombre d'essais.
La ligne 280 attend l'indication de l'interrupteur manoeuvré au moyen d'un CALL KEY.
Les lignes 330 à 420 définissent le résultat de la manoeuvre.
- POUR JOUER** : RUN provoque l'affichage de l'immeuble, avec toutes les fenêtres allumées.
A la question INTER, on répond en indiquant le numéro de l'interrupteur à manoeuvrer, suivi de ENTER.
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 lors l'affichage du nombre d'essais réalisés.
- EXTENSIONS POSSIBLES** : On pourra envisager un graphisme très amélioré, avec si possible l'utilisation de la couleur. On pourra aussi définir un jeu à deux joueurs, augmenter le nombre de fenêtres, modifier la formule définissant l'allumage ou l'extinction...

```

10 CALL CLEAR
20 RANDOMIZE
30 CALL CHAR(128,"FFFFFFFFF
FFFFF")
40 S=1
50 M=0
60 FOR I=S TO 23
70 CALL HCHAR(6,I+5,128)
80 CALL HCHAR(6+S,I+5,128)
90 CALL HCHAR(6+S+S,I+5,128)
100 CALL HCHAR(9,I+5,128)
110 CALL HCHAR(10,I+5,128)
120 NEXT I
130 T$="INTERRUPTEUR NUMERO
:"
140 FOR I=1 TO 21
150 CALL HCHAR(18,4+I,ASC(SE
G$(T$,I,1)))
160 NEXT I
170 T=0
180 FOR I=S TO 10
190 T=T+A(I)
200 CALL HCHAR(S+S+6,2*I+S+5
,128)

210 IF A(I) THEN 230
220 CALL HCHAR(6+2*S,2*I+S+5
,32)
230 CALL HCHAR(10,2*I+S+5,47
+I)
240 NEXT I
250 IF T=10 THEN 460
260 CALL HCHAR(18,27,32)
270 CALL SOUND(100,-1,0)
280 CALL KEY(5, KK, SS)
290 IF SS=0 THEN 280
300 CALL SOUND(40,800,0)
310 CALL HCHAR(18,27, KK)
320 N=KK-48
330 N=N+S
340 M=M+S
350 IF A(N)=S THEN 440
360 A(N)=S
370 R=TAN(RND+N/RND-N)-SIN(R
ND/N)+336*SIN(8*N)
380 N=INT(10*(R-INT(R)))
390 IF N THEN 410
400 N=INT(S+10*RND)
410 IF A(N)=S THEN 440
420 A(N)=S
430 GOTO 170
440 A(N)=0
450 GOTO 370
460 PRINT "TROUVE EN";M;"ESS
AIS"

```

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 vont apparaître en début de partie, et chaque joueur peut prendre autant de carrés qu'il le désire, mais ceci dans un seul tas à la fois. Celui qui prend le dernier carré est considéré comme perdant.
- LE PROGRAMME** : La boucle I (30-70) dessine 4 rangées de 20 carrés.
La variable F, dimensionnée en ligne 20, servira à enregistrer le nombre de carrés pris dans chaque rangée.
La variable T contiendra le nombre total de carrés enlevés.
La ligne 50 imprime les numéros de ligne.
La ligne 120 va contrôler qu'il reste suffisamment de carrés sur la ligne indiquée.
La boucle I (160-200) redessine le terrain de jeu après modification.
La ligne 240 teste la fin de partie.
- POUR JOUER** : Lancer le programme par RUN ; le terrain de jeu s'affiche, puis les inscriptions JR 1 et LGN., CASES ? On répond en entrant un numéro de ligne (entre 1 et 4), et le nombre de carrés à enlever suivi de ENTER.
Les carrés sont alors supprimés, et c'est au tour du deuxième joueur.
La partie se termine, lorsque le dernier carré a été enlevé, par l'affichage du joueur gagnant.
- EXTENSIONS POSSIBLES** : Outre une amélioration de la qualité du graphisme et des contrôles, de nombreuses variantes sont envisageables, en modifiant si besoin est les règles de départ : on peut autoriser l'enlèvement de seulement un nombre impair de carrés, ou alterner pair et impair, ou tirer au hasard le nombre à enlever, ou encore jouer à plusieurs...

```

10 CALL CLEAR
20 DIM F(4)
30 FOR I=1 TO 4
40 F(I)=0
50 CALL HCHAR(9+2*I,6,48+I)
60 CALL HCHAR(9+2*I,9,30,20)
70 NEXT I
80 T=0
90 B=1
100 PRINT "JOUER";B: :
110 INPUT "LIGNE,CASES ? : "
:L,C
120 IF (C+F(L)>20)+(L>4) THEN
110
130 F(L)=F(L)+C
140 T=T+C
150 CALL CLEAR
160 FOR I=1 TO 4
170 CALL HCHAR(9+2*I,6,48+I)
180 CALL HCHAR(9+2*I,9+F(I),
30,20-F(I))
190 CALL HCHAR(9+2*I,9,42,F(
I))
200 NEXT I
210 B=B+A
220 IF B<>3 THEN 240
230 B=1
240 IF T<80 THEN 100
250 CALL CLEAR
260 PRINT "LE JOUEUR";B;"GAG
NE"

```

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** : Le codage du dessin est, dans le programme retenu, figé dans une variable A\$, définie en ligne 50 à l'aide de caractères redéfinis. On pourra éventuellement essayer d'autres dessins en entrant directement leurs codes : il suffit de modifier la ligne concernée par INPUT A\$.
- La boucle N (60-140) réalise le décodage de A\$.
- La ligne 100 détermine les positions de la tête d'écriture.
- La ligne 120 détecte le caractère 0, qui sert à indiquer le retour à la ligne.
- POUR JOUER** : Le dessin devra auparavant être défini par une variable A\$, en utilisant les caractères redéfinis, et en donnant la position d'impression de chaque caractère, par rapport au début de la ligne, avec un chiffre de 1 à 9, ou une lettre (A : 10 ; B : 11 ; C : 12 ; etc.).
- Le passage à la ligne suivante sera indiquée par le caractère 0.
- EXTENSIONS POSSIBLES** : La fonction CHAR 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.
- Remarque** : Pour entrer directement dans une chaîne un caractère redéfini au moyen de CHAR, on peut se servir de la touche CTRL. Par exemple : CTRL A correspond au caractère de code ASCII 129. Se référer au manuel d'utilisation pour de plus amples détails.

```

10 CALL CLEAR
20 CALL CHAR(129,"FFFFFFFFF
  FFFFFF")
30 CALL CHAR(130,"00000000FF
  FFFFFF")
40 CALL CHAR(131,"0000001818
  ")
50 A$="AOAKROAKROAKRO309BM
  DOANO"
60 FOR N=1 TO LEN(A$)
70 S=ASC(SEG$(A$,N,1))
80 IF S<128 THEN 100
90 PRINT SEG$(A$,N,1);
100 IF ((S<49)+(S>57))*((S<6
  5)+(S>90)) THEN 120
110 PRINT TAB(S-48+7*(S>=65)
  );
120 IF S<>48 THEN 140
130 PRINT ""
140 NEXT N

```

Dé truqué

- REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté de programmation : facile.
- LE JEU** : On veut essayer de fabriquer un dé truqué selon sa propre convenance. La méthode proposée ici consiste à définir au départ une probabilité de réalisation pour chaque type de tirage (soit la probabilité d'obtenir un 1, un 2, un 3, etc., que chacune de ces probabilités soit comprise entre 0 et 1, et que la somme de toutes les probabilités soit un nombre très voisin de 1).
- LE PROGRAMME** : Il utilise le générateur aléatoire du TI-99. supposé de qualité convenable (on pourra aisément utiliser tout autre générateur, si besoin était).
La boucle I (40-100) s'occupe de l'entrée de chacune des probabilités, avec vérification à chaque entrée et sur le total (ligne 90) ; chaque probabilité est conservée dans une variable indicée E (I) : ligne 60 du programme.
Le lancer de dé truqué est effectué à l'aide d'une autre boucle I (200-220), la ligne permettant le trucage se trouvant en 210.
Le résultat est affiché grâce à la ligne 230.
Un nouveau tirage pourra être effectué sur simple pression d'une touche (lignes 240 et 250).
- POUR JOUER** : Après le lancement de l'exécution par RUN, on obtient l'affichage de la question PROBABILITE DU DE 1, à laquelle on devra répondre par un nombre compris entre 0 et 1. Il conviendra de ne pas oublier que le total de toutes les probabilités entrées ne devra en aucun cas dépasser 1.1 (tolérance de 0.1). Des messages d'erreurs pourront être affichés et le programme sera éventuellement relancé.
Lorsque toutes les probabilités auront été définies, le premier tirage sera automatiquement effectué, et il suffira de presser une touche quelconque (sauf CLEAR) pour effectuer les tirages suivants. On arrête le programme en tapant CLEAR.
- EXTENSIONS POSSIBLES** : Il sera bien évidemment très 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...

```

10 CALL CLEAR
20 P=0
30 DIM E(6)
40 FOR I=1 TO 6
50 PRINT "PROBABILITE DU DE"
;I
60 INPUT E(I)
70 PRINT
80 P=P+E(I)
90 IF (E(I)<0)+(E(I)>1)+(P>1
.1) THEN 270
100 NEXT I
110 PRINT : :
120 FOR I=1 TO 6
130 PRINT "DE";I,E(I)
140 NEXT I
150 IF P<.9 THEN 10
160 FOR I=1 TO 300
170 NEXT I
180 RANDOMIZE
190 N=INT(RND*6+1)
200 FOR I=1 TO 6
210 IF (N=I)*(RND>E(I)) THEN
190
220 NEXT I
230 PRINT : "TIRAGE :";N
240 CALL KEY(5,K,S)
250 IF S=0 THEN 240
260 GOTO 190
270 PRINT : : "IMPOSSIBLE : 0<
= P <=1"
280 FOR L=1 TO 700
290 NEXT L
300 P=P-E(I)
310 CALL CLEAR
320 GOTO 50

```


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 rouges, 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 contraindrait 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 (90-120) initialise une table A (I), qui contient les positions des moutons. La boucle I (140-200) dessine le terrain de jeu et les moutons. Les lignes 210 et 220 s'occupent du texte et des entrées. La ligne 230 contrôle la validité du coup. Le changement de joueur est réalisé en ligne 260.
- POUR JOUER** : On commence l'exécution du programme par RUN. On obtient alors l'affichage JRI, qui indique le numéro du joueur, suivi de la question DEBUT, FIN ? On répond en indiquant le numéro du pion à déplacer, puis le numéro de la case d'arrivée, suivi de ENTER. Le déplacement est alors réalisé ; puis le deuxième joueur procède de la même façon. Le gagnant est en principe celui qui réussit à placer ses trois moutons côte à côte sur les emplacements initialement occupés par l'adversaire.
- EXTENSIONS POSSIBLES** : On commencera par rajouter un contrôle du joueur gagnant. On peut également améliorer la présentation en dessinant de vrais moutons, rendre le jeu plus attractif en les numérotant, et en imposant un certain ordre à l'arrivée. Enfin, on pourra définir une stratégie pour l'ordinateur, en figeant les règles.

```

10 CALL CLEAR
20 CALL SCREEN(16)
30 CALL CHAR(128,"007E7E7E7E
7E7E7E")
40 CALL COLOR(13,7,1)
50 B=1
60 DIM A(8)
70 A(1)=0
80 A(8)=0
90 FOR I=B TO 3
100 A(B+I)=B
110 A(4+I)=-B
120 NEXT I
130 H=B
140 FOR I=B TO 8
150 CALL HCHAR(17,3*I+B+3,4B
+I)
160 IF A(I)<>B THEN 180
170 CALL HCHAR(15,3*I+B+3,30
)
180 IF A(I)<>-B THEN 200
190 CALL HCHAR(15,3*I+B+3,12
B)
200 NEXT I
210 PRINT "JOUER"; (H=B)+2
220 INPUT "DEBUT,FIN ? ":D,F
230 IF (A(D)<>H)+(A(F)<>1-B)
THEN 210
240 A(D)=1-B
250 A(F)=B*H
260 H=-H
270 CALL CLEAR
280 GOTO 140

```

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 TI-99 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 (80-120) choisit dans l'alphabet entre 7 et 12 lettres, en fonction de la variable aléatoire L, tirée en ligne 50.
Pour éviter d'avoir trop de consonnes, on utilise en ligne 110 une petite astuce, qui consiste à refaire au hasard un tirage dans la table A\$, contenant des voyelles (on peut en avoir plus ou moins en utilisant d'autres RND).
La boucle 130-190 demande à chacun des joueurs la longueur de son mot.
Les lignes 200-210 comparent les résultats et totalisent le score de chaque joueur, affiché par la ligne 220.

POUR JOUER

: On lance l'exécution du jeu avec RUN.
Les lettres sont alors affichées, puis la question FR.1LONG, 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 une touche quelconque (sauf la touche CLEAR qui permettra de finir).

EXTENSIONS POSSIBLES

: On pourra bien entendu songer à introduire une temporisation avant de demander leurs réponses aux joueurs.
On pourra aussi entrer non point la longueur du mot, mais le mot lui-même, l'adversaire devant alors répondre à la question CORRECT OU NON ?
On peut aussi étendre le jeu à plus de deux joueurs. Et pourquoi ne pas essayer de définir un jeu contre le TI-99, auquel on aurait auparavant entré une liste de mots assez longue, dont il afficherait les lettres dans le désordre, avec éventuellement quelques lettres de complément choisies au hasard.

```

10 S=0
20 J=S
30 RANDOMIZE
40 CALL CLEAR
50 L=INT(6*RND+7)
60 A$="AEIOUE"
70 PRINT "LETTRES :": : : :
: : : : :
80 FOR I=1 TO L
90 CALL HCHAR(15,7+2*I,INT(2
6*RND+65))
100 IF RND>=.4 THEN 120
110 CALL HCHAR(15,7+2*I,ASC(
SEG$(A$,INT(6*RND+1),1)))
120 NEXT I
130 I=1
140 T$="JOUER " & STR$(I) & "
LONGUEUR ? "
150 INPUT T$:L
160 IF I=1 THEN 180
170 P=L
180 I=I+1
190 IF I<3 THEN 140
200 S=S-(P*(P>L))
210 J=J-(L*(L>P))
220 PRINT "JOUER 1: ";S,"JOU
EUR 2: ";J
230 CALL KEY(S,K,T)
240 IF T THEN 40 ELSE 230

```

Espion

- REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : facile...
- LE JEU** : Le programme peut servir simultanément de codeur et de décodeur de messages 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 surveiller à utiliser toujours le même nombre (noyau) pour commencer le codage ou le décodage de deux messages identiques.
- LE PROGRAMME** : Le noyau initial (qui sera un entier quelconque) est entré en ligne 20 ; le message est entré à la ligne 40.
La ligne 50 permet de choisir entre les options de codage ou de décodage.
La boucle I (60-150) est utilisée soit pour effectuer le codage, soit pour effectuer le décodage.
La ligne 80 appelle le générateur aléatoire.
La ligne 90 réalise la transposition des lettres.
Les lignes 100 et 120 vérifient que l'on reste bien dans les limites autorisées.
Les résultats seront imprimés par la ligne 160.
Les lignes 180 à 210 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 ENTER.
Choisir l'option C pour CODER, et l'option D pour DECODER, et taper ENTER.
Le message, codé ou décodé selon le cas, sera alors affiché.
- EXTENSIONS POSSIBLES** : On pourra étendre le codage à d'autres signes que les lettres de l'alphabet, utiliser éventuellement plusieurs générateurs, introduire dans le message un codage du ou des nombres ayant servi de générateurs, etc.

```

10 CALL CLEAR
20 INPUT "CODE ? ":N
30 A=1/N
40 INPUT "MESSAGE :":A$
50 INPUT "(C)ODER OU (D)ECOD
ER :":R$
60 FOR I=1 TO LEN(A$)
70 T=ASC(SEG$(A$,I,1))
80 GOSUB 180
90 T=T+A*((R$="D")-(R$="C"))
100 IF T<=90 THEN 120
110 T=T-26
120 IF T>=65 THEN 140
130 T=T+26
140 A$=SEG$(A$,1,I-1)&CHR$(T
)&SEG$(A$,I+1,LEN(A$)-I)
150 NEXT I
160 PRINT "RESULTAT :";A$
170 END
180 A=(A+3.14159)*8
190 A=A-INT(A)
200 A=INT(26*A+1)
210 RETURN

```

Accrochez les wagons

- REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : facile.
- LE JEU** : Il faut essayer d'accrocher, le plus rapidement possible, la locomotive à son wagon, qui apparaissent tous deux en début de partie, en deux endroits plus ou moins éloignés de l'écran. On devra pour cela déplacer la locomotive en indiquant la suite des mouvements à effectuer (sous forme codée : H pour une case vers le haut ; B pour une case vers le bas ; D pour une case vers la droite, et G pour une case vers la gauche). On tapera, par exemple : BBGDHHH.
- LE PROGRAMME** : Les lignes 10 à 40 définissent les caractères qui seront utilisés pour dessiner la locomotive et le wagon.
La position initiale de la locomotive est définie par les lignes 60 et 70, celle du wagon par les lignes 80 et 90.
La ligne 100 évite qu'ils soient initialement trop proches l'un de l'autre.
Les lignes 120 à 170 réalisent l'affichage du wagon, les lignes 180 à 250 celui de la locomotive.
Le test de fin est en ligne 260.
Les mouvements à effectuer sont entrés dans une variable alphanumérique R\$, en ligne 270, et sont décodés dans une boucle I (310-340), pour être exécutés en lignes 350 et 360.
- 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 ENTER.
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 CALL CHAR(131,"000000000F
0F0F0F")
20 CALL CHAR(130,"FFFFFFFFFF
FFFFFF")
30 CALL CHAR(129,"F0F0F0F0FF
FFFFFF")
40 CALL CHAR(132,"00000000FF
FFFFFF")
50 RANDOMIZE
60 LX=INT(21*RND+2)
70 LY=INT(25*RND+3)
80 WX=INT(21*RND+2)
90 WY=INT(25*RND+3)
100 IF ABS(LX-WX)<4 THEN 60
110 CALL CLEAR
120 CALL HCHAR(WX,WY,129)
130 CALL HCHAR(WX,WY+1,132)
140 CALL HCHAR(WX,WY+2,129)
150 CALL HCHAR(WX+1,WY,48)
160 CALL HCHAR(WX+1,WY+1,32)
170 CALL HCHAR(WX+1,WY+2,48)
180 CALL HCHAR(LX-1,LY,131)
190 CALL HCHAR(LX-1,LY+1,132
,2)
200 CALL HCHAR(LX-1,LY+3,131
)
210 CALL HCHAR(LX,LY,132)
220 CALL HCHAR(LX,LY+1,130,3
)
230 CALL HCHAR(LX+1,LY+1,48)
240 CALL HCHAR(LX+1,LY+2,32)
250 CALL HCHAR(LX+1,LY+3,48)
260 IF (LX=WX)*(WY=LY-3) THEN
380
270 INPUT "CHEMIN A SUIVRE :
":R$
280 X=0
290 Y=X
300 N=N+1
310 FOR I=1 TO LEN(R$)
320 X=X+(SEG$(R$,I,1)="B")*(
LX<20)-(SEG$(R$,I,1)="H")*(L
X>2)
330 Y=Y+(SEG$(R$,I,1)="D")*(
LY<28)-(SEG$(R$,I,1)="G")*(L
Y>2)
340 NEXT I
350 LX=LX+X
360 LY=LY+Y
370 GOTO 110
380 PRINT "BRAVO , TROUVE EN
";N;"COUPS"

```


Tirage d'un dé truqué

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : moyenne.
- LE JEU** : Il reprend une idée d'un programme précédent (DE TRUQUE) dans une application statistique : on fabrique un dé volontairement truqué, que l'on va lancer un certain nombre de fois, et l'on dresse un tableau des résultats.
- LE PROGRAMME** : Les probabilités désirées pour chaque tirage sont introduites par une boucle I (40-90), avec toutes les vérifications de validité nécessaire. Le nombre de tirages demandés est introduit en ligne 120.
La boucle J (150-260) comptabilise les tirages.
La boucle I (170-190) assure la conformité de chaque tirage avec les probabilités introduites en début de jeu.
Les résultats sont comptabilisés en ligne 200 et affichés grâce aux lignes 220 à 250.
- POUR JOUER** : Commencer l'exécution en frappant RUN, puis, à la question PROBABILITE DU DE 1, introduire un nombre compris entre 0 et 1.
Faire de même pour les dés 2, 3, etc., sans oublier que le total des probabilités introduites devra être très proche de 1.
Ensuite, à l'affichage de la question NOMBRE DE TIRAGES on introduira un entier suffisamment grand, suivi de ENTER. L'affichage des résultats mettra d'autant plus de temps à apparaître que l'entier introduit sera grand. Les résultats sont affichés sous les numéros des tirages. Ainsi 32.3 apparaissant sous le 1 signifie que le tirage de 1, auquel on avait affecté une probabilité de 0.3 a été réalisé 32 fois.
- EXTENSIONS POSSIBLES** : On peut envisager d'améliorer la représentation des résultats en utilisant par exemple un histogramme, ou tout autre méthode graphiquement plus parlante.

```

10 CALL CLEAR
20 P=0
30 DIM E(6)
40 FOR I=1 TO 6
50 P$="PROBABILITE DU DE "&S
TR$(I)&" : "
60 INPUT P$:E(I)
70 P=P+E(I)
80 IF (E(I)<0)+(E(I)>1)+(P>1
.1) THEN 280
90 NEXT I
100 IF P<.9 THEN 10
110 PRINT : : :
120 INPUT "NOMBRE DE TIRAGES
:" : P
130 RANDOMIZE
140 PRINT : : : : : : : :
150 FOR J=1 TO P
160 N=INT(6*RND+1)
170 FOR I=1 TO 6
180 IF (N=I)*(RND>(E(I)-INT(
E(I)))) THEN 160
190 NEXT I
200 E(N)=E(N)+1
210 E$=STR$(E(N))

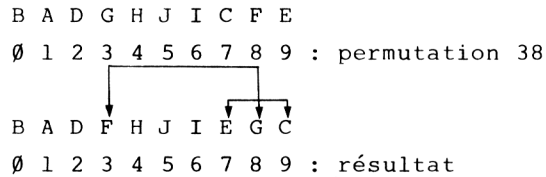
220 FOR H=1 TO LEN(E$)
230 CALL HCHAR(22,5*N+H-2,AS
C(SEG$(E$,H,1)))
240 NEXT H
250 CALL HCHAR(20,5*N-1,48+N
)
260 NEXT J
270 END
280 PRINT : : "IMPOSSIBLE : 0
<= P <= 1"
290 P=P-E(I)
300 FOR II=1 TO 250
310 NEXT II
320 GOTO 60

```

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'elles les deux lettres qui entourent celles-ci, suivant le mécanisme expliqué ci-dessous :



LE PROGRAMME : Les boucles emboîtées I (40-100) et J (60-80) définissent une variable alphanumérique A\$, contenant les 10 premières lettres de l'alphabet dans le désordre.
La ligne 120 affiche A\$, et les numéros de lettres à utiliser pour la permutation.
On entre la permutation en ligne 130.
La ligne 150 effectue la première inversion.
La ligne 170 effectue la deuxième.
La ligne 180 comptabilise les résultats.
La ligne 190 contrôle les résultats.

POUR JOUER : Lancer le programme par RUN.
Le mot initial est affiché, chaque lettre étant repérée par un numéro de 0 à 9.
A la question : INVERSION (I, J), on répond en entrant deux nombres x, y ; x est le numéro de la première lettre, et y le numéro de la seconde, sachant que $x < y$. Les deux inversions sont alors successivement effectuées, et l'on recommence.
Le nombre de coups sera affiché en fin de partie

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

```

10 CALL CLEAR
20 A$=""
30 RANDOMIZE
40 FOR I=1 TO 10
50 R$=CHR$(INT(10*RNDR+65))
60 FOR J=1 TO I-1
70 IF R$=SEG$(A$,J,1) THEN 50
80 NEXT J
90 A$=A$&R$
100 NEXT I
110 S=1

120 PRINT : : : A$ : "0123456
789" : "INVERSION (I,J) ; ES
SAI";S
130 INPUT I,J
140 IF J<I THEN 130
150 A$=SEG$(A$,1,I)&SEG$(A$,
J+1,1)&SEG$(A$,I+2,J-I-1)&SE
G$(A$,I+1,1)&SEG$(A$,J+2
,9-J)
160 IF J=9 THEN 180
170 A$=SEG$(A$,1,J-1)&SEG$(A
$,J+2,1)&SEG$(A$,J+1,1)&SEG$
(A$,J,1)&SEG$(A$,J+3,8-J
)
180 S=S+1
190 IF A$<>"ABCDEFGHIJ" THEN
120
200 PRINT : : : "TROUVE EN";"
COUPS"

```

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 TI-99 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 (80-200) affiche les chevaux dans leur position de départ. Une deuxième boucle I (110-220) les fera avancer ; la position de chaque cheval sera repérée par un chiffre ou une lettre dans une variable alphanumérique A\$. La ligne 130 fera avancer les chevaux. La ligne 140 gère les arrivées. Les lignes 160 et 170 réalisent les impressions. Le sous-programme 290-330 fait avancer le compteur des arrivées, et imprime les résultats.
- POUR JOUER** : On lance le programme par RUN. A la question NBRE DE CHEVAUX ?, on répond par un entier de 1 à 10, suivi de ENTER. 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.

```

10 CALL CLEAR
20 RANDOMIZE
30 A$=""
40 INPUT "NOMBRE DE CHEVAUX
:";N
50 P=0
60 T=1
70 A=P
80 FOR I=T TO N
90 A$=A$&"1"
100 GOTO 130
110 FOR I=1 TO N
120 IF SEG$(A$,I,1)="K" THEN
220
130 A$=SEG$(A$,1,I-1)&CHR$(A
SC(SEG$(A$,I,1))+INT(4*RND+1
))&SEG$(A$,I+1,LEN(A$)-I
)
140 IF SEG$(A$,I,1)<"K" THEN
160
150 GOSUB 290
160 CALL HCHAR(I+I,ASC(SEG$(
A$,I,1))-49,32,4)
170 CALL HCHAR(I+I,ASC(SEG$(
A$,I,1))-45,96+I)
180 FOR IP=1 TO 200
190 NEXT IP
200 NEXT I
210 GOTO 230
220 NEXT I
230 P=P+T
240 T=0
250 FOR IP=1 TO 400
260 NEXT IP
270 IF A<N THEN 110
280 STOP
290 A$=SEG$(A$,1,I-1)&"K"&SE
G$(A$,I+1,LEN(A$)-I)
300 CALL HCHAR(I+1,3,48+P)
310 T=1
320 A=A+T
330 RETURN

```

Le jeu de Marienbad

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : logique (2 joueurs).
Difficulté du jeu : moyenne.
- LE JEU** : Un jeu rendu très célèbre grâce à un film qui le popularisa, présenté toutefois ici dans une version plus conforme à ses origines à savoir le jeu de NIM.
En effet, il est possible, en début de partie de choisir le nombre 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 60 dimensionne une variable A (I) qui contiendra le nombre d'allumettes de chaque tas.
La boucle I (70-110) permet d'entrer ce nombre.
Les boucles I (140-160, 180-200) impriment les tas.
La ligne 220 assure le changement de joueur.
La ligne 250 contrôle si le coup est autorisé.
La ligne 320 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 ENTER.
On entrera ensuite le nombre d'allumettes dans chaque tas (faire à chaque entrée N suivi de ENTER). On obtiendra alors l'affichage des différents tas, avec le numéro du joueur concerné (1 ou 2), ainsi que le message N° DU TAS ; on répond en tapant le numéro du tas dans lequel on désire faire un retrait suivi de ENTER.
On obtient ensuite la question NBRE. A OTER, et l'on répond en indiquant le nombre d'allumettes que l'on désire enlever du tas concerné.
Puis ce sera au tour du deuxième joueur, et l'on jouera alternativement jusqu'à ce qu'il n'y ait plus d'allumettes disponibles.
En fin de partie, on obtiendra l'affichage du numéro du joueur gagnant.
- EXTENSIONS POSSIBLES** : On peut désigner chaque joueur par son nom, faire un graphisme plus élaboré, écrire un programme permettant au TI-99 de servir de partenaire, etc.

```

10 CALL CLEAR
20 INPUT "NOMBRE DE TAS ":N
30 PRINT : :
40 S=0
50 IF N>=8 THEN 10
60 DIM A(7)
70 FOR I=1 TO N
80 T$="TAS "&STR$(I)&","NOMBR
E D'ALLUMETTES "
90 INPUT T$:A(I)
100 S=S+A(I)
110 NEXT I
120 H=1
130 CALL CLEAR
140 FOR I=1 TO N
150 PRINT TAB(4*I+1);A(I);
160 NEXT I
170 PRINT : : "TAS";
180 FOR I=1 TO N
190 PRINT TAB(1+4*I);I;
200 NEXT I
210 PRINT : : :TAB(4);"JOUEU
R";H: :
220 IF H=1 THEN 240
230 H=0
240 INPUT "PRISE DANS LE TAS
":T.
250 IF (T>N)+(A(T)=0)THEN 24
0
260 PRINT : :
270 INPUT "NOMBRE A OTER :":
0
280 IF (0>A(T))+(0=0)THEN 27
0
290 A(T)=A(T)-0
300 S=S-0
310 H=H+1
320 IF S=0 THEN 340
330 GOTO 130
340 PRINT : : "LE JOUEUR";H;"
A GAGNE"

```


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 TI-99 choisit un nombre de n chiffres (selon le niveau de difficulté indiqué au début). Ce nombre (le même au début pour les deux joueurs) sera régulièrement décrémenté à chaque tour, d'une quantité également choisie au hasard par TI-99 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 10, pour contenir les résultats successifs des deux joueurs.
La ligne 40 permettra d'entrer le niveau de difficulté choisi.
Le nombre initial est confectionné par la ligne 80. La quantité qui interviendra dans les décréments est obtenue en ligne 100.
La ligne 130 teste si le résultat n'est pas trop petit.
La ligne 150 affiche le numéro du joueur et le nombre. Le jeu du joueur est entré en ligne 160 (deux chiffres).
La ligne 170 incrémente la colonne choisie et la ligne 180 décrément le nombre de la quantité choisie au départ.
Le résultat sera affiché un court instant (lignes 190-210). La boucle I (120-135) permet de vérifier si tous les chiffres sont identiques.
L'impression et la fin de partie sont effectuées en ligne 260. Les lignes 290-300 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 x, y suivi de ENTER ; 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 CLEAR.

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 DIM M(2)
20 H=1
30 CALL CLEAR
40 INPUT "NIVEAU (2-7) :":N
50 U=10
60 P=0
70 RANDOMIZE
80 M(1)=INT((U^N-U^(N-1))*RN
D+U^(N-1))
90 M(2)=M(1)
100 A=INT(2*U^(N-1)*RND+U^(N
-1))
110 CALL CLEAR
120 P=P-(H=1)
130 IF M(H)>=U^(N-1) THEN 150
140 M(H)=U^N-ABS(M(H))
150 PRINT "Joueur";H,"ESSAI"
;P: : "NOMBRE";TAB(15-N);M(
H):TAB(9);"7654321": :
160 INPUT "VOTRE JEU (N,C) ?
":R1,R2
170 M(H)=M(H)+R1*U^(R2-1)
180 M(H)=M(H)-A
190 PRINT : : "RESULTAT";M(
H)
200 FOR I=1 TO 400
210 NEXT I
220 R$=STR$(M(H))
230 FOR I=1 TO LEN(R$)
240 IF SEG$(R$,I,1)<>SEG$(R$
,I+1,1) THEN 280
250 NEXT I
260 PRINT : : "BRAVO , JOUE
UR";H;": ";P;"ESSAIS"
270 END
280 H=H+1
290 IF H<3 THEN 110
300 H=1
310 GOTO 110

```

Mastermind

- REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : logique (un seul joueur).
Difficulté du jeu : difficile.
- LE JEU** : Devenu un grand classique des jeux sur micro-ordinateur, ce jeu développe le raisonnement déductif ; il faut réussir à trouver le plus vite possible un nombre de n chiffres (on pourra prendre n compris entre 1 et 5), en disposant à chaque essai de renseignements. En effet, le TI-99 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 même emplacement.
- LE PROGRAMME** : Le choix du nombre de chiffres du nombre à découvrir est effectué à la ligne 30 et à la ligne 40. Le nombre est fabriqué à la ligne 60 (il pourra y avoir des chiffres répétés).
L'affichage est préparé par la ligne 70.
La saisie de E\$ se fait au moyen de KEY (lignes 100 à 180) afin de ne pas décoder l'affichage. La ligne 140 teste si l'on a pressé sur ENTER.
La ligne 210 s'occupe de la fin de partie.
Une première boucle I (240-330) teste les chiffres qui sont à la bonne place.
Une deuxième boucle J (260-280) teste les chiffres qui sont présents et mal placés.
Les compteurs sont aux lignes 300 et 320.
La ligne 340 affiche le résultat, et la ligne 350 comptabilise les essais.
- POUR JOUER** : On tape RUN et l'on choisit la longueur du nombre désiré (de 1 à 6), suivie de ENTER. Après l'affichage N°. NOMBRE BON PLACE, on introduit son propre essai, suivi de ENTER. On obtient immédiatement la réponse (en cas de double chiffre, les résultats peuvent être un peu délicats à interpréter). On introduit ensuite le nouvel essai, et ainsi de suite.
Si l'écran est plain (message d'erreur 5), il faudra faire CONT pour effacer l'écran et poursuivre. Lorsque l'on a trouvé, on obtiendra l'affichage TROUVE EN...COUPS. On peut sortir du programme à tout moment en entrant une lettre, suivie de ENTER, à la place du nombre essayé.
- EXTENSIONS POSSIBLES** : On peut chercher à éviter la présence de chiffres doubles dans le nombre à trouver, on peut aussi prévoir un changement de page automatique, avec rappel des résultats pré-

cédents ; on pourra aussi réaliser des jeux analogues avec des mots, ou d'autres symboles (couleur, taille,...).

```

10 CALL CLEAR
20 RANDOMIZE
30 INPUT "NOMBRE DE CHIFFRES
   ":N
40 IF N>=6 THEN 10
50 E=1
60 X$=STR$(INT(RND*9^N+10^(N
-1)))
70 PRINT : : : "NO.   NOMBRE
   BON   PLACE": : :
80 C=0
90 E$=""
100 CALL HCHAR(24,8,62)
110 CALL SOUND(100,-1,0)
120 CALL KEY(S,K,S)
130 IF (S=0)+(S=-1) THEN 120
140 IF K=13 THEN 190
150 C=C+1
160 CALL HCHAR(24,8+C,K)
170 E$=E$&CHR$(K)
180 GOTO 120
190 X=0
200 Y=X
210 IF E$<>X$ THEN 240
220 PRINT : : : "TROUVE EN";E
   ;"COUPS"
230 STOP
240 FOR I=1 TO N
250 IF SEG$(X$,I,1)=SEG$(E$,
   I,1) THEN 320
260 FOR J=1 TO N
270 IF (J<>I)*(SEG$(X$,I,1)=
   SEG$(E$,J,1)) THEN 300
280 NEXT J
290 GOTO 330
300 Y=Y+1
310 GOTO 330
320 X=X+1
330 NEXT I
340 PRINT STR$(E);TAB(7);E$;
   TAB(17);X;TAB(25);Y
350 E=E+1
360 GOTO 80

```

A travers

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : moyenne.
- LE JEU** : Sur l'écran apparaît un carré de 9 cases, avec sur le bord vertical des lettres et sur le bord horizontal des chiffres.
L'un des joueurs manipule les chiffres, l'autre les lettres. 3 déplacements sont possibles pour les chiffres, et 3 autres pour les lettres.
Les lettres peuvent se déplacer vers le haut, à droite et à gauche, et les chiffres vers la droite, en haut ou en bas.
Le gagnant est le premier qui arrive à faire sortir tous ses symboles (les lettres sortent en haut, et les chiffres à droite).
Il n'y a pas de prise, et les sauts ne sont pas autorisés ; on ne peut pas non plus se placer sur une case déjà occupée.



- LE PROGRAMME** : Les positions des chiffres et des lettres sont enregistrées dans une table N (4,4) ; les lignes 100 à 110 définissent les valeurs de départ.
Les lignes 130 à 200 dessinent le terrain de jeu et la ligne 210 affiche le joueur qui a la main.
La ligne 220 contrôle le coup joué, de même que les boucles I (230-270) et J (240-260).
Le jeu proprement dit est géré par les lignes 290 à 360, et le changement de joueur est assuré par la ligne 370.
- POUR JOUER** : Le terrain de jeu sera affiché après la frappe de RUN et l'ordinateur attend le coup du joueur désigné par les pions qu'il possède. 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 ENTER. 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 TI-99.

```

10 K=1
20 J$(0)="LETTRES"
30 J$(1)="CHIFFRES"
40 F=2
50 S=4
60 DIM N(4,4)
70 FOR I=K TO S
80 N(I,K)=48+I
90 N(S,I)=63+I
100 NEXT I
110 N(S,K)=0
120 CALL CLEAR
130 FOR J=1 TO S
140 FOR I=1 TO S
150 IF N(I,J) THEN 180
160 CALL HCHAR(F*I+B,F*J+12,
46)
170 GOTO 190
180 CALL HCHAR(F*I+B,F*J+12,
N(I,J))
190 NEXT I
200 NEXT J
210 INPUT "JR. AYANT LES "&J
$(K)&" ";R$
220 IF K*(SEG$(R$,F,1)="O")+
(K=0)*(SEG$(R$,F,1)="S") THEN
210
230 FOR I=1 TO S
240 FOR J=1 TO S
250 IF N(I,J)=ASC(SEG$(R$,1,
1)) THEN 290
260 NEXT J
270 NEXT I
280 GOTO 210
290 A=I
300 B=J
310 I=I-(SEG$(R$,F,1)="S")+(
SEG$(R$,F,1)="N")
320 J=J-(SEG$(R$,F,1)="E")+(
SEG$(R$,F,1)="O")
330 IF (I=0)*(K=0)+(J>S)*K T
HEN 360
340 IF N(I,J) THEN 210
350 N(I,J)=ASC(SEG$(R$,1,1))
360 N(A,B)=0
370 K=ABS(K-1)
380 GOTO 120

```

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 20 vous permet de choisir la taille de l'échiquier.
Une variable D est surdimensionnée en ligne 30.
Une boucle d'exploration principale I (40-90) contient une sous-boucle J (60-80) qui réalise le contrôle de non alignement (ligne 70). Si, au cours de l'exploration, on rencontre un alignement, on revient en arrière, en utilisant les lignes 220 à 250, et on recommence.
La boucle F (110-190) assure l'impression de l'échiquier, et la boucle T (140-160), que l'on pourra éventuellement supprimer, assure l'impression du résultat sous forme d'un code numérique correspondant à la position des pièces dans chaque colonne.
La ligne 210 permet un arrêt après l'affichage.
- POUR JOUER** : On lance le programme par RUN, puis on tape la dimension de l'échiquier désiré, et on fait ENTER.
Il suffit d'attendre, avec patience (plus l'échiquier sera grand, plus l'attente sera longue).
Il faudra taper CLEAR en fin d'exécution.
- EXTENSIONS POSSIBLES** : On peut envisager de trouver des algorithmes pour tous les problèmes analogues (position de tours, déplacement d'un cavalier, etc.).

```

10 CALL CLEAR
20 INPUT S
30 DIM D(20)
40 FOR I=1 TO S
50 D(I)=1
60 FOR J=1 TO I-1
70 IF (D(I)=D(J))+(ABS(D(I)-
D(J))=I-J)THEN 220
80 NEXT J
90 NEXT I
100 CALL CLEAR
110 FOR F=1 TO S
120 CALL HCHAR(3,2*F+3,48+F)
130 CALL HCHAR(2*F+3,3,48+F)
140 FOR T=1 TO S
150 CALL HCHAR(2*F+3,2*T+3,4
6)
160 NEXT T
170 CALL HCHAR(2*F+3,2*D(F)+
3,30)
180 CALL HCHAR(24,3+F,48+D(F
))
190 NEXT F
200 CALL SOUND(70,-1,0)
210 GOTO 210
220 D(I)=D(I)+1
230 IF D(I)<=S THEN 60
240 I=I-1
250 IF I THEN 220

```


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 TI-99 (version entièrement automatique) soit par vous-même (version manuelle). Cette population va évoluer sous vos yeux, en respectant certaines règles, très élémentaires, de naissance, de vie, et de mort. L'évolution se poursuivra jusqu'à ce que l'on décide d'arrêter le programme.
- LE PROGRAMME** : Deux tables (B et C) sont utilisées pour enregistrer l'état de la population.
Les boucles F (70-210) et G (80-200) sont utilisées en début de jeu pour définir la population initiale.
Le calcul de la nouvelle population est ensuite réalisé à l'aide d'une quadruple boucle : 70-210 ; 80-200 ; 100-150 ; 110-140 .
La ligne 120 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 (230-320) et G (240-310).
La ligne 330 permet de recommencer.
- POUR JOUER** : Dans la version automatique, il suffit de lancer le programme par RUN et regarder.
Pour arrêter, il faudra taper CLEAR.
Dans la version manuelle, la configuration de départ sera entrée, après le lancement du programme par RUN.
L'évolution se déroulera ensuite de façon automatique.
- EXTENSIONS POSSIBLES** : Il sera facile d'augmenter les dimensions des tables B et C.
On pourra aussi introduire la couleur, améliorer la précision du dessin en redéfinissant des caractères.

```

10 CALL CLEAR
20 INPUT "DIMENSION (MAX 10)
:" : A
30 D=1
40 I=0
50 GOTO 210
60 FOR F=D TO A
70 FOR G=D TO A
80 S=0
90 FOR I=-D TO D
100 FOR J=-D TO D
110 IF (F+J=0)+(G+I=0)+(F+J=
A+D)+(G+I=A+D) THEN 130
120 S=S+B(F+J,D+I)
130 NEXT J
140 NEXT I
150 S=2*S-B(F,G)
160 C(F,G)=0
170 IF (S<5)+(S>7) THEN 190
180 C(F,G)=D
190 NEXT G
200 NEXT F
210 CALL CLEAR
220 FOR F=D TO A
230 FOR G=D TO A
240 IF I THEN 330
250 CALL HCHAR(2*F,2,62)
260 CALL HCHAR(1,2*G+2,86)
270 FOR P=1 TO 300
280 NEXT P
290 CALL KEY(S,KK,C(F,G))
300 C(F,G)=ABS(C(F,G))
310 CALL HCHAR(F*2,2,32)
320 CALL HCHAR(1,2*G+2,32)
330 IF C(F,G)-1 THEN 350
340 CALL HCHAR(2*F,2*G+2,30)
350 B(F,G)=C(F,G)
360 NEXT G
370 NEXT F
380 GOTO 60

```

```

10 CALL CLEAR
20 RANDOMIZE
30 INPUT "DIMENSION (MAX 10)
:" : A
40 D=1
50 I=0
60 GOTO 220
70 FOR F=D TO A
80 FOR G=D TO A
90 S=0
100 FOR I=-D TO D
110 FOR J=-D TO D
120 IF (F+J=0)+(G+I=0)+(F+J=
A+D)+(G+I=A+D) THEN 140
130 S=S+B(F+J,D+I)
140 NEXT J
150 NEXT I
160 S=2*S-B(F,G)
170 C(F,G)=0
180 IF (S<5)+(S>7) THEN 200
190 C(F,G)=D
200 NEXT G
210 NEXT F
220 CALL CLEAR
230 FOR F=D TO A
240 FOR G=D TO A
250 IF I THEN 280
260 IF RND>.5 THEN 280
270 C(F,G)=D
280 IF C(F,G)-1 THEN 300
290 CALL HCHAR(2*F,2*G+2,30)
300 B(F,G)=C(F,G)
310 NEXT G
320 NEXT F
330 GOTO 70

```


NIVEAU 3

<i>N°</i>	<i>Nom du programme</i>	<i>Thème et particularités</i>
1	Losanges	Pour vous aider à vous endormir...
2	Echiquier	Des bases graphiques, pour vos propres créations.
3	Couloirs	Un jeu de logique, à découvrir.
4	Jour de la semaine	Profitez-en pour faire le point sur les anniversaires.
5	Le pendu	Un jeu de lettres très classique (2 joueurs).
6	Bridge	Mettez au point vos propres problèmes de bridge.
7	Horloge 1	Ne jetez pas votre vieille télé : transformez-la en horloge !
8	Le loup et les agneaux	Un jeu de réflexion, pour 2 joueurs.
9	Belote	Vous n'aimez pas le bridge ? Alors, utilisez Belote !
10	Un taquin de calcul	Un bon entraînement au calcul (2 joueurs).
11	Poker	Une base de départ, pour mettre au point votre programme.
12	Hanoï	Un conseil : commencez petit !!!
13	Musique	Révisez vos notions de solfège, grâce au TI-99.
14	Pendu inversé	Cette fois, c'est TI-99 qui essaie de deviner votre mot !
15	Mathieu	Vous ne savez pas dessiner ? Laissez faire TI-99.
16	Machinamot	Peut-être TI-99 trouvera-t-il des mots qui n'existent pas ?
17	Compose	Pour voir, pour apprendre, et pour entendre...
18	Nicomaque	Entraînement au calcul mental (1 ou plusieurs joueurs).
19	Permutations	Un jeu de réflexion, pas toujours évident.
20	Tour de cartes	Et, pour finir, un peu de prestidigitation, par TI-99;
21	Parcours du cavalier	Très spectaculaire et instructif à plus d'un titre.
22	Mur de briques	Un grand classique, dans une version très complète.

Manipulation des chaînes et graphiques haute résolution

LE TRAITEMENT DES CHAINES DE CARACTERES : SEG\$

Nous avons déjà parlé, au niveau 2, de l'instruction SEG\$ en la comparant aux instructions équivalentes dans d'autres Basics.

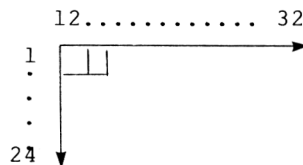
Rappelons que SEG\$ (EX\$, P, L) extrait de la chaîne EX\$ la sous-chaîne débutant au Pième élément et de la longueur L. Si $P + L$ est supérieur à la longueur de EX\$, la sous-chaîne est tronquée ; elle est vide si P est supérieur à cette même longueur. Si la valeur de P ou de L est négative ou nulle, le programme stoppe en affichant un message d'erreur.

En TI-Basic le signe & permet la concaténation de 2 chaînes. Ainsi, "VEHIC" et "ULE" assemblés selon A\$ = "VEHIC"&"ULE" équivalait à A\$ = "VEHICULE".

INSTRUCTIONS HCHAR et VCHAR

La haute résolution n'est pas directement accessible en TI-Basic. Les instructions HCHAR et VCHAR permettent d'afficher un caractère quelconque en un point de l'écran. On accèdera parfois à une fausse "haute résolution" en redéfinissant ce caractère au moyen de CHAR.

Rappelons que HCHAR et UCHAR travaillent avec une résolution 24 x 32, l'écran étant ainsi divisé en 768 points.



La syntaxe d'utilisation de ces instructions est la suivante :

CALL HCHAR (n° de ligne, n° de colonne, code, repet)

où . n° de ligne est compris entre 1 et 24
 . n° de colonne est compris entre 1 et 32

- . code est le code ASCII du caractère affiché (que l'on peut donc redéfinir).
- . repet est le nombre de répétitions de ce caractère ; ces répétitions sont HORIZONTALES avec HCHAR, VERTICALES avec VCHAR. Pour afficher 1 seul caractère ces 2 routines sont donc équivalentes.

Ainsi CALL VCHAR (10, 15, 84, 3) affiche la lettre T à la 10, 11 et 12ème lignes sur la 15ième.

INSTRUCTION GCHAR

Cette puissante instruction du TI-Basic permet de "récupérer" le code ASCII d'un caractère affiché sur l'écran. Ainsi CALL GCHAR (L, C, CODE) met dans la variable CODE la valeur ASCII du caractère de la Lième ligne et de la Cième colonne.

Losanges

- REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : interlude.
Difficulté du jeu : aucune.
- LE JEU** : Le programme va construire progressivement, sur l'écran, un losange noir, puis il va l'effacer progressivement, le reconstruire, et ainsi de suite.
Vous pourrez envisager de l'utiliser pour vous endormir, si les moutons ne vous suffisent plus. Une version, très légèrement modifiée, vous est également proposée, qui vous permettra de faire apparaître des motifs aléatoires après la construction des losanges noirs.
- LE PROGRAMME** : Une première boucle K (50-140) définit les dimensions des tracés successifs.
La deuxième boucle I (60-130) trace les bandes de points, en utilisant les instructions HCHAR (lignes 90 à 120).
Si la variable S (ligne 70) est à 1, alors il y aura effacement, par utilisation des instructions HCHAR avec le caractère 32 (lignes 170 à 200).
Pour obtenir un effacement aléatoire (deuxième programme), il suffira de rajouter la ligne 80 : 80 ON INT (1 + 4 * RND) GOTO 170, 180, 190, 200.
On obtient, alors, un effacement aléatoire.
- POUR JOUER** : On commence l'exécution du programme avec RUN.
Il suffit, dès lors, d'attendre et de regarder.
On pourra arrêter le déroulement à tout moment, en tapant CLEAR.
- EXTENSIONS POSSIBLES** : On pourra immédiatement, bien sûr, songer à introduire la couleur et, éventuellement, les sons.
On peut, à partir de là, imaginer toute une série de programmes fonctionnant sur le même principe (kaléidoscope, tapisseries, etc.).

```

10 CALL CLEAR
20 RANDOMIZE
30 CALL CHAR(128,"FFFFFFFFF
FFFFF")
40 S=0
50 FOR K=0 TO 15
60 FOR I=0 TO K
70 IF S THEN 170
90 CALL HCHAR(13+I/2,16+(I-K
)/2,128)
100 CALL HCHAR(13-I/2,16+(I-
K)/2,128)
110 CALL HCHAR(13-I/2,16+(K-
I)/2,128)
120 CALL HCHAR(13+I/2,16+(K-
I)/2,128)
130 NEXT I
140 NEXT K
150 S=1-S
160 GOTO 50
170 CALL HCHAR(13+I/2,16+(I-
K)/2,32)
180 CALL HCHAR(13-I/2,16+(I-
K)/2,32)
190 CALL HCHAR(13-I/2,16+(K-
I)/2,32)
200 CALL HCHAR(13+I/2,16+(K-
I)/2,32)
210 GOTO 130

```

```

10 CALL CLEAR
20 RANDOMIZE
30 CALL CHAR(128,"FFFFFFFFF
FFFFF")
40 S=0
50 FOR K=0 TO 15
60 FOR I=0 TO K
70 IF S=0 THEN 90
80 ON INT(1+4*RND)GOTO 170,1
80,190,200
90 CALL HCHAR(13+I/2,16+(I-K
)/2,128)
100 CALL HCHAR(13-I/2,16+(I-
K)/2,128)
110 CALL HCHAR(13-I/2,16+(K-
I)/2,128)
120 CALL HCHAR(13+I/2,16+(K-
I)/2,128)
130 NEXT I
140 NEXT K
150 S=1-S
160 GOTO 50
170 CALL HCHAR(13+I/2,16+(I-
K)/2,32)
180 CALL HCHAR(13-I/2,16+(I-
K)/2,32)
190 CALL HCHAR(13-I/2,16+(K-
I)/2,32)
200 CALL HCHAR(13+I/2,16+(K-
I)/2,32)
210 GOTO 130

```


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 cases sur 10 cases). Un repérage a été prévu, horizontalement, par des lettres (de A à H) et, verticalement par des nombres (de 1 à 8).
- LE PROGRAMME** : La ligne 20 définit un "pavé plein" qui sera utilisé pour dessiner les cases noires.
La boucle L (30-110) parcourt les lignes de l'échiquier, la boucle C (40-100) les colonnes.
La ligne 50 affiche les lettres, la ligne 60 les chiffres.
La ligne 70 détermine (par parité) si l'on doit imprimer une case blanche ou une case noire.
- 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 composées de 4 pavés côte à côte.
Les cases sont obtenues en largeur par une tabulation du type 2*C et 2*L en hauteur.
- EXTENSIONS POSSIBLES** : On pourra utiliser ce programme pour toute sorte de jeux se déroulant sur un damier (échecs, dames anglaises, le loup et les agneaux, etc.).

```
10 CALL CLEAR
20 CALL CHAR(128,"FFFFFFFFF
  FFFFFF")
30 FOR L=3 TO 10
40 FOR C=5 TO 12
50 CALL HCHAR(3,2*C,60+C)
60 CALL HCHAR(2*L,7,46+L)
70 IF (C+L)/2=INT((C+L)/2)TH
  EN 100
80 CALL HCHAR(2*L,2*C,128,2)
90 CALL HCHAR(2*L+1,2*C,128,
  2)
100 NEXT C
110 NEXT L
120 GOTO 120
```

Couloirs

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.
- LE JEU** : Il fait appel à des notions de logique élémentaire. Le joueur peut se présenter à l'entrée soit du couloir numéro 1, soit du couloir numéro 2. Dans ces deux entrées, sont disposés des tourniquets qui basculent à votre passage et vous laissent passer, une fois vers la droite, et la fois suivante vers la gauche.
De plus, un tourniquet analogue a été disposé dans le couloir central ; pourrez-vous prévoir où (sortie A ou B) va s'effectuer votre prochaine sortie ?
- LE PROGRAMME** : Les lignes 70 à 150 dessinent les couloirs et les trois tourniquets.
Les lignes 160 à 190 impriment les noms des entrées (1 et 2) et des sorties (A et B).
Les lignes 230 et 240 attendent votre choix.
Le décodage du chemin est fait par le TI-99 en utilisant trois variables logiques I, J et H, représentant les états des trois tourniquets (lignes 270 à 390).
- POUR JOUER** : Après la frappe de RUN, le couloir est dessiné sur l'écran, et l'on devra taper 1 ou 2 en réponse à la question du TI-99.
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 TI-99.
- EXTENSIONS POSSIBLES** : Elles seront très nombreuses ; il est très facile d'imaginer un couloir plus complexe avec d'autres entrées et d'autres sorties, et d'autres tourniquets. On peut imaginer un affichage graphique des résultats en dessinant, un bref instant, l'état des tourniquets après la réponse du joueur.
On peut aussi ajouter un test des réponses (jeu à plusieurs), décompter les points de chaque joueur, etc.).

```

10 CALL CLEAR
20 I=0
30 J=0
40 H=0
50 CALL CHAR(128,"1818181818
181818")
60 PRINT ,"ENTREE SORTIE":
: : : : : : : : : : : : : :
: : : : :
70 CALL VCHAR(3,5,128,20)
80 CALL VCHAR(3,13,128,20)
90 CALL VCHAR(3,9,128,10)
100 CALL VCHAR(19,9,128,4)
110 CALL VCHAR(11,7,128,7)
120 CALL VCHAR(11,11,128,7)
130 CALL HCHAR(14,9,84)
140 CALL HCHAR(20,7,84)
150 CALL HCHAR(20,11,84)
160 CALL HCHAR(3,11,66)
170 CALL HCHAR(3,7,65)
180 CALL HCHAR(23,7,49)
190 CALL HCHAR(23,11,50)
200 L=6
210 CALL HCHAR(L,19,62)
220 CALL SOUND(80,-1,0)
230 CALL KEY(S,K,S)
240 IF (S=0)+(K=49)*(K=50)TH
EN 230
250 CALL HCHAR(L,19,32)
260 CALL HCHAR(L,20,K)
270 IF K=49 THEN 320
280 J=1-J
290 IF J THEN 380
300 O=66
310 GOTO 350
320 I=1-I
330 IF I THEN 380
340 O=65
350 CALL HCHAR(L,24,O)
360 L=L+1
370 GOTO 210
380 H=1-H
390 IF H THEN 300 ELSE 350

```

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 (sous forme un peu abrégée).
- LE PROGRAMME** : Les lignes 10 à 50 contrôlent les entrées.
La ligne 20 s'occupe de la fin de partie.
La ligne 90 permet une correction pour les dates antérieures à 1582 (mise en service du calendrier actuel).
Les lignes 100 et 110 permettront l'affichage en clair du mois dans la réponse (sur 4 lettres).
La formule utilisée apparaît aux lignes 140 et 150.
Les lignes 160 et 170 permettront l'affichage en clair du jour dans la réponse.
La ligne 180 imprime le résultat.
Le sous-programme 200-220 intervient pour tenir compte des mois de janvier et février.
- POUR JOUER** : Lancer le programme par RUN, puis entrer le jour suivi de ENTER, le mois (entre 1 et 12) suivi de ENTER, et l'année (sous la forme 1845, par exemple) suivie de ENTER.
Le résultat est très rapidement affiché et le programme revient sur la question JOUR.
Pour continuer, il suffit de retaper un jour, un mois, une année.
Pour terminer, il faut répondre 0 à la question JOUR.
- EXTENSIONS POSSIBLES** : On pourra entrer les mois en clair et faire afficher le nom du jour en entier.
On peut envisager de faire afficher les dates de fêtes particulières (Noël, Pâques, les Rameaux, Pentecôte...).
- On peut écrire un programme qui afficherait les jours fériés pour une année donnée, etc.

```

10 INPUT "JOUR      ":"J
20 IF J THEN 40
30 STOP
40 INPUT "MOIS(1-12) ":"M
50 INPUT "ANNEE      ":"A
60 B=A
70 S=INT(A/100)-INT(A/400)
80 IF A>1582 THEN 100
90 S=2
100 M$="JANVFEVRMARSABRILMAI
    JUINJUILAOUTSEPTOCTONOVEDECE
    "
110 M$=SEG$(M$,4*M-3,4)
120 IF M>=3 THEN 140
130 GOSUB 200
140 S=J-1+INT(13*(M+1)/5)+INT
    (5*A/4)-S
150 S=S-7*INT(S/7)
160 J$="DIMANCHELUNDI  MARD
    I  MERCREDIJEUDI  VENDREDI
    SAMEDI  "
170 J$=SEG$(J$,8*S+1,8)
180 PRINT : : : "LE";J;M$;B;"
    ETAIT UN ";J$: : : :
190 GOTO 10
200 M=M+12
210 A=A-1
220 RETURN

```

Le pendu

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de lettres (version deux joueurs).
Difficulté du jeu : moyenne.
- LE JEU** : Il serait sans doute assez difficile de retrouver le nom de l'inventeur de ce jeu très à la mode, mais il est certain qu'il existe depuis des temps immémoriaux, quelquefois pratiqué sous des formes assez différentes (par exemple, reconstituer un petit cochonnet en lançant des dés.).
L'apparition des jeux sur micro-ordinateur et la magie des écrans de télévision lui ont fait connaître un renouveau certain ; dans la version retenue, l'un des joueurs doit entrer un mot de longueur quelconque. L'ordinateur affiche alors des tirets à l'emplacement des lettres de ce mot et demande, au deuxième joueur, de proposer une lettre.
Chaque réponse correcte fait apparaître la lettre à son emplacement exact dans le mot, et chaque réponse fausse fait progresser le dessin de la potence.
Pour gagner, il suffit de réussir à trouver le mot inconnu avant d'être pendu.
- LE PROGRAMME** : Le mot à deviner est introduit en début de programme (ligne 30) ; une variable B\$, en ligne 20 contient tous les éléments du dessin.
La boucle I (50-70) fabrique un mot P\$ qui contient des tirets à l'emplacement de chacune des lettres du mot initial.
La boucle I (110-130) s'occupe du dessin (plus ou moins complet, selon la valeur de P).
Les lignes 160 et 190 affichent le résultat et arrêtent le jeu.
La lettre essayée est entrée en ligne 210 et testée dans une boucle I (230-260) ; le sous-programme 300-320 remplace chaque tiret par la lettre correspondante.
- POUR JOUER** : Après le lancement du programme par RUN, l'un des joueurs doit entrer son mot et taper ENTER.
Des tirets sont affichés à l'emplacement de chacune des lettres du mot, et la question LETTRE apparaît sur l'écran ; l'autre joueur répond alors en proposant une lettre suivie de ENTER.
Si la lettre est bonne, elle viendra rempla-

cer un ou plusieurs tirets ; sinon, la po-
tence va commencer à se dessiner.
Le mot exact sera affiché dans le cas où
l'on aurait perdu.

EXTENSIONS POSSIBLES : Il sera intéressant de constituer une
liste des mots dans lesquels TI-99 pourra
piocher au hasard ; on pourra aussi amélio-
rer le dessin et modifier le nombre d'essais
conduisant à la pendaïson, en fonction de la
difficulté du mot.

```

10 CALL CLEAR
20 B$="I II I888HHHHHHHHH I
   A ! ! ____"
30 INPUT "VOTRE MOT :";R$
40 P$=""
50 FOR I=1 TO LEN(R$)
60 P$="_"&P$
70 NEXT I
80 P=11
90 CALL CLEAR
100 PRINT TAB(5);P$
110 FOR I=10 TO P STEP -1
120 PRINT TAB(20);SEG$(B$,1+
3*I,3)
130 NEXT I
140 PRINT : : :
150 IF P$<>R$ THEN 180
160 PRINT "BRAVO"
170 STOP
180 IF P=1 THEN 210
190 PRINT "LE MOT ETAIT ";R$
200 STOP
210 INPUT "VOTRE LETTRE ? ";
S$
220 T=0
230 FOR I=1 TO LEN(R$)
240 IF SEG$(R$,I,1)<>S$ THEN
260
250 GOSUB 300
260 NEXT I
270 IF T THEN 90
280 P=P-1
290 GOTO 90
300 T=1
310 P$=SEG$(P$,1,I-1)&S$&SEG
$(P$,I+1,LEN(P$)-I)
320 RETURN

```


Bridge

- REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : variable, selon le problème.
- 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 30 et 40 définissent un jeu de 52 cartes.
La variable B\$, en ligne 60 permettra de donner en abrégé le nom des couleurs et aussi d'améliorer la désignation de certaines cartes (10, valet, dame, roi et as).
La boucle J (40-140) réalise le tirage de 4 jeux de 13 cartes.
La boucle I (80-200) réalise tirage et impression. La carte est choisie parmi les 52 disponibles, en ligne 100.
La ligne 110 calcule la couleur de la carte tirée.
La ligne 120 vérifie que cette carte n'a pas déjà été tirée.
Les lignes 150 et 160 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 140 et 170.
La ligne 180 permet d'éliminer les cartes déjà tirées.
- POUR JOUER** : On lance le programme par RUN.
Quatre séries de 13 cartes sont affichées verticalement.
- EXTENSIONS POSSIBLES** : Sur un matériel disposant, comme c'est le cas du TI-99 d'un affichage performant, on peut imaginer de présenter 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 TI-99, par exemple), ou introduire des possibilités de jeu de la carte, y compris pour TI-99.

```

10 CALL CLEAR
20 RANDOMIZE
30 R$="234567891VDRA"
40 R$=R$&R$&R$&R$
50 U=7
60 B$="TRCACOP10AAS"
70 V=LEN(B$)
80 FOR I=1 TO V
90 FOR J=1 TO 4
100 X=INT(LEN(R$)*RND+1)
110 S=2*INT((X-1)/V)+1
120 IF SEG$(R$,X,1)="0" THEN
130 T=X-V*INT((X-1)/V)
140 PRINT TAB(U*J-U+1);SEG$(
R$,X,1);
150 IF T<9 THEN 170
160 PRINT SEG$(B$,T,1);
170 PRINT TAB(U*J-U+4);SEG$(
B$,S,2);
180 R$=SEG$(R$,1,X-1)&"0"&SE
G$(R$,X+1,LEN(R$)-X)
190 NEXT J
200 NEXT I

```

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 20 vous permettra d'entrer l'heure sous forme d'un entier de 4 chiffres (par exemple : 1024 signifiera 10 heures 24 minutes).
La ligne 80 effectue le décodage de chaque chiffre.
Les lignes 140 à 280 réalisent le dessin, en décalant l'impression au fur et à mesure (à l'aide de la variable K, incrémentée en ligne 290).
La ligne 320 assure le passage de 23 heures 59 minutes à 00 heure, 00 minute.
La ligne 360 permet d'ajouter une heure toutes les 60 minutes.
Les lignes 500 et 510 permettent de modifier la durée de la pause pour régler l'horloge.
- POUR JOUER** : Dès le lancement par RUN, le programme se place en attente de l'heure ; on tape au prochain top sonore et on attend le top pour appuyer sur ENTER. L'heure s'affiche alors assez rapidement et l'horloge est en route. Pour l'arrêter, il faudra appuyer sur la touche CLEAR.
L'heure est entrée sous forme d'un entier ayant impérativement 4 chiffres ; pour 5 heures, 6 minutes, on veillera à taper 0506 suivi de ENTER.
- EXTENSIONS POSSIBLES** : Il sera intéressant de modifier le programme pour en faire un réveil, avec alarme ; on peut aussi améliorer le graphisme, introduire la couleur.

```

10 CALL CLEAR
20 INPUT "HEURE (FORME XXXX)
?:A$
30 F=6
40 S=1
50 G=F-S
60 CALL CLEAR
70 K=S
80 A=VAL (SEG$(A$,K,1))
90 F$="FFFFFFF"
100 CALL CHAR(128,F$%F$)
110 CALL CHAR(129,F$)
120 CALL CHAR(130,"00000000"
&F$)
130 IF (A=S)+(A=G-S) THEN 150
140 CALL HCHAR(S+S,5+F*K-G,1
28,3)
150 FOR J=S TO G
160 IF A*((A<=3)+(A=F+S))THE
N 180
170 CALL HCHAR(5+J,5+F*K-G,1
28)
180 IF (A=G)+(A=F) THEN 200
190 CALL HCHAR(5+J,F*K+2,128
)
200 IF A*(A-2*S)*(A-F)*(A-F-
2*S) THEN 220
210 CALL HCHAR(5+G+J-S,5+F*K
-G,128)
220 IF A=2*S THEN 240
230 CALL HCHAR(5+G+J-S,F*K+2
,128)
240 NEXT J
250 IF A<=S THEN 270

260 CALL HCHAR(5+G,5+F*K-G+S
,128,2)
270 IF (A=S)+(A=G-S)+(A=F+S)
THEN 290
280 CALL HCHAR(F+8,5+F*K-G,1
28,3)
290 K=K+S
300 IF K<G THEN 80
310 A=S+VAL(A$)
320 IF A-2360 THEN 340
330 A=0
340 A$="000"&STR$(A)
350 A$=SEG$(A$,LEN(A$)-3,4)
360 IF SEG$(A$,2,2)<>"60" TH
EN 380
370 A$=STR$(VAL (SEG$(A$,1,2*
S)+S)&"00")
380 H=S
390 FOR J=S TO 120
400 FOR I=16 TO S
410 IF H=0 THEN 450
420 CALL HCHAR(5+G,5+F+G,129
)
430 CALL HCHAR(5+G+S,5+F+G,1
30)
440 GOTO 470
450 CALL HCHAR(5+G,5+F+G,32)
460 CALL HCHAR(5+G+S,5+F+G,3
2)
470 NEXT I
480 H=1-H
490 NEXT J
500 FOR PAUSE=1 TO 15000
510 NEXT PAUSE
520 GOTO 60

```

Le loup et les agneaux

REFERENCES

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

LE JEU

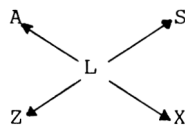
: C'est un jeu assez connu qui peut être pratiqué indifféremment sur un damier 8 sur 8, ou 10 sur 10. Il est présenté, ici, dans une version pour un damier de 8 cases ; l'un des joueurs choisit le loup qui se déplace suivant les diagonales, en restant sur les cases noires, d'une case à chaque coup, et 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 140 contient les positions de départ.
Les boucles L (160-210) et C (170-200) dessinent le damier, et les lignes 220 à 260 placent le loup et les agneaux.
La ligne 270 gère le changement de joueur.
La ligne 280 affiche L si c'est le joueur qui manipule le loup qui a le trait, et A si c'est son adversaire, de plus elle attend le coup du joueur.
La ligne 290 choisira le programme analysant la réponse, en fonction du joueur concerné.

POUR JOUER

: RUN déclenche l'arrêt sur l'attente d'une entrée : Ø et ENTER si l'on veut que les agneaux commencent, et 1 pour le loup. Après quoi, le damier s'affiche et la phrase COUP A pour les agneaux, et COUP L pour le jeu du loup.
Les agneaux répondent par un numéro, suivi de D ou G (pour droite ou gauche).
Ainsi 2D provoque le déplacement de l'agneau numéro 2 qui monte en diagonale d'une case sur la droite. Le déplacement effectué est affiché, puis on obtient une question identique pour le loup qui répondra en tapant A, S, 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 TI-99 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 TI-99).

<pre> 10 CALL CLEAR 20 DIM B\$(1) 30 B\$(1)="A" 40 B\$(0)="L" 50 CALL SCREEN(15) 60 CALL CHAR(128,"O") 70 CALL CHAR(129,"0040404040 40407C") 80 CALL CHAR(130,"0010301010 101038") 90 CALL CHAR(131,"0038440408 10207C") 100 CALL CHAR(132,"003844041 8044438") 110 CALL CHAR(133,"000818284 87C0808") 120 CALL COLOR(13,15,2) 130 INPUT "AGNEAUX (O) OU LO UP (1) ? ":J 140 A\$="6163656714" 150 CALL CLEAR 160 FOR L=8 TO 15 170 FOR C=10 TO 17 180 IF (L+C)/2=INT((L+C)/2) T HEN 200 190 CALL HCHAR(L,C,128) 200 NEXT C 210 NEXT L 220 CALL HCHAR(VAL(SEG\$(A\$,9 ,1))+7,VAL(SEG\$(A\$,10,1))+9, </pre>	<pre> 129) 230 CALL HCHAR(VAL(SEG\$(A\$,1 ,1))+7,VAL(SEG\$(A\$,2,1))+9,1 30) 240 CALL HCHAR(VAL(SEG\$(A\$,3 ,1))+7,VAL(SEG\$(A\$,4,1))+9,1 31) 250 CALL HCHAR(VAL(SEG\$(A\$,5 ,1))+7,VAL(SEG\$(A\$,6,1))+9,1 32) 260 CALL HCHAR(VAL(SEG\$(A\$,7 ,1))+7,VAL(SEG\$(A\$,8,1))+9,1 33) 270 J=ABS(J-1) 280 INPUT " COUP "&B\$(J)&" ":R\$ 290 IF J THEN 330 300 C=10 310 A=VAL(SEG\$(A\$,C-1,10))- (R\$="X")*(C+1)-(R\$="Z")*(C-1) +(R\$="A")*(C+1)+(R\$="S") *(C-1) 320 GOTO 350 330 C=2*VAL(SEG\$(R\$,1,1)) 340 A=VAL(SEG\$(A\$,C-1,2))+9* (SEG\$(R\$,2,1)="D")+11*(SEG\$(R\$,2,1)="G") 350 A\$=SEG\$(A\$,1,C-2)&STR\$(A)&SEG\$(A\$,C+1,10) 360 GOTO 150 </pre>
--	--

Belote

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de cartes.
Difficulté de programmation : 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 des problèmes faisant intervenir des données de 20 cartes sur 32.
- LE PROGRAMME** : En ligne 30, on crée un tableau pouvant contenir les 20 cartes.
La boucle I (50-100) assure le tirage aléatoire de 20 nombres, pris parmi 32 ; la boucle J (70-90) vérifie que l'on ne réalise pas deux fois le même tirage.
La deuxième boucle I (110-340) assure, ensuite, l'affichage, les noms des cartes étant affichés en clair, et la couleur étant décodée et affichée en clair, grâce aux lignes 170, 190, 210 et 230.
- 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 ; on pourra également se servir de ce programme comme d'un noyau pour la mise au point de programmes de jeu de cartes sur l'ordinateur (qui pourra, si nécessaire, servir de partenaire: voir l'exemple du POKER, dans le même ouvrage).

```

10 CALL CLEAR
20 RANDOMIZE
30 DIM A(20)
40 K=1
50 FOR I=K TO 20
60 A(I)=K+INT(32*VRND)
70 FOR J=K TO I-K
80 IF A(I)=A(J) THEN 60
90 NEXT J
100 NEXT I
110 FOR I=1 TO 20
120 S$="CA"
130 S=1+A(I)-8*INT(A(I)/8)
140 R$=STR$(S+6)
150 R$=SEG$(R$,1,2)
160 IF S=5 THEN 180
170 R$="VA"
180 IF S=6 THEN 200
190 R$="DA"
200 IF S=7 THEN 220
210 R$="RD"
220 IF S=9 THEN 240
230 R$="AS"
240 IF A(I)>24 THEN 260
250 S$="TR"
260 IF A(I)>16 THEN 280
270 S$="PI"
280 IF A(I)>8 THEN 300
290 S$="CO"
300 PRINT TAB(7*I-6);R$;TAB(
7*I-3);S$;
310 K=K+1
320 IF K=6 THEN 340
330 K=1
340 NEXT I

```


Un taquin de calcul

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (deux 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 TI-99), et doit le transformer à l'aide de 5 opérations qu'il choisit lui-même, ainsi d'ailleurs que les opérateurs. Le résultat qu'il a obtenu et les opérations qu'il a utilisées seront affichés sur l'écran, le décodeur devra alors essayer de retrouver le nombre initial en effectuant les opérations en sens inverse.
- LE PROGRAMME** : La ligne 30 permet d'obtenir l'entier de départ compris entre 5 et 20. Une boucle I (100-180) permet d'entrer successivement les 5 opérations. 4 sous-programmes sont utilisés pour les calculs (aux lignes 230 à 290). Après chaque calcul, le résultat est affiché (seul, le codeur devra regarder l'écran pendant cette opération) et le résultat définitif restera affiché pour le décodeur à la fin des calculs (ligne 170). La ligne 190 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 ENTER. 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 TI-99 participer davantage en lui confiant le rôle du codage, etc.

```

10 CALL CLEAR
20 RANDOMIZE
30 X=INT(16*RND+5)
40 PRINT "NOMBRE INITIAL";X
50 L=1
60 PRINT "ENTREZ 5 OPERATION
S"
70 F=X
80 PRINT "FORME +6,/5...."
90 DIM R$(5)
100 FOR I=L TO 5
110 INPUT R$(I)
120 A=ASC(SEG$(R$(I),L,1))-4
1
130 IF (A<=0)+(A=3)+(A=5)+(A
>6) THEN 110
140 IF A<=3 THEN 160
150 A=A/2+1
160 ON A GOSUB 230,250,270,2
90
170 PRINT "NOMBRE OBTENU";F
180 NEXT I
190 PRINT : "OPERATIONS":R$(4
);TAB(5);R$(1);TAB(10);R$(5)
;TAB(15);R$(3);TAB(20);R
$(2);"BON COURAGE"
200 CALL KEY(5,KK,SS)
210 IF SS=0 THEN 200
220 GOTO 10
230 F=F*VAL(SEG$(R$(I),2,3))
240 RETURN
250 F=F+VAL(SEG$(R$(I),2,3))
260 RETURN
270 F=F-VAL(SEG$(R$(I),2,3))
280 RETURN
290 F=F/VAL(SEG$(R$(I),2,3))
300 RETURN

```

Poker

- REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : jeu contre le TI-99
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 TI-99.
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 TI-99 de jouer.
- LE PROGRAMME** : Les lignes 30 et 40 définissent les cartes.
Une variable N est dimensionnée à 15, en ligne 50, pour recevoir 10 cartes joueur et 5 cartes TI-99.
Le tirage sera effectué par une boucle I (70-120) et le contrôle par une boucle J (90-110).
L'affichage des cartes est réalisé par le sous-programme 240-320.
Une autre boucle I (180-200) réalise le deuxième tirage du joueur, la ligne 220 permettant le deuxième affichage, et la boucle I (240-290), 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 ENTER s'il veut arrêter la partie.
Pour jouer, il répondra simplement ENTER s'il s'estime servi, ou il entrera les numéros des cartes qu'il refuse suivis de ENTER.
Par exemple, s'il tape 235 et ENTER, son nouveau jeu conservera les cartes 1 et 4.
Son jeu sera alors affiché ainsi que celui du TI-99, ce dernier n'ayant pas, dans ce programme, la possibilité de reprendre des cartes.
Il suffira ensuite de faire ENTER 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 TI-99, l'introduction des règles sur les paris et même, pourquoi pas, des possibilités de bluff.

```

10 CALL CLEAR
20 S=6
30 V$="7 8 9 10VADAR0AS"
40 C$="TRCAC0PI"
50 DIM N(15)
60 RANDOMIZE
70 FOR I=1 TO 15
80 N(I)=INT(32*RND)
90 FOR J=1 TO I-1
100 IF N(I)=N(J) THEN 80
110 NEXT J
120 NEXT I
130 K=0
140 PRINT "CARTES REFUSEES"
150 GOSUB 240
160 IF R$<>"0" THEN 180
170 STOP
180 FOR I=1 TO LEN(R$)
190 N(VAL(SEG$(R$,I,1)))=N(I
+S-1)
200 NEXT I
210 CALL CLEAR
220 GOSUB 240
230 K=10
240 FOR I=1 TO 5
250 J=2*(N(I+K)-4*INT(N(I+K)
/4)+1)-1
260 F=2*(N(I+K)-7*INT(N(I+K)
/7)+1)-1
270 R$=SEG$(V$,F,2)
280 PRINT TAB(S*I-4);I;TAB(S
*I-S);R$;";";SEG$(C$,J,2)
290 NEXT I
300 INPUT R$
310 IF K THEN 10
320 RETURN

```

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** : Le caractère 128, en ligne 70, servira à dessiner les disques.
Le nombre de disques est entré en ligne 80 et refusé s'il y a plus de 8 disques.
La boucle I (100-120) permet de fabriquer une variable A (I) qui contient l'état de la pile sous forme numérique (en ligne 20, trois variables numériques sont prévues pour conserver l'état des trois piles).
Les socles sont dessinés par les lignes 140 à 160, et les trois piles par les boucles emboîtées I (200-260) et J (210-250).
Le contrôle du coup est effectué par la ligne 320 et l'état des piles est modifié par les lignes 330 et 340.
- POUR JOUER** : Après RUN qui lance le programme, on entre le nombre de disques demandés, suivi de ENTER. 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 ENTER 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 CALL CLEAR
20 DIM A(3)
30 U=9
40 A(1)=U
50 A(2)=U
60 A(3)=U
70 CALL CHAR(128,"FFFFFFFFF
FFFFF")
80 INPUT "NOMBRE DE DISQUES
:" :N
90 IF N>7 THEN 10
100 FOR I=0 TO N-1
110 A(1)=VAL(STR$(A(1))&STR$(
(N-I))
120 NEXT I
130 CALL CLEAR
140 CALL HCHAR(15,3,94,9)
150 CALL HCHAR(15,13,94,9)
160 CALL HCHAR(15,23,94,9)
170 CALL HCHAR(17,7,49)
180 CALL HCHAR(17,17,50)
190 CALL HCHAR(17,27,51)
200 FOR I=0 TO N-1
210 FOR J=1 TO 3
220 B$=STR$(A(J))
230 IF LEN(B$)<=I+1 THEN 250
240 CALL HCHAR(U-I+5,(U+1)*J
-U+3,128,VAL(SEG$(B$,I+2,1))
)
250 NEXT J
260 NEXT I
270 INPUT "TOUR DE DEPART " :
D
280 IF A(D)=U THEN 270
290 INPUT "TOUR D'ARRIVEE " :
W
300 B$=STR$(A(D))
310 C$=STR$(A(W))
320 IF SEG$(B$,LEN(B$),1)>SE
G$(C$,LEN(C$),1) THEN 270
330 A(W)=VAL(C$&SEG$(B$,LEN(
B$),1))
340 A(D)=VAL(SEG$(B$,1,LEN(B
$)-1))
350 GOTO 130

```

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 (110-130) dessine la portée. Les deux boucles I (140-160) et (170-190) affichent les numéros des lignes. La ligne 200 définit les noms des notes, et la ligne 210 réalise le tirage aléatoire d'une note. Le nom de la note est calculé à la ligne 85 et il est affiché par la ligne 230, suivi de la mention GRAVE ou AIGUE, selon le cas. La réponse est entrée en ligne 240. La ligne 250 s'occupe du décodage de la réponse. Les lignes 260 et 270 affichent la note. La ligne 280 regarde si l'on a gagné ou perdu et affiche le message correspondant. Les lignes 290 à 310 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 LIGNE. On répond en donnant un numéro de ligne (de 1 à E), suivi de ENTER. La note est alors dessinée et le message BRAVO ou PERDU apparaît, dans ce cas, à l'écran. On peut taper une touche quelconque pour continuer, et s'arrêter en tapant CLEAR.
- EXTENSIONS POSSIBLES** : On peut étendre à une autre clé que la clé de sol, on peut introduire de la couleur, des sons, etc.

```

10 RANDOMIZE
20 DIM TON$(1),RR$(1)
30 RR$(1)="BRAVO"
40 RR$(0)="PERDU"
50 TON$(1)="GRAVE"
60 TON$(0)="AIGUE"
70 CALL CHAR(128,"0000FFFFFF
FF")
80 CALL CHAR(129,"0C0C0C0C0C
0C0C0C")
90 CALL CHAR(130,"1C3C7C7C7C
7C3C1C")
100 CALL CLEAR
110 FOR I=1 TO 5
120 CALL HCHAR(9+2*I,8,128,2
4)
130 NEXT I
140 FOR I=1 TO 9
150 CALL HCHAR(I+7,3,48+I)
160 NEXT I
170 FOR I=10 TO 14
180 CALL HCHAR(I+7,3,55+I)
190 NEXT I
200 N$="SI LA SOLFA MI RE DO
"
210 X=INT(14*RND+1)
220 T=X-1-7*INT((X-1)/7)
230 PRINT "NOTE ";SEG$(N$,3*
T+1,3);" ";TON$(-(X>7))
240 INPUT "LIGNE :";L$
250 L=(L$>"@")*(55-ASC(L$))-
(ASC(L$)-48)*(L$<" ":)
260 CALL VCHAR(3+X,15,129,2)
270 CALL HCHAR(X+5,15,130)
280 PRINT RR$(-(L=X))
290 CALL KEY(5,KK,SS)
300 IF SS=0 THEN 290
310 GOTO 100

```


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 TI-99 qui va chercher à deviner votre mot, mais il joue quelque peu au hasard et ne possède pas votre intuition ; il lui faudra donc, en règle générale, beaucoup d'essais pour découvrir la solution. Pour augmenter ses chances, on a fait appel à une formule qui lui permet de tirer d'avantage dans les voyelles puis, dans les consonnes courantes et, enfin, dans les consonnes plus rares.
- LE PROGRAMME** : La ligne 30 permet d'enregistrer, dans une variable alphanumérique B\$, les lettres de l'alphabet, dans l'ordre : voyelles, consonnes usuelles, autres consonnes. Le mot à deviner est introduit en ligne 60 (en fait, il suffirait d'introduire sa longueur). La boucle I (70-90) imprime des tirets aux emplacements des lettres à trouver. La formule, placée en ligne 110, permet de modifier légèrement la probabilité de tirage, en faveur des voyelles et des consonnes usuelles. .
La ligne 130 provoque un nouveau tirage, si la lettre avait déjà été essayée.
La ligne 160 affiche la lettre choisie, et la boucle I (170-200) réimprime les tirets et les lettres trouvées.
La ligne 210 incrémente le compteur d'essais.
La ligne 230 affiche le nombre d'essais réalisés, en fin de partie.
- POUR JOUER** : Lancer le programme par RUN.
Entrer le mot à découvrir, suivi de ENTER.
Des tirets sont alors affichés (un par lettre) et le TI-99 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 CLEAR.
- 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 TI-99 ; On pourra aussi imaginer d'ajouter un affichage graphique, par exemple une potence ou une voiture qui avance vers un précipice, etc.

Avec une minidisquette, on pourra faire retenir, au TI-99 les mots les plus couramment utilisés par les joueurs, etc.

```

10 CALL CLEAR
20 RANDOMIZE
30 D$="YAEIOUBCDFGJLMNPRSTVW
HKQXZ"
40 C=1
50 T$=""
60 INPUT "VOTRE MOT :":R$
70 FOR I=1 TO LEN(R$)
80 T$="_"&T$
90 NEXT I
100 PRINT :T$: : "JE CHERCH
E... ESSAI";C
110 X=-6*(RND<.7)-13*(RND<.5
)-7*(RND<.4)
120 X=INT(X*RND+1)
130 IF SEG$(D$,X,1)="_" THEN
110
140 E$=SEG$(D$,X,1)
150 D$=SEG$(D$,1,X-1)&_"%SE
G$(D$,X+1,26-X)
160 PRINT : "JE PRENDS :";E
$
170 FOR I=1 TO LEN(R$)
180 IF E$<>SEG$(R$,I,1) THEN
200
190 T$=SEG$(T$,1,I-1)&E$&SEG
$(T$,I+1,LEN(T$)-I)
200 NEXT I
210 C=C+1
220 IF T$<>R$ THEN 100
230 PRINT : "TROUVE EN";C-1; "
COUPS"

```

Mathieu

- REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : dessin aléatoire.
Difficulté du jeu : aucune.
- LE JEU** : On utilise la définition d'une courbe en coordonnées polaires, et la fonction HCHAR, pour obtenir, sur l'écran, des dessins aléatoires (de type fleur), plus ou moins élégants.
Les deux paramètres principaux, A et B, pourront être, si nécessaire, entrés à la main (instruction INPUT) pour faire une étude plus systématique des diverses courbes possibles.
- LE PROGRAMME** : La ligne 20 permet de définir le rayon maximum, et la ligne 30 corrige ce paramètre pour utiliser, au mieux, la surface de l'écran.
On pourra, évidemment, modifier ces deux paramètres en veillant à rester dans les limites de l'écran.
Les lignes 50 et 60 définissent deux variables aléatoires A et B, choisies ici entre 0 et 3 ; l'allure des courbes dépendant de ces variables, on pourra modifier leur intervalles de valeurs, ou même les entrer à la main, en écrivant 50 INPUT A, B.
La boucle I (70-130) réalise le tracé, point par point ; le rayon R est calculé à la ligne 80, et le passage en coordonnées cartésiennes (indispensable pour HCHAR) est effectué aux lignes 90 et 100.
La ligne 120 réalise l'impression des points.
La ligne 140 affiche un message FINI.
Les lignes 150 et 160 permettent l'exécution d'un autre dessin, dès que l'on effleure une touche du clavier.
- POUR JOUER** : On lance le programme en tapant RUN.
Le dessin est alors automatiquement exécuté.
Le message FINI (en bas, à gauche) signale la fin du tracé.
Pour exécuter un autre dessin, taper une touche quelconque.
- EXTENSIONS POSSIBLES** : On pourra essayer d'utiliser beaucoup d'autres fonctions pour réaliser des dessins encore plus extraordinaires (par exemple, des dessins simulant une troisième dimension).

```

10 CALL CLEAR
20 H=2.5
30 K=5
40 RANDOMIZE
50 A=3*RND
60 B=3*RND
70 FOR I=0 TO 6.28 STEP .052
80 R=H*SIN(3*I)
90 X=12+INT(K*R*COS(A*I))
100 Y=17+INT(K*R*SIN(B*I))
110 IF (X<1)+(X>24)+(Y<2)+(Y
>32) THEN 130
120 CALL HCHAR(X,Y,42)
130 NEXT I
140 PRINT "FINI"
150 CALL KEY(S,C,S)
160 IF S=0 THEN 150 ELSE 327
67

```

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 TI-99 de faire la démonstration de ses capacités à manipuler des chaînes de caractères : on part simplement d'un préfixe et d'un suffixe d'origine grecque, que l'on va assembler au hasard ; on tombera bien sûr, de temps à autre, sur des mots absurdes ou même légèrement équivoques.
Un nouveau mot est confectionné à chaque essai, mais il est possible d'obtenir, plusieurs fois, le même mot.
- LE PROGRAMME** : La ligne 10 permet d'entrer la liste des préfixes dans une variable A\$ (il sera bien sûr possible d'ajouter d'autres préfixes, en ne dépassant pas 5 lettres et en ajoutant, si nécessaire, des blancs).
La ligne 20 contiendra, de la même façon, la liste des suffixes qui sont placés dans une autre variable alphanumérique B\$.
La ligne 30 effectue le choix d'un suffixe au hasard.
La ligne 40 effectue le choix d'un préfixe au hasard.
Le mot obtenu est affiché par la ligne 50, et la ligne 70 permettra de recommencer un tirage.
- POUR JOUER** : Lancer l'exécution du programme par RUN.
Un premier mot sera immédiatement affiché.
Pour obtenir un nouveau mot, il convient d'appuyer sur une touche quelconque (sauf CLEAR) ; un nouveau mot sera alors affiché à la suite du précédent.
Taper CLEAR pour sortir du programme.
- EXTENSIONS POSSIBLES** : On pourra très facilement augmenter la liste des mots disponibles (sur TI-99, il sera préférable d'utiliser des instructions DATA qui permettront de ne pas avoir de contraintes sur la longueur des mots).
On pourra aussi utiliser un procédé analogue pour faire confectionner des phrases à l'ordinateur, lui faire inventer des poèmes, etc.
On pourra aussi transformer le programme initial en un jeu de société qui consisterait par exemple, à prendre des paris sur le prochain mot qui va apparaître. Et confier, éventuellement, la gestion des paris au TI-99.

```

1 CALL CLEAR
2 RANDOMIZE
10 A$="ANDROCOSMOMICROHELIO
PEDOPORNOFALEOMACROHYDROSTEN
OPATHO OLEOMYTHOHOMEO OC
TO"
20 B$="CRATEPHILEPHOBETROPEG
YNE MANE NAUTEPHAGEFUGE PATH
ECARDEDERMESCOPEVORE PHO
NE"
30 I=5*INT((LEN(A$)/5)*RND)+
1
40 J=5*INT((LEN(B$)/5)*RND)+
1
50 PRINT TAB(5);SEG$(A$,I,5)
;SEG$(B$,J,5)
60 CALL KEY(5,K,S)
70 IF 'S THEN 10 ELSE 60

```

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 de 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 (60-80) dessine la portée.
La variable N\$, en ligne 90, contient les noms des notes.
La deuxième boucle I (110-160) permet de faire le choix d'une note de façon aléatoire à la ligne 120 ; son nom est calculé en ligne 130 et la note est dessinée par les lignes 140 et 150.
Le nom de la note est affichée en ligne 130.
La ligne 180 permet de recommencer.
- POUR JOUER** : Lancer l'exécution du programme en tapant RUN.
La portée sera alors très rapidement affichée, puis les notes sont représentées, avec leur nom affiché en bas de l'écran.
Une pression sur une touche quelconque provoquera l'affichage d'une nouvelle série de notes, et une pression sur la touche CLEAR provoque la fin de partie.
- EXTENSIONS POSSIBLES** : On pourra affiner les règles de composition, afficher davantage de notes, introduire des dièses et des bémols, améliorer le graphisme, faire jouer la mélodie affichée.

```

10 RANDOMIZE
20 CALL CHAR(128,"0000FFFFFF
FF")
30 CALL CHAR(129,"0C0C0C0C0C
0C0C0C")
40 CALL CHAR(130,"1C3C7C7C7C
7C3C1C")
50 CALL CLEAR
60 FOR I=1 TO 5
70 CALL HCHAR(5+2*I,3,128,30
)
80 NEXT I
90 N$="SI LA SOLFA MI RE DO
"
100 PRINT " ";
110 FOR I=1 TO 9
120 X=INT(14*RND+1)
130 PRINT SEG$(N$,3*(X+7*(X>
7))-2,3);
140 CALL VCHAR(1+X,3*I+2,129
,2)
150 CALL HCHAR(X+3,3*I+2,130
)
160 NEXT I
170 CALL KEY(5,KK,SS)
180 IF SS=0 THEN 170 ELSE 50

```


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 identiques sans ambiguïté par leurs restes dans les divisions par 3, 5 et 7.
Le programme, proposé ici, est un programme codeur qui affiche simplement les restes d'un entier donné N dans les divisions indiquées. Le jeu consiste à essayer de découvrir l'entier (mais, dans cette version, vous n'avez droit qu'à un seul essai).
- LE PROGRAMME** : La ligne 60 permet de confectionner l'entier N.
La boucle I (70-120) calcule les trois restes successifs, et la boucle emboîtée J (90-110) permet de les afficher sous une forme assez visuelle, aux bons emplacements.
La ligne 130 est là pour indiquer de quelle division il s'agit.
Le nombre proposé par le joueur est entré à la ligne 140.
Il est testé dans une variable logique R\$ qui permettra, ensuite, l'affichage de fin de partie (BRAVO ou PERDU).
La ligne 150 réalise l'affichage de fin.
Les lignes 160 et 170 s'occupent de gérer la prochaine partie, selon la réponse du joueur.
- POUR JOUER** : On lance le programme en tapant RUN.
Les trois restes seront alors affichés sur la même ligne, symbolisés par des points (l'absence de points signifiant : reste 0).
A la question VOTRE ESSAI, on répondra en donnant un entier compris entre 0 et 100, suivi de ENTER.
On obtiendra, en réponse, un affichage PERDU ou BRAVO, puis la question AUTRE PARTIE (O/N) ?
Pour faire un nouvel essai, on répondra O, suivi de ENTER.
- EXTENSIONS POSSIBLES** : On peut envisager de réaliser le programme décodeur (on donne les restes et TI-99 devra vous donner N) et chercher si la propriété peut être étendue (en effectuant davantage de divisions, par exemple).

```

10 DIM R$(1)
20 R$(0)="PERDU"
30 R$(1)="BRAVO"
40 RANDOMIZE
50 CALL CLEAR
60 X=INT(100*RND+1)
70 FOR I=1 TO 3
80 A=X-(I+I+1)*INT(X/(I+I+1))
90 FOR J=1 TO A
100 CALL HCHAR(19,2+8*I+J,30)
110 NEXT J
120 NEXT I
130 PRINT "RESTE      PAR 3  P
AR 5  PAR 7": : : : : :
140 INPUT "VOTRE ESSAI :":N
150 PRINT : :R$(-(N=X)): :
160 INPUT "AUTRE PARTIE (O/N
) ? ":T$
170 IF T$="O" THEN 50

```

Permutations

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : moyenne.
- LE JEU** : TI-99 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 30 permet de choisir le nombre de lettres, et la ligne 50 contrôle que l'on n'en demande pas plus de 12.
La ligne 80 réalise le brouillage, en tirant au hasard la nouvelle position de chaque lettre.
Les lettres sont ensuite ajoutées au fur et à mesure de leur tirage, dans une variable B\$ (ligne 100), supprimées dans A\$ et remplacées en C\$, dans le bon ordre (lignes 110 et 120).
La ligne 140 contrôle le nombre de lettres de B\$.
Les affichages sont effectués aux lignes 160 et 170.
Le retournement est fait, lettre par lettre, dans la boucle I (200-220) ; le résultat est placé en B\$, en ligne 230, affiché par la ligne 240 et comparé au résultat final en ligne 250.
Le nombre d'essais sera affiché en fin de partie, grâce à la ligne 260.
- POUR JOUER** : Après avoir tapé RUN, on obtient l'affichage de la question NBRE. DE LETTRES (DE 1 A 12) ; on répond en frappant N, suivi de ENTER.
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 ENTER, on obtiendra AEGBDFC.
On poursuivra jusqu'à l'obtention de l'ordre désiré.
Le score sera affiché en fin de partie, ainsi qu'une question AUTRE PARTIE (O ou N) ; répondre O si l'on désire faire une autre partie.

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

```

10 CALL CLEAR
20 A$="ABCDEFGHJKLM"
30 INPUT "NBRE DE LETTRES (1
  A 12) ":N
40 C$=" "
50 IF N>12 THEN 10
60 B$=""
70 RANDOMIZE
80 X=INT(RND*N+1)
90 IF SEG$(A$,X,1)=" " THEN
80
100 B$=SEG$(A$,X,1)&B$
110 C$=SEG$(C$,1,X-1)&SEG$(A
$,X,1)&SEG$(C$,X+1,12-X)
120 A$=SEG$(A$,1,X-1)&" "&SE
G$(A$,X+1,12-X)
130 S=1
140 IF LEN(B$)-N THEN 80
150 C$=SEG$(C$,1,LEN(B$))
160 PRINT "DEBUT :";B$:"FIN
  :";C$
170 INPUT "PERMUTATION "&STR
$(S)&" :":L
180 S=S+1
190 A$=""
200 FOR I=1 TO L
210 A$=A$&SEG$(B$,L-I+1,1)
220 NEXT I
230 B$=A$&SEG$(B$,L+1,12)
240 PRINT "RESULTAT :";B$ :
  :
250 IF B$<>C$ THEN 170
260 PRINT : : "TROUVE EN";S
  ;"ESSAIS"
270 INPUT "UNE AUTRE PARTIE
(O OU N) ":A$
280 IF A$="O" THEN 10

```

Tour de cartes

- REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : jeu de prestidigitation.
Difficulté du jeu : facile.
- LE JEU** : C'est un tour assez classique mais, ici, les cartes sont manipulées sous vos yeux par le TI-99 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 TI-99 dans quelle colonne (1ère, 2ème ou 3ème à partir de la gauche) se trouve la carte choisie ; les cartes sont alors redistribuées et le joueur indique de nouveau la colonne correspondant à sa carte ; ceci est fait à nouveau encore une fois et, au 4ème affichage, le TI-99 est en mesure de deviner la carte choisie.
- LE PROGRAMME** : La boucle I (60-120) permet de choisir, au hasard, 21 cartes sur 52, et la boucle emboîtée J (80-100) permet de s'assurer que l'on ne réalise pas deux fois le même tirage. La ligne 150 contient les couleurs des cartes, ainsi que les noms des cartes figures (VA, DA, RO, AS) dans une variable alphanumérique B\$.
La boucle I (160-270) réalise l'affichage du jeu.
La ligne 170 décode la couleur de la carte, la ligne 180 choisit la position d'affichage la ligne 190 décode la hauteur.
L'affichage proprement dit est effectué en ligne 260.
La ligne 300 permet, au joueur, d'entrer le numéro de la colonne où se trouve sa carte. La boucle J (350-370) réalise le brouillage, et les boucles emboîtées J (430-470) et I (440-460) 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 trois colonnes décalées.
On indique alors le numéro (1, 2 ou 3) de la colonne où se trouve la carte choisie, suivi de ENTER.
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 la dernière nommée par le TI-99.

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 CALL CLEAR
20 A$=""
30 R=2
40 Q=7
50 RANDOMIZE
60 FOR I=1 TO 21
70 B$=CHR$(38+52*RND)
80 FOR J=1 TO LEN(A$)
90 IF SEG$(A$,J,1)=B$ THEN 7
0
100 NEXT J
110 A$=A$&B$
120 NEXT I
130 H=3
140 P=1
150 B$="TRCACOFIVADAROAS"
160 FOR I=1 TO LEN(A$)*P+11*(1-P)
170 S=INT((ASC(SEG$(A$,I,1))-38)/13)
180 J=Q*INT((I-1)/Q)
190 T=ASC(SEG$(A$,I,1))-37-13*S
200 IF P THEN 220
210 PRINT "VOTRE CARTE EST : ";
220 IF T>Q+R THEN 250
230 P1$=STR$(T+1)
240 GOTO 260

250 P1$=SEG$(B$,ABS(2*T-11),2)
260 PRINT TAB(J*P+18*(1-P));
P1$;TAB((J+3)*P+21*(1-P));SE
G$(B$,R*S+1,R)
270 NEXT I
280 IF P THEN 300
290 STOP
300 INPUT "
    COLONNE?";N
310 CALL CLEAR
320 H=H-1
330 R$="20120"
340 B$=""
350 FOR J=0 TO R
360 B$=B$&SEG$(A$,Q*VAL(SEG$(R$,N+J,1))+1,Q)
370 NEXT J
380 IF H THEN 420
390 A$=B$
400 P=0
410 GOTO 150
420 A$=""
430 FOR J=0 TO R
440 FOR I=1 TO Q
450 A$=A$&SEG$(B$,3*I-R+J,1)
460 NEXT I
470 NEXT J
480 GOTO 150

```

Parcours du cavalier

- REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.
- LE JEU** : Il se présente, ici, comme un jeu de hasard total, mais provient, en réalité, d'un problème assez difficile à résoudre : comment déplacer un cavalier sur un échiquier de façon qu'il passe une fois, et une seule, sur chacune des 64 cases du damier.
Le cavalier se déplace suivant les règles communément admises aux échecs, et sa position de départ est définie par le joueur ; par contre, ses déplacements sont effectués au hasard. Arrivera-t-il à parcourir les 64 cases ?
- LE PROGRAMME** : Les lignes 40 et 50 définissent les divers déplacements possibles, dans deux variables alphanumériques A\$ et B\$; les deux boucles emboîtées I (60-100) et J (70-90) dessinent le terrain de jeu.
On entre les coordonnées de la case de départ (ligne, puis colonne) ligne 110.
La ligne 180 permet le tirage au hasard d'un déplacement du cavalier.
La boucle Z (190-250) cherche si ce déplacement est possible ; si ce n'est pas le cas, on essaie tous les autres (toujours dans le même ordre).
Les lignes 200 et 210 effectuent le déplacement.
La ligne 220 contrôle que l'on ne sorte pas du terrain de jeu.
La ligne 240 envoie en ligne 280 si le coup est possible.
La ligne 260 gère la fin de partie (cavalier bloqué) et affiche le score.
Les lignes 280, 290 et 300 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), puis un numéro de colonne, suivi de ENTER.
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 CLEAR.
- EXTENSIONS POSSIBLES** : On peut imaginer de nombreuses variantes sur ce thème, ne serait-ce qu'en commençant par faire varier la taille des échiquiers ; on peut aussi améliorer considérablement le graphisme, utiliser la couleur. Mais, le plus intéressant, est d'arriver à définir un

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

```

10 CALL CLEAR
20 RANDOMIZE
30 S=1
40 A$="56653223"
50 B$="653223"&A$&"56"
60 FOR I=S TO LEN(A$)
70 FOR J=S TO LEN(A$)
80 CALL HCHAR(2*I+4,3*J+3,46
)
90 NEXT J
100 NEXT I
110 INPUT I,J
120 A$=A$&A$
130 IF S>9 THEN 160
140 CALL HCHAR(2*I+3,3*J+3,4
8+S)
150 GOTO 180
160 CALL HCHAR(2*I+3,3*J+3,4
8+INT(S/10))
170 CALL HCHAR(2*I+3,3*J+4,4
8+S-INT(S/10)*10)
180 Z=INT(8*RND+1)
190 FOR Z=Z TO Z+8
200 A=I+VAL(SEG$(A$,Z,1))-4
210 B=J+VAL(SEG$(B$,Z,1))-4
220 IF (A<1)+(B<1)+(A>8)+(B>
8) THEN 250
230 CALL GCHAR(2*A+3,3*B+3,C
)
240 IF C=46 THEN 280
250 NEXT Z
260 PRINT "MAX :";S
270 STOP
280 I=A
290 J=B
300 S=S+1
310 GOTO 130

```


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, 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 zéro) 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 (70-90) et les lignes 100 à 110 dessinent le terrain de jeu et les briques.
Les lignes 140 à 160 dessinent la raquette dont la position est contrôlée par la ligne 180.
Les lignes 260, 270 et 280 regardent si l'on a touché le mur, les briques ou la raquette.
Les rebonds seront gérés en conséquence avec des éventuels aléas, par les lignes 310 à 370.
La ligne 210 affichera le score en fin de partie.
- POUR JOUER** : Le programme sera lancé par l'instruction RUN, ce qui provoque l'affichage du terrain de jeu puis, très rapidement, l'envoi de la balle.
Le joueur devra alors essayer de déplacer sa raquette à l'aide des touches de déplacement horizontal (S et D).
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 CLEAR.
- EXTENSIONS POSSIBLES** : On peut envisager d'agrandir le terrain de jeu, d'augmenter le nombre de briques, de prévoir plusieurs balles, de dessiner chaque rangée de briques d'une couleur différente, d'introduire des effets sonores.
On peut aussi, en s'inspirant des jeux d'ar-

caides, introduire de petits ouvriers qui viendront reconstruire votre mur de briques au fur et à mesure que vous le démolissez.

```

10 CALL CLEAR
20 RANDOMIZE
30 CALL CHAR(45,"FFFF")
40 S=0
50 Y=14
60 J=14
70 FOR I=2 TO 8
80 CALL HCHAR(I,10,35,12)
90 NEXT I
100 CALL VCHAR(1,9,124,22)
110 CALL VCHAR(1,22,124,22)
120 H=-1
130 K=H
140 CALL HCHAR(23,Y,32)
150 CALL HCHAR(23,Y+4,32)
160 CALL HCHAR(23,Y+1,45,3)
170 CALL KEY(5,G,E)
180 Y=Y+(G=68)*(Y<18)-(G=83)*
(Y>9)
190 CALL HCHAR(I,J,32)
200 IF I<23 THEN 230
210 PRINT,"SCORE :";S
220 END
230 I=I+H
240 J=J+K
250 CALL GCHAR(I,J,C)
260 IF C=32 THEN 290
270 IF C=124 THEN 330
280 IF C=35 THEN 350 ELSE 31
0
290 CALL HCHAR(I,J,48)
300 GOTO 140
310 H=-H
320 K=RND-RND
330 K=-K
340 GOTO 230
350 S=S+1
360 K=-(K+(2*(RND<.5)-1)*RND
/3)
370 H=-H
380 GOTO 140

```


NIVEAU 4

<i>N°</i>	<i>Nom du programme</i>	<i>Thème et particularités</i>
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	Employez les possibilités de traitement de chaîne du TI-99.
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 réflexes.
12	Le jeu du 15	Un jeu de réflexion utilisant des chiffres (2 joueurs).
13	Adresse	Jeu de réflexion pour un joueur, très visuel.
14	Sous-marin	Un classique des jeux vidéo (1 joueur).
15	Zombies	Un jeu farfelu, mais très visuel (1 joueur).
16	Diago	Etes-vous vraiment sûr de vos réflexes ? (1 joueur).
17	Topologie	Une direction à explorer, pour d'autres jeux (1 ou 2 joueurs).
18	Chasse au sous-marin	Il vous faudra pas mal de réflexion et beaucoup de chance.
19	Le lézard d'Alice	Beaucoup de chance, mais très spectaculaire (1 joueur).
20	Calculatrice	Un programme de simulation qui pourra facilement être amélioré.
21	Smokey	Fumez sans risque, avec ce programme original pour 1 joueur.

102 PROGRAMMES POUR TI-99/4A

- | | | |
|----|-------------------------------|---|
| 22 | Course de voitures | Un jeu de réflexes, pas trop difficile, pour un joueur. |
| 23 | Echecs | Un programme très simplifié, pour deux joueurs. |
| 24 | Danger spatial | Serez-vous assez adroit pour évitez toutes les météorites ? |
| 25 | Tic tac toe | Un classique, mais pas sans intérêt (1 joueur). |
| 26 | Gouttes | Plutôt un gag visuel et sonore. A exploiter. |
| 27 | Obstacles | Beaucoup d'obstacles à éviter dans ce jeu d'adresse pour un joueur. |
| 28 | Captures dans l'espace | Votre mission serait facile, sans ces damnées météorites ! |
| 29 | Plusieurs en un | Un terrain de jeu, et des pions, pour deux joueurs ! choisissez ! |

Animation

SAISIE " AU VOL " D'UN CARACTERE FRAPPE AU CLAVIER : CALL KEY

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.

Dans le Basic Texas, cette propriété est offerte par la **fonction KEY** qui effectue une scrutation de toutes les touches du clavier et enregistre le code de la touche frappée.

On pourra utiliser KEY de deux façons :

- soit dans une boucle (réalisation d'un mouvement), on saisit une variable au moyen de KEY et on la teste pour sortir éventuellement de la boucle ;
- soit, au contraire, tant que la variable d'état est nulle (on n'a rien frappé), on attend et l'on sort dès que l'on a frappé quelque chose.

Rappel : La syntaxe de l'instruction KEY est KEY (C, K, S) où C le mode de clavier choisi (de 0 à 5), S la variable d'état qui vaut notamment zéro si aucune touche n'a été pressée et K le code de la variable saisie.

Tirages de dés

- REFERENCES** : Difficulté de programmation : aucune.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : aucune.
- LE JEU** : Le programme proposé ne réalise pas un jeu en lui-même, mais pourra être utilisé en sous-programme pour de nombreux jeux. Il offre simplement la possibilité de lancer de 1 à 9 dés, par simple pression d'une touche. Les dés sont ensuite affichés avec un graphisme très simplifié.
- LE PROGRAMME** : La ligne 50 saisit le caractère frappé au clavier et le transforme en nombre. Ligne 60, un test est prévu pour refaire une saisie si on a frappé une autre que celle d'un chiffre. Une boucle J (100-120) est utilisée pour dessiner un fond gris, et une autre boucle I (90-150) sert au tirage (tirage en ligne 50 et impression en ligne 70). Les lignes 160 et 170 offrent la possibilité de refaire un tirage ; elles ne sont pas indispensables.
- POUR JOUER** : On lance le programme par l'ordre RUN et on obtient l'affichage NOMBRE DE DES A LANCER auquel on répond en frappant un chiffre. Le résultat est alors automatiquement affiché ; il suffit alors de presser une touche quelconque pour recommencer le programme. Pour arrêter, il est nécessaire de taper CLEAR.
- EXTENSIONS POSSIBLES** : On pourra, bien sûr, utiliser le canevas fourni par ce programme pour la réalisation de tout jeu servant de dés (421, YAM'S, CRAPS, etc.).

```

10 CALL CLEAR
20 CALL CHAR(128,"AASSAA55AA
55AA55")
30 PRINT "NOMBRE DE DES A LA
NCER : "
40 CALL SOUND(100,-1,0)
50 CALL KEY(5,K,S)
60 IF (S=0)+(K>58)+(K<49) THE
N 50
70 N=K-48
80 RANDOMIZE
90 FOR I=1 TO N
100 FOR J=1 TO 3
110 CALL HCHAR(8+J,3*I,128,3
)
120 NEXT J
130 D(I)=INT(6*RND+1)
140 CALL HCHAR(10,3*I+1,D(I)
+48)
150 NEXT I
160 CALL KEY(5,K,S)
170 IF S THEN 10 ELSE 160

```


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 30, pour enregistrer les résultats. Le tirage des nombres est effectué dans une boucle I (40-90), une deuxième boucle J (60-80) étant utilisée pour contrôler qu'il n'y a pas deux tirages identiques. Le tirage (ou le retraitage) est effectué en ligne 50. La boucle I (110-130) affiche le résultat, et la ligne 140 affiche le numéro complémentaire. Les lignes 150 à 170 s'occupent du cas où l'on voudrait refaire une partie.

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

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

```

10 CALL CLEAR
20 RANDOMIZE
30 DIM N(7)
40 FOR I=1 TO 7
50 N(I)=INT(49*RND+1)
60 FOR J=1 TO I-1
70 IF N(I)=N(J) THEN 50
80 NEXT J
90 NEXT I
100 PRINT TAB(5);"TIRAGE DU
    LOTO": : : :
110 FOR I=1 TO 6
120 PRINT TAB(4*I);N(I);
130 NEXT I
140 PRINT : : "NUMERO COMPLEM
    ENTAIRE :";N(7): : : "VOULEZ
    VOUS UN AUTRE TIRAGE (O/
    N) "
150 CALL KEY(S,K,S)
160 IF S=0 THEN 150
170 IF K=79 THEN 10

```

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 est dimensionnée à 3 en ligne 40, pour contenir les trois coordonnées définissant la position du joueur.
La position de la chambre contenant le colis piégé est définie par un tirage aléatoire de 3 variables A, B, C, aux lignes 60, 70, 80. Les lignes 110 à 150 dessinent le détecteur. Le signal de proximité est calculé en ligne 100, et la ligne 170 permet l'affichage des coordonnées de la chambre à découvrir, dans le cas où l'on a perdu.
La ligne 120 permet l'affichage, à la fois du signal et du temps restant à courir. La ligne 180 réalise le décompte du temps. La ligne 190 contrôle si l'on a trouvé ou non. La boucle I (230-270) permet d'entrer les nouvelles coordonnées, en utilisant le sous-programme placé aux lignes 290 à 380.
- 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 45, puis ENTER 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'ex-

plosion, d'afficher symboliquement la position dans l'hôtel, etc.

```

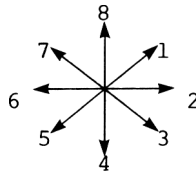
10 CALL CHAR(128,"FFFFFFFF
  FFFFFF")
20 L=100
30 T=L+L
40 DIM R(3)
50 RANDOMIZE
60 A=INT(RND*L)
70 B=INT(RND*L)
80 C=INT(RND*L)
90 CALL CLEAR
100 S=ABS((A/L+B/C*L)-(R(1)/
  L+R(2)+R(3)*L))
110 CALL HCHAR(23,5,128,24)
120 PRINT TAB(7);S;TAB(17);T
130 CALL HCHAR(23,5,128,5)
140 CALL HCHAR(23,17,128,3)
150 CALL HCHAR(23,24,128,5)
160 IF T THEN 180
169 CALL CLEAR
170 PRINT : "PERDU : ";A;B;C
171 STOP
180 T=T-10
190 IF S THEN 210
199 CALL CLEAR
200 PRINT : "BRAVO"
201 STOP
210 PRINT : : : : ">X=?" : : ,
  ">Y=?" : : , ">Z=?" : : : :
220 CALL HCHAR(12,5,128,24)
230 FOR I=1 TO 3
240 R$=""
250 GOSUB 290
260 R(I)=VAL(R$)
270 NEXT I
280 GOTO 90
290 CC=20
300 CALL SOUND(100,700,0)
310 CALL KEY(5, KK, SS)
320 IF (SS=0)+(SS=-1) THEN 31
  0
330 IF KK=13 THEN 380
340 R$=R$&CHR$(KK)
350 CALL HCHAR(14+2*I, CC, KK)
360 CC=CC+1
370 GOTO 310
380 RETURN

```

Ardoise magique

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

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



On dispose aussi de la possibilité de déplacer le point sur l'écran, sans laisser de trace, en appuyant sur la touche 0 (ceci permettra aussi l'effacement partiel), et l'on pourra réinscrire en tapant 9. Pour effacer complètement un dessin, on devra frapper la touche P.

LE PROGRAMME : La ligne 60 réalise l'affichage d'un point sur l'écran.
Les lignes 70 et 80 saisissent au vol la touche frappée (de 0 à 9, ainsi que P).
La ligne 160 appelle l'un des sous-programmes effectuant le déplacement.
Les lignes 170 et 180 contrôlent que l'on ne sorte pas de l'écran.
La ligne 80 permet de tout effacer si l'on appuie sur la touche P.

POUR JOUER : On lancera le programme en tapant RUN, puis on pourra déplacer le point qui apparaît en bas à gauche de l'écran, en utilisant les 8 touches de déplacement.
La touche 0 permet de déplacer le point sans laisser de trace (il y aura effacement si l'on repasse sur un ancien trait).
La touche 9 permet d'inscrire le déplacement.
La touche P efface tout, et l'on recommence.
Pour sortir du programme, il faut taper CLEAR.

EXTENSIONS POSSIBLES : On pourra envisager des possibilités particulières pour tracer des droites, des cercles, dessiner avec plusieurs couleurs, améliorer la finesse.
On pourra ajouter, si besoin est, un programme qui sauve le dessin obtenu pour une utilisation ultérieure.

```

10 CALL CLEAR
20 X=4
30 Y=20
40 CALL CHAR(128,"FFFFFFFFF
FFFFF")
50 S=0
60 CALL HCHAR(Y,X,128)
70 CALL KEY(S,K,E)
80 IF K=80 THEN 10
90 IF (E=0)+(E=-1)+(K<48)+(K
>57) THEN 70
100 IF K=48 THEN 230
110 IF K=57 THEN 50
120 R=K-48
130 F=-2*(R<5)-1
140 A=X
150 B=Y
160 ON R+2*(F-1)GOSUB 250,26
0,280,290,310
170 C=(X>=1)*(X<=32)*(Y>=1)*
(Y<=24)
180 IF C THEN 200
190 GOSUB 310
200 IF S=0 THEN 220
210 CALL HCHAR(B,A,32)
220 GOTO 60
230 S=1
240 GOTO 60
250 Y=Y-F
260 X=X+F
270 RETURN
280 GOSUB 260
290 Y=Y+F
300 RETURN
310 X=A
320 Y=B
330 RETURN

```

Affichage publicitaire

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : affichage publicitaire.
Difficulté du jeu : aucune.
- LE JEU** : Un texte quelconque devra être introduit en début de programme. Il pourra être relativement long (4 ou 5 lignes) ; on choisira ensuite le sens et la vitesse de défilement, à la suite de quoi le texte va apparaître (une seule ligne à la fois) et commencera à défiler, à la vitesse choisie, et dans le sens choisi.
- LE PROGRAMME** : Le texte sera introduit dans une variable P\$, à la ligne 20.
L'affichage sera tronqué à 28 caractères au plus, grâce à la ligne 120.
Le défilement vers la gauche est préparé à la ligne 170, et le défilement vers la droite à la ligne 190.
Le texte sera affiché en ligne 150, et la durée de l'affichage (vitesse de défilement) est choisie par la ligne 200.
La ligne 220 permettra de changer, si besoin est, les conditions du défilement.
- POUR JOUER** : On lancera le programme par RUN, et, à l'affichage de la question VOTRE TEXTE, on introduira le texte à faire défiler, suivi de ENTER. On obtient, ensuite l'affichage SENS D ou G, auquel on répondra en tapant, soit D, soit G suivi de ENTER.
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 ENTER.
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 CLEAR pour interrompre le programme.
- EXTENSIONS POSSIBLES** : On pourra introduire de la couleur, réaliser un affichage ondulant, faire remonter les lignes sur l'écran, etc.

```

10 CALL CLEAR
20 INPUT "ENTREZ VOTRE TEXTE
   ":P$
30 P$=P$&" "
40 INPUT "DEFILEMENT G OU D:
   ":S$
50 IF SEG$(S$,1,1)="G" THEN
80
60 PRINT "DROITE"
70 GOTO 90
80 PRINT "GAUCHE"
90 INPUT "VITESSE DE 0 A 50:
   ":V
100 V=51-V
110 T=LEN(P$)
120 IF T<=28 THEN 150
130 T=28
140 CALL CLEAR
150 PRINT SEG$(P$,1,T)
160 IF SEG$(S$,1,1)="D" THEN
190
170 P$=SEG$(P$,2,LEN(P$)-1)&
SEG$(P$,1,1)
180 GOTO 200
190 P$=SEG$(P$,LEN(P$),1)&SE
G$(P$,1,LEN(P$)-1)
200 FOR P=1 TO V
210 NEXT P
220 CALL KEY(S,K,S)
230 IF K=48 THEN 40 ELSE 140

```


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 TI-99 vous ait auparavant touché par l'un des nombreux obus explosifs qu'il va tirer sur vous.
Vous pouvez vous déplacer dans 4 directions (à l'aide des 4 flèches), et vous devez réussir à sortir du terrain par le bord droit, sans, bien sûr, avoir été touché une seule fois.
- LE PROGRAMME** : Les lignes 30 et 40 définissent la position de départ du joueur.
La ligne 60 choisit la coordonnée horizontale de la case où TI-99 va tirer (liée à votre position), et la ligne 70 choisit la coordonnée verticale (arbitraire) de cette case.
Le déplacement du joueur est contrôlé par les lignes 130 et 140 ; sa trajectoire est tracée par la ligne 110, et sa position actuelle (sur l'écran le caractère 128) est imprimée par la ligne 160.
La ligne 150 regarde s'il est touché ou non.
La ligne 200 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 le caractère 128.
Les 4 touches de direction (5, 6, 7, 8) 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 CLEAR.
- 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 CALL CLEAR
20 CALL CHAR(128,"1818087E08
182424")
30 X=10
40 Y=10
50 RANDOMIZE
60 H=2*(Y-1)+6*RND
70 K=19*RND+3
80 FOR J=1 TO 7*RND+3
90 CALL HCHAR(K+4*RND-2,H+4*
RND-2,30)
100 NEXT J
110 CALL HCHAR(Y,X,46)
120 CALL KEY(S,K,S)
130 X=X+(K=54)-(K=55)
140 Y=Y+(K=56)-(K=53)
150 CALL GCHAR(Y,X,U)
160 CALL HCHAR(Y,X,128)
170 IF U=30 THEN 200
180 PRINT "PERDU"
190 STOP
200 IF Y<31 THEN 60
210 PRINT "BRAVO"

```

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 20 à 50 définissent les caractères utilisés pour le dessin.
Les lignes 60 à 90 déterminent les positions initiales des personnages et leurs chapeaux au départ.
La ligne 190 teste leur rencontre et la ligne 210 saisit "au vol" la réponse d'un des 2 joueurs.
La ligne 220 teste s'il y a eu réponse.
Les lignes 230 à 330 éditent le résultat en fonction de la touche pressée.
Les lignes 350 et 400 constituent le sous-programme d'échange des chapeaux.
Les lignes 410 à 490 dessinent le bonhomme.
- POUR JOUER** : Lancer le programme par RUN.
Les personnages commencent immédiatement à se déplacer l'un vers l'autre.
Le joueur de gauche dispose des touches 1 et 2, et celui de droite, des touches 9 et 0.
S'il y a eu changement de chapeau, les touches à appuyer seront 1 (à gauche) et 0 (à droite).
S'il n'y a pas eu de changement, il faudra, pour gagner, être le premier à appuyer sur 2 (à gauche) et sur 9 (à droite).
L'affichage indiquera alors BRAVO GAUCHE ou BRAVO DROITE ou, encore, PERDU GAUCHE ou PERDU DROITE.
- EXTENSIONS POSSIBLES** : Sur un appareil comme le TI-99, il sera possible de prévoir un affichage couleur, avec une animation des personnages. On pourra aussi compliquer le jeu à plaisir avec plusieurs objets échangés de façon aléatoire (il faudrait alors en trouver la liste complète).

```

10 RANDOMIZE
20 CALL CHAR(128,"FFFFFFFF
FFFFF")
30 CALL CHAR(129,"AA55AA55AA
55AA55")
40 CALL CHAR(130,"000000003C
3CC3C3")
50 CALL CHAR(131,"00000000C3
C33C3C")
60 X1=5
70 C1=130
80 X2=29
90 C2=131
100 CALL CLEAR
110 X=X1
120 C=C1
130 GOSUB 410
140 X=X2
150 C=C2
160 GOSUB 410
170 X1=X1+1
180 X2=X2-1
190 IF X1=X2 THEN 210
200 GOSUB 350
210 CALL KEY(S,K,S)
220 IF S=0 THEN 100
230 IF ((I<>1)+(K<>49))*((I<
>0)+(K<>50)) THEN 260
240 PRINT "BRAVO GAUCHE"
250 END
260 IF ((I<>1)+(K<>48))*((I<
>0)+(K<>57)) THEN 290
270 PRINT "BRAVO DROITE"
280 STOP
290 IF ((I<>1)+(K<>50))*((I<
>0)+(K<>49)) THEN 320
300 PRINT "PERDU GAUCHE"
310 END
320 PRINT "PERDU DROIT"
330 END
340 IF S=0 THEN 100
350 IF RND>.5 THEN 400
360 C=C1
370 C1=C2
380 C2=C
390 I=1
400 RETURN
410 CALL HCHAR(15,X,C)
420 CALL HCHAR(16,X,65)
430 CALL HCHAR(17,X,73)
440 CALL HCHAR(18,X-1,129,3)
450 CALL HCHAR(19,X-1,129,3)
460 CALL HCHAR(20,X-1,128,3)
470 CALL HCHAR(21,X-1,128)
480 CALL HCHAR(21,X+1,128)
490 RETURN

```

Combat aérien

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : réflexe (pour un joueur).
Difficulté du jeu : difficile.
- LE JEU** : Au centre de l'écran, se trouve le viseur de votre appareil, et l'avion ennemi se déplace devant vous, d'une façon qui pourra vous sembler au début incohérente.
Vous pouvez vous déplacer dans toutes les directions, grâce aux flèches de mouvement, et vous allez essayer d'amener l'appareil ennemi au centre de votre viseur.
Vous tirerez sur lui en frappant 0, et vous voyez vos projectiles qui convergent vers 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** : Les lignes 60 à 80 dessinent l'appareil ennemi dont le déplacement aléatoire sera réalisé par les lignes 120 et 130.
Les lignes 150 et 160 vous permettent le pilotage de votre appareil.
Les lignes 180 à 230 dessinent le viseur de votre appareil.
Le sous-programme, gérant le tir, va de la ligne 270 à la ligne 350, et le test de fin de partie est effectué en ligne 340.
La ligne 240 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 5, 6, 7, 8) et vous pouvez tirer à tout moment en pressant la touche 0.
L'appareil ennemi peut continuer à se déplacer de manière aléatoire, même lorsque vous avez tiré.
La partie se terminera avec l'affichage 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 CLEAR.
- EXTENSIONS POSSIBLES** : Il sera assez facile de faire un affichage au graphisme plus soigné, utilisant la couleur. On pourrait aussi envisager de sta-

biliser la position de l'avion ennemi pendant le tir (il suffit de modifier la ligne 160). On peut aussi introduire plusieurs avions, compter les points ou le temps, tenir compte de sa propre réserve de carburant, etc.

```

10 CALL CLEAR
20 A=8
30 B=8
40 S=2
50 N=1
60 CALL HCHAR(A,B,30)
70 CALL HCHAR(A,B+2,30)
80 CALL HCHAR(A,B+1,45)
90 RANDOMIZE
100 U=A
110 V=B
120 A=A+S*INT(S*RND)-1
130 B=B+S*INT(S*RND)-1
140 CALL KEY(S,K,E)
150 A=A+2*(K=54)-(K=55)
160 B=B+2*(K=56)-(K=53)
170 CALL HCHAR(U,V,32,3)
180 CALL HCHAR(11,16,92)
190 CALL HCHAR(11,18,47)
200 CALL HCHAR(12,17,88)
210 CALL HCHAR(12,18,32)
220 CALL HCHAR(13,16,47)
230 CALL HCHAR(13,18,92)
240 IF K<>48 THEN 260
250 GOSUB 270
260 GOTO 60
270 CALL HCHAR(24-2*N,5+2*N,
42)
280 CALL HCHAR(24-2*N,30-2*N
,42)
290 CALL HCHAR(26-2*N,3+2*N,
32)
300 CALL HCHAR(26-2*N,32-2*N
,32)
310 N=N+1
320 IF N>=7 THEN 340
330 RETURN
340 IF ((A=11)+(A=12)+(A=13)
)*(B=16)+(B=17)+(B=18)) THEN
350 ELSE 50
350 PRINT "GAGNE"

```

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 4 dés sur l'écran, en utilisant une technique de dessin un peu particulière. Chaque réponse à la question NBRE. DE DES provoque immédiatement un nouveau lancer du nombre de dés indiqué.
- LE PROGRAMME** : Les lignes 40 à 120 définissent les caractères qui seront utilisés pour les dessins. La ligne 150 permet de contrôler s'il y a eu ou non action sur une touche de 1 à 4. Les lignes 200 et 210 initialisent les deux variables de dessin des dés. La boucle I (220-410) dessinent le cadre du dé. La ligne 290 réalise le tirage, et la boucle J (300-400) dessine les points en relation avec le tirage réalisé (en appelant les variables A\$ et B\$).
- POUR JOUER** : On lance le programme par l'ordre RUN et, à la question NBRE. DE DES, on répond en frappant une touche de 1 à 4. Les dés demandés sont alors représentés, et la question NBRE. DE DES est de nouveau posée. Il suffira de taper CLEAR pour sortir du programme.
- EXTENSIONS POSSIBLES** : On pourra conserver la partie dessin utilisée dans ce programme pour réaliser toute sorte de jeux de dés, avec une représentation graphique de qualité convenable et facile à modifier si besoin était.

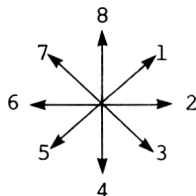
```

10 CALL CLEAR
20 FF$="FFFFFFFF"
30 FO$="FOFOFOF"
40 CALL CHAR(128,FF$&FF$)
50 CALL CHAR(129,"0"&FO$&"0"
&FO$)
60 CALL CHAR(130,FO$&"0"&FO$
)
70 CALL CHAR(131,FF$&FO$)
80 CALL CHAR(132,FO$&"0"&FF$
)
90 CALL CHAR(133,FF$&"0"&FO$
)
100 CALL CHAR(134,"0"&FO$&FF
$)
110 CALL CHAR(135,FF$)
120 CALL CHAR(136,"00000000"
&FF$)
130 PRINT "NOMBRE DE DES ?"
140 CALL KEY(S,K,S)
150 IF (S=0)+(K<49)+(K>52) TH
EN 140
160 CALL CLEAR
170 U=8
180 RANDOMIZE
190 N=K-48
200 A$="          "&CHR$(128)&"
"&CHR$(128)&"          "&CHR
$(128)&CHR$(128)&"          "&C
HR$(128)
210 B$="121314324515525555"
220 FOR I=1 TO N
230 CALL HCHAR(3,U*I-U+2,131
)
240 CALL HCHAR(3,U*I-U+3,135
,5)
250 CALL HCHAR(3,U*I-U+8,133
)
260 CALL HCHAR(9,U*I-U+2,132
)
270 CALL HCHAR(9,U*I-U+3,136
,5)
280 CALL HCHAR(9,U*I-U+8,134
)
290 X=INT(6*RND+1)
300 FOR J=1 TO 5
310 IF J>3 THEN 380
320 S=5*VAL(SEG$(B$,3*(X-1)+
J,1))-4
330 CALL HCHAR(2*J+2,U*I-5,A
SC(SEG$(A$,S,1)))
340 CALL HCHAR(2*J+2,U*I-4,A
SC(SEG$(A$,S+1,1)))
350 CALL HCHAR(2*J+2,U*I-3,A
SC(SEG$(A$,S+2,1)))
360 CALL HCHAR(2*J+2,U*I-2,A
SC(SEG$(A$,S+3,1)))
370 CALL HCHAR(2*J+2,U*I-1,A
SC(SEG$(A$,S+4,1)))
380 CALL HCHAR(J+3,U*I-U+2,1
30)
390 CALL HCHAR(J+3,U*I,129)
400 NEXT J
410 NEXT I
420 GOTO 130

```


Le jeu du missile

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : contre le TI-99.
Difficulté du jeu : moyenne.
- LE JEU** : Le TI-99 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) et de revenir sain et sauf sur votre base ; mais, dès que le TI-99 vous aura repéré, il va envoyer, à votre poursuite, un missile très perfectionné qui n'aura de cesse de vous détruire. Votre avion peut se déplacer dans toutes les directions, grâce aux touches de 1 à 8 ; réussirez-vous à remplir votre mission ?
- LE PROGRAMME** : La ligne 50 dessine la base ennemie à survoler (sous forme d'un H).
La ligne 60 dessine votre base de départ.
Les lignes 100 et 120 choisissent l'instant de départ du missile.
La ligne 140 dessine votre avion.
Les lignes 170 et 180 définissent son mouvement suivant 8 directions, en utilisant les sous-programmes en lignes 300, 310, 330 et 340.
La ligne 190 assure le départ du missile qui est dessiné par la ligne 200 et dont le déplacement est géré par les lignes 210 et 220.
Les lignes 230 et 260 contrôlent la fin de partie.
- POUR JOUER** : Lancer le programme par RUN.
Le terrain de jeu s'affiche et vous pouvez déplacer votre appareil en utilisant les touches de 1 à 8, avec la convention suivante :



Vous devez lui faire contourner le territoire ennemi (si vous, ou le missile, touchez ce territoire, une deuxième partie commence). Si le missile vous détruit, vous obtenez un message TOUCHE, et la partie s'arrête. Tapez CLEAR 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 CALL CHAR(128,"FFFFFFFFF
   FFFFFF")
20 CALL CLEAR
30 A=2
40 B=5
50 CALL HCHAR(B,20,72)
60 CALL HCHAR(15,3,128,3)
70 U=15
80 X=14
90 RANDOMIZE
100 S=INT(RND*U)+1
110 Y=1
120 S=S-1
130 V=29
140 CALL HCHAR(X,Y+3,43)
150 CALL KEY(5,K,E)
160 IF E=0 THEN 150
170 F=-A*(K-48<B)-1
180 ON K-48+A*(F-1)GOSUB 300
   ,310,330,340
190 IF S THEN 120
200 CALL HCHAR(U,V+3,60)
210 U=U-SGN(U-X)
220 V=V-SGN(V-Y)
230 IF (U-X<>0)+(V-Y<>0) THEN
   260
240 PRINT "TOUCHE...."
250 STOP
260 IF (U=B)*(V=17) THEN 20
270 IF (X-B<>0)+(Y-17<>0) THE
   N 140
280 PRINT "MISSION ACCOMPLIE
   ..."
290 STOP
300 X=X-F
310 Y=Y+F
320 RETURN
330 GOSUB 310
340 X=X+F
350 RETURN

```

Le singe

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : difficile.
- LE JEU** : Il s'inspire de l'idée déjà ancienne du singe qui, frappant au hasard sur une machine à écrire, a des chances non nulles de finir par écrire un texte signifiant.
Le rôle du singe est, ici, tenu par le TI-99 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 40 contient une liste V\$ des voyelles courantes, et la ligne 50, 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 60 définit la longueur du mot (en principe, de 3 à 6 lettres, mais on peut également modifier cette longueur).
La boucle I (80-150) permet la confection d'un "mot" en prenant une lettre, soit dans l'alphabet complet, soit dans la liste C\$ (ligne 100), soit dans la liste V\$ (ligne 120).
La ligne 170 temporise l'apparition du résultat.
La ligne 200 gère la fin de partie, avec affichage du score.
La ligne 230 regarde si l'un des joueurs a réagi.
Enfin, les lignes 250 et 260 comptabilisent les points de chaque joueur.
- POUR JOUER** : On lancera le programme en tapant RUN.
La première série de lettres est alors affichée sur l'écran. Le joueur de gauche devra taper 1 s'il voit qu'il peut réaliser un mot, celui de droite devra taper 0.
Si rien n'a été frappé, une autre série de lettres sera affichée après très peu de temps. Si l'un des joueurs a frappé 0 ou 1, la question J ou F est alors affichée. On répond J si le mot est correct, F si le mot

est incorrect, suivi de ENTER. Les points sont alors calculés par TI-99 et la série de lettres suivantes est alors affichée. Pour sortir du programme, il faudra taper P au lieu de 0 ou 1 : le score final est alors affiché.

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

```

10 CALL CLEAR
20 A=0
30 B=A
40 V$="AEIOU"
50 C$="BCDGLMNPSTV"
60 X=INT(4*RND+3)
70 M$=""
80 FOR I=1 TO X
90 E$=CHR$(INT(26*RND+65))
100 IF RND>.4 THEN 120
110 E$=SEG$(C$,INT(LEN(C$)*RND)+1,1)
120 IF RND>.5 THEN 140
130 E$=SEG$(V$,INT(LEN(V$)*RND)+1,1)
140 M$=M$&E$
150 NEXT I
160 PRINT M$
170 FOR P=1 TO 100
180 NEXT P
190 CALL KEY(S,K,S)
200 IF K<>80 THEN 230
210 PRINT : "SCORE A";A,"SCORE B";B
220 STOP
230 IF (K<>48)*(K<>49) THEN 60
240 INPUT "(J)USTE OU (F)AUX ?":R$
250 A=A+(R$="J")*(K=49)-(K=49)*(R$="F")
260 B=B+(R$="J")*(K=48)-(K=48)*(R$="F")
270 GOTO 60

```

Le jeu du 15

- REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (deux 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 (110-160) dessine le terrain de jeu, et la ligne 190 s'occupe de la gestion du tour des joueurs (numéro 1 ou 2 pour chaque joueur, et jeton O ou X correspondant). La ligne 240 attend le coup joué.
La boucle I (250-280) vérifie la validité du coup.
La ligne 300 teste la fin de partie, en cas de match nul, et la ligne 320 refuse les coups non conformes (avec retour à la situation initiale, ligne 180).
Les lignes 340 et 380 enregistrent les coups respectifs de chaque joueur dans des variables alphanumériques O\$ et X\$.
Les lignes 350 et 370 s'occupent de l'affichage du coup, et la ligne 390 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 400 à la ligne 530 : 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 540.
- 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 FINI.
On fera une autre partie en tapant RUN.

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

```

10 CALL CLEAR
20 F1$="FFFFFFF"
30 F2$="F0F0F0F"
40 CALL CHAR(128,F2$&"0"&F1$)
50 CALL CHAR(129,"00000000"&F1$)
60 CALL CHAR(130,"0"&F2$&F1$)
70 F=0
80 R$="123456789"
90 O$=""
100 X$=""
110 FOR I=F TO 24 STEP 3
120 CALL HCHAR(15,I+3,128)
130 CALL HCHAR(15,I+4,129)
140 CALL HCHAR(15,I+5,130)
150 CALL HCHAR(16,I+4,49+I/3)
160 NEXT I
170 PRINT TAB(8);"JOUEUR  :"
180 CALL HCHAR(18,17,49+F)
190 IF F THEN 220
200 CALL HCHAR(18,20,88)
210 GOTO 230
220 CALL HCHAR(18,20,48)
230 CALL KEY(5,K,S)
240 IF (S=0)+(K<49)+(K>57) THEN 230
250 FOR I=1 TO LEN(R$)
260 IF CHR$(K)<>SEG$(R$,I,1) THEN 280
270 GOSUB 330
280 NEXT I
290 IF VAL(R$) THEN 320
300 PRINT "FINI"
310 STOP
320 GOTO 180
330 IF F THEN 370
340 X$=X$&CHR$(K)
350 CALL HCHAR(8,3*I+1,88)
360 GOTO 390
370 CALL HCHAR(8,3*I+1,48)
380 O$=O$&CHR$(K)
390 R$=SEG$(R$,1,I-1)&"0"&SEG$(R$,I+1,LEN(R$)-I)
400 IF F THEN 430
410 T$=X$
420 GOTO 440
430 T$=O$
440 IF LEN(T$)<3 THEN 540
450 FOR II=1 TO 3
460 FOR J=II+1 TO LEN(T$)
470 FOR K=J+1 TO LEN(T$)
480 IF VAL(SEG$(T$,II,1))+VAL(SEG$(T$,J,1))+VAL(SEG$(T$,K,1))<>15 THEN 510
490 PRINT "BRAVO !!"
500 STOP
510 NEXT K
520 NEXT J
530 NEXT II
540 F=ABS(F-1)
550 RETURN

```

Adresse

- REFERENCES** : Difficulté de programmation : difficile.
Catégorie de 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 100 et 110 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 TI-99 est choisi aux lignes 130 et 140, et le chiffre est tiré en ligne 150.
La ligne 160 réalise l'impression, et la ligne 170 détermine la durée d'affichage.
La ligne 210 définit la durée du jeu, et la fin de partie est traitée par la ligne 230.
La boucle I (290-330) réalise le dessin grand format du chiffre frappé par le joueur.
Le contrôle de l'exactitude du résultat est effectué par la ligne 340.
- POUR JOUER** : Le programme sera lancé par RUN, puis un premier chiffre apparaît, après une courte pause vers le milieu de l'écran. Le joueur devra réagir suffisamment vite, tant qu'il reste à l'affichage et le retaper.
Le chiffre qu'il a tapé se dessinera alors en haut 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 CLEAR.
- EXTENSIONS POSSIBLES** : On peut envisager de modifier le jeu pour faire participer plusieurs joueurs, en modifiant éventuellement l'affichage et en codant les touches du clavier mises à la disposition de chacun des joueurs.

On pourra aussi enlever des points à chaque résultat faux, modifier les règles de calcul du score et la durée du jeu, etc.

```

10 M=10
20 T=2*M
30 CALL CLEAR
40 FOR I=1 TO 200
50 NEXT I
60 S=0
70 CALL CHAR(128,"FFFFFFFF
FFFFFF")
80 CALL CHAR(129,"FOFOFOFOFO
FOFOFO")
90 CALL CHAR(130,"OFOFOFOFOF
OFOFOF")
100 C#=CHR$(128)&CHR$(128)&C
HR$(129)&" "&CHR$(130)&CHR$
(129)&CHR$(130)
110 B#="14441333331312113131
4413312131121411333314141141
31"
120 RANDOMIZE
130 X=INT(20*RND)+3
140 Y=INT(28*RND)+4
150 N=INT(M*RND)
160 CALL HCHAR(X,Y,48+N)
170 FOR P=1 TO 200+200*RND
180 NEXT P
190 CALL KEY(S,K,E)
200 IF E THEN 280
210 T=T-N
220 CALL HCHAR(X,Y,32)
230 IF T>=0 THEN 130
240 CALL CLEAR
250 PRINT "SCORE";S;"/100"
260 CALL KEY(S,K,E)
270 IF E THEN 10 ELSE 260
280 U=K-48
290 FOR I=1 TO 5
300 A=2*VAL(SEG$(B#,5*U+I,1)
)-1
310 CALL HCHAR(I,3,ASC(SEG$(
C#,A,1)))
320 CALL HCHAR(I,4,ASC(SEG$(
C#,A+1,1)))
330 NEXT I
340 IF U=N THEN 360
350 S=S+M
360 GOTO 210

```


Sous-marin

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : réflexes (un seul joueur).
Difficulté du jeu : facile.
- LE JEU** : Vous manœuvrez en surface un chasseur de sous-marins, et des sous-marins passent très régulièrement en dessous de votre unité, à différentes profondeurs.
Vous pouvez, à tout moment, larguer des grenades sous-marines et vous pouvez même envisager de toucher plusieurs fois le même sous-marin.
Votre score sera affiché en permanence, sur la gauche de l'écran (rapport nombre de coups au but sur nombre de sous-marins apparus).
- LE PROGRAMME** : La ligne 120 permet au TI-99 de choisir la profondeur du sous-marin.
Les lignes 160 à 210 dessinent votre bâtiment.
La ligne 130 affiche le score et la ligne 220 dessine la surface de l'eau.
Les lignes 230 à 260 dessinent le sous-marin.
Le tir est effectué par un programme spécifique placé aux lignes 400 à 440.
Le déplacement (très rapide) de votre navire est réalisé par la ligne 380.
Le test, permettant de savoir si l'on a touché ou non, est effectué en ligne 290.
- POUR JOUER** : Taper RUN pour commencer la partie.
Le bâtiment sera déplacé grâce aux flèches droite et gauche (S et D).
La touche F 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 CLEAR.
- EXTENSIONS POSSIBLES** : Il sera assez élémentaire d'améliorer la qualité graphique du jeu, d'utiliser la couleur et les sons, de faire apparaître plusieurs sous-marins se déplaçant à des vitesses différentes, éventuellement même, de permettre aux sous-marins de tirer...

```

10 RANDOMIZE
20 F$="FFFFFFF"
30 O$="00000000"
40 CALL CHAR(128,F$&F$)
50 CALL CHAR(129,O$&F$)
60 CALL CHAR(130,F$)
70 C=15
80 S=3
90 P=0
100 CALL CLEAR
110 T=28
120 S=INT(6*RND+7)
130 PRINT "SCORE";P
140 P=P+.01
150 X=1
160 CALL HCHAR(4,C+3,32,3)
170 CALL HCHAR(4,C+6,129,2)
180 CALL HCHAR(4,C+8,128)
190 CALL HCHAR(4,C+9,129)
200 CALL HCHAR(4,C+10,47)
210 CALL HCHAR(4,C+11,32,3)
220 CALL HCHAR(5,3,130,29)
230 CALL HCHAR(S,T,129)
240 CALL HCHAR(S,T+1,128)
250 CALL HCHAR(S,T+2,129)
260 CALL HCHAR(S,T+3,32)
270 IF X=1 THEN 290
280 Y=C+X+X
290 IF (S<>X)+(T+2<>Y) THEN 3
20
300 P=P+1
310 CALL HCHAR(S,T,42,3)
320 T=T-1
330 IF P=99 THEN 360
340 PRINT "C'EST FINI.."
350 STOP
360 IF T<3 THEN 100
370 CALL KEY(S,K,E)
380 C=C+3*((K=68)*(C<21)-(K=
83)*(C>1))
390 IF (K<>70)*(X=1) THEN 160
400 CALL HCHAR(X,Y,32)
410 X=X+1
420 IF X>S THEN 150
430 CALL HCHAR(X,Y,48)
440 GOTO 160

```

Zombies

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

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

LE PROGRAMME : La position des ZOMBIES est définie par un tableau Z.
Les lignes 50 à 140 (boucle I) dessinent le cadre, les ZOMBIES et les trous noirs.
La ligne 230 définit la position initiale du joueur.
Une deuxième boucle I (170-220) réalise le déplacement des Z.
Les lignes 280 et 290 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 CLEAR.

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

```

10 CALL CLEAR
20 RANDOMIZE
30 CALL CHAR(128,"FFFFFFFFF
FFFFF")
40 L=3
50 FOR I=L TO 21
60 CALL HCHAR(L,I+1,128)
70 CALL HCHAR(I,L,128)
80 CALL HCHAR(I,23,128)
90 IF RND>.5 THEN 110
100 CALL HCHAR(3+INT(19*RND)
,I,128)
110 S=INT(14*RND)+4
120 Z((I-1)/2)=S
130 CALL HCHAR(21,I+1,128)
140 NEXT I
150 K=INT(18*RND)+4
160 L=2
170 FOR I=1 TO 4
180 CALL HCHAR(3+Z(I),Z(I+4)
,32)
190 Z(I)=Z(I)-SGN(Z(I)+3-S)
200 Z(I+4)=Z(I+4)-SGN(Z(I+4)
-K)
210 CALL HCHAR(3+Z(I),Z(I+4)
,90)
220 NEXT I
230 CALL HCHAR(S,K,48)
240 FOR PAUSE=1 TO 250
250 NEXT PAUSE
260 CALL HCHAR(S,K,32)
270 CALL KEY(S,IK,E)
280 K=K+L*(IK=83)-L*(IK=68)
290 S=S+L*(IK=69)-L*(IK=88)
300 GOTO 170

```

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 une touche quelconque pour arrêter l'affichage des points dès que vous estimez que le nombre de points demandé au début est apparu à l'écran. On obtient alors un affichage BRAVO ou PERDU, selon le résultat obtenu.
- LE PROGRAMME** : Le nombre de points noirs sera choisi au hasard, en ligne 30.
Le temps est décompté par les lignes 90 à 130.
Les points sont imprimés par la ligne 170.
La pression d'une touche est détectée en ligne 210 et la ligne 220 contrôle la fin de partie (aucun réflexe).
Les affichages de fin sont exécutés par les lignes 240 à 290.
- POUR JOUER** : On lancera l'exécution par RUN.
On obtient alors l'affichage de la phrase LE BON REFLEXE EST A ... qui indique le nombre de points gagnants.
Le temps sera ensuite décompté sous la forme ATTENTION ... SECONDES qui indique le temps restant avant l'apparition de la diagonale.
A l'affichage de 0, la diagonale apparaît et commence à se déplacer sur l'écran.
Si l'on estime que le nombre de points réclamés a été atteint, on devra presser la touche 0.
L'affichage indique alors BRAVO si l'on a gagné, ou PERDU dans le cas contraire ; puis la phrase AUTRE ESSAI (O ou N) ? est affichée, répondre O pour faire une autre partie.
- EXTENSIONS POSSIBLES** : On pourra envisager une adaptation du jeu à deux joueurs ou plus, en laissant la diagonale se former complètement et en testant la coïncidence lors de la pression de la touche affectée au joueur.
On peut aussi réaliser un graphisme plus soigné et modifier le principe du déplacement des objets sur l'écran (déplacement horizontal, vertical ou apparition au hasard...).

```

10 CALL CLEAR
20 RANDOMIZE
30 X=INT(3+18*RND)
40 PRINT "LE BON REFLEXE EST
  A";X: : : :
50 FOR P=1 TO 300
60 NEXT P
70 S=10
80 PRINT "ATTENTION : SECO
  NDES..": : : :
90 S=S-1
100 CALL HCHAR(19,15,48+S)
110 FOR P=1 TO 100
120 NEXT P
130 IF S THEN 90
140 CALL CLEAR
150 U=24
160 V=32
170 CALL HCHAR(U,V,42)
180 U=U-1
190 V=V-1
200 CALL KEY(S,K,S)
210 IF S THEN 230
220 IF 24-U-X THEN 170 ELSE
280
230 IF 24-U-X THEN 280
240 PRINT : :,"BRAVO"
250 INPUT "AUTRE ESSAI (O/N)
  ? ":R$
260 IF R$="O" THEN 10
270 STOP
280 PRINT :,"PERDU"
290 GOTO 250

```

Topologie

REFERENCES

: Difficulté de programmation : moyenne.
Catégorie de jeu : graphisme.
Difficulté du jeu : moyenne.

LE JEU

: Un dessin, très élémentaire, à base de traits apparaît sur l'écran (il peut aussi être entré par un partenaire).
Le joueur dispose d'un plot mobile symbolisant une gomme et devra essayer d'effacer le dessin précédent en repassant, une fois et une seule, sur chacun des traits.

LE PROGRAMME

: La ligne 30 contient un dessin préprogrammé (variable B\$). On pourra, bien entendu, soit définir un autre dessin, soit entrer le dessin en cours de jeu en tapant 30 INPUT B\$. Le codage du dessin est réalisé de la façon suivante :

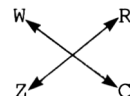
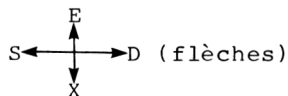
Début
→ x x || x x || x x | x x || x x | x x | ... etc.
 x y └─ Déplacement en y (0,1 ou 2)
 └─ Déplacement en x (0, 1 ou 2)

Position de départ

0	↑	0	←
x : 1	immobile	y : 1	immobile
2	↓	2	→

Les lignes 40 et 50 calculent la position de départ.

La boucle I (60-100) assure le décodage, l'impression étant assurée par le sous-programme 200-250. La partie effacement commence en ligne 110. Le décodage du déplacement du joueur est assuré par les lignes 180 et 170 (8 directions).



POUR JOUER

: On commence avec l'instruction RUN et, si le dessin n'a pas été programmé, on le rentre en suivant les instructions données précédemment. On tape ensuite ENTER.
A l'affichage du dessin, un point lumineux clignotant apparaît en bas à gauche de l'écran. On peut le déplacer en utilisant les 4 touches de déplacement d'une part, et en utilisant les touches pour les déplacements en diagonale d'autre part.
On devra essayer d'effacer le dessin initial,

sans lever le crayon et sans repasser deux fois sur le même trait.

EXTENSIONS POSSIBLES : On pourra introduire la couleur, définir une table de dessins topologiquement intéressants, ajouter des contrôles sur le fait que l'on ne quitte pas le dessin initial, etc.

```

10 CALL CLEAR
20 CALL CHAR(128,"FFFFFFFFF
FFFFF")
30 B$="020422112011122800110
2111028"
40 S=VAL(SEG$(B$,1,2))
50 T=VAL(SEG$(B$,3,2))
60 FOR I=5 TO LEN(B$)STEP 4
70 U=VAL(SEG$(B$,I,1))-1
80 V=VAL(SEG$(B$,I+1,1))-1
90 GOSUB 200
100 NEXT I
110 X=1
120 Y=3
130 CALL HCHAR(X,Y,30)
140 CALL HCHAR(X,Y,32)
150 CALL KEY(S,K,E)
160 IF E=0 THEN 130
170 Y=Y+(K=87)+(K=90)+(K=83)
  -(K=68)-(K=67)-(K=82)
180 X=X+(K=69)+(K=82)+(K=87)
  -(K=67)-(K=88)-(K=90)
190 GOTO 130
200 FOR K=1 TO VAL(SEG$(B$,I
+2,2))
210 CALL HCHAR(S,T,128)
220 S=S+U
230 T=T+V
240 NEXT K
250 RETURN

```


Chasse au sous-marin

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : moyenne.
- LE JEU** : Vous êtes à bord d'un bâtiment de surface, charger 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 (70-100) affiche le terrain. Les lignes 150 et 160 définissent la position initiale du sous-marin ; les lignes 170 et 180 permettent son déplacement, contrôlé par la ligne 190 (on reste dans les limites du terrain). Les lignes 280 et 290 contrôlent votre déplacement. La distance est calculée en ligne 300, affichée en ligne 310 et testée en ligne 320. Le test de fin de partie est effectué à la ligne 410.
- 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 S, D, E, X. 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 CLEAR.
- 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 CALL CLEAR
20 A=0
30 INPUT "DIMENSION (max 9):
":H
40 CALL CLEAR
50 PRINT "DISTANCE"
60 G=2
70 FOR I=0 TO H
80 CALL HCHAR(I+9,5,48+I)
90 CALL HCHAR(7,8+I*2,48+I)
100 NEXT I
110 I=1
120 X=7
130 Y=2
140 RANDOMIZE
150 V=INT(H*RND)+7
160 U=INT(H*RND)+2
170 U=U+I
180 V=V+I
190 IF (U>=2)*(U<=H+2)*(V>=7)
)*(V<=H+7) THEN 210
200 I=-I
210 CALL HCHAR(X+G,2+G*Y,32,
2)
220 CALL HCHAR(X+G,4+G*Y,30)
230 CALL HCHAR(X+G,5+G*Y,32,
2)
240 CALL HCHAR(X+1,4+G*Y,32)
250 CALL HCHAR(X+3,4+G*Y,32)
260 CALL KEY(5,K,E)
270 IF E=0 THEN 260
280 Y=Y+(K=68)*(Y<H+2)-(K=83)
)*(Y>2)
290 X=X+(K=88)*(X<H+7)-(K=69)
)*(X>7)
300 D=INT(SQR((X-V)^2+(Y-U)^
2))
310 CALL HCHAR(23,12,48+D)
320 IF D=0 THEN 340
330 IF D<3 THEN 170 ELSE 210
340 CALL HCHAR(23,22,80)
350 CALL HCHAR(23,24,63)
360 CALL SOUND(100,700,0)
370 CALL KEY(5,K,E)
380 IF E=0 THEN 370
390 CALL HCHAR(23,22,32,3)
400 D=K-48
410 IF D=INT(3*RND)+2 THEN 1
50
420 PRINT "BRAVO !!"
430 STOP

```

Le lézard d'Alice

- REFERENCES** : Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez difficile.
- LE JEU** : Il s'inspire très vaguement d'une idée de LEWIS CAROLL, dans "Alice aux Pays des Merveilles".
Dans l'histoire originale, Alice grandissait ou diminuait, alors que dans le jeu c'est un lézard qui change de taille.
Deux verres apparaissent sur l'écran : l'un contient le liquide qui fait grandir, l'autre le liquide qui fait rapetisser (malheureusement, ce n'est pas toujours le même verre).
Le joueur devra indiquer la quantité de liquide qu'il désire boire et le lézard changera de taille en fonction de cette quantité.
Le but du jeu est de parvenir à lui rendre sa taille initiale.
- LE PROGRAMME** : Les lignes 100 à 150 dessinent les verres.
La boucle I (210-230) les remplit avec la quantité indiquée en ligne 180.
La variation de taille du lézard est définie par la ligne 240, avec contrôle par les lignes 250 et 270.
La ligne 290 dessine le corps du lézard, les lignes 300 et 320 dessinent l'oeil et les pattes.
Un message gagnant sera affiché par la ligne 360 si l'on a trouvé la bonne taille.
- POUR JOUER** : On lance le programme par RUN : les deux verres se dessinent. A la question VERRE ? on répond en tapant G ou D suivi de ENTER.
A la question COMBIEN ? on entre alors une quantité (de 0 à 9) suivi de ENTER.
Le verre désigné se remplit sous vos yeux puis, après quelques instants, le lézard est affiché.
On devra alors frapper une touche quelconque (sauf CLEAR qui arrêterait la partie) ; si le lézard n'avait pas la bonne taille, aucun message n'est affiché et l'on revient au dessin des verres.
- EXTENSIONS POSSIBLES** : Puisqu'il s'agit d'un jeu essentiellement graphique, il sera intéressant d'améliorer les dessins, d'introduire la couleur, de modifier le caractère aléatoire du choix des verres, etc.
Pourquoi ne pas relire Alice aux Pays des Merveilles, les idées n'y manquent pas.

```

10 F$="FFFFFFF"
20 CALL CHAR(128,F$&F$)
30 CALL CHAR(136,F$&F$)
40 CALL COLOR(14,5,1)
50 CALL CHAR(129,F$)
60 RANDOMIZE
70 CALL SCREEN(15)
80 N=INT(15*RND)+1
90 CALL CLEAR
100 CALL VCHAR(11,23,128,9)
110 CALL VCHAR(11,9,128,9)
120 CALL VCHAR(11,13,128,9)
130 CALL VCHAR(11,19,128,9)
140 CALL HCHAR(20,9,129,5)
150 CALL HCHAR(20,19,129,5)
160 PRINT TAB(7);"GAUCHE";TAB
B(17);"DROITE": : : :
170 INPUT "   VERRE   ":V$
180 INPUT "   COMBIEN   ":Q
190 PRINT : : : :
200 C=-10*(V$="G")-20*(V$="D")
210 FOR I=10-Q TO 9
220 CALL HCHAR(I,C,136,3)
230 NEXT I
240 N=N+(RND>.5)*Q
250 IF N<=30 THEN 270
260 N=30
270 IF N>=1 THEN 290
280 N=1
290 CALL HCHAR(22,17-N/2,129,N)
300 CALL HCHAR(23,17-N/3,94)
310 CALL HCHAR(23,17+N/3,94)
320 CALL HCHAR(21,16+N/2,64)
330 CALL KEY(5,K,E)
340 IF E=0 THEN 330
350 IF N=10 THEN 90
360 PRINT ,"BRAVO"

```

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 100 et 110 contiennent les informations codées qui permettront le dessin des chiffres en grand format.
Les lignes 180 à 270 testent les différentes entrées possibles.
Les lignes 280 à 310 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 320 à 350.
Le nombre total de symboles entrés est contrôlé par la ligne 360.
Le résultat est inscrit grâce à un programme fonctionnant en automatique qui se trouve aux lignes 380 et 450 (l'automatisme est géré par la ligne 370, dans le programme principal).
- POUR JOUER** : On devra lancer le programme par RUN.
Puis, on tape les chiffres du premier nombre à entrer qui vont s'afficher un par un, au fur et à mesure (on ne pourra entrer plus de 7 symboles, y compris le signe - 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 CLEAR, puis RUN.
Taper CLEAR pour sortir du programme.
- EXTENSIONS POSSIBLES** : On devra, tout d'abord, améliorer l'affichage du signe - (représenté, ici, par un seul caractère) puis on pourra améliorer le dessin des chiffres ; on peut aussi envisager une extension à 8 chiffres, autoriser des opérations plus complexes (COS, SIN,...), ajouter des mémoires, etc).

```

10 CALL CLEAR
20 U=2
30 F$="FFFFFFF"
40 D$="F0F0F0F0F0F0F0F"
50 CALL CHAR(128,F$&F$)
60 CALL CHAR(129,D$)
70 CALL CHAR(130,"0"&D$)
80 CALL CHAR(131,"00000000"&
F$)
90 CALL CHAR(132,"00000000F0
F0F0F0")
100 C$=CHR$(128)&CHR$(128)&C
HR$(129)&" "&CHR$(130)&CHR$
(129)&CHR$(130)
110 B$="14441333331312113131
4413312131121411333314141141
31"
120 R$=""
130 T=0
140 N=0
150 H=0
160 CALL KEY(S,K,S)
170 IF S=0 THEN 160
180 IF K=61 THEN 380
190 R$=R$&CHR$(K)
200 IF R$<>"-" THEN 230
210 P$=R$
220 GOTO 320
230 IF (K<46)+(K>57)+(K=47) T
HEN 140
240 P$=CHR$(K)
250 IF N THEN 270
260 CALL CLEAR
270 IF (P$=".")+(P$="-") THEN
320
280 FOR I=1 TO 5
290 S=U*VAL(SEG$(B$,I+5*VAL(
P$),1))-1
300 CALL HCHAR(I,N+3,ASC(SEG
$(C$,S,1)))
301 CALL HCHAR(I,N+4,ASC(SEG
$(C$,S+1,1)))
310 NEXT I
320 IF P$<>"-" THEN 340
330 CALL HCHAR(3,5,131)
340 IF P$<> "." THEN 360
350 CALL HCHAR(3,N+3,132)
360 N=N-4*(N<24)
370 IF T=0 THEN 160 ELSE 440
380 N=0
400 H=1-H
410 FOR T=1 TO LEN(R$)
420 P$=SEG$(R$,T,1)
430 GOTO 250
440 NEXT T
450 GOTO 160

```

Smokey

- REFERENCES** : Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire
Difficulté du jeu : moyenne.
- LE JEU** : Une colonne de fumée noire apparaît sur l'écran.
Le joueur, représenté par un petit bonhomme devient visible sur la droite. Il doit dans un premier temps se positionner sur la colonne de fumée, puis à partir de là, il va automatiquement avaler la fumée, à condition qu'il réussisse à se maintenir sur sa trace.
- LE PROGRAMME** : La ligne 100 fait apparaître la fumée au bas de l'écran ; la ligne 110 permet de changer (vers la droite ou vers la gauche) la direction de la colonne, et la ligne 130 fait monter la colonne.
Le joueur apparaît grâce à la ligne 150, et son déplacement est effectué par les lignes 210 et 230.
Les lignes 290 à 340 se chargent de l'affichage du résultat.
- POUR JOUER** : On doit lancer l'exécution du programme par RUN.
La fumée commence à s'élever, tandis que le joueur apparu sous forme d'un carré noir dans un coin de l'écran, va se déplacer sur elle à l'aide des 4 flèches. Dès qu'il arrive sur la fumée; il va automatiquement commencer à l'avalier, et devra essayer de suivre la colonne sans dévier (grâce aux flèches S et D).
Il gagne s'il arrive en bas de l'écran avant que la fumée n'arrive en haut.
L'affichage final indique alors BRAVO ou PERDU.
- EXTENSIONS POSSIBLES** : On pourra, tout en conservant l'idée de départ, imposer d'autres règles : par exemple avaler toute la fumée, qui pourra aussi repartir d'un autre endroit si l'on dévie de la colonne initiale.
On pourra aussi améliorer les contrôles de fin de partie, et la qualité du graphisme.

```

10 CALL CLEAR
20 CALL CHAR(128,"18180A7E4B
182424")
30 CALL CHAR(129,"FFFFFFFF
FFFFFF")
40 S=0
50 X=20
60 U=1
70 V=12
80 RANDOMIZE
90 Y=INT(14*RND)+3
100 CALL HCHAR(X,Y,129)
110 Y=Y+(RND<.5)*(Y<17)-(RND
<.5)*(Y>3)
120 T=1
130 X=X-T
140 IF X=1 THEN 290
150 CALL HCHAR(U,V,128)
160 IF T=0 THEN 190
170 FOR PAUSE=1 TO 200
180 NEXT PAUSE
190 CALL HCHAR(U,V,32)
200 CALL KEY(S,K,S)
210 U=U-(K=88)*(T<20)*(X<20)
+(K=69)*(T>0)*(X>1)
220 U=U+1-T
230 V=V+(K=68)*(V<17)-(K=83)
*(V>1)
240 CALL GCHAR(U,V,COD)
250 IF COD=128 THEN 100
260 T=0
270 S=1
280 IF U=20 THEN 150
290 CALL CLEAR
300 IF T THEN 330
310 PRINT "BRAVO"
320 GOTO 340
330 PRINT "PERDU"
340 PRINT "SCORE";S

```


Course de voitures

- REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe (1 joueur).
Difficulté du jeu : moyenne.
- LE JEU** : Vous circulez sur une route très sinueuse, ne comportant aucun obstacle, et vous devez essayer de la suivre le plus longtemps possible. La trace de votre parcours reste affichée en permanence, et votre score est affiché dès que vous avez un accident.
- LE PROGRAMME** : L'impression de la route est assurée par les lignes 100 à 120.
La ligne 130 imprime la voiture.
La ligne 160 contrôle son mouvement, et la ligne 170 détermine si la route fait ou non des virages.
La ligne 200 regarde si l'on a touché le bord de la route.
La ligne 260 incrémente le score, et la ligne 270 efface l'ancienne position de la voiture (une trace claire apparaît à l'emplacement de sa trajectoire).
- POUR JOUER** : Lancer l'exécution du programme par l'ordre RUN, le dessin de la route commence à évoluer, et votre véhicule, représenté par un caractère redéfini apparaît.
Pour le déplacer vers la droite ou vers la gauche, vous devez utiliser les touches S et D (flèches de déplacement).
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, suivi de l'indication sur le score réalisé.
On pourra arrêter le déroulement de la partie à tout moment en pressant la touche CLEAR.
- EXTENSIONS POSSIBLES** : On peut envisager de jouer à plusieurs, chacun son tour, en enregistrant le meilleur score.
On peut aussi envisager d'élargir la piste, et de faire courir plusieurs voitures.
On pourra aussi s'inspirer des jeux d'arcades qui sont particulièrement nombreux sur ce thème.

```

10 CALL CLEAR
20 RANDOMIZE
30 CALL CHAR(128,"D6FEC61010
447C54")
40 A=9
50 B=A
60 X=A+1
70 Y=A-3
80 S=0
90 PRINT
100 CALL HCHAR(X,Y,30,2)
110 CALL HCHAR(X,Y+2,46,2)
120 CALL HCHAR(X,Y+4,30,2)
130 CALL HCHAR(A,B,128)
140 C=B
150 CALL KEY(S,K,E)
160 B=B-(K=68)+(K=83)
170 Y=Y+2*RND*(Y>3)-2*RND*(Y
<16)
180 FOR PAUSE=1 TO 100
190 NEXT PAUSE
200 CALL GCHAR(A+1,B,COD)
210 IF COD=30 THEN 260
220 PRINT "ACCIDENT","SCORE"
;S
230 FOR PAUSE=1 TO 150
240 NEXT PAUSE
250 CALL HCHAR(23,2,32,30)
260 S=S+1
270 CALL HCHAR(A,C,32)
280 GOTO 90

```

Echecs

- REFERENCES** : Difficulté de programmation : difficile.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : difficile.
- LE JEU** : Il n'était pas question en utilisant le Basic de réaliser un programme permettant au TI-99 de jouer aux échecs.
Le programme proposé permettra seulement à 2 joueurs de disputer une partie en se servant de l'écran du téléviseur comme plateau de jeu. Il n'y a pas de contrôle de validité des coups ; mais seulement gestion de la position des pièces sur l'échiquier. Les pièces sont représentées par leur initiale, les pions par des P, les tours par T, les fous par F, les cavaliers par C, les dames par D et les rois par R. La couleur du fond permet de distinguer pièces noires et pièces blanches.
- LE PROGRAMME** : La ligne 10 définit l'échiquier qui est affiché par la boucle I (200-220).
Les lignes 30 à 90 redéfinissent les caractères T, C, F, R, D, P (écrits blanc sur fond noir) dont la couleur sera fixée en ligne 110.
La boucle I (130-150) lit l'échiquier initial stocké en DATA.
La ligne 260 change de joueur. Les modifications des lignes de départ et d'arrivée de l'échiquier sont respectivement effectuées par les lignes 290 et 300.
La ligne 270 détermine si la case de départ est noire ou blanche.
- POUR JOUER** : On lance l'exécution par RUN.
L'échiquier est dessiné, et l'on voit apparaître la question JOUEUR BLANC L,C (DEPART) ; on introduit la ligne, puis la colonne de la pièce à manoeuvrer, sous la forme, par exemple : 2,4 (pion blanc de la ligne 2 et de la colonne 4) suivi de ENTER.
On a ensuite L,C (ARRIV.) : on tape alors la ligne et la colonne de la case d'arrivée, sous la forme, par exemple : 4,4, suivi de ENTER.
- EXTENSIONS POSSIBLES** : On pourra chercher à réaliser un graphisme plus précis (un exemple possible est d'ailleurs fourni dans ce même ouvrage). On pourra aussi sans trop de difficultés introduire le contrôle du mouvement des pièces. Il ne restera alors plus qu'à écrire le programme de jeu du TI-99 ! Bon courage !

```

10 DATA " 12345678","1TCFRDF
CT","2PPPPPPPP","3# # # #",
"4 # # # #","5# # # #",
"6 # # # #","7pppppppp","8tc
frdfct"
20 CALL SCREEN(15)
30 CALL CHAR(116,"007C101010
101010")
40 CALL CHAR(99,"00384440404
04438")
50 CALL CHAR(102,"007C404078
404040")
60 CALL CHAR(114,"0078242424
242478")
70 CALL CHAR(100,"0078444478
504844")
80 CALL CHAR(112,"0078444478
404040")
90 CALL CHAR(35,"FFFFFFFFF
FFFF")
100 FOR I=9 TO 11
110 CALL COLOR(I,15,2)
120 NEXT I
130 FOR I=0 TO 8
140 READ L$(I)
150 NEXT I
160 C$(0)="BLANC"
170 C$(1)="NOIR"
180 H=0
190 CALL CLEAR
200 FOR I=0 TO 8
210 PRINT L$(I)
220 NEXT I
230 PRINT : : "Joueur ";C$(
H)
240 INPUT "L,C (DEPART) :":L
D,CD
250 INPUT "L,C (ARRIV.) :":L
A,CA
260 H=ABS(1-H)
270 COD=-32*((LD+CD)/2<>INT(
(LD+CD)/2))-35*((LD+CD)/2=IN
T((CD+LD)/2))
280 S$=SEG$(L$(LD),CD+1,1)
290 L$(LD)=SEG$(L$(LD),1,CD)
&CHR$(COD)&SEG$(L$(LD),CD+2,
8)
300 L$(LA)=SEG$(L$(LA),1,CA)
&S$&SEG$(L$(LA),CA+2,8)
310 GOTO 190

```

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 la ligne 50 ; elles sont imprimées par la ligne 70.
L'avance des météorites est réalisé par une instruction PRINT ligne 80.
Les lignes 110 et 120 vérifient si vous avez ou non heurté une météorite.
La ligne 150 dessine votre vaisseau spatial et la ligne 170 permet de la déplacer.
Le compteur se trouve en ligne 180.
- 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 (S et D).
On peut arrêter la partie à tout moment en tapant CLEAR. Sinon, le jeu s'arrête lorsque vous heurtez une météorite, et le score est alors affiché.
- EXTENSIONS POSSIBLES** : Il serait intéressant de rendre le jeu de plus en plus difficile en faisant apparaître de plus en plus de météorites au fur et à mesure du déroulement de la partie.
On peut aussi essayer de détruire les météorites en les pulvérisant au fur et à mesure de leur apparition.

```

10 CALL CLEAR
20 RANDOMIZE
30 X=12
40 Z=0
50 H=INT(30*RND)+3
60 IF RND>.7 THEN 90
70 CALL HCHAR(20,H,90)
80 PRINT
90 IF RND>.6 THEN 110
100 CALL HCHAR(20,3+INT(30*RND),90)
110 CALL GCHAR(12,X+1,C1)
120 CALL GCHAR(12,X,C2)
130 IF (C1=90)+(C2=90) THEN 200
140 CALL HCHAR(10,X-3,32,6)
150 CALL HCHAR(11,X,72)
160 CALL KEY(5,K,S)
170 X=X+(K=68)*(X<32)-(K=83)*(X>3)
180 Z=Z+1
190 GOTO 50
200 CALL CLEAR
210 PRINT "SCORE";Z

```

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 Ø ou un X dans une case libre de la grille de jeu, et le premier qui réussit un alignement de 3 symboles identiques a gagné (l'alignement peut être horizontal, vertical, ou suivant l'une des deux diagonales).
- LE PROGRAMME** : Le terrain de jeu est dessiné à l'aide de deux boucles I (5Ø-1ØØ et 11Ø-16Ø).
La ligne 18Ø s'occupe d'afficher le symbole X ou Ø correspondant au joueur qui a le trait.
Un symbole clignotant (ligne 25Ø) est affiché en haut et à gauche du terrain, même si la case est déjà occupée ; les lignes 28Ø et 29Ø permettent de déplacer le symbole sur le terrain de jeu.
La ligne 3ØØ vérifie si la case où l'on veut jouer est libre, et si l'on désire y jouer.
La ligne 31Ø affiche le coup, et la ligne 36Ø 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 S, D, E, X).
Lorsqu'il arrive sur la case où il désire jouer, il suffit qu'il appuie sur la touche Ø ; le coup sera refusé si la case n'est pas libre.
Si la case 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 CLEAR pour sortir du programme.
On pourra ensuite faire une autre partie avec RUN.
- EXTENSIONS POSSIBLES** : On pourrait réaliser après chaque coup un contrôle pour vérifier si l'un des joueurs a gagné. On peut aussi réaliser un jeu machine, mais il faudrait éviter qu'il soit par trop systématique (par exemple, la machine jouant toujours son premier coup sur la case centrale).

```

10 CALL CLEAR
20 CALL CHAR(128,"AASSAASSAA
SSAASS")
30 CALL CHAR(129,"FFFFFFFFF
FFFFFF")
40 PRINT TAB(10);"JOUER": :
:
50 FOR I=5 TO 9
60 CALL HCHAR(6,I,128)
70 CALL HCHAR(8,I,128)
80 CALL HCHAR(1,6,128)
90 CALL HCHAR(1,8,128)
100 NEXT I
110 FOR I=4 TO 10
120 CALL HCHAR(4,I,129)
130 CALL HCHAR(10,I,129)
140 CALL HCHAR(1,4,129)
150 CALL HCHAR(1,10,129)
160 NEXT I
170 K=0
180 IF K THEN 210
190 CALL HCHAR(21,19,88)
200 GOTO 220
210 CALL HCHAR(21,19,48)
220 I=1
230 J=I
240 CALL GCHAR(2*I+3,2*J+3,C
)
250 CALL HCHAR(2*I+3,2*J+3,4
2)
260 CALL HCHAR(2*I+3,2*J+3,C
)
270 CALL KEY(S,V,E)
280 I=I+(V=88)*(I<3)-(V=69)*
(I>1)
290 J=J+(V=68)*(J<3)-(V=83)*
(J>1)
300 IF (C<>32)+(V<>48) THEN 2
40
310 IF K=0 THEN 340
320 CALL HCHAR(2*I+3,2*J+3,4
8)
330 GOTO 350
340 CALL HCHAR(2*I+3,2*J+3,8
8)
350 K=ABS(K-1)
360 GOTO 180

```


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 carré gris, apparaît au-dessus du récipient et tombe en remplissant le récipient au fur et à mesure. Sa trajectoire, lorsqu'elle rencontre un obstacle, est entièrement déterminée par le hasard : elle part à droite ou à gauche, jusqu'à ce qu'il n'y ait plus aucune possibilité. Alors, une autre goutte tombe, et ainsi de suite, jusqu'au débordement final.
- LE PROGRAMME** : Les lignes 90 à 110 dessinent le bassin et la boucle (120-140) les obstacles. La série de gouttes est dessinée en ligne 230.
Le sous-programme (480-580) regarde si la voie est libre ou si un obstacle se trouve sur la trajectoire de la série de gouttes. Les diverses positions de toutes les gouttes composant la série sont stockées dans 2 tableaux L et C périodiquement remis à jour à chaque déplacement.
En changeant la ligne 80 on peut modifier le nombre de gouttes constituant une série.
- POUR JOUER** : Le RUN déclenche le dessin du récipient et des obstacles, une série de gouttes apparaît, se met à tomber et finit par se trouver une place. Une autre série apparaît et l'on recommence.
Le programme s'arrête de lui-même lorsque le récipient déborde ; on peut aussi l'arrêter à tout moment en tapant CLEAR.
- EXTENSIONS POSSIBLES** : On peut, bien entendu, ajouter la couleur et le son et aussi envisager un jeu dans lequel un petit personnage devrait échapper aux gouttes ou à la noyade.

10 CALL CLEAR	280 U=L(LNG)+1
20 CALL CHAR(128,"1818181818181818")	290 V=C(LNG)
30 CALL CHAR(129,"FFFF")	300 GOSUB 480
40 CALL CHAR(130,"FFFFFFFFFFFF")	310 IF COD=32 THEN 280
50 CALL CHAR(131,"00183C7E7E3C18")	320 U=L(LNG)+1
60 RANDOMIZE	330 V=C(LNG)+1
70 ROC=80	340 GOSUB 480
80 LNG=6	350 IF COD=32 THEN 280
90 CALL HCHAR(24,3,129,30)	360 U=L(LNG)+1
100 CALL VCHAR(1,3,128,23)	370 V=C(LNG)-1
110 CALL VCHAR(1,32,128,23)	380 GOSUB 480
120 FOR I=1 TO ROC	390 IF COD=32 THEN 280
130 CALL HCHAR(2+21*RND,4+25	400 U=L(LNG)
*RND,130,1+RND)	410 V=C(LNG)-1
140 NEXT I	420 GOSUB 480
150 FOR I=3 TO 31	430 IF COD=32 THEN 280
160 CALL GCHAR(1,I,CK)	440 U=L(LNG)
170 IF CK=32 THEN 200	450 V=C(LNG)+1
180 NEXT I	460 GOSUB 480
190 END	470 IF COD=32 THEN 280 ELSE
200 CC=4+27*RND	150
210 CALL GCHAR(1,CC,LL)	480 CALL GCHAR(U,V,COD)
220 IF LL=131 THEN 200	490 IF COD<>32 THEN 580
230 CALL VCHAR(1,CC,131,LNG)	500 CALL HCHAR(U,V,131)
240 FOR I=1 TO LNG	510 CALL HCHAR(L(1),C(1),32)
250 C(I)=CC	520 FOR I=1 TO LNG-1
260 L(I)=I	530 C(I)=C(I+1)
270 NEXT I	540 L(I)=L(I+1)
	550 NEXT I
	560 L(LNG)=U
	570 C(LNG)=V
	580 RETURN

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 8, qui dispose de la possibilité de se déplacer vers la gauche ou vers la droite pour les éviter et qui devra tenir la 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 voiture du joueur est imprimée grâce à la ligne 80.
Les lignes 90, 110 et 130 fabriquent la route (les lignes 100 et 120, bien qu'identiques sont toutes deux indispensables, car la valeur de RND ne sera, en principe, pas la même que ces deux lignes).
La ligne 150 affiche la route et le score.
La ligne 170 permettra de déplacer le véhicule.
Les lignes 180, 190 et 220 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 200.
- 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 8.
La voiture peut se déplacer à l'aide des flèches de déplacement horizontal (touches S et D).
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 CLEAR.
- 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.).

```

10 CALL CLEAR
20 RANDOMIZE
30 Y=7
40 S=0
50 CALL CLEAR
60 CALL CHAR(35,"FFFFFFFFFFFF")
70 CALL CHAR(124,"1818181818181818")
80 CALL HCHAR(1,Y,56)
90 A$="!";          ";"
100 R=7*RND+2
110 A$=SEG$(A$,1,R-1)&"#"%SE
    G$(A$,R+1,10)
120 R=7*RND+2
130 A$=SEG$(A$,1,R-1)&"#"%SE
    G$(A$,R+1,10)
140 CALL HCHAR(23,20,32,10)
150 PRINT A$;TAB(23);S
160 CALL KEY(5,K,E)
170 Y=Y+(K=83)-(K=68)
180 CALL GCHAR(1,Y,C)
190 IF (C=35)+(C=124) THEN 22
    0
200 S=S+1
210 GOTO 80
220 END

```

Captures dans l'espace

- REFERENCES** : Difficulté de programmation : facile.
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 30 permet de faire apparaître les météorites.
La ligne 50 fera apparaître le minerai d'urane.
Leur déplacement, sur l'écran, est assuré par un PRINT ligne 60.
Les lignes 70 et 90 contrôlent le déroulement du jeu (a-t'on heurté une météorite ou a-t'on capturé un bloc d'urane).
La rencontre d'une météorite provoque la fin de partie.
Votre vaisseau spatial est imprimé par la ligne 120 et son déplacement est contrôlé par la ligne 140.
- 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 (S et D) 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 CLEAR.
- 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...

```
10 X=12
20 Z=0
30 CALL HCHAR(20,INT(30*RND)
+2,30)
40 IF RND>.5 THEN 60
50 CALL HCHAR(20,INT(30*RND)
+2,48)
60 PRINT : :
70 CALL GCHAR(12,X,C)
80 IF C=30 THEN 160
90 IF C<>48 THEN 110
100 Z=Z+1
110 CALL HCHAR(10,X-2,32,6)
120 CALL HCHAR(12,X,86)
130 CALL KEY(5,K,E)
140 X=X+(K=68)*(X<32)-(K=83)
*(X>1)
150 GOTO 30
160 CALL CLEAR
170 PRINT "SCORE";Z
```

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 redéfinis et mis en place par les lignes 100 à 130.
Le changement de joueur est effectué par la ligne 150.
Le joueur qui a le trait voit le symbole qu'il a choisi (O ou X) se placer en limite du cadre (lignes 160 et 170).
Les lignes 220 et 230 permettent, aux deux joueurs, de déplacer le symbole sur le terrain.
Les lignes 250 et 260 contrôlent si le déplacement est possible (case libre et non dépassement des limites).
La ligne 310 permet d'écrire sur une case libre.
- POUR JOUER** : RUN provoque l'affichage des limites du terrain et du numéro de 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 (S, D, E, X).
Il n'est pas possible de venir sur une case déjà occupée et il faudra appuyer sur la touche 0 pour inscrire son propre symbole dans une case disponible.
Dès que cela a été fait, le symbole du second joueur apparaît dans la colonne de gauche, à peu près au milieu, et le joueur procédera de la même manière que son adversaire.
Pour sortir du programme, il suffira de taper CLEAR.
- EXTENSIONS POSSIBLES** : On pourra utiliser l'écran tout entier comme terrain de jeu et réaliser des contrôles en fonction des différents jeux que l'on

désire pratiquer : contrôle d'alignements de 5 pièces pour le morpion, contrôle du nombre de pièces capturées pour le GO, etc.

```

10 CALL CLEAR
20 D(0)=48
30 D(1)=88
40 F$="FFFFFFFF"
50 O$="FOFOFOF"
60 CALL CHAR(128,F$&O$)
70 CALL CHAR(129,F$&"O"&O$)
80 CALL CHAR(130,"O"&O$&F$)
90 CALL CHAR(131,O$&"O"&F$)
100 CALL HCHAR(1,3,128)
110 CALL HCHAR(1,32,129)
120 CALL HCHAR(24,32,130)
130 CALL HCHAR(24,3,131)
140 J=0
150 J=ABS(J-1)
160 X=12
170 Y=3
180 CALL HCHAR(X,Y,D(J))
190 CALL KEY(S,K,S)
200 A=X
210 B=Y
220 X=X+(K=88)*(X<24)-(K=69)
    *(X>1)
230 Y=Y+(K=68)*(Y<32)-(K=83)
    *(Y>3)
240 CALL HCHAR(A,B,32)
250 CALL GCHAR(X,Y,C)
260 IF (C<>48)*(C<>88) THEN 3
    00
270 CALL HCHAR(X,Y,C)
280 X=A
290 Y=B
300 IF K=48 THEN 180
310 CALL HCHAR(X,Y,D(J))
320 GOTO 150

```


NIVEAU 5

N°	Nom du programme	Thème et particularités
1	Horloge 2	Utilisation du générateur de caractères incorporé.
2	Dames	Pour deux joueurs : classique.
3	Anagrammes	Un bon jeu de réflexes, pour 2 joueurs.
4	Oeufs	Ne laissez pas faire la pondeuse diabolique ! (1 joueur).
5	Affichage tournant (vertical)	Une possibilité à exploiter de votre TI-99.
6	Quatre à quatre	Un jeu de réflexion un peu difficile, pour 2 joueurs.
7	Taquin de lettres	Un proverbe est caché dedans ; à vous de le découvrir.
8	Cibles	Un peu d'adresse, beaucoup de chance. Très visuel.
9	Affichage publicitaire (2)	Un affichage de lettres géantes !!!
10	Anomalie	Exercez votre perspicacité dans la reconnaissance des symboles.
11	Othello	Pour deux joueurs ; il retournera les pions à votre place.

Présentation de certaines fonctions du Basic étendu

Les programmes présentés dans les quatre premières parties sont exécutables en TI-BASIC avec la console seule (sans qu'aucune extension ne soit nécessaire).

Le Basic Etendu (qui se présente sous la forme d'un module enfichable) offre des possibilités, notamment au niveau graphique, largement accrues. Certains des programmes qui suivent sont écrits dans ce langage et nécessitent le module.

SPRITES :

La caractéristique la plus impressionnante de ce Basic est la faculté de créer et de gérer de petits lutins animés qui seront ensuite manipulés par le processeur vidéo. Nous ne reviendrons pas sur l'importante palette d'instructions y donnant accès.

DISPLAY AT :

Comme vous avez pu le remarquer, tout au long des programmes précédents, il est pratiquement impossible d'afficher une chaîne en un endroit donné de l'écran. Avec l'instruction **DISPLAY AT** du Basic Etendu (qui possède d'ailleurs d'autres capacités comme la mise au format) il devient donc possible d'afficher des chaînes sans que l'écran "ne monte".

Remarque : En comparant les listings de 2 programmes de même nature, l'un en TI-Basic, l'autre en Basic Etendu vous vous apercevrez rapidement que le second apporte une flexibilité et une souplesse d'utilisation importantes.

Horloge 2

- REFERENCES** : Basic Etendu.
Difficulté de programmation : difficile.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.
- LE JEU** : Ce deuxième programme de simulation d'une horloge digitale grand format présente la particularité intéressante d'utiliser les **lutins** (SPRITES) du Basic Etendu.
Ce sont donc les caractères d'origine qui seront reproduits agrandis par l'instruction MAGNIFY (2).
- LE PROGRAMME** : La ligne 10 permet l'introduction de l'heure sous la forme d'un entier de 4 chiffres.
La ligne 20 décode cet entier.
L'heure est affichée au moyen de SPRITES par la ligne 30.
La ligne 50 incrémente les minutes.
Le réglage du temps est fait en ligne 40.
- POUR JOUER** : Faire RUN, puis introduire un entier de 4 chiffres (les deux premiers représentent l'heure, les deux suivants les minutes : 0416 signifiera 4 heures 16 minutes).
Attendre le quatrième top pour taper ENTER, et l'affichage commence.
On pourra essayer de régler l'horloge en jouant sur la durée de la pause en ligne 40
- EXTENSIONS POSSIBLES** : On peut faire clignoter les deux points, ajouter une possibilité de réveil, prévoir un agenda automatique, faire apparaître un coucou chantant au moment des heures, etc.

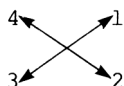
```

10 CALL MAGNIFY(2):: CALL CL
EAR :: INPUT "HEURE (HHMM) :
":H$ :: CALL CLEAR :: CA
LL VCHAR(10,17,46,2)
20 FOR I=1 TO 4 :: S(I)=VAL(
SEG$(H$,I,1)):: NEXT I
30 FOR I=1 TO 4 :: CALL SPRI
TE(#I,48+S(I),2,75,75+20*I):
: NEXT I
40 FOR PAUSE=1 TO 15000 :: N
EXT PAUSE
50 S(4)=S(4)+1
60 IF S(4)=10 THEN S(4)=0 ::
  S(3)=S(3)+1 :: IF S(3)=6 TH
EN S(3)=0 :: S(2)=S(2)+1
  :: IF S(2)=10 THEN S(2)=0 :
: S(1)=S(1)+1
70 IF S(1)=2 AND S(2)=4 THEN
  S(1),S(2)=0
80 GOTO 30

```

Dames

- REFERENCES** : Basic Etendu
 Difficulté de programmation : très difficile.
 Catégorie de jeu : deux joueurs.
 Difficulté du jeu : difficile.
- LE JEU** : C'est le jeu classique de dames à la française, mais un peu incomplet puisqu'il ne prévoit pas la possibilité de faire des dames. La règle du jeu reste classique, mais les contrôles ont été un peu simplifiés. Les mouvements des pièces sont entrés en une seule 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 0 pour pouvoir rejouer s'il y avait eu saut ou si le coup n'était pas possible.
- LE PROGRAMME** : La boucle I ligne 30 dessine le damier (10 cases sur 10 cases) et les pions (X ou O). La ligne 60 indique qui a le trait. Le changement de joueur est traité par la ligne 60 (entrée d'un 0) également. La ligne 70 calcule le déplacement, qui sera réalisé lorsqu'il est possible, par la boucle I (80-120). La ligne 80 calcule la case d'arrivée (elle tient compte du cas où l'on arriverait en ligne 0). La ligne 90 teste les cases à parcourir. La ligne 100 traite le cas d'un déplacement, et 110 celui d'un saut. La ligne 130 effectue les éventuelles modifications, et la ligne 140 réalise le changement de joueur.
- POUR JOUER** : Dès le lancement du programme par RUN, le damier est affiché avec ses pions, et un caractère 0 inversé apparaît en-dessous pour indiquer que le trait appartient au joueur qui a choisi le 0. On joue en indiquant par un nombre de 3 chiffres (pouvant commencer par un 0) la ligne et la colonne de la pièce à déplacer, puis la direction du déplacement (codée de 1 à 4) suivant la convention :



Par exemple 931 signifie que le O situé en ligne 9, colonne 3, se déplace dans la direction 1 (ne pas oublier de faire ENTER). De même, 063 signifiera pièce X, ligne 0, colonne 6, direction 3. Le changement de joueur étant effectué auto-

matiquement, il faudra taper Ø pour rejouer si le coup est refusé ou s'il n'y a pas eu prise (toujours terminer par ENTER). 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 du terrain). On pourra envisager la création de dames, leur déplacement, et éventuellement écrire un programme permettant au TI-99 de se mêler avec de bonnes chances de réussite à vos passionnantes parties.

```

10 CALL CLEAR :: CALL CHAR(3
5,"FFFFFFFFFFFFFF")
20 CALL CHARPAT(79,A$):: CAL
L CHAR(111,A$):: CALL CHARPA
T(88,A$):: CALL CHAR(120
,A$):: CALL COLOR(12,15,2)::
CALL COLOR(10,15,2):: CALL
SCREEN(15)
30 FOR I=0 TO 9 :: DISPLAY A
T(3,12+I):STR$(I):: DISPLAY
AT(I+5,9):I;TAB(12-(I/2<
>INT(I/2)));"o o # x x" :: N
EXT I
40 H=1
50 IF H THEN A$="o" ELSE A$=
"x"
60 DISPLAY AT(20,5):"JOUEUR
";A$ :: ACCEPT AT(20,14)BEEP
:R$ :: IF R$="O" THEN 14
0
70 D=VAL(SEG$(R$,3,1)):: D=9
*((D=1)-(D=3))+11*((D=4)-(D=
2)):: B$=R$ :: T=VAL(SEG
$(R$,1,2)):: R=2
80 FOR I=1 TO 3 :: R$=STR$(A
BS(T+(I-1)*D)):: IF LEN(R$)<
2 THEN R$="O"&R$
90 X=5+VAL(SEG$(R$,1,1)):: Y
=12+VAL(SEG$(R$,2,1)):: CALL
GCHAR(X,Y+2,COD):: IF R
=2 THEN B$=SEG$(B$,1,I-1)&CH
R$(COD)&SEG$(B$,I+1,3)
100 AA$=A$&"#" :: IF SEG$(B$
,1,2)=AA$ THEN B$="#"&A$&SEG
$(B$,3,1)
110 IF SEG$(B$,1,1)=A$ AND S
EG$(B$,2,1)<>A$ AND SEG$(B$,
1,1)="#" THEN B$="###"&A$
120 DISPLAY AT(X,Y)SIZE(1):S
EG$(B$,I,1):: NEXT I
130 R=R-1 :: IF R THEN 80
140 H=1-H :: GOTO 50

```


Anagrammes

REFERENCES

: Basic Etendu.
 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 TI-99, 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 est délivré.

LE PROGRAMME

: Le niveau du jeu est entré à la ligne 1Ø.
 La boucle I (2Ø-3Ø) réalise le choix des lettres.
 Une deuxième boucle I (5Ø-11Ø) effectue le brouillage, et la ligne 7Ø permet de détecter la pression d'une touche.
 La ligne 11Ø affiche le nouveau mot, et fait monter d'une ligne sur l'écran les mots déjà obtenus.
 Les lignes 12Ø et 13Ø cherchent quel est celui des deux joueurs qui a pressé une touche.
 La boucle I (14Ø-17Ø) explore les mots présents sur l'écran ligne par ligne, alors que la boucle J (14Ø-15Ø) saisit un par un les caractères de ces mots. On obtient ainsi un nouveau mot qui est comparé au mot initial en ligne 16Ø, et le résultat BRAVO apparaît si c'est exact.
 Sinon, la partie s'arrête sur l'affichage PERDU (ligne 18Ø).

POUR JOUER

: On lance le programme par RUN.
 On devra ensuite entrer le niveau de difficulté (en principe de 1 à 3) et faire ENTER.
 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 CALL CLEAR :: INPUT "NIVE
AU (1/3) :":N :: CALL CLEAR
20 RANDOMIZE :: S$="" :: FOR
  I=1 TO 3+N
30 S$=S$&CHR$(INT(65+26*RND)
):: NEXT I
40 A$="" :: B$=S$
50 FOR I=1 TO LEN(S$)
60 N=INT(LEN(S$)*RND+1)
70 CALL KEY(S,K,E):: IF K=48
  OR K=49 THEN 120
80 IF SEG$(B$,N,1)="-" THEN
60
90 A$=A$&SEG$(B$,N,1)
100 B$=SEG$(B$,1,N-1)&"-%SE
G$(B$,N+1,LEN(B$))
110 NEXT I :: DISPLAY AT(20,
9):A$ :: S$=A$ :: PRINT :: G
OTO 40
120 IF K=48 THEN H$="DROIT"
130 IF K=49 THEN H$="GAUCHE"
140 FOR I=1 TO 18 :: B$="" :
: FOR J=1 TO LEN(S$)
150 CALL GCHAR(I,J+10,CC)::
B$=B$&CHR$(CC):: NEXT J
160 IF B$=S$ THEN DISPLAY AT
(I,LEN(S$)+11):"BRAVO ";H$ :
: STOP
170 NEXT I
180 PRINT TAB(LEN(S$)+11);"P
ERDU ";H$

```

Oeufs

- REFERENCES** : Basic Etendu.
 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 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 20 et 30 ; son déplacement aléatoire est géré par le sous-programme 60-120 (en particulier les lignes 100 et 110).
 La ligne 30 reboucle le programme sur la ligne 20.
 Le déplacement du joueur est régi par la ligne 40, et le test permettant de détecter la présence d'un oeuf est effectué à la ligne 50.
 La ligne 50 permet l'affichage du score.
 La fin du jeu est détectée en ligne 60.
- POUR JOUER** : On lancera le programme avec RUN.
 L'oiseau commence alors à se déplacer au hasard sur l'écran, et se met à pondre des oeufs.
 Lorsque le symbole qui représente le joueur (caractère redéfini) 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 CLEAR.
- 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.

```

10 CALL CLEAR :: P=200 :: X=
INT(20*RND):: Y=INT(20*RND):
: R,S=1 :: T=0
20 DISPLAY AT(X,Y):"\/" :: C
ALL KEY(S,K,Z):: IF Z THEN G
OSUB 40
30 GOSUB 60 :: DISPLAY AT(X,
Y):"/\" :: GOSUB 60 :: GOTO
20
40 DISPLAY AT(R,S):" " :: R=
R+(K=88)*(R<20)-(K=69)*(R>1)
:: S=S+(K=68)*(S<30)-(K=
83)*(S>1)
50 CALL GCHAR(R,S+2,CC):: E=
CC=79 :: T=T-E :: DISPLAY AT
(22,12):"SCORE :";T :: R
ETURN
60 A=X :: B=Y :: P=P-1 :: IF
P=0 THEN PRINT "C'EST FINI.
..." :: STOP
70 DISPLAY AT(R,S):"H"
80 FOR U=1 TO 150 :: NEXT U
90 DISPLAY AT(A,B):" O"
100 X=ABS(X+2*RND-1):: Y=ABS
(Y+4*RND-2):: IF X>19 THEN X
=1
110 IF Y>29 THEN Y=1
120 RETURN

```

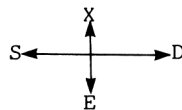
Affichage tournant (vertical)

- REFERENCES** : Basic Etendu.
Difficulté de programmation : facile.
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 des SPRITES a permis de simplifier considérablement le programme.
- LE PROGRAMME** : La ligne 10 permet l'introduction du texte; qui sera rangé dans une variable alphanumérique A\$.
La boucle I ligne 20 réalise l'affichage lettre après lettre en créant pour chacune d'entre-elles une SPRITE défilant verticalement. Les SPRITES étant gérées par le processeur vidéo, il suffit ensuite de faire une boucle infinie afin de ne pas les détruire.
- POUR JOUER** : On doit d'abord lancer le programme par l'instruction RUN, puis on entre le texte, on tape ensuite ENTER, et on regarde.
Le programme boucle sur lui-même et le défilement ne pourra être interrompu que par un appui sur la touche CLEAR.
- EXTENSIONS POSSIBLES** : Il suffira d'avoir un peu d'imagination pour adapter ce programme à toute sorte d'affichage 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.

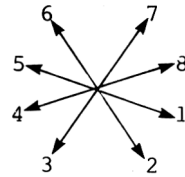
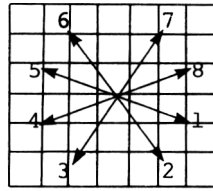
```
10 ACCEPT AT(10,1)SIZE(9)ERA  
SE ALL BEEP:A$ :: CALL CLEAR  
:: CALL MAGNIFY(2)  
20 FOR I=1 TO LEN(A$):: CALL  
  SPRITE(#I,ASC(SEG$(A$,I,1))  
  ,2,5+20*I,100,-20,0):: N  
EXT I  
30 GOTO 30
```

Quatre à quatre

- REFERENCES** : Basic Etendu.
 Difficulté de programmation : réflexion (un ou deux joueurs).
 Difficulté du jeu : difficile.
- LE JEU** : Il pourra être pratiqué à un ou deux 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 10, contient la table des coups possibles (déplacement du cavalier). Les boucles emboîtées I et J (20-60) dessinent le terrain de jeu, et les positions de départ.
 La ligne 80 permet d'afficher O ou X, selon le joueur qui a le trait.
 La ligne 90 permet l'affichage du curseur clignotant (ici un +) qui indique sur quelle case l'on se trouve.
 La ligne 110 regarde si l'on a appuyé sur une touche de 1 à 8, donnant la direction du déplacement.
 La ligne 130 réimprime la figure qui était sous le curseur.
 La ligne 140 réalise le contrôle du départ. Les lignes 150 et 160 effectuent le déplacement.
 Le résultat est imprimé par la ligne 170, le changement de joueur est réalisé par la ligne 180, et un . est imprimé à l'emplacement de départ. Le sous-programme 190 est utilisé pour les divers contrôles.
- POUR JOUER** : On lance le programme par RUN. Le terrain de jeu est affiché, ainsi qu'un 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 S, D, E, X.



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 CLEAR.

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 TI-99 de participer au jeu avec de bonnes chances de l'emporter.

```

10 CALL CLEAR :: H=1 :: A$=""
5665322356"
20 FOR I=1 TO 5 :: FOR J=1 TO 5
0 5
30 CALL HCHAR(10,10+2*I,48+I)
):: CALL HCHAR(10+I+I,10,48+
I):: CALL HCHAR(10+I+I,1
0+J+J,46)
40 IF I<3 AND J<3 THEN CALL
HCHAR(I+I+10,10+J+J,79)
50 IF I>3 AND J>3 THEN CALL
HCHAR(I+I+10,10+J+J,88)
60 NEXT J :: NEXT I
70 X,Y=H
80 DISPLAY AT(22,5):"JOUEUR"
:: IF H THEN DISPLAY AT(22,
12):"O" ELSE DISPLAY AT(
22,12):"X"
90 GOSUB 190 :: I=J :: CALL
HCHAR(10+X+X,10+Y+Y,43)
100 A=X :: B=Y :: CALL KEY(5
,K,E)
110 X=X+(K=88)*(X<5)-(K=69)*
(X>1):: Y=Y+(K=68)*(Y<5)-(K=
83)*(Y>1)
120 IF K<57 AND K>48 THEN 14
0
130 CALL HCHAR(A+A+10,B+B+10
,I):: GOTO 80
140 J=K-48 :: IF J>9 OR(I<>7
9 AND H<>0)OR(I<>88 AND H=0)
THEN 130
150 X=X+VAL(SEG$(A$,J,1))-4
160 Y=Y+VAL(SEG$(A$,J+2,1))-
4 :: GOSUB 190 :: IF J-46 TH
EN 130
170 IF H THEN CALL HCHAR(X+X
+10,Y+Y+10,79)ELSE CALL HCHA
R(X+X+10,Y+Y+10,88)
180 H=ABS(H-1):: I=46 :: GOT
O 130
190 CALL GCHAR(10+X+X,10+Y+Y
,J):: RETURN

```


Taquin de lettres

- REFERENCES** : Basic Etendu.
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.
- LE JEU** : Qui d'entre-nous n'a pas, un jour ou l'autre, eu dans les mains un de ces petits puzzles où l'on essaie de déplacer d'une case à une autre des petits carrés de plastique colorés, jusqu'à l'obtention d'un résultat bien précis ? En son temps, le 14-15, de SAM LOYD, a eu autant de succès que notre célèbre RUBIK'S. Le taquin qui vous est proposé ici utilise 35 lettres, disposées au hasard dans un carré de 6 cases sur 6 cases, et ces lettres peuvent, replacées dans leur ordre initial, vous permettre de reconstituer un proverbe très connu. Une case grise vous permet de déplacer les lettres, et un compteur affiche en permanence le nombre d'essais : arriverez-vous à retrouver le proverbe ?
- LE PROGRAMME** : La phrase à reconstituer (qui pourra bien sûr être modifiée) est placée dans une variable B\$, en ligne 10.
Le mélange des lettres et l'impression sont faits par la boucle I (20-50).
La ligne 70 réalise l'affichage du carré noir ; celui-ci est déplacé par la ligne 90 en utilisant les 4 flèches.
La ligne 100 réimprime la lettre concernée à l'emplacement du carré noir.
Elle comptabilise le nombre d'essais, qui est imprimé grâce à la ligne 70.
- 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 noir à l'aide des 4 touches de direction (S, D, E, X).
- EXTENSIONS POSSIBLES** : Le même principe peut être appliqué à toute sorte de taquins (manipulation de nombres, de couleurs, de sons éventuellement). On peut aussi prévoir plusieurs phrases, avec un choix effectué par un générateur aléatoire, etc.

```

10 CALL CLEAR :: A$="" :: R$
="ONASOUVENTBESOIND.UNPLUSPE
TITQUESOI" :: RANDOMIZE
20 FOR I=1 TO 35
30 X=INT(35*RND+1):: IF SEG$
(R$,X,1)="-" THEN 30
40 A$=A$&SEG$(R$,X,1):: R$=S
EG$(R$,1,X-1)&"- "&SEG$(R$,X+
1,35-X)
50 CALL HCHAR(INT(I/6)+5,I-6
*INT(I/6)+5,ASC(SEG$(A$,I,1)
)):: NEXT I
60 X,Y,S=0
70 DISPLAY AT(1,7):"NOMBRE D
'ESSAIS :";S :: CALL HCHAR(X
+5,Y+5,30)
80 CALL KEY(S,K,E):: IF E=0
THEN 80
90 I=X :: J=Y :: X=X+(K=88)*
(X<5)-(K=69)*(X>0):: Y=Y+(K=
68)*(Y<5)-(K=83)*(Y>0)
100 CALL GCHAR(X+5,Y+5,C)::
CALL HCHAR(I+5,J+5,C):: S=S+
1 :: GOTO 70

```

Cibles

- REFERENCES** : Basic Etendu.
 Difficulté de programmation : difficile.
 Catégorie de jeu : hasard.
 Difficulté du jeu : facile.
- LE JEU** : Il consiste simplement à tirer sur des cibles en essayant de marquer le maximum de points ; le nombre de cibles (de 1 à 3) sera choisi en début de partie, et le score réalisé sera affiché en permanence. Si l'on touche le bord d'une cible, la partie s'arrête. On marquera d'autant plus de points que le nombre sur lequel on tombe sera plus grand.
- LE PROGRAMME** : Le nombre de cibles est entré en ligne 20.
 La boucle I ligne 50 effectue l'impression des cibles.
 La ligne 60 affichera le score en permanence.
 La boucle est affichée par la ligne 70.
 La boucle I (80-90) réalise l'impression des chiffres contenus dans les cibles, et permet le déplacement de la boule (ligne 90).
 La ligne 110 regarde si l'on a tiré, et la ligne 120 regarde si la boule est encore sur la ligne de départ.
 La ligne 130 fait progresser la boule d'une case vers la droite et vérifie si l'on a ou non atteint les cibles ; puis la ligne 140 regarde ce que l'on a touché.
 La ligne 150 incrémente le score. (Le nombre de points marqués correspond au chiffre affiché).
 La ligne 150 s'occupe également de la fin de partie.
- POUR JOUER** : Lancer le programme par RUN, et taper le nombre de cibles désirées, suivi de ENTER.
 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 S et D), et vous pouvez tirer à tout moment en tapant 0. Le score sera modifié en fonction de votre résultat, et une nouvelle balle apparaît.
 Il est possible d'arrêter à tout moment en tapant CLEAR, sinon le jeu s'arrête au bout d'un certain nombre de tirs.
- EXTENSIONS POSSIBLES** : On pourra sans difficulté augmenter le nombre de cibles, modifier la règle d'apparition des nombres dans les cibles pour la rendre moins hasardeuse, s'arranger pour que la balle elle-même soit représentée par un chiffre qui devrait coïncider à l'arrivée avec celui de la cible.

```

10 RANDOMIZE :: S=0
20 ACCEPT ERASE ALL BEEP:N :
: CALL CLEAR
30 P=10 :: J=25
40 CALL CHAR(128,"FFFF"&RPT$(
"03",20)&"FFFF")
50 FOR I=1 TO N :: CALL HCHA
R(3*I,J,128):: CALL HCHAR(3*
I+1,J,129):: CALL HCHAR(
3*I+2,J,130):: NEXT I
60 K=2 :: A=24 :: DISPLAY AT
(A,10):"SCORE";S
70 CALL HCHAR(A,K,48):: D=A
80 FOR I=1 TO N :: U=INT(P*R
ND):: CALL HCHAR(3*I+1,J-1,4
8+U)
90 CALL KEY(S,T,E):: A=A+(T=
69)-(T=88):: NEXT I
100 CALL HCHAR(D,K,32)
110 IF T=48 THEN 130
120 IF K=2 THEN 70
130 K=K+1 :: IF K<=J-3 THEN
70
140 CALL GCHAR(D,K+1,C):: IF
C=32 THEN PRINT "PERDU" ::
END
150 S=S+C-48 :: P=P-1 :: IF
P=0 THEN STOP ELSE GOTO 30

```

Affichage publicitaire (2)

- REFERENCES** : Basic Etendu.
 Difficulté de programmation : difficile.
 Catégorie de jeu : graphique.
 Difficulté du jeu : aucune.
- LE JEU** : Ce programme va permettre l'affichage tournant d'un texte en grand format ; il utilise les SPRITES et le générateur de caractères. On devra redéfinir les caractères que l'on veut afficher car ils apparaissent sur l'écran au moyen de SPRITES magnifiés (ligne 10). L'exemple donné page suivante a redéfini le O, le H et le E. La redéfinition débute pour A au code 40 et va, bien évidemment de quatre en quatre.
- LE PROGRAMME** : La ligne 10 permettra l'introduction du texte. Les lignes 20 à 40 redéfinissent chaque lettre par 4 caractères (ici O, H, E). La ligne 60 permet de reboucler indéfiniment sur le défilement. La boucle I (50-60) réalise l'affichage lettre par lettre.
- POUR JOUER** : On devra d'abord lancer le programme par RUN, puis on entrera le texte à afficher, suivi de ENTER. (en ajoutant 10 INPUT A\$). Le texte commence à défiler sur l'écran; les lettres apparaissent les unes derrière les autres avec un maximum de 4 sur l'écran. On peut arrêter le programme à tout moment en tapant CLEAR.
- EXTENSIONS POSSIBLES** : Il sera possible de reprendre cette idée pour réaliser toute sorte d'affichages fantaisie (couleurs, caractères spéciaux, etc.).

```

10 A$="DHHEEE" :: CALL CLEAR
   :: CALL MAGNIFY(4)
20 CALL CHAR(68,"E0E0E0E0E
  0E0FFFFE0E0E0E0E0E0E007070
  7070707FFFF0707070707070
  7")!H
30 CALL CHAR(56,"FFFFFFE0E
  0E0FFFFE0E0E0E0E0E0E0F0F0
  0000000E0E000000000F0F0F
  0")!E
40 CALL CHAR(96,"FFFFFFFE0E
  0E0E0E0E0E0E0E0E0E0E0E0E0E
  F0707070707070707070707FFF
  F")!D
50 FOR I=1 TO LEN(A$):: CALL
   SPRITE(#{I-4*INT((I-1)/4)},
   (ASC(SEG$(A$,I,1))-64)*4
   +36,2,100,240,0,-30)
60 FOR P=1 TO 150 :: NEXT P
   :: NEXT I :: CALL DELSPRITE(
   ALL):: GOTO 50

```

Anomalie

- REFERENCES** : Basic Etendu.
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. Les lettres de l'alphabet sont affichées en des endroits différents des deux grilles, mais l'une des lettres a été modifiée sur la grille de droite. Le jeu consiste à trouver laquelle, en entrant ses coordonnées dans l'ordre ligne-colonne.
- LE PROGRAMME** : La boucle I ligne 10 imprime les symboles des deux grilles.
La boucle principale I ligne 20 imprime les symboles des deux grilles, dans un ordre différent.
La ligne 20 choisit également le caractère à imprimer.
Le sous-programme (60-80) s'occupe de faire l'impression à un emplacement libre (on l'appelle deux fois pour les deux grilles).
La ligne 30 change, de façon aléatoire, l'un des caractères.
Les coordonnées du point essayé sont entrées à cette même ligne.
Le contrôle est effectué par la ligne 40, qui s'occupe également 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, et son numéro de colonne, suivi de ENTER.
Le programme répondra alors BRAVO ou PERDU.
- EXTENSIONS POSSIBLES** : Il sera possible d'utiliser des symboles graphiques de diverses couleurs, ce qui pourra simplifier la recherche.
On peut aussi envisager d'utiliser des dessins plus sophistiqués, améliorer le contrôle du résultat, admettre la bonne réponse sur la grille de droite ou sur celle de gauche, introduire un minutage, etc.

```

10 CALL CLEAR :: T=5 :: FOR
I=1 TO T :: CALL HCHAR(5,5+I
+I,48+I):: CALL HCHAR(5,
21+I+I,48+I):: CALL HCHAR(5+
I+I,5,48+I):: CALL HCHAR(5+I
+I,21,48+I):: NEXT I
20 RANDOMIZE :: FOR I=1 TO 2
2 :: AL=INT(26*RND+65):: L=1
:: S=T :: GOSUB 60 :: L
=9 :: GOSUB 60 :: NEXT I
30 S=0 :: GOSUB 60 :: INPUT
" LIGNE,COLONNE : ":I,T
:: U=2*I-1 :: V=2*T+16
40 GOSUB 70 :: CALL CLEAR ::
IF AL=C THEN DISPLAY AT(10,
10)BEEP:"BRAVO" ELSE DIS
PLAY AT(10,10)BEEP:"PERDU"
50 END
60 U=2*INT(T*RND+1):: V=2*IN
T(S*RND+L)
70 CALL GCHAR(U+5,V+5,C):: I
F C<>32 AND S<>0 THEN 60
80 CALL HCHAR(U+5,V+5,AL)::
RETURN

```


Othello

- REFERENCES** : Basic Etendu.
 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 procédera 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** : La boucle P ligne 20 dessine le terrain de jeu et la ligne 50 affiche un X ou un O, selon le joueur qui commence.
 La boucle P (70-130) contrôle la validité du coup (la case est-elle libre ? : ligne 120), en utilisant une partie de programme qui se trouve aux lignes 150 à 180 (a-t-on pris un ou plusieurs ennemis en ligne 150, et retournement éventuel, autorisé par la ligne 160 : on refait un balayage dans toutes les directions, si l'encadrement s'est révélé possible).
 La ligne 70 définit les 8 directions d'exploration.
 La ligne 80 calcule les déplacements à effectuer pour explorer ces 8 directions.
 La ligne 120 réalise les déplacements.
 La ligne 140 assure le changement de joueur, s'il y a eu retournement (contrôlé en ligne 130).
- POUR JOUER** : Après RUN, chaque joueur entrera à son tour les coordonnées (ligne, puis colonne, suivi de ENTER) de la case où il désire jouer ; si le coup est accepté, les pions sont retournés et l'on change de joueur ; si aucun coup

n'est possible, on entre 0 en première coordonnée pour le changement de joueur.

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 TI-99.

Remarque : Après avoir lancé le programme par RUN, il convient d'entrer la dimension de la grille (3, ENTER, ou 4, ...).

```

10 CALL CLEAR :: INPUT "DIME
NSION : ":A :: CALL CLEAR
20 B=1 :: FOR P=B TO A+A ::
DISPLAY AT(1,1+P+P):P :: DIS
PLAY AT(P+P+1,1):P :: NE
XT P
30 DISPLAY AT(A+A+1,A+A+2)SI
ZE(3):"X 0"
40 DISPLAY AT(A+A+3,A+A+2)SI
ZE(3):"0 X" :: K=B
50 IF K THEN DISPLAY AT(23,2
8):"X" ELSE DISPLAY AT(23,28
):"0"
60 DISPLAY AT(24,2):"LIGNE:
COLONNE:" :: ACCEPT AT(
24,8)BEEP SIZE(-1):U ::
IF U=0 THEN 140
70 ACCEPT AT(24,22)BEEP:V ::
H=PI :: FOR P=B TO 8 :: S=0
80 W=0 :: T=U+U :: R=V+V ::
F=(P>3 AND P<7)-(P<3 OR P>7)
:: G=(P>B AND P<5)-(P>=6
)
90 IF S=1 THEN R$="X"
100 IF S=2 THEN R$="0"
110 DISPLAY AT(T+1,R+2)SIZE(
1):R$
120 T=T+F+F :: R=R+G+G :: CA
LL GCHAR(T+1,R+4,C):: IF C=8
8 OR C=79 THEN 150
130 NEXT P :: IF H THEN 50
140 K=ABS(K-B):: GOTO 50
150 IF (K<>0 AND C=79)OR(C=8
8 AND K=0)THEN W=B
160 DISPLAY AT(T+1,R+2)SIZE(
1):R$ :: IF W<>0 AND((C=79 A
ND K<>0)OR(C=88 AND K=0)
)THEN 120
170 IF W=0 OR S<>0 THEN 130
180 S=B+1-K :: H=H*(1-S):: G
OTO 80

```


Achevé d'imprimer en février 1984
sur les presses de l'imprimerie Laballery et C[®]
58500 Clamecy
Dépôt légal : janvier 1984

N° d'impression : 401007
N° d'édition : 86595-114-1
ISBN : 2-86595-114-6



Enseigner et distraire est le double objectif de cet ouvrage.

Ces 102 programmes de jeux vous entraînent
à la découverte du TI 99/4A.

La classification par niveaux de ces programmes vous
permet d'acquérir progressivement des connaissances
nouvelles et une plus grande maîtrise du Basic.

Pour chaque niveau, la description des jeux
proposés est suivie d'un programme abondamment
commenté et d'un exemple d'utilisation. Le dernier
niveau constitue une illustration claire de
l'utilisation du Basic étendu.

DES LIVRES
POUR VOTRE
ORDINATEUR



110 FF
ISBN : 2-86595-114-6



102 PROGRA
MME
S
POUR
TI-99/4A

PAR JACQUES DECONCHAT

