

# ABC

N° 75

COURS  
D'INFORMATIQUE  
PRATIQUE  
ET FAMILIALE

## INFORMATIQUE



L'ordinateur à l'école

Premiers mots en pascal

Le portable Apicot

Listage du jeu Digitaya

EDITIONS  
**ATLAS**



# L'ordinateur à l'école

Quel est l'impact des ordinateurs sur les méthodes pédagogiques? Pour le savoir, voyons d'abord comment on s'est efforcé de formaliser et de mécaniser l'apprentissage de la connaissance.



« A long terme, il faut s'attendre à ce que l'informatique ait sur le système éducatif des effets aussi profonds que ceux qu'a entraînés le livre imprimé. » Cette phrase apparaît dans tous les rapports gouvernementaux commandés pour évaluer l'impact de l'informatique dans l'éducation. Des professeurs n'hésitent pas à déclarer que les écoles, telles qu'elles existent aujourd'hui, sont appelées à disparaître dans un avenir proche; l'idée a déjà été exprimée par un spécialiste comme Seymour Papert.

C'est là un point de vue sans doute excessif. Les changements qu'on a constatés jusqu'ici sont très lents, et ils sont le fait des enseignants eux-mêmes, plus que des machines dont ils se servent. Il n'en reste pas moins que beaucoup de gens

s'accordent à penser que les ordinateurs pourront jouer un très grand rôle en pédagogie, pourvu qu'on sache les mettre en œuvre de façon appropriée.

Il y a des précédents en ce domaine, puisqu'on a entrepris des recherches et des expériences en ce domaine depuis plus de vingt ans. Mais des machines à apprendre très simples existaient déjà aux alentours des années vingt. Il fallut pourtant presque trente ans avant qu'elles ne connaissent un certain succès. Aux États-Unis, le chef de file de l'école dite « behavioriste », B.F. Skinner mit près d'une vingtaine d'années à élaborer ses théories, après de multiples expériences sur les rats et les pigeons. Il avait par exemple eu l'idée de procéder au guidage d'un missile à l'aide de trois

## L'ordinateur à l'école

L'ordinateur peut dès maintenant remplacer dans une salle de classe le tableau noir, le projecteur de diapositives, la télévision, le lecteur de cassette et la bibliothèque — pourvu, bien sûr, qu'il dispose de logiciels appropriés. Certaines activités ne se prêtent pas à une telle substitution; mais l'ordinateur peut pourtant renforcer certaines méthodes d'enseignement. (Cl. Tony Lodge.)



pigeons installés dans le nez de l'engin... Il les avait conditionnés à pointer un endroit sur une carte en leur offrant de la nourriture à chaque bonne réponse.

Le Pentagone repoussa le projet, dont il ne subsista qu'« un amas d'appareils tout à fait inutiles, et des pigeons obsédés par un point particulier sur la côte du New Jersey »!

Skinner avait sur l'éducation des idées qui concurrent un meilleur sort. Il était partisan d'une méthode de « programmation linéaire » (le terme « programmation » ne doit pas s'entendre au sens informatique). Il s'agissait avant tout de présenter à chaque étudiant un ensemble d'informations par petits fragments à la fois, et en récompensant chaque réponse correcte. Nous ne sommes pas très loin du système de gratification des pigeons...

## La programmation linéaire

Deux concepts importants sont issus de la programmation linéaire : celui de *rétroaction* et celui d'*individualisation*. Le premier recouvre tout simplement le fait qu'après chaque réponse de l'étudiant, on lui dit aussitôt s'il a deviné juste ou non. Les machines de Skinner avaient le gros défaut de n'accepter que des réponses extrêmement précises. L'individualisation exprime le fait que le sujet peut travailler au rythme qui lui convient, au lieu de subir celui qu'impose le professeur.

L'apparition des ordinateurs coïncida presque exactement avec la vogue des idées de Skinner. On les employa donc, tout naturellement, à créer des systèmes de programmation linéaire, qui sont loin d'avoir disparu aujourd'hui. « Allons, encore un effort » ou « Excellente réponse ! » s'affichent sur bien des écrans. Mais les questions posées sont toujours aussi dramatiquement ennuyeuses.

Les créateurs de programmes s'efforcèrent de dépasser les limites de ce type de méthode. A chaque réponse incorrecte, l'étudiant se voyait proposer des éléments de réponse, puis on lui posait une nouvelle question. Cela paraît assez logique, bien que Skinner lui-même ait vu d'un très mauvais œil la « programmation intrinsèque »,

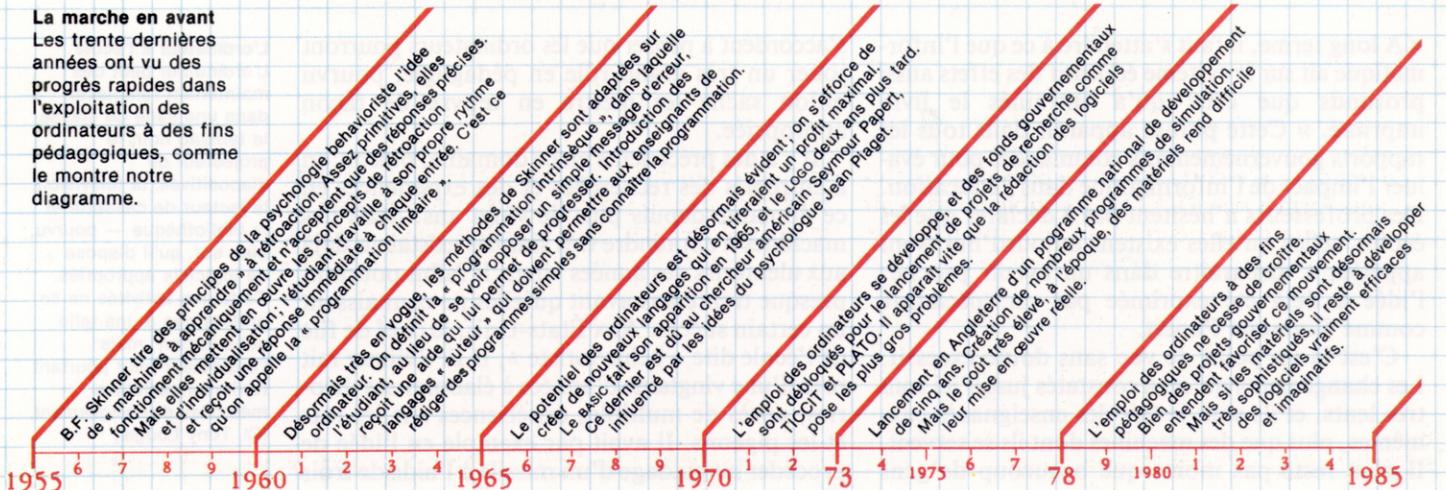
comme elle fut bientôt appelée. Il n'en reste pas moins que, pour le sujet, l'épreuve était moins redoutable et mieux adaptée à son propre cas. On mit aussi au point des « langages auteur » destinés à permettre aux enseignants d'écrire des programmes de ce type, sans connaître la programmation.

Programmation linéaire et programmation intrinsèque ont toutefois de gros inconvénients. Le cerveau de l'étudiant est considéré un peu comme un récipient vide, qu'il faut remplir de connaissances ; il s'agit d'emmagasiner des faits, non d'acquérir de l'expérience. Tout cela ne laisse aucune place à la créativité, à l'imagination, à l'expression de soi-même. D'une certaine façon, c'est à des conceptions de ce genre que l'enseignement assisté par ordinateur doit, très injustement, sa mauvaise réputation.

Les ordinateurs étaient pourtant riches de possibilités. Une des retombées inattendues des premières recherches fut la création, vers 1965, d'un langage « aussi proche de l'anglais que possible », et qui devait permettre aux scientifiques, aux ingénieurs et aux étudiants d'écrire leurs propres programmes. On était au Dartmouth College, aux États-Unis, et ce fut là que naquit le BASIC, qui devait plus tard connaître une fortune éclatante.

Les conditions d'alors n'étaient pas les mêmes que celles d'aujourd'hui : un ordinateur de 16 K coûtait près de 40 000 dollars et occupait une pièce entière. Il fallait donc mettre au point un langage qui occupe aussi peu d'espace mémoire que possible. D'où certaines limitations — d'ailleurs liées à sa facilité d'apprentissage — qui sont indirectement responsables de la prolifération des dialectes BASIC.

Deux ans plus tard, une équipe de chercheurs du Massachusetts Institute of Technology définit un nouveau langage de programmation qui devait être « facile pour le programmeur, non pour l'ordinateur ». L'objectif était en fait d'apprendre aux très jeunes enfants à manipuler un ordinateur, ce qui était alors une idée tout à fait révolutionnaire. Le projet était dirigé par Seymour Papert, qui avait travaillé avec Jean Piaget, le spécialiste de la psychologie enfantine. Les conceptions de ce dernier étaient aux anti-





podés de celles de Skinner, qui restait très mécaniste. Papert voulait créer un environnement au sein duquel l'enfant pourrait apprendre en découvrant. Sa formule était : « Plutôt que de laisser l'ordinateur programmer l'enfant, laissons l'enfant programmer l'ordinateur. » Il en sortit le langage LOGO, qui connaît un succès grandissant dans les écoles, et qui est désormais disponible sur la plupart des micro-ordinateurs. Le BASIC et le LOGO arrachaient la programmation aux professionnels pour la rendre accessible à quiconque voulait s'en donner la peine.

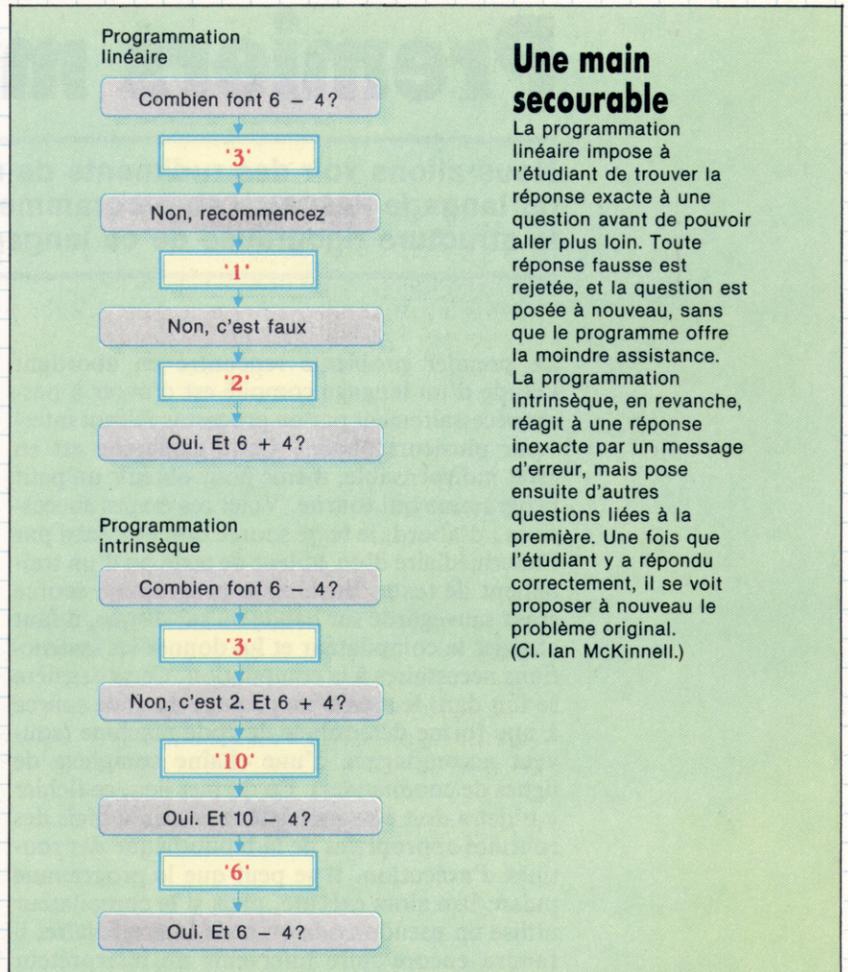
Les vieilles méfiances n'en étaient pas levées pour autant. Pour en venir à bout, la National Science Foundation of America décida en 1970 d'investir dix millions de dollars dans des recherches qui devaient s'étendre sur cinq ans. Deux projets furent mis sur pied : TICCIT et PLATO.

## TICCIT et PLATO

TICCIT (*Time-shared Interactive Computer Controlled Information Television*, ou « télévision interactive contrôlée par ordinateur en temps partagé ») avait pour objectif d'assurer une formation supérieure à celle fournie par les méthodes classiques, et ce à moindre coût. La compagnie chargée de la réalisation du projet devait créer matériels et logiciels. Elle mit au point un clavier spécial, avec des touches spéciales, un écran vidéo, un haut-parleur. Il y avait en tout cent vingt-huit terminaux reliés à deux mini-ordinateurs. Les résultats restèrent mitigés : les logiciels n'étaient pas toujours bien adaptés, et certains étudiants déclarèrent forfait en cours de route. Ceux qui allèrent jusqu'au bout obtinrent effectivement de meilleurs résultats que les autres. TICCIT est encore en usage dans deux universités pilotes, mais n'a été adopté nulle part ailleurs.

PLATO (*Programmed Logic for Automatic Teaching Operation*, ou « logique programmée pour la mise en œuvre de l'enseignement automatique ») est constitué d'un réseau de mille terminaux reliés à un gros système installé en Illinois. Il renferme une véritable bibliothèque de leçons qu'il peut faire parvenir à la demande à tel ou tel terminal qui communique aussi avec les autres. L'ensemble est très sophistiqué ; écrans transparents permettant la projection de diapositives, écrans tactiles pour simplifier la communication avec l'ordinateur central. Un langage « auteur » appelé TUTOR facilite aux enseignants la rédaction de leurs programmes. Là encore, on n'a pu parvenir à des conclusions définitives. La création des logiciels a demandé beaucoup plus de temps et d'efforts qu'on ne pensait. Les étudiants formés par PLATO ne semblent pas avoir plus progressé que leurs camarades soumis à une pédagogie de type classique. Le système paraît, il est vrai, plus « amical » que TICCIT.

Dès 1973, le gouvernement britannique avait parrainé un projet de recherche qui devait s'étendre sur cinq ans et coûter deux millions et demi de livres. Il s'agissait de mettre au point un système d'enseignement assisté par ordinateur. La plus grande part des fonds alloués fut consa-



## Une main secourable

La programmation linéaire impose à l'étudiant de trouver la réponse exacte à une question avant de pouvoir aller plus loin. Toute réponse fautive est rejetée, et la question est posée à nouveau, sans que le programme offre la moindre assistance. La programmation intrinsèque, en revanche, réagit à une réponse inexacte par un message d'erreur, mais pose ensuite d'autres questions liées à la première. Une fois que l'étudiant y a répondu correctement, il se voit proposer à nouveau le problème original. (Cl. Ian McKinnell.)

crée à la création de logiciels de simulation, l'ordinateur recréant des situations très difficiles à observer dans la vie courante. L'un d'eux, par exemple, simulait la concentration d'un marqueur chimique dans le flot sanguin ; d'autres illustraient les réactions chimiques. Mais presque tous les programmes manquaient cruellement d'imagination, et, de plus, le coût très élevé — à l'époque — des ordinateurs empêchait qu'on s'en serve réellement.

Tout système éducatif est déterminé par des conditions économiques, politiques, sociales. Les ordinateurs eux-mêmes en ressentent les effets, et ils sont de surcroît toujours menacés d'obsolescence technologique. Il y a beaucoup à faire avant qu'on puisse convaincre ceux qui ne voient en eux que de simples outils de mémorisation inquiétants et limités.

Un facteur nouveau est toutefois à prendre en compte : la phénoménale baisse de prix de la mémoire. Un micro-ordinateur de 64 K n'étonne plus personne ; il y a trente ans, c'était encore un miracle. Il se peut, bien entendu, que l'avancée au niveau du matériel ait devancé l'évolution des logiciels eux-mêmes, et même qu'elle l'ait entravée. Mais les responsables gouvernementaux paraissent désormais conscients du problème et, si la baisse de prix des ordinateurs se poursuit, on peut s'attendre à des bouleversements décisifs du système éducatif dans les dix années qui viennent.

# Premiers mots

**Nous allons voir des rudiments de syntaxe et de vocabulaire du langage PASCAL. Les programmeurs, sevrés de BASIC, apprécieront la structure rigoureuse de ce langage et sa grande économie.**

Le premier problème rencontré en abordant l'étude d'un langage compilé est d'avoir à passer nécessairement par un processus faisant intervenir plusieurs phases. Cette démarche est en effet indispensable, même pour obtenir un petit programme qui tourne. Voici ces étapes successives : d'abord, le texte source doit être saisi par l'intermédiaire d'un éditeur de texte ou d'un traitement de texte. Ensuite, le programme source étant sauvegardé sur bande ou sur disque, il faut charger le compilateur et lui donner les instructions nécessaires à la compilation. Cette dernière se fait dans le sens d'un passage du code source à une forme déterminée de code machine (souvent accompagnée d'une chaîne complexe de lignes de commandes). En dernier lieu, ce fichier « objet » doit être traduit et lié par le biais des routines appropriées de la bibliothèque des routines d'exécution. Il se peut que le programme puisse être alors exécuté, mais si le compilateur utilise un pseudo-code ou code intermédiaire, il faudra encore faire intervenir un interpréteur d'exécution de programme.

Consolez-vous en apprenant que la plupart des PASCAL disponibles pour les micro-ordinateurs simplifient ce processus à l'extrême. Il faut néanmoins, lors du développement du programme, que votre texte source, le compilateur, et le programme en code objet résident tous en mémoire. L'efficacité du PASCAL et sa petite taille le rendent possible, et le système qui en résulte n'est souvent pas plus compliqué que celui d'un BASIC résident. C'est pourquoi, seuls les programmes importants nécessitent le processus complexe décrit plus haut.

Chaque machine a son propre jeu de commandes de l'éditeur et du compilateur. Le plus souvent, il s'agira d'une simple lettre, « E » pour édition, « C » pour compilation, « X » pour eXécution... Ce qui nous intéresse ici, c'est la syntaxe d'un programme, quel que soit son degré de complexité. Par bonheur, le PASCAL est si bien standardisé qu'il ne sera presque jamais fait mention de « variantes », comme avec le BASIC pour lequel il s'agissait de traiter de très nombreuses versions. Nous verrons cependant quelques exceptions. Pour le moment, écrivons notre premier programme PASCAL :

```
Premier Program (résultat);
Const
  Message = 'Programmation en Pascal';
Begin
  write (message)
End
```

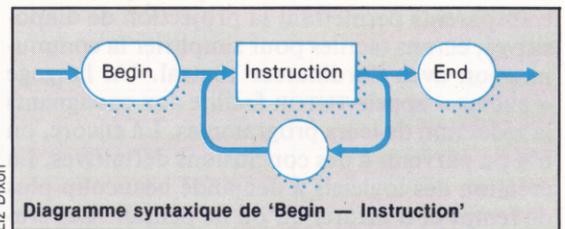
Avant d'étudier cet exemple plus en détail, essayez de le saisir, de le compiler et de l'exécuter. Si le compilateur vous adresse un message d'erreur, pesez-en bien les termes pour trouver l'erreur. Chaque symbole doit être à sa place. Veillez à ne pas omettre le point-virgule à la fin des première et troisième lignes, ainsi que le point à la fin du programme. Lorsque vous serez parvenu à faire exécuter ce programme sans erreur, le message suivant s'affichera :

Programmation en Pascal

Ce programme peut sembler simpliste, mais il illustre bien la forme générale que doit prendre chaque module de programmation PASCAL (un programme, une procédure ou une fonction). Il doit comprendre trois parties :

1. L'en-tête, ici, un titre de programme.
2. Les déclarations et définitions, ici une seule définition de constante.
3. Le « corps » du module qui comporte toutes les instructions exécutables.

Les règles syntaxiques du PASCAL, au moins en ce qui concerne les rudiments du langage, peuvent être définies par des diagrammes dits *chemins syntaxiques*, que l'on peut comparer à la circulation dans une ville. Un chemin valide dans le diagramme va du coin supérieur gauche au coin inférieur droit. Chaque pavé ou étape dans ce chemin de décisions peut être soit une *entité syntaxique* (qui se décrit elle-même), contenue dans un bloc de forme ronde, soit un autre élément qui est décrit ailleurs par un autre diagramme de syntaxe et indiqué par un pavé rectangulaire.



Liz Dixon

En considérant le diagramme d'ensemble du programme, on voit que les mots BEGIN (début) et END (fin) sont définis comme appartenant au vocabulaire PASCAL. Ils n'ont besoin d'aucun autre diagramme pour établir leur signification. PASCAL comprend seulement trente-cinq mots fixes. Nous en donnons ici la liste complète. Notre premier programme en fait seulement intervenir quatre : Program, Const (pour constante),

Begin et End. Le mot qui suit Program est un identificateur qui donne le nom du programme, et qui doit être un identificateur légal.

En commençant par une lettre, et en utilisant exclusivement des lettres ou des chiffres, tous les noms sont possibles. Néanmoins, il est impossible d'utiliser un mot réservé. Voici des exemples :

```
Name
PASCAL
ProgramUn
N
XYZ123
Address12
JulietteDelabarre
UnTrèsLongIdentificateur
```

Tous ces identificateurs sont légaux, en revanche :

```
Prog-1
10%
Jacques.Le.Bûcheron
and
LeTemps$L'argent
JeN'avaisRienSu
101Dalmatiens
Cela représente
```

sont autant d'identificateurs illégaux, soit parce qu'ils contiennent des symboles qui ne sont pas alphanumériques, soit parce qu'ils reprennent un mot réservé (ici, and). Le dernier exemple est illégal dans la mesure où il comprend un espace entre les lettres, les deux mots sont illégaux. En PASCAL, les majuscules n'ont pas de signification propre. Il existe cependant des versions non standard qui insistent sur le fait de mettre des majuscules aux mots réservés.

A part les espaces à la fin des lignes, il existe un autre élément syntaxique pouvant servir de séparateur — le commentaire. Ce dernier peut figurer n'importe où dans un texte, sauf bien sûr au milieu d'un mot. Les commentaires doivent figurer entre parenthèses.

Soyons un peu plus téméraires, et abordons un programme plus authentiquement PASCAL :

```
Program Program Deux (input, output);
  (Donne le carré d'un nombre)

Const
  Message = "Tapez un nombre:";

Var
  nombre : entier;

Begin
  WriteLn;
  WriteLn;
  Write (Message);
  read (nombre);
  WriteLn (nombre, 'le carré est', nombre * nombre)
End
```

Notez que nous mettons maintenant l'identificateur entrées dans l'en-tête. Les mots input et output sont ici nécessaires, car ils désignent les fichiers



#### Compilateurs Grand-Public

Un développement professionnel du langage PASCAL coûte souvent plus cher qu'un ordinateur personnel, mais il existe depuis peu sur le marché des compilateurs d'un prix abordable pour les micros les plus répandus. (Cl. Riby/Éd. Atlas.)

externes avec lesquels le programme doit communiquer. Ils sont normalement représentés sur les micros respectivement par le clavier et la vidéo : les entrées se font au clavier, et les sorties sur écran. En mentionnant input, nous pouvons faire lire à l'ordinateur la saisie au clavier par la procédure standard PASCAL read. Comme pour write, tous les paramètres doivent être donnés entre parenthèses.

La zone mémoire utilisée pour sauvegarder ces paramètres est réservée par la déclaration Var, ici pour un seul nombre entier. Contrairement au BASIC qui ne peut ordinairement faire la distinction qu'entre nombres et chaînes (au moyen du signe dollar après l'identificateur), PASCAL possède un nombre presque illimité de types de données. Il est donc important de dire au compilateur quelle place mémoire réserver pour chaque élément de données. Toutes les variables doivent être signalées dans une déclaration Var.

Le curseur reste positionné immédiatement après le message, comme si l'on avait utilisé une instruction BASIC PRINT suivie d'un point-virgule. Il ne s'agit pas d'une erreur, mais de ce que nous avons voulu en utilisant la procédure write. Nous devons utiliser pour chaque nouvelle ligne la procédure qui constitue l'alternative à cette dernière, WriteLn. Ln est l'abrégé de Ligne. Une instruction WriteLn sans paramètres ne fera que créer une nouvelle ligne.

Le PASCAL vous permet de différencier les données qui sont susceptibles de changer et les éléments qui resteront constants pendant toute l'exécution du programme.

### Les 35 mots réservés du PASCAL

And	Case	Do	End	Function	In	Not	Procedure	Set	Until
Array	Const	Downto	File	Goto	Label	Of	Program	Then	Var
Begin	Div	Else	For	If	Mod	Or	Record	To	While
					Nil	Packed	Repeat	Type	With



# Regards sur le futur

**Nous avons vu MacProject, un programme graphique destiné au Macintosh d'Apple. Nous étudions ici une application similaire mais destinée aux machines MS-DOS (IBM PC et compatibles).**

Bien que Project de Microsoft n'ait pas les ressources graphiques de MacProject, il tire le meilleur parti possible des capacités graphiques limitées de l'IBM PC. Par exemple, en combinant le signe égal, ou le signe tiret, au signe supérieur à, Project obtient des « flèches » différentes :

=====> ----->

La première flèche peut figurer en caractères gras pour représenter le chemin critique, la seconde intervenant en caractères normaux pour indiquer les phases non essentielles. Les facteurs d'appréciation du programme sont cependant plus complexes que ce genre de détails. Ce sont les suivants :

- Quel est le degré de souplesse du programme, dans quelle mesure peut-il assimiler les changements intervenant dans la réalisation du projet ?
- Quel est son degré de clarté dans la présentation des rapports existant entre les activités ; cette présentation apporte-t-elle vraiment quelque chose à une gestion efficace ?
- Le programme peut-il être utilisé pour concevoir le projet au départ ? Ou bien, les utilisateurs doivent-ils d'abord mettre au point leurs idées de base ?

Pour répondre à la dernière question, l'utilisateur doit effectivement développer son projet sur le papier, avant de le saisir. Et, de fait, le manuel le dit bien :

« Avant de saisir les phases de votre projet, vous devez réunir les informations suivantes sur chacune d'elle :

- sa description ;
- sa durée ou le temps qu'elle doit prendre ;
- sa préparation (les autres tâches antérieures) ;
- le moment où elle doit commencer. »

Pour mettre en évidence l'importance de cette préparation, Microsoft Project adjoint à son manuel une annexe de plusieurs pages où il est en particulier expliqué :

« Une bonne gestion de projet doit se faire sentir dès le début. Définissez d'abord clairement les objectifs, leur raison d'être, les contraintes qui pèsent sur eux, les paramètres nécessaires à leur bonne réalisation. Vous devez ensuite recenser toutes les activités spécifiques au projet. »

« Aucun programme informatique ne peut suppléer au manque de clarté et de précision dans la définition des tâches. Celles-ci sont déterminantes pour la planification des ressources, leur attribution et leur contrôle, ainsi que pour le suivi des coûts du budget. Une définition hasardeuse se manifesterait dès le début de sa réalisation. »

La nécessité d'une préparation minutieuse n'étonnera pas ceux qui sont familiers des techniques de gestion. Mais ceux qui y voient précisément le rôle de l'ordinateur pourront penser qu'il s'agit d'une grave lacune. En effet, le but d'un logiciel n'est-il pas de prolonger l'esprit de méthode de l'homme ? Il faut néanmoins reconnaître que Microsoft Project contribue à donner l'ordre des priorités. En partant du travail initial de l'utilisateur, il lui permet d'avoir toute latitude de modifier son projet au fur et à mesure de son développement et de sa réalisation.

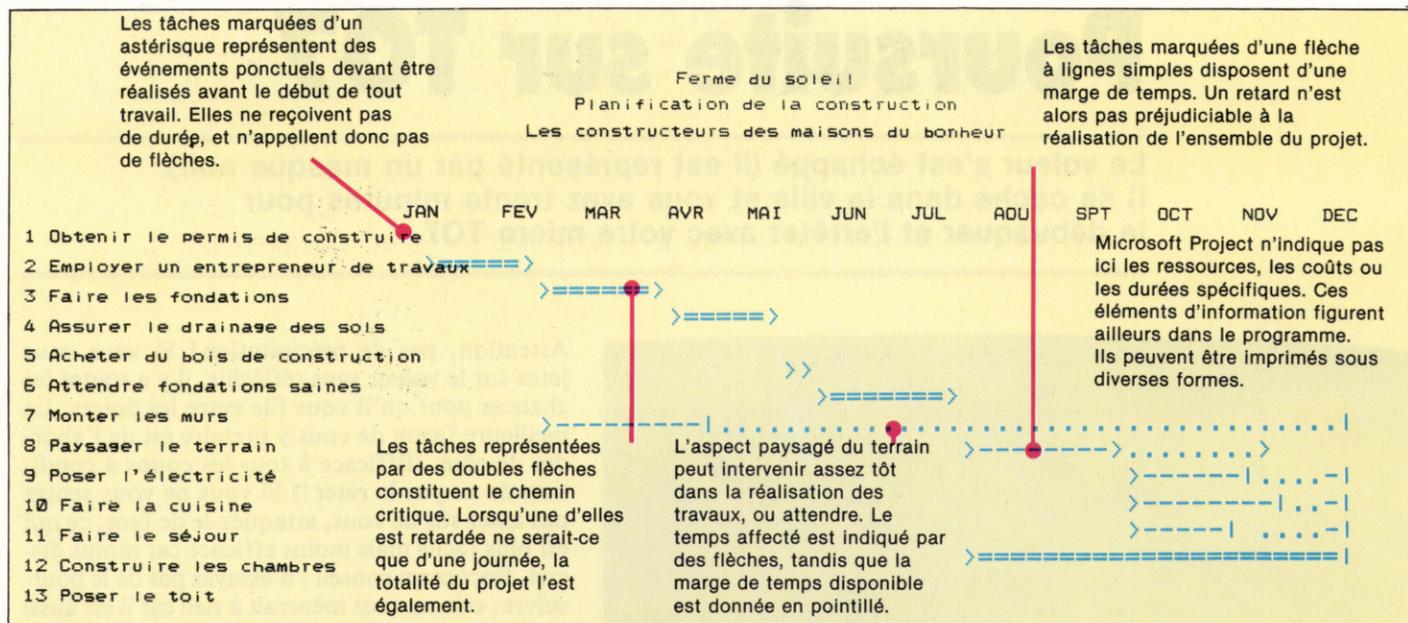
La date et l'heure sont données dans la séquence d'initialisation et déterminent le calendrier utilisé par le programme. Ce calendrier peut néanmoins être assez facilement modifié, même après saisie de plusieurs activités. Le diagramme du chemin critique est recalculé instantanément lorsque le temps alloué à une tâche est modifié. Certaines tâches deviennent cruciales, étant alors précédées d'une flèche en caractères gras ==>. Inversement, des tâches auparavant critiques redeviennent annexes ---->. Non seulement cela permet d'observer l'influence mutuelle des activités, mais encore, cela met en évidence l'importance de telles modifications. Il semble que Microsoft ait mis à profit son expérience de Multiplan avec Project. En effet, le plus célèbre tableur depuis le pionnier Visicalc en 1979 a inspiré les auteurs de Project quant à son mode d'utilisation. Aussi, quiconque sait exploiter un tableur, saura instantanément utiliser Project.

## Exploitation

Après affichage rapide du logo de Microsoft, le menu principal est proposé avec dix-neuf possibilités de tâches. Le menu ne s'affiche pas en l'absence de nom de fichier. Le menu des activités ou tableau se présente sous forme d'une grille allant de 1 à 19. L'échelle de temps est donnée en haut des colonnes ; elle commence par les date et heure données initialement. Un menu d'options en bas de l'écran permet en sélectionnant CHOIX de modifier les tâches données à la saisie. Ce menu comprend également les options suivantes :

EFFACER CALENDRIER SUPPRIMER ÉDITER ALLER-À AIDE  
INSERTION DÉPLACEMENT IMPRESSION QUITTER RESSOURCE  
TRI TRANSFERT EXTERNE

L'option en cours est affichée sous la légende « choisissez une option ou tapez la lettre correspondant à la commande ». L'option de début est



ACTIVITÉ. Le nom du projet en cours apparaît dans le coin inférieur droit. En l'absence de nom, le programme donne le titre TEMPORAIRE.

L'échelle de temps peut porter sur des jours, des semaines ou des mois, et cela peut être modifié à tout moment pour pouvoir visionner en un seul tableau l'ensemble du projet. Avec une échelle de temps portant sur des jours, on ne peut voir à la fois qu'une période portant sur deux mois alors qu'une unité de temps d'un mois permet en revanche de visionner à la fois deux ans et trois mois. Les touches fléchées droite et gauche correspondent à des déplacements droite et gauche sur l'écran ACTIVITÉ, à la manière d'un tableau. L'option CALENDRIER permet de tenir compte, dans la planification, des jours non travaillés. On peut modifier cette répartition, mais elle doit alors se faire pour chaque jour ainsi dénaturé. Les mois situés après la période affichée sont appelés à l'écran par l'option <défilement de la page vers le bas>, les mois précédant la période courante, par l'option <défilement de la page vers le haut>.

Comme pour la plupart des options concernant les dates sur les machines IBM PC et compatibles, il faut penser à utiliser l'ordre américain MM/JJ/AA (Mois/Jour/Année). Une fois le calendrier défini, il est possible de le sauvegarder au moyen de l'option TRANSFERT, selon la manière habituelle à Multiplan. Le programme ajoute à ce fichier le suffixe .CAL pour le distinguer des fichiers d'activité (suffixe .ACT) et ressource (.RES).

L'utilisateur peut alors commencer à saisir les tâches ou activités, accompagnées du temps qui leur est affecté. Le programme attribue alors aux activités situées en amont d'une nouvelle activité un certain nombre de relations séparées par des virgules. La colonne des titres d'activités est limitée à quinze caractères, sans possibilité d'extension. Cependant, lorsque plus de quinze caractères sont saisis pour un titre, les caractères excédentaires sont mis en mémoire, et restitués en cas

de recopie d'écran. Au fur et à mesure de la saisie de chaque nouvelle activité, avec sa durée, ses antécédents, et sa date de commencement, les relations ou rapports mutuels entre activités sont immédiatement donnés, avec la marge de temps dont la nouvelle activité dispose pour sa réalisation. Ces rapports sont indiqués par des tirets.

La table peut être imprimée dans sa totalité, selon le mode page à page, dès lors que toutes les données ont été saisies. Cependant, avec la carte graphique et une imprimante appropriée (l'imprimante graphique de l'IBM PC, ou la MX-80, MX-100, FX-80, FX-100 de Epson), la totalité du projet peut être rendue dans le sens de la largeur de la page. Cela est important dans la mesure où l'on peut afficher un tableau entier, sans avoir à faire des collages. Il est également possible d'obtenir des états imprimés individuels à chaque activité. Ces derniers font alors figurer les dates minimales et maximales de début et de fin pour l'activité, ses ressources, ses activités précédentes, la marge de temps dont elle dispose, ainsi qu'une table abrégée de la totalité du projet.

En fin de compte, le manque de ressources graphiques sophistiquées de Microsoft Project se révèle peu préjudiciable. En outre, bien que son excellente documentation insiste beaucoup sur la préparation initiale, les informations restent tout à fait modifiables par la suite. Les temps ne sont pas tous prévisibles non plus. Le programme sera particulièrement facile pour ceux qui connaissent les tableurs — et surtout Multiplan — du fait de sa grande ressemblance d'utilisation. Multiplan est bien sûr beaucoup plus rapide, mais il est aussi trois fois plus cher.

#### Projet de construction

Ce tableau représente la planification de la construction d'une maison. Les dates figurent en haut des colonnes. Les tâches sont données au début de chaque ligne de la grille selon l'ordre de leur exécution. La durée prévue pour chaque activité est représentée par des flèches tapées au clavier par des combinaisons de caractères de ponctuation (<>. = -).

**Microsoft Project : pour machines sous MS-DOS.**

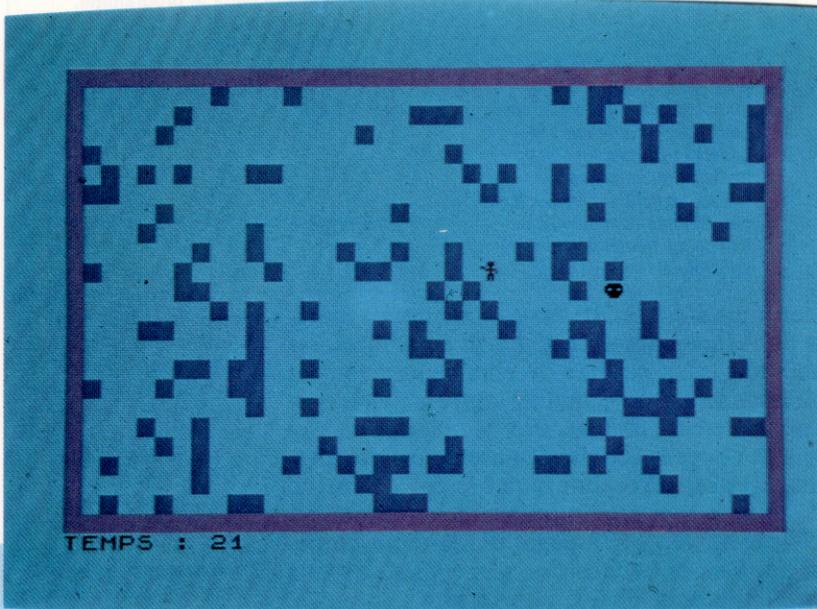
**Distributeurs : Microsoft.**

**Auteurs : MAS, SEATTLE, USA.**

**Format : disque.**

# Poursuite sur T07

Le voleur s'est échappé (il est représenté par un masque noir). Il se cache dans la ville et vous avez trente minutes pour le débusquer et l'arrêter avec votre micro T07.



Attention, pas de précipitation! Si vous vous jetez sur le voleur sans réfléchir, il y a toutes les chances pour qu'il vous file entre les doigts. La meilleure façon de vous y prendre est de l'aborder de côté. (Efficace à tous les coups à condition de ne pas le rater!) Si vous ne vous sentez pas assez sûr de vous, attaquez-le de face, ce qui est plus facile mais moins efficace car moins discret. Encore un conseil : n'essayez pas de le poursuivre; cela ne vous mènerait à rien car il est aussi rapide que vous. Observez plutôt ses mouvements comme un vrai détective. Quand vous le verrez tourner en rond, approchez-vous sans faire de bruit et surgissez au bon moment. Mais rappelez-vous, le temps presse!

Utilisez le joystick ou les touches suivantes : Z : haut, Q : gauche, S : droite, W : bas, pour vous déplacer.

```

10 REM *****
20 REM * POURSUITE *
30 REM *****
40 CLEAR ,,2
50 GOSUB 1320
60 S=0
70 N$=CHR$(32)
80 V$=GR$(1)
90 P$=GR$(0)
100 GOSUB 820
110 ON JK GOTO 180
120 D$=INKEY$
130 DH=(D$="Q")-(D$="S")
140 DV=(D$="Z")-(D$="W")
150 IF DH<>0 THEN DX=DH: DY=0
160 IF DV<>0 THEN DY=DV: DX=0
170 GOTO 210
180 ST=STICK(0)
190 DX=(ST=7)-(ST=3)
200 DY=(ST=1)-(ST=5)
210 Z=Z-.2
220 LOCATE 0,24
230 COLOR 0
240 PRINT "TEMPS :";INT(Z+1);
250 IF Z<0 THEN 500
260 PX=PX+DX
270 PY=PY+DY
280 C=POINT(PX*8+4,PY*8+4)
290 IF C=0 THEN 1270
300 IF C<>7 THEN PX=XP:PY=YP
310 LOCATE XP,YP
320 PRINT N$;
330 LOCATE PX,PY
340 COLOR 1
350 PRINT P$;
360 YP=PY
370 XP=PX
380 VX=VX+CX
390 VY=VY+CY
400 IF SCREEN(VX,VY)<>32 THEN GOSUB 660
410 IF SCREEN(VX,VY)<>32 THEN 380
420 LOCATE XV,VY
430 COLOR 0
440 PRINT N$;
450 LOCATE VX,VY
460 PRINT V$;
470 XV=VX
480 VY=VY
490 GOTO 110

```

```

500 IF INKEY$<>" " THEN 500
510 IF R<S THEN R=S
520 COLOR 0
530 LOCATE 10,6
540 PRINT "TEMPS ECOULE";
550 LOCATE 10,10
560 PRINT "SCORE :";S;
570 LOCATE 10,14
580 PRINT "RECORD :";R;
590 LOCATE 10,18
600 PRINT "UNE AUTRE ?";
610 D$=INKEY$
620 IF D$=" " THEN 610
630 IF D$<"N" THEN RUN
640 CLS
650 END
660 DT=DT+1
670 GOSUB 770
680 IF SCREEN(XV+CX,VY+CY)=32 THEN VX=XV
+CX:VY=VY+CY:RETURN
690 DT=DT-2
700 GOSUB 770
710 IF SCREEN(XV+CX,VY+CY)=32 THEN VX=XV
+CX:VY=VY+CY:RETURN
720 DT=DT-1
730 GOSUB 770
740 VX=XV+CX
750 VY=VY+CY
760 RETURN
770 IF DT>4 THEN DT=DT-4
780 IF DT<1 THEN DT=DT+4
790 CX=(DT=1)-(DT=3)
800 CY=(DT=2)-(DT=4)
810 RETURN
820 CLS
830 COLOR 5
840 FOR VX=0 TO 39
850 LOCATE VX,0
860 PRINT CHR$(127);
870 LOCATE VX,23
880 PRINT CHR$(127);
890 NEXT VX
900 FOR VY=1 TO 22
910 LOCATE 0,VY
920 PRINT CHR$(127);
930 LOCATE 39,VY
940 PRINT CHR$(127);
950 NEXT VY
960 COLOR 4

```

```

970 FOR VX=1 TO 150
980 GOSUB 1230
990 LOCATE PX,PY
1000 PRINT CHR$(127);
1010 NEXT VX
1020 GOSUB 1230
1030 VX=PX
1040 VY=PY
1050 COLOR 0
1060 LOCATE VX,VY
1070 PRINT V$;
1080 XV=VX
1090 VY=VY
1100 GOSUB 1230
1110 COLOR 1
1120 LOCATE PX,PY
1130 PRINT P$;
1140 XP=PX
1150 YP=PY
1160 Z=30
1170 CX=0
1180 CY=0
1190 DX=0
1200 DY=0
1210 DT=0
1220 RETURN
1230 PX=INT(RND*38)+1
1240 PY=INT(RND*22)+1
1250 IF SCREEN(PX,PY)<>32 THEN 1230
1260 RETURN
1270 FOR I=1 TO 5
1280 PLAY "T15REDO"
1290 NEXT I
1300 S=S+1
1310 GOTO 100
1320 SCREEN 4,6,6
1330 CLS
1340 DEFINT A-Y
1350 DEFGR$(0)=2B,2B,200,62,9,2B,20,20
1360 DEFGR$(1)=0,126,255,153,255,126,60,0
1370 ATTRB 1,1
1380 LOCATE 10,10,0
1390 PRINT "JOYSTICK ?";
1400 D$=INKEY$
1410 PX=RND
1420 IF D$=" " THEN 1400
1430 IF D$="D" THEN JK=1
1440 ATTRB 0,0
1450 RETURN

```



# Vous avez dit portable

**ACT (Apricot) a exploité les dernières avancées technologiques sur son portable : des liaisons sans câble, une réponse à des commandes vocales et l'utilisation d'une souris/boule roulante. Mais...**

Les machines portables représentent une part grandissante du marché de la micro-informatique. Les hommes d'affaires peuvent les emporter et les utiliser partout où un branchement est autorisé. En réalité, la machine n'est pas toujours utilisée de cette façon.

ACT (Apricot) est l'un des derniers fabricants d'ordinateurs (ils sont nombreux) à avoir produit une machine portable compatible avec son ordinateur de table le plus populaire.

Il est vendu avec un boîtier robuste en plastique — pour le transport — de la dimension et de la forme d'une mallette avec un renflement dans la partie inférieure pour accueillir l'écran de l'ordinateur.

## Analyse du matériel

A l'intérieur, le clavier et l'ordinateur sont maintenus au moyen de courroies. On trouve également dans le boîtier deux manuels, un microphone, et une souris/boule roulante. Comme dans le cas de l'Exelvision, aucun câble ne relie le clavier et la souris à l'ordinateur. Les données sont transmises par un faisceau infrarouge à l'aide de deux petites ampoules situées à l'arrière du clavier et à l'avant de la souris. Les signaux provenant de ces dispositifs sont captés par une grande ampoule située sur l'ordinateur. L'utilisateur peut donc employer le clavier là où il le désire, même si l'ordinateur se trouve à quelques mètres.

Ce système infrarouge comporte deux désavantages. D'abord, si plusieurs portables Apricot fonctionnent simultanément dans la même pièce, les signaux provenant de machines différentes peuvent créer des interférences. Pour remédier à ce problème, Apricot a fabriqué un câble à fibre optique. Celui-ci connecte le clavier à l'ordinateur de façon conventionnelle. Le second problème se pose lors de l'utilisation de la souris dans la mesure où le clavier peut faire écran s'il est interposé entre la souris et l'ordinateur, car son ampoule réceptrice est placée presque au niveau de la surface de la table. D'autre part, si vous déplacez le clavier sur le côté, l'ordinateur risque de ne pas pouvoir capter ses signaux.

Pour produire une version portable d'un ordinateur de table, on doit réunir toutes les caractéristiques dans un espace réduit. Le clavier présente des difficultés particulières, et les fabricants ont eu recours à toutes sortes de techniques ingénieuses pour loger les fonctions nécessaires sur un clavier plus petit et plus léger. En transposant

le clavier de l'Apricot dans la version portable, le fabricant a dégagé l'affichage à cristaux liquides et a resserré la position des quatre-vingt-sept touches. La disposition est identique à celle de l'Apricot de table, à l'exception des touches de fonction, qui sont situées à l'extrême droite, et non au-dessus du clavier.

Pour compenser partiellement l'insuffisance de l'affichage à cristaux liquides qui, dans la version de table, sert également pour les commandes à une touche, Apricot a ajouté une touche de fonction supplémentaire soit un total de neuf, plus une servant à afficher l'heure et la date. A gauche du clavier se trouvent des touches de commande de chaque côté; les touches de commande du curseur se trouvent au bas à droite. Entre les touches du clavier et les touches de fonction, on retrouve un clavier numérique dont les touches sont d'un gris plus pâle que le reste du clavier.

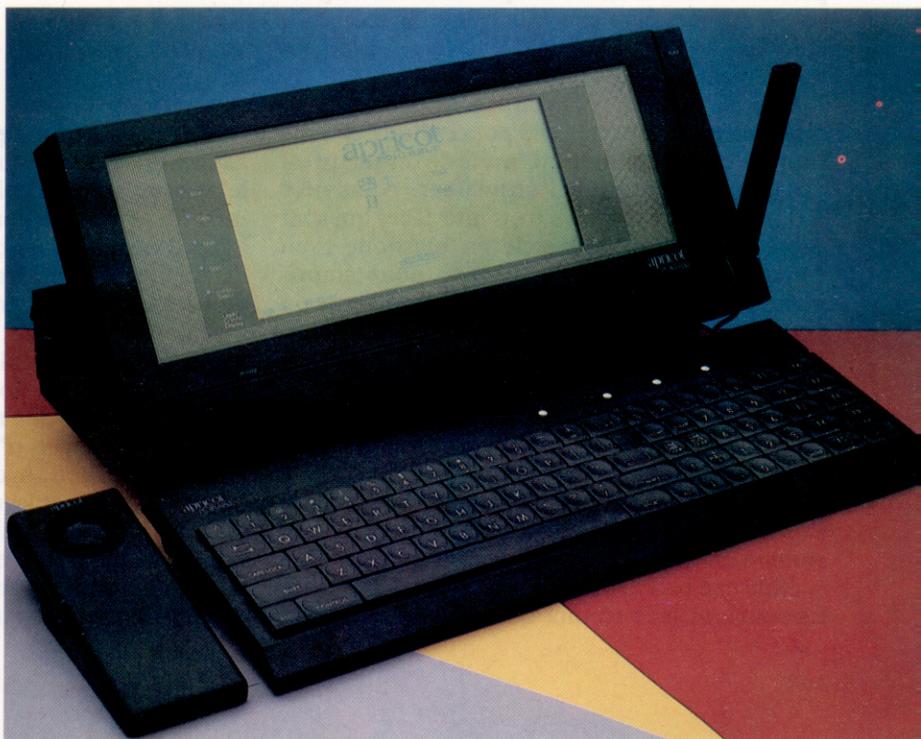
## Pas la touche!

Bien que le clavier soit bien conçu, les touches semblent être trop rapprochées pour taper confortablement. Au lieu d'être moulées conventionnellement, les touches du Portable sont plates. Cela rend le clavier plus compact, puisque celles-ci ne dépassent pas le niveau du corps principal du clavier. Mais beaucoup éprouveront de la difficulté à utiliser le clavier, puisqu'il est difficile de localiser les touches. Leur proximité

### Déplacement facile

Le Portable Apricot utilise les tout derniers développements technologiques.

La machine est dotée d'un écran texte 80 x 25 qui permet d'afficher une large variété d'applications MS-DOS. La photo présente également le clavier et la souris/boule roulante : ces dispositifs ne sont pas reliés à l'ordinateur de façon conventionnelle, mais transmettent des signaux au moyen d'un faisceau infrarouge, comme pour le micro français, Exelvision.





accentue ce problème : il n'y a pas d'espace entre le clavier, la calculatrice et les touches de fonction. La disposition des touches offre cependant certaines compensations. Les touches Shift, Stop et Caps Lock sont deux fois plus grosses que sur l'Apricot, ce qui facilite leur localisation.

Au-dessus du clavier et à l'emplacement de l'affichage de la calculatrice, on retrouve quatre boutons permettant d'exécuter des fonctions supplémentaires. Le premier est un bouton de remise à zéro qui réamorçe la disquette présente dans le lecteur. Le deuxième bouton permet à l'utilisateur de modifier la vitesse de répétition des touches. Le troisième appelle une routine qui règle l'heure mémorisée par l'ordinateur, et le quatrième verrouille le clavier. Ce dernier bouton est utile si vous utilisez la souris et désirez empêcher qu'une touche soit pressée inopinément.

L'écran à cristaux liquides intégré est situé sur la partie principale de l'ordinateur. Sa définition est moins bonne que celle de machines comparables, bien que le fond vert se dégage mieux que la plupart des autres. Un microphone se trouve à la droite de l'écran; il est connecté au moyen d'un fil très fin qui se branche dans une prise microjack située à gauche de l'ordinateur.

## Reconnaissance de la parole

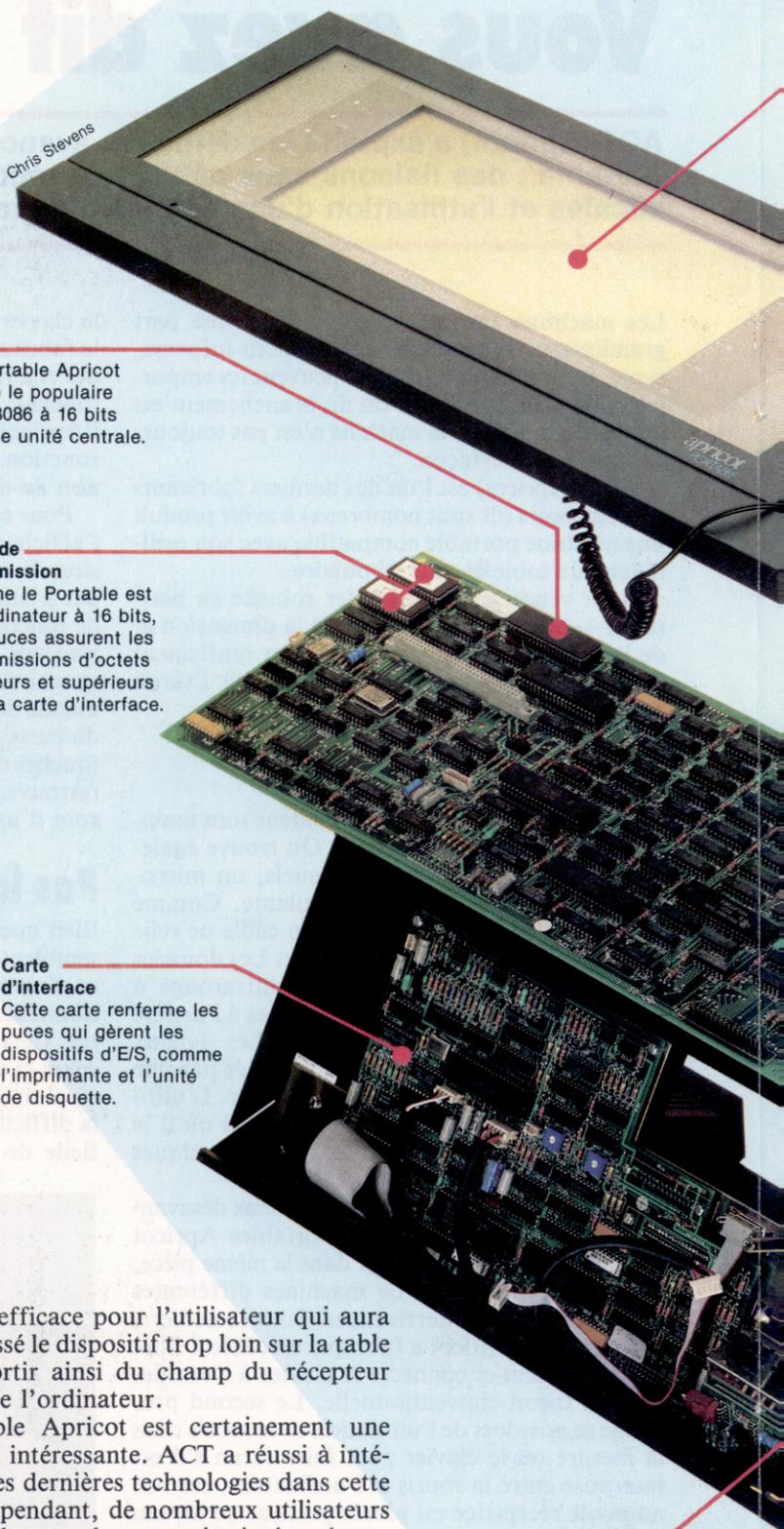
La caractéristique probablement la plus intéressante du portable est le programme de reconnaissance de la parole qui, en reconnaissant des commandes vocales, les transmet comme entrées et les exécute normalement. Le mode Voice Driven Application accepte jusqu'à soixante-trois mots pour tout programme d'application, comme WordStar ou SuperCalc. Un programme demande à l'utilisateur si la voix utilisée est celle d'un homme, d'une femme ou d'un enfant, et s'il y a beaucoup de bruit de fond. Après avoir prononcé une commande à plusieurs reprises, l'ordinateur accepte et compare plusieurs variantes. Cependant, il peut lui arriver de mal interpréter certains mots ou simplement de les ignorer. Pour cette raison, les commandes DELETE et FORMAT ne devraient pas être données vocalement.

Comme le reste de la gamme Apricot, le Portable utilise les disquettes 3 pouces Sony comme support de programmes. L'unité de disquette est placée au côté droit de l'ordinateur. A l'arrière de la machine, on trouve trois interfaces qui sont protégées par un couvercle en plastique : un connecteur parallèle Centronics pour imprimante, une prise RS232 pour modem externe ou pour un autre dispositif série et un port de manche à balai de type Atari. Ce port n'est pas destiné au jeu mais sert plutôt à brancher une souris ou une boule roulante dans les cas où l'utilisation de la souris infrarouge n'est pas appropriée.

La souris infrarouge à piles fournie avec le Portable est très bien conçue et peut être utilisée comme une souris sur la surface d'un bureau ou comme une boule roulante si on la tient à la main. La souris utilisée en tant que boule roulante est

en fait plus efficace pour l'utilisateur qui aura souvent poussé le dispositif trop loin sur la table le faisant sortir ainsi du champ du récepteur infrarouge de l'ordinateur.

Le Portable Apricot est certainement une machine très intéressante. ACT a réussi à intégrer les toutes dernières technologies dans cette machine. Cependant, de nombreux utilisateurs professionnels sont plus ou moins intéressés par les technologies de pointe; ce qu'ils recherchent avant tout, c'est une machine fiable qui exécute leurs programmes d'application facilement et confortablement. A ce niveau, l'Apricot laisse entrevoir certaines lacunes. Les difficultés de lecture d'écran et les faiblesses quant à la fiabilité des dispositifs d'entrée pourraient décourager des utilisateurs professionnels préférant opter pour des machines plus sûres.



Chris Stevens

**UC**  
Le Portable Apricot utilise le populaire Intel 8086 à 16 bits comme unité centrale.

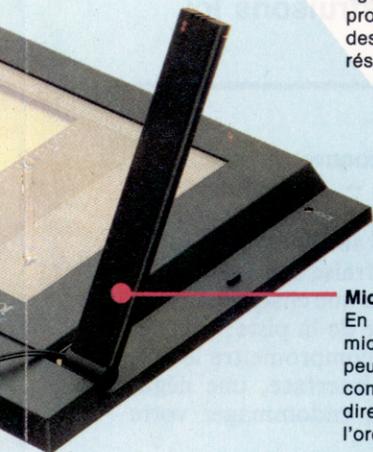
**ROM de transmission**  
Comme le Portable est un ordinateur à 16 bits, ces puces assurent les transmissions d'octets inférieurs et supérieurs vers la carte d'interface.

**Carte d'interface**  
Cette carte renferme les puces qui gèrent les dispositifs d'E/S, comme l'imprimante et l'unité de disquette.

**Unité de disquette**  
L'ordinateur a une seule unité de disquette double face, double densité, qui est compatible avec tous les autres produits Apricot.

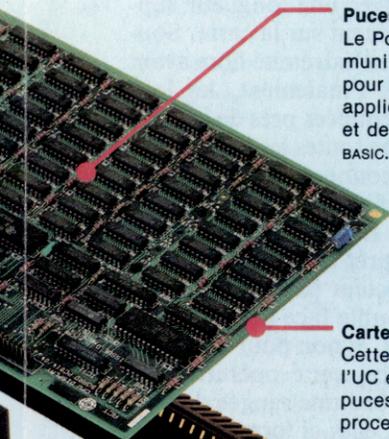


**Affichage à cristaux liquides**  
Cet écran produit un affichage de 80 par 25 caractères. Il est également possible de produire sur cet écran des graphiques haute résolution.

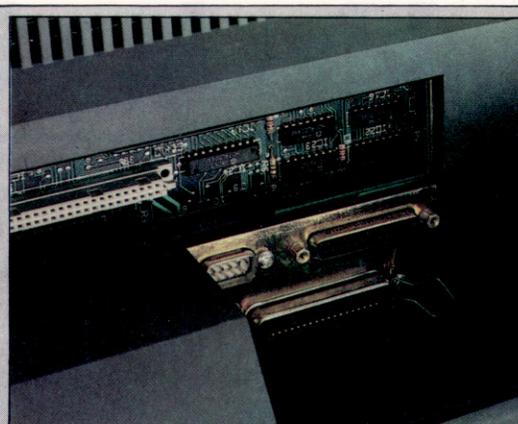


**Microphone**  
En parlant dans le microphone, l'utilisateur peut donner des commandes vocales directement à l'ordinateur.

**Puces RAM**  
Le Portable Apricot est muni de 256 K de RAM pour exécuter des applications de gestion et des programmes BASIC.



**Carte UC**  
Cette carte renferme l'UC et toutes les autres puces utilisées par le processeur.



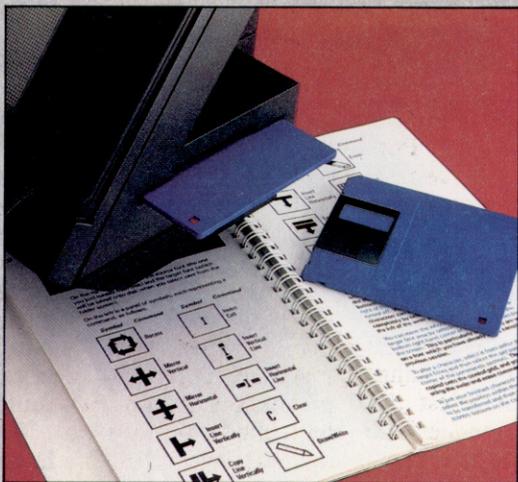
#### Interfaces de périphériques

A l'arrière de l'ordinateur, protégées par un boîtier en plastique, on retrouve les interfaces de périphériques. Le port parallèle Centronics est situé tout en bas. Au-dessus et à droite, on aperçoit la prise RS232 (qui permet de connecter l'ordinateur à un modem externe) et à gauche, un port pour une souris.



#### Boule roulante infrarouge

En faisant tourner la boule, le curseur se déplace dans la direction correspondante sur l'écran. Lorsque le curseur se trouve sur l'image choisie, il suffit d'appuyer sur l'un des boutons latéraux pour appeler l'application. Le dispositif peut être retourné afin que la boule roule sur la surface de la table, à la manière d'une souris.



#### Logiciels Apricot

De nombreux logiciels d'application permettent à l'utilisateur d'effectuer diverses tâches. Ces programmes sont livrés sur disquettes 3 pouces Sony. Parmi ces programmes, mentionnons SuperCalc (un tableur), SuperPlanner (un répertoire et un calendrier), et SuperWriter (un programme de traitement de texte).

## PORTABLE APRICOT

### PRIX

★ ★ ★ ★

### DIMENSIONS

450 × 335 × 200 mm.

### UC

Intel 8086.

### MÉMOIRE

32 K de ROM et 256 K de RAM, dont 211 disponibles pour des programmes d'application.

### ÉCRAN

Écran à cristaux liquides électrofluorescent produisant une résolution texte de 80 × 25 caractères et une résolution graphique de 640 × 256 points. Branché sur un moniteur externe, le Portable affiche jusqu'à 16 couleurs.

### INTERFACES

Série RS232, parallèle Centronics et une connexion « souris ». Connexion fibre optique également pour le clavier et prise pour le microphone.

### UNITÉS DE DISQUETTES

Le Portable possède une unité de disquette 3 pouces double face Sony de 720 K.

### SYSTÈME D'EXPLOITATION

MS-DOS, CP/M-86 et autres CP/M.

### CLAVIER

Clavier compatible IBM à 87 touches dont 10 de fonction. Il y a aussi quatre boutons de fonction.

### DOCUMENTATION

Le manuel du débutant donne tous les détails concernant le montage de l'ordinateur et l'utilisation du système d'exploitation et des fonctions de reconnaissance de la parole. Le manuel des applications explique comment utiliser les programmes SuperCalc, SuperPlanner et SuperWriter.

### FORCES

Le Portable Apricot est certainement l'une des machines les plus perfectionnées sur le marché. La technologie mise au point pour la conception de cet ordinateur séduira de nombreux utilisateurs.

### FAIBLESSES

La technologie de pointe utilisée n'est pas encore infaillible. L'écran à cristaux liquides est très difficile à lire.



# Comblar le vide

**Nous avons expliqué comment concevoir une interface qui joue le rôle d'un port utilisateur pour le Spectrum. Nous construisons ici l'interface qui permettra de commander le robot.**

Le connecteur plat conçu pour le Spectrum doit être obstrué à la cinquième position à partir de la gauche (en regardant vers le connecteur). Notre première tâche est donc d'insérer une fiche d'obstruction dans le connecteur plat qui sera connecté dans le port d'extension du Spectrum. Cette fiche assure que l'interface construite ne peut être insérée que dans la bonne orientation.

Après avoir coupé la carte de montage suivant la forme donnée ici, l'un des coins peut servir de fiche d'obstruction. Enlevez d'abord la piste de cuivre et enfoncez le bout de carte dénudée dans

la cinquième position du connecteur plat — en utilisant un peu de colle —, puis coupez les connexions des deux broches. Effectuez maintenant les coupures de pistes sur la carte de montage, en utilisant soit une fraise soit une mèche torche de 5 mm. Veillez à ce que chaque coupure sectionne la largeur entière de la piste; cela est important, car en plus de compromettre le bon fonctionnement de votre interface, une négligence à ce stade pourrait endommager votre Spectrum.

Une fois les coupures de pistes effectuées, nous commençons à assembler les composants sur la carte, en débutant par les liaisons câblées. Les fils doivent être coupés avec une longueur supplémentaire et posés librement sur la carte. Soudez une extrémité et tendez l'extrémité libre avant de faire l'autre soudure. Finalement, les longueurs superflues seront coupées près de la carte avec un coupe-fils. Lorsque toutes les liaisons ont été soudées, branchez le connecteur isolé en utilisant un court brin de câble ruban.

Après avoir fixé toutes les liaisons, le connecteur plat est maintenant prêt à être soudé. Mais puisque le système connecteur plat est en réalité conçu pour des cartes double face (comme celle utilisée dans le Spectrum), et non pour une carte de montage à simple face, cette opération est assez délicate. Pliez d'abord une rangée de broches du connecteur plat. Celles-ci forment la rangée inférieure lorsque le connecteur plat est positionné sur la carte. Maintenant, soudez des bouts de fils de cuivre de 3 cm sur chacune de la rangée supérieure. Insérez les broches pliées dans la rangée de trous située près du bord de la carte et pliez chaque fil à 90°, en les insérant dans l'autre rangée de trous (indiquée sur le diagramme).

Mettez maintenant un peu de colle sur le bord de la carte entre les deux coins coupés, et pressez le connecteur plat contre la carte. Soudez soigneusement chaque broche aux pistes de la carte. Avant de poursuivre, examinez le côté cuivre de la carte et enlevez à l'aide d'un couteau affûté toute soudure shuntant les pistes.

Soudez maintenant le reste des composants sur la carte. Les trois prises DIL seront traitées en premier, puis les quatre résistances. Finalement, insérez les trois puces, en veillant à les mettre dans le bon sens.

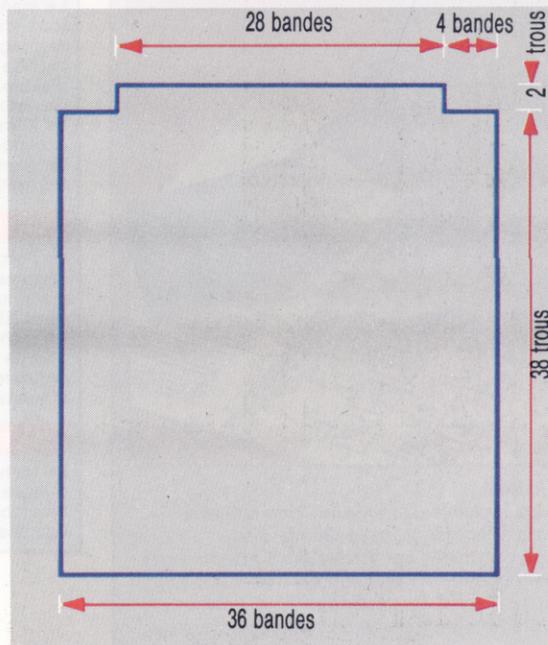
Les encoches apparaissant à l'une des extrémités des puces doivent être toutes dirigées dans la même direction — vers la droite lorsque le connecteur plat est tout en haut de la carte.

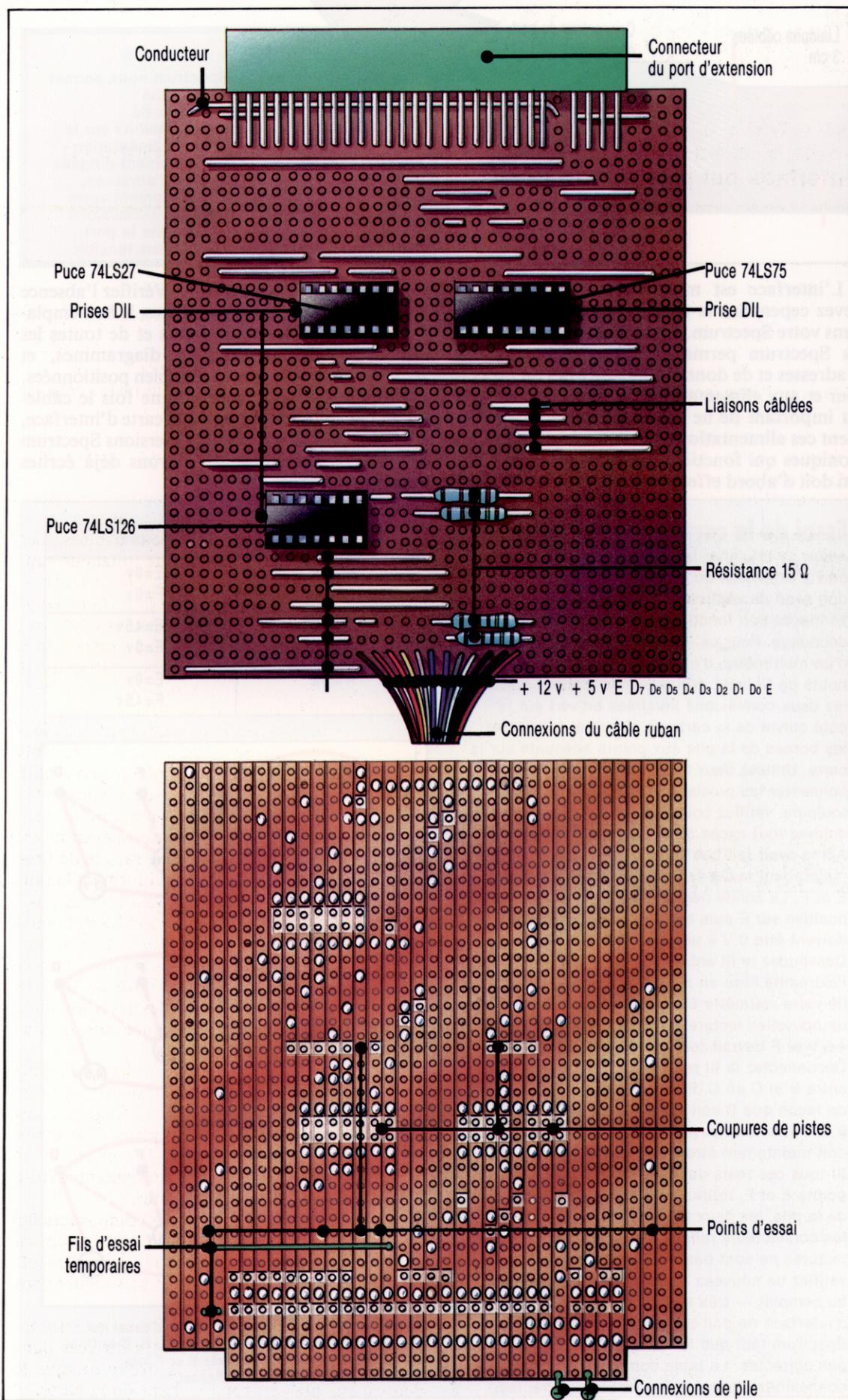
## Liste des pièces

Nombre	Article
1	74LS27
1	74LS75
1	74LS126
4	15 kΩ, 0,3 watts
1	Bobine de fil de cuivre 20 swg
2	Prise DIL 14 voies
1	Prise DIL 16 voies
5 m	Câble ruban 20 voies
1	Connecteur plat 2 × 28 voies
1	Connecteur D 15 voies
1	Couvercle de connecteur D 15
1	Carte de montage 36 × 50

### Remarques concernant la découpe

Les composants de l'interface du Spectrum se montent sur une carte de montage coupée comme il est indiqué ici. Utilisez l'un des coins comme fiche d'obstruction dans le connecteur du port d'extension. C'est sur ce bord découpé que nous placerons le connecteur lorsque nous le souderons.





## Disposition sur la carte

Soudez d'abord les liaisons câblées et les résistances sur le côté composant de la carte, puis installez le connecteur plat et les prises DIL. Veillez à ce que les trois puces soient enfichées dans le bon sens dans les prises DIL; Les encoches seront dirigées dans la même direction que sur l'illustration. Comparez soigneusement votre travail au diagramme avant de tenter de brancher la carte d'interface dans votre Spectrum.

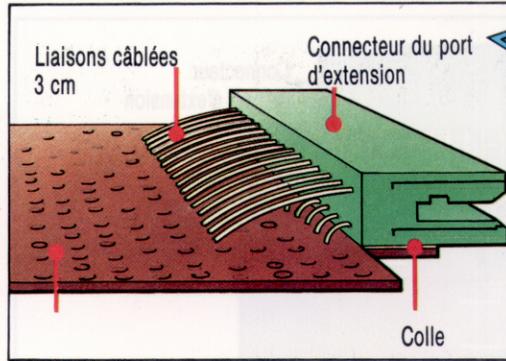
Sur le côté cuivre, veillez à ce que toutes les coupures de pistes aient sectionné entièrement la piste; utilisez un voltmètre pour vérifier la continuité. Les liaisons marquées en vert sont des connexions temporaires destinées à une vérification électronique de la carte et sont enlevées à la fin d'un essai positif. Comme toujours, vérifiez soigneusement l'ensemble de votre travail.



**Vue latérale**

Pour nous permettre d'utiliser un connecteur plat standard sur la carte d'interface, nous devons apporter de légères modifications aux broches. Après avoir enlevé celles de cinquième position, pliez les broches inférieures à angle droit et soudez des bouts de 3 cm de fil aux broches supérieures. Poussez-les dans les trous correspondants de la carte, en vérifiant la localisation de chacune. La rangée inférieure doit être connectée dans la troisième rangée de trous, et la rangée supérieure doit être connectée dans la sixième rangée de trous. Collez le connecteur avant de souder les broches.

(Cl. Kevin Jones.)



L'interface est maintenant terminée. Vous devez cependant la tester avant de la brancher dans votre Spectrum. Puisque le port d'extension du Spectrum permet un libre accès aux bus d'adresses et de données, aux broches du processeur et aux alimentations de 12 et de 9 volts, il est important de ne pas connecter accidentellement ces alimentations aux délicats circuits électroniques qui fonctionnent à des niveaux TTL. On doit d'abord effectuer un test visuel. Vérifiez

**Avertissement**

Le port d'extension du Spectrum nous permet d'avoir accès aux délicats circuits électroniques situés à l'intérieur de l'ordinateur. Sont également présents sur le port d'extension des sorties d'alimentation qui, si elles étaient accidentellement dirigées vers les lignes de données ou d'adresses, pourraient endommager sérieusement votre ordinateur. Vérifiez TOUTES les connexions avant de brancher l'interface dans le port utilisateur et avant de mettre sous tension.

soigneusement la carte entière. Vérifiez l'absence de pont de soudure entre les pistes, le bon emplacement de tous les composants et de toutes les liaisons, conformément aux diagrammes, et assurez-vous que les puces sont bien positionnées.

Nous installerons la prochaine fois le câble-ruban qui connecte le robot à la carte d'interface, et nous examinerons certaines versions Spectrum des programmes, que nous avons déjà écrites pour notre robot.

**Essai de la carte**

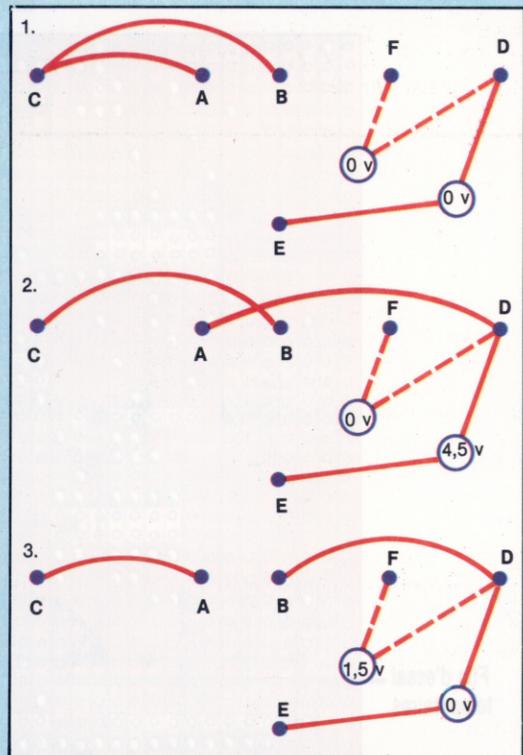
Avant de brancher la carte d'interface dans le port d'extension du Spectrum, nous effectuons une série de vérifications électriques pour vérifier le bon fonctionnement de la logique de décodage. Pour ce faire, nous avons besoin d'un multimètre, d'une pile 4-5 V, et de plusieurs bouts de fil isolé. Nous devons d'abord établir les deux connexions illustrées en vert sur le côté cuivre de la carte de circuit et connecter les bornes de la pile aux points adéquats sur la carte. Utilisez deux bouts de fil isolé pour connecter les points A et B au point C. Comme toujours, vérifiez soigneusement votre travail et enlevez tout excès de soudure entre les pistes. Après avoir fait ces connexions, nous devons maintenant tester les niveaux de tension en E et F. La sonde négative sur D, posez la sonde positive sur E puis sur F. Ces deux lectures doivent être 0 V à un volt près.

Dessoudez le fil entre A et C et reconnectez l'extrémité libre au point D. La sonde négative de votre voltmètre étant toujours en D, prenez de nouvelles lectures en E et en F. E doit être 4-5 V et F devrait toujours être 0 V.

Déconnectez le fil reliant A à D en D, et le fil entre B et C en C. Placez les extrémités de fils de façon que B soit relié à D et que A soit relié à C. Faites de nouvelles lectures en E et en F. F doit maintenant être 4-5 V et E doit être 0 V.

Si tous ces tests donnent les bons résultats aux points E et F, retirez de la carte les connexions de la pile, les deux liaisons câblées vertes et les connecteurs reliant A et C, et B et D. Si les lectures ne sont pas conformes aux prévisions, vérifiez de nouveau la construction de la carte au complet — très soigneusement. La carte d'interface ne doit pas être connectée au Spectrum tant que les lectures d'essai ne sont pas correctes. La table donne les diverses connexions d'essai et les lectures correctes.

Connexions	Lectures
A-C, B-C	E=0 v F=0 v
A-D, B-C	E=4,5 v F=0 v
A-C, B-D	E=0 v F=4,5 v



Ce diagramme montre les connexions d'essai qui doivent être établies entre les points A, B, C et D, et les lectures obtenues aux points E et F lors de chacune des trois étapes d'essai.



# De l'organigramme au basic

Les symboles utilisés dans la construction des organigrammes trouvent une parfaite correspondance au niveau des instructions du langage BASIC.

Tous les symboles que nous avons jusqu'à présent associés essentiellement à des idées peuvent être assimilés à des instructions précises. Ainsi, le symbole d'entrée par clavier se codifiera avec le commandement INPUT (voir 1); le rectangle, symbole représentant une opération, correspondra à LET (voir 2). La plupart des systèmes se dispenseront de leur utilisation en préférant une représentation sous forme algébrique :  $A = B$  équivaut à LET  $A = B$ .

Le losange (symbole de décision) se traduit par IF (voir 3). La visualisation et l'impression, symboles (voir 4) ou (voir 5), se transcrivent par PRINT, alors que les lignes de flux et les connecteurs sont représentés par GOTO. En ce qui concerne les actions « terminales », le final (voir 6) équivaut à END (dans le Spectrum, STOP), et le démarrage peut se représenter avec un REM, qui servira également pour inclure d'éventuels commentaires. Prenons un exemple.

Nous allons établir la visualisation des nombres pairs compris entre 2 et 50, ainsi que leurs carrés (fig. 1). On a utilisé A pour dénommer le compteur et C pour dénommer le carré. Les lignes 60 et 70, qui expriment la décision, peuvent se transcrire ainsi :

```
60 IF A < 50 OR A = 50 THEN GOTO 30
70 END
```

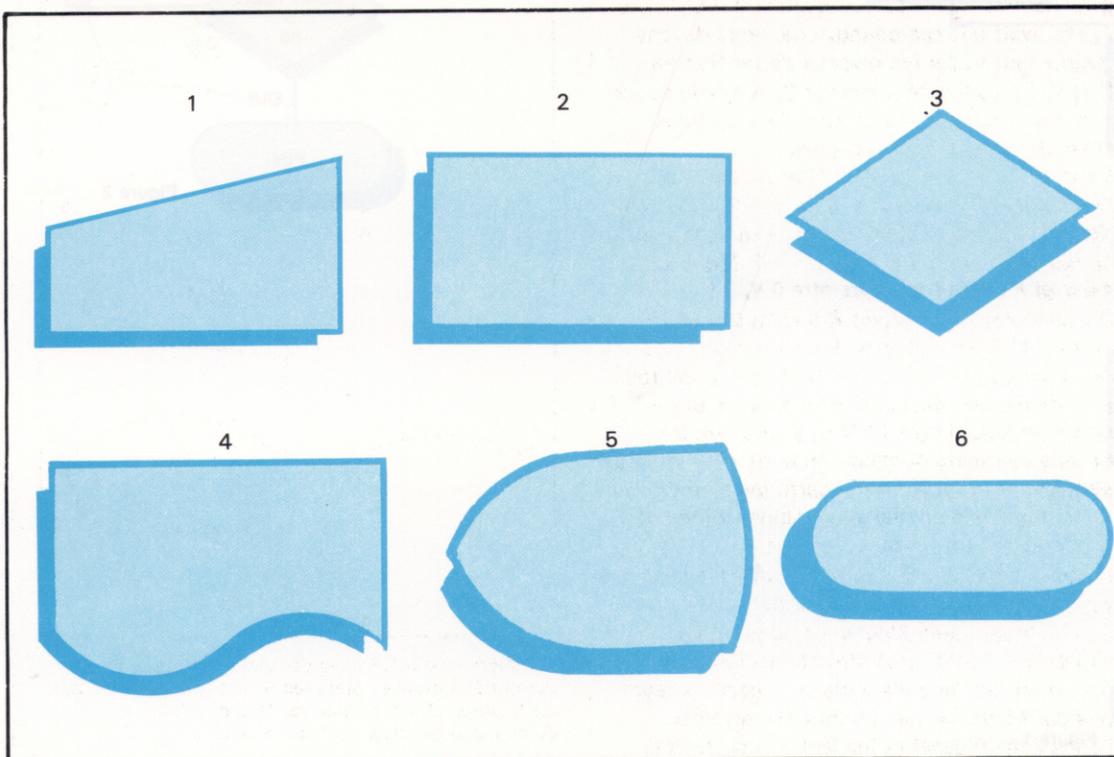
ou bien

```
60 IF A > 50 THEN END
70 GOTO 30
```

Modifions la situation, en supposant qu'au lieu de deux nombres déterminés on introduise, par le clavier, deux chiffres quelconques (fig. 2). On observe alors que la première quantité doit toujours être paire, pour que l'accroissement de deux en deux redonne le résultat souhaité. Mais cet objectif n'est pas signifié à la personne qui, elle, entre les chiffres qu'elle désire.

C'est à partir du programme que l'on doit contrôler cette condition. De même, il est nécessaire de vérifier que, étant donné l'accroissement du compteur, la première quantité doit être inférieure à la deuxième; dans le cas contraire, il faudra inverser l'ordre des quantités. De cette façon, les nombres pairs apparaîtront toujours en ordre croissant.

```
10 REM NOMBRES PAIRS ET CARRÉS
20 INPUT «PREMIERE QUANTITÉ»;A
30 INPUT «DEUXIEME QUANTITÉ»;B
35 REM CONTRÔLE DES QUANTITÉS
```





```

40 IF A<B THEN GOTO 60
50 X=B: B=A: A=X
55 REM CONTRÔLE PARITÉ DE A

60 IF A/2=INT (A/2) THEN GOTO 80
70 A=A+1

```

```

75 REM DÉBUT DE BOUCLE
80 C=A+A
90 PRINT A,C
100 A=A+2
110 IF A>B THEN END
120 GOTO 80

```

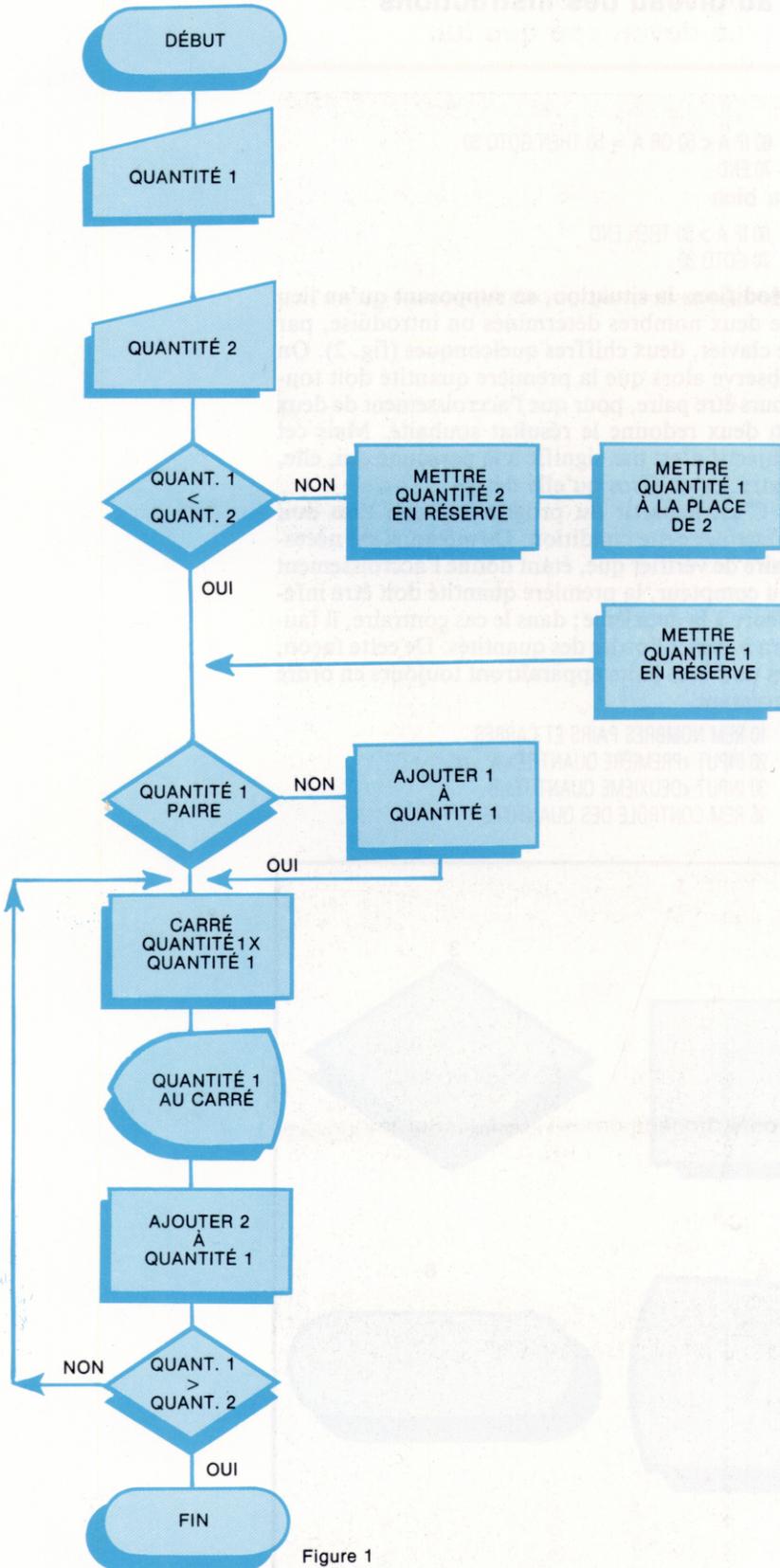


Figure 1

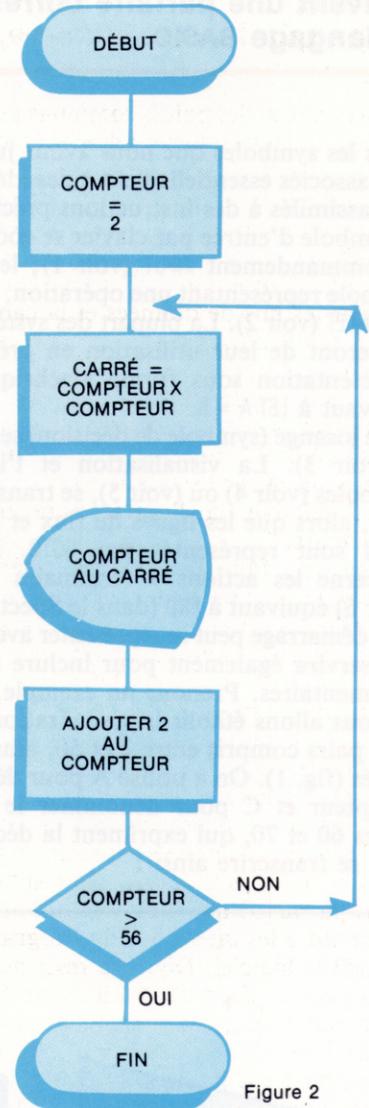


Figure 2



# Digitaya

**Nous terminons notre série sur la programmation des jeux d'aventures avec le listage complet de Digitaya, présenté en même temps que la Forêt hantée, mais bien plus développé que lui.**

Digitaya a bien des points communs avec la Forêt hantée; c'est simplement l'échelle du jeu qui change, où l'on ne compte pas moins de cent lieux différents sur le parcours qui est proposé. Vous vous retrouvez à l'intérieur d'un ordinateur, et devez en parcourir tous les éléments (depuis la mémoire jusqu'au processeur, en passant par les bus de données et la carte qui gère les dispositifs d'entrée/sortie) avant de retrouver la mystérieuse Digitaya, qui est retenue prisonnière quelque part dans la machine, elle aussi. Vous serez amené à traverser le port joystick (manche à balai), à affronter des bogues aléatoires, et vous aurez besoin de toutes vos connaissances relatives à l'architecture interne des ordinateurs.

Digitaya fait usage des mêmes sous-programmes de base que la Forêt hantée : aller d'un lieu à l'autre, prendre et abandonner des objets, regarder autour de soi. Mais nous leur avons ajouté d'autres modules spécifiques, qui permettent de gérer les nombreux pièges et dangers qui guettent l'aventurier.

N'oubliez jamais que la méthode la plus sûre pour jouer à un jeu d'aventures consiste à prendre un papier et un crayon, et à faire un plan des lieux traversés. Refaire soi-même un plan du jeu est souvent un bon moyen pour arriver au but. C'est, par la même occasion, un bon moyen de comprendre les intentions du programmeur qui a conçu le logiciel. Digitaya reste assez simple, et un plan en deux dimensions suffira. Mais il serait tout à fait possible, en reprenant sa structure de base, d'écrire un jeu d'aventures à trois dimensions. On pourrait même envisager d'en ajouter une quatrième, qui serait le temps. Le plan des lieux — à supposer qu'on puisse en établir un — serait de la sorte soumis à des changements incessants, et seuls les aventuriers confirmés seraient capables d'en venir à bout.

## Trois dimensions?

Le listage est destiné au Commodore 64, mais il peut tourner sur tout appareil pourvu d'un BASIC de type Microsoft avec des modifications mineures.

En tapant le programme, prenez garde aux REM dont il est parsemé. La plupart donnent le nom d'un sous-programme, et le numéro de ligne correspondant est employé quelque part dans un GOSUB. Pour éviter le message d'erreur UNDEFINED STATEMENT, et pour vous épargner du travail (et de l'espace mémoire), vous pouvez vous borner à taper 4660 REM :

## Variantes de basic

### Spectrum

Le Spectrum manipule les tableaux et les variables de chaînes de caractères de façon bien particulière; et les modifications à apporter au listage de Digitaya sont trop nombreuses pour être toutes mentionnées. Reportez-vous aux exemples donnés précédemment. Voici toutefois quelques conseils qui vous aideront à procéder aux transpositions nécessaires. Tout d'abord, chaque variable de chaîne ne peut comporter qu'une seule lettre. A l'occasion de la publication du listage de la Forêt hantée, nous vous avons donné un tableau de conversion que nous vous suggérons de consulter de près. Par ailleurs, le Spectrum n'autorise que des tableaux à longueur fixe, laquelle est déterminée par une instruction DIM. Cela peut parfois être très gênant. Disons que cette longueur est fixée à 20 caractères. Si une chaîne particulière n'en comporte que quinze, elle sera automatiquement suivie de cinq blancs, qu'il faudra faire disparaître avant d'ajouter cette chaîne à la phrase que nous sommes en train de construire. Nous aurons donc besoin d'une variable A\$ qui fasse passer cette chaîne dans le sous-programme, et qui soit déclarée avant tout appel de ce sous-programme. Un exemple permettra de bien comprendre le procédé.

### Version Microsoft :

```
3650 SN$=IV$(F,1)+«NE SERT À RIEN, LA FORCE AUGMENTE»
```

### Version Spectrum :

```
3650 A$=IV$(F,1) : LET S$=A$ : GOSUB 8500 :  
LET S$=S$+«NE SERT À RIEN, LA FORCE AUGMENTE»
```

Le seul autre problème est lié à l'effacement de l'écran. Sur le Commodore 64, on y procède par PRINT CHR\$(147). Sur le Spectrum, il suffit de remplacer toutes les instructions de ce type par CLS.

### BBC Micro

Les lignes suivantes doivent être substituées à celles du listage d'origine :

```
1400 A$=GET$  
1410 CLS  
2630 REPEAT:A$=GET$:UNTIL A$=«O»OR A$=«N»  
2750 RA=RND(1)  
2820 P=RND(40)+7  
3890 RD=RND(1):IF RD>.65 THEN 4110:BEM TOUCHE  
4090 P=RND(40)+7  
4520 IV$(4,2)=STR$(RND(40)+7):REM NOUVELLE POSITION POUR  
LE TICKET  
4570 RN=RND(3)+1  
5560 RA=RND(1)
```

# L'aventure de Digitaya

```

1030 REM ** DIGITAYA **
1040 REM ** JEU D'AVENTURE **
1050 REM ** SUR ORDINATEUR **
1090 :
1100 GOSUB6090:REM READ LECTURE DATA
1110 GOSUB1290:REM ON Y VA !
1120 P=A7:REM POINT DE DEPART
1130 :
1140 REM ***** DEBUT BOUCLE PRINCIPALE *****
1150 :
1160 MF=0:PRINT
1170 GOSUB1440:REM DESCRIPTION POSITION
1180 GOSUB1560:REM SORTIES
1190 GOSUB2670:REM LIEU SPECIAL ?
1200 IF SF=1 THEN 1250:REM NOUVELLE BOUCLE
1210 PRINT:INPUT"INSTRUCTIONS":IS
1220 GOSUB1700:REM ANALYSE INSTRUCTIONS
1225 IF F=0 THEN 1210:REM NON VALABLE
1230 GOSUB 1900:REM INSTRUCTIONS NORMALES
1240 IF VF=0 THENPRINT"JE NE COMPRENS PAS"
1250 IF MF=1 THEN 1160:REM NOUVELLE POSITION
1260 IF MF=0 THEN 1210:REM NOUVELLE INSTRUCTION
1270 END
1280 :
1290 REM ***** ON Y VA ! *****
1300 SN$="BIENVENUE A 'DIGITAYA'"
1310 GOSUB5880:REM FORMAT
1320 PRINT
1330 SN$=" L'APPAREIL BOURDONNE ET VOUS REGARDEZ AUTOUR
DE VOUS"
1340 SN$=SN$+ " AU NORD ET AU SUD PASSE UNE AUTOROUTE".
1350 SN$=SN$+ " VOTRE MISSION EST DE RETROUVER DIGITAYA ET"
1360 SN$=SN$+ " DE RESSORTIR PAR L'UN DES PORTS DE SORTIE."
1370 SN$=SN$+ "... MAIS LEQUEL ?"
1380 GOSUB5800
1390 PRINT:PRINT "APPUYEZ SUR UNE TOUCHE"
1400 GETA:IFA$=""THEN:400
1410 PRINTCHR(147):REM EFFACE ECRAN
1420 RETURN
1430 :
1440 REM ***** S/P DESCRIPTION POSITION *****
1450 SN$=" VOUS ETES" + LN$(P):GOSUB5880
1460 SN$=" CE QUE VOUS VOYEZ:"
1470 REM ** RECHERCHE OBJET **
1480 F=0:SP$=""
1490 FOR I=1TO9
1500 IF VAL(IV$(I,2))=P THEN SN$=SN$+SP$+V$(I,1):F=1:SP$=""
1510 NEXT I
1520 IF F=0 THENSN$=SN$+ "... RIEN DU TOUT."
1530 GOSUB5880:REM FORMATAGE
1540 RETURN
1550 :
1560 REM ***** S/P SORTIES *****
1570 EX$=EX$(P)
1580 NR=VAL(LEFT$(EX$,2))
1590 EA=VAL(MID$(EX$,3,2))
1600 SD=VAL(MID$(EX$,5,2))
1610 ME=VAL(RIGHT$(EX$,2))
1620 IF(NR OR EA OR SD OR ME)=0THEN RETURN
1630 PRINT:SN$=" SORTIES : "
1640 IF NR<>0 THEN SN$=" AU NORD"
1650 IF EA<>0 THEN SN$=" A L'EST"
1660 IF SD<>0 THEN SN$=" AU SUD"
1670 IF ME<>0 THEN SN$=" A L'OUEST"
1675 GOSUB 5880:REM FORMATAGE
1680 PRINT:RETURN
1690 :
1700 REM ***** S/P ANALYSE INSTRUCTIONS *****
1705 F=0:REM DRAPEAU A ZERO
1710 IFIS$="FIN" OR IS$="LISTE" THEN VB$=IS$:F=1:RETURN
1720 IF IS$="REGARDE" THEN VB$=IS$:F=1:RETURN
1730 :
1740 REM ** DECOMPOSE INSTRUCTION **
1750 VB$=""IN$=""REM VERBE ET NOMS MIS A ZERO
1770 LS=LEN(IS$)
1780 FOR C=1TO LS
1790 A$=MID$(IS$,C,1)
1800 IF A$=" " THEN VB$=LEFT$(IS$,C-1):NN$=RIGHT$(IS$,LS-C):
F=1:C=LS
1810 NEXT C
1830 IF F=0 THEN PRINT:PRINT"J'AI BESOIN D'AU MOINS DEUX MOTS"
1840 RETURN
1850 :
1900 REM ***** S/P ACTIONS NORMALES *****
1910 VF=0
1920 PRINT
1930 IF VB$="ALLER"ORVB$="PASSER"THENVF=1:GOSUB2000
1940 IF VB$="PRENDRE"ORVB$="SAISIR"THEN VF=1:GOSUB2140
1950 IF VB$="LAISSER"ORVB$="LACHER"THENVF=1:GOSUB2350
1960 IF VB$="LISTER"ORVB$="INVENTORIER"THENVF=1:GOSUB2540
1965 IF VB$="REGARDE" THEN VF=1:MF=1:RETURN
1970 IF VB$="FIN"ORVB$="QUITTER"THENVF=1:GOSUB2610
1980 RETURN
1990 :
2000 REM ***** S/P DEPLACEMENT *****
2010 MF=1:REM REMETTRE DRAPEAU
2015 GOSUB8500:REM RECHERCHE DIRECTION
2020 DR$=LEFT$(NN$,1)
2030 IFDR$<"N"ANDDR$<"E"ANDDR$<"S"ANDDR$<"O"THEN2100
2040 IF DR$="N" AND NR<>0 THEN P=NR:RETURN

```

```

2050 IF DR$="S" AND SD<>0 THEN P=SD:RETURN
2060 IF DR$="E" AND EA<>0 THEN P=EA:RETURN
2070 IF DR$="O" AND ME<>0 THEN P=ME:RETURN
2080 PRINT "IMPOSSIBLE !"
2090 MF=0:RETURN
2100 REM NOM INCONNU ?
2110 PRINT "NN$"? "QU'EST-CE DONC ."
2120 NF=0:RETURN
2130 :
2140 REM ***** S/P PRENDRE Y *****
2145 IV$(4,1)="UN TICKET TROIS-ETATS"
2150 GOSUB5730:REM VALABLE ?
2160 IF F=0 THEN PRINT "IL N'Y EN A PAS !":RETURN
2170 REM ** OBJET POSSEDE ? **
2180 OV=F:GOSUB5830
2190 IFHF=1 THEN SN$=" VOUS L'AVEZ DEJA !":GOSUB5880
:RETURN
2200 :
2210 REM ** OBJET PRESENT **
2220 IF VAL(IV$(F,2))>P THENSN$=" PAS ICI. EN TOUT CAS !":
GOSUB5880:RETURN
2230 :
2240 REM ** AJOUTER OBJET A LISTE **
2250 AF=0:FOR J=1TO4
2260 IFIC$(J)="THENIC$(J)=IV$(F,1):AF=1:J=4
2270 NEXTJ
2280 :
2290 REM ** QUOTA ATTEINT ? **
2300 IF AF=0THENPRINT "VOUS AVEZ DEJA 4 OBJETS":RETURN
2310 :
2320 SN$=" VOILA QUI EST FAIT":GOSUB5880
2330 IV$(F,2)=-1:REM SUPPRIME POSITION
2340 RETURN
2350 :
2360 REM ** S/P LAISSER **
2370 GOSUB5730:REM OBJET VALABLE ?
2380 IF F=0 THEN PRINT "IL N'Y EN A PAS !":RETURN
2390 :
2400 REM ** IS OBJET POSSEDE ? **
2410 OV=F:GOSUB5830
2420 IFHF=0THENPRINT"VOUS N'EN AVEZ PAS !":RETURN
2430 :
2440 REM ** LAISSER OBJET **
2450 SN$=" VOILA QUI EST FAIT":GOSUB5880
2460 IV$(F,2)=STR$(P):REM MISE A JOUR POSITION
2470 :
2480 REM ** SUPPRIMER OBJET DE LA LISTE **
2490 FORJ=1TO4
2500 IF IC$(J)=IV$(F,1)THENIC$(J)=""J=4
2510 NEXTJ
2520 RETURN
2530 :
2540 REM ***** S/P INVENTAIRE *****
2550 PRINT "VOUS AVEZ AVEC VOUS : "
2560 FORI=1TO4
2570 PRINT " "IC$(I)
2580 PRINT I
2590 RETURN
2600 :
2610 REM ***** S/P FIN DE PARTIE *****
2620 PRINT:PRINT"ETES-VOUS SUR ? (O/N)"
2630 GETA:IFA$="O"AND A$="N"THEN2630
2640 IFA$="N"THEN RETURN
2650 END
2660 :
2670 REM ***** S/P ENDROIT SPECIAL *****
2680 SF=0:REM MISE A ZERO DRAPEAU
2690 IF P=37 THEN2700:REM TABLE VECTEUR
2700 IF F 7 THEN 2750:REM BOGUE ALEATOIRE
2710 ON P GOSUB 2850,2960,3450,3830,4180,4550,5150
2720 RETURN
2730 :
2740 REM ** BOGUE ALEATOIRE **
2750 RA=RND(TI)
2760 IF RA=0.05THEN GOSUB5420:REM BOGUE
2770 RETURN
2780 REM ** TABLE VECTEUR **
2790 SF=1
2800 SN$=" VOUS FILEZ A VIVE ALLURE VERS UNE NOUVELLE
DIRECTION":GOSUB5880
2810 FORJ=1TO1000:NEXTJ:REM PAUSE
2820 P=INT(RND(TI)*40+7)
2830 MF=1:RETURN
2840 :
2850 REM ***** S/P SORTIE T.V. *****
2860 SF=1
2870 SN$=" VOUS ETES PRIS AU PIED DANS LA SORTIE T.V."
2880 SN$=SN$+ " VOUS ETES CONDAMNE A VIE A FAIRE PARTIE
DU COLLARO SHOW"
2890 GOSUB 5880:REM FORMATAGE
2900 PRINT
2910 PRINT "SOUS LE NOM DE PENCASSINE"
2920 FORJ=1TO500:NEXTJ
2930 GOTO 2910
2940 END
2950 :
2960 REM ***** S/P PORT UTILISATEUR *****
2970 SF=1
2980 SN$=" LA SORTIE EST PROCHE MAIS UN EMPLOYE
2990 SN$=SN$+ " VOUS BARRE LA ROUTE. IL A POUR FONCTION DE
N'ACCEPTER "
3000 SN$=SN$+ " QUE LES ENTREES, MAIS IL VEUT BIEN PRENDRE"
3010 SN$=SN$+ " DES CARTES DE CREDIT"
3020 GOSUB 5880:REM FORMATAGE

```



```

3030 :
3040 PRINT:INPUT"INSTRUCTIONS":IS#
3050 GOSUB1700:REM ANALYSE INSTRUCTIONS
3060 GOSUB1900:REM ACTIONS NORMALES
3070 IF MF=1 THEN RETURN:REM SORTIE
3080 IF VF=1 THEN3040:REM NOUVELLE INSTRUCTION
3090 IF VB<>" " THENPRINT"JE NE COMPRENDS PAS":GOTO3040
3100 :
3110 REM ** INSTRUCTION : DONNER **
3120 GOSUB5730:REM OBJET VALABLE
3130 IFF=0THENPRINT"IL N'Y EN A PAS !":GOTO3040:REM NEXT
INSTRUCTION
3140 :
3150 REM ** OBJET : CARTE DE CREDIT **
3160 IF F<>5THENPRINT"IL N'ACCEPTÉ QUE LES CARTES DE CREDIT":
GOTO3040
3170 :
3180 REM ** CARTE POSSEDEE **
3190 OV=5:GOSUB5830
3200 IFHF=0THENPRINT"VOUS NE L'AVEZ PAS !":GOTO 3040
3210 :
3220 SN#=" L'EMPLOYE PREND LA CARTE ET DIT : 'CA IRA TOUT
A FAIT'"
3230 GOSUB5880:REM FORMATAGE
3240 SN#=" VOUS PASSEZ LA BARRIERE ET ENTREZ DANS LE PORT
UTILISATEUR."
3250 GOSUB5880:REM FORMATAGE
3260 :
3270 REM ** DIGITAYA ? **
3280 CV=6:GOSUB5830
3290 IF HF=1 THEN 3380:REM GAGNE
3300 :
3310 REM ** PERDU ! **
3320 SN#=" VOUS ETES SORTI DES ENTRAÎLLES"
3330 SN#=" DE LA MACHINE"
3340 SN#=" MAIS SANS DIGITAYA..."
3350 GOSUB5880:REM FORMATAGE
3360 END
3370 :
3380 REM ** GAGNE **
3390 SN#=" FELICITATIONS ! VOUS AVEZ REUSSI VOTRE MISSION"
3400 SN#=" ET ARRACHE DIGITAYA AUX ENTRAÎLLES"
3410 SN#=" DE LA MACHINE"
3420 GOSUB5880:REM FORMATAGE
3430 END
3440 :
3450 REM **** S/P PORT CASSETTE ****
3460 SF=1
3470 SN#=" UNE FORCE MAGNETIQUE VOUS POUSSE IRRESISTIBLEMENT"
3480 SN#=" VERS L'OUBLI MAGNETIQUE
3490 GOSUB5880:REM FORMATAGE
3500 NS=0:REM COMPTE INSTRUCTIONS
3510 REM ** INSTRUCTIONS **
3520 NS=NS+1:IFNS>3THEN3770:REM ASPIRE !
3530 PRINT:INPUT"INSTRUCTIONS":IS#
3540 GOSUB1700:REM ANALYSE INSTRUCTIONS
3550 GOSUB1900:REM ACTIONS NORMALES
3560 IFMF=1THENMF=0:PRINT "VOUS NE POUVEZ BOUGER":GOTO3510
3570 IFVF=1THEN3510:REM INSTRUCTION SUIVANTE
3580 IFVB<>"UTILISER" THENPRINT"JE NE COMPRENDS PAS":GOTO
3510
3590 REM ** INSTRUCTION : UTILISER **
3600 GOSUB5730:REM OBJET VALABLE ?
3610 IFF=0THENPRINT"IL N'Y EN N'A PAS !":GOTO3510
3620 :
3630 REM ** OBJET : ACTIVATEUR TAMPON ? **
3640 IF F=8 THEN3680:REM OK
3650 SN#=" C'EST INUTILE. LA FORCE AUGMENTE"
3660 GOSUB 5880:GOTO3510:REM INSTRUCTION SUIVANTE
3670 :
3680 OV=8:GOSUB5830:REM ACTIVATEUR POSSEDE ?
3690 IFHF=0THENSN#=" VOUS N'EN N'AVEZ PAS !":GOSUB5880:GOTO
3510
3700 :
3710 REM ** SAUVE ! **
3720 SN#=" L'ACTIVATEUR TAMPON CONTRAIRE"
3730 SN#=" LA FORCE QUI DIMINUE"
3740 GOSUB5880:REM FORMATAGE
3750 RETURN
3760 :
3770 REM ** ASPIRE **
3780 SN#=" LA FORCE DEVIENT TROP FORTE ET"
3790 SN#=" VOUS ETES ASPIRE DANS LE PORT CASSETTE, VERS
L'OUBLI MAGNETIQUE"
3800 GOSUB 5880:REM FORMATAGE
3810 END
3820 :
3830 REM **** PORT JOYSTICK ****
3840 SF=1
3850 SN#=" UN USAGER AUX YEUX ROUGIS TIRE SUR VOUS AU LASER"
3860 GOSUB5880:REM FORMATAGE
3870 :
3880 REM ** INSTRUCTIONS **
3890 RD=RD(TI):IF RD<.65THEN 4110:REM TOUCHE
3900 PRINT:INPUT"INSTRUCTIONS":IS#
3910 GOSUB1700:GOSUB1900:REM ANALYSE INSTRUCTION
3920 IFMF=1THENMF=0:PRINT "VOUS NE POUVEZ BOUGER":GOTO3880
3930 IFVF=1THEN3880:REM INSTRUCTION SUIVANTE
3940 IFVB<>"UTILISER" THENPRINT "JE NE COMPRENDS PAS":GOTO
3880
3950 GOSUB5730:REM OBJET VALABLE
3960 IFF=0THENPRINT "IL N'Y EN N'A PAS":GOTO3880:REM
3970 :
3980 REM ** OBJET : BOULIER LASER ? **
3990 IF F=3 THEN4020:REM OK

```

```

4000 SN#=" CELA NE BERT A RIEN !":GOSUB5880:GOTO3880
4010 :
4020 OV=3:GOSUB5830:REM BOULIER LASER POSSEDE ?
4030 IFHF=0THENSN#=" VOUS NE L'AVEZ PAS !":GOSUB5880:
GOTO3880
4040 :
4050 REM ** SAUVE ! **
4060 SN#=" LE BOULIER LASER VOUS PROTEGE. UN CHOC"
4070 SN#=" VOUS PROJETTE HORS DU PORT JOYSTICK, DANS
LA MACHINE"
4080 GOSUB5880:REM FORMATAGE
4090 P=INT(RND(TI)*40+7):MF=1:RETURN
4100 :
4110 REM ** TOUCHE **
4120 SN#=" FRAPPE PAR LE LASER VOUS SENTEZ CONFUSEMENT"
4130 SN#=" QUE VOUS DISPARAISSEZ"
4140 SN#=" DANS LE NEANT"
4150 GOSUB5880:REM FORMATAGE
4160 END
4170 :
4180 REM **** S/P APPAREIL 3 ETATS ****
4190 SF=1
4200 SN#=" UNE PANCARTE INDIQUE ICI E/S MAIS UN CONTROLEUR"
4210 SN#=" ARRIVE ET CRIE 'TICKETS SVP'"
4220 GOSUB5880:REM FORMAT
4230 :
4240 REM ** INSTRUCTIONS **
4250 PRINT:INPUT"INSTRUCTIONS":IS#
4260 GOSUB1700:GOSUB1900:REM ANALYSE
4270 IFMF=1 THEN RETURN
4280 IFVF=1THEN4240:REM INSTRUCTION SUIVANTE
4290 IFVB<>"DONNER" ANDVB<>"OFFRIR" THENPRINT "JE NE
COMPRENDS PAS":GOTO4240
4300 REM ** INSTRUCTION : DONNER **
4310 GOSUB5730:REM OBJET VALABLE ?
4320 IFF=0THENPRINT"IL N'Y EN N'A PAS !":GOTO4240:REM NOUVELLE
INSTRUCTION
4330 :
4340 REM ** OBJET : TICKET ? **
4350 IF F=4 THEN4400:REM OK
4360 SN#=" LE CONTROLEUR BECOULE LA TETE ET DIT '"
4370 SN#=" JE NE PEUX ACCEPTER CA"
4380 GOSUB5880:GOTO4240:REM NOUVELLE INSTRUCTION
4390 :
4400 OV=4:GOSUB5830:REM TICKET POSSEDE ?
4410 IFHF=0THENPRINT"VOUS N'AVEZ PAS LE TICKET":GOTO4240
4420 :
4430 REM ** OK **
4440 SN#=" LE CONTROLEUR PREND VOTRE TICKET ET"
4450 SN#=" VOUS LAISSE PASSER"
4460 GOSUB5880:REM FORMAT
4470 REM ** SUPPRIME TICKET DE LA LISTE **
4480 F=0
4490 FORJ=1TO4
4500 IF IC*(J)=IV*(4,1)THENIC*(J)="" :J=4
4510 NEXT J
4520 IV*(4,2)=STR$(INT(RND(TI)*40+8)):REM NOUVELLE POSITION
TICKET
4530 P=15:MF=1:RETURN
4540 :
4550 REM **** ULA ****
4560 SF=1
4570 RN=INT(RND(TI)*3+1)
4580 IF RN=1 THEN CD#="ET"
4590 IF RN=2 THEN CD#="OU"
4600 IF RN=3 THEN CD#="NON"
4610 SN#=" SUR LE MUR IL Y A 3 BOUTONS"
4620 SN#=" 'ET', 'OU' ET 'NON'. SI VOUS APPUYEZ SUR LE
BON"
4630 SN#=" VOUS ACCEDEREZ A L'ACCUMULATEUR"
4640 GOSUB5880:REM FORMATAGE
4650 :
4660 REM ** INSTRUCTIONS **
4670 PRINT:INPUT"INSTRUCTIONS":IS#
4680 GOSUB1700:GOSUB1900:REM ANALYSE
4690 IF MF=1THEN RETURN:REM SORTIE
4700 IF VF=1THEN 4670:REM NOUVELLE INSTRUCTION
4710 IFVB="UTILISER" THEN4740
4720 PRINT "JE NE COMPRENDS PAS":GOTO4670
4730 :
4740 REM ** COMMANDE VALABLE ? **
4750 IF VB#="APPUYER" THEN 4930
4760 REM ** COMMANDE **
4770 GOSUB5730:REM OBJET VALABLE ?
4780 IFF=0THENPRINT "IL N'Y EN N'A PAS !":GOTO4670:REM
NOUVELLE INSTRUCTION
4790 :
4800 REM ** LIVRE DE CODE ? **
4810 IF F=7 THEN4850:REM OK
4820 SN#=" CA NE SERVIRA A RIEN !":GOSUB5880
4830 GOTO4670:REM INSTRUCTION SUIVANTE
4840 :
4850 OV=7:GOSUB5830:REM LIVRE DE CODE POSSEDE ?
4860 IFHF=1THEN4900:REM OK
4870 SN#=" VOUS NE L'AVEZ PAS !"
4880 GOSUB5880:GOTO4670:REM INSTRUCTION SUIVANTE
4890 :
4900 SN#=" VOUS OUVREZ LE LIVRE ET TROUVEZ LE MOT" + CD# +
" ECRIE DEDANS"
4910 GOSUB5880:GOTO4670:REM INSTRUCTION SUIVANTE
4920 :
4930 REM ** COMMANDE PRESSER **
4940 IFNN#=" ET" OR NN#=" OU" OR NN#=" NON" THEN4970
4950 SN#=" IL N'Y EN N'A PAS !":GOSUB5880:GOTO4670:REM

```

Ian McKinnell

```

INSTRUCTION SUIVANTE
4960 :
4970 REM ** VRAI OU FAUX **
4980 IF NN=C THEN GOSUB5100:RETURN
4990 GOSUB5010:RETURN
5000 :
5010 REM ** S/P FAUX **
5020 SN=" FAUX : UNE TRAPPE S'OUVRE ET"
5030 SN="SN+" VOUS VOUS RETROUVEZ DANS LA MEMOIRE"
5040 GOSUB5800:REM FORMAT
5050 IF RN=1 THEN P=39
5060 IF RN=2 THEN P=35
5070 IF RN=3 THEN P=29
5080 MF=1:RETURN
5090 :
5100 REM ** S/P VRAI **
5110 SN=" LA PORTE DE L'ACCUMULATEUR S'OUVRE"
5120 SN="SN+" ET VOUS LA FRANCHISSEZ":GOSUB5800
5130 P=30:MF=1:RETURN
5140 :
5150 REM **** S/P ENTREE MEMOIRE ****
5160 SF=1
5170 SN=" UN GARDIEN VOUS ACCUEILLE ET VOUS PREVIENT"
5180 SN="SN+" QUE POUR ENTRER IL FAUT UNE ADRESSE":GOSUB5800
5190 REM ** INSTRUCTIONS **
5200 PRINT:INPUT"INSTRUCTIONS":IS
5210 GOSUB1700:GOSUB1900:REM ANALYSE
5220 IF MF=1 THEN RETURN:REM SORTIE
5230 IF VF=1 THEN S200:REM INSTRUCTION SUIVANTE
5240 IF VB<>"DONNER" THEN PRINT "JE NE COMPRENS PAS":
GOTO 5200
5250 :
5260 GOSUB5730:REM OBJET VARIABLE ?
5270 IF F=0 THEN PRINT "IL N'Y EN N'A PAS":GOTO5200:REM
INSTRUCTION SUIVANTE
5280 :
5290 REM ** OBJET : ADRESSE ? **
5300 IF F=1 THEN S330:REM OK
5310 PRINT"IL VEUT VOTRE ADRESSE":GOTO5200
5320 :
5330 DV=1:GOSUB5830:REM ADRESSE TRANSPORTEE ?
5340 IF HF=1 THEN S370
5350 SN=" VOUS NE L'AVEZ PAS":GOSUB5800:GOTO5200
5360 :
5370 REM ** VOUS PASSEZ **
5380 SN=" LE GARDIEN VERIFIE ET"
5390 SN="SN+" VOUS LAISSE PASSER":GOSUB5800
5400 P=40:MF=1:RETURN
5410 :
5420 REM **** BOGUE ALERTOIRE ****
5430 SF=1
5440 SN=" UNE HORRIBLE BOGUE APPARAIT"
5450 SN="SN+" ET RAMPE VERS VOUS":GOSUB5800
5460 :
5470 REM ** INSTRUCTIONS **
5480 PRINT:INPUT"INSTRUCTIONS":IS
5490 GOSUB1700:GOSUB1900:REM ANALYSE
5500 IF MF=1 THEN MF=0:PRINT"VOUS NE POUVEZ BOUGER":GOTO5480
5510 IF VF=1 THEN S480:REM NEXT INSTRUCTION
5520 IF VB=" TUER" OR VB=" COMBATTRE" THEN S550
5530 PRINT"JE NE COMPRENS PAS":GOTO5480
5540 :
5550 REM ** COMBATTRE/TUER **
5560 RR=RND(TI)
5570 IFR<0.5 THEN GOSUB5600
5580 GOSUB5670:RETURN
5590 :
5600 REM **** S/P TUE ****
5610 SN=" VOUS AFFRONTÉZ LA BOGUE"
5620 SN="SN+" ELLE VOUS INONDE DE DONNEES FRAUSSE ET"
5630 SN="SN+" VOTRE CRANE EXPLOSE"
5640 GOSUB5800
5650 END
5660 :
5670 REM **** S/P GAGNE ****
5680 SN=" VOUS COMBATTEZ LA BOGUE ET FINALEMENT"
5690 SN="SN+" VOUS EN VENEZ A BOUT":GOSUB5800
5700 RETURN
5710 :
5720 :
5730 REM **** S/P OBJET VARIABLE ****
5740 NN="NN+" ":LN=LEN(NN):F=0:C=1
5745 FOR K=1 TO LN
5750 IF MID$(NN,K,1)<>" " THEN NEXTK:RETURN
5755 W=MID$(NN,K,C):C=C+1:W=LEN(W)
5760 FOR J=1 TO B
5770 LI=LEN(LI*(J,1)):REM LONGUEUR OBJET
5780 FOR I=1 TO LI
5790 IF MID$(LI*(J,1),I,W)=W THEN F=J:I=LI:J=B:K=LN
5800 NEXT I,J,K
5810 RETURN
5820 :
5830 REM **** OBJET POSSEDE ? ****
5840 HF=0
5850 IF IV*(OV,2)=-1 THEN HF=1
5860 RETURN
5870 :
5880 REM **** S/P FORMATAGE ****
5890 LC=0:REM COMPTEUR LIGNE/CARACTERE
5900 OC=1:REM ANCIEN COMPTE
5910 OW="":REM ANCIEN MOT
5920 LL=40:REM LONGUEUR LIGNE
5930 SN="SN+" BIDON "
5940 PRINT
5950 FOR C=1 TO LEN(SN)
5960 LC=LC+1
5970 IF MID$(SN,C,1)="" THEN GOSUB6020
5980 NEXTC
5990 PRINT
6000 RETURN
6010 :
6020 REM **** S/P FIN DE LIGNE ****
6030 NN=MID$(SN,OC,C-OC+1)
6040 IF LC<LL THEN PRINTNN+";GOTO6050
6050 PRINTNN+LC=LEN(NN)
6060 OC=C+1:OW=NN
6070 RETURN
6080 :
6090 REM **** S/P DATA ****
6100 REM ** LECTURE INVENTAIRE **
6110 DIM IV*(B,2),IC*(4)
6120 FOR C=1 TO B
6130 READ IV*(C,1),IV*(C,2)
6140 NEXT C
6150 :
6160 REM ** LECTURE DATA LIEUX ET SORTIES **
6170 DIM LN*(55),EX*(55)
6180 C1=0:C2=0:REM INITIALISATION CHECKSUM
6190 FOR C=1 TO 54
6200 READ LN*(C),EX*(C)
6210 C1=C1+VAL(LEFT$(EX*(C),4))
6220 C2=C2+VAL(RIGHT$(EX*(C),4))
6230 NEXT C
6240 READ CA:IF CA<>C1 THEN PRINT ERREUR DE DATA:STOP
6250 READ CB:IF CB<>C2 THEN PRINT ERREUR DE DATA:STOP
6260 RETURN
6270 REM **** DATA INVENTAIRE ****
6280 DATA ADRESSE,45,CLE,34,BOULIER LASER,25
6290 DATA TICKET TROIS-ETATS,26,CARTE CREDIT DATA,28
6300 DATA DIGITAYA,30,LIVRE DE CODE,19,ACTIVATEUR,13
6310 :
6320 REM **** DATA LIEUX/SORTIES ****
6330 DATA DANS LA SORTIE T.V.,00000000
6340 DATA DANS LE PORT USAGER,00000100
6350 DATA DANS LE PORT CASSETTE,00100000
6360 DATA DANS LE PORT JOYSTICK,00130000
6370 DATA DANS L'APPAREIL TROIS ETATS,00130000
6380 DATA DANS L'ULA,00310016
6390 DATA A LA PORTE DE LA MEMOIRE,00490000
6400 DATA SUR L'AUTOROUTE,09000001
6410 DATA SUR L'AUTOROUTE,10000002
6420 DATA SUR L'AUTOROUTE,11000000
6430 DATA SUR L'AUTOROUTE,12001003
6440 DATA SUR L'AUTOROUTE,13531100
6450 DATA SUR L'AUTOROUTE,14001204
6460 DATA SUR L'AUTOROUTE,15001300
6470 DATA SUR L'AUTOROUTE UNE PANCARTE INDIQUE 'S U D',
00001400
6480 DATA DANS LE REGISTRE DATA,00061700
6490 DATA SUR UNE AUTOROUTE A 8 VOIES,16001805
6500 DATA SUR UNE AUTOROUTE A 8 VOIES,17001900
6510 DATA SUR UNE AUTOROUTE A 8 VOIES,18002000
6520 DATA SUR UNE AUTOROUTE A 8 VOIES,19292100
6530 DATA SUR UNE AUTOROUTE A 8 VOIES,20292200
6540 DATA SUR UNE AUTOROUTE A 8 VOIES,21272300
6550 DATA SUR UNE AUTOROUTE A 8 VOIES,22262400
6560 DATA SUR UNE AUTOROUTE A 8 VOIES,23250000
6570 DATA DANS LA MATRICE CARACTERES,26360024
6580 DATA EN MEMOIRE HAUTE,27352523
6590 DATA EN MILIEU DE MEMOIRE,28342622
6600 DATA EN MILIEU DE MEMOIRE,29332721
6610 DATA EN MEMOIRE BASSE,00542820
6620 DATA DANS L'ACCUMULATEUR,00000000
6630 DATA DANS UN LONG CORRIDOR,00420006
6640 DATA DANS LE REGISTRE D'INDICES,31000000
6650 DATA EN MEMOIRE BASSE,54403428
6660 DATA EN MILIEU DE MEMOIRE,33393527
6670 DATA EN MEMOIRE HAUTE,34383626
6680 DATA DANS LA MATRICE CARACTERES,35370025
6690 DATA DANS UNE TABLE DE VECTEURS,00000000
6700 DATA EN MEMOIRE HAUTE AU DESSUS D'UNE AUTOROUTE,39003735
6710 DATA EN MILIEU DE MEMOIRE,40003834
6720 DATA EN MEMOIRE - A L'EST UNE PORTE,41003933
6730 DATA EN MEMOIRE BASSE,00004034
6740 DATA DANS UN CORRIDOR,00430031
6750 DATA DANS UN CORRIDOR,00440042
6760 DATA DANS UN CORRIDOR,00004543
6770 DATA DANS LE REGISTRE ADRESSES,00004600
6780 DATA SUR UNE AUTOROUTE A 8 VOIES,45004700
6790 DATA SUR UNE AUTOROUTE A 8 VOIES,46004800
6800 DATA SUR UNE AUTOROUTE A 8 VOIES,47004900
6810 DATA SUR UNE AUTOROUTE A 8 VOIES,48005007
6820 DATA DANS UN VECTEUR VERS LA MEMOIRE,49005100
6830 DATA DANS UN VECTEUR VERS LA MEMOIRE,50005200
6840 DATA DANS UN VECTEUR VERS LA MEMOIRE,51000000
6850 DATA EN MEMOIRE BASSE,00290012
6860 DATA TOTAL DE CONTROLE,00413329
6870 REM ** DATA TOTAL DE CONTROLE **
6880 DATA 100169,103973
8599 :
8600 REM **** S/P DIRECTIONS ****
8610 NN="NN+" ":LN=LEN(NN):C=1
8620 FOR I=1 TO LN
8630 IF MID$(NN,I,1)<>" " THEN NEXT I:RETURN
8640 W=MID$(NN,C,I):C=C+1
8650 IF W=" NORD" OR W=" EST" THEN NN=W+I:LN
8660 IF W=" SUD" OR W=" OUEST" THEN NN=W+1:LN
8670 NEXT I
8680 RETURN

```