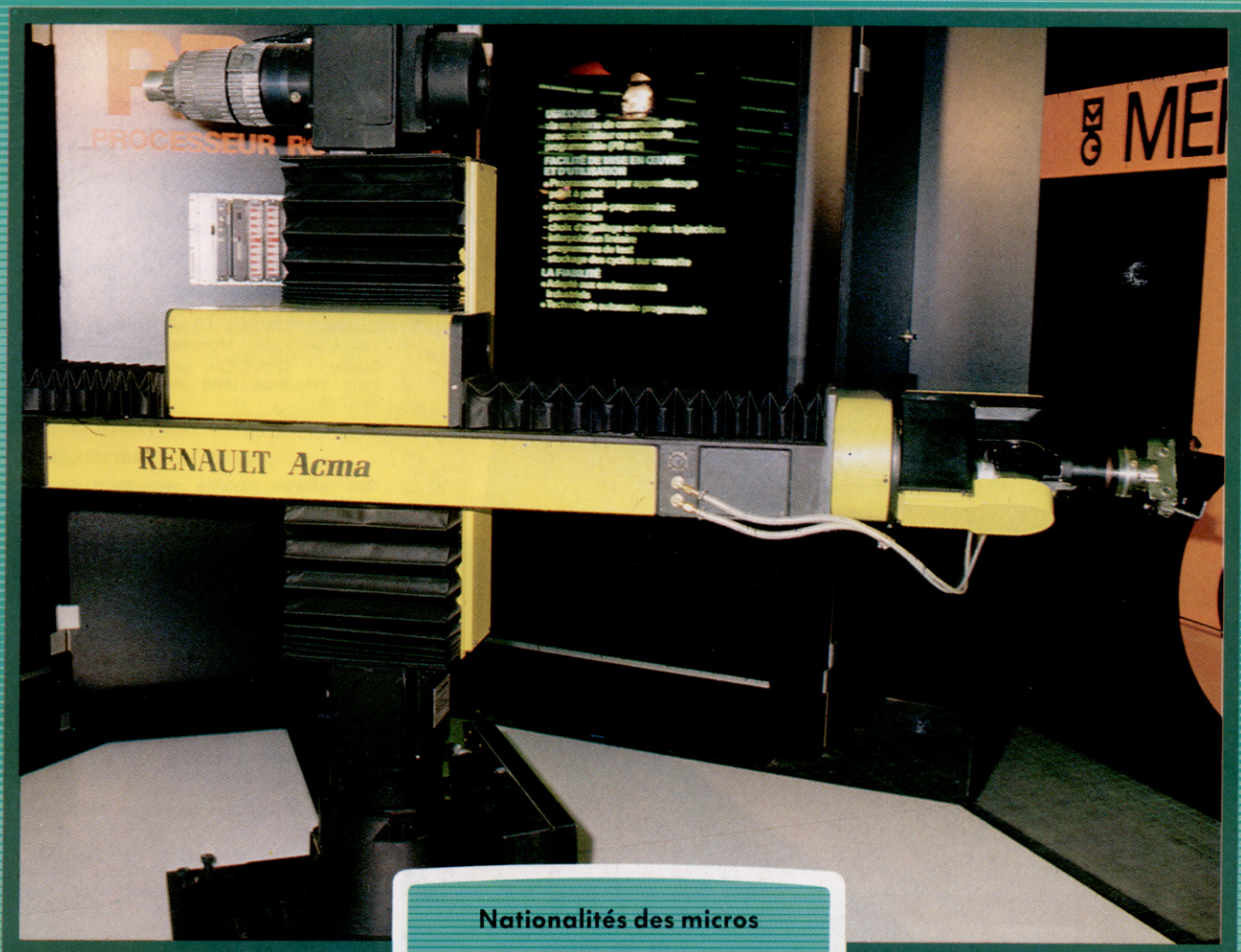


# abc

N° 46

COURS  
D'INFORMATIQUE  
PRATIQUE  
ET FAMILIALE

# INFORMATIQUE



Nationalités des micros

L'Amstrad CPC

Table de décision

Un jeu sur Atari

EDITIONS  
**ATLAS**

# GRAND ATLAS GÉOGRAPHIQUE ET ENCYCLOPÉDIQUE

LE LIVRE  
ÉVÉNEMENT  
DE L'ANNÉE

GRAND  
ATLAS  
GÉO-  
GRAPHIQUE  
ET ENCYCLOPÉDIQUE

EDITIONS ATLAS

Une cartographie unique au monde, entièrement renouvelée grâce aux techniques les plus modernes.

La formation de la Terre et l'apparition de la vie en 100 pages luxueusement illustrées.

- 67 cartes continentales inédites.
- 7 grandes cartes détaillées de la France et de la Belgique.
- 500 schémas et illustrations en couleurs.
- 100 000 lieux géographiques répertoriés et localisés en index.

L'ouvrage de référence attendu avec impatience par le grand public, les étudiants, les enseignants ou, tout simplement, les passionnés de voyages.

Un outil de travail indispensable à toute la famille.

**DISPONIBLE DÈS À PRÉSENT  
CHEZ VOTRE LIBRAIRE**

EDITIONS  
ATLAS

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. : (37) 28-10-10. Services administratifs et commerciaux : 3, rue de la Taye, 28110 Lucé. Tél. : (37) 28-10-10.

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., boîte 185, 33, avenue du Maine, 75755 Paris Cedex 15.

Direction générale : J.-P. Turbergue. Service technique et artistique : F. Givone et J.-Cl. Bernar. Iconographie : J. Pierre. Correction : B. Noël.

Publicité : Anne Cayla. Tél. : 202-09-80.

## VENTE AU NUMÉRO

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 aux services commerciaux des ÉDITIONS ATLAS, Z.I. de Lucé, 3, rue de la Taye, 28110 Lucé. Tél. : (37) 28-10-10.

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 sont en vente 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 IV, n° 46.

ABC INFORMATIQUE est réalisé avec la collaboration de Trystan Mordrel (secrétariat de rédaction), Jean-Pierre Bourcier (coordination), Patrick Bazin, Jean-Paul Mourlon, Claire Rémy (traduction), Ghislaine Goullier (fabrication), Marie-Claire Jacquet (iconographie), Claire Bischoff (correction).  
Crédit photographique, couverture : Maillac/Réa.

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 : novembre 1984. 308411. Dépôt légal en Belgique : D/84/2783/27.

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

## A NOS LECTEURS

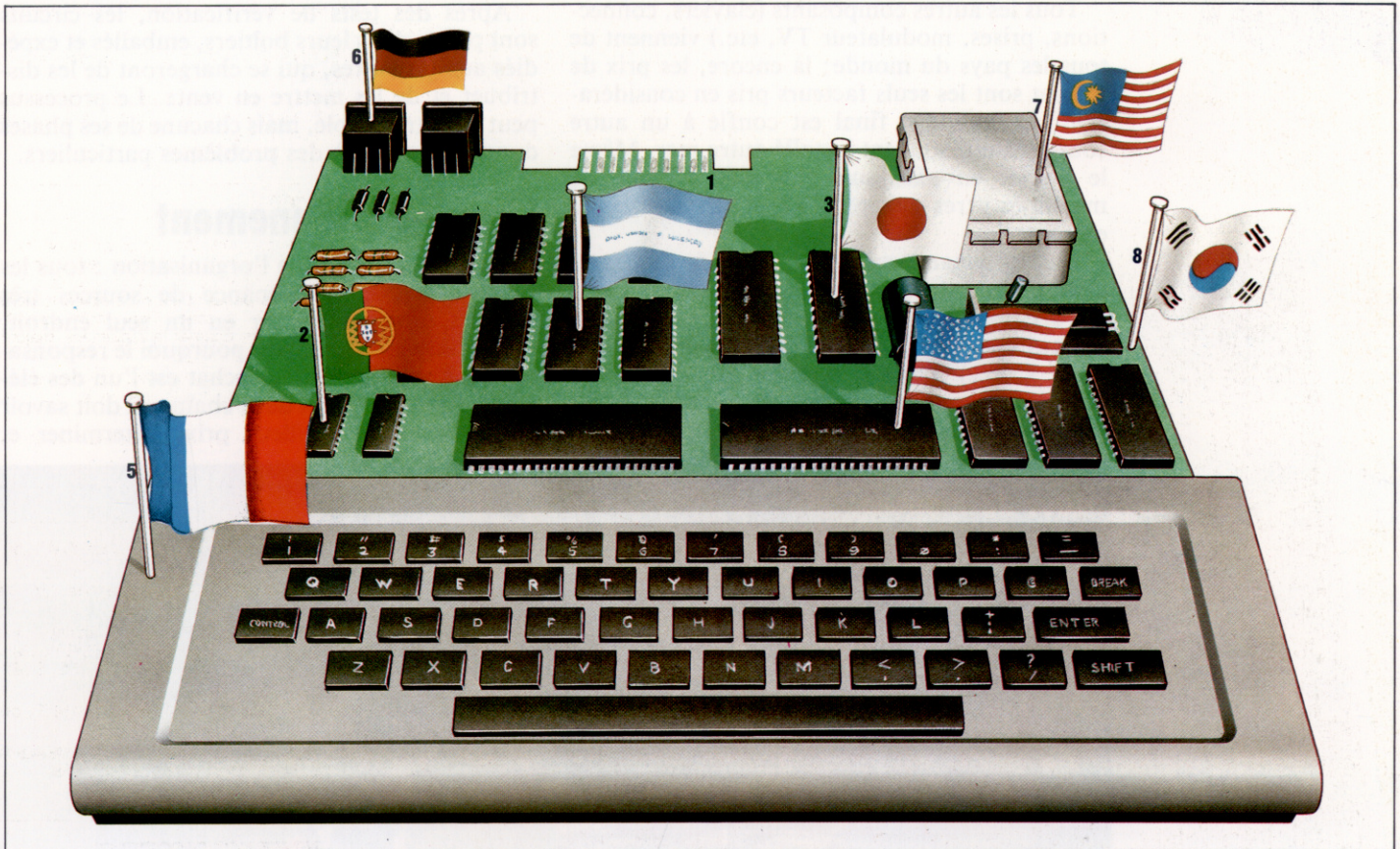
En achetant chaque semaine votre fascicule chez le même marchand de journaux, vous serez certain d'être immédiatement servi, en nous facilitant la précision de la distribution. Nous vous en remercions d'avance.

Les Éditions Atlas



# Tour du monde

Quelle est la nationalité de votre micro-ordinateur? Un examen détaillé des composants révélera souvent la participation de plusieurs pays, dont certains vous étonneront.



La construction des micro-ordinateurs se fait sur une base multinationale. L'Amstrad CPC 464, dont nous avons déjà parlé, est entièrement réalisé en Corée, et de nombreux BBC Micro d'Acorn le sont à Hong Kong. Sinclair Research, de son côté, entend se consacrer uniquement à la recherche et à la conception, et confie la construction et l'assemblage de ses machines à des sous-traitants.

A chaque fois, il s'agit d'abaisser au minimum le prix de revient. C'est même la préoccupation principale des concepteurs. Le circuit imprimé sera ainsi petit et simple, et comportera aussi peu de puces que possible. Ce n'est pas qu'elles soient chères en elles-mêmes, mais c'est leur montage qui se révèle coûteux; de plus, la fiabilité de l'appareil en est parfois affectée. On en comprend parfaitement la raison lorsqu'on voit le procès qui est fait à Texas Instruments aux États-Unis à propos du manque de fiabilité de certains des composants que cette entreprise fabrique.

Cela explique la présence de puces de type ULA (*uncommitted logic array*, « unité arithmétique et logique ») sur la plupart des micro-ordinateurs domestiques. Elles sont chères mais remplacent des douzaines de puces plus petites sur le circuit imprimé.

Presque toutes sont fabriquées en Californie. On appelle même Silicon Valley (la vallée du silicium) l'endroit où les constructeurs sont rassemblés. Une fois créées, ces puces doivent être incluses dans des boîtiers de plastique et de céramique. Comme c'est une opération qui demande moins de compétence technique et a surtout recours au travail humain intensif, elles sont expédiées à l'étranger, dans des pays où la main-d'œuvre est moins chère.

Une fois la conception du micro-ordinateur menée à bien, la recherche des sous-traitants commence. La fabrication des circuits imprimés, comme celle des puces, est une entreprise complexe, domaine réservé de firmes spécialisées qui disposent des moyens nécessaires à la

## Du monde entier

Cet ordinateur imaginaire est fabriqué à partir de composants venus du monde entier. Les puces sont originaires : d'El Salvador (1); du Portugal (2); du Japon (3); des États-Unis (4). Le boîtier, le clavier et le montage final sont réalisés en Grande-Bretagne ou en France (5); les prises viennent de RFA (6). Le modulateur haute fréquence est fabriqué en Malaisie (7), et le circuit imprimé en Corée (8). (Cl. Steve Cross.)



réalisation des circuits, à partir des plans détaillés fournis par la maison mère. Celle-ci doit d'ailleurs tenir compte, dès le départ, des coûts de fabrication; il est possible de recourir à des circuits multi-couches, dans lesquels des isolants sont placés entre deux couches de métal : mais c'est une solution très onéreuse, et donc rarement employée. On se sert plutôt de trous percés dans le circuit, et dont les bords sont garnis de métal, afin d'améliorer les contacts électriques. Cela a de surcroît l'avantage de renforcer la fiabilité de l'ensemble.

Tous les autres composants (claviers, connexions, prises, modulateur TV, etc.) viennent de tous les pays du monde; là encore, les prix de revient sont les seuls facteurs pris en considération. Le montage final est confié à un autre sous-traitant, souvent installé outre-mer. Même le boîtier en plastique, fabriqué à l'aide de machines très chères, vient d'une autre entreprise.

L'assemblage peut se faire de deux façons différentes : soit par l'automatisation (États-Unis, Europe, Japon), soit manuellement, grâce à une main-d'œuvre bon marché (Hong Kong, Singapour, Corée). Dans le premier cas, des robots installés le long des chaînes de montage fixent les composants sur les circuits, et sont sans cesse

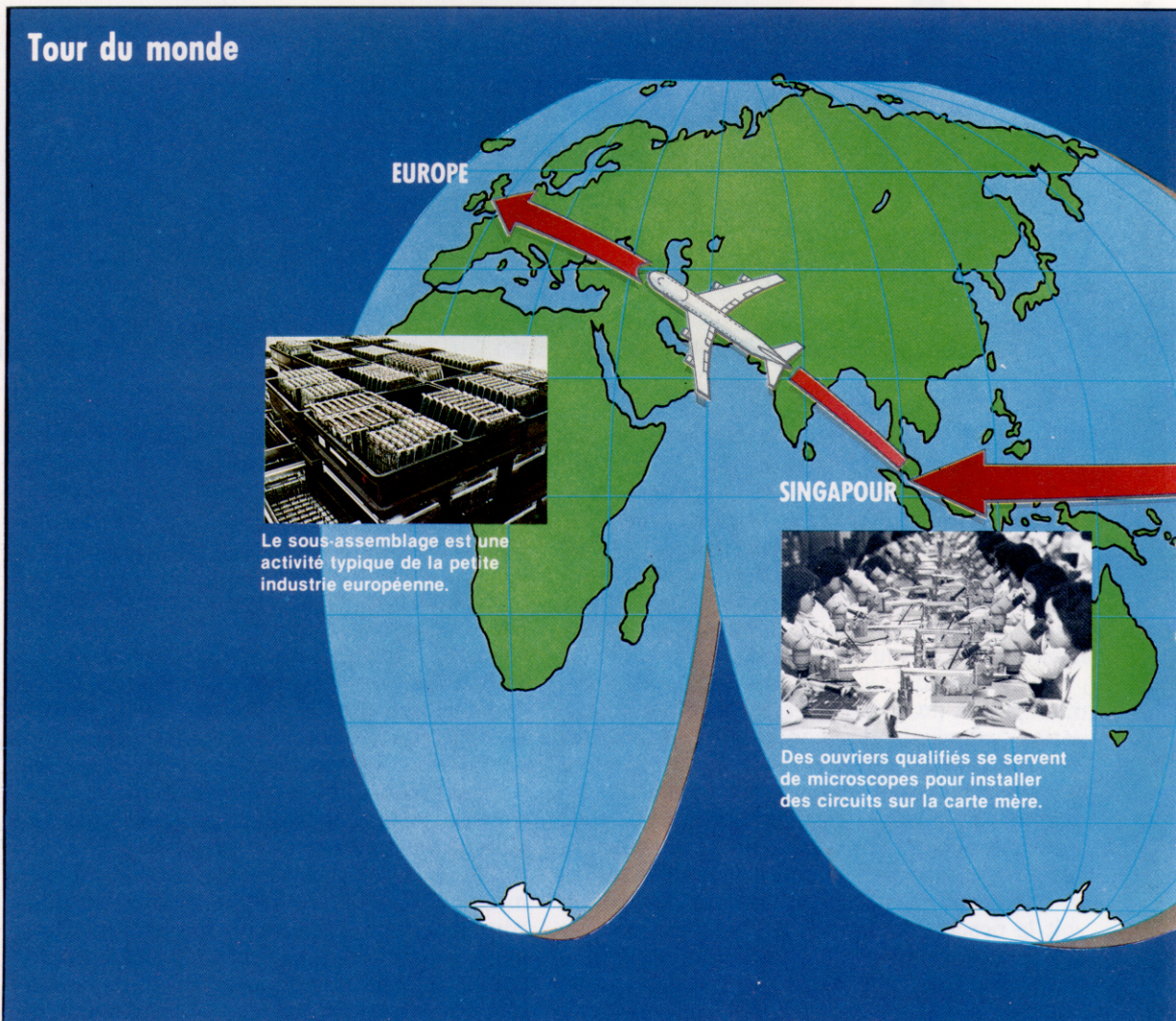
approvisionnés en éléments de base (des condensateurs aux puces mémoire). Le travail des ouvriers se borne à vérifier qu'aucun d'eux n'est à court de composants. Ceux-ci, quelle que soit la méthode utilisée, sont placés sur le circuit imprimé, de façon que leurs connexions soient en saillie par rapport à la partie inférieure du circuit. Des machines à souder continue déposent alors sur ces connexions un alliage liquide qui passe à travers les trous et se solidifie ensuite, assurant de bons contacts électriques.

Après des tests de vérification, les circuits sont placés dans leurs boîtiers, emballés et expédiés aux grossistes, qui se chargeront de les distribuer et de les mettre en vente. Le processus peut paraître simple, mais chacune de ses phases donne naissance à des problèmes particuliers.

## L'approvisionnement

Le premier est celui de l'organisation : tous les composants, en provenance de sources très diverses, doivent arriver en un seul endroit, prêts à être montés. C'est pourquoi le responsable qui se charge de leur achat est l'un des éléments fondamentaux de la chaîne. Il doit savoir les négocier au meilleur prix, déterminer et

### Tour du monde





synchroniser leur date de livraison. Tous les constructeurs savent que l'absence d'un seul composant peut contraindre les unités de production au chômage technique. Tout cela coûte de l'argent, et fait perdre des clients. Il est également vital d'obtenir les prix les plus bas possible. Une différence de quelques centimes sur un condensateur peut se révéler très profitable, vu le très grand nombre d'appareils fabriqués. D'où l'intérêt des pays à main-d'œuvre bon marché!

Par ailleurs, le montage des ordinateurs, qu'il soit manuel ou automatisé, donne inévitablement lieu à des erreurs. Les composants peuvent être placés au mauvais endroit, ou même être oubliés, la soudure peut se révéler médiocre. Parfois, certains éléments sont défectueux dès l'origine. Il faut donc faire des tests très complets, au départ comme à l'arrivée. La vérification des circuits, qui peut prendre des formes plus ou moins sophistiquées, est très coûteuse, parce qu'elle exige un matériel important. Mais c'est là un investissement nécessaire : un sous-traitant perdra vite son contrat si les appareils qu'il livre présentent des défauts. Il est d'ailleurs fréquent qu'un représentant du client s'assure sur place que les composants, comme le produit fini, sont en état de fonctionner.

La firme responsable du montage de l'Oric a même inventé un test de vérification original. Chaque machine terminée est pesée : si son poids est inférieur à la normale, c'est qu'il manque un ou plusieurs éléments. C'est pourquoi chaque boîtier est muni d'une étiquette bleue spécifiant son poids.

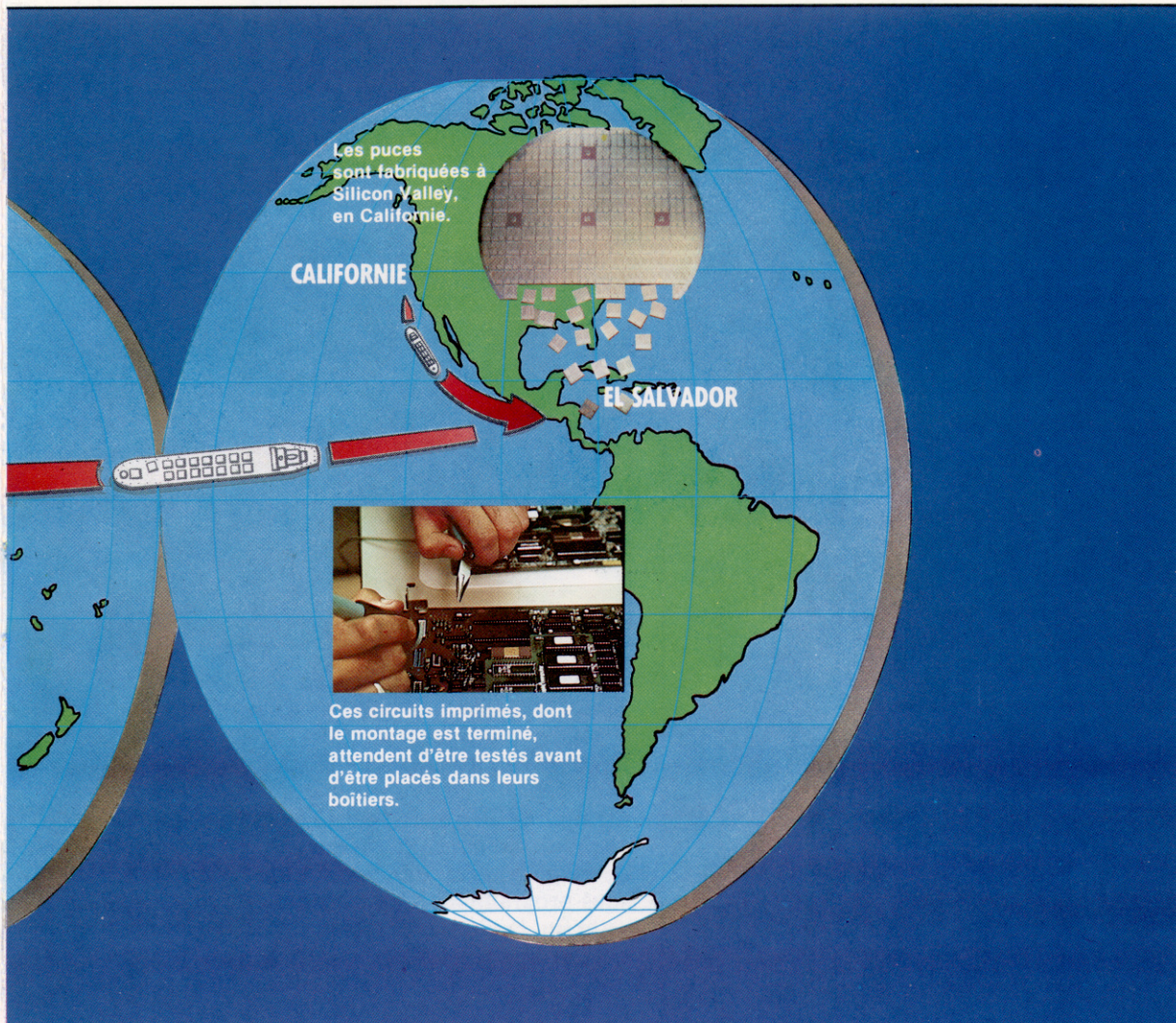
Enfin toutes les machines sont mises sous tension et raccordées à un récepteur de télévision. Les ordinateurs de gestion, par exemple, sont fréquemment soumis à un test très simple, qui consiste à les laisser branchés pour un jour ou deux d'affilée, tandis qu'ils exécutent leurs routines de fonctionnement, ou un programme particulier : on peut être ainsi sûr que tout va bien.

Vu le nombre de ces facteurs, il n'est donc pas surprenant que les micro-ordinateurs ne soient pas toujours disponibles, ou sujets à des pannes. La firme qui assure le montage dépend de ses fournisseurs si elle veut livrer à temps. Et la compagnie qui conçoit et distribue l'engin est liée à elle, pour le bon fonctionnement et les délais de livraison du produit. Un maximum de tests pour vérifier la fiabilité des micros coûte très cher. Répercutés sur les prix de vente, ils risquent de dissuader les clients. La voie des constructeurs d'ordinateurs familiaux est donc étroite.



#### Réduction des coûts

On fait aller et venir les composants d'un ordinateur d'un point du globe à l'autre pour réduire les coûts de fabrication, et par conséquent les prix de vente au détail. Dans le tiers monde, la main-d'œuvre est très bon marché. Toutefois les progrès de l'automatisation font que désormais il est possible de monter des ordinateurs en Europe, sans que ce soit plus cher.



#### Composants baladeurs

Cette carte du monde montre le mouvement des composants au cours du processus d'assemblage. (Cl. Kevin Jones.)

# Étapes d'une décision

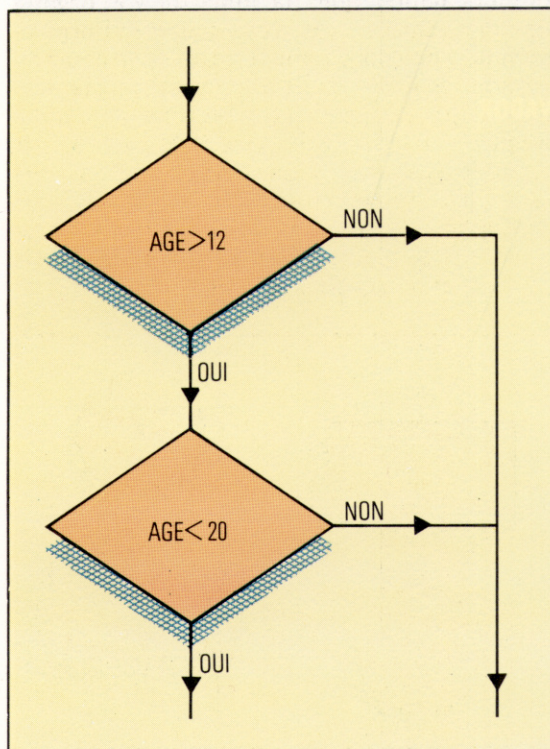
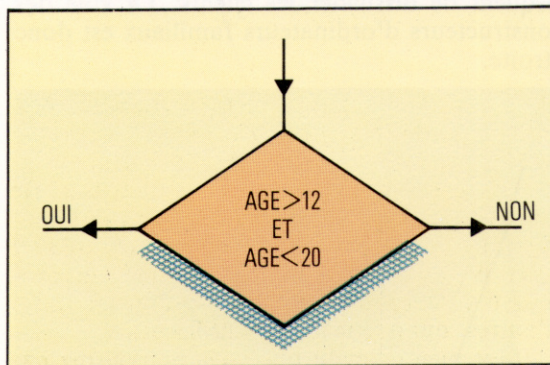
Grâce aux organigrammes, des décisions complexes peuvent être décomposées en éléments de décisions simples : les tables de décision servent aux cas qui ne peuvent être réduits.

Les programmeurs veulent souvent introduire des décisions complexes dans leurs programmes comme :

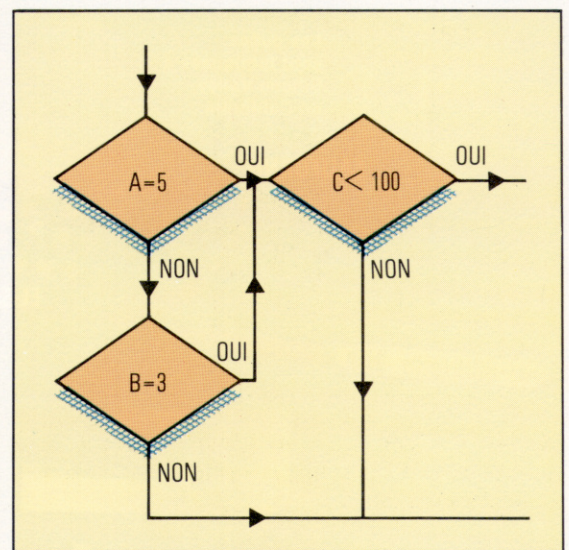
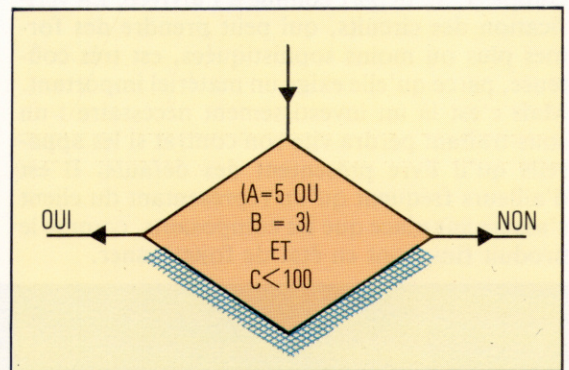
```
IF AGE>12 AND AGE<20 THEN
  STATUT="ADOLESCENT"
```

Les algorithmes d'instructions de ce type sont plus lisibles lorsque la décision est décomposée en ses diverses phases :

Nous avons fait figurer notre exemple en BASIC sous forme de diagrammes afin de mettre en évidence la plus grande clarté de la représentation « éclatée » de l'algorithme de décision.

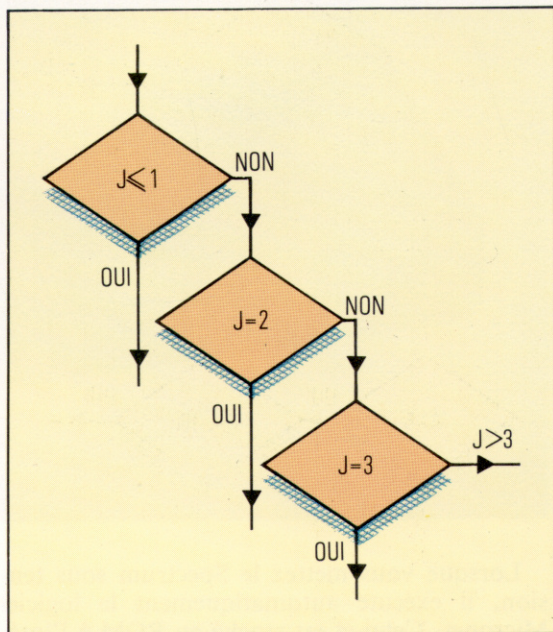
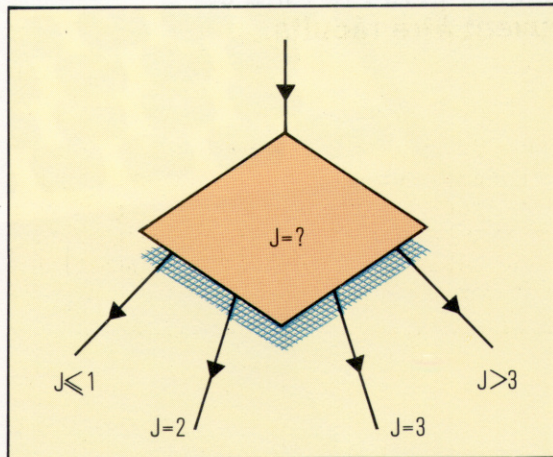


Notre deuxième exemple (ci-dessous) traite de trois décisions constitutives. Le circuit logique entre les pavés élémentaires de test de décision est davantage explicite que le seul algorithme. Un tel circuit fait apparaître une certaine similitude avec les règles booléennes.



Dans les exemples précédents, toutes les décisions reposent sur une alternative simple : répondre par OUI ou par NON. Il est cependant fréquent qu'un algorithme mette en jeu des décisions à plus de deux issues. S'il s'agit par exemple de lire une réponse au clavier qui correspond au choix d'un menu, le programme devra pouvoir se brancher sur un des nombreux sous-programmes pour exécuter cette option. A cette fin, la plupart des langages de programmation proposent diverses constructions syntaxiques de branchement, telles que CASE...OF... en PASCAL, ON...GOTO et ON...GOSUB en BASIC. Les

règles de décisions binaires s'appliquent également aux décisions à plus de deux solutions. Une seule issue est possible au pavé de test de décision, et chaque issue doit être explicitée, les chemins logiques possibles s'excluant mutuellement et couvrant toutes les possibilités.



Toutes les décisions multiples peuvent être représentées ainsi sous la forme d'un ensemble de décisions binaires.

### La table de décision

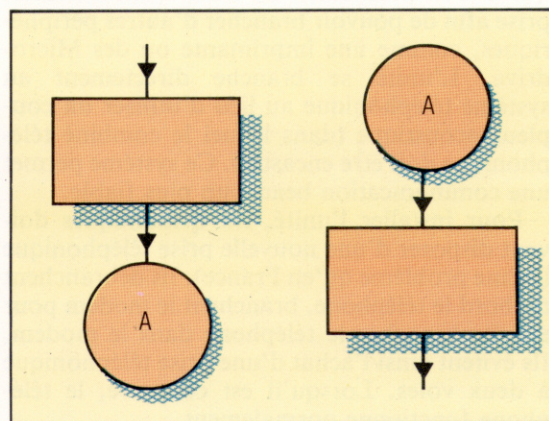
Les tables de décision constituent une alternative aux organigrammes lorsque les décisions sont multiples. Voici un exemple de table constituée d'un ensemble de règles pour prendre des décisions. La table comporte quatre rubriques principales : un texte décrivant les conditions définies par les règles, un texte énumérant les décisions possibles, une grille décrivant les règles en fonction des conditions, et une grille donnant les correspondances entre les décisions et les règles. Les cases de la grille « conditions-règles » donnent les valeurs des variables et,

CONDITIONS	RÈGLES							
	1	2	3	4	5	6	7	8
MISE À FEU	✓	✗	✗	✗	✓	✗	✓	✗
NIVEAU DE JEU	1	1	2	2	1	1	2	2
NIVEAU DU JOUEUR	NOVICE	NOVICE	NOVICE	NOVICE	EXPERT	EXPERT	EXPERT	EXPERT
DÉCISIONS								
ÉVASION DE L'ÉTRANGER			✓	✓			✓	✓
BOUCLERS MINÉS ALÉATOIRES			✓	✓			✓	✓
RÉDUCTION DE L'ÉNERGIE DE...	1 %	1 %	2 %	2 %	2 %	2 %	4 %	4 %

lorsqu'elles sont cochées en regard d'une décision, indiquent qu'il convient de les exécuter. Des valeurs servent de paramètres d'entrée pour la décision retenue. La règle 4, par exemple, se lira de la sorte : « Lorsque l'on appuie sur le bouton de mise à feu et que le niveau de jeu est 2, et lorsque le joueur est de niveau « novice », mettre en œuvre les boucliers minés aléatoires et réduire le niveau d'énergie de 2 % ». »

Un dernier mot à propos des organigrammes : efforcez-vous, partout où cela est possible, d'en limiter la taille à une page. En effet, lorsqu'il faut fouiller sur plusieurs pages pour essayer de s'y reconnaître dans un organigramme, cela devient fastidieux et constitue une perte de temps. Si vos algorithmes deviennent trop importants, faites en sorte de les subdiviser. De sorte que chaque routine d'un programme peut très bien être écrite pour représenter un seul pavé de traitement dans la totalité du programme. Et ce, même si cette routine en utilise d'autres, qui à leur tour en utilisent...

Il est bien entendu que vous ne pourrez pas toujours limiter ainsi la taille de vos organigrammes. Dans ce cas, vous couperez l'organigramme à un endroit qui ne gêne pas la compréhension générale du programme (après une décision par exemple). Vous représenterez sur la première page l'endroit exact de la coupure par un cercle, identifié par un symbole. Cela permet de lire, au premier coup d'œil, où passe le circuit logique. Sur la deuxième page, vous reproduirez en début d'organigramme le même cercle qui fera en quelque sorte la charnière.



Liz Dixon



# Liaisons

**Il se passe toujours quelque chose en Grande-Bretagne. Regardons comment le modem Prism VTX5000 favorise le développement du réseau Micronet, spécialisé dans l'informatique.**

La base de données Prestel, homologue en Grande-Bretagne de notre Télétel, a eu du mal à dénicher ses premiers clients. Les terminaux Prestel étaient chers et la quantité d'informations offerte était insuffisante pour attirer les utilisateurs potentiels. Cependant, avec un modem et un logiciel adéquat, la plupart des micros domestiques peuvent être reliés à Prestel. Afin d'exploiter cette possibilité, Prism constitua alors Micronet, un département séparé de Prestel consacré à l'actualité micro-informatique. Prism proposa rapidement des modems et des logiciels permettant d'être reliés à Micronet et à Prestel. Le succès ne tarda pas.

D'autant qu'un dispositif permettait de connecter le ZX Spectrum à Micronet, machine qui n'était pas destinée à ce genre d'utilisation. Elle ne possède pas d'interface RS232 ou interface série, ce qui signifie que des modems ordinaires ne peuvent pas être connectés. Son affichage est de 32 colonnes par 24 lignes, contre 40 par 24 pour Prestel. La société a donc produit une unité intégrée conçue spécialement pour cette tâche : le modem Prism VTX5000.

L'unité renferme tous les dispositifs requis pour effectuer une connexion avec le Spectrum : un modem à connexion directe de 1 200/75 bauds et un logiciel permettant d'interroger Prestel. Cela permet non seulement d'introduire les fonctions standards dans le système Prestel, mais aussi d'utiliser l'unité d'affichage du Spectrum comme un véritable écran télétexte.

Le VTX5000 se place sous le Spectrum et se connecte à son port d'extension. Le câble-ruban qui relie les deux unités est muni d'une troisième prise afin de pouvoir brancher d'autres périphériques, comme une imprimante ou des Microdrive. L'unité se branche directement au système téléphonique au lieu d'utiliser un coupleur acoustique (dans lequel le combiné téléphonique doit être encastré). Ce système permet une communication beaucoup plus fiable.

Pour installer l'unité, les Britanniques doivent disposer d'une nouvelle prise téléphonique (même problème qu'en France). Ils débranchent d'abord le téléphone, branchent le modem pour ensuite brancher le téléphone dans le modem. Ils évitent ainsi l'achat d'une prise téléphonique à deux voies. Lorsqu'il est connecté, le téléphone fonctionne normalement.

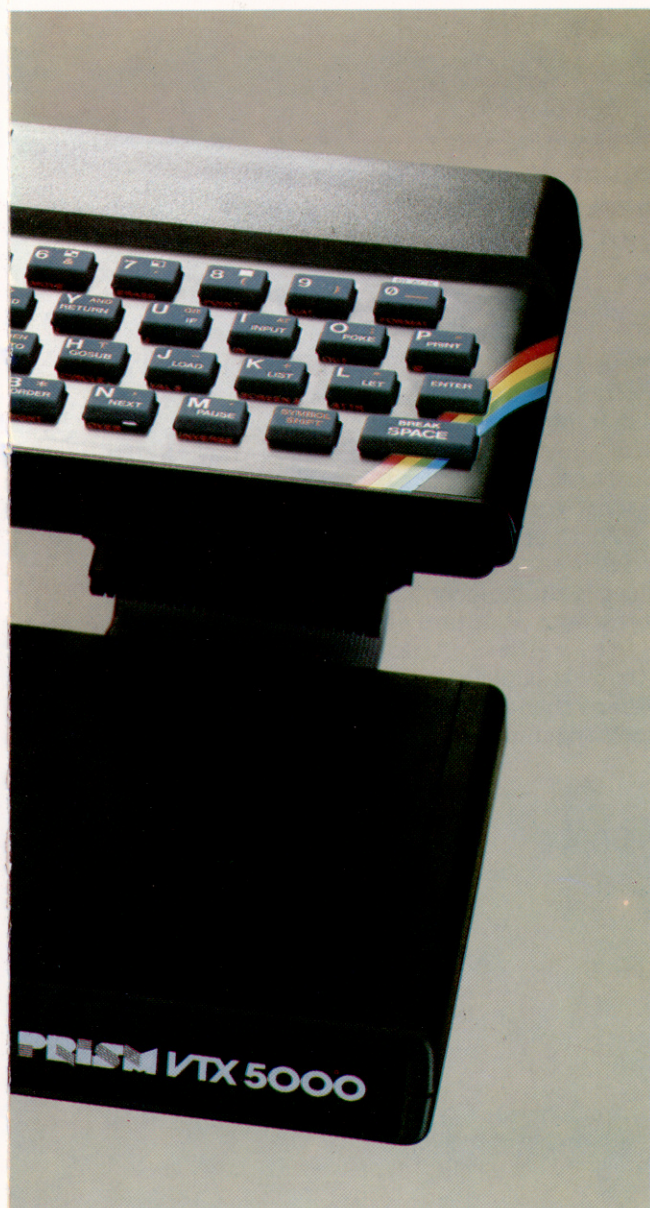


Lorsque vous mettez le Spectrum sous tension, il exécute automatiquement le logiciel Micronet. Celui-ci est stocké en ROM à l'intérieur du modem ; il ne doit donc pas être chargé avant son utilisation. Le logiciel Micronet est similaire aux programmes proposés pour d'autres micros. Ainsi, après avoir appris à s'en servir, il n'y a aucune difficulté à utiliser une autre machine. Le logiciel est commandé au moyen d'une série de menus ; il est simple à apprendre et à utiliser.

La première option du menu appelle la procédure d'entrée en communication qui implique l'entrée d'un numéro d'identification de dix chiffres. Ce nombre sera mémorisé par l'ordinateur tant qu'il sera sous tension et n'a donc à être entré qu'une seule fois au cours de chaque session, même si l'utilisateur appelle Prestel plusieurs fois.

L'étape suivante consiste à téléphoner à l'un des quatre ordinateurs qui gèrent la base de données Prestel. Lorsque le récepteur produit





un son aigu, il faut presser le bouton Line du modem et poser le récepteur. Le Spectrum communique alors avec Prestel. Il faut d'abord entrer un mot de passe de quatre lettres. Cela pour empêcher toutes fraudes sur les comptes Prestel. Il est possible de changer de mot de passe aussi souvent que l'on veut.

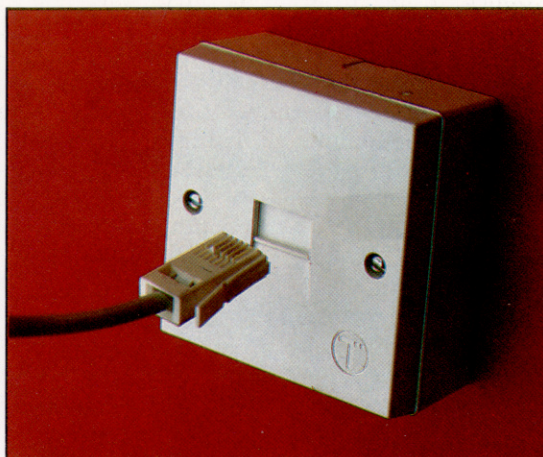
Après avoir accédé à la base de données, l'utilisation est la même qu'avec tout appareil Prestel. Les caractères \* et £ permettent de passer d'une page de Prestel à l'autre. Le Spectrum utilise Enter et Shift pour les mêmes fonctions. Il est possible d'appeler toute autre fonction Micronet à tout moment pendant la communication avec le système. Vous pouvez ainsi copier des pages de la base de données, et les stocker sur bande ou les imprimer. Il est même possible de copier des programmes entiers dans le Spectrum. Les pages télélogiciel de Micronet proposent des programmes gratuits et payants. Les programmes gratuits sont parfaits pour s'amuser, mais ils sont limités.

Micronet et Prestel offrent une immense gamme d'informations. En plus des nouvelles, ces systèmes proposent des conseils techniques, des blagues, des jeux, des lettres, des petites annonces, et ainsi de suite. Vous pouvez même envoyer du courrier électronique à des personnes qui utilisent le système régulièrement. Tout cela sur des pages « écran ».

Cette diffusion à grande échelle qu'offre Prestel est prometteuse. Mais certains des problèmes du début ne sont pas complètement résolus, surtout dans Micronet. Plusieurs pages nouvelles ne sont pas mises à jour, et le cheminement d'une page à l'autre n'est pas toujours évident. Cela dit, pourquoi ne pas essayer de communiquer avec d'autres utilisateurs de Spectrum. Le VTX5000 est aussi conçu pour relier deux ordinateurs Spectrum au moyen du réseau téléphonique, afin qu'ils puissent s'envoyer directement des messages et des programmes. La vitesse de transmission est de 1 200 bauds. Les premières versions du modem étaient vendues sans le logiciel de communication; mais maintenant une bande qui facilite cette transmission de données est fournie avec chaque unité. Évidemment, ce logiciel n'est destiné qu'aux utilisateurs de Spectrum ayant des amis qui en possèdent un muni du modem VTX5000.

Plusieurs autres normes de communication sont utilisées fréquemment par des modems, mais le VTX5000 ne peut s'y conformer. La restriction la plus importante est l'impossibilité pour le VTX5000 d'accéder aux systèmes de communiqués qui sont mis en place tout autour de la planète par des mordus de communication entre ordinateurs. Ceux-ci fonctionnent à 300 bauds, et le VTX5000 ne peut transmettre aussi lentement.

Toute personne désirant explorer les communications entre ordinateurs avec un Spectrum aurait avantage à se procurer une interface RS232 (en particulier l'interface 1 ZX de Sinclair) et à utiliser un modem d'usage général. Cependant, cela impliquera probablement la nécessité d'écrire son propre logiciel, de monter les câbles et ainsi de suite. Le VTX5000 est idéal pour les personnes ne désirant utiliser que Prestel.



#### Connexion directe

Le VTX5000 se connecte directement au Spectrum sans interface. Le téléphone, lui, se branche directement sur le VTX5000 qui est branché dans la prise téléphonique modulaire. (Cl. Chris Stevens.)

#### Nouveaux branchements

Voilà à quoi ressemblent les nouvelles prises du téléphone en Grande-Bretagne. Bien sûr, si l'idée est semblable, le produit équivalent en France (ou ailleurs) est différent.

# Parachutiste

Le but de ce jeu est d'obtenir le maximum de points en dix sauts de parachute. Mais parfois, ça ne s'ouvre pas! Paul Bunn a écrit ce programme pour les ordinateurs Atari.



Vous êtes dans un avion qui passe en haut de l'écran. Pour sauter, il vous suffit d'appuyer sur le bouton rouge du manche à balai; vous êtes alors en chute libre. Lorsque vous estimez être assez bas, dirigez le manche à balai vers le bas, et votre parachute s'ouvrira.

Vous aurez un bonus lorsque vous réussirez à atterrir sur l'aire prévue à cet effet. Ce bonus dépend du moment auquel vous déployez votre parachute. Plus tard vous l'ouvrirez, plus votre score sera élevé. Attention, il y a une limite à partir de laquelle le parachute ne s'ouvre plus. Bonne chance! Ce programme utilise des caractères redéfinis.

```

10 REM ** PARACHUTE- ECRIT PAR **
20 REM ** PAUL BUNN DECEMBRE 1982 **
25 IF PEEK(203)=PEEK(106)-8 THEN 40
30 GOSUB 10000:REM INITIALISATION
40 DIM AVION$(1),HOM$(1),CHUTE$(1),BASE$(1),LIGNE$(1)
50 AVION$=CHR$(ASC("%")+128):HOM$="!":LIGNE$=CHR$(ASC("#")+96)
55 FOR SAUT=1 TO 10
60 GRAPHICS 17:SETCOLOR 2,12,10:POKE 16,64:POKE 53774,64
61 POKE 53277,3:REM * MODE GRAPHIQUE "PLAVER MISSILE *
62 POKE 559,46:REM * RESOLUTION DOUBLE LIGNE *
63 POKE 54279,PEEK(204):REM * LOCALISATION DES GRAPHIQUES PA *
64 POKE 53256,3:POKE 53257,3:REM * TAILLE DES NUAGES (QUADRUPLE) *
65 POKE 704,8:POKE 705,14:REM * COULEUR DES NUAGES *
66 X3=INT(RND(0)*100)+75:POKE 53248,X3:REM ** POSITION NUAGE 1 ALEATOIRE **
67 X4=INT(RND(0)*100)+75:POKE 53249,X4:REM ** POSITION NUAGE 2 ALEATOIRE **
68 POKE 623,1:REM * PRIORITE ENTRE PLANS GRAPHIQUES *
69 POKE 53278,255:REM * INITIALISE LES REGISTRES DE DETECTION DE COLLISION *
70 POKE 756,PEEK(203):SETCOLOR 3,3,4:REM * POINT SUR JEU CARACTERE RAM AU LIEU D'E ROM *
90 X=1:BASE$="*":CHUTE$=CHR$(ASC("%")+128)
95 REM ** LIGNE DE BASE **
98 FOR P=0 TO 19
100 POSITION P,23
110 PRINT #6:LIGNE$;
120 NEXT P
130 BASE=INT(RND(0)*16)+2
140 POSITION BASE,22
150 ? #6:BASE$
155 REM ** BOUCLE PRINCIPALE **
160 POSITION X,0: ? #6:AVION$:POSITION X-1,0: ? #6;"
170 X=X+1:IF X<19 THEN 190
180 POSITION X-1,0: ? #6;" *X=1
185 REM ** SOUS-PROGRAMME MANCHE A BALAI S **
190 S1=STICK(0):S2=STRIG(0)
191 X3=X3-3:IF X3<20 THEN X3=200
192 X4=X4+3:IF X4>220 THEN X4=20
193 POKE 53248,X3:POKE 53249,X4
200 IF HOM=0 AND S2=0 THEN GOSUB 1000
205 IF HOM=1 THEN LOCATE X2,Y2,Z
210 IF HOM=1 THEN POSITION X2,Y2: ? #6:HOM$
220 IF CHUTE=1 THEN POSITION X2,Y2-1: ? #6:CHUTE$:FOR P=1 TO 100:NEXT P
222 IF PEEK(53252)<>0 OR PEEK(53253)<>0

```

```

225 IF Y2=22 AND HOM=1 THEN 2000
230 OX=X2:Y2=X2+(S1=7)-(X2>18)
240 OY=Y2:X2=X2-(S1=11)+(X2<1):Y2=Y2+1
250 IF S1=13 AND HOM=1 AND CHUTE=0 AND Y2<19 THEN CHUTE=1:FOR T=0 TO 50:SOUND 0,T,120,15:NEXT T:SOUND 0,0,0,0:C=Y2
255 IF HOM=0 THEN 160
260 POSITION OX,OY-1: ? #6;" "
270 POSITION OX,Y2-1: ? #6;" "
280 GOTO 160
290 REM ** OUVERTURE DU PARACHUTE **
1000 HOM=1:Y2=2:X2=X
1010 FOR P=30 TO 0 STEP -2:SOUND 0,P,120,15:NEXT P:SOUND 0,0,0,0:RETURN
2000 IF Z=ASC(BASE$) THEN GOTO 3000
2010 FOR G=0 TO 255
2020 SOUND 0,0,10,15:POKE 708,G
2030 NEXT G
2040 FOR G=255 TO 0 STEP -3
2050 SOUND 0,0,10,15:POKE 708,G
2060 NEXT G
2070 SOUND 0,0,0,0
2080 POSITION 0,0:POKE 708,104
2090 ? #6;">>>>PERDU<<<<"
2095 ? #6:"SCORE:";SCORE
2100 ? #6: ? #6;"TAPEZ START POUR JOUER"
2110 POKE 53279,0
2120 IF PEEK(53279)<>6 THEN 2110
2130 RUN
3000 POSITION 0,0
3010 IF CHUTE=1 THEN 3500
3020 ? #6;" VOUS N'AVEZ PAS"
3030 ? #6;" VOTRE PARACHUTE OUVERT"
3040 ? #6: ? #6;" OHCH ..."
3050 ? #6
3060 FOR P=0 TO 150 STEP 2
3070 SOUND 0,P,12,15
3080 NEXT P
3090 SOUND 0,0,0,0
3100 GOTO 2095
3500 ? #6:"BIEN JOUE..."
3510 ? #6:"ATTERRISSAGE REUSSI"
3540 POSITION 3,18: ? #6:"score:"
3550 BONUS=C*75
3560 FOR P=0 TO BONUS STEP 5
3570 POSITION 0,10: ? #6:SCORE+P
3580 SOUND 0,P*25,10,15
3590 NEXT P
3600 SCORE=SCORE+BONUS
3610 SOUND 0,0,0,0
3620 HOM=0:CHUTE=0:NEXT SAUT
3630 GRAPHICS 2:POKE 53248,0:POKE 53249,0
3640 ? #6;"VOUS AVEZ REUSSI"

```

```

3650 ? #6;" 10 ATERRISSAGES"
3660 ? #6
3670 ? #6:"VOTRE SCORE FINAL :-"
3680 ? #6;" ";SCORE
3690 SETCOLOR 2,0,0
3700 FOR P=5 TO 200 STEP 3
3710 FOR G=P-5 TO P+5
3720 SOUND 0,6,10,15
3730 NEXT G
3740 SETCOLOR 1,RND(0)*15,8
3750 NEXT P
3760 SOUND 0,0,0,0
3770 END
10000 RAM=PEEK(106)-8:A=256*RAM
10010 GRAPHICS 0
10020 SETCOLOR 4,9,2:SETCOLOR 2,9,2
10030 PRINT CHR$(127);"PATIENTEZ UN INSTANT"
10040 PRINT CHR$(127);"*****"
10050 POKE 752,1:PRINT
10055 REM ** COPIE LE JEU CARACTERE ROM EN RAM **
10060 FOR P=0 TO 1023
10070 POKE A+P,PEEK(57344+P)
10080 SOUND 0,PEEK(53770),10,2
10090 NEXT P:SOUND 0,0,0,0
10100 READ X
10110 IF X<0 THEN POKE 203,RAM:GOTO 1022
10120 FOR P=0 TO 7
10130 READ Y
10140 POKE (X*8)+A+P,Y
10150 NEXT P
10160 GOTO 10100
10165 REM ** DATA POUR REDEFINITION CARACTERES **
10170 DATA 1,153,189,153,126,60,60,66,129
10180 DATA 5,0,0,0,24,60,126,195,129
10190 DATA 3,255,255,255,255,255,255,255,255,255
10200 DATA 6,116,0,132,194,255,2,4,8
10210 DATA -1
10220 R=RAM-8
10225 REM
10230 FOR P=512 TO 768
10240 POKE R*256+P,0: NEXT P
10245 REM
10250 FOR P=0 TO 7
10260 READ X
10270 POKE R*256+70+P+512,X
10280 POKE R*256+50+P+640,X
10290 NEXT P
10300 POKE 204,R
10310 RETURN
10315 REM
10320 DATA 16,126,127,255,126,62,28,8

```



# Un ensemble complet

**Analysons l'Amstrad CPC qui, pour un prix moindre, offre un plus grand potentiel de traitement que la plupart des ordinateurs domestiques que l'on rencontre sur le marché.**

L'ordinateur Amstrad est offert en deux versions. Elles comportent le même ordinateur de base, mais proposent deux moniteurs différents, l'un monochrome, l'autre couleur RVB. L'alimentation est logée à l'intérieur du boîtier du moniteur et un câble transmet l'alimentation et les signaux du moniteur à l'ordinateur. Comme l'unité à cassette est aussi intégrée dans le boîtier de l'ordinateur, le système entier ne compte qu'un seul fil d'alimentation, ce qui limite le nombre de fils de connexion.

Le moniteur monochrome possède un écran vert produisant un affichage très net et très lisible. Il est parfaitement adapté à une utilisation professionnelle. Toutefois, l'image vibre légèrement; cela est probablement dû à la proximité de l'alimentation dans le boîtier du moniteur. Puisque le moniteur couleur n'autorise qu'une résolution moyenne, cela signifie qu'il ne peut produire de façon lisible un texte sur 80 colonnes.

Pour accompagner le moniteur monochrome de type gestion, l'Amstrad possède un clavier complet de type machine à écrire et un clavier numérique séparé. Toutes les touches peuvent être redéfinies pour créer d'autres caractères que ceux qui y sont indiqués. De plus, le clavier

numérique peut devenir un ensemble de touches de fonction programmables. Toute chaîne comptant jusqu'à 32 caractères peut être programmée sur chacune de ces touches. Une seule peut donc servir à charger ou à lister un programme ou encore à effacer l'écran.

L'unité à cassette est de conception standard. L'ordinateur peut commander le démarrage et la mise à l'arrêt du moteur d'entraînement lors d'un chargement ou d'une sauvegarde de programme. Deux vitesses de chargement sont disponibles : 1 000 ou 2 000 bauds; ces vitesses peuvent être sélectionnées par logiciel. Lors du chargement, l'ordinateur détecte automatiquement la vitesse utilisée pendant l'enregistrement et effectue les ajustements correspondants. Bien que la vitesse prudente de 1 000 bauds soit retenue par défaut, la vitesse rapide est parfaitement fiable dans la plupart des cas. Il s'agit de la vitesse utilisée par la plupart des bandes de programme offertes pour l'Amstrad.

L'Amstrad dispose, en outre, d'une large gamme de périphériques. Une imprimante peut être branchée directement dans le micro au moyen d'un connecteur plat. Il est possible d'ajouter des unités de disquettes, mais celles-ci ne sont pas encore offertes sur le marché (un



#### Options de moniteur

La version la plus économique de l'Amstrad offre un moniteur monochrome, mais la version livrée avec un moniteur couleur reste d'un prix raisonnable. La machine ne peut être achetée sans moniteur. Les deux programmes exécutés sont Wordhang, qui est une version de Hangman, et Admiral Graf, un jeu de combat naval. (Cl. Chris Stevens.)



lecteur de disquettes donnant une mémoire additionnelle à l'ordinateur, le langage LOGO et le système d'exploitation CP/M sont promis pour bientôt). L'Amstrad possède un seul port pour manche à balai de type Atari. Il dispose d'un haut-parleur intégré (avec commande de volume). Mais le son peut aussi être transmis à un amplificateur externe afin de produire un son stéréo à l'aide des trois voix distinctes de l'ordinateur. Une voix est transmise au haut-parleur de droite, une autre à celui de gauche et la troisième est mixée entre les deux. A l'aide d'une programmation adéquate, un objet peut non seulement être déplacé sur l'écran, mais le son qu'il produit peut sembler suivre son déplacement.

Il est très facile de commander le son à partir du BASIC, malgré la complexité du système. Il est non seulement possible de produire et d'interrompre une note dans l'octave et au volume désirés, mais l'enveloppe peut aussi être définie. L'enveloppe peut être ajustée pour imiter le son d'un piano ou d'une cloche. L'enveloppe du registre peut aussi être commandée indépendamment, afin de permettre la création d'effets sonores comme des sirènes ou des sifflets, tandis que chaque voix peut être mixée à du bruit, ce qui permet de simuler facilement des explosions et des tirs d'armes à feu.

La caractéristique la plus intéressante de l'Amstrad est son potentiel graphique. Trois modes d'affichage sont offerts, chacun produisant un nombre différent de caractères et de couleurs à l'écran. Chacun d'eux utilise les mêmes 16 K de mémoire. Le nombre de couleurs est inversement proportionnel au niveau de résolution et de format de texte. Le mode de résolution le plus élevé ne permet l'utilisation que de deux couleurs à la fois (premier plan et arrière-plan). L'affichage sur 80 colonnes est disponible et la machine autorise une résolution graphique impressionnante de 640 x 200. A l'autre extrême, le mode 20 colonnes permet d'utiliser 16 couleurs à la fois. Le troisième mode permet quatre couleurs sur 40 colonnes.

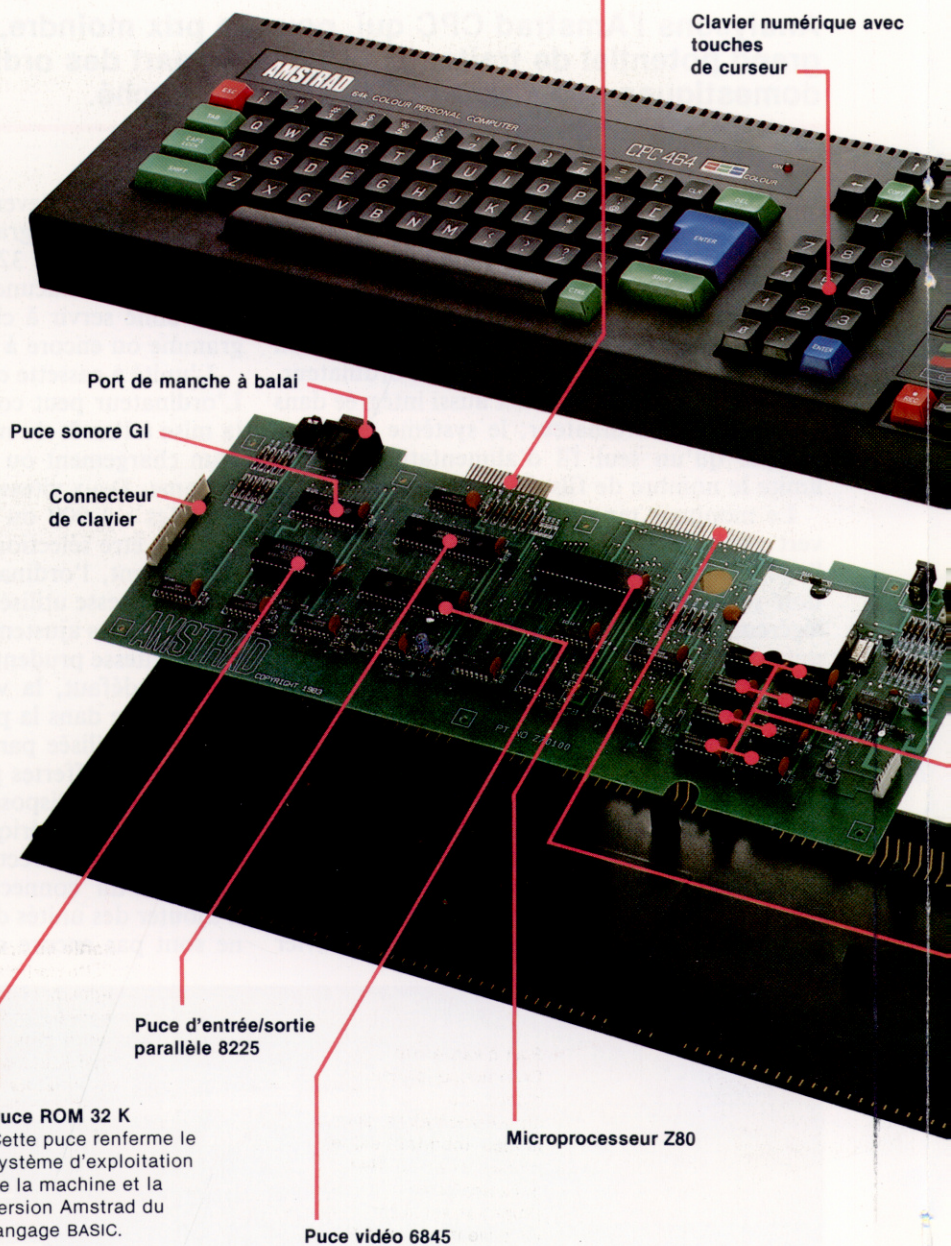
Si le nombre de couleurs pouvant être affichées simultanément est limité, on peut quand même choisir parmi 27 couleurs. Les bruns et les tons pastels peuvent être affichés comme les rouge, vert et bleu habituels. En plus de pouvoir créer facilement des dessins immobiles, l'utilisateur de l'Amstrad n'aura aucun mal à produire de l'animation. La seule lacune est l'absence de plan objet, mais la manipulation de l'écran à l'aide du BASIC permet de minimiser cet inconvénient.

Les 16 K occupés par l'affichage écran laissent de l'espace mémoire pour l'utilisateur. La RAM écran et la ROM BASIC occupent la même zone dans la topographie mémoire du processeur. Une puce spéciale à l'intérieur de l'Amstrad passe d'une utilisation à l'autre selon les besoins; 42 K sont donc disponibles pour les programmes et pour les données.

Le BASIC de l'Amstrad est l'une des versions les plus évoluées sur le marché. L'excellent

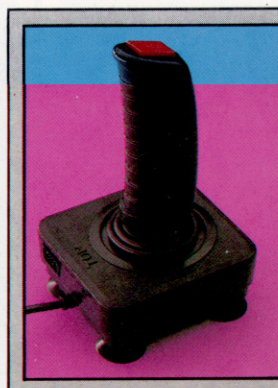
**Interface d'imprimante Centronics**  
Cette interface d'imprimante standard est très pratique;

malheureusement Amstrad a décidé d'utiliser un connecteur plat pour ce branchement et non le connecteur normal.



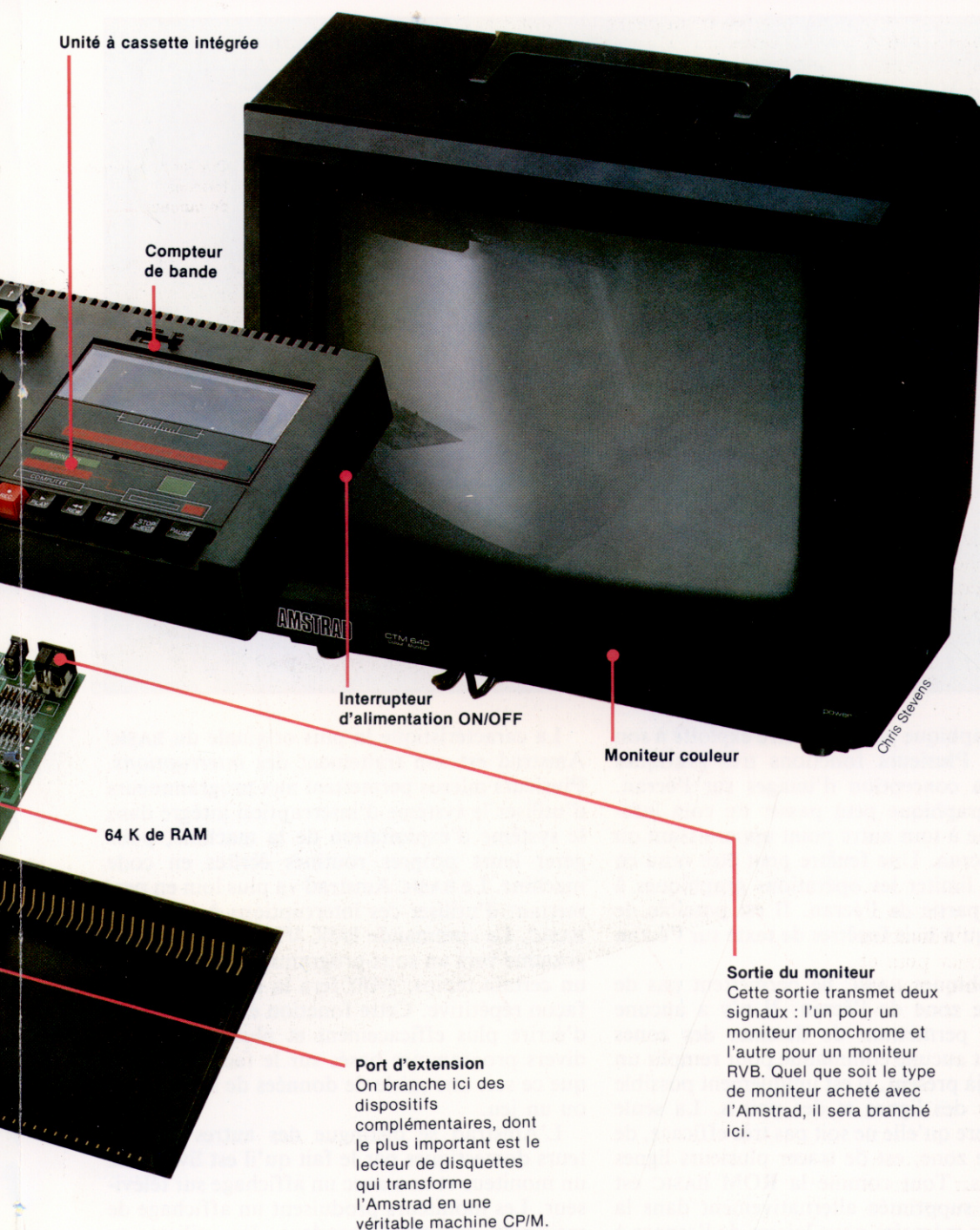
**Puce ROM 32 K**  
Cette puce renferme le système d'exploitation de la machine et la version Amstrad du langage BASIC.

**Microprocesseur Z80**



**Manches à balai**

L'Amstrad a une prise de manche à balai qui permet d'utiliser un manche à balai Atari standard. Cependant, la société fabrique son propre produit ce qui permet d'en brancher un deuxième. Les signaux du second sont transmis par le premier.



Unité à cassette intégrée

Compteur de bande

Interrupteur d'alimentation ON/OFF

Moniteur couleur

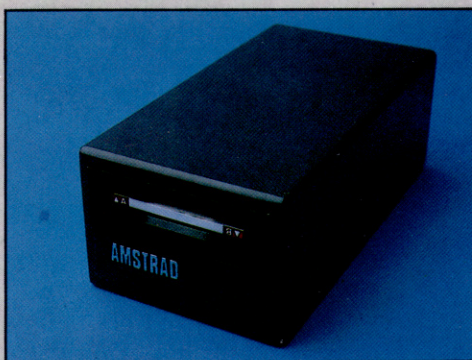
64 K de RAM

**Port d'extension**  
On branche ici des dispositifs complémentaires, dont le plus important est le lecteur de disquettes qui transforme l'Amstrad en une véritable machine CP/M.

**Sortie du moniteur**  
Cette sortie transmet deux signaux : l'un pour un moniteur monochrome et l'autre pour un moniteur RVB. Quel que soit le type de moniteur acheté avec l'Amstrad, il sera branché ici.

## Amélioration à venir

Amstrad envisage d'attaquer le bas de gamme des ordinateurs de gestion en lançant un lecteur de disquettes. Celui-ci comprendra le système d'exploitation CP/M requis par la plupart des programmes de gestion.



## Amstrad CPC 464

### PRIX

\*\* avec moniteur monochrome  
\*\*\* avec moniteur couleur

### DIMENSIONS

Clavier/unité à cassette  
565 x 170 x 70 mm  
Moniteur couleur :  
380 x 250 x 350 mm

### UC

Z80

### MÉMOIRE

64 K de RAM, dont 42 K sont disponibles pour les programmes BASIC.  
32 K de ROM.

### ÉCRAN

Trois modes avec mixage total du texte et des graphiques :  
640 x 200 (2 couleurs)  
320 x 200 (4 couleurs)  
160 x 200 (16 couleurs)  
Choix dans une palette de 27 couleurs.

### INTERFACES

Deux manches à balai, un port Centronics, un bus d'extension (lecteurs de disquettes), une sortie de son stéréo, une sortie moniteur.

### LANGAGES DISPONIBLES

BASIC (inclus), PASCAL (sur cassette).

### CLAVIER

De type machine à écrire, 74 touches.

### DOCUMENTATION

Le guide du débutant fourni avec la machine est facile à comprendre. Un manuel de référence BASIC et un manuel technique sont aussi disponibles.

### POINTS FORTS

Le moniteur et l'unité à cassette intégrés font de l'Amstrad un système complet par lui-même. Il dispose d'une mémoire importante, d'un excellent potentiel graphique et d'un son stéréo évolué.

### FAIBLESSE

L'unité à cassette n'est pas très robuste. Les commandes graphiques du BASIC ne permettent pas (encore) d'exploiter à fond les possibilités offertes par le matériel.



## Amsoft

Division Logiciel d'Amstrad, Amsoft a produit plusieurs douzaines de programmes pour l'Amstrad avant le lancement de la machine. Il s'agit surtout de jeux, versions écrites à l'origine pour d'autres machines. Cependant, plusieurs sociétés développent

actuellement des logiciels pour cette machine; la situation va donc changer très bientôt. Les photographies montrent Roland On The Ropes (gauche), Roland In The Caves (centre) de Indescomp et Oh Mummy de Gem Software (droite).

Chris Stevens

matériel graphique peut ainsi être exploité à son maximum. Plusieurs fonctions très pratiques facilitent la conception d'images sur l'écran. L'origine graphique peut passer du coin inférieur gauche à tout autre point apparaissant ou non sur l'écran. Une fenêtre peut être mise en place pour limiter les opérations graphiques à une petite partie de l'écran. Il est possible de définir jusqu'à huit fenêtres de texte sur l'écran à la fois.

Les graphiques BASIC ne permettent pas de colorer une zone de l'écran. Il n'y a aucune commande permettant de dessiner des zones continues et aucune commande pour remplir un contour déjà présent. Il est uniquement possible de dessiner des lignes et des points. La seule façon, encore qu'elle ne soit pas très efficace, de colorer une zone, est de tracer plusieurs lignes rapprochées. Tout comme la ROM BASIC est intégrée et supprimée alternativement dans la topographie mémoire pour laisser de l'espace à la mémoire écran, d'autres ROM peuvent être ajoutées et occuper le même espace. Les fonctions absentes dans la version actuelle du BASIC Amstrad pourraient apparaître sur une ROM accessoire et être fusionnées à l'ancien BASIC. De nouveaux langages comme le PASCAL, le FORTH et le LOGO pourraient ainsi être ajoutés. Ces ROM supplémentaires n'occupent pas plus d'espace mémoire que la version actuelle de BASIC; 42 K de RAM seront donc toujours disponibles pour leur utilisation.

Toutefois, une mémoire supplémentaire peut être inscrite dans la topographie mémoire, d'une manière analogue à ce que l'on a vu précédemment. Ainsi, la RAM de 64 K peut être étendue.

La caractéristique la plus originale du BASIC Amstrad est son traitement des *interruptions*. Plusieurs micros permettent aux programmeurs d'utiliser le système d'interruption intégré dans le système d'exploitation de la machine, pour gérer leurs propres routines écrites en code machine. Le BASIC Amstrad va plus loin en permettant d'utiliser ces interruptions à partir du BASIC. La commande BASIC AFTER dirige un programme vers un sous-programme spécifié après un certain temps. EVERY fera la même chose de façon répétitive. Cette fonction évoluée permet d'écrire plus efficacement et plus simplement divers programmes basés sur le facteur temps, que ce soit une saisie de données de laboratoire ou un jeu.

L'Amstrad se distingue des autres ordinateurs domestiques par le fait qu'il est livré avec un moniteur et non avec un affichage sur téléviseur. Les moniteurs produisent un affichage de meilleure qualité. Cependant, les utilisateurs qui ont acheté la version monochrome auront peut-être occasionnellement besoin d'un affichage couleur. Avec la version de base de l'Amstrad, il est impossible d'utiliser un téléviseur couleur bien qu'un adaptateur soit disponible en option. Il est possible d'utiliser un moniteur couleur séparé avec l'Amstrad sans l'aide d'un adaptateur, mais, puisque le micro tire son alimentation du moniteur monochrome, les deux moniteurs devront être sous tension en même temps.

Avec ses superbes graphiques, son matériel très fiable, un BASIC évolué et un excellent potentiel d'extension, l'Amstrad CPC 464 est l'un des ordinateurs domestiques les plus perfectionnés sur le marché.

# Perdu dans la jungle

**Ultimate, Play The Game, s'est fait une réputation par ses jeux d'arcades de grande qualité destinés au Spectrum. Regardons son dernier gros succès, Sabre Wulf.**

La baisse des ventes de jeux d'arcades a amené les compagnies productrices de logiciels à mettre au point des programmes de type nouveau, qui intègrent certains éléments du jeu d'aventures. Sabre Wulf est un bon exemple de cette nouvelle tendance. Cela confirme également l'intérêt du public pour l'usage des micros domestiques aux dépens des consoles uniquement destinées aux jeux. L'engouement pour ces derniers produits semble bien en déclin depuis un an.

Dans le jeu d'aventures classique, le joueur transporte le héros — souvent un personnage venu de la science-fiction ou de la littérature fantastique — d'un endroit à un autre, tout en résolvant de nombreuses énigmes. L'action s'interrompt fréquemment, parce qu'il faut venir à bout d'un problème d'apparence insoluble. Le jeu d'arcades, par contre, n'est guère porté sur la réflexion et ne nécessite que des réflexes rapides et un excellent doigté.

Sabre Wulf entend combiner les deux points de vue. C'est d'abord un jeu de labyrinthes, lointainement inspiré du célèbre Pacman, et qui se passe dans la jungle. Le héros fait un peu penser à l'Indiana Jones des *Aventuriers de l'Arche perdue*; il parcourt le labyrinthe en évitant les assaillants et en amassant des trésors. Le but ultime du jeu est de retrouver les quatre fragments d'une amulette magique brisée.

Le cadre est superbe, et le graphisme est l'un des plus beaux et des plus détaillés qu'on ait vus sur le Spectrum. Animaux, végétation, montagnes et grottes font parfois penser à une peinture du Douanier Rousseau. Le labyrinthe est extrêmement complexe, et en parcourir ne serait-ce que vingt pour cent en une seule partie est très difficile.

Le héros se débarrasse de la plupart de ses adversaires d'un simple coup d'épée. Il est pourtant contraint parfois de les affronter (tout comme dans Pacman), ce qui nécessite le recours à des armes particulières, disséminées çà et là, et qu'il faut découvrir. D'autres objets vous donnent une vie supplémentaire — ce qui sera fort utile, le jeu étant très difficile. Les orchidées en fleur sont plus ambiguës. Selon leur couleur, elles peuvent vous immuniser contre tout danger, vous changer en légume (temporairement!), doubler la vitesse à laquelle vous vous déplacez ou, pire encore, inverser les effets des contrôles! Mais leurs effets sont de courte durée, heureusement, et laissent une chance. Le jeu est admirablement conçu et réa-

lisé, bien qu'il souffre de certains petits défauts très courants. Les effets sonores, d'abord très amusants et très travaillés, deviennent vite lassants, et Ultimate n'a pas jugé utile de vous permettre de les supprimer. Par ailleurs, bien que ce soit en théorie un jeu à un ou deux joueurs, en fait un seul peut y jouer; il y a simplement deux panneaux pour indiquer le score, et un palmarès qui enregistre les six meilleurs résultats. A la manière des jeux d'arcades, on y entre ses initiales en déplaçant le manche à balai, ou en appuyant sur les touches du clavier.

Ultimate a, en revanche, eu l'heureuse idée de prévoir l'emploi de trois manches à balai de marques différentes — il en existe de plus en plus pour le Spectrum. Le jeu au clavier est beaucoup moins satisfaisant : déplacements et coups d'épée se règlent grâce aux touches des lettres. Elles sont l'une à côté de l'autre, et se révèlent donc très difficiles à manier.

Toutefois, ces quelques réserves mises à part, Sabre Wulf reste un jeu remarquable, un mélange très équilibré de jeu d'action et de jeu d'aventures. Il est digne des précédentes productions d'Ultimate et connaîtra sans doute le même succès.

**Sabre Wulf** : Spectrum 48 K.

**Éditeurs** : Ashby Computers and Graphics Ltd.

**Auteurs** : Ultimate, Play The Game.

**Manches à balai** : Kempston ou Interface 2.

**Format** : cassette.

#### Générique

On voit ici la page de titre du jeu. Sabre Wulf est un jeu de labyrinthes, où il faut affronter des « méchants » très réussis. Les bords de l'écran sont ornés de graphismes superbes évoquant des scènes de jungle. (Cl. Liz Heaney.)



# Le cours de collisions

Pour notre jeu « Champ de mines », nous allons maintenant regarder le contrôle au clavier des déplacements à l'écran et les parties du programme qui détectent les collisions.

Le BASIC BBC ne comporte pas moins de 4 commandes par touche du clavier. Le choix dépendra évidemment du but poursuivi. INKEY\$ et INKEY sont normalement utilisées lorsque l'on désire un délai avant une réponse au clavier et la suite du programme. GET et GET\$ stopperont toujours l'exécution du programme dans l'attente d'une frappe au clavier. Ces deux dernières commandes ont tendance à être utilisées lorsqu'une réponse à une question est nécessaire, par exemple « Une autre partie O/N? ». Si l'on utilise GET et GET\$, le programme attendra la réponse qui sera soit "O", soit "N". Nous pouvons utiliser une boucle qui répétera l'instruction GET jusqu'à ce que (UNTIL) la réponse soit "O" ou "N"; soit le programme suivant :

```
1000 PRINT « UNE AUTRE PARTIE O/N » ?
1010 REPEAT
1020 A$=GET$
1030 UNTIL A$="O" OR A$="N"
1040 Etc.
```

En utilisant GET\$ ou INKEY\$, la touche frappée est interprétée en tant que chaîne de caractères comme dans l'exemple ci-dessus. Si, en revanche, c'est GET ou INKEY, une valeur numérique est alors transmise du clavier; il s'agit du code ASCII pour la touche choisie en réponse. Ces options peuvent servir au programmeur pour découvrir les touches n'ayant pas d'équivalence standard, telle la touche « retour-chariot » ou les touches de mouvement du curseur. Cela aura alors les résultats suivants :

Curseur vers la gauche	136
Curseur vers la droite	137
Curseur vers le bas	138
Curseur vers le haut	139

Supposons que nous n'acceptons que les « curseurs vers la gauche » et « vers la droite » pour notre programme.

Le segment de programme suivant utilise INKEY pour attendre une réponse pendant un quart de seconde :

```
1000 *FX4,1:REM ACTIVER LE MODE VALEUR DU CODE ASCII
    POUR LE CURSEUR
1010 REPEAT
1020 A=INKEY(25)
1030 UNTIL A=136 OR A=137
1040 *FX4,0:REM RESTAURER LE CURSEUR EN MODE ÉDITION
```

Le paramètre 25 de la ligne 1020 demande à l'ordinateur d'attendre un vingt-cinquième de seconde avant de poursuivre le programme.

Ces instructions ne concernent pas le clavier lui-même mais cette zone de la mémoire de l'ordinateur appelée « tampon mémoire du clavier ». Il s'agit d'un espace mémoire temporaire de sauvegarde pour les caractères entrés au clavier. Les derniers caractères frappés se mettent à la queue des précédents, et le processeur « prend » les caractères par le début. Si vous frappez des caractères au clavier plus vite que le processeur ne peut les traiter, ils ne sont pas perdus, mais attendent leur tour dans le tampon. Du fait que INKEY, GET, INKEY\$ et GET\$ testent tous le début du tampon, il vous est impossible de savoir combien de temps un caractère y séjourne. Pour les jeux commandés au clavier, cela peut avoir pour conséquence des réponses trop longues et un manque de vivacité dans les mouvements à l'écran.

Il existe deux solutions à ce problème. La première consiste à toujours vider le tampon mémoire avant de le solliciter. Cela peut être fait en utilisant l'instruction \*FX15,1. Une autre possibilité est d'utiliser une autre variante de INKEY. Comme il a été dit, INKEY() attend un certain temps, spécifié par le nombre de la parenthèse, une frappe au clavier, avant de poursuivre avec le programme. Il est possible de faire en sorte que INKEY teste directement le clavier au lieu de son tampon, en spécifiant un nombre négatif dans les parenthèses qui suivent la commande. Chaque touche a un nombre négatif associé. La liste complète des nombres négatifs et leur signification est donnée à la page 275 du Guide de l'utilisateur. Nous utiliserons dans notre programme les touches de déplacement du curseur pour contrôler les mouvements à l'écran. Les valeurs des touches à utiliser avec INKEY sont les suivantes :

Curseur vers la gauche	-26
Curseur vers la droite	-122
Curseur vers le bas	-42
Curseur vers le haut	-58

La procédure suivante utilise INKEY pour tester directement le clavier sur la base des quatre flèches curseurs tour à tour. Lorsque l'une des touches est sollicitée, une autre procédure est engagée (« déplacement ») et deux paramètres sont remplis. Ces derniers contiennent des



informations relatives à la direction que doit prendre le détecteur de mines.

```

3000 DEF PROCtest_clavier
3010 REM ** vers le haut ? **
3020 IF INKEY(-58)=-1 THEN PROCdéplacement (0,-1)
3030 REM ** vers le bas ? **
3040 IF INKEY (-42)=-1 THEN PROCdéplacement (0,1)
3050 REM ** vers la droite ? **
3060 IF INKEY (-122)=-1 THEN PROCdéplacement (1,0)
3070 REM ** vers la gauche ? **
3080 IF INKEY (-26)=-1 THEN PROCdéplacement (-1,0)
3090 ENDPROC
    
```

## La procédure « déplacement »

La procédure « Lire-clavier » passe deux paramètres à la procédure « déplacement ». Ils sont affectés aux variables  $x$ -delta et  $y$ -delta pour être utilisés par la procédure « déplacement ». Ils correspondent aux modifications à apporter respectivement à la coordonnée  $x$  et à la coordonnée  $y$  du détecteur de mines. Ainsi, si l'on frappe par exemple sur la touche « curseur vers le haut », les valeurs 0 et -1 sont affectées à la procédure « déplacement ». Les instructions  $x_{det}=x_{det}+x\_delta$  et  $y_{det}=y_{det}+y\_delta$  mettent à jour les coordonnées du détecteur. Dans le cas « curseur vers le haut », 0 est ajouté à  $x_{det}$  et -1 à  $y_{det}$  (avec pour effet de soustraire 1 à  $y_{det}$ ).

Cela peut sembler signifier une sortie de l'écran vers le bas, mais il faut nous rappeler que la position d'origine est le coin supérieur gauche et que les valeurs en  $y$  s'accroissent vers le bas de l'écran. Réduire d'une unité  $y_{det}$  revient donc à monter la position sur l'axe des  $y$  d'une case. Vous pouvez vérifier que les valeurs passées pour les trois autres directions correspondent en fait à des modifications appropriées de  $x_{det}$  et  $y_{det}$ . Il serait tout à fait possible d'utiliser ce système pour ajouter des mouvements diagonaux. Voici le listage de cette procédure :

```

3220 DEF PROCdéplacement X_delta, y_delta
3230 REM ** EFFACEMENT DES ANCIENNES POSITIONS **
3240 COLOUR 1
3250 PRINTTAB(xdet, ydet): " "
3260 PRINTTAB(xhomme, yhomme): " "
3270 REM ** DEPLACEMENT DU DETECTEUR **
3280 xdet = xdet + x_delta
3290 ydet = ydet + y_delta
3300 REM ** TEST DES LIMITES **
3310 IF xdet>17 THEN xdet=17
3320 IF ydet>25 THEN ydet=25
3330 IF xdet<2 THEN xdet=2
3340 IF ydet<1 THEN ydet=1
3350 REM ** CALCUL DES COORDONNEES DE L'HOMME **
3360 xhomme = 19 - xdet
3370 yhomme = 26 - ydet
3380 PROCconversion(xhomme, yhomme)
3390 IF POINT(xgraph, ygraph)=2 THEN PROCexplosion (xgraph,
ygraph)
3400 PROCconversion(xdet, ydet)
3410 IF POINT(xgraph, ygraph)=2 THEN PROCmine trouvée
3420 PROCposition_chars
3430 ENDPROC
    
```

Avant que les coordonnées "x" et "y" du détecteur soient modifiées, il faut d'abord effacer les positions du détecteur et de l'homme. Les lignes 3250 et 3260 utilisent les anciennes valeurs de  $x_{det}$ ,  $y_{det}$ ,  $x_{homme}$  et  $y_{homme}$  pour recouvrir les anciens caractères. Du fait que les nouveaux sont affichés (PRINT) en rouge (couleur logique n° 1), la commande de couleur est utilisée à la ligne 3240 pour mettre la couleur courante du fond à 1. Les lignes 3280 et 3290 mettent à jour les coordonnées du détecteur ainsi que cela a été décrit plus haut. Avant d'afficher (PRINT) le détecteur

sur sa nouvelle position, il faut encore effectuer des tests pour s'assurer que les coordonnées ne sont pas incrémentées ou décrémentées en dehors de la zone définie comme « champ de mines ». Les limites inférieures et supérieures de  $x_{det}$  et  $y_{det}$  sont testées au segment de programme compris entre les lignes 3310 et 3340. Il a été décidé ici que lorsque le détecteur arrive en bordure de la zone, il y reste jusqu'à un ordre contraire. Par exemple la ligne 3310 vérifie si la limite droite du champ de mines a été atteinte, limite d'abscisse 17. Si l'on essaye d'attribuer à  $x_{det}$  une valeur supérieure à 17, cette ligne de programme remet la valeur à 17. Il aurait été également possible de créer un effet d'« enroulement » qui aurait rapporté le détecteur de la limite droite de la zone à la limite gauche de l'écran, donnant ainsi l'impression d'un déroulement continu. Il faudrait pour cela modifier la ligne 3310 de la sorte :

```
3310 IF xdet>17 THEN xdet=2
```

Vous pouvez ainsi modifier les quatre tests de bordure de zone et créer un déroulement continu dans tous les sens de déplacement à l'intérieur du champ de mines.

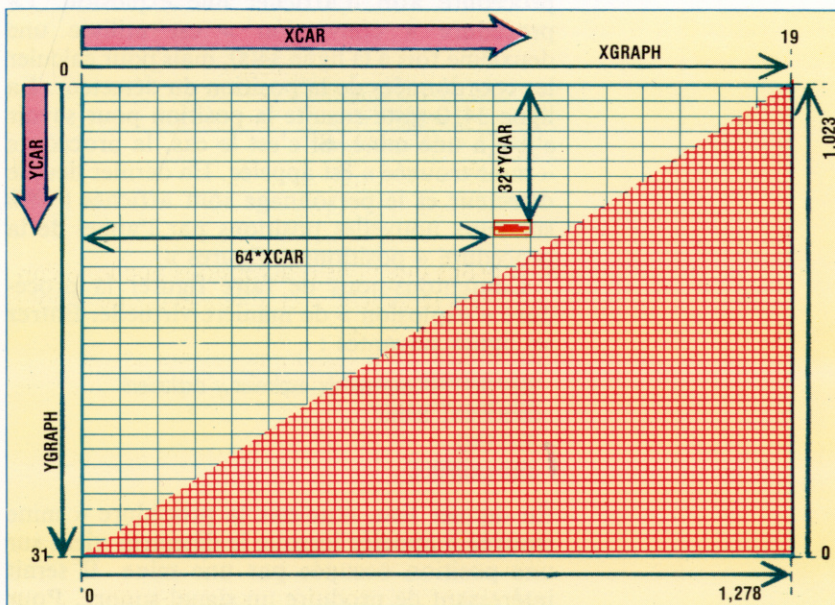
Une des règles de ce jeu est que le personnage représentant le joueur à l'écran (l'homme) se déplace automatiquement avec le détecteur, et selon un effet de symétrie de type miroir (orientation inversée). Ainsi, lorsque le détecteur va vers la gauche, le personnage va vers la droite et avance au même rythme. Pour programmer cette particularité, il faut automatiquement mettre à jour les coordonnées du personnage à chaque déplacement du détecteur. Elles sont liées à celles du détecteur par le biais de deux formules mathématiques indiquées aux lignes 3360 et 3370. Pour mettre en évidence cet effet de miroir, étudions la relation entre les abscisses ( $x_{homme} = 19 - x_{det}$ ).

A l'origine  $x_{det} = 2$  et  $x_{homme} = 17$ . Si le détecteur se déplace d'une position sur la droite,  $x_{det}$  va prendre la valeur 3. En appliquant la formule

### Étalonnage

Ce jeu utilise le mode graphique haute résolution conjointement avec l'affichage de type texte du BBC Electron. Cela présente certains avantages, mais cela suppose également deux systèmes distincts de coordonnées à concilier pour les graphiques et pour les textes. Le BBC Electron dispose de plusieurs formats d'affichage texte, chacun ayant son propre système de coordonnées. Ce jeu utilise le mode n° 5 qui permet 20 colonnes et 32 lignes.

Cet ordinateur dispose, par ailleurs, de trois modes de résolution graphique; mais ces derniers utilisent tous le même système de coordonnées. Il considère que tous les modes graphiques ont une résolution de 10280 par 1024. Le programme utilise le système de coordonnées haute résolution pour lire l'affichage texte basse résolution. Ce qui signifie convertir la position d'un caractère en coordonnées haute résolution. A cette fin, la coordonnée horizontale (XCHAR dans le programme) doit être multipliée par 64, et la coordonnée verticale (YCHAR), par 32. Un autre problème a dû être résolu: les coordonnées « texte » de l'écran commencent en 0 en haut de l'écran et décroissent, alors que le graphique commence en 0 en bas de l'écran et s'ajoute.



Liz Dixon

donnée ci-dessus, `xhomme` prendra la valeur :  $19 - 3 = 16$ . Ce qui signifie que le personnage se déplace à l'écran d'une position vers la gauche. Si `xdet` se déplace à nouveau vers la droite, il prendra la valeur 4 et, `xhomme` la valeur 15, et ainsi de suite. L'ordonnée `yhomme` suit la même progression.

Avant de pouvoir afficher (PRINT) le détecteur et le personnage étant sur de nouvelles positions ainsi définies, il nous faut vérifier s'ils vont aller sur des positions occupées par des mines. Le BASIC du BBC nous permet de tester un point de l'écran pour connaître sa couleur. `POINT(X,Y)` indique la couleur du pixel à la position définie par les coordonnées. Cela peut être utilisé pour vérifier si la couleur de la position qui va être rejointe est verte (si elle contient une mine, pour notre jeu). Il y a pourtant un problème : `POINT(X,Y)` utilise des systèmes de coordonnées haute résolution pour spécifier le point consulté. Si nous voulons utiliser cette commande pour notre jeu, il nous faut d'abord transcrire les coordonnées de la position en coordonnées graphiques. Le plus simple est de préciser le centre de la position.

La procédure suivante peut donc être adoptée pour convertir les coordonnées d'un caractère en coordonnées graphiques :

```
3720 DEF PROCconversion(xchar,ychar)
3730 xgraph=64*xchar+32
3740 ygraph=1023-(32*char+16)
3750 ENDPROC
```

Nous comprenons l'intérêt de pouvoir passer des paramètres d'une procédure à l'autre, en étudiant la procédure « déplacement ». La procédure « conversion » est utilisée deux fois : d'abord pour convertir les coordonnées du personnage en coordonnées graphiques, valeurs utilisées alors à la ligne 3390 pour tester si la couleur est le vert (souvenez-vous que, bien que les coordonnées du personnage aient été mises à jour, le caractère n'est pas encore affiché PRINT sur sa nouvelle position). Si la couleur est le vert, le programme se branche sur une autre procédure afin d'afficher une explosion. La procédure « conversion » est utilisée une deuxième fois à la ligne 3400, mais pour calculer les coordonnées de la position du détecteur. La ligne 3410 teste ensuite la position pour savoir s'il y a une mine. Si c'est le cas, la procédure « mine trouvée » est appelée. En dernier lieu, le détecteur et le personnage sont affichés (PRINT) sur leurs nouvelles positions par l'appel de la procédure « positionner figures ».

Contentons-nous de faire figurer la procédure « explosion » de manière virtuelle. Entrez les lignes suivantes :

```
3550 DEFPROCexplosion(x-explosion,y-explosion)
3560 PRINT"BANG"
3570 END
3580 ENDPROC
```

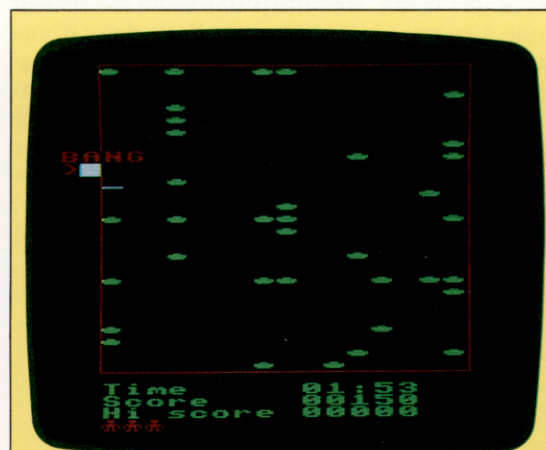
En dernier lieu, étudions la procédure « mine trouvée » appelée lorsque le détecteur vient sur une position occupée par une mine. Il serait intéressant de produire un signal sonore. Pour

le moment, il nous faut savoir que l'instruction `SOUND` à la ligne 3790 produit un son strident, sorte de « ping ». La fonction principale de cette routine est cependant d'incrémenter le score du joueur. Le programme utilise deux variables pour le score, la première étant une variable numérique incrémentée selon un pas de 150. Afin que le score soit toujours affiché (PRINT) sous forme d'un nombre à 5 chiffres, il faut ajouter des zéros en tête de ses valeurs numériques par des méthodes de traitement de chaînes de caractères. La procédure devient :

```
3770 DEF PROCmine_trouvée
3780 REM ** EFFET SONORE **
3790 SOUND 2,-15,170,3
3800 REM ** INCREMENTER LE SCORE **
3810 COLOUR 2
3820 score = score + 150
3830 score $ = STR$(score)
3840 score $ = LEFT$(zero$,5 - LEN(score$)) + score$
3850 PRINTTAB(11,28);score$
3860 ENDPROC
```

Les procédures indiquées ici peuvent être insérées dans vos programmes avec leurs numéros de lignes. La seule modification à apporter, pour que le programme puisse être exploitable, est d'insérer un appel de procédure « test-clavier » dans la boucle principale du programme d'appel. A cette fin vous ajouterez :

```
55 PROCtest-clavier
```



Ian McKinnell/Liz Heaney

### « Bang » sur la cible

Au point où nous en sommes du projet de jeu de « champ de mines », le programme remplira l'écran de mines disposées de façon aléatoire. Il créera votre personnage à l'écran et une représentation en miroir; il définira des tables pour le son et animera l'affichage. Comme le programme est encore incomplet, si vous heurtez une mine vous verrez seulement le mot `BANG` apparaître à l'écran. Prochainement, nous créerons une routine d'explosion avec les effets sonores que cela suppose. Il se peut aussi que vous rencontriez des messages d'erreur à certains moments. Ils sont dus à une structure incomplète de programme mais disparaîtront avec l'intégration des futurs sous-programmes finaux. Il reste que le programme est maintenant cohérent et représente un jeu d'action très rapide.

# Des réflexes rapides

Il faut reconnaître que nos réflexes sont de moins en moins rapides à mesure que nous avançons en âge. Testez vos temps de réaction en jouant au jeu très simple que nous vous proposons.

Nous aimons tous penser que nous avons de bons réflexes, et que notre réaction aux événements est quasi instantanée. Malheureusement, il nous faut, au mieux, un tiers de seconde pour réagir à un stimulus quelconque. Cela paraît satisfaisant jusqu'à ce qu'on se rende compte

que dans le même temps une voiture parcourt une dizaine de mètres... Par ailleurs, nos résultats peuvent varier en fonction de la fatigue, de la maladie ou de l'abus de boisson... Notre programme vous permettra de vous faire une idée précise de vos réflexes. Pour que le jeu soit plus équitable, vous aurez droit à cinq essais, qui donneront lieu ensuite au calcul d'une moyenne.

Vous jouez le rôle d'un pilote de l'espace chargé de faire parvenir des médicaments à une colonie installée dans une ceinture d'astéroïdes. Vous devez arriver le plus vite possible, mais aussi vous arrêter d'un seul coup dès qu'une collision se révèle imminente. Il suffit pour cela d'appuyer sur la barre d'espace. Le programme affiche le temps écoulé entre l'apparition des astéroïdes sur l'écran et l'appui sur la barre. Au bout de cinq essais, votre temps moyen de réaction est calculé. Pour prévenir toute tentation, l'ordinateur vérifie que vous ne maintenez pas une pression continue... Si oui, les moteurs de votre vaisseau s'arrêtent, et un signal d'avertissement se fait entendre. Votre radar de bord émettra une note très aiguë dès que les astéroïdes seront en vue. Essayez donc de modifier le programme de façon à recevoir un signal sonore ou visuel — mais pas les deux à la fois.

Il serait sans doute très intéressant de pouvoir choisir entre deux ou plusieurs options affichées à l'écran. Voyez si c'est possible. Le programme cesse alors d'être un simple indicateur de vos réactions et devient un test de vos capacités de jugement.

## Temps de réaction

```

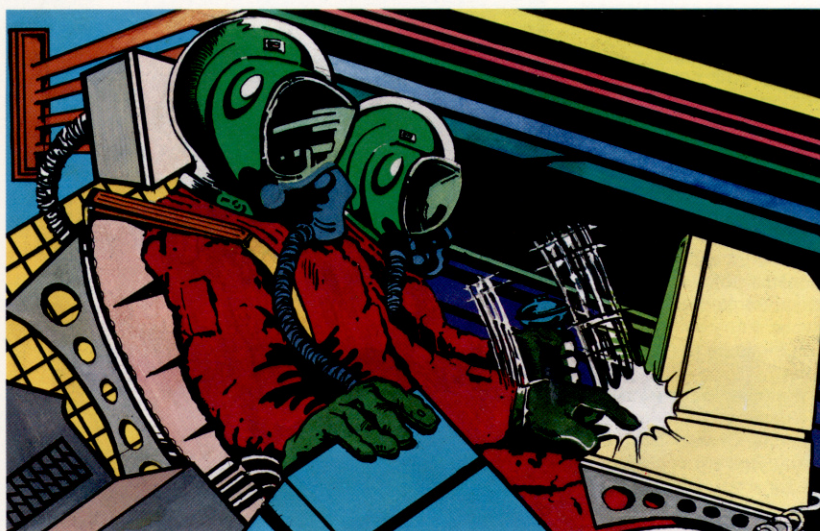
10 REM * pour le Spectrum
20 INK 7 : PAPER 0 : BORDER 0
30 FOR T=0 TO 31 : READ A
40 POKE USR "A"+T, A : NEXT T
50 DATA 123, 231, 255, 255, 245, 233, 123, 100
70 DATA 34, 187, 208, 49, 86, 96, 178, 205
80 DATA 67, 234, 45, 123, 198, 255, 29, 245
90 DATA 78, 108, 245, 68, 58, 168, 245, 188
100 LET H=0 : LET A=0 : LET V=0 : DIM R(5) : CLS
120 FOR Q=1 TO 5
150 FOR P=1 TO RND*158+58
160 PLOT INT (RND*258), INT (RND*178)
170 IF INKEY="" THEN GO TO 280
175 BEEP .1, -.3 : GO TO 160
180 NEXT P
190 FOR A=144 TO 147
220 PRINT AT RND*28, RND*38 : CHR$(A)
230 NEXT A
250 BEEP .05, .15 : LET C=0
260 LET C=C+1 : LET A=INKEY$
410 IF A=C THEN GO TO 360
420 LET R(Q)=C/66+.05
430 PRINT AT 19,81 : "IL VOUS A FALLU " : INT
(R(Q)*100) /100
440 PRINT " SECONDES POUR ARRÊTER "
450 PRINT " LES MOTEURS "
460 LET H=H+R(Q)
470 FOR J=1 TO 300 : NEXT J
475 NEXT Q
480 LET A=H/5 : CLS
490 PRINT "SUR CINQ ESSAIS VOTRE TEMPS DE REACTION"
495 PRINT " MOYEN A ETE DE " : INT (A*100)/100
495 PRINT " SECONDES "
500 FOR Q=1 TO 5 : LET V=V+ABS (R(Q)-A)
510 NEXT Q
520 PRINT " PRINT " VOTRE TEMPS DE REACTION A VARIE DE "
530 PRINT INT (V*28/5) " POUR CENT "
540 PRINT " PRINT " UNE AUTRE PARTIE ? (O/N) "
550 LET R=INKEY$ : IF R="O" THEN GO TO 150
560 IF R="N" THEN GO TO 550
570 INK 0 : PAPER 7 : BORDER 7 : CLS

```

```

10 REM * pour le BBC MICRO et l'ELECTRON
20 MODE 1 : CLS : VDU 23:228:181:0 : DIM REACT(5)
30 REM *réaction caractères (astéroïdes)
40 FOR J=0 TO 3
50 VDU 23:237:J,78:100,245:68:58:168:245:189
60 VDU 23:233:J,89:67:34:156:123:208:256:29
70 NEXT J
80 SOMME=0 : MOYENNE=0 : VARIE=0 : CLS
90 FOR Q=1 TO 5
100 REM *trace les étoiles
110 FOR DELAI=1 TO 188:RND(128)
120 PLOT 59,RND(1200), RND(1812)
130 IF INKEY(-99)=0 THEN 150 ELSE
140 SOUND 1,-15,.5,1 : GOTO 130
150 NEXT RETARD
160 REM *trace les astéroïdes sur l'écran
170 FOR ASTEROIDES =233 TO 248
180 X=RND(28) : Y=RND(28)
190 PRINT TAB(X,Y) "H*(ASTEROIDE)"
200 NEXT ASTEROIDE
210 SOUND 2,-15,158,2
220 REM *toste la barre d'espace
230 TIME=0
240 IF INKEY(-99)=0 THEN 240
250 REACT(Q)=TIME/100
260 PRINT TAB(3,26) : "IL VOUS A FALLU "
270 PRINT REACT(Q) : " SECONDES "
280 PRINT TAB(3,28) : "PLUS ARRÊTER LES MOTEURS"
290 FOR I=1 TO 300 : NEXT I
300 SOMME=SOMME+REACT(Q)
310 NEXT Q
320 FOR I=1 TO 3000 : NEXT I
330 REM *CALCUL LE TEMPS DE REPONSES
340 MOYENNE=SOMME/5
350 FOR Q=1 TO 5
360 VARIE=VARIE+ABS(REACT(Q)-MOYENNE)
370 NEXT Q
380 CLS : PRINT : PRINT
390 PRINT "SUR CINQ ESSAIS VOTRE TEMPS DE REACTION"
400 PRINT " MOYEN A ETE DE " : MOYENNE : "SECONDES."
410 PRINT " VOTRE TEMPS DE REACTION A VARIE DE "
420 PRINT INT(VARIE*28/MOYENNE) : "POUR CENT"
430 PRINT " PRINT " UNE AUTRE PARTIE ? (O/N) "
440 R=INKEY$(0) : IF R="O" THEN 80
450 IF R="N" THEN 440 ELSE MODE 4

```



Mike Cilewicz

# Cercle de lumière

**Malgré ses grandes qualités, il manque au BASIC BBC une fonction importante : celle de tracer des cercles. Voici comment écrire une routine en langage machine pour le faire.**

Tracer un cercle est facile, mais traiter une équation mathématique pour tracer les points sur un cercle avec rapidité l'est moins. Le plus simple est d'utiliser les fonctions COS et SIN, comme dans cet exemple :

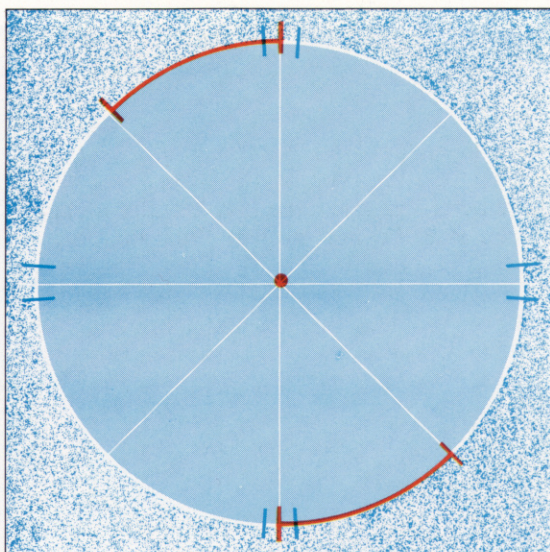
```
DEF PROCCIRCLE (XORG,YORG,R)
  MOVE XORG+R, YORG
  FOR THETA=0 TO 2*PI STEP PI/32
    X=R*COS(THETA)
    Y=R*SIN(THETA)
    DRAW X+XORG, Y+YORG
  NEXT THETA
ENDPROC
```

Du fait que cette procédure utilise les fonctions SIN et COS, elle est assez longue dans son exécution. Le tracé du cercle est relativement lent. Cependant, on peut l'accélérer en utilisant l'algorithme suivant, que l'on peut dériver de la géométrie élémentaire et du calcul différentiel :

```
3 MODE1
4 PROCCIRCLE(500,600,200)
7 END
10 DEF PROCCIRCLE(XORG,YORG,R)
20 Y=R
30 FOR X=1 TO Y*.707
40 Y=Y-X/Y
50 PROCPOINTS(X,Y)
60 NEXT
70 ENDPROC
80 DEF PROCPOINTS(X,Y)
```

## Images miroir

Le programme en langage machine utilise une équation pour tracer le cercle, qui est spécialement choisie pour sa rapidité. Néanmoins, elle serait lente à utiliser pour dessiner tout le tour du cercle, en calculant chaque point. C'est parce que les moitiés supérieure et inférieure du cercle sont symétriques que nous n'avons besoin de calculer que l'équation de la moitié supérieure. Ce même principe peut être étendu parce que la moitié gauche d'un cercle est symétrique de la moitié droite. En pratique, le programme ne calcule qu'un huitième de cercle. Chaque point de ce huitième est copié sept fois sur d'autres parties pour former le cercle entier. (Cl. Liz Dixon.)



```
90 PLOT 69,XORG+X,YORG+Y
100 PLOT 69,XORG-X,YORG+Y
110 PLOT 69,XORG-X,YORG-Y
120 PLOT 69,XORG+X,YORG-Y
130 PLOT 69,XORG+Y,YORG+X
140 PLOT 69,XORG-Y,YORG+X
150 PLOT 69,XORG-Y,YORG-X
160 PLOT 69,XORG+Y,YORG-X
170 ENDPROC
```

Cette routine dessine le cercle en huit sections en même temps, ce qui rend le tracé plus rapide. L'algorithme est également meilleur que notre routine initiale parce qu'il ne requiert pas le calcul des sinus et cosinus pour chaque point, mais seulement d'effectuer une division pour chacun. C'est toujours assez long.

Voici une autre routine qui ne fait appel à aucune mathématique compliquée :

```
10 MODE4
20 PNUM=69
30 PROCCIRCLE(500,600,200)
40 END
50 :
60 DEF PROCCIRCLE(X,Y,R)
70 VDU29,X,Y;:REM ORIGINE DU GRAPHIQUE
80 X=0:Y=R:D=3-2*R:REM VARIABLES
90 REPEAT
100 PROCCPLOT
110 IFD<0:D=D+4*X+6:ELSE D=D+4*(X-Y)+10:Y=Y-4
120 X=X+4
130 UNTIL X>Y:ENDPROC
140 :
150 DEF PROCCPLOT
160 PLOT PNUM,X,Y
170 PLOT PNUM,Y,X
180 PLOT PNUM,Y,-X
190 PLOT PNUM,-X,Y
200 PLOT PNUM,-X,-Y
210 PLOT PNUM,-Y,-X
220 PLOT PNUM,-Y,X
230 PLOT PNUM,X,-Y
240 ENDPROC
```

Elle est connue sous le nom d'algorithme de Bresenham. Elle est bien plus rapide car elle n'effectue que des additions, soustractions et multiplications par 2 ou 4 (qui peuvent se faire par décalage de bit). Nous l'utilisons dans notre programme de tracé de cercle en langage machine.



## Cercles BBC

```

30PNUM=69
40Sx=2
50OSWRCH=&FFEE
60DIM CODEX 600
70DIM DX 12
80X=DX+2
90Y=DX+4
100D=DX+6
110N1=DX+8
120N2=DX+10
130PROCCOMPIL
140MODE4
150PROC_OLYMPIC
160END
170#
180DEF PROCCOMPIL
190FOR IX =0 TO SX STEP SX
200KX=PK
210PX=CODEX
220COPT IX
230.CIRCLE
240JSR INIT
250:
260.LOOP
270JSR COMPXY:BMI DOIT:JSR CPLOT:RTS
280.DOIT
290JSR CPLOT
300LDA D+1:BPL D_IS_POS
310:
320.D_IS_NEG
330JSR DNEG
340JSR ADD_4_TO_X
350JMP LOOP
360:
370.D_IS_POS
380JSR DPOS
390JSR ADD_4_TO_X
400JMP LOOP
410:
420.INIT
430LDY #0
440.L7
450LDA #0:Y:STA #0:Y
460DEY:BPL L7
470INY
480LDA (#0):Y:STA X
490LDA (#0):Y:STA Y
500LDA (#0):Y:STA D
510INY
520LDA (#0):Y:STA X+1
530LDA (#0):Y:STA Y+1
540LDA (#0):Y:STA D+1
550LDA #29:STA DX+1
560LDA #0:STA DX
570JSR PSTR
580LDA #25:STA DX
590LDA #PNUM:STA DX+1
600JSR SETD
610RTS
620:
630.COMPY
640LDA X:STA N1:LDA X+1:STA N1+1
650LDA Y:STA N2:LDA Y+1:STA N2+1
660JSR SUB
670LDA N1+1
680RTS
690:
700.CPLOT
710LDA #4
720.L2
730JSR P2
740DEX:ENE L2
750RTS
760:
770.DNEG
780LDA X:STA N1:LDA X+1:STA N1+1
790JSR TIMES4
800LDA #6:STA N2:LDA #0:STA N2+1
810JSR ADD
820LDA D:STA N2:LDA D+1:STA N2+1
830JSR ADD
840LDA N1:STA D:LDA N1+1:STA D+1
850RTS
860:
870.DPOS
880LDA X:STA N1:LDA X+1:STA N1+1
890LDA Y:STA N2:LDA Y+1:STA N2+1
900JSR SUB
910JSR TIMES4
920LDA #10:STA N2:LDA #0:STA N2+1
930JSR ADD
940LDA D:STA N2:LDA D+1:STA N2+1
950JSR ADD
960LDA N1:STA D:LDA N1+1:STA D+1
970JSR SUB_4_FORM_Y
980RTS
990:
1000.ADD_4_TO_X
1010LDA #4:STA N1:LDA #0:STA N1+1
1020LDA X:STA N2:LDA X+1:STA N2+1
1030JSR ADD
1040LDA N1:STA X:LDA N1+1 STA X+1
1050RTS
1060:
1070.SUB_4_FORM_Y
1080LDA#4:STA N2:LDA #0:STA N2+1
1090LDA Y:STA N1:LDA Y+1:STA N1+1
1100JSR SUB
1110LDA N1:STA Y:LDA N1+1:STA Y+1
1120RTS
1130.SETD
1140LDA #0:STA X:STA X+1
1150LDA D:STA Y:LDA D+1:STA Y+1
1160ASL D:ROL D+1
1170LDA #3:STA N1:LDA #0:STA N1+1
1180LDA D:STA N2:LDA D+1:STA N2+1
1190JSR SUB
1200LDA N1:STA D:LDA N1+1:STA D+1
1210RTS
1220:
1230.P2
1240JSR PSTR
1250JSR SWAPXY
1260JSR PSTR
1270JSR NEGY
1280RTS
1290:
1300.TIMES4
1310ASL N1:ROL N1+1
1320ASL N1:ROL N1+1
1330RTS
1340:
1350.ADD
1360CLC
1370LDA N1:ADC N2:STA N1
1380LDA N1+1:ADC N2+1:STA N1+1
1390RTS
1400:
1410.SUB\ (N1=N1-N2)
1420SEC
1430LDA N1:SBC N2:STA N1
1440LDA N1+1:SBC N2+1:STA N1+1
1450RTS
1460:
1470.PSTR
1480LDY #250
1490.L1
1500LDA DX-250:Y
1510JSR OSWRCH
1520INY
1530ENE L1
1540RTS
1550:
1560.SWAPXY
1570LDA X:PHA:LDA X+1:PHA
1580LDA Y:STA X:LDA Y+1:STA X+1
1590PLA:STA Y+1:PLA:STA Y
1600RTS
1610:
1620.NEGY
1630LDA #0:STA N1:STA N1+1
1640LDA Y:STA N2
1650LDA Y+1:STA N2+1
1660JSR SUB
1670LDA N1:STA Y
1680LDA N1+1:STA Y+1
1690RTS
1700:
1710NEXT
1720ENDPROC
1730#
1740DEF PROCCIRCLE(P1%,P2%,P3%)
1750CALL CIRCLE,P1%,P2%,P3%
1760ENDPROC
1770#
1780DEF PROC_OLYMPIC
1790PROCCIRCLE(300,600,150)
1800PROCCIRCLE(650,600,150)
1810PROCCIRCLE(1000,600,150)
1820PROCCIRCLE(475,450,150)
1830PROCCIRCLE(825,450,150)
1840VDU29,0:0:
1850MOVE100,250
1860DRAW100,800
1870DRAW1200,800
1880DRAW1200,250
1890DRAW100,250
1900ENDPROC

```

Cette routine permet de dessiner des cercles sur le BBC et l'Electron. L'utilisation de la routine en langage machine consiste simplement à mettre les valeurs requises dans trois variables entières (INTEGER, soit X %, Y % et R %) et à appeler la routine avec la commande : CALL CIRCLE, X %, Y %, R %. On dessinera un cercle de rayon R % et de centre (X %, Y %). Notez que l'origine du graphique est déplacée vers le centre du cercle par la routine. On peut réinitialiser avec VDU 29,0:0;. Malheureusement, l'instruction CALL ne permet pas d'utiliser des expressions comme paramètres, et donc X %, Y % et R % ne peuvent être que des variables. Pour surmonter cette difficulté nous utilisons la procédure PROCCIRCLE (comme indiqué dans le listage).

Pour permettre au programme de marcher dans tous les modes graphiques, nous avons utilisé la commande VDU 25 pour tracer tous les points sur le cercle. Le programme est structuré en sous-programmes autonomes qui aident à la compréhension. On n'a utilisé que des techniques simples pour éviter toute confusion. Le prix de cette clarté a été une perte de vitesse dans l'exécution du programme. Néanmoins, pour dessiner un cercle de rayon de 300 unités, le programme l'exécute en langage machine en 0,52 seconde seulement, comparé au 1,9 seconde pour la version BASIC initiale.

Des améliorations utiles peuvent être apportées à la routine en modifiant la valeur de PNUM à la ligne 30, de 69 à 5. Le programme tracera

alors des lignes au lieu de points, et dessinera ainsi un disque plein en couleurs au lieu d'un cercle. Ce changement aura pour effet secondaire de tracer une ligne involontaire à chaque cercle. Pour l'éviter, ajouter l'instruction :

```
1745 VDU 29,0,0:MOVE P1%,P2%n
```

Cela peut être fait en assembleur, en ajoutant les instructions :

```
575 LDA#25:JSR OSWRCH:LDA#4:JSR
OSWRCH:LDA#0:JSR OSWRCH:LDA#0:JSR
OSWRCH:LDA#0:JSR OSWRCH:LDA#0:JSR
OSWRCH
```

Des améliorations plus compliquées pourraient être apportées pour tracer des arcs et des ellipses.



# L'étoile du Texas

**Texas Instruments a joué un grand rôle dans l'explosion de la micro-informatique. C'est l'un de ses ingénieurs qui inventa, en 1958, le circuit intégré, indispensable aux micros.**



**Le président de TI**  
Depuis le mois d'avril 1976, J. Fred Bucy est président et directeur exécutif de Texas Instruments.

Apparu en 1978, le TI99/4A était construit autour du microprocesseur 16 bits TMS9900, fabriqué par la firme elle-même. Il disposait de 16 K de RAM, de seize couleurs et d'un synthétiseur trois canaux. Cependant, malgré des ventes importantes, il fut victime de la féroce guerre des prix lancée par Commodore, qui ravagea le marché américain en 1982 et 1983. En dépit de très importantes baisses de tarif, la production du TI99 fut arrêtée fin 1983.

Les responsables expliquent : « Il se vendait très bien, mais sans rapporter de vrais bénéfices, et nous avons perdu plusieurs centaines de millions de dollars. » Si on leur demande s'ils ont l'intention de sortir un nouveau modèle, ils répondent aussitôt : « Absolument pas. Nous l'avons clairement expliqué à l'époque. Il est plus intéressant pour nous de vendre des appareils de prix plus élevé. » La firme entend pourtant se maintenir sur le marché de l'informatique professionnelle.

Elle propose, il est vrai, une gamme de produits très variés, qui va des jouets électroniques, comme « La dictée magique », aux mini-ordinateurs et aux détecteurs sismographiques sophistiqués. Ses quinze usines de production à travers le monde lui permettent un chiffre d'affaires annuel de 4 milliards de dollars.

Texas Instruments fut fondé par deux scientifiques : John Karchner et Eugen McDermott. On était en 1930 et la compagnie s'appelait alors Geophysical Service Incorporated. Karchner

avait eu l'idée d'envoyer des ondes sonores à travers les couches géologiques, afin de déterminer leur épaisseur et leur profondeur. La société fut fondée à Dallas et elle entreprit de convaincre l'industrie pétrolière locale.

GSI connut au cours des années trente une croissance rapide. Elle mit au point de nouvelles techniques de détection sismographique. Pendant la Seconde Guerre mondiale, plusieurs d'entre elles se révélèrent utiles dans la détection des sous-marins. Un laboratoire et un département de fabrication furent créés à cette occasion, et ils prirent une telle ampleur qu'en 1951 il fut décidé de leur accorder leur autonomie. La nouvelle compagnie reçut le nom de Texas Instruments.

Dès l'année suivante, TI obtint un brevet des Laboratoires Bell, qui lui permettait de fabriquer des transistors, alors de création toute récente; en 1954, la firme mit sur le marché le premier poste de radio à transistors. En 1958, un ingénieur de TI, Jack Kirby, inventa le circuit intégré — domaine où, depuis lors, la compagnie a toujours gardé la prééminence. Kirby joua aussi un rôle dans la mise au point en 1967 de la première calculatrice électronique de poche.

En 1983, TI a lancé sur le marché un ordinateur professionnel, construit autour du 8088, et qui peut donc accueillir le CP/M-86 et le MS-DOS. La même année, un ordinateur de gestion portable et un micro-ordinateur de poche, le CC40, firent leur apparition.

Le point fort de la firme reste cependant la fabrication des composants électroniques : elle en propose une grande variété (comprenant par exemple des puces RAM de 64 K). Presque tous les micro-ordinateurs domestiques sont équipés de puces TI, du Commodore 64 au Sinclair Spectrum. Tout comme les autres constructeurs, TI met en vente plusieurs microprocesseurs 16 bits, dont le fameux TMS9900 qui équipait le TI99/4A. Il avait la particularité d'être dépourvu de registres internes, regroupés au sein d'une mémoire « bloc-notes » extérieure à l'unité centrale. Les responsables déclarent : « Il était un peu en avance sur son temps. Il avait beaucoup de possibilités puissantes, mais peut-être la conception d'ensemble n'était-elle pas tout à fait à la hauteur. » TI compte bien prendre sa revanche avec un nouveau 16 bits, le TMS99000, qui, espère-t-elle, aura plus de succès.

Il n'empêche que l'échec de TI sur le marché grand public de l'informatique est à méditer.

## L'usine

La plus grande part des appareils de Texas Instruments est fabriquée dans cette usine installée à Dallas, Texas.



# PROGRAMME N° 28

## COMPTE CLIENT

Après le petit programme d'histogramme que nous avons vu précédemment, nous allons passer à une application encore plus pragmatique de la micro-informatique : la mise à jour des comptes clients. Le sujet est très sérieux, mais ce programme est aussi simple que le précédent. En effet, l'écriture du « compte client » est absolument standard et ne varie pas d'un micro à l'autre. Quant aux intérêts pratiques d'un tel programme, ils sont considérables, ne serait-ce que dans la gestion des commerces de détail, ou encore dans toutes les P.M.E. et P.M.I.

Plus une minute à perdre. Commençons...

```
10 REM Mise à jour de comptes clients
20 ? "NOM DU CLIENT ... "
30 INPUT NM$
```

NM\$ : nom du client. On tapera FIN au clavier si on désire arrêter le programme!

```
40 IF NM$ = "FIN" THEN GOTO 300
50 ? "SOLDE ANTERIEUR..."
60 INPUT SA
```

SA : valeur du solde antérieur.

```
70 N = SA
```

N : valeur du nouveau solde.

N est égal à SA avant tout mouvement de compte du client NM\$.

```
80 ? "CODE MOUVEMENT ... "
90 INPUT CM$
```

CM\$ : code de mouvement (C pour « crédit », D pour « débit », F pour arrêter la saisie des mouvements du client en cours de traitement et permettre l'affichage des caractéristiques multiples du compte de ce client).

```
95 IF CM$ = "D" THEN GOTO 200
```

```
100 IF CM$ = "F" THEN GOTO 250
```

```
120 IF CM$ = "C" THEN GOTO 160
```

Si le code client CM\$ est C (crédit) GOTO 160

```
130 ? "X X ERREUR SUR LE CODE MOUVEMENTS X"
```

Le code CM\$ a une valeur différente de F, C ou D. On retourne à la lecture de CM\$ en ligne 80.

```
140 GOTO 80
150 ? "SOMME A VOTRE CREDIT"
170 INPUT SC
```

SC : valeur du crédit.

```
180 N = N + SC
```

N, le nouveau solde est actualisé.

```
190 GOTO 80
```

Retour à la lecture d'un nouveau code, code mouvement CM\$.

```
200 ? "SOMME A VOTRE DEBIT"
210 INPUT SD
```

SD : valeur de débit.

```
220 N = N - SD
230 GOTO 80
```

Retour à la lecture d'un nouveau code de mouvement CM\$.

```
250 ? "CLIENT:";NM$
```

Édition du nom du client de ses anciens et nouveaux soldes, 250 à 270.

```
260 ? "ANCIEN SOLDE:";SA
270 ? "NOUVEAU SOLDE:";N
280 ?
```

Saut de ligne.

```
290 GOTO 20
```

Retour au début et traitement d'un nouveau client.

```
300 END
```

A l'aide de ce petit programme, vous pourrez commencer l'ébauche de ce qui sera une comptabilité clients. Puis, en le sophistiquant encore, vous pourrez aboutir à l'élaboration d'une gestion de fichier clients.

# ORGANIGRAMME

