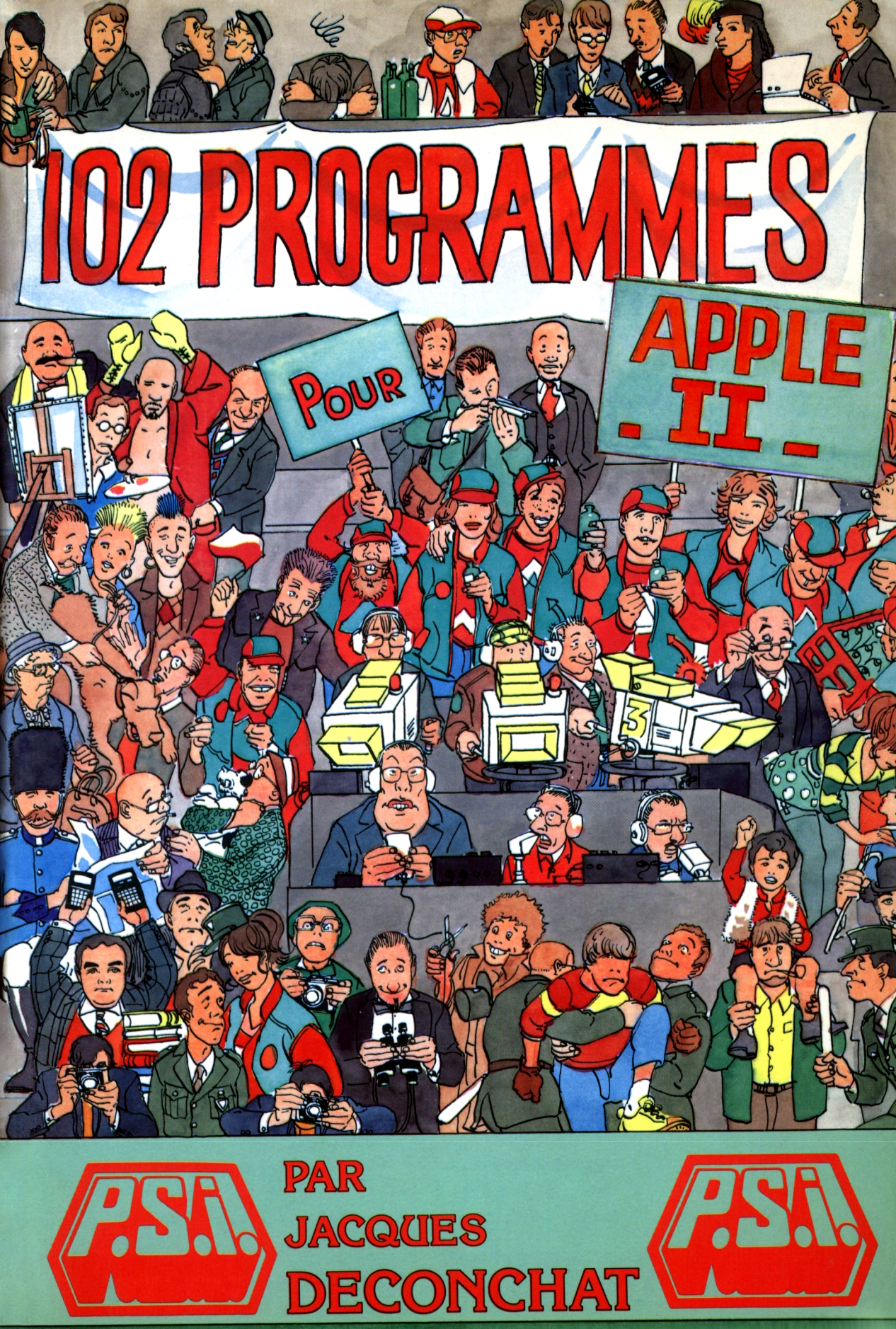




PAR
JACQUES
DECONCHAT



**102 PROGRAMMES POUR
APPLE II**

Autres ouvrages relatifs au matériel APPLE

Certains de ces ouvrages disposent d'une disquette d'accompagnement*.

-
- Apple pour tous - Jacques Boisgontier
 - 36 programmes Apple II pour tous - Jacques Boisgontier
 - La pratique de l'Apple II, tome 1 - Nicole Bréaud-Pouliquen
 - La pratique de l'Apple II, tome 2 - Nicole Bréaud-Pouliquen
 - La pratique de l'Apple II, tome 3 - Nicole Bréaud-Pouliquen et Daniel-Jean David
 - La découverte de l'Apple II - Dominique Schraen et Frédéric Lévy
 - Exercices pour Apple II - Frédéric Lévy, traduit par André Babéanu
 - Du logo pour Apple - Nicole Bréaud-Pouliquen
 - Recueil Pom's n° 1
 - Recueil Pom's n° 2
 - Gestion de fichiers et de périphériques pour Apple II - Hervé Haut
 - *Visicalc sur Apple - Hervé Thiriez
 - Pascal UCSD sur Apple II, tome 1 - Jacques Rouault et Patrice Girard
 - Pascal UCSD sur Apple II, tome 2 - Jacques Rouault et Patrice Girard
 - *Multiplan pour Apple II - Hervé Thiriez
 - Les bases de données sur Apple II - Michel Keller
 - *L'Apple et ses fichiers - Jacques Boisgontier
 - Clefs pour l'Apple II - Nicole Bréaud-Pouliquen
 - Microbook: base de données pour Apple II - Ted Lewis, traduit par André Babéanu
 - Macintosh le magnifique - Merl K. Miller et Mary A. Myers, adapté par Raymond Borraz

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

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

**JACQUES
DECONCHAT**

**102 PROGRAMMES POUR
APPLE II**

adapté par Jacques LABIDURIE



EDITIONS DU P.S.I.

1984

SOMMAIRE

Avant-propos	8
Organisation de l'ouvrage	9

Niveau 1

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

Niveau 2

Dimensionnement d'un tableau - Recherche d'un élément dans un tableau et dans une chaîne	60
Dé truqué	62
Visions	64
421	66
L'électricien fou	68
La course au 20	70
Musique	72
Poker	74
Le jeu de saute-mouton	76
Lettres	78
Espion	80

Accrochez les wagons	82
Tirage d'un dé truqué	84
Tour de cartes	86
L'indésirable	89
Le jeu de Marienbad	92
Compose	94
Jour de la semaine	96
Le jeu de la vie	98

Niveau 3

Manipulation des chaînes et traitement de données multiples	102
Machinamot	104
Plus grand, plus petit	106
Ordre	108
Le pendu	111
Belote	114
Un taquin de calcul	116
Pendu inversé	118
Nicomaque	120
Course de chevaux	122
A travers	124
Le loup et les agneaux	127
Mastermind	130
Idem	132
Horloge 1	134
Reines sur l'échiquier	136
Répétitions	138
Mots	140
Permutations	142
Bridge	144
Tirage de dés	146
Hanoï	148

Niveau 4

Les techniques d'animation élémentaire	152
Jeu de loto	154
La bombe	156
Le jeu du 15	159
Affichage publicitaire	162
Chapeaux	164
Combat aérien	166
Dessin de dés	168
Le jeu du missile	170
Zombies	173
Le singe	176
Adresse	178

Sous-marin	180
Diago	182
Décomaque	184
Chasse au sous-marin	186
Clavier	188
Le lézard d'Alice	190
Calculatrice	193
Echecs	196
Horloge 2	199
Danger spatial	202
Dames	204

Niveau 5

Les graphiques haute et basse résolution sur APPLE II	210
Un dé	212
Losanges	214
Mathieu	216
Ardoise magique	218
Evasion	220
Smokey	222
Course de voitures	224
Tic tac toe	226
Quatre à quatre	228
Taquin de lettres	231
Anagrammes	234
Œufs	236
Cibles	238
Gouttes	240
Obstacles	242
Captures dans l'espace	244
Anomalie	246
Plusieurs en un	248
Parcours du cavalier	250
Mur de briques	252
Othello	254

Avant-propos

Vous trouverez dans ce livre plus de 100 programmes (essentiellement des jeux), rédigés en Basic Applesoft, pour l'ordinateur APPLE II. La plupart de ces programmes seront d'ailleurs utilisables sans grande modification sur d'autres ordinateurs, utilisant aussi un Basic APPLE II, et ils sont tous utilisables tels quels sur les appareils de la gamme APPLE II, II plus, IIe et IIc.

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

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

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

Les programmes n'utilisent pas toutes les particularités d'un appareil comme l'APPLE II, par ailleurs très bien détaillées dans le manuel de référence.

Les niveaux 4 et 5 donnent une idée sur l'utilisation de certaines techniques un peu spécifiques, comme les POKE ou les instructions graphiques. Ces techniques pourront être reprises avec profit pour améliorer les programmes précédents.

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

Organisation de l'ouvrage

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

- **Le niveau 1** permet l'étude des instructions élémentaires du langage Basic : **LET** (optionnel ici), **PRINT**, **INPUT**, **IF...THEN**, **FOR...NEXT**, **GOTO**, **GOSUB**, **ON...GOTO**, **ON...GOSUB**.
- **Au niveau 2**, on utilisera les instructions de confection et de manipulation des éléments d'un tableau, et de recherche d'un élément dans une chaîne : **DIM**, **R(I)**, **MID\$**.
- **Au niveau 3**, on utilise la panoplie complète des instructions de traitement des chaînes de caractères : **MID\$**, **RIGHT\$**, **LEFT\$**, **DATA** et **READ**.
- **Le niveau 4** fait appel à une instruction très particulière qui facilite la réalisation de jeux interactifs : **POKE**.
- Enfin, **le niveau 5** nous fait découvrir des instructions plus spécifiques : utilisation du graphique, identification d'objets sur l'écran : **GR**, **HGR**, **SCRN (X, Y)**.

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

Le programme, fonctionnant sur l'APPLE II, est ensuite reproduit, ainsi qu'un exemple d'exécution. Les listings ont été obtenus directement d'après les originaux de l'auteur, sur une imprimante matricielle, ce qui devrait garantir contre tout risque de dysfonctionnement. Les minuscules ne sont pas utilisées.

Niveau 1

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

Instructions élémentaires

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

- *RND (1) pour obtenir un nombre aléatoire entre 0 et 1.*
- *INT pour prendre la partie entière d'un résultat.*
- *REM pour placer un commentaire.*
- *END pour terminer un programme.*
- *ASC pour obtenir le code ASCII d'un caractère.*
- *CHR\$ pour obtenir un caractère dont on connaît le code ASCII.*

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

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

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

L'INSTRUCTION LET OU INSTRUCTION D'AFFECTATION

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

- **Si la quantité est numérique**, le nom sera une simple lettre de l'alphabet, ou un ensemble de plusieurs chiffres ou lettres, commençant obligatoirement par une lettre.

- Si la quantité est alphanumérique, la même règle s'applique pour le nom, qui devra cependant être suivi du symbole \$, et la variable devra être écrite entre guillemets.

Exemple :

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

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

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

L'INSTRUCTION PRINT OU INSTRUCTION DE SORTIE (ECRAN)

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

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

- **PRINT TAB(N)** provoque sur la ligne d'impression courante un affichage commençant à la *énième* colonne à partir de la gauche ($1 \leq N \leq 40$).
- Un ; (**point-virgule**) en fin de PRINT permet soit d'enchaîner une autre impression sans répéter l'instruction PRINT, soit d'indiquer que la prochaine impression sera, en l'absence d'indication contraire, effectuée exactement à la suite de l'impression précédente.
- La , (**virgule**) provoque une tabulation horizontale automatique pré-définie.
- **PRINT CHR\$ (7)** provoque l'émission d'un BIP.

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

LES INSTRUCTIONS VTAB ET HTAB

L'APPLE II dispose de deux instructions de tabulation permettant de positionner le curseur sur l'écran, en vue de la prochaine écriture. Il s'agit de **VTAB L** et de **HTAB C**. La première (tabulation verticale) positionne

le curseur à la ligne L (compris entre 1 et 24), la seconde à la colonne C (compris entre 1 et 40). Ainsi, pour afficher le mot "BONJOUR" à la 5^e colonne de la 8^e ligne, on écrira : **VTAB 8 : HTAB 5 : PRINT "BONJOUR"**.

LES MODES D’AFFICHAGE

L’APPLE II permet d’afficher un caractère en trois modes différents :

- INVERSE positionne le mode en inverse (noir sur fond blanc).
- FLASH positionne en clignotement.
- NORMAL revient au mode normal (blanc sur fond noir).

EFFACEMENT DE LIGNE

CALL-868 permet d’effacer la ligne en cours, depuis la position actuelle du curseur (éventuellement modifiée par VTAB et HTAB) jusqu’au bout de la ligne.

EFFACEMENT DE L’ECRAN

HOME efface l’écran entier et repositionne le curseur en haut à gauche (LIGNE 1, COLONNE 1).

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

Cette instruction, qui provoque un arrêt et une attente de l’appareil, ***permettra d’entrer un nombre, ou un texte, qui sera affecté à la variable suivant l’ordre INPUT.*** Ceci est fait en mode conversationnel, c’est-à-dire au fur et à mesure des besoins ; en effet, lorsque l’APPLE II rencontre une instruction INPUT, il s’arrête et vous rend la main ; on peut alors lui entrer une information de nature convenable (nombre ou texte), et lui signaler que cette entrée est terminée à l’aide de la touche **RETURN**, ce qui lui rendra la main.

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

Il est possible, comme pour le PRINT, d’afficher des messages avant l’entrée de la variable, et d’utiliser les instructions **VTAB** et **HTAB**.

On pourra, par exemple, écrire :

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

L'INSTRUCTION GOTO OU INSTRUCTION DE RENVOI INCONDITIONNEL

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

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

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

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

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

IF condition = vraie THEN numéro de ligne,

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

On pourra aussi utiliser le test sous la forme :

IF condition THEN instruction ;

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

On peut par exemple écrire :

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

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

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

Ainsi, si l'on écrit :

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

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

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

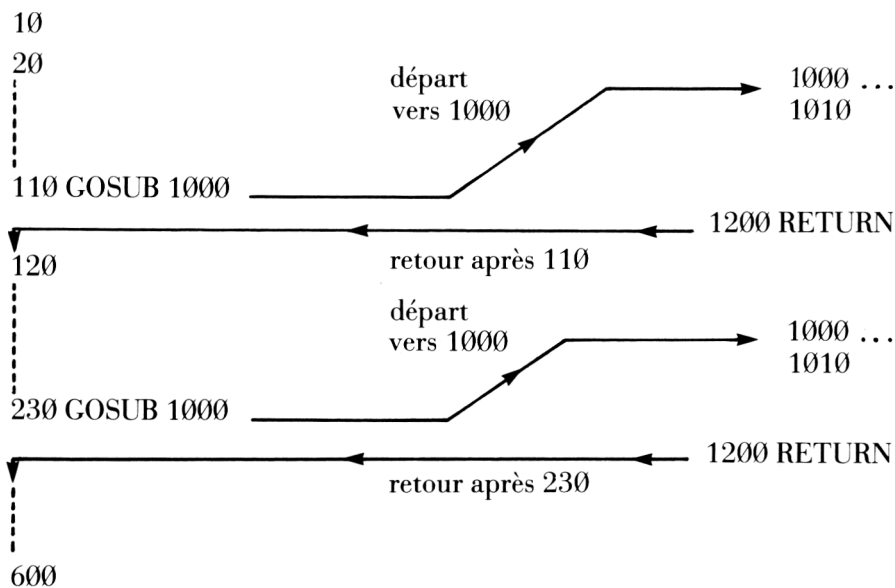
Il est possible d'emboîter plusieurs boucles de noms différents, et d'omettre l'indication de la variable dans le NEXT.

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

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

Voici le schéma théorique de fonctionnement :

Programme principal



Fin du programme principal

1000
1200 RETURN

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

Sous-programme 1

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

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

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

Le nombre mystérieux

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu de déduction pour un joueur.

Difficulté du jeu : facile.

LE JEU :

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

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

LE PROGRAMME :

La ligne 60 permet de fabriquer le nombre à deviner.

Les entrées sont faites par les lignes 80 à 90.

La ligne 100 teste si le résultat est exact.

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

Le compteur d'essais est en ligne 130.

Les lignes 160 à 180 s'occupent de l'affichage du résultat, et de la gestion de la partie suivante.

POUR JOUER :

On lancera le programme par RUN.

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

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

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

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

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

On peut arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :

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

```

10 REM LE NOMBRE MYSTERIEUX
20 HOME
30 PRINT "NOMBRE MYSTERIEUX": PRINT
   "-----"
40 PRINT : PRINT : INPUT "NB. DE
   CHIFFRES (1 A 9) : ";NC
50 IF NC > 9 THEN 40
60 X = INT ( RND (1) * 10 ^ NC)
70 J = 1
80 PRINT : PRINT "ESSAI N.";J
90 INPUT "VOTRE NOMBRE : ";Y
100 IF Y = X THEN 160
110 IF Y < X THEN PRINT "    ==
    === TROP PETIT"
120 IF Y > X THEN PRINT "    ==
    === TROP GRAND"
130 J=J+1
140 PRINT : PRINT
150 GOTO 80
160 HOME
170 PRINT "VOUS AVEZ TROUVE ";X;
   " EN ";J;" COUPS...."
180 PRINT : INPUT " UNE AUTRE PAR
   TIE (O/N) : ";K$: IF K$ = "O
   " THEN 20

```

Cube

REFERENCES :

Difficulté de programmation : aucune.

Catégorie de jeu : dessin.

Difficulté du jeu : aucune.

LE JEU :

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

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

LE PROGRAMME :

Le dessin du cube est réalisé par l'impression de points, en utilisant deux boucles : la boucle I (30-120), et la boucle I (130-180).

Le nom de l'objet représenté est affiché en ligne 190 (ici : CUBE).

POUR JOUER :

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

EXTENSIONS POSSIBLES :

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

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

```

10  REM CUBE
20  HOME
30  FOR I = 1 TO 10
40  VTAB 7 + I: HTAB 1: PRINT "."

50  VTAB I: HTAB 8: PRINT "."
60  VTAB 7 + I: HTAB 11: PRINT "."
   "
70  VTAB I: HTAB 18: PRINT "."
80  VTAB 1: HTAB 7 + I: PRINT "."

90  VTAB 11: HTAB 7 + I: PRINT "."
   "
100 VTAB 8: HTAB I: PRINT "."
110 VTAB 18: HTAB I: PRINT "."
120 NEXT I
130 FOR I = 1 TO 8
140 VTAB I: HTAB 9 - I: PRINT "."
   "
150 VTAB 10 + I: HTAB 9 - I: PRINT
   "."
160 VTAB I: HTAB 19 - I: PRINT "
   ."
170 VTAB 10 + I: HTAB 19 - I: PRINT
   "."
180 NEXT I
190 VTAB 18: HTAB 25: PRINT "CUB
   E"

```

Chaud, froid

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

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

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

LE PROGRAMME :

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

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

La ligne 110 regarde si vous avez gagné.

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

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

POUR JOUER :

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

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

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

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

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

EXTENSIONS POSSIBLES :

Si l'on dispose de la couleur, il sera intéressant de trouver un moyen plus visuel pour symboliser la distance, par exemple des pavés allant du rouge au bleu, aux emplacements essayés.

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


```

10  REM CHAUD-FROID
20  HOME
30  INPUT "DIMENSION DE LA GRILLE
      : ";D
40  X = INT ( RND (1) * D + 1)
50  Y = INT ( RND (1) * D + 1)
60  PRINT : PRINT
70  INPUT "DONNEZ X : ";A
80  INPUT "DONNEZ Y : ";B
90  S = S + 1
100 T = ABS (X - A) + ABS (Y -
      B)
110 IF T = 0 THEN 270
120 PRINT "          VOUS ETES ";

130 N = INT ( SQRT (D) / T) + 1
140 IF N > 5 THEN N = 5
150 IF T < = SQRT (D) THEN ON
      N GOTO 1000,190,210,230,250
160 PRINT "COMPLETEMENT GELE"
170 GOTO 70
180 GOTO 70
190 PRINT "TIEDE"
200 GOTO 70
210 PRINT "CHAUD"
220 GOTO 70
230 PRINT "TRES CHAUD"
240 GOTO 70
250 PRINT "BRULANT"
260 GOTO 70
270 HOME
280 PRINT : PRINT "VOUS AVEZ TRO
      UVE EN ";S;" COUPS."

```

Les allumettes

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : contre l'APPLE II.

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 l'APPLE II. Chacun prend à son tour autant d'allumettes qu'il le désire, dans la limite du nombre maximum autorisé.

Celui qui prend la dernière allumette est considéré comme perdant.

LE PROGRAMME :

Le jeu du joueur commence en ligne 70.

Celui de l'Apple II commence en ligne 110.

La ligne 80 contrôle la légalité du coup du joueur.

La ligne 90 gère le changement de joueur.

Le coup de l'APPLE II est calculé dans une boucle non formelle, de la ligne 120 à la ligne 150.

Le résultat est affiché par la ligne 220.

Si la partie est terminée, la ligne 230 le détecte et envoie en ligne 250, pour le message de fin.

POUR JOUER :

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

Après chaque coup, le nombre d'allumettes restantes est affiché, et l'APPLE II affiche VOUS RETIREZ lorsque c'est au joueur de prendre.

Celui qui prend la dernière allumette a perdu.

EXTENSIONS POSSIBLES :

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

```

10  REM LES ALLUMETTES
20  HOME
30  INPUT "NB.TOTAL D'ALLUMETTES
    : ";N
40  INPUT "PRISE MAXIMUM : ";I
50  INPUT "QUI COMMENCE (0=VOUS /
    1=MOI) : ";D
60  IF D = 1 THEN 110
70  INPUT "VOUS RETIREZ : ";P
80  IF P < > INT (P) OR P < 1 OR
    P > N OR P > I THEN 70
90  D = 1
100 GOTO 210
110 S = I + D
120 T = (N - S) / (I + D)
130 IF T = INT (T) THEN 160
140 S = S - D
150 GOTO 120
160 P = ABS (S - D)
170 IF P = 0 THEN P = D
180 D = 0
190 PRINT "JE RETIRE ";
200 PRINT P
210 N = N - P
220 PRINT "IL EN RESTE ";N
230 IF N = 0 THEN 250
240 GOTO 60
250 PRINT
260 IF D = 1 THEN PRINT : PRINT
    "=== J'AI GAGNE"
270 IF D < > 1 THEN PRINT : PRINT
    "=== BRAVO !"

```

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 à l'APPLE II 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 30 à 40 permettent les différentes entrées.

Les lignes 70 à 100 réalisent les impressions.

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

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

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

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

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

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

```

10  REM ALLUMETTES A PLUSIEURS
20  HOME
30  INPUT "NB.TOTAL D'ALLUMETTES
      : ";N
40  INPUT "NB.DE JOUEURS : ";J
50  F = 1
60  K = 1
70  HOME
80  PRINT "TOUR DU JOUEUR N.";K
90  PRINT "TOTAL RESTANT : ";N
100 PRINT "MAXI. A ENLEVER ";F
110 INPUT "VOTRE COUP .. ";C
120 IF C = 0 OR C > F THEN 110
130 N = N - C
140 IF N < 0 THEN N = N + C: GOTO
      110
150 F = C + C
160 IF N = 0 THEN 200
170 K = K + 1
180 IF K > J THEN K = 1
190 GOTO 70
200 PRINT : PRINT
210 PRINT "LE JOUEUR N.";K;" A G
      AGNE"

```

Calendrier perpétuel

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : gadget publicitaire.

Difficulté du jeu : aucune.

LE JEU :

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

LE PROGRAMME :

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

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

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

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

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

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

```

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

```

Bataille navale

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : réflexion.

Difficulté du jeu : facile.

LE JEU :

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

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

LE PROGRAMME :

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

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

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

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

La ligne 180 permet l'affichage de la distance.

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

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


```

10  REM BATAILLE NAVALE
20  HOME
30  PRINT "    0  1  2  3  4  5  6
      7  8  9"
40  FOR I = 0 TO 9
50  PRINT I;"  .  .  .  .  .  .
      .  .  .  .  .  ."
60  PRINT
70  NEXT I
80  S = 1
90  X = INT ( RND (1) * 10)
100 Y = INT ( RND (1) * 10)
110 VTAB 22: HTAB 1: INPUT "LIGN
    E : ";H
120 VTAB 22: HTAB 21: INPUT "COL
    ONNE : ";V
130 VTAB 2 * H + 2: HTAB 3 * V +
    4: PRINT S
140 D = INT ( SQR ((X - H) * (X -
    H) + (Y - V) * (Y - V)))
150 IF D = 0 THEN 190
160 S = S + 1
170 IF D = 1 THEN VTAB 23: HTAB
    1: PRINT "PFFFFF..."
    ": GOTO
    90
180 VTAB 23: HTAB 1: PRINT "DIST
    ANCE .... : ";D;"    ": GOTO
    110
190 HOME
200 PRINT X;" , ";Y;" TROUVE EN ";
    S;" COUPS."

```

Golf

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez facile.

LE JEU :

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

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

LE PROGRAMME :

La ligne 50 permet de définir la position du trou, et la ligne 60 dessine le terrain de jeu.

Le déplacement à réaliser est calculé en ligne 100.

La boucle I (120-170) réalise le déplacement de la balle.

La ligne 200 regarde si le coup est, ou non, gagnant ; s'il l'est, on recommence (ligne 230) jusqu'à ce que l'on ait réalisé 18 trous (boucle S (20-230) ; fin de boucle en 250, avec affichage de fin).

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

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

```

10  REM GOLF
20  FOR S = 1 TO 18
30  HOME
40  X = 1
50  T = INT (29 * RND (1) + 9)
60  VTAB 7: HTAB 1: PRINT "*****
      *****"
70  VTAB 7: HTAB T + 1: PRINT " "
      : VTAB 6: HTAB X + 1: PRINT
      "O"
80  VTAB 9: HTAB 16: PRINT "
      "
90  VTAB 9: HTAB 1: PRINT "TROU "
      ;S;: INPUT "  FORCE : ";R
100 F = INT (R * .16 - ( ABS (R)
      * ( SQR ( RND (1)) + .1) ^
      2) / 1000)
110 IF F = 0 THEN 80
120 FOR I = SGN (F) TO F STEP SGN
      (F)
130 X = X + SGN (F)
140 IF X < 40 AND X > = 2 THEN
      VTAB 6: HTAB X + 1 - SGN (
      F): PRINT " ": VTAB 6: HTAB
      X + 1: PRINT "O"
150 IF X > = 39 THEN X = 39
160 IF X < = 2 THEN X = 2
170 NEXT I
180 IF X = T THEN 210
190 E = E + 1
200 GOTO 60
210 VTAB 6: HTAB X + 1: PRINT "
      ": VTAB 7: HTAB X + 1: PRINT
      "O"
220 FOR P = 1 TO 500: NEXT P
230 NEXT S
240 HOME
250 PRINT "18 TROUS EN ";E;" COU
      PS"

```

Solfège

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion.

Difficulté du jeu : facile.

LE JEU :

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

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

LE PROGRAMME :

La boucle I (30-50) dessine une portée.

La note est choisie en ligne 60 et affichée en lignes 70-80-90.

La ligne 100 affiche le score.

La ligne 110 affiche le nom de la clé (ici clé de sol) et vous interroge.

La ligne 120 s'occupe de la fin de la partie.

Les lignes 130 et 140 effectuent le décodage de la note entrée.

La ligne 170 contrôle le résultat.

POUR JOUER :

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

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

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

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

EXTENSIONS POSSIBLES :

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

```

10 REM SOLFEGE
20 HOME
30 FOR I = 1 TO 5
40 VTAB 2 * (I + 1): HTAB 1: PRINT
   "-----"
   "
50 NEXT I
60 X = INT ( RND (1) * 11)
70 VTAB X + 1: HTAB 19: PRINT "!"
   "
80 VTAB X + 2: HTAB 19: PRINT "!"
   "
90 VTAB X + 3: HTAB 18: PRINT "O"
   "
100 VTAB 21: HTAB 1: PRINT "SCOR
    E : ";S;" SUR ";P;" ESSAIS."

110 INPUT "CLE DE SOL. QUELLE ES
    T LA NOTE : ";R$
120 IF R$ = "O" THEN END
130 Y = (R$ = "FA") + 2 * (R$ = "
    MI") + 3 * (R$ = "RE") + 4 *
    (R$ = "DO")
140 Y = Y + 5 * (R$ = "SI") + 6 *
    (R$ = "LA")
150 PRINT Y
160 P = P + 1
170 IF Y = X OR Y = X - 7 THEN S
    = S + 1
180 GOTO 20

```

Le jeu du 21

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu contre l'APPLE II.

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 l'APPLE II qui essaiera alors d'avoir un meilleur score.

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

LE PROGRAMME :

Les lignes 60 et 70 réalisent le contrôle de celui qui doit jouer (joueur ou APPLE II).

Le sous-programme (360-370) lance les dés.

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

Le total de l'APPLE II est calculé en ligne 160 et affiché par la ligne 170.

La ligne 190 contrôle la fin du jeu pour l'APPLE II.

Les lignes 220-230 affichent les résultats, et la ligne 260 cherche quel est le gagnant.

Le gain (ou la perte) sera affiché par la ligne 280, et le total par la ligne 310.

La ligne 340 contrôle la fin de partie.

POUR JOUER :

Taper RUN pour lancer le jeu.

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

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

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

EXTENSIONS POSSIBLES :

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

```

10  REM JEU DU 21
20  G = 0
30  HOME
40  T = 0
50  PRINT : INPUT "VOTRE MISE ...
    . : ";M
60  PRINT "TAPER 1 POUR TIRER": INPUT
    "      0 POUR FINIR : ";R
70  IF R = 0 THEN 120
80  GOSUB 360
90  T = T + X
100  PRINT "          VOTRE TOTAL
    EST DE ";T
110  GOTO 60
120  HOME
130  PRINT "=== LA BANQUE JOUE ..
    . ==="
140  S = 0
150  GOSUB 360
160  S = S + X
170  PRINT "  TOTAL DE LA BANQUE
    : ";S
180  FOR I = 1 TO 700: NEXT I
190  IF (S > T) OR S > 17 THEN 21
    0
200  GOTO 150
210  HOME
220  PRINT "VOUS AVEZ ";T
230  PRINT
240  PRINT "LA BANQUE A ";S
250  PRINT
260  IF S > 21 OR ((S < T) AND T <
    = 21) THEN 280
270  M = - M
280  PRINT "GAIN .... : ";M
290  PRINT
300  G = G + M
310  PRINT "VOTRE TOTAL ACTUEL :
    ";G
320  PRINT
330  INPUT "ON CONTINUE ? (O/N) "
    ;R$
340  IF R$ = "N" THEN  END
350  GOTO 30
360  X = INT ( RND (1) * 6 + 1)
370  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. L'APPLE II lance simultanément deux dés, non visibles et pose trois questions : quelle est la parité du total, quel est le total, et enfin donner la valeur de l'un des dés. On peut ensuite soit poursuivre le jeu, soit l'interrompre à tout moment.

LE PROGRAMME :

Les dés sont tirés par un sous-programme placé aux lignes 360 et 370.

La ligne 60 tient le compte du nombre de parties, et les lignes 160, 240 et 270 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 300 et les suivantes réalisent l'affichage de fin de partie : total, nombre de points, nombre de parties jouées.

La ligne 340 provoque l'arrêt si l'on répond N à la question ON CONTINUE.

POUR JOUER :

On lance le programme par l'instruction RUN.

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

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

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

EXTENSIONS POSSIBLES :

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

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


```

10  REM PAIR-IMPAIR
20  T = 0:R = 0
30  GOSUB 360
40  A = X
50  GOSUB 360
60  R = R + 1
70  S = A + X
80  HOME
90  INPUT "TOTAL PAIR (0) OU IMPA
    IR (1) : ";N
100 PRINT
110 P = S / 2 - INT (S / 2)
120 IF N = P OR N - .5 = P THEN
    150
130 PRINT "=== FAUX"
140 GOTO 170
150 PRINT "=== EXACT"
160 T = T + 1
170 PRINT
180 INPUT "TOTAL PRESUME : ";N
190 IF N = S THEN 230
200 PRINT "=== PERDU"
210 FOR I = 1 TO 900: NEXT I
220 GOTO 300
230 PRINT
240 T = T + 2
250 INPUT "BRAVO. DONNEZ UN DE :
    ";N
260 IF N < > A AND N < > (S -
    A) THEN 200
270 T = T + 3
280 PRINT
290 PRINT "**** BRAVO !": GOTO 2
    10
300 HOME
310 PRINT "VOUS AVEZ UN TOTAL DE
    ";T;" SUR ";R * 6
320 PRINT
330 INPUT "ON CONTINUE (O/N) : "
    ;R$
340 IF R$ = "N" THEN END
350 GOTO 30
360 X = INT ( RND (1) * 6 + 1)
370 RETURN

```

Devine un chiffre

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

Le tirage des dés est réalisé par un sous-programme placé en lignes 300 et 310.

L'affichage est réalisé par les lignes 130 et 140.

Les lignes 160 à 190 calculent le gain dans les diverses configurations possibles.

La totalisation est effectuée en ligne 210 pour les gains, et en ligne 240 pour les pertes.

La variable R\$, testée en ligne 280, détermine la poursuite éventuelle du jeu.

POUR JOUER :

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

EXTENSIONS POSSIBLES :

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

```

10  REM DEVINE UN CHIFFRE
20  G = 0
30  HOME
40  INPUT "VOTRE MISE ... : ";N
50  PRINT
60  INPUT "TAPER UN CHIFFRE DE 1
    A 6 : ";D
70  GOSUB 300
80  A = X
90  GOSUB 300
100 B = X
110 GOSUB 300
120 C = X
130 PRINT : PRINT : PRINT "    A
    B          C"
140 PRINT : PRINT "    ";A;"
    ";B;"          ";C
150 T = 0
160 IF (D = A) THEN T = T + N
170 IF D = B THEN T = T + N
180 IF D = C THEN T = T + N
190 IF T = 0 THEN 230
200 PRINT : PRINT : PRINT "=== V
    OUS GAGNEZ ";T
210 G = G + T
220 GOTO 250
230 PRINT : PRINT : PRINT "=== P
    ERDU !"
240 G = G - N
250 PRINT : PRINT : PRINT "TOTAL
    ACTUEL : ";G
260 PRINT
270 INPUT "ON CONTINUE (O/N) : "
    ;R$
280 IF R$ = "N" THEN END
290 GOTO 30
300 X = INT ( RND (1) * 6 + 1)
310 RETURN

```

Le jeu de Wytthof

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion (deux joueurs).

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

Les lignes 20, 30 et 40 permettent de définir une position de départ, en dehors de la diagonale.

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

La ligne 110 imprime la position du point mobile.

Le test de fin est effectué en ligne 120.

Le changement de joueur est assuré par la ligne 160.

Les directions de déplacement disponibles, 1, 2 et 3, sont décodées par trois sous-programmes qui commencent aux lignes 180, 200 et 210.

POUR JOUER :

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

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

C'est ensuite au tour du joueur 2.

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

EXTENSIONS POSSIBLES :

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

```

10 REM LE JEU DE WYTTTHOF
20 X = 3 * INT ( RND (1) * 5) +
   4
30 Y = INT ( RND (1) * 5)
40 IF X = 3 * Y + 5 THEN 30
50 HOME
60 PRINT "    0  1  2  3  4  5  6
   7  8  9"
70 FOR I = 0 TO 9
80 PRINT : PRINT I;" . . . .
   . . . . ."
90 NEXT I
100 H = 1
110 VTAB 2 * Y + 3: HTAB X: PRINT
   "+"
120 IF X = 31 AND Y = 9 THEN END

130 VTAB 23: HTAB 1: PRINT " JOU
   EUR ";H;: INPUT " - DIRECTIO
   N (1,2,3) : ";D$
140 D = VAL (D$)
150 ON D GOSUB 180,200,210
160 H = 2 - ABS (H - 1)
170 GOTO 110
180 X = X + 3
190 RETURN
200 GOSUB 180
210 Y = Y + 1
220 RETURN

```

Rebonds

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu d'adresse.

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

LE PROGRAMME :

La ligne 30 imprime les chiffres et les lettres en haut de l'écran.

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

Les lignes 60 et 70 effectuent leur impression.

Le symbole correspondant à la direction choisie est entré en ligne 80.

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

La ligne 160 imprime la nouvelle position.

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

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

Le nombre de rebonds est compté en ligne 230.

POUR JOUER :

On lance le programme par RUN.

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

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

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

EXTENSIONS POSSIBLES :

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

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

```

10 REM REBONDS
20 HOME
30 PRINT "0123456789ABCDEFGHIJKL
   MNOPQRSTUVWXYZ"
40 X = INT ( RND (1) * 18 + 3)
50 Y = INT ( RND (1) * 18 + 3)
60 VTAB X: HTAB 1: PRINT "#"
70 VTAB Y: HTAB 40: PRINT "+"
80 VTAB 23: HTAB 1: INPUT "VOTRE
   COUP : ";R$
90 IF R$ = "O" THEN 80
100 S = 0
110 T = X
120 ZZ = ( ASC (R$) - ASC ("O") +
   7 * (R$ > = "A"))
130 FOR J = 2 TO 40
140 T = T - X / ( ASC (R$) - ASC
   ("O") - 7 * (R$ > = "A"))
150 IF T > = 23 OR T < = 1 THEN
   220
160 VTAB T: HTAB J: PRINT "#"
170 NEXT J
180 IF ABS (T - Y) < 3 THEN PRINT
   "BRAVO : ";S;" REBONDS"
190 INPUT "UNE AUTRE PARTIE (O/N
   ) : ";R$
200 IF R$ = "N" THEN END
210 GOTO 10
220 X = - X
230 S = S + 1
240 GOTO 170

```

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

LE PROGRAMME :

La ligne 40 permet d'appeler le sous-programme 250 à 350 qui réalise l'affichage du terrain de jeu, le contrôle de la position des joueurs et l'affichage de cette position. Cet affichage sera d'ailleurs refait en cours de partie, pour tenir compte du déplacement réellement effectué (qui n'est pas nécessairement celui indiqué par le dé).

La ligne 50 s'occupe du changement de joueur, et la ligne 60 effectue le lancer de dé.

La ligne 110 traite le cas où l'un des joueurs termine.

La ligne 120 tire un nombre aléatoire entre 0 et 3, qui servira s'il y a lieu pour déterminer le vrai déplacement du joueur, en envoyant dans l'un des sous-programmes de "chance".

Les lignes 250 et 260 font avancer les pions, et les lignes 270 à 330 redessinent à chaque affichage le terrain de jeu en entier.

POUR JOUER :

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

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

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

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

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

On peut aussi s'arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :

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


```

10 REM JEU DE L'OEIE ALEATOIRE
20 HOME
30 X = 0:Y = 0:J = 1:I = X
40 GOSUB 250
50 J = ABS (J - 1)
60 I = INT ( RND (1) * 6 + 1)
70 IF X = 35 OR Y = 35 THEN VTAB
    21: HTAB 1: PRINT CHR$ (7);
    CHR$ (7);"=== BRAVO ...": END

80 PRINT CHR$ (7):: VTAB 12: HTAB
    1: PRINT "JOUEUR N.":J + 1;"
    - VOTRE COUP : ":I
90 PRINT : INPUT "APPUYER SUR 'R
    ETURN' : ":R$
100 GOSUB 250
110 IF X = 35 OR Y = 35 THEN VTAB
    21: HTAB 1: PRINT CHR$ (7);
    CHR$ (7);"=== BRAVO ...": END

120 I = INT ( RND (1) * 4)
130 F = INT ( RND (1) * 5 + 1)
140 ON F GOSUB 160,180,200,220,1
    50
150 GOTO 40
160 I = I + 3
170 RETURN
180 I = - I
190 RETURN
200 I = 2 * I
210 RETURN
220 IF J = 0 THEN X = INT ( RND
    (1) * 30): RETURN
230 Y = INT ( RND (1) * 30): RETURN

240 END
250 IF J = 0 THEN X = X + I * ((
    X + I) < = 35)
260 Y = Y + J * I * ((Y + I) < =
    35)
270 VTAB 3: HTAB 1: PRINT "01234
    56789ABCDEFGHIJKLMNPOQRSTUVWXYZ"
280 FOR I = 0 TO 35
290 VTAB 5: HTAB I + 1: PRINT "
    ";
300 IF I = X THEN VTAB 5: HTAB
    I + 1: PRINT "1";
310 VTAB 6: HTAB I + 1: PRINT "
    ";
320 IF I = Y THEN VTAB 6: HTAB
    I + 1: PRINT "2";
330 NEXT I
340 FOR I = 1 TO 500: NEXT I
350 RETURN

```

Roulette

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : hasard (plusieurs joueurs).

Difficulté du jeu : facile.

LE JEU :

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

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

LE PROGRAMME :

La boule K (30-90) dessine le terrain de jeu, auquel la ligne 100 rajoute les trois symboles P, 00 et I.

La boucle K (110-180) réalise le déplacement de la boule, l'arrêt étant contrôlé par les lignes 140 et 160.

La ligne 190 tire la position d'arrêt.

La ligne 210 gère les essais suivants.

POUR JOUER :

Il suffit de faire RUN, et de regarder.

Une fois la boule arrêtée, il suffira de répondre par O ou N (suivi de RETURN) à la question : AUTRE PARTIE.

EXTENSIONS POSSIBLES :

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

```

10  REM ROULETTE
20  HOME
30  FOR K = 1 TO 4
40  FOR I = 1 TO 9
50  S = I + I
60  PRINT TAB( 4 * I + 1);S * (K
    = 1) + (S + 18) * (K = 2) +
    (S - 1) * (K = 3) + (S + 17)
    * (K = 4);
70  NEXT I
80  PRINT : PRINT : PRINT : PRINT

90  NEXT K
100 VTAB 3: HTAB 1: PRINT "P": VTAB
    7: HTAB 1: PRINT "00": VTAB
    11: HTAB 1: PRINT "I"
110 FOR K = 1 TO 4
120 FOR I = 1 TO 10
130 IF I = 10 AND K < 4 THEN VTAB
    4 * K - 1: HTAB 3:: GOTO 150

140 VTAB 4 * K - 3: HTAB 4 * I
150 X = X - 1: PRINT CHR$( 7);
160 IF X = 1 THEN PRINT "*": GOTO
    210
170 NEXT I
180 NEXT K
190 X = INT ( RND (1) * 38)
200 GOTO 100
210 VTAB 21: HTAB 1: INPUT "====
    AUTRE ESSAI (O/N) : ";R$
220 IF R$ = "N" THEN END
230 GOTO 20

```

Couloirs

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

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

LE PROGRAMME :

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

La ligne 100 attend votre choix.

Le décodage du chemin est fait par l'APPLE II en utilisant trois variables logiques I, J et K, représentant les états des trois tourniquets (lignes 140 à 280).

POUR JOUER :

Après la frappe de RUN, le couloir est dessiné sur l'écran, les tourniquets étant visibles, et l'on devra taper 1 ou 2 en réponse à la question de l'APPLE II (ou 0 pour arrêter).

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 l'APPLE II.

EXTENSIONS POSSIBLES :

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

```

10  REM COULOIRS
20  HOME
30  PRINT "-----
    "
40  PRINT "B          -----  X
    2"
50  PRINT "-----  X          -----
    "
60  PRINT "A          -----  X
    1"
70  PRINT "-----
    "
80  I = 0:J = 0:K = 0
90  R$ = "":T$ = ""
100 VTAB 14: HTAB 1: INPUT "ENTR
    EE ? (1/2 0=FIN) : ";C$
110 IF C$ = "0" THEN END
120 IF C$ < > "1" AND C$ < > "
    2" THEN 100
130 R$ = R$ + C$
140 PRINT : PRINT R$
150 IF C$ = "1" THEN 200
160 J = NOT (J)
170 IF J < > 0 THEN 260
180 S$ = "B"
190 GOTO 230
200 I = NOT (I)
210 IF I < > 0 THEN 260
220 S$ = "A"
230 T$ = T$ + S$
240 VTAB 18: HTAB 1: PRINT "SORT
    IE EN ";S$: PRINT : PRINT T$

250 GOTO 100
260 K = NOT (K)
270 IF K < > 0 THEN 180
280 GOTO 220

```

Echiquier

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : graphisme.

Difficulté du jeu : variable, selon utilisation.

LE JEU :

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

LE PROGRAMME :

La ligne 40 définit deux variables A\$ et N\$ qui contiennent les éléments de dessin. La ligne 30 imprime les lettres horizontales et la première ligne du dessin.

La boucle I (50-160) permet l'impression, le motif à imprimer étant affiché ou non selon la valeur de $(I + J)/2$; les lignes 100 et 110 réalisent l'impression.

POUR JOUER :

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

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

EXTENSIONS POSSIBLES :

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

```

10 REM ECHIQUEUR
20 HOME
30 PRINT "      A  B  C  D  E  F
   G  H": PRINT "
"
40 B$ = "      ":N$ = "####"
50 FOR I = 1 TO 8
60 FOR J = 1 TO 2
70 I$ = "      ": IF J = 2 THEN I$ =
   STR$ (I) + " "
80 PRINT I$;"!";
90 FOR K = 1 TO 8
100 IF AG = 0 THEN AG = 1: PRINT
   B$;: GOTO 120
110 AG = 0: PRINT N$;
120 NEXT K
130 PRINT "!"
140 NEXT J
150 AG = NOT (AG)
160 NEXT I
170 VTAB 18: HTAB 1: PRINT "8 !#
   ##_###_###_###_!"

```

Le jeu de la fève

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : réflexion (deux joueurs).

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

LE PROGRAMME :

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

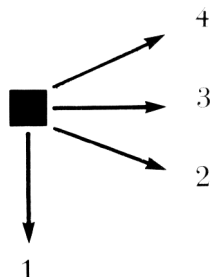
La ligne 100 réalise le test de fin de partie.

La ligne 130 permet d'exécuter l'un des sous-programmes de déplacement.

Les lignes 140 et 150 assurent le changement de joueur.

POUR JOUER :

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



Le joueur 1 entre son déplacement, et tape RETURN.

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

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

EXTENSIONS POSSIBLES :

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

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


```

10 REM LE JEU DE LA FEVE
20 HOME
30 PRINT "      0  1  2  3  4  5
      6  7  8  9"
40 FOR I = 1 TO 9
50 PRINT : PRINT I;"      .  .  .
      " " " " " " " "
60 NEXT I
70 X = 5:Y = 1
80 J = 1
90 VTAB 2 * Y + 1: HTAB X + 1: PRINT
  "#"
100 IF X = 32 AND Y = 9 THEN 170

110 VTAB 23: HTAB 1: PRINT "JOUER
    UR ";J;: INPUT " - DIRECTION
      (1/2/3/4) : ";I$
120 I = VAL (I$): PRINT CHR$ (7
    );
130 ON I GOSUB 190,210,220,240
140 J = J + 1
150 IF J = 3 THEN 80
160 GOTO 90
170 PRINT : PRINT CHR$ (7); CHR$
    (7); CHR$ (7);: PRINT "LE JO
    UEUR ";J;" A GAGNE."
180 END
190 Y = Y + 1
200 RETURN
210 GOSUB 190
220 X = X + 3
230 RETURN
240 GOSUB 220
250 Y = Y - 1
260 RETURN

```

Jackpot

REFERENCES :

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

LE JEU :

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

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

LE PROGRAMME :

Les instructions 50 à 110 permettent le dessin de la machine.

La mise sera entrée en ligne 120, et testée en ligne 130.

Les lignes 140, 160 et 180 permettent le tirage des 3 chiffres, en appelant un sous-programme qui se trouve en ligne 450.

Le résultat est affiché par la ligne 310, puis testé en ligne 320, et en ligne 420, si l'on a gagné.

POUR JOUER :

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

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

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

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

Pour terminer, il suffit de répondre F à la question CONTINUER.

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 REM JACKPOT
20 HOME
30 VTAB 5: INPUT "POT DE DEBUT .
   .. : ";P
40 HOME
50 HTAB 10: PRINT " _____
   ^"
60 HTAB 10: PRINT "! _____
   ! _#"
70 HTAB 10: PRINT "!!    !    !
   !! _#"
80 HTAB 10: PRINT "!! . ! . ! .
   !!==#"
90 HTAB 10: PRINT "!!____!____!____
   !!"
100 HTAB 10: PRINT "! _____
   _!"
110 HTAB 10: PRINT "#####
   ###"
120 VTAB 11: INPUT "VOTRE MISE :
   ";M
130 IF M < = 0 OR M > P THEN 12
   0
140 GOSUB 450
150 A = X
160 GOSUB 450
170 B = X
180 GOSUB 450
190 VTAB 1: HTAB 27: PRINT " ": VTAB
   2: HTAB 27: PRINT "^"
200 PRINT CHR$( 7);
210 VTAB 2: HTAB 27: PRINT " ": VTAB
   3: HTAB 27: PRINT "^"
220 PRINT CHR$( 7);
230 VTAB 3: HTAB 27: PRINT " ": VTAB
   4: HTAB 27: PRINT "^"
240 PRINT CHR$( 7);
250 VTAB 3: HTAB 27: PRINT "^": VTAB
   4: HTAB 27: PRINT "#"
260 PRINT CHR$( 7);
270 VTAB 2: HTAB 27: PRINT "^": VTAB
   3: HTAB 27: PRINT "#"
280 PRINT CHR$( 7);
290 VTAB 1: HTAB 27: PRINT "^": VTAB
   2: HTAB 27: PRINT "#"
300 PRINT CHR$( 7);
310 VTAB 4: PRINT TAB( 13);A; TAB(
   17);B; TAB( 21);X
320 IF A = B AND A = X THEN 410

```

```

330 N = - M
340 P = P + N: VTAB 13: PRINT "VO
    TRE GAIN : ";N;" "
350 VTAB 15: PRINT "IL VOUS REST
    E $";P;" "
360 IF P = 0 THEN 400
370 VTAB 17: PRINT "TAPER 'RETUR
    N' POUR CONTINUER : "
380 VTAB 18: INPUT "
    F' POUR TERMINER : ";R$
390 IF R$ < > "F" THEN 120
400 END
410 N = M * 40
420 IF A = 7 THEN N = M * 500
430 GOTO 340
440 END
450 X = INT ( RND (1) * 10)
460 RETURN

```

Niveau 2

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

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

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

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

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

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

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

Sur l'APPLE II il n'est pas obligatoire de dimensionner un tableau qui contient moins de 10 éléments.

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

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

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

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

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

Ainsi, si A\$ = "BONJOUR", MID\$ (A\$, 3, 1) donne le caractère N ; de plus MID\$ (A\$, 3, 3) donne la sous-chaîne "NJO" et MID\$ (A\$, 6) la sous-chaîne "UR".

Dé truqué

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

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

Le lancer de dé truqué est effectué à l'aide d'une autre boucle I (170-190), la ligne permettant le truquage se trouvant en 180. Un nouveau tirage pourra être effectué sur pression de RETURN (ligne 210).

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 la touche RETURN pour effectuer les tirages suivants. On arrête le programme en répondant F.

EXTENSIONS POSSIBLES :

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


```

10 REM DE TRUQUE
20 DIM E(6)
30 P = 0
40 HOME
50 FOR I = 1 TO 6
60 VTAB 4: HTAB 18: PRINT "
"
70 VTAB 4: HTAB 1: PRINT "PROBAB
  ILITE DU DE ";I:: INPUT " :
  ";E(I)
80 P = P + E(I)
90 IF E(I) < 0 OR E(I) > 1 OR P >
  1.1 THEN 240
100 VTAB 7: HTAB 6 * I - 5: PRINT
  "DE ";I
110 VTAB 9: HTAB 6 * I - 5: PRINT
  E(I)
120 NEXT I
130 IF P < .9 THEN PRINT CHR$
  (7)::P = 0: GOTO 30
140 VTAB 4: CALL - 868
150 FOR I = 1 TO 500: NEXT I
160 N = INT ( RND (1) * 6 + 1)
170 FOR I = 1 TO 6
180 IF N = I AND RND (1) > E(I)
  THEN 160
190 NEXT I
200 VTAB 16: HTAB 1: PRINT "TIRA
  GE : ";N
210 VTAB 20: HTAB 1: PRINT "TAPE
  R 'RETURN' POUR UN AUTRE TIR
  AGE": INPUT "          OU 'F'
  POUR ARRETER : ";R$
220 IF R$ < > "F" THEN 160
230 END
240 VTAB 5: HTAB 1: PRINT CHR$
  (7); CHR$ (7); "=== IMPOSSIB
  LE : 0 <= P < 1 ..."
250 P = P - E(I)
260 FOR J = 1 TO 1500: NEXT J
270 VTAB 5: CALL - 868
280 GOTO 70

```

Visions

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : dessin.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

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

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

POUR JOUER :

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

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 l'APPLE II deviendra de plus en plus savant. On peut aussi réaliser un jeu à deux joueurs dans lequel l'APPLE II vous demande de trouver le mot qui correspond à un certain dessin, etc.

```

10  REM VISIONS
20  HOME
30  INPUT "QUEL DESSIN : ";R$
40  FOR I = 1 TO LEN (R$)
50  N = 9 * ASC ( MID$ (R$,I,1)) -
      340
60  J = 1
70  S = N / 2 - INT (N / 2)
80  N = INT (N / 2)
90  IF S < > 0 THEN VTAB I + 5:
      HTAB J + 14: INVERSE : PRINT
      " ": NORMAL
100 J = J + 1
110 IF J < 10 THEN 70
120 NEXT I
130 PRINT : PRINT

```

421

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : lancer de dés.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

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

La ligne 110 affiche le numéro de joueur et le nombre d'essais déjà effectués.

Un sous-programme placé en ligne 360, utilisé 3 fois dans la boucle J (190-240), réalise le lancer de dés.

Les retirages sont effectués par les lignes 300, 320 et 340.

La variable R\$ gère les 2^e et 3^e tirages. L'impression du résultat est réalisée dans tous les cas par les lignes 120-130.

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

POUR JOUER :

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

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

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

Les joueurs suivants procéderont de la même façon.

EXTENSIONS POSSIBLES :

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

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

```

10  REM  4 2 1
20  HOME : INPUT "NOMBRE DE JOUEU
    RS : ";N
30  P = 1
40  I = 1
50  K = 0
60  HOME
70  R$ = "ABC"
80  GOTO 190
90  K = K + 1
100 VTAB 1: HTAB 11: PRINT "===
    4 2 1 ==="
110 VTAB 5: PRINT "JOUEUR N.";I;
    " - ESSAI N.";K;" - PARTIE N
    .";P
120 VTAB 8: PRINT TAB( 8);A; TAB(
    18);B; TAB( 28);C
130 VTAB 10: PRINT TAB( 8);"A";
    TAB( 18);"B"; TAB( 28);"C"
140 R$ = ""
150 VTAB 15: CALL - 868: INPUT
    "ABC : ";R$
160 IF R$ = "FIN" THEN END
170 IF R$ = "" THEN 260
180 R$ = R$ + "DDD"
190 FOR J = 1 TO 3
200 GOSUB 360
210 X$ = MID$(R$,J,1)
220 X = ASC (X$) - 64
230 ON X GOSUB 300,320,340,360
240 NEXT J
250 IF K < 3 THEN 90
260 I = I + 1
270 IF I < = N THEN 50
280 P = P + 1
290 GOTO 40
300 A = D
310 RETURN
320 B = D
330 RETURN
340 C = D
350 RETURN
360 D = INT ( RND (1) * 6 + 1)
370 RETURN

```

L'électricien fou

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez difficile.

LE JEU :

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

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

LE PROGRAMME :

Une table A, en ligne 20, contiendra l'état (allumé ou éteint) des 10 fenêtres.

Les lignes 50-70 et la boucle I (80-120) dessinent l'immeuble dans son état initial.

La ligne 130 regarde si tout est éteint, auquel cas l'on ira en ligne 270 afficher le nombre d'essais.

La ligne 140 attend l'indication de l'interrupteur à manœuvrer.

Les lignes 150 à 250 définissent le résultat de la manœuvre.

POUR JOUER :

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

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

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

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

EXTENSIONS POSSIBLES :

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

```

10  REM L'ELECTRICIEN FOU
20  DIM A(10)
30  M = 0
40  HOME
50  VTAB 3: PRINT " _ _ _ _ "
60  VTAB 4: PRINT " ! ! ! ! ! ! ! ! "
   PRINT " ! ! ! ! ! ! ! ! ! ! ! ! ! ! "
70  VTAB 6: PRINT " 0  1  2  3
   4  5  6  7  8  9"
80  FOR I = 0 TO 9
90  X$ = "_": IF A(I) = 0 THEN X$ =
   "#"
100 VTAB 4: HTAB I * 4 + 2: PRINT
   X$
110 T = T + A(I)
120 NEXT I
130 IF T = 10 THEN 270
140 VTAB 13: INPUT "INTERROMPTEUR
   N. ";N
150 N = N + 1
160 M = M + 1
170 IF A(N) = 1 THEN 250
180 A(N) = 1
190 R = TAN ( RND (1) + N / RND
   (1) - N) - SIN ( RND (1) /
   N) + 336 * SIN (8 * N)
200 N = INT (10 * (R - INT (R))
   )
210 IF N = 0 THEN N = INT ( RND
   (1) * 10 + 1)
220 IF A(N) = 1 THEN 250
230 A(N) = 1
240 GOTO 80
250 A(N) = 0
260 GOTO 190
270 PRINT : PRINT "=== GAGNE EN
   ";M;" ESSAIS."

```

La course au 20

REFERENCES :

Difficulté de programmation : facile.

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

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

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

La variable F, dimensionnée en ligne 70, servira à enregistrer le nombre d'étoiles prises dans chaque rangée.

La variable T contiendra le nombre total d'étoiles enlevées (ligne 60).

La ligne 120 va contrôler qu'il reste suffisamment d'étoiles sur la ligne indiquée.

Les boucles I (150-190) et J (160-180) suppriment les étoiles ; la ligne 220 teste la fin de partie.

POUR JOUER :

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

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

EXTENSIONS POSSIBLES :

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


```

10  REM LA COURSE AU 20
20  HOME
30  FOR I = 1 TO 4
40  VTAB 2 * I: PRINT I;"  *****
    *****"
50  NEXT I
60  T = 0
70  DIM F(4)
80  B = 1
90  VTAB 13: PRINT "JOUEUR .... :
    ";B
100 VTAB 15: INPUT "LIGNE .....
    ";L
110 VTAB 17: INPUT "CASES .....
    ";C
120 IF C + F(L) > 20 THEN 90
130 F(L) = F(L) + C
140 T = T + C
150 FOR I = 1 TO 4
160 FOR J = F(I) - C TO F(I)
170 VTAB 2 * I: HTAB 24 - J: PRINT
    " "
180 NEXT J
190 NEXT I
200 B = B + 1
210 IF B = 3 THEN B = 1
220 IF T < 80 THEN 90
230 PRINT "LE JOUEUR ";B;" A GAG
    NE ..."

```

Musique

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu éducatif.

Difficulté du jeu : moyenne.

LE JEU :

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

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

LE PROGRAMME :

La boucle I (30-50) dessine la portée. Une deuxième boucle I (60-100) affiche les numéros des lignes.

La ligne 110 définit le nom des notes, et la ligne 120 réalise le tirage aléatoire d'une note.

Le nom de la note est calculé à la ligne 140 et il est affiché par la ligne 150, suivi de la mention GRAVE ou AIGUE, selon le cas.

La réponse sera entrée en ligne 170.

La ligne 180 s'occupe du décodage de la réponse.

La ligne 190 affiche la note.

Les lignes 210 et 220 regardent si l'on a gagné ou perdu et affichent le message correspondant.

La ligne 230 permet de faire un nouvel essai.

POUR JOUER :

On lance le programme par RUN.

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

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

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

On doit taper la touche RETURN pour continuer, et s'arrêter en tapant F.

EXTENSIONS POSSIBLES :

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

```

10 REM MUSIQUE
20 HOME
30 FOR I = 1 TO 5
40 VTAB 2 * I + 4: HTAB 1: PRINT
   "-----"
   "-----"
50 NEXT I
60 FOR I = 1 TO 14
70 VTAB I + 2
80 IF I < = 9 THEN PRINT I;" "

90 IF I > 9 THEN PRINT CHR$ (I
   + 55);" "
100 NEXT I
110 N$ = "SI LA SOLFA MI RE DO"
120 X = INT ( RND (1) * 14 + 1)
130 T = X - 1 - 7 * INT ((X - 1)
   / 7)
140 VTAB 17: PRINT "-- NOTE : ";
   MID$ (N$,3 * T + 1,3);
150 IF X < = 7 THEN PRINT " AI
   GUE"
160 IF X > 7 THEN PRINT " GRAVE
   "
170 VTAB 19: INPUT "NUMERO DE LA
   LIGNE : ";L$
180 L = ( ASC (L$) - 48) * (L$ <
   ":") + ( ASC (L$) - 55) * (L
   $ > = "A")
190 VTAB X: HTAB 15: PRINT "!": VTAB
   X + 1: HTAB 15: PRINT "!": VTAB
   X + 2: HTAB 14: PRINT "O"
200 VTAB 19: HTAB 26
210 IF L = X THEN PRINT "BRAVO
   !"
220 IF L < > X THEN PRINT "PER
   DU ..."
230 VTAB 21: PRINT "TAPER 'RETUR
   N' POUR CONTINUER": INPUT "
   OU 'F' POUR ARRETER :
   ";R$
240 IF R$ < > "F" THEN 20

```

Poker

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu contre l'APPLE II.

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, l'APPLE II.

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 de l'APPLE II de jouer.

LE PROGRAMME :

Les lignes 30 et 40 définissent les cartes. Une variable N est dimensionnée à 16, en ligne 20, pour recevoir 10 cartes joueur et 5 cartes APPLE.

Le tirage sera effectué par une boucle I (60-120) et le contrôle par une boucle J (90-110).

L'affichage des cartes est réalisé par le sous-programme 240-350.

Une autre boucle I (170-190) réalise le deuxième tirage du joueur, la ligne 230 permettant le deuxième affichage, et la boucle I (240-310) où l'on entre directement après, réalisera l'affichage du jeu machine.

POUR JOUER :

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

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

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

Il suffira ensuite de faire RETURN pour démarrer une autre partie et F pour arrêter.

EXTENSIONS POSSIBLES :

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

```

10  REM POKER
20  DIM N(16)
30  V$ = "7 8 9 10VADARQAS"
40  C$ = "TRCACOFI"
50  HOME
60  FOR I = 1 TO 16
70  N(I) = INT ( RND (1) * 32)
80  IF I = 1 THEN 120
90  FOR J = 1 TO I - 1
100  IF N(I) = N(J) THEN 70
110  NEXT J
120  NEXT I
130  K = 0:P = 0
140  VTAB 8: PRINT "CARTES REFUSE
    ES"
150  GOSUB 240
160  IF R$ = "0" THEN END
170  FOR I = 1 TO LEN (R$)
180  N( VAL ( MID$ (R$,I,1))) = N(
    I + 5)
190  NEXT I
200  HOME
210  P = 1
220  GOSUB 240
230  K = K + 10
240  FOR I = 1 TO 5
250  J = 2 * (N(I + K) - 4 * INT
    (N(I + K) / 4) + 1) - 1
260  F = 2 * (N(I + K) - 7 * INT
    (N(I + K) / 7) + 1) - 1
270  R$ = MID$ (V$,F,2)
280  VTAB K + 1: HTAB 7 * I - 3: PRINT
    I
290  VTAB K + 3: HTAB 7 * I - 4: PRINT
    R$;" ":" MID$ (C$,J,2)
300  VTAB K + 4: HTAB 7 * I - 4: PRINT
    "-----"
310  NEXT I
320  IF P = 1 THEN P = 0: RETURN

330  IF K = 10 THEN GOTO 360
340  VTAB 8: HTAB 17: INPUT " : "
    ;R$
350  RETURN
360  VTAB 21: PRINT "TAPER 'RETUR
    N' POUR CONTINUER": INPUT "
    OU 'F' POUR ARRETER :
    ";R$
370  IF R$ < > "F" THEN 50

```

Le jeu de saute-mouton

REFERENCES :

Difficulté de programmation : facile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

La boucle I (50-80) initialise une table A (I), qui contient les positions des moutons. La boucle I (100-150) dessine le terrain de jeu et les moutons. Les lignes 170 et 180 s'occupent du texte et des entrées. La ligne 190 contrôle la validité du coup. Le changement de joueur est réalisé en ligne 220.

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

On commencera par rajouter un contrôle du joueur gagnant.

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

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

```

10  REM LE JEU DE SAUTE-MOUTON
20  HOME
30  DIM A(8)
40  A(1) = 0:A(8) = 0
50  FOR I = 1 TO 3
60  A(I + 1) = 1
70  A(I + 4) = - 1
80  NEXT I
90  H = 1
100 FOR I = 1 TO 8
110  VTAB 9: HTAB 4 * I + 1: PRINT
    I
120  VTAB 7: HTAB 4 * I + 1: PRINT
    " "
130  IF A(I) = 1 THEN  VTAB 7: HTAB
    4 * I + 1: PRINT "O"
140  IF A(I) = - 1 THEN  VTAB 7:
    HTAB 4 * I + 1: PRINT "*"
150  NEXT I
160  J = 2 - (H = 1)
170  VTAB 14: PRINT "JOUEUR N.":J

180  VTAB 16: INPUT "---- DEBUT, F
    IN ... : ";D,F
190  IF A(D) < > H OR A(F) < >
    0 THEN 180
200  A(D) = 0
210  A(F) = H
220  H = - H
230  GOTO 100

```

Lettres

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

La règle est fortement inspirée d'un jeu de lettres très connu en France, pratiqué notamment dans une émission télévisée (les chiffres et les lettres). L'APPLE II 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 (70-110) choisit dans l'alphabet entre 7 et 12 lettres, en fonction de la variable aléatoire L, tirée en ligne 40. Pour éviter d'avoir trop de consonnes, on utilise en ligne 90 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 110-150 demande à chacun des joueurs la longueur de son mot. Les lignes 160 et 170 comparent les résultats et totalisent le score de chaque joueur, affiché par la ligne 180.

POUR JOUER :

On lance l'exécution du jeu avec RUN.

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

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

On devra faire F pour finir.

EXTENSIONS POSSIBLES :

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

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


```

10  REM LETTRES
20  S = 0:J = 0
30  HOME
40  L = INT ( RND (1) * 6 + 7)
50  A$ = "AEIOUEA"
60  VTAB 7: PRINT "LETTRES : "
70  FOR I = 1 TO L
80  VTAB 7: HTAB 9 + 2 * I: PRINT
      CHR$ ( INT ( RND (1) * 26 +
      65))
90  IF RND (1) < .4 THEN VTAB 7
      : HTAB 9 + 2 * I: PRINT MID$
      (A$, INT ( RND (1) * 7 + 1),
      1)
100 NEXT I
110 I = 1
120 VTAB 11: CALL - 868: PRINT
      "JOUEUR N.":I:: INPUT " - LO
      NGUEUR : ";L
130 IF I = 1 THEN P = L
140 I = I + 1
150 IF I < 3 THEN 120
160 S = S + P * (P > L)
170 J = J + L * (L > P)
180 VTAB 14: PRINT "=SCORES= JOU
      EUR 1 : ";S;" JOUEUR 2 : "
      ;J
190 VTAB 21: PRINT "TAPER 'RETUR
      N' POUR CONTINUER": INPUT "
      OU 'F' POUR ARRETER :
      ";R$
200 IF R$ < > "F" THEN 30

```

Espion

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : jeu de réflexion.

Difficulté du jeu : facile.

LE JEU :

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

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

LE PROGRAMME :

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

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

La boucle I (80-150) est utilisée soit pour effectuer le codage, soit pour effectuer le décodage.

La ligne 100 appelle le générateur aléatoire.

La ligne 110 réalise la transposition des lettres.

Les lignes 120 et 130 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 RETURN.

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

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

EXTENSIONS POSSIBLES :

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

```

10  REM ESPION
20  HOME
30  INPUT "TAPER VOTRE CODE : ";N

40  A = 1 / N
50  PRINT : INPUT "TAPER VOTRE ME
    SSAGE : ";A$
60  PRINT : INPUT "(C)ODAGE OU (D
    )ECODAGE ? (C/D) : ";R$
70  B$ = ""
80  FOR I = 1 TO LEN (A$)
90  T = ASC ( MID$ (A$,I,1))
100  GOSUB 180
110  T = T - A * ((R$ = "C") - (R$
    = "D"))
120  IF T > 90 THEN T = T - 26
130  IF T < 65 THEN T = T + 26
140  B$ = B$ + CHR$ (T)
150  NEXT I
160  PRINT : PRINT : PRINT "RESUL
    TAT : ";B$
170  END
180  A = (A + 3.14159) ^ B
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 20 et 30 dessinent la locomotive, la ligne 40 le wagon, et les roues sont en ligne 50.

La position initiale de la locomotive est définie par les lignes 70 et 80, celle du wagon par les lignes 90 et 110.

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

Les lignes 130 et 140 réalisent l'affichage du wagon et les lignes 150, 160 et 170 celui de la locomotive.

Le test de fin est en ligne 180.

Les mouvements à effectuer sont entrés dans une variable alphanumérique R\$, en ligne 190, et sont décodés dans une boucle I (220-250) pour être exécutés en ligne 260.

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 DDDHHHBHCG, suivie de RETURN.

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

On obtiendra alors l'affichage du score.

EXTENSIONS POSSIBLES :

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

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

```

10  REM ACCROCHEZ LES WAGONS
20  D$ = "  *"
30  A$ = "S=!"
40  B$ = "!_!"
50  C$ = "O O"
60  N = 0
70  LY = INT ( RND (1) * 33 + 1)
80  LX = INT ( RND (1) * 20 + 2)
90  WY = INT ( RND (1) * 33 + 1)
100 WX = INT ( RND (1) * 20 + 2)

110 IF ABS (LX - WX) < 4 THEN 7
    0
120 HOME
130 VTAB WX: HTAB WY: PRINT B$
140 VTAB WX + 1: HTAB WY: PRINT
    C$
150 VTAB LX - 1: HTAB LY: PRINT
    D$
160 VTAB LX: HTAB LY: PRINT A$
170 VTAB LX + 1: HTAB LY: PRINT
    C$
180 IF LX = WX AND WY = LY - 3 THEN
    280
190 VTAB 22: INPUT "CHEMIN A SUI
    VRE (DGGHBB...) ";R$
200 X = 0:Y = 0
210 N = N + 1
220 FOR I = 1 TO LEN (R$)
230 X = X + ( MID$ (R$,I,1) = "B"
    AND LX < 23) - ( MID$ (R$,I
    ,1) = "H" AND LX > 1)
240 Y = Y + ( MID$ (R$,I,1) = "D"
    AND LY < 38) - ( MID$ (R$,I
    ,1) = "G" AND LX > 1)
250 NEXT I
260 LX = LX + X:LY = LY + Y
270 GOTO 120
280 VTAB 22: PRINT "BRAVO ... TR
    OUVRE 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 (60-130) avec toutes les vérifications de validité nécessaires. Le nombre de tirages demandés est introduit en ligne 150.

La boucle J (160-250) comptabilise les tirages.

La boucle I (190-210) assure la conformité de chaque tirage avec les probabilités introduites en début de jeu.

Les résultats sont comptabilisés en ligne 220, et affichés grâce aux lignes 230 et 240.

POUR JOUER :

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

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

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

Les résultats sont affichés sous les numéros des tirages, alors que le nombre total de tirages décroît au fur et à mesure.

EXTENSIONS POSSIBLES :

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

```

10  REM DE TRUQUE
20  DIM E(6)
30  DIM T(6)
40  P = 0
50  HOME
60  FOR I = 1 TO 6
70  VTAB 4: HTAB 18: PRINT "
    "
80  VTAB 4: HTAB 1: PRINT "PROBAB
    ILITE DU DE ";I;: INPUT " :
    ";E(I)
90  P = P + E(I)
100 IF E(I) < 0 OR E(I) > 1 OR P
    > 1.1 THEN 270
110 VTAB 7: HTAB 6 * I - 5: PRINT
    "DE ";I
120 VTAB 9: HTAB 6 * I - 5: PRINT
    E(I)
130 NEXT I
140 IF P < .9 THEN 40
150 PRINT : INPUT "NB. DE TIRAGE
    S ... : ";P
160 FOR J = 1 TO P
170 VTAB 11: HTAB 22: PRINT P -
    J;" "
180 N = INT ( RND (1) * 6 + 1)
190 FOR I = 1 TO 6
200 IF N = I AND RND (1) > E(I)
    - INT (E(I)) THEN 180
210 NEXT I
220 T(N) = T(N) + 1
230 VTAB 16: HTAB 6 * N - 3: PRINT
    N
240 VTAB 18: HTAB 6 * N - 3: PRINT
    T(N)
250 NEXT J
260 END
270 VTAB 5: PRINT "=== IMPOSSIBL
    E : 0 <= P < 1"
280 P = P - E(I)
290 FOR J = 1 TO 1700: NEXT J
300 VTAB 5: CALL - 868
310 GOTO 80

```

Tour de cartes

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de prestidigitation.

Difficulté du jeu : facile.

LE JEU :

C'est un tour assez facile mais, ici, les cartes sont manipulées sous vos yeux par l'APPLE II 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 à l'APPLE II dans quelle colonne (1^{re}, 2^e ou 3^e à partir de la gauche) se trouve la carte choisie ; les cartes sont alors redistribuées et le joueur indique de nouveau la colonne correspondant à sa carte ; ceci est fait encore une fois et, au 4^e affichage, l'APPLE II est en mesure de deviner la carte choisie.

LE PROGRAMME :

La boucle I (40-100) permet de choisir, au hasard, 21 cartes sur 52, et la boucle emboîtée J (60-80) permet de s'assurer que l'on ne réalise pas deux fois le même tirage. La ligne 120 contient les couleurs des cartes ainsi que les noms des cartes figures (VA, DA, RO, AS), dans une variable alpha-numérique B\$.

La boucle I (130-220) réalise l'affichage du jeu.

La ligne 140 décode la couleur de la carte.

La ligne 150 choisit la position d'affichage.

La ligne 160 décode la hauteur.

L'affichage proprement dit est effectué de la ligne 170 à la ligne 210.

La ligne 240 permet au joueur d'entrer le numéro de la colonne où se trouve sa carte. La boucle J (280-300) réalise le brouillage, et les boucles emboîtées J (330-370) et I (340-360) fabriquent la nouvelle variable A\$ à utiliser pour le prochain affichage.

POUR JOUER :

On lance le programme par RUN.

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

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

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

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

On sortira du programme en tapant CNT et C à la place du numéro de la colonne.

EXTENSIONS POSSIBLES :

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

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


```

10 REM TOUR DE CARTES
20 HOME
30 A$ = ""
40 FOR I = 1 TO 21
50 B$ = CHR$ ( INT ( RND (1) * 5
      2 + 38))
60 FOR J = 1 TO LEN (A$)
70 IF B$ = MID$ (A$,J,1) THEN 5
      0
80 NEXT J
90 A$ = A$ + B$
100 NEXT I
110 H = 3:P = 1
120 B$ = "TRCACDFIVADARQAS"
130 FOR I = 1 TO LEN (A$) * P -
      11 * (P = 0)
140 S = INT (( ASC ( MID$ (A$,I,
      1)) - 38) / 13)
150 J = 7 * INT ((I - 1) / 7)
160 T = ASC ( MID$ (A$,I,1)) - 3
      7 - 13 * S
170 IF P = 0 THEN VTAB 1: PRINT
      "VOTRE CARTE EST : "
180 VTAB (I - J) * P + 1: HTAB J
      * P + INT (J / 2) + 16 * (
      P = 0) + 1
190 IF T < = 9 THEN PRINT T +
      1
200 IF T > 9 THEN PRINT MID$ (
      B$, ABS (2 * T - 11),2)
210 VTAB (I - J) * P + 1: HTAB (
      J + 3) * P + 2 + INT (J / 2
      ) + 21 * (P = 0): PRINT MID$
      (B$,2 * S + 1,2)
220 NEXT I
230 IF P = 0 THEN END
240 VTAB 17: INPUT "-- QUELLE CO
      LONNE ? (1 A 3) : ";N
250 HOME
260 H = H - 1
270 R$ = "20120":B$ = ""
280 FOR J = 0 TO 2

```

```

290 B$ = B$ + MID$ (A$,7 * VAL
    ( MID$ (R$,N + J,1)) + 1,7)
300 NEXT J
310 IF H = 0 THEN A$ = B$:F = 0:
    GOTO 120
320 A$ = ""
330 FOR J = 0 TO 2
340 FOR I = 1 TO 7
350 A$ = A$ + MID$ (B$,3 * I - 2
    + J,1)
360 NEXT I
370 NEXT J
380 GOTO 120

```

L'indésirable

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : hasard et réflexion (1 joueur).

Difficulté du jeu : moyenne.

LE JEU :

Vous êtes le propriétaire d'un immeuble de 16 pièces vides, et vous n'arrivez pas à mettre la main sur votre locataire indésirable. Vous visitez les pièces au hasard, mais, si vous visitez la pièce voisine, votre locataire, prévenu par le bruit, va déménager. Il peut même se réinstaller dans une pièce que vous avez déjà visitée, et marquée comme vide. Il vous faudra arriver à le déloger en moins de 9 coups pour gagner.

LE PROGRAMME :

Une table B est dimensionnée à 16, en ligne 20, pour recevoir les numéros des cases déjà visitées.

La boucle I (40-120) dessine le terrain de jeu.

La position initiale du locataire est tirée au hasard en ligne 130.

Le numéro de la case à visiter est donné en ligne 140.

Les lignes 150 à 280 contrôlent les diverses réponses possibles et dessinent le nouvel état de l'immeuble.

Les lignes 270 et 280 assurent le déplacement du locataire.

POUR JOUER :

On lance l'exécution par RUN.

A la question CASE A VISITER, on répond en tapant une lettre, de A à P. Si on a trouvé le locataire, celui-ci est affiché, et la partie est terminée ; sinon, deux cas peuvent se produire : soit l'on est suffisamment loin, et le locataire ne bouge pas, soit l'on est à une case, il apparaît un court instant, et il se recache, aléatoirement, dans une case voisine. Les cases déjà visitées se colorent en rouge, et vous ne pouvez pas y revenir. Si, au bout de 9 coups, vous n'avez pas trouvé, la position du locataire s'affiche, et vous avez perdu.

EXTENSIONS POSSIBLES :

On pourra, bien sûr, jouer sur plus de 16 cases, utiliser des caractères graphiques pour représenter le locataire et le propriétaire, utiliser un algorithme plus subtil pour le déplacement du locataire, lui interdire éventuellement de revenir dans des cases déjà visitées, etc.

```

10 REM L'INDESIRABLE
20 DIM B(16)
30 HOME
40 FOR I = 0 TO 3
50 PRINT
60 FOR J = 1 TO 4: PRINT "  ";;
  INVERSE : PRINT "  ";; NORMAL
70 NEXT J: PRINT
80 FOR J = 1 TO 4: PRINT "  ";; INVERSE :
  PRINT "  ";; NORMAL : PRINT CHR$
    (64 + I * 4 + J) ;; INVERSE :
    PRINT "  ";; NORMAL
90 NEXT J: PRINT
100 FOR J = 1 TO 4: PRINT "  ";;
  INVERSE : PRINT "  ";; NORMAL
  : NEXT J: PRINT
110 PRINT
120 NEXT I
130 X = INT ( RND (1) * 16 + 1):
  S = 1:K = 0
140 VTAB 21: INPUT "CASE A VISIT
  ER : ";C$: IF LEN (C$) < >
  1 THEN 140
150 C = ASC (C$) - 64: IF C < 1 OR
  C > 16 THEN 140
160 IF B(C) = 1 THEN VTAB 22: INVERSE :
  PRINT CHR$ (7); CHR$ (7); "===
  CASE DEJA VISITEE ! ===": NORMAL
  : FOR I = 1 TO 1300: NEXT I:
  VTAB 22: CALL - 868: GOTO
  140
170 IF C = X THEN VTAB 22: INVERSE
  : PRINT CHR$ (7); CHR$ (7);
  "=== BRAVO. GAGNE EN ";S;" C
  OUPS ! ===": NORMAL :K = 1: GOTO
  250
180 D = ABS (X - C): IF D = 1 OR
  D = 3 OR D = 4 OR D = 5 THEN
  250
190 VTAB 22: INVERSE : PRINT CHR$
  (7); CHR$ (7); "=== IL N'EST
  PAS LA ! ===": NORMAL
200 S = S + 1:B(C) = 1:
210 I = INT ((C - 1) / 4):J = C -
  1 - I * 4

```

```

220 VTAB I * 5 + 2: HTAB J * 7 +
3: PRINT "-----": VTAB I * 5
+ 3: HTAB J * 7 + 3: PRINT
"(( " + C$ + " ))": VTAB I * 5
+ 4: HTAB J * 7 + 3: PRINT
"-----"
230 IF S = 9 THEN VTAB 21: CALL
- 868: INVERSE : PRINT "===
VOUS AVEZ PERDU ===": PRINT
"=== IL ETAIT EN "; CHR$ (X
+ 64); " ===": NORMAL :K =
1: GOTO 250
240 FOR I = 1 TO 600: NEXT I: VTAB
22: CALL - 868: GOTO 140
250 I = INT ((X - 1) / 4):J = X -
1 - I * 4
260 IF K = 1 THEN INVERSE
270 VTAB I * 5 + 2: HTAB J * 7 +
3: PRINT "=====": VTAB I * 5
+ 3: HTAB J * 7 + 3: PRINT
"! " + CHR$ (X + 64) + " !"
: VTAB I * 5 + 4: HTAB J * 7
+ 3: PRINT "====="
280 IF K = 1 THEN NORMAL : VTAB
23: END
290 Z = INT ( RND (1) * 10 + 1) -
5: IF ABS (Z) = 2 THEN 290
300 X = X + Z: IF X > 16 OR X < 1
THEN X = X - Z: GOTO 290
310 GOTO 200

```

Le jeu de Marienbad

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : logique (2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

LE PROGRAMME :

La ligne 60 dimensionne une variable A (I) qui contiendra le nombre d'allumettes de chaque tas.

La boucle I (70-100) permet d'entrer ce nombre.

La boucle I (130-170) imprime les tas.

La ligne 180 assure l'affichage du joueur.

La ligne 200, ainsi que la ligne 210, contrôlent si le coup est autorisé.

La ligne 280 contrôle la fin de la partie.

POUR JOUER :

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

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

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

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

EXTENSIONS POSSIBLES :

On peut désigner chaque joueur par son nom, faire un graphisme plus élaboré, écrire un programme permettant à l'APPLE II de servir de partenaire, etc.

```

10 REM LE JEU DE MARIENBAD
20 HOME
30 INPUT "NB. DE TAS ? ( <= 7 )
   : ";N
40 IF N > 8 OR N < 1 THEN PRINT
   CHR$(7);: GOTO 20
50 S = 0
60 DIM A(N)
70 FOR I = 1 TO N
80 PRINT : PRINT "TAS ";I;: INPUT
   " - NB.D'ALLUMETTES ... : ";
   A(I)
90 S = S + A(I)
100 NEXT I
110 H = 1
120 HOME
130 VTAB 7: PRINT "TAS"
140 FOR I = 1 TO N
150 VTAB 5: HTAB 5 * I + 3: PRINT
   A(I)
160 VTAB 7: HTAB 5 * I + 3: PRINT
   I
170 NEXT I
180 VTAB 11: PRINT "=== JOUEUR
   ";H;" ==="
190 VTAB 13: INPUT "QUEL TAS ...
   : ";T
200 IF T < 1 OR T > N THEN PRINT
   CHR$(7);: GOTO 190
210 IF A(T) = 0 THEN PRINT CHR$(
   7);: GOTO 190
220 VTAB 15: INPUT "QUANTITE ...
   : ";Q
230 IF Q < 1 OR Q > A(T) THEN PRINT
   CHR$(7);: GOTO 220
240 A(T) = A(T) - Q
250 S = S - Q
260 H = H + 1
270 IF H = 3 THEN H = 1
280 IF S < 0 THEN GOTO 120
290 VTAB 17: INVERSE : PRINT "=="
   =
   ==="
300 VTAB 18: PRINT CHR$(7) + CHR$(
   7);: PRINT "=== LE JOUEUR "
   ;H;" A GAGNE ==="
310 VTAB 19: PRINT "===
   ===": NORMAL

```

Compose

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : composition musicale.

Difficulté du jeu : aucune, a priori.

LE JEU :

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

LE PROGRAMME :

La boucle I (30-50) dessine la portée. La variable N\$, en ligne 60, contient les noms des notes.

La deuxième boucle I (70-120) permet de faire le choix d'une note de façon aléatoire, à la ligne 80 ; son nom est calculé en ligne 90 et la note est dessinée par la ligne 100.

La ligne 130 permet de recommencer en tapant RETURN.

POUR JOUER :

Lancer l'exécution du programme en tapant RUN.

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

Une pression sur la touche RETURN provoquera l'affichage d'une nouvelle série de notes, et une pression sur la touche F provoque la fin de la partie.

EXTENSIONS POSSIBLES :

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


```

10 REM COMPOSE
20 HOME
30 FOR I = 1 TO 5
40 VTAB 2 * I + 4: HTAB 1: PRINT
   "-----"
   "
50 NEXT I
60 N$ = "SI LA SO FA MI RE DO "
70 FOR I = 1 TO 13
80 X = INT ( RND (1) * 14 + 1)
90 D$ = MID$ (N$,3 * (X - 7 * (X
   > 7)) - 2,3)
100 VTAB X: HTAB 3 * I - 1: PRINT
   "!": VTAB X + 1: HTAB 3 * I -
   1: PRINT "!": VTAB X + 2: HTAB
   3 * I - 2: PRINT "O"
110 VTAB 19: HTAB 3 * I - 2: PRINT
   D$
120 NEXT I
130 VTAB 21: PRINT "TAPER 'RETUR
   N' POUR CONTINUER": INPUT "
   OU 'F' POUR ARRETER :
   ";R$
140 IF R$ < > "F" THEN 20

```

Jour de la semaine

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : utilitaire.

Difficulté du jeu : facile.

LE JEU :

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

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

LE PROGRAMME :

Les lignes 20 à 50 contrôlent les entrées. La ligne 80 permet une correction pour les dates antérieures à 1582 (mise en service du calendrier actuel).

Les lignes 90-100 permettront l'affichage en clair du mois dans la réponse (sur 4 lettres).

La formule utilisée apparaît aux lignes 120 et 130.

Les lignes 140 et 150 permettront l'affichage en clair du jour dans la réponse. La ligne 160 imprime les résultats.

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 RETURN, le mois (entre 1 et 12) suivi de RETURN, et l'année (sous la forme 1845, par exemple) suivie de RETURN.

Le résultat est très rapidement affiché.

On tape RETURN pour recommencer et F pour arrêter.

EXTENSIONS POSSIBLES :

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

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

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

```

10 REM JOUR DE LA SEMAINE
20 HOME
30 INPUT "JOUR ..... : ";J
40 INPUT "MOIS ..... : ";M
50 INPUT "ANNEE ..... : ";A
60 B = A
70 S = INT (A / 100) - INT (A /
    400)
80 IF A < = 1582 THEN S = 2
90 M$ = "JANVIER FEVRIER MARS
    AVRIL MAI JUIN
    JUILLET AOÛT SEPTEMBR
    EOCTOBRE NOVEMBRE DECEMBRE
    "
100 M$ = MID$ (M$,9 * M - 8,9)
110 IF M < 3 THEN GOSUB 200
120 S = J - 1 + INT (13 * (M + 1
    ) / 5) + INT (5 * A / 4) -
    S
130 S = S - 7 * INT (S / 7)
140 J$ = "DIMANCHELUNDI MARDI
    MERCREDIJEUDI VENDREDISAM
    EDI DIMANCHE"
150 J$ = MID$ (J$,8 * S + 1,8)
160 VTAB 16: PRINT "LE ";J;" ";M
    $;" ";B;" ETAIT UN ";J$
170 VTAB 21: PRINT "TAPER 'RETUR
    N' POUR CONTINUER": INPUT "
    OU 'F' POUR ARRETER :
    ";R$
180 IF R$ < > "F" THEN 20
190 END
200 M = M + 12
210 A = A - 1
220 RETURN

```

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

LE PROGRAMME :

Deux tables (B et C) sont utilisées en ligne 50, pour enregistrer l'état de la population. Les boucles F (230-310) et G (240-300) sont utilisées en début de jeu pour définir la population initiale (lignes 250 et 260). Le calcul de la nouvelle population est ensuite réalisé à l'aide d'une quadruple boucle : 80-210, 90-200, 110-160, 120-150. La ligne 130 réalise la gestion du nouvel état suivant les règles.

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

La ligne 320 permet de recommencer.

POUR JOUER :

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

EXTENSIONS POSSIBLES :

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

```

10  REM JEU DE LA VIE (AUTO)
20  HOME
30  INPUT "----  DIMENSION ( <=10
      ) : ";A
40  IF A < 1 OR A > 10 THEN PRINT
      CHR$(7): GOTO 20
50  DIM B(A,A),C(A,A)
60  I = 0
70  GOTO 220
80  FOR F = 1 TO A
90  FOR G = 1 TO A
100 S = 0
110 FOR I = - 1 TO 1
120 FOR J = - 1 TO 1
130 IF F + J = 0 OR G + I = 0 OR
      F + J = A + 1 OR G + I = A +
      1 THEN 150
140 S = S + B(F + J,G + I)
150 NEXT J
160 NEXT I
170 S = 2 * S - B(F,G)
180 C(F,G) = 0
190 IF S > = 5 AND S < = 7 THEN
      C(F,G) = 1
200 NEXT G
210 NEXT F
220 HOME
230 FOR F = 1 TO A
240 FOR G = 1 TO A
250 IF I < > 0 THEN 270
260 IF RND (1) < .5 THEN C(F,G)
      = 1
270 VTAB 2 * F - 1: HTAB 3 * G: IF
      C(F,G) = 1 THEN PRINT "*"
280 B(F,G) = C(F,G)
290 NEXT G
300 NEXT F
310 GOTO 80

```

```

10  REM JEU DE LA VIE (MANUEL)
20  HOME
30  INPUT "----  DIMENSION ( <=10
    ) : ";A
40  IF A < 1 OR A > 10 THEN PRINT
    CHR$(7): GOTO 20
50  DIM B(A,A),C(A,A)
60  I = 0
70  GOTO 220
80  FOR F = 1 TO A
90  FOR G = 1 TO A
100 S = 0
110 FOR I = - 1 TO 1
120 FOR J = - 1 TO 1
130 IF F + J = 0 OR G + I = 0 OR
    F + J = A + 1 OR G + I = A +
    1 THEN 150
140 S = S + B(F + J,G + I)
150 NEXT J
160 NEXT I
170 S = 2 * S - B(F,G)
180 C(F,G) = 0
190 IF S > = 5 AND S < = 7 THEN
    C(F,G) = 1
200 NEXT G
210 NEXT F
220 HOME
230 VTAB 22: PRINT "GENERATION N
    .";GE:GE = GE + 1
240 FOR F = 1 TO A
250 FOR G = 1 TO A
260 IF I < > 0 THEN 300
270 VTAB 22: HTAB 1: PRINT "LIGN
    E ";F;" COLONNE ";G;" TAPE
    R 1 OU 0"
280 VTAB 2 * F - 1: HTAB 3 * G: INPUT
    " ";R$: IF R$ < > "0" AND R$
    < > "1" THEN PRINT CHR$(
    7): GOTO 280
290 C(F,G) = VAL (R$): IF R$ = "
    0" THEN VTAB 2 * F - 1: HTAB
    3 * G: PRINT " "
300 VTAB 2 * F - 1: HTAB 3 * G: IF
    C(F,G) = 1 THEN PRINT "*"
310 B(F,G) = C(F,G)
320 NEXT G
330 NEXT F
340 GOTO 80

```

Niveau 3

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Machinamot	Laisser l'ordinateur créer des vocables nouveaux.
Plus grand, plus petit	Un jeu de déduction intéressant, pour un joueur.
Ordre	Un peu d'astuce pour remettre en ordre, le plus vite possible.
Le pendu	Un jeu de lettres très classique (2 joueurs).
Belote	Pour simuler une donne, avec un jeu de 32 cartes.
Un taquin de calcul	Un bon entraînement (2 joueurs).
Pendu inversé	Cette fois, c'est l'ordinateur qui essaie de deviner le mot.
Nicomaque	Pour se perfectionner en calcul mental (1 ou plusieurs joueurs).
Course de chevaux	Assez visuel : un jeu de hasard, pour plusieurs joueurs.
A travers	Un jeu de stratégie pour 2 joueurs.
Le loup et les agneaux	Un jeu de réflexion pour 2 joueurs.
Mastermind	Avec des chiffres, mais très classique. Un seul joueur.
Idem	Un jeu de calcul mental, avec plusieurs niveaux, pour plusieurs joueurs.
Horloge 1	Ne jetez pas votre vieille télévision : transformez-la en horloge.
Reines sur l'échiquier	Un vieux problème, pour vous aider à réfléchir.
Répétitions	Retrouver une séquence de chiffres de plus en plus longue.
Mots	Une idée très simple, mais ce n'est pas si facile de trouver !
Permutations	Un jeu de réflexion, pas toujours évident.
Bridge	Mettez au point vos propres problèmes de bridge.
Tirage de dés	Suffisamment souple pour être utilisé partout.
Hanoi	Un conseil : commencez petit ! ! !

Manipulation des chaînes et traitement de données multiples

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

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

Ainsi si,

A\$ = "MONSIEUR"

on aura :

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

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

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

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

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

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

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

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

$$A\$ = \text{MID\$}(\text{LEFT\$}(" " + A\$, I) + \text{"nouvelle lettre"} + \text{MID\$}(A\$, I + 1), 2).$$

TRAITEMENT DE DONNEES MULTIPLES

INSTRUCTIONS READ, DATA, RESTORE

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

Ainsi, on pourra écrire :

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

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

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

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

LE PROGRAMME :

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

La boucle I (50-70) lit les lignes DATA (140-160) concernant les préfixes suivis des suffixes.

La ligne 80 effectue le choix d'un préfixe au hasard.

La ligne 90 effectue le choix d'un suffixe au hasard.

Le mot obtenu en ligne 100 est affiché par la ligne 110, puis on recommence (ligne 130).

POUR JOUER :

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

Taper CNT et C pour sortir du programme.

EXTENSIONS POSSIBLES :

On pourra très facilement augmenter la liste des mots disponibles.

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

```

10 REM MACHINAMOT
20 HOME
30 DIM A$(15)
40 DIM B$(15)
50 FOR I = 1 TO 15
60 READ A$(I),B$(I)
70 NEXT I
80 I = INT ( RND (1) * 15 + 1)
90 J = INT ( RND (1) * 15 + 1)
100 C$ = A$(I) + B$(J)
110 PRINT C$
120 FOR S = 1 TO 1200: NEXT S
130 GOTO 80
140 DATA ANDRO,CRATE,COSMO,PHILE
    ,MICRO,PHOBE,HELIO,TROPE,SEX
    O,GYNE,STEREO,MANE
150 DATA PALED,NAUTE,MACRO,PHAGE
    ,HYDRO,FUGE,STENO,PATHE,PATH
    O,CARDE,OLEO,DERME
160 DATA MYTHO,SCOPE,HOMEO,VORE,
    OCTO,PHONE

```

Plus grand, plus petit

REFERENCES :

Difficulté de programmation : facile.

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

Difficulté du jeu : difficile.

LE JEU :

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

LE PROGRAMME :

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

Le nombre à deviner est placé aux lignes 60-70, dans une variable alphanumérique X\$.

Les comparaisons avec le nombre essayé sont faites à la ligne 110 (qui teste si l'on a trouvé juste), et dans la boucle I (130-160) où l'on fait la comparaison "lettre à lettre".

L'affichage du résultat est réalisé grâce à la ligne 170.

Le compteur du nombre d'essais est en ligne 180.

POUR JOUER :

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

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

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

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

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

Si l'on a trouvé l'APPLE II affichera VOUS AVEZ TROUVE... EN... COUPS. Puis AUTRE PARTIE (O/N) ; on fera O suivi de RETURN pour commencer une nouvelle partie.

EXTENSIONS POSSIBLES :

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

On peut aussi prévoir de laisser jouer l'APPLE II qui tiendra alors le rôle de decodeur.

```

10 REM PLUS GRAND, PLUS PETIT
20 HOME
30 VTAB 1: INPUT "---- NB. DE CHI
    FFRES <=8 : ";C
40 IF C > 8 OR C < 1 THEN PRINT
    CHR$(7): GOTO 30
50 E = 1:F = 1
60 X = INT ( RND (1) * (10 ^ C))

70 X$ = RIGHT$ ("00000000" + STR$
    (X),C)
80 VTAB 3: HTAB 1: PRINT "ESSAI
    N.      NOMBRE      >
    <": PRINT "=====
    ====="
90 VTAB F + 4: HTAB 5: PRINT E;;
    HTAB 13: CALL - 868: INPUT
    "";P$
100 IF LEN (P$) < > C THEN PRINT
    CHR$(7):: GOTO 90
110 IF P$ = X$ THEN 200
120 A = 0:B = 0
130 FOR I = 1 TO C:P1$ = MID$ (
    P$,I,1):X1$ = MID$ (X$,I,1)

140 IF P1$ > X1$ THEN A = A + 1
150 IF P1$ < X1$ THEN B = B + 1
160 NEXT I
170 VTAB F + 4: HTAB 31: PRINT A
    ; TAB( 36);B
180 E = E + 1:F = F + 1: IF F = 1
    5 THEN F = 1
190 GOTO 90
200 VTAB 21: INVERSE : PRINT "VO
    US AVEZ TROUVE ";X$;" EN ";E
    ;" COUPS.": NORMAL
210 VTAB 23: PRINT "AUTRE PARTIE
    ? (O/N) : ";: GET K$: IF K$
    < > "N" THEN 20

```

Ordre

REFERENCES :

Difficulté de programmation : moyenne.

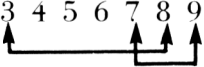
Catégorie de jeu : jeu de réflexion.

Difficulté du jeu : moyenne.

LE JEU :

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

B A D G H I J C F E
0 1 2 3 4 5 6 7 8 9 : permutation 83



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

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 la ligne 130 affiche les numéros de lettres de l'alphabet pour la permutation.

On entre la permutation en ligne 170.

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

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

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

La ligne 260 appelle une troisième fois le sous-programme d'inversion.

La ligne 270 comptabilise les essais.

La ligne 160 contrôle les résultats.

POUR JOUER :

Lancer le programme par RUN.

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

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

Le nombre de coups sera affiché en fin de partie.

EXTENSIONS POSSIBLES :

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

```
10 REM ORDRE
20 HOME
30 A$ = ""
40 FOR I = 1 TO 10
50 R$ = CHR$ ( INT ( RND (1) * 1
    0 + 65)) + " "
60 FOR J = 1 TO I * 2 - 3
70 IF R$ = MID$ (A$,J,2) THEN 5
    0
80 NEXT J
90 A$ = A$ + R$
100 NEXT I
110 S = 1
120 VTAB 3: PRINT "RESULTAT : ";
    A$
130 VTAB 4: HTAB 12: PRINT "O 1
    2 3 4 5 6 7 8 9"
140 VTAB 9: HTAB 1: PRINT "ESSAI
    NUMERO ";S
150 VTAB 13: CALL - 868
160 IF A$ = "A B C D E F G H I J
    " THEN 290
170 VTAB 13: INPUT "INVERSION (S
    OUS LA FORME 'IJ') : ";R$
180 I = VAL ( LEFT$ (R$,1)) + 1:
    IF I < 1 OR I > 10 THEN PRINT
    CHR$ (7): GOTO 170
190 J = VAL ( RIGHT$ (R$,1)) + 1
    : IF J < 1 OR J > 10 OR I =
    J THEN PRINT CHR$ (7): GOTO
    170
200 GOSUB 310
210 A = J:I = I - 1: IF I < 1 THEN
    I = 10:J = 2: GOTO 230
```

```

220 J = I + 2: IF J > 10 THEN J =
    1
230 GOSUB 310
240 I = A - 1: IF I < 1 THEN I =
    10: J = 2: GOTO 260
250 J = A + 1: IF J > 10 THEN J =
    1
260 GOSUB 310
270 S = S + 1
280 GOTO 120
290 INVERSE : VTAB 20: PRINT "BR
    AVO. GAGNE EN ";S;" COUPS.":
    NORMAL
300 END
310 IF I > J THEN X = I:I = J:J =
    X
320 A$ = " " + A$ + " "
330 B$ = LEFT$ (A$,I * 2) + MID$
    (A$,2 * J + 1,2) + MID$ (A$
    ,2 * I + 3,(J - I - 1) * 2) +
    MID$ (A$,2 * I + 1,2) + MID$
    (A$,2 * J + 3)
340 A$ = MID$ (B$,3, LEN (B$) -
    4)
350 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 affichera alors des tirets à l'emplacement des lettres de ce mot et demande au deuxième joueur de proposer une lettre. Chaque réponse correcte fait apparaître la lettre à son emplacement exact dans le mot, et chaque réponse fausse fait progresser le dessin de la potence.

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

LE PROGRAMME :

Le mot à deviner est introduit en ligne 70 ; une variable B\$, en ligne 20, contient tous les éléments du dessin.

La ligne 100 fabrique un mot P\$ qui contient des tirets à l'emplacement de chacune des lettres du mot initial.

Le sous-programme 280-310 s'occupe du dessin (selon la valeur de M).

Les lignes 190-200 affichent le résultat et arrêtent le jeu.

La lettre essayée est entrée en ligne 120 et testée dans une boucle I (140-160) ; le sous-programme 220-260 remplace chaque tiret par la lettre correspondante.

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

Il sera intéressant de constituer une liste de mots dans lesquels l'APPLE II pourra piocher au hasard ; on pourra aussi améliorer le dessin et modifier le nombre d'essais permis selon la difficulté du mot.

```

10 REM LE PENDU
20 B$ = "### # ! ##### - !
    ! - ! ! ! ! ! !
    "
30 DATA 3,8,6,8,6,5,6,11,9,6,9,
    10
40 FOR I = 1 TO 6: READ V(I),H(I)
    ): NEXT I
50 HOME
60 FOR I = 1 TO 13: VTAB I: PRINT
    ".....": NEXT
70 VTAB 3: HTAB 20: PRINT "VOTRE
    MOT :": VTAB 5: HTAB 20: INPUT
    "====> ";R$
80 L = LEN (R$): IF L > 14 OR L =
    0 THEN PRINT CHR$ (7);: GOTO
    70
90 VTAB 3: HTAB 20: PRINT "MOT D
    E ";L;" LETTRES."
100 P$ = LEFT$ ("-----"
    ,L)
110 VTAB 5: HTAB 26: PRINT P$
120 VTAB 17: HTAB 1: PRINT "TAPE
    R VOTRE LETTRE : ";: GET K$:
    PRINT K$
130 T = 0
140 FOR I = 1 TO L
150 IF MID$ (R$,I,1) = K$ THEN
    GOSUB 220
160 NEXT I
170 IF T = 0 THEN M = M + 1: GOSUB
    280
180 VTAB 5: HTAB 26: PRINT P$
190 IF P$ = R$ THEN VTAB 21: INVERSE
    : PRINT "BRAVO. VOUS AVEZ TR
    OUVÉ ...": NORMAL : END
200 IF M = 6 THEN VTAB 21: INVERSE
    : PRINT "DOMMAGE. VOUS AVEZ
    PERDU ...": VTAB 23: PRINT "
    LE MOT ETAIT ";: NORMAL : PRINT
    " ";R$: END

```

```

210 GOTO 120
220 T = 1
230 P$ = " " + P$ + " "
240 P$ = LEFT$ (P$,I) + K$ + RIGHT$
      (P$,L - I + 1)
250 P$ = MID$ (P$,2,L)
260 RETURN
270 END
280 VTAB V(M): HTAB H(M): PRINT
      MID$ (B$,M * 9 - 8,3)
290 VTAB V(M) + 1: HTAB H(M): PRINT
      MID$ (B$,M * 9 - 5,3)
300 VTAB V(M) + 2: HTAB H(M): PRINT
      MID$ (B$,M * 9 - 2,3)
310 RETURN

```

Belote

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de cartes.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

En ligne 30, on crée un tableau pouvant contenir les 20 cartes.

La boucle I (60-120) assure le tirage aléatoire de 20 nombres, pris parmi 32 ; la boucle J (90-110) vérifie que l'on ne réalise pas deux fois le même tirage.

La deuxième boucle I (140-280) assure ensuite l'affichage, les noms des cartes étant affichés en clair, et la couleur étant décodée et affichée en clair, grâce aux lignes 180 à 250.

POUR JOUER :

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

Un nouveau RUN permet d'obtenir une nouvelle distribution.

EXTENSIONS POSSIBLES :

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

```

10  REM BELOTE
20  HOME
30  DIM A(20)
40  K = 1
50  VTAB 10: FLASH : PRINT "PATIE
    NTEZ ! JE DISTRIBUE ...": NORMAL

60  FOR I = 1 TO 20
70  A(I) = INT ( RND (1) * 32 + 1
    )
80  IF I = 1 THEN 120
90  FOR J = 1 TO I - 1
100  IF A(I) = A(J) THEN 70
110  NEXT J
120  NEXT I
130  VTAB 10: CALL - 868
140  FOR I = 1 TO 20
150  S$ = "CA"
160  S = A(I) + 1 - 8 * INT (A(I)
    / 8)
170  R$ = LEFT$ ( STR$ (S + 6) +
    " ",2)
180  IF S = 5 THEN R$ = "VA"
190  IF S = 6 THEN R$ = "DA"
200  IF S = 7 THEN R$ = "RO"
210  IF S = 8 THEN R$ = "AS"
220  IF A(I) < = 24 THEN S$ = "T
    R"
230  IF A(I) < = 16 THEN S$ = "P
    I"
240  IF A(I) < = 8 THEN S$ = "CO
    "
250  VTAB 2 * K + 1: HTAB 10 * INT
    ((I - 1) / 5) + 1: PRINT R$;
    " ";S$
260  K = K + 1
270  IF K = 6 THEN K = 1
280  NEXT I

```

Un taquin de calcul

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : difficile.

LE JEU :

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

Le codeur part d'un nombre (proposé par l'APPLE II) 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 (80-180) permet d'entrer successivement les 5 opérations.

4 sous-programmes sont utilisés pour les calculs (aux lignes 250 à 320).

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).

Les lignes 210-220 affichent les 5 opérations utilisées, ainsi que les opérateurs correspondants.

POUR JOUER :

On frappe RUN pour lancer le programme.

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

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

EXTENSIONS POSSIBLES :

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

```

10 REM UN TAQUIN DE CALCUL
20 HOME
30 X = INT ( RND (1) * 16 + 5)
40 VTAB 1: INPUT "VOTRE NOMBRE I
  INITIAL : ";X
50 VTAB 3: PRINT "ENTRER CINQ OP
  ERATIONS": PRINT "SOUS LA FO
  RME +5 OU /3 OU *8 ..."
60 F = X
70 DIM R$(5)
80 FOR I = 1 TO 5
90 VTAB I + 5: PRINT "OPERATION
  N.":I;
100 VTAB I + 5: HTAB 14: INPUT "
  : ";R$(I)
110 A = ASC ( LEFT$ (R$(I),1)) -
  41
120 IF A < = 0 OR A = 3 OR A =
  5 OR A > 6 THEN PRINT CHR$
  (7);: GOTO 100
130 IF LEN (R$(I)) < 2 THEN PRINT
  CHR$ (7);: GOTO 100
140 IF A > 3 THEN A = (A / 2) +
  1
150 ON A GOSUB 250,270,290,310
160 VTAB 15: CALL - 868
170 PRINT "NOMBRE OBTENU : ";F
180 NEXT I
190 VTAB 1: HTAB 24: PRINT "****
  ***"
200 FOR I = 1 TO 5: VTAB I + 5: CALL
  - 868: NEXT I
210 VTAB 7: PRINT "LES OPERATION
  S UTILISEES SONT : "
220 VTAB 9: PRINT R$(4);" " ;R$
  (1);" " ;R$(5);" " ;R$(3);
  " " ;R$(2)
230 VTAB 22: INVERSE : PRINT "=="
  = BON COURAGE ! ===": NORMAL
240 END
250 F = F * VAL ( MID$ (R$(I),2)
  )
260 RETURN
270 F = F + VAL ( MID$ (R$(I),2)
  )
280 RETURN
290 F = F - VAL ( MID$ (R$(I),2)
  )
300 RETURN
310 F = F / VAL ( MID$ (R$(I),2)
  )
320 RETURN

```

Pendu inversé

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu sur les mots.

Difficulté du jeu : facile.

LE JEU :

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

LE PROGRAMME :

La ligne 40 permet d'enregistrer, dans une variable alphanumérique D\$, les lettres de l'alphabet, dans l'ordre : voyelles, consonnes usuelles, autres consonnes. Le mot à deviner est introduit en ligne 60 (en fait, il suffirait d'introduire sa longueur).

La boucle I (90-110) fabrique un mot T\$ contenant des tirets à l'emplacement de chacune des lettres du mot initial. La ligne 120 imprime T\$.

La boucle en ligne 140 sert uniquement à simuler le temps de recherche d'une lettre.

La formule, placée en ligne 150, permet de modifier légèrement la probabilité de tirage, en faveur des voyelles et des consonnes usuelles.

La ligne 180 provoque un nouveau tirage, et la boucle I (220-250) réimprime les tirets et les lettres trouvées.

La ligne 260 incrémente le compteur d'essais.

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

POUR JOUER :

Lancer le programme par RUN.

Entrer le mot à découvrir, suivi de RETURN.

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

On pourra sortir du programme à tout moment en tapant CNT et C.

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 l'APPLE II ; on pourra aussi imaginer d'ajouter un affichage graphique, par exemple une potence ou une voiture qui avance vers un précipice, etc.


```

10 REM PENDU INVERSE
20 HOME
30 D$ = "YAEIOUBCDFGLMNPRSTVWHKJQ
   XZ"
40 D$ = "YAEIOUBCDFGLMNPRSTVWHKJQ
   XZ"
50 C = 1: T$ = ""
60 INPUT "---- ENTREZ VOTRE MOT
   : "; R$
70 HOME
80 VTAB 11: PRINT D$
90 FOR I = 1 TO LEN (R$)
100 T$ = T$ + "-"
110 NEXT I
120 VTAB 5: PRINT T$
130 VTAB 8: PRINT "JE CHERCHE ..
   . ESSAI N."; C
140 FOR J = 1 TO 700: NEXT J
150 X = 6 * ( RND (1) < .7) + 13 *
   ( RND (1) < .5) + 7 * ( RND
   (1) < .4)
160 X = INT (X * RND (1) + 1)
170 E$ = MID$ (D$, X, 1)
180 IF E$ = "-" THEN 150
190 VTAB 11: HTAB X: PRINT "-"
200 D$ = MID$ ( LEFT$ (" " + D$,
   X) + "-" + MID$ (D$, X + 1),
   2)
210 VTAB 14: PRINT "---- JE PREN
   DS LE "; E$
220 FOR I = 1 TO LEN (R$)
230 IF E$ = MID$ (R$, I, 1) THEN
   T$ = MID$ ( LEFT$ (" " + T$,
   I) + E$ + MID$ (T$, I + 1),
   2)
240 VTAB 5: HTAB I: PRINT MID$
   (T$, I, 1)
250 NEXT I
260 C = C + 1
270 IF T$ < > R$ THEN 130
280 VTAB 5: INVERSE : PRINT T$: NORMAL
290 VTAB 19: INVERSE : PRINT "J'
   AI TROUVE EN "; C - 1; " COUPS
   ... ": NORMAL

```

Nicomaque

REFERENCES :

Difficulté de programmation : moyenne.

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

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

LE PROGRAMME :

La ligne 30 permet de confectionner l'entier.

La boucle I (70-130) calcule les trois restes successifs, et la boucle emboîtée J (100-120) permet de les afficher sous une forme assez visuelle, aux bons emplacements.

Les lignes 40 et 50 sont là pour indiquer de quelle division il s'agit.

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

La ligne 170 s'occupe de gérer la prochaine partie, selon la réponse du joueur.

POUR JOUER :

On lance le programme en tapant RUN.

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

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

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

Pour faire un nouvel essai, on répondra O.

EXTENSIONS POSSIBLES :

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

```

10 REM NICOMAUQUE
20 HOME
30 X = INT ( RND (1) * 90 + 1)
40 PRINT "RESTE DANS LA DIVISION
    PAR : "
50 PRINT : PRINT TAB( 7);3; TAB(
    17);5; TAB( 27);7
60 PRINT : PRINT TAB( 7);".."; TAB(
    17);"...."; TAB( 27);".....
    "
70 FOR I = 1 TO 3
80 A = X - (2 * I + 1) * INT (X /
    (2 * I + 1))
90 IF A = 0 THEN 130
100 FOR J = 1 TO A
110 VTAB 5: HTAB 10 * I - 4 + J:
    PRINT "*"
120 NEXT J
130 NEXT I
140 VTAB 10: INPUT "---- VOTRE E
    SSAI : ";N
150 IF N = X THEN VTAB 14: INVERSE
    : PRINT "==== BRAVO !": NORMAL
160 IF N < > X THEN VTAB 14: INVERSE :
    PRINT "==== PERDU !";: NORMAL
    : PRINT " C'ETAIT ";X
170 VTAB 18: PRINT "UNE AUTRE PA
    RTIE ? (O/N) : ";: GET K$: IF
    K$ < > "N" THEN 20

```

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 l'APPLE II s'occupe de faire avancer les chevaux les uns après les autres, en commençant par le numéro 1 ; le classement sera affiché au fur et à mesure des arrivées (et en tenant compte des ex aequo).

LE PROGRAMME :

Une première boucle non formelle (60-170) affiche les chevaux dans leur position de départ.

Une deuxième boucle I (80-180) les fera avancer ; la position de chaque cheval sera repérée par un chiffre ou une lettre dans une variable alpha-numérique A\$.

La ligne 100 fera avancer les chevaux.

La ligne 120 gère les arrivées.

La ligne 130 réalisera les impressions.

Le sous-programme 230-280 fait avancer le compteur des arrivées, et imprime les résultats.

POUR JOUER :

On lance le programme par RUN.

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

La course se déroulera ensuite automatiquement.

EXTENSIONS POSSIBLES :

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

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

```

10  REM COURSE DE CHEVAUX
20  HOME
30  INPUT "NB. DE CHEVAUX (<=9) :
      ";N: IF N < 1 OR N > 9 THEN
      PRINT CHR$ (7):: GOTO 20
40  A$ = "":P = 0:A = 0:T = 1:K =
      0
50  J = T
60  A$ = A$ + "O":I = J
70  GOTO 130
80  FOR I = 1 TO N
90  IF MID$ (A$,I,1) = "K" THEN
      180
100  B$ = CHR$ ( ASC ( MID$ (A$,I
      ,1)) + INT ( RND (1) * 4 +
      1))
110  A$ = MID$ ( LEFT$ (" " + A$,
      I) + B$ + MID$ (A$,I + 1),2
      )
120  IF B$ > = "K" THEN GOSUB 2
      30
130  VTAB 2 * I + 4: HTAB ASC ( MID$
      (A$,I,1)) - 45: PRINT "      "
      ;I
140  FOR D = 1 TO 200: NEXT D
150  IF K < > 0 THEN 180
160  J = J + 1: IF J < = N THEN 6
      0
170  K = 1: GOTO 80
180  NEXT I
190  T = 0
200  IF A < N THEN 80
210  VTAB 21
220  END
230  A$ = MID$ ( LEFT$ (" " + A$,
      I) + "K" + MID$ (A$,I + 1),
      2)
240  P = P + 1
250  VTAB 2 * I + 4: PRINT P
260  T = 1
270  A = A + T
280  RETURN

```

A travers

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

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

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

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

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



LE PROGRAMME :

Les positions des chiffres et des lettres sont enregistrées dans une table N (4, 4) ; les lignes 10 à 70 définissent les valeurs de départ.

Les lignes 80 à 130 dessinent le terrain de jeu et les lignes 140-150 affichent le joueur qui a le trait.

La ligne 190 contrôle le coup joué, de même que les boucles I et J (200 à 240), et le changement de joueur est assuré par la ligne 340.

POUR JOUER :

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

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

On pourra ainsi faire 2E suivi de RETURN.

Si le coup est invalide, il ne sera pas exécuté, et le code du joueur restera à l'affichage ; s'il est autorisé, le déplacement sera effectué, et l'on obtient l'affichage correspondant au deuxième joueur.

Le premier qui aura réussi à faire disparaître ses 3 pièces aura gagné.

EXTENSIONS POSSIBLES :

On peut envisager d'étendre le jeu avec davantage de chiffres ou de lettres et d'améliorer les contrôles. On peut aussi définir une stratégie pour l'APPLE II.

```

10  REM A TRAVERS
20  DIM N(4,4):K = 1
30  FOR I = 1 TO 4
40  N(I,1) = ASC ("O") + I
50  N(4,I) = ASC ("O") + I
60  NEXT I
70  N(4,1) = 0
80  HOME : VTAB 3: HTAB 11: PRINT
    "-----"
90  FOR U = 1 TO 4: VTAB U * 3 +
    1: HTAB 10: PRINT "!": VTAB
    U * 3 + 2: HTAB 10: PRINT "!
    ": VTAB U * 3 + 3: HTAB 10: PRINT
    "!"
100  FOR V = 1 TO 4: VTAB U * 3 +
    1: HTAB V * 4 + 7: PRINT "
    !": VTAB U * 3 + 3: HTAB V *
    4 + 7: PRINT "___!": VTAB U *
    3 + 2: HTAB V * 4 + 7: PRINT
    "___!"
110  IF N(U,V) < > 0 THEN VTAB
    U * 3 + 2: HTAB V * 4 + 8: PRINT
    CHR$(N(U,V))
120  NEXT V
130  NEXT U
140  VTAB 18: PRINT "=== AU JOUEU
    R AYANT LES ";R$ = "CHIFFRE
    S": IF K = 0 THEN R$ = "LETT
    RES"
150  PRINT R$;" : "
160  VTAB 22: PRINT "      VOTRE CO
    UP ? (N,S,E,O) : "
170  VTAB 22: HTAB 30: INPUT R$: IF
    LEN (R$) < > 2 THEN PRINT
    CHR$ (7);: GOTO 170
180  K$ = RIGHT$ (R$,1):N$ = LEFT$
    (R$,1)
190  IF (K = 1 AND K$ = "O") OR (
    K = 0 AND K$ = "S") THEN PRINT
    CHR$ (7);: GOTO 170

```

```

200 FOR I = 1 TO 4
210 FOR J = 1 TO 4
220 IF N(I,J) = ASC (N$) THEN 2
    60
230 NEXT J
240 NEXT I
250 GOTO 140
260 A = I;B = J
270 I = I + (K$ = "S") - (K$ = "N
    ")
280 J = J + (K$ = "E") - (K$ = "O
    ")
290 IF (I = 0 AND K = 1) OR (J >
    4 AND K = 0) OR J = 0 OR I >
    4 THEN PRINT CHR$ (7);: GOTO
    170
300 IF (I = 0 AND K = 0) OR (J >
    4 AND K = 1) THEN 330
310 IF N(I,J) < > 0 THEN 140
320 N(I,J) = ASC (N$)
330 N(A,B) = 0
340 K = ABS (K - 1)
350 GOTO 80

```


Le loup et les agneaux

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

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

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

LE PROGRAMME :

Une variable A\$, en ligne 20 contient les positions de départ.

La boucle I (70-180) dessine le damier, et les lignes 200 à 230 placent le loup et les agneaux.

La ligne 240 gère le changement de joueur.

La ligne 260 affiche JEU DU LOUP si c'est le joueur qui manipule le loup qui a le trait, et la ligne 770 JEU DES AGNEAUX dans le cas contraire.

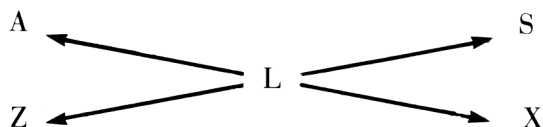
Les lignes 290 à 310 analysent la réponse, en fonction du joueur concerné.

POUR JOUER :

On lance le programme en tapant RUN. Après quoi, le damier s'affiche et la phrase JEU DU LOUP (A, S, Z, X) pour le loup et JEU DES AGNEAUX (ND et NG) pour les agneaux.

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

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



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

EXTENSIONS POSSIBLES :

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

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

```
10 REM LE LOUP ET LES AGNEAUX
20 A$ = "8284868815"
30 HOME
40 PRINT "      A  B  C  D  E  F
      G  H": PRINT "      _____
      "
50 B$ = "      ":N$ = "####"
60 VTAB 3
70 FOR I = 1 TO 8
80 FOR J = 1 TO 2
90 I$ = "      ": IF J = 2 THEN I$ =
      STR$ (I) + " "
100 PRINT I$;"!";
110 FOR K = 1 TO 8
120 IF AG = 0 THEN AG = 1: PRINT
      B$;: GOTO 140
130 AG = 0: PRINT N$;
140 NEXT K
150 PRINT "!"
160 NEXT J
170 AG = NOT (AG)
180 NEXT I
190 VTAB 18: HTAB 1: PRINT "8 !#
      ##      ###      ###      ###      !"
200 INVERSE : VTAB 2 * VAL ( MID$
      (A$,9,1)) + 1: HTAB 3 * VAL
      ( MID$ (A$,10,1)) + 2: PRINT
      "L": NORMAL
210 FOR I = 1 TO 4
220 INVERSE : VTAB 2 * VAL ( MID$
      (A$,I * 2 - 1,1)) + 1: HTAB
      3 * VAL ( MID$ (A$,I * 2,1)
      ) + 2: PRINT I: NORMAL
230 NEXT
240 S = ABS (S - 1)
250 VTAB 20: CALL - 868
```

```

260 VTAB.20: IF S = 0 THEN INPUT
    "JEU DU LOUP (A,S,Z,X) : ";R$
270 IF S = 1 THEN INPUT "JEU DE
    S AGNEAUX (nD OU nG) : ";R$:
    GOTO 320
280 C = 10
290 A = VAL ( RIGHT$ (A$,2)) + (
    C + 1) * (R$ = "X") + (C - 1
    ) * (R$ = "Z")
300 A = A - (C + 1) * (R$ = "A") -
    (C - 1) * (R$ = "S")
310 GOTO 340
320 C = 2 * VAL ( MID$ (R$,1,1))

330 A = VAL ( MID$ (A$,C - 1,2))
    - 9 * ( RIGHT$ (R$,1) = "D"
    ) - 11 * ( RIGHT$ (R$,1) = "
    G")
340 A$ = MID$ ( LEFT$ (" " + A$,
    C - 1) + RIGHT$ ("0" + STR$
    (A),2) + MID$ (A$,C + 1),2)

350 GOTO 60

```

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

LE PROGRAMME :

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

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

L'affichage est préparé par la ligne 80.

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

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

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

Les compteurs sont aux lignes 200 et 220.

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

POUR JOUER :

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

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

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

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

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

On peut sortir du programme à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :

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

```

10 REM MASTERMIND
20 HOME
30 INPUT "COMBIEN DE CHIFFRES <=
   6 : ";N
40 IF N > 6 OR N < 1 THEN PRINT
   CHR$ (7): GOTO 20
50 E = 1:F = 1
60 X$ = RIGHT$ ("000000" + STR$
   ( INT ( RND (1) * 10 ^ N)),N
   )
70 HOME
80 PRINT "NUM.      NOMBRE      B.
   PLACE  M.PLACE"
90 PRINT "=====
   ====="
100 VTAB F + 2: HTAB 2: PRINT E;
   TAB( 9); LEFT$ (".....",N)
   : VTAB F + 2: HTAB 9: INPUT
   "";E$
110 IF LEN (E$) < > N THEN PRINT
   CHR$ (7):: GOTO 100
120 X = 0:Y = 0
130 IF E$ = X$ THEN VTAB 21: INVERSE
   : PRINT "BRAVO ! TROUVE EN "
   ;E;" COUPS.": NORMAL : END
140 FOR I = 1 TO N
150 IF MID$ (X$,I,1) = MID$ (E
   $,I,1) THEN 220
160 FOR J = 1 TO N
170 IF J < > I AND MID$ (X$,I,
   1) = MID$ (E$,J,1) THEN 200

180 NEXT J
190 GOTO 230
200 Y = Y + 1
210 GOTO 230
220 X = X + 1
230 NEXT I
240 VTAB F + 2: HTAB 25: PRINT X
   : TAB( 34);Y
250 E = E + 1:F = F + 1: IF F = 1
   8 THEN F = 1
260 GOTO 100

```

Idem

REFERENCES :

Difficulté de programmation : moyenne.

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

Difficulté du jeu : selon le niveau choisi.

LE JEU :

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

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

LE PROGRAMME :

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

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

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

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

La ligne 130 affiche le numéro du joueur.

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

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

Le résultat sera affiché un court instant (ligne 210). La boucle I (250-270) permet de vérifier si tous les chiffres sont identiques.

L'impression et la fin de partie sont effectuées en ligne 280. Les lignes 290 et 310 s'occupent du changement de joueur.

POUR JOUER :

Lancer l'exécution du programme par RUN.

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

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

EXTENSIONS POSSIBLES :

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

```

10 REM IDEM
20 HOME
30 H = 1
40 DIM N(2)
50 INPUT "NIVEAU (2-7) : ";N
60 U = 10:P = 0
70 N(1) = INT ((U ^ N - U ^ (N -
    1)) * RND (1) + U ^ (N - 1)
    )
80 N(2) = N(1)
90 A = INT (2 * U ^ (N - 1) * RND
    (1) + U ^ (N - 1))
100 IF H = 1 THEN P = P + 1
110 HOME
120 IF N(H) < U ^ (N - 1) THEN N
    (H) = U ^ N - ABS (N(H))
130 VTAB 3: HTAB 1: PRINT "JOUEU
    R # ";H;"          ESSAI # ";P
140 VTAB 6: PRINT "NOMBRE : "; TAB(
    17 - N);N(H)
150 VTAB 7: HTAB 10: PRINT "7654
    321"
160 VTAB 10: INPUT "VOTRE JEU (
    SOUS LA FORME NC ) : ";R$
170 N(H) = N(H) + VAL ( LEFT$ (R
    $,1)) * U ^ ( VAL ( RIGHT$ (
    R$,1)) - 1)
180 N(H) = N(H) - A
190 IF N(H) > U ^ N THEN N(H) =
    N(H) - U ^ N
200 N(H) = INT (N(H))
210 VTAB 16: PRINT "RESULTAT : "
    ;N(H)
220 FOR I = 1 TO 1200: NEXT I
230 R$ = STR$ (N(H))
240 R$ = MID$ (R$,2)
250 FOR I = 1 TO LEN (R$) - 1
260 IF MID$ (R$,I,1) < > MID$
    (R$,I + 1,1) THEN 290
270 NEXT I
280 VTAB 21: INVERSE : PRINT "BR
    AVO, JOUEUR # ";H: NORMAL : END

290 H = H + 1
300 IF H > 2 THEN H = 1
310 GOTO 100

```

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

Les lignes 90-180 réalisent le dessin, en décalant l'impression au fur et à mesure (à l'aide de la variable K, incrémentée en ligne 160).

Les lignes 190 à 220 assurent la progression de l'heure.

Les lignes 230 à 290 effectuent l'affichage de deux points clignotants pour les secondes.

POUR JOUER :

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

L'heure s'affiche alors assez rapidement et l'horloge est en route. Pour l'arrêter, il faudra appuyer sur les touches CNT et C, puis taper l'ordre NORMAL.

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

EXTENSIONS POSSIBLES :

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


```

10 REM HORLOGE
20 HOME
30 INPUT "HEURE (FORME XXXX) : "
   ;A$
40 HOME
50 INVERSE
60 K = 1
70 A = VAL ( MID$ (A$,K,1))
80 IF A < > 1 AND A < > 4 THEN
   VTAB 1: HTAB 6 * K - 4: PRINT
   " "
90 FOR J = 1 TO 5
100 IF A = 0 OR (A > 3 AND A < >
   7) THEN VTAB J: HTAB 6 * K -
   4: PRINT " "
110 IF A < > 5 AND A < > 6 THEN
   VTAB J: HTAB 6 * K - 2: PRINT
   " "
120 IF A = 0 OR A = 2 OR A = 6 OR
   A = 8 THEN VTAB J + 4: HTAB
   6 * K - 4: PRINT " "
130 IF A < > 2 THEN VTAB J + 4
   : HTAB 6 * K - 2: PRINT " "
140 NEXT J
150 IF A > 1 THEN VTAB 5: HTAB
   6 * K - 3: PRINT " "
160 IF A < > 1 AND A < > 4 AND
   A < > 7 THEN VTAB 9: HTAB
   6 * K - 4: PRINT " "
170 K = K + 1
180 IF K < 5 THEN 70
190 A = VAL (A$) + 1
200 IF A = 2360 THEN A = 0
210 A$ = RIGHT$ ("000" + STR$ (
   A),4)
220 IF RIGHT$ (A$,2) = "60" THEN
   A$ = RIGHT$ ("0" + STR$ ( VAL
   ( LEFT$ (A$,2)) + 1),2) + "0"
   "0"
230 H = 1
240 FOR J = 1 TO 120
250 FOR I = 1 TO 40
260 VTAB 9: HTAB 12: X$ = " ": IF
   H = 1 THEN X$ = ":"
270 PRINT X$: NEXT I
280 H = ABS (H - 1)
290 NEXT J
300 GOTO 40

```

Reines sur l'échiquier

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

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

LE PROGRAMME :

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

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

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

Si, au cours de l'exploration, on rencontre un alignement, on revient en arrière, en utilisant les lignes 240 à 280 et on recommence.

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

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

POUR JOUER :

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

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

EXTENSIONS POSSIBLES :

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

```

10  REM REINES SUR ECHIQUE
20  HOME
30  INPUT "---- TAILLE DE L'ECHIQUE
    IER : ";S
40  DIM D(S)
50  I = 1
60  D(I) = 1
70  IF I = 1 THEN 110
80  FOR J = 1 TO I - 1
90  IF D(I) = D(J) OR ( ABS (D(I)
    - D(J)) = I - J) THEN 250
100 NEXT J
110 I = I + 1: IF I < = S THEN 60
120 HOME
130 FOR F = 1 TO S
140 VTAB 1: HTAB 2 * F + 1: PRINT
    F
150 VTAB 2 * F + 1: HTAB 1: PRINT
    F
160 FOR K = 1 TO S
170 VTAB 2 * F + 1: HTAB 2 * K +
    1: PRINT "."
180 NEXT K
190 VTAB 2 * F + 1: HTAB 2 * D(F
    ) + 1: PRINT "#"
200 VTAB 21: HTAB F: PRINT D(F)
210 NEXT F
220 VTAB 23: INVERSE : PRINT "TA
    PER UNE TOUCHE POUR CONTINUE
    R : ";; NORMAL : GET K$
230 VTAB 23: HTAB 1: CALL - 868

240 I = I - 1
250 D(I) = D(I) + 1
260 IF D(I) < = S THEN 70
270 I = I - 1
280 IF I < > 0 THEN 250
290 VTAB 20: HTAB 10: INVERSE : PRINT
    "RECHERCHE TERMINEE ...": NORMAL

```

Répétitions

REFERENCES :

Difficulté de programmation : facile.

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

Difficulté du jeu : moyenne.

LE JEU :

Des nombres vont apparaître un par un.

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

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

LE PROGRAMME :

La ligne 40 appelle un sous-programme (lignes 230 à 290) qui donne la règle du jeu.

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

La ligne 70 permet l'impression du dernier chiffre au hasard sur la quatrième ligne de l'écran, et la durée de l'affichage est déterminée par la ligne 80.

La ligne 100 attend la réponse du joueur, qui est comparée à la réponse correcte en ligne 110.

La ligne 140 marque les points.

La ligne 160 contrôle le meilleur score.

Les affichages de fin de partie sont effectués de la ligne 180 à la ligne 220.

POUR JOUER :

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

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

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

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

EXTENSIONS POSSIBLES :

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

```

10 REM REPETITIONS
20 M = 0
30 A$ = ""
40 GOSUB 230
50 H = 0
60 X = INT ( RND (1) * 10):A$ =
    A$ + STR$ (X)
70 HOME : VTAB 4: HTAB INT ( RND
    (1) * 20 + 1): PRINT X
80 FOR I = 1 TO 200 + RND (1) *
    300: NEXT I
90 HOME
100 VTAB 10: INPUT "VOTRE SEQUEN
    CE : ";R$
110 IF R$ < > A$ THEN PRINT CHR$
    (7); CHR$ (7);: GOTO 170
120 PRINT CHR$ (7);"EXACT - ATT
    ENTION ..."
130 FOR I = 1 TO 500: NEXT
140 H = H + 1
150 IF M < H THEN M = H
160 GOTO 60
170 HOME
180 INVERSE : VTAB 3: PRINT "VOT
    RE SCORE EST DE ";: NORMAL :
    PRINT " ";: FLASH : PRINT
    H: NORMAL
190 VTAB 6: INVERSE : PRINT "MEI
    LLEUR SCORE ACTUEL : ";M
200 VTAB 20: PRINT "AUTRE PARTIE
    ? (O/N) : ";: NORMAL : GET
    K$
210 IF K$ < > "N" THEN 30
220 END
230 HOME
240 VTAB 3: PRINT "VOUS DEVEZ RE
    PETER TOUTE LA SERIE DES"
250 VTAB 5: PRINT "NOMBRES APPAR
    US DEPUIS LE DEBUT DU JEU"
260 VTAB 8: PRINT "IL Y EN A UN
    DE PLUS A CHAQUE FOIS !"
270 VTAB 11: PRINT "TAPER UNE TO
    UCHE POUR COMMENCER ";
280 GET K$
290 RETURN

```

Mots

REFERENCES :

Difficulté de programmation : facile.

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

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

La ligne 20 permet de dimensionner la variable A\$ (I) qui contiendra les mots.

La boucle I (30) saisit la liste des mots (inscrits en DATA aux lignes 220-260).

La ligne 50 choisit un mot au hasard ; la boucle I (80-130) contrôle si une lettre n'a pas déjà été tirée, et reprend les lettres dans le désordre.

La ligne 140 affiche les lettres, et la ligne 150 attend votre proposition, qui est testée en ligne 160.

Le nombre d'essais est incrémenté en ligne 170.

La fin de partie est contrôlée par la ligne 180, et le score est affiché en ligne 200.

POUR JOUER :

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

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

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

EXTENSIONS POSSIBLES :

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

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

```

10  REM MOTS
20  DIM A$(30):S = 0:E = 0
30  FOR I = 1 TO 30: READ A$(I): NEXT
    I
40  HOME
50  Z = INT ( RND (1) * 30 + 1)
60  IF A$(Z) = "" THEN 50
70  R$ = A$(Z):A$ = "":A$(Z) = "":
    V$ = R$
80  FOR I = 1 TO LEN (R$)
90  Z = INT ( RND (1) * LEN (R$)
    + 1)
100 IF MID$ (R$,Z,1) = "-" THEN
    90
110 A$ = A$ + MID$ (R$,Z,1)
120 R$ = MID$ ( LEFT$ (" " + R$,
    Z) + "-" + MID$ (R$,Z + 1),
    2)
130 NEXT I
140 VTAB 3: PRINT "LES LETTRES S
    ONT : ";A$
150 VTAB 6: INPUT "VOTRE SOLUTIO
    N...: ";S$
160 IF S$ = V$ THEN S = S + 1
170 E = E + 1
180 VTAB 21: PRINT "AUTRE ESSAI
    ? (O/N) : ";: GET K$
190 IF K$ < > "N" THEN 40
200 VTAB 21: HTAB 1: CALL - 868
    : INVERSE : PRINT " VOTRE SC
    ORE EST DE ";S;" SUR ";E;". "
    : NORMAL
210 END
220 DATA JAMBON,ENERGUMENE,AFFIL
    IE,ERUDIT,INSIGNIFIANT,REALI
    STE
230 DATA IMPRESSIONANT,ALLECHANT
    ,EPIQUE,INTREPIDE,COUARD,ORD
    INATEUR
240 DATA TREVE,BANDEROLE,SENAT,A
    CTUEL,CONSACRE,DIEDRE
250 DATA LIVRE,TERMINE,SIGNATURE
    ,MALODORANT,AIMANT,ARBITRAIR
    E
260 DATA CHALEUR,OPPORTUN,PELURE
    ,ALTRUISTE,RAMONEUR,NOVICE

```

Permutations

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexion.

Difficulté du jeu : moyenne.

LE JEU :

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

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

LE PROGRAMME :

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

La ligne 70 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 90), supprimées dans A\$ et replacées en C\$, dans le bon ordre (lignes 100-110).

La ligne 130 contrôle le nombre de lettres de B\$.

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

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

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

POUR JOUER :

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

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

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

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

EXTENSIONS POSSIBLES :

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


```

10  REM PERMUTATIONS
20  HOME
30  INPUT "---- NOMBRE DE LETTRES
      (DE 1 A 12) : ";N
40  IF N < 1 OR N > 12 THEN 20
50  A$ = "ABCDEFGHIJKL":C$ = "
      ":B$ = ""
60  HOME
70  X = INT ( RND (1) * N + 1)
80  IF MID$ (A$,X,1) = " " THEN
    70
90  B$ = MID$ (A$,X,1) + B$
100 C$ = MID$ ( LEFT$ (" " + C$,
      X) + MID$ (A$,X,1) + MID$
      (C$,X + 1),2)
110 A$ = MID$ ( LEFT$ (" " + A$,
      X) + " " + MID$ (A$,X + 1),
      2)
120 S = 1
130 IF LEN (B$) < > N THEN 70
140 C$ = LEFT$ (C$, LEN (B$))
150 VTAB 4: PRINT "DEBUT ... : "
      ;B$
160 VTAB 6: PRINT "FIN ..... : "
      ;C$
170 VTAB 10: PRINT "PERMUTATION
      N. ";S
180 VTAB 12: INPUT "VOTRE QUANTI
      TE : ";L: IF L < 1 OR L > N THEN
    180
190 S = S + 1
200 A$ = ""
210 FOR I = 1 TO L
220 A$ = A$ + MID$ (B$,L - I + 1
      ,1)
230 NEXT I
240 B$ = A$ + MID$ (B$,L + 1)
250 VTAB 15: PRINT "RESULTAT ...
      . : ";B$
260 IF B$ < > C$ THEN 170
270 VTAB 18: HTAB 1: INVERSE : PRINT
      " TROUVE EN ";S - 1;" ESSAIS
      "
280 VTAB 21: PRINT "UNE AUTRE PA
      RTIE ? (O/N) : ";: NORMAL : GET
      K$
290 IF K$ < > "N" THEN 20

```

Bridge

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : variable, selon le problème posé.

LE JEU :

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

LE PROGRAMME :

Les lignes 30 et 40 définissent un jeu de 52 cartes.

La variable B\$, en ligne 50, 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 (60-200) réalise le tirage de quatre jeux de 13 cartes.

La boucle I (70-170) réalise tirage et impression. La carte est choisie parmi les 52 disponibles, en ligne 80.

La ligne 90 calcule la couleur de la carte tirée.

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

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

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

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

POUR JOUER :

On lance le programme par RUN.

Une première série de 13 cartes est affichée verticalement. Il suffira d'appuyer sur une touche pour obtenir la deuxième série de 13 cartes. On procédera de même pour la troisième et la quatrième.

EXTENSIONS POSSIBLES :

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

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

```

10  REM BRIDGE
20  HOME
30  R$ = "234567891VDRA"
40  R$ = R$ + R$ + R$ + R$
50  B$ = "TRCACOP10AAOS"
60  FOR J = 1 TO 4
70  FOR I = 1 TO 13
80  X = INT ( RND (1) * LEN (R$)
      + 1)
90  S = 2 * INT ((X - 1) / 13) +
      1
100  IF MID$ (R$,X,1) = "O" THEN
      80
110  T = X - 13 * INT (X / 13)
120  T = T + 13 * (T = 0)
130  VTAB I + 1: HTAB 10 * J - 9:
      PRINT MID$ (R$,X,1);
140  IF T > = 9 THEN PRINT MID$
      (B$,T,1)
150  VTAB I + 1: HTAB 10 * J - 5:
      PRINT MID$ (B$,S,2)
160  R$ = MID$ ( LEFT$ (" " + R$,
      X) + "O" + MID$ (R$,X + 1),
      2)
170  NEXT I
180  VTAB 19: HTAB 1: INVERSE : PRINT
      "TAPER UNE TOUCHE POUR CONTI
      NUER : ";; NORMAL : GET K$
190  VTAB 19: HTAB 1: CALL - 86B
200  NEXT J

```

Tirage 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 40 saisit le caractère frappé au clavier et le transforme en nombre (si ce n'est pas possible, le programme refuse l'entrée).

Une variable D non dimensionnée reçoit la valeur indiquée par chaque dé.

Une boucle I (100-130) sert au tirage (tirage en ligne 110 et impression en ligne 120).

La ligne 140 offre la possibilité de refaire un tirage ; elle n'est pas indispensable.

POUR JOUER :

On lance le programme par l'ordre RUN et on obtient l'affichage NOMBRE DE DES A LANCER auquel on répond en frappant un chiffre.

Le résultat est alors automatiquement affiché et le programme recommence.

Pour arrêter, il est nécessaire de taper 0.

EXTENSIONS POSSIBLES :

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

```

10  REM TIRAGE DE DES
20  HOME
30  VTAB 1: PRINT "NB. DE DES A L
    ANCE : ";
40  GET K$: PRINT K$
50  IF K$ = "0" THEN END
60  IF K$ < "1" OR K$ > "9" THEN
    PRINT CHR$ (7);: GOTO 30
70  K = VAL (K$)
80  HOME
90  VTAB 10: CALL - 868
100 FOR I = 1 TO K
110 D(I) = INT ( RND (1) * 6 + 1
    )
120 VTAB 10: HTAB 4 * I - 2: PRINT
    D(I)
130 NEXT I
140 GOTO 30

```

Hanoi

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

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

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

LE PROGRAMME :

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

Le nombre de disques est entré en ligne 50 et refusé s'il y a plus de 8 disques.

La boucle I (70-100) permet de fabriquer une variable A (1) qui contient l'état de la première pile sous forme numérique (en ligne 30, trois variables sont prévues pour conserver l'état des trois piles).

Les socles sont dessinés par la ligne 120 et les trois piles par les boucles emboîtées I (140-200) et J (150-190).

Le contrôle du coup est effectué par la ligne 300 et l'état des piles est modifié par les lignes 310 et 320.

POUR JOUER :

Après un RUN qui lance le programme, on entre le nombre de disques demandé, suivi de RETURN. On obtient le dessin de l'état initial et, à la question TOUR DE DEPART, on répond en donnant le numéro de la tour de départ. Ce numéro (1, 2 ou 3) sera refusé si la tour est vide. Puis 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 retaper les numéros des deux tours. Si le coup est possible, le déplacement est réalisé et l'on recommence.

EXTENSIONS POSSIBLES :

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

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

```

10  REM TOURS DE HANOI
20  HOME
30  DIM A(3):A(1) = 9:A(2) = 9:A(
    3) = 9
40  D$ = "======"
50  INPUT "NB. DE TOURS (MOINS DE
    8) : ";N
60  IF N < 1 OR N > 8 THEN PRINT
    CHR$(7)::VTAB 1:GOTO 50
70  FOR I = 0 TO N - 1
80  A(1) = VAL ( STR$ (A(1)) + STR$
    (N - I))
90  PRINT A(1)
100 NEXT I
110 HOME
120 INVERSE : VTAB 15: PRINT D$:
    : HTAB 13: PRINT D$: HTAB 2
    5: PRINT D$: NORMAL
130 VTAB 15: HTAB 5: PRINT 1:: HTAB
    17: PRINT 2:: HTAB 29: PRINT
    3
140 FOR I = 0 TO N - 1
150 FOR J = 1 TO 3
160 IF LEN ( STR$ (A(J))) = 1 THEN
    B$ = "": GOTO 180
170 B$ = STR$ (A(J))
180 IF LEN (B$) > 1 + 1 THEN INVERSE :
    VTAB 14 - I: HTAB 12 * J - 10: PRINT
    LEFT$ (D$, VAL ( MID$ (B$,I
    + 2,1))) : NORMAL
190 NEXT J
200 NEXT I
210 VTAB 17: HTAB 1: PRINT "TOUR
    DE DEPART :      TOUR D'ARRI
    VEE : "
220 VTAB 17: HTAB 18: GET K$: PRINT
    K$::D = VAL (K$)
230 IF K$ = "F" THEN END
240 IF D < 1 OR D > 3 THEN PRINT
    CHR$(7)::GOTO 220

```

```

250 IF A(D) = 9 THEN PRINT CHR$
    (7);: GOTO 220
260 VTAB 17: HTAB 38: GET K$: PRINT
    K$:A = VAL (K$)
270 IF A < 1 OR A > 3 THEN PRINT
    CHR$ (7);: GOTO 260
280 B$ = RIGHT$ ( STR$ (A(D)),1)
290 C$ = RIGHT$ ( STR$ (A(A)),1)
300 IF B$ > C$ THEN PRINT CHR$
    (7);: GOTO 220
310 A(A) = VAL ( STR$ (A(A)) + B
    $)
320 A(D) = INT (A(D) / 10)
330 GOTO 110

```


Niveau 4

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Jeu de loto	Gagnez sans problème, au prochain tirage.
La bombe	Saurez-vous la découvrir à temps ? (1 joueur).
Le jeu du 15	Un jeu de réflexion utilisant des chiffres (2 joueurs).
Affichage publicitaire	Si vous ne savez pas quoi mettre dans votre vitrine.
Chapeaux	Soyez très observateur, et vous réussirez (2 joueurs).
Combat aérien	Pas si facile de descendre un appareil qui bouge tout le temps.
Dessin de dés	Améliorez vos jeux de dés en incorporant ce programme.
Le jeu du missile	Une mission bien périlleuse (un seul joueur).
Zombies	Un peu farfelu, mais très visuel (1 joueur).
Le singe	Un jeu de lettres pour 2 joueurs, où il faut de bons réflexes.
Adresse	Jeu de réflexe pour 1 joueur, très visuel.
Sous-marin	Un classique des jeux vidéo (1 joueur).
Diago	Etes-vous vraiment sûr de vos réflexes ?
Décomaque	Pour vous éviter de sécher trop longtemps sur "Nicomaque".
Chasse au sous-marin	Il vous faudra pas mal de réflexion, et beaucoup de chance.
Clavier	Apprenez à taper à la machine en vous amusant.
Le lézard d'Alice	Beaucoup de hasard, mais très spectaculaire.
Calculatrice	Un programme de simulation très réaliste.
Echecs	Un programme assez simplifié, pour deux joueurs.
Horloge 2	Utilisation du crayon optique de l'ordinateur.
Danger spatial	Serez-vous assez adroit pour éviter toutes les météorites ?
Dames	Pour deux joueurs : classique.

Les techniques d'animation élémentaire

SAISIE AU VOL D'UN CARACTERE FRAPPE AU CLAVIER

Le principe d'un jeu graphique, en animation, est de laisser le jeu se dérouler tout seul, 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.

On va pour cela utiliser une instruction et une fonction classiques en Basic : **PEEK** et **POKE**.

PEEK est la fonction permettant de lire le contenu d'une case mémoire de l'APPLE.

Exemple : **X = PEEK (1024)** mettra dans X la valeur de l'octet se trouvant à l'adresse 1024. (Le résultat sera toujours compris entre 0 et 255 inclus).

POKE est une instruction permettant d'écrire une valeur à une adresse donnée.

Exemple : **POKE 1024, 193** mettra dans l'octet se trouvant à l'adresse 1024 la valeur 193.

Sur l'APPLE II, on trouve à l'adresse 49152 (ou - 16384 modulo 65536) l'octet du clavier. Si une touche a été frappée, cet octet contient le code ASCII du caractère correspondant + 128. Sinon, il contient le code de la dernière touche frappée, soit une valeur au plus égale à 127.

Ainsi pour savoir si le joueur a eu une action quelconque sur le clavier :
X = PEEK (- 16384)

Si **X ≥ 128**, alors une touche a été frappée, correspondant au caractère de code ASCII X-128.

Dès que l'on a détecté cette frappe, il faut repositionner le système pour pouvoir détecter la prochaine. Ceci se fait en écrivant une valeur quelconque à l'adresse 49168 (ou - 16368 modulo 65536).

Généralement, on fait :
POKE - 16368, 0.

BRUITAGE

Chaque référence à l'octet situé en 49200 (ou - 16336), que ce soit en lecture ou en écriture, produira l'émission par le haut-parleur d'un "click" beaucoup plus discret et agréable que le BIP du **PRINT CHR\$ (7)**. Ainsi :
X = PEEK (- 16336)

La valeur renvoyée dans X n'a aucune signification.

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 20, pour enregistrer les résultats.

Le tirage des nombres est effectué dans une boucle I (40-100), une deuxième boucle J (70-90) étant utilisée pour contrôler qu'il n'y a pas deux tirages identiques.

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

La boucle I (130-190) affiche le résultat, et la ligne 220 affiche le numéro complémentaire.

Les lignes 230 et 240 s'occupent du cas où l'on voudrait refaire une autre partie.

POUR JOUER :

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

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

EXTENSIONS POSSIBLES :

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

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

```

10  REM JEU DE LOTO
20  DIM N(7)
30  HOME
40  FOR I = 1 TO 7
50  N(I) = INT ( RND (1) * 49 + 1
    )
60  IF I = 1 THEN 100
70  FOR J = 1 TO I - 1
80  IF N(I) = N(J) THEN 50
90  NEXT J
100 NEXT I
110 VTAB 4: INVERSE : PRINT "===
    = T I R A G E   D U   L O T
    0   =====
120 PRINT "-----
    -----": NORMAL

130 FOR I = 1 TO 6
140 FOR J = 1 TO 20
150 VTAB 11: HTAB 7 * I - 5: PRINT
    INT ( RND (1) * 49 + 1); " "

160 FOR K = 1 TO 100: NEXT
170 NEXT J
180 VTAB 11: HTAB 7 * I - 5: PRINT
    N(I); " "
190 NEXT I
200 VTAB 17: PRINT "-- NO. COMPL
    EMENTAIRE : ";
210 FOR I = 1 TO 20: VTAB 17: HTAB
    25: PRINT INT ( RND (1) * 4
    9 + 1); " ": FOR J = 1 TO 100
    : NEXT J: NEXT I
220 VTAB 17: HTAB 25: PRINT N(7)
    ; " "
230 VTAB 21: INVERSE : PRINT "AU
    TRE TIRAGE ? (O/N) : ";: NORMAL
    : GET K$
240 IF K$ < > "N" THEN 30

```

La bombe

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

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

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

Pourrez-vous découvrir la bombe avant l'explosion ?

LE PROGRAMME :

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

La position de la chambre contenant le colis piégé est définie par un tirage aléatoire de 3 variables A, B, C, en ligne 30.

Les lignes 50 à 70 dessinent le détecteur.

Le signal de proximité est calculé en lignes 80-90 et la ligne 100 permet l'affichage des coordonnées de la chambre à découvrir, dans le cas où l'on a perdu.

La ligne 110 permet l'affichage, à la fois du signal et du temps restant à courir.

La ligne 120 réalise le décompte du temps.

La ligne 130 contrôle si l'on a trouvé ou non.

La boucle I (170-200) permet d'entrer les nouvelles coordonnées, en utilisant le sous-programme placé aux lignes 220 à 280.

POUR JOUER :

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

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

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

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

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

EXTENSIONS POSSIBLES :

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

```

10 REM LA BOMBE
20 L = 100:T = 200
30 A = INT ( RND (1) * L):B = INT
   ( RND (1) * L):C = INT ( RND
   (1) * L)
40 HOME
50 VTAB 4: INVERSE : HTAB 6: PRINT
   "      DISTANCE      RESTE
   "
60 VTAB 5: HTAB 6: PRINT "      ";;
   HTAB 20: PRINT "      ";; HTAB
   30: PRINT "      "
70 VTAB 6: HTAB 6: PRINT "
   " : NORMAL

80 S = ABS ((A / L + B + C * L) -
   (R(1) / L + R(2) + R(3) * L)
   )
90 S = INT (S * 100 + .5) / 100:
   S$ = LEFT$ ( RIGHT$ ("0000"
   + STR$ (S + .001),8),7)
100 IF T = 0 THEN VTAB 21: INVERSE
   : PRINT "= PERDU. LA SOLUTI
   ON ETAIT ";A;"",";B;"",";C;"":
   : NORMAL : END
110 VTAB 5: HTAB 11: PRINT S$;: HTAB
   25: PRINT T;" "
120 T = T - INT ( SQR (L))
130 IF S = 0 THEN VTAB 21: INVERSE
   : PRINT "=== BRAVO. VOUS AVE
   Z GAGNE ...": NORMAL : END
140 VTAB 10: HTAB 1: PRINT "----
   X .... : "
150 VTAB 12: HTAB 1: PRINT "----
   Y .... : "
160 VTAB 14: HTAB 1: PRINT "----
   Z .... : "
170 FOR I = 1 TO 3
180 GOSUB 220
190 R(I) = VAL (R$)
200 NEXT I
210 GOTO 50

```

```

220 K = 0:R$ = ""
230 VTAB 2 * I + 8: HTAB 15 + K:
    CALL - 868: GET K$: PRINT
    K$
240 IF ASC (K$) = 13 AND K > 0 AND
    K < 4 THEN RETURN
250 IF ( ASC (K$) = 13 AND K = 0
    ) OR ( ASC (K$) < > 13 AND
    K > = 3) THEN PRINT CHR$
    (7):: GOTO 230
260 IF K$ < "0" OR K$ > "9" THEN
    PRINT CHR$ (7):: GOTO 230
270 K = K + 1:R$ = R$ + K$
280 GOTO 230

```


Le jeu du 15

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : moyenne.

LE JEU :

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

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

LE PROGRAMME :

Les lignes 50 à 90 dessinent le terrain de jeu, et les lignes 110-120 s'occupent de la gestion du tour des joueurs (numéro 1 ou 2 pour chaque joueur, et jeton O ou X correspondant).

La ligne 130 attend le coup joué.

La boucle I (140-160) vérifie la validité du coup.

La ligne 170 teste la fin de partie, en cas de match nul, et la ligne 180 refuse les coups non conformes (avec retour à la situation initiale, ligne 100).

Les lignes 190 et 200 s'occupent de l'affichage du coup, et la ligne 210 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 240 à la ligne 330 : 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 340.

POUR JOUER :

Lancer le jeu en tapant RUN.

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

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

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

On fera une autre partie en tapant RUN.

EXTENSIONS POSSIBLES :

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

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

```

10 REM LE JEU DU 15
20 HOME
30 F = 0:R$ = "123456789"
40 O$ = "":X$ = ""
50 INVERSE : VTAB 4: PRINT "

      ": VTAB 8: PRINT "

"
60 FOR I = 5 TO 7: VTAB I: PRINT
  " "
70 FOR J = 1 TO 9: VTAB I: HTAB
  J * 4 + 1: PRINT " ": NEXT J

80 NEXT I: NORMAL
90 VTAB 10: PRINT "  1    2    3
    4    5    6    7    8    9"
100 K$ = "X": IF F = 1 THEN K$ =
    "O"
110 K$ = "X": IF F = 1 THEN K$ =
    "O"
120 VTAB 14: HTAB 1: PRINT "---->
    JOUEUR ";F + 1;" : ";K$;" -
    --> ";
130 GET K$: IF K$ < "1" OR K$ >
    "9" THEN PRINT CHR$(7):: GOTO
    130
140 FOR I = 1 TO LEN (R$)
150 IF K$ = MID$(R$,I,1) THEN
    GOSUB 190
160 NEXT I
170 IF VAL (R$) = 0 THEN VTAB
    20: HTAB 1: INVERSE : PRINT
    "=== C'EST FINI ...": NORMAL
    : END
180 GOTO 100
190 IF F = 1 THEN O$ = O$ + K$: VTAB
    6: HTAB I * 4 - 1: PRINT "O"

200 IF F = 0 THEN X$ = X$ + K$: VTAB
    6: HTAB I * 4 - 1: PRINT "X"

```

```

210 R$ = MID$ ( LEFT$ (" " + R$,
    I) + "O" + MID$ (R$,I + 1),
    2)
220 IF F = 1 THEN T$ = O$
230 IF F = 0 THEN T$ = X$
240 IF LEN (T$) < 3 THEN 340
250 FOR I = 1 TO 3
260 IF I = LEN (T$) THEN 340
270 FOR J = I + 1 TO LEN (T$)
280 IF J = LEN (T$) THEN 340
290 FOR K = J + 1 TO LEN (T$)
300 IF VAL ( MID$ (T$,I,1)) + VAL
    ( MID$ (T$,J,1)) + VAL ( MID$
    (T$,K,1)) = 15 THEN 360
310 NEXT K
320 NEXT J
330 NEXT I
340 F = ABS (F - 1)
350 RETURN
360 K$ = "X": IF F = 1 THEN K$ =
    "O"
370 VTAB 20: HTAB 1: INVERSE : PRINT
    "=== BRAVO, JOUEUR ";K$;" .
    ..": NORMAL : END

```

Affichage publicitaire

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : affichage publicitaire.

Difficulté du jeu : aucune.

LE JEU :

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

LE PROGRAMME :

Le texte sera introduit lettre par lettre aux lignes 60 à 90.

L'affichage sera tronqué à 40 caractères au plus, grâce à la ligne 170.

Le défilement vers la gauche est préparé à la ligne 180, et le défilement vers la droite à la ligne 190.

Le texte sera affiché en ligne 170, et la durée de l'affichage (vitesse de défilement) est choisie par la ligne 200.

La ligne 240 permettra de changer, si besoin est, les conditions du défilement (176 est le code ASCII du "0" + 128). La ligne 230 permet de terminer le programme (198 est le code ASCII du "F" + 128).

POUR JOUER :

On lancera le programme par RUN, et à l'affichage de la question ENTREZ VOTRE TEXTE, on introduira le texte à faire défiler, suivi de RETURN.

On obtient ensuite l'affichage DEFILEMENT A DROITE (D) OU A GAUCHE (G) auquel on répondra en tapant, soit D, soit G.

Puis à la question VITESSE, on répondra en introduisant un petit nombre pour une vitesse rapide, et un grand pour une vitesse lente (par exemple, de 0 à 50 ou plus).

Le texte commencera à défiler dès que l'on aura frappé RETURN.

Si l'on désire changer les conditions du défilement, il suffit d'appuyer sur la touche 0 pour retrouver les questions SENS et VITESSE. On tapera F pour interrompre le programme.

EXTENSIONS POSSIBLES :

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

```

10  REM AFFICHAGE PUBLICITAIRE
20  HOME
30  B$ = ""
40  INVERSE : PRINT "===  ENTREZ
    VOTRE TEXTE : ": NORMAL
50  K = 1
60  VTAB 4: HTAB K: GET K$
70  IF ASC (K$) = 13 THEN B$ = B
    $ + " ": GOTO 100
80  B$ = B$ + K$:K = K + 1
90  PRINT K$: GOTO 60
100 VTAB 7: HTAB 1: PRINT "DEFIL
    EMENT GAUCHE OU DROITE ? (G/
    D) : ": GET K$: PRINT K$
110 IF K$ < > "D" AND K$ < > "
    G" THEN PRINT CHR$ (7):: GOTO
    100
120 VTAB 9: INPUT "VITESSE (DE 0
    A 50) ... : ":V: IF V < 0 OR
    V > 50 THEN PRINT CHR$ (7)
    :: GOTO 120
130 V = 50 - V
140 HOME
150 T = LEN (B$)
160 IF T < 40 THEN B$ = B$ + B$:
    GOTO 150
170 VTAB 14: HTAB 1: PRINT LEFT$
    (B$,40)
180 IF K$ = "G" THEN B$ = MID$
    (B$,2) + LEFT$ (B$,1)
190 IF K$ = "D" THEN B$ = RIGHT$
    (B$,1) + LEFT$ (B$,T - 1)
200 FOR J = 0 TO V
210 X = PEEK ( - 16384): IF X <
    128 THEN 250
220 POKE - 16368,0
230 IF X = 198 THEN END
240 IF X = 176 THEN 100
250 NEXT J
260 GOTO 160

```

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 de chapeaux lors du croisement.

LE PROGRAMME :

La ligne 20 contient le dessin des deux chapeaux (dans deux variables alphanumériques B\$ et C\$).

La ligne 30 contient, dans une autre variable A\$, les symboles qui permettront de dessiner les personnages.

La ligne 50 permet, s'il y a lieu, l'échange des chapeaux.

La variable T de la ligne 60 sert à repérer s'il y a eu échange ou non.

Le dessin des deux personnages est réalisé par une boucle I (110-140), et leur déplacement est réalisé par la ligne 150.

Le test, permettant ou non le changement des couvre-chefs, est effectué en ligne 160.

Les lignes 170 à 240 déterminent si le premier joueur ayant appuyé sur une touche a gagné ou perdu.

POUR JOUER :

Lancer le programme par RUN.

Les personnages commencent immédiatement à se déplacer l'un vers l'autre.

Le joueur de gauche dispose des touches 1 et 2 et celui de droite, des touches 9 et 0.

S'il y a eu changement de chapeau, les touches à appuyer seront 1 (à gauche) et 0 (à droite).

S'il n'y a pas eu de changement, il faudra, pour gagner, être le premier à appuyer sur 2 (à gauche) et sur 9 (à droite).

L'affichage indiquera alors BRAVO GAUCHE ou BRAVO DROITE, ou, encore, PERDU GAUCHE ou PERDU DROITE.

EXTENSIONS POSSIBLES :

Il sera possible de prévoir un affichage couleur, avec une animation des personnages.

On pourra aussi compliquer le jeu à plaisir avec plusieurs objets échangés de façon aléatoire (il faudrait alors en trouver la liste complète).

```

10 REM CHAPEAUX
20 B$ = " =q=":C$ = " =/="
30 A$ = " ### ## # # ####
    ##### # # # # #
    # ## ##"
40 X = 1:Y = 35:T = 1
50 D$ = C$:C$ = B$:B$ = D$
60 T = ABS (T - 1)
70 HOME
80 IF X > 35 OR Y < 1 THEN 170
90 VTAB 1: HTAB X: PRINT " ";C$:
    " "
100 VTAB 1: HTAB Y: PRINT " ";D$
    ;" "
110 FOR I = 0 TO 11
120 VTAB I + 3: HTAB X: PRINT "
    "; MID$ (A$,I * 5 + 1,5);" "
130 VTAB I + 3: HTAB Y: PRINT "
    "; MID$ (A$,I * 5 + 1,5);" "
140 NEXT I
150 X = X + 1:Y = Y - 1
160 IF X = Y AND RND (1) < .5 THEN
    50
170 K = PEEK ( - 16384): IF K <
    128 THEN 80
180 POKE - 16368,0:K$ = CHR$ (
    K - 128)
190 IF K$ < > "1" AND K$ < > "
    2" AND K$ < > "9" AND K$ <
    > "0" THEN PRINT CHR$ (7)
    ;: GOTO 170
200 IF X < = Y THEN 220
210 IF (K$ = "1" AND T = 1) OR (
    K$ = "2" AND T = 0) THEN VTAB
    21: INVERSE : PRINT "=== BRA
    VO GAUCHE !": NORMAL : END
220 IF (K$ = "0" AND T = 1) OR (
    K$ = "9" AND T = 0) THEN VTAB
    21: INVERSE : PRINT "=== BRA
    VO DROITE !": NORMAL : END
230 IF K$ = "2" OR K$ = "1" THEN
    VTAB 21: INVERSE : PRINT "=
    == PERDU GAUCHE !": NORMAL :
    END
240 IF K$ = "9" OR K$ = "0" THEN
    VTAB 21: INVERSE : PRINT "=
    == PERDU DROITE !": NORMAL :
    END

```

Combat aérien

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexe (1 joueur).

Difficulté du jeu : difficile.

LE JEU :

Au centre de l'écran, se trouve le viseur de votre appareil, et l'avion ennemi se déplace devant vous, d'une façon qui pourra vous sembler au début incohérente.

Vous pouvez vous déplacer dans toutes les directions, grâce aux flèches de mouvement, et vous allez essayer d'amener l'appareil ennemi au centre de votre viseur.

Vous tirez sur lui en frappant Ø, et vous voyez vos projectiles qui convergent sur lui. Mais il continue à se déplacer, même pendant votre tir, et il n'est pas du tout évident que vous réussirez à l'avoir.

La partie continuera à se dérouler jusqu'à ce que vous l'ayez touché (affichage de GAGNE) ou qu'il soit sorti de votre champ de vision.

LE PROGRAMME :

La ligne 50 dessine l'appareil ennemi dont le déplacement aléatoire sera réalisé par les lignes 70 et 80.

Les lignes 100 et 110 vous permettent le pilotage de votre appareil.

La ligne 150 dessine le viseur de votre appareil.

Le sous-programme, gérant le tir, va de la ligne 180 à la ligne 230 et le test de fin de partie est effectué en ligne 220.

La ligne 160 regarde si vous avez tiré ou non.

POUR JOUER :

Dès que le programme a été lancé par l'ordre RUN, vous devez manœuvrer pour essayer de faire en sorte que l'appareil ennemi soit cadré dans votre viseur.

Pour cela, vous disposez des flèches de déplacement (touches I, J, K, M) et vous pouvez tirer à tout moment en pressant la touche Ø.

L'appareil ennemi peut continuer à se déplacer de manière aléatoire, même lorsque vous avez tiré.

La partie se terminera avec l'affichage de GAGNE, si vous touchez l'avion ennemi.

On peut aussi arrêter à tout moment en frappant CNT et C.

EXTENSIONS POSSIBLES :

Il sera assez facile de faire un affichage au graphisme plus soigné, utilisant la couleur. On pourrait envisager de stabiliser la position de l'avion ennemi pendant le tir (il suffit de modifier la ligne 160).

On peut aussi introduire plusieurs avions, compter les points ou le temps, tenir compte de sa propre réserve de carburant, etc.


```

10  REM COMBAT AERIEN
20  HOME
30  A = 5:B = 15
40  N = 21
50  VTAB A: HTAB B: PRINT "#-#"
60  U = A:V = B
70  A = A + 2 * INT ( RND (1) * 2
    ) - 1
80  B = B + 2 * INT ( RND (1) * 2
    ) - 1
90  K = PEEK ( - 16384) - 128: POKE
    - 16368,0: IF K < 0 THEN K =
    0
100 B = B - 2 * (K = 74) + 2 * (K
    = 75)
110 A = A - 2 * (K = 73) + 2 * (K
    = 77)
120 IF B < 1 OR B > 38 THEN B =
    ABS (B - 38)
130 IF A < 4 OR A > 20 THEN A =
    ABS (A - 20) + 3
140 VTAB U: HTAB V: PRINT "  "
150 INVERSE : VTAB 9: HTAB 18: PRINT
    "* *": VTAB 10: HTAB 18: PRINT
    " X ": VTAB 11: HTAB 18: PRINT
    "* *": NORMAL
160 IF K = 48 OR N < 21 THEN GOSUB
    180
170 GOTO 50
180 INVERSE : VTAB N: HTAB 29 -
    N: PRINT "/": VTAB N: HTAB N
    + 9: PRINT "°": NORMAL
190 VTAB N: HTAB 29 - N: PRINT "
    ": VTAB N: HTAB N + 9: PRINT
    " "
200 N = N - 1
210 IF N > 9 THEN RETURN
220 IF A > 8 AND A < 12 AND B >
    17 AND B < 21 THEN VTAB 22:
    INVERSE : PRINT "=== GAGNE
    ...": END
230 GOTO 40

```

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 NOMBRE DE DES provoque immédiatement un nouveau lancer du nombre de dés indiqué.

LE PROGRAMME :

La ligne 50 permet de contrôler s'il y a eu ou non action sur une touche de 1 à 5.

Les lignes 80 et 90 initialisent les deux variables de dessin des dés.

La boucle I (100-210) et les lignes 110 à 150 dessinent le cadre du dé.

La ligne 160 réalise le tirage, et la boucle J (170-200) dessine les points en relation avec le tirage réalisé (en appelant les variables A\$ et B\$).

POUR JOUER :

On lance le programme par l'ordre RUN, et, à la question NOMBRE DE DES, on répond en frappant une touche de 1 à 5. Les dés demandés sont alors représentés, et la question NOMBRE DE DES est de nouveau posée. Il suffira de taper 0 pour sortir du programme.

EXTENSIONS POSSIBLES :

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

```

10 REM DESSIN DE DES
20 HOME
30 VTAB 20: PRINT "NOMBRE DE DES
   (1 A 4) : ";
40 GET K$: IF K$ = "0" THEN END

50 K = VAL (K$): IF K < 1 OR K >
   4 THEN PRINT CHR$ (7);: GOTO
   40
60 PRINT K$
70 HOME
80 A$ = "      # #      ##
   #"
90 B$ = "121314324515525555"
100 FOR I = 1 TO K
110 VTAB 3: HTAB 10 * I - 9: PRINT
   "#####"
120 VTAB 11: HTAB 10 * I - 9: PRINT
   "#####"
130 FOR J = 4 TO 10
140 VTAB J: HTAB 10 * I - 9: PRINT
   "#      #"
150 NEXT J
160 X = INT ( RND (1) * 6 + 1)
170 FOR J = 1 TO 3
180 S = VAL ( MID$ (B$,3 * (X -
   1) + J,1))
190 VTAB J * 2 + 3: HTAB 10 * I -
   7: PRINT MID$ (A$,S * 5 - 4
   ,5)
200 NEXT J
210 NEXT I
220 GOTO 30

```

Le jeu du missile

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : contre l'APPLE II.

Difficulté du jeu : moyenne.

LE JEU :

L'APPLE II 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, et de revenir sain et sauf sur votre base ; mais, dès que l'APPLE II vous aura repéré, il va envoyer à votre poursuite, un missile très perfectionné qui n'aura de cesse de vous détruire.

Votre avion peut se déplacer dans toutes les directions, grâce aux touches de 1 à 8 ; réussirez-vous à remplir votre mission ?

LE PROGRAMME :

La ligne 30 dessine la base ennemie à survoler.

La ligne 40 dessine votre base de départ.

Les lignes 60 et 80 choisissent l'instant de départ du missile.

La ligne 100 dessine votre avion.

Les lignes 160 et 170 définissent son mouvement suivant 8 directions, en utilisant les sous-programmes en lignes 280, 290, 310 et 320.

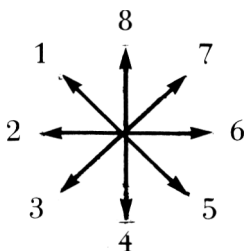
La ligne 180 assure le départ du missile qui est dessiné par la ligne 190 et dont le déplacement est géré par les lignes 200 et 210.

Les lignes 220 et 230 contrôlent la fin de partie.

POUR JOUER :

Lancer le programme par RUN.

Le terrain de jeu s'affiche et vous pouvez déplacer votre appareil en utilisant les touches de 1 à 8, avec la convention suivante :



Vous devez lui faire contourner le territoire ennemi (si vous, ou le missile, touchez ce territoire, une deuxième partie commence).

Si le missile vous détruit, vous obtenez un message TOUCHE, et la partie s'arrête.

Taper F pour arrêter la partie.

EXTENSIONS POSSIBLES :

Il sera sans doute nécessaire de rajouter divers contrôles dans le programme (l'avion a-t-il survolé la base ennemie, est-il revenu à sa propre base...). On peut aussi prévoir divers niveaux de jeu en faisant, par exemple, varier la vitesse du missile, prévoir l'utilisation de leurres par l'avion, l'explosion du missile au bout d'un certain temps, etc.

```
10 REM LE JEU DU MISSILE
20 HOME :M = 0
30 INVERSE : VTAB 5: HTAB 18: PRINT
   "###": VTAB 6: HTAB 18: PRINT
   "# #": VTAB 7: HTAB 18: PRINT
   "###": NORMAL
40 INVERSE : VTAB 20: HTAB 2: PRINT
   "===": PRINT "====": NORMAL

50 U = 15: X = 20
60 S = INT ( RND (1) * U + 6)
70 Y = 1
80 S = S - 1
90 V = 33
100 VTAB X: HTAB Y: PRINT "Y"
110 PRINT CHR$(7);
120 K = PEEK ( - 16384): IF K <
    128 THEN 120
130 VTAB 22: HTAB 30: PRINT K"
    "
140 POKE - 16368,0:K = K - 176:
    IF K < 1 OR K > 8 THEN 120
150 VTAB 22: HTAB 35: PRINT K"
    "M
160 F = 2 * (K < 5) - 1
170 ON K + 2 * (F - 1) GOSUB 280
    ,290,310,320
180 IF S < > 0 THEN 80
190 VTAB U: HTAB V: PRINT "<"
200 U = U - SGN (U - X)
210 V = V - SGN (V - Y)
220 IF U = X AND V = Y THEN VTAB
    22: HTAB 20: INVERSE : PRINT
    CHR$(7) + CHR$(7):"TOUCH
    E ...": NORMAL : END
230 IF (U > = 5 AND U < = 7) AND
    (V > = 18 AND V < = 20) THEN
    20
240 IF (X > = 5 AND X < = 7) AND
    (Y > = 18 AND Y < = 20) THEN
    20
```

```

250 IF X < = 5 AND Y > = 20 THEN
    M = 1
260 IF M = 1 AND X = 20 AND Y <
    = 4 THEN VTAB 22: HTAB 15:
    INVERSE : PRINT "MISSION AC
    COMPLIE ...": NORMAL : END
270 GOTO 100
280 X = X - F
290 Y = Y - F
300 RETURN
310 GOSUB 290
320 X = X + F
330 RETURN

```

Zombies

REFERENCES :

Difficulté de programmation : assez difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez facile.

LE JEU :

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

La partie se termine, soit avec la disparition du joueur, soit par l'élimination de tous les ZOMBIES.

LE PROGRAMME :

La position des ZOMBIES est définie par un tableau Z et un tableau T indique l'occupation des cases.

Les lignes 50 à 150 (boucle I) dessinent le cadre, les ZOMBIES et les trous noirs.

La ligne 160 définit la position initiale du joueur.

Les lignes 190 à 400 régissent les déplacements du joueur et des zombies.

Vous pouvez modifier le nombre de zombies et de trous noirs en ligne 30.

POUR JOUER :

Après avoir lancé l'exécution par RUN, le joueur devra se déplacer dans les 8 directions, en veillant à ne pas sortir des limites du terrain.

Il devra manœuvrer de façon à faire venir les ZOMBIES sur les trous noirs, sans se faire lui-même coïncider.

Pour arrêter la partie avant la fin, il suffira de taper CNT et C.

EXTENSIONS POSSIBLES :

Il serait intéressant d'agrandir le terrain de jeu, d'augmenter le nombre de ZOMBIES ou de trous noirs, d'améliorer les contrôles, de placer le joueur de façon plus aléatoire, éventuellement de modifier la règle.

On pourra aussi, bien sûr, améliorer la qualité du dessin et introduire la couleur.

```

10 REM ZOMBIES
20 HOME
30 Z = 4:N = 11
40 DIM T(22,22),ZX(Z - 1),ZY(Z -
    1)
50 INVERSE : FOR I = 1 TO 22: VTAB
    1: HTAB I: PRINT " ": VTAB 2
    2: HTAB I: PRINT " ": VTAB I
    : HTAB 1: PRINT " ": VTAB I:
    HTAB 22: PRINT " "
60 T(1,I) = 1:T(22,I) = 1:T(I,1) =
    1:T(I,22) = 1
70 IF RND (1) < N / 22 THEN X =
    INT ( RND (1) * 20 + 2):Y =
    INT ( RND (1) * 20 + 2): VTAB
    X: HTAB Y: PRINT "#":T(X,Y) =
    2
80 NEXT I: NORMAL
90 FOR I = 1 TO Z
100 X = INT ( RND (1) * 20 + 2):
    Y = INT ( RND (1) * 20 + 2)

110 IF T(X,Y) < > 0 THEN 100
120 ZX(J) = X:ZY(J) = Y:J = J + 1

130 VTAB X: HTAB Y: PRINT "Z":T(
    X,Y) = 3
140 NEXT I
150 VTAB 7: HTAB 28: PRINT "U I
    0": HTAB 29: PRINT "Q!/: HTAB
    28: PRINT "J--*--K": HTAB 29: PRINT
    "/!Q": HTAB 28: PRINT "N M ,
    "
160 X = INT ( RND (1) * 20 + 2):
    Y = INT ( RND (1) * 20 + 2)

170 IF T(X,Y) < > 0 THEN 160
180 VTAB X: HTAB Y: PRINT "O":A =
    X:B = Y
190 K = PEEK ( - 16384): IF K <
    128 THEN 190
200 POKE - 16368,0
210 X = X - (K = 213 OR K = 201 OR
    K = 207) + (K = 206 OR K = 2
    05 OR K = 172)
220 Y = Y - (K = 213 OR K = 202 OR
    K = 206) + (K = 207 OR K = 2
    03 OR K = 172)
230 VTAB A: HTAB B: PRINT " ": VTAB
    X: HTAB Y

```



```

240 IF T(X,Y) = 1 THEN INVERSE
   : PRINT "O": VTAB 16: HTAB 2
4: PRINT "SORTI ..." + CHR$
(7) + CHR$ (7) + CHR$ (7):
   NORMAL : END
250 IF T(X,Y) = 2 THEN INVERSE
   : PRINT "O": VTAB 16: HTAB 2
4: PRINT "TROU NOIR ..." + CHR$
(7) + CHR$ (7) + CHR$ (7):
   NORMAL : END
260 IF T(X,Y) = 3 THEN INVERSE
   : PRINT "O": VTAB 16: HTAB 2
4: PRINT "BON APPETIT ." + CHR$
(7) + CHR$ (7) + CHR$ (7):
   NORMAL : END
270 PRINT "O"
280 T(A,B) = 0: T(X,Y) = 9
290 A = X: B = Y
300 FOR I = 0 TO Z - 1
310 IF ZX(I) = 0 THEN 380
320 C = ZX(I) - SGN (ZX(I) - X)
330 D = ZY(I) - SGN (ZY(I) - Y)
340 T(ZX(I),ZY(I)) = 0: VTAB ZX(I)
   : HTAB ZY(I): PRINT " ": ZX(I)
   : C: ZY(I) = D: VTAB C: HTAB
   D
350 IF T(C,D) = 1 OR T(C,D) = 2 THEN
   INVERSE : PRINT "Z" + CHR$
(7): ZX(I) = 0: NORMAL : P = P
   + 1: GOTO 380
360 IF T(C,D) = 9 THEN INVERSE
   : PRINT "*" + CHR$ (7) + CHR$
(7): VTAB 15: HTAB 25: PRINT
"=== PERDU ...": NORMAL : END

370 T(C,D) = 3: PRINT "Z":
380 NEXT I
390 IF P = Z THEN INVERSE : VTAB
18: HTAB 25: PRINT "GAGNE ..
. ": NORMAL : END
400 GOTO 190

```

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 l'APPLE II 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 20 contient une liste V\$ des voyelles courantes, et 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 30 définit la longueur du mot (en principe, de 3 à 6 lettres, mais on peut également modifier cette longueur).

La boucle I (60-110) permet la confection d'un "mot" en prenant une lettre, soit dans l'alphabet complet, soit dans la liste C\$ (ligne 80), soit dans la liste V\$ (ligne 90).

La ligne 130 temporise l'apparition du résultat.

La ligne 160 gère la fin de partie, avec affichage du score.

La ligne 170 regarde si l'un des joueurs a réagi.

Enfin, les lignes 190 et 200 comptabilisent les points de chaque joueur.

POUR JOUER :

On lance le programme en tapant RUN.

La première série de lettres est alors affichée sur l'écran. Le joueur de gauche devra taper 1 s'il voit qu'il peut réaliser un mot, celui de droite devra taper 0.

Si rien n'a été frappé, une autre série de lettres sera affichée après peu de temps.

Si l'un des joueurs a frappé 0 ou 1, la question (J)USTE ou (F)AUX est alors affichée. On répond J si le mot est correct, F si le mot est incorrect.

Les points sont alors calculés par l'APPLE II, et la série de lettres suivante est alors affichée.

Pour sortir du programme, il faudra taper P au lieu de 0 ou 1 ; le score final est alors affiché.

EXTENSIONS POSSIBLES :

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

```
10 REM LE SINGE
20 A = 0:B = 0:V$ = "AEIOU":C$ =
   "BCDGLMNFIRSTV"
30 X = INT ( RND (1) * 4 + 3)
40 M$ = ""
50 HOME
60 FOR I = 1 TO X
70 E$ = CHR$ ( INT ( RND (1) * 2
   6 + 65))
80 IF RND (1) < .4 THEN E$ = MID$
   (C$, INT ( RND (1) * LEN (C
   $) + 1),1)
90 IF RND (1) < .5 THEN E$ = MID$
   (V$, INT ( RND (1) * LEN (V
   $) + 1),1)
100 M$ = M$ + E$
110 NEXT I
120 VTAB 5: INVERSE : PRINT LEFT$
   ("      ",X + 2): VTAB 6: PRINT
   " ";M$;" ": VTAB 7: PRINT LEFT$
   ("      ",X + 2): NORMAL
130 FOR J = 1 TO 500: NEXT J
140 K = PEEK ( - 16384): IF K <
   128 THEN 30
150 POKE - 16368,0:K = K - 128
160 IF K = 80 THEN VTAB 13: INVERSE
   : PRINT "SCORE GAUCHE ";A: TAB(
   21): "SCORE DROITE ";B:".": NORMAL
   : END
170 IF K < > 48 AND K < > 49 THEN
   30
180 VTAB 8: PRINT "(J)USTE OU (F
   )AUX : ";; GET K$: PRINT K$:
   IF K$ < > "J" AND K$ < >
   "F" THEN PRINT CHR$ (7): GOTO
   180
190 A = A + (K$ = "J" AND K = 49)
   - (K$ = "F" AND K = 49)
200 B = B + (K$ = "J" AND K = 48)
   - (K$ = "F" AND K = 48)
210 GOTO 30
```

Adresse

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexe (pour 1 joueur).

Difficulté du jeu : difficile.

LE JEU :

Des chiffres, entre 0 et 9, vont apparaître au hasard, en divers emplacements de l'écran. Ils restent affichés pendant un temps très court. Le joueur doit réussir à appuyer sur la touche numérique correspondante, au moment où le chiffre est affiché.

Le chiffre tapé par le joueur sera alors affiché en grand format sur la gauche de l'écran ; et chaque réponse exacte rapporte 10 points. La durée de la partie est aléatoire et le score sera affiché en fin de partie.

LE PROGRAMME :

Les lignes 50 et 60 contiennent, dans deux variables alphanumériques C\$ et B\$, les références qui seront utilisées pour dessiner les chiffres en grand format.

L'emplacement du chiffre tiré par l'APPLE II est choisi aux lignes 70 et 80, et le chiffre est tiré en ligne 90.

La ligne 100 réalise l'impression, et la ligne 120 détermine la durée d'affichage. La ligne 140 définit la durée du jeu, et la fin de partie est traitée par la ligne 160.

La boucle I (210-240) réalise le dessin grand format du chiffre frappé par le joueur.

Le contrôle de l'exactitude du résultat est effectué par la ligne 250.

POUR JOUER :

Le programme sera lancé par RUN, puis un premier chiffre apparaît, après une courte pause, vers le milieu de l'écran. Le joueur devra réagir suffisamment vite, tant qu'il reste à l'affichage, et le retaper. Le chiffre qu'il a tapé se dessinera alors en haut à gauche de l'écran, en grand format.

Si le résultat est exact, le joueur marquera autant de points que la valeur du chiffre.

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 CNT et C.

EXTENSIONS POSSIBLES :

On peut envisager de modifier le jeu pour faire participer plusieurs joueurs, en modifiant éventuellement l'affichage et en codant les touches du clavier mises à la disposition de chacun des joueurs.

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

```

10  REM ADRESSE
20  HOME
30  S = 0:T = 50
40  T1 = T
50  C$ = "#####    ##  #"
60  B$ = "144413333313121131314413
      31213112141133331414114131"
70  X = INT ( RND (1) * 15 + 1)
80  Y = INT ( RND (1) * 15 + 16)
90  N = INT ( RND (1) * 10)
100 VTAB X: HTAB Y: PRINT N
110 R = R + 10
120 FOR I = 0 TO RND (1) * 350 +
      500: NEXT I
130 K = PEEK ( - 16384): IF K >
      128 THEN K = K - 128: GOTO 2
      00
140 T = T - N
150 VTAB X: HTAB Y: PRINT "    "
160 IF T > = 0 THEN 70
170 HOME
180 PRINT : PRINT "VOTRE SCORE E
      ST DE ";S;"/";T1
190 END
200 POKE - 16368,0: IF K < 48 OR
      K > 57 THEN 140
210 FOR I = 1 TO 5
220 A = 4 * VAL ( MID$ (B$,5 * (
      K - 48) + I,1)) - 3
230 VTAB 2 + I: HTAB 2: PRINT MID$
      (C$,A,4)
240 NEXT I
250 IF K - 48 = N THEN S = S + N
260 GOTO 140

```

Sous-marin

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : réflexe (1 seul joueur).

Difficulté du jeu : facile.

LE JEU :

Vous manœuvrez en surface un chasseur de sous-marins, et des sous-marins passent très régulièrement en dessous de votre unité, à différentes profondeurs.

Vous pouvez, à tout moment, larguer des grenades sous-marines et vous pouvez même envisager de toucher plusieurs fois le même sous-marin.

Votre score sera affiché en permanence, sur la gauche de l'écran (rapport nombre de coups au but sur nombre de sous-marins apparus).

LE PROGRAMME :

La ligne 50 permet à l'APPLE II de choisir la profondeur du sous-marin.

La ligne 70 dessine votre bâtiment.

Les lignes 80 et 90 affichent le score et dessinent la surface de l'eau.

La ligne 100 dessine le sous-marin.

Le tir est effectué par un programme spécifique placé aux lignes 180 à 220.

Le déplacement (très rapide) de votre navire est réalisé par les lignes 150 et 160.

Le test, permettant de savoir si l'on a touché ou non, est effectué en ligne 120.

POUR JOUER :

Taper RUN pour commencer la partie.

Le bâtiment sera déplacé grâce aux flèches droite et gauche.

La touche 0 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 CNT et C ou d'attendre 99 sous-marins.

EXTENSIONS POSSIBLES :

Il sera assez élémentaire d'améliorer la qualité graphique du jeu, d'utiliser la couleur et les sons, de faire apparaître plusieurs sous-marins se déplaçant à des vitesses différentes, éventuellement même, de permettre aux sous-marins de tirer...

```

10  REM LE SOUS-MARIN
20  P = 0: C = 15: S = 5
30  HOME
40  VTAB S: CALL - 868
50  T = 34: S = INT ( RND (1) * 10
    + 5): P = P + .1
60  X = 1
70  VTAB 2: HTAB C: PRINT "   ###
    ###  ": VTAB 3: HTAB C: PRINT
    "   C###/   "
80  VTAB 4: PRINT "=====
    ====="
90  VTAB 19: PRINT "SCORE ... : "
    ;P
100 VTAB S: HTAB T: PRINT " C==/
    "
110 IF X = 1 THEN Y = C + 5
120 IF S = X AND (Y > T) AND Y <
    T + 5 THEN P = P + 1: VTAB S
    : HTAB T: PRINT CHR$ (7) +
    "*" + CHR$ (7) + "*" + CHR$
    (7) + "*" + CHR$ (7) + "*" +
    CHR$ (7) + "*" + CHR$ (7) +
    "*"
130 T = T - 1: IF P = 99 THEN VTAB
    19: INVERSE : PRINT "=== C'E
    ST FINI ... ": NORMAL : END

140 IF T < 1 THEN 40
150 K = PEEK ( - 16384): IF K >
    128 THEN POKE - 16368, 0
160 C = C + 2 * ((K = 149 AND C <
    31) - (K = 136 AND C > 1))
170 IF K < > 176 AND X = 1 THEN
    70
180 VTAB X: HTAB Y
190 PRINT " ": X = X + 1
200 IF X > S THEN 60
210 VTAB X: HTAB Y: PRINT "0"
220 GOTO 70

```

Diago

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : réflexe (1 seul joueur).

Difficulté du jeu : difficile.

LE JEU :

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

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

LE PROGRAMME :

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

Le temps est décompté par les lignes 40 à 90.

Les points sont imprimés et la ligne diagonale est fabriquée par la ligne 120.

La pression de la touche 0 est détectée en ligne 180, et la ligne 160 contrôle la fin de partie (aucun réflexe).

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

POUR JOUER :

On lancera l'exécution par RUN.

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

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

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

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

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

EXTENSIONS POSSIBLES :

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

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

```
10 REM DIAGO
20 HOME
30 X = INT ( RND (1) * 18 + 3)
40 S = 5
50 PRINT "=== LE BON RÉFLEXE EST
   A ";X
60 VTAB 8: PRINT "=== ATTENTION
   === IL VOUS RESTE ";S;" S. "

70 S = S - 1
80 PRINT CHR$( 7);: FOR I = 1 TO
   760: NEXT I
90 IF S < > 0 THEN 60
100 HOME
110 U = 0
120 VTAB 24: HTAB 1: PRINT TAB(
   U + 1);"O"
130 FOR I = 1 TO 30: NEXT I
140 U = U + 1
150 K = PEEK ( - 16384): IF K >
   128 THEN 180
160 IF U < = 22 THEN 120
170 PRINT : PRINT : PRINT : INVERSE
   : PRINT "=== PERDU ... ";U;
   " AU LIEU DE ";X;".": NORMAL
   : END
180 POKE - 16368,0: IF K < > 1
   76 THEN 160
190 IF U < > X THEN GOTO 170
200 PRINT : PRINT : PRINT : INVERSE
   : PRINT "=== BRAVO ...": NORMAL

210 PRINT : PRINT "=== UN AUTRE
   ESSAI ? (O/N) : ";: GET K$
220 IF K$ < > "N" THEN 20
```

Décomaque

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : solution du programme Nicomaque.

Difficulté du jeu : aucune.

LE JEU :

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

LE PROGRAMME :

Les lignes 30 à 50 affichent quelques explications.

La ligne 110 attend que vous frappiez un reste, qui sera refusé s'il est incompatible avec la division par le nombre affiché.

La ligne 120 emmagasine les restes dans une variable non dimensionnée A (S).

La boucle J (130) dessine des étoiles selon les restes introduits.

Les lignes 180 à 200 gèrent la fin de partie.

POUR JOUER :

On lance le programme par RUN. Un texte de présentation est affiché, puis le message "RESTE DANS LA DIVISION PAR 3" s'affiche.

On doit répondre 0, 1 ou 2, et l'on obtient le message "RESTE DANS LA DIVISION PAR 5" etc.

La solution est affichée dès que l'on a donné les 3 restes.

Pour une autre partie, taper O à la question UN AUTRE ESSAI (O/N).

Tapez N pour finir.

EXTENSIONS POSSIBLES :

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

```

10  REM DECOMAQUE
20  HOME
30  VTAB 4: PRINT "VOUS DEVEZ ENT
    RER LES TROIS RESTES"
40  VTAB 6: PRINT "DANS LES DIVIS
    IONS PAR 3, 5 ET 7"
50  VTAB 8: PRINT "D'UN NOMBRE EN
    TIER INFERIEUR A 100"
60  I = 3:S = 1
70  VTAB 11: PRINT "RESTE DANS LA
    DIVISION PAR : "
80  PRINT : PRINT TAB( 7);3; TAB(
    17);5; TAB( 27);7
90  PRINT : PRINT TAB( 7);".."; TAB(
    17);"...."; TAB( 27);".....
    "
100 VTAB 11: HTAB 28: PRINT I;"
    : ";
110 VTAB 11: HTAB 32: GET K$: IF
    K$ < "0" OR K$ > CHR$ (47 +
    I) THEN PRINT CHR$ (7);: GOTO
    110
120 A(S) = ASC (K$) - 48: IF K$ =
    "0" THEN 140
130 FOR J = 1 TO A(S): VTAB 15: HTAB
    S * 10 - 4 + J: PRINT "*": NEXT
    J
140 S = S + 1:I = I + 2: IF I < 8
    THEN 100
150 VTAB 11: HTAB 28: CALL - 86
    8: PRINT " : "
160 T = A(1) * 70 + A(2) * 21 + A
    (3) * 15
170 IF T > 105 THEN T = T - 105:
    GOTO 170
180 VTAB 18: INVERSE : PRINT " L
    A SOLUTION EST ";T;". "
190 VTAB 22: PRINT "UN AUTRE ESS
    AI ? (O/N) : ";: NORMAL : GET
    K$
200 IF K$ < > "N" THEN 20

```

Chasse au sous-marin

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de réflexion.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

La boucle I (50-110) affiche le terrain.

La ligne 140 définit la position initiale du sous-marin ; la ligne 150 permet son déplacement, contrôlé par la ligne 160 (on reste dans les limites du terrain).

La ligne 180 affiche votre position.

Les lignes 200 à 220 contrôlent votre déplacement.

La distance est calculée en ligne 230, affichée en ligne 240 et testée en lignes 250 et 260.

Le test de fin de partie est effectué à la ligne 290.

POUR JOUER :

On lance le programme par RUN et on entre la dimension ; la grille de jeu s'affiche. Vous êtes au départ en position (0, 0) et vous vous déplacez avec les flèches.

Après chaque déplacement, la nouvelle distance du sous-marin s'affiche ; à chaque coup le sous-marin se déplace. Si vous réussissez à vous placer à sa verticale, la distance 0 est affichée et il ne bouge plus.

Vous tirez en donnant une profondeur entière (entre 2 et 4, bornes comprises). Si vous le ratez, il se replace n'importe où, et on recommence ; si vous le coulez, vous obtenez l'affichage BRAVO et la partie s'arrête.

On peut arrêter à tout moment le programme en utilisant les touches CNT et C.

EXTENSIONS POSSIBLES :

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

```

10 REM CHASSE AU SOUS-MARIN
20 HOME
30 VTAB 5: INPUT "DIMENSION DE L
  A GRILLE (2 A 8) : ";H
40 HOME
50 FOR I = 0 TO H
60 VTAB 1: HTAB I * 3 + 4: PRINT
  I
70 VTAB I * 2 + 3: HTAB 1: PRINT
  I
80 FOR J = 0 TO H
90 VTAB I * 2 + 3: HTAB J * 3 +
  4: PRINT "."
100 NEXT J
110 NEXT I
120 VTAB 3: HTAB 32: PRINT " I"
  : HTAB 32: PRINT " !": HTAB
  32: PRINT "J-*K": HTAB 32: PRINT
  " !": HTAB 32: PRINT " M"
130 I = 1: X = 0: Y = 0: A = 0: B = 0

140 V = INT ( RND (1) * H): U = INT
  ( RND (1) * H)
150 U = U + I: V = V + I
160 IF (U < 0 OR U > H OR V < 0 OR
  V > H) THEN I = - I
170 VTAB 2 * A + 3: HTAB 3 * B +
  4: PRINT "."
180 VTAB 2 * X + 3: HTAB 3 * Y +
  4: PRINT "#"
190 A = X: B = Y
200 VTAB 10: HTAB 31: PRINT "A V
  OUS ";: GET K$
210 Y = Y + (K$ = "K" AND Y < H) -
  (K$ = "J" AND Y > 0)
220 X = X + (K$ = "M" AND X < H) -
  (K$ = "I" AND X > 0)
230 D = INT ( SQRT ( ABS (X - V) ^
  2 + ABS (Y - U) ^ 2))
240 VTAB 13: HTAB 31: CALL - 86
  8: PRINT "DIST : ";D
250 IF D = 0 THEN 280
260 IF D < 3 THEN 150
270 GOTO 170
280 VTAB 14: HTAB 31: CALL - 86
  8: PRINT "PROF : ";: GET D$:
  PRINT D$: D = VAL (D$): IF
  D < 2 OR D > 4 THEN PRINT CHR$
  (7);: GOTO 280
290 IF D = INT ( RND (1) * 3 +
  2) THEN VTAB 21: INVERSE : PRINT
  "=== BRAVO ... ===": NORMAL
  : END
300 VTAB 14: HTAB 31: CALL - 86
  8
310 GOTO 140

```

Clavier

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : jeu de réflexe (apprentissage du clavier).

Difficulté du jeu : difficile.

LE JEU :

Une reproduction du clavier de votre ordinateur apparaît sur l'écran ; le caractère qu'il convient de frapper au clavier apparaît en inversé. Plus l'on frappe vite, et plus le point se déplace rapidement (le temps écoulé depuis le début s'affiche en continu).

LE PROGRAMME :

Deux tableaux X et Y, dimensionnés à 26, contiennent les positions de chacune des lettres de l'alphabet (ligne, colonne). Vous pouvez modifier la disposition du clavier en modifiant les DATA en 40 (nombre de touches par ligne) et en 50 (ordre des touches).

Le sous-programme (280-310) affiche une touche.

La ligne 150 choisit la lettre.

La ligne 200 teste votre frappe.

POUR JOUER :

On lance le programme par RUN. L'on dispose alors de quelques secondes pour frapper la touche qui occupe la colonne au-dessus de l'étoile, puis une autre étoile est affichée, etc.

La durée de la partie n'est pas limitée ; il suffit de taper 0 pour obtenir l'affichage du score, et du temps total.

Faire RUN pour une autre partie.

EXTENSIONS POSSIBLES :

On pourra envisager de faire clignoter les lettres, ou de les afficher en couleur, au lieu d'indiquer la colonne par une balle. On pourra afficher les lettres devinées justes en bas de l'écran, et éventuellement s'entraîner à taper un texte qui défile en haut de l'écran, lettre par lettre.

```

10 REM CLAVIER
20 HOME
30 DIM N(3),X(26),Y(26)
40 DATA 10,19,26
50 DATA Q,W,E,R,T,Y,U,I,O,P,A,S
   ,D,F,G,H,J,K,L,Z,X,C,V,B,N,M

60 FOR I = 1 TO 3: READ N(I): NEXT

70 FOR I = 1 TO 26: READ K$:K =
   ASC (K$) - 64
80 FOR J = 3 TO 1 STEP - 1: IF
   I < = N(J) THEN X(K) = J
90 NEXT J
100 Y(K) = I - N(X(K) - 1)
110 GOSUB 280
120 NEXT I
130 VTAB 18: INVERSE : PRINT " O
   ESSAIS JUSTES SUR O"
140 VTAB 20: PRINT "VOTRE TEMPS
   EST DE 0": NORMAL
150 K = INT ( RND (1) * 26 + 1)
160 INVERSE : GOSUB 280: NORMAL

170 FOR F = 1 TO 200 * RND (1) +
   500
180 L = PEEK ( - 16384):T = T +
   1: INVERSE : VTAB 20: HTAB 2
   0: PRINT T;". ": NORMAL
190 IF L < 128 THEN NEXT F: GOTO
   220
200 POKE - 16368,0: IF L = K +
   192 THEN S = S + 1
210 IF L = 176 THEN 250
220 P = P + 1: VTAB 18: INVERSE :
   PRINT " ";S;: HTAB 23: PRINT
   P;". ": NORMAL
230 GOSUB 280
240 GOTO 150
250 VTAB 18: INVERSE : PRINT " "
   ;S;" ESSAIS JUSTES SUR ";P;"
   . "
260 VTAB 20: PRINT "VOTRE TEMPS
   EST DE ";T;". ": NORMAL
270 END
280 VTAB X(K) * 4 - 3: HTAB Y(K)
   * 4 - 4 + X(K): PRINT " "
290 VTAB X(K) * 4 - 2: HTAB Y(K)
   * 4 - 4 + X(K): PRINT " "; CHR$
   (K + 64);" "
300 VTAB X(K) * 4 - 1: HTAB Y(K)
   * 4 - 4 + X(K): PRINT " "
310 RETURN

```

Le lézard d'Alice

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : solitaire.

Difficulté du jeu : assez difficile.

LE JEU :

Il s'inspire très vaguement d'une idée de LEWIS CARROLL, dans *Alice au Pays des Merveilles*. Dans l'histoire originale, Alice grandissait ou diminuait, alors que dans le jeu, pour des raisons d'encombrement de la mémoire, c'est un lézard qui change de taille.

Deux verres apparaissent sur l'écran : l'un contient le liquide qui fait grandir, l'autre le liquide qui fait rapetisser (malheureusement ce n'est pas toujours le même verre). Le joueur devra indiquer la quantité de liquide qu'il désire boire et le lézard changera de taille en fonction de cette quantité. Le but du jeu est de parvenir à lui rendre sa taille initiale.

LE PROGRAMME :

Les lignes 80 à 130 dessinent les verres et le sous-programme 330-380 dessine le lézard.

La boucle I (220-240) remplit les verres avec la quantité indiquée en ligne 210.

La variation de taille du lézard est définie par la ligne 270, avec contrôle par les lignes 280 et 290.

Un message gagnant sera affiché par la ligne 310 si l'on a trouvé la bonne taille.

POUR JOUER :

On lance le programme par RUN : les deux verres se dessinent, puis apparaît la question QUEL COTE à laquelle on répond en tapant G ou D ; on obtient l'affichage QUELLE QUANTITE ; on entre alors une quantité (de 0 à 130) et on frappe RETURN.

Le verre désigné se remplit sous vos yeux, puis, après quelques instants, le lézard est affiché.

EXTENSIONS POSSIBLES :

Puisqu'il s'agit d'un jeu essentiellement graphique, il sera intéressant d'améliorer les dessins, d'introduire la couleur, de modifier le caractère aléatoire du choix des verres, etc.

Pourquoi ne pas relire *Alice au Pays des Merveilles*, les idées n'y manquent pas.


```

10 REM LE LEZARD D'ALICE
20 HOME
30 D$ = "#####
    ###"
40 L$ = "   ###   ###0#   #####
    #####   ##   ## "
50 F$ = "       #   #####  ##
    ##  "
60 N = 12
70 A$ = "à"
80 FOR I = 1 TO 9
90 VTAB I + 8: HTAB 3: PRINT A$;
    : HTAB 10: PRINT A$;; HTAB 1
    3: PRINT A$;; HTAB 20: PRINT
    A$
100 NEXT
110 GOSUB 330
120 FOR I = 3 TO 10: VTAB 18: HTAB
    I: PRINT A$; NEXT I
130 FOR I = 13 TO 20: VTAB 18: HTAB
    I: PRINT A$; NEXT I
140 VTAB 20: HTAB 4: PRINT "GAUC
    HE";: HTAB 14: PRINT "DROITE
    "
150 VTAB 13: HTAB 23: PRINT "COT
    E ? (G/D) : ";: GET K$
160 IF K$ = "D" THEN VTAB 13: HTAB
    28: CALL - 868: PRINT "DROI
    T": GOTO 190
170 IF K$ = "G" THEN VTAB 13: HTAB
    28: CALL - 868: PRINT "GAUC
    HE": GOTO 190
180 PRINT CHR$(7);: GOTO 150
190 U = 4 + 10 * (K$ = "D")
200 FOR I = 1 TO 9: VTAB I + 8: HTAB
    4: PRINT "       ";: HTAB 14:
    PRINT "       ": NEXT I
210 VTAB 16: HTAB 23: INPUT "QUA
    NTITE : ";Q

```

```

220  FOR I = 1 TO Q / 14.45 + 1
230  VTAB 18 - I: HTAB U: INVERSE
      : PRINT "      "
240  NEXT I
250  NORMAL
260  H = 4 * ( RND (1) < .5) - 2
270  N = N + H * SQR (Q)
280  IF N < 0 THEN N = 0
290  IF N > 27 THEN N = 27
300  GOSUB 330
310  IF N = 22 THEN VTAB 22: INVERSE
      : PRINT "=== BRAVO ... ": NORMAL
      : END
320  GOTO 150
330  FOR I = 1 TO 6
340  VTAB I: CALL - 868: PRINT MID$
      (L$,8 * I - 7,8);
350  IF I = 4 AND N < > 0 THEN PRINT
      LEFT$ (D$,N);
360  HTAB N + 9: PRINT MID$ (F$,
      5 * I - 4,5)
370  NEXT
380  RETURN

```

Calculatrice

REFERENCES :

Difficulté de programmation : très difficile.

Catégorie de jeu : simulation d'une calculatrice.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

Les variables B\$ et C\$, des lignes 20 et 30, contiennent les informations codées qui permettront le dessin des chiffres en grand format.

Les lignes 60 à 150 s'occupent de la gestion des entrées.

La boucle I (160-190) réalise la représentation, en grand format, des chiffres sur l'écran, au fur et à mesure des entrées.

Le point décimal et le signe - sont gérés par les lignes 200 et 210.

Le nombre total des symboles entrés est contrôlé par la ligne 220.

Le résultat est inscrit grâce à un programme fonctionnant en automatique qui se trouve aux lignes 270 à 400 (l'automatisme est géré par la ligne 230, dans le programme principal).

Les sous-programmes (lignes 400 à 470) effectuent l'opération indiquée, selon la valeur de Z (ligne 310).

Chaque résultat est mémorisé en A (F-1) (ligne 330).

POUR JOUER :

On devra lancer le programme par RUN.

Puis, on tape les chiffres du premier nombre à entrer qui vont s'afficher un par un, au fur et à mesure (on ne pourra entrer plus de 8 symboles, y compris le signe - et le point décimal). La frappe d'un signe opératoire laisse l'affichage intact, mais l'introduction du prochain chiffre ou symbole provoque l'effacement immédiat et l'affichage du nouveau symbole. Il est à remarquer que les chiffres s'introduisent de gauche à droite, sens de l'écriture usuelle.

L'appareil accepte, comme signes opératoires les symboles +, -, *, / ; le signe = provoquera l'affichage du résultat. Les opérations sont enchaînées automatiquement, le résultat précédent étant pris en compte comme opérateur du calcul suivant.

Pour faire une autre opération, il faudra taper CNT et C, puis RUN.

Taper F pour sortir du programme.

EXTENSIONS POSSIBLES :

On devra, tout d'abord, améliorer l'affichage du signe - (représenté, ici, par un trait), puis on pourra améliorer le dessin des chiffres, on peut aussi envisager une extension à plus de 8 chiffres, autoriser des opérations plus complexes (COS, SIN, ...), ajouter des mémoires, etc.

```
10 REM CALCULATRICE
20 C$ = "#####    ##  #"
30 B$ = "144413333313121131314413
      31213112141133331414114131"
40 HOME : F = 1 : L = 0
50 N = 0 : T = 0 : K = 0 : R$ = ""
60 GET K$
70 IF K = 1 THEN N = 0 : R$ = "" : K
   = 0
80 IF K$ = "=" THEN 260
90 X = (K$ = "+") + 2 * (K$ = "--"
   ) + 3 * (K$ = "*") + 4 * (K$
   = "/" )
100 R$ = R$ + K$
110 IF X < > 0 THEN 260
120 IF R$ = "-" THEN K$ = R$ : GOTO
   200
130 IF K$ < "." OR K$ > "9" OR K
   $ = "/" THEN 60
140 IF N = 0 THEN HOME
150 IF K$ = "." OR K$ = "--" THEN
   200
160 FOR I = 1 TO 5
170 S = 4 * VAL ( MID$ (B$, 5 * VAL
   (K$) + I, 1)) - 3
180 VTAB 3 + I : HTAB N + 1 : PRINT
   MID$ (C$, S, 4) "   "
190 NEXT I
200 IF K$ = "--" THEN HOME : VTAB
   7 : HTAB N + 2 : PRINT "###"
210 IF K$ = "." THEN VTAB 8 : HTAB
   N + 3 : PRINT "#"
220 N = N + 5 * (N < 35)
230 IF T < > 0 THEN T = T + 1 : IF
   T < = LEN (R$) THEN 390
240 IF K = 1 AND L < > 1 THEN 5
   0
250 I = 0 : GOTO 60
260 IF R$ = "-" AND A(F) = 0 THEN
   120
270 N = 0 : K = 1
```

```

280 IF (K$ = "*" OR K$ = "/" ) AND
    VAL (R$) = 0 THEN A(F) = 1:
    GOTO 300
290 A(F) = VAL (R$)
300 IF X < > 0 THEN Z = X
310 ON Z GOSUB 410,420,430,440
320 IF A$ < > "=" THEN F = F +
    1: GOTO 340
330 F = 2
340 A(F - 1) = VAL (R$)
350 IF VAL (R$) = 0 THEN A(F -
    1) = A(F)
360 R$ = STR$ ( VAL (R$))
370 IF LEFT$ (R$,1) = " " THEN
    R$ = MID$ (R$,2)
380 T = 1
390 K$ = MID$ (R$,T,1)
400 GOTO 140
410 R$ = STR$ (A(F - 1) + A(F)):
    GOTO 450
420 IF F < > 1 THEN R$ = STR$
    (A(F - 1) - A(F)): GOTO 450
430 IF F < > 1 THEN R$ = STR$
    (A(F - 1) * A(F)): GOTO 450
440 IF F < > 1 THEN R$ = STR$
    (A(F - 1) / A(F))
450 IF LEN (R$) > 7 THEN R$ = LEFT$
    (R$,7)
460 A(F) = VAL (R$):L = 1
470 RETURN

```

Echecs

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de réflexion, pour deux joueurs.

Difficulté du jeu : difficile.

LE JEU :

Le programme propose seulement à deux joueurs de disputer une partie en se servant de l'écran du téléviseur comme plateau de jeu. Il n'y a pas de contrôle de la validité des coups, mais seulement gestion de la position des pièces sur l'échiquier. La couleur permet de distinguer chaque camp.

LE PROGRAMME :

Les lignes 50 à 190 dessinent l'échiquier. Les lignes 200 à 260 définissent les conditions initiales, à l'aide d'un tableau : R (I, J).

Les lignes 300 à 400 permettent l'entrée d'un coup sous la forme classique (E2-E4).

Le sous-programme 460-510 affiche une pièce.

Le sous-programme 520-570 ré-affiche une case vide.

POUR JOUER :

On lance l'exécution par RUN.

L'échiquier est dessiné, puis les pièces, et enfin l'inscription JOUEUR NOIR ; on entre la colonne et la ligne de départ, sous la forme E 7 par exemple, puis la colonne et la ligne d'arrivée. Le déplacement est alors effectué, et c'est au tour de l'autre joueur.

Pour arrêter, il suffit de taper le slash (/) en colonne de départ.

EXTENSIONS POSSIBLES :

On pourra chercher à réaliser un graphisme plus précis, introduire le contrôle du mouvement des pièces. Et pourquoi pas, écrire un programme qui permette à l'APPLE II de se comporter en partenaire honorable. Bon courage !

```

10  REM ECHIQUEUR
20  HOME
30  DIM R(8,8)
40  C$(0) = "BLANC":C$(1) = "NOIR"
   "
50  PRINT "      A  B  C  D  E  F
      G  H": PRINT "      "
   "
60  B$ = "      ":N$ = "####"
70  FOR I = 1 TO 8
80  FOR J = 1 TO 2
90  I$ = "      ": IF J = 2 THEN I$ =
      STR$(I) + " "
100 PRINT I$;"!";
110 FOR K = 1 TO 8
120 IF AG = 0 THEN AG = 1: PRINT
      B$:: GOTO 140
130 AG = 0: INVERSE : PRINT B$:: NORMAL
140 NEXT K
150 PRINT "!"
160 NEXT J
170 AG = NOT (AG)
180 NEXT I
190 VTAB 18: HTAB 1: PRINT "8 !"
   :: FOR I = 1 TO 4: INVERSE :
      PRINT B$:: NORMAL : PRINT "
      ____": NEXT I: PRINT "!"
200 DATA R,D,F,C,T,O
210 DATA 5,4,3,2,1,3,4,5,6,6,6,6
   ,6,6,6,6
220 FOR I = 1 TO 6: READ P$(I): NEXT
      I
230 FOR A = 1 TO 2: FOR B = 1 TO
      8
240 I = A:J = B: READ R(I,J):C =
      0: GOSUB 460
250 I = 9 - A:J = B:R(I,J) = R(A,
      B) + 10:C = 1: GOSUB 460
260 NEXT B,A
270 H = 1
280 VTAB 4: HTAB 29: INVERSE : PRINT
      "JOUEUR ":C$(H): NORMAL
290 VTAB 7: HTAB 30: PRINT "VOTR
      E COUP": HTAB 30: PRINT "EX
      : E2 E4"
300 VTAB 14: HTAB 31: GET K$: PRINT
      K$: IF K$ = "/" THEN VTAB 2
      0: END

```

```

310 IF K$ < "A" OR K$ > "H" THEN
    PRINT CHR$(7);: GOTO 300
320 VTAB 14: HTAB 32: GET L$: PRINT
    L$: IF ASC (L$) = 8 THEN 300
330 IF L$ < "1" OR L$ > "8" THEN
    PRINT CHR$(7);: GOTO 320
340 I = VAL (L$):J = ASC (K$) -
    64:C = (R(I,J) > = 10): IF
    R(I,J) - C * 10 = 0 OR C < >
    H THEN PRINT CHR$(7);: GOTO
    300
350 VTAB 14: HTAB 34: GET K$: PRINT
    K$: IF ASC (K$) = 8 THEN 300
360 IF K$ < "A" OR K$ > "H" THEN
    PRINT CHR$(7);: GOTO 350
370 VTAB 14: HTAB 35: GET L$: PRINT
    L$: IF ASC (L$) = 8 THEN 350
380 IF L$ < "1" OR L$ > "8" THEN
    PRINT CHR$(7);: GOTO 370
390 I1 = VAL (L$):J1 = ASC (K$)
    - 64
400 IF R(I1,J1) < > 0 AND ((R(I
    1,J1) > 10) = (R(I,J) > 10))
    THEN PRINT CHR$(7);: GOTO
    350
410 GOSUB 520
420 R(I1,J1) = R(I,J):R(I,J) = 0:
    I = I1:J = J1: GOSUB 460
430 H = ABS (H - 1)
440 GOTO 280
450 END
460 REM
470 C = (R(I,J) > 10):R = R(I,J) -
    C * 10
480 X = 2 * I + 2:Y = 3 * J + 2
490 IF C = 1 THEN INVERSE
500 VTAB X: HTAB Y: PRINT P$(R)
510 NORMAL : RETURN
520 X = 2 * I + 2:Y = 3 * J + 2
530 A$ = " "
540 IF I - INT (I / 2) * 2 < >
    J - INT (J / 2) * 2 THEN INVERSE : GOTO 560
550 IF I = 8 THEN A$ = " _ "
560 VTAB X: HTAB Y: PRINT A$: NORMAL
570 RETURN

```


Horloge 2

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : simulation.

Difficulté du jeu : aucune.

LE JEU :

Ce programme simule une horloge digitale grand format en utilisant les caractères d'origine de l'APPLE II, notablement agrandis. Il présente la particularité de faire appel au crayon optique de l'APPLE II pour la mise à l'heure, et éventuellement l'arrêt ou le départ de l'horloge.

LE PROGRAMME :

Ce programme a pour base HORLOGE.

Les lignes 340 à 610 permettent la remise à l'heure des heures (390 à 490) et des minutes (500 à 590).

POUR JOUER :

Dès le lancement par RUN, le programme se place en attente de l'heure. Une fois l'horloge en marche, la frappe de la touche RETURN permet de la régler.

Faire CTRL et C pour arrêter le programme. On pourra régler l'horloge en jouant sur la valeur de I, en ligne 280.

EXTENSIONS POSSIBLES :

On peut améliorer l'affichage des secondes, ajouter une possibilité de réveil, prévoir un agenda automatique, faire apparaître un coucou chantant au moment du passage de l'heure.

```

10 REM HORLOGE 2
20 HOME
30 INPUT "HEURE (FORME XXXX) : "
   ;A$
40 HOME
50 INVERSE
60 VTAB 18: PRINT "POUR REMETTRE
   A L'HEURE, TAPER 'RETURN'"
70 K = 1
80 A = VAL ( MID$ (A$,K,1))
90 IF A < > 1 AND A < > 4 THEN
   VTAB 1: HTAB 6 * K - 4: PRINT
   " "
100 FOR J = 1 TO 5
110 IF A = 0 OR (A > 3 AND A < >
   7) THEN VTAB J: HTAB 6 * K -
   4: PRINT " "
120 IF A < > 5 AND A < > 6 THEN
   VTAB J: HTAB 6 * K - 2: PRINT
   " "
130 IF A = 0 OR A = 2 OR A = 6 OR
   A = 8 THEN VTAB J + 4: HTAB
   6 * K - 4: PRINT " "
140 IF A < > 2 THEN VTAB J + 4
   : HTAB 6 * K - 2: PRINT " "
150 NEXT J
160 IF A > 1 THEN VTAB 5: HTAB
   6 * K - 3: PRINT " "
170 IF A < > 1 AND A < > 4 AND
   A < > 7 THEN VTAB 9: HTAB
   6 * K - 4: PRINT " "
180 K = K + 1
190 IF K < 5 THEN 80
200 B$ = A$
210 A = VAL (A$) + 1
220 IF A = 2360 THEN A = 0
230 A$ = RIGHT$ ("000" + STR$ (
   A),4)
240 IF RIGHT$ (A$,2) = "60" THEN
   A$ = RIGHT$ ("0" + STR$ ( VAL
   ( LEFT$ (A$,2)) + 1),2) + "0
   0"
250 H = 1
260 FOR J = 1 TO 120
270 X = PEEK ( - 16384): IF X >
   127 THEN 340
280 FOR I = 1 TO 40
290 VTAB 9: HTAB 12:X$ = " ": IF
   H = 1 THEN X$ = ":"
300 PRINT X$: NEXT I
310 H = ABS (H - 1)
320 NEXT J
330 GOTO 40

```

```

340 POKE - 16368,0: IF X < > 1
    41 THEN 280
350 VTAB 15: HTAB 1: CALL - 958
    : NORMAL
360 PRINT "VOUS POUVEZ REMETTRE
    A L'HEURE ": PRINT "SUCCESSI
    VEMENT HEURES PUIS MINUTES."
    : PRINT : PRINT "UTILISEZ LE
    S FLECHES <- ET -> ": PRINT
    "
370 Y$ = LEFT$ (B$,2):Z$ = RIGHT$
    (B$,2):Y = VAL (Y$):Z = VAL
    (Z$)
380 VTAB 21: HTAB 5: INVERSE : PRINT
    " ";Y$;" ";Z$;" ": NORMAL
390 VTAB 21: HTAB 6: FLASH : PRINT
    Y$
400 X = PEEK ( - 16384): IF X <
    128 THEN 400
410 POKE - 16368,0
420 IF X < > 136 THEN 450
430 Y = Y - 1: IF Y = - 1 THEN Y
    = 23
440 Y$ = RIGHT$ ("0" + STR$ (Y)
    ,2): GOTO 390
450 / IF X < > 149 THEN 480
460 Y = Y + 1: IF Y = 24 THEN Y =
    0
470 Y$ = RIGHT$ ("0" + STR$ (Y)
    ,2): GOTO 390
480 IF X < > 141 THEN PRINT CHR$
    (7);: GOTO 400
490 VTAB 21: HTAB 6: INVERSE : PRINT
    Y$
500 VTAB 21: HTAB 10: FLASH : PRINT
    Z$
510 X = PEEK ( - 16384): IF X <
    128 THEN 510
520 POKE - 16368,0
530 IF X < > 136 THEN 560
540 Z = Z - 1: IF Z = - 1 THEN Z
    = 59
550 Z$ = RIGHT$ ("0" + STR$ (Z)
    ,2): GOTO 500
560 IF X < > 149 THEN 590
570 Z = Z + 1: IF Z = 60 THEN Z =
    0
580 Z$ = RIGHT$ ("0" + STR$ (Z)
    ,2): GOTO 500
590 IF X < > 141 THEN PRINT CHR$
    (7);: GOTO 510
600 A$ = Y$ + Z$
610 GOTO 40

```

Danger spatial

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de réflexe, pour un joueur.

Difficulté du jeu : moyenne.

LE JEU :

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

LE PROGRAMME :

La position de chaque météorite est déterminée par les lignes 40 et 60 ; elle est imprimée par les lignes 50 et 70.

L'avance des météorites est réalisée par la ligne 80.

Les lignes 80 et 90 vérifient si vous avez ou non heurté une météorite.

La ligne 110 dessine votre vaisseau spatial, et la ligne 140 permet de le déplacer.

Le compteur se trouve en ligne 150.

POUR JOUER :

Le jeu est lancé par l'ordre RUN.

Votre vaisseau apparaît vers le milieu de l'écran, et les météorites viennent à votre rencontre en partant du bas de l'écran. Pour vous déplacer, vous pouvez utiliser les touches de déplacement horizontal (→ et ←).

On peut arrêter la partie à tout moment en tapant CNT et C. Sinon, le jeu s'arrête lorsque vous heurtez une météorite, et le score est alors affiché.

EXTENSIONS POSSIBLES :

Il serait intéressant de rendre le jeu de plus en plus difficile en faisant apparaître de plus en plus de météorites au fur et à mesure du déroulement de la partie.

On peut aussi essayer de détruire les météorites en les pulvérisant au fur et à mesure de leur apparition.

```

10  REM DANGER SPATIAL
20  HOME
30  X = 18:Z = 0:Y = X
40  H = INT ( RND (1) * 20 + 1)
50  IF RND (1) < .7 THEN VTAB 2
    4: HTAB H: INVERSE : PRINT "
      ";; NORMAL :Z = PEEK ( - 1
        6336)
60  K = INT ( RND (1) * 20 + 1)
70  IF RND (1) < .6 THEN VTAB 2
    4: HTAB K: INVERSE : PRINT "
      ";; NORMAL :Z = PEEK ( - 1
        6336)
80  PRINT :B = PEEK (1447 + X):C
    = PEEK (1448 + X)
90  VTAB 11: HTAB Y - 1: PRINT "
    " : VTAB 10: HTAB Y - 1: PRINT
    "
100 IF B = 32 OR C = 32 THEN 170

110 VTAB 12: HTAB X: PRINT "##"
120 K = PEEK ( - 16384): IF K >
    127 THEN POKE - 16368,0
130 Y = X
140 X = X + (K = 149 AND X < 38) -
    (K = 136 AND X > 2)
150 W = W + 1
160 GOTO 40
170 VTAB 12: HTAB X: FLASH : PRINT
    "##": NORMAL
180 VTAB 24: PRINT : INVERSE : PRINT
    "DOMMAGE ... VOTRE SCORE EST
    DE ":W;". ": NORMAL

```

Dames

REFERENCES :

Difficulté de programmation : très difficile.

Catégorie de jeu : deux joueurs.

Difficulté du jeu : difficile.

LE JEU :

C'est le jeu classique de dames à la française, mais un peu incomplet puisqu'il ne prévoit pas la possibilité de faire des dames. La règle du jeu reste classique, mais les contrôles ont été un peu simplifiés. Les mouvements des pièces sont entrés en une seule manœuvre (position de départ et direction du déplacement) et les prises sont réalisées automatiquement, si elles sont possibles.

Par contre le changement de joueur étant réalisé à chaque coup, il faudra éventuellement taper 0 pour pouvoir rejouer s'il y avait eu saut ou si le coup n'était pas possible.

LE PROGRAMME :

Les lignes 60 à 250 dessinent le damier et définissent les positions initiales des pions noirs et des pions blancs dans une table R (X, Y).

L'impression des pions proprement dite est effectuée par le sous-programme 530-590.

La ligne 280 indique le joueur qui a le trait.

Le coup est entré aux lignes 290 à 410.

La ligne 320 réalise le changement de joueur (on tape 0).

La ligne 420 regarde si le déplacement est possible.

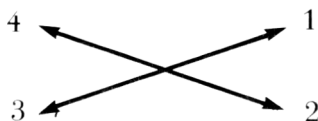
La ligne 420 traite le cas d'un déplacement et les lignes 440 à 480 celui d'un saut.

La ligne 490 assure le changement de joueur.

POUR JOUER :

Dès le lancement du programme par RUN, le damier est affiché avec ses pions, et le joueur ayant le trait est indiqué.

On joue en indiquant par un mot de 3 lettres (pouvant commencer par un 0) la colonne et la ligne de la pièce à déplacer, puis la direction du déplacement (codée de 1 à 4) suivant la convention :



Par exemple D91 signifie que le O situé en ligne 9, colonne D, se déplace dans la direction 1.

De même 0G3 signifiera pièce noire, ligne 0, colonne G, direction 3.

Les sauts multiples n'étant pas contrôlés, il faudra taper 0 pour rejouer. Le programme ne contrôle pas l'arrivée à dame, et n'interdit pas les déplacements en arrière.

EXTENSIONS POSSIBLES :

Il sera assez facile de commencer par améliorer les contrôles (changement de joueur, coup en arrière qui ne correspondrait pas à une prise, sortie de terrain). On pourra envisager la création de dames, leur déplacement, et éventuellement écrire un programme permettant à l'APPLE II de se mêler avec de bonnes chances de réussite à vos passionnantes parties.

```
10 REM DAMES
20 HOME
30 DIM R(10,10)
40 C$(0) = "BLANC":C$(1) = "NOIR"
   "
50 A1$ = "c/":A2$ = "/c"
60 PRINT "  A B C D E F G H I J
   ": PRINT "
   "
70 B$ = "  "
80 FOR I = 0 TO 9
90 FOR J = 1 TO 2
100 I$ = "  ": IF J = 2 THEN I$ =
    STR$(I) + " "
110 PRINT I$:"!";
120 FOR K = 1 TO 10
130 IF AG = 0 THEN AG = 1: PRINT
    B$:: GOTO 150
140 AG = 0: INVERSE : PRINT B$:: NORMAL
150 NEXT K
160 PRINT "!"
170 NEXT J
180 AG = NOT (AG)
190 NEXT I
200 VTAB 22: HTAB 1: PRINT "9 !"
   :: FOR I = 1 TO 5: INVERSE :
    PRINT B$:: NORMAL : PRINT "
   " :: NEXT I: PRINT "!"
210 A = 0: FOR B = 1 TO 4: FOR C =
    1 TO 5
220 A = A + 2 - 9 * (A = 9) - 11 *
    (A = 10)
230 I = B:J = A:R(I,J) = 1: GOSUB
    530
```

```

240 I = 11 - B:J = 11 - A:R(I,J) =
    2: GOSUB 530
250 NEXT C,B
260 H = 1
270 VTAB 4: HTAB 29: INVERSE : PRINT
    "JOUEUR "
280 VTAB 3 + H: HTAB 37: PRINT A
    1$: VTAB 4 - H: HTAB 37: PRINT
    A2$: NORMAL
290 VTAB 7: HTAB 25: PRINT "VOTR
    E COUP CLD": HTAB 31: PRINT
    "EX : H64"
300 VTAB 10: HTAB 27: PRINT "C :
    A-J": PRINT : HTAB 27: PRINT
    "L : 0-9": PRINT : HTAB 31: PRINT
    "4 1": HTAB 27: PRINT "D :
    *": HTAB 31: PRINT "3 2"
310 VTAB 20: HTAB 31: GET K$: PRINT
    K$: IF K$ = "/" THEN VTAB 2
    2: END
320 IF K$ = "0" THEN 490
330 IF K$ < "A" OR K$ > "J" THEN
    PRINT CHR$ (7);: GOTO 310
340 VTAB 20: HTAB 32: GET L$: PRINT
    L$: IF ASC (L$) = 8 THEN 31
    0
350 IF L$ < "0" OR L$ > "9" THEN
    PRINT CHR$ (7);: GOTO 340
360 I = VAL (L$) + 1:J = ASC (K
    $) - 64: IF R(I,J) < > H +
    1 THEN PRINT CHR$ (7);: GOTO
    310
370 VTAB 20: HTAB 33: GET K$: PRINT
    K$: IF ASC (K$) = 8 THEN 34
    0
380 IF K$ < "1" OR K$ > "4" THEN
    PRINT CHR$ (7);: GOTO 370
390 I1 = I + (K$ = "2" OR K$ = "3
    ") - (K$ = "4" OR K$ = "1")
400 J1 = J + (K$ = "2" OR K$ = "1
    ") - (K$ = "4" OR K$ = "3")
410 IF I1 > 10 OR J1 > 10 OR I1 <
    1 OR J1 < 1 THEN PRINT CHR$
    (7);: GOTO 310
420 IF R(I1,J1) = 0 THEN R(I,J) =
    0: GOSUB 600:I = I1:J = J1:R
    (I,J) = H + 1: GOSUB 530: GOTO
    490

```



```

430 IF R(I1,J1) = H + 1 THEN PRINT
    CHR$(7):: GOTO 310
440 I2 = I1 + (K$ = "2" OR K$ = "
    3") - (K$ = "4" OR K$ = "1")
450 J2 = J1 + (K$ = "2" OR K$ = "
    1") - (K$ = "4" OR K$ = "3")
460 IF I2 > 10 OR J2 > 10 OR I2 <
    1 OR J2 < 1 THEN PRINT CHR$(
    7):: GOTO 310
470 IF R(I2,J2) < > 0 THEN PRINT
    CHR$(7):: GOTO 310
480 R(I,J) = 0: GOSUB 600: I = I1:
    J = J1: R(I,J) = 0: GOSUB 600
    : I = I2: J = J2: R(I,J) = H +
    1: GOSUB 530
490 H = ABS (H - 1)
500 GOTO 270
510 GOTO 270
520 END
530 REM
540 INVERSE
550 K = (R(I,J) = 2)
560 X = 2 * I + 1: Y = 2 * J + 2
570 VTAB X + K: HTAB Y: PRINT A1
    $
580 VTAB X + 1 - K: HTAB Y: PRINT
    A2$
590 NORMAL : RETURN
600 X = 2 * I + 1: Y = 2 * J + 2
610 A$ = " ": INVERSE
620 VTAB X: HTAB Y: PRINT A$
630 IF I = 8 THEN A$ = " _ "
640 VTAB X + 1: HTAB Y: PRINT A$
650 NORMAL : RETURN

```


Niveau 5

<i>Nom du programme</i>	<i>Thèmes et particularités</i>
Un dé	Tirage d'un dé avec graphisme.
Losanges	Pour vous aider à vous endormir...
Mathieu	Un jeu de réflexion pour deux joueurs.
Ardoise magique	Mettez au point vos créations graphiques sans papier ni crayon.
Evasion	Il vous faudra beaucoup de chance pour réussir.
Smokey	Fumez sans risque, avec ce programme original pour un joueur.
Course de voitures	Un jeu de réflexe pas trop difficile, pour un joueur.
Tic Tac Toe	Un classique, mais pas sans intérêt (deux joueurs).
Quatre à quatre	Un jeu de réflexion un peu difficile, pour deux joueurs.
Taquin de lettres	Un proverbe est caché dedans : à vous de le découvrir.
Anagrammes	Un bon jeu de réflexe : pour deux joueurs.
Oeufs	Ne laissez pas faire la pondeuse diabolique (un joueur).
Cibles	Un peu d'adresse, beaucoup de chance : très visuel.
Gouttes	Un véritable gag visuel. A exploiter.
Obstacles	Beaucoup d'obstacles à éviter dans ce jeu d'adresse pour un joueur.
Captures dans l'espace	Votre mission serait facile, sans ces dam-nées météorites.
Anomalie	Exercez votre perspicacité dans la reconnaissance des symboles.
Plusieurs en un	Un terrain de jeu et les pions, pour deux joueurs.
Parcours du cavalier	Très spectaculaire, et instructif à plus d'un titre.
Mur de briques	Un grand classique dans une version très complète.
Othello	Pour deux joueurs ; il retourne les pions à votre place.

Les graphiques haute et basse résolution sur APPLE II

GRAPHIQUE BASSE RESOLUTION

L'APPLE II dispose d'un graphique de basse résolution, affichant dans une matrice de 40×40 , le $(0, 0)$ étant en haut à gauche. L'instruction **GR** permet de passer dans ce mode, l'instruction **TEXT** provoquant le retour à la normale. En **GR**, on dispose de quatre lignes de texte en bas de l'écran.

- **COLOR** permet le choix d'une couleur (16 couleurs de 0 à 15).
- **COLOR = 7** permet de choisir le blanc (code 7) comme couleur d'affichage des pavés suivants.
- **PLOT C, L** affiche un pavé situé à l'intersection des colonnes C et lignes L (C et L compris entre 0 et 39 inclus).
- **HLIN C1, C2 AT L** trace une ligne horizontale de la colonne C1 à la colonne C2, sur la ligne L.
- **VLIN L1, L2 AT C** trace une ligne verticale de la ligne L1 à la ligne L2, sur la colonne C.
- **SCRN (C, L)** est une fonction renvoyant la couleur du pavé situé à la colonne C et ligne L.

GRAPHIQUE HAUTE RESOLUTION

On passe dans ce mode par l'ordre **HGR**. On est alors en haute résolution (280 points par 160) et, comme en **GR**, on a toujours quatre lignes de texte en bas de l'écran.

- **HCOLOR** permet le choix d'une couleur (entre 0 et 7).
- **HPOINT CL** affiche un point en colonne C et ligne L. C doit être compris entre 0 et 279 et L entre 0 et 159. On peut également utiliser **HPOINT** pour tracer une ligne.

Exemple :

HPOINT 0, 0 TO 279, 159 tracera une diagonale de l'écran graphique.

REPERAGE DU CARACTERE AFFICHE A UN EMPLACEMENT DONNE DE L'ECRAN

L'affichage à l'écran est une image de la mémoire écran. Cette mémoire s'étend de 1024 à 2047. L'écran s'affiche par tiers. A partir de 1024, on trouve les quarante caractères de la première ligne, puis les quarante caractères de la 9^e ligne, puis les quarante caractères de la 17^e ligne. Ensuite huit autres caractères sont inutilisés pour s'aligner sur une puissance de 2 ($40 + 40 + 40 + 8 = 128 = 2^7$). Viennent ensuite la 2^e ligne, la 10^e, la 18^e, etc. La valeur de l'octet ainsi trouvé contient le code ASCII du caractère affiché à cet endroit ± 64 ou ± 128 , suivant qu'il est en normal, inverse ou flash (cf. table, manuel de référence APPLE II, page 15).

$A = \text{INT}((X - 1) / 8) : B = X - A * 8$

$C = 895 + 128 * B + 40 * A + Y$

vous donneront dans C l'adresse de l'octet correspondant au caractère ligne X et colonne Y.

D = PEEK (C) vous permettra de récupérer le code écran de ce caractère.

Un dé

REFERENCES :

Difficulté de programmation : facile.

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

Difficulté du jeu : aucune.

LE JEU :

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

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

LE PROGRAMME :

La ligne 80 utilise le générateur aléatoire de l'APPLE II pour tirer un nombre de 1 à 6.

Les lignes 50 et 60 dessinent l'entourage du dé.

Les lignes 100 à 130 sont utilisées pour le dessin des points.

Les lignes 140 et 150 permettront le cas échéant de refaire un tirage.

POUR JOUER :

Lancer le programme par RUN.

Un dé s'affiche, en couleurs, et en grand format.

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

EXTENSIONS POSSIBLES :

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

```

10  REM UN DE
20  HOME
30  GR
40  COLOR= 1
50  HLIN 14,22 AT 14: HLIN 14,22 AT
    22
60  VLIN 14,22 AT 14: VLIN 14,22 AT
    22
70  COLOR= 3
80  X = INT ( RND (1) * 6 + 1)
90  VTAB 21: HTAB 1: PRINT "TIRAG
    E DE : ";X
100 IF X > = 4 THEN PLOT 16,16
    : PLOT 16,20: PLOT 20,16: PLOT
    20,20
110 IF X > = 2 AND X < 4 THEN PLOT
    16,16: PLOT 20,20
120 IF X = 6 THEN PLOT 18,16: PLOT
    18,20
130 IF X / 2 < > INT (X / 2) THEN
    PLOT 18,18
140 VTAB 23: PRINT "AUTRE TIRAGE
    ? (O/N) : ";; GET K$: IF K$
    < > "N" THEN 30
150 TEXT : HOME

```

Losanges

REFERENCES :

Difficulté de programmation : facile.

Catégorie de jeu : interlude.

Difficulté du jeu : aucune.

LE JEU :

Le programme va progressivement construire sur l'écran un losange, 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 de chaque losange.

LE PROGRAMME :

Une première boucle K (50-120) définit les dimensions des tracés successifs.

La deuxième boucle I (60-110) trace des bandes de points, en utilisant des instructions HPLLOT (lignes 70-100).

La variable C contient la couleur (déterminée à la ligne 40). C est incrémentée (jusqu'à 7 inclus) à la ligne 130.

POUR JOUER :

On commence l'exécution du programme avec RUN et on regarde.

On peut arrêter à tout moment en tapant CNT et C.

Ne pas oublier de taper TEXT pour revenir en mode normal.

EXTENSIONS POSSIBLES :

On pourra introduire le son, imaginer sur des principes analogues divers programmes (kaléidoscope, tapisserie...).


```

10  REM LOSANGE
20  HOME : VTAB 21: HGR
30  C = 1
40  HCOLOR= C
50  FOR K = 0 TO 20
60  FOR I = 0 TO K
70  HPLOT 130 - K + I,80 - I
80  HPLOT 130 - K + I,80 + I
90  HPLOT 130 + K - I,80 + I
100 HPLOT 130 + K - I,80 - I
110 NEXT I
120 NEXT K
130 C = C + 1: IF C = 8 THEN C =
    0
140 GOTO 40

```

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 l'instruction PLOT pour obtenir sur l'écran des dessins aléatoires (de type fleur), plus ou moins élégants.

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

LE PROGRAMME :

Les 3 premiers paramètres Z, K et T, introduits en ligne 30, définissent le rayon maximum et l'occupation de l'écran.

Les variables aléatoires A et B, choisies entre 0 et 3, de la ligne 60, définissent l'allure de la courbe ; il pourra être intéressant de les introduire manuellement (par exemple : 60 INPUT A, B).

La boucle H (70-120) réalise le tracé, point par point. Le rayon R est calculé en ligne 80, et le passage en coordonnées cartésiennes est effectué aux lignes 90 et 100.

L'affichage des points est réalisé en ligne 110.

La ligne 130 imprime un message de fin.

La ligne 140 permet de refaire un nouveau dessin.

POUR JOUER :

On lance le programme en tapant RUN.

Le dessin sera ensuite exécuté automatiquement.

Le message "AUTRE DESSIN ?" signale la fin du tracé.

Si on veut un nouveau tracé, on tape O.

Pour arrêter, faire N.

EXTENSIONS POSSIBLES :

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

```

10  REM MATHIEU
20  HOME : VTAB 21: HGR
30  Z = 7:K = 9:T = 3
40  HCOLOR= 7
50  HPLOT 130,80
60  A = 3 * RND (1):B = 3 * RND
    (1)
70  FOR H = 0 TO 6.28 STEP 3.14 /
    180
80  R = Z * SIN (H * T)
90  X = 130 + INT (K * R * COS (
    A * H))
100 Y = 80 + INT (K * R * SIN (
    B * H))
110 HPLOT TO X,Y
120 NEXT H
130 VTAB 21: INVERSE : PRINT "AU
    TRE DESSIN ? (O/N) : ";; NORMAL
    : GET K$: PRINT
140 IF K$ < > "N" THEN 20
150 VTAB 21: CALL - 868: INVERSE
    : PRINT "FAITES 'TEXT' POUR
    REVENIR A LA NORMALE": NORMAL

```

Ardoise magique

REFERENCES :

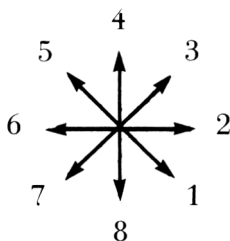
Difficulté de programmation : facile.

Catégorie de jeu : dessin.

Difficulté du jeu : facile.

LE JEU :

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



On dispose aussi de la possibilité de déplacer le point sur l'écran sans laisser de trace en appuyant sur la touche 0 (ce qui permet l'effacement partiel) et de laisser de nouveau une trace en réappuyant sur 0.

On peut choisir la couleur du point en appuyant sur une touche de A à H ; et il est possible d'effacer tout l'écran en appuyant sur P.

LE PROGRAMME :

La ligne 40 contient le choix des possibilités de couleur.

La ligne 60 imprime le point élémentaire.

Les lignes 70 à 120 contrôlent les entrées.

La ligne 150 appelle l'un des quatre sous-programmes effectuant le déplacement.

La ligne 160 vérifie que l'on ne sort pas des limites de l'écran.

La ligne 180 effectue l'effacement s'il y a lieu.

La ligne 80 choisit la couleur ; la ligne 90 positionne ou non l'effacement.

POUR JOUER :

On lance le programme par RUN ; un point apparaît vers le milieu de l'écran. On devra taper 0 pour le déplacer sans afficher.

Si on frappe une lettre de A à H, la couleur du point change.

La touche P efface tout, et l'on recommence.

On peut sortir à tout moment, en tapant Z.

EXTENSIONS POSSIBLES :

On pourra, bien sûr, envisager d'écrire un programme identique en utilisant des tracés de segments, prévoir un sous-programme qui permette la sauvegarde du dessin réalisé pour une utilisation ultérieure, etc.

```
10 REM ARDOISE MAGIQUE V.1
20 HOME : VTAB 21: HGR
30 PRINT "543": PRINT "6+2": PRINT
   "781"
40 C$ = "01234567"
50 X = 140:Y = 80:C = 3
60 HCOLOR= C: HPLLOT X,Y
70 VTAB 22: HTAB 12: CALL - 868
   : PRINT "X = ";X;"   Y = ";Y
   : "   "   :: GET K$
80 IF K$ > = "A" AND K$ < = "H
   " THEN C = VAL ( MID$ (C$, ASC
   (K$) - 64,1)): GOTO 60
90 IF K$ = "0" THEN S = ABS (S -
   1): GOTO 70
100 IF K$ = "P" THEN 20
110 IF K$ = "Z" THEN END
120 IF K$ < "1" OR K$ > "8" THEN
   PRINT CHR$ (7):: GOTO 70
130 F = 2 * ( VAL (K$) < 5) - 1:A
   = X:B = Y
140 U = VAL (K$) + 2 * (F - 1)
150 ON U GOSUB 200,210,230,240
160 D = (X > = 0 AND X < = 279 AND
   Y > = 0 AND Y < = 159)
170 IF D = 0 THEN GOSUB 260
180 IF S < > 0 THEN HCOLOR= 0:
   HPLLOT A,B
190 GOTO 60
200 Y = Y + F
210 X = X + F
220 RETURN
230 GOSUB 210
240 Y = Y - F
250 RETURN
260 X = A:Y = B
270 PRINT CHR$ (7);
280 RETURN
```

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 l'APPLE II vous ait auparavant touché par l'un des nombreux obus explosifs qu'il va tirer sur vous.

Vous pouvez vous déplacer dans 4 directions et vous devez réussir à sortir du terrain par le bord droit, sans, bien sûr, avoir été touché une seule fois.

LE PROGRAMME :

La ligne 50 définit la position de départ du joueur.

La ligne 60 choisit la coordonnée horizontale de la case où l'APPLE II va tirer (liée à votre position), et la ligne 70 choisit la coordonnée verticale (arbitraire) de cette case.

La boucle I (90-110) affiche les impacts d'obus.

Le déplacement du joueur est contrôlé par les lignes 150 et 160 ; sa trajectoire est tracée par la ligne 120.

La ligne 210 regarde s'il est touché ou non.

La ligne 220 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.

Les 4 touches I, J, K et M 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 CNT et C.

EXTENSIONS POSSIBLES :

La couleur et un graphisme plus soigné amélioreront considérablement l'attrait du jeu. On pourra aussi rendre le tir des bombes plus aléatoire, introduire des obstacles à contourner, ajouter des contrôles, faire participer plusieurs joueurs, etc.

```

10  REM EVASION
20  HOME : VTAB 21
30  HGR
40  HCOLOR= 7
50  X = 0:Y = 12
60  H = 7 * (X - 1) + RND (1) * 1
    6
70  K = 160 * RND (1) + 5
80  IF X < 3 THEN 120
90  FOR J = 1 TO RND (1) * 7 + 3

100  HPLOT H + 6 * RND (1) - 2,K
    + 6 * RND (1) - 2
110  NEXT J
120  HPLOT X * 7,Y * 8
130  T = PEEK ( - 16384): IF T <
    128 THEN 60
140  POKE - 16368,0: IF T < 201 OR
    T > 205 OR T = 204 THEN 60
150  X = X + (T = 203) - (T = 202)

160  Y = Y + (T = 205) - (T = 201)

170  X1 = X * 7:Y1 = Y * 8
180  FOR I = - 1 TO 1: FOR J = -
    1 TO 1
190  HPLOT X1 + I,Y1 + J
200  NEXT J,I
210  IF X > 3 AND ( ABS (H - X1) +
    ABS (K - Y1) < = 10) THEN
    X$ = "PERDU A " + STR$ (X) +
    "/38 ...": GOTO 240
220  IF X < 38 THEN 60
230  X$ = "BRAVO"
240  VTAB 21: FOR I = 1 TO 5: PRINT
    CHR$ (7):: NEXT : INVERSE :
    PRINT "==== ":X$: NORMAL
250  POKE - 16368,0

```

Smokey

REFERENCES :

Difficulté de programmation : moyenne.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

Une colonne de fumée grise apparaît sur l'écran. Le joueur, représenté par un dièse (#), 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 60 fait apparaître la fumée au bas de l'écran ; la ligne 70 permet de changer (vers la droite ou vers la gauche) la direction de la colonne, et la ligne 90 fait monter la colonne.

Le joueur apparaît grâce à la ligne 110, et son déplacement est effectué par les lignes 170 à 190.

Les lignes 250 et 260 se chargent de l'affichage du résultat.

POUR JOUER :

On doit lancer l'exécution du programme par RUN.

La fumée commence à s'élever, tandis que le joueur, apparu sous forme d'un dièse en haut de l'écran, va se déplacer sur elle à l'aide des touches I, J, K, M. Dès qu'il arrive sur la fumée, il va automatiquement commencer à l'avalier, et devra essayer de suivre la colonne sans dévier. Il gagne s'il arrive en bas de l'écran avant que la fumée n'arrive en haut.

L'affichage final indique alors BRAVO ou PERDU, ainsi que le score obtenu.

EXTENSIONS POSSIBLES :

On pourra, tout en conservant l'idée de départ, imposer d'autres règles : par exemple avaler toute la fumée, qui pourra aussi repartir d'un autre endroit si l'on dévie de la colonne initiale.

On pourra aussi améliorer les contrôles de fin de partie, et la qualité du graphisme.


```

10  REM SMOKEY
20  A$ = "#":B$ = "+"
30  S = 0:X = 23:Y = INT ( RND (1
    ) * 24)
40  U = 1:V = 15
50  HOME
60  VTAB X: HTAB Y: PRINT B$
70  Y = Y + ( RND (1) < .5 AND Y <
    24) - ( RND (1) < .5 AND Y >
    1)
80  T = 1
90  X = X - T
100 IF X = 0 THEN 250
110 VTAB U: HTAB V: PRINT A$
120 IF T = 0 THEN 160
130 FOR I = 1 TO 400
140 K = PEEK ( - 16384): IF K <
    128 THEN NEXT I
150 POKE - 16368,0
160 VTAB U: HTAB V: PRINT " "
170 U = U + (K = 205 AND T = 1 AND
    X < 24)
180 U = U + (T = 0)
190 V = V + (K = 203 AND V < 24) -
    (K = 202 AND V > 1)
200 A = INT ((U - 1) / 8):B = U -
    A * 8:K = 895 + 128 * B + 40
    * A + V:C = PEEK (K)
210 IF C < > 171 THEN 60
220 T = 0:S = S + 1
230 PRINT CHR$ (7);
240 IF U < 23 THEN 110
250 IF T = 0 THEN VTAB 20: CALL
    - 958: INVERSE : PRINT "===
    BRAVO ... ": GOTO 270
260 VTAB 20: CALL - 958: INVERSE
    : PRINT "=== PERDU ... "
270 HTAB 16: PRINT "SCORE : ";S;
    ". ": NORMAL

```

Course de voitures

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de réflexe (1 joueur).

Difficulté du jeu : moyenne.

LE JEU :

Vous circulez sur une route très sinueuse, ne comportant aucun obstacle, et vous devez essayer de la suivre le plus longtemps possible. La trace de votre parcours reste affichée en permanence, et votre score est affiché dès que vous avez un accident.

LE PROGRAMME :

L'impression de la route est assurée par la ligne 90.

La ligne 110 fait avancer la route, et la ligne 170 imprime la voiture.

La ligne 200 contrôle son mouvement, et la ligne 210 détermine si la route fait ou non des virages.

La ligne 150 regarde si l'on a touché le bord de la route.

La ligne 160 ne compte que 1/2 point si l'on s'approche du bord.

La ligne 220 incrémente le score, et la ligne 230 efface l'ancienne position de la voiture (une trace claire apparaît à l'emplacement de sa trajectoire).

La ligne 240 affiche le temps et le score.

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 +, apparaît.

Pour le déplacer vers la droite ou vers la gauche, vous devrez utiliser les touches → et ← (flèches de déplacement du curseur). Il vous faudra éviter le plus longtemps possible de toucher une bordure noire, ce qui provoquera l'arrêt du programme et l'affichage de la mention ACCIDENT.

On pourra arrêter le déroulement de la partie à tout moment en pressant les touches CNT et C.

EXTENSIONS POSSIBLES :

On peut envisager de jouer à plusieurs, chacun son tour, en enregistrant le meilleur score.

On peut aussi envisager d'élargir la piste, et de faire courir plusieurs voitures.

On pourra aussi s'inspirer des jeux d'arcades qui sont particulièrement nombreux sur ce thème.

```

10 REM COURSE DE VOITURES
20 POKE 34,0: HOME : POKE 34,3
30 VTAB 1: HTAB 1: INVERSE : PRINT
   " SCORE :          TEMPS : ": NORMAL

40 A$ = "+"
50 B$ = "((      ))"
60 T = 0: HOME
70 X = 24:Y = 11:A = 10:B = 15:S =
   0
80 J = INT ((A - 1) / 8):M = A -
   J * 8:AD = 895 + 128 * M + 4
   0 * J
90 VTAB X: HTAB Y: INVERSE : PRINT
   B$:
100 NORMAL
110 PRINT
120 W = PEEK ( - 16336) + PEEK
   ( - 16336)
130 Z = Z + 1: IF Z < (X - A) THEN
   S = S - 1: GOTO 170
140 J = INT ((A - 1) / 8):M = A -
   J * 8:K = 895 + 128 * M + 40
   * J + B:L = PEEK (K)
150 IF L < > 32 AND L < > 40 AND
   L < > 41 THEN 260
160 IF L < > 32 THEN W = PEEK
   ( - 16336) - PEEK ( - 16336
   ):S = S - .5
170 VTAB A: HTAB B: PRINT A$
180 T = T + 1:K = PEEK ( - 16384
   )
190 POKE - 16368,0:C = B
200 B = B + (K = 149) - (K = 136)

210 Y = Y + ( INT (2 * RND (1)) AND
   Y < 25) - ( INT (2 * RND (1
   )) AND Y > 3)
220 S = S + 1
230 POKE 1191 + C,160
240 VTAB 1: HTAB 10: INVERSE : PRINT
   INT (S):: HTAB 23: PRINT T:
   " ": NORMAL
250 GOTO 90
260 FLASH : VTAB A: HTAB B: PRINT
   A$: NORMAL
270 VTAB 24: PRINT : FLASH : PRINT
   CHR$ (7) + CHR$ (7) + "===
   ACCIDENT ... " + CHR$ (7)
   + CHR$ (7): NORMAL

```

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 des lignes 40 à 100.

La ligne 120 s'occupe d'afficher le symbole cœur ou carreau correspondant au joueur qui a le trait.

Un symbole clignotant est affiché en haut et à gauche du terrain, même si la case est déjà occupée ; les lignes 190 à 240 permettent de déplacer le symbole sur le terrain de jeu.

La ligne 250 vérifie si la case où l'on veut jouer est libre, et la ligne 180 si l'on désire jouer.

La ligne 260 affiche le coup, et la ligne 270 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 touches I, J, K, M.

Lorsqu'il arrive sur la case où il désire jouer, il suffit qu'il appuie sur la touche Ø, le coup sera refusé si la case n'est pas libre.

Si la case est libre, le symbole qui correspond à ce joueur est inscrit, et on change de joueur.

Lorsque la partie est finie, il convient de taper CNT et C 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 REM TIC TAC TOE
20 HOME
30 C$(0) = "O":C$(1) = "X":C(0) =
   79:C(1) = 88
40 VTAB 1: PRINT "
   ": VTAB 14: PRINT "
   "
50 FOR I = 2 TO 14: VTAB I: HTAB
   1: PRINT "!": HTAB 15: PRINT
   "!": NEXT I
60 FOR I = 1 TO 3
70 FOR J = 1 TO 3
80 VTAB 4 * J: HTAB 4 * I: INVERSE
   : PRINT " "
90 NEXT J
100 NEXT I: NORMAL
110 K = 0:T = 0
120 VTAB 18: HTAB 1: PRINT "===
   JOUEUR ":C$(K)
130 I = 1:J = 1
140 U = 4 * I:V = 4 * J
150 A = INT ((U - 1) / 8):B = U -
   A * 8:C = 895 + 128 * B + 40
   * A + V:D = PEEK (C)
160 POKE C,D + 64
170 VTAB 18: HTAB 15: GET K$
180 IF K$ = "O" THEN 250
190 I1 = I + (K$ = "M" AND I < 3)
   - (K$ = "I" AND I > 1)
200 J1 = J + (K$ = "K" AND J < 3)
   - (K$ = "J" AND J > 1)
210 IF I1 = I AND J1 = J THEN PRINT
   CHR$ (7):: GOTO 170
220 POKE C,D
230 I = I1:J = J1
240 GOTO 140
250 IF D < > 32 THEN PRINT CHR$
   (7):: GOTO 170
260 POKE C,C(K) - 64
270 K = ABS (K - 1)
280 T = T + 1: IF T < 9 THEN 120
290 VTAB 18: HTAB 1: CALL - 868
   : INVERSE : PRINT " PARTIE T
   ERMINEE. ": NORMAL

```

Quatre à quatre

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion (un ou deux joueurs).

Difficulté du jeu : difficile.

LE JEU :

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

LE PROGRAMME :

La variable alphanumérique A\$, en ligne 40, contient la table des coups possibles (déplacement du cavalier).

Les lignes 50 à 160 dessinent le terrain de jeu, et les positions de départ.

La ligne 180 permet d'afficher O ou X, selon le joueur qui a le trait.

La ligne 200 permet l'affichage du curseur clignotant qui indique sur quelle case on se trouve.

Les lignes 330 à 350 permettent le déplacement du curseur.

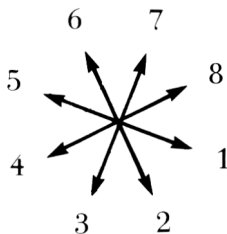
La ligne 230 regarde si l'on a appuyé sur une touche de 1 à 8, donnant la direction du déplacement.

Les lignes 240 à 320 contrôlent et impriment le déplacement.

POUR JOUER :

On lance le programme par RUN. Le terrain de jeu est affiché, ainsi que le symbole permettant d'identifier le joueur qui a le trait, les 8 directions de déplacement du cavalier et les 4 directions de déplacement du curseur. Le joueur peut choisir le pion qu'il désire déplacer en manœuvrant le curseur clignotant grâce aux touches de déplacement du curseur (I, J, K, M).

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



Le coup sera refusé si la case de départ n'est pas conforme ou si la case d'arrivée est occupée. Sinon, le coup est réalisé, et le trait passe à l'autre joueur, qui déplace le curseur clignotant pour l'amener sur l'un des pions, et joue de la même façon.

Le joueur gagnant est celui qui réussit à amener le premier tous ses pions dans les cases initialement occupées par l'adversaire. On peut interrompre la partie à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :

On peut améliorer les contrôles, prévoir un test de fin, utiliser un terrain de jeu différent ; on pourra aussi envisager l'écriture d'un programme permettant à l'APPLE II de participer au jeu avec de bonnes chances de l'emporter.

```

10 REM QUATRE A QUATRE
20 C$(0) = "X":C$(1) = "O"
30 HOME
40 H = 1:A$ = "5665322356"
50 FOR I = 0 TO 5
60 IF I > 0 THEN NORMAL : VTAB
   I * 4 + 2: HTAB 2: PRINT I::
   VTAB 2: HTAB I * 4 + 2: PRINT
   I:
70 FOR J = 0 TO 5
80 FOR K = 1 TO 3: VTAB I * 4 +
   K: HTAB (J + 1) * 4: INVERSE
   : PRINT " ": NEXT K
90 INVERSE : VTAB (I + 1) * 4: HTAB
   J * 4 + 1: PRINT "   ":
100 NEXT J,I
110 NORMAL
120 FOR I = 1 TO 2: FOR J = 1 TO
   3 STEP 2
130 VTAB I * 4 + 2: HTAB J * 4 +
   2: PRINT "O"
140 VTAB 26 - I * 4: HTAB 26 - J
   * 4: PRINT "X"
150 NEXT J,I
160 GOSUB 400
170 X = 5 - 4 * H:Y = 5 - 4 * H
180 VTAB 18: INVERSE : HTAB 27: PRINT
   "Joueur ";C$(H):" ";
190 GOSUB 360
200 POKE C,D - 128 * (D > = 128
   ) + 64 * (D = 160)
210 X1 = X:Y1 = Y
220 VTAB 18: HTAB 36: GET K$: PRINT
   K$:K = VAL (K$)
230 IF K$ < "1" OR K$ > "8" THEN
   330

```

```

240 IF D = 128 * (D > 128) < >
    ASC (C$(H)) THEN PRINT CHR$
    (7):: GOTO 220
250 X = X + VAL ( MID$ (A$,K,1))
    - 4
260 Y = Y + VAL ( MID$ (A$,K + 2
    ,1)) - 4
270 C1 = C:D1 = D: GOSUB 360: IF
    D > < 160 THEN 290
280 IF X > 0 AND Y > 0 AND X < 6
    AND Y < 6 THEN 300
290 X = X1:Y = Y1: GOSUB 360: PRINT
    CHR$ (7):: GOTO 220
300 POKE C1,160: POKE C, ASC (C$
    (H)) + 128
310 H = ABS (H - 1)
320 GOTO 170
330 X = X + (K$ = "M" AND X < 5) -
    (K$ = "I" AND X > 1)
340 Y = Y + (K$ = "K" AND Y < 5) -
    (K$ = "J" AND Y > 1)
350 POKE C,D + 128 * (D < 96): GOTO
    190
360 U = X * 4 + 2:V = Y * 4 + 2:A
    = INT ((U - 1) / 8):B = U -
    A * 8:C = 895 + 128 * B + 40
    * A + V:D = PEEK (C)
370 RETURN
380 VTAB 1: HTAB 35: GET K$
390 END
400 DATA .6,,7,,5,,,,8,,,*,.,4,,
    ,,1,,3,,2,
410 FOR A = 1 TO 25: READ K$
420 NORMAL
430 IF X = 0 THEN INVERSE
440 X = ABS (X - 1)
450 J = J + 1: IF J > 5 THEN J =
    1:I = I + 1
460 IF LEN (K$) = 0 THEN K$ = "
    "
470 VTAB 1: HTAB 30 + J: PRINT K
    $
480 NEXT A
490 VTAB 11: HTAB 31: PRINT " I
    ": HTAB 31: PRINT " ! ":
    HTAB 31: PRINT "J-*~K": HTAB
    31: PRINT " ! ": HTAB 31: PRINT
    " M "
500 RETURN

```


Taquin de lettres

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

Qui d'entre nous n'a pas, un jour ou l'autre, eu dans les mains un de ces petits puzzles où l'on essaie de déplacer d'une case à une autre des petits carrés de plastique colorés, jusqu'à l'obtention d'un résultat bien précis ? En son temps le 14-15 de SAM LOYD, a eu autant de succès que notre célèbre RUBIK'S.

Le taquin qui vous est proposé ici utilise 35 lettres, disposées au hasard dans un carré de 6 cases sur 6 cases, et ces lettres peuvent, replacées dans leur ordre initial, vous permettre de reconstituer un proverbe très connu. Une case blanche en flash vous permet de déplacer les lettres, et un compteur affiche en permanence le nombre d'essais : arriverez-vous à retrouver le proverbe ?

LE PROGRAMME :

La phrase à reconstituer (qui pourra bien sûr être modifiée) est placée dans une variable R\$ en ligne 50.

L'impression en est réalisée par la boucle K (70-80).

Le mélange des lettres est fait par la boucle I (90-170).

La ligne 210 réalise l'affichage du carré coloré ; celui-ci est déplacé par les lignes 250 et 260 en utilisant les touches I, J, K, M.

La ligne 280 réimprime la lettre concernée à l'emplacement du carré coloré.

La ligne 290 comptabilise le nombre d'essais qui est imprimé grâce à la ligne 300.

POUR JOUER :

On lance le programme par RUN.

La partie est facilitée par l'affichage du proverbe à reconstituer. On peut déplacer les lettres et les faire passer à l'emplacement du carré coloré, à l'aide des 4 touches I, J, K, M.

Le nombre d'essais est affiché en permanence, à droite du puzzle.

On peut arrêter à tout moment en tapant le slash (/).

EXTENSIONS POSSIBLES :

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

```

10 REM TAQUIN DE LETTRES
20 HOME
30 INVERSE : FOR I = 0 TO 14: FOR
    J = 0 TO 7: VTAB 2: HTAB I +
    2: PRINT " ": VTAB I + 2: HTAB
    2: PRINT " "
40 VTAB 16: HTAB I + 2: PRINT "
    ": VTAB I + 2: HTAB 16: PRINT
    " ": NEXT I: NORMAL
50 A$ = "":R$ = "ONASOUVENTBESOIN
    D'UNPLUSPETITQUESOI"
60 VTAB 12: HTAB 25: INVERSE : PRINT
    "      ": VTAB 19: HTAB 25
    : PRINT "      ": NORMAL
70 I = 0:J = 6: FOR K = 1 TO 35:J
    = J + 1: IF J > 6 THEN J =
    1:I = I + 1: VTAB 12 + I: HTAB
    25: INVERSE : PRINT " ";; HTAB
    32: PRINT " ": NORMAL
80 VTAB 12 + I: HTAB 25 + J: PRINT
    MID$ (R$,K,1): NEXT K
90 FOR I = 1 TO 35
100 X = INT ( RND (1) * 35 + 1)
110 IF MID$ (R$,X,1) = "-" THEN
    100
120 A$ = A$ + MID$ (R$,X,1)
130 R$ = MID$ ( LEFT$ (" " + R$,
    X) + "-" + MID$ (R$,X + 1),
    2)
140 F = ASC ( MID$ (A$,I,1))
150 Y = INT (I / 6):X = I - 6 *
    Y
160 VTAB Y * 2 + 4: HTAB X * 2 +
    4: PRINT CHR$ (F)
170 NEXT I
180 X = 0:Y = 0:S = 0
190 VTAB 3: HTAB 20: PRINT "NB.E
    SSAIS : ";S
200 VTAB 6: HTAB 20: PRINT "DIRE
    CTION IJKM : "
210 GOSUB 340: POKE C,96
220 VTAB 6: HTAB 38: GET K$
230 IF K$ = "/" THEN 320
240 I = X:J = Y

```

```

250 X = X + (K$ = "K" AND X < 5) -
    (K$ = "J" AND X > 0)
260 Y = Y + (K$ = "M" AND Y < 5) -
    (K$ = "I" AND Y > 0)
270 IF X = I AND Y = J THEN PRINT
    CHR$(7);: GOTO 220
280 C1 = C: GOSUB 340: POKE C1, PEEK
    (C): POKE C, 96
290 S = S + 1
300 VTAB 3: HTAB 32: PRINT S
310 GOTO 220
320 VTAB 21
330 END
340 U = X * 2 + 4: V = Y * 2 + 4
350 A = INT ((V - 1) / 8): B = V -
    A * 8: C = 895 + 128 * B + 40
    * A + U
360 RETURN

```

Anagrammes

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : jeu de réflexion (2 joueurs).

Difficulté du jeu : selon le niveau.

LE JEU :

Des lettres sont choisies au hasard par l'APPLE II, le nombre de lettres étant fonction du niveau de difficulté désiré. Puis les lettres du mot ainsi obtenu sont brouillées, et affichées au bas de l'écran ; un nouveau mot est alors confectionné, de façon aléatoire, à partir des mêmes lettres et il est affiché sous le premier, et ainsi de suite ; les deux joueurs (gauche et droit) disposent chacun d'une touche (1 ou 0) et doivent la presser dès qu'ils voient s'afficher au bas de l'écran un mot déjà présent sur l'écran ; leur score est alors évalué, et un message correspondant à leur résultat sera délivré.

LE PROGRAMME :

Le niveau du jeu est entré à la ligne 30. La boucle I (60-80) réalise le choix des lettres.

Une deuxième boucle I (100-160) effectue le brouillage, et la ligne 120 permet de détecter la pression d'une touche.

La ligne 170 affiche un nouveau mot, et fait monter d'une ligne sur l'écran les mots déjà obtenus.

La ligne 210 cherche quel est celui des deux joueurs qui a pressé une touche.

La boucle I (230-310) explore les mots présents sur l'écran ligne par ligne, alors que la boucle J (260-290) saisit un par un les caractères de ces mots. On obtient ainsi un nouveau mot qui est comparé au mot initial en ligne 300 et le résultat BRAVO apparaît si c'est exact.

Sinon, la partie s'arrête sur l'affichage PERDU (ligne 320).

POUR JOUER :

On lance le programme par RUN.

On devra ensuite entrer le niveau de difficulté (en principe de 1 à 3) et faire RETURN. Les mots s'affichent en partant du bas de l'écran, et montent au fur et à mesure ; si l'un des joueurs croit reconnaître dans le mot qui apparaît un des mots déjà affichés, il devra immédiatement presser sa touche (1 ou 0), et son résultat sera affiché après un court instant.

La partie sera alors terminée.

EXTENSIONS POSSIBLES :

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

```

10  REM ANAGRAMMES
20  HOME
30  INPUT "NIVEAU (DE 1 A 3) : ";
    N
40  IF N < 1 OR N > 3 THEN 20
50  S$ = ""
60  FOR I = 1 TO N + 3
70  S$ = S$ + CHR$ ( INT ( RND (1
    ) * 26 + 65))
80  NEXT I
90  A$ = "":B$ = S$
100 FOR I = 1 TO LEN (S$)
110 N = INT ( RND (1) * LEN (B$
    ) + 1)
120 K = PEEK ( - 16384): IF K >
    127 THEN POKE - 16368,0
130 IF K = 176 OR K = 177 THEN 2
    10
140 A$ = A$ + MID$ (B$,N,1)
150 B$ = MID$ ( LEFT$ (" " + B$,
    N) + MID$ (B$,N + 1),2)
160 NEXT I
170 VTAB 24: PRINT A$
180 FOR I = 1 TO 30 * LEN (A$):
    NEXT I
190 S$ = A$
200 GOTO 90
210 J$ = "JOUER DE GAUCHE": IF K
    = 176 THEN J$ = "JOUER DE
    DROITE"
220 VTAB 23: INVERSE : PRINT S$:
    NORMAL
230 FOR I = 1 TO 22
240 B$ = ""
250 A = INT ((I - 1) / 8):B = I -
    A * 8:C = 895 + 128 * B + 40
    * A
260 FOR J = 1 TO LEN (S$)
270 R$ = CHR$ ( PEEK (C + J) - 1
    28)
280 B$ = B$ + R$
290 NEXT J
300 IF B$ = S$ THEN VTAB I: HTAB
    LEN (S$) + 4: PRINT "<---- B
    RAVO ";J$: VTAB 23: END
310 NEXT I
320 VTAB 23: HTAB LEN (S$) + 4:
    PRINT "=== PERDU ";J$

```

Œufs

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

Un oiseau mystérieux se déplace en battant des ailes au hasard sur l'écran.

Il pond des œufs (représentés par des O) que vous (un carré blanc) devez essayer de récupérer. Votre score (nombre d'œufs récupérés) sera affiché en permanence, et la partie se déroule en un laps de temps défini.

LE PROGRAMME :

L'oiseau est dessiné par le sous-programme en 340 ; son déplacement aléatoire est géré par le sous-programme 220-330 (en particulier les lignes 270 à 320).

La ligne 140 permet de reboucler le programme sur la ligne 70.

Le déplacement du joueur est régi par les lignes 160 et 170, et le test permettant de détecter la présence d'un œuf est effectué en ligne 190, après appel au sous-programme en 350-360.

La ligne 190 permet aussi l'affichage du score.

La fin de jeu est détectée en ligne 250.

POUR JOUER :

On lancera le programme par RUN.

L'oiseau commence alors à se déplacer au hasard sur l'écran, et se met à pondre ses œufs. Lorsque le symbole qui représente le joueur (un carré blanc) apparaît sur l'écran, celui-ci peut se déplacer en utilisant les 4 touches de déplacement (I, J, K, M). Il vous faut dès lors essayer de récupérer un maximum d'œufs avant la fin de la partie.

On pourra arrêter de jouer à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :

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

```

10 REM OEUF5
20 A$(0) = "c="/":A$(1) = "/=c"
30 HOME
40 VTAB 23: PRINT "TEMPS 200 ...
   SCORE 0"
50 P = 200:R = 18:S = 10:T = 0
60 X = INT ( RND (1) * 36 + 1):Y
   = INT ( RND (1) * 21 + 1)
70 GOSUB 340
80 FOR Q = 1 TO 5:K = PEEK ( -
   16384): IF K > 128 THEN POKE
   - 16368,0: GOSUB 150
90 NEXT Q
100 GOSUB 220
110 GOSUB 340
120 FOR Q = 1 TO 35: NEXT Q
130 GOSUB 220
140 GOTO 70
150 IF G < > 207 THEN VTAB S: HTAB
   R: PRINT " "
160 S = S + (K = 205 AND S < 21) -
   (K = 201 AND S > 1)
170 R = R + (K = 203 AND R < 36) -
   (K = 202 AND R > 1)
180 GOSUB 350:G = PEEK (C)
190 IF G = 207 THEN T = T + 1: POKE
   C,15: VTAB 23: HTAB 21: PRINT
   T: RETURN
200 POKE C,32
210 RETURN
220 A = X:B = Y
230 P = P - 1
240 VTAB 23: HTAB 7: PRINT P;" "

250 IF P = 0 THEN VTAB 24: HTAB
   1: INVERSE : PRINT CHR$ (7)
   + "=== " + CHR$ (7) + "C'E
   ST " + CHR$ (7) + "FINI ...
   " + CHR$ (7): NORMAL : END

260 VTAB B: HTAB A: PRINT " 0 "
270 X = ABS (X + INT (1 - 2 * (
   RND (1) < .5)))
280 Y = ABS (Y + INT (1 - 2 * (
   RND (1) < .5)))
290 IF X > 35 THEN X = 1
300 IF X < 1 THEN X = 36
310 IF Y > 21 THEN Y = 1
320 IF Y < 1 THEN Y = 21
330 RETURN
340 Z = ABS (1 - Z): VTAB Y: HTAB
   X: PRINT A$(Z): RETURN
350 A = INT ((S - 1) / 8):B = S -
   A * 8:C = 895 + 128 * B + 40
   * A + R
360 RETURN

```

Cibles

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : hasard.

Difficulté du jeu : facile.

LE JEU :

Il consiste simplement à tirer sur des cibles, en essayant de marquer le maximum de points ; le nombre de cibles (de 1 à 5) sera choisi en début de partie, et le score réalisé sera affiché en permanence. Si l'on touche le bord d'une cible, la partie s'arrête. On marquera d'autant plus de points que le nombre sur lequel on tombe sera plus grand.

LE PROGRAMME :

Le nombre de cibles est entré en ligne 30. La boucle I (60-80) effectue l'impression des cibles.

La ligne 100 affichera le score en permanence.

La balle est affichée par la ligne 110.

La boucle I (130-170) réalise l'impression des chiffres contenus dans les cibles, et permet le déplacement de la boule (ligne 180).

La ligne 200 regarde si l'on a tiré, et la ligne 210 regarde si la boule est encore sur la ligne de départ.

La ligne 220 fait progresser la boule d'une case vers la droite, et la ligne 230 vérifie si l'on a ou non atteint les cibles ; puis les lignes 240 et 250 regardent ce que l'on a touché par l'appel du sous-programme en 300-320.

La ligne 260 incrémente le score.

Les lignes 270 et 280 s'occupent de la fin de partie.

POUR JOUER :

Lancer le programme par RUN, et taper le nombre de cibles désiré suivi de RETURN. Les cibles s'affichent, et la partie commence. Votre balle, représentée par un * est affichée en bas à gauche ; vous pouvez la déplacer avec les flèches haut et bas (touches A et Z), et vous pouvez tirer, à tout moment en tapant Ø.

Le score sera modifié en fonction de votre résultat, et une nouvelle balle apparaît. Il est possible d'arrêter à tout moment en tapant CNT et C, 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 REM CIBLES
20 HOME
30 INPUT "NOMBRE DE CIBLES (DE 1
   A 5) : ";N: IF N < 1 OR N >
   5 THEN PRINT CHR$(7);: GOTO
   20
40 P = 10:J = 30:S = 0:B = 1:X$ =
   "*"
50 HOME
60 INVERSE : FOR I = 1 TO N
70 VTAB I * 4 - 3: HTAB 38: PRINT
   "   ": FOR K = 1 TO 3: VTAB
   I * 4 - 3 + K: HTAB 40: PRINT
   "   ": NEXT K
80 NEXT I: VTAB N * 4 + 1: HTAB
   38: PRINT "   ": NORMAL
90 K = 1:A = 18
100 VTAB 22: HTAB 1: PRINT "----
   SCORE : ";S: TAB( 20);"----
   BALLE : ";B
110 VTAB A: HTAB K: PRINT X$
120 D = A
130 FOR I = 1 TO N
140 U = INT ( RND (1) * P)
150 VTAB I * 4 - 1: HTAB 38: PRINT
   U
160 G = PEEK ( - 16384): IF G >
   127 THEN POKE - 16368,0: GOTO
   180
170 NEXT I
180 A = A + (G = 218 AND A < N *
   4 + 1) - (G = 193 AND A > 1)

190 VTAB D: HTAB K: PRINT " "
200 IF G = 176 AND K = 1 THEN B =
   B + 1: GOTO 220
210 IF K = 1 THEN 110
220 K = K + 1
230 IF K < 38 THEN 110
240 GOSUB 310:F = PEEK (E)
250 IF F < 176 OR C > 185 THEN VTAB
   23: INVERSE : PRINT "=== PE
   RDU ...": NORMAL : END
260 S = S + F - 176
270 P = P - 1
280 IF P < > 0 THEN 90
290 END
300 U = X * 2 + 4:V = Y * 2 + 4
310 C = INT ((A - 1) / 8):D = A -
   C * 8:E = 895 + 128 * D + 40
   * C + K
320 RETURN

```

Gouttes

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : dessin.

Difficulté du jeu : aucune.

LE JEU :

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

LE PROGRAMME :

Les deux boucles emboîtées I (50-110) et J (60-100) dessinent le récipient et les obstacles.

La ligne 150 dessine la goutte et la ligne 190 envoie dans un sous-programme (350-380) qui permet de déterminer s'il y a ou non rencontre avec un obstacle.

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

Si c'est oui, la goutte part à droite ou à gauche (ligne 210) selon la valeur définie pour K en ligne 130.

On regarde de nouveau s'il y a ou non obstacle, ou si la goutte peut continuer à tomber (lignes 220, 230 ou 240).

Selon le résultat du test, on utilisera 250, 270, 290 ou 320.

POUR JOUER :

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

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

EXTENSIONS POSSIBLES :

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

```

10  REM GOUTTES
20  C1 = 7:C2 = 5:C3 = 4
30  HOME
40  GR
50  FOR I = 23 TO 38
60  FOR J = 1 TO 38
70  COLOR= C1: PLOT 0,I: PLOT 39,
    I
80  PLOT J,39
90  IF I < > 23 AND RND (1) < .
    3 THEN COLOR= C2: PLOT J,I
100 NEXT J
110 NEXT I
120 Y = INT ( RND (1) * 36 + 2):
    X = 6
130 K = 2 * ( RND (1) < .5) - 1
140 I = 0
150 Z = PEEK ( - 16336) + PEEK
    ( - 16336): COLOR= C3: PLOT
    Y,X
160 FOR J = 1 TO 100: NEXT
170 COLOR= 0: PLOT Y,X
180 U = X + 1:V = Y
190 GOSUB 350
200 IF J = 0 THEN 270
210 U = X:V = Y + K
220 GOSUB 350
230 IF J = 0 THEN 290
240 IF I = 0 THEN 320
250 COLOR= C3: PLOT Y,X
260 GOTO 120
270 X = X + 1
280 GOTO 130
290 Y = Y + K
300 I = 1
310 GOTO 150
320 K = - K
330 I = 1
340 GOTO 210
350 C = SCRN( V,U)
360 J = (C = C2) OR (C = C1) OR (
    C = C3)
370 IF U = 39 THEN I = 1
380 RETURN

```

Obstacles

REFERENCES :

Difficulté de programmation : assez difficile.

Catégorie de jeu : jeu de réflexion (pour un joueur).

Difficulté du jeu : moyenne.

LE JEU :

Des piétons se déplacent sur l'écran pour venir à la rencontre d'une petite voiture qui dispose de la possibilité de se déplacer vers la gauche ou vers la droite pour les éviter et qui devra tenir le plus longtemps possible.

La distance parcourue sera affichée en permanence, et le jeu se termine lorsque la voiture rencontre, soit le bord du terrain, soit un piéton.

LE PROGRAMME :

La ligne 50 fait avancer les piétons vers la voiture du joueur qui est imprimée grâce à la ligne 40.

Les lignes 50, 60, 70 fabriquent la route (les lignes 60 et 70 bien qu'identiques sont toutes deux indispensables, car la valeur de RND ne sera, en principe, pas la même sur ces deux lignes).

La ligne 90 affiche le score.

La ligne 110 permettra de déplacer le véhicule.

Les lignes 120 et 130 permettent de contrôler s'il y a ou non un obstacle et arrêtent le programme s'il y a lieu.

Le score est incrémenté en ligne 140.

POUR JOUER :

On lance le programme en tapant RUN.

La route et les piétons apparaissent en bas de l'écran et montent vers la voiture. La voiture peut se déplacer à l'aide des flèches de déplacement horizontal (touches → et ←).

La rencontre avec un piéton ou la bordure interrompt le déroulement du programme et provoque l'affichage du score.

On peut, cependant, sortir du programme à tout moment en utilisant les touches CNT et C.

EXTENSIONS POSSIBLES :

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

```

10  REM OBSTACLES
20  HOME
30  S = 0:Y = 5
40  VTAB 1: HTAB Y: PRINT "#"
50  VTAB 24: HTAB 1: PRINT CHR$
      (127) + "      " + CHR$ (
      127);
60  IF RND (1) < .7 THEN VTAB 2
      4: HTAB INT ( RND (1) * 8 +
      2): PRINT "^";
70  IF RND (1) < .8 THEN VTAB 2
      4: HTAB INT ( RND (1) * 8 +
      2): PRINT "^"
80  VTAB 8: HTAB 16: CALL - 868
90  VTAB 9: HTAB 16: PRINT "SCORE
      : ";S
100 K = PEEK ( - 16384): IF K >
      127 THEN POKE - 16368,0
110 Y = Y + (K = 149 AND Y < 10) -
      (K = 136 AND Y > 1)
120 E = 1023 + Y
130 IF PEEK (E) = 222 OR PEEK
      (E) = 255 THEN POKE E, PEEK
      (E) - 128: END
140 S = S + 1
150 GOTO 40

```

Captures dans l'espace

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexe (1 seul joueur).

Difficulté du jeu : moyenne.

LE JEU :

Votre vaisseau spatial navigue dans un dangereux champ de météorites et votre mission est de vous emparer de la plus grande quantité possible de minerai d'urane, un métal indispensable que l'on ne peut trouver que dans ces lieux particulièrement dangereux.

Vous devez arriver à manœuvrer pour recueillir l'urane, sans vous faire heurter par une météorite.

LE PROGRAMME :

La ligne 50 permet de faire apparaître les météorites.

La ligne 70 fera apparaître le minerai d'urane et la ligne 80 assure le déplacement des météorites sur l'écran.

Les lignes 100, 130 et 140 contrôlent le déroulement du jeu (a-t-on heurté une météorite ou a-t-on capturé un bloc d'urane ?).

La rencontre d'une météorite provoque la fin de partie.

Votre vaisseau spatial est imprimé par les lignes 150 et 160 et son déplacement est contrôlé par la ligne 120.

POUR JOUER :

On lance le programme par RUN et les météorites commencent à monter sur l'écran, ainsi que les blocs d'urane.

Les touches de déplacement horizontal (→ et ←) seront utilisées pour déplacer votre vaisseau et récupérer l'urane, mais le moindre heurt avec une météorite provoquera la fin de la partie et l'affichage de la quantité d'urane capturée.

On pourra aussi arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :

On peut rendre le jeu de plus en plus difficile en augmentant progressivement le nombre de météorites, ou encore, en minutant la durée de la partie.

On peut aussi prévoir un affichage permanent du score, introduire un deuxième vaisseau spatial manœuvré par une puissance étrangère et qui cherche aussi à s'emparer de l'urane...

```

10  REM CAPTURES DANS L'ESPACE
20  HOME
30  X = 15:Z = 0
40  S = INT ( RND (1) * 39 + 1)
50  INVERSE : VTAB 23: HTAB S: PRINT
    "#";: NORMAL :Y = PEEK ( -
    16336) + PEEK ( - 16336)
60  S = INT ( RND (1) * 39 + 1)
70  IF RND (1) < .7 THEN VTAB 2
    3: HTAB S: PRINT "*";:Y = PEEK
    ( - 16336)
80  PRINT : PRINT
90  FOR I = 1 TO 10: NEXT
100 IF C = 170 THEN PRINT CHR$
    (7);:Z = Z + 1
110 X1 = X
120 X = X + (K = 21 AND X < 38) -
    (K = 8 AND X > 2)
130 C = PEEK (1447 + X)
140 IF C = 35 THEN INVERSE : PRINT
    "—— SCORE : ";Z;". ": NORMAL
    : END
150 VTAB 11: HTAB X1: PRINT " "
160 VTAB 12: HTAB X: PRINT "+"
170 K = PEEK ( - 16384): IF K >
    127 THEN POKE - 16368,0
180 GOTO 40

```

Anomalie

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : solitaire.

Difficulté du jeu : moyenne.

LE JEU :

Il repose essentiellement sur une bonne capacité d'observation. Deux grilles de 25 cases chacune sont dessinées sur l'écran. Des symboles identiques sont affichés en des endroits différents des deux grilles, mais l'un des symboles a été modifié sur la grille de droite. Le jeu consiste à trouver lequel, en entrant ses coordonnées dans l'ordre ligne-colonne.

LE PROGRAMME :

La boucle I (40-70) imprime les chiffres de repérage des grilles.

La boucle principale I (80-140) imprime les symboles des deux grilles, dans un ordre différent.

La ligne 90 choisit le caractère à imprimer.

Le sous-programme (260-300) s'occupe de faire l'impression à un emplacement libre (on l'appelle deux fois pour les deux grilles).

Les lignes 150 et 160 changent, de façon aléatoire, l'un des caractères.

Les coordonnées du point essayé sont entrées aux lignes 170 et 180, et le contrôle est effectué par les lignes 190 et 200.

Les lignes 210 et 220 s'occupent de l'affichage du résultat.

POUR JOUER :

Après avoir lancé le programme par l'instruction RUN, on obtient l'affichage des deux grilles. On devra alors chercher, par comparaison, quel est le caractère qui a été modifié (sur la grille de droite).

On entre alors son numéro de ligne, suivi de RETURN, et son numéro de colonne, suivi de RETURN.

Le programme répondra alors BRAVO ou PERDU.

EXTENSIONS POSSIBLES :

Il sera possible d'utiliser des symboles graphiques de diverses couleurs, ce qui pourra simplifier la recherche.

On peut aussi envisager d'utiliser des dessins plus sophistiqués, améliorer le contrôle du résultat, admettre la bonne réponse sur la grille de droite ou celle de gauche, introduire un minutage, etc.


```

10 REM ANOMALIE
20 HOME
30 T = 5
40 FOR I = 1 TO 5
50 VTAB 1: HTAB 2 * I + 2: PRINT
   I: VTAB 2 * I + 1: HTAB 1: PRINT
   I
60 VTAB 1: HTAB 2 * I + 22: PRINT
   I: VTAB 2 * I + 1: HTAB 21: PRINT
   I
70 NEXT I
80 FOR I = 1 TO 22
90 X = INT ( RND (1) * 26 + 1)
100 L = 2: S = T
110 GOSUB 260
120 L = 22
130 GOSUB 260
140 NEXT I
150 S = 0
160 GOSUB 260
170 VTAB 15: HTAB 1: INPUT "LIGN
   E ... : ";A
180 VTAB 15: HTAB 20: INPUT "COL
   ONNE ... : ";B
190 U = 2 * A + 1: V = 2 * B + 22
200 GOSUB 240
210 IF X + 192 = K THEN VTAB 18
   : HTAB 1: PRINT "=== BRAVO
   ... ": NORMAL : END
220 VTAB 18: HTAB 1: INVERSE : PRINT
   "=== PERDU ... IL FALLAIT TR
   OOVER "; CHR$ (X + 64): ". ":
   NORMAL
230 END
240 C = INT ((U - 1) / 8): D = U -
   C * 8: E = 895 + 128 * D + 40
   * C + V: K = PEEK (E)
250 RETURN
260 U = 2 * INT ( RND (1) * T +
   1) + 1: V = 2 * INT ( RND (1
   ) * S + 1) + L
270 GOSUB 240
280 IF K < > 160 AND S < > 0 THEN
   260
290 VTAB U: HTAB V: PRINT CHR$
   (X + 64)
300 RETURN

```

Plusieurs en un

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : deux joueurs.

Difficulté du jeu : variable, selon l'utilisation.

LE JEU :

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

LE PROGRAMME :

Le terrain de jeu est délimité par un cadre mis en place par les lignes 60-70 suivant les valeurs indiquées à la ligne 40.

Le changement de joueur est effectué par la ligne 90.

Les lignes 170 et 180 permettent aux deux joueurs de déplacer le curseur sur le terrain.

La ligne 210 contrôle si le déplacement est possible (case libre et non dépassement des limites).

La ligne 220 permet d'écrire sur une case libre.

POUR JOUER :

RUN provoque l'affichage des limites du terrain et du numéro du joueur, ainsi que d'un curseur que l'on devra déplacer sur l'écran à l'aide des 4 touches I, J, K, M.

Il n'est pas possible de venir sur une case déjà occupée et il faudra appuyer sur la touche RETURN pour inscrire son propre symbole dans une case disponible.

Dès que cela a été fait, le symbole du second joueur réapparaît 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 /(slash).

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.

On pourra aussi utiliser le joystick ou les paddles pour déplacer le curseur.

```

10 REM PLUSIEURS EN UN
20 HOME
30 A$(0) = "X":A$(1) = "0"
40 L1 = 3:L2 = 19:C1 = 3:C2 = 37:
    REM LIMITE ECRAN
50 INVERSE
60 FOR I = L1 TO L2: VTAB I: HTAB
    C1: PRINT " ": VTAB I: HTAB
    C2: PRINT " ": NEXT I
70 FOR I = C1 TO C2: VTAB L1: HTAB
    I: PRINT " ": VTAB L2: HTAB
    I: PRINT " ": NEXT I
80 NORMAL
90 J = ABS (J - 1)
100 X = INT ((L2 - L1) / 2):Y =
    INT ((C2 - C1) / 2)
110 GOSUB 250: POKE E,K - 128
120 Z = PEEK ( - 16384): IF Z <
    128 THEN 120
130 POKE - 16368,0
140 IF Z = 175 THEN 240
150 IF Z = 141 THEN 210
160 X1 = X:Y1 = Y:E1 = E:K1 = K
170 X = X + (Z = 205 AND X < L2 -
    1) - (Z = 201 AND X > L1 + 1
    )
180 Y = Y + (Z = 203 AND Y < C2 -
    1) - (Z = 202 AND Y > C1 + 1
    )
190 POKE E,K
200 GOTO 110
210 IF K < > 160 THEN PRINT CHR$(
    7): GOTO 120
220 VTAB X: HTAB Y: PRINT A$(J)
230 GOTO 90
240 END
250 C = INT ((X - 1) / 8):D = X -
    C * 8:E = 895 + 128 * D + 40
    * C + Y:K = PEEK (E)
260 RETURN

```

Parcours du cavalier

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion.

Difficulté du jeu : difficile.

LE JEU :

Il se présente, ici, comme un jeu de hasard total, mais provient, en réalité, d'un problème assez difficile à résoudre : comment déplacer un cavalier sur un échiquier de façon qu'il passe une fois, et une seule, sur chacune des 64 cases du damier.

Le cavalier se déplace suivant les règles communément admises aux échecs, et sa position de départ est définie par le joueur ; par contre, ses déplacements sont effectués au hasard. Arrivera-t-il à parcourir les 64 cases ?

LE PROGRAMME :

La ligne 40 définit les divers déplacements possibles, dans deux variables alphanumériques A\$ et B\$; la boucle I (50-80) dessine le terrain de jeu.

On entre les coordonnées de la case de départ (ligne, puis colonne) aux lignes 90 à 130.

La ligne 140 imprime le numéro du coup.

La ligne 150 permet le tirage au hasard d'un déplacement du cavalier.

La boucle Z (160-220) cherche si le déplacement est possible ; si ce n'est pas le cas, on essaie tous les autres (toujours dans le même ordre).

Les lignes 170 et 180 effectuent le déplacement.

La ligne 190 contrôle que l'on ne sorte pas du terrain de jeu.

Les lignes 200-210 envoient en ligne 240 si le coup est possible.

La ligne 230 gère la fin de partie (cavalier bloqué) et affiche le score.

Les lignes 240 et 250 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 son numéro de colonne. 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 CNT et C.

EXTENSIONS POSSIBLES :

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

```

10 REM PARCOURS DU CAVALIER
20 HOME
30 S = 1:T = 0
40 A$ = "56653223":B$ = "653223" +
  A$ + "65":A$ = A$ + A$
50 FOR I = 1 TO 8
60 VTAB 1: HTAB 3 * I + 2: PRINT
  I
70 VTAB 2 * I + 1: HTAB 1: PRINT
  I;" .. .. .
  "
80 NEXT I
90 VTAB 19: HTAB 1: PRINT "COORD
  ONNEES DE DEPART ..... : ": VTAB
  21: PRINT "LIGNE ... : C
 OLONNE ... : "
100 VTAB 21: HTAB 13: GET K$:I =
  VAL (K$): IF I < 1 OR I > 8
  THEN PRINT CHR$ (7):: GOTO
  100
110 PRINT I
120 VTAB 21: HTAB 32: GET K$:J =
  VAL (K$): IF J < 1 OR J > 8
  THEN PRINT CHR$ (7):: GOTO
  120
130 PRINT J
140 VTAB 2 * I + 1: HTAB 3 * J +
  1: PRINT RIGHT$ ("0" + STR$
  (S),2)
150 Z = INT ( RND (1) * 8 + 1)
160 FOR Z = Z TO Z + 8
170 A = I + VAL ( MID$ (A$,Z,1))
  - 4
180 B = J + VAL ( MID$ (B$,Z,1))
  - 4
190 IF A < 1 OR B < 1 OR A > 8 OR
  B > 8 THEN 220
200 X = 2 * A + 1:Y = 3 * B + 1: GOSUB
  270
210 IF K = 174 THEN 240
220 NEXT Z
230 VTAB 22: INVERSE : PRINT "TO
  TAL ..... : ";S;" ". : NORMAL
  : END
240 I = A:J = B
250 S = S + 1: GOTO 140
260 END
270 C = INT ((X - 1) / 8):D = X -
  C * 8:E = 895 + 128 * D + 40
  * C + Y:K = PEEK (E)
280 RETURN

```

Mur de briques

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexe (pour 1 joueur).

Difficulté du jeu : difficile.

LE JEU :

S'il est un jeu que l'on peut présenter comme un grand classique des jeux d'arcades pour micro-ordinateur, c'est bien le mur de briques (en anglais BREAKOUT).

Il est offert, ici, dans une version simplifiée, mais offre néanmoins de bonnes performances : affichage du score en fin de partie, rebonds aléatoires, etc. Le seul problème en est la rapidité.

La règle en est très élémentaire : vous disposez d'une raquette qui peut être déplacée horizontalement pour renvoyer une balle (matérialisée par un point) qui va démolir les briques dessinées en haut de l'écran.

Votre problème est d'en démolir le plus grand nombre possible, sans rater la balle, ce qui provoquerait la fin de partie et l'affichage du score.

LE PROGRAMME :

Les lignes 60 à 80 dessinent le terrain de jeu et les briques sont mises en place par les boucles emboîtées I (90-130) et J (100-120).

La ligne 160 dessine la raquette dont la position est contrôlée par la ligne 180.

Les lignes 230 à 260 regardent si l'on a touché le mur, les briques ou la raquette.

Les rebonds sont gérés en conséquence avec des éventuels aléas, par les lignes 270 à 320.

La ligne 200 affiche le score en permanence.

POUR JOUER :

Le programme sera lancé par l'instruction RUN, ce qui provoque l'affichage du terrain de jeu puis, très rapidement, l'envoi de la balle. Le joueur devra alors essayer de déplacer sa raquette à l'aide des touches de déplacement horizontal (→ et ←).

Si, à un moment quelconque, sa balle ne touche pas la raquette, le score sera affiché et la partie s'arrête.

On peut aussi terminer à tout moment par CNT et C.

EXTENSIONS POSSIBLES :

On peut envisager d'agrandir le terrain de jeu, d'augmenter le nombre de briques, de prévoir plusieurs balles, d'introduire des effets sonores.

On peut aussi, en s'inspirant des jeux d'arcades, introduire de petits ouvriers qui viendront reconstruire votre mur de briques, au fur et à mesure que vous le démolissez.

```

10 REM MUR DE BRIQUES
20 HOME
30 GR
40 C1 = 3:C2 = 15:C3 = 7:C(0) = 4
   :C(1) = 5
50 COLOR= C1
60 VLIN 0,37 AT 6
70 HLIN 6,33 AT 0
80 VLIN 0,37 AT 33
90 FOR I = 1 TO 12:C = ABS (C -
   1)
100 FOR J = 7 TO 32
110 C = ABS (C - 1): COLOR= C(C)
   : PLOT J,I
120 NEXT J
130 NEXT I
140 H = - 1:K = - 1:J = 19:I =
   37:Y = J
150 COLOR= 0: HLIN Y - 3,Y + 1 AT
   38
160 COLOR= C2: HLIN Y - 2,Y AT 3
   8
170 A = PEEK ( - 16384): IF A >
   127 THEN POKE - 16368,0
180 Y = Y + (A = 149 AND Y < 32) -
   (A = 136 AND Y > 9)
190 IF T = 1 THEN COLOR= 0: PLOT
   J,I
200 VTAB 22: PRINT "SCORE : ";S
210 IF I > 38 THEN PRINT CHR$
   (7);: END
220 I = I + H:J = J + K
230 C = SCRN( J,I)
240 IF C = 0 THEN COLOR= C3: PLOT
   J,I:T = 1: GOTO 150
250 IF C = C1 OR J < 7 OR K > 32
   THEN 290
260 IF C = C(0) OR C = C(1) THEN
   COLOR= C3: PLOT J,I:Z = PEEK
   ( - 16336) + PEEK ( - 16336
   ):T = 1: GOTO 320
270 H = - H
280 K = RND (1) - RND (1)
290 K = - K:T = 0
300 Z = PEEK ( - 16336) - PEEK
   ( - 16336)
310 GOTO 150
320 S = S + 13 - I
330 K = - (K - (2 * ( RND (1) <
   .5) - 1) * RND (1) / 2)
340 H = - H
350 GOTO 150

```

Othello

REFERENCES :

Difficulté de programmation : difficile.

Catégorie de jeu : réflexion (2 joueurs).

Difficulté du jeu : difficile.

LE JEU :

Devenu un grand classique des jeux sur micro-ordinateur, le jeu d'Othello ou Reversi gagne, même si comme ici il est pratiqué à deux joueurs, à être géré sur ordinateur, avec visualisation : en effet, le jeu consistant essentiellement en des retournements des pions de l'adversaire, ceci peut se révéler, à la longue, très fastidieux, surtout si vous êtes un joueur assidu. Le principe est que le joueur possédant les X doit pour pouvoir jouer, réussir à encadrer une ou plusieurs des pièces de son adversaire, dans une ou plusieurs directions ; s'il y réussit, les O seront alors retournés pour devenir des X (le retournement pourra être fait horizontalement, verticalement ou suivant les diagonales).

Le joueur ayant les O procédera de la même façon, et ceci jusqu'à ce que le damier soit rempli ou que la personne ne puisse plus jouer (si un joueur ne peut retourner des pions, il doit passer son tour).

LE PROGRAMME :

La boucle I (40-60) affiche le terrain de jeu et la boucle H (70-80) les 4 premières pièces.

La ligne 110 assure le changement de joueur et affiche celui qui a le trait.

La ligne 130, par un appel au sous-programme 470-480, fait flasher la case où l'on se trouve (ou l'inverse, si c'est une case libre).

La ligne 140 permet la frappe d'une touche.

S'il s'agit d'un déplacement, il est géré par les lignes 180 à 220.

Si c'est un slash (/), il y a changement de joueur (ligne 160).

Si c'est un RETURN, le contrôle et l'affichage du coup sont réalisés par les lignes 230 à 450. La boucle P (260-400) examine les 8 directions possibles.

La ligne 450 détecte la fin du jeu.

POUR JOUER :

Après le démarrage du programme par RUN, le joueur qui a le trait peut déplacer le curseur grâce aux 4 touches (I, J, K, M).

Il tape RETURN quand il est sur la case qu'il veut jouer. Si le coup est valide, l'APPLE II effectue le retournement des pions, comptabilise et affiche le score, et passe la main à l'adversaire.

Sinon, l'APPLE II le signale par un BIP et redonne la main au même joueur.

En cas d'impossibilité de jeu, il suffit de taper un slash (/) pour changer de main.

EXTENSIONS POSSIBLES :

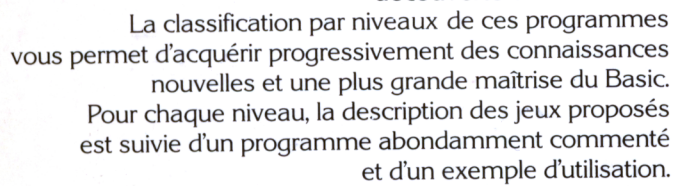
Avant tout, bien sûr, améliorer l'affichage avec des couleurs, une grille plus aérée.

On pourra aussi imaginer un jeu machine pour l'APPLE II.

```
10 REM OTHELLO
20 HOME
30 A$(0) = "X":A$(1) = "O"
40 FOR I = 1 TO 8: VTAB 1: HTAB
   I * 3 + 1: PRINT I
50 VTAB I * 2 + 1: PRINT I;"
   . . . . .
60 NEXT I
70 FOR H = 0 TO 1: VTAB 9: HTAB
   13 + 3 * H: PRINT A$(H)
80 VTAB 11: HTAB 16 - 3 * H: PRINT
   A$(H): VTAB 6 + 2 * H: HTAB
   27: PRINT "JOUER ";A$(H): NEXT
   H
90 S(0) = 2:S(1) = 2:H = 1
100 VTAB 6 + 2 * H: HTAB 34: PRINT
   A$(H)
110 H = ABS (H - 1): VTAB 6 + 2 *
   H: HTAB 34: FLASH : PRINT A$(
   H): NORMAL
120 I = 3:J = 3
130 GOSUB 470: POKE E,K - 128 *
   (K > 127)
140 Z = PEEK ( - 16384): IF Z <
   128 THEN 140
150 POKE - 16368,0
160 IF Z = 175 THEN POKE E,K: GOTO
   100
170 IF Z = 141 THEN 230
180 X1 = X:Y1 = Y:E1 = E:K1 = K
190 I = I + (Z = 205 AND I < 8) -
   (Z = 201 AND I > 1)
200 J = J + (Z = 203 AND J < 8) -
   (Z = 202 AND J > 1)
210 POKE E,K
220 GOTO 130
230 IF K < > 174 THEN PRINT CHR$(
   7):: GOTO 140
240 E1 = E:I1 = I:J1 = J:X1 = X:Y
   1 = Y:K1 = K
250 S = 0
260 FOR P = 1 TO 8:T = 0
270 I = I1:J = J1
```

Achevé d'imprimer
sur les presses de l'imprimerie IBP
à Rungis (Val-de-Marne 94) 686.73.54
Dépôt légal - octobre 1984

N° d'impression: 4825
N° d'édition: 86595-196-1
N° d'ISBN: 2-86595-196-0



Coverture GIBRAT



102 PROGRAMMES POPULAIRES

PAR JACQUES DECROIX



PROGRAMMES