

ABC

N° 9

COURS
D'INFORMATIQUE
PRATIQUE
ET FAMILIALE

INFORMATIQUE



Les ordinateurs portables

L'aventure avec l'ordinateur

Un langage pour les enfants

Les robots arrivent

EDITIONS
ATLAS

M6062-9-12F

85FB-3,80FS-\$1.95

Sommaire

Matériel



La taille des ordinateurs a diminué à un tel point qu'ils peuvent maintenant être transportés dans une mallette. Comment un tel équipement peut-il vous être utile? 166

L'Epson HX-20 est le premier ordinateur vraiment portable muni d'un affichage, d'une imprimante et d'une unité à cassette intégrés. 169

Logiciel



Les jeux d'aventures rivalisent maintenant en popularité avec les jeux d'arcades. 161

Programmation basic



Il n'est pas facile pour un ordinateur de produire des nombres véritablement aléatoires, mais ils sont essentiels dans les jeux. 172

Le marché



Le langage Logo permet de construire sur ordinateur des micro-environnements où l'enfant peut découvrir et explorer le monde réel. 164

Un robot peut être à la fois divertissant et instructif. 176

Mots de passe



La base 16 est une des plus efficaces en informatique. 179

Les pionniers



Chuck Peddle est un des pères de la micro-informatique. Cet homme conçut la puce 6502, le PET et le Sirius. 180

Prochain fascicule

Nous examinons l'Alice, qui, pour un prix modique, offre un potentiel étonnant.

En stockant une représentation numérique des composantes de la parole et en les enchaînant ultérieurement à la bonne vitesse et au bon rythme, les ordinateurs ont maintenant la possibilité de nous répondre oralement.

Nous explorons le domaine de l'animation sur ordinateur : s'agit-il d'un nouvel outil pour l'illustrateur ou d'un nouvel art?



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° 9.

ABC INFORMATIQUE est réalisé avec la collaboration de Trystan Mordrel (secrétariat de rédaction), S.I.-André Laroche (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 Italie par I.G.D.A., Officine Grafiche, Novara. Distribution en France : N.M.P.P. Tax. Dépôt légal : mars 1984. 2843. Dépôt légal en Belgique : D/84/2783/27.

© Orbis Publishing Ltd., London.
© Éditions Atlas, Paris, 1984.

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



Les jeux d'aventures

Dans un jeu d'aventures, vous n'êtes pas uniquement spectateur, mais vous participez réellement à l'action décrite.

Le mot « aventure » évoque pour la plupart des personnes un livre, un film, un programme télévisé ou peut-être une expérience personnelle. Mais ce mot rappelle immédiatement à certains utilisateurs d'ordinateurs un jeu particulier.

Les jeux d'aventures peuvent être comparés à des livres. La lecture d'une histoire d'aventures séduit par ses moments de suspense et de danger, mais ces événements demeurent étrangers au lecteur. Dans une aventure sur ordinateur, vous n'êtes pas qu'observateur, vous devenez participant. A titre de principal acteur, vous êtes plongé dans l'action et vous vivez une réelle expérience.

Dans un livre, le lecteur ne peut pas intervenir. Le déroulement des événements ne peut être modifié. Dans un jeu d'aventures, vos décisions et interventions déterminent et orientent l'intrigue. Une multitude d'options vous sont offertes, occasionnant des résultats quelquefois heureux, mais qui peuvent aussi être désagréables. Confortablement installé au foyer, vous prenez part à une aventure excitante.

Chaque aventure se déroule dans un environnement particulier. Ce peut être un étrange monde souterrain, une ville fantôme, une autre planète... De plus, l'action peut se dérouler dans le passé, dans le présent ou dans l'avenir.

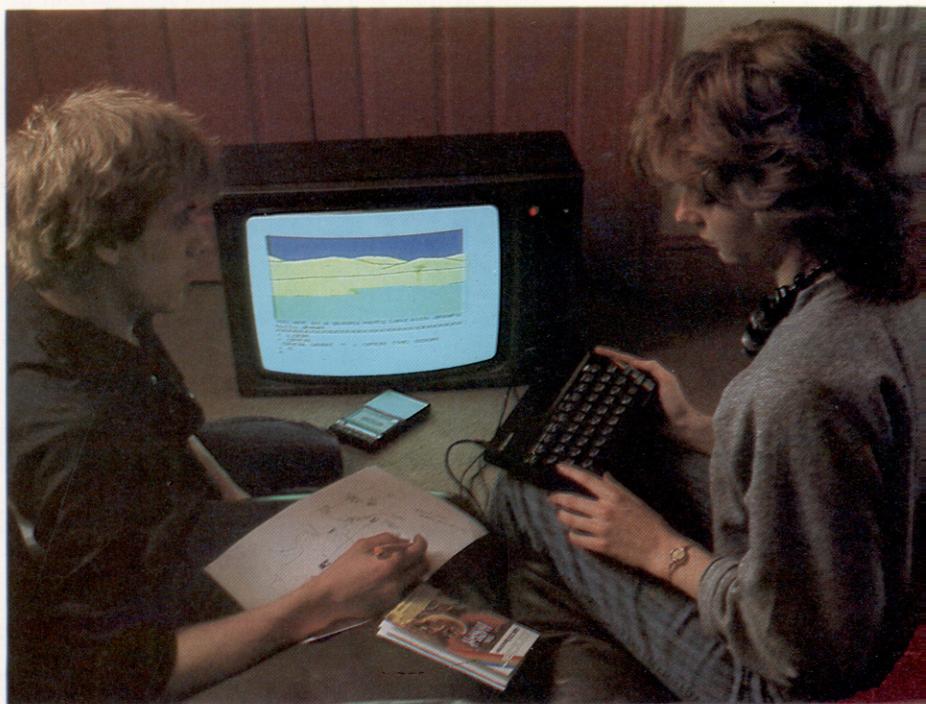
L'aventure a généralement un thème qui comporte un objectif ultime. Par exemple, il est possible que vous ayez à vous évader d'une planète étrangère, trouver et supprimer un sorcier, secourir une princesse, rechercher un trésor, découvrir et arrêter un criminel.

La résolution des différentes énigmes apporte une réelle satisfaction au joueur. Elles font partie de l'aventure. Très souvent, l'énigme doit être résolue avant de pouvoir poursuivre : la traversée d'un pont dangereusement chancelant par exemple. A l'autre extrême, l'énigme ne peut être qu'une diversion : après avoir franchi une gorge pour atteindre un personnage étrange qui se trouve de l'autre côté, vous vous retrouvez devant un énorme miroir. La solution d'une énigme peut quelquefois être utile mais non essentielle pour la réussite du jeu, comme par exemple la découverte d'un passage secret plus facile. La solution d'une énigme peut enfin être vitale pour vous et pour vos compagnons.

Ces énigmes peuvent être résolues avec un peu de bon sens et ne requièrent aucune connaissance particulière. Cependant, l'aventurier doit être attentif, puisque d'abondants indices sont fournis ou peuvent être déduits facile-

ment. Les éléments aléatoires (simulant un coup du sort) sont très rares dans un programme d'aventures bien écrit.

Même si dans un jeu d'aventures certains événements semblent isolés, vous devez garder à l'esprit que tout est intimement lié. Un fait anodin ne doit pas être négligé, il peut avoir une signification importante. Toutes les énigmes proposées composent l'intrigue. Lorsque vous rencontrez des objets qui semblent étranges et sans aucun rapport avec ce que vous recherchez, vous devez vous douter qu'ils peuvent vous être utiles ultérieurement.



Plusieurs aventures comportent des labyrinthes composés de pièces parfaitement identiques. La seule façon de vous y retrouver est de laisser des objets témoins, ce qui vous permet d'identifier les pièces par la suite. Cependant, cette méthode étant maintenant bien connue, les auteurs ont ajouté de nouvelles difficultés ; par exemple, un voleur vous suit et dérobe ou redisse les objets que vous avez laissés derrière vous.

Dans certaines aventures, même si vous devez résoudre toutes les énigmes et atteindre les divers objectifs pour être vainqueur, le parcours que vous empruntez n'a pas d'importance. Dans d'autres, le parcours est imposé.

Un rôle actif

Les jeux d'aventures existent depuis de nombreuses années, mais l'ordinateur leur a donné une nouvelle dimension. Les jeux d'aventures rivalisent de plus en plus avec les jeux d'arcades. Bien que la plupart ne puissent être joués que par un joueur à la fois, ils constituent un excellent divertissement pour toute la famille. (Cl. Ian Dobbie.)

Une bonne aventure ne vous révélera tous ses secrets qu'après des heures, voire des semaines de recherche. Elle doit vous permettre de stocker sur cassette ou sur disquette l'état de vos recherches. Vous pouvez ainsi reprendre le jeu où vous l'avez laissé lors de la dernière séance, ou sauvegarder les étapes déjà franchies avant d'entreprendre un passage dangereux, évitant ainsi d'avoir à reprendre au début en cas de difficultés lors de ce passage.

Même la mort de l'aventurier ne signifie pas la fin du voyage. Certains auteurs vous permettent de revenir à la vie moyennant une perte de points et l'attribution d'une position moins favorable dans le jeu.

Comment participer à l'aventure, comment communiquer avec le programme? Vous pouvez être sollicité directement ou représenté par un personnage que vous commandez. L'ordinateur est à la fois interprète de vos désirs et narrateur. Le joueur entre ses commandes au clavier et l'ordinateur répond à l'écran.

Certaines aventures n'affichent que du texte à l'écran, certaines sont essentiellement graphiques, d'autres sont un compromis entre ces deux genres. Les effets sonores sont surtout utilisés dans les aventures graphiques. Les aven-

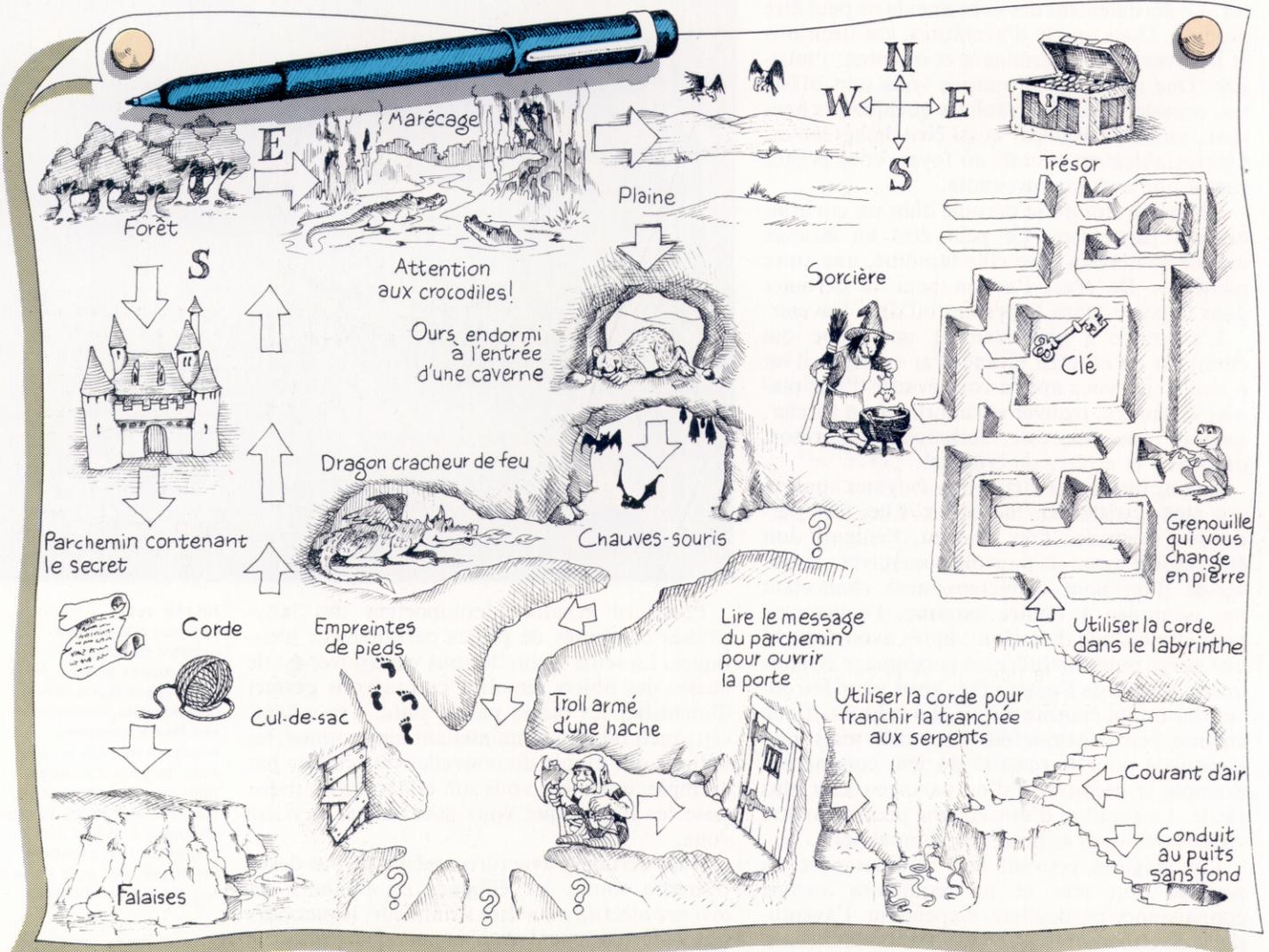
tures à texte sont comparables à un livre sans illustration où les lieux, les objets et les événements sont décrits avec des mots. Les graphiques dans les jeux texte-graphiques servent à compléter les descriptions textuelles. Dans les aventures strictement graphiques, les images décrivent à elles seules l'environnement, les personnages et les objets. Ces derniers n'offrent au joueur que quelques commandes qui servent à diriger un personnage.

L'affichage textuel d'une aventure donne généralement trois éléments d'information : où vous êtes, ce que vous voyez et où vous pouvez aller. Par exemple : « Vous êtes dans une forêt. Le ciel est voilé par un épais feuillage. Vous êtes sur un sentier allant d'est en ouest, devant l'entrée d'une caverne. Vous pouvez voir une épée autour de laquelle s'est enroulé un serpent. » Le texte vous présente votre environnement, les directions que vous pouvez prendre et les objets qui sont devant vous.

Les commandes consistent généralement en deux mots, un verbe et un nom, quoique les programmes les plus sophistiqués comprennent des phrases complètes. Voici certains verbes utilisés fréquemment : CHERCHER, POUSSER, TIRER, LAN-CER, ÉCLAIRER, TUER, MANGER et BOIRE. ALLER AU NORD,

Lecture de carte

Dans un jeu d'aventures, le joueur ou le personnage qu'il représente doit se déplacer sur un vaste territoire. Certains auteurs donnent des indices dans le manuel d'instructions afin d'aider les joueurs vraiment perdus. De tels jeux peuvent durer plusieurs semaines, il est donc essentiel de composer un croquis illustrant les endroits visités et les obstacles rencontrés. (Cl. David Higham.)



par exemple, est une façon de spécifier le mouvement, des abréviations peuvent être utilisées.

EXAMINER est un verbe essentiel, il permet d'acquérir plus de renseignements. EXAMINER SERPENT peut entraîner la réponse « c'est une couleuvre », ou encore « le serpent s'apprête à vous attaquer ». Certains verbes ne nécessitent pas de nom.

INVENTAIRE sert à vous annoncer quels objets vous transportez. Certains objets peuvent être contenus dans d'autres (une hache dans un sac), d'autres peuvent être portés (un anneau, une veste).

POINTAGE indique le nombre d'objectifs atteints. Taper AIDE peut entraîner l'affichage d'un indice, mais plus souvent le programme ne fait que vous donner un simple encouragement. Parfois certaines commandes sont refusées en raison du contexte. Il importe de trouver les commandes qui correspondent au problème du moment. « Je ne vous comprends pas » est la réponse généralement donnée par le programme quand vous lui transmettez des commandes qu'il ne reconnaît pas.

La plupart des aventures sont livrées avec une documentation qui fournit de subtils indices.

Certaines aventures n'entrent pas entièrement dans la mémoire de l'ordinateur. Pour résoudre ce problème, elles sont stockées sur disquette; le programme principal est chargé en mémoire et les divers messages et graphiques sont appelés à tour de rôle. Mais, grâce à de meilleures techniques de programmation en code machine, une aventure assez complexe peut maintenant être stockée en mémoire, ce qui signifie que de telles aventures peuvent être livrées sur cassette.

Des aventures sont offertes pour presque tous les micros, et peuvent vous apporter de nombreuses heures de plaisir.

Pour mieux comprendre la mécanique d'un jeu d'aventures, prenons un exemple parmi les très nombreux logiciels existant sur le marché : Wizardry, distribué en France par Sivéa et dont une version française a été mise au point par Hachette. Dans ce jeu, la part réservée au visuel est très faible et se limite aux silhouettes des personnages et à leurs déplacements (en perspective!) dans le labyrinthe d'un château diabolique. Tous les autres renseignements nous sont fournis par le texte, affiché à l'écran. L'avantage de Wizardry est qu'il peut se jouer à plusieurs ou seul face à la machine.

Avant de se lancer dans l'aventure, chaque joueur doit créer le personnage qui le représentera tout au long de la partie. Ces êtres possèdent tous des caractéristiques propres qui les différencient les uns des autres et qui obligent les participants humains à imaginer des stratégies en fonction du profil des personnages. Ainsi, un moine n'a pas les mêmes pouvoirs qu'un guerrier et un évêque qu'un voleur. Mais les dons de chaque être se complètent au sein des équipes qui doivent se former pour affronter les périls pour trouver un trésor. Une fois l'équipe constituée, tout ce petit monde part à l'assaut du château, sans posséder ni l'expé-



Chasse au trésor

The Hobbit est une aventure graphique pour le Sinclair Spectrum et est inspiré de l'ouvrage de J.R.R. Tolkien, dont un exemplaire est fourni avec la cassette. Vous vivez les aventures de Bilbo le Hobbit et vous rencontrez plusieurs des personnages du livre à la recherche du dragon et de son trésor. (Cl. Ian McKinnell.)

rience nécessaire ni les armes indispensables. En effet, une des caractéristiques importantes de ce jeu est la capacité qu'ont les personnages d'acquérir de l'expérience au fur et à mesure du déroulement de la partie, et de retenir les leçons de leurs échecs ou de leurs réussites. Ainsi, progressivement, le guerrier deviendra plus valeureux, le voleur plus adroit et le moine plus efficace dans ses prières. Face aux dangers toujours changeants que recèle le labyrinthe, cette faculté vous sera précieuse. Très vite, vous oublierez le caractère purement artificiel des créatures qui vivent sous vos yeux et vous vous identifieriez à l'une d'entre elles, avec laquelle vous partagerez toutes les péripéties du jeu. N'est-ce pas une belle aventure?



Enquête

Dans Deadline, vous jouez le rôle d'un détective qui doit trouver un meurtrier. Votre dossier renferme des informations telles que les résultats de l'autopsie, certaines pilules trouvées près du corps, et d'autres éléments d'enquête. Le jeu n'est pas joué en temps réel (il dure au moins deux semaines), mais chaque intervention, comme fouiller des pièces ou interroger un suspect, utilise des minutes précieuses des douze heures de recherches qui vous sont allouées. (Cl. Ian Dobbie.)

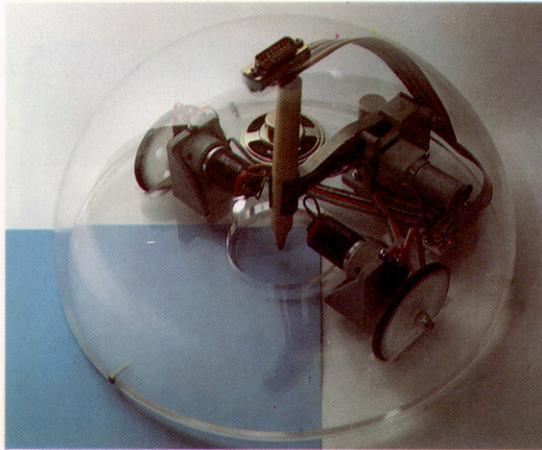


Langage pour enfants

Les ordinateurs sont largement utilisés à l'école, mais la plupart des programmes éducatifs ne sont que des livres électroniques. Le LOGO est différent; il offre à l'enfant un environnement pédagogique.

Tortue

La tortue est un outil de réflexion particulièrement utile pour assimiler des notions de géométrie et de relations dans l'espace. Lorsque les enfants ne sont pas certains des commandes qu'ils doivent envoyer à la tortue pour lui faire effectuer une manœuvre particulière, ils ont tendance à simuler eux-mêmes l'effet des instructions LOGO en marchant dans la pièce. L'apprentissage devient alors une expérience plus réelle et plus vivante. (Cl. Ian McKinnell.)



Depuis l'apparition du premier micro-ordinateur en 1977, certains éducateurs reconnaissent rapidement son potentiel pédagogique. Bientôt toutes les écoles auront au moins une machine, et intégreront l'informatique dans le programme scolaire. Néanmoins, le micro n'a modifié que très peu les méthodes d'enseignement traditionnelles.

Cette lacune s'explique par le fait que de nombreux programmes offerts pour les ordinateurs domestiques souffrent d'un manque remarquable d'imagination. La plupart ne sont en fait que des « livres électroniques » qui ne proposent à l'élève qu'une série d'affichages (équivalant aux pages d'un livre) et vérifient l'acquisition des données fournies à l'aide d'une série de questions à choix multiples.

De tels programmes sont faciles à écrire sur un ordinateur domestique et comportent des graphiques, souvent animés, qui accompagnent le texte. Mais cela ne fait rien de plus qu'automatiser les méthodes pédagogiques existantes, sans exploiter réellement le potentiel de l'ordinateur.

Le LOGO est différent. Cette création du professeur Seymour Papert, du Massachusetts Institute of Technology (MIT), est « une philosophie de l'enseignement, et une famille de langages de programmation conçue pour mettre en œuvre cette philosophie ».

De nombreuses personnes croient que le LOGO est un langage de programmation et comparent ses commandes et fonctions avec celles du BASIC, ce qui les amène à conclure que le LOGO est un meilleur langage pour les débutants, ce qui n'est pas exact. Papert n'a jamais désiré que son système soit une méthode d'apprentissage de la programmation. Il l'a conçu comme un environnement grâce auquel

les enfants peuvent apprendre divers sujets, dont l'art d'apprendre.

Cette philosophie s'est inspirée dans une large mesure des travaux du psychologue suisse Jean Piaget, qui estime que l'enfant placé dans un environnement adéquat peut apprendre tout sujet par lui-même, un peu comme il apprend à marcher et à parler. Le travail de Piaget n'a jamais dépassé le stade théorique. Papert a mis en place l'environnement préconisé par Piaget.

Le fait que la majorité des adultes ont peur d'apprendre et ne sont pas séduits par l'acquisition de nouvelles connaissances démontre que les méthodes traditionnelles d'enseignement n'atteignent pas ce résultat. Dans son livre *Mindstorms - Children, Computers, and Powerful Ideas*, Papert indique que le symptôme le plus répandu de ce problème est la peur des mathématiques.

Cela s'explique par le fait que la plupart des matières sont enseignées de la même façon, alors qu'elles sont entièrement différentes. Les enfants apprennent à multiplier de la même manière qu'ils apprennent le nom des capitales, de façon machinale. Le processus d'apprentissage devrait être défini en fonction du sujet traité.

Papert voit l'apprentissage d'une nouvelle technique comme un hobby. Il attribue cette attitude à son enfance, lorsqu'il découvrit encore très jeune comment fonctionnaient les engrenages et qu'il appliqua la même approche chaque fois qu'il devait résoudre un problème. Einstein disait lui aussi que, pour comprendre quelque chose, il utilisait des notions qu'il avait acquises avant l'âge de cinq ans.

Ces notions de compréhension peuvent être acquises avec le LOGO, comme l'illustre notre exemple. Le principal outil du LOGO est sa tortue, qui donne à l'enfant un « instrument pour penser », un peu comme les engrenages pour Papert. Pour les jeunes enfants, la tortue se présente sous la forme d'un robot sur roues (voir page 176), qui est relié à un micro et peut être déplacé quand on tape des commandes en LOGO. La tortue possède généralement un stylo afin de tracer des dessins sur le sol, et peut être munie d'un haut-parleur et de détecteurs de collision qui lui permettent de contourner des obstacles.

Les enfants passent généralement de la tortue de sol à la tortue d'écran, des formes qui peuvent être déplacées à l'écran. La tortue est un dispositif puissant grâce auquel les enfants peuvent apprendre des notions de base d'espace et de géométrie.



Dessin d'une fleur

```
TO COURBE
REPEAT 80
  FORWARD 1
  RIGHT 1
END
COURBE
```



Cet exemple illustre comment le LOGO encourage un groupe d'enfants à résoudre un problème nouveau pour eux. (Cl. Liz Dixon.)

« Si nous voulons des pétales, nous devons dessiner des courbes. »
 « Mais la tortue se déplace toujours en ligne droite. »
 « Si elle se déplaçait sur une très courte distance, et tournait juste un peu, puis continuait à avancer, et ainsi de suite, ça ressemblerait à une courbe. »
 « D'accord, la distance la plus courte est 1, et l'angle le plus petit est 1°. Essayons cela 80 fois. »

```
COURBE
COURBE
```



« C'est exactement ce que nous voulions. »
 « Deux comme cela et nous avons notre pétale. »

```
TO PETALE
COURBE
RIGHT 90
COURBE
END
PETALE
```



« Ça ne ressemble pas à un pétale, que s'est-il passé? »
 « Ça n'est que la suite de l'autre courbe, nous devons lui dire d'aller dans une autre direction. »
 « Mais de quel angle devons-nous tourner? »
 « Essayons 90°, ça fonctionne souvent. »
 « Et utilisons un nouveau mot, PÉTALE, nous gagnerons du temps. »

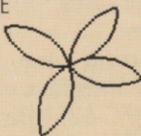
```
TO PETALE
COURBE
RIGHT 100
COURBE
END
PETALE
```



« C'est mieux mais 90° n'est pas assez, quel angle devons-nous essayer? »
 « Au lieu d'essayer de deviner, faisons d'autres essais. »
 « Oui, rappelez-vous que pour faire un tour complet, la tortue doit tourner de 360°. »

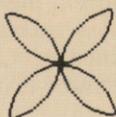
« Écoutez, nous savons qu'elle a tourné de 80° pour la première courbe, et pour revenir à son point de départ elle doit faire demi-tour, donc tourner de 180° en tout. »
 « 180 - 80, nous devons donc tourner de 100° au bout du pétale. »

```
PETALE
PETALE
PETALE
PETALE
```



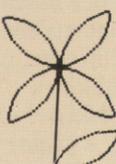
« Fantastique, quatre comme cela et nous avons notre fleur. »

```
TO FLEUR
REPEAT 4
  PETALE
  RIGHT 10
END
FLEUR
```



« Ce n'est pas parfaitement symétrique. »
 « Nous avons oublié d'indiquer de combien tourner entre les pétales. »
 « Mais pourquoi alors les pétales n'ont-ils pas été dessinés les uns sur les autres? »
 « Parce que la tortue, après avoir dessiné un pétale, a tourné de 100° vers la gauche par rapport à son point de départ. »
 « Les pétales tournent donc de 100° vers la gauche à chaque fois. »

```
FLEUR
RIGHT 180
FORWARD 100
RIGHT 180
PETALE
```



« Je crois; ce que nous voulons c'est 90°, tournons donc de 10° vers la droite entre chaque pétale. »
 « Enfin! Faisons tourner la tortue pour dessiner la tige. »
 « Ça devrait être 180°. »
 « Plaçons une feuille à la fin de la tige, utilisons un pétale. »
 « N'oublions pas qu'elle doit encore tourner de 180°. »

La commande de la tortue n'est qu'une application du LOGO, mais puisque c'est visuellement l'aspect le plus intéressant du système, c'est celui qui retient le plus l'attention. Il est beaucoup plus important de souligner les notions d'analyse et de synthèse qu'il communique aux enfants : partir d'une idée simple pour aboutir à une idée générale ou inversement. Le LOGO permet aux enfants de se familiariser avec ces concepts.

Ce processus peut être clairement illustré par une conversation fictive d'un groupe d'enfants apprenant à dessiner une fleur avec la tortue (voir encadré). Ils débutent avec uniquement trois commandes disponibles : FORWARD, qui déplace la tortue en ligne droite sur une distance spécifiée; RIGHT, qui fait tourner la tortue d'un angle spécifié; et REPEAT, qui répète les lignes préprogrammées un nombre de fois spécifié par l'ordinateur.

Les enfants construisent d'abord un « outil » — un programme — qui dessine une courbe (TO COURBE... END). Cette séquence peut maintenant être appelée simplement en tapant COURBE. De façon similaire, après expérimentation et période d'apprentissage, une commande PÉTALE est définie et utilise la commande COURBE. Finalement, une commande FLEUR effectue le dessin au complet.

Le LOGO n'est pas le seul langage qui comporte de telles structures (il y a aussi le FORTH, voir page 150), mais c'est le seul qui soit destiné à de jeunes enfants. Il ne comporte pas les formalités et les procédures associées à d'autres langages. L'objectif est d'amener l'enfant à comprendre le processus de résolution d'un problème.

Dans certaines situations, l'élève n'est même pas impliqué à ce niveau. L'enseignant prépare une série d'outils tous associés à un sujet particulier ou à un domaine de connaissances. L'enfant peut alors, de façon autonome, explorer le sujet à l'aide de ces outils. On crée ainsi des « micro-univers », des environnements dans lesquels l'ordinateur sert à simuler le monde réel ou un champ de connaissances.

Le meilleur exemple de micro-univers est probablement le modèle de physique newtonienne du LOGO. Même si la première loi de Newton énonce que, sans l'influence de forces externes, un corps en mouvement se déplacerait indéfiniment en ligne droite à une vitesse constante, les jeunes enfants observent dans le monde réel le ralentissement éventuel de tout mouvement. Cela gêne leur compréhension. A l'aide du LOGO, il est possible de créer un micro-univers qui obéit parfaitement aux lois newtoniennes. Avec l'aide d'outils, l'enfant peut pousser des objets sur l'écran et bientôt apprendre par lui-même les trois lois de Newton.

Le LOGO est un système puissant qui a tout pour intéresser les utilisateurs d'ordinateurs domestiques. Les tortues de sol sont disponibles mais ne sont pas bon marché. Les ordinateurs domestiques les plus populaires offrent eux aussi des versions du LOGO qui utilisent des tortues sur écran.



L'ère des portables

Plus sophistiqués et de moins en moins encombrants, certains ordinateurs sont maintenant devenus véritablement portables.



Un outil de travail

Les ordinateurs portables comme l'Epson HX-20, qui ont jusqu'à 32 K de mémoire, sont munis de périphériques intégrés et d'une vaste gamme de programmes, ont grandement accru la quantité de données dont peut disposer un homme d'affaires et ce quel que soit l'endroit où il se trouve. Chaque microcassette peut contenir jusqu'à 230 K (environ 40 000 mots, soit l'équivalent d'une puissante base de données). Il est ainsi possible d'utiliser des programmes de traitement de texte ou de feuilles de calcul électronique presque partout, même dans un taxi ! (Cl. Ian Dobbie.)

Le développement du micro portable a deux origines. L'une est la sophistication du calculateur de poche, comme le Sharp PC-1251 et le Casio FX-700P. L'autre est l'évolution des processus de miniaturisation qui a donné des machines comme l'ordinateur individuel Tandy TRS-80 modèle 100 et l'Epson HX-20 (voir page 169).

Ces progrès sont dus au développement de puces plus denses qui permettent le traitement de plus de données dans le même espace physique.

Quand, à partir de 1972, les microprocesseurs furent logés sur une seule puce, il devint théoriquement possible de construire un ordinateur complet dans un boîtier pas plus grand qu'un paquet de cigarettes. Cependant, la dimension de l'écran et la présence du clavier imposèrent certaines restrictions à la miniaturisation.

Les calculatrices de poche devinrent en fait plus petites, et aujourd'hui certaines montres numériques comportent toutes les touches d'une véritable calculatrice. Mais la plus simple opération arithmétique est assez difficile à effectuer sur ces montres, et même la personne la plus enthousiaste renoncerait à entrer un programme BASIC de cinquante lignes. Il est donc peu probable que les ordinateurs personnels

atteignent cette dimension. Mais ils ont atteint la taille de la calculatrice de poche.

Dès que les calculatrices de poche devinrent programmables dans leur propre langage, on pensa aussitôt à intégrer un langage de programmation évolué, et le choix du BASIC s'imposa. Environ au même moment, les fabricants commencèrent à utiliser des RAM rémanentes — une mémoire qui, en maintenant une petite charge électrique, ne perd pas son contenu lors de sa mise hors tension — et un générateur de caractères plus complet qui permit l'affichage de caractères alphabétiques.

Les ordinateurs de poche peu coûteux, comme le Sharp et le Casio, qui possèdent une gamme de commandes BASIC comparable à celle des ordinateurs domestiques ainsi qu'une capacité de mémoire similaire, envahissent progressivement le marché de la calculatrice programmable. Pouvant être logé dans une poche — même une poche de chemise —, certains offrent la possibilité d'être branchés directement à une imprimante ou à une unité à cassette. Et bientôt, si la demande est suffisante, une interface de communication sera disponible, permettant ainsi le transfert d'information *via* le réseau téléphonique. La gamme Casio intéresse particulièrement les scientifiques et les ingénieurs, puisqu'elle offre de nombreuses fonctions mathématiques et scientifiques.

D'autres modèles comportant une imprimante intégrée et munis d'une interface cassette coûtent environ 1 500 F. Le PC-1251 de Sharp offre des caractéristiques similaires à ce prix.

On retrouve dans la catégorie suivante des ordinateurs comme l'Epson HX-20, le Tandy 100 et le NEC 8220. Ces micros offrent un BASIC complet (Microsoft), 16 à 64 K de RAM, un affichage intégré à cristaux liquides de dimensions raisonnables (20 × 4 caractères sur l'Epson, 40 × 8 sur les deux autres), et la possibilité de connecter une vaste gamme de périphériques standards.

Évidemment, en les couplant à un moniteur normal (vous devrez peut-être acheter une interface), vous obtenez un micro dont les possibilités sont comparables à un ordinateur domestique de même prix. La différence essentielle de ces machines réside dans le fait qu'elles possèdent leur propre alimentation interne, ce qui les rend ainsi véritablement portables.

L'Epson, par exemple, possède des piles intégrées au nickel-cadmium qui, à pleine charge, autorisent une autonomie d'environ cinquante



heures; le système s'arrête automatiquement lorsque les piles sont trop faibles, de façon à protéger les données essentielles.

Par contre, le NEC et le Tandy utilisent des circuits MOS complémentaires qui consomment moins de courant et qui permettent l'emploi de piles ordinaires.

L'Epson, qui fut l'objet d'une vaste campagne de publicité, était initialement destiné au gestionnaire, mais il est probable qu'on le rencontre plus souvent à l'extérieur qu'à l'intérieur des bureaux. Il est parfait pour la saisie de données en usine ou sur les chantiers; ces données peuvent être traitées ultérieurement au bureau.

L'Epson permet divers modes de transmission de données. Le constructeur propose des dispositifs complémentaires permettant d'utiliser le micro comme un terminal de télex; à l'aide d'un modem, il peut être relié à d'autres machines *via* le réseau téléphonique; ou ses données peuvent être simplement et économiquement stockées sur cassette et envoyées par la poste.

Cette souplesse a rendu ces machines populaires. Par exemple, un vendeur qui effectue un minimum de vingt visites par jour simplement pour prendre des commandes peut utiliser ce type de micro au lieu de remplir des imprimés. Le programme de saisie de commandes lui proposerait une liste de produits où il n'aurait qu'à taper la quantité désirée par le client. A la fin de chaque visite, les données de la commande seraient stockées sur bande, et à la fin de la journée le vendeur pourrait poster cette bande à son bureau ou transmettre les données à l'ordinateur de sa société.

L'imprimante intégrée du micro fournirait au vendeur une copie des commandes de la journée et au client une confirmation de commande immédiate.

Certaines versions plus sophistiquées pourraient aussi consulter le niveau des stocks et annoncer au vendeur un niveau devenu trop bas. Mais, s'il y avait plusieurs vendeurs, le micro devrait être connecté à l'ordinateur central lors de chaque visite. Un simple coupleur acoustique portable permet d'établir cette communication. Notre ordinateur portable devient alors un terminal interactif qui peut interroger la base de données de la société et obtenir une réponse immédiate.

En plus des avantages évidents de cet accès immédiat aux dernières mises à jour, les économies réalisées par ce type de saisie en ligne peuvent rentabiliser en quelques mois un système portable complet.

Tout comme les petits ordinateurs de poche ont supplanté les calculatrices programmables, ces machines portables remplacent actuellement les dispositifs traditionnels de saisie de données. Ces « terminaux non intelligents » alimentés par piles, qui ne peuvent être programmés, ont été pratiques mais jamais très populaires en raison de leur coût élevé et de leur complexité.

Avec un logiciel adéquat, connectées à des moniteurs et à des imprimantes de qualité pro-

fessionnelle, et éventuellement équipées de lecteurs de disquettes ou d'unités à cassette plus rapide, de telles machines deviendront probablement de plus en plus populaires. Ce type d'ordinateurs constitue le premier véritable marché pour les mémoires à bulles — une mémoire autonome, rémanente, qui offre une capacité énorme de stockage dans un petit boîtier. Certaines applications de cette technologie, comme le PC-5000 de Sharp, offrent 128 K de stockage par cartouche enfichable, avec un accès beaucoup plus rapide que sur disquettes. On peut déjà envisager la conception de telles machines munies d'une mémoire d'un million d'octets.

Cette capacité de stockage est déjà offerte par le troisième type de machines portables qui comprend l'Osborne Executive, l'Ajile Hyperion et le Portico Miracle. Coûtant jusqu'à cinq

Pour toutes les saisons

(Sens des aiguilles d'une montre, à partir du coin supérieur droit.)
L'Epson HX-20, le premier des ordinateurs autonomes, offre jusqu'à 32 K de RAM et une vaste gamme de périphériques. Moins puissant et moins coûteux, le PS-1251 de Sharp est muni ici d'une imprimante et d'une unité à microcassette. Le Casio FX-700P utilise une unité à cassette domestique standard. Le modèle Executive de la société Osborne offre un écran plus grand et d'autres perfectionnements.
(Cl. Ian McKinnell.)



ou six fois le prix du HX-20 ou du TRS-100, elles sont plutôt dites « transportables », puisqu'elles doivent être branchées au secteur. Des batteries sont disponibles pour certaines machines, mais elles offrent rarement plus d'une heure de fonctionnement.



Écriture électronique

Voici le Microwriter, un appareil de traitement de texte révolutionnaire. Les six touches pressées dans diverses combinaisons produisent toutes les lettres de l'alphabet. La sortie peut être envoyée sur imprimante, avec certaines commandes de mise en pages, ou vers un système de traitement de texte conventionnel afin d'y être stockée. (Cl. Ian McKinnell.)



Les machines au travail

Des ordinateurs portables peu coûteux sont utilisés dans des endroits auparavant inaccessibles aux systèmes informatiques.

Les journalistes peuvent composer le texte de leurs articles sur le lieu du reportage et même le



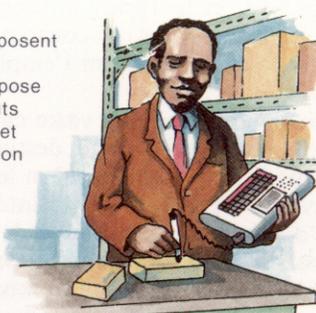
modifier dans le train ou dans l'avion pendant le voyage de retour.



Les pharmaciens disposent d'un outil simple et économique qui compose les étiquettes des produits délivrés sur ordonnance et qui se charge de la gestion de leurs stocks.



Les vendeurs peuvent préparer instantanément des devis impliquant des calculs complexes.



Le contrôle des stocks est très important même dans une petite société, mais dans l'industrie, où la valeur des stocks atteint plusieurs millions de francs, il est essentiel de maintenir une gestion efficace et précise. (Cl. David Higham.)

Ces ordinateurs ont généralement deux lecteurs de disquettes, un moniteur intégré, un clavier et un système d'exploitation comme CP/M.

L'Osborne 1 fut le premier modèle sur le marché et offrit pour moins de 30 000 F une gamme complète de programmes, incluant BASIC, WordStar et SuperCalc avec les utilitaires CP/M.

Son point faible était les dimensions de son écran — 100 × 85 mm — qui affichait des caractères plutôt petits et exigeait l'utilisation du défilement horizontal pour l'emploi de logiciels comme WordStar. Cela ne sembla pas trop gêner les utilisateurs qui firent de cette machine un succès commercial.

Certains concurrents proposèrent des machines plus sophistiquées; si bien que cette catégorie de machines rejoignit, quant aux performances, les systèmes de bureau. Souvent elles offrent une entière compatibilité avec d'autres machines, notamment l'IBM PC, ce qui encourage leur emploi comme « second ordinateur ».

La seule différence notable entre ces systèmes portables et les modèles de bureau se situe au niveau du volume qu'ils occupent. En effet, si un programme d'application fonctionnant sur un système normal tournera bien sur un portable muni du même microprocesseur, il n'en sera pas de même en ce qui concerne la sortie des informations. Le système portable restera limité par les dimensions réduites de l'écran et, en conséquence, par le nombre de caractères affichés.

La puce microprocesseur est responsable de cette miniaturisation. Les restrictions provenant de l'accessibilité d'entrée et de sortie illustrent notre dépendance vis-à-vis du clavier et du moniteur. Mais si nous pouvions utiliser la

reconnaissance et la synthèse de la parole? Il n'y aurait alors aucune raison pour qu'un ordinateur de la puissance de l'Ajile Hyperion, par exemple, occupe plus de volume qu'un Walkman. Si nous disposions de la technologie nécessaire, il se présenterait probablement ainsi.

La technologie progresse actuellement à un tel rythme que les développements dans ce domaine seront probablement considérables, même si nous abandonnons la notion encore utopique d'entrée/sortie vocale. Imaginons par exemple un système composé du Sinclair ZX Spectrum, de l'écran plat Sinclair et du Microdrive, le tout alimenté par un bloc d'alimentation rechargeable au nickel-cadmium : nous pourrions déjà composer ce système, mais Sinclair le fera peut-être pour nous...

En raison de sa taille, le ZX Spectrum est parfaitement adapté à cette application, mais la plupart des ordinateurs utilisent le type de tension fourni par les piles sèches standards : tout ordinateur pourrait donc devenir portable de cette façon.

Une tendance intéressante est exprimée dans la conception de l'ACT Apricot, un système de bureau normal qui permet de travailler de façon autonome sans moniteur et qui devient ainsi un système portable. Le qualificatif « portable » est bien sûr arbitraire. Certains micros qui sont dits tels ne sont même pas munis d'une poignée et n'ont pas été conçus en véritables unités autonomes.

Avec l'introduction d'une vaste gamme de micros portables, l'industrie informatique s'est approchée un peu plus de son réel potentiel : rendre l'ordinateur accessible à tous tant au niveau du coût qu'au niveau des connaissances nécessaires à l'utilisateur.



Epson HX-20

Non seulement cet ordinateur peut être transporté n'importe où, mais il peut être utilisé pour travailler en dehors du bureau.

L'Epson HX-20 fut le premier ordinateur véritablement portable. Puisqu'il possède un BASIC complet, il permet beaucoup plus d'applications que les calculatrices les plus sophistiquées. De plus, ses dimensions et son poids lui permettent d'être facilement transporté dans une mallette.

Des machines beaucoup plus grosses qui avaient été jusqu'alors vendues comme « portables » perdirent rapidement ce qualificatif au sein de la profession. Le HX-20 a séduit une vaste gamme d'utilisateurs : utilisateurs domestiques, hommes d'affaires, ingénieurs.

Puisque la conception du HX-20 est entièrement nouvelle, peu de programmes sont disponibles. Cependant, c'est un appareil idéal pour apprendre à programmer en BASIC. Souvent l'intention du client est d'écrire un programme qui exécutera une tâche spécifique (peut-être inhabituelle) : il existe quand même quelques programmes d'application pour l'Epson.

L'unité est munie d'un affichage à cristaux liquides qui offre quatre lignes de vingt caractères, ou une résolution maximale de 120 x 32. Dans la plupart des applications, cet affichage sert de fenêtre qui peut être déplacée à l'aide des touches du curseur (touches fléchées) afin d'afficher toute autre section d'un texte beau-

coup plus large pouvant être géré par l'ordinateur.

L'imprimante intégrée utilise des rouleaux de papier ordinaires de 5 cm de largeur, sur lesquels elle peut imprimer jusqu'à vingt-quatre colonnes de texte ou dessiner des graphiques du plus bel effet.

L'unité à microcassette est offerte en option, mais la plupart des personnes la demandent. L'emplacement qu'elle occupe peut loger des cartouches logiciel, quoique aucune n'ait été produite jusqu'ici. L'unité à microcassette est supérieure à une unité à cassette domestique. L'ordinateur peut détecter la position de la bande en avance rapide et ainsi trouver très vite le programme ou les données recherchés.

La variété d'interfaces que comportent le côté et l'arrière du boîtier reflète la diversité des applications possibles; il y a même une prise pour un crayon-lecteur de codes à barres (voir page 40). Les 16 K de RAM peuvent être portés à 32 K à l'aide d'un boîtier latéral.

Matériel et logiciel peu coûteux permettent de communiquer *via* le réseau téléphonique avec d'autres machines similaires ou avec de gros ordinateurs afin d'accéder aux bases de données centrales.



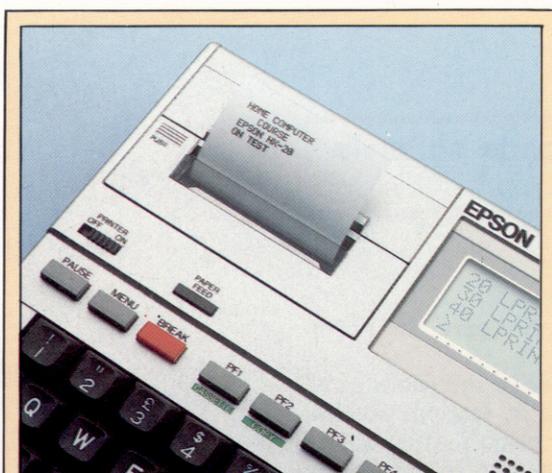
Clavier du HX-20

C'est le clavier du HX-20 qui détermine en fait ses dimensions. Bien que son toucher soit légèrement différent de celui d'une machine à écrire, son utilisation est confortable et efficace.

En plus des touches numériques, la touche NUM convertit les touches U, I, O, J, K, M en un clavier numérique. Cela permet d'accélérer les saisies de données numériques.

Les touches de déplacement du curseur et d'autres touches d'édition sont situées dans le coin supérieur droit; on retrouve également une touche SCRN qui permet le défilement de l'écran vers le haut ou vers le bas.

Les cinq touches de fonction programmables (PF1 à PF5) diffèrent physiquement des autres touches. (Cl. Chris Stevens.)



Imprimante intégrée

Les imprimantes matricielles sont maintenant parties intégrantes des caisses enregistreuses. Epson mettant à profit son expérience de constructeur d'imprimantes décida d'en intégrer une dans son HX-20. Cette imprimante imprime des caractères et peut également tracer des graphiques.



Prise d'alimentation

Le bloc d'alimentation branché à cet endroit fournit l'alimentation du HX-20 et recharge ses piles internes.

Touche Reset

Connecteur de lecteur de codes à barres

Un crayon-lecteur de codes à barres peut être branché à cet endroit.

Port cassette

Permet d'utiliser une unité à cassette domestique. L'efficacité n'est alors pas comparable à celle de l'unité à microcassette.

Carte de circuits imprimés

La carte a été photographiée à l'envers, la carte est montée de l'autre côté, ce qui permet d'accéder facilement aux puces importantes en ne démontant que le bas du boîtier.

Les microprocesseurs

Deux microprocesseurs 6301 (fabriqués par Epson) commandent l'ordinateur et ses interfaces. Chacun de ces microprocesseurs a 4 Koctets de ROM et 128 octets de RAM, en plus des puces externes.

RAM

Le HX-20 en version standard est livré avec une mémoire RAM de 16 K, configurée en huit puces de 2 K.



Stockage de masse

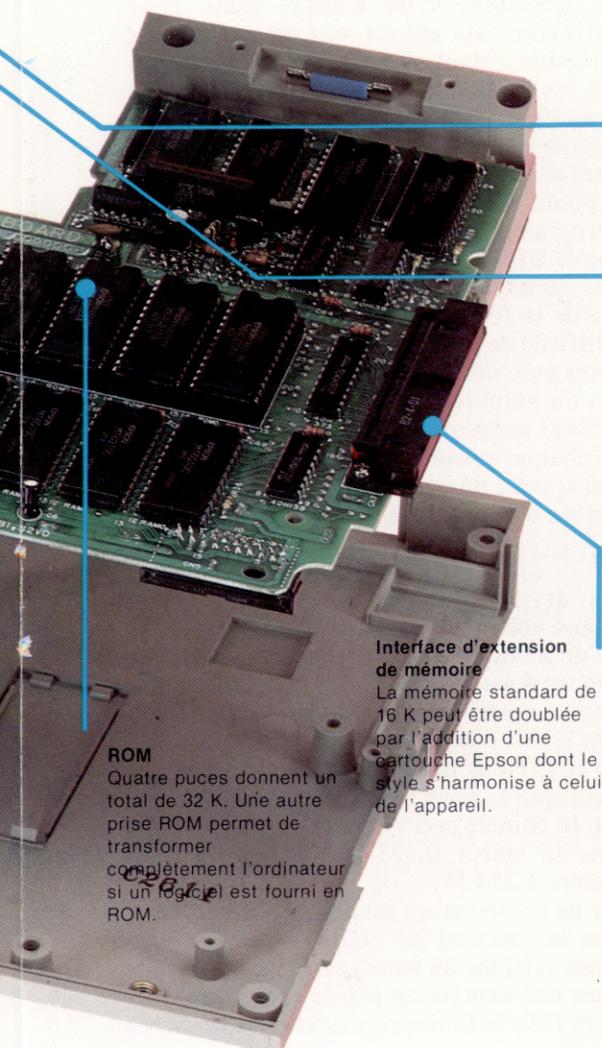
L'unité à microcassette intégrée emploie les cassettes utilisées dans les appareils à dicter et est beaucoup plus efficace que les unités à cassette domestiques puisque sa fonction « avance rapide » est commandée par l'ordinateur. Le temps de recherche des données est donc beaucoup plus court. Le fait que l'unité soit intégrée et donc non encombrante constitue un autre avantage.

**Bouton pour cartouches**

Ce bouton permet la dépose de l'unité de cassette. Théoriquement, ce module pourrait être remplacé par une cartouche si vous ne désirez utiliser l'appareil que pour une seule application.

Réglage de l'angle de vision

Les affichages à cristaux liquides doivent être regardés sous un certain angle pour être visibles. Ce bouton permet à l'utilisateur de régler cet angle.

Interrupteur d'alimentation**Interface série**

Bien que ce port soit aussi une interface série, il est destiné à divers périphériques qu'Epson introduira pour le HX-20.

Port RS232

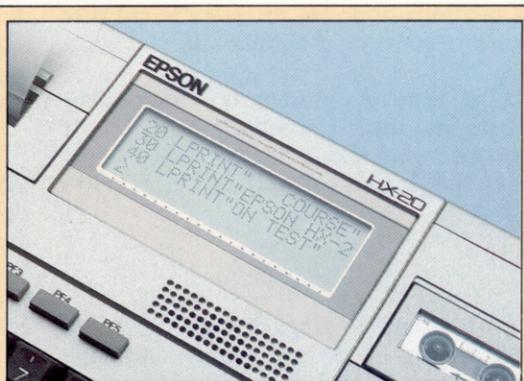
Norme de connexion d'interface série, ce port est utilisé pour connecter une imprimante ou un modem afin de communiquer avec un autre ordinateur via les lignes téléphoniques.

Interface d'extension de mémoire

La mémoire standard de 16 K peut être doublée par l'addition d'une cartouche Epson dont le style s'harmonise à celui de l'appareil.

ROM

Quatre puces donnent un total de 32 K. Une autre prise ROM permet de transformer complètement l'ordinateur si un logiciel est fourni en ROM.

**Inclinaison de l'affichage**

La majeure partie du volume occupé par un micro-ordinateur est consacrée au tube cathodique. Epson décida de contourner ce problème en utilisant un affichage à cristaux liquides. Ce type d'affichage est utilisé depuis longtemps sur les calculatrices, mais celui-ci est beaucoup plus confortable puisque l'angle de vision est ajustable.

EPSON HX-20**PRIX**

6 000 F ou 7 000 F avec unité à microcassette.

DIMENSIONS

290 x 210 x 45 mm.

POIDS

1,7 kg.

UC

6301 Hitachi.

HORLOGE

2,45 MHz.

MÉMOIRE

16 K de RAM, extension jusqu'à 32 K à l'aide d'une unité latérale. 32 K de ROM, pouvant être portée à 64 K de la même façon.

AFFICHAGE

Affichage à cristaux liquides (angle de vision ajustable) avec 4 lignes de 20 caractères agissant comme une fenêtre sur une zone d'affichage de 255 x 255 caractères. Chaque point de l'écran peut être adressé, ce qui donne une résolution graphique maximale de 120 x 32.

INTERFACES

Port d'extension de mémoire, cassette, RS232 (pour imprimante externe), série (pour unité de disquettes disponibles bientôt), crayon-lecteur de codes à barres.

LANGAGE INTÉGRÉ

BASIC.

AUTRES LANGAGES DISPONIBLES

Interpréteur FORTH.

ACCESSOIRES FOURNIS

Boîtier de voyage, unité d'alimentation, manuel d'instruction.

CLAVIER

60 touches mécaniques, 5 touches de fonction programmables et 4 boutons de fonction dédiés.

DOCUMENTATION

L'ordinateur est livré avec deux manuels : un manuel d'utilisation et un manuel de référence BASIC. Les deux sont complets mais ne comportent pas d'index. Epson décida d'intégrer le BASIC de Microsoft, et le manuel du langage est probablement assez bien fait pour servir à la fois de manuel d'instruction général et de guide de référence.

Le manuel d'utilisation fournit des données spécialisées comme une liste complète des connexions E/S, des interfaces RS232 et série.

Nombres aléatoires

Nous continuons notre étude des fonctions BASIC. Voyons cette fois la fonction RDN; cette fonction produit des nombres aléatoires utilisés dans les jeux ou dans les programmes statistiques.

Après avoir découvert plusieurs fonctions BASIC, examinons l'une des plus fréquemment utilisées, la fonction RDN. RDN sert à générer des nombres aléatoires et à introduire le hasard dans les jeux.

En effet, dans les programmes les plus rationnels, il peut être nécessaire d'introduire le hasard. Par exemple, pour simuler les paramètres de l'évolution d'une situation météo dans un simulateur de vol, ou bien déclencher des événements imprévisibles au cours d'un jeu d'aventures.

Il n'est pas facile de simuler les coups du sort. Il est possible que des nombres reviennent plus fréquemment que d'autres, dans une proportion infiniment suffisante pour altérer les résultats. C'est pourquoi il est difficile d'utiliser l'ordinateur au cours d'expériences scientifiques pour la parfaite simulation du hasard. Mais pour les besoins plus modestes d'un programme de jeu, il suffit de faire appel à des nombres quasi aléatoires qui satisfont les besoins limités de l'intrigue. Dans vos programmes, sachez introduire la fonction aléatoire qui permet de remplir d'inattendu vos logiciens.

Tentez de rédiger un programme qui vous permette de jouer aux dés avec votre ordinateur. Comment faire? Si vous ne savez pas, nous vous proposons dans cette leçon un programme type.

Malheureusement, RDN est un des mots les moins standards en BASIC. Notre description peut ne pas correspondre à la version de votre micro. Clarifions donc les différences entre le BASIC de notre cours de programmation et votre BASIC.

La plupart de nos programmes correspondent au BASIC de Microsoft (ou MBASIC). Microsoft est une société américaine et leur BASIC fut le premier à être largement diffusé. Il n'existe



aucune norme pour le langage BASIC, mais la version de Microsoft est presque devenue une norme. Microsoft a servi de modèle à de nombreuses autres versions, et cette société a produit des versions pour plusieurs ordinateurs populaires.

La principale différence entre le MBASIC et la plupart des versions plus récentes se situe au niveau des fonctions graphiques; les ordinateurs ont aujourd'hui un potentiel graphique

qui n'existait pas lors du développement du MBASIC. Pour exploiter entièrement le potentiel de votre machine et utiliser toutes ses fonctions graphiques, vous devez étudier soigneusement votre manuel de l'utilisateur.

Parmi les BASIC fournis avec les ordinateurs domestiques populaires, le Sinclair BASIC (utilisé sur le ZX-81 et sur le Spectrum) et le BBC BASIC sont ceux qui diffèrent le plus du MBASIC. La version de Texas Instruments (utilisée sur le TI-99/4A) a aussi de nombreuses différences significatives. Nous essayons, autant que possible, de vous indiquer comment modifier nos programmes dans les encadrés « Variantes de BASIC », afin de vous permettre de les adapter à votre ordinateur.

Comme nous l'avons mentionné précédemment, la fonction RDN diffère d'une version à l'autre. Consultez votre manuel BASIC pour découvrir comment elle est mise en œuvre dans votre version. Nous illustrons son utilisation dans un jeu de dés très simple. Comme dans les programmes précédents, presque tout le travail est effectué par des sous-programmes. Les programmes sont ainsi plus faciles à écrire et à mettre au point.

Le programme principal commence par l'instruction RANDOMIZE à la ligne 20. La plupart des versions de BASIC doivent redéfinir le nombre de base de la fonction RDN. Il est généralement assez difficile de produire de véritables nombres aléatoires avec des ordinateurs. Sans cette redéfinition du nombre de base, la même séquence de nombres supposés être aléatoires serait produite à chaque utilisation de la fonction RDN. La ligne 50 appelle un sous-programme qui affecte un nombre aléatoire à la variable D.

```
320 LET D = INT (10 * RND)
```

Vous devrez probablement modifier cette ligne dans votre programme. L'encadré « Variantes de BASIC » décrit le fonctionnement de diverses versions de RND. Mais voyons comment fonctionne notre version MBASIC. Dans l'instruction LET A = RND, la valeur de A sera comprise entre 0 et 1. Nous ne voulons pas d'un nombre plus petit que 1, nous multiplions donc le nombre par 10 comme ceci : LET A = 10 * RND. Si RND retourne la valeur 0,125 455, la valeur de A serait alors 1,254 55.

Pour ne conserver qu'un nombre entier, nous utilisons la fonction INT : LET A = INT (10 * RND). Certaines versions de BASIC permettent de définir, dans une expression placée entre parenthèses après RND, la limite supérieure des nombres

aléatoires générés. Par exemple, le BASIC du Dragon générera un nombre entier compris entre 1 et 6 en réponse à l'instruction : PRINT RND (6).

Puisque cela est impossible avec MBASIC, nous utilisons un test pour ne retenir que les nombres compris entre 1 et 6, nombres dont nous avons besoin dans notre jeu.

```
330 IF D > 6 THEN GOTO 320
340 IF D < 1 THEN GOTO 320
```

Si D n'est pas compris entre 1 et 6, l'instruction GOTO sollicite un nouveau nombre.

Après avoir produit une valeur aléatoire comprise entre 1 et 6, le sous-programme revient au programme principal. Cela produit le message VOUS AVEZ LANCÉ UN, suivi de l'illustration d'un dé. Notez comment l'image adéquate est sélectionnée. Le sous-programme SELECTION le fait. Par exemple, si 1 est choisi, la ligne 410 appelle le sous-programme de la ligne 530 :

```
410 IF D = 1 THEN GOSUB 530
```

Ce sous-programme n'est rien d'autre qu'une série d'instructions PRINT produisant un graphique rudimentaire. Votre BASIC peut produire de bien meilleurs graphiques; si c'est le cas, remplacez votre sous-programme par vos instructions graphiques.

Après avoir joué pour vous, le programme fait de même pour l'ordinateur. La partie du programme qui détermine qui est le vainqueur a été intégrée dans le programme principal; elle aurait pu être placée dans un sous-programme, mais sa longueur ne le justifiait pas. La ligne 200 teste l'égalité de M (mon dé) et de O (dé ordinateur). S'ils sont égaux, le mot ÉGALITÉ est affecté à la variable S\$. La ligne 210 affecte les mots VOUS AVEZ GAGNÉ à la variable S\$ si M est supérieur à 0. La ligne 220 affecte les mots L'ORDINATEUR A GAGNÉ à la variable S\$ si M est inférieur à 0. La ligne 240 ne fait qu'afficher le résultat et le jeu est terminé. Malgré sa longueur, ce programme est très simple. Il n'utilise qu'une fonction, RND, ne comporte aucune boucle, aucune variable indicée et rien de plus compliqué que quelques instructions IF-THEN.

En raison des nombreuses variantes de la fonction RND et puisque certaines versions (MBASIC par exemple) nécessitent l'instruction RANDOMIZE pour générer une nouvelle séquence de nombres aléatoires, nous devrions trouver une autre façon de générer des nombres véritablement aléatoires sans utiliser ces fonctions. Plusieurs approches sont possibles.

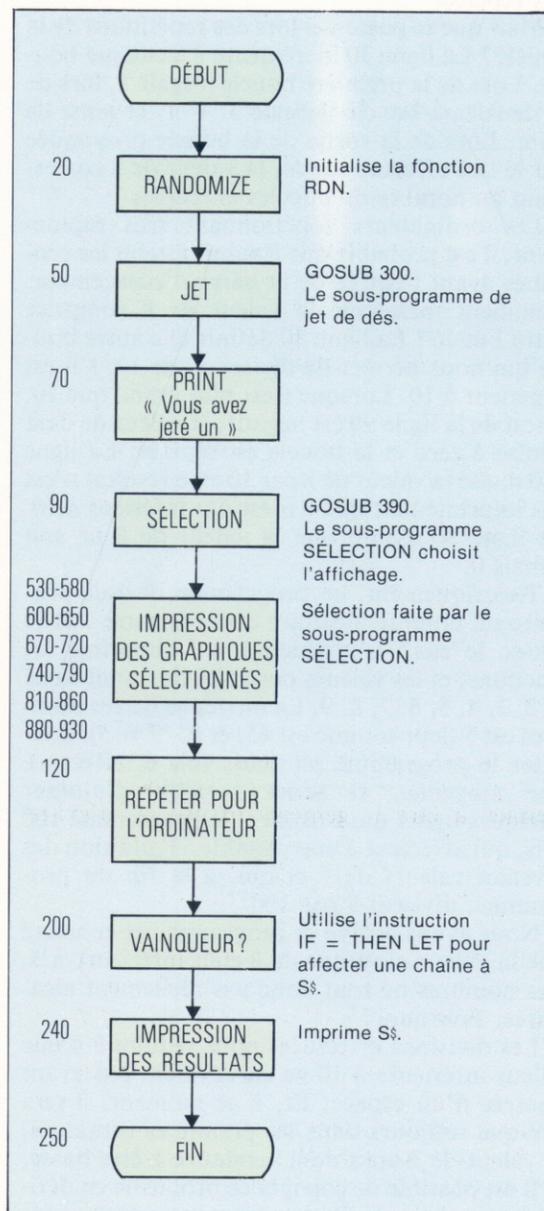
Nous ne nous sommes pas encore servis de la fonction INKEY\$. Chaque fois qu'il rencontre le mot INKEY\$, le programme recherche l'entrée d'un caractère au clavier. Le programme n'attend pas cette entrée comme il le fait avec une instruction INPUT. La fonction INKEY\$ est donc généralement placée dans une boucle. Le programme analyse alors continuellement le clavier, attendant l'entrée d'un caractère. Il y a généralement un test à l'intérieur de la boucle qui y met fin lorsque le caractère attendu est

entré. Cela permet d'écrire un programme comportant une boucle de comptage qui se terminera lors de l'entrée d'un caractère spécifique. Quel serait le résultat de ce programme ?

```
10 PRINT « APPUYEZ SUR LA BARRE D'ESPACEMENT »
20 FOR X = 0 TO 1
30 LET R = R + 1
40 LET A$ = INKEY$
50 IF A$ = « » THEN GOTO 80
60 LET X = 0
70 NEXT X
80 FOR Q = 0 TO 1
90 IF R < 10 THEN GOTO 130
100 LET Q = 0
110 LET R = R/10
120 NEXT Q
130 PRINT INT (R)
140 END
```

R est un nombre aléatoire. Examinons ce programme.

La ligne 10 affiche APPUYEZ SUR LA BARRE D'ESPA



Déroulement du programme

Cet organigramme illustre de façon simplifiée les principales interventions du programme. Les numéros de lignes correspondants sont donnés à gauche et de courtes explications sont données à droite. Il ne s'agit pas d'un organigramme complet puisque plusieurs « décisions » et branchements du programme ne sont pas illustrés.

CEMENT. Avant d'avoir le temps de répondre à ce message, le programme est déjà entré dans la boucle FOR X = 0 TO 1 de la ligne 20. 0 et 1 peuvent sembler des limites de boucle assez étranges, mais nous verrons bientôt comment cette structure est utilisée. La ligne 30 affecte la valeur 1 à la variable R lors de la première boucle. La ligne 40 affecte le caractère tapé à la variable de chaîne A\$ en utilisant la fonction INKEY\$. Si vous tapez la lettre V, V serait affecté à A\$. La ligne 50 fait un branchement à la ligne 80 si A\$ est un espace; si ce n'est pas le cas, le programme continue à la ligne suivante.

La ligne 60 affecte 0 à la variable X qui est l'indice de la boucle. L'instruction NEXT X de la ligne 70 renvoie le programme au début de la boucle, à la ligne 20. Puisque X est remis à zéro à la ligne 60, la boucle sera répétée indéfiniment tant que le test IF A\$ = « » sera négatif.

Si vous appuyez sur la barre d'espacement, un caractère espace est affecté à A\$, le programme passera à la ligne 80 et la boucle ne sera pas répétée.

Mais que se passe-t-il lors des répétitions de la boucle? La ligne 30 incrémente R à chaque boucle. Lors de la première boucle R égale 1, lors de la deuxième boucle il égale 1 + 1, et ainsi de suite. Lors de la sortie de la boucle provoquée par le test effectué sur A\$, la valeur de R correspond au nombre de boucles exécutées.

Les ordinateurs fonctionnant très rapidement, il est probable que R aurait atteint les centaines avant l'entrée de la barre d'espacement. Comment maintenir la valeur de R comprise entre 1 et 10? La ligne 80 définit une autre boucle qui nous permet de diviser R par 10, s'il est supérieur à 10. Lorsque R est plus grand que 10, le test de la ligne 90 est négatif, la valeur de R est remise à zéro et la boucle est répétée. La ligne 110 divise la valeur de R par 10 et le résultat n'est pas imprimé tant que R n'est pas inférieur à 10. La ligne 30 assure que la valeur de R ne soit jamais 0.

Théoriquement, ce programme devrait produire un nombre aléatoire compris entre 1 et 9. Est-ce le cas? L'instruction INT supprime les fractions, et les valeurs possibles de R sont donc 1, 2, 3, 4, 5, 6, 7, 8, 9. La moyenne de ces nombres est 5 (leur somme est 45, et $45/9 = 5$). Exécutez le programme plusieurs fois et effectuez une moyenne. Il serait possible d'ajouter quelques lignes qui exécutent le programme 100 fois, qui affectent à une variable S l'addition des diverses valeurs de R et qui, à la fin du programme, divisent S par 100.

Nous avons essayé ce programme et constaté que la valeur moyenne de R était inférieure à 5. Ces nombres ne sont donc pas réellement aléatoires. Pourquoi?

Les divisions effectuées pour réduire R à une valeur inférieure à 10 ne surviennent pas avant l'entrée d'un espace. Et, à ce moment, R sera presque toujours dans les premières centaines, la valeur de R aura donc tendance à être basse.

Il est possible de corriger ce problème en définissant un processus de comptage assez rapide

```

10 REM JEU DE DÉS - PROGRAMME PRINCIPAL
20 RANDOMIZE
30 REM VOTRE JET DE DÉS
40 REM ALLER AU SOUS-PROGRAMME « JET »
50 GOSUB 300
60 LET M = D
70 PRINT « VOUS AVEZ JETÉ UN »
80 REM ALLER AU SOUS-PROGRAMME « SÉLECTION »
90 GOSUB 390
100 PRINT
110 REM LE JET DE DÉS DE L'ORDINATEUR
120 REM ALLER AU SOUS-PROGRAMME « JET »
130 GOSUB 300
140 LET O = D
150 PRINT « L'ORDINATEUR A JETÉ UN »
160 REM ALLER AU SOUS-PROGRAMME « SÉLECTION »
170 GOSUB 390
180 PRINT
190 REM VAINQUEUR!
200 IF M = 0 THEN LET S$ = « ÉGALITÉ »
210 IF MF > 0 THEN LET S$ = « VOUS GAGNEZ »
220 IF M < 0 THEN LET S$ = « L'ORDINATEUR A GAGNÉ »
230 REM IMPRESSION DES RÉSULTATS
240 PRINT S$
250 END
260 REM
270 REM
280 REM
290 REM
300 REM SOUS-PROGRAMME DE JET DE DÉS ALÉATOIRE
310 REM
320 LET D = INT (10 * RND)
330 IF D > 6 THEN GOTO 320
340 IF D < 1 THEN GOTO 320
350 RETURN
360 REM
370 REM
380 REM
390 REM SOUS-PROGRAMME DE SÉLECTION
400 REM
410 IF D = 1 THEN GOSUB 530
420 IF D = 2 THEN GOSUB 600
430 IF D = 3 THEN GOSUB 670
440 IF D = 4 THEN GOSUB 740
450 IF D = 5 THEN GOSUB 810
460 IF D = 6 THEN GOSUB 880
470 RETURN
480 REM
490 REM
500 REM
510 SOUS-PROGRAMMES GRAPHIQUES
520 REM
530 PRINT « »
540 PRINT « »
550 PRINT « »
560 PRINT « »
570 PRINT « »
580 PRINT « »
590 RETURN
600 PRINT « »
610 PRINT « »
620 PRINT « »
630 PRINT « »
640 PRINT « »
650 PRINT « »
660 RETURN
670 PRINT « »
680 PRINT « »
690 PRINT « »
700 PRINT « »
710 PRINT « »
720 PRINT « »
730 RETURN
740 PRINT « »
750 PRINT « »
760 PRINT « »
770 PRINT « »
780 PRINT « »
790 PRINT « »
800 RETURN
810 PRINT « »
820 PRINT « »
830 PRINT « »
840 PRINT « »
850 PRINT « »
860 PRINT « »
870 RETURN
880 PRINT « »
890 PRINT « »
900 PRINT « »
910 PRINT « »
920 PRINT « »
930 PRINT « »
940 RETURN

```

pour que notre temps de réaction au message APPUYEZ SUR LA BARRE D'ESPACEMENT ne puisse être planifié. Il suffit de faire le test « limite supérieure » dans la première boucle; examinez ce programme :

```
10 PRINT « APPUYEZ SUR LA BARRE D'ESPACEMENT »
20 FOR X = 0 TO 1
30 LET R = R + 1
40 IF R < 9 THEN LET R = 1
50 IF INKEY$ = « » THEN GOTO 80
60 LET X = 0
70 NEXT X
80 PRINT R
90 END
```

Dans ce programme, R ne peut jamais être plus petit que 1 ou plus grand que 9. Lorsqu'un espace est tapé au clavier, R a une valeur comprise entre 1 et 9.

Ce programme fut testé 1 000 fois et a rendu une valeur moyenne de 5,014. Puisqu'une moyenne parfaite aurait été 5, l'erreur n'est que de 0,28 %. On peut donc dire que ce programme produit un nombre aléatoire très près de la moyenne théorique. Un programme qui semble parfait sur papier peut cacher certains défauts, il est donc préférable de le tester soigneusement.

Certains lecteurs se demandent peut-être pourquoi ces programmes n'ont pas été simplifiés par l'utilisation d'instructions GOTO au lieu

de boucles FOR-NEXT. Vous comprendrez dans quelque temps pourquoi.

Exercices

● **Fonction RND.** Modifiez le dernier programme afin d'obtenir un nombre aléatoire compris entre 1 et 6 (inclusivement).

● **Boucle et moyenne.** Ajoutez des lignes au dernier programme qui l'exécutent 100 fois et qui calculent la moyenne de 100 résultats.

● **Utilisation d'un sous-programme.** Remplacez les lignes 50 et 130 par une instruction GOSUB appelant votre « générateur de nombres aléatoires » du premier exercice.

● **INKEY\$.** A l'aide de la fonction INKEY\$, écrivez un programme qui lit toute touche tapée au clavier et qui affiche : LA TOUCHE QUE VOUS AVEZ TAPÉE EST : * (* représente la touche tapée).

● **Boucle de synchronisation.** Écrivez une boucle de synchronisation et utilisez la fonction INKEY\$ pour déterminer la valeur d'une variable croissante après 10 secondes (vous devrez utiliser une montre). La sortie finale devrait être LA VALEUR DE R APRÈS 10 SECONDES EST : * (* = valeur de R).

● **Tests IF-THEN.** Écrivez un programme de jeu dans lequel l'ordinateur génère un nombre aléatoire compris entre 1 et 100 (inclusivement) et qui doit permettre au joueur de deviner quel est ce nombre. Le programme répondra VOTRE RÉPONSE EST TROP ÉLEVÉE, ou VOTRE RÉPONSE EST TROP BASSE, ou C'EST EXACT, FÉLICITATIONS, ou PLUS D'ESSAI, VOUS AVEZ PERDU.

Les réponses seront publiées dans une prochaine leçon.

Variantes de basic



Sur le micro BBC et sur l'Oric-1, effacez la ligne 20 et remplacez la ligne 320 par :

```
320 LET D = INT (10 * RND (1))
```



Sur le Dragon 32, effacez la ligne 20, et remplacez la ligne 320 par :

```
320 LET D = RND (6)
```

et effacez les lignes 330 et 340.

Sur le Lynx, remplacez la ligne 20 par :

```
20 RANDOM
```

Sur le Vic-20 et sur le Commodore 64, remplacez la ligne 20 par :

```
20 LET X = RND (- TI)
```

et remplacez la ligne 320 par :

```
320 LET D = INT (10 * RND (1))
```

Sur le Spectrum, le mot RANDOMIZE est abrégé sur le clavier en RAND, mais il est écrit RANDOMIZE à l'écran.

Sur l'Oric-1 et sur le Lynx, remplacez INKEY\$ PAR KEY\$.



Sur le Vic-20 et sur le Commodore 64, remplacez la ligne 40 par :

```
40 GET A$
```

Puis remplacez la ligne 50 par :

```
50 GET A$:IF A$ = « » THEN GOTO 80
```

Sur le micro BBC, remplacez INKEY\$ par INKEY\$(10). Le nombre entre parenthèses est le temps (exprimé en centièmes de seconde) pendant lequel le système attend l'entrée d'un caractère; pour obtenir une réponse rapide, utilisez un nombre peu élevé, et vice versa.

Réponses aux exercices des pages 148-149

Boucles 1

LA VALEUR DE A EST 450

Boucles 2

C'EST PARTI
STOP

Boucles 3

A VAUT MAINTENANT 160

Boucles 4

J'AIME LE BASIC
J'AIME LE BASIC
J'AIME LE BASIC

:

Jusqu'à l'interruption du programme

Boucles 5

SANS VOULOIR ME RÉPÉTER
(15 fois)

Read-Data 1

NOUS TESTONS L'INSTRUCTION READ
170

Read-Data 2

X = 1
X = 2
:
:
X = 23



Sur deux roues

La robotique se développe rapidement. Les robots de sol comme le BBC Buggy peuvent être programmés pour détecter des obstacles.

Les robots de sol et les tortues sont amusants autant qu'instructifs. Les principes de commande d'un dispositif comme le BBC Buggy sont les mêmes que ceux des robots industriels. Même s'ils ne se chargent pas encore de travaux domestiques, ils devraient représenter la prochaine génération de robots familiaux.

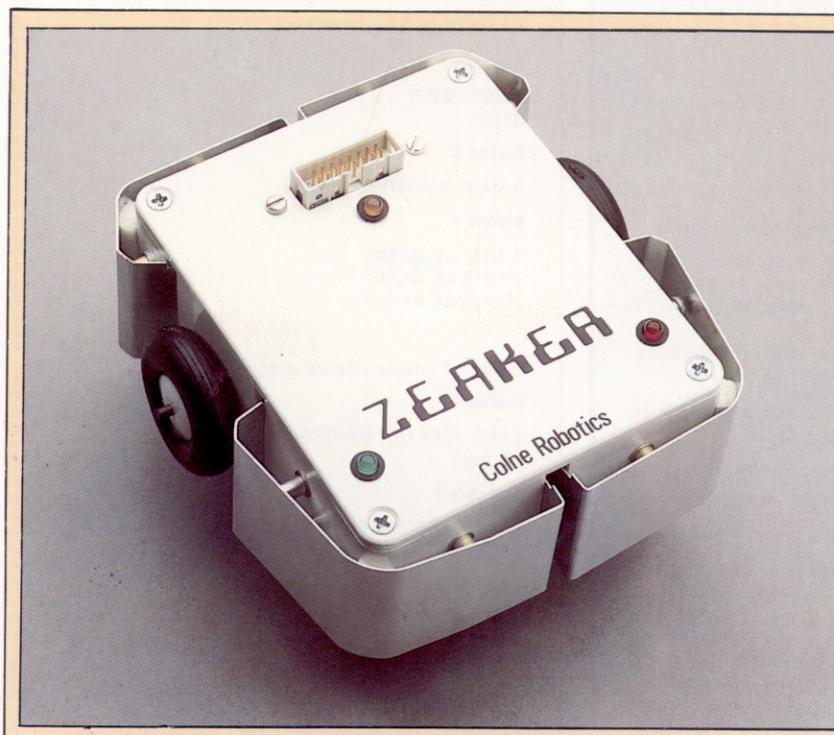
Les robots doivent être positionnés précisément, c'est pourquoi ils possèdent généralement des moteurs pas à pas. Contrairement aux moteurs traditionnels, les moteurs pas à pas ne tournent pas lors de leur mise sous tension. L'axe n'effectue qu'une fraction prédéterminée de rotation à chaque impulsion électrique appliquée. Le nombre d'impulsions correspondant à une révolution complète dépend du type de moteur. La direction de rotation peut aussi être commandée. Il est possible de faire se déplacer sur des distances très précises, et cela dans toutes les directions, un robot de sol, ou tortue, en permettant à l'ordinateur de commander séparément les moteurs. Un robot de sol peut tourner sur place en commandant un mouvement inverse aux deux roues.

Cependant, le robot doit également être en mesure de signaler à l'ordinateur la rencontre d'un obstacle. Les collisions sont généralement

détectées par des pare-chocs montés autour du véhicule et qui sont connectés à des microcontacts. Ceux-ci sont reliés, à leur tour, au port d'entrée de l'ordinateur. L'ouverture, ou la fermeture, de ces microcontacts provoque une information à l'ordinateur sous forme de bits de valeur « 0 » ou « 1 ».

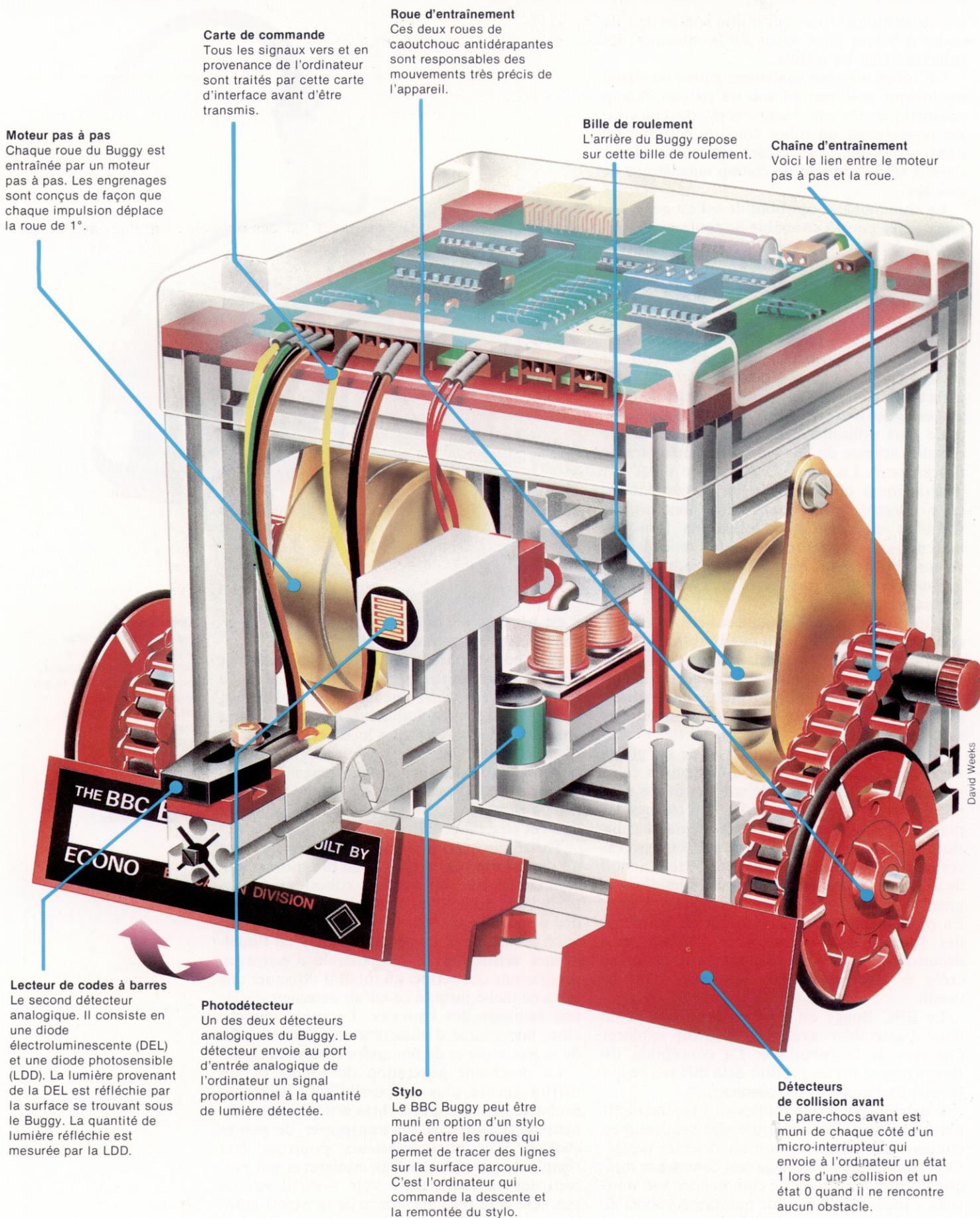
D'autres types d'entrées provenant du robot sont aussi nécessaires. La possibilité de suivre une ligne blanche sur un sol noir peut être utile. Pour ce faire, le robot est muni d'une source lumineuse qui éclaire le sol et d'un photodétecteur qui mesure la quantité de lumière réfléchie. Cette quantité variera selon la nature de la surface sur laquelle le robot se déplacera à un moment donné. Très souvent, il s'agit d'une quantité analogique et non numérique. Le micro BBC est muni d'une entrée analogique qui permet l'utilisation d'un tel détecteur. La plupart des autres systèmes doivent convertir le signal en un signal numérique avant de l'envoyer à l'ordinateur.

Une autre utilisation de ce type de détecteur apparaît dans les lecteurs de codes à barres. Les codes correspondant à la description des articles d'un magasin, par exemple, peuvent être analysés par un robot et permettre ainsi la recherche du produit désiré. Le BBC est livré avec un logi-



Fonctionnement

Cette micro-tortue possède des détecteurs de collision et peut être dotée d'un certain niveau d'« intelligence ». Le Zeaker peut se déplacer vers l'avant, l'arrière, la gauche ou la droite. Il peut abaisser et remonter un stylo. Les commandes très simples sont données à l'aide d'une version de LOGO appelée SNAIL LOGO. Des commandes comme FORWARD et BACKWARD peuvent être utilisées directement. Les ordinateurs autres que le BBC nécessitent une interface, mais comme celle-ci possède sa propre alimentation, l'alimentation de l'ordinateur n'est nullement sollicitée. (Cl. Chris Stevens.)



Moteur pas à pas

Chaque roue du Buggy est entraînée par un moteur pas à pas. Les engrenages sont conçus de façon que chaque impulsion déplace la roue de 1°.

Carte de commande

Tous les signaux vers et en provenance de l'ordinateur sont traités par cette carte d'interface avant d'être transmis.

Roue d'entraînement

Ces deux roues de caoutchouc antidérapantes sont responsables des mouvements très précis de l'appareil.

Bille de roulement

L'arrière du Buggy repose sur cette bille de roulement.

Chaîne d'entraînement

Voici le lien entre le moteur pas à pas et la roue.

Lecteur de codes à barres

Le second détecteur analogique. Il consiste en une diode électroluminescente (DEL) et une diode photosensible (LDD). La lumière provenant de la DEL est réfléchiée par la surface se trouvant sous le Buggy. La quantité de lumière réfléchiée est mesurée par la LDD.

Photodétecteur

Un des deux détecteurs analogiques du Buggy. Le détecteur envoie au port d'entrée analogique de l'ordinateur un signal proportionnel à la quantité de lumière détectée.

Stylo

Le BBC Buggy peut être muni en option d'un stylo placé entre les roues qui permet de tracer des lignes sur la surface parcourue. C'est l'ordinateur qui commande la descente et la remontée du stylo.

Détecteurs de collision avant

Le pare-chocs avant est muni de chaque côté d'un micro-interrupteur qui envoie à l'ordinateur un état 1 lors d'une collision et un état 0 quand il ne rencontre aucun obstacle.



ciel de démonstration qui utilise son lecteur de codes à barres pour jouer de la musique; les principes sont les mêmes.

Un robot pourrait également suivre un signal analogique créé par un son ou par un champ magnétique. De tels champs sont souvent utilisés pour guider un robot dans un entrepôt ou dans une usine. Des câbles spéciaux enfouis dans le sol produisent un champ magnétique de guidage.

La commande d'un robot de sol est généralement gérée par un ensemble de routines spécialement écrites. Ces routines gèrent l'entrée/sortie de l'information *via* le câble reliant le port utilisateur de l'ordinateur au robot. Dans le cas du BBC Buggy, quatre bits servent à commander les moteurs. Les données sont envoyées par le robot sur le même câble. Les sorties analogiques du photodétecteur et du lecteur de codes à barres vont sur le port analogique, et les détecteurs de collision sont connectés à deux autres lignes d'entrée sur le port utilisateur.

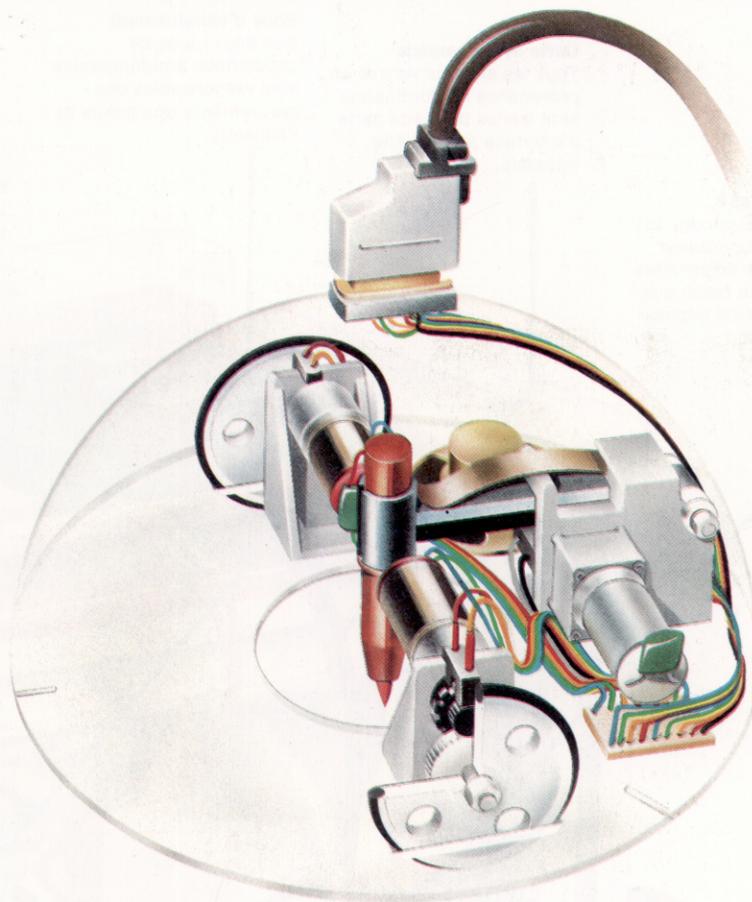
Le port utilisateur peut être examiné à une certaine adresse de la topographie mémoire de l'ordinateur. La commande PEEK de BASIC est généralement utilisée pour lire le contenu de cette adresse. Pour modifier le contenu, de façon à changer la direction de l'un des moteurs, le programmeur doit modifier la valeur du bit approprié de cette adresse. La commande POKE permet cette intervention dans un programme BASIC.

L'information analogique peut être examinée de la même façon, pourvu que l'ordinateur soit muni d'un convertisseur analogique-numérique intégré. Sinon, une unité d'interface doit être ajoutée au robot pour lui permettre de convertir les signaux analogiques en données numériques avant leur envoi.

Une tortue est en fait un robot de sol utilisé avec le langage LOGO, mais la distinction entre les tortues et les robots de sol s'estompe progressivement (voir page 164). Les tortues les plus récentes sont munies de détecteurs de collision, et de nombreux robots de sol peuvent tracer des lignes comme des tortues. Cette fonction de dessin à grande échelle a pour objet d'enseigner les relations entre distance, angle et forme. En plaçant un stylo sous le robot ou la tortue, des formes sont dessinées et l'enfant peut découvrir les relations existant entre la forme créée et les mouvements impliqués dans ce dessin.

Le BBC Buggy est livré en kit, vous devez donc l'assembler avant de pouvoir explorer l'univers de la robotique. La conception du Buggy repose sur un produit déjà diffusé, ce qui permet de l'améliorer facilement.

L'assemblage d'un tel dispositif est instructif par lui-même. On peut apprendre beaucoup en étudiant les relations entre les diverses pièces. Cependant, l'apprentissage réel commence lorsque l'utilisateur essaie de commander son nouveau « jouet ». Même si de nombreux robots de sol sont vendus avec un logiciel de commande, il est beaucoup plus amusant d'écrire le sien. Ici,



il est nécessaire d'adopter une nouvelle approche de la programmation.

Pour commander le robot, le programme doit constamment contrôler ses capteurs afin de savoir s'il a rencontré une ligne sur le plancher, détecté une source lumineuse intense, heurté une chaise, etc. Dès que le détecteur signale quelque chose, l'ordinateur doit réagir automatiquement afin d'éviter d'endommager le robot. On dit que les programmes de ce type fonctionnent « en temps réel » parce que leurs réponses doivent être immédiates.

Théoriquement, il y a peu de différence entre un programme de ce type et un programme qui commande une application robotique industrielle. Les techniques apprises en jouant avec des dispositifs comme le BBC Buggy peuvent aussi permettre de comprendre ce qu'est l'intelligence artificielle. Il est possible d'écrire un programme qui permet au robot d'effectuer une certaine tâche jusqu'à ce qu'un détecteur décele une faiblesse des batteries. Le robot cherche alors une source d'alimentation adéquate afin de se recharger et de reprendre ensuite sa tâche.

La prochaine génération de robots de sol offrira encore plus de possibilités. Ils seront probablement équipés de bras articulés leur permettant de saisir et de transporter de petites charges. Les photodétecteurs pourront être remplacés par des caméras miniatures qui permettront au robot de « voir » où il va. La synthèse et la reconnaissance de la parole autoriseront un dialogue entre le robot et l'opérateur.

Robot dédié

Une tortue est un robot spécialisé qui, sous la commande de l'ordinateur, trace des dessins sur le sol avec un stylo feutre rétractable. Les tortues sont généralement associées au langage éducatif LOGO (voir page 164), mais elles peuvent être pilotées par un programme BASIC. (Cl. David Weeks.)

La base 16

Les ordinateurs travaillent en notation binaire parce qu'ils ne peuvent interpréter que deux états. Pourquoi alors les programmeurs utilisent-ils parfois les nombres hexadécimaux, la base 16?

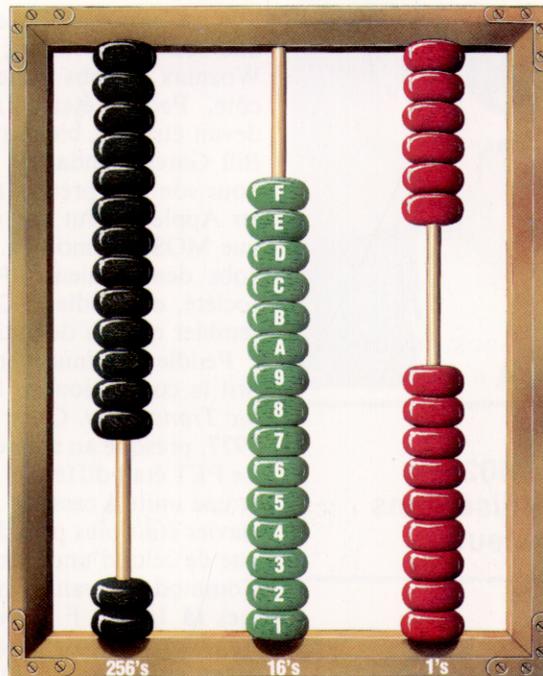
Il est facile de comprendre pourquoi le système binaire est utilisé dans les ordinateurs; la représentation de nombres à l'aide de 0 et de 1 correspond parfaitement aux états des signaux électriques de l'ordinateur. Il est aussi facile de comprendre pourquoi le système décimal est utilisé presque universellement dans les sociétés humaines; l'homme ayant dix doigts, cette base numérique s'est imposée naturellement.

Mais pourquoi utiliser l'hexadécimal (la base 16)? Les nombres hexadécimaux se convertissent beaucoup plus facilement en nombres binaires que les nombres décimaux. Puisque les ordinateurs « comprennent » les nombres binaires, pourquoi les programmeurs ne programment-ils pas en chiffres binaires? A cela, deux raisons qui sont simples à comprendre. D'abord, les nombres binaires sont beaucoup plus longs que leurs équivalents en base 16 ou 10. Par exemple, le nombre décimal 356 correspond au nombre binaire 0000000101100100. Deuxièmement, il est beaucoup plus facile de faire des erreurs avec des nombres ne comprenant que des 1 et des 0.

Le système hexadécimal a seize chiffres, incluant zéro, le plus grand nombre pouvant être représenté dans la colonne des 1 correspond donc à 15 décimal. Les mathématiciens auraient pu inventer six nouveaux symboles pour les nombres entre 10 et 15, mais optèrent pour la convention des six premières lettres de l'alphabet. Pour compter, nous utilisons donc les chiffres numériques normaux et les chiffres alphabétiques A à F; F est l'équivalent hexadécimal de 15. Là, nous avons épuisé tous les symboles et devons commencer une nouvelle colonne, la colonne des 16. Lorsque les colonnes des 16 et des 1 sont pleines, additionner 1 signifie commencer une nouvelle colonne, la colonne des 256. Le tableau ci-contre montre des nombres décimaux avec leurs équivalents binaires et hexadécimaux.

Le nombre 65 535 représenté en hexadécimal est FFFF, ainsi l'un des avantages du système hexadécimal est assez évident. Un nombre de cinq chiffres décimaux, ou de seize chiffres binaires, peut être représenté par quatre chiffres hexadécimaux. En plus, ils offrent un autre avantage. Quatre chiffres binaires peuvent être représentés par un chiffre hexadécimal, ce qui rend la conversion binaire-hexadécimal très facile.

Pour convertir un nombre binaire en hexadécimal, divisez le nombre binaire en groupes de



Le boulier hexadécimal

Un boulier utilisé dans un système hexadécimal aurait 15 boules sur chaque tringle. Sur la tringle d'extrême droite, les boules sont descendues une à la fois jusqu'à la boule F. Le système hexadécimal fonctionne en base 16 et des lettres de l'alphabet sont utilisées pour les chiffres supérieurs à 9. Comme nous l'avons indiqué sur la tringle du milieu, A correspond à 10 décimal, B à 11, C à 12, D à 13, E à 14 et F à 15. Après le nombre F, nous avons épuisé les chiffres disponibles et nous devons passer à la colonne suivante, la colonne des 16. La colonne des 1 dans l'illustration montre un compte de 9. La colonne des 16 montre un compte de F. Et la colonne des 256 un compte de 2. Il s'agit donc du nombre « 2F 9 hex ». Son équivalent décimal est 761 [(256 × 2) + (16 × 15) + (1 × 9)]. (Cl. Mark Watkinson.)

quatre chiffres, en commençant par la droite, et convertissez chaque groupe en hexadécimal. Voici quelques exemples :

		1110	1001	binaire
		= E	9	hex.
		= 233		décimal
	1111	1000	1100	binaire
	= F	8	C	hex.
	= 3980			décimal
0111	1110	0010	1101	binaire
= 7	E	2	D	hex.
= 32 301				décimal

Quand devez-vous utiliser des nombres hexadécimaux? Bien que des langages évolués comme le BASIC ne nécessitent pas l'utilisation de nombres hexadécimaux (ou binaires), vous devrez les utiliser en assembleur ou en code machine.

Les nombres hexadécimaux sont souvent suivis d'un H pour les distinguer des nombres décimaux. Par conséquent, 256 (décimal) serait écrit 100H et prononcé « un zéro zéro hex ».

Nous utiliserons rarement les nombres hexadécimaux dans le *Cours de micro-informatique*. Cependant, vous les rencontrerez sans doute dans le manuel de votre ordinateur. Vous savez maintenant comment trouver leurs équivalents décimaux et binaires.

Binaire	Décimal	Hexadécimal
00000001	1	1
00000010	2	2
00000011	3	3
00000100	4	4
00000101	5	5
00000110	6	6
00000111	7	7
00001000	8	8
00001001	9	9
00001010	10	A
00001011	11	B
00001100	12	C
00001101	13	D
00001110	14	E
00001111	15	F
00010000	16	10
00010001	17	11
00010010	18	12
00010011	19	13
00010100	20	14
00010101	21	15

Chuck Peddle



L'homme qui conçut le microprocesseur 6502; ce composant est utilisé dans la plupart des ordinateurs.

Issu de la génération antérieure à celle des Wozniak et Jobs, Chuck Peddle commença à s'intéresser aux microprocesseurs en 1973 lorsqu'il rejoignit Motorola pour travailler au projet de conception du microprocesseur 6800.

Puisque c'était le premier microprocesseur offert sur le marché, Motorola pouvait demander un prix élevé pour le 6800, 200 dollars pièce. Peddle estima que ce prix était trop élevé et quitta Motorola pour rejoindre MOS Technology.

Dans cette petite société, il commença à travailler à un autre projet de microprocesseur qui allait devenir le 6502 MPU, le microprocesseur qui obtint le plus de succès dans la première décennie de la micro-informatique. A l'époque, cependant, personne ne se doutait que ce produit allait devenir le pilier d'une industrie, et contribuer dans une large mesure à une véritable révolution sociale, telle que le monde n'en avait pas connu depuis deux cents ans.

Une des rares personnes à avoir saisi l'importance du microprocesseur et le potentiel particulier du 6502 fut le président de Commodore, Jack Tramiel. Jusqu'alors, Commodore Business Machine n'avait produit que divers articles de bureau et des calculatrices de poche sans succès significatif.

Commodore était un des principaux clients de MOS Technology, achetant régulièrement un très grand nombre de puces destinées à des calculatrices 4 fonctions. Mais, malgré ses difficultés pour maintenir financièrement à flot

son entreprise Commodore, Tramiel avait une très grande confiance et donc un intérêt certain pour le 6502. Il réussit à trouver l'argent nécessaire pour racheter MOS Technology. Et en reprenant cette société, il s'adjudgeait par la même occasion les services de Chuck Peddle, responsable du développement des microprocesseurs.

A ce moment, Peddle commença à rêver à une réelle innovation, l'ordinateur individuel. Wozniak et Jobs avaient la même idée de leur côté. Peddle était d'avis que la technologie devait être très bien utilisée, aussi il s'associa à Bill Gates, fondateur de Microsoft — célèbre pour son interpréteur BASIC — et tenta d'acheter Apple qui fut mis en vente en même temps que MOS Technology. Cependant, Wozniak et Jobs demandaient 150 000 dollars pour leur société, et Peddle et Gates ne réussirent à rassembler que les deux tiers de cette somme.

Peddle continua chez Commodore et entreprit la conception du PET (*Personal Electronic Transactor*). Cette machine fut présentée en 1977, presque au même moment que l'Apple II. Le PET était différent : il possédait un moniteur et une unité à cassette intégrés, et le toucher du clavier était plus près de celui d'une calculatrice que de celui d'une machine à écrire. Peu après, Commodore avait un millier de commandes fermes (à 10 000 F l'unité); la première génération d'ordinateurs conçus spécialement pour un usage individuel était née.

Trois ans plus tard, Peddle réalisa un second rêve, celui de former sa propre société. Avec Chris Fish, l'un des experts financiers de Commodore, il rejoignit Victor United, une filiale du géant Walter Kidde Corporation, et démarra la société Sirius Systems Technology.

Les travaux de recherches en micro-informatique étaient concentrés sur les puces 16 bits comme l'Intel 8088. IBM travaillait également à un projet d'ordinateur individuel fondé sur la même puce. Heureusement pour Peddle, Sirius put présenter les fruits de ses travaux quelques semaines avant IBM. La machine fut très bien accueillie et établit rapidement une forte présence sur le marché, étant le premier micro-ordinateur produit en série et peu coûteux à offrir les avantages des microprocesseurs 16 bits.

Le Sirius 1 était relativement bon marché et facile à utiliser. Ses diverses caractéristiques établirent de nouvelles normes pour les systèmes professionnels. Les utilisateurs profitèrent énormément des possibilités offertes par ce microprocesseur au niveau vitesse et adressage grâce à l'utilisation d'un 16 bits terriblement efficace.

Chuck Peddle a presque atteint son but : mettre la puissance de l'ordinateur à la portée de tous. Et par cette démarche il mit au point un standard que d'autres constructeurs de micro-ordinateurs suivent aujourd'hui.



Le premier ordinateur individuel

Après avoir développé le microprocesseur 6502, Chuck Peddle commença à travailler à la conception d'un appareil qu'il nomma « ordinateur individuel ». Cet ordinateur fonctionnait de façon entièrement autonome, et l'utilisateur n'avait qu'à le brancher pour pouvoir l'utiliser dans de multiples applications. Le PET de Commodore, qui fit son apparition sur le marché presque au même moment que l'Apple II, offrait un écran et une unité à cassette intégrés, ainsi qu'un interpréteur BASIC. Bien qu'il ait subi plusieurs modifications depuis son lancement, PET est toujours populaire. (Cl. Sirius Computing.)

LE « GUIDE D'ACHAT » DE VOTRE ENCYCLOPÉDIE

Cher lecteur,

Vous avez entre les mains les premiers numéros d'ABC informatique, et nous vous remercions de votre achat.

Que vous soyez déjà un lecteur d'encyclopédies en fascicules, ou que celle-ci soit la première que vous envisagiez d'acquérir : dans les deux cas vous trouverez ci-dessous quelques informations et conseils pratiques qui vous aideront à constituer sans problèmes, chaque semaine, votre encyclopédie.

LES AVANTAGES DE L'ENCYCLOPÉDIE EN FASCICULES

- Chaque semaine, pour le prix d'un magazine, vous pouvez acquérir un ouvrage complet sur un sujet qui vous passionne, et votre dépense est étalée dans le temps.
- Vous disposez à chaque livraison d'une quantité de lecture raisonnable.
- Vous vous constituez un ouvrage de référence, que toute la famille peut consulter à loisir.
- Des reliures vous permettent de transformer vos fascicules en luxueux volumes qui mettront en valeur votre bibliothèque.

QUELQUES CONSEILS

- Achetez vos fascicules régulièrement chez le même marchand de journaux, vous serez mieux servi et vous faciliterez notre distribution.
- Si pour une raison quelconque — absence momentanée, vacances, fermeture annuelle de votre point de vente, etc. — vous ne pouviez vous procurer un ou plusieurs numéros, commandez-les chez votre marchand : dans les meilleurs délais, celui-ci recevra votre commande. Pendant toute la durée de la collection, nous vous garantissons la disponibilité de tous les numéros (et même après la fin de la collection, pendant au minimum 6 mois).
- Pour faciliter votre commande, rappelez à votre marchand de journaux :
 - le titre de l'encyclopédie : *Abc informatique*
 - la codification (c'est le numéro à 4 chiffres qui est indiqué dans un petit rectangle sur la première page de couverture) :

Abc informatique M. 6062

— les numéros des fascicules souhaités, par exemple :

*Abc informatique M. 6062
numéros 7, 8 et 9*

— si vous aviez malgré tout des difficultés pour trouver régulièrement votre encyclopédie à votre point de vente habituel, adressez une simple lettre à notre : Service de ventes, Éditions Atlas, 33, avenue du Maine, 75755 PARIS Cedex 15, en indiquant clairement :

- votre nom et votre adresse,
- l'adresse de votre marchand de journaux,
- la nature du problème (service irrégulier, numéros manquants, etc.).

Nous prendrons immédiatement les dispositions nécessaires pour y remédier, et vous en tiendrons informé.

LES RELIURES

Nous mettons à votre disposition des reliures qui contiennent chacune 12 numéros.

Ces reliures sont identiques pour tous les volumes de la collection. Un système de transfert, livré avec chaque reliure, permet d'indiquer sur le dos le numéro du volume. S'il arrivait que votre point de vente manque momentanément de reliures, il peut vous les commander et vous les fournir rapidement.

La codification des reliures d'ABC informatique est : M. 6103.

Passez votre commande en indiquant, par exemple :

*reliure Abc informatique M 6103
3 exemplaires*

Pour être certain de pouvoir toujours conserver vos fascicules en parfait état, ayez toujours une ou plusieurs reliures d'avance. Vous y classerez vos fascicules au fur et à mesure de leur parution.

Encore une fois, n'hésitez pas à nous contacter pour tout problème que vous pourriez rencontrer dans l'acquisition de votre encyclopédie. Nous n'en connaissons pas qui ne puissent se résoudre.

ÉDITIONS ATLAS

**Chez tous les marchands de journaux,
une encyclopédie exceptionnelle en 12 fascicules :**

le Livre d'Or du
FROMAGE

**Les saisons • Les variétés
Les recettes • L'historique • L'élevage
La conservation**



Tout sur tous les fromages
expliqué par Pierre Androuet,
le meilleur spécialiste du monde,
dont le restaurant fromager,
à Paris, reçoit les amateurs des
cinq continents.

- Pour préparer un plateau de fête**
- Pour choisir un fromage de tous les jours**
- Pour conserver vos fromages chez vous**
- Pour réussir des croquettes et des soufflés**
- Pour trouver le bon vin sur le bon fromage**



Jean Riby/Ed. Atlas

**EDITIONS
ATLAS**

**Chaque semaine,
chez tous les marchands
de journaux.**