

abc

N° 76

COURS
D'INFORMATIQUE
PRATIQUE
ET FAMILIALE

INFORMATIQUE



Les variables de PASCAL

Commencez à remonter le temps

Le Yamaha CX5M

Madeleine Hodé, une pionnière

EDITIONS
ATLAS



Salle de classe

Nous allons voir comment les ordinateurs peuvent aider les très jeunes enfants à maîtriser l'écriture et le dessin. C'est un problème qui prête à controverses.

Un jour un petit chien
 alla dans la forêt et
 vit une abeille et du
 miel. Il aimait beaucoup
 le miel. Il dit : j'aime
 beaucoup le miel et je
 vais le prendre et il
 s'avança et mit son
 nez dans l'ouverture
 de la ruche. Mais
 une grosse abeille la
 piqua. Il partit en
 courant et il n'est
 plus jamais revenu dans
 le bois. fin.

Un jour un petit chien alla dans la forêt
 et il vit une abeille et du miel. Il aimait
 beaucoup le miel. Il dit : j'aime beaucoup le
 miel et je vais le prendre et il s'avança
 et mit son nez dans l'ouverture de la ruche.
 Mais une grosse abeille l'a piqué. Il partit
 en courant et il n'est plus jamais revenu
 dans le bois. Fin.

Les innovations techniques se heurtent souvent aux pesanteurs politiques et sociales : les gens répugnent à faire des expériences, à défendre des idées qui n'ont pas encore fait leurs preuves. Même un outil aussi familier aujourd'hui que le cahier a représenté, lors de son apparition, un progrès révolutionnaire.

Le papier ne servait alors qu'aux transactions commerciales et aux corps constitués : gouvernement, église, universités. Et n'oublions pas qu'il était coûteux. Dans de telles conditions, comment les enfants feraient-ils pour s'en servir correctement sans le gaspiller ?

Le cahier se révéla pourtant un instrument des plus utiles apparaissant, notamment, comme une sorte de « mémoire » du travail. N'a-t-il pas permis, et aujourd'hui encore, de progresser sur différents exercices à la fois sans effacer ce qui avait été réalisé auparavant.

Il en va de même pour les ordinateurs, qui se voient opposer des arguments tout à fait semblables. Il paraît utopique cependant d'offrir un appareil à chaque enfant, et, d'ailleurs, ce serait ruineux. L'introduction de nouvelles techniques

au sein de l'Éducation nationale (et dans tous les pays) s'est toujours heurtée à des problèmes de prix.

Les enseignants sont, pour leur part, parfois très réticents à utiliser une technologie déjà mal vue, sans doute à cause de la « programmation linéaire » dont nous avons déjà parlé. Une étude précise de la façon de mettre en œuvre l'ordinateur dans une salle de classe permettra peut-être de lever bien des objections. Non seulement il permet d'obtenir de meilleurs résultats — et se révèle donc économique malgré son prix d'achat encore trop élevé — mais, de surcroît, il contribue grandement à favoriser la créativité dans le processus éducatif.

Savoir rédiger

Voyons d'abord comment les micros peuvent jouer un rôle dans la rédaction d'un texte. Écrire est un processus assez complexe qui comprend différentes étapes : il faut inventer une histoire, la mettre en forme, la rédiger, la corriger, lui faire subir des améliorations avant d'obtenir un texte

Jouer avec les mots

Lorsque les enfants ont à exprimer leur créativité par le biais de l'expression écrite, ils sont souvent handicapés par leur maîtrise plus ou moins bonne de la calligraphie : on a vite fait de perdre le fil de sa pensée quand il faut former nettement chaque lettre. C'est d'ailleurs pourquoi leur écriture paraît si dispersée. Un traitement de texte ne résoudra pas ce problème — il faut du temps pour trouver la bonne touche — mais taper à la machine nécessite bien moins de capacités psychomotrices. Comme par ailleurs il est possible d'éditer et de corriger le texte avant qu'il ne soit imprimé, la mise en forme définitive exige moins de temps, et les idées et les mots sont mieux liés les uns aux autres.



L'informatique à la rescousse

L'ordinateur met en œuvre des capacités d'écriture et de dessin très semblables à celles qu'on emploie quand on travaille « à la main ». Mais il a l'avantage de les développer plus aisément et d'en faire apprendre de nouvelles. Nous donnons ci-contre quelques-uns de ses domaines d'application.

Langue et écriture

- Capacités psychomotrices fines (mouvement de la main et des doigts).
- Reconnaissance des lettres.
- Formation des lettres (par emploi d'une imprimante et d'un jeu de caractères complet).
- Vérification et correction.
- Orthographe (et les multiples problèmes qui y sont liés).
- Coordination des idées (remise en ordre logique des événements).
- Mémoire visuelle (apprendre où sont les touches pour ne pas avoir à regarder le clavier).

Dessin

- Coordination de l'œil et de la main.
- Capacités psychomotrices fines.
- Reconnaissance des formes.
- Sens des volumes.
- Relations de cause à effet.
- Aisance d'expression (génération rapide des idées).
- Souplesse d'esprit (« pensée latérale » : repérer la nouveauté derrière les idées connues).
- Imagination.



présentable. Tout cela s'apparente d'assez près, en fait, à la peinture et à la musique.

Malheureusement, l'exercice de rédaction, tel qu'il se déroule dans beaucoup d'écoles, est beaucoup trop structuré pour en tenir compte. Les enfants se voient souvent imposer un temps limité, une présentation aussi soignée que possible sans rature ni « pâté ». C'est que l'aspect proprement créatif (corriger, réécrire, modifier) est sacrifié à une norme, implicite ou non, du « bien écrire ». L'attitude des enseignants favorise cette situation puisque d'une part, ils poussent les enfants à écrire le plus vite possible et, d'autre part, à ce qu'il y ait le moins de ratures possible pour pouvoir corriger le plus facilement et sans trop de perte de temps.

L'emploi d'un ordinateur et d'un traitement de texte simplifierait certainement bien les choses : les enfants pourraient apprendre sans trop de difficultés à éditer et corriger leur texte. On pourrait ensuite obtenir une copie imprimée ; le texte serait également stocké sur disquette, afin que l'enseignant puisse le revoir à loisir, chez lui ou à l'école ou pour faire circuler les différents écrits parmi les élèves pour établir des comparaisons. Il existe d'ailleurs des logiciels qui pointent et corrigent toutes les fautes d'orthographe. Le résultat en serait une copie présentable, et son auteur ne manquerait pas d'en être très fier.

Barrière du clavier

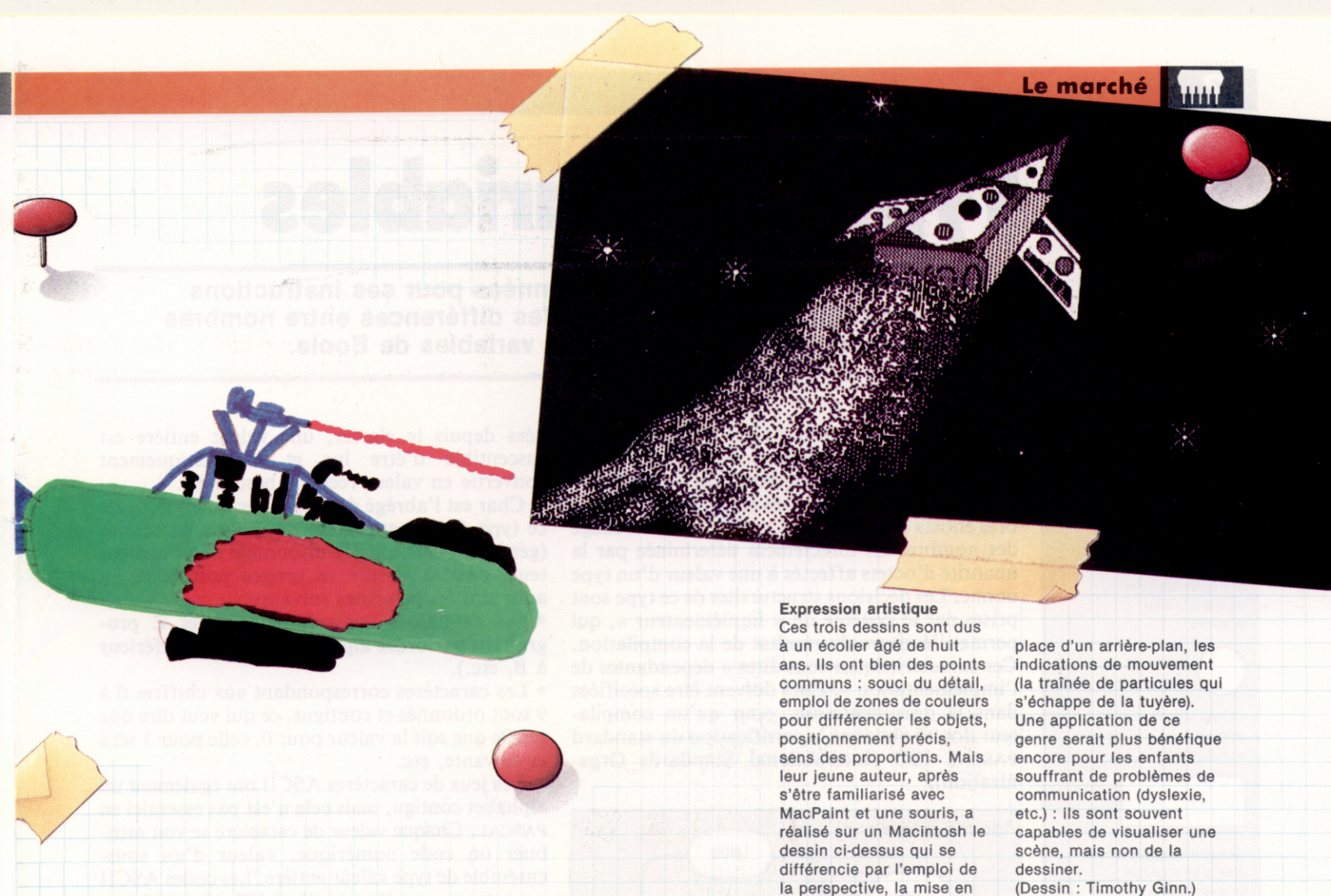
On a fait remarquer, toutefois, qu'avec ce système, l'enfant avait moins conscience de ses fautes, et qu'il ressentait moins vivement la néces-

sité de les corriger. Il est de fait que tous ceux qui font usage d'un traitement de texte préfèrent souvent corriger leurs fautes sur le papier parce qu'elles sont plus visibles, et non à l'écran. En outre, il est possible d'écrire des commentaires dans la marge.

Mais c'est un type de programme qui demande à être encore amélioré ; le problème disparaîtra sans doute quand il sera devenu aussi simple d'emploi que le crayon et le papier.

On parle beaucoup de « la barrière du clavier », lorsque les enfants apprennent à se servir d'un ordinateur ; mais on ne parle guère de « la barrière du crayon », lorsqu'ils doivent écrire ! Il leur faut savoir comment tenir le crayon, mémoriser la forme des lettres, puis savoir les reproduire correctement, à la bonne taille, en ligne droite... Les lettres doivent être dessinées de façon particulière, en tenant compte d'un certain ordonnancement dans la mise en forme. C'est une tâche écrasante qui demande plusieurs années ; et, il faut reconnaître que bien des gens ne sauront jamais s'en acquitter correctement. Il faut recommencer tous les jours, tenir compte des différents jeux de caractères...

En comparaison, apprendre à taper à la machine est bien plus simple. Il faut identifier les lettres sur le clavier, savoir quelle est la position de chacune, et établir des rapports entre elles et tel ou tel doigt. Bien entendu, il sera toujours utile — et nécessaire — de savoir écrire « à la main » ; mais c'est parfois un inconvénient. La créativité d'un enfant est souvent limitée par sa maîtrise imparfaite de la calligraphie. Entre cinq et sept ans, on attend de l'enfant qu'il améliore



Expression artistique

Ces trois dessins sont dus à un écolier âgé de huit ans. Ils ont bien des points communs : souci du détail, emploi de zones de couleurs pour différencier les objets, positionnement précis, sens des proportions. Mais leur jeune auteur, après s'être familiarisé avec MacPaint et une souris, a réalisé sur un Macintosh le dessin ci-dessus qui se différencie par l'emploi de la perspective, la mise en

place d'un arrière-plan, les indications de mouvement (la traînée de particules qui s'échappe de la tuyère). Une application de ce genre serait plus bénéfique encore pour les enfants souffrant de problèmes de communication (dyslexie, etc.) : ils sont souvent capables de visualiser une scène, mais non de la dessiner.

(Dessin : Timothy Ginn.)

progressivement, en qualité et en quantité, le nombre de lignes qu'il écrira correctement. Le clavier permettrait très certainement d'obtenir des résultats bien supérieurs, qui surprendraient enseignants et parents. Des expériences de ce type ont déjà eu lieu tant en Europe qu'en Amérique du Nord et les résultats — même si l'on manque encore de recul — sont surprenants.

Il ne fait, par ailleurs, aucun doute qu'avec le temps, le traitement de texte deviendra une nécessité sociale, et non plus une commodité commerciale, comme c'est le cas actuellement.

Fais-moi un dessin

Un autre aspect de la micro-informatique est la facilité avec laquelle un ordinateur permet d'obtenir des créations de graphismes très élaborés. On s'est longtemps heurté à la barrière du prix de revient ; il y fallait des puces coûteuses et plus complexes que pour faire du traitement de texte. Mais les ordinateurs destinés aux écoles et à l'éducation en général deviennent chaque jour plus puissants et moins chers. La tablette graphique qui permet des fonctions « artistiques » variées existe depuis plusieurs années ; on s'en sert en conjonction avec un moniteur. Plus récemment, certains ordinateurs comme le Macintosh d'Apple se sont vus équiper d'une « souris » qui en reprend les principes de base.

La souris contrôle une série d'« icônes » qui permet la maîtrise, immédiate et précise, de possibilités les plus diverses : dessin libre, travail sur

des formes, extension ou réduction de celles-ci, insertion de trames, manipulation des images, incrustation, etc. La liste n'est pas fermée.

Dans le même ordre d'idées, il est très facile pour l'écolier d'intégrer ou de rassembler simultanément texte et image. Une fois achevée, cette dernière sera « découpée » et « collée » là où on en a besoin. Un logiciel graphique comme MacPaint ne peut évidemment se substituer aux activités d'expression graphique telles qu'elles existent déjà dans les écoles ; mais il leur offre un champ d'activité nouveau, extrêmement riche et prometteur.

Les enfants sont en fait très désireux d'apprendre — il est vrai, pas toujours dans le cadre de la classe... Mais ils font preuve d'une curiosité incessante, et d'une volonté d'expression qui se satisfait à la moindre occasion : un crayon, un bout de craie et un mur suffisent... L'ordinateur leur permet de retrouver cette même dimension.

Les très jeunes enfants, placés devant un ordinateur, apprennent en appuyant simplement sur les touches, pour voir des formes bizarres apparaître à l'écran... Ils savent assez vite manipuler l'appareil, parce qu'il représente pour eux quelque chose de familier. Le système éducatif ne fait pas toujours un bon usage de l'informatique ; il la regarde encore comme une fantaisie coûteuse. Mais il est certain que dans un avenir très proche, il sera inévitablement amené à y recourir de façon généralisée. Ce n'est qu'une question de temps. Quand les écoles ne regardent plus les micros uniquement comme des jouets coûteux.

Types de variables

Le PASCAL utilise quatre types de données pour ses instructions de déclaration de variables. Voyons les différences entre nombres entiers, valeurs réelles, caractères et variables de Boole.

Formes symboliques

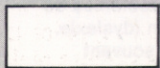
Ces trois formes géométriques constituent les symboles couramment utilisés dans les diagrammes syntaxiques PASCAL :



• Le symbole arrondi ou en losange est mis pour les mots réservés ou pour les caractères qui se passent d'explication (une lettre ou un chiffre par exemple).



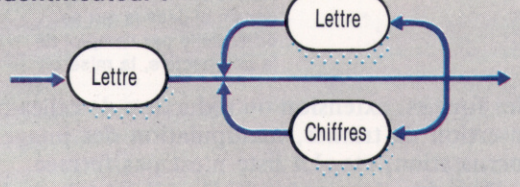
• Un cercle représente un opérateur PASCAL (+, -, *, /, etc.).



• Le symbole rectangulaire représente un mot ou une phrase ayant son propre diagramme syntaxique.

Le PASCAL nous fournit quatre types prédéfinis de données, d'identificateurs Entier, Réel, Char et Booléen. Les nombres sont classés selon leur aptitude à être fractionnés (nombres réels), ou non (nombres entiers ou naturels). La capacité de stockage des nombres est directement déterminée par la quantité d'octets affectés à une valeur d'un type donné. Les décisions structurelles de ce type sont prises par le système dit « implémenteur », qui permet l'écriture du résultat de la compilation. Ces caractéristiques sont dites « dépendantes de l'implémentation ». Elles doivent être spécifiées dans la documentation, pour qu'un compilateur donné obtienne la certification du standard PASCAL ISO (International Standards Organization).

Identificateur :



La fourchette des valeurs possibles pour un entier sera probablement comprise, sur votre compilateur, entre $-32\,768$ et $32\,767$, ou entre $-2\,147\,483\,648$ et $+2\,147\,483\,647$, selon que les nombres sont représentés sur deux ou quatre octets. Le PASCAL donne un nom à la plus grande valeur entière possible, représentée par l'identificateur prédéfini de constante `MaxInt`. Vous pouvez donc facilement trouver cette valeur par l'instruction :

```
WriteLn ('MaxInt :', MaxInt)
```

Les nombres réels devront également être compris dans une certaine fourchette, généralement $1,7E + 38$, avec une précision de six ou sept chiffres dans le cas le plus défavorable. Cette manière d'écrire les nombres réels correspond au mode par défaut en PASCAL. Vous pouvez néanmoins les écrire de façon traditionnelle, par exemple 123.456, lorsque cela reste lisible.

Vous devez pourtant savoir qu'un nombre réel comporte *toujours* un point pour séparer sa partie entière de sa partie fractionnaire. Ces deux dernières doivent être *systématiquement* présentes. 0.1 et $1.0E - 1$ sont valides, mais, 1 ou $1E - 1$ sont incorrects. Heureusement, ces règles ne s'appliquent qu'aux nombres devant être reconnus comme réels dans le texte du programme. Ainsi, lorsque vous saisissez des don-

nées depuis le clavier, une valeur entière est susceptible d'être lue et automatiquement convertie en valeur réelle si besoin est.

Char est l'abrégié de Character, une valeur de ce type devant appartenir au jeu de caractères (généralement ASCII) disponible sur l'ordinateur. PASCAL assure sa propre portabilité en adoptant les principes suivants :

- Les caractères compris entre A et Z progressent par ordre alphabétique. A est inférieur à B, etc.).
- Les caractères correspondant aux chiffres 0 à 9 sont ordonnés et contigus, ce qui veut dire que quelle que soit la valeur pour 0, celle pour 1 sera la suivante, etc.

Les jeux de caractères ASCII ont également un alphabet contigu, mais cela n'est pas essentiel en PASCAL. Chaque valeur de caractère se voit attribuer un code numérique, valeur d'un sous-ensemble de type valeur entière. Les codes ASCII appartiennent à l'intervalle 0-127, de nombreuses machines étendant cet éventail jusqu'à 255 pour les codes supplémentaires destinés aux caractères graphiques. Il est assez facile de coder tout jeu de caractère ordonné, selon l'échelle des valeurs ordinales utilisée de manière interne par l'ordinateur. PASCAL fournit la fonction prédéfinie `Ord`; elle produit le code pour la valeur entière de son argument. Ainsi `Ord (A)` est l'équivalent de 65 dans la table ASCII. Une autre fonction, `chr`, assure le codage en sens inverse — `chr (65)` donne le caractère A (vous remarquez que le signe \$ n'est pas utilisé avec `chr`).

L'éventail des valeurs de caractères et des entiers est défini pour chaque implémentation; il figure selon un ordre croissant de constantes connues. C'est pourquoi ces dernières sont appelées type « ordinal » ou « scalaire ». Quelle que soit la valeur, nous pouvons toujours savoir la précédente ou la suivante, lorsqu'elle existe. Ces valeurs voisines sont obtenues par les fonctions scalaires suivantes :

```
pred (élément) (antérieur)
succ (élément) (postérieur)
```

Ainsi `succ (3)` correspondra toujours à la valeur de caractère 4, tandis que `pred (Z)` ne donnera Y qu'avec certains jeux de caractères, tel le jeu ASCII. `Pred(MaxInt)` sera soit 32 766, soit 2 147 483 646. La fonction `chr` ne peut être utilisée qu'avec un argument-code de caractère. Toutes les autres fonctions scalaires peuvent être utilisées avec tout type scalaire, même si `ord`, employé avec une valeur entière, donne simple-

ment la valeur de son argument. Les variables booléennes sont les plus simples de tous les types scalaires, ne représentant que deux valeurs dans l'échelle, vrai ou faux (et dans cet ordre). Les fonctions scalaires peuvent recevoir toutes valeurs booléennes du fait qu'elles sont de type scalaire simple. La valeur ordinaire de faux est 0, et ord(vrai) est égal à 1. Les autres fonctions scalaires, pred et succ, ne sont pas très utiles ici. Votre compilateur PASCAL réagira très violemment lorsque vous lui demanderez WriteLn(pred(faux)) : vous ne pouvez raisonnablement lui demander d'évaluer une valeur qui n'existe pas.

Le passage suivant de définition de constante dans un programme fait figurer tous les types simples ordinaux tels qu'ils peuvent apparaître dans un texte-source PASCAL.

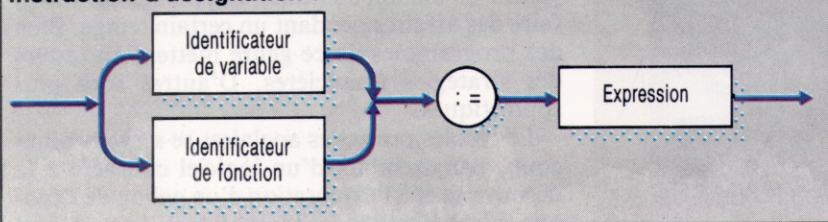
```
CONST
  VAT           = 0,15;
  colonnes     = 40;
  espace       = " ";
  recherche d'erreur = faux;
```

Le symbole (=) signifie *toujours* égal en PASCAL. Il est utilisé pour faire correspondre, à un identificateur, la valeur qui lui est assignée. A la déclaration des variables dans la partie déclaration VAR, le signe (:) sert à séparer l'identificateur de variable nouvellement déclaré, de son type. Par exemple :

```
VAR
  genre       : réel;
  nombre     : entier;
  symbole     : char;
  effectué   : booléen;
```

Lorsque nous désirons affecter des valeurs à ces variables, nous utilisons l'opérateur composite d'assignation (: =). Cela permet de faire nettement la distinction entre les trois modes d'opérations. Les définitions CONST font correspondre des valeurs fixes; les déclarations VAR se contentent de réserver de la place mémoire; enfin, l'assi-

Instruction d'assignation :



gnation donne une valeur (qui peut être temporaire) à l'identificateur.

Lorsque deux instructions ou davantage doivent être exécutées dans le cadre d'un même processus, nous pouvons les inclure dans une parenthèse syntaxique entre les mots BEGIN et END. Ils forment alors une instruction composée. Rappelez-vous simplement que chaque élément doit être séparé des autres par le signe (:). Nous avons déjà rencontré les instructions composées, puisque le corps de toute instruction exécutable prend cette forme. Nous adopterons la convention qui consiste à écrire en lettres majuscules les mots réservés afin de les distinguer des identificateurs.

Vous remarquez les sauts de lignes :

```
PROGRAM Cercle (entrées, sorties);
CONST
  pi = 3.1415926536;
  message = « tapez le rayon : »;
VAR
  rayon,
  surface : réel;
BEGIN
  WriteLn;
  write (message);
  read (rayon);
  surface := pi * rayon * rayon;
  WriteLn;
  WriteLn (« la surface d'un cercle »,
    « de rayon », rayon : 8 : 3);
  WriteLn (c'est : « , surface : 10 : 3)
END
```

Cet exemple illustre deux aspects syntaxiques intéressants. D'abord, la partie VAR déclare deux identificateurs de même type, réel. Il n'est pas nécessaire de faire deux déclarations, puisque les listes d'éléments sont simplement séparées par des virgules, ce qui est universel en PASCAL. C'est pourquoi la même syntaxe s'applique à plusieurs arguments d'une procédure dans les instructions WriteLn. La deuxième caractéristique nouvelle est le formatage des sorties utilisé pour suppléer la notation scientifique par défaut des valeurs réelles. Nous pouvons en outre donner deux entiers séparés par une virgule, pour attribuer une certaine longueur de zone à la représentation du nombre entier et de sa partie fractionnaire.

Trois affectations décimales seront données dans chaque cas pour le rayon et la surface. La surface étant un nombre plus important (et donc plus long), nous lui attribuons une longueur totale de dix positions au lieu de huit seulement pour le rayon. Ces valeurs entières doivent être supérieures à zéro, accepter une virgule, comporter au moins un chiffre, et permettre l'écriture de la virgule avant la partie fractionnaire. Les valeurs illégales susciteront un message d'erreur à l'exécution, ou (au mieux) restitueront le format de la notation scientifique. WriteLn (X : 6 : 2) ne ménagerait pas suffisamment de place pour les nombres supérieurs à 99,99 par exemple.

Atout encore plus précieux, PASCAL arrondit automatiquement le dernier chiffre, ce qui nous donne la meilleure précision possible pour tout champ numérique souhaité. En outre, il est possible d'utiliser toute variable ou expression, et non pas seulement des constantes. Cela confère une très grande souplesse, notamment en ce qui concerne la tabulation. Pour tous les autres types de données, une seule valeur entière est nécessaire, ou du moins suffisante, pour préciser la longueur de la zone affectée. Une longueur d'une unité entraînera l'écriture des entiers dans la zone minimale et sans espace. Il nous faudra donc nous souvenir des entiers si les résultats doivent être disposés en tables. Par exemple :

```
WriteLn (« Total : » :20, poids : 1, « tonnes. »)
```

PASCAL refusera normalement de donner un résultat ambigu.



Maîtres du temps

Les jeux de simulation recréent fidèlement des événements réels. Nous allons mettre au point un programme qui nous fera voyager et commercer dans l'Amérique du XV^e siècle.

Chacun sait que l'ordinateur est par excellence le support rêvé du jeu d'arcade qui permet de tester nos réflexes ou d'assouvir nos pires instincts de destruction... sans que personne n'en souffre physiquement. Mais il se prête aussi parfaitement à la simulation, autre forme de défi que l'on peut se lancer à soi-même : il recrée toutes les apparences d'un univers réel, dans lequel il faut apprendre à se déplacer.

Le programme peut aussi comporter des éléments de réflexion stratégique. Chaque joueur doit accomplir certaines tâches, et reçoit les informations nécessaires. Exemple classique, celui du commerce : chaque participant se voit accorder une certaine somme d'argent et doit se livrer à diverses transactions pour tirer un profit maximal de sa mise de départ. Les périls et les coups du sort évoquent un peu ceux des jeux d'aventures, mais, ici, il est possible de prévoir un peu plus à l'avance, et de faire usage de ses propres ressources — surtout intellectuelles, contrairement à ce qui vous est demandé dans les jeux d'arcades — suivant les circonstances.

Ces types de jeux demandent donc à ceux qui s'y intéressent d'être capables de réfléchir et de calculer, plutôt que de réagir instantanément, presque par instinct. Nous sommes dans un

monde qui se veut réaliste, bien sûr ; il interviendra donc des facteurs imprévus, impossibles à planifier. Le bon joueur est celui qui saura à la fois tirer le meilleur parti des événements et faire face aux difficultés inattendues grâce à la rapidité de ses réflexes et à son intelligence.

L'avantage de ces jeux de simulation est qu'on peut y jouer à plusieurs et former des équipes qui affronteront l'ordinateur. C'est d'ailleurs bien pourquoi on l'utilise fréquemment dans les écoles : c'est un bon moyen de se familiariser avec certains types de conduite sociale, ceux qui régissent le monde du travail par exemple. De surcroît, les jeux de simulation sont tout à fait passionnants et amusants...

Des stratégies

Le plus significatif, pourtant, est qu'ils ont toujours une dimension morale : l'appât du gain n'est pas tout ; il faudra parfois faire des choix d'ordre moral : le profit ne passe pas avant la vie des employés, même d'un strict point de vue économique. Heureusement ! Si de telles décisions doivent être prises collectivement, on imagine sans mal qu'elles risquent fort de donner lieu à des débats passionnés...

Bien des situations se prêtent à une simulation sur ordinateur : il faut avant tout définir une tâche spécifique visant à accomplir un but précis que parfois le joueur choisira lui-même : gagner un million de francs, ou, plus simplement, faire des affaires pendant un certain temps. Bien des programmes de ce genre mettent en œuvre des stratégies financières. D'autres sont plus sophistiqués.

Les écoles primaires anglaises se servent beaucoup, par exemple, d'un logiciel consacré à la découverte et à l'exploration d'un galion de l'époque élisabéthaine, le *Mary Rose*, dont il faut retrouver l'épave. Les joueurs ont droit également à des documents qui leur fournissent des indices supplémentaires.

Nous nous efforcerons ici de mettre sur pied un programme de simulation, dans lequel les participants auront à équiper un navire, puis à le lancer en direction du Nouveau Monde à peine découvert (sachez que nous sommes au XV^e siècle). Nous l'écrirons sous forme de modules interdépendants, en prenant bien soin de permettre de profondes modifications si le besoin s'en fait sentir ; ce sera en fait une sorte de squelettes qui pourront être aisément adaptés. Il vous sera facile, par exemple, de créer un voyage dans

Des univers réalistes

On voit ici quelques-uns des grands types de jeux de simulation. *Space Shuttle* vous place aux commandes d'une navette spatiale.

Air Traffic Control est une simulation en temps réel : vous devez gérer le départ et l'arrivée de nombreux avions sur un aéroport. *The Great Nordic War* est un wargame qui a pour cadre la guerre de Trente Ans ; vous y incarnez le roi de Suède, Charles XII.

(Cl. Liz Heaney.)





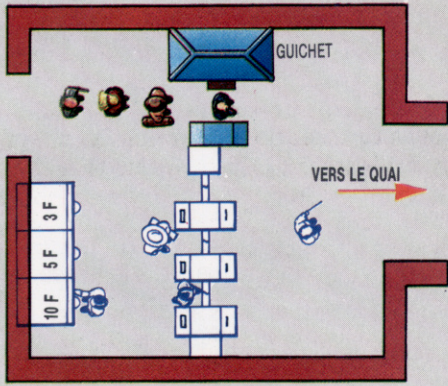
Avoir le ticket

Les jeux de simulation ont pour origine lointaine les exigences de la gestion. Les techniques mises en œuvre se prêtent, en effet, particulièrement bien à l'étude du trafic routier par exemple. Elles utilisent des méthodes de calculs mathématiques et statistiques qui permettent de définir avec précision la fréquence de distribution des variables considérées.

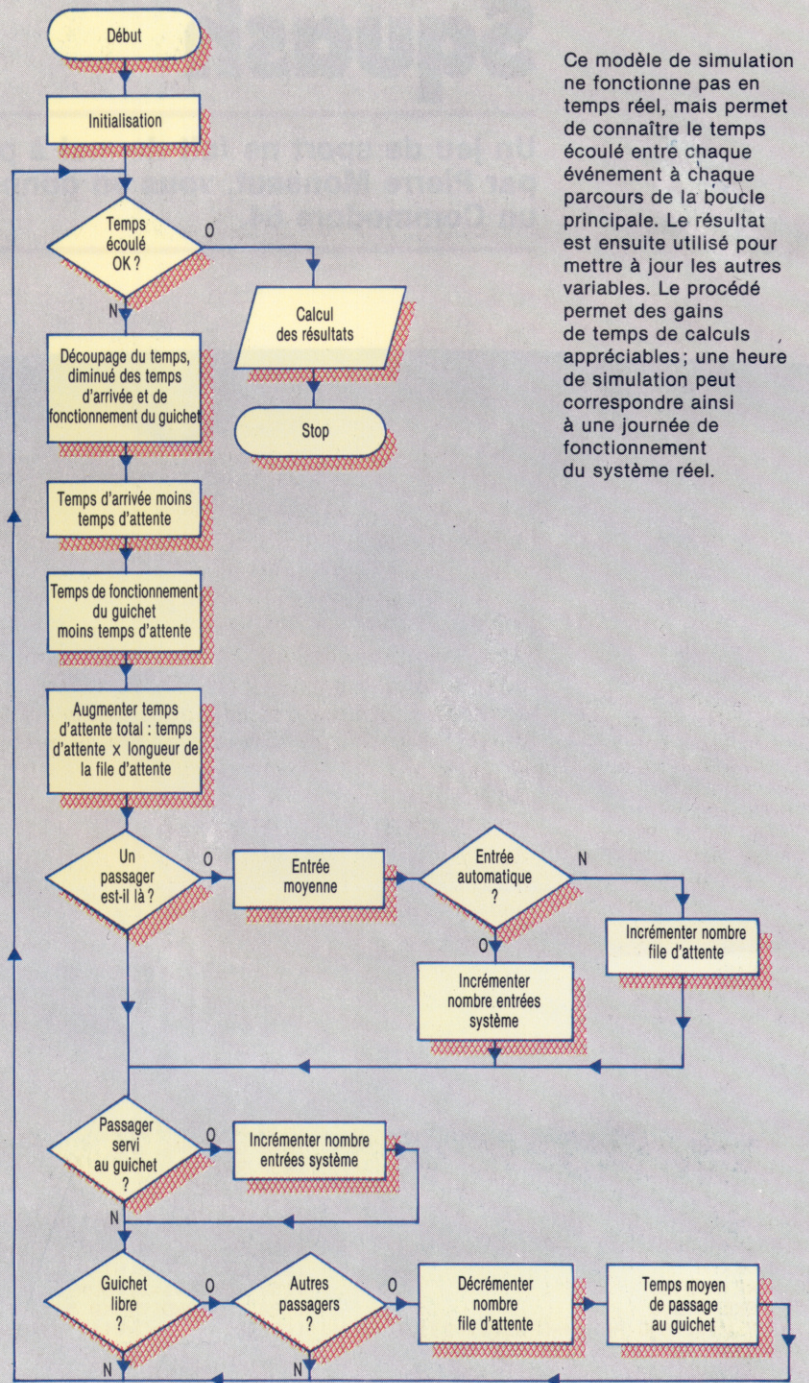
Un exemple nous permettra d'avoir une idée plus exacte de ces procédés. Supposons que, dans une station de métro, on ne puisse accéder au quai qu'en achetant un ticket au guichet. Imaginons maintenant qu'on veuille mettre en place un système automatique composé de trois poinçonneuses automatiques, qui accepteront des tickets achetés à l'avance. L'organigramme ci-contre montre comment simuler ce système. Il faudra tenir compte de trois grands facteurs : le débit d'arrivée des passagers, le temps de mise en œuvre de chaque guichet, et la façon (manuelle et automatique) de présenter le ticket.

Les deux premiers sont définis en fonction d'observations statistiques et de méthodes mathématiques; le troisième peut être modifié à chaque fois, afin d'étudier les conséquences liées aux deux types d'entrée et à leur rapport entre elles. Notre programme permettra en tout cas de calculer le temps d'attente moyen de chaque passager qui se présentera.

FILE D'ATTENTE MANUELLE



PROJET DE FILE D'ATTENTE AUTOMATISÉE



Ce modèle de simulation ne fonctionne pas en temps réel, mais permet de connaître le temps écoulé entre chaque événement à chaque parcours de la boucle principale. Le résultat est ensuite utilisé pour mettre à jour les autres variables. Le procédé permet des gains de temps de calculs appréciables; une heure de simulation peut correspondre ainsi à une journée de fonctionnement du système réel.

d'autres galaxies, en direction d'une planète lointaine, si vous êtes attiré par des espaces plus neufs.

Ce jeu, destiné à un ou plusieurs joueurs, vise à commercer dans l'espoir d'atteindre un profit maximal. Il ne doit pas, cependant, passer avant la vie et la santé de l'équipage. Nous sommes des gentlemen, n'est-ce pas? Il comprend trois parties : préparatifs, voyage et commerce. Lors de la première phase, vous aurez à rassembler tout ce dont vous aurez besoin et à préparer minutieusement votre équipée.

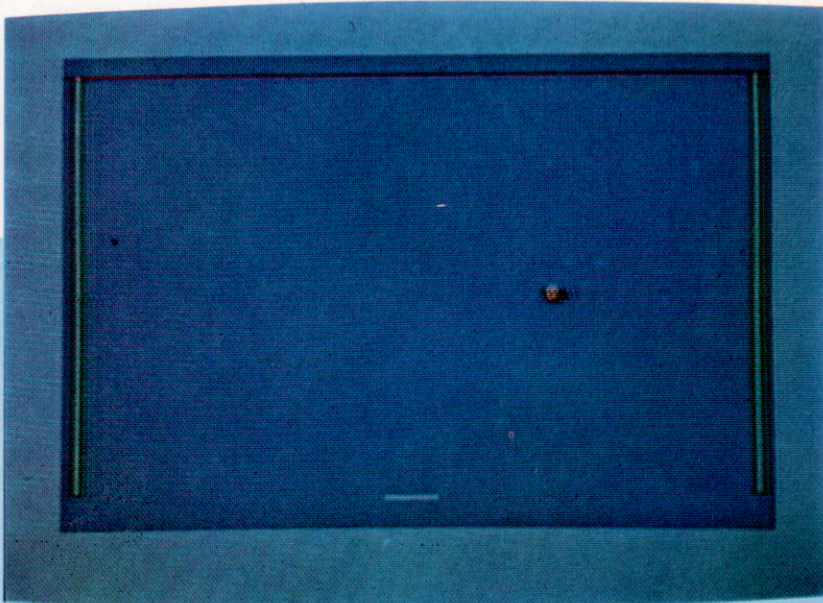
La deuxième sera émaillée d'événements aléatoires que vous surmonterez d'autant mieux que vous aurez su prendre vos précautions au départ.

Enfin, troisième phase, vous aurez à établir des relations amicales avec les indigènes et à commercer avec eux de façon aussi profitable que possible.

Pour tenir compte de l'opinion des programmeurs intéressés par ce jeu, l'organisation du projet, c'est-à-dire sa construction, sera composée d'une série de modules indépendants. Nous donnerons des variantes BASIC pour le Spectrum et le Commodore 64. Il s'agira avant tout de créer des sous-programmes gérant les décisions prises, créant des situations inattendues ou gardant trace des succès obtenus par le joueur. Nous traiterons du premier module lors du prochain article de cette série.

Squash

Un jeu de sport ne fait de mal à personne. Ce jeu, écrit par Pierre Monsaut, vous en donne la possibilité si vous avez un Commodore 64.



Grâce à votre ordinateur, vous pouvez jouer au squash assis dans un fauteuil. La raquette se déplace à l'aide des touches de contrôle du curseur (à droite de la barre d'espacement). Vous disposez de dix balles que vous devez garder en jeu le plus longtemps possible. Chaque balle renvoyée rapporte un point.

```

5 REM *****
10 REM * SQUASH *
15 REM *****
20 GOTO 1000
50 U=U+DU
60 H=H+DH
70 POKE M+B,BN
80 B=U*X+H
90 POKE M+B,B0
100 POKE N+B,C
110 IF U=U1 AND ABS(H-(POS(X)+3))>1 THEN
2000
120 IF U=U1 OR U=U0 THEN DU=-DU
130 IF H>=H1 OR H<=H0 THEN DH=-DH
140 IF U=U1 THEN S=S+1
150 GET X$
160 IF X$=G$ AND POS(X)<>0 THEN PRINT GG
$: IF U=U1 THEN H=H-1
170 IF X$=D$ AND POS(X)<>34 THEN PRINT D
D$: IF U=U1 THEN H=H+1
180 PRINT R$;
190 PRINT T$;
200 GOTO 50
1000 X=40
1010 M=1024
1020 N=55296
1030 D$=CHR$(29)
1040 G$=CHR$(17)
1050 R0=0
1060 R1=33
1070 BN=32
1080 B=0:C=8:B0=81
1110 H=INT(RND(TI)*(X-4)+2)
1120 U=22:H0=2:H1=X-2
1150 U0=1:U1=22:DU=-1
1180 DH=(RND(TI)<0.5)*2+1
1190 R$="" +CHR$(196)+CHR$(196)+CHR$(19
6)+""
1200 R=17
1220 B=U*X+H
1230 IF NB=0 THEN GOSUB 1500
1240 GOTO 50
1500 PRINT CHR$(147);
1510 FOR I=0 TO 40
1520 POKE M+I,100
1530 POKE M+I,2
1540 NEXT I
1550 FOR I=1 TO 22
1560 POKE M+I*40,103
1570 POKE M+I*40,5
1580 POKE M+I*40+39,101
1590 POKE M+I*40+39,5
1600 NEXT I
1610 DD$=CHR$(29)+CHR$(29)
1620 GG$=CHR$(157)+CHR$(157)
1630 T$=""
1640 FOR I=1 TO 7
1650 T$=T$+CHR$(157)
1660 NEXT I
1670 PRINT CHR$(19)
1680 FOR I=1 TO 22
1690 PRINT G$;
1700 NEXT I
1710 FOR I=1 TO 16
1720 PRINT " ";
1730 NEXT I
1740 S=0
2000 NB=NB+1
2010 POKE M+B,BN
2130 FOR I=1 TO 500
2140 NEXT I
2150 IF NB=11 THEN 3000
2160 GOTO 1000
3000 IF S>RR THEN RR=S
3010 NB=0
3020 PRINT CHR$(147);
3030 FOR I=1 TO 7
3040 PRINT
3050 NEXT I
3060 PRINT TAB(16)"SCORE :";S
3090 PRINT:PRINT:PRINT:PRINT
3110 PRINT TAB(16)"RECORD :";RR
3130 FOR I=1 TO 4
3140 PRINT
3150 GET X$
3160 NEXT I
3170 PRINT TAB(16)"UNE AUTRE ?"
3180 GET X$
3190 IF X$="" THEN 3180
3200 IF X$<>"N" THEN 20
3210 END

```




Gammes de bits

Le micro CX5M aux normes MSX, qui vient de faire son apparition, est le premier ordinateur domestique dédié à la musique. Son fabricant : Yamaha, évidemment !

Bien qu'il réponde aux normes MSX, ce qui lui permet d'être utilisé pour des applications de jeu ou de traitement de texte, l'ordinateur Yamaha CX5M vise presque exclusivement le marché des musiciens. Un clavier de piano externe YK-10 ou YK-01 peut être branché sur le côté de la machine où se trouve également une paire de ports MIDI (Musical Instrument Digital Interface). Tout synthétiseur ou dispositif musical muni d'une interface MIDI peut donc être commandé par l'ordinateur. L'ordinateur ressemble beaucoup aux autres machines de la gamme MSX. Il possède quarante-huit touches, type machine à écrire, entourées de dix touches de commande, exécutant des fonctions MSX standard comme le retour arrière destructif. Une touche Graph affiche une série de caractères graphiques, et une touche Code qui, lorsqu'elle est appuyée en même temps que l'une des touches du clavier, imprime divers caractères étrangers. Au-dessus du clavier apparaissent cinq touches de fonction capables d'appeler dix fonctions programmables. Les quatre touches de commande de curseur se trouvent dans le coin inférieur droit du clavier ; au-dessus, cinq autres touches correspondent à d'autres fonctions MSX standard comme INSert, DElete et STOP. Un port de cartouche est situé en haut du clavier.

Architecture de la machine

Sur le côté droit apparaît une paire de ports pour manche à balai de type Atari. A l'arrière de l'ordinateur, on aperçoit un connecteur plat parallèle, l'interface d'imprimante Centronics, un port de cassette, un jack sonore pour une sortie mono externe, un jack de moniteur vidéo composite, un jack de téléviseur HF et l'entrée de l'alimentation. Les ports MIDI et l'interface du clavier, qui donnent à la machine son potentiel musical, se trouvent dans une boîte unique. Celle-ci se glisse dans un connecteur plat situé sous la machine. Cela s'explique par le fait que Yamaha pense produire une série d'interfaces avec différentes configurations. Lorsque de nouveaux dispositifs seront disponibles, les utilisateurs pourront changer simplement d'interface en retirant une vis et en glissant la nouvelle interface dans le connecteur plat.

Le clavier YK-01 à trois octaves et demie — vendu avec l'ordinateur et apparaissant sur notre photographie — offre un toucher agréablement professionnel, bien que certains musiciens puissent trouver les touches un peu trop petites. A



l'aide d'un logiciel intégré, le clavier peut être partagé, c'est-à-dire qu'un thème programmé est exécuté sur la partie inférieure du clavier tandis qu'un autre thème complètement différent se joue sur le reste des touches. Cela signifie, par exemple, que vous pouvez jouer des « cordes » sur les touches supérieures tout en produisant un accompagnement de « cuivres » sur les touches inférieures. Le clavier peut aussi être utilisé en mode monophonique ou polyphonique (ou les deux) ; il est ainsi possible de jouer huit notes simultanément.

Lors de la mise sous tension, l'écran affiche l'écran bleu MSX standard. Le programme musical appelé par la commande CALL MUSIC affiche alors un menu renfermant cinq blocs de diverses options disponibles. L'utilisateur parcourt la liste à l'aide de la touche RETURN, et les paramètres de chaque option sont modifiés en appuyant sur les touches de commande du curseur. Les blocs portant les étiquettes POLY et MONO permettent à l'utilisateur de sélectionner quels thèmes pré-programmés parmi les quarante-six disponibles seront utilisés et quel sera le mode retenu. Ces thèmes vont des instruments musicaux conventionnels — comme l'orgue, la guitare ou les instruments à percussion comme le vibraphone et la cloche de vache — à une variété de sons de tous les jours (sirène d'ambulance, averse, etc.). En raison de la nature de certains

Boîtes musicales

On peut s'attendre à voir le Yamaha CX5M se vendre mieux dans les boutiques d'instruments musicaux que chez les revendeurs d'ordinateurs. Pour mettre l'accent sur son potentiel musical, l'ordinateur est vendu avec un clavier musical YK-01 ou YK-10. Le clavier musical se branche dans une petite interface musicale qui est vissée sous l'ordinateur.
(Cl. Chris Stevens.)



Clavier

Le clavier possède toutes les touches de fonction qui sont normalement associées aux normes MSX, dont cinq de fonction programmables et un groupe de commande du curseur.

Alimentation

En raison des restrictions d'espace imposées par l'interface musicale, l'alimentation est installée sur le côté, sur la carte principale.

Connecteur plat

La boîte amovible renfermant l'interface musicale se glisse dans ce connecteur.

ROM BASIC

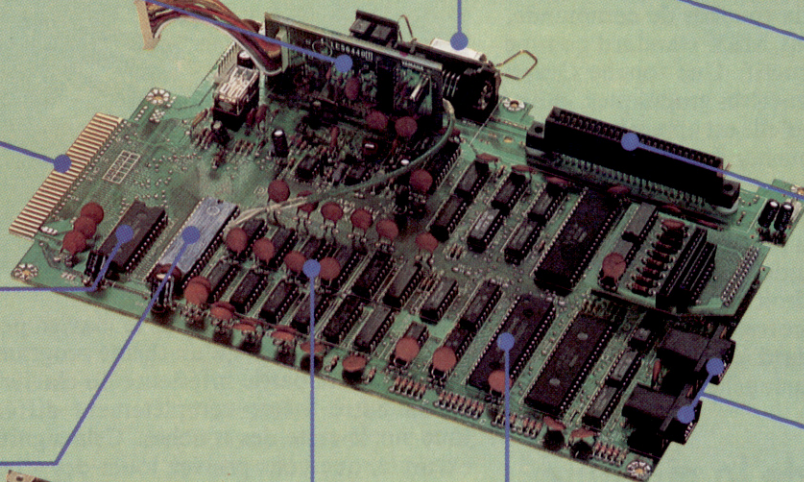
Le basic MSX 32 K est contenu dans cette puce.

Puce vidéo

L'ordinateur renferme la puce vidéo TMP8255 qui fournit les 32 lutins et les 16 couleurs dont dispose le programmeur.

Modulateur HF

Il permet de brancher l'ordinateur à un téléviseur standard.



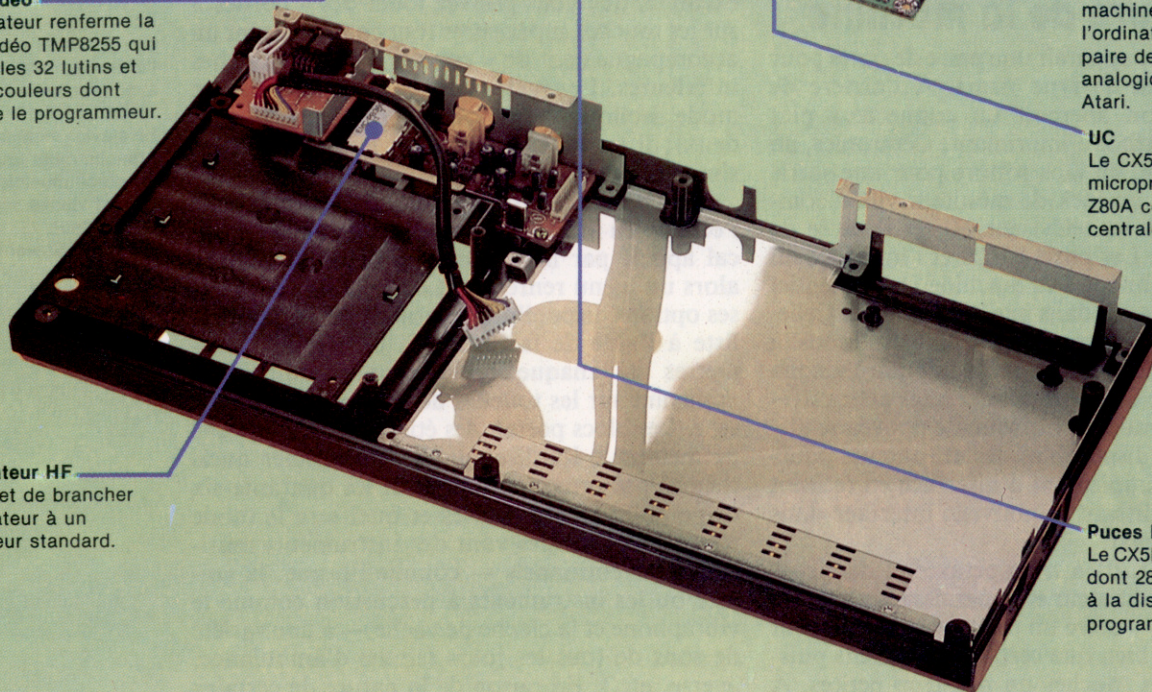
Interface d'imprimante
Le CX5M est muni d'un port d'imprimante compatible Centronics.

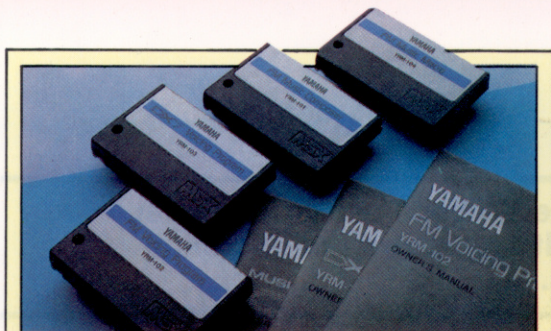
Port de cartouche
Ce port de cartouche parallèle permet l'exécution de programmes de musique additionnels ainsi que l'utilisation des autres cartouches MSX.

Ports de manche à balai
Comme les autres machines MSX, l'ordinateur possède une paire de ports analogiques de type Atari.

UC
Le CX5M utilise le microprocesseur à 8 bits Z80A comme unité centrale de traitement.

Puces RAM
Le CX5M a 48 K de RAM dont 28 K sont mis à la disposition du programmeur BASIC.





Voici certaines des cartouches proposées pour le CX5M. Chacun de ces programmes permet à l'ordinateur d'effectuer une fonction différente — de la programmation du synthétiseur DX-7 à la composition de programmes musicaux. Chaque cartouche est accompagnée d'un manuel. Ces programmes sont relativement coûteux.

thèmes, on ne peut les utiliser en maintien prolongé. Le fait de modifier les thèmes des modes POLY et MONO permet aux utilisateurs de définir des sons contrastés pour les utiliser avec un clavier partagé — c'est-à-dire mettre certaines touches en mode monophonique et d'autres en mode polyphonique.

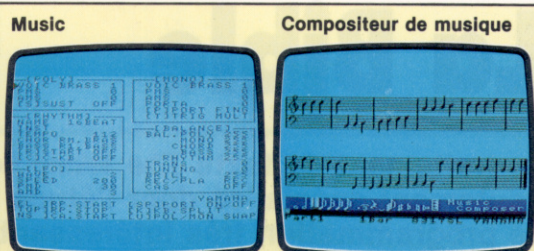
Mêmes défauts

Les sons obtenus sont remarquables, même lorsqu'ils sont reproduits par un haut-parleur de téléviseur. Toutefois, les sons « trains », par exemple, ressemblent aux sons d'un orgue électrique ordinaire. C'est un défaut commun chez les fabricants de synthétiseurs.

Un autre bloc, RHYTHM, fournit un accompagnement rythmique à une mélodie. Six rythmes différents peuvent être construits à l'aide d'accords, de basses et de percussions. Les sons de basses et d'accords peuvent être définis par l'utilisateur. Le tempo du rythme peut varier et une option permet au joueur de varier le son de la section rythmique à partir du clavier. En modifiant la forme de l'onde sonore avec le bloc LFO (Low Frequency Oscillator), les sons sont modulés pour produire différentes sonorités. Le bloc BALANCE permet un réglage indépendant du volume de chaque élément rythmique, dans le mode POLY et dans le mode MONO. Par conséquent, il est possible de baisser le volume de l'accompagnement rythmique de fond et d'élever le volume du mode POLY pour faire ressortir la mélodie.

Lorsque le joueur est satisfait par les sons produits, les données peuvent être sauvegardées sur cassette et rechargées ultérieurement. Pas besoin d'écrire les divers paramètres sélectionnés et de les redéfinir à chaque exécution.

Yamaha a également mis sur le marché une série de cartouches destinées au CX5M. Cette gamme inclut le programme FM Voicing, qui est une version étendue du logiciel intégré. Le programme permet au joueur de manipuler entièrement la forme d'une voix en ajustant les fréquences et les algorithmes qui la construisent. Un second programme de définition des voix présenté sur cartouche permet à l'ordinateur de pro-



On voit ici deux des applications disponibles pour le CX5M. Le programme Music est intégré et est appelé au moyen de la commande CALL MUSIC. En déplaçant le curseur avec les touches de fonction et RETURN, les paramètres peuvent être modifiés en appuyant sur les touches de curseur. Le programme compositeur de musique permet au joueur d'écrire une partition musicale à l'écran qui peut être éditée puis envoyée sur imprimante.

grammer le fameux synthétiseur Yamaha DX7. Le programme FM Music Macro permet de programmer l'ordinateur avec une série de commandes BASIC additionnelles; FM Music Composer affiche une portée vierge, permet d'entrer une partition conventionnelle puis de l'envoyer vers une imprimante externe.

On peut constater que le logiciel du CX5M n'exploite pas entièrement le potentiel de la machine. Les programmes de modulation, en particulier, semblent un peu limités — il est parfois difficile de discerner un changement effectif dans le son d'une note. C'est surprenant, car Yamaha a fondé le système d'exploitation du CX5M sur celui utilisé par le synthétiseur DX7; l'immense gamme de sons de celui-ci ridiculise un peu le CX5M.

De plus, bien que les méthodes choisies par Yamaha pour modifier les paramètres des blocs semblent correctes, elles sont parfois curieuses comme celle qui permet d'utiliser les touches de curseur pour modifier les paramètres ou de déplacer le curseur avec la touche RETURN. Yamaha affirme cependant qu'il est sur le point de proposer des logiciels améliorés.

Trop cher?

Le CX5M est évidemment destiné aux personnes qui désirent posséder un ordinateur domestique et que la musique électronique intéresse. Pour ces personnes, le CX5M représente un achat plutôt séduisant. Cet ordinateur est non seulement un véritable système MIDI à clavier, mais il offre un potentiel largement supérieur à un système comme le Commodore 64, par exemple, dont le prix est voisin. On peut, cependant, penser que le CX5M est destiné à devenir un ordinateur spécialisé.

Il est difficile de justifier son existence à une personne qui n'est pas intéressée par la musique électronique, d'autant qu'il est maintenant possible d'acheter un ordinateur MSX similaire pour moitié moins cher. Le développement futur du logiciel et des interfaces de la machine sera donc déterminant dans la tentative de Yamaha de s'imposer sur le marché de l'ordinateur domestique dédié à la musique.

YAMAHA CX5M

PRIX

Avec le clavier YK-01 : ★★★★★; avec le clavier YK-10 : ★★★★★ (les deux avec écran).

DIMENSIONS

413 x 216 x 64 mm.

UC

Z80A, fonctionnant à 3,58 MHz.

MÉMOIRE

RAM de 48 K dont 28 K sont à la disposition des programmes BASIC.

ÉCRAN

Écran texte de 24 x 40; écran graphique de 256 x 192 points, avec 16 couleurs et jusqu'à 32 lutins.

INTERFACES

Imprimante Centronics, téléviseur, moniteur composite, sortie audio, sortie stéréo, deux ports manche à balai, port cassette, port cartouche ROM, bus d'extension, interface de clavier, ports EIS MIDI.

LANGAGES DISPONIBLES

BASIC, PASCAL, Assembleur.

CLAVIER

67 touches « machine à écrire » avec touches de commande du curseur, et 5 touches de fonction programmables. Le clavier piano permet une polyphonie à 8 notes sur 3 octaves et demie.

DOCUMENTATION

Le manuel est très décousu. Tout en donnant quelques détails concernant l'utilisation du logiciel musical, il ne propose aucun exercice de formation. Le BASIC MSX est à peine mentionné.

FORCES

Comme instrument musical, le CX5M n'a aucun compétiteur à ce niveau de prix.

FAIBLESSES

Le logiciel doit être grandement amélioré si le fabricant désire s'assurer un succès à long terme. Le manuel est très limité et la machine trop coûteuse pour qui ne s'intéresse pas à la musique électronique.



La Bible

Le traitement de texte *The Word* (« Le Verbe ») est un progiciel hautement spécialisé dont l'étude fait apparaître des techniques de programmation très intéressantes.

« The Word Processor »
Pour machines MS-DOS
Auteurs :
Bible Research Systems,
Austin-Texas
Format :
Disque

Certaines occupations et recherches universitaires supposent de laborieuses recherches de références dans des textes très longs. Les étudiants sur les textes sacrés, par exemple, doivent avoir recours à des « concordances », ou index spécifiques à la Bible, pour mener à bien leurs recherches. Ces dernières portent en effet sur les trente-neuf livres de l'Ancien Testament et sur les vingt-sept du Nouveau Testament; c'est-à-dire sur un corpus de quelque 750 000 mots.

Les limitations de mémoire interne et de place sur les disques magnétiques sont souvent un obstacle sur les ordinateurs familiaux. C'est dans ce contexte qu'un programme écrit spécialement pour une telle recherche peut être appréciable. Le progiciel appelé, sans complexe, « Le Verbe » permet non seulement la recherche de références déterminées, mais encore le temps nécessaire aux recherches.

L'énormité du sujet couvert par « The Word » apparaît déjà en ouvrant son étui. En effet, si le manuel est une mince documentation de trente-six pages, le programme complet comprend les fichiers — texte de la Bible, c'est-à-dire sept disquettes double face, dont la totalité doit être recopiée.

Traitement du Verbe

L'installation de « The Word » est délicate. Et comme l'on peut penser que les étudiants de la Bible ne sont pas forcément des informaticiens, il faut bien reconnaître que le programme est un peu compliqué dans son exploitation. Lorsque nous l'avons utilisé, nous nous sommes aperçus qu'il ne répondait pas au MS-DOS, mais au système d'exploitation PC-DOS. L'interpréteur BASIC et le code système de la machine doivent être copiés depuis le disque système sur le disque programme. L'étudiant choisit alors entre l'exécution du programme principal (TWP), par RUN, s'il utilise un IBM-PC (une autre version du même programme, en l'espèce ATWP pour les machines compatibles), ou une version spéciale pour les ordinateurs avec seulement 64 K (TWP64). L'utilisateur averti pensera qu'une fois le système, le BASIC et les programmes appropriés chargés, le logiciel se charge lui-même, mais il n'existe pas de programme AUTOEXEC.BAT à cette fin.

Après le chargement LOAD et l'exécution RUN, la page titre est un avertissement. Non seulement l'utilisateur se voit exposé à la réglementation américaine (ce logiciel est protégé par la législa-

tion américaine des copyright : « Toute reproduction ou utilisation non autorisée de copies illécitales sera l'objet de poursuites pénales, conformément à... »), mais encore, il est fait cette mention très opportune à la Bible :

EXODE 20:15 « Tu ne voleras pas. »

Cette injonction d'ordre moral est suivie d'un menu général :

<F1> - AFFICHAGE D'INSTRUCTION D'AIDE
<F2> - ATTRIBUTION DES OPTIONS DE COMMANDE
<F3> - MODE AFFICHAGE INHIBÉ, ACTIVEZ-LE
<F4> - INTERVALLE
<F5> - FIN DU TRAVAIL
<F6> - CRÉATION D'UN INDEX
<F7> - AFFICHAGE OU MODIFICATION DE L'INDEX
<F8> - FUSION DE TABLES D'INDEX
<F9> - SUPPRESSION D'UN INDEX
<F10> - AFFICHAGE DU TEXTE DE L'ÉCRITURE

<F2> affiche un sous-menu de neuf options de commande, dont la détermination de la largeur de l'écran (40 ou 80 caractères), la tabulation pour les marges, l'espacement, le nombre de lignes par page, la taille maximale pour un index, l'affichage d'un texte à la fois ou du texte entier avec les versets, et les noms de toutes les tables d'indices disponibles sur un disque.

L'option <F4> concilie le grand volume de la Bible et le nombre important de disques. Une désignation sur trois lettres pour chaque livre de la Bible, de « Gen » à « Rev », ainsi que les références de chapitre et de verset, donnent le début et la fin du paramètre de recherche. Lorsque l'utilisateur s'intéresse à un livre particulier, les références au chapitre et au verset peuvent être omises, le programme supposant que le début du premier chapitre et la fin du dernier constituent la plage désirée.

Une fois le disque approprié mis en place, il est possible de parcourir un chapitre déterminé. <F10> permet d'afficher le premier verset du chapitre; les touches fléchées assurent le développement avant et arrière. L'option de commande <F8> fait passer le mode d'affichage d'un verset à la fois, à la totalité de l'écran, permettant de lire selon le contexte.

Les critères de recherche sont donnés par l'option <F2>, option de balayage du texte SCAN. Il est possible de localiser toutes les occurrences d'une chaîne de caractères, même lorsqu'elle fait partie d'autres mots (« cherche » sera localisé dans « recherche »). Mais vous pouvez également restreindre les recherches à la seule chaîne de

caractères demandée. Les recherches sont également possibles en ne demandant qu'une lettre de commencement ou de fin de chaîne. Les références demandées sont données une par une, soit dans leur contexte, soit individuellement, selon ce qui a été demandé.

Les indices sont créés par l'option <F6>, soit automatiquement, soit manuellement. Ils peuvent être édités (par exemple pour supprimer les références non souhaitées) et fusionnés, pourvu que la limite des 1 020 éléments dans la table ne soit pas dépassée. Cette limite peut être augmentée jusqu'à 53 040 par l'option <F2> DONNER LES OPTIONS DE COMMANDE du menu principal. Lorsqu'une table comporte plus de 1 020 références, elle est sauvegardée (à l'aide de la commande OPTIONS DE CONTRÔLE du sous-menu), par tranches, une pause étant marquée après la 1 020^e. Il est également possible de créer un index des sujets couverts par chaque verset. Cela doit se faire de toute évidence à l'initiative de l'utilisateur qui définit les sujets et leur localisation.

Décalage

En dehors de la difficulté de traiter une telle quantité de textes, le principal problème est celui de l'ancienneté de la langue. Le traitement de texte biblique *The Word* reproduit une ancienne traduction de la Bible; le sens de nombreux termes est donc aujourd'hui inusité, voire incompréhensible. Il y aurait quelque 300 mots « obscurs » pour le lecteur contemporain. Il

conviendrait en outre d'ajouter à ces ombres les erreurs de traduction.

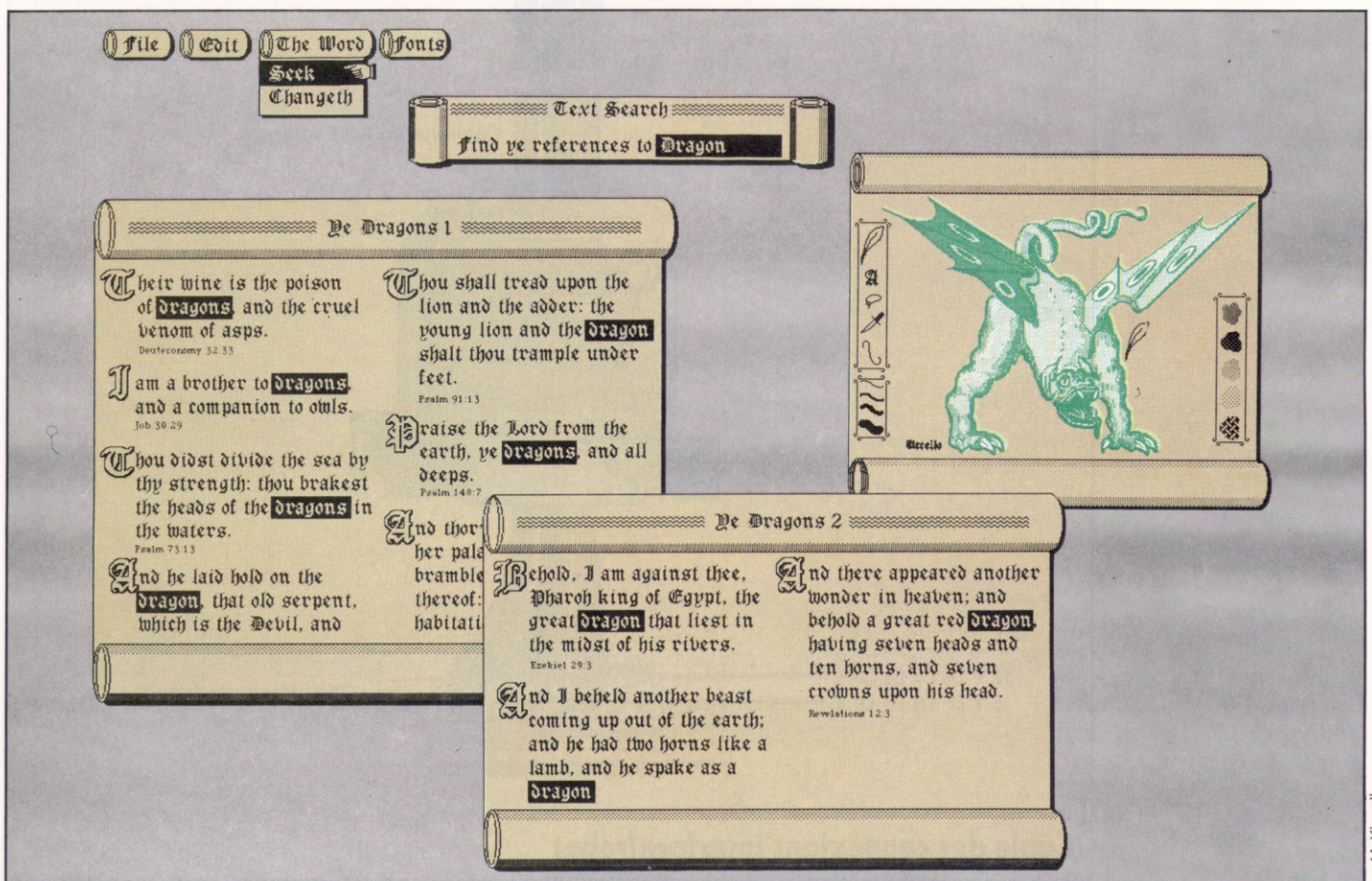
Un autre programme vient heureusement effacer ce décalage linguistique, « Traducteur du grec ». Ce programme peut être associé et éviter des erreurs de sens, tout du moins pour ce qui est de l'Ancien Testament. Il vous permet de vous référer au texte grec.

Les mots grecs sont affichés selon une transcription phonétique. Le mot ainsi trouvé est défini et étudié. Les indices sont interchangeables entre le logiciel de traitement de texte biblique et le logiciel de lexicologie grecque. Il est également possible de créer des index de termes grecs. Le prix des deux logiciels est le même.

Jusqu'ici, aucun système de lexicologie comparée n'a été produit pour ce qui concerne l'Ancien Testament, même si un système informatique central a été programmé à Jérusalem pour contenir le Talmud et d'autres ouvrages rabbiniques. Il s'agit d'ailleurs probablement de l'idéal en matière d'études bibliques : l'informatique de consultation d'un centre serveur, base de données d'accès public, et non pas système individuel de restitution de données. Les méthodes utilisées par le progiciel *The Word* n'en sont pas moins intéressantes et pourraient être mises en œuvre avec profit dans d'autres domaines d'application. L'étude de grands textes littéraires ou d'intérêt général peut donc parfaitement s'appuyer sur un traitement informatisé de la critique stylistique et sémantique. Pourquoi pas Shakespeare, Molière ou Karl Marx ?

Saint Georges et le dragon

Interrogé sur le mot « dragon », le progiciel *The Word* repère diverses occurrences dans sa base de données. Il fait office de concordance écrite (référence biblique), et simplifie la recherche de thèmes associés, symboles correspondants, ou personnages impliqués, selon la traduction anglaise de King James, 1611. (Nous avons utilisé la version en langue anglaise du logiciel; nous avons donc gardé la version anglaise de la Bible dont un extrait apparaît ci-dessous.)





La liaison

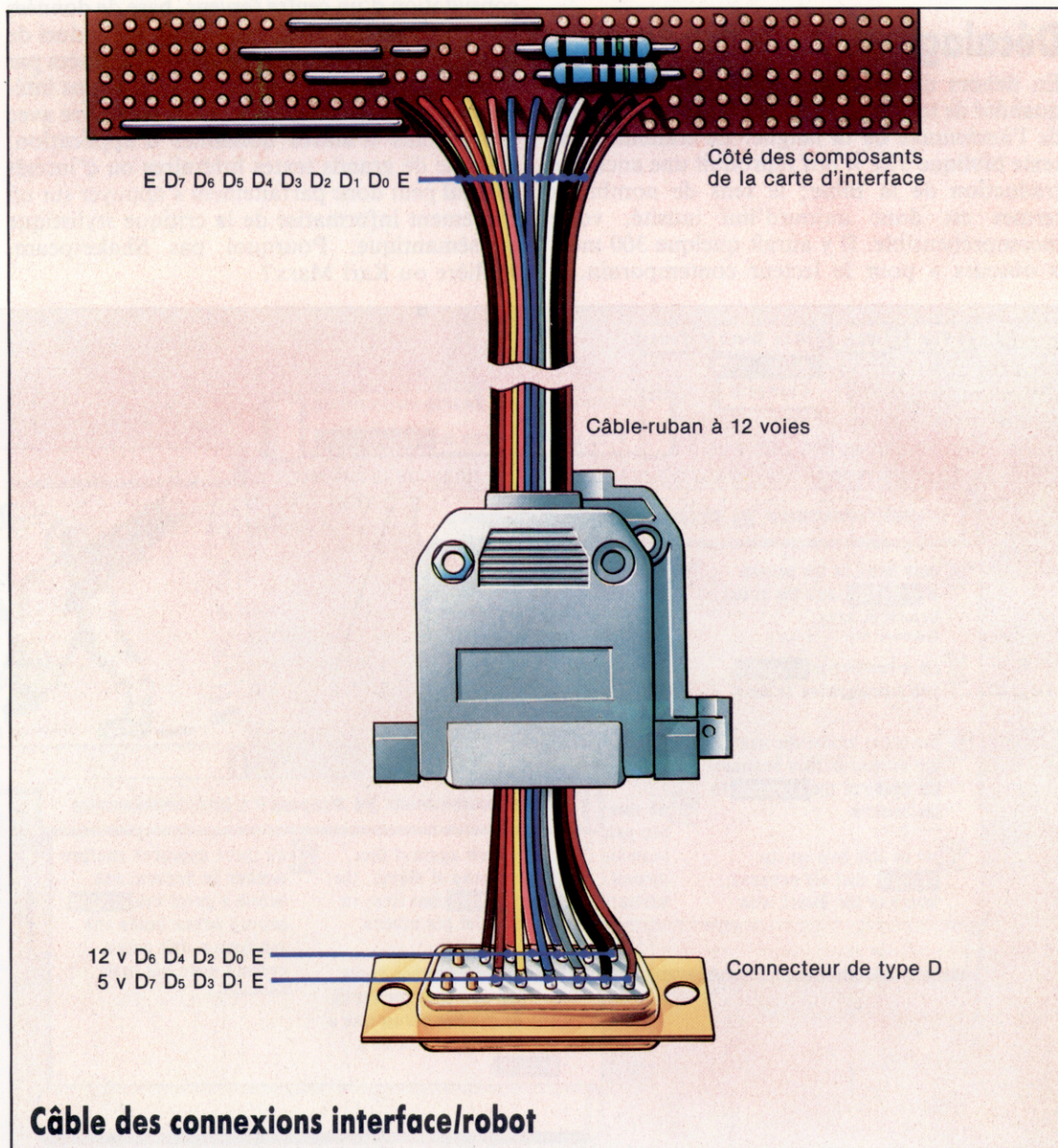
Nous avons construit une interface qui permet de commander le robot à partir du Spectrum. Construisons maintenant le conducteur qui reliera le robot à l'interface.

Les alimentations de 5 V et de 12 V sont accessibles directement à partir du port d'extension du Spectrum. Cela signifie que nous n'avons pas besoin de fournir une autre source de courant pour alimenter le robot. Préparez le câble-ruban en enlevant quatre fils du câble-ruban spécifié dans la liste des pièces. Dénudez les deux extrémités de chacun des douze brins qui restent et soudez-les à la carte et au connecteur de type D, comme nous le voyons sur le diagramme. Véri-

fiez soigneusement votre travail, en vous assurant bien que les alimentations 12 V et 5 V sont acheminées sur les bonnes broches du connecteur de type D.

L'interface est maintenant terminée, et nous pouvons la brancher dans le Spectrum pour la tester. L'ordinateur étant hors tension, branchez avec précaution la carte d'interface dans le port d'extension situé à l'arrière du Spectrum. La carte doit être insérée en veillant à ce que les

Lier un ruban à 12 voies
Reliez le connecteur de type D à la carte d'interface de la façon illustrée. Ce montage est facile à réaliser — les fils passent successivement des rangées supérieures aux rangées inférieures des broches du connecteur de type D.





composants soient sur le dessus de la carte; si vous avez inséré la fiche d'obstruction à la position 5 du connecteur du port d'extension, il est impossible d'installer la carte d'une autre manière. La carte peut être légèrement difficile à insérer. En imposant à la carte un léger mouvement de gauche à droite, vous pourrez enfoncer facilement le connecteur de la carte d'interface dans celle du port d'extension. Branchez le câble-ruban au robot et mettez sous tension. Si tout va bien, le message familier de droits d'auteur de Sinclair apparaîtra à l'écran. Nous pouvons maintenant essayer un programme test.

Commande du moteur

Le robot utilise le port d'E/S 31; ses huit bits servent à commander les moteurs et à recevoir les entrées provenant des capteurs du robot. Les quatre bits inférieurs commandent le mouvement. Les bits 1 et 2 commandent la direction de rotation des deux moteurs pas à pas utilisés. Les quatre directions sont sélectionnées en mettant ces bits dans diverses combinaisons. Le bit 3 est le bit d'impulsion. Faire passer ce bit de 0 à 1 provoque un pas de rotation pour les deux moteurs dans la direction spécifiée par le bit de direction correspondant. Le bit 0 est le bit de remise à zéro et est normalement mis à 1.

Nous pouvons tester le côté sortie de l'interface en tapant et en exécutant le programme « Commande du robot ». Les lettres T, B, F et H correspondent respectivement à avant, arrière, gauche et droite.

La routine d'initialisation définit quatre variables correspondant aux quatre directions possibles du robot. Leurs valeurs sont définies par les quatre combinaisons des bits 1 et 2 dans le port d'E/S 31. Si nous examinons ces variables et leurs quatre valeurs binaires (voir le tableau), nous pouvons comprendre le mécanisme.

La direction adoptée par le robot est définie dans la variable *dr*. Pour faire déplacer le robot dans la direction désirée, nous devons fournir des impulsions aux moteurs en mettant le bit 3 au niveau élevé puis au niveau bas. Le sous-programme de la ligne 2000 se charge de cette

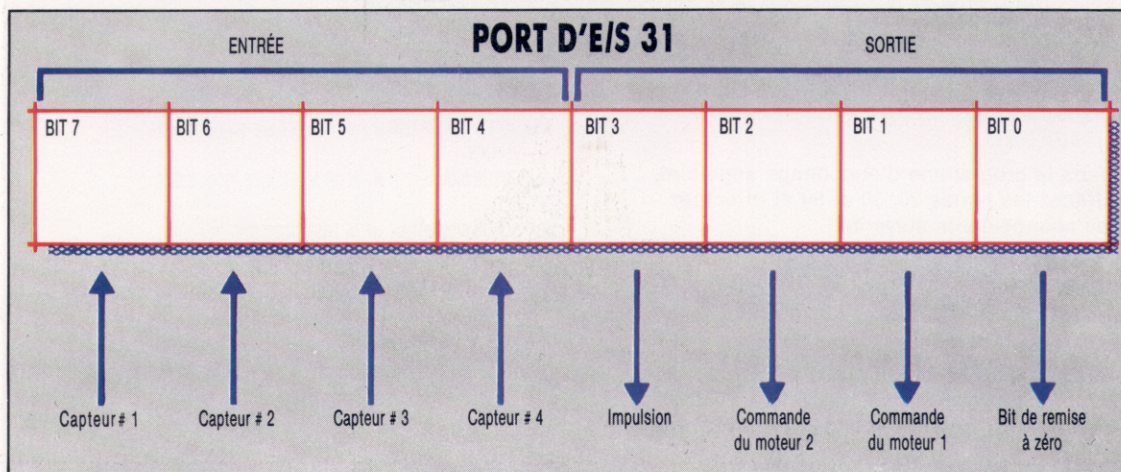
Direction	Variable	Valeur décimale	Valeur binaire
Avant	fw	4	0100
Arrière	bw	2	0010
Gauche	lf	6	0110
Droite	rt	0	0000

tâche, au moyen de OUT, ressemblant à POKE, pour placer une valeur dans le port d'E/S 31. La valeur est d'abord égale à la direction présente, *dr*, plus 8 pour mettre le bit 3 à 1, plus 1 pour mettre le bit 0 à 1 — c'est-à-dire $dr+9$. Pour mettre le 3 à 0, nous n'avons qu'à ne pas ajouter la valeur 8, ce qui nous donne $dr+1$. Passer le bit 3 de 1 à 0 fait tourner les moteurs d'un pas de $7,5^\circ$. Cela se traduit par un mouvement de moins de 1 mm au niveau de la roue; le processus d'impulsion est placé à l'intérieur d'une boucle pour entraîner les répétitions, déterminées par la variable *m*.

La direction est modifiée en appuyant sur une des touches du clavier, ce qui modifie la valeur de *dr*. Lors de l'impulsion suivante, les bits de direction des moteurs sont donc changés; le robot se déplace alors dans une autre direction.

Les quatre bits supérieurs servent à détecter les entrées provenant des capteurs du robot; chaque bit correspond à une prise du système de connexion sur le couvercle du robot. Ces bits sont normalement maintenus au niveau élevé (c'est-à-dire qu'ils ont la valeur 1), à moins qu'ils ne soient mis à la terre par la fermeture d'un interrupteur. Dans ce cas, la valeur du bit passe à 0. Pour détecter des entrées indépendantes vers chaque bit, nous devons isoler chacun des bits supérieurs et tester leurs valeurs. Dans la plupart des variantes de BASIC, c'est possible en utilisant l'instruction logique AND pour masquer les bits qui ne nous intéressent pas. Le BASIC Spectrum a une instruction AND, mais celle-ci n'est pas assez évoluée pour masquer des bits individuels dans un nombre binaire.

Le jeu des instructions du Z80 renferme cependant une opération AND qui correspond à nos besoins. Nous devons donc écrire un programme très simple en langage machine qui exécutera une opération logique AND sur deux nombres et qui



Les bits de l'action

L'interface du Spectrum met en mode sortie les quatre bits inférieurs du port 31 d'E/S et les quatre bits supérieurs en mode entrée. Les bits de sortie commandent les fonctions du moteur pas à pas; les quatre bits supérieurs servent à accepter l'entrée des capteurs.



retournera le résultat. Pour utiliser la routine, nous devons écrire les nombres que nous désirons soumettre à l'opération AND dans deux adresses de mémoire. Le résultat est retourné par la commande USR. Cette méthode indirecte nous permet de lire chaque ligne d'entrée de façon indépendante.

Le programme « Capteurs » du Spectrum teste l'action de la composante entrée de l'interface en faisant avancer le robot jusqu'à ce qu'il rencon-

tre un objet et qu'il regagne alors sa position de départ.

Les lecteurs doivent consulter les pages précédentes correspondant à cette même série d'articles pour apporter les modifications nécessaires aux versions Commodore 64. Les utilisateurs du Spectrum peuvent également remplacer les noms de variable à caractères majuscules utilisés par les versions Commodore par des noms de variable à caractères minuscules.

Programme « Commande du Robot »

```

10 REM **** COMMANDE DU ROBOT SPECTRUM ****
20 GOSUB 1000: REM INITIALISER
30 LET a$=INKEY$: IF a$<>"x" THEN
40 SUB 3000: REM LIRE TOUCHE
40 LET m=10: GOSUB 2000: REM IMPULSION
50 IF a$<>"x" THEN GOTO 30
60 OUT 31,0
70 STOP
80:
1000 REM **** initialiser ****
1010 LET fw=4: LET bw=2: LET lf=6: LET rt=0
1020 LET dr=fw
1030 RETURN
1040:
2000 REM **** IMPULSION ****
2010 FOR c=1 TO m
2020 OUT 31,dr+9
2030 OUT 31,dr+1
2040 NEXT c
2050 RETURN
2060:
3000 REM **** LIRE TOUCHES ****
3010 IF a$="t" THEN LET dr=fw: RETURN
3020 IF a$="b" THEN LET dr=bw: RETURN
3030 IF a$="f" THEN LET dr=lf: RETURN
3040 IF a$="h" THEN LET dr=rt: RETURN
3050 RETURN

```

Programme « Capteurs » du Spectrum

```

10 REM **** spectrum bumpers ****
15 CLEAR 32499: LET st=32500
20 GOSUB 500: REM initialiser
25 GOSUB 3000: RM CHANGER CODE MACHINE
30 LET dr=fw
40 REM **** IMPULSION AVANT ****
50 LET cc=0
60 GO SUB 1000: LET cc=cc+1: REM IMPULSION
70 REM ** TEST DES CAPTEURS **
80 LET nm=192: GOSUB 2000: REM OPERATION AND
90 IF USR st=192 THEN GOTO 60
100 REM **** RETOUR AU DEPART ****
110 LET dr=bw
120 FOR i=1 TO cc
130 GOSUB 1000: REM IMPULSION
140 NEXT i
150 OUT 31,0: STOP
500 REM **** initialiser ****
510 LET fw=4: LET bw=2: LET lf=6: LET rt=0
520 RETURN
1000 REM **** IMPULSION ****
1010 OUT 31,dr+9
1020 OUT 31,dr+1
1030 RETURN
2000 REM **** OPERATION AND ****
2010 POKE st+1, IN 31
2020 POKE st+3, nm: RETURN
3000 REM **** CHARGEUR CODE MACHINE ****
3010 FOR i=st TO st+8
3020 READ a: POKE i, a
3030 NEXT i
3040 DATA 62,0,14,0,161,6,0,79,201
3050 RETURN

```

Variantes basic

Dans le programme d'étalonnage linéaire donné auparavant, effacez les lignes 20, 30 et 70, et effectuez les changements suivants :

```

80 LET dr=fw
100 LET a$=INKEY$: IF a$=<> THEN GOTO 100
280 OUT 31,dr+9
290 OUT 31,dr+1

```

Dans le programme d'étalonnage angulaire, effacez les lignes 20, 30 et 50 et effectuez les changements suivants :

```

60 LET dr=rt
80 LET dr=fw
100 OUT 31,dr+9
110 OUT 31,dr+1

```

Dans le programme de mesure Robot, effacez les lignes 1010, 1020, 1030, 7010 et 7510 et effectuez les changements suivants :

```

2050 GOSUB 8100: IF USR st<> lb THEN GOTO 2040

```

```

2090 GOSUB 8100: IF USR st<> lb THEN GOTO 2080
2150 GOSUB 8100: IF USR st<> lb THEN GOTO 2140
2190 GOSUB 8100: IF USR st<> lb THEN GOTO 2180
2200 OUT 31,0
3010 CLS
3520 GOSUB 8100: IF USR st=nb THEN GOTO 3510
4010 GOSUB 8100: IF USR st=rb THEN LET ss=rt: GOSUB 5000:
RETURN
4020 GOSUB 8100: IF USR st=lb THEN LET ss=lf: GOSUB 5000:
RETURN
5020 GOSUB 8100: IF USR st=nb THEN GOTO 5010
6080 GOSUB 8100: IF USR st=nb THEN GOTO 6070
8100 REM **** perform AND ****
8110 POKE st+1, 192: POKE st+3, IN 31: RETURN
8200 REM **** chargeur code machine ****
8210 FOR i=st TO st+8
8220 READ a: POKE i, a
8230 NEXT i
8240 DATA 62,0,14,0,161,6,0,79,201
8250 RETURN

```


Des comptes clairs

Nous continuons notre étude sur les registres. Nous allons prendre un exemple pratique qui va nous permettre de voir comment fonctionnent les compteurs.

Analysons le problème suivant : un enfant possède des sacs contenant des billes de deux couleurs : blanches et noires. Il désire contrôler le nombre de sacs de billes qu'il possède, ainsi que le nombre de billes de couleurs différentes qu'il y a dans chaque sac. Il écrit toutes ces données sur papier, en indiquant le numéro du sac de billes qu'il a contrôlé et le nombre de billes selon l'ordre de la couleur prédominante. Voyons un exemple :

SAC DE BILLES 5 : BLANCHES 10, NOIRES 7

SAC DE BILLES 9 : NOIRES 20, BLANCHES 12

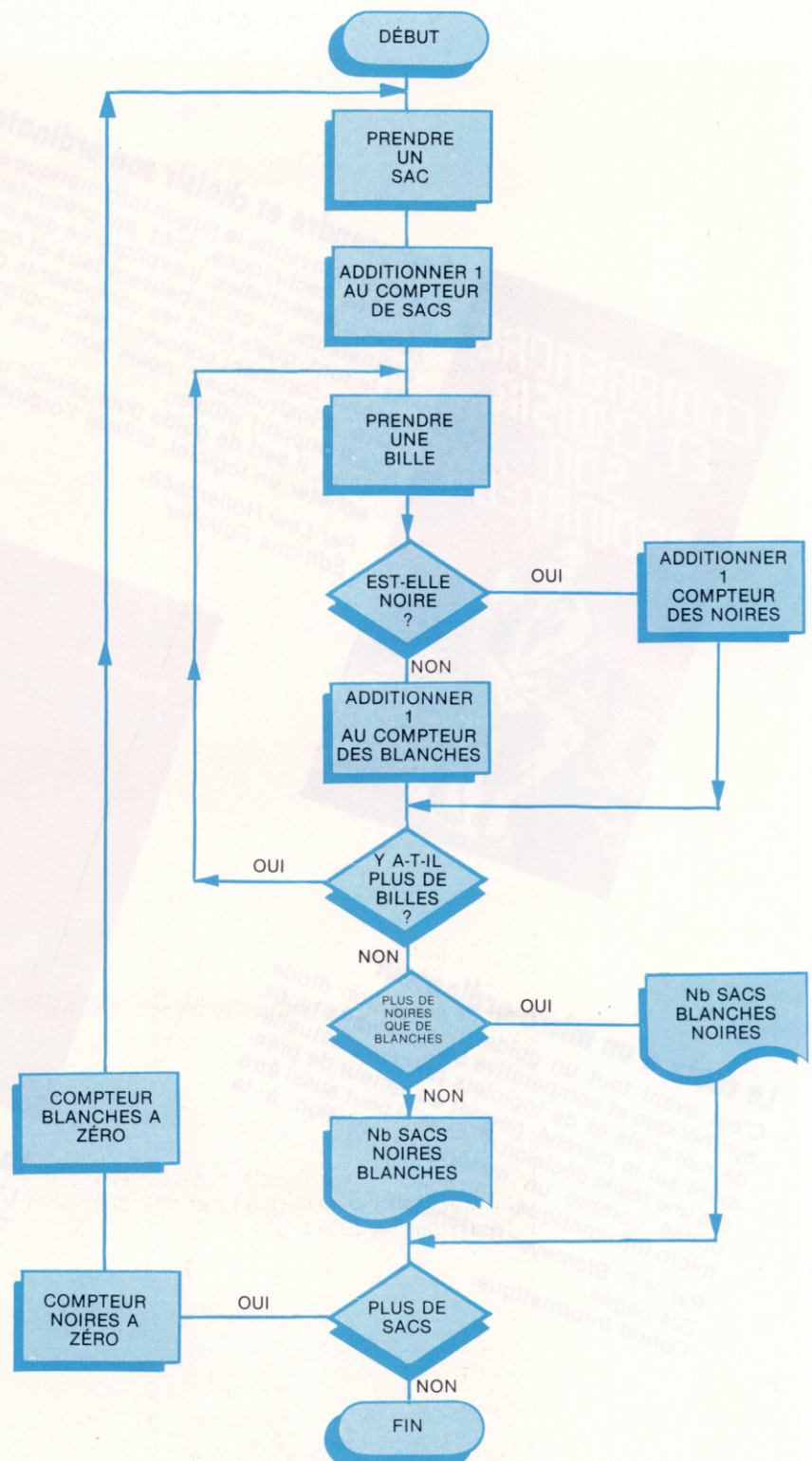
On observe ici que le programme va faire appel à trois compteurs : celui du sac de billes, qui augmentera à chaque fois avec l'arrivée d'un nouveau sac, et un autre pour chaque couleur de bille; ces deux derniers compteurs seront remis à zéro dès que l'on commencera un nouveau sac de billes. Pourquoi une telle opération? Parce que ne pas effacer le contenu du compteur au moment de prendre un nouveau sac provoquerait un faux résultat : on obtiendrait le nombre de billes accumulées de tous les sacs. Sans savoir, au préalable, le nombre de sacs disponibles, ni la quantité de billes par sac, il est possible d'obtenir des informations en interrogeant l'opérateur.

Dans le cas contraire, il ne serait pas nécessaire de consulter l'opérateur, puisque, grâce à la programmation, on contrôlerait le nombre de sacs.

```

10 REM SACS DE BILLES
15 PRINT «PRENDRE UN SAC»
20 X = X + 1
25 PRINT «PRENDRE UNE BILLE»
30 INPUT «EST-ELLE BLANCHE» (O/N);A$
40 IF A$ = «O» THEN B=B+1 : GOTO 60
50 IF A$ <> «N» THEN GOTO 30
55 N = N + 1
60 INPUT «Y A-T-IL ENCORE DES BILLES? (O/N)»;B$
70 IF B$ = «S» THEN GOTO 25
80 IF B$ <> «N» THEN GOTO 60
90 REM COULEUR PRÉDOMINANTE
100 IF B > N THEN PRINT «SAC»;X, «BLANCHES»;B,
«NOIRES»;N : GOTO 120
110 PRINT «SAC»;X, «NOIRES»;N, «BLANCHES»;B
120 INPUT «Y A-T-IL ENCORE DES SACS? (S/N)»;C$
130 IF C$ = «N» THEN END
140 IF C$ <> «O» THEN GOTO 120
150 B=0 : N=0 : GOTO 15

```



Livres pour débiter

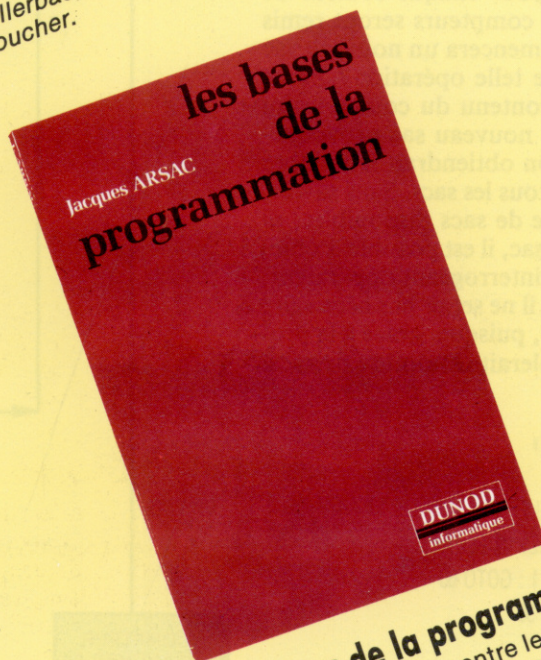
Voici une sélection de livres pour vous initier à ces étranges et merveilleuses machines que sont les micro-ordinateurs. Ils vous feront entrer dans le cercle des amateurs éclairés, puis des passionnés de micro-informatique.



Comprendre et choisir son ordinateur

Ce livre démystifie le jargon informatique et ses données techniques, tout en présentant les notions essentielles. Il explique ce que sont les ordinateurs, ce qu'ils peuvent faire et comment ils le font, quels sont les composants de l'ordinateur, comment concevoir les programmes ou jeux d'instructions, quels sont ses langages, quel support utiliser. Enfin, il sert de guide pour choisir un système, acheter un logiciel, utiliser l'ordinateur.

Par Lew Hollerbach.
Éditions Foucher.



Les bases de la programmation

L'étude des relations entre les trois piliers de la programmation que sont la récurrence, la récursivité et l'itération : comment passer de l'un à l'autre ; comment s'en servir pour fabriquer des programmes justes et les améliorer pour les rendre plus clairs ou plus performants.

Par Jacques Arsac.
288 pages. Format 15,5 cm x 24 cm.
Dunod informatique (1983).

Le choix d'un micro-ordinateur

C'est avant tout un guide qui, par son étude synthétique et comparative des différents types de matériels et de logiciels présents actuellement sur le marché, permet au lecteur de prendre une réelle décision d'achat. Il peut aussi être utilisé comme un matériel d'initiation à la micro-informatique.

Par H.P. Blomeyer-Bartenstein.
224 pages.
Dunod Informatique.



Partie de rien

Avec son Apple II et la volonté de créer un outil agréable à utiliser pour mener à bien son projet, Madeleine Hodé vient de publier un super logiciel de traitement de texte français.



L'informatique, c'est bien connu, est le domaine des jeunes, et surtout de sexe masculin. Est-ce pour démentir cette affirmation statistique qu'une femme de cinquante-cinq ans, française de surcroît, crée un logiciel de traitement de texte extraordinairement complet, performant et agréable à utiliser ?

A l'origine, Madeleine Hodé n'était pas informaticienne, elle n'était qu'utilisatrice de logiciels de traitement de texte pour son travail. Mais, se sentant « des atomes crochus avec l'informatique », elle a toujours essayé d'approcher les ateliers d'informatique partout où elle travaillait. Cependant, il n'est pas facile d'accéder à ce domaine, à moins d'être programmeur, ce que Mme Hodé n'était pas, jusqu'au jour où elle s'est décidée à acheter un micro-ordinateur Apple II+. Cet achat fut motivé, avant tout, par le traitement de texte : « Cela me paraissait être une des choses les plus intéressantes qu'on pouvait faire avec un micro-ordinateur », précise-t-elle.

Ainsi, Madeleine Hodé se lance-t-elle dans l'informatique en 1981, équipée de son Apple II+ et de la première version du traitement de texte AppleWriter. C'est alors que commence son aventure. AppleWriter lui convient parce qu'il lui permet de taper, d'enregistrer ses textes, de les corriger et de les imprimer. L'essentiel y est, mais, pense Mme Hodé, l'essentiel ne suffit pas, et peu à peu les défauts de presque tous les logiciels de traitement de texte lui apparaissent : ils ne coupent pas les mots en fin de ligne.

La plupart de ces logiciels, étant conçus à partir de traductions de programmes anglais ou américains, avaient, en effet, peu de chance de faire des coupures. Celles-ci nécessitent des algorithmes très différents des français, et l'on n'y observe pas la même régularité. En français, au contraire, les règles sont relativement simples : il faut couper un mot entre deux syllabes.

Mme Hodé commence donc à chercher comment on pourrait s'y prendre pour résoudre ce

A la ligne !

Madeleine Hodé n'était pas informaticienne mais utilisait des logiciels de traitement de texte. A force de trouver ces logiciels peu adaptés au français, elle en vint à créer le sien, Gribouille.



problème. Elle y réfléchit à tout moment, en faisant autre chose, pendant ses loisirs, au restaurant ou dans l'autobus. Tout de suite, ce problème la passionne, comme un exercice intellectuel amusant.

Elle aboutit d'abord à trouver un algorithme (voir encadré), puis elle commence par le programmer en BASIC. Ayant créé des textes avec son AppleWriter, elle les enregistre et les reprend ensuite avec son programme BASIC. Elle parvient ainsi à réaliser des coupures de mots correctes. Mais la tâche n'est pas simple, car il faut toujours passer d'un programme à l'autre.

Le doigt était pris dans l'engrenage, et il n'était plus possible à notre pionnière de reculer. Un énorme travail l'attendait encore : la conception d'un logiciel de traitement de texte exige au moins un ou deux ans de travail, surtout si l'on exerce un métier à côté. Mais Madeleine Hodé, qui n'avait aucune formation préalable, a dû passer beaucoup plus de temps que cela.

Tout d'abord, elle a dû apprendre l'assembleur pour comprendre le contenu du programme AppleWriter. Elle passa deux ou trois semaines avant de trouver où le programme commençait et ce qu'il faisait. Ensuite, ce fut comme une boîte qui s'ouvre : petit à petit, elle réussit à lire, à comprendre comme une langue étrangère qu'on apprend à parler. Et, enfin, elle arrive à programmer elle-même directement dans ce langage. Dès lors, le problème qui se posait n'était plus le programme proprement dit, c'était la façon de commander aux périphériques que sont l'imprimante, l'écran et le lecteur de disquette. En achetant des livres, en cherchant, en lisant, elle parvient enfin à réaliser son programme, rudimentaire certes, mais qui coupait les mots.

Un beau jour, elle se décide à envoyer le résultat à Jean-Louis Gassée, le patron d'Apple France. Celui-ci lui répond avec enthousiasme que le programme l'intéresse beaucoup.

Dès lors, Madeleine Hodé n'est plus seule. Elle a chez Apple des correspondants prêts à l'aider.

Qu'est-ce qu'un algorithme?

Pour Madeleine Hodé, il s'agissait de trouver un algorithme pour couper les mots à la fin des lignes.

Lorsque, par exemple, on analyse un mot et la façon dont il est structuré dans ses lettres, il y a tout un raisonnement à faire pour trouver à quel endroit s'effectue le passage d'une syllabe à l'autre, afin de pouvoir y placer la coupure. Le fait de réitérer ce raisonnement et d'appliquer toujours la même analyse à différents stades successifs, afin de trouver la solution du problème, est appelé algorithme.

Non pas à programmer, car lorsqu'on fait un programme en assembleur, tel un pianiste, on ne peut se faire remplacer pour les passages difficiles. Elle fait donc le travail complètement seule, mais Apple lui fournit toute la documentation nécessaire sur les machines. Elle ne s'inspire nullement des autres logiciels car, en voyant ce qu'ont fait les autres, elle a le sentiment de perdre toute créativité. En revanche, ses correspondants d'Apple la conseillent, lui disent quelles sont les trouvailles originales, quels points doivent être développés, ils lui recommandent d'utiliser la carte 80 colonnes. Ils lui donnent leur avis, compte tenu de l'évolution des techniques et de la concurrence existant entre les différents traitements de texte.

C'est ainsi qu'au bout d'un ou deux ans, Madeleine Hodé a fini par élaborer un programme qui, d'amélioration en amélioration, était devenu présentable. A partir de ce moment s'est posé le problème de la commercialisation. Ce n'était pas facile du tout. Les premières tentatives n'ayant pas abouti, Mme Hodé perd du temps, mais elle profite de ce délai pour continuer à améliorer son logiciel, à développer des fonctions nouvelles, par exemple la création de caractères sur l'imprimante pour la notation mathématique.

De même qu'un roman, un programme n'est jamais entièrement fini; il peut toujours être perfectionné, ses applications peuvent être élargies. Mais il arrive un moment où il faut s'arrêter et passer au second acte qui consiste à savoir comment le commercialiser. Cette étape comporte la rédaction d'un manuel. Notre auteur dut rédiger plusieurs manuels successifs, les premiers étant soit trop courts, soit trop difficiles à comprendre. Cette rédaction lui permet d'améliorer encore son programme afin de le rendre plus agréable pour les utilisateurs.

Enfin, elle parvient à prendre contact avec l'agence Berlingot, qui distribue son programme. Avec ce partenaire qui connaît bien la partie commerciale, le nouveau logiciel est enfin lancé sous le nom de « Gribouille ».

A présent, Madeleine Hodé espère que son programme remportera tout le succès mérité par tant de travail. Pourtant, ces années de travail acharné, elle ne les regrette pas du tout; au contraire, affirme-elle, « c'est l'une des périodes les plus heureuses de mon existence ».

Gribouille

Exemple d'écran où l'on voit que le logiciel de traitement de texte Gribouille « s'explique ».

