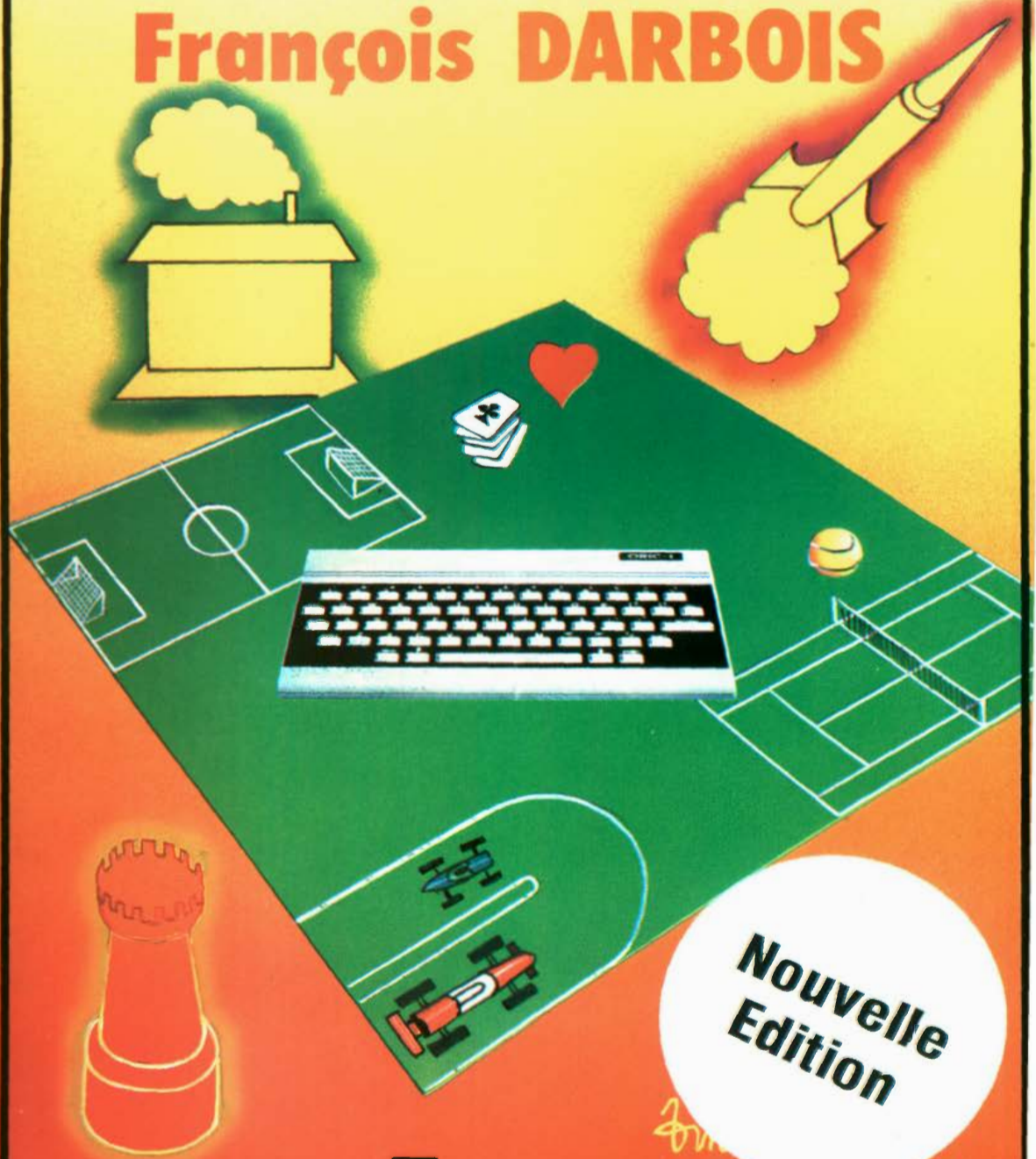
JEUX SUR ORIC

David CHANE-HUNE
François DARBOIS



Nouvelle Edition

Edimicro

JEUX SUR ORIC

Cet ouvrage a été réalisé sous la direction de Benoît de MERLY.

Première édition

ISBN : 2-904457-06-2

EDIMICRO

DÉPARTEMENT ÉDITIONS DE F.D.S. SARL

121-127, avenue d'Italie, 75013 Paris

JEUX SUR ORIC

par

David CHANE-HUNE
François DARBOIS

Edimicro

Avertissement

Le livre "Jeux sur Oric" contient un nombre assez important de listings. Pour ceux d'entre vous qui n'ont pas encore une grande expérience de la programmation, nous fournissons une liste non exhaustive de conseils :

- Recopier scrupuleusement tous les caractères avant de faire d'éventuelles modifications.

- Respecter les majuscules et les minuscules.

- Conserver le même nombre de caractères d'espacement dans les chaînes de caractères.

- L'imprimante utilisée pour les listings édite 67 caractères par ligne. Pour les instructions plus longues, elle continue à la ligne suivante. En conséquence, ne pas appuyer sur la touche [RETURN] à la fin de la ligne, mais à la fin de l'instruction (cette fin d'instruction est mise en évidence par le numéro de l'instruction suivante.

- Ne pas confondre le chiffre zéro et la lettre O, ni le chiffre un et la lettre I (sur les listings de jeu, les zéros sont barrés : Ø).

- Enfin, sauvegarder votre programme sur cassette avant toute exécution, car l'exécution d'une mauvaise instruction peut vous faire perdre la main. Le seul moyen de pouvoir dialoguer avec l'Oric étant parfois de le débrancher puis de le rebrancher, vous perdriez alors le fruit d'un long travail de patience.

Remarque : Les programmes où l'on utilise le clavier pour déplacer un mobile sur l'écran peuvent être améliorés si vous changez la vitesse de l'auto repeat avant toute exécution. Vous l'obtiendrez en plaçant en #306 et #307 une valeur comprise entre #960 et #2710, par l'instruction DOKE #306, #1000 (si l'on veut mettre #1000 en #306, 307).

Sommaire

CHAPITRE **1**

Techniques de programmation des jeux

Le but de ce chapitre est d'amener le lecteur à programmer lui-même ses propres jeux sur son ordinateur personnel Oric. Des connaissances générales sur le langage BASIC seront supposées connues (si cela n'est pas encore le cas, il sera utile de lire, dans la même collection, le livre "Guide de l'Oric" où les principes de la programmation BASIC sont exposés).

Plutôt qu'une longue et fastidieuse énumération d'instructions et de fonctions, ce chapitre sera composé d'une série de courts exemples montrant comment se pose un problème de programmation de jeu et comment le résoudre à partir des possibilités - très étendues au demeurant - de l'Oric en matière de jeux vidéo. Quand vous aurez lu ce qui suit, vous serez en mesure de créer des jeux qui seront aussi enthousiasmants et passionnants que les jeux du commerce, ou ceux que l'on trouve dans les cafés.

1.1 COMMENT PROGRAMMER UN JEU

C'est la question que vous vous posez tous. Voici une liste des opérations à effectuer dans le cas général et dans un ordre chronologique. Cette liste n'est pas limitative et en matière de jeux toute initiative est la bienvenue.

Tout d'abord il faut vous astreindre à ne pas commencer directement à "pianoter" sur le clavier de l'Oric. La création d'un jeu se fait au brouillon - c'est préférable car cela permet de mieux structurer son programme, rendant ainsi la mise au point plus rapide et plus simple.

Donc, au départ, il va falloir choisir un jeu soit en faisant appel à votre imagination débordante et fertile soit en prenant un jeu classique et éprouvé.

Puis vient la phase de définition des règles du jeu. Tout jeu doit la comporter. Celle-ci doit être écrite avec précision et détail. Combien y aura-t-il de joueurs? Comment se compteront les points ? Quelle en sera la durée ? Qu'est-ce qui détermine la fin du jeu ? ... etc.

A ce niveau, des limitations du jeu sont à définir ainsi que la spécification des coups permis et ceux qui sont interdits. Des restrictions à la règle du jeu permettent souvent, si elles sont bien choisies, de simplifier beaucoup la programmation sans pour autant limiter l'attrait et l'amusement qu'apporte le jeu.

Là commence l'aspect informatique du jeu car il va vous falloir maintenant choisir la représentation matérielle du jeu (mode graphique utilisé, messages apparaissant en cours de jeu, nom des variables utilisées ...).

Ces choix sont fonction de vos dons artistiques car le graphisme de l'Oric permet une présentation impeccable des jeux.

Les étapes préliminaires ont pour but de simplifier l'étape suivante qui est l'écriture du programme proprement dit. Elles permettent d'éviter les ambiguïtés pouvant intervenir en cours d'exécution du jeu si toutes les possibilités n'ont pas été étudiées préalablement. Il est

plus simple de prévoir une éventualité plutôt que de la corriger sur un programme après coup. L'écriture se fera de préférence sur papier, en commençant par le programme principal qui assurera les fonctions minimales du jeu, c'est-à-dire l'appel aux sous-programmes utilitaires qui gèrent le calcul des points, l'affichage, les déplacements, la stratégie de l'ordinateur, les sons, le dessin du terrain de jeu, etc. L'intérêt d'avoir un programme principal et de nombreux sous-programmes est que la mise au point de ceux-ci peut être effectuée indépendamment l'un de l'autre, permettant ainsi de bâtir petit-à-petit un jeu long et compliqué, en partant d'un programme principal simple.

Si l'on veut effectuer des modifications dans l'aspect du jeu, il suffira de modifier un seul sous-programme au lieu de modifier l'ensemble du programme.

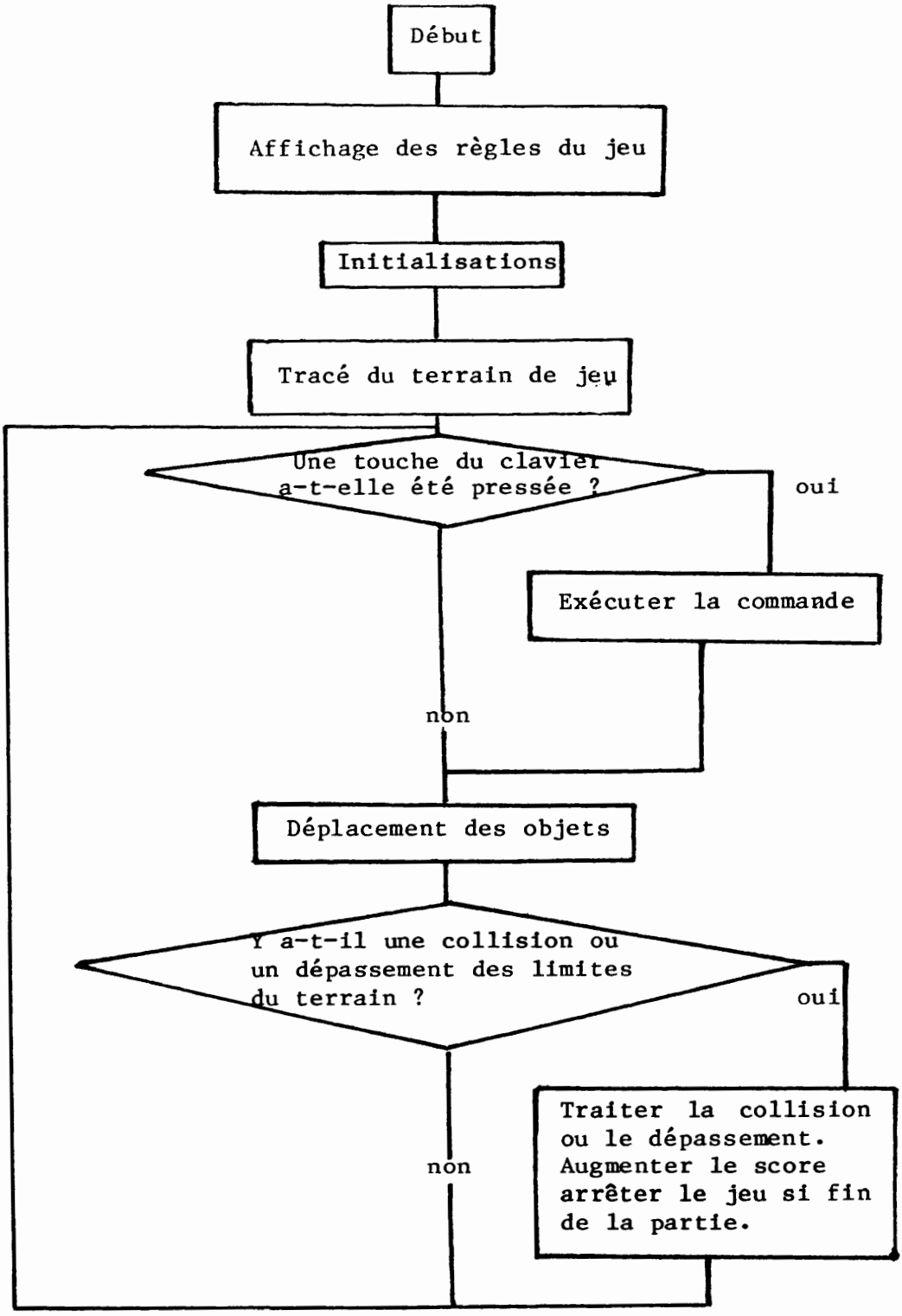
1.2 STRUCTURE GÉNÉRALE D'UN PROGRAMME DE JEU

Voyons maintenant quel est l'agencement d'un programme. On peut distinguer deux sortes de structures selon qu'il s'agit d'un jeu rapide ou d'un jeu lent.

La première catégorie correspond aux jeux dits "en temps réel". Le jeu se déroule alors tout seul sans avoir besoin du joueur, l'ordinateur déplaçant des objets (par exemple une balle) sur l'écran selon les règles du jeu. Le joueur n'influe que sur la valeur de certains paramètres (par exemple la position de la raquette), par action sur le clavier ou sur une poignée de jeu. L'acquisition de ces paramètres ne doit pas demander l'interruption de l'exécution du programme. Cela est obtenu par une structure rebouclée sur elle-même.

Le programme ne se termine pas par un END mais par un GOTO vers le début du programme.

Cette boucle principale contient une série de tests qui déterminent l'évolution future du terrain de jeu. Leur nombre doit être le plus réduit possible, pour des raisons de rapidité. Ils devront comprendre une scrutation du clavier qui sera périodique puisqu'elle se fera à chaque tour de la boucle. Voici une représentation de la structure générale de ce type de jeu :



Ordinogramme d'un jeu d'action rapide

La boucle principale doit s'exécuter le plus rapidement possible. Elle ne doit servir qu'à détecter la pression d'une touche, l'arrivée d'un événement (collision, dépassement des limites du jeu, du temps, impact, etc.) qui modifiera le déroulement du jeu. Le traitement de ces événements est effectué, lui, en dehors de cette boucle principale. On utilisera pour cela un certain nombre de sous-programmes effectuant ces opérations. Il pourra s'agir du comptage des points par incrémentation d'une variable, du déplacement du joueur d'après la touche pressée, de tests plus fins de débordement ou de collision permettant un traitement spécifique de ces événements. Ils pourront être aussi longs que l'on veut, leur durée étant moins critique.

Dans la deuxième catégorie on trouve les jeux lents. Ce sont tous les autres jeux qui font appel au joueur pour continuer la partie, en lui laissant un temps de réflexion. Les instructions peuvent alors s'ordonner linéairement selon la structure suivante :

1) Présentation du jeu.

2) Jeu proprement dit.

3) Détermination du score et affichage des résultats.

4) Fin du jeu, ou demande de nouvelle partie (on relance alors en 2).

L'ordinateur sert alors d'arbitre et d'adversaire avec qui l'on communique par le biais de questions et de réponses au clavier. L'ordinateur attend une réponse pour réagir selon les règles établies.

Le temps d'exécution n'est pas critique. On peut alors établir des stratégies de jeu beaucoup plus complexes que dans les jeux d'action.

Voyons maintenant comment l'on doit utiliser les possibilités du langage BASIC pour la programmation des jeux.

1.3 PROGRAMMATION DES JEUX D'ACTION

Ce qui fait la qualité d'un jeu d'action, c'est sa rapidité à déplacer des objets sur l'écran.

Pour recréer l'illusion du mouvement une méthode d'animation - couramment utilisée- consiste à visualiser un caractère (graphique ou redéfini à votre goût) à une coordonnée puis, après une courte attente, à imprimer un caractère "blanc" (espacement) à la même coordonnée. Ce qui effacera le caractère précédent. Ensuite il faut calculer les nouvelles coordonnées puis recommencer au début. Cela consiste à faire de l'animation image par image, tout comme au cinéma.

On voit alors que la vitesse de déplacement dépend de l'écart entre chaque nouvelle coordonnée (qui ne doit pas être trop grand si l'on désire des mouvements qui n'apparaissent pas saccadés). Elle dépend aussi du temps mis par la machine pour effectuer les impressions, de la durée de l'attente entre l'impression et l'effacement. Cette attente permet de laisser visible le caractère pendant une durée suffisante pour qu'il ne semble pas clignoter lors des déplacements.

Si dans un jeu il suffisait de déplacer un seul objet sans rien faire d'autre, on obtiendrait de grandes vitesses (et un jeu plutôt ennuyeux!). En général il y a plusieurs objets en mouvement en même temps. Cela allonge la durée de la boucle d'impression et limite en fait le nombre de mobiles, si l'on veut garder un jeu suffisamment rapide.

Les nouvelles coordonnées dépendent d'éventuelles collisions, de débordements de l'écran. La boucle doit donc comporter aussi des tests qui permettront de traiter cela. Tout ceci montre que, si l'on veut optimiser la vitesse, il faut agir sur les tests, la structure de la boucle et sur la prise en compte des ordres du clavier.

1.3.1 Déplacements

Pour commander le déplacement d'un objet sur l'écran, il est commode d'utiliser les touches comportant des flèches qui sont situées en bas du clavier. On utilise pour

cela la fonction KEY$ qui retourne une variable caractère dont le code ASCII vaut en décimal :

8 = vers la gauche

9 = vers la droite

10 = vers le bas

11 = vers le haut

La touche d'espacement peut être aussi utilisée (pour effectuer un tir par exemple) ; elle correspond au code "32".

La touche "Return" vaut "13".

Voyons ce que cela donne sur un exemple.

Pour scruter l'enfoncement d'une des quatre touches fléchées, on peut utiliser les instructions suivantes :

```
100  REM EXEMPLE A NE PAS SUIVRE !
110  IF KEY$ = CHR$(8) THEN GOSUB 1000
120  IF KEY$ = CHR$(9) THEN GOSUB 2000
130  IF KEY$ = CHR$(10) THEN GOSUB 3000
130  IF KEY$ = CHR$(11) THEN GOSUB 4000
150  GOTO 100
```

Les sous-programmes 1000 à 4000 contiennent les opérations à effectuer pour chacune des touches. Cette façon de procéder est lente. En effet, à chaque test, on prend un nouvel échantillon du clavier pour KEY$ (ce qui est assez long à effectuer) alors qu'il suffit de le faire une seule fois avant de commencer la série de tests. D'autre part, cette batterie de tests peut avantageusement être remplacée par la structure ON GOSUB ou ON GOTO qui s'exécutera plus vite car il n'y aura qu'une condition à évaluer.

Voici une meilleure façon de procéder :

```
100  M$ = KEY$
110  ON ASC(M$) - 7 GOSUB 1000, 2000, 3000,
     4000
120  GOTO 100
```

L'expression ASC(M$) - 7 vaut successivement 1, 2, 3, 4 suivant que l'on a appuyé sur la touche flèche gauche, droite, bas ou haut.

Après évaluation de cette expression, le programme effectue le sous-programme dont le numéro de ligne est à la place donnée par le résultat de cette expression.

Il peut malgré tout y avoir un problème si l'on appuie sur une autre touche que les flèches car la condition de saut prend alors une valeur différente de 1, 2, 3 ou 4. On peut alors faire précéder le ON GOSUB par un test préliminaire qui détectera si l'on a bien pressé sur une des touches valides.

```
 90  REM TEST DE DEPLACEMENT
100  M$ = KEY$
110  IF M$ >= CHR$(8) AND CHR$(11) >= M$
     THEN ON ASC(M$) - 7 GOSUB 1000, 2000,
     3000, 4000
120  GOTO 100
```

La ligne 120 relance le programme à la scrutation du clavier qui devient ainsi périodique. Cette boucle est parcourue assez rapidement pour que la prise en compte de la touche apparaisse instantanée au joueur, alors qu'en réalité elle n'est pas effectuée constamment. On effectue alors ce que l'on appelle un "polling" du clavier. Cette notion peut être étendue aux autres variables susceptibles d'être modifiées en cours de jeu, le résultat de la modification n'étant actif qu'à chaque rebouclage du "polling". Cela peut s'appliquer au changement de position sur l'écran.

La position d'un mobile sur l'écran peut être repérée par deux variables, x et y par exemple, qui seront ses coordonnées et que l'on pourra utiliser pour le visualiser par un PLOT en basse résolution ou un CURSET et un CHAR en haute résolution.

Pour un déplacement, il est préférable de modifier deux variables auxiliaires, Dx et Dy par exemple, qui représenteront les déplacements relatifs du mobile par rapport à sa position précédente.

Voici un exemple d'animation :

```
 10  REM SUPERBALLE
 20  CLS : LORES 0
 30  X = 3 : Y = 3 : DX = 1 : DY = 1
 50  REPEAT
120  X = X + DX
130  Y = Y + DY
140  PLOT X, Y, "O"
150  IF X < 3 OR X > 35 OR Y < 3 OR Y > 23
     THEN GOSUB 5000
170  PLOT X, Y, " "
190  UNTIL KEY$ = "S"
200  END
5000 IF X < 3 OR X > 35 THEN DX = -DX
5010 IF Y < 3 OR Y > 23 THEN DY = -DY
5020 RETURN
```

Comme précédemment, pour éviter d'avoir plusieurs tests de suite dans la boucle (réalisée ici par REPEAT UNTIL, ce qui permet un arrêt du jeu quand on presse la touche S (STOP!)), on effectue un test global pour savoir si la balle (la lettre "O") est sortie des limites du terrain. Si cela est le cas, on exécute alors le sous-programme 5000 qui calcule les déplacements relatifs de la balle pour que l'on ait un "rebond" sur les limites.

On peut améliorer ce sous-programme en introduisant un rebond et une vitesse aléatoires.

On écrira alors :

```
5000 IF X < 3 OR X > 35 THEN DX =
     - SGN(DX) * (INT(RND(1) * 2) + 1)
5010 IF Y < 3 OR Y > 23 THEN DY =
     - SGN(DY) * (INT(RND(1) * 2) + 1)
5020 RETURN
```

Si vous êtes fatigué de voir cette balle rebondir sans arrêt sur les bords de l'écran, il n'y a qu'à ajouter une raquette qui vous permettra de dévier la balle. Pour cela nous allons réutiliser l'exemple TEST DE DEPLACEMENT pour faire bouger la raquette.

Ajoutez dans le programme SUPERBALLE les lignes 100 et 110 de TEST DE DEPLACEMENT.

Si l'on veut que la raquette puisse bouger dans toutes les directions, il faut écrire quatre sous-programmes modifiant les coordonnées de la raquette XR, YR.

Ces coordonnées sont initialisées préalablement par la ligne supplémentaire :

```
40  XR = 10 : YR = 10
```

Voici les sous-programmes de déplacement de la raquette :

```
1000  IF XR < > 3 THEN XR = XR - 1 : PLOT XR,
      YR, "I" : PLOT XR + 1, YR, " "
1010  RETURN
2000  IF XR < > 35 THEN XR = XR + 1 : PLOT
      XR, YR, "I" : PLOT XR - 1, YR, " "
2010  RETURN
3000  IF YR < > 23 THEN YR = YR + 1 : PLOT
      XR, YR, "I" : PLOT XR, YR - 1, " "
3010  RETURN
4000  IF YR < > 3 THEN YR = YR - 1 : PLOT XR,
      YR, "I" : PLOT XR, YR + 1, " "
4010  RETURN
```

Il faut aussi tester la collision de la balle et de la raquette. Ajoutez donc la ligne suivante :

```
160  IF X = XR AND Y = YR THEN  DX =-DX :
     DY = - DY : PING
```

Que manque-t-il pour finir ce jeu ? Eh bien la création d'un score qui motivera le joueur. On peut compter le nombre de fois que vous avez touché la balle en un temps imparti.

Ajoutez alors à la suite de la ligne 160 :

```
: SCORE = SCORE + 1
```

Ce qui comptabilisera les coups de raquette.

Il faut aussi compter le temps. On comptera en fait le nombre de tours de la boucle REPEAT UNTIL.

Ajoutez :

```
180  TEMPS = TEMPS + 1
190  UNTIL KEY$ = "S" OR TEMPS = 1000
200  PRINT "SCORE:", SCORE
210  END
```

Amusez-vous bien à la superballe.

1.3.2 Ecriture condensée et choix des variables

Le BASIC de l'Oric permet certains accommodements d'écriture permettant un formalisme plus compact qui accélère légèrement l'exécution.

Vous pouvez écrire les instructions et les expressions sans blancs, mettre plusieurs instructions par ligne, supprimer les REM, supprimer les indices suivant les NEXT.

Tout cela réduit le nombre d'opérations élémentaires effectuées par l'interprêteur BASIC lors de l'exécution d'une ligne.

Mais cela entraîne un manque de lisibilité du programme et un risque d'erreur lors de modifications ultérieures du programme. Il vaut donc mieux ne recourir à ces méthodes qu'en dernier lieu, et - de toute façon - seulement lorsque le programme est déjà au point.

Il est préférable également de ne pas utiliser des variables sous forme de tableaux indicés qui demandent plus de temps pour trouver la valeur correspondant à l'indice, que lorsque l'on utilise une variable non indicée.

Utilisez de préférence XA, XB, XC que X(1), X(2), X(3) ...

N'oubliez pas que seuls les deux premiers caractères sont pris en compte dans le nom d'une variable. Cela fait énormément de noms possibles.

En fait pour obtenir des jeux très rapides et ayant beaucoup d'objets en mouvement simultanément, il faut écrire vos programmes directement en langage assembleur ou au moins certaines parties du jeu.

1.4 JEUX DE HASARD ET JEUX DE RÉFLEXION

1.4.1 Utilisation de RND

Un jeu est bien morne s'il ne comporte pas une part d'inattendu et de surprise apportée par l'intervention du hasard. Que ce soit le mélange de cartes à jouer au Poker, le lancé des dés au 421, l'arrêt de la bille à la roulette, ces événements dirigent le déroulement de la partie de manière imprévisible sans que l'on puisse agir dessus. Cela suffit à renouveler le jeu constamment.

Le BASIC permet facilement de créer un nombre dont la valeur est calculée au hasard en utilisant la fonction RND(1). Voyons cela sur un exemple :

```
 10  REM JEU DE LA FACE MAXIMUM
 20  INPUT "JET DU DE (O/N)" ; M$
 30  IF N$ <>"O" THEN END
 40  A = INT (RND(1) * 6) + 1
 50  B = INT (RND(1) * 6) + 1
 60  PRINT : PRINT "VOUS AVEZ FAIT UN : "; A
 70  PRINT : PRINT "J'AI FAIT UN : " ; B :
     PRINT
 80  IF A = B THEN PRINT "PARTIE NULLE" :
     GOTO 20
 90  IF A > B THEN PRINT "VOUS AVEZ GAGNE"
     ELSE PRINT "J'AI GAGNE !"
100  PRINT
110  GOTO 20
```

Le résultat de RND(1) est un nombre compris entre 0 et 1, mais toujours inférieur à 1. Tapez :

```
PRINT RND(1)
```

Ce qui est affiché est une nombre "aléatoire" en ce sens que vous ne pouvez prévoir quelle valeur sera affichée, mais cela n'est pas réellement aléatoire au sens mathématique du terme. Vous pouvez utiliser RND(1) pour générer un nombre aléatoire dans la gamme de variation que vous voulez.

Supposons que vous désiriez obtenir un nombre entier compris entre 1 et 6 (comme dans le jeu de dé), en utilisant RND(1), vous obtenez un nombre compris entre 0 et 1,

en le multipliant par 6 cela donne un nombre compris entre 0 et 6 (non compris).

En prenant la partie entière par la fonction INT, on obtient un nombre entier compris entre 0 et 6 (non compris). Il suffit de lui ajouter 1 pour obtenir un entier entre 1 et 6 (compris). Cela simulera le lancé d'un dé non pipé. L'Oric est honnête ! Cela vous permettra de réaliser tous les jeux à base de dés tels le Yam, le 421, le Jeu de l'Oie, le Monopoly et bien d'autres.

On peut toutefois faire mieux encore. Il est possible de tirer des lettres, des cartes à jouer, des couleurs, de générer une trajectoire au hasard.

Tapez :

```
PRINT CHR$(INT (RND(1) * 26) + 65)
```

qui imprime le caractère dont le code ASCII est tiré aléatoirement entre 65 et 90 ce qui correspond aux 26 lettres majuscules de l'alphabet.

Tapez :

```
PAPER (INT(RND(1) * 8))
```

et vous obtiendrez une couleur de fond au hasard. Le programme du Mastermind est un bon exemple de la manipulation des couleurs.

On peut tout rendre aléatoire, il suffit de trouver une correspondance entre ce que vous voulez rendre hasardeux et un nombre.

1.4.2 Utilisation des chaînes de caractères

Dans beaucoup de jeux de réflexion, l'ordinateur attend votre réponse sous la forme d'un mot ou d'une suite de caractères. C'est le cas du Jeu du Pendu ou des jeux utilisant des coordonnées pour se déplacer sur un damier ou un échiquier.

L'acquisition de votre réponse se fait par l'intermédiaire d'un INPUT ou d'un GET qui arrêtent le programme en attendant une entrée au clavier.

Exemple d'entrée de coordonnées :

```
10  INPUT "QUELLE POSITION (VER,HOR)" ; A$
20  X = ASC(LEFT$ (A$,1)) - 64
30  Y = VAL(RIGHT$ (A$,1))
```

Cette partie de programme entre les coordonnées d'une pièce sur un échiquier. La ligne est repérée par une lettre de A à H, et la colonne par un chiffre de 1 à 8.

A la ligne 10 on entre au clavier une suite de caractères dont le premier est une lettre et le dernier est un chiffre. Il peut y avoir d'autres caractères entre les deux. Il ne peut pas y avoir de virgule, toutefois.

La ligne 20 assigne à la variable numérique X, une valeur calculée à partir de la chaîne de caractères A$. En effet, par la fonction LEFT$(A$,1), on prend le premier caractère à gauche de A$, puis on évalue son code ASCII en décimal par la fonction ASC. On soustrait ensuite 64 de manière à se ramener à des valeurs comprises entre 1 et 8.

A la ligne 30, la fonction VAL donne la valeur numérique correspondant au caractère le plus à droite de A$ qui est un chiffre. Notez que la fonction VAL appliquée à un caractère autre qu'un chiffre donne la valeur 0, ce qui peut être utile pour repérer les erreurs de frappe et redemander alors les coordonnées.

Pour le Jeu du Pendu, il faut vérifier qu'une lettre se trouve bien dans le mot secret qu'a tiré l'ordinateur dans une table de vocabulaire.

Supposons que la chaîne de caractères B$ contienne le mot à découvrir, on peut effectuer le test de la manière suivante :

```
10  GET A$
20  FOR I = 1 TO LEN(B$)
30  IF MID$(B$, I, 1) = A$ THEN PRINT "IL Y A
    UN" ; A$ ; "EN" ; I
40  NEXT I
```

A la ligne 20, la fonction LEN permet de connaître le nombre de caractères du mot secret. A la ligne 30, la fonction MID$ donne le caractère situé à la nième position dans le mot secret et le compare avec la lettre rentrée en 10.

Dans tous les cas d'entrée de données au clavier, il est essentiel que vous protégiez votre programme contre les erreurs de frappe, car la poursuite d'un jeu avec des données fausses peut être catastrophique.

Pour cela testez systématiquement les caractéristiques des valeurs ou caractères entrés pour filtrer les valeurs non licites.

Exemple :

```
10  GET A$
20  IF A$ < "1" OR A$ > "9" THEN GOTO 10
30  A = VAL(A$)
```

A étant une variable numérique, on ne peut lui donner une valeur alphanumérique. Ce qui arriverait si l'on avait fait un GET A à la ligne 10 au lieu d'un GET A$ et si l'on avait appuyé sur une lettre au lieu d'un chiffre.

La ligne 20 vérifie si le caractère entré est bien un chiffre et ne conserve que ceux compris entre 0 et 9. La ligne 30 donne la valeur numérique correspondant au caractère entré en 10.

1.5 UTILISATION DU GRAPHIQUE DANS LES JEUX

1.5.1 Présentation d'un jeu

Cette partie d'un programme de jeu, bien qu'accessoire, améliore celui-ci. C'est une introduction qui doit présenter le titre et exposer rapidement les règles du jeu. On utilisera le mode Texte car il est mieux adapté pour l'édition de caractères alphabétiques. Il est facile d'imprimer une phrase dans un emplacement quelconque de l'écran grâce à la commande PLOT X, Y, Z$.

X et Y étant les coordonnées de la première lettre de la phrase, Z$ étant une chaîne de caractères définie auparavant ou une chaîne de caractères écrite entre guillemets.

Pour rendre la présentation plus attrayante, vous pouvez colorer les différents messages. Cela peut se faire facilement grâce à la possibilité de concaténation des chaînes de caractères, c'est-à-dire d'accoler bout-à-bout plusieurs chaînes entre elles.

Voyons ce que cela permet.

Exemple :

```
10  A$ = CHR$(4) + "Règle" + CHR$(2) +
    "du" + CHR$(1) + "Jeu"
20  PLOT 2, 13, A$
```

imprimera le message "Règle du Jeu" en bleu, vert et rouge au milieu de l'écran.

La chaîne CHR$(4) + place l'attribut encre bleue juste devant le mot règle. Les autres attributs utilisables sont donnés en annexe.

Vous pouvez décorer cette présentation avec les caractères semi-graphiques ou les caractères spéciaux utilisés dans le jeu.

Avant le début de chaque jeu, et de chaque changement de décor, il faut effectuer un effacement de l'écran. Pour cela utilisez CLS en mode basse résolution ou HIRES en haute résolution. Cependant, dans ce cas, la couleur de l'encre et celle du papier devient noire et blanche à chaque fois. Pensez donc à remettre les couleurs désirées par un PAPER et un INK.

Si vous utilisez les différents modes graphiques dans un jeu, la définition des couleurs doit se faire à l'intérieur de chaque mode, indépendamment.

1.5.2 Organisation de la mémoire d'écran

Votre écran de télévision doit être considéré comme une page sur laquelle vous allez dessiner successivement l'évolution du jeu. Chaque point visualisé sur l'écran est défini dans la mémoire vive de l'Oric.

Les caractères et les couleurs sont mémorisés dans une zone bien définie que l'on peut adresser directement.

Les commandes PRINT, PLOT en basse résolution et CURSET, DRAW, CIRCLE et FILL en haute résolution évitent de se préoccuper de la relation écran-mémoire et permettent de travailler avec des coordonnées verticales et horizontales.

— Mode basse résolution

En mode Texte, Lores 0 et 1, la page écran est divisée en 28 lignes et 40 colonnes, mais seules les lignes 0 à 26 et les colonnes 1 à 38 sont utilisables avec les commandes graphiques usuelles. En effet, la ligne en haut de l'écran est réservée aux messages apparaissant lors de l'enregistrement ou de la lecture de programme avec un magnétophone à cassette. Elle sert aussi à indiquer le mode majuscule pour le clavier par le message CAPS apparaissant en haut à gauche. Pour supprimer ce message, disgracieux dans vos décors de jeux, faites :

```
PRINT CHR$(20)
```

ce qui équivaut à presser [CONTROL] [T] qui inhibe le mode majuscule. Pour le faire réapparaître, faites une nouvelle fois cette commande.

Le dessin du curseur clignotant peut être aussi supprimé par [CONTROL] [Q] ou PRINT CHR$(17).

Les deux premières colonnes servent à placer l'attribut de couleur de papier et celui de l'encre. On ne peut écrire dans la première mais un PLOT 1, 10, 3 changera la couleur d'encre de la 10ème ligne en vert. Attention donc dans les jeux à ne pas faire se déplacer vos mobiles avec une coordonnée horizontale inférieure ou égale à 1. Une coordonnée dépassant le cadre de l'écran est signalée par le message "ILLEGAL QUANTITY ERROR".

Les coordonnées horizontales et verticales repèrent une portion de l'écran qui couvre une grille de 6 sur 8 points élémentaires de l'écran (appelés pixels). Un caractère est formé de 48 pixels. Ce bloc de 48 points demande une seule case mémoire pour être mémorisé en basse résolution. La mémoire d'écran en basse résolution est donc de 28 x 40 = 1 120 cases mémoire ou octets (appelés ainsi car ils contiennent 8 informations binaires élémentaires valant 0 ou 1).

Chacune de ces cases peut être remplie par la valeur que vous désirez grâce à la commande POKE, par exemple POKE 48162,65 met la valeur 65 dans la 48 162ème mémoire de l'Oric qui peut en adresser 65 536 en tout.

L'effet produit est d'afficher la lettre A (de code ASCII 65) à la 3ème colonne de la 2ème ligne de l'écran.

On peut donc faire apparaître sur toute partie de l'écran un caractère ou un attribut graphique si l'on connaît l'adresse de la mémoire correspondant à celle-ci.

Pour déterminer cette adresse, utilisez la formule suivante :

adresse en basse résolution =
48040 + (40 x Y) + 2 + X

Y étant la coordonnée verticale, X la coordonnée horizontale. La colonne couleur papier est trouvée par X = -1, la colonne encre par X = 0. La ligne CAPS par Y = -1.

Pour colorier et encrer les différentes zones de l'écran, déterminez les cases mémoire où inscrire un attribut. N'oubliez pas qu'un attribut affecte les caractères placés à droite de celui-ci jusqu'à ce qu'il trouve un autre attribut de même type ou la fin de la ligne.

Exemple de tracé de décors :

```
10  PAPER 6 :
20  READ J, K
30  FOR I = 1 TO 5
40  POKE J, K
50  NEXT I
60  DATA 48880,1
```

donnera un ciel bleu cyan sur un sol rouge.

La différence entre le mode Texte et le mode Lores est que la mémoire est initialisée dans le premier cas avec la première colonne correspondant au papier blanc, la 2ème colonne à l'encre noire et les colonnes suivantes au caractère vide (code 32), alors que dans le deuxième cas la première colonne contient l'attribut papier 0, la deuxième encre blanche, et les autres colonnes l'attribut

papier noir ce qui fait que les attributs de couleur placés n'affectent que les caractères ajoutés juste après, et non l'ensemble de la ligne. Pour colorer en Lores, il faut donc mettre un attribut dans la case précédant la zone à colorier et la remplir de caractères vides pour que l'attribut puisse se propager sur la zone. Utilisez comme précédemment une boucle FOR NEXT et un POKE. Notez que cela demande un certain temps pour être exécuté.

Pour effacer un caractère sur l'écran, on peut utiliser PLOT X, Y, " " ou PLOT X, Y, 32 ou POKE W, 32 (W étant l'adresse correspondant aux coordonnées X et Y).

Le caractère "pavé plein" a comme code 127 et le damier 126.

L'ensemble des caractères peut être imprimé en couleur inverse. Il suffit pour cela d'ajouter 128 au code ASCII du caractère considéré.

Exemple :

```
        PLOT 10, 10, 193
ou      PRINT CHR$(193)
```

imprime un A en inverse.

Cela est très utile pour avoir des objets en déplacement de couleurs différentes. C'est en effet plus pratique que d'utiliser un attribut placé devant le caractère pour modifier sa couleur, car il faut ajouter un autre attribut derrière pour rétablir la couleur sur le reste de la ligne. Ce qui demande en tout trois cases mémoire et surtout trois emplacements sur l'écran pour un seul caractère visualisé. De plus, il ne faut pas écrire sur un attribut car cela modifierait son action, ce qui est très gênant pour simuler les collisions.

Les mobiles doivent donc être imprimés en couleur d'encre normale ou inverse. Les décors peuvent, eux, être colorés par attribut s'ils ne se trouvent pas sur la trajectoire des mobiles. Cela vous amènera donc à dessiner votre terrain de jeu avec des zones de mouvements et des zones de décors et de messages (score par exemple).

La plupart des jeux peuvent être réalisés en basse résolution mais pour obtenir des graphiques plus fins, le mode haute résolution offre de plus grandes possibilités, bien que plus délicat à utiliser surtout pour imprimer des textes.

— Mode haute résolution

Il doit son nom au fait que l'on peut modifier, pixel par pixel, le contenu de l'écran. On obtient ainsi une définition de 240 sur 200 points, plus 3 lignes de texte en bas de l'écran. Comme en basse résolution l'écran est mémorisé en mémoire mais cela demande beaucoup plus de place car il faut une case mémoire pour un groupe de 6 pixels seulement au lieu de 48 en basse résolution.

Il y a donc (200 x 240) + 3 x 40 = 8 120 cases mémoire. La correspondance entre adresse de la mémoire et coordonnée se fait selon la formule :

40960 + (Y x 40) + X

Y étant la coordonnée verticale (de 0 à 199).

X étant le numéro de la colonne de 6 points considérée (de 0 à 39).

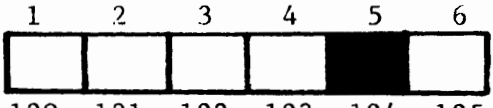
La mémoire s'étend donc de 40 960 à 49 119.

Exemple d'utilisation de la formule :

La case mémoire où est stocké le point de coordonnées 125, 120 est en :

124 = 6 x 20 + 4 donc X = 20 Y = 120

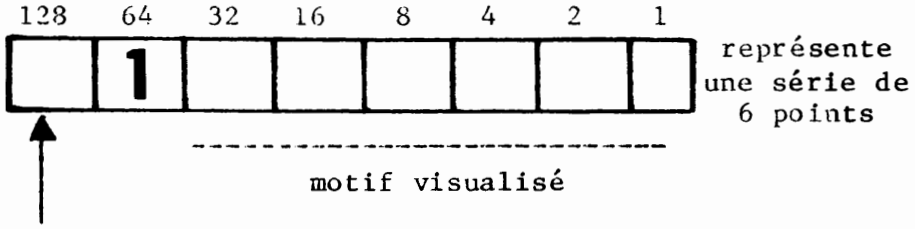
L'adresse cherchée est en 40960 + (120 x 40) + 20 = 45780



120 121 122 123 124 125

mémorisé en un octet en 45780

Une case mémoire peut contenir l'information pour 6 points ou un attribut.



si à 1 en inverse

1 = encre
0 = papier

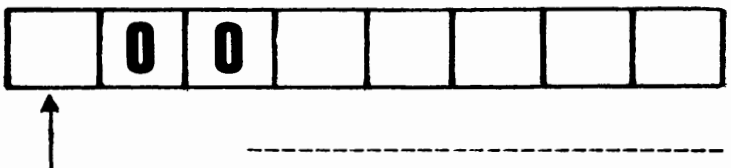
Calcul de la valeur à mettre dans la mémoire pour obtenir le motif désiré : somme des nombres au-dessus de la case si la case est à 1.

Exemple :

en couleur normale donne :
64 + 32 + 8 + 2 = 106

Si l'on fait POKE 45780,106 on obtient le motif à partir de la coordonnée 120, 120.

Codage d'un attribut en haute résolution :



inverse si 1

code binaire de l'attribut

On voit que l'on retrouve les mêmes codes qu'en basse résolution pour les attributs.

Le programme suivant vous montrera la définition en couleur maximale que l'on peut obtenir.

```
10  HIRES
20  K = 16
30  FOR I = 40960 TO 49120
40  K = K + 1
50  POKE I, K
60  IF K = 23 THEN K = 16
70  NEXT I
```

Chaque case de la mémoire d'écran est remplie avec un attribut de couleur de fond qui change à chaque case. Il en résulte, sur l'écran, une multitude de petits segments de couleurs différentes.

Pour imprimer un caractère en mode haute résolution, et donc pour faire mouvoir les mobiles d'un jeu, il faut utiliser la commande CHAR N, M, FB qui imprime sur l'écran à partir de la position précédente du curseur, le caractère de code ASCII N, choisi dans le jeu normal (M = 0) ou alterné (M = 1), avec la couleur d'encre (FB = 1), avec celle du fond (FB = 0), en inverse (FB = 2) ou n'imprime rien (FB = 3). Cette dernière possibilité peut être intéressante pour faire une impression conditionnée par la valeur FB. La position du curseur n'est pas modifiée par cette commande. On voit que l'on ne peut afficher qu'un seul caractère à la fois.

Pour des mobiles plus grands (composés de plusieurs caractères), il faut utiliser une boucle FOR NEXT.

Exemple d'impression de texte en haute résolution :

```
10  A$ = "SCORE :": N = 6 : X = 10 : Y = 100
20  GOSUB 30
30  FOR I = 1 TO N
40  CURSET X + I * 8, Y, 0
50  CHAR ASC (MID$(A$, I, 1)), 0, 1
60  NEXT I
```

En ligne 40, il faut découper le mot à afficher en tranches d'un caractère.

Pour chaque message ou mobile composé, faites exécuter un sous-programme semblable à celui-ci.

N est la longueur du mot.

X et Y les coordonnées de la première lettre.

Pour effacer un caractère, il faut imprimer le même caractère avec FB = 0 donc en couleur de fond ou imprimer le caractère plein (code 127) en FB = 0.

1.5.3 Les caractères graphiques sur ORIC

Tout jeu visuel demande l'utilisation de caractères. L'Oric a des possibilités très variées dans la forme des caractères utilisables.

A la mise en route, l'Oric charge en mémoire vive deux jeux de caractères. Le jeu standard contient les caractères alphabétiques, les chiffres et la ponctuation. Le jeu "alterné" contient, lui, les caractères graphiques basse résolution utilisés pour le réseau télématique Prestel en Grande-Bretagne. Ils conviennent très bien pour délimiter les limites d'un terrain de jeu, car ils sont formés d'une mosaïque de rectangles élémentaires.

— Codage des caractères PRESTEL

Le motif de base de ces caractères est illustré à la figure A. Il comprend six rectangles formant un grand rectangle de la taille d'un caractère normal. Chacun des sous-rectangles peut être noirci selon le code du caractère.

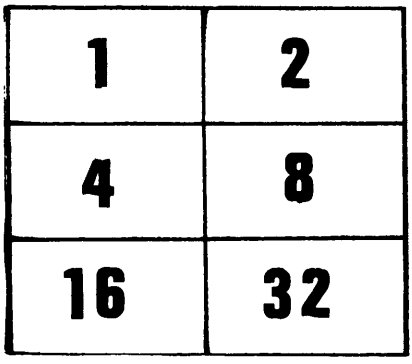


Fig. A : Motif de base d'un caractère Prestel

Pour trouver ce code, tracez le caractère désiré sur le motif et faites la somme des nombres contenus dans les sous-rectangles noircis.

Exemple :

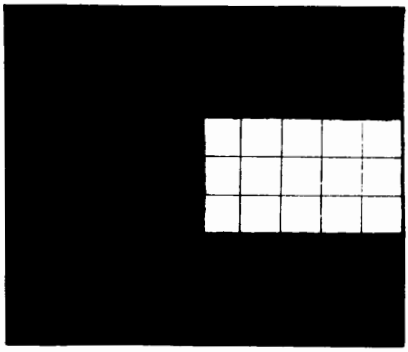


Fig. B

1 + 2 + 4 + 16 + 32 = 55

Pour obtenir le code, il faut ajouter 32 à ce nombre pour avoir le code ASCII utilisable.

Pour faire apparaître le caractère correspondant, pressez [ESC] [I] (ce qui met en place le mode semi-graphique pour les caractères suivants de la ligne) puis pressez la touche dont le code ASCII vaut le code que l'on vient de calculer. Ici on a 55 + 32 = 87 qui est le code ASCII de la lettre "W". Pressez [W] et il apparaît le caractère dessiné à la Fig. B.

Essayez, pour d'autres codes, de trouver la touche correspondante. Si au cours d'un programme, vous désirez faire apparaître ces caractères spéciaux, vous pouvez faire comme ceci :

```
PRINT CHR$(27) + "IW"
```

Cela fera apparaître le caractère de Fig. B.

27 est le code ASCII de la touche [ESC].

I peut être remplacé par d'autres attributs selon que l'on veut faire clignoter, changer l'encre ou le papier sur le reste de la ligne. On doit faire un CHR$(27) pour chaque attribut. Les caractères qui suivent I sont pris dans le jeu "alterné".

— Caractères personnalisés

Malgré le double jeu de caractères de l'Oric, cela peut vous sembler insuffisant et peut-être désirez-vous utiliser d'autres caractères spéciaux. Si vous désirez un trèfle, un carreau, un pique et un coeur pour visualiser un jeu de cartes, c'est possible.

En effet, il est possible de redéfinir tous les caractères de l'Oric car ils sont définis dans la mémoire vive que vous pouvez donc modifier par programme.

C'est très utile pour créer vos personnages et vos mobiles dans les jeux. Vous obtiendrez ainsi de très beaux graphismes et donnerez une touche professionnelle à vos jeux.

Pour redéfinir un caractère, commencez par dessiner une grille de 6 sur 8 cases. Puis dessinez à l'intérieur, le motif que vous désirez. Noircissez les cases où passe un trait.

Après quelques essais, vous obtiendrez un motif correspondant à la forme désirée.

Exemple : définition d'un monstre.

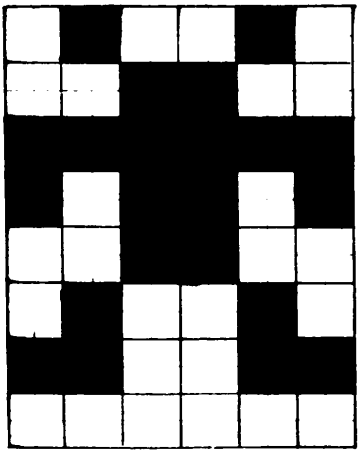


Fig. C : Dessin du motif du caractère à définir

Chaque caractère demande 8 places en mémoire correspondant aux 8 lignes de la grille.

Pour trouver le contenu de ces mémoires, faites la somme des nombres contenus dans les cases noircies pour chaque ligne.

32	16	8	4	2	1

Pour notre exemple cela donne :

ligne 1 : 16 + 2 = 18
ligne 2 : 8 + 4 = 12
ligne 3 : 32 + 16 + 8 + 4 + 2 + 1 = 63
ligne 4 : 32 + 8 + 4 + 1 = 45
ligne 5 : 8 + 4 = 12
ligne 6 : 16 + 2 = 8
ligne 7 : 32 + 16 + 2 + 1 = 51
ligne 8 : 0

Il reste maintenant à déterminer à quelle adresse de la mémoire il faut ranger ces valeurs. Dans la mémoire vive de l'Oric, une zone est réservée aux définitions des caractères. Cette zone commence en 46 080 et s'arrête en 47 999. les caractères sont rangés dans l'ordre croissant de leur code ASCII par suites de 8 nombres (dans l'ordre, ligne 1 à 8). Les caractères standard étant définis avant les caractères "alternés" (voir table de correspondance entre code ASCII et emplacement en mémoire dans l'annexe). Pour modifier un caractère (par exemple £, ©, ↑, qui sont peu usités et qui ont comme code ASCII 94, 95, 96), faites une série de POKE dans les adresses données en annexe (46080 + (code ASCII du caractère x 8) + numéro de la ligne concernée pour un caractère standard et 47104 + (code ASCII du caractère x 8) + numéro de la ligne pour un caractère "alterné").

Exemple de sous-programme type effectuant le changement de caractère :

```
 5  REM * DEFINITION DE CARACTERE
10  READ J
20  FOR I = 0 TO 7
30  READ K
40  POKE J + I, K
50  NEXT I
60  DATA 46600, 18, 12, 63, 45, 12, 18, 51, 0
70  RETURN
```

Ce sous-programme peut être utilisé pour d'autres caractères en modifiant la première valeur du DATA.

Vous trouverez de nombreux exemples de caractères ainsi définis dans le jeu des envahisseurs de l'espace.

Pour vous aider dans la définition de caractères spéciaux, voici un programme très utile et pratique permettant de dessiner un motif de caractère et d'obtenir ensuite les valeurs à mettre dans le DATA du sous-programme DEFINITION DE CARACTERE.

Pour l'utiliser, pressez les touches [G] pour gommer la prochaine case, [espace] pour noircir la prochaine case, les flèches pour déterminer la position de la prochaine case.

Le dessin du caractère apparaît en grand sur l'écran.

Modifiez-le jusqu'à ce que vous soyez satisfait de la forme obtenue puis tapez [F] pour obtenir les DATA correspondant au dessin.

Le programme vous demandera quel caractère doit être modifié. Vous pouvez taper la touche correspondant au caractère ou bien entrer le code ASCII pour les caractères qui ne sont pas sur le clavier. Faites RETURN ensuite. Attention pour le code ASCII, veillez à ne rentrer que les chiffres correspondants du code sinon les valeurs obtenues ne seraient pas bonnes. De toute façon le programme vous donne le code avec lequel les calculs ont été faits. Choisissez ensuite le jeu standard ou alterné et vous obtiendrez les résultats.

Notez que les valeurs des adresses données ici, correspondent au mode TEXTE et qu'elles ne sont pas les mêmes en haute résolution. Cela n'a pas d'importance car si vous définissez vos caractères avant de faire un HIRES les mémoires seront automatiquement modifiées lors du passage en haute résolution.

```
10  REM * AIDE A LA DEFINITION DES
    CARACTERES SPECIAUX *
20  X = 16 : Y = 10 : CLS
30  PLOT X, Y, 64
40  GET K$
```

```
 50  IF K$ = CHR$(32) THEN A$ = CHR$(127)
 60  IF K$ = "G" THEN A$ = CHR$(32)
 70  IF K$ = "F" THEN GOTO 120
 80  IF K$ < CHR$(8) OR K$ > CHR$(11) THEN
     GOTO 40
 90  ON ASC (K$) - 7 GOSUB 320, 330, 340, 350
100  PLOT X, Y, A$
110  GOTO 40
120  PLOT 2, 20,"QUEL CARACTERE VOULEZ VOUS
     DEFINIR ? "
130  INPUT B$
140  IF LEN (B$) > 1 THEN B = VAL(B$) : GOTO
     160
150  B = ASC (B$)
160  PLOT 2, 22, CODE ASCII : +STR$(B)
170  INPUT "STANDARD(S) OU SEMI-GRAPHIQUE(A)"
     ; C$
180  IF C$ = "S" THEN N = 46080 : GOTO 210
190  IF C$ = "A" THEN N = 47104 : GOTO 210
200  GOTO 160
210  X = 16 ; Y = 10
220  FOR I = 0 TO 5
230  FOR J = 0 TO 7
240  IF SCRN(X + 1, Y + J) = 127 THEN D(J) =
     D(J) + 2"(5 - I)
260  NEXT J, I
270  PLOT 2, 23, "VOICI L'ADRESSE DE DEBUT ET
     LES DATA"
280  PLOT 1, 24, STR$(N + B * 8) : FOR I = 0
     TO 7 : PLOT 8 + I * 4, 24, STR$(D(I)) :
     NEXT I
290  INPUT "VOULEZ VOUS DEFINIR UN AUTRE
     CARACTERE ; C$"
300  IF C$ = "O" THEN RUN
310  END
320  IF X - 1 > 15 THEN X = X - 1
325  RETURN
330  IF X + 1 < 22 THEN X = X + 1
335  RETURN
340  IF Y + 1 < 18 THEN Y = Y + 1
345  RETURN
350  IF Y - 1 > 7 THEN Y = Y - 1
355  RETURN
```

Pour retrouver les caractères normaux, vous pouvez appuyer sur le bouton de Reset qui se trouve sous l'appareil ou faire CALL # 555 qui effectue la même opération.

Si vous trouvez que les caractères sont trop petits, vous pouvez en accoler plusieurs pour former un plus grand motif. Définissez alors chaque caractère selon le motif qui est recouvert par la taille de ce sous-caractère. Un bel exemple de ce procédé est visible dans le jeu Cavalier.

1.5.4 Terrain de jeu et déplacement

Après avoir choisi le mode de définition qui vous semblera le mieux adapté au jeu, dessinez sur une grille (voir annexe) représentant l'écran, les décors et la zone d'évolution des mobiles. Fixez l'emplacement des messages (score, temps limite, instructions, etc). Si vous voulez avoir plusieurs couleurs sur l'écran, prenez garde à laisser un espace vide devant chaque début de zone à colorer, cela pour pouvoir y placer l'attribut assurant le coloriage. Etudiez bien les limites du terrain de jeu. Grâce à la grille vous déterminerez les coordonnées de chaque partie de l'écran ainsi que l'emplacement en mémoire de celle-ci.

Pour planter le décor, utilisez une série de POKE pour placer les attributs de couleurs ou de choix de jeu de caractères.

L'ordre dans lequel vous l'effectuez doit être étudié car cela peut donner un curieux effet.

Exemple :

```
10  PAPER 2
20  N = 48452
30  FOR I = 1 TO 10
40  POKE N + I * 40, 17
50  POKE N + 5+I * 40, 18
60  NEXT I
```

Ce programme trace un rectangle de 5 sur 10 caractères en rouge sur fond vert.

Tel qu'il est écrit, vous constaterez à l'exécution qu'il fait apparaître des bandes rouges fugitives sur l'écran. Il est facile de l'éviter, en permutant l'ordre

des lignes 40 et 50. En effet, la ligne 40 met un attribut rouge qui affecte donc toute la ligne après lui. La ligne 50 met un attribut vert qui rétablit la couleur du fond. La présence de cet attribut vert arrête l'effet de l'attribut rouge. Donc, si l'on place l'attribut vert auparavant, l'attribut rouge n'agira que sur 5 caractères.

En prêtant attention à ces détails, vous obtiendrez un décor qui s'établira sans traînée sur l'écran, ce qui est tout de même plus satisfaisant esthétiquement.

1.6 UTILISATION DES POSSIBILITÉS SONORES

Avec le générateur de sons programmable incorporé à l'Oric, la création de bruitages devient un jeu en lui-même.

La plupart des jeux demandent une sonorisation pour renforcer le réalisme des décors ou pour marquer l'évolution de la partie en cours. Les tirs de projectiles sont accompagnés d'un SHOOT retentissant, ou d'un ZAP sidéral, les collisions ou les cibles touchées résonnent d'un EXPLODE, les différentes manches des matchs sont annoncées par des PING. Tout ces bruitages ne doivent pas être considérés comme décoratifs mais au contraire faire partie intégrante du jeu. Ils doivent donc être peu nombreux car leur cumul les rend inopérants et fatigants. Donc les sons ne seront utilisés que lors de modifications du déroulement du jeu pour signifier ces changements au joueur qui réagira à leur signal. L'effet de surprise est à rechercher, et il est bon d'avoir un bruit spécifique à chaque mobile. L'Oric permet la création de sons originaux et variés.

1.6.1 Sons prédéfinis

Avec les commandes PING, ZAP, EXPLODE, SHOOT, l'Oric couvre la grande majorité des bruitages pour jeux d'action.

Utilisez-les donc souvent car l'effet obtenu est garanti.

Ces sons ressemblent beaucoup à ceux des jeux du commerce. On peut créer de nouvelles possibilités avec ces sons prédéfinis en les répétant successivement de manière rapide.

Par exemple :

```
10  FOR I = 1 TO 10
20  PING
30  WAIT 10
40  NEXT I
```

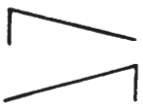
donne l'effet d'une cloche de voiture de pompiers d'autrefois ou signale l'immersion d'un sous-marin.

Essayez aussi avec ZAP.

1.6.2 Sons préfabriqués

Le générateur de sons possède trois canaux indépendants et un générateur de bruits qui peuvent être mélangés aux autres sons. La puissance du son (l'amplitude de la tension de sortie) peut être modulée par sept formes d'enveloppe dans l'instruction PLAY (TE, NE, EM, EP). EM valant

1 :
2 :



1 ou 2 donne un son de durée finie dont l'amplitude monte rapidement puis baisse lentement (1) et inversement (2). La durée du son dépend de la période d'enveloppe EP.

Pour EM valant de 3 à 7 le son est continu,

3 :
4 :
5 :
6 :
7 :

3 et 6 sont équivalents à 1 et 2 mais en se répétant.

La fréquence du son émis est programmée par un MUSIC ou un SOUND. La note ou la période étant définie par un nombre, elle peut être une variable BASIC et donc ainsi permettre la modulation de fréquence par variation de la période dans une série de SOUND répétés.

Exemple :

```
 10  REM * SIRENE AMERICAINE *
 20  N = 100
 30  DN = 1
 40  REPEAT
 50  REPEAT
 60  N = N + DN
 70  SOUND 1, N, 10
 80  UNTIL N = 120 OR N = 80
 90  DN = - DN
100  UNTIL KEY$ = CHR$(32)
110  SHOOT
120  END
```

Ce court programme crée une variable N qui croît de 80 à 120 puis décroît de 120 à 80 et ainsi de suite. N étant le paramètre période de la ligne 70, celle-ci est modulée et passe du grave à l'aigu puis au grave imitant ainsi le son des sirènes de voitures de police américaines.

Essayez d'autres valeurs de N ou de DN et du test ligne 80. La ligne 100 arrête le son en appuyant sur la barre d'espacement.

Beaucoup de bruits de moteurs peuvent être imités en utilisant le générateur de bruits.

Par exemple l'ambiance de la cabine d'une caravelle :

```
10  REM * CARAVELLE *
20  SOUND 4, 10, 0
30  PLAY 1, 1, 7, 0
```

La ligne 20 autorise le mélange canal 1 et bruit. La période de 10 donne un son aigu et perçant imitant les turbines des réacteurs. Le son est produit par la ligne 30. Le paramètre de période d'enveloppe est à 0 pour avoir le son à sa hauteur normale sans modulation d'amplitude. L'enveloppe 7 est choisie pour avoir un son continu.

Si l'on module l'amplitude du bruit, on peut obtenir le bruit d'une machine à vapeur.

```
10  REM * LOCOMOTIVE *
20  SOUND 4, 100, 0
30  PLAY 1, 1, 3, 200
```

A la ligne 30, une augmentation du dernier paramètre ralentit la cadence.

Pour signaler le rebond d'une balle sur les limites du terrain, il serait bon de faire entendre un "CLOP".

Faites :

```
CALL #FB03
```

et vous entendrez le son émis par une pression sur une touche du clavier mais sans toucher celui-ci. On exécute en fait un sous-programme de la ROM BASIC de l'Oric.

CHAPITRE **2**

Jeux de hasard

2.1 ALPHABET

A comme Ananas, B comme Banane, C comme Citron, ... L'alphabet est l'une des premières choses que l'on nous apprend lorsque nous arrivons sur les bancs des écoles communales. Mais il ne suffit pas d'apprendre. Il faut aussi retenir. Nous vous proposons dans ce jeu un test qui vous permettra d'exercer votre mémoire. Le principe est le suivant : une lettre apparaît sur l'écran, il faut alors taper la touche correspondante pour pouvoir continuer. Si la touche appuyée est la bonne, deux lettres sont alors affichées, la première étant identique à la précédente. La seconde étant nouvelle, vous devez alors taper dans l'ordre ces deux lettres. Si aucune erreur n'est commise, la suite des lettres réapparaît et une lettre y est rajoutée... Ce jeu peut être très long si vous avez une mémoire infaillible, ou très court si vous ne faites pas un effort de concentration suffisant.

Le programme que nous vous proposons fera marcher votre machine comme répétiteur. Celle-ci générera la suite de lettres et comptabilisera le nombre que vous aurez retenu. Les touches de clavier utilisées sont évidemment celles marquées par une lettre alphabétique.

REMARQUE : L'utilisation fréquente de ce programme vous apportera une meilleure connaissance de l'emplacement des touches sur le clavier.

Structure du programme :

Lignes 9 à 20, présentation.

Lignes 30 à 42, choix du niveau.

Lignes 110 à 310 boucle principale.

Lignes 300 à 350 sortie du programme.

Lignes 360 à 390, affichage nombre de lettres retenues.

```
9 REM  ************
10 REM * ALPHABET *
11 REM ************
12 DIM X(100),Y(100),C(100)
15 CLS
16 PRINTCHR$(17)
20 PRINT:PRINT:PRINTSPC(10)"ALPHABET"
30 PRINT:PRINT:PRINT"NIVEAUX DES DIFFICULTES"
35 PRINT"  1:FAIBLE"
36 PRINT"  2:MOYEN"
37 PRINT"  3:FORT"
38 INPUT"QUEL NIVEAU CHOISISSEZ VOUS";R$
40 IF ASC(R$)<49 OR ASC(R$)>51 THEN GOTO 38
42 NI=-VAL(R$)*50+200
110 CLS
120 K=0
130 REPEAT
140 K=K+1
150 X(K)=INT(38*RND(1))+1
160 Y(K)=INT(26*RND(1))+1
170 C(K)=INT(26*RND(1))+65
180 FOR I=1 TO K
190 PLOTX(I),Y(I),CHR$(C(I))
200 A=INT(7*RND(1))
210 INKA:PAPER7-A
220 WAITNI
230 CLS
235 ZAP
240 NEXT I
250 INK0
260 FOR I=1 TO K
270 GET R$
275 PING
280 IF R$<>CHR$(C(I)) THEN GOTO 360
```

```
290 IF K=100 THEN PRINT"BRAVO!! QUELLE MEMOIRE PHENOMENALE"
300 NEXT I
310 UNTIL K=100
320 PRINT:PRINT:INPUT"VOULEZ VOUS RECOMMENCER";R$
330 IF R$="O" THEN GOTO 30
340 PRINT"FIN"
350 END
360 IF R$<CHR$(65) OR R$>CHR$(90) THENI=I-1: GOTO 390
370 PRINT:PRINT"VOUS AVEZ RETENU ";K-1;" LETTRES":I=K
380 K=100
390 GOTO 300
```

2.2 JACK-POT

Votre porte-monnaie est bien rembourré ? Alors vous pouvez continuer la lecture de cette page, car nous entrons ici dans le domaine des machines à sous où la passion du jeu l'emporte sur la raison. Ce bandit manchot, comme il est appelé dans l'ouest américain, comporte trois fenêtres dans lesquelles peuvent apparaître les chiffres 1, 2 ou 3. Pour jouer, misez 5 francs en pressant la touche "O". Les combinaisons gagnantes sont celles qui font apparaître les trois as ou un trois. Le gain est alors supérieur à votre mise et est évalué aléatoirement.

L'activité lucrative des machines à sous étant interdite, l'Oric vous prêtera en début de partie un pécule de 100 F. A vous de le conserver si vous pouvez résister à l'attrait du gain. Comme toute bonne attraction d'argent, la machine est truquée et statistiquement elle finit toujours par vous faire dépenser au-delà de vos moyens.

Dès que vous lancez le programme, la machine tire une première série de chiffres. Après il vous faudra presser la touche "O" pour la relancer.

En route pour Las Vegas !

Structure du programme :

Les lignes 90 à 140 assurent le tirage de la série de chiffres par incrémentation aléatoire de trois compteurs L, M, N.

La ligne 150 teste si un des compteurs est à 3 et donc si le coup est gagnant.

La ligne 160 teste s'il y a les trois as.

Les lignes 170 à 190 décomptent vos économies et testent si la ruine survient.

Les lignes 200 à 230 augmentent vos économies et calculent vos gains.

Les lignes 240 à 300 relancent ou terminent la partie en cours.

Le sous-programme de 1000 à 1080 assure l'affichage du Jack-Pot.

Le sous-programme de 2000 à 2040 affiche les séries de chiffres dans les fenêtres du Jack-Pot.

```
5 REM  ************
10 REM * JACK POT *
15 REM ************
20 PAPER4:INK2:CLS
30 NC=0
40 TT=100
50 GOSUB 1000
55 WAIT 300
60 L=0
70 M=0
80 N=0
90 FOR I=1 TO 3
95 CALL#FB03
100 A=RND(1)*9+1
110 IF A=<3 THEN L=L+1:GOTO140
120 IF A=<5 THEN M=M+1:GOTO140
130 N=N+1
140 NEXT I
150 IF L=3 OR M=3 OR N=3 THEN GOTO 200
160 IF L=1 AND M=1 AND N=1 THEN GOTO 200
170 PLOT 5,20,"Helas,vous avez Perdu 5 francs!"
180 TT=TT-5
190 IF TT<1 THEN GOTO 270 ELSE GOTO 220
200 PLOT 5,20,"Vous etes chanceux,felicitation"
210 TT=TT+5+INT(RND(1)*5)
220 PLOT 2,22,"Vous avez encore"+STR$(TT)+"francs en Poche  "
230 NC=NC+1
235 GOSUB2000
240 PLOT2,24,"Voulez vous rejouer(O/N)?"
245 GET A$
250 IF A$="O" THEN GOTO 60
260 END
270 PLOT 1,22,"Vous avez tout dilaPide en"+STR$(NC)+"coups"
275 SHOOT
280 NC=0
290 TT=100
300 GOTO240
1000 PLOT10,5,"**************"
```

```
1010 FOR I=1 TO 9
1020 PLOT10,5+I,"*   *   *   *
1030 NEXT I
1040 PLOT10,15,"**********"
1050 PLOT0,3,1
1060 PLOT3,3,12
1070 PLOT12,3,"JACK-POT"
1080 RETURN
2000 PLOT11,10,STR$(L)
2010 PLOT15,10,STR$(M)
2020 PLOT19,10,STR$(N)
2030 PING
2040 RETURN
```

2.3 SIMON

Bleu, jaune, rouge, magenta... non ce n'est pas un arc-en-ciel qui va apparaître devant vos yeux, mais tout simplement une suite de couleurs qu'il vous faudra mémoriser. Combien de ces couleurs pourrez-vous retenir ? Pour le savoir, il vous suffit de mettre en route le programme. A chaque fois, un nombre croissant de bandes colorées apparaîtront sur l'écran. Il vous suffira alors de rentrer, dans l'ordre d'apparition, le code de la couleur enregistrée.

Les codes utilisés sont :

Rouge : 1	Bleu : 4
Vert : 2	Magenta : 5
Jaune : 3	Cyan : 6

Le jeu comporte trois niveaux, la cadence d'apparition croît avec le niveau. De plus chaque couleur est agrémentée d'un son, ce qui vous facilitera la tâche. Dans ce dernier but, chaque bande colorée apparaîtra avec son code sauf au niveau 3. Un signal sonore "PING" vous indiquera le moment où il vous faudra rentrer les codes. Si l'une des touches autre que 1 à 6 est enfoncée, la machine vous préviendra grâce à un son "ZAP", il vous faut alors continuer à rentrer les codes.

Attention : si la suite de codes rentrés est bonne, la machine recommence de suite une autre séquence, alors pas de moment d'inattention ... A vous de jouer.

REMARQUE : Contrairement au "Simon" classique la séquence est renouvelée à chaque fois, ce qui rend le jeu bien plus difficile.

Pour sortir du programme, appuyez sur un mauvais code.

Structure du programme :

Lignes 9 à 30, initialisations.

Lignes 40 à 60, choix du niveau.

Lignes 110 à 300, boucle principale.

Lignes 310 à 340, sortie, fin.

Lignes 400 à 430, validation code de couleur.

```
 9 REM  *********
 10 REM * SIMON *
 11 REM *********
 12 DIM C(100)
 16 CLS :PAPER0 :INK2
 20 PRINTCHR$(17)
 30 PRINT:PRINT:PRINTSPC(10)"SIMON"
 40 PRINT:PRINT:PRINT"NIVEAUX DES DIFFICULTES"
 42 PRINT"  1:FAIBLE"
 44 PRINT"  2:MOYEN"
 46 PRINT"  3:FORT"
 48 PRINT:INPUT"QUEL NIVEAU CHOISISSEZ VOUS";R$
 50 IF R$<CHR$(49) OR R$>CHR$(51) THEN GOTO 48
 60 NI=-VAL(R$)*50+200
 110 INK0
 120 K=0
 130 REPEAT
 135 CLS
 140 K=K+1
 150 FOR I=1 TO K
 160 C(I)=INT(7*RND(1))+1
 165 PRINTCHR$(27)+CHR$(C(I)+80);
 166 IF NI>50 THEN PRINTCHR$(C(I)+48);
 167 SOUND1,15*C(I),5
 168 WAIT NI
 169 PRINT :PRINT
 170 NEXT I
 180 PING
 190 CLS
 200 FOR I=1 TO K
 210 GETR$
 220 IF R$<>CHR$(C(I)+48) THEN GOTO 400
 240 PRINTCHR$(27)+CHR$(C(I)+80)
 250 PRINT
```

```
260 SOUND1,15*C(I),5
270 WAIT 50
280 IF K=100 THEN PRINT"PIC DE LA MIRANDOLE,JE VOUS AI RECONNU"
290 NEXT I
300 UNTIL K=100
305 INK3
310 INPUT"VOULEZ VOUS RECOMMENCER (O/N)";R$
320 IF R$="O" THEN GOTO 48
330 PRINT"FIN"
340 END
400 IF R$<CHR$(49) OR R$>CHR$(55) THENZAP:I=I-1:GOTO 430
405 INK3
410 PRINT:PRINT"VOUS AVEZ RETENU ";K-1;" BARETTES COLOREES"
415 ZAP
420 I=K :K=100
430 GOTO 290
```

CHAPITRE 3

Jeux d'adresse

3.1 MUR

Un mur de briques multicolores se dresse devant vous. A l'aide de trois balles et de deux raquettes, vous devez le détruire brique par brique. Profitez des rebonds.

Si la balle dépasse la ligne des raquettes, elle est perdue. Essayez de totaliser un maximum de briques détruites. Si vous réussissez à détruire totalement un mur, un second se reconstruit aussitôt pour que vous puissiez continuer la partie.

La raquette est déplacée par les touches fléchées droite et gauche.

Structure du programme :

Lignes 20 à 40, initialisations des variables.

Lignes 50 à 90, tracé des limites du jeu.

Lignes 100 à 230, tracé du mur et des messages.

Lignes 300 à 470, double boucle imbriquée gérant le comptage des balles et le déroulement de la partie.

Lignes 340 à 350, testent s'il y a une demande de déplacement.

Ligne 360 teste si la balle arrive sur le mur.

Lignes 370 à 380, testent si la balle arrive sur les limites du terrain ou sur la raquette.

Lignes 390 à 430, déplacement de la balle, testent si la balle sort du terrain.

Lignes 1000 à 1040 et 3000, sous-programme redéfinissant le caractère de la balle.

Lignes 1500 à 1550, comptabilisent les briques détruites et les effacent de l'écran.

Ligne 2000, sous-programme qui assure le déplacement de la raquette.

```
14 REM ******************
15 REM * MUR DE BRIQUE *
16 REM ******************
17 PAPER0:INK6:CLS
20 GOSUB 1000
30 B=3:TP=0:R$=CHR$(254)+CHR$(254)+"      "+CHR$(254)+CHR$(254)+" "
35 S$=" "+CHR$(254)+CHR$(254)+"      "+CHR$(254)+CHR$(254)
40 XR=17:YR=22:XB=20
50 REM * Trace du jeu *
60 FOR I=1 TO 10
70 READ E,F
80 POKE E,F
90 NEXT I
100 REM *Trace du mur *
110 FOR I=4 TO 32:PLOT I,3,254:NEXT I
120 FOR I=4 TO 22:PLOT 4,I,254:PLOT32,I,254:NEXT I
130 FOR I=5 TO 9:PLOTI,22,254:NEXT I
140 FOR I=27 TO 32:PLOTI,22,254:NEXT I
150 FOR I=48286 TO 48446 STEP 40
160 FOR J=I TO I+26
170 POKEJ,127
180 NEXTJ,I
190 NB=135
200 PLOT21,1,"NB DE BALLES:"+STR$(B)
210 PLOT3,1,"NB DE BRIQUES:"+STR$(BQ)
220 PLOT1,24,"TEMPS:"+STR$(TP)
230 PLOTXR,YR,R$
290 REM *Boucles Principales *
300 REPEAT
310 YB=18:DX=1:DY=-1
320 REPEAT
330 TP=TP+1:PLOT7,24,STR$(TP)
340 K$=KEY$
350 IF K$=CHR$(8) OR K$=CHR$(9) THEN GOSUB 2000:GOTO340
```

```
360 IF SCRN(XB+DX,YB+DY)=127 THEN GOTO 1500
370 IF SCRN(XB+DX,YB+DY)=254 AND(YB+DY=30ORYB+DY=22)THENDY=-DY:CALL# FB24
FB03
380 IFSCRN(XB+DX,YB+DY)=254THENDX=-DX:CALL#FB03   FB14
390 PLOTXB,YB," "
400 XB=XB+DX
410 YB=YB+DY
420 PLOTXB,YB,"C"
430 UNTIL YB>25
440 B=B-1:EXPLODE
450 PLOTXB,YB," "
460 PLOT34,1,STR$(B)
470 UNTIL B=0
480 PLOT4,25,"VOULEZ VOUS REJOUER?(O/N)"
490 GET A$
500 IF A$="O" THEN RUN
510 IF A$="N" THEN END
520 GOTO490
990 REM * Def caracteres *
1000 FOR I=46808 TO 46815
1010 READ J
1020 POKEI,J
1030 NEXT I
1040 RETURN
1490 REM * Brique detruite *
1500 PLOTXB+DX,YB+DY," "
1510 DY=-DY:PING
1520 B0=B0+1
1530 PLOT17,1,STR$(B0)
1540 NB=NB-1
1550 IF NB=0 THEN GOTO150 ELSE GOTO370
1990 REM * Depl raquette *
2000 ON ASC(K$)-7 GOTO 2010,2050
2010 IF XR<11 THEN GOTO 2040
2020 XR=XR-1
2030 PLOTXR,YR,R$
2040 RETURN
2050 IF XR>17 THEN GOTO 2080
2060 XR=XR+1
2070 PLOTXR-1,YR,S$
2080 RETURN
3000 DATA 0,12,30,63,63,30,12,0
3010 DATA 48081,2,48281,1,48321,2,48361,3,48401,5,48441,7
3020 DATA49000,21,49040,21,49080,21,48960,21
```

3.2 BARON

Ce jeu retrace un des périples les plus connus de la vie du baron légendaire. Au cours d'une de ses innombrables batailles, il a chevauché un boulet de canon durant le trajet qui le séparait des lignes ennemies. Vous devez aider le baron à réaliser cet exploit hors du commun. Vous êtes canonnier et vous vous tenez prêt derrière votre pièce d'artillerie.

Par votre grande expérience de la balistique, le baron vous a choisi pour orienter le tir. Il faut que le baron se pose sur la façade de la forteresse des opposants. S'il atterrissait à l'intérieur de celle-ci, il risquerait d'être pris par vos adversaires et sa mission échouerait.

Pour régler la trajectoire de l'obus et de son passager de fortune, vous devez choisir la charge de poudre à mettre dans le fût ainsi que la hausse du tir. Celle-ci doit être un nombre entier positif et inférieur à 89. Elle représente l'angle du fût par rapport au sol.

Fort heureusement, le baron a plus d'un tour dans son sac, et même s'il ne réussit pas à atteindre la muraille, il peut revenir au canon pour essayer de nouveau.

La taille de la forteresse ainsi que sa position varient aléatoirement au début de chaque partie.

Structure du programme :

Les lignes 100 à 110 fixent les dimensions et la position de la forteresse.

Les lignes 120 à 150 assurent le choix de la charge de poudre.

Les lignes 160 à 200 permettent de fixer la valeur de la hausse.

Les lignes 210 à 290 calculent la vitesse initiale et le paramètre de la trajectoire de l'obus.

Les lignes 300 à 340 calculent la trajectoire.

Les lignes 340 à 390 assurent le déroulement de la fin du jeu.

Les lignes 1000 à 1030 et 4000 sont le sous-programme de redéfinition des caractères.

Les lignes 2000 à 2080 dessinent la forteresse et le canon.

Les lignes 3000 à 3080 dessinent le baron sur sa trajectoire.

```
5 REM  ************************
10 REM * BARON DE MUNSCHAUSEN *
15 REM ************************
20 PAPER2:INK5:GOSUB 1000
30 HIRES
40 PAPER6:INK1
50 FILL40,1,23
60 XP=10:YP=189
100 XR=INT(RND(1)*90+120)
110 YR=189-INT(RND(1)*50+10)
115 GOSUB 2000
120 PRINT"Charge simple(1),double(2)ou triple(3)?"
130 GET A$
140 IF A$>"3" OR A$<"1" THEN GOTO 130
150 R=42+8*VAL(A$)
160 INPUT"Hausse en degres";A$
170 FOR I=1 TO LEN(A$)
124 IF MID$(A$,I,1)<"0" OR MID$(A$,I,1)>"9" THEN GOTO 160
190 NEXT I
195 IF VAL(A$)<0 OR VAL(A$)>88 THEN GOTO 160
200 A=VAL(A$)*0.0174533
210 VX=R*COS(A)
220 VY=R*SIN(A)
230 YM=VY^2/19.62
240 IF YM>181 THEN PRINT"Hausse trop forte,recalculez!":GOTO 160
260 X=20:Y=189
270 B=4.9/VX^2
280 C=-TAN(A)
290 D=189
295 EXPLODE
300 REPEAT
305 IF X>XR+6 AND Y>YR+22 THEN SHOOT:PRINT"AIE!":WAIT150:GOTO 120
310 IFY>191 OR X>231 THEN GOTO120
315 GOSUB 3000
320 X=X+5
330 Y=(B*(X-20)+C)*(X-20)+D
340 UNTILY>YR-6ANDY<YR+6ANDX>XR-6ANDX<XR+6
345 CALL#FB03
350 PRINT"Bravo,le baron est dans la forteresse"
360 WAIT 200
370 INPUT"VOULEZ VOUS REJOUER";Z$
380 IF Z$="O" THEN GOTO 30
390 END
1000 FOR I=46840 TO 46855
1010 READ J
1020 POKE I,J
1030 NEXT I
1040 RETURN
2000 CURSETXR-6,YR,1
2010 FILL200-YR,2,63
2020 CURSETXR+6,YR+6,1
2030 FILL194-YR,39-INT(XR/6),44
2040 CURSET20,189,1
2050 DRAW-7,3,1
2060 CURSET17,195,1
2070 CIRCLE4,1
2080 RETURN
3000 CURSETX,Y,1
3005 CHAR95,0,1
3010 CURMOV0,-8,0
3020 CHAR96,0,1
```

```
3030 CURSETXP,YP,1
3040 CHAR95,0,0
3050 CURMOV0,-8,0
3060 CHAR96,0,0
3065 CURSETXP,YP,1
3070 XP=X:YP=Y
3080 RETURN
4000 DATA 58,57,62,18,58,19,0,0,0,0,0,24,60,24,16,60
```

3.3 MISSILE

Pourquoi donc cette planète désolée intéresse-t-elle tous les vaisseaux hostiles qui veulent atterrir pour s'emparer des lieux ? C'est un mystère pour vous. De toute manière ce n'est pas votre problème, vous devez suivre les ordres du commandement suprême sans en chercher les motifs. Vous êtes l'ultime sentinelle au front, vous devez interdire l'accès à la planète. Pour cela vous disposez d'un arsenal ultra-moderne de missiles ultra-sophistiqués. Ils sont munis d'une tête chercheuse qui fait que ces missiles font mouche à tout coup. Encore faut-il que l'ordre de tir soit donné au bon moment, c'est-à-dire quand un missile est disposé dans le silo de tir. L'ordre peut être alors validé par l'ordinateur central dirigeant l'organisation du poste de tir. Le missile peut alors partir et ne laisse aucune chance à l'adversaire.

Et l'homme dans tout cela ? Il est indispensable car la machine n'est pas habilitée à ordonner le tir.

Pour donner cet ordre, pressez la touche d'espacement. Le jeu comporte trois niveaux de jeu modifiant la vitesse de déplacement du pas de tir, et la durée de mise en fonction des missiles.

La séance de tir est limitée à la durée de votre tour de garde.

Ne décevez pas les espoirs qui sont mis en vous. Défendez ardemment votre planète.

Structure du programme :

Lignes 20 à 170, présentation du jeu et initialisations.

Lignes 180 à 220, dessin du sol de la planète et coloration de l'écran.

Lignes 230 à 250, dessin du ciel étoilé.

Lignes 260 à 270, impression des messages de score.

Lignes 280 à 380, double boucle imbriquée qui compte le temps et assure le déplacement du pas de tir.

Ligne 320, teste si l'ordre de tir est donné.

Ligne 350, autorise le déplacement des vaisseaux.

Lignes 390 à 450, fin du jeu.

Lignes 1000 à 1040 et 4000 à 4060, définition des caractères spéciaux représentant les vaisseaux et le missile.

Lignes 2000 à 2130, sous-programme de tir. Interdit le tir si le missile n'est pas prêt (compteur G inférieur au temps F dépendant du niveau).

Lignes 3000 à 3040 : déplacement des vaisseaux par rotation de la chaîne de caractères les contenant et ajout d'un nouveau vaisseau.

```
5 REM  **********************
10 REM * MISSILE GALACTIQUE *
15 REM **********************
20 CLS:PAPER 3:INK 0:PRINTCHR$(20);CHR$(17)
30 GOSUB 1000
40 FOR I=1 TO 15
50 A$=A$+CHR$(100+INT(RND(1)*10))+" "
60 NEXT I
70 PLOT 2,6,12
80 PLOT10,6,"MISSILE GALACTIQUE"
90 PLOT2,10,"QUEL NIVEAU CHOISISSEZ VOUS"
100 PLOT5,12,"1:DEBUTANT"
110 PLOT5,14,"2:ENTRAINE"
120 PLOT5,16,"3:CHAMPION"
130 GET E$
140 IF E$<"1" OR E$>"3" THEN GOTO 130
150 DX=VAL(E$)/1.5+1
160 F=3+VAL(E$)
170 CLS
180 PAPER0:INK6
190 FOR I=1 TO 6
200 READ J,K
```

```
210 POKE J,K
220 NEXT I
230 FOR I=1 TO 50
240 PLOT RND(1)*36+1,RND(1)*20,"."
250 NEXT I
260 TP=500:N=1:XC=10:YC=23:YT=3:G=0
270 PLOT 1,25,"TEMPS:":PLOT 24,25,"SCORE:"
280 REPEAT
290 REPEAT
300 XC=XC+DX
310 PLOT XC,YC,"   no   "
320 IF KEY$=CHR$(32) THEN GOSUB 2000
330 TP=TP-1:TP$=STR$(TP):PLOT11-LEN(TP$),25,TP$
340 N=N+1:G=G+1
350 IF N=15 THEN GOSUB 3000
360 UNTIL XC>30 OR XC<4 OR TP=0
370 DX=-DX
380 UNTIL TP=0
390 POKE 48160,INT(RND(1)*6+1)
400 PLOT 3,10,"PARTIE TERMINEE"
410 PLOT 3,12,"VOULEZ VOUS REJOUER?(O/N)"
415 PRINTCHR$(20);CHR$(17)
420 GET E$
430 IF E$="O" THEN RUN
440 IF E$="N" THEN END
450 GOTO420
990 REM * Def caracteres speciaux *
1000 FOR I=46880 TO 46983
1010 READ J
1020 POKEI,J
1030 NEXT I
1040 RETURN
1990 REM * TIR *
2000 IF G<F THEN RETURN
2010 ZAP
2020 SN=SCRN(XC+3,YT)
2030 G=0
2040 IF SN>110 OR SN<100 THEN RETURN
2050 FOR I=1 TO YC-YT
2060 PLOTXC+3,YC-I-1,"P"
2070 PLOTXC+3,YC-I," "
2080 NEXT I
2090 EXPLODE
2100 PLOTXC+3,YT-1," "
2110 A$=LEFT$(A$,XC-1)+" "+RIGHT$(A$,30-XC)
2120 SC=SC+SN-95:PLOT30,25,STR$(SC)
2130 RETURN
2990 REM * Depl cible *
3000 N=0
3010 PING
3020 A$=" "+CHR$(INT(RND(1)*10)+100)+LEFT$(A$,28)
3030 PLOT4,YT,A$
3040 RETURN
4000 DATA0,32,16,40,31,40,16,32,60,16,56,55,56,16,60,0
4010 DATA 56,32,34,52,30,15,28,0,12,26,55,26,55,26,12,0
4020 DATA 20,28,8,28,63,8,63,20,48,32,20,43,20,32,48,0
4030 DATA 42,8,28,62,28,8,42,0,18,63,45,63,33,18,33,18
4040 DATA 32,40,58,30,53,30,12,56,20,63,43,63,3,15,9,37
4050 DATA 1,1,1,3,3,7,15,57,0,0,0,32,32,48,56,14
4060 DATA 8,8,8,8,8,8,28,42
4070 DATA 48160,2,49000,17,49040,17,49080,17,49041,3,48961,5
```

3.4 BRACONNIER

"Alors ! le gibier ? Beaucoup, beaucoup trop. Les lièvres sont très prolifiques cette année et, si cela continue, les cultures risquent d'être endommagées. Dommage que la chasse ne soit pas encore ouverte ..."

Voilà des propos que vous avez surpris ce matin, alors que vous sirotiez votre pastis à la terrasse du café. Ils vous reviennent à l'esprit par cette soirée de pleine lune, et incitent vos instincts de braconnier à remonter en surface. La science des collets est un art que vous dominez depuis fort longtemps. Le matériel rassemblé, la gibecière en bandoulière, vous vous lancez en rase campagne en quête de ces rongeurs destructeurs. Votre but : poser le plus de pièges possibles. Votre tactique : repérer les traces de ces rongeurs, et poser un nombre de collets proportionnel à la densité des traces. Attention bien sûr de ne pas mettre les pieds dans un des collets déjà posés. Prenez garde aussi à regagner votre domicile dans les délais imposés par votre femme, celle-ci étant très jalouse, son courroux pourrait être grand lorsque vous reviendrez.

Vous pourrez vous déplacer en utilisant les quatre flèches. Les traces citées seront symbolisées par un chiffre : si vous passez sur ce chiffre vous poserez un à un autant de pièges que le chiffre indiqué. Les collets posés seront indiqués par des "*". Un "ZAP" vous indiquera le moment où il vous faudra rejoindre votre véhicule, regagnez-le dans les plus brefs délais. Surtout, ne repassez pas sur vos pièges. Bon braconnage.

Structure du programme :

Lignes 9 à 95, initialisations.

Lignes 99 à 200, dépôt des pièges.

Lignes 209 à 370, retour.

Lignes 399 à 470, sortie chiffres.

Lignes 499 à 530, affichage pièges.

Lignes 599 à 710, déplacement braconnier.

```
9 REM  ***************
10 REM * BRACONNIER *
11 REM ***************
19 REM * Initialisations *
20 CLS
30 PAPER4 :INK6
40 PLOT10,25,"PIEGES POSES="
50 FOR I=1 TO 2
55 READ K
60 FOR J=0 TO 7
65 READ L
70 POKEK+J,L
80 NEXT J
85 NEXT I
87 DATA 46360,0,0,0,30,18,63,63,18
88 DATA 46592,12,12,63,18,30,18,12,0
90 D=1 :T=0 :S=0 :TR=0
92 GOSUB 465
95 X=20 :Y=13
99 REM * Depot des Pieges *
100 REPEAT
105 T=T+1
110 M$=KEY$
115 XE=X :YE=Y
120 IF M$>=CHR$(8) AND M$<=CHR$(110) THEN ON ASC(M$)-7 GOSUB600,630
,660,690
130 IF RND(1)<.1 THEN GOSUB 400
140 IF SCRN(X,Y)>48 AND SCRN(X,Y)<58 THEN GOSUB 450
150 IF SCRN(X,Y)=42 THEN EXPLODE :D=0
155 GOSUB 500
160 UNTIL D=0 OR T=1000
170 PING :V=0 :T=0
180 XC=INT(33*RND(1))+3 :YC=INT(22*RND(1))+1
190 IF SCRN(XC,YC)<>32 THEN GOTO 180
195 ZAP
200 PLOTXC,YC,163
209 REM * Retour *
210 REPEAT
220 XE=X :YE=Y
230 M$=KEY$
240 IF M$>=CHR$(8) AND M$<=CHR$(11) THEN ON ASC(M$)-7 GOSUB 600,630
,660,690
250 IF SCRN(X,Y)=42 THEN EXPLODE :D=0
260 IF X=XC AND Y=YC THEN V=1
270 PLOTXE,YE," "
280 PLOTX,Y,192
290 T=T+1
300 UNTIL T=300 OR D=0 OR V=1
310 CLS
320 IF D=0 THEN PRINT"VOUS ETES TOMBES DANS VOTRE PIEGE":PRINT"C'ES
T MALIN"
330 IF V=1 THEN PRINT"BRAVO VOUS ETES DANS LES TEMPS"
340 IF D<>0 AND V<>1 THEN PRINT"GARE AU ROULEAU A PATISSERIE"
350 PRINT :INPUT"VOULEZ VOUS REJOUER (O/N)";R$
360 IF R$="O" THEN RUN
370 END
399 REM * Sortie chiffres "
400 XC=INT(33*RND(1))+3
410 YC=INT(22*RND(1))+1
420 PLOTXC,YC,9*RND(1)+49
430 RETURN
```

```
449 REM * Affichage nbr Pieces *
450 S=S+SCRN(X,Y)-48
460 TR=TR+SCRN(X,Y)-48
465 PLOT24,25,STR$(S)
470 RETURN
499 REM * Affichage Pieces *
500 PLOTXE,YE," "
510 IF TR<>0 THEN PLOTXE,YE,"*"
515 IF(X<>XE OR Y<>YE) AND TR<>0 THEN TR=TR-1
520 PLOTX,Y,192
530 RETURN
599 REM * Deplacement braconnier *
600 X=X-1
610 IF X<1 THEN X=X+1
620 RETURN
630 X=X+1
640 IF X>36 THEN X=X-1
650 RETURN
660 Y=Y+1
670 IF Y>23 THEN Y=Y-1
680 RETURN
690 Y=Y-1
700 IF Y<1 THEN Y=Y+1
710 RETURN
```

CHAPITRE **4**

Jeux d'action

4.1 ASTÉROÏDES

Une expédition d'exploration scientifique est en route vers Jupiter. Vous êtes le commandant du vaisseau spatial. Hélas vous devez traverser la barrière d'astéroïdes après avoir dépassé l'orbite de Mars. Cela peut être périlleux et demande toute l'expérience du vétéran de l'espace que vous êtes. Heureusement, un dispositif révolutionnaire équipe votre vaisseau. Un désintégrateur est fixé à l'avant de celui-ci et pulvérise les astéroïdes arrivant devant vous. Mieux encore, la matière constituant ces blocs de rochers est récupérée et utilisée comme carburant. Pour poursuivre votre route vous devez donc détruire un maximum d'astéroïdes pour emplir vos réservoirs. Mais des fusées ennemies patrouillent également dans cette portion du cosmos et votre désintégrateur est sans effet sur elles après trois salves de grande puissance. Prenez donc garde à ne pas en rencontrer plus de deux sur votre route.

Pour déplacer le vaisseau, utilisez les flèches droite et gauche.

Structure du programme :

Lignes 15 à 45, présentation et initialisations.

Lignes 50 à 140, boucle principale, assurent :

- la création des astéroïdes et le scroll ligne 60
- le comptage des astéroïdes touchés ligne 70
- le déplacement du vaisseau lignes 90 à 110
- le comptage et l'arrêt du jeu lignes 120 à 140
- la création des fusées ennemies ligne 130.

Lignes 150 à 210, assurent la fin du jeu et l'affichage des résultats.

Lignes 1000 à 1060 et 3000, sous-programme assurant la redéfinition des caractères.

Lignes 2000 à 2100, traitement de la collision avec les fusées ennemies.

```
5 REM  ***************
10 REM * ASTEROIDES *
12 REM ***************
15 CLS:PAPER4:INK3:PRINTCHR$(17);CHR$(20)
20 PLOT 5,10,"DETRUISEZ LES ASTEROIDES"
30 XV=10:YV=0:TP=1000:V=3
40 GOSUB 1000:WAIT100
45 CLS:PAPER0
50 REPEAT
60 PRINTSPC(RND(1)*18)"*"SPC(RND(1)*18)"*"
70 IF SCRN(XV,YV+1)=42 THEN SC=SC+1:SHOOT
80 IF SCRN(XV,YV+1)=96 THEN GOTO2000
90 K$=KEY$
100 XV=XV+(K$=CHR$(8) AND XV>2)-(K$=CHR$(9) AND XV<37)
110 PLOTXV,YV,95
120 TP=TP-1
130 IF RND(1)>0.7 THEN PLOT RND(1)*37+1,RND(1)*24+2,96
140 UNTIL TP=<0
150 PING
160 PLOT2,10,"VOUS AVEZ DETRUIT "+STR$(SC)+" ASTEROIDES"
170 PLOT2,12,"VOULEZ VOUS REJOUER?"
175 PRINTCHR$(17);CHR$(20)
180 GETA$
190 IF A$="O" THEN RUN
200 IF A$="N" THEN END
```

```
210 GOTO180
1000 FOR I=1 TO 2
1010 READ K
1020 FOR J=0 TO 7
1030 READ L
1040 POKEK+J,L
1050 NEXTJ,I
1060 RETURN
2000 V=V-1
2010 EXPLODE
2020 FOR I=7 TO 0 STEP-1
2030 PAPERI
2040 WAIT10
2050 NEXT I
2060 PLOTXV,YV,223
2080 PLOTXV,YV+1," "
2090 IF V=0 THEN PLOT2,8,"VOTRE VAISSEAU A EXPLOSE":GOTO160
2100 GOTO90
3000 DATA 46840,21,31,31,10,14,4,4,4,46848,8,8,28,8,28,62,54,34
```

4.2 CHAMPS DE FORCE

Terra, an 3047, alors que l'expansion galactique bat son plein essor, alors que toutes les nations belligérantes ont déposé leurs armes, alors que "tout va pour le mieux dans le meilleur des mondes", votre planète se trouve menacée par une force étrangère. Nul ne connaît son origine, mais son ultimatum est précis : "la planète sera détruite dans 500 unités de temps". Tous les grands de chaque nation se sont réunis. Leur décision : la destruction de cette entité ennemie. Le moyen a été déterminé par une entente de toutes les unités de recherche techniques, scientifiques et militaires : une charge doit être déposée au centre du champ de force. C'est dans le cadre de cette mission que vous pilotez actuellement la fusée interplanétaire transportant la charge. Vous devez dans une première étape rejoindre votre navette, puis, à l'aide de la navette, aller déposer la charge. Ces deux étapes sont primordiales, car la fusée serait détruite par le champ de force, alors que la navette plus puissante pourra y entrer et en ressortir facilement. Mais attention, l'ennemi n'est pas dupe : il essaiera de détruire la fusée, mais fuira la navette lorsque celle-ci sera en possession de la charge. Pour la plus grande réussite de votre mission, vous devez sortir du champ de force avant que la charge à retardement n'explose. Tous les espoirs de vos co-planétaires reposent sur vous ... Trois tentatives vous sont permises.

Le programme vous propose trois niveaux de jeu, la vitesse de déplacement du champ de force croît avec le niveau. Les touches du clavier utilisées sont les flèches pour les déplacements, la barre d'espacement pour le tir.

Structure du programme :

Lignes 10 à 54, présentation.

Lignes 60 à 86, définition des caractères spéciaux.

Lignes 89 à 104, choix du niveau de jeu.

Lignes 109 à 240, initialisations.

Lignes 299 à 390, première boucle, trajet terra-navette.

Lignes 399 à 610, déplacement fusée.

Lignes 699 à 750, destruction fusée.

Lignes 799 à 860, deuxième boucle, trajet navette vers champ de force.

Lignes 899 à 990, déplacement champ de force.

Lignes 999 à 1130, fin, sortie du programme.

Lignes 1199 à 1300, déplacement vaisseau.

Lignes 1399 à 1455, dépôt de la charge.

Lignes 1459 à 1510, explosion de la charge.

Lignes 1599 à 1640, commentaires si partie gagnée.

Lignes 1699 à 1810, poursuite de la navette par le champ de force.

```
---
9 REM  ******************
10 REM * CHAMP DE FORCE *
11 REM ******************
15 TEXT
20 PAPER0 :INK2
25 PRINTCHR$(17)
50 CLS
51 PRINT:PRINT:PRINTSPC(10)"CHAMP DE FORCE"
52 PRINT:PRINT:PRINTSPC(7)"SAUVEZ VOTRE PLANETE"
53 PRINT:PRINT:PRINTSPC(8)"DETRUISEZ L'ENNEMI"
54 PRINT:PRINT:PRINT
60 REM * DEFINITION CARACTERES *
62 FOR I=1 TO 7
64 READ K
66 FOR J=0 TO 7
68 READ L
70 POKEK+J,L
72 NEXT J
74 NEXT I
80 DATA46856,0,1,15,59,15,1,0,0
81 DATA46864,12,63,63,45,63,63,12,0
82 DATA46872,0,32,60,47,60,32,0,0
83 DATA46592,4,4,4,14,14,31,21,17
84 DATA46360,0,7,12,62,12,7,0,0
85 DATA46376,0,56,12,31,12,56,0,0
86 DATA46344,17,21,31,14,14,4,4,4
89 REM * CHOIX DU NIVEAU *
90 TEXT
91 PRINT"NIVEAUX DES DIFFICULTES"
92 PRINT"  1:FAIBLE"
94 PRINT"  2:MOYEN"
96 PRINT"  3:FORT"
98 INPUT"QUEL NIVEAU CHOISISSEZ VOUS";NI$
100 IF ASC(NI$)<49 OR ASC(NI$)>51 THEN GOTO 98
102 NI=VAL(NI$)*2
104 PAPER4
109 REM * INITIALISATIONS *
110 X=30 :Y=190 :FU=3 :XN=200 :YN=40
120 XC=120 :YC=100 :R=20 :TE=0
121 HIRES :PAPER0 :INK3
122 FOR I=1 TO 100
123 CURSETINT(220*RND(1))+10,INT(190*RND(1))+2,3
124 FILL1,1,16
125 CURMOV-6,0,3
126 FILL1,1,INT(7*RND(1))+16
127 NEXTI
130 CURSETXC,YC,3
140 CIRCLER,1
150 CURSETX,Y,3
160 CHAR64,0,1
170 FOR I=1 TO 3
180 CURSETXN+(I-2)*6,YN,3
190 CHAR96+I,0,1
200 NEXT I
210 FOR I=1 TO 2
220 POKE49042+2*I,64
230 NEXT I
240 XF=49046
299 REM * PREMIERE BOUCLE *
300 M$=KEY$
310 IF M$>=CHR$(8) AND M$<=CHR$(11) THEN GOSUB 400
320 EC=(X-XC)*(X-XC)+(Y-YC)*(Y-YC)
330 EN=(X-XN)*(X-XN)+(Y-YN)*(Y-YN)
```

```
340 IF EC<R*R THEN GOSUB 700
350 IF EN<64 THEN GOSUB 800
360 IF RND(1)<NI/10 THEN GOSUB 900
370 TE=TE+1
375 GOSUB 1740
380 IF TE=500 THEN GOTO 1000
382 FOR I=1 TO 3
384 CURSETXN+(I-2)*6,YN,3
386 CHAR96+I,0,1
388 NEXT I
390 GOTO 300
399 REM * DEPLACEMENT FUSEE *
400 XE=X :YE=Y
410 ON ASC(M$)-7 GOSUB500,530,560,590
420 CURSETXE,YE,0
430 CHAR127,0,0
440 CURSETX,Y,3
450 CHARCG,0,1
460 RETURN
500 X=X-4
510 IF X<20 THEN X=X+4
515 CG=35
520 RETURN
530 X=X+4
540 IF X>220 THEN X=X-4
545 CG=37
550 RETURN
560 Y=Y+4
570 IF Y>180 THEN Y=Y-4
575 CG=33
580 RETURN
590 Y=Y-4
600 IF Y<20 THEN Y=Y+4
605 CG=64
610 RETURN
699 REM * FUSEE DANS CHAMP *
700 EXPLODE
710 XE=X :YE=Y
720 X=30 :Y=190
730 FU=FU-1
740 IF FU=0 THEN GOTO 1000
745 POKEXF,20
747 XF=XF-2
750 GOTO 420
799 REM * DEUXIEME BOUCLE *
800 TR=1000 :BOUM=0 :DEST=0
801 CURSETX,Y,3
802 CHAR127,0,0
805 X=XN :Y=YN
806 XF=XF+2
807 POKEXF,64
809 M$=KEY$
810 IF M$>=CHR$(8) AND M$<=CHR$(11) THEN GOSUB 1200
820 IF M$=CHR$(32) THEN GOSUB 1400
830 TE=TE+1
835 TR=TR-1
837 GOSUB 1740
840 IF TE=500 THEN GOTO 1000
845 IF TR=0 THEN GOTO 1460
850 IF RND(1)<NI/10 THEN GOSUB 1700
860 GOTO 809
```

```
899 REM * DEPLACEMENT CERCLE FUSEE *
900 DX=SGN(X-XC)*INT(10*RND(1))
910 DY=SGN(Y-YC)*INT(10*RND(1))
920 IF XC+DX<40 OR XC+DX>200 THEN DX=0
930 IF YC+DY<40 OR YC+DY>160 THEN DY=0
940 CURSETXC,YC,3
950 CIRCLER,0
960 R=20+INT(10*RND(1))
970 XC=XC+DX :YC=YC+DY
975 CURSETXC,YC,3
980 CIRCLER,1
990 RETURN
999 REM * FIN DU JEU *
1000 P=1
1010 FOR I=1 TO 10
1020 PAPERP
1030 IF P=1 THEN P=3 :GOTO 1050
1040 P=1
1050 EXPLODE
1060 NEXT I
1070 PAPER0
1080 PRINT"MISSION ECHOUEE"
1090 PRINT"VOTRE PLANETE EST DETRUITE
1095 FOR I=49000 TO49080 STEP40
1096 POKEI,16
1097 NEXT I
1100 WAIT 100
1110 INPUT"VOULEZ VOUS RECOMMENCER(O/N)";R$
1120 IF R$="O" THEN RUN 90
1130 END
1199 REM * DEPLACEMENT VAISSEAU *
1200 XE=X :YE=Y
1210 ON ASC(M$)-7 GOSUB500,530,560,590
1220 FOR I=1 TO 3
1230 CURSETXE+(I-2)*6,YE,0
1240 CHAR96+I,0,0
1250 NEXT I
1260 FOR I=1 TO 3
1270 CURSETX+(I-2)*6,Y,0
1280 CHAR96+I,0,1
1290 NEXT I
1300 RETURN
1399 REM * DEPOT BOMBE *
1400 IF BOUM=1 THEN GOTO 1450
1410 ZAP
1415 TR=15
1420 BOUM=1
1430 EC=(X-XC)*(X-XC)+(Y-YC)*(Y-YC)
1440 IF EC<R*R/4 THEN DEST=1
1450 IF EC<R*R THEN TE=TE-50
1455 RETURN
1459 REM * MISE A FEU *
1460 EXPLODE
1470 EC=(X-XC)*(X-XC)+(Y-YC)*(Y-YC)
1475 IF DEST=1 THEN GOTO 1600
1480 IF EC<R*R THEN GOTO 1000
1490 XN=X :YN=Y
1500 X=30 :Y=190
1505 POKEXF,20
1506 XF=XF-2
```

```
1507 CG=64
1510 GOTO 150
1599 REM * PARTIE GAGNEE *
1600 CURSETXC,YC,3
1610 CIRCLER,0
1611 FOR I=1 TO 3
1612 POKE49070+I,32
1613 NEXT I
1620 PRINT:PRINT"VOTRE PLANETE EST SAUVEE"
1622 WAIT200
1625 PRINT:PRINT"IL VOUS RESTAIT UNE MARGE DE:";500-TE
1626 WAIT 200
1630 PRINT:PRINT"LA POPULATION VOUS FELICITE"
1635 PRINT
1640 GOTO 1110
1699 REM * DEPLACEMENT CERCLE NAV *
1700 DX=SGN(X-XC)*INT(10*RND(1))
1710 DY=SGN(Y-YC)*INT(10*RND(1))
1720 IF BOUM=0 THEN DX=-DX :DY=-DY
1730 GOTO 920
1740 T=500-TE
1750 T(1)=INT(T/100)+48
1760 T(2)=INT(T/10)-10*(T(1)-48)+48
1770 T(3)=T-10*(T(2)-48)-100*(T(1)-48)+48
1780 FOR I=1 TO 3
1790 POKE49070+I,T(I)
1800 NEXT I
1810 RETURN
3000 FOR I=#EB0 TO #F00
3010 PRINTHEX$(PEEK(I));
3020 PRINTHEX$(I);
3030 NEXT I
```

4.3 DÉMINAGE

La guerre approche de sa fin. Pour exterminer l'un des derniers bastions ennemis, une colonne de chars d'assaut doit emprunter la vallée d'Abdel-Kir. Cette vallée étant un lieu de passage stratégique, l'ennemi y a enterré, dans le sable, des mines. Ces mines insensibles au poids d'un matériel léger sont un obstacle certain pour des véhicules lourds. L'état-major a donc décidé d'envoyer sur les lieux des unités du corps du génie. Votre rôle en tant qu'artificier sera de désarmorcer les explosifs. Mais cette tâche ne sera pas facile, en effet des chars ennemis surveillent ce champ de mines et tenteront de vous anéantir. Votre seule chance de leur échapper est de regagner votre point de départ, ce qui vous ralentira dans votre tâche. Non contents de vous retarder, les ennemis redéposeront des mines en regagnant leurs positions. L'impératif temps est très important, il faut que la colonne alliée puisse passer avant que l'ennemi ne détruise tous les documents compromettants.

Vous pourrez déplacer votre véhicule grâce aux quatre flèches. Il vous suffira de passer sur la bombe pour la désamorcer. Le temps qui vous est imparti est limité.

Maintenant, ne perdez plus une minute: au travail!

Structure du programme :

Lignes 9 à 40, mise en place du décor.

Lignes 50 à 86, définitions des caractères spéciaux.

Lignes 100 à 160, pose des mines.

Lignes 170 à 190, initialisations.

Lignes 200 à 290, campagne de déminage.

Lignes 400 à 610, déplacement du véhicule du génie.

Lignes 700 à 830, poursuite du véhicule par char ennemi.

Lignes 1000 à 1080, retour char ennemi sur ses positions.

Lignes 1200 à 1230, affichage temps.

Lignes 1500 à 1540, explosion char sur mine.

Lignes 1700 à 1730, pose des mines lors du retour du char.

```
_____
9 REM  ************
10 REM * DEMINAGE *
11 REM ************
25 CLS
27 TEXT:PAPER3:INK0
30 POKE49040,16
40 POKE49041,2
50 FOR I=1 TO 3
55 READ K
60 FOR J=0 TO 7
65 READ L
70 POKEK+J,L
75 NEXT J
80 NEXT I
82 DATA 46840,0,63,33,33,45,63,63,18
84 DATA 46592,21,14,31,14,21,0,0,0
86 DATA 46848,30,30,63,63,18,0,0,0
100 TEXT
120 FOR I=1 TO 20
130 X=INT(34*RND(1)+3)
140 Y=INT(20*RND(1)+2)
145 IF SCRN(X,Y)=64 THEN I=I-1
```

```
150 PLOTX,Y,64
160 NEXT I
170 MI=20 :TE=1000
175 PLOT4,25,64 :GOSUB 1525
177 PLOT25,25,"T" :GOSUB 1210
180 X=36 :Y=2
190 PLOTX,Y,95
200 REPEAT
205 GOSUB 1200
210 M$=KEY$
220 IF M$>=CHR$(8) AND M$<=CHR$(11) THEN GOSUB 400
230 IF RND(1)<.03 THEN GOSUB 700
250 UNTIL MI=0 OR TE=0
260 INPUT"VOULEZ VOUS REJOUER (O/N)";R$
270 IF R$="O" THEN RUN
280 PRINT:PRINT"FIN"
290 END
400 XE=X :YE=Y
410 ON ASC(M$)-7 GOSUB500,530,560,590
420 IF SCRN(X,Y)=64 THEN PING:GOSUB 1520
430 PLOTXE,YE," "
440 PLOTX,Y,95
450 RETURN
500 X=X-1
510 IF X<2 THEN X=X+1
520 RETURN
530 X=X+1
540 IF X>36 THEN X=X-1
550 RETURN
560 Y=Y+1
570 IF Y>23 THEN Y=Y-1
580 RETURN
590 Y=Y-1
600 IF Y<2 THEN Y=Y+1
610 RETURN
700 XC=2 :YC=23 :D=1
710 D=-D
711 GOSUB 1200
720 M$=KEY$
730 XD=XC :YD=YC
740 IF M$>=CHR$(8) AND M$<=CHR$(11)   THEN GOSUB 400
745 IF RND(1)>.5 THEN GOTO 720
750 IF D=1 THEN GOTO 810
760 YC=INT(3*RND(1))*SGN(Y-YC)+YC
770 IF XC=X OR YC=Y THEN GOTO 1000
780 PLOTXD,YD," "
785 IF SCRN(XC,YC)=64 THEN GOSUB 1500
790 PLOTXC,YC,96
800 IF XC<>20R YC<>23 THEN GOTO 710
805 RETURN
810 XC=INT(3*RND(1))*SGN(X-XC)+XC
820 IF YC=Y THEN GOTO 1000
830 GOTO 780
1000 IF X=36 AND Y=2 THEN GOTO 1032
1005 WAIT10
1010 EXPLODE
1020 PLOTX,Y," "
1026 WAIT5
1030 X=36 :Y=2
1032 PLOTX,Y,"£"
1035 PLOTXC,YC," "
```

```
1040 XC=INT(3*RND(1))*SGN(2-XC)+XC
1050 YC=INT(3*RND(1))*SGN(23-YC)+YC
1058 GOSUB 1200
1060 IF SCRN(XC,YC)=64 THEN GOSUB 1500
1063 PLOTXD,YD," "
1065 IF RND(1)<.1 THEN GOSUB 1700
1066 PLOTXC,YC,96
1067 XD=XC :YD=YC
1070 IF XC<>2 OR YC<>23 THEN GOTO 1040
1080 GOTO 805
1200 TE=TE-1
1205 IF TE-INT(TE/10)*10=9 THEN PLOT26,25,"      "
1210 PLOT26,25,STR$(TE)
1220 IF TE=0 THEN GOTO 260
1230 RETURN
1500 EXPLODE
1505 PLOTXC,YC," "
1510 XC=2 :YC=23
1520 MI=MI-1
1525 IF MI=9 THEN PLOT6,25,"    "
1530 PLOT5,25,STR$(MI)
1540 RETURN
1700 IF (XD=2 AND YD=23) OR (XD=36 AND YD=2) THEN GOTO 1730
1705 IF XC=XD AND YC=YD THEN GOTO 1730
1707 PLOTXD,YD,64
1710 MI=MI+1
1715 ZAP
1720 GOSUB 1530
1730 RETURN
```

4.4 ENVAHISSEURS

Ca y est, cela est arrivé ! Les extraterrestres attaquent. Ils veulent débarquer en grand nombre sur la terre. Il y en a de six espèces différentes plus étranges et monstrueuses les unes que les autres. Leur hostilité ne fait aucun doute. Que faire pour éviter cette invasion terrible ? Il faut les exterminer tous. Aucun ne doit rester. Mais ils se défendent ardemment à l'aide de bombes qu'ils envoient en grappes. Vous avez trois canons-laser à votre disposition. Cependant ces armes sont assez lentes à manoeuvrer et la durée entre chaque tir est assez longue. D'autre part vous ne pouvez tirer qu'un seul coup à la fois. Il faut donc viser juste et ne pas rater un seul envahisseur. Pour les aider, ils disposent d'un vaisseau spatial qui patrouille sporadiquement dans le ciel. Détruisez-le à tout prix car il contient 8 envahisseurs ce qui augmentera de 100 votre total. Néanmoins, les envahisseurs jouent sur le nombre, et si vous arrivez à détruire une escouade complète une autre prend la relève immédiatement. Sauvez votre planète natale de la menace cosmique ! Pour déplacer le canon utilisez les touches fléchées droite et gauche, pour tirer la barre d'espacement.

Structure du programme :

```
5 REM  ********************************
10 REM * LES ENVAHISSEURS DE L'ESPACE *
15 REM ********************************
20 GOSUB1000:GOSUB7000
30 DIM A$(6),B(6),X(20),Y(20)
40 PAPER0:INK3
42 SC=0:TP=0:NC=3:NS=0
45 CLS:NE=6:M2=10
50 FOR I=1 TO 6
60 READ C,D
70 POKE C,D
80 NEXT I
100 B$="                "
110 FOR I=1 TO 6
120 A$(I)=" "
130 FOR J=1 TO 6
140 A$(I)=A$(I)+CHR$(I+111)+" "
150 NEXT J,I
170 XE=8:YE=2:XC=10:YC=23:PLOTXC,YC," bc"
180 GOSUB 4000
190 NX=1:MY=1:T=0:S=0:YS=1:YL=1
200 FOR I=1 TO 6:B(I)=6:NEXT I
210 FOR I=0 TO NC-1
220 PLOT4+3*I,25,"bc "
230 NEXTI
240 FOR I=1 TO 8
250 X(I)=INT(RND(1)*12+XE+1)
260 Y(I)=16+INT(RND(1)*3.1)
270 NEXTI
300 REM * Programme Principal *
310 REPEAT
320 M=1
330 REPEAT
340 N=1
350 REPEAT
360 K$=KEY$
370 IF(K$<CHR$(10)ANDK$>CHR$(7)) THEN GOSUB 2000
380 IF SCRN(XC+1,YC-1)=101 THEN GOSUB 6000
390 IF K$=CHR$(32)AND N3>2 THEN ZAP:GOSUB 9000
395 IF S=1 THEN GOSUB 2400
400 N=N+1:N1=INT((N+1)/2):N3=N3+1
410 TP=TP+1:PLOTXC,YC," bc "
412 PLOTX(N1),Y(N1)," "
414 Y(N1)=Y(N1)+0.5
416 IF Y(N1)=24 THEN X(N1)=INT(RND(1)*11+XE+1):Y(N1)=YE+NE*2
418 PLOTX(N1),Y(N1),"e"
420 UNTIL N=15
425 M=M+1
430 XE=XE+NX
440 IF XE<5 OR XE>22 THEN NX=-NX
450 GOSUB4000
455 IF SC>450 THEN M2=2
460 UNTILM=M2
470 YE=YE+MY:G=FRE("")
480 GOSUB4000
490 IF S=0 THEN IF RND(1)<0.5 THEN S=1:XS=2
500 UNTIL YE=YC-2*NE
510 GOTO 5000
520 END
```

```
990 REM * Definition des caracteres *
1000 FOR I=46976 TO 47023
1010 READ J
1020 POKEI,J
1030 NEXT I
1040 FOR I=46856 TO 46895
1050 READJ
1060 POKEI,J
1070 NEXTI
1080 RETURN
1990 REM * Depl canon *
2000 REM
2010 ON ASC(K$)-7 GOSUB 2200,2300
2020 RETURN
2090 REM * Tir laser *
2100 IF T=1 THEN RETURN
2110 T=1
2120 XL=XC+1
2130 YL=YC-1
2140 ZAP
2150 RETURN
2200 IF XC>7 THEN XC=XC-1:PLOTXC,YC," bc "
2210 RETURN
2300 IF XC<28 THEN XC=XC+1:PLOTXC,YC," bc "
2310 RETURN
2390 REM * Depl soucoupe *
2400 IFXS<38THENXS=XS+0.5:PLOTXS,YS,"d":PLOTXS-1,YS," "ELSEPLOTXS,Y
S," ":S=0
2405 SOUND1,1500,8
2410 RETURN
2990 REM * Effacement envahisseurs *
3000 SHOOT
3002 IF SR=100 THEN SC=SC+100:NS=NS+1:PLOTXS,YS," ":S=0:GOTO3035
3003 IF SR=101 THEN Y(N)=YE+NE*2:X(N)=INT(RND(1)*10+XE):RETURN
3005 W=(YL-YE)/2:Z1=XL-XE:Z2=12-XL+XE
3007 PLOTXL,YL-1,"  "
3010 A$(W)=LEFT$(A$(W),Z1)+" "+RIGHT$(A$(W),Z2)
3020 B(W)=B(W)-1
3030 SC=SC+13
3031 NE=NE+1
3032 REPEAT
3033 NE=NE-1
3034 UNTILB(NE)<>0 OR NE=0
3035 IF NE=0 THEN NE=6:GOTO3037
3037 SC$=STR$(SC):PLOT20,0,SC$:PLOTXL,YL," ":YL=24
3040 IF(B(1)+B(2)+B(3)+B(4)+B(5)+B(6))<>0THEN GOSUB4000:T=0:RETURN
3050 RESTORE
3060 FOR I=1 TO 88
3070 READ J
3080 NEXTI
3090 GOTO45
3990 REM * AFFICHAGE DES ENVAHISSEURS *
4000 FOR I=1 TO NE
4010 PLOT XE,YE+2*I,A$(I)
4020 PLOT XE,YE-1+2*I,B$
4030 NEXT I:G=FRE("")
4040 RETURN
4990 REM * FIN DU JEU *
5000 EXPLODE
4903 WAIT500
5020 CLS
5030 PRINT:PRINT:PRINT:PRINT"LES ENVAHISSEURS ONT ATTERRI!!!!"
5040 PRINT:PRINT:PRINT"VOUS EN AVEZ EMPECHES ";(SC-NS*100)/13;"DE S
E POSER"
5045 PRINT:PRINT:PRINT"EN ";TP;"QUANTUM DE TEMPS"
5048 PRINT:PRINT:PRINT"SCORE:";SC;"POINTS"
```

```
5050 PRINT:PRINT:PRINT"VOULEZ VOUS SUBIR UNE NOUVELLE ATTAQUE?(O/N)
"
5055 PRINTCHR$(20);CHR$(17)
5060 INPUT C$
5070 IF C$="O" THEN RUN
5080 IF C$="N" THEN RELEASE:END
5090 GOTO 5060
5990 REM * Canon detruit *
6000 NC=NC-1:EXPLODE:G=FRE("")
6010 IF NC=0 THEN CLS:GOSUB 5040:END
6020 PLOT4+(NC)*3,25," ":PLOTXC,YC-1," "
6030 PLOT XC,YC,"   "
6040 XC=10:PLOTXC,YC," bc":PLOTXC+1,YC-1," "
6050 RETURN
6990 REM * Presentation *
7000 CLS:PAPER1:INK3
7010 PRINT:PRINTCHR$(20);CHR$(17)
7020 PRINT:PRINTSPC(5)CHR$(27);"DP q r s t u P q r s t u"
7030 PRINT:PRINT"LES ENVAHISSEURS VEULENT ATTERRIR"
7040 PRINT:PRINT"SAUVEZ LA PLANETE AVEC LE CANON LASER"
7050 PRINT:PRINTSPC(10)CHR$(27);"EP q r s t u P"
7060 PRINT:PRINTSPC(12)CHR$(27)"Fq r s t u"
7070 PRINT:PRINTSPC(14)CHR$(27)"Bp q r"
7080 PRINT:PRINTSPC(16)CHR$(27)"Ds"
7090 PRINT:WAIT300
7100 RETURN
8000 DATA 33,45,63,12,30,49,33,51,12,18,51,63,45,12,18,51
8010 DATA 10,63,53,63,48,60,36,54,12,18,62,62,42,42,42,42
8020 DATA 40,20,30,63,45,63,18,45,36,46,63,53,31,30,18,33
8030 DATA 0,8,8,28,8,8,28,54,1,5,5,7,3,31,31,21
8040 DATA 0,16,16,48,32,60,60,20,0,4,46,53,46,4,0,0,0,14,2,4,8,4,2,
4
8050 DATA 48081,1,48961,2,49000,20,49040,20,49080,20,48041,6
9000 YL=YC-1:N3=0
9010 XL=XC+1
9020 REPEAT
9030 SR=SCRN(XL,YL-1)
9040 IF(SR<119 AND SR>111)OR SR=100 OR SR=101 THEN GOSUB 3000:U=1
9050 YL=YL-1
9060 PLOTXL,YL,"a":PLOTXL,YL+1," "
9070 UNTILYL<2 OR U=1
9080 PLOTXL,YL," ":U=0
9090 RETURN
```

4.5 LABYRINTHE

Une expérience scientifique vous fait remonter dans le temps. Vous revêtez les traits et le corps d'une personne de l'époque. La machine vous a emmené en Crète antique et vous êtes Thésée pour quelques heures. Hélas, Thésée est très courageux et a accepté une mission suicide. Il doit faire périr le Minotaure qui hante le labyrinthe construit par Dédale. Vous ne pouvez refuser car vous ne pouvez pas intervenir sur le cours du temps, et la légende raconte que Thésée est entré dans le labyrinthe... Soyez aussi courageux que Thésée et affrontez le monstre. Heureusement, vous bénéficiez de l'aide d'Ariane

qui vous a donné une bobine de fil que vous déroulerez derrière vous et qui vous permettra de retrouver rapidement la sortie du labyrinthe. Bonne chasse ! Gare au Minotaure qui est très rapide.

Pour vous déplacer à l'intérieur du labyrinthe, utilisez les quatre touches fléchées. Le jeu comporte trois niveaux selon la taille du labyrinthe, la densité des murs. Seul le niveau 1 permet de voir les murs. Si le labyrinthe vous semble sans sortie, ou pour vous arrêter, tapez [F] pour avoir la solution.

Structure du programme :

Lignes 20 à 120, présentation du jeu, choix du niveau de jeu.

Lignes 130 à 380, création du labyrinthe.

Lignes 400 à 500, déplacement dans le labyrinthe.

Lignes 510 à 570, initialisation de la deuxième phase du jeu si le test ligne 520 est faux.

Lignes 1000 à 1040 et lignes 5000 et 5010, redéfinition des caractères.

Lignes 2000 à 2340, sous-programme assurant les déplacements.

Lignes 3000 à 3040, tracé du labyrinthe complet (niveau 1).

Lignes 3500 à 3580, tracé des limites extérieures du labyrinthe (niveau 2 et 3).

Lignes 4000 à 4110, déplacement du minotaure - sa stratégie est de minimiser la distance entre lui et Thésée.

```
10 REM ***************
15 REM * LABYRINTHE *
16 REM ***************
20 DIM L(20,20)
30 GOSUB 1000
40 CLS:PAPER6:INK5
50 PRINT:PRINT:PRINTSPC(3)"*** LE LABYRINTHE DE MINOS ***"
60 PRINT:PRINT:PRINT"VOUS DEVEZ OCCIR LE MINOTAURE QUI EST"
65 PRINT:PRINT:PRINT"AU FOND DU LABYRINTHE"
70 PRINT:PRINT:PRINT"ATTENTION AUX PIEGES!"
80 PRINT:PRINT:PRINT"Quel niveau desirez vous?(1,2,3)"
90 GET A$
100 IF A$<"1" OR A$>"3" THEN GOTO 90
110 A=VAL(A$)
120 PRINT:PRINT"Attendez un instant"
125 REM *Limites du labyrinthe *
130 N=10+5*(A-1)
140 FOR I=1 TO N
150 L(I,1)=1
160 L(I,10)=1
170 NEXT I
180 FOR I=2 TO 9
190 L(1,I)=1
200 L(N,I)=1
210 NEXT I
220 REM * Creation du labyrinthe *
230 Z=0.18*SQR(A)
240 FOR I=2 TO N-1
250 FOR J=2 TO 9
260 L(I,J)=INT(RND(1)+Z-J/75)
270 NEXT J,I
280 REM * Position des Pie9es *
290 W=A
300 FOR I=1 TO W
310 L(INT(RND(1)*7+2),INT(RND(1)*7+2))=2
320 NEXT I
330 X=1:Y=INT(RND(1)*6+3):YD=Y:PH=1:F=94
340 L(X,Y)=0
350 XS=N:YS=INT(RND(1)*6+3)
351 FOR I=1 TO 3
352 L(XS-1,YS-2+I)=0
353 L(X+1,Y-2+I)=0
354 NEXT I
360 L(XS,YS)=0:CLS:PAPER4:INK6
370 IF A=1 THEN GOSUB 3000 ELSE GOSUB 3500
380 PLOT 3,3,"TEMPS:":PLOTXS+5,YS+8,96
390 REM * Exploration du labyrinthe *
400 A$=KEY$
405 IF A$="F" THEN GOTO 610
410 IF A$<CHR$(8) OR A$>CHR$(11) THEN GOTO 470
420 ON ASC(A$)-7 GOSUB 2000,2100,2200,2300
430 IF L(X,Y)=2 THEN EXPLODE:GOTO590
440 PLOTX+5,Y+8,F
450 IF X=XS AND Y=YS AND PH=1 THEN GOTO520
460 IF X=1 AND Y=YD AND PH=2 THEN GOTO 600
470 TP=TP+1
480 PLOT9,3,STR$(TP)
490 IF PH=2 THEN GOSUB 4000
500 GOTO400
510 REM * Phase 2 *
520 IFRND(1)<0.5THENPLOT2,21,"Le minotaure est mort,vous avez 9a9ne
":GOTO610
530 PLOT2,21,"Le minotaure est blesse,enfuyez-vous"
```

```
540 PLOT2,23,"avant qu'il ne vous ratrappe"
550 TM=TP:F=95
560 PH=2:XM=XS:YM=YS
570 GOTO 400
580 REM * Fin *
590 PLOT2,23,"Vous etes tombe dans un piege":GOTO610
600 PLOT2,23,"Vous avez echappe au minotaure,bravo"
610 GOSUB 3000
620 PLOT2,25,"VOULEZ VOUS AFFRONTER CE DEDALE A"
625 PLOT2,26,"NOUVEAU?(O/N)"
630 GET A$
640 IF A$="O" THEN RUN
650 IF A$="N" THEN END
660 GOTO 630
990 REM * Def caracteres *
1000 FOR I=46832 TO 46855
1010 READ J
1020 POKE I,J
1030 NEXT I
1040 RETURN
1990 REM * Deplacement *
2000 IF X=1 THEN RETURN
2010 IF L(X-1,Y)=1THEN RETURN
2020 PLOTX+5,Y+8,"."
2030 L(X,Y)=-1
2040 X=X-1
2050 RETURN
2100 IF X=N THEN RETURN
2110 IF L(X+1,Y)=1 THEN RETURN
2120 PLOTX+5,Y+8,"."
2130 L(X,Y)=-1
2140 X=X+1
2150 RETURN
2200 IF L(X,Y+1)=1 THEN RETURN
2210 PLOTX+5,Y+8,"."
2220 L(X,Y)=-1
2230 Y=Y+1
2240 RETURN
2300 IF L(X,Y-1)=1 THEN RETURN
2310 PLOTX+5,Y+8,"."
2320 L(X,Y)=-1
2330 Y=Y-1
2340 RETURN
2990 REM * Trace complet *
3000 FOR I=1 TO N
3010 FOR J=1 TO 10
3020 IF L(I,J)=1 THEN PLOTI+5,J+8,255
3030 NEXTJ,I
3040 RETURN
3490 REM * Trace du Perimetre *
3500 FOR I=1 TO N
3510 PLOTI+5,9,255
3520 PLOTI+5,18,255
3530 NEXTI
3540 FOR I=2 TO 9
3550 IF L(1,I)=1 THEN PLOT6,I+8,255
3560 IF L(N,I)=1 THEN PLOTN+5,I+8,255
3570 NEXT I
3580 RETURN
3990 REM * Depl minotaure *
4000 IF TP-TM<20 THEN RETURN
```

```
4010 E=0
4020 XT=XM-SGN(XM-X)*INT(RND(1)*2+0.5)
4030 YT=YM-SGN(YM-Y)
4040 E=E+1
4050 IF E=8 THEN GOTO 4070
4060 IF L(XT,YT)=1 THEN GOTO 4020
4065 PLOTXM+5,YM+8," "
4070 XM=XT
4080 YM=YT
4090 PLOTXM+5,YM+8,224
4100 IFXM=XANDYM=YTHENPLOT2,23,"Le minotaure vous a devore,desole":
GOTO 610
4110 RETURN
5000 DATA 12,12,62,45,44,14,18,19,12,12,31,45,13,28,22,50
5010 DATA 18,12,12,30,45,12,18,33
```

4.6 PILOTE DE CHASSE

"Alerte à toutes les unités ! Alerte à toutes les unités ! Des avions inconnus ont violé notre espace aérien. Identifiez-les, et abattez-les sans sommation, je répète. Identifiez-les, et abattez-les sans sommation".

Depuis le début des conflits qui opposent votre patrie à un pays expansionniste, les hautes instances militaires ont décidé de surveiller de près certaines installations ultra-secrètes. Cette surveillance comprend bien évidemment une surveillance terrestre, mais aussi une surveillance aérienne. C'est dans le cadre de cette mission que vous êtes aujourd'hui sur l'un des chasseurs les plus rapides à ce jour. Et il s'avère que cette surveillance n'est pas inutile puisque l'ennemi a en effet décidé de passer à l'action. Pour le salut de votre pays, détruisez les appareils espions.

Le programme proposé ici vous mettra dans la situation citée plus haut. Une mire située au centre de l'écran vous permettra de viser les appareils ennemis. Vous amènerez ceux-ci au centre de la mire grâce à vos commandes de déplacement (les quatre flèches). La barre d'espacement vous permettra de faire feu. Votre but est d'abattre le plus d'appareils possible. Attention à votre réserve de kérosène, et au nombre de missiles restants. Votre mérite sera comptabilisé par la boîte noire de votre appareil : un avion abattu incrémente votre score de 100 points, un avion sortant de votre champ de vue le décrémentera de 10 points. Bonne chasse.

Structure du programme :

Lignes 9 à 45, présentation.

Lignes 47 à 64, définition des caractères spéciaux.

Lignes 68 à 99, choix du niveau.

Lignes 99 à 172, initialisation.

Lignes 173 à 280, boucle principale.

Lignes 999 à 5060, déplacement mire.

Lignes 999 à 6020, tir roquette.

Lignes 6030 à 6110, destruction avion.

Lignes 7000 à 7080, déplacement avion.

Lignes 7999 à 8070, affichage indicateurs.

Lignes 8099 à 8160, affichage kérosène.

Lignes 8199 à 8295, affichage score.

Lignes 8299 à 8340, affichage avions touchés.

Lignes 8399 à 8430, affichage stock roquettes.

Lignes 8999 à 9040, affichage trajectoire roquettes.

```
9 REM  ********************
10 REM * PILOTE DE CHASSE *
11 REM ********************
20 CLS
25 PAPER0 :INK2
30 PRINT:PRINTSPC(10)"PILOTE DE CHASSE"
35 PRINT:PRINT:PRINTSPC(5)"ABATTEZ L'ESCADRILLE INCONNUE"
40 PRINT:PRINTSPC(5)"QUI VIOLE VOTRE ESPACE AERIEN"
45 PRINT:PRINT:WAIT100
47 REM * Definition caracteres *
48 FOR J=1 TO 4
50 READ K
52 FOR I=0 TO 7
54 READL
56 POKEK+I,L
58 NEXT I
59 NEXT J
60 DATA46840,0,1,1,3,15,60,4,0
62 DATA46592,0,0,0,32,56,30,16,0
63 DATA46848,59,42,63,51,45,45,51,63
```

```
64 DATA46832,4,14,14,14,14,4,10,0
68 REM * Choix du niveau *
69 TEXT
70 PRINT"NIVEAU DE DIFFICULTE"
71 PRINT"  1:FAIBLE"
72 PRINT"  2:MOYEN"
73 PRINT"  3:FORT"
74 INPUT"QUEL NIVEAU CHOISISSEZ VOUS";NI$
75 IF ASC(NI$)<49 OR ASC(NI$)>51 THEN GOTO 74
76 NI=VAL(NI$)
99 REM * Initialisations *
100 HIRES
105 PRINTCHR$(17)
110 PAPER6 :INK1
120 CURSET110,90,3
130 FILL20,1,19
140 CURSET130,90,3
150 FILL20,1,22
160 REM
170 X=115 :Y=180 :KE=1000 :PNT=0 :MUN=20 :XE=80 :YE=80 :AV=0
171 GOSUB 8000
172 GOSUB 5020
173 REM * Boucle Principale *
174 REPEAT
175 GOSUB 8100
180 M$=KEY$
190 IF M$>=CHR$(8) AND M$<=CHR$(11) THEN GOSUB 5000
200 IF M$=CHR$(32) THEN MUN=MUN-1 :GOSUB 6000
210 IF RND(1)<NI/10 THEN GOSUB 7000
220 KE=KE-1
230 UNTIL KE=0
240 PRINT"MANQUE DE CARBURANT.RENTREZ A LA BASE"
245 PING
250 WAIT500
260 INPUT"VOULEZ VOUS REPARTIR (O/N)";R$
270 IF R$="O" THEN GOTO 69
280 END
999 REM * Deplacement mire *
1000 X=X+4
1010 IF X>220 THEN X=115:Y=20:INK0:PNT=PNT-10
1020 GOTO 5020
2000 X=X-4
2010 IF X<20 THEN X=115:Y=180:INK1:PNT=PNT-10
2020 GOTO 5020
3000 Y=Y-4
3010 IF Y<20 THEN X=30:Y=95:INK4:PNT=PNT-10
3020 GOTO 5020
4000 Y=Y+4
4010 IF Y>180 THEN X=210:Y=95:INK5:PNT=PNT-10
4020 GOTO 5020
5000 XE=X :YE=Y
5010 ON ASC(M$)-7 GOTO1000,2000,3000,4000
5020 GOSUB 8200
5025 CURSETXE,YE,0
5030 CHAR64,0,0
5035 CURSET XE-6,YE,3
5037 CHAR95,0,0
5040 CURSETX,Y,3
5050 CHAR64,0,1
5055 CURSETX-6,Y,3
5057 CHAR95,0,1
```

```
5060 RETURN
5999 REM * Tir roquette *
6000 ZAP:WE=1:GOSUB 9000
6005 WAIT10 :WE=0 :GOSUB 9000
6007 GOSUB 8400
6010 IF ABS(X-115)<9-NI AND ABS(Y-95)<9-NI THEN EXPLODE :GOTO6030
6015 IF MUN=0 THEN GOTO 6090
6020 RETURN
6030 REM * Destruction avion *
6031 P=1
6032 FOR I=1 TO 10
6034 CURSET110,90,3
6035 FILL20,1,16+P
6036 IF P=1 THEN P=3 :GOTO 6038
6037 P=1
6038 NEXT I
6040 PNT=PNT+100
6045 GOSUB 8200
6047 AV=AV+1 :GOSUB 8300
6050 IF MUN=0 THEN GOTO 6090
6060 XE=X :YE=Y
6070 X=115 :Y=20+160*INT(RND(1)*2)
6080 GOTO 5020
6090 PRINT:PRINT"PLUS DE MUNITION.RENTREZ A VOTRE BASE"
6100 PING
6110 GOTO 250
6999 REM * Deplacement avion *
7000 XE=X :YE=Y
7005 S=SGN(X-120)
7010 X=X+INT(10*RND(1))*S
7020 N=SGN(Y-100)
7030 Y=Y+INT(10*RND(1))*N
7040 IF X>220 THEN X=30:Y=95:PNT=PNT-10
7050 IF X<20 THEN X=115:Y=180:PNT=PNT-10
7060 IF Y>180 THEN X=210:Y=95:PNT=PNT-10
7070 IF Y<20 THEN X=115:Y=20:PNT=PNT-10
7080 GOTO 5020
7999 REM * Affichage indicateurs *
8000 POKE49042,96
8010 POKE49050,95
8020 POKE49051,64
8030 POKE49060,94
8040 FOR I=1 TO 3
8050 POKE48990+I*40,124
8060 NEXT I
8062 GOSUB 8400
8065 GOSUB 8200
8066 GOSUB 8300
8070 RETURN
8099 REM * Affichage kerosene *
8100 E(1)=INT(KE/100)
8110 E(2)=INT(KE/10)-10*E(1)
8120 E(3)=KE-10*E(2)-100*E(1)
8130 FOR I=1 TO 3
8140 POKE49043+I,E(I)+48
8150 NEXT I
8160 RETURN
8199 REM * Affichage score *
8200 AP=ABS(PNT)
8210 E(1)=0
8220 E(4)=INT(AP/1000)
```

```
8230 E(3)=INT(AP/100)-10*E(4)
8240 E(2)=INT(AP/10)-10*E(3)-100*E(4)
8250 IF PNT<0 THEN E(5)=-3:GOTO 8270
8260 E(5)=-16
8270 FOR I=1 TO 5
8280 POKE49077-I,E(I)+48
8290 NEXT I
8295 RETURN
8299 REM * Affichage avions touches *
8300 E(1)=INT(AV/10)
8310 E(2)=AV-10*E(1)
8320 FOR I=1 TO 2
8330 POKE49052+I,E(I)+48
8335 NEXT I
8340 RETURN
8399 REM * Affichage stock roquettes *
8400 E(1)=INT(MUN/10)
8410 E(2)=MUN-10*E(1)
8420 POKE49062,E(1)+48
8430 POKE49063,E(2)+48
8999 REM * Trajectoire roquettes *
9000 CURSET115,100,3
9010 DRAW-60,90,WE
9020 CURSET115,100,3
9030 DRAW60,90,WE
9040 RETURN
```

CHAPITRE **5**

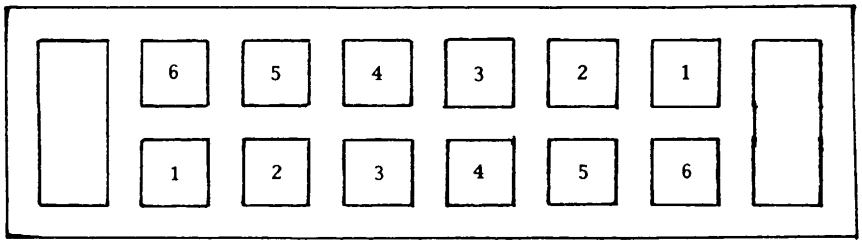
Jeux de réflexion

5.1 AWELÉ

Découvrez l'univers des jeux de réflexion africains. Passionnez-vous pour l'Awélé, jeu connu aux Philippines sous le nom de Côkajon, en Java sous le nom de Pakân Dakon, à Ceylan sous le nom de Câka, au Brésil sous le nom de Adi, et dans bien d'autres pays encore. Partagez l'engouement de ces peuples à cet exercice d'esprit.

Imaginez-vous en Afrique, dans un village de paillotes, le soleil arrose la savane de ses dards acérés. A l'ombre d'un baobab, vous affrontez l'un des plus grands stratèges de la région, le sorcier du village. Vos gestes sont rythmés par les tam-tams qui sonnent au loin, à ces battements se mélangent parfois le rugissement des lions, le barissement des éléphants... L'enjeu : devenir dignitaire du village ou subir les maléfices du sorcier. Alors n'hésitez pas, prenez votre destinée en main, et soyez le plus fort.

Le matériel du jeu traditionnel se compose de 48 graines et d'un plateau en bois dans lequel sont creusées 14 cases. Ces cases sont réparties de la manière suivante.



Cette aire de jeu ne nécessite aucun matériel particulier et est souvent tracée à même le sol. Pour plus de commodité nous numérotons les deux rangées de petites cases et indiquons chacune de ces rangées par Nord ou Sud.

Les règles du jeu sont aussi multiples que les noms sous lesquels l'Awélé est connu. Nous adopterons ici les règles suivantes :

- les 48 graines sont réparties au départ, de manière équivalente dans chacune des cases numérotées.

- à chacun des adversaires sont attribuées une rangée et une maison, celle-ci étant la case située au bout de sa rangée.

- chaque joueur prend à son tour les graines d'une case quelconque de son côté du tableau, selon son libre choix (cases 1 à 6).

- il les répartit alors une à une dans les cases suivantes dans le sens inverse des aiguilles d'une montre (il "sème").

- si son adversaire avait joué au coup précédent et que sa dernière graine tombe dans sa maison alors il rejoue.

- si sa dernière graine tombe dans une case vide et que la case vis-à-vis de l'autre rangée est occupée, alors les graines de celle-ci et la dernière graine sont mises dans sa maison (remarque : vis-à-vis ne veut pas dire portant le même numéro, le vis-à-vis de 1 est 6, de 2, 5...).

- s'il arrive qu'une des rangées soit vide, le propriétaire de celle-ci passera son tour jusqu'à ce que son adversaire lui fournisse des graines à déplacer.

- le jeu est fini lorsque toutes les cases numérotées sont vides, le gagnant est celui qui a le plus de graines dans sa maison.

Le programme que nous vous proposons suit mot pour mot les règles énoncées ci-dessus. Le côté Nord est attribué à la machine et le côté Sud vous est attribué. Vous possédez le trait au départ. Les commandes utilisables sont seulement les numéros 1 à 6 qui indiqueront à l'ordinateur la case que vous voulez vider. Vous noterez la présentation originale de ce jeu, les graines sont remplacées par des petits hommes, et les cases maison par des paillotes. Vous avez maintenant en main toutes les informations : alors "Bon jeu".

Sructure du programme :

Lignes 10 à 120, initialisations.

Lignes 130 à 160, gestion du tracé du décor et de la mise en place des petits hommes.

Lignes 170 à 330, arbitre de la partie.

Lignes 1000 à 1070, acquisition et validation de la commande clavier.

Lignes 2000 à 2100, test du contenu des rangées.

Lignes 3000 à 3090, déplacement des petits hommes.

Lignes 3100 à 3190, transfert des petits hommes lorsque la case vis-à-vis vient de recevoir la dernière graine du déplacement précédent.

Lignes 4000 à 4170, stratégie de l'ordinateur.

Lignes 5000 à 5130, affichage décor.

Lignes 6000 à 6080, dessin des paillotes.

Lignes 7000 à 7070, numérotation des cases de chaque rangée.

Lignes 8010 à 8050, effacement petit homme côté Nord.

Lignes 8080 à 8140, affichage petit homme.

Lignes 8200 à 8250, effacement petit homme côté Sud.

Lignes 8300 à 8470, affichage du nombre des petits hommes dans les paillotes.

Lignes 8500 à 8520, décalage curseur pour affichage en double rang.

Lignes 9000 à 9180, mise en place initiale des petits hommes.

```
9 REM  *********
10 REM * AWELE *
11 REM *********
20 DIM B(14),G(14),X0(14),Y0(14),XC(14),YC(14)
30 READK
40 FOR I=0 TO 7
50 READL
60 POKEK+I,L
70 NEXT I
80 DATA46848,64,64,64,66,71,66,69,69
100 REM
120 N=1 :E=0 :C=0 :DRAP=0 :SUM=0 :VID=0 :COT=0
130 GOSUB 5000
140 FOR I=1 TO 14 :B(I)=4 :NEXT I
150 B(7)=0 :B(14)=0
160 GOSUB 9000
170 IFVID=1THENPRINT:PRINT"COTE VIDE.VOUS PERDEZ VOTRE TOUR":WAIT20
0:GOTO 215
180 GOSUB 1000
190 IF E=0 THEN GOTO 260
195 IF VID=1 THEN GOTO 170
200 IF M=H THEN PRINT :PRINT"REJOUEZ" :GOSUB 1000
210 IF E=0 THEN GOTO 260
215 IF COT=1 THEN PRINT :PRINT"JE PERDS MON TOUR":WAIT 100 :GOTO 25
0
220 PRINT: PRINT"A MON TOUR DE JOUER" :GOSUB 4000
230 IF E=0 THEN GOTO 260
235 WAIT 300
240 IF M=H THEN PRINT :PRINT"JE REJOUE" :GOSUB 4000
250 IF E>0 THEN GOTO 170
260 PRINT:PRINT"FIN DU JEU" :WAIT 500
270 D=B(7)-B(14)
280 IF D<0 THEN PRINT :PRINT"J'AI GAGNE ";-D;" POINTS":GOTO 310
290 IF D=0 THEN PRINT :PRINT"PARTIE NULLE" :GOTO 310
300 PRINT :PRINT"VOUS AVEZ GAGNE ";D;" POINTS"
310 WAIT 500 :PRINT :PRINT :INPUT"VOULEZ VOUS REJOUER?(O/N)";R$
320 IF R$="O"THEN GOTO 30
330 END
1000 PRINT:PRINT:INPUT"CASE A VIDER ";M
```

```
1010 IF M<7 AND M>0 THEN GOTO 1040
1020 PRINT:PRINT"IL N'Y A PAS DE CASE NUMERO ";M
1030 WAIT 300 :GOTO 1000
1040 IF B(M)=0 THEN PRINT:PRINT"CASE VIDE" :GOTO 1030
1050 H=7
1060 REM
1070 REM
2000 K=M :GOSUB 3000
2020 VID=1 :COT=1
2030 FOR I=1 TO 6
2040 IF B(I)<>0 THEN VID=0 :I=6
2050 NEXT I
2060 FOR I=8 TO 13
2070 IF B(I)<>0 THEN COT=0 :I=13
2080 NEXT I
2090 E=1-VID*COT
2100 RETURN
3000 P=B(M) :B(M)=0
3010 FOR P=P TO 1 STEP -1
3020 IF DRAP=0 THEN GOSUB 8010
3030 M=M+1
3040 IF M>14 THEN M=M-14
3050 IF (M=7 OR M=14) ANDDRAP=0 THEN SUM=1 :H=M :GOSUB 8300
3060 IF DRAP=0 AND M<>7 ANDM<>14 THEN GOSUB 8080
3065 B(M)=B(M)+1
3070 NEXT P
3080 IF B(M)=1 AND M<>7 AND M<>14 AND B(14-M)<>0 THEN GOTO 3100
3090 RETURN
3100 IF K<7 THEN H=7 ELSE H=14
3110 K=M :IF DRAP=0 THEN GOSUB 8010
3120 SUM=1 :IF DRAP =0 THEN GOSUB 8300
3125 B(H)=B(H)+1
3130 P=B(14-M) :K=14-K
3135 IF DRAP=1 THEN GOTO 3180
3140 FOR P=P TO 1 STEP -1
3150 GOSUB 8010
3160 SUM=1 :GOSUB 8300
3165 B(H)=B(H)+1
3170 NEXT P
3180 B(M)=0 :B(14-M)=0
3190 RETURN
4000 D=-40 :H=14
4010 FOR I=1 TO 14 :G(I)=B(I) :NEXT I
4015 DRAP=1
4020 FOR J=8 TO 13 :IF B(J)=0 THEN GOTO 4140
4030 Q=0 :M=J :GOSUB 3000
```

```
4040 FOR I=1 TO 6:IF B(I)=0 THEN GOTO 4100
4050 L=B(I)+I :R=0
4060 IF L>14 THEN L=L-14 :R=1 :GOTO 4060
4070 IF B(L)=0 AND L<>7 AND L<>14 THEN R=B(14-L)+R
4090 IF R>Q THEN Q=R
4100 NEXT I
4110 Q=B(14)-B(7)
4120 FOR I=1 TO 14 :B(I)=G(I) :NEXT I
4130 IF Q>D THEN A=J :D=Q
4140 NEXT J
4150 DRAP=0
4160 M=A :PRINT :PRINT"CASE VIDEE:";M-7 :WAIT10 :GOTO 2000
4170 REM
5000 HIRES
5010 PAPER3 :INK0
5020 CURSET15,90,3
5030 GOSUB 6000
5040 CURSET195,105,3
5050 GOSUB 6000
5060 X=30 :Y=15
5070 CURSET45,130,3
5080 GOSUB 7000
5090 X=-X
5100 CURSET190,70,3
5110 GOSUB 7000
5120 RETURN
5130 REM
6000 DRAW30,0,1
6010 DRAW-15,-10,1
6020 DRAW-15,10,1
6030 CURMOV5,0,3
6040 DRAW0,15,1
6050 DRAW20,0,1
6060 DRAW0,-15,1
6070 RETURN
6080 REM
7000 FOR I=1 TO 6
7010 A=48+I
7020 CHARA,0,1
7030 IF I=4 THEN Y=-Y
7040 CURMOVX,Y,3
7050 NEXT I
7070 RETURN
8010 IF K<7 THEN GOTO  8200
8015 IF XC(K)-X0(K)<=0 THEN XC(K)=X0(K)+30 :YC(K)=YC(K)-8
8020 XA=XC(K)-5 :YA=YC(K)
```

```
8030 CURSETXA,YA,3
8040 CHAR96,0,0
8050 XC(K)=XA :RETURN
8080 XA=XC(M) :YA=YC(M)
8090 CURSETXA,YA,3
8100 CHAR96,0,1
8110 IF M>7 THEN GOTO 8500
8120 XC(M)=XC(M)-5
8130 IF X0(M)-XC(M)>25 THEN XC(M)=X0(M) :YC(M)=YC(M)-8
8140 RETURN
8200 IF X0(K)-XC(K)<=0 THENXC(K)=X0(K)-30:YC(K)=YC(K)+8
8210 XA=XC(K)+5 :YA=YC(K)
8220 CURSETXA,YA,3
8230 CHAR96,0,0
8240 XC(K)=XA :RETURN
8250 REM
8300 SIG=B(H) :AT=0
8310 GOSUB 8400
8320 SIG=SIG+SUM :AT=1
8330 GOSUB 8400
8340 RETURN
8350 REM
8400 DIZ=INT(SIG/10)
8410 UNIT=SIG-10*DIZ
8420 CURSETX0(H),Y0(H),3
8430 CHAR48+UNIT,0,AT
8440 CURMOV-6,0,3
8450 CHAR48+DIZ,0,AT
8460 RETURN
8470 REM
8500 XC(M)=XC(M)+5
8510 IF XC(M)-X0(M)>+25 THEN XC(M)=X0(M) :YC(M)=YC(M)+8
8520 RETURN
9000 DATA55,120,85,135,115,150,145,165,175,150,205,135,210,110
9010 DATA180,80,150,65,120,50,90,35,60,50,30,65,30,95
9020 FOR I=1 TO 14 :READ X0(I) :READ Y0(I) :NEXT I
9025 RESTORE
9030 FOR I=1 TO 14 :XC(I)=X0(I) :YC(I)=Y0(I) :NEXT I
9040 FOR I=1 TO 6 :DEC=-5
9050 GOSUB 9110 :NEXT I
9060 FOR I=8 TO 13 :DEC=5
9070 GOSUB 9110 :NEXT I
9080 H=7 :GOSUB 8300
9090 H=14 :GOSUB 8300
9095 RETURN
9100 RETURN
```

```
9110 FOR J=1 TO 4
9120 XA=XC(I) :YA=YC(I)
9130 CURSETXA,YA,3
9140 CHAR96,0,1
9150 XC(I)=XC(I)+DEC
9160 NEXT J
9170 RETURN
9180 END
```

5.2 CAVALIER

Comment vous échapper de ce château mystérieux où votre curiosité vous a mené ? Heureusement pour vous, le propriétaire de ces lieux, un ermite enchanteur, vous laisse une alternative pour vous en sortir. Il vous a remis un échiquier et un cavalier magique. Chaque case de cet échiquier représente une grille que vous aurez à franchir. Pour pouvoir passer les soixante quatre grilles, il vous faut obtenir la clef de chacune d'elles. Le moyen d'obtenir ces clefs est le suivant : vous choisissez une case de départ sur l'échiquier, vous y placez votre cavalier, puis vous le déplacez conformément au déplacement d'un cavalier d'un jeu d'échecs. Chaque case parcourue vous donnera une clef, il vous faudra évidemment parcourir tout l'échiquier pour pouvoir retrouver votre liberté.

Si vous voulez voir une solution, ou lorsque votre cavalier n'a plus de liberté de déplacement, ou encore si vous avez trouvé une solution, le magicien vous montrera sa solution. Vous pourrez alors tenter de trouver une autre solution.

Le programme proposé ici vous fera chercher une solution du problème précédent à partir d'une case que vous aurez choisie. Une solution vous sera proposée dans les cas cités plus haut.

Les commandes à utiliser :

- entrée des coordonnées de la case d'arrivée

L C RETURN

L étant une lettre de A à H et C un chiffre de 1 à 8,

- demande de solution de l'ordinateur

S

Sructure du programme :

Lignes 10 à 150, initialisations.

Lignes 150 à 220, moniteur principal.

Lignes 1000 à 1220, tracé de l'échiquier.

Lignes 2000 à 2270, acquisition case à parcourir. Validation déplacement. Affichage cavalier. Test cavalier bloqué ou non.

Lignes 3000 à 3110, validation commande entrée.

Lignes 4000 à 4200, recherche et affichage d'une solution.

Lignes 5000 à 5560, affichage du numéro du coup joué.

Lignes 6000 à 6570, affichage du cavalier.

Lignes 8000 à 8050, recherche des cases accessibles à partir d'une position.

Lignes 9000 à 9070, définition des caractères pour dessin du cavalier.

```
9 REM  ************
10 REM * CAVALIER *
11 REM ************
90 PAPER2 :INK5
100 REM * Init deplacements *
110 DIM H(8),V(8),T(8,8)
120 DATA1,2,2,1,2,-1,1,-2,-1,-2,-2,-1,-2,1,-1,2
130 FOR I=1 TO 8
140 READ H(I),V(I)
150 NEXT I
155 GOSUB9000
159 REM * Saut sur Partie joueur *
160 GOSUB 1000
170 GOSUB 2000
179 REM * Saut sur Partie computer *
180 PRINT :PRINT"VOILA UNE SOLUTION"
185 FORI=1TO8:FORJ=1TO8:T(I,J)=0:NEXTJ,I
```

```
190 GOSUB 4000
200 PRINT :INPUT"VOULEZ VOUS RECOMMENCER(O/N)";R$
210 IF R$="O" THEN RUN110
220 END
1000 REM * Dessin echiquier *
1005 HIRES
1010 PAPER2 :INK1
1020 CURSET24,6,0
1030 C=19.5 :D=3.5
1040 FOR I=1 TO 8
1050 FOR J=1 TO 8
1060 FILL23,1,C+D
1070 D=-D
1080 NEXT J
1090 CURSET24*(I+1),6,0
1095 D=-D
1100 NEXT I
1110 FILL184,1,18
1120 CURSET18,14,0
1130 FOR I=8 TO 1 STEP -1
1140 CHAR48+I,0,1
1145 IF I=1 THEN GOTO 1160
1150 CURMOV0,23,0
1160 NEXT I
1170 CURSET32,191,0
1180 FOR I=1 TO 8
1190 CHAR64+I,0,1
1200 CURMOV24,0,0
1210 NEXT I
1220 RETURN
2000 REM * Partie Joueur *
2005 K=0
2010 INPUT"CASE DE DEPART(VER/HOR)";X$
2020 GOSUB 3005
2030 X0=X :Y0=Y
2040 T(X,Y)=0
2050 GOSUB 6000
2060 REPEAT
2070 T(X,Y)=T(X,Y)+1
2080 INPUT"CASE D'ARRIVEE";X$
2085 K=K+1
2090 GOSUB 3000
2100 IF X$="S" THEN GOTO 180
2210 GOSUB 5000
2220 DP=0
2230 GOSUB 8000
2240 UNTIL DP=0
2250 IF K=63 THEN PRINT"BRAVO" :EXPLODE:WAIT100 :GOTO2270
2260 PRINT :PRINT"VOUS AVEZ PERDU" :WAIT500
2270 RETURN
2999 REM * Decodage coordonnees *
3000 IF X$="S" THEN GOTO 3110
3005 Y=ASC(RIGHT$(X$,1))-48
3010 X=ASC(LEFT$(X$,1))-64
3015 IFK=0 THEN GOTO 3060
3020 XH=X-HO :YV=Y-VE
3030 FOR J=1 TO 8
3040 IFXH=H(J)THENIFYV=V(J)THENGOTO3060
3050 NEXT J
3051 ZAP
3055 GOTO 3090
3056 PING
```

```
3060 IF X<1 OR X>8 OR Y<1 OR Y>8 THEN GOTO 3090
3080 IF T(X,Y)=0 THEN GOTO 3100
3090 INPUT"CODE INVALIDE.ENTREZ(VER/HOR)";X$
3095 GOTO 3000
3100 HO=X :VE=Y
3110 RETURN
4000 REM * Partie computer *
4002 GOSUB 1000
4005 X=X0 :Y=Y0
4007 W=30+(X-1)*24 :Z=10+(8-Y)*23
4010 T(X0,Y0)=0 :K=0 :DP=8
4020 REPEAT
4030 T(X,Y)=T(X,Y)+1
4040 MIN=8
4050 FOR I=1 TO 8
4060 A=X+H(I) :B=Y+V(I) :DP=0
4070 IF A<1 OR A>8 OR B<1 OR B>8 THEN GOTO 4130
4080 IF T(A,B)<>0 THEN GOTO 4130
4090 FORJ=1TO8
4095 C=A+H(J):D=B+V(J)
4100 IF C>0ANDC<9ANDD>0ANDD<9 THEN IF T(C,D)=0 THEN DP=DP+1:CALL#FB
03
4110 NEXT J
4120 IF DP<=MIN THEN MIN=DP :DH=A :DV=B
4130 NEXT I
4140 X=DH :Y=DV
4150 K=K+1
4160 GOSUB 5000
4180 UNTIL MIN=0
4200 RETURN
4999 REM * Numerotage coup joue *
5000 CURSETW,Z,3
5010 Q=0 :GOSUB 6500
5500 REM
5510 CURSETW,Z+4,3
5520 DIZ=INT(K/10)
5530 UNIT=K-DIZ*10
5540 CHAR48+DIZ,0,1
5550 CURMOV6,0,3
5560 CHAR48+UNIT,0,1
6000 REM
6010 W=30+(X-1)*24 :Z=10+(8-Y)*23
6020 CURSETW,Z,3
6030 Q=1 :GOSUB 6500
6040 RETURN
6500 CHAR64,0,Q
6510 CURMOV6,0,3
6520 CHAR95,0,Q
6530 CURMOV-6,8,3
6540 CHAR126,0,Q
6550 CURMOV6,0,3
6560 CHAR96,0,Q
6570 RETURN
7999 REM * Recherche coups possibles *
8000 REM
8010 FOR J=1 TO 8
8020 C=X+H(J) :D=Y+V(J)
8030 IF C>0ANDC<9ANDD>0ANDD<9 THEN IF T(C,D)=0 THEN DP=DP+1
8040 NEXT J
8050 RETURN
8999 REM * Definition caracteres *
9000 FORI=1TO4
```

```
9010 READK
9020 FORJ=0TO7
9030 READL
9040 POKEK+J,L
9050 NEXTJ,I
9060 DATA46592,64,67,69,79,82,126,100,124
9062 DATA46840,96,80,72,84,66,97,93,83
9064 DATA46848,72,72,72,120,80,78,65,127
9066 DATA47088,100,124,100,127,66,92,96,127
9070 RETURN
```

5.3 MASTERMIND

Mastermind est un jeu de réflexion inventé par un mathématicien : Mordechai Meirovich. Deux partis sont en présence, le codeur et le décodeur. Le rôle du codeur est d'élaborer une combinaison de quatre fiches, et de donner des informations sur cette combinaison. Le rôle du décodeur est de trouver la combinaison élaborée en s'aidant des informations données par le codeur. Les informations sont de deux types : nombre de bonnes couleurs à la bonne place, et nombre de bonnes couleurs à la mauvaise place. Le nombre de tentatives du décodeur est limité à dix.

Le programme proposé joue le rôle du codeur.

Six couleurs sont possibles :

Rouge	: R	Bleu	: B
Vert	: V	Magenta	: M
Jaune	: J	Cyan	: C

La combinaison élaborée est formée de quatre fiches, chacune de ces fiches pouvant avoir une des six couleurs précédentes. Les combinaisons multiples sont donc possibles. Pour entrer votre combinaison, vous devez procéder de la manière suivante :

Tapez

$$C_1\ C_2\ C_3\ C_4 \text{ puis [RETURN]}$$

C_1 = code de couleur

Les indications vous seront fournies dans les colonnes C et P :

- C : bonne couleur, mauvaise place,

- P : bonne couleur, bonne place.

Lorsque vous aurez trouvé la bonne combinaison ou utilisé vos dix tentatives, vous verrez s'afficher le bon code sur la première ligne du tableau. Et maintenant à vous de jouer.

REMARQUE : si vous entrez

$$C_1\ C_2\ C_3 \ldots\ C_{n-3}\ C_{n-2}\ C_{n-1}$$

C_i = code de couleur

seuls les quatre derniers codes sont pris en compte.

Sructure du programme :

Lignes 10 à 110, commentaires. Appel au tracé du décor.

Lignes 120 à 140, élaboration du code secret.

Lignes 147 à 148, affichage de C et P.

Lignes 150 à 310, arbitre de la partie.

Lignes 1000 à 1220, tracé du décor.

Lignes 2000 à 2050, affichage du code secret.

Lignes 3000 à 3130, lecture et validation de la combinaison entrée au clavier.

Lignes 4000 à 4090, table de traduction des codes couleur.

Lignes 5000 à 5090, affichage de la combinaison entrée au clavier.

Lignes 6000 à 6160, recherche des indications à visualiser.

Lignes 7000 à 7060, affichage indications.

Lignes 8000 à 8090, effacement du contenu du tableau.

```
9 REM  ***************
10 REM * MASTERMIND *
11 REM ***************
100 REM
110 GOSUB 1000
120 FOR I=1 TO 4
130 COMB(I)=INT(6*RND(1)+1)
140 NEXT I
145 B=0
147 W=19:B=32:Y=15
148 GOSUB7000
150 FOR K=1 TO 10
160 GOSUB 3000
170 GOSUB 5000
180 GOSUB 6000
190 GOSUB 7000
200 IF B=4 THEN GOTO 260
210 NEXT K
220 PRINT :PRINT"VOUS AVEZ PERDU" :WAIT500
260 PRINT :PRINT"BONNE COMBINAISON "
270 X=90 :Y=15
280 GOSUB 2000
290 PRINT:INPUT"VOULEZ VOUS REJOUER?(O/N)";R$
300 IF R$="O" THEN GOTO 8000
310 END
1000 HIRES
1010 INK2
1020 FOR I=102 TO 222 STEP 18
1030 CURSETI,10,3
1040 DRAW0,165,1
1050 NEXT I
1060 FOR I=10 TO 175 STEP 15
```

```
1070 CURSET102,I,3
1080 DRAW108,0,1
1090 NEXT I
1100 FORI=0TO5
1110 READCL
1120 CURSET20,15*(I+1),3
1130 FILL5,1,17+I
1140 CURSET26,15*(I+1),3
1150 FILL5,1,16
1160 CURSET38,15*(I+1),3
1170 CHARCL,0,1
1180 NEXTI
1190 DATA82,86,74,66,77,67
1200 RETURN
1220 RETURN
2000 REM
2010 FOR I=1 TO 4
2020 C(I)=COMB(I)
2030 NEXT I
2040 GOSUB 5020
2050 RETURN
3000 REM
3010 PRINT :PRINT :INPUT"ENTREZ VOTRE COMBINAISON";CHOIX$
3020 FOR I=4 TO 1 STEP -1
3040 C$(I)=RIGHT$(CHOIX$,5-I)
3050 C$(I)=LEFT$(C$(I),1)
3060 NEXT I
3070 FOR I=1 TO 4
3080 COD$=C$(I)
3090 GOSUB 4000
3100 IF TRAD=0 THEN PRINT :PRINT"CODE INVALIDE":WAIT 300 :GOTO 3000
3110 C(I)=TRAD
3120 NEXT I
3130 RETURN
4000 REM
4010 IF COD$="R" THEN TRAD=1 :GOTO 4090
4020 IF COD$="V" THEN TRAD=2 :GOTO 4090
4030 IF COD$="J" THEN TRAD=3 :GOTO 4090
4040 IF COD$="B" THEN TRAD=4:GOTO 4090
4050 IF COD$="M" THEN TRAD=5 :GOTO 4090
4060 IF COD$="C" THEN TRAD=6 :GOTO 4090
4070 TRAD=0
4080 RETURN
4090 RETURN
5000 REM
5010 Y=15+15*K
```

```
5020 FOR I=1 TO 4
5030 X=108+18*(I-1)
5040 CURSETX,Y,3
5050 FILL5,1,16+C(I)
5060 CURSETX+6,Y,3
5070 FILL5,1,16
5080 NEXT I
5090 RETURN
6000 REM
6005 B=0 :W=0
6010 FOR  I=1 TO 4
6020 IF C(I)=COMB(I) THEN B=B+1
6030 NEXT I
6040 FOR I=1 TO 4
6060 R(I)=COMB(I)
6070 NEXT I
6080 FOR I=1 TO 4
6090 FOR J=1 TO 4
6100 IF C(I)<>R(J) THEN GOTO 6130
6120 W=W+1 :R(J)=0
6125 GOTO6140
6130 NEXT J
6140 NEXT I
6150 W=W-B
6160 RETURN
7000 REM
7010 AT=1 :CAR=0
7020 CURSET180,Y-1,3
7030 CHAR48+W,CAR,AT
7040 CURSET198,Y-1,3
7050 CHAR48+B,CAR,AT
7060 RETURN
8000 REM
8010 FOR K=0 TO 10
8020 FOR I=1 TO 4
8030 C(I)=0
8040 NEXT I
8050 GOSUB 5000
8060 W=79 :B=W :AT=0 :CAR=0
8070 GOSUB 7020
8080 NEXT K
8090 GOTO 120
```

5.4 PENDU

Voici un grand classique des jeux de devinettes. Un mot est choisi par votre Oric dans son dictionnaire. Vous devez le découvrir en un minimum de coups. Pour vous aider, le nombre de lettres du mot vous est donné par des étoiles recouvrant les lettres. Vous devez proposer une lettre et si elle fait partie du mot secret, elle apparaîtra à sa place dans le mot. Mais si elle n'appartient pas au mot ou si vous l'avez déjà proposée, une partie du pendu se dessinera sur l'écran. D'abord le socle et la potence puis la corde, la tête, le corps, les bras et enfin les jambes. Quand la jambe gauche se dessinera, vous aurez perdu. L'Oric vous donnera la solution. Attention, soyez "fair-play" et ne faites pas exprès de perdre pour voir se dessiner le pendu !

Structure du programme :

Lignes 20 à 190, tracés des décors et présentation du jeu.

Lignes 220 à 270, recherche du mot dans le dictionnaire.

Lignes 300 à 330, effacement du pendu.

Lignes 350 à 380, entrée de la proposition de lettre.

Lignes 390 à 410, tri des lettres déjà jouées.

Lignes 470 à 540, testent si lettre valable et si mot trouvé.

Lignes 1000 à 1050, sous-programme de traitement de lettre trouvée.

Lignes 2000 à 3070, tracé du pendu.

Lignes 4000 et suivantes, dictionnaire.

REMARQUE IMPORTANTE :

Le dictionnaire est écrit en DATA. La première donnée à la ligne 4000 est le nombre de mots contenus dans le dictionnaire. Les autres données sont les mots. Le nombre de mots par ligne de DATA est sans importance. Evitez toutefois de dépasser la contenance de la ligne. Ecrivez les mots entre guillemets. Vous pouvez donc modifier et allonger le dictionnaire à votre guise. Créez-vous plusieurs dictionnaires contenant un vocabulaire spécialisé ou de niveaux différents et enregistrez-les sur cassette.

```
5 REM  ****************
10 REM * JEU DU PENDU *
15 REM ****************
20 PAPER 0:INK 5
30 HIRES:INK3
40 E=15
50 E$="JEU DU PENDU"
60 FOR I=1 TO 12
70 CURSET E+I*8,20,0
80 CHAR ASC(MID$(E$,I,1)),0,1
85 NEXTI
90 CURSETE,10,0
100 DRAW110,0,1
110 DRAW0,28,1
120 DRAW-110,0,1
130 DRAW0,-28,1
140 CURSET137,0,0
150 FILL20,1,23
160 FILL120,1,22
170 FILL50,1,18
180 FOR I=1 TO 20
190 CURSET10+I*6,100,0:CHAR127,0,0:NEXT I
200 B$=" ":S=0:P=0
210 REM * Recherche du mot *
220 READJ
230 N=INT(RND(1)*J+1)
235 I=0
240 REPEAT
250 I=I+1
260 READ J$
265 U=FRE("")
270 UNTIL I=N
280 LM=LEN(J$)
290 FOR I=1 TO LM
300 CURSET10+I*6,100,0
310 CHAR 127,0,0
320 CHAR 42,0,1
330 NEXT I
340 REM * Debut *
350 PRINT"Quelle lettre jouez-vous?":T=0
360 GET A$
370 IF A$<"A"OR A$>"Z"THEN GOTO 360
380 PRINT"Vous avez joue la lettre ";A$
```

```
390 FOR I=1 TO LEN(B$)
400 IF A$=MID$(B$,I,1) THEN PRINT"deja jouee":B$=B$+A$:GOTO 460
410 NEXT I
420 B$=B$+A$
430 FOR I=1 TO LM
440 IF A$=MID$(J$,I,1) THEN GOSUB 1000
450 NEXT I
460 IF T=0 THEN GOTO2000
470 IF S<>LM THEN GOTO 350
480 PRINT"Vous avez trouve en ";LEN(B$)-1;" coups"
490 PRINTB$
495 WAIT400
500 PRINT"Voulez vous rejouer?(O/N)
510 GETD$
520 IF D$="O" THENPRINT"OK!":RESTORE:GOTO3500
530 IF D$="N" THEN END
540 GOTO 510
990 REM * Lettre trouvee *
1000 T=1
1010 S=S+1
1020 CURSET10+I*6,100,0
1030 CHAR 127,0,0
1040 CHAR ASC(A$),0,1
1050 RETURN
1990 REM * Trace du Pendu *
2000 P=P+1
2010 GOTO 2000+100*P
2100 CURSET120,130,0
2110 FILL20,1,1
2120 FOR I=1 TO 10
2130 CURSET150,129+I,0
2140 DRAW60,0,1
2150 NEXT I
2160 CURSET140,140,0
2170 DRAW 80,0,1
2180 CURMOV 0,5,0
2185 DRAW-80,0,1
2190 GOTO 350
2200 CURSET 140,40,0
2210 FILL90,1,5
2220 FOR I=1 TO 8
2230 CURSET 199+I,40,0
2240 DRAW 0,90,1
2250 NEXT I
2260 GOTO 350
2300 CURSET 140,41,0
2310 FILL 10,1,4
2320 FOR I=1 TO 10
2330 CURSET 150,40+I,1
2340 DRAW 60,0,1
2350 NEXT I
2360 CURSET 180,50,1
2370 DRAW 20,20,1
2380 GOTO 350
2400 CURSET 175,50,1
2410 DRAW 0,10,1
2420 GOTO 350
2500 CURSET175,67,0:CIRCLE 7,1
2510 CURSET173,64,1
2520 CURMOV4,0,1
2530 CURSET173,67,0:CHAR45,0,1
```

```
2540 GOTO 350
2600 CURSET 175,74,1
2610 DRAW0,28,1
2620 GOTO 350
2700 CURSET 175,80,1
2710 DRAW-10,15,1
2720 GOTO 350
2800 CURSET 175,80,1
2810 DRAW 10,15,1
2820 GOTO 350
2900 CURSET175,100,1
2910 DRAW-10,20,1
2920 GOTO 350
3000 CURSET175,100,1
3010 DRAW 10,20,1
3020 PING
3050 PRINT"Desole,vous avez Perdu"
3060 PRINT"Le mot etait:";J$
3070 GOTO 495
3500 FOR I=1 TO 15
3510 FOR J=1 TO 15
3520 CURSET130+J*6,20+I*8,0
3530 CHAR 127,0,0
3540 NEXTJ,I
3550 GOTO 180
3990 REM * Dictionnaire *
4000 DATA40,"KIWI","ZIGOMAR","ORIC","DEVELOPPEMENT","ILLUSTRATION"
4010 DATA "REVOLUTION","THALASSOTHERAPIE","OUIE","RESOLUTION","KYST
E"
4020 DATA "ALEATOIRE","ORDINATEUR","UNIVERSEL","MICROPROCESSEUR","M
EMOIRE"
4030 DATA "TERMINAL","INFORMATIQUE","AZIMUT","ANALYSE","TECHNOLOGIE
"
4040 DATA "COCASSERIE","HESITER","KSAR","ORTHOGRAPHE","LUXE"
4050 DATA "ZEN","FISTULE","PLASMODESME","VACUOLE","HYPOTHALAMUS"
4060 DATA "RESISTANCE","CONDENSATEUR","INDUCTANCE","DIODE","TRANSIS
TOR"
4070 DATA "GRAVITATION","RELATIVITE","ATTRACTION","RAYONNEMENT","GA
LAXIE"
```

5.5 REVERSE

Ce jeu pourrait s'appeler le jeu des contorsionnistes de l'esprit. Vous comprendrez vite pourquoi en y jouant !

La règle du jeu est simple : il s'agit de remettre dans l'ordre croissant une série de neuf chiffres préalablement mélangés. Simple, oui, mais pour effectuer ce réarrangement, vous devez dire combien de chiffres à partir de la gauche vous voulez inverser.

Par exemple si vous demandez d'inverser quatre chiffres à la série : 4 3 2 1 5 6 7 8 9 vous obtiendrez : 1 2 3 4 5 6 7 8 9 et vous aurez gagné.

Evidemment, quand les chiffres sont mélangés davantage, de nombreuses inversions sont nécessaires avant d'arriver à la solution. Le but du jeu est justement de trouver la méthode permettant d'en effectuer un minimum.

Il existe une méthode systématique qui permet d'arriver facilement au bout mais qui est longue. Vous pouvez vous fier à votre chance et inverser un peu au hasard.

Faites le concours du plus rapide et que le meilleur l'emporte ! Pour répondre aux questions, entrez la réponse puis pressez la touche [RETURN].

Le programme mélange la série de chiffres, vous donne le nombre de coups joués et détecte la fin de la partie. Si vous voulez vous arrêter en cours de partie, pressez le chiffre 0.

Méfiez-vous des torticolis que vous pourriez subir en essayant de regarder les chiffres inversés !

Structure du programme :

Ligne 40, initialisation de la chaîne de caractères donnant la solution.

Lignes 50 à 140, mélange des chiffres.

Lignes 170 à 200, entrée du nombre des chiffres à retourner.

Lignes 210 à 250, inversion des chiffres.

Ligne 270, test de fin de partie.

Lignes 280 à 320, fin du jeu.

La ligne 100 vérifie que la série ne contient pas le même chiffre plusieurs fois.

```
5 REM  ***********
10 REM * REVERSE *
15 REM ***********
20 CLS:PAPER3:INK4:PRINT"REVERSE":PRINT
30 E$=""
40 FOR I=1 TO 9:E$=E$+STR$(I):NEXT I
50 L$=STR$(INT(RND(1)*9+1))
60 N=2
70 REPEAT
80 A=INT(RND(1)*9+1)
90 FOR I=2 TO N STEP 2
100 IF VAL(MID$(L$,I,1))=A THEN GOTO 80
110 NEXT I
120 N=N+2
```

```
130 L$=L$+STR$(A)
140 UNTIL N=18
150 PRINTL$:G$=""
160 PRINT"COUP NUMERO ";R
170 PRINT:INPUT"COMBIEN DE CHIFFRES RETOURNEZ VOUS";C$
175 IF LEN(C$)>1 THEN GOTO 170
180 IF C$<"0" OR C$>"9" THEN GOTO 170
190 IF C$="0" THEN GOTO 300
200 C=VAL(C$)
210 D$=LEFT$(L$,C*2)
220 FOR I=1 TO C
230 G$=MID$(D$,2*I-1,2)+G$
240 NEXT I
250 L$=G$+RIGHT$(L$,18-2*C)
260 R=R+1
270 IF L$<>E$ THEN GOTO 150
280 PRINT L$
290 PRINT"VOUS AVEZ REUSSI EN ";R;" COUPS"
300 INPUT"VOULEZ VOUS REJOUER";F$
310 IF F$="O" THEN RUN
320 END
```

5.6 SOLITAIRE

Quand Montaigne jouait les ermites dans sa bibliothèque, il aimait se divertir grâce à ce jeu. Comme son nom l'indique, le solitaire se joue seul. Le principe est simple. Le plateau de jeu est constitué d'une grille formée par l'intersection de six lignes, formant ainsi trente-trois intersections où l'on peut loger trente-deux fiches. Pour déplacer une fiche, il faut prendre une autre fiche. Cela peut se faire si l'intersection suivant celle-ci est inoccupée. Le déplacement est horizontal ou vertical seulement. Cela ressemble à la prise au jeu de dames sauf que l'on ne se déplace pas en diagonale. Le but du jeu est de n'avoir plus qu'une seule fiche à la fin de la partie. Pour y arriver, il faut de la méthode et surtout de l'astuce. Ne laissez pas votre matière grise au chômage !

Un inconvénient du jeu tel qu'il se présente habituellement, est que comme on y joue seul, par définition, il n'y a pas d'arbitre. Il est bien difficile dans ce cas de résister à la tentation de pousser un peu le sort en fin de partie pour se rapprocher du but. Avec l'Oric, vous disposerez d'un juge-arbitre impartial qui vous interdira les déplacements illicites et vérifiera que la partie est terminée.

L'écran est utilisé comme plateau de jeu. Chaque intersection est repérée par ses coordonnées horizontale et verticale. Vous rentrerez une lettre de A à G pour l'horizontale puis un chiffre de 1 à 7 pour la verticale. Entrez les deux caractères à la suite puis pressez la touche [RETURN]. Vous pouvez toujours corriger vos coordonnées grâce à la touche [DEL] avant d'avoir pressé [RETURN]. Tout autre manipulation sera refusée hormis [CTRL] qui arrêtera le programme en cours d'exécution si vous désirez interrompre la partie. Celle-ci peut être reprise grâce à la commande CONT.

Bon divertissement et ne fatiguez pas trop vos neurones !

Structure du programme :

Lignes 20 à 220, assurent l'initialisation des tableaux contenant les déplacements possibles (F et G) et la grille du jeu (S). La valeur 0 indique une intersection vide, 1 une occupée, et -1 une intersection interdite.

Lignes 230 à 350, dessinent la grille sur l'écran.

Lignes 360 à 420, permettent l'entrée des coordonnées de la fiche à déplacer et testent si elles sont valides.

Lignes 430 à 540, permettent l'entrée des coordonnées de la fiche déplacée et testent si l'intersection d'arrivée est autorisée.

Lignes 550 à 580, modifient le tableau représentant la grille du jeu.

Lignes 590 à 660, déplacent et ôtent les fiches correspondant au coup joué.

Lignes 670 à 770, servent à tester la fin de la partie.

Lignes 780 à 820, assurent l'affichage des résultats finaux et la fin du jeu.

Le sous-programme 1000 à 1050 redéfinit les caractères de code ASCII 95 et 96 qui servent à représenter le plateau du jeu.

```
5 REM  *************
10 REM * SOLITAIRE *
15 REM *************
20 DIM S(11,11)
30 N=32:TEXT:CLS
40 F(1)=0:G(1)=1
50 F(2)=0:G(2)=-1
60 F(3)=-1:G(3)=0
70 F(4)=1:G(4)=0
80 X=70:Y=60
90 PAPER3:INK4
100 PLOT10,10,"JEU DU SOLITAIRE"
110 GOSUB 1000
120 FOR I=1 TO 11
130 FOR J=1 TO 11
140 S(I,J)=-1
150 NEXTJ,I
160 FOR I=3 TO 9
170 FOR J=5 TO 7
180 S(I,J)=1
190 S(J,I)=1
200 NEXTJ,I
210 S(6,6)=0
220 HIRES:INK5
230 PAPER6
240 FOR I=3 TO 9
250 FOR J=3 TO 9
260 CURSET X+I*6,Y+J*8,0
270 IF S(I,J)=1 THEN CHAR 96,0,1
280 IF S(I,J)=0 THEN CHAR 95,0,1
290 NEXTJ,I
300 FOR I=3 TO 9
310 CURSET X+I*6,Y,0
320 CHAR 46+I,0,1
330 CURSET X,Y+I*8,0
340 CHAR 62+I,0,1
350 NEXT I
360 INPUT"Coordonnees de la fiche a deplacer";A$
370 C=ASC(LEFT$(A$,1)):A=FRE("")
380 B=ASC(RIGHT$(A$,1))
390 IF ((C-65)*(C-71)<=0)*((B-49)*(B-57)<=0) <>1 THEN GOTO 360
400 B=B-46
410 C=C-62
420 IF S(B,C)<>1 THEN GOTO 360
430 INPUT"Coordonnees d'arrivee de la fiche";A$
440 E=ASC(LEFT$(A$,1)):A=FRE("")
450 D=ASC(RIGHT$(A$,1))
460 IF ((E-65)*(E-71)<=0)*((D-49)*(D-57)=<0) <>1 THEN GOTO 430
470 D=D-46
480 E=E-62
490 FOR I=1 TO 4
500 IF D=B+F(I)*2 AND E=C+G(I)*2 THEN GOTO 540
510 NEXT I
520 PRINT"NON VALABLE"
530 GOTO 360
540 IF S(D,E)<>0 OR S(B+F(I),C+G(I))<>1 THEN GOTO 360
550 N=N-1
560 S(B,C)=0
570 S(D,E)=1
580 S(B+F(I),C+G(I))=0
590 CURSET X+B*6,Y+C*8,0
600 CHAR 127,0,0
610 CHAR 95,0,1
```

```
620 CURSET X+(B+F(I))*6,Y+(C+G(I))*8,0
630 CHAR 127,0,0
640 CHAR 95,0,1
650 CURSET X+D*6,Y+E*8,0
660 CHAR 96,0,1
670 IF N=1 THEN GOTO 780
680 FOR I=3 TO 9
690 FOR J=3 TO9
700 IF S(I,J)=1 AND S(I,J+1)=1 AND S(I,J+2)=0 THEN GOTO 360
710 IF S(J,I)=1 AND S(J+1,I)=1 AND S(J+2,I)=0 THEN GOTO 360
740 IF S(12-I,12-J)=1 AND S(12-I,11-J)=1 AND S(12-I,10-J)=0 THEN GO
TO 360
750 IF S(12-J,12-I)=1 AND S(11-J,12-I)=1 AND S(10-J,12-I)=0 THEN GO
TO 360
760 NEXT J
770 NEXT I
780 PRINT"FIN DE LA PARTIE"
790 PRINT"IL RESTE ";N;"FICHES"
800 WAIT500
810 INPUT "Voulez vous rejouer(O/N)";A$
820 IF A$="O" THEN RUN
830 END
1000 FOR I=46840 TO 46855
1010 READ J
1020 POKE I,J
1030 NEXT I
1040 RETURN
1050 DATA 8,8,8,8,63,8,8,8,8,8,28,62,63,62,28,8
```

5.7 TÉLÉCRAN

Développez les facultés de dessinateur qui sommeillent en vous. Télécran vous aidera dans cette tâche. Vous pourrez grâce à lui faire naître sur votre écran des graphismes plus beaux les uns que les autres. Vous voulez dessiner une voiture, un avion, une maison, un paysage... c'est possible. C'est votre imagination qui crée l'objet que Télécran va visualiser. Laissez votre côté artistique s'épanouir.

Le programme que nous vous proposons ici permet d'effectuer les mêmes fonctions que sur un Télécran traditionnel, c'est à dire : déplacer le crayon à l'aide de plusieurs commandes, ce crayon laissant une trace visualisant sa trajectoire, et effacer complètement l'écran. En sus de ces fonctions le programme vous permettra de travailler en mode "déplacement", le crayon se déplaçant sans laisser de trace, ou en mode "gomme", le crayon étant alors remplacé par une gomme qui effacera tout sur son passage.

Commandes à utiliser :

- Les flèches donnent le sens de déplacement du crayon.

- E : mode écriture, le déplacement du crayon laisse une trace sur l'écran.

- D : mode déplacement, le déplacement du crayon ne laisse pas de trace de son passage.

- G : mode gomme, le crayon efface tout sur son passage.

- T : commande d'effacement de l'écran.

- F : commande de sortie du programme.

Structure du programme :

Lignes 10 à 100, impression des commandes possibles.

Lignes 110 à 170, tracé du cadre limitant l'évolution du pointeur.

Lignes 180 à 190, initialisations.

Lignes 200 à 260, acquisition et validation de la commande clavier.

Lignes 1000 à 1030, déplacement du curseur vers la gauche.

Lignes 2000 à 2030, déplacement du curseur vers la droite.

Lignes 3000 à 3030, déplacement du curseur vers le haut.

Lignes 4000 à 4020, déplacement du curseur vers le bas.

Lignes 5000 à 5030, affichage curseur.

REMARQUE : Il est possible de changer les couleurs du "Papier" et de l'"Encre", pour cela,

Taper F.

Si vous voulez conserver votre dessin

Entrez

PAPER p : INK i (p et i étant des attributs couleur)

puis

GOTO 130

Si vous voulez effacer le dessin

Entrez

HIRES

puis

PAPER p : INK i

puis

GOTO 130

```
_________
9 REM  ************
10 REM * TELECRAN *
11 REM ************
20 CLS
30 PRINT
40 PRINT"Vous pouvez:"
50 PRINT"Ecrire(E),"
60 PRINT"Gommer(G),"
70 PRINT"Deplacer le crayon(D),"
80 PRINT"utiliser les fleches,"
90 PRINT"effacer tout(T) ou vous arreter(F)."
:WAIT500
100 REM
110 HIRES
120 PAPER6:INK4
130 CURSET20,10,1
140 DRAW200,0,1
150 DRAW0,180,1
160 DRAW-200,0,1
170 DRAW0,-180,1
180 CURSET120,100,1
185 X=120 :Y=100 :H=1 :G=1 :T=0
190 PRINT:PRINT"ECRITURE"
200 A=FRE(""):GET M$
210 IF M$="G" THEN H=0 :G=0 :T=0 :PRINT :PRINT"GOMME"
```

```
220 IF M$="D" THEN H=1 :G=3:T=1 :PRINT :PRINT"DEPLACEMENT"
230 IF M$="E" THEN H=1 :G=1 :T=0 :PRINT :PRINT"ECRITURE"
240 IF M$="F" THEN END
242 IF M$="T" THEN GOTO 110
245 CURSETX,Y,G
247 IF T=1 THEN G=0
250 IF M$)=CHR$(8)ANDCHR$(11)>=M$ THENONASC(M$)-7 GOTO1000,2000,300
0,4000
255 IFT=1ANDPOINT(X,Y)=-1THENG=1
260 GOTO 200
1000 REM
1010 X=X-1
1020 IF X<20 THEN X=X+1
1030 GOTO 5000
2000 REM
2010 X=X+1
2020 IF X>220 THEN X=X-1
2030 GOTO 5000
3000 REM
3010 Y=Y+1
3020 IF Y>190 THEN Y=Y-1
3030 GOTO 5000
4000 REM
4010 Y=Y-1
4020 IF Y<10 THEN Y=Y+1
5000 REM
5015 IF POINT(X,Y)=-1 AND T=1 THENG=1
5020 CURSETX,Y,H
5030 GOTO 200
```

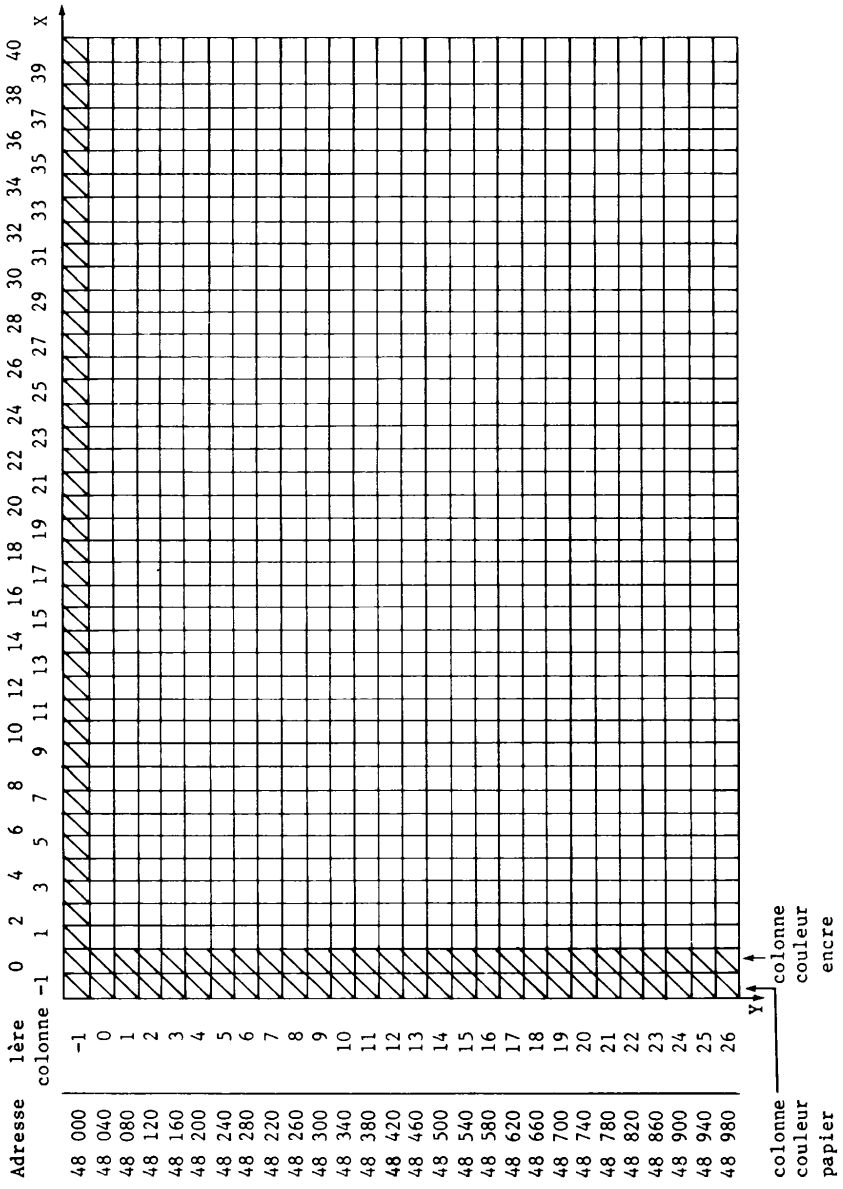
Annexes

1 Coordonnées sur l'écran

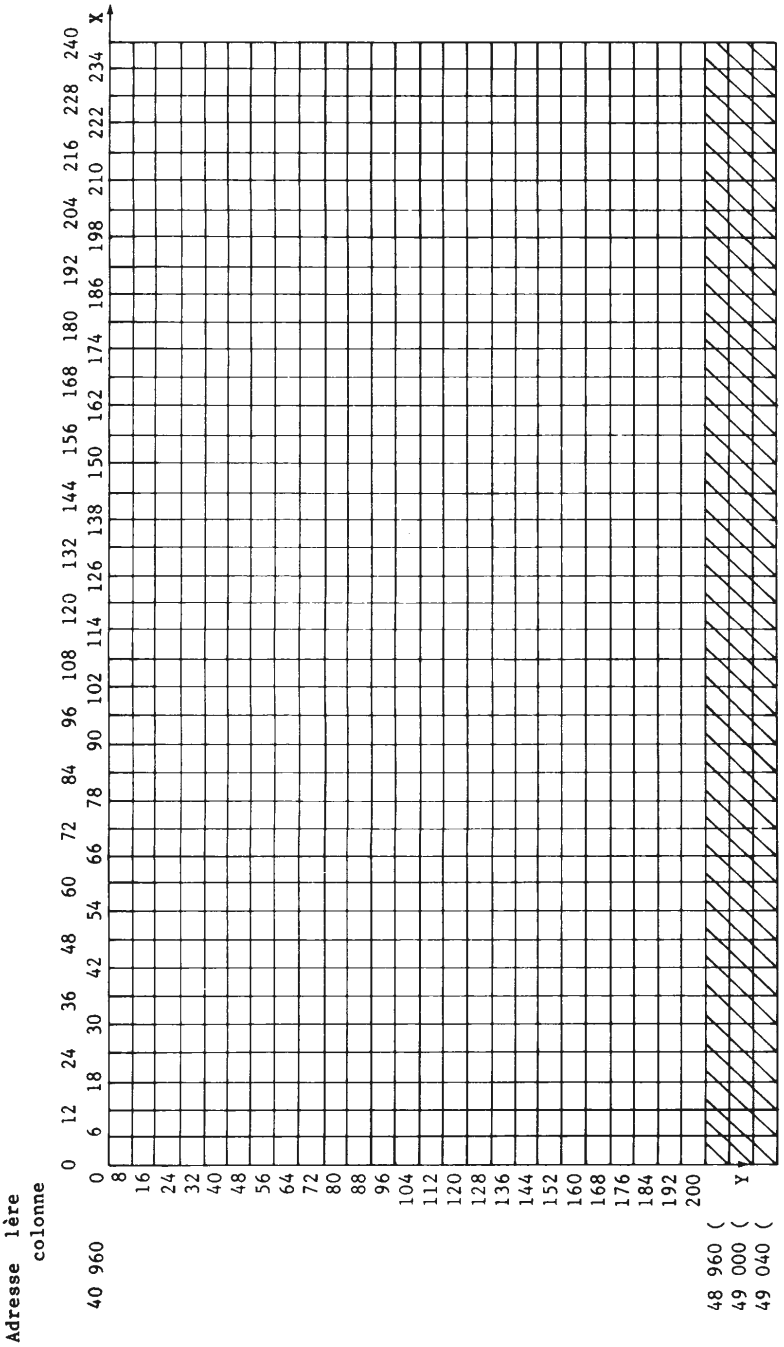
Grilles donnant les coordonnées sur l'écran.

A. — Basse résolution

La zone non hachurée est accessible par les instructions PRINT et PLOT. La colonne 0 est accessible par un PLOT.



B. — Haute résolution



2 Caractères ASCII

Table des caractères contenus en RAM
emplacement du 1er octet en mémoire

1	2	3	4	5	1	2	3	4	5
32	160	vide	46336	47360	69	197	E	46632	47656
33	161	!	46344	47368	70	198	F	46640	47664
34	162	"	46352	47376	71	199	G	46648	47672
35	163	#	46360	47384	72	200	H	46656	47680
36	164	$	46368	47392	73	201	I	46664	47688
37	165	%	46376	47400	74	202	J	46672	47696
38	166	&	46384	47408	75	203	K	46680	47704
39	167	'	46392	47416	76	204	L	46688	47712
40	168	(	46400	47424	77	205	M	46696	47720
41	169	)	46408	47432	78	206	N	46704	47728
42	170	*	46416	47440	79	207	O	46712	47736
43	171	+	46424	47448	80	208	P	46720	47744
44	172	,	46432	47456	81	209	Q	46728	47752
45	173	-	46440	47464	82	210	R	46736	47760
46	174	.	46448	47472	83	211	S	46744	47768
47	175	/	46456	47480	84	212	T	46752	47776
48	176	0	46464	47488	85	213	U	46760	47784
49	177	1	46472	47496	86	214	V	46768	47792
50	178	2	46480	47504	87	215	W	46776	47800
51	179	3	46488	47512	88	216	X	46784	47808
52	180	4	46496	47520	89	217	Y	46792	47816
53	181	5	46504	47528	90	218	Z	46800	47824
54	182	6	46512	47536	91	219	[	46808	47832
55	183	7	46520	47544	92	220	ç	46816	47840
56	184	8	46528	47552	93	221	]	46824	47848
57	185	9	46536	47560	94	22	^	46832	47856
58	186	:	46544	47568	95	223	£	46840	47864
59	187	;	46552	47576	96	224	©	46848	
60	188	<	46560	47584	97	225	a	46856	
61	189	=	46568	47592	98	226	b	46864	
62	190	>	46576	47600	99	227	c	46872	
63	191	?	46584	47608	100	228	d	46880	
64	192	@	46592	47616	101	229	e	46888	
65	193	A	46600	47624	102	230	f	46896	
66	194	B	46608	47632	103	231	g	46904	
67	195	C	46616	47640	104	232	h	46912	
68	196	D	46624	47648	105	233	i	46920	

1 : Code ASCII ; 2 : Code en inverse ; 3 : Caractère
4 : Jeu normal ; 5 : Jeu semi-graphique

ANNEXE 2 - CARACTERES ASCII

Table des caractères contenus en RAM
emplacement du ler octet en mémoire

1	2	3	4	1	2	3	4
106	234	j	46928	117	245	u	47016
107	235	k	46936	118	246	v	47024
108	236	l	46944	119	247	w	47032
109	237	m	46952	120	248	x	47040
110	238	n	46960	121	249	y	47048
111	239	o	46968	122	250	z	47056
112	240	p	46976	123	251	{	47064
113	241	q	46984	124	252		47072
114	242	r	46992	125	253	}	47080
115	243	s	47000	126	254	▒	47088
116	244	t	47008	127	255	■	47096

1 : Code ASCII ; 2 : Code en inverse ; 3 : Caractère
4 : Jeu normal

3 Commandes FILL

Syntaxe : FILL L, C, A

L est le nombre de lignes à remplir (de 1 à 200) à partir de la position du curseur qui est donnée par un CURSET précédent, ou à partir de la dernière ligne du FILL précédent. Ne pas donner à L une valeur telle que l'on puisse déborder de l'écran.

C est le nombre de colonnes de 6 pixels à partir du curseur. Il est compris entre 1 et 40. On ne doit pas déborder l'écran.

A est un attribut du FILL (de 0 à 255).

Si $0 < A < 32$, on obtient les mêmes effets que pour l'attribut du PLOT.

. de 0 à 7 modifie la couleur de l'encre.

. de 16 à 23 modifie la couleur du papier.

La commande agit alors sur toutes les colonnes suivantes sur la ligne quelle que soit la valeur de C. Son effet s'arrête quand une autre commande FILL est rencontrée sur la ligne.

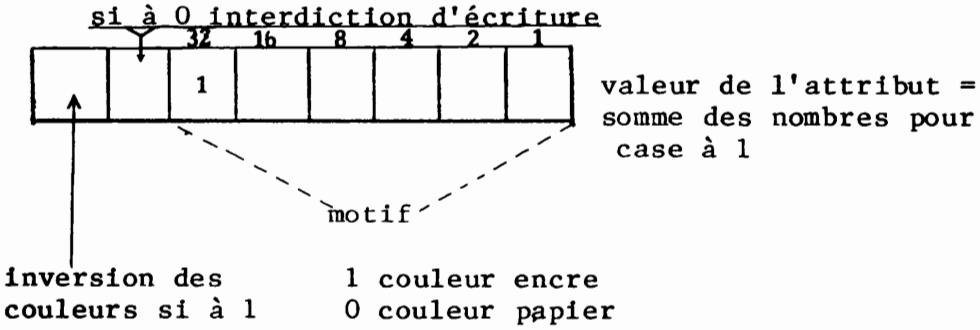
Les attributs en mémoire d'écran sont lus de la gauche vers la droite de l'écran.

On ne peut écrire sur les cases contenant les codes.

. de 12 à 15 provoque le clignotement.

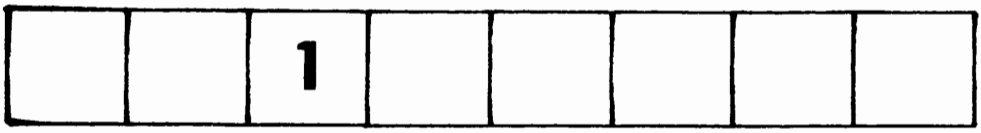
. de 8 à 11 redonne l'affichage normal.

. de 24 à 31 modifie les paramètres de synchronisation de l'image C (cela peut permettre une adaptation à un moniteur aux normes américaines). Ces valeurs sont à éviter car l'affichage peut se déstabiliser. On peut toutefois l'utiliser pour des effets spéciaux (explosion). Pour le resynchroniser utilisez l'attribut 31 au delà de 32, le bit 6 de l'attribut passe à 1, l'octet est reconnu comme un motif.



L'attribut 63 colorie les 6 pixels en encre, très utile pour remplir des zones de l'écran avec la couleur de l'encre.

Les motifs obtenus peuvent être modifiés par un **DRAW** ou un CIRCLE si le bit est à 1.



4 Les Attributs

Décimal	Escape	Fonction
0	@	encre noire
1	A	rouge
2	B	verte
3	C	jaune
4	D	bleue
5	E	magenta
6	F	cyan
7	G	blanche
8	H	caractères (standards) simple
		( ) hauteur
9	I	non (alternés)
		(
10	J	clignotants (standards) double
		( ) hauteur
11	K	(alternés)
12	L	caractères (standards) simple
		( ) hauteur
13	M	clignotants (alternés)
		(
14	N	(standards) double
		( ) hauteur
15	O	(alternés)
16	P	papier noir
17	Q	rouge
18	R	vert
19	S	jaune
20	T	bleu
21	U	magenta
22	V	cyan
23	W	blanc
24	X)	synchronisation mode Texte sur 60
25	Y)	Hz
26	Z)	synchronisation mode Texte sur 50
27	{)	Hz
28	!)	synchronisation mode Graphique sur
29	})	60 Hz
30	~)	synchronisation mode Graphique sur
31	←)	50 Hz

Les codes de la première colonne s'utilisent avec un PLOT.

Les codes de la deuxième colonne s'utilisent après un ESC ou après un CHR$(27) avec un PLOT.

LA BIBLIOTHÈQUE EDIMICRO

- *Collection « Guides microordinateurs »*

de Merly	**GUIDE DE L'APPLE**
	Tome 1 L'APPLE standard
	Tome 2 Les extensions
	Tome 3 Les applications
	Tome 4 Les langages
Bayvejiel	**GUIDE DE L'ORIC**
Bieber, Perbost, Renucci	**GUIDE DU TO 7**

- *Collection « Jeux »*

Chane-Hune, Darbois	**JEUX SUR ORIC**
Perbost, Renucci	**JEUX SUR TO 7**

- *Collection « Logiciels »*

Bonnet, Dinh	**MULTIPLAN SUR APPLE**
	Exercices de Gestion

- *Ouvrages de base*

Viguier	**Premiers pas en programmation sur ORIC**

Les séminaires de formation FDS/Edimicro

- *Pour utilisateurs non spécialistes*

  VISICALC - MULTIPLAN - GESTION DE FICHIERS - BASES DE DONNÉES - CHOIX D'UN MICROORDINATEUR - BUREAUTIQUE...

- *Pour programmeurs débutants*

  PREMIERS PAS EN BASIC - FICHIERS EN BASIC...

- *Pour professionnels de l'informatique*

  TÉLÉMATIQUE - UNIX - XENIX - MS-DOS - LANGAGE C...

EDIMICRO
121-127, avenue d'Italie, 75013 PARIS

Imprimé en France. — JOUVE, 18, rue Saint-Denis, 75001 PARIS
N° 12068. Dépôt légal : Novembre 1983

JEUX SUR ORIC

David CHANE-HUNE
François DARBOIS

Vingt jeux captivants pour votre ORIC :

* Jeux de hasard
  ALPHABET - JACK-POT - SIMON

* Jeux d'adresse
  MUR - BARON - MISSILE - BRACONNIER

* Jeux d'action
  ASTÉROÏDES - CHAMPS DE FORCE - DÉMINAGE - ENVAHISSEURS - LABYRINTHE - PILOTE DE CHASSE

* Jeux de réflexion
  AWELÉ - CAVALIER - MASTERMIND - PENDU - REVERSE - SOLITAIRE - TÉLÉCRAN

Le premier chapitre est consacré aux techniques de programmation des jeux mettant en œuvre les remarquables possibilités graphiques et sonores de l'ORIC.

Un excellent moyen d'apprendre, en vous amusant, à tirer le meilleur parti de votre ORIC.

Edimicro

121-127 Avenue d'Italie 75013 Paris

ISBN 2-904457-06-2

PRIX 79 F