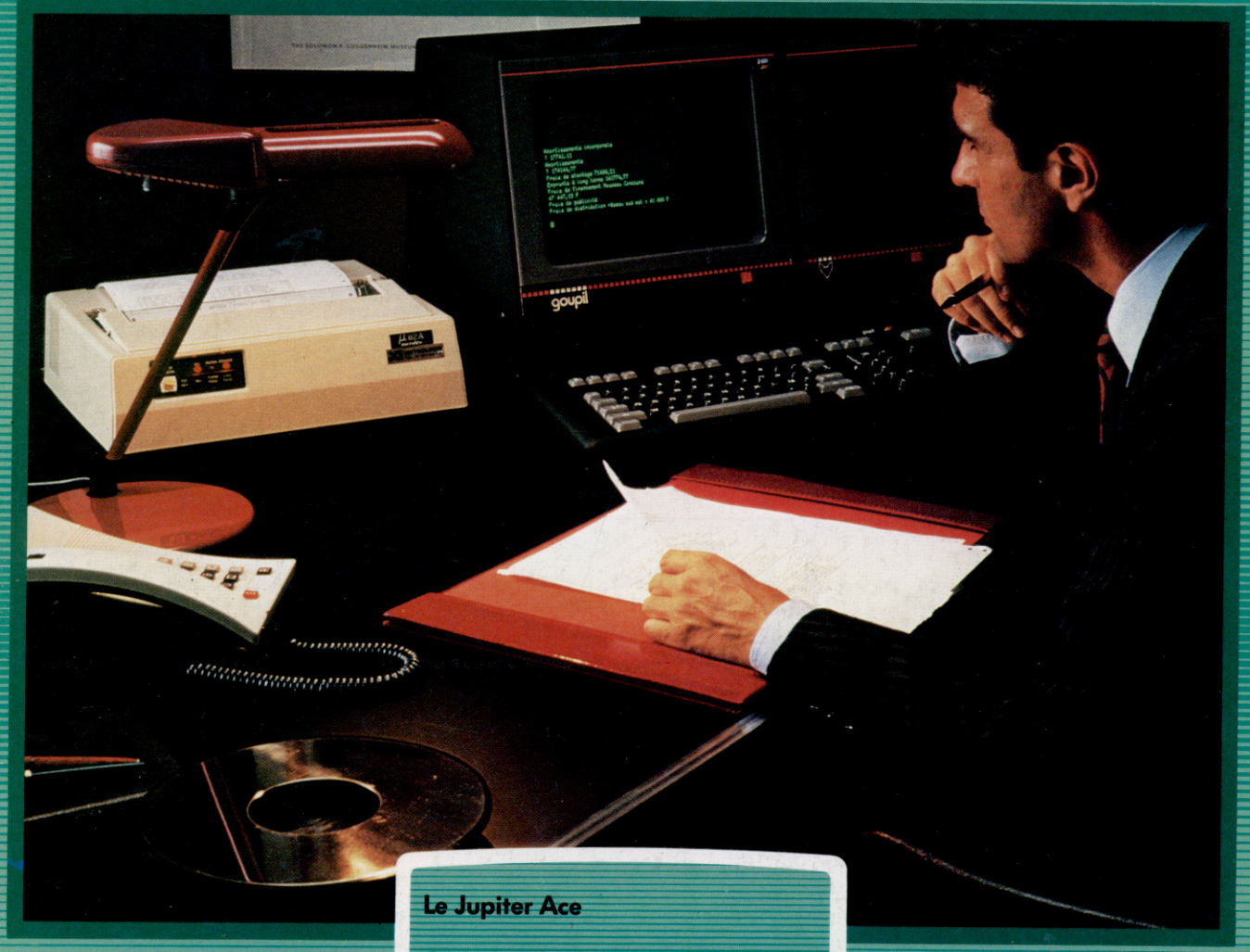
ABC INFORMATIQUE

N° 8

COURS D'INFORMATIQUE PRATIQUE ET FAMILIALE



Le Jupiter Ace

Gérer son budget avec l'ordinateur

L'ordinateur musicien

L'homme à la pomme

EDITIONS ATLAS

M 6062-8-12 F 85FB-3,80FS-$1.95

Sommaire

Prochain fascicule

● Les ordinateurs portatifs vont de l'ordinateur de poche à l'ordinateur en mallette, mais tous offrent un réel potentiel. Nous passons en revue les derniers développements.

● Vous aimeriez intervenir dans le déroulement d'une histoire ? Les jeux d'aventures vous en donnent la possibilité, mais attention au dragon !

● Nous présentons l'ordinateur portatif Epson HX-20, la première machine à offrir une imprimante et une unité à cassette intégrées ainsi qu'un BASIC comportant un éventail complet de commandes.

aBc INFORMATIQUE
cours d'informatique pratique et familiale
EDITIONS ATLAS

Édité par ÉDITIONS ATLAS s.a., tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris *Cedex 15*. Tél. : 538-52-70.

Belgique : ÉDITIONS ATLEN s.a., Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, Montréal Nord.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, Mezzovico.

Réalisé par EDENA s.a., 29, boulevard Edgar-Quinet, 75014 Paris. Tél. : 320-15-01.

Direction éditoriale : J.-Fr. Gautier. *Service technique et artistique* : F. Givone et J.-Cl. Bernar. *Iconographie* : J. Pierre. *Correction* : B. Noël.

VENTE

Les numéros parus peuvent être obtenus chez les marchands de journaux ou, à défaut, chez les éditeurs, au prix en vigueur au moment de la commande. Ils resteront en principe disponibles pendant six mois après la parution du dernier fascicule de la série. (Pour toute commande par lettre, joindre à votre courrier le règlement, majoré de 10 % de frais de port.)

Pour la France, s'adresser à ÉDITIONS ATLAS, tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris *Cedex 15*. Tél. : 538-52-70.

Pour les autres pays, s'adresser aux éditeurs indiqués ci-dessous.

SOUSCRIPTION

Les lecteurs désirant souscrire à l'ensemble de cet ouvrage peuvent s'adresser à :

France : DIFFUSION ATLAS, 3, rue de la Taye, 28110 Lucé. Tél. : (37) 35-40-23.

Belgique : ÉDITIONS ATLEN s.a., 55, avenue Huart-Hamoir, 1030 Bruxelles. Tél. : (02) 242-39-00. Banque Bruxelles-Lambert, compte n° 310-0018465-24 Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, 11450 boulevard Albert-Hudon, Montréal Nord, H 1G 3J9.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, zona industriale 6849 Mezzovico-Lugano. Tél. : (091) 95-27-44.

RELIEZ VOS FASCICULES

Des reliures mobiles, permettant de relier 12 fascicules, seront en vente en permanence chez votre marchand de journaux.

ATTENTION : ces reliures, présentées sans numérotation, sont valables indifféremment pour tous les volumes de votre collection. Vous les numéroterez vous-même à l'aide du décalque qui est fourni (avec les instructions nécessaires) dans chaque reliure.

En vente tous les vendredis. Volume I, n° 8.

ABC INFORMATIQUE est réalisé avec la collaboration de Trystan Mordrel *(secrétariat de rédaction)*, S.I.-André Larochelle *(traduction)*, Ghislaine Goullier *(fabrication)*, Marie-Claire Jacquet *(iconographie)*, Patrick Boman *(correction)*. Crédit photographique, couverture : Photo SMT-Goupil.

Directeur de la publication : Paul Bernabeu. Imprimé en Belgique par Proost, à Turnhout. Distribution en France : N.M.P.P. Tax. Dépôt légal : février 1984. 24842. Dépôt légal en Belgique : D/84/2783/27.

A NOS LECTEURS

En achetant chaque semaine votre fascicule chez le même marchand de journaux, vous serez certain d'être immédiatement servi, en nous facilitant la précision de la distribution. Nous vous en remercions d'avance.

Les Éditions Atlas

La voix de son maître

Grâce aux synthétiseurs, les ordinateurs font déjà partie de l'univers musical professionnel. De tels instruments commencent à être intégrés dans de nombreux ordinateurs domestiques.

Outre traiter l'information et exécuter des jeux, les ordinateurs peuvent aussi servir à faire de la musique, ils ont alors pour nom synthétiseurs musicaux. On peut également se servir de l'ordinateur pour enseigner la musique. Bientôt, nous vous expliquerons comment faire de la musique sur votre ordinateur domestique, mais nous nous limiterons pour le moment au travail des professionnels. Le secteur professionnel est important parce que c'est de lui que proviennent toutes les idées qui sont plus tard appliquées dans les produits à grande diffusion.

Les instruments automatiques ont toujours été populaires et ont plusieurs points en commun avec les ordinateurs. Les pianos mécaniques ont eu beaucoup de succès au siècle dernier et étaient actionnés par une bande de papier perforée; des boîtes musicales comportaient un tambour ou un disque métallique muni de « dents » qui jouaient une mélodie sur un peigne métallique.

Les orgues de Barbarie étaient en un sens programmables puisqu'il était possible de modifier leur mélodie. Néanmoins, ces instruments ne plurent pas à Charles Babbage, l'un des pères de l'informatique, puisqu'il demanda leur interdiction.

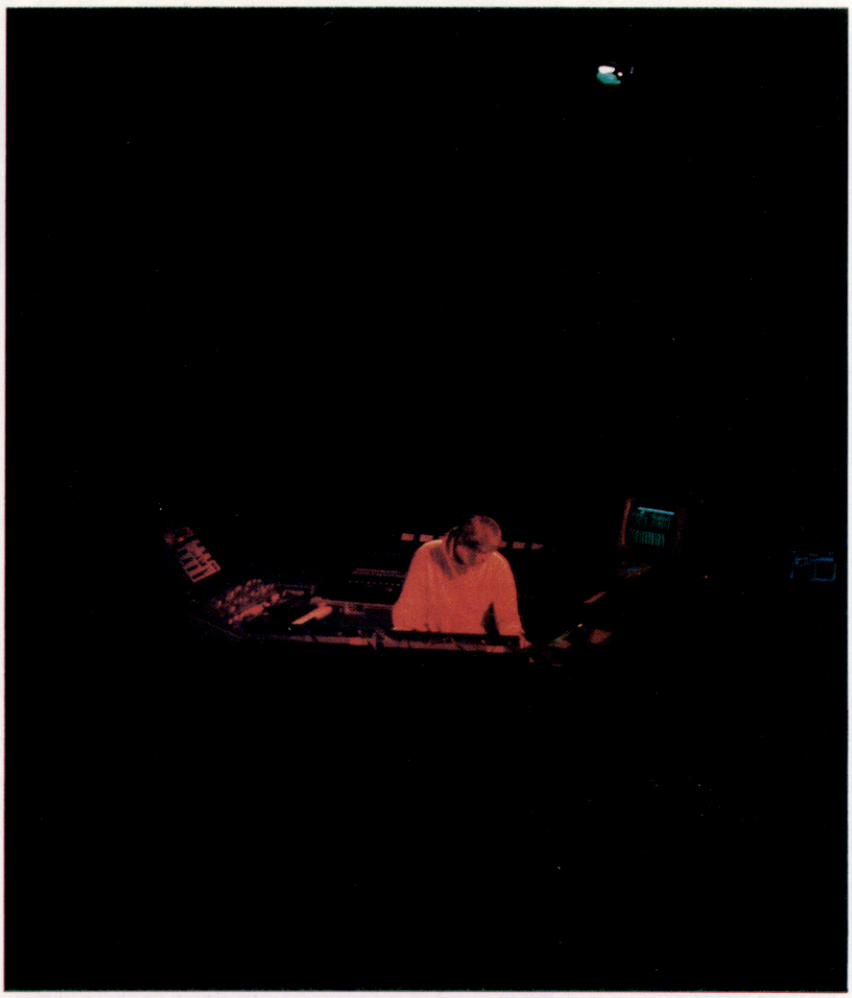
Aujourd'hui, plusieurs syndicats de musiciens sont partis en guerre contre les instruments programmables; en 1982, la section londonienne vota l'interdiction de leur usage lors des sessions d'enregistrements et lors des concerts. Ils estiment évidemment que ces appareils menacent leurs emplois.

Les synthétiseurs musicaux sont utilisés depuis de nombreuses années, mais l'arrivée des techniques numériques a accru leur potentiel. Plus besoin de sans cesse manipuler des boutons pour produire chaque son, il est maintenant possible d'enregistrer un son, de l'analyser et de le reproduire fidèlement.

On peut comparer le son numérique à une photo de journal. Si vous examinez soigneusement la page, vous verrez que l'image est composée de milliers de points pour représenter là toutes les nuances de photographie initiale (analogique). De même, le son analogique ordinaire peut être exprimé par une série de nombres. On nomme cette technique échantillonnage.

La digitalisation des sons permet non seulement leur traitement par les ordinateurs à des fins musicales, mais aussi facilite la transmission de données et de la parole. Les militaires ont fait appel à des techniques de ce type pour rendre plus fiable leur réseau de transmissions. Ainsi, la parole est découpée en une multitude de segments qui sont transmis en ordre dispersé à travers une large gamme d'ondes. Le récepteur les recueille et les rend à nouveau intelligibles. Si, pendant le parcours, l'adversaire tente de brouiller la communication, il lui sera impossible de couvrir toute la gamme d'ondes utilisée et le récepteur recevra néanmoins suffisamment d'éléments de parole pour que le message soit compréhensible. A l'inverse, toute tentative d'interception est vouée à l'échec car les parties de message captées seront en nombre insuffisant pour avoir un sens.

De tels systèmes sont coûteux — le Fairlight et le Synclavier sont les modèles les plus connus et les plus évolués — mais peuvent reproduire les sons de nombreux instruments et donc s'avérer une solution plus économique que l'embauche de plusieurs musiciens.

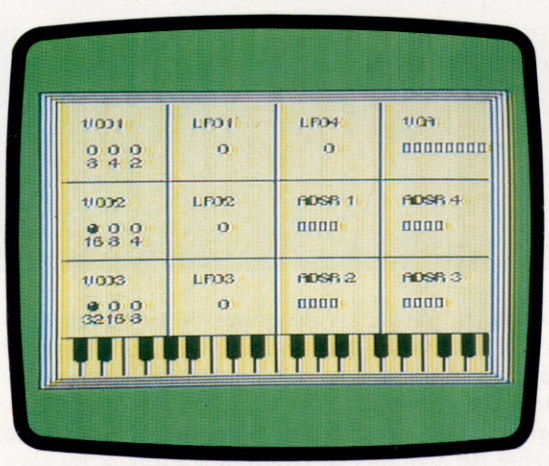


Homme-orchestre
Les joueurs de synthétiseurs comme Klaus Schultz (photographié ici) exploitent de plus en plus le potentiel de leurs « instruments » électroniques et réussissent à produire sur scène un son qui, il y a vingt ans, aurait nécessité le concours d'un orchestre au complet. (Cl. EMI.)

Effets spéciaux
Les synthétiseurs musicaux commandés par ordinateur deviennent de plus en plus populaires. Les effets musicaux qui, il y a dix ans, n'étaient possibles que sur des équipements professionnels très coûteux, sont maintenant produits par des équipements valant entre 1 500 F et 3 000 F. L'appareil photographié a la possibilité de charger en mémoire un morceau de musique codé à l'aide de codes à barres. Ce morceau peut dès lors être reproduit ou modifié par le musicien. Plusieurs de ces synthétiseurs domestiques peuvent être reliés directement à un ordinateur domestique pour ainsi profiter de l'écran et d'une mémoire additionnelle. En outre, les ordinateurs domestiques intègrent de plus en plus des fonctions de synthèse musicale. (Cl. Ian McKinnell.)

Avec la baisse du prix des ordinateurs et des mémoires, les machines numériques deviennent de plus en plus populaires, mais cela n'implique pas la disparition des synthétiseurs analogiques. Ces derniers exploitent une technique dite « synthèse soustractive » qui est comparable à l'exécution d'une sculpture. Un son de base créé électroniquement est soumis à divers processus électroniques. Chaque processus soustrait des composantes du son pour lui donner la nature désirée. La synthèse soustractive encourage l'essai de diverses combinaisons de processus et est relativement à la portée du débutant.

Par contre, toute création sur synthétiseur numérique doit être soigneusement planifiée



Boîtes à musique
Plusieurs ordinateurs domestiques offrent maintenant des programmes qui exploitent leur potentiel musical. L'affichage peut être utilisé par ces programmes pour illustrer la musique qui est jouée, ou peut aider le musicien débutant à se servir du clavier de l'ordinateur comme d'un clavier musical. (Cl. Ian McKinnell.)

puisque la machine effectue une synthèse additive; le son final est obtenu par l'addition cumulative des composantes. Le son n'est identifiable qu'à la toute fin du processus. Il est cependant possible de prendre un son conventionnel, d'extraire ses composantes élémentaires, de les stocker en RAM, sur disquette ou sur bande, et de les utiliser pour construire le son recherché.

En plus de pouvoir créer une variété de sons, les ordinateurs peuvent aussi stocker des séquences musicales et des compositions. Les synthétiseurs les plus populaires comprennent ou offrent en option un séquenceur (le dispositif qui stocke et rappelle les séquences sonores). Par exemple, le Fairlight peut stocker jusqu'à trente minutes de sons sur disquette et jusqu'à huit « voix » pouvant représenter des instruments individuels. Le synclavier double ce nombre.

On peut voir facilement qu'avec les synthétiseurs les plus coûteux (200 000 F ou plus), une seule personne peut faire le travail du compositeur et d'un orchestre au complet et doit être assez qualifiée en programmation.

Si vous essayez d'écouter une bande à une vitesse trop rapide, vous noterez que le ton monte considérablement (comme lorsqu'on écoute un disque 33 tours à un réglage 45 tours). Les synthétiseurs commandés par ordinateur peuvent annuler cet effet et jouer un morceau musical plus lentement ou plus rapidement sans modifier le ton, ou réciproquement peuvent passer à un autre registre sans modifier la vitesse.

Il est possible d'enregistrer un son de trompette, de le reproduire et de créer simultanément un son de cor français. Les deux instruments peuvent alors jouer à l'unisson ou créer une harmonie.

Plusieurs de ces machines sont commandées au moyen d'un langage de composition (celui du Synclavier se nomme SCRIPT) un peu similaire au BASIC (avec des numéros de lignes) mais peut-être un peu plus difficile. La fonction la plus remarquable est la compilation inverse; vous jouez un morceau de musique au clavier, et l'ordinateur produit un listing en SCRIPT de votre composition. C'est comme si vous inventiez un jeu sur l'écran de votre ordinateur et n'ayez qu'à appuyer sur un bouton pour obtenir le listing complet de votre nouveau jeu!

Si l'exécution du morceau n'est pas parfaite, il est possible d'éditer le listing SCRIPT à l'aide d'un clavier conventionnel et d'un écran tout comme en BASIC. Un système moins souple mais plus attrayant accompagne les synthétiseurs

Yamaha : ils impriment les notes jouées sur une portée.

Les synthétiseurs sont aussi utilisés au cinéma. Le film *Tron* de Walt Disney suscita d'élogieux commentaires pour ses remarquables graphiques créés par ordinateur. Mais on ignore généralement que des micro-ordinateurs furent employés pour la musique et pour les effets sonores. En plus de la « véritable » musique exécutée par le London Philharmonic Orchestra, une grande variété de sons furent produits à l'aide d'un Fairlight.

Un tel nombre de sons étaient impliqués qu'un catalogue complet a dû être établi sur un Atari 800 à l'aide du programme File Manager 800 et d'un logiciel de base de données. Certaines des voix du film furent créées à l'aide d'un ordinateur domestique et d'un synthétiseur de parole qui permettait de mixer voix et musique. Atari fournit également un logiciel inédit à l'extérieur de leur société. Il s'agit en quelque sorte d'un programme de synthèse additive qui a été employé pour créer les effets sonores complexes de leurs propres programmes de jeux et de machine d'arcade. Ce système crée des sons aussi facilement qu'un programme de traitement de texte produit un texte.

Il exite de nombreux dispositifs complémentaires qui offrent la possibilité de transformer un ordinateur domestique en synthétiseur. L'Apple est un ordinateur très populaire pour cette application en raison de ses connecteurs permettant de brancher des dispositifs additionnels. Certains de ces dispositifs confient toute la création à l'ordinateur et ne fournissent qu'un synthétiseur analogique de bonne qualité comme sortie, d'autres possèdent un véritable clavier musical.

Dans la portion domestique du marché des synthétiseurs, mentionnons le Casio CT7000 qui comporte un « séquenceur polyphonique » qui, à l'aide d'une unité à cassette et d'une mémoire RAM importante, peut accomplir le travail d'un système d'enregistrement multipistes professionnel. Une cassette ordinaire n'a qu'une piste de chaque côté, une bande stéréo en a deux, mais une unité professionnelle peut en compter plus de 24, de façon à ce que chaque instrument puisse être enregistré séparément, puis mixé pour former le son global. Avec le CT7000, vous pouvez composer vos propres polyphonies de la même manière.

Son prédécesseur, le CT701, utilise un lecteur de codes à barres (voir page 40) pour entrer en mémoire de la musique à partir de codes à barres imprimés. Malheureusement, l'utilisateur ne peut créer lui-même ces codes à barres, cette méthode est donc limitée à la lecture de morceaux écrits et diffusés par Casio.

Mais le groupe allemand Trio a prouvé qu'il n'est pas nécessaire de posséder le meilleur équipement pour réussir dans le monde de la musique pop. Ce groupe a eu plus de succès avec sa chanson « Da Da Da », utilisant un Casiotone VL1 ne coûtant que 450 F, que Peter Gabriel avec son Fairlight valant une fortune.

Même si le VL1 est un dispositif monophonique (ne pouvant jouer qu'une note à la fois), il peut imiter divers instruments et stocker une séquence de notes. Il est également possible de définir les sons produits en modifiant l'enveloppe sonore (attaque, chute, maintien, fin - ADSR). Certains ordinateurs domestiques, comme le Commodore 64 et l'Oric-1, offrent exactement les mêmes fonctions, et puisqu'ils peuvent être programmés en BASIC, ils comportent eux aussi un séquenceur intégré.

Plusieurs ordinateurs domestiques disposent de logiciels de composition musicale — même ceux dotés d'un potentiel musical limité. Certains de ces programmes affichent une portée où un morceau est composé en choisissant des notes à l'aide d'un manche à balai ou d'un crayon électronique et en les plaçant sur la portée. Il ne reste plus alors qu'à appuyer sur le bouton de mise à feu ou à donner une commande tout aussi simple pour entendre le morceau composé. L'affichage peut aussi représenter un clavier de piano, les notes sont alors de nouveau sélectionnées à l'aide d'un crayon électronique ou d'un manche à balai, ou à partir du clavier de l'ordinateur.

Des effets musicaux peuvent aussi être programmés en BASIC sans l'aide de tels programmes. Tout comme pour les fonctions graphiques, le niveau de raffinement des commandes associées aux fonctions sonores et leurs méthodes d'utilisation varient considérablement d'une machine à l'autre. Le Dragon n'a qu'une voix mais comporte une commande PLAY qui définit une séquence de note (A à G). Le Commodore 64 offre un excellent potentiel musical au niveau matériel, mais son BASIC ne comporte pas de commandes aussi faciles à utiliser. Nous expliquerons dans un prochain article comment produire de la musique avec votre machine.

François Couturier
Ce musicien français de réputation mondiale est un ardent défenseur des synthétiseurs musicaux commandés par ordinateur.
(Cl. Jean-Pierre Leloir.)

L'adressage direct

L'UC doit localiser les instructions et les données stockées dans les milliers d'adresses de la mémoire de l'ordinateur. Voici ce qui se passe à l'intérieur de l'UC lors de l'exécution d'un programme.

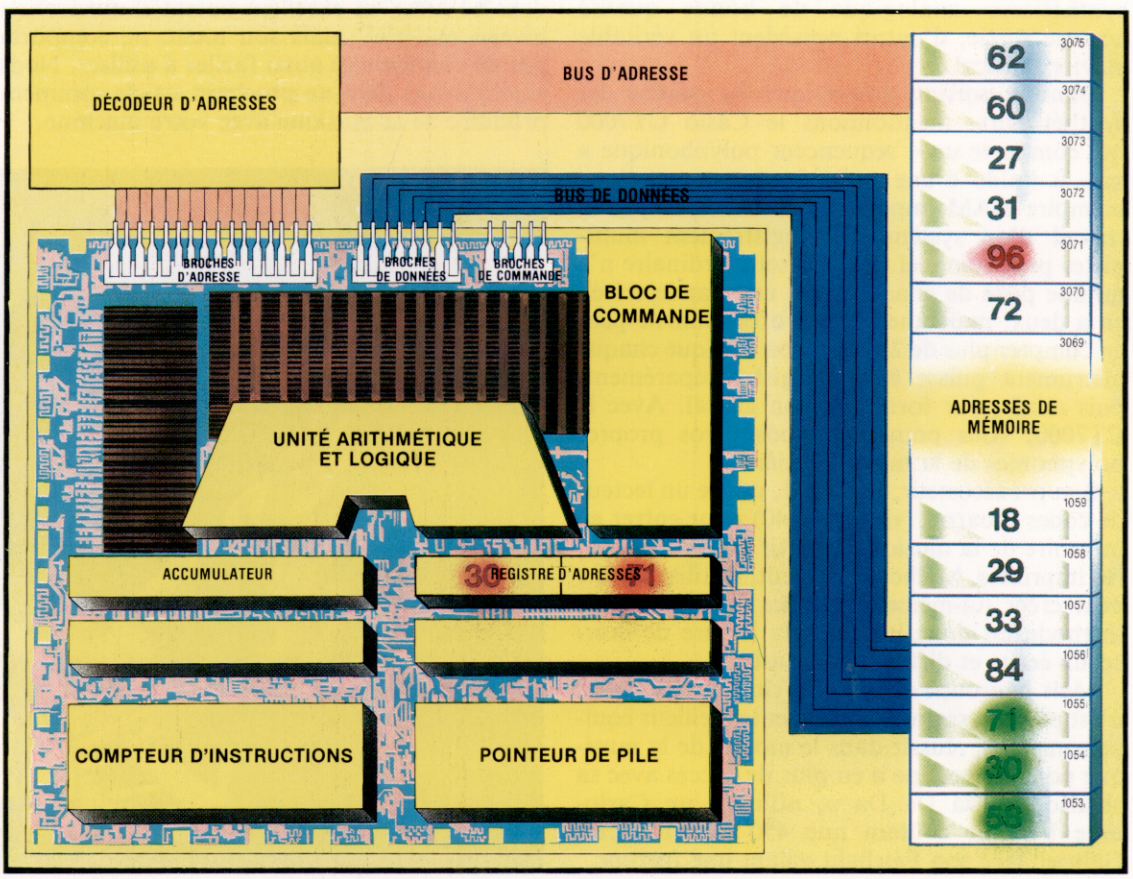
L'unité centrale (UC) reçoit ses instructions et ses données en provenance de la mémoire en réglant ses broches d'adresse selon le code binaire de la mémoire; la lecture des instructions de la mémoire se faisant *via* le bus de données. Mais le processus réel est en fait plus complexe.

Les octets (codes binaires de huit bits) contenus dans les milliers de cellules de la mémoire de l'ordinateur peuvent être des instructions commandant l'UC, ou des données qui doivent être traitées par l'UC. Comment l'UC peut-elle discerner, sans risque de se tromper, les instructions des données?

Reconnaissance des codes

Voyons d'abord ce qu'est une instruction. C'est un code binaire qui fait effectuer une séquence spécifique d'opérations à l'UC. Si par exemple, le code 00111010, S est reconnu par l'UC comme une instruction et non comme un élément de données, celui-ci commandera à l'UC de lire les deux prochains octets de S mémoire, de mettre leurs données dans un « registre d'adresses », de régler S broches d'adresse selon ce nombre, d'aller à cette adresse, dans la mémoire, d'extraire l'information contenue à cette adresse *via* le bus de données, puis de charger cette information dans son accumulateur.

Cela peut sembler un peu compliqué. Et pourtant ce n'est là qu'une des méthodes d'adressage de mémoire utilisées par la traditionnelle UC Z80. La figure illustre le processus de transfert d'un octet de la mémoire à l'UC. Supposons que l'UC sache déjà que le prochain octet de la mémoire sera une instruction (et non pas une donnée) et que cet octet réside à l'adresse 1053 de la mémoire (tous les nombres de l'illustration sont volontairement décimaux). Cette adresse, 1053, sera définie sur le bus d'adresse en langage binaire par : 0000010000011101. Les 16 broches d'adresse sont alors mises sur « on » ou sur « off » afin de correspondre à ce nombre. Lorsque le « décodeur d'adresses » reçoit cette adresse par le bus



Événements en série
Même la plus simple opération de l'UC implique de nombreuses interventions. Des instructions, aussi appelées « codes opération », sont lues par l'UC dans la mémoire. Ces instructions sont décodées par le bloc de commande et entraînent le déroulement d'interventions spécifiques. Dans cet exemple, l'instruction 58 est lue dans l'adresse de mémoire 1053. Cette instruction provoque la série d'événements suivants : l'octet de l'adresse (1054) est lu et stocké dans une moitié du registre d'adresse à 16 bits de l'UC. L'octet de l'adresse suivante (1055) est lu et stocké dans l'autre moitié. Ces deux octets représentent l'adresse de mémoire où certaines données sont stockées. Le contenu du registre d'adresse est maintenant placé sur le bus d'adresse, de façon que la prochaine adresse de mémoire sollicitée soit l'adresse 3071. Le contenu de cette adresse est mis sur le bus de données et lu par l'UC. Cet octet (96 dans notre exemple) est ensuite placé dans l'accumulateur de l'UC, où il demeure jusqu'à ce qu'il soit traité par la prochaine instruction. Le bus d'adresse revient alors à l'adresse précédente + 1, il accède donc à l'adresse 1056. L'UC sait que le contenu de cette adresse est une instruction et qu'une séquence similaire d'interventions devra être répétée. Ici l'instruction suivante est 84, elle est interprétée par le bloc de commande comme un ordre d'inversion des bits présents dans l'accumulateur. Puisque 84 est une instruction « à un octet », l'UC sait que l'octet placé à l'adresse suivante, 1057, sera encore une instruction.

d'adresse, il la décode et sollicite l'une de ses lignes de sortie. C'est cette ligne qui choisit l'adresse 1053 dont le contenu est 58.

Ce contenu de l'adresse 1053 (00111010 en binaire) est alors placé sur le bus de données et chargé dans l'UC. Ici, puisque l'UC attend une instruction, l'octet est interprété par le bloc de commande et entraîne l'exécution d'une séquence très précise d'opérations. Cette instruction particulière spécifie que les deux prochains octets en mémoire contiendront 16 bits qui désigneront une adresse de mémoire et que le contenu de cette adresse doit être chargé dans l'accumulateur de l'UC. Dès que l'UC reconnaît cette instruction, elle sait que les deux prochains octets en mémoire désigneront une adresse et que le contenu de cette adresse doit être chargé dans l'accumulateur. Elle sait par conséquent qu'elle ne recevra pas d'autre instruction en provenance de la mémoire tant que ces opérations n'auront pas été effectuées, et que la prochaine instruction résidera à l'adresse 1056.

L'instruction utilisée comme exemple incrémente le bus de données de un. La prochaine adresse sollicitée est donc 1054. Le contenu de cette adresse (30) est alors mis sur le bus de données et chargé dans l'UC, non pas dans l'accumulateur, mais dans la première moitié du registre d'adresses. Après quoi, l'UC incrémente de nouveau le bus d'adresse de façon à ce qu'il fasse apparaître l'adresse 1055. Le contenu de cette adresse (71) va sur le bus de données et est chargé dans l'UC de façon similaire, mais dans l'autre moitié du registre d'adresses.

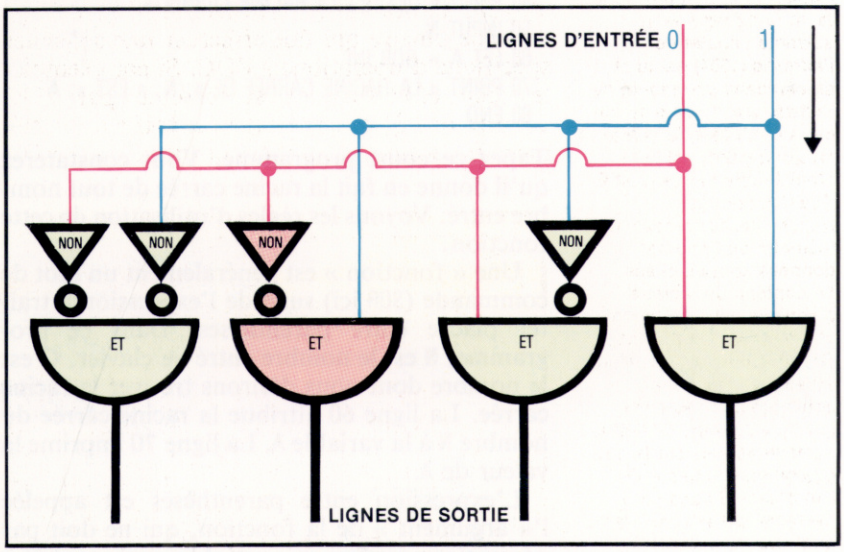
Transfert de nombres

Les nombres 30 et 71 dans le registre d'adresses de l'UC sont alors transférés au bus d'adresse (toutes ces interventions se produisant automatiquement en réponse à l'instruction initiale). Ces deux nombres couplés correspondent en fait à 3071. L'emplacement mémoire à sélectionner est donc désormais 3071. Cette nouvelle adresse (0000101111111111 en binaire) est décodée par le décodeur d'adresses et sélectionne la cellule mémoire 3071. Le contenu de celle-ci, adresse 96 (01100000 en binaire), est mis sur le bus de données et chargé, cette fois dans l'accumulateur de l'UC. Après quoi, le bus de données est réglé à 1056 et l'UC s'attend à y trouver une autre instruction.

Maintenant que l'UC a un élément de données dans son accumulateur, quel type d'instruction peut-on désormais prévoir pour l'UC? A vrai dire cela pourrait être n'importe quelle instruction. Une UC peut ainsi recevoir de 10 à 100 instructions selon son type. Supposons à présent que nous désirions inverser les données dans l'accumulateur (inverser signifie changer les 1 en 0 et les 0 en 1) et que l'instruction correspondant à cette intervention se trouve à l'adresse 1056. Dans notre UC, le code de cette instruction est 84. Lorsque ce nombre est reçu par l'UC, les données présentes dans l'accumulateur sont inversées. Le nombre qui est dans l'accumulateur est 96 (01100000). L'instruction d'inversion le changera en 10011111 en binaire. Puisque l'instruction d'inversion est une instruction d'un octet, l'UC sait déjà que le contenu de la prochaine adresse de mémoire, 1057, contiendra de nouveau une instruction et non des données.

Cette façon d'adresser un emplacement mémoire pour extraire un élément de données est l'une des méthodes offertes au programmeur. Les codes spécifiques que nous avons utilisés dans notre exemple (58 pour charger l'accumulateur et 84 pour inverser le contenu de l'accumulateur) correspondent aux instructions de notre UC. Le même principe s'applique à tous les autres microprocesseurs. Mais des codes différents sont utilisés pour les diverses instructions et chaque marque de microprocesseur possède son propre « jeu d'instructions ».

Les emplacements d'E/S (entrées-sorties) doivent eux aussi avoir des adresses uniques et les principes d'adressage sont les mêmes. Les microprocesseurs 8 bits ne possèdent généralement que 8 lignes d'adresses pour les E/S. Le nombre maximal d'adresses d'E/S est donc de 256. Ce qui est amplement suffisant pour la plupart des applications des micro-ordinateurs. Le décodage d'adresse est toujours nécessaire de façon que le dispositif sélectionné par l'UC (que ce soit une adresse de mémoire ou d'E/S) soit sollicité exclusivement. On nomme ce processus « validation ». Lorsqu'il y a peu de lignes à décoder, il est possible d'utiliser de simples portes logiques pour effectuer le décodage. Le principe d'un décodeur de lignes deux-quatre est illustré ci-dessus. Ce type de décodage simple est utilisé fréquemment pour sélectionner des dispositifs d'E/S. Cependant, la complexité du circuit de décodage est proportionnelle au nombre de lignes d'adresse. Lorsque 65 536 adresses de mémoire doivent être sélectionnées individuellement, presque tout le décodage est généralement effectué à l'intérieur des puces mémoire.



Décodage d'adresse
Les 16 lignes composant le bus d'adresse peuvent désigner une adresse parmi 65 636 adresses de mémoire distinctes. La combinaison de 1 et de 0 sur le bus d'adresse est décodée dans le décodeur d'adresses. Une partie du décodage est effectuée par des décodeurs d'adresses composés de simples portes logiques montées sur le circuit, mais la majeure partie du décodage est effectuée par des circuits équivalents situés à l'intérieur des puces mémoire. L'illustration montre comment deux lignes d'adresse peuvent être décodées pour sélectionner une et une seule puce parmi quatre. (Cl. Liz Dixon.)

Les fonctions intégrées

Le BASIC comporte des fonctions intégrées qui définissent certaines des opérations sans avoir à les programmer. Bien les connaître et savoir les utiliser améliorera votre programmation.

Supposons qu'un de vos programmes doive calculer la racine carrée d'un nombre. Il y a plusieurs façons de le faire. La manière la plus maladroite serait de créer une table de valeurs de racines carrées et de vous en servir pour obtenir la valeur correspondant à un nombre donné. Il vaut mieux utiliser la « fonction » racine carrée, intégrée dans la plupart des versions du BASIC. Cette fonction se charge du problème mathématique sans que le programmeur ait à s'en préoccuper. Examinons cet exemple :

```
10 REM CE PROGRAMME TROUVE LA RACINE
20 REM CARRÉE D'UN NOMBRE
30 PRINT « ENTREZ LE NOMBRE DONT VOUS »
40 PRINT « DÉSIREZ LA RACINE CARRÉE »
50 INPUT N
60 LET A = SQR (N)
70 PRINT « LA RACINE CARRÉE DE »; N; « EST »; A
80 END
```

Tapez ce court programme. Vous constaterez qu'il donne en fait la racine carrée de tout nombre entré. Voyons les règles d'utilisation de cette fonction.

Une « fonction » est généralement un mot de commande (SQR ici) suivi de l'expression à traiter placée entre parenthèses. Dans ce programme, N est le nombre entré au clavier. C'est le nombre dont nous désirons trouver la racine carrée. La ligne 60 attribue la racine carrée du nombre N à la variable A. La ligne 70 imprime la valeur de A.

L'expression entre parenthèses est appelée l'« argument » de la fonction, qui ne doit pas nécessairement être une variable, mais peut être un nombre. Tapez ce programme et voyez ce que donne son exécution :

```
10 PRINT SQR(25)
20 END
```

Vous constatez qu'il fonctionne tout aussi bien. Vous pouvez aussi introduire des arguments plus complexes :

```
10 LET A = 10
20 LET B = 90
30 LET C = SQR(A + B)
40 PRINT C
50 END
```

Ce court programme peut être simplifié en combinant ainsi les lignes 30 et 40 :

```
10 LET A = 10
20 LET B = 90
30 PRINT SQR(A + B)
40 END
```

Les fonctions sont en fait de courts programmes BASIC que le programmeur peut utiliser en tout temps. Il est important de bien les mémoriser et les comprendre afin de les employer à bon escient. La plupart des versions de BASIC ont de nombreuses fonctions et offrent au programmeur la possibilité d'en définir de nouvelles. Vous découvrirez plus loin comment le faire. Mais voyons d'abord quelques autres fonctions généralement intégrées. Il existe deux catégories de fonctions : les fonctions numériques, où l'argument (l'expression entre parenthèses) est un nombre ou une expression numérique, et les fonctions de chaînes, où l'argument est une chaîne de caractères ou une expression composée de chaînes de caractères. Voyons d'abord quelques fonctions numériques.

Nous avons précédemment écrit un programme qui calculait le nombre de carreaux nécessaires dans une pièce. Il y avait un petit problème dans ce programme : la réponse pouvait impliquer des fractions de carreau. La réponse pouvait être 988,24. Dans un tel cas, il est souhaitable d'arrondir au nombre entier le plus proche. L'une des fonctions du BASIC est précisément d'effectuer cette opération. Voyons comment fonctionne celle-ci :

```
10 PRINT « ENTREZ UN NOMBRE AVEC FRACTION
  DÉCIMALE »
20 INPUT N
30 PRINT « L'ÉQUIVALENT ENTIER DU NOMBRE EST »;
40 PRINT INT(N)
50 END
```

Si le nombre entré est 3,14, le programme affichera :

```
L'ÉQUIVALENT ENTIER DU NOMBRE EST 3
```

Dans le cas de nos carreaux, nous devrons naturellement ajouter 1 à la réponse, afin de garantir l'achat d'un nombre suffisant de carreaux.

Il peut être nécessaire de connaître le signe d'un nombre afin de savoir s'il est négatif, égal à zéro ou positif. Pour ce faire, la plupart des BASIC comprennent une fonction SGN. Voici un exemple :

```
10 PRINT « ENTREZ UN NOMBRE »
20 INPUT N
30 LET S = SGN(N)
40 IF S = 1 THEN GOTO 100
```

```
50 IF S = 0 THEN GOTO 120
60 IF S = 1 THEN GOTO 140
100 PRINT « LE NOMBRE EST NÉGATIF »
110 GOTO 999
120 PRINT « LE NOMBRE EST ZÉRO »
130 GOTO 999
140 PRINT « LE NOMBRE EST POSITIF »
150 GOTO 999
999 END
```

Il n'existe que trois valeurs possibles retournées par la fonction SGN : − 1 si l'argument est négatif, 0 si l'argument est zéro et + 1 si l'argument est un nombre positif. L'utilisation de la fonction SGN à la ligne 30 épargne plusieurs lignes de programmation. Nous aurions pu en effet procéder de cette façon :

```
IF N<0 THEN LET S = - 1
IF N = 0 THEN LET S = 0
IF N>0 THEN LET S = 1
```

L'intervention d'une fonction BASIC peut toujours être définie par une programmation normale; l'emploi d'une fonction ne fait que vous faire gagner du temps, et économiser l'espace mémoire.

Voici quelques autres fonctions numériques : ABS retourne la valeur absolue d'un nombre. La valeur absolue est en fait sa valeur réelle sans signe. La valeur absolue de − 6 est donc 6. Essayez ce programme :

```
10 LET X = - 9
20 LET Y = ABS(X)
30 PRINT Y
40 END
```

MAX trouve la valeur maximale parmi deux nombres :

```
10 LET X = 9
20 LET Y = 7
30 LET Z = X MAX Y
40 PRINT Z
50 END
```

MIN est semblable mais trouve le plus petit nombre parmi deux nombres :

```
10 PRINT « ENTREZ UN NOMBRE »
20 INPUT X
30 PRINT « ENTREZ UN AUTRE NOMBRE »
40 INPUT Y
50 LET Z = X MIN Y
60 PRINT Z
70 END
```

Notez que ces deux dernières fonctions reçoivent deux arguments et que ceux-ci n'ont pas à être placés entre parenthèses. La plupart des fonctions BASIC comportent de nombreuses autres fonctions numériques, comme LOG pour trouver le logarithme d'un nombre, TAN pour trouver la tangente, COS pour trouver le cosinus et SIN pour le sinus. Nous étudierons plus tard l'utilisation de ces fonctions trigonométriques.

Ces programmes ont une grande importance car ils permettent de faciliter l'écriture de programmes plus vastes. En effet, rien de plus fastidieux que de refaire un programme qui est déjà disponible dans le langage BASIC intégré de l'ordinateur.

Le BASIC comporte également des fonctions agissant sur les chaînes de caractères. Nous en avons utilisé certaines dans notre programme de tri (voir page 135), mais nous n'avions pas alors analysé leur fonctionnement. Examinons maintenant ces fonctions ainsi que d'autres.

Une des fonctions de chaînes les plus utiles est LEN. Elle compte le nombre de caractères qui composent une chaîne placée entre guillemets ou le nombre de caractères affectés à une variable de chaîne. Voici un exemple :

```
10 LET A$ = « ORDINATEUR »
20 LET N = LEN(A$)
30 PRINT « LE NOMBRE DE CARACTÈRES DANS LA CHAINE
  EST »; N
40 END
```

Pourquoi peut-il être nécessaire de connaître le nombre de caractères dans une variable de chaîne? Pour le constater, entrez et exécutez ce court programme construisant un « mot triangulaire ». Il imprime d'abord la première lettre du mot, puis les première et deuxième, et ainsi de suite jusqu'à ce que le mot soit imprimé au complet.

```
5 REM IMPRIME UN MOT TRIANGULAIRE
10 LET A$ = « GILET »
20 FOR L = 1 TO 5
30 LET B$ = LEFT$(A$,L)
40 PRINT B$
50 NEXT L
60 END
```

Maintenant, exécutez ce programme. L'affichage devrait être :

```
G
GI
GIL
GILE
GILET
```

Ce programme utilise la fonction LEFT$ pour extraire des caractères dans une chaîne. LEFT$ reçoit deux arguments. Le premier spécifie la chaîne et le second (suivant une virgule) spécifie le nombre de caractères à extraire à partir de la gauche de la chaîne. La chaîne GILET a été affectée à A$, LEFT$(A$1) retournera donc la lettre G, LEFT$(A$,2) retournera GI. Ce programme utilise l'indice L, qui va de 1 à 5, le second argument de la fonction LEFT$ passe donc de 1 à 5. Nous connaissions le nombre de caractères que comptait le mot à imprimer, il était donc facile de choisir 5 comme limite supérieure de la boucle FOR-NEXT. Mais que faire si nous ne connaissons pas le nombre de caractères du mot?

C'est ici que nous faisons appel à la fonction LEN. LEN reçoit une chaîne (entre guillemets) ou une variable de chaîne comme argument. Voici quelques exemples illustrant son fonctionnement :

```
10 REM PROGRAMME POUR TESTER LA FONCTION LEN
20 PRINT LEN (« ORDINATEUR »)
30 END
```

Le programme devrait imprimer 10. Il a compté le nombre de caractères du mot ORDINATEUR et retourné cette valeur. Faisons la même chose d'une manière légèrement différente :

```
10 REM TROUVER LA LONGUEUR D'UNE CHAINE
20 LET A$ = « ORDINATEUR DOMESTIQUE »
30 LET L = LEN(A$)
40 PRINT L
50 END
```

Lors de l'exécution de ce programme, l'ordinateur affichera 21. Il y a vingt et un caractères dans cette chaîne et non vingt, n'oubliez pas que l'espace est aussi un caractère.

Maintenant, utilisons la fonction LEN pour modifier notre programme d'impression d'un mot triangulaire.

```
10 REM CE PROGRAMME IMPRIME UN MOT TRIANGULAIRE
20 PRINT « TAPEZ UN NOMBRE »
30 INPUT A$
40 LET N = LEN(A$)
50 FOR L = 1 TO N
60 LET B$ =LEFT$(A$,L)
70 PRINT B$
80 NEXT L
90 END
```

Lors de chaque boucle, la valeur de L est incrémentée de 1 et cela jusqu'à N (la longueur de la chaîne). Si vous entrez le mot MARTEAU, la ligne 40 équivaut à LEFT N = LEN(« MARTEAU »), N sera donc égal à 7. Lors de la première boucle, la ligne 50 définit L égal à 1 et la ligne 60 équivaut à LET B$ = LEFT$(MARTEAU,1), un caractère de la chaîne à partir de la gauche est donc affecté à B$. Ce caractère est M.

Lors de la seconde boucle, L est égal à 2 et la ligne 60 équivaut à LET B$ = LEFT$(MARTEAU,2). Cela extrait les deux premiers caractères de la chaîne et les affecte à B$. B$ contient donc MA.

La fonction LEN a calculé qu'il y avait sept caractères dans la chaîne MARTEAU et a affecté cette valeur à la variable N; ainsi, lors de la dernière boucle, tous les caractères sont affectés à la variable B$ et la chaîne est affichée.

RIGHT$, tout comme LEFT$, extrait des caractères d'une chaîne, mais cette fois à partir de la droite.

Voyons finalement une autre fonction de chaîne utilisée dans notre programme de tri. C'est la fonction INSTR. Elle sert à trouver l'emplacement de la première apparition de la chaîne spécifiée (sous-chaîne) à l'intérieur d'une chaîne. INSTR servit dans notre programme de tri à localiser l'espace entre le prénom et le nom. Voici un exemple :

```
10 LET A$ = « INFORMATIQUE »
20 LET P = INSTR5(A$,« TIQUE »)
30 PRINT P
40 END
```

Avant d'entrer et d'exécuter ce programme, essayez de prévoir quelle valeur sera affichée pour P. Rappelons que INSTR trouve la position du premier caractère de la première apparition d'une « sous-chaîne » à l'intérieur d'une chaîne. Si la chaîne est INFORMATIQUE, la position du début de la sous-chaîne TIQUE est 8 (le T de TIQUE est la huitième lettre de INFORMATIQUE). Certains BASIC n'utilisent pas INSTR, mais offrent une fonction similaire nommée INDEX. Voici comment localiser un espace à l'intérieur d'une chaîne à l'aide de INSTR (ou de INDEX) :

```
10 REM TROUVER LA POSITION D'UN ESPACE DANS UNE
   CHAINE
20 LET A$ = « ORDINATEUR PERSONNEL »
30 LET P = INSTR (A$, « »)
40 PRINT P
50 END
```

Notez que le second argument de la fonction INSTR (ligne 30) est « ». Un espace est placé entre guillemets, c'est le caractère à rechercher. Le programme affichera le nombre 11 puisque l'espace est à la onzième position. Qu'est-ce qui serait imprimé si la ligne 30 était modifiée comme ceci?

```
LET P = INSTR(A$, « P »)
```

Voici enfin une autre fonction très pratique pour tous ceux qui souhaitent faire apparaître des textes à l'écran avec un minimum de présentation : PRINT. En effet, rien de plus désagréable et de si peu engageant qu'un texte dont les marges gauche et droite se confondent avec les bords respectifs de l'écran. Exécutez ce programme :

```
10 PRINT « CETTE LIGNE EST A LA MARGE »
20 PRINT TAB(5) « CETTE LIGNE N'EST PAS A LA MARGE »
30 END
```

Voyez-vous ce qui s'est passé? La deuxième ligne est affichée à cinq positions de la marge gauche. TAB est comparable à la touche TAB d'une machine à écrire. Voici un autre exemple de la fonction TAB :

```
10 REM UTILISATION DE LA FONCTION TAB
20 PRINT « ENTREZ LA TABULATION »
30 INPUT T
40 LET W$ = « TABULATION »
50 PRINT TAB(T); W$
60 END
```

Vous pouvez revenir à la page 136 pour voir comment nous avons employé ces fonctions dans le programme de tri de noms.

Exercices

• ***Boucles 1.*** Qu'est-ce qui sera affiché lors de l'exécution de ce programme?

```
10 LET A = 500
20 FOR L = 1 TO 50
30 LET A = A - 1
40 NEXT L
50 PRINT « LA VALEUR DE A EST »; A
```

• ***Boucles 2.*** Quel sera l'affichage lors de l'exécution de ce programme?

```
10 REM
20 REM BOUCLE DE SYNCHRONISATION
30 REM VOYONS LE TEMPS REQUIS
40 REM
```

Variantes de basic

TAB

Remplacez par nombre TAB sur le Spectrum.

LEFT$

RIGHT$

Aucune de ces commandes n'est disponible sur le Spectrum, mais vous pouvez créer vos propres versions avec l'instruction DEF FN ; ainsi, vous devez insérer ces lignes dans votre programme :

```
9900 DEF FN L$(X$,N) = X$(TO N)
9910 DEF FN R$(X$,N) = X$ (LEN
   (X$) – N + 1 TO)
9920 DEF FN M$(X$,P,N) = X$(P TO
   P + N – 1)
```

MID$

Maintenant :

```
FN L$(X$,N) remplace LEFT$(X$,N)
FN M$(X$,P,N) remplace MID$(X$,P,N)
FN R$(X$,N) remplace RIGHT$(X$,N)
```

INSTR

Cette fonction n'est pas disponible sur le Spectrum, sur le Vic-20, sur le Commodore 64 et sur l'Oric-1, mais vous pouvez écrire une routine de remplacement. Supposons que la ligne initiale soit :

```
20 LET P = INSTR(A$, « TIQUE »)
```

Remplacez-la par :

```
20 LET X$ = A$ : LET Z$ = « TIQUE » :
   GOSUB 9930 P = U
```

et ajoutez ces lignes :

```
9929 STOP
9930 LET U = 0 : LET X = LEN(X$) :
   LET V = LEN (Z$)
9940 FOR W = 1 TO L – V + 1 : IF
   MID$(X$,W,V) = Z$ THEN
   LET U = W
9950 IF U < > O THEN
   LET W = L – V + 1
9960 NEXT W : RETURN
```

```
50 PRINT « C'EST PARTI »
60 FOR X = 1 TO 5000
70 NEXT X
80 PRINT « STOP »
90 END
```

• ***Boucles 3.*** Si vous entrez le nombre 60, quel résultat sera affiché par ce programme ?

```
10 PRINT « ENTREZ UN NOMBRE »
20 INPUT N
30 LET A = 100
40 FOR L = 1 TO N
50 LET A = A + 1
60 NEXT L
70 PRINT « A VAUT MAINTENANT »; A
80 END
```

• ***Boucles 4.*** Que se passe-t-il lors de l'exécution de ce programme ?

```
10 PRINT « J'AIME LE BASIC »
20 GOTO 10
30 END
```

• ***Boucles 5.*** Qu'est-ce qui sera affiché lors de l'exécution de ce programme ?

```
10 FOR Q = 1 TO 15
20 PRINT « SANS VOULOIR ME RÉPÉTER »
30 NEXT Q
40 END
```

• ***Read-data 1.*** Quel résultat sera affiché ?

```
10 READ X
20 READ Y
30 READ Z
40 PRINT « NOUS TESTONS L'INSTRUCTION READ »
50 DATA 50, 100, 20
60 PRINT X + Y + Z
```

• ***Read-data 2.*** Quel sera l'affichage de ce programme ?

```
100 FOR L = 1 TO 10
110 READ X
120 PRINT « X = »; X
130 NEXT L
140 DATA 1, 3, 5, 7, 11, 13, 17, 19, 23
```

Les réponses seront données dans une prochaine leçon.

Réponses des exercices des pages 136 et 137

Variables

(A) (B6) ~~Z7~~ D$ ~~16~~ X$ (A12) (D9) (Q81) (Q5) ~~0E~~ H$

Arithmétique 1

```
10 LET B = 6
20 PRINT B
```

Arithmétique 2

```
10 LET A = 5
20 LET B = 7
30 LET C = 9
40 LET D = A + B + C
50 PRINT D
```

Arithmétique 3

17

Arithmétique 4

25
25

Comparaisons 1

5

Comparaisons 2

601 (avec des entiers)

Comparaisons 3

10000

Impression 1

```
PRINT « LA VALEUR DE T EST »; T
```

Impression 2

```
640 PRINT « DÉSOLÉ, VOTRE POINTAGE »; S; « EST
TROP BAS »
```

Impression 3

Cela était une erreur volontaire.
Le point-virgule à la fin de la ligne causera une erreur lors de l'exécution. Le programme doit se présenter comme ceci :

```
200 LET A$ = « LE COURS DE MICRO-
   INFORMATIQUE ? »
210 LET B$ = « AIMEZ-VOUS »
220 PRINT B$;A$
```

Et le résultat devrait être le suivant :

AIMEZ-VOUS LE COURS DE MICRO-INFORMATIQUE ?

Entrée 1

6

Entrée 2

TAPEZ VOTRE NOM SVP
BONJOUR (VOTRE NOM) ICI VOTRE ORDINATEUR

Les réponses de « Variables » seront différentes sur certaines machines qui ne permettent pas autre chose qu'une seule lettre (aucun suffixe numérique).

Jupiter Ace

Le seul ordinateur domestique peu coûteux offrant le langage FORTH comme langage de programmation standard; un défi pour les programmeurs ambitieux qui souhaitent dépasser le BASIC.

Le Jupiter Ace est une machine fantastique et l'un des rares ordinateurs à ne pas avoir choisi le BASIC comme langage intégré. Ses concepteurs : des ingénieurs de Sinclair soucieux de produire un ordinateur qu'ils aimeraient utiliser.

Sa principale caractéristique est donc d'utiliser le langage FORTH. Et il consiste sans nul doute, pour tous ceux désireux de travailler sur ce langage de programmation, une solution relativement bon marché. Il est en fait plus avantageux d'acheter l'Ace plutôt que d'installer le langage sur son ordinateur. Le langage FORTH est par ailleurs très bien présenté dans la documentation livrée avec l'ordinateur.

Langage forth

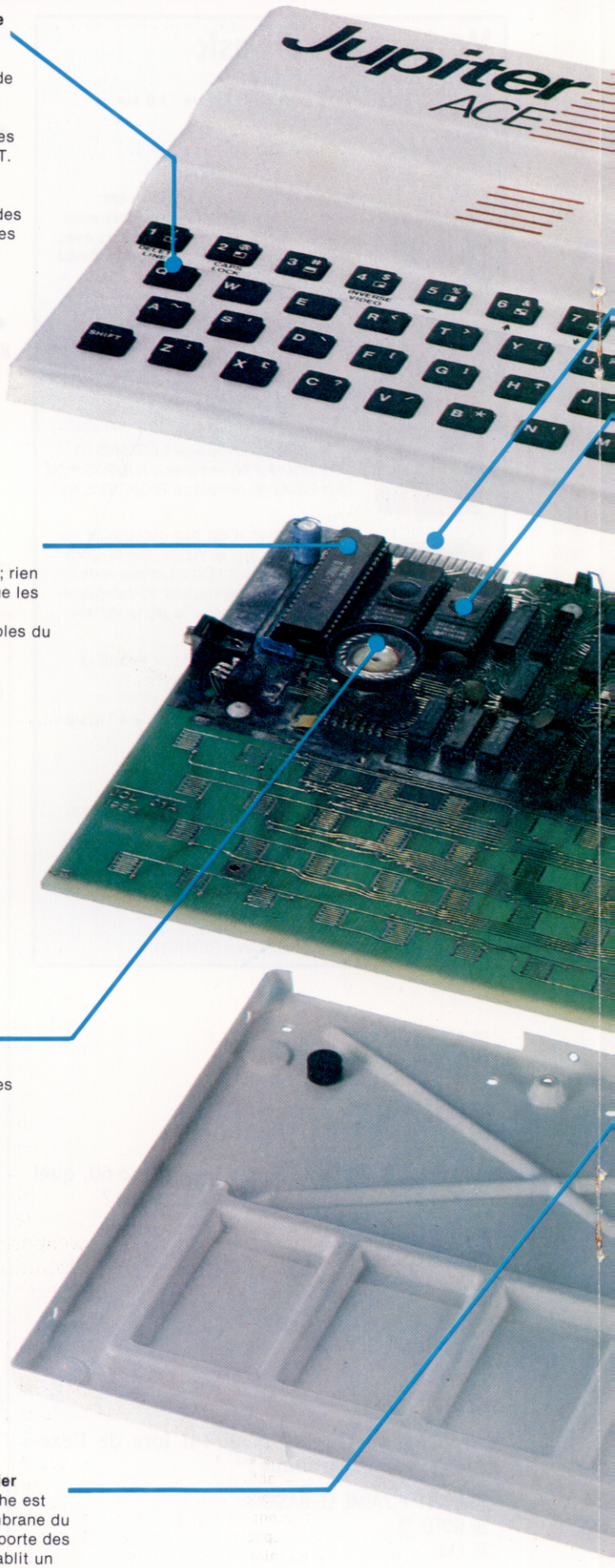
Version BASIC

```
100 REM UN PROGRAMME BASIC POUR IMPRIMER
  « SHASAM ! »
110 FOR X = 1 TO 6
120 PRINT « SHASAM ! »
130 NEXT X
140 END
    RUN
```

Version forth

```
:SHOUT. « SHASAM ! »;
:CHORUS 6 0 DO SHOUT  LOOP;
 CHORUS
```

Ces deux programmes font exactement la même chose, mais la version BASIC semble un peu moins compliquée !
Le FORTH offre un dictionnaire de mots de commande et permet la création de nouveaux mots de commande. Dans notre programme nous avons ajouté deux nouveaux mots au dictionnaire : SHOUT désigne une chaîne à imprimer et CHORUS est un mélange de mots « primitifs » (mots prédéfinis dans le dictionnaire) et du nouveau mot SHOUT.
Le FORTH a aussi une mémoire (nommée pile) et peut y traiter des nombres. Le programme FORTH effectue les mêmes opérations arithmétiques et logiques que le BASIC, mais en manipulant la pile au lieu de passer par des expressions algébriques.
Le FORTH est un langage un peu exaspérant (genre Rubik cube); c'est aussi un langage très puissant et une nouvelle façon de penser. Certains programmeurs l'adorent, d'autres le détestent.
Le FORTH doit toute sa puissance au fait de pouvoir définir et utiliser de nouvelles commandes.
L'utilisateur peut ainsi configurer son langage de programmation à la tâche à accomplir. Le FORTH est particulièrement indiqué pour programmer des robots domestiques.

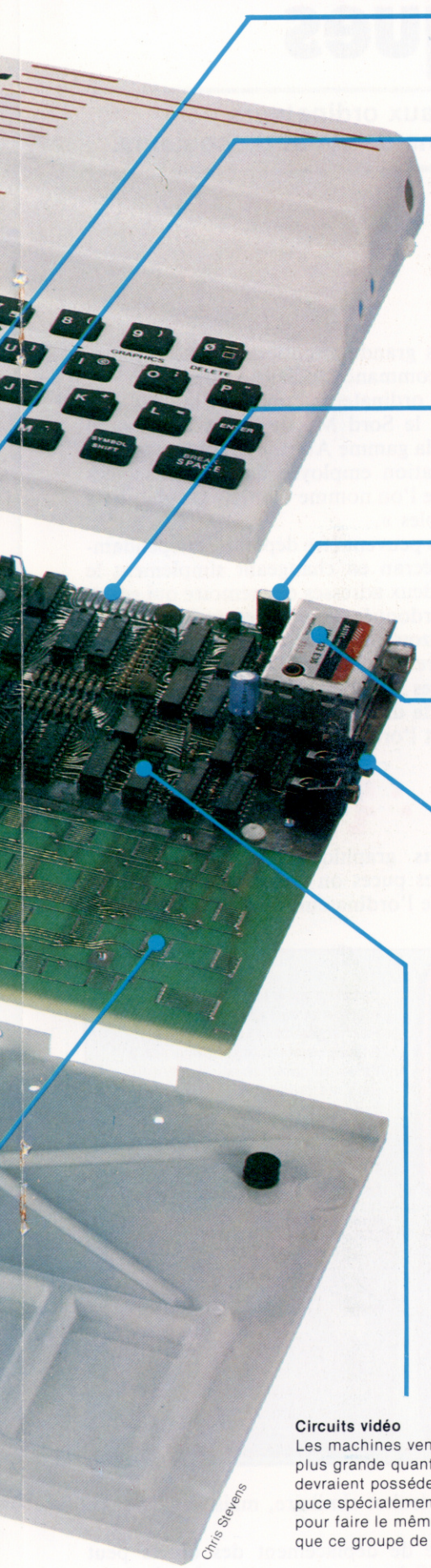


Le clavier de Jupiter Ace
Ce clavier à membrane ressemble à celui du Spectrum. Les touches de la rangée supérieure ont trois fonctions qui sont sollicitées par les touches SHIFT et SYMBOL SHIFT. En plus, sept symboles graphiques peuvent être utilisés pour construire des diagrammes et graphiques simples.

Microprocesseur
Le Z80A a été utilisé; rien de surprenant puisque les concepteurs sont également responsables du Sinclair Spectrum.

Haut-parleur
Dispositif à semi-conducteur qui produit des bips semblables à ceux d'une montre numérique, mais qui peut être utilisé pour produire des sons simples.

Circuits du clavier
Quand une touche est pressée, la membrane du clavier, qui comporte des conducteurs, établit un contact entre deux pistes métalliques du circuit imprimé.



Port d'extension de la mémoire
Muni d'un adapteur adéquat, l'Ace peut aussi utiliser les modules d'extension du ZX81.

Langage forth
Deux EPROM de 4 K contiennent le langage. Les ROM doivent être commandées par groupes de mille, les EPROM permettent donc une production en plus petite série. La partie supérieure de chaque EPROM doit être protégée, sinon tout ultraviolet effacerait son contenu.

Port utilisateur
Pour une imprimante ou un autre périphérique.

Horloge
Pilote le microprocesseur à 1 MHz.

Modulateur RF
Fournit une sortie noir et blanc pour téléviseur.

Port cassette
Un des systèmes les plus fiables sur le marché.

Circuits vidéo
Les machines vendues en plus grande quantité devraient posséder une puce spécialement conçue pour faire le même travail que ce groupe de puces.

Chris Stevens

L'apparence extérieure de l'Ace est assez proche de celle du Sinclair ZX80. Il possède un petit dispositif à circuits intégrés produisant toute une gamme de bips, mais qui peut également générer des sons complexes. Tout comme les ordinateurs Sinclair, l'Ace possède un connecteur d'alimentation, une prise d'antenne de télévision et deux autres connecteurs destinés à des éléments d'équipement spéciaux.

Le support cassette est muni d'une prise à deux plots et est l'une des plus faibles interfaces cassette sur le marché. L'affichage noir et blanc comporte 32 caractères par ligne, et offre une résolution graphique de 64 par 48. Chacun des caractères peut être défini par l'utilisateur pour créer des symboles mathématiques ou des silhouettes de jeux.

Utilisant le FORTH, le Jupiter Ace dispose déjà de nombreuses applications. Divers dispositifs complémentaires sont également disponibles. Les programmes FORTH étant généralement assez courts, des applications complexes peuvent donc être logées dans ses 3 K de mémoire. Pour écrire des programmes plus longs, il est nécessaire d'avoir recours à plus de mémoire : des modules de 16 ou de 48 K sont prévus à cet effet. Les modules RAM du ZX81 peuvent être utilisés avec un adaptateur.

JUPITER ACE

PRIX
* *

DIMENSIONS
215 × 190 × 30 mm.

POIDS
246 g.

HORLOGE
1 MHz.

MÉMOIRE
3 K RAM, extensions externes possibles jusqu'à 51 K; 8 K ROM.

AFFICHAGE VIDÉO
Noir et blanc, 32 × 22 lignes de texte, graphiques 64 × 48.

INTERFACES
Connecteur pour téléviseur, pour cassette, prise d'alimentation, deux connecteurs plats; le premier possède toutes les lignes de données et d'adresse du processeur, le second dispose des lignes de données et de certaines lignes de sélection.

LANGAGE INTÉGRÉ
FORTH.

AUTRE LANGAGE OFFERT
Aucun.

ACCESSOIRES FOURNIS
Bloc d'alimentation, prises de cassette et de téléviseur.

CLAVIER
Clavier à membrane semblable à celui du Spectrum Sinclair. Les touches doivent être pressées en plein centre. Toutes les touches sont à répétition et deux modes « shift » permettent la définition de tous les codes ASCII.

DOCUMENTATION
De loin le meilleur manuel pour cette catégorie de machines, il pourrait servir d'exemple à de nombreux constructeurs de plus gros ordinateurs. L'auteur a écrit les manuels du ZX81 et du Spectrum. On trouve à la page 180 une introduction au langage FORTH et une description complète de l'Ace, toutes deux accompagnées de nombreux exemples.

Objets graphiques

Des mémoires à grande capacité permettent aux ordinateurs domestiques de produire des graphiques pleins de vie et de couleurs.

La possibilité de produire des graphiques et des animations constitue un des principaux attraits des ordinateurs domestiques. Sur la plupart des micro-ordinateurs, l'utilisateur peut tracer des points individuels, dessiner des lignes et des cercles et modifier les couleurs d'arrière-plan et de premier plan.

Dans les jeux à action rapide et dans les simulations, le mouvement doit être simulé. La façon la plus simple est de produire une série d'images immobiles les unes après les autres. Cela doit être fait assez rapidement pour créer l'illusion du mouvement. Les images télévisées sont produites ainsi.

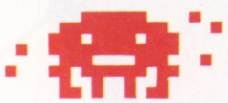
Vitesse de l'action

L'illusion du mouvement peut être créée d'une autre façon : un personnage est affiché, effacé et affiché de nouveau avec un léger décalage. Afin d'obtenir un mouvement uniforme, ce décalage doit être minimal, la période affichage-suppression doit être la plus courte possible.

tence et très grande attention, chaque personnage étant commandé individuellement.

Plusieurs ordinateurs, notamment le Commodore 64, le Sord M5, le Texas Instrument TI99/4A et la gamme Atari, offrent une technique d'animation employée dans les machines d'arcade que l'on nomme « objets graphiques à plans multiples ».

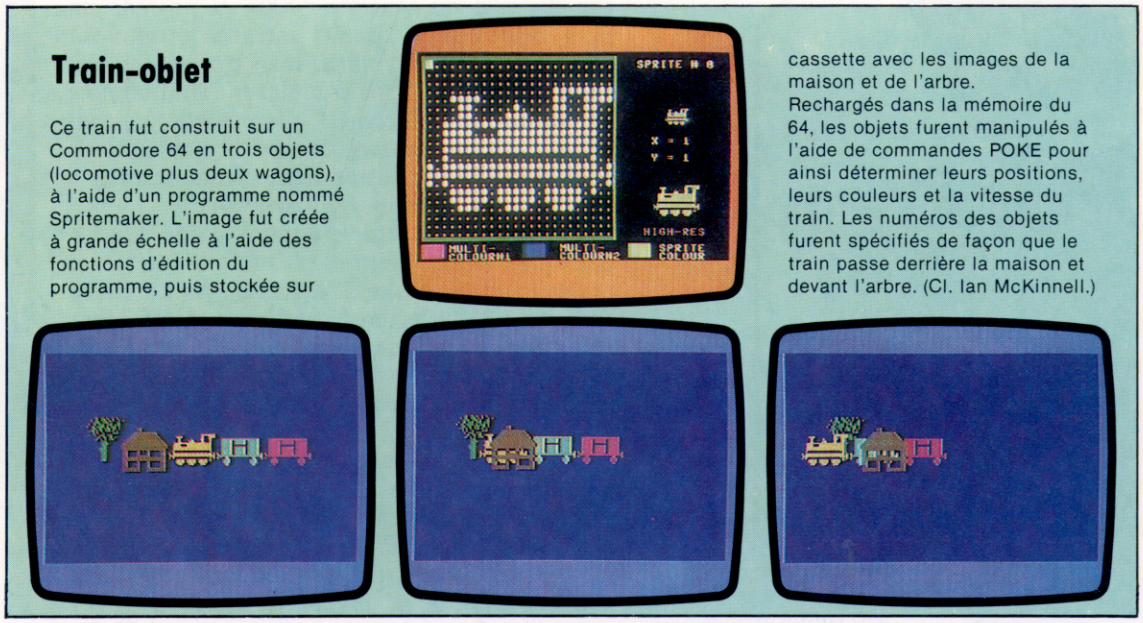
Les objets peuvent être déplacés indépendamment sur l'écran en changeant simplement le contenu de deux adresses de mémoire qui spécifient les coordonnées X et Y (les positions verticales et horizontales). Par exemple, X peut être compris entre 0 et 255, et Y entre 0 et 191. Certains systèmes vous permettent même de définir la vitesse et la direction du mouvement de chaque objet, et l'ordinateur fait le reste.



Les objets graphiques sont généralement gérés par des puces ou des circuits spéciaux à l'intérieur de l'ordinateur. On peut obtenir par

Train-objet

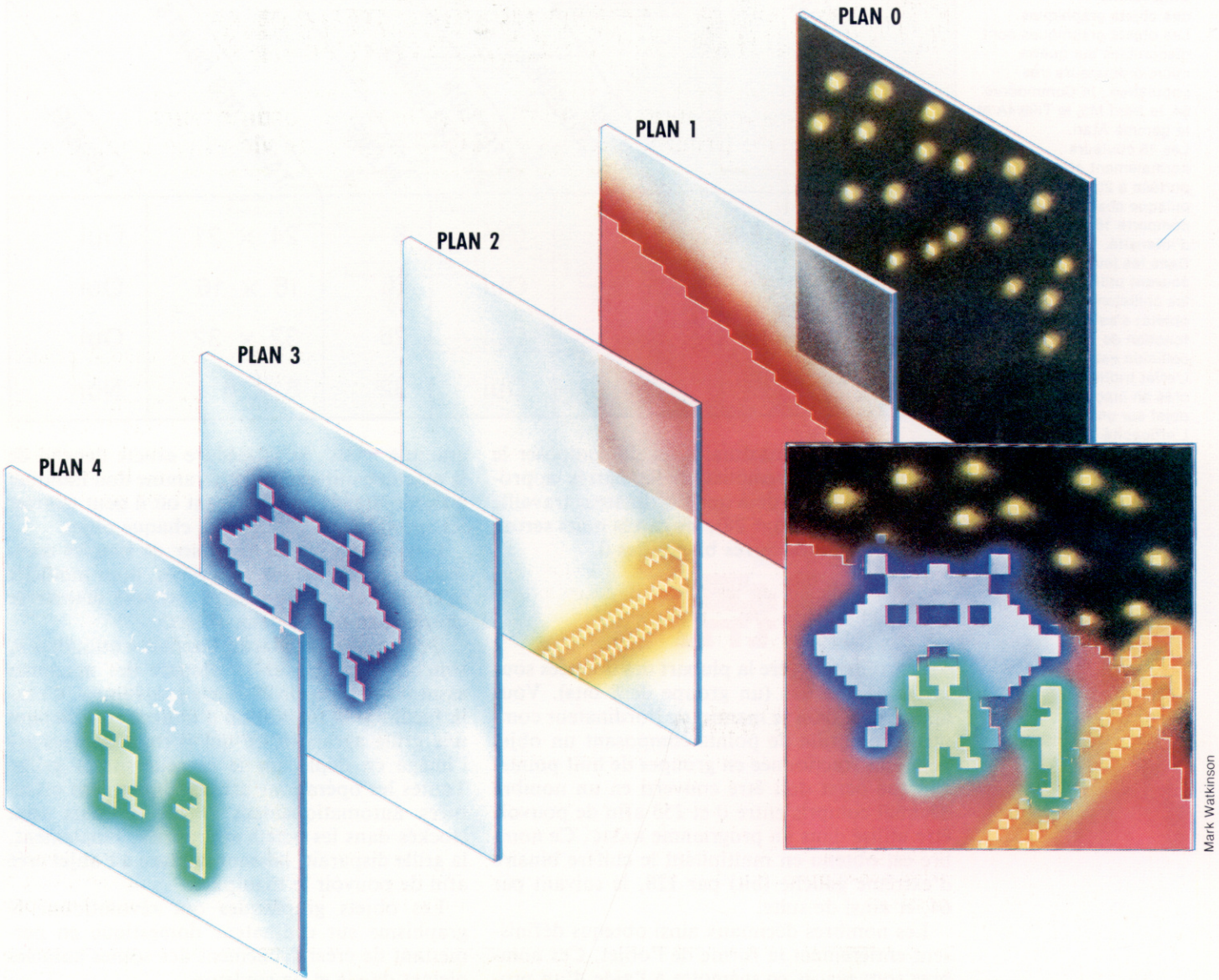
Ce train fut construit sur un Commodore 64 en trois objets (locomotive plus deux wagons), à l'aide d'un programme nommé Spritemaker. L'image fut créée à grande échelle à l'aide des fonctions d'édition du programme, puis stockée sur cassette avec les images de la maison et de l'arbre. Rechargés dans la mémoire du 64, les objets furent manipulés à l'aide de commandes POKE pour ainsi déterminer leurs positions, leurs couleurs et la vitesse du train. Les numéros des objets furent spécifiés de façon que le train passe derrière la maison et devant l'arbre. (Cl. Ian McKinnell.)



Avec le BASIC, on ne peut créer que des animations assez lentes. Ce problème peut être résolu en programmant en langage machine, mais ce langage exige du programmeur compé-

logiciel un résultat similaire, mais jamais aussi satisfaisant.

La vitesse de déplacement des objets peut varier. L'affichage est constitué de plusieurs



plans superposés de façon que l'œil interprète le tout comme un affichage unique.

Un effet tridimensionnel peut être obtenu en faisant passer un objet devant ou derrière d'autres objets. Les objets sont numérotés de 0 au nombre maximal (32 sur le Sord M5). Si deux objets chevauchent, l'objet portant le numéro inférieur est affiché. En ordonnant soigneusement vos objets, il est donc possible d'obtenir un effet tridimensionnel : un train passant devant un arbre le cache pendant son passage. Le train peut lui-même être caché en passant derrière une maison ayant un numéro inférieur.

Grâce à la gamme de couleurs offerte par votre ordinateur, chaque objet peut être coloré individuellement. La taille d'un objet peut être modifiée en changeant le contenu d'une adresse mémoire.

Les objets graphiques sont évidemment surtout utilisés dans les programmes de jeux, et se prêtent très bien à la « détection de collision ». Lorsqu'il y a chevauchement (par exemple lorsque vos missiles touchent la cible ennemie), le système peut être programmé pour passer à une autre partie du programme qui crée une explosion et qui incrémente le pointage du joueur.

Avant d'utiliser des objets graphiques, vous devez les créer, et le processus ressemble à la conception d'un nouveau caractère. Les lettres, chiffres et symboles graphiques sont stockés à l'intérieur de l'ordinateur dans une puce que l'on nomme « générateur de caractères ».

Comme nous l'avons déjà mentionné, les caractères sont généralement construits sur une matrice de huit points par huit. Les dimensions maximales d'un plan-objet varient d'une machine à l'autre, mais ils ont généralement plusieurs caractères de longueur et de largeur. Sur le Commodore 64, le maximum est de vingt-quatre points par vingt et un.

La meilleure façon de construire un objet graphique est de dessiner une grille correspondant

Sur des plans différents
A l'aide d'objets graphiques, une image complexe peut être construite en plaçant chaque composante sur des plans différents qui sont superposés les uns sur les autres pour créer l'affichage. Le principal avantage de ce système est que des objets situés sur des plans différents peuvent se déplacer de façon entièrement indépendante. Le programmeur affecte aux plans un ordre de priorité, de façon que lorsque deux objets chevauchent, l'objet le plus prioritaire cache l'autre objet, ce qui crée un effet tridimensionnel. La plupart des ordinateurs qui utilisent des objets graphiques peuvent signaler l'éventuelle collision de deux objets au programme, afin que celui-ci provoque une explosion.

Dimensions des objets graphiques
Les objets graphiques sont disponibles sur quatre micro-ordinateurs très populaires : le Commodore 64, le Sord M5, le TI99/4A et la gamme Atari.
Les 16 couleurs normalement offertes sont portées à 256 sur l'Atari, puisque chaque couleur comporte 16 degrés d'intensité.
Dans les jeux, il est très souvent utile de détecter les collisions entre deux objets ; c'est pourquoi une fonction de détection de collision est intégrée.
L'effet tridimensionnel est créé en plaçant chaque objet sur un plan différent. L'efficacité du système est proportionnelle au nombre de plans. Les dimensions maximales de chaque objet sont exprimées en points. Ces dimensions peuvent varier et les objets peuvent être déplacés sur l'écran.
Des programmes utilitaires sont offerts pour simplifier la création des objets.

MICRO-ORDINATEUR	NOMBRE DE COULEURS	DÉTECTION DE COLLISIONS	PLANS	TAILLE DES OBJETS	LOGICIEL
Commodore 64	16	Oui	8	24 × 21	Oui
Atari 800	256	Oui	16	16 × 16	Oui
Texas TI 99/4A	16	Oui	28	32 × 32	Oui
Sord M5	16	Oui	32	8 × 8	Non

aux dimensions maximales, et de composer le motif désiré en remplissant les carrés appropriés. Vous savez déjà que l'ordinateur travaille en binaire (voir page 28), les carrés noirs seront donc des 1 et les carrés blancs des 0.

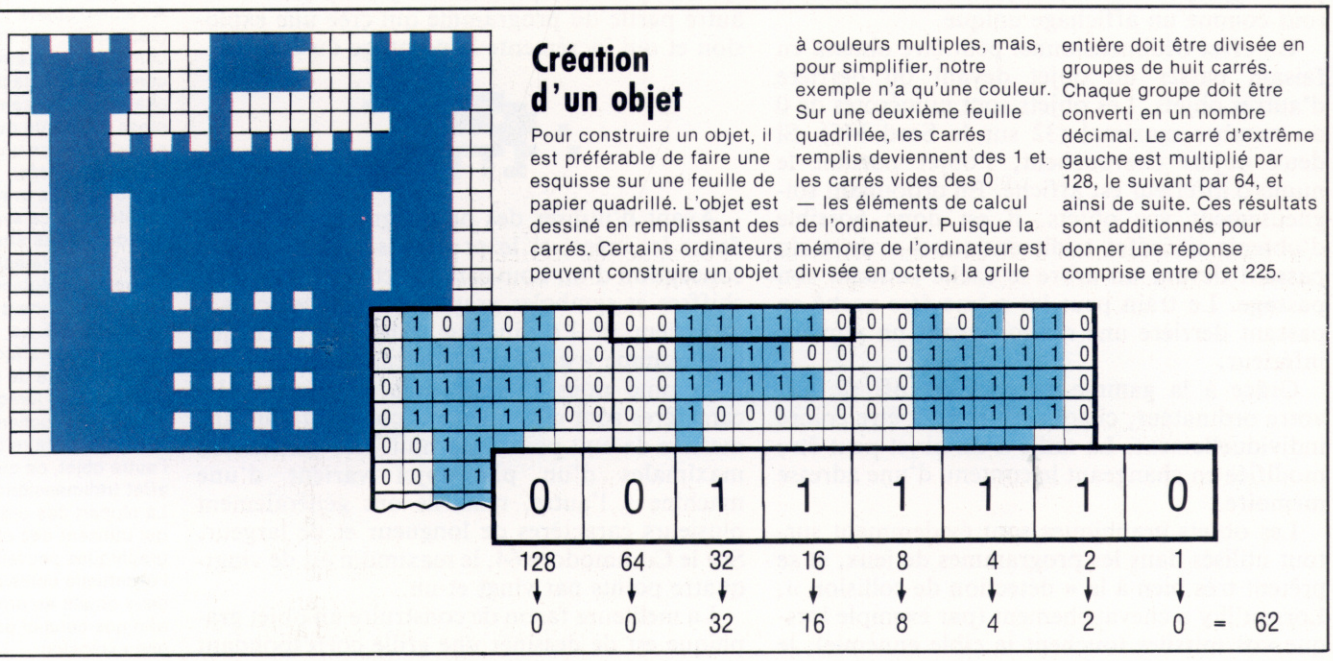
L'ordinateur gère la plupart des données sous la forme d'octets (un groupe de 8 bits). Vous apprendrez dans le manuel de l'ordinateur comment une grille de points composant un objet peut être fractionnée en groupes de huit points. Chaque octet doit être converti en un nombre décimal compris entre 0 et 255 afin de pouvoir être utilisé dans un programme BASIC. Ce nombre est obtenu en multipliant le chiffre binaire d'extrême gauche (bit) par 128, le suivant par 64, et ainsi de suite.

Les nombres décimaux ainsi obtenus définissent entièrement la forme de l'objet. Ces nombres sont placés en mémoire à l'aide d'un programme BASIC, la procédure exacte dépend de la machine utilisée. Le programme doit indiquer à l'ordinateur l'emplacement où il peut trouver les spécifications relatives à chaque objet.

Il ne reste plus qu'à utiliser de simples commandes pour positionner l'objet, pour modifier sa couleur et ses dimensions, ou pour détecter le chevauchement de deux objets.

Des logiciels standards, baptisés « utilitaires », sont disponibles sur la plupart des machines assurant la création d'objets graphiques. Ceux-là facilitent la réalisation d'images en affichant une grille agrandie, ce qui permet de dessiner l'image en déplaçant le curseur sur la grille. Toutes les opérations arithmétiques sont effectuées automatiquement et les résultats sont stockés dans les octets appropriés. Finalement, la grille disparaît, laissant la place à l'objet créé afin de pouvoir le manipuler.

Les objets graphiques ont révolutionné le graphisme sur ordinateur domestique en permettant de créer facilement des scènes animées pleines de vie et de couleurs.



Mark Watkinson

Steve Wozniak

Apple Computers

Des farces électroniques à la réussite économique, l'histoire fantastique de la création des ordinateurs Apple.

Aux États-Unis, Steve Wozniak est sans doute plus connu comme organisateur de concerts de rock que comme concepteur des Apple I et II. Mais, dans l'industrie informatique, il est connu comme le « génie électronique » qui fit plus pour simplifier et pour populariser les micro-ordinateurs que toute autre personne. Les machines de Wozniak furent les premières à offrir la couleur, des graphiques, un clavier et un écran vidéo dans la configuration standard. Plus d'un million d'Apple II ont déjà été vendus.

L'histoire de Wozniak a tout du conte de fées californien. Il est né et a été élevé en Californie, dans la Silicon Valley, lieu de naissance des « puces ». Son père était ingénieur. Même s'il enseigna à son fils les règles fondamentales de l'électronique, Wozniak acquit par lui-même ses connaissances techniques.

Adolescent, il aimait bien s'amuser avec des composants électroniques. Il réussit à construire un dispositif électronique nommé « boîte bleue » : une idée qu'il trouva dans un périodique. Ce dispositif pouvait imiter certaines tonalités du système téléphonique. Ces tonalités servent à signifier le paiement d'une communication dans une cabine, ce qui permit à Wozniak de téléphoner gratuitement un peu partout. Il téléphona souvent en Europe, et parla même au pape !

Wozniak n'a aucun diplôme d'ingénieur ; il était doué en mathématiques et en électronique, mais décida de quitter le lycée. Il commença à travailler comme technicien chez Hewlett Packard, où il concevait des calculatrices. Mais son but était de concevoir des ordinateurs. Il se mit à travailler en solitaire, généralement la nuit, et conçut un ordinateur qu'il proposa à Hewlett Packard, mais son projet fut rejeté. Déçu, il quitta Hewlett et, avec son ami d'enfance Steve Jobs, construisit et vendit cinquante ordinateurs. L'Apple I était né. Le nom Apple fut choisi simplement parce que Jobs avait déjà travaillé dans un verger.

Entre 1975 et 1976, Wozniak s'enferma dans son garage. Toutes ces longues journées de travail aboutirent finalement à la production de l'Apple II. Il avait vingt-six ans. Les experts considèrent toujours l'Apple II comme une étonnante démonstration de simplicité de conception.

Une des innovations importantes qu'apporta l'Apple II fut la simplification du lecteur de disquettes. Avant Wozniak, les lecteurs nécessitaient trente puces ; il en conçut un qui n'utilisait que cinq puces. En fait, Wozniak n'inventa rien de nouveau, il ne fit que simplifier les composants et les assembla dans un ordinateur à la portée de tous.

Wozniak n'eut jamais l'intention de devenir gestionnaire. Cette tâche revint à son ami Steve Jobs qui se chargea du marketing des Apple et de la mise sur pied de l'Apple Corporation. La société emploie maintenant plus de trois mille personnes dans le monde entier et a fait plus de cinquante millionnaires. Wozniak ne détient que 4 % de la société Apple et n'a jamais été impliqué dans sa gestion. Il préfère jouer avec les ordinateurs et concevoir de nouveaux produits.

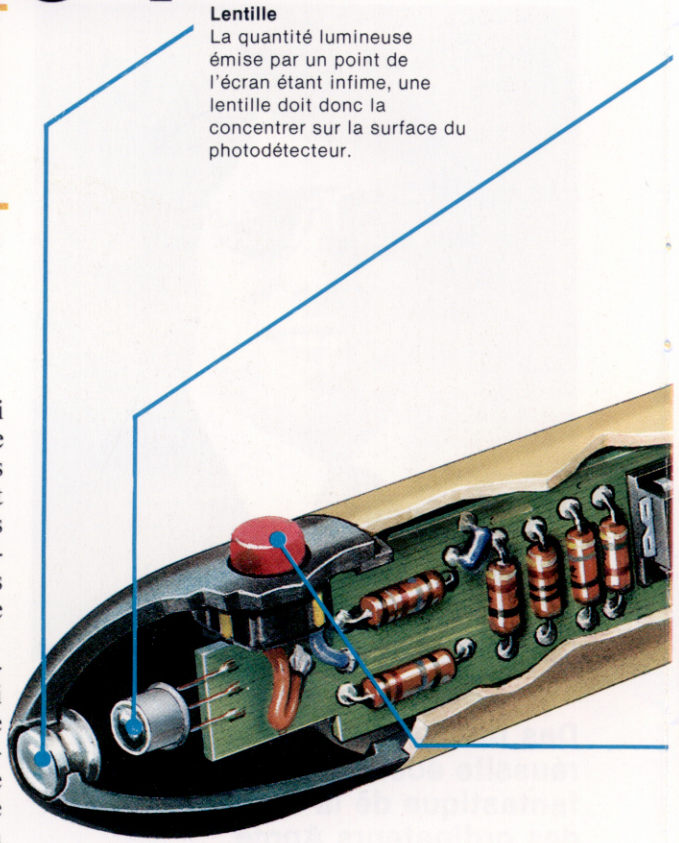
Après un arrêt de deux ans, pendant lequel il organisa surtout des concerts de rock, il décida de revenir chez Apple au cours de l'été 1983. Personne ne connaît l'objet de ses travaux actuels. Il écrit probablement des programmes d'application. Il participe également à un projet Apple II concernant un système de production et d'édition vidéo comportant des graphiques de haute qualité pouvant produire des dessins animés. Mais Wozniak ne s'intéresse pas uniquement à l'amélioration des machines existantes, il croit aussi qu'il sera bientôt possible de concevoir un micro-ordinateur très intelligent. Cette machine pourrait apprendre n'importe quoi à l'aide d'un seul programme. Nous devrons attendre pour découvrir la prochaine trouvaille de ce brillant ingénieur pas très conventionnel.

Le président

Steven Jobs, président d'Apple Computer Inc., était un des camarades d'école de Steve Wozniak. Si Wozniak était doué pour l'électronique, Jobs l'était tout autant pour les affaires. En 1975, quand l'Apple fut conçu, Jobs avait vingt ans. Il vit le potentiel commercial de la machine non seulement pour les mordus d'ordinateurs du moment, mais encore pour un nouveau marché d'informatique individuelle. Wozniak fut le créateur de la machine, mais Jobs fut responsable de la mise en production et de la stratégie de vente. L'Apple II fut présenté dans un boîtier complet, prêt à être branché dans n'importe quel foyer. Au début, pour financer leur projet, Jobs vendit sa voiture et Wozniak sa calculatrice programmable. Mais bientôt un jeune millionnaire, Mike Markkula, leur apporta un soutien financier pour ce qu'il considérait comme un projet intéressant. Lorsque la société fut cotée en Bourse en 1980, ses actions s'envolèrent littéralement, et les espoirs mis dans cette société ne furent pas déçus, puisque son chiffre d'affaires en 1982 a atteint 583 millions de dollars. La progression phénoménale d'Apple Corporation fut applaudie partout aux États-Unis. Steve Jobs fit la couverture du prestigieux magazine *Time*, qui n'hésita pas à dire de lui qu'il était le jeune homme d'affaires le plus talentueux des États-Unis.

Crayons magiques

Les crayons optiques servent à dessiner sur l'écran ou à effectuer des sélections. Voici comment fonctionnent ces dispositifs remarquables.



La « phobie du clavier » est sans doute ce qui freine le plus l'utilisation de l'ordinateur, que ce soit à la maison ou au bureau. Les personnes qui ne savent pas taper et qui, pis encore, voient sur le clavier de l'ordinateur des symboles inconnus, craignent de se ridiculiser en l'utilisant. Le crayon optique, en plus de ses autres usages, est une solution à ce problème (l'entrée vocale de données en est une autre).

Un crayon optique est un dispositif cylindrique (ressemblant à un stylo ordinaire) muni d'un fil à l'une de ses extrémités. Au bout de ce fil, une fiche se branche à l'arrière de l'ordinateur. Quand le crayon optique est placé contre l'écran (sur certains systèmes, il est nécessaire d'appuyer le crayon contre l'écran de façon à activer l'interrupteur situé à l'intérieur du crayon), l'ordinateur peut détecter quelle position de l'écran est désignée par le crayon.

Pour ce faire, un photodétecteur placé au bout du crayon envoie une impulsion électrique chaque fois qu'il reçoit le faisceau électronique qui balaie l'écran de façon permanente pour créer l'image. Certains circuits du contrôleur vidéo calculent la position de balayage au moment de l'émission du signal.

Le crayon optique est surtout utilisé pour sélectionner un élément affiché à l'écran. Connaissant le point de l'écran où se trouve le crayon, l'ordinateur peut déduire quel caractère est désigné. Plusieurs programmes d'application comportent des « menus ». Un menu est simplement une liste d'options parmi lesquelles l'utilisateur doit faire une sélection — tout comme au restaurant. Un programme de gestion domestique pourrait proposer le menu suivant :

1) Effectuer un paiement.
2) Solde bancaire.
3) Entrer un reçu,

et ainsi de suite.

Normalement, l'utilisateur spécifie le type d'intervention désiré en appuyant sur une touche (1, 2 ou 3) ou en tapant un mot de commande. Avec un crayon optique, il n'a qu'à désigner l'option désirée. L'ordinateur répond généralement en faisant clignoter l'option en question afin de signifier l'acceptation de cette entrée. Certains programmes sophistiqués sont pilotés entièrement à l'aide de tels menus, et là l'utilisateur n'a à utiliser le clavier que quand des données spécifiques doivent être fournies.

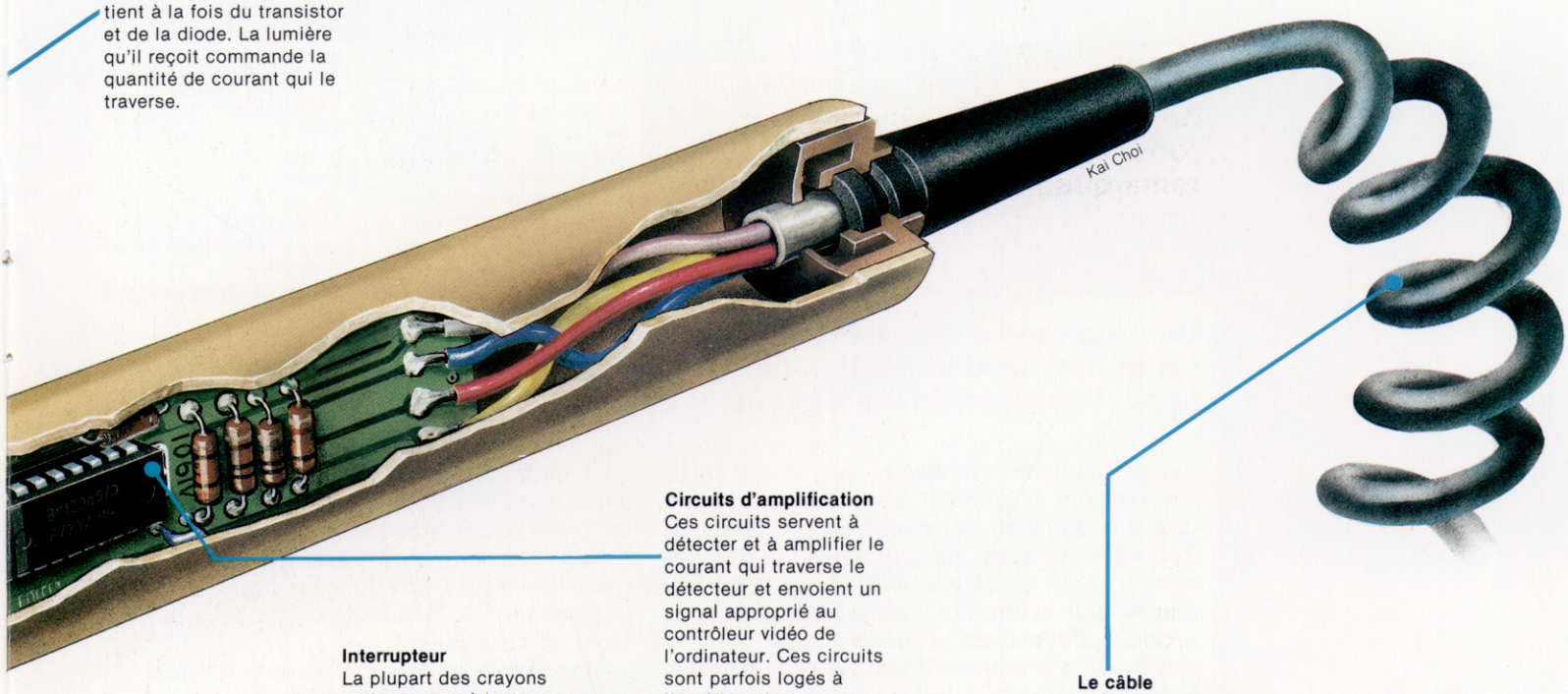
Routines spéciales

Mais un tel programme doit tenir compte de l'utilisation du crayon lumineux. Il ne s'agit que d'inclure dans le programme une courte routine qui reçoit à partir du contrôleur vidéo les coordonnées de la position occupée par le crayon et qui déduit quelle option est sélectionnée. Malheureusement, peu de fabricants fournissent des versions avec un crayon optique.

Cependant, en plus de la sélection des options, un crayon optique peut servir à créer des images à l'écran. La plupart des ordinateurs domestiques qui disposent d'un crayon optique offrent un logiciel à cet effet. Le programme propose un écran vierge à l'utilisateur sur lequel celui-ci peut dessiner à volonté, et affiche sur une portion distincte de l'écran (généralement au bas de l'écran) une série de fonctions d'aide. Une de ces fonctions affiche une palette de couleurs comparable à la palette d'un artiste peintre. Le crayon est placé sur la prochaine couleur désirée, et à partir de ce moment les lignes tracées par le crayon sont de cette couleur.

L'utilisateur peut aussi sélectionner diverses largeurs de traits au bas de l'écran et a la possi-

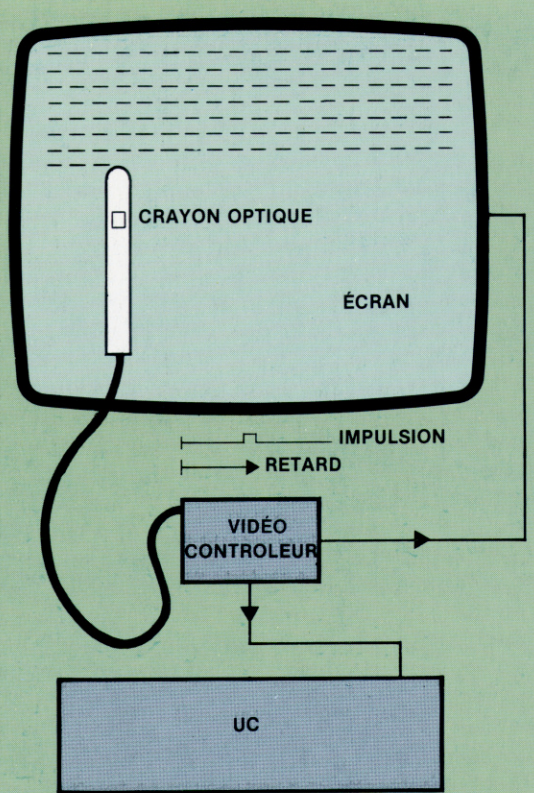
Photodétecteur
Ce dispositif électronique tient à la fois du transistor et de la diode. La lumière qu'il reçoit commande la quantité de courant qui le traverse.



Circuits d'amplification
Ces circuits servent à détecter et à amplifier le courant qui traverse le détecteur et envoient un signal approprié au contrôleur vidéo de l'ordinateur. Ces circuits sont parfois logés à l'extérieur du crayon.

Interrupteur
La plupart des crayons optiques possèdent un interrupteur quelconque, actionné soit par une pression du doigt, soit en appuyant le crayon contre l'écran. Cet interrupteur sert à ce que le crayon ne réagisse pas à la lumière ambiante lorsqu'il n'est pas utilisé sur l'écran.

Le câble
Ce fil se branche directement à l'arrière de l'ordinateur et sert à acheminer des signaux vers le contrôleur vidéo. (Cl. Kai Choi.)



Fonctionnement

Le fonctionnement d'un crayon optique est généralement sous la responsabilité du contrôleur vidéo. Cette puce produit les images vidéo sur le téléviseur ou sur le moniteur en balayant l'écran plusieurs fois par seconde (balayage de trame). En France, il y a 625 lignes sur l'écran et la fréquence de balayage est de 50 par seconde. Le signal de sortie du crayon est envoyé directement dans cette puce. Quand le crayon est placé contre l'écran et qu'un signal est envoyé, le contrôleur vidéo calcule le temps écoulé entre le moment où il a balayé le haut de l'écran et l'émission du signal par le crayon optique. A partir de ce résultat, il peut calculer les coordonnées exactes de la position occupée par le crayon. (Cl. Liz Dixon.)

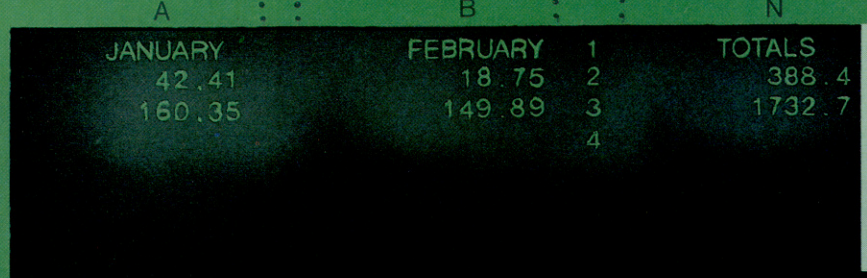
bilité de dessiner des cercles et des carrés. En bref, toutes les opérations que nous avions présentées dans « L'artiste électronique » (voir page 26) peuvent être réalisées avec un crayon optique, probablement beaucoup plus rapidement qu'au clavier.

On commence à voir apparaître des jeux utilisant des crayons optiques. La poursuite d'un monstre est beaucoup plus facile avec un crayon optique qu'au clavier, et les jeux doivent donc être beaucoup plus difficiles. On peut jouer à des jeux plus calmes avec un crayon optique, aux échecs par exemple. Le joueur n'a qu'à désigner l'emplacement où il désire déplacer la pièce, et l'ordinateur se charge du reste.

Les crayons optiques sont surtout utilisés par les concepteurs et les dessinateurs. Les systèmes de conception assistée par ordinateur (CAO) sont en fait des ordinateurs ordinaires, mais sont dotés de logiciels spécialisés et d'une excellente qualité graphique. Un système CAO utilisé pour concevoir un nouveau dispositif optique affiche des représentations de tous les composants pouvant être requis par le concepteur; ces composants peuvent être sélectionnés par le concepteur et placés à l'endroit désiré sur l'écran.

Le crayon optique est un excellent exemple de dispositif complémentaire joignant l'utile à l'agréable.

Planification



Champs
Une feuille multizone est divisée en lignes et en colonnes pour former des champs ou des cellules qui peuvent être adressés par leurs coordonnées (A1, B3, etc.). Chaque champ peut contenir un titre (JANVIER), un nombre (149.89) ou une formule. Le champ N2 contient la formule « SUM(A2:M2) » qui est la somme des chiffres de la rangée supérieure, de janvier à décembre. Le résultat de ce calcul est affiché : 338.4. Notez que la feuille a été divisée en deux fenêtres.

Un programme de « feuille de calcul électronique » peut accroître les domaines d'utilisation de votre ordinateur. Ses principales applications sont l'établissement de budget et les prévisions.

Les gestionnaires consacrent environ 30 % de leur temps à la préparation de budgets, opération qui implique de nombreuses hypothèses. Traditionnellement, on utilisait une feuille de papier qui était divisée verticalement en une douzaine de colonnes ou plus. Les dépenses du mois des diverses catégories de dépenses étaient entrées dans chaque colonne. De cette façon, il était possible d'additionner les colonnes et de connaître ainsi les dépenses par mois, ou d'additionner les rangées pour connaître les dépenses totales d'une catégorie pendant l'année.

Mais, quand les prévisions de dépenses étaient trop élevées — ou, pire encore, trop basses —, le gestionnaire devait modifier de nombreuses données et recalculer les divers totaux.

Grâce à un programme de feuille de calcul, les éléments d'une page entière peuvent être recalculés automatiquement à chaque fois qu'un élément est modifié; il suffit d'appuyer sur une touche. Si, par exemple, le coût du transport du mois de janvier est modifié, les dépenses totales de ce mois sont modifiées, ainsi que les dépenses totales de l'année de cette catégorie. Pas étonnant que les programmes de feuille de calcul électronique soient les logiciels les plus vendus !

Tout comme la plupart des programmes professionnels, les progammes de feuille de calcul sont normalement écrits en segments de recouvrement, c'est-à-dire que le programme n'est jamais totalement résident en mémoire. Imaginons le programme écrit en plusieurs sous-programmes (voir page 77). Un sous-programme qui n'est pas utilisé lors de l'opération en cours n'est pas chargé en mémoire tant qu'un « appel » ne lui est pas adressé. Le système d'exploitation recouvre tout sous-programme devenu inutile. Cette méthode est certes pratique, mais implique des périodes d'attente lors du chargement des divers segments du programme.

Ces programmes de calcul ont très souvent un nom se terminant par « calc » et sont offerts sur la plupart des ordinateurs domestiques et de gestion. Le plus populaire est Visicalc; il a été écrit initialement pour l'Apple II et diffusé à partir de 1979. Les concepteurs de logiciels pour micro-ordinateurs réagissent très rapidement au succès et ils ne firent pas exception à cette occasion. Le marché fut immédiatement envahi de programmes de feuille de calcul pour tous les types de machines.

Une feuille de calcul électronique, pour être vraiment utile, doit être de dimensions suffisantes (pas nécessairement ce que vous voyez à l'écran puisque, comme vous le verrez plus loin, l'écran n'est qu'une « fenêtre » sur l'ensemble) et doit disposer d'une bonne gamme de fonctions de mise en page et de commande.

Toutes les machines ne peuvent pas utiliser pleinement le potentiel de ce type de programme. Les exigences minimales requises au niveau du matériel sont les suivantes : 32 K de mémoire RAM et un affichage sur quatre-vingts colonnes, bien qu'un affichage sur quarante colonnes puisse suffire à une application domestique.

Les utilisateurs d'ordinateurs domestiques trouveront certains de ces programmes disponibles sur cassettes. Ces logiciels ont des caractéristiques évidemment restreintes, mais peuvent quand même se révéler très utiles.

Puisque les feuilles de calcul électronique peuvent tenir compte de diverses hypothèses, il est possible de définir des modèles informatisés (voir page 101). L'utilisation d'une feuille de calcul électronique nécessite au préalable une excellente planification. Les bases de données, comme nous l'avons mentionné, consistent en une masse d'informations brutes qui, lors de leur extraction, sont sélectionnées et triées selon les spécifications de l'utilisateur. Les programmes de traitement de texte offrent à l'utilisateur des fonctions puissantes pour créer des documents. Les programmes de feuille de calcul sont un peu différents, puisque l'utilisateur doit faire un important travail de planification avant de s'en servir.

Par exemple, si vous utilisez un programme de feuille de calcul pour analyser vos dépenses, il est préférable de regrouper toutes les dépenses relatives à la maison — loyers ou rembourse-

La ligne du bas
Le curseur indique dans quel champ apparaissent les données tapées. Le contenu de la cellule occupée par le curseur est aussi affiché sur la ligne de commande de la feuille, qui ici se trouve au bas de l'écran.

FORM:SUM(A2:M2)

ments, assurances, etc. — et d'utiliser ce résultat dans un calcul plus global. Vous devez veiller à inclure toutes les dépenses « maison » avant de transférer le résultat.

Chaque case individuelle d'une feuille de calcul électronique se nomme une cellule et est adressée au moyen de ses coordonnées X et Y. On utilise normalement les lettres A-Z, AA-ZZ, et quelquefois BA-BM pour identifier les colonnes sur toute la largeur de la feuille. Verticalement, on utilise les numéros 0-256. Chaque cellule peut contenir une étiquette (comme « ventes » ou « profit »), une valeur entrée ou calculée (comme 1 000) ou la formule de ce calcul, comme B4 + B6 * B5. Les formules, étant souvent trop longues pour être affichées dans la case, sont généralement affichées en haut de l'écran. Lors du chargement initial du programme, la dimension des cellules est définie par défaut à huit ou neuf caractères ou chiffres. Cette dimension peut être modifiée, et la plupart des versions permettent de changer la dimension des cellules occupées. Si la cellule devient plus petite que son contenu, la partie non affichée n'est pas perdue mais simplement voilée.

La ligne de commande est également très importante; elle est affichée en haut ou en bas de l'écran en réponse à une commande : / par exemple. Ces commandes servent à formater et à manipuler la feuille. La plupart des versions permettent une grande variété d'opérations sur la base de données. Par exemple, vous pouvez effacer, déplacer ou copier des colonnes ou des lignes entières. Vous pouvez fractionner la fenêtre pour afficher ensemble des parties non adjacentes de la feuille, et vous pouvez faire défiler ces fenêtres séparément.

Normalement, les déplacements entre les diverses cellules s'effectuent à l'aide des touches de commande du curseur, mais une commande permet de passer à la cellule spécifiée. Les opérations de sauvegarde, d'effacement et de protection sont faites à partir de la ligne de commande. Rappelons l'importance de la sauvegarde. Il est généralement beaucoup plus long de préparer la feuille que d'y entrer les données, et il est donc préférable de faire une sauvegarde avant de commencer l'entrée des données. Ainsi, votre travail ne sera pas perdu en cas d'erreur de manipulation.

Une commande provoque l'impression des résultats, mais la partie de la feuille à imprimer doit être définie soigneusement au moyen des paramètres d'impression. L'écran n'est qu'une fenêtre sur la feuille ou une portion de la feuille, et il en est de même pour une sortie imprimée. Si vous devez imprimer une feuille plus large que votre imprimante, vous pouvez faire deux impressions et assembler les deux feuilles.

L'utilisation de fenêtres permet d'afficher deux parties différentes de la feuille en même temps. Une feuille fractionnée peut aussi être imprimée. Cette caractéristique est surtout utile lors de la saisie de données, puisque l'opérateur peut voir les données antérieures. La plupart des programmes permettent de conserver la ligne de titre ou les premières lignes qui sont très utiles puisqu'elles contiennent les titres, ou étiquettes, des colonnes.

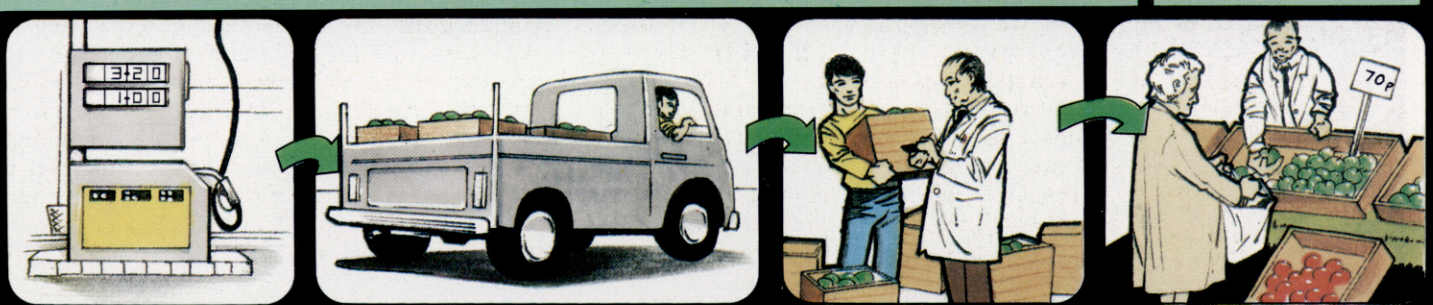
Nous avons jusqu'à maintenant considéré la feuille comme un tableau où tous les éléments sont rangés les uns après les autres en colonnes, mais cet agencement peut être modifié selon les besoins de l'utilisateur. Soulignons qu'un agencement sophistiqué risque d'impliquer un travail de planification plus long qu'à l'ordinaire.

Les programmes à vocation professionnelle comme Visicalc, Supercalc et Masterplan offrent tous à l'utilisateur la possibilité de transférer les données vers des programmes de bases de données ou de traitement de texte, et il existe de nombreux programmes permettant des sorties sous diverses formes graphiques : graphique circulaire ou diagramme de Gantt.

Nous avons déjà mentionné une application possible des feuilles de calcul électronique pour l'utilisateur d'ordinateur domestique : l'analyse des dépenses domestiques et l'établissement d'un budget. Une autre application domestique pourrait planifier l'installation d'un système de chauffage central. Ici, on doit tenir compte

Hypothèses...
Si le contenu d'un champ est modifié, la feuille multizone recalcule automatiquement tous les autres champs qui dépendent d'une façon ou d'une autre de ce champ. La vitesse et la simplicité du processus encouragent les utilisateurs à tester diverses hypothèses.

Mark Watkinson



Kevin Jones

Si le prix de l'essence montait de X %...

Les coûts mensuels de transport grimperaient de Y %...

Cela se traduirait par une augmentation du prix de gros des fruits...

qui devrait être répercutée sur le prix payé par le consommateur.

d'un grand nombre de variables : le type de combustible à utiliser, le nombre et le type de radiateurs, la température du liquide, etc. En fait, tout processus de prise de décision peut être facilité par l'utilisation d'une feuille de calcul électronique.

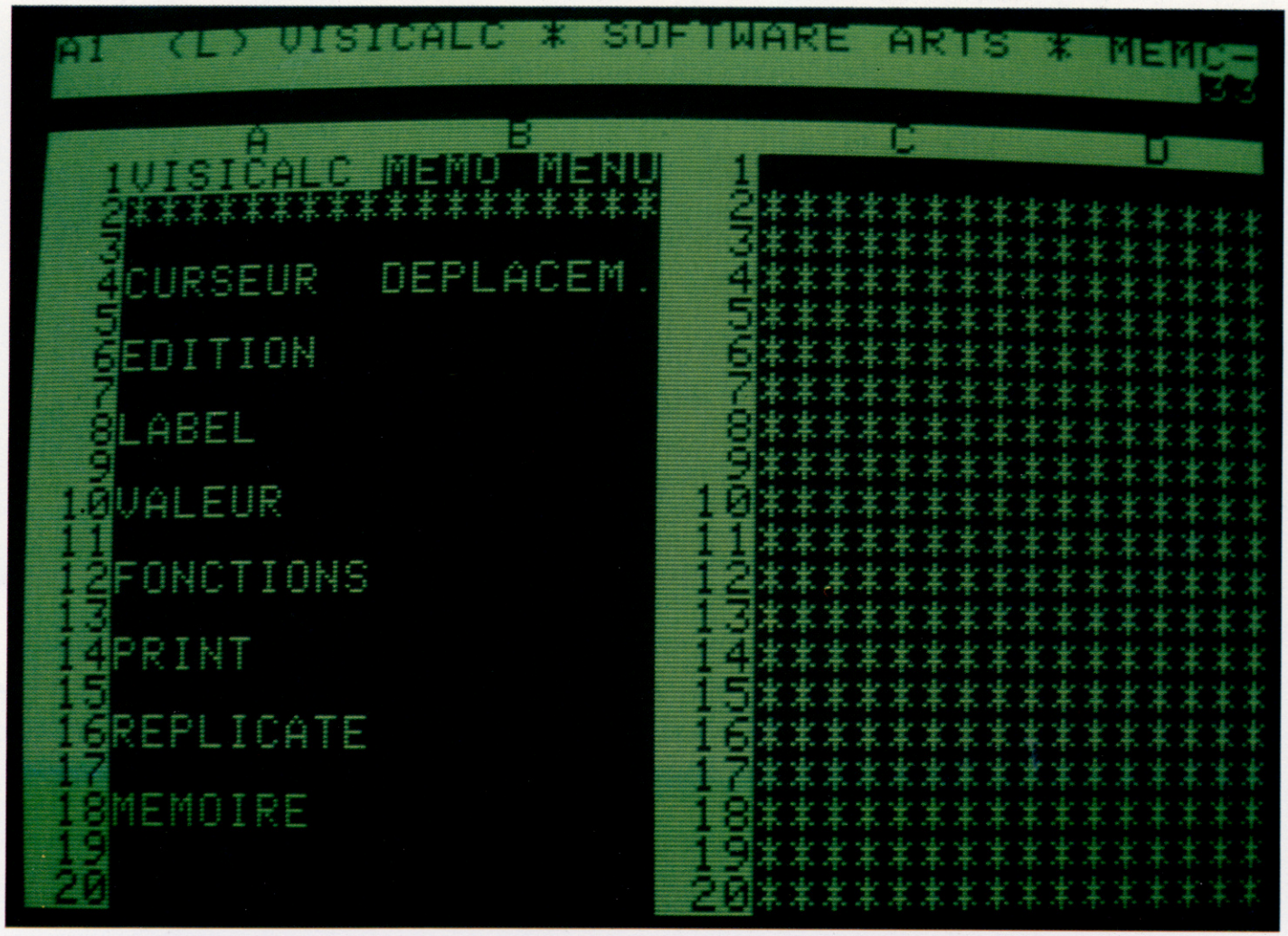
La principale qualité des feuilles de calcul à vocation professionnelle est leur vitesse d'exécution, due à la programmation en code machine. La vitesse d'un programme écrit en BASIC pour un petit ordinateur domestique risque d'être assez décevante.

Voyons les problèmes qu'aurait à résoudre une personne désirant écrire un tel programme en BASIC.

D'abord, chaque cellule doit être définie de trois façons. Elle doit être capable de contenir des variables de chaînes, comme « janvier » ou « charges »; elle doit pouvoir contenir des données numériques, le montant des charges payées en janvier, par exemple; elle doit pouvoir contenir une formule, qui est en fait une ligne d'instruction, comme « charges annuelles/12 = charges mensuelles ». Puis la taille de chaque cellule doit pouvoir être modifiée sans perdre son contenu. Chaque cellule doit être reproduite : une cellule est destinée à l'affichage, l'autre, cachée à l'intérieur du programme, au stockage des données.

Comme vous le voyez, seule la manipulation des données représente une tâche très complexe. Rappelons que les programmes les plus évolués peuvent gérer plus de seize mille cellules individuelles! L'écriture de tels programmes est comparable à l'écriture d'interpréteurs de langage comme BASIC ou FORTH, ou de logiciels de gestion de base de données.

Tout cela explique le coût élevé des programmes professionnels. Un logiciel comme Visicalc ou Supercalc peut coûter des milliers de francs, mais on ne doit pas oublier les économies qu'il peut entraîner. L'application très simple que nous avons présentée, l'établissement de budgets par un gestionnaire, peut entraîner une économie de 10 à 20 % par an. Le coût du logiciel est donc insignifiant par rapport à une telle économie. C'est pourquoi Visicalc est probablement le meilleur ami du vendeur Apple.



Le Visicalc
Le menu du Visicalc permet à l'utilisateur de l'adapter en fonction de ses besoins. Une fois la feuille multizone établie, pour se déplacer, il suffit de déplacer le curseur vers le haut, le bas, la gauche ou la droite. Il est ainsi possible d'afficher la totalité de la feuille à l'écran.
Certains ordinateurs portatifs dotés de possibilités d'affichage limitées, comme l'Epson HX-20 et l'Osborne-1, utilisent ce système de défilement dans toutes leurs opérations.
(Cl. *L'Expansion.*)

PROGRAMME N° 2

Nous allons étudier comment rendre « portable » le premier programme. Portable, c'est-à-dire présentable et compréhensible par tout utilisateur.

Rappel du premier programme :

```
§LIST

10 INPUT A
20 INPUT B
30 S = A + B
40 PRINT S
50 END
```

A ce programme, nous allons d'abord adjoindre des *messages,* ou commentaires.

Exemples :

10 INPUT « DONNER LA VALEUR DE A »; A Dans cette ligne de programme, l'instruction INPUT est suivie d'un petit texte qui est affiché avant que l'ordinateur ne lise les données introduites par l'utilisateur au clavier.

En exécutant cette introduction, on verra apparaître à l'écran :

```
DONNER LA VALEUR DE A
```

Le programme deviendra :

```
§LIST

10 INPUT « DONNER UNE VALEUR A »; A
20 INPUT « DONNER UNE VALEUR B »; B
30 S = A + B
40 PRINT « LA SOMME A + B EST EGALE A »; S
50 END
```

On pourrait avoir aussi :

```
§LIST

10 INPUT « QUELLE EST LA VALEUR
   DE A : »; A
20 INPUT « QUELLE EST LA VALEUR
   DE B : »; B
30 S = A + B
40 PRINT « LA SOMME DE A ET DE B
   EST : »; S
50 END

§RUN
QUELLE EST LA VALEUR DE A : 12
QUELLE EST LA VALEUR DE B : 45
LA SOMME DE A ET DE B EST : 57
```

De la même manière, pour le programme somme de trois nombres, on obtiendrait :

```
10 INPUT « DONNEZ A : »; A
20 INPUT « DONNEZ B : »; B
30 INPUT « DONNEZ C : »; C
40 S = A + B + C
50 PRINT « LA SOMME DE A DE B ET DE C
   EST : »; S
60 END

§RUN
DONNEZ A : 34
DONNEZ B : 78
DONNEZ C : 986
LA SOMME DE A DE B ET DE C EST : 1098
```

L'ORGANIGRAMME

Les programmes précédents ont montré comment on pourrait facilement apprendre les instructions de base du langage BASIC. Cependant, dès que le programme devient plus complexe, il n'est plus possible de l'écrire directement. Il faut procéder tout d'abord à une analyse du problème et écrire l'*organigramme* correspondant. En effet, l'expérience montre que celui-ci est une aide précieuse, voire indispensable, à l'élaboration du programme, en particulier pour les débutants.

Le but de cette leçon est de vous expliquer comment s'élabore la construction d'un organigramme.

L'organigramme constitue une visualisation graphique du traitement envisagé à l'usage de l'utilisateur.

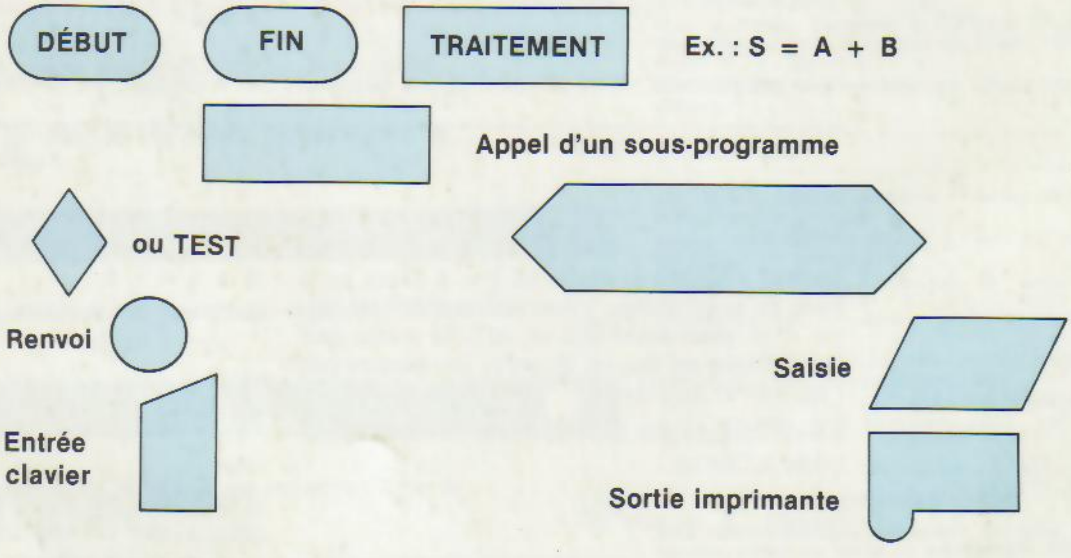
L'organigramme permet de vérifier, lors de l'analyse du problème, qu'aucun sous-problème n'a été omis. Il facilite l'échange d'informations entre les différentes personnes participant à la rédaction d'un programme. Pour le néophyte, l'organigramme constitue une étape préliminaire facilitant une bonne programmation; il reste, dans la mesure du possible, indépendant du langage de programmation utilisé. Les techniques d'organigramme ont fait l'objet d'une norme AFNOR (Z 67-010) homologuée en avril 1966. Nous verrons ci-après quelques symboles utilisés.

REMARQUES

Ordinogramme, terme qui n'est pas normalisé, est un synonyme d'organigramme; l'équivalent anglais est *flowchart*.

Il existe des méthodes qui permettent de décrire de façon plus concise des algorithmes (en métalangage), mais elles nécessitent de bonnes connaissances du programme.

QUELQUES EXEMPLES DE SYMBOLES UTILISÉS



Essayons, à l'aide de ces symboles, de créer l'organigramme du premier programme.