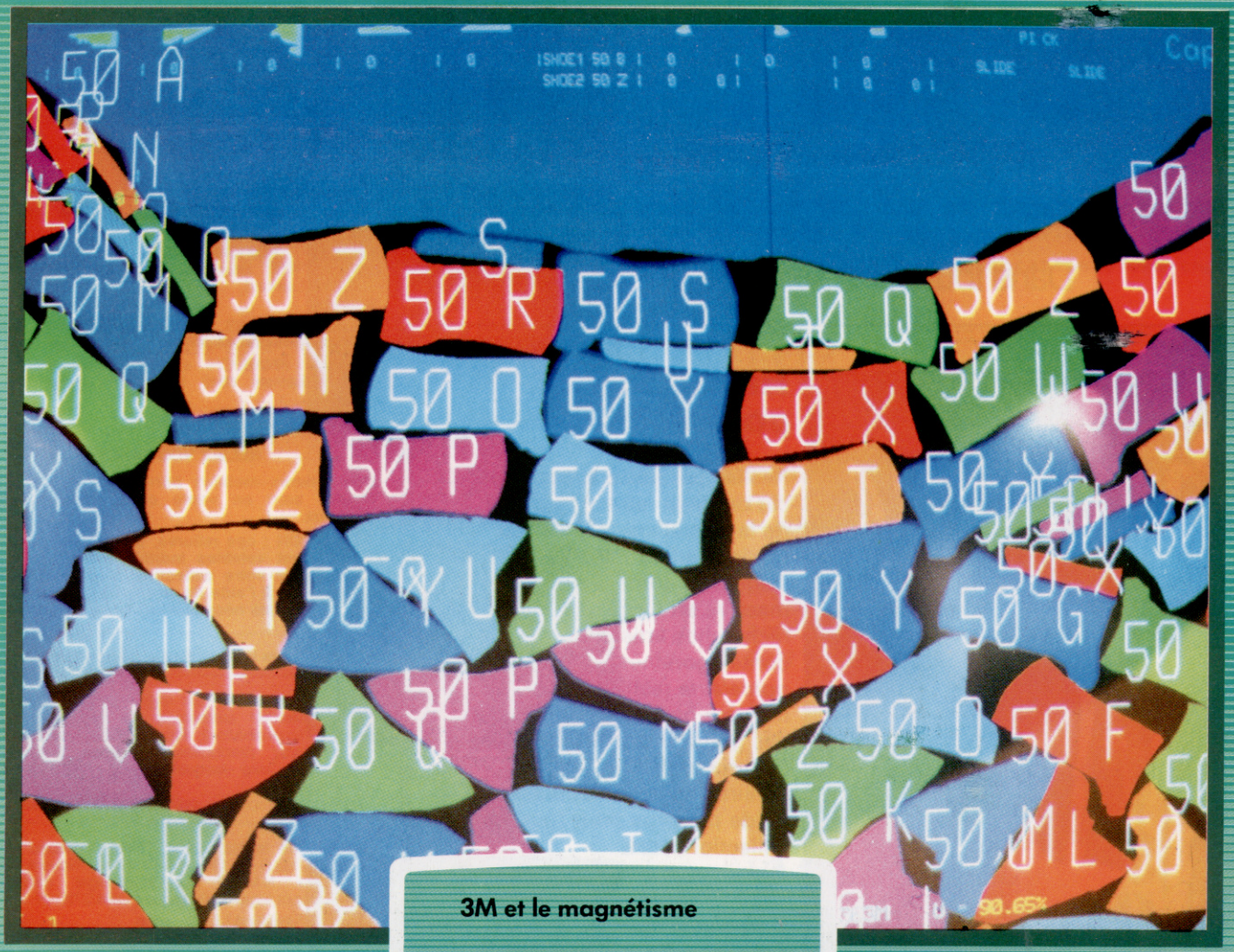
ABC INFORMATIQUE

Nº 47

COURS D'INFORMATIQUE PRATIQUE ET FAMILIALE



3M et le magnétisme

Programmes de bridge

Formes en trois dimensions

Structures modulaires

EDITIONS ATLAS

M 6062-47-12 F 85 FB - 3,80 FS - $ 1.95

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° 47.

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. Depôt légal : décembre 1984. 78412. Dépôt légal en Belgique : D/84/2783/27.

A NOS LECTEURS

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

Les Éditions Atlas

Magnétisme à la Une

La communication moderne ne se conçoit plus sans l'aide des supports magnétiques, bandes et disques. La société 3M s'est taillé une solide réputation mondiale dans ces domaines.

Le composant principal d'une bande magnétique est l'oxyde de fer (plus précisément : Fe_2O_3 dans la forme cristalline gamma) dispersé dans un liant avec lequel il forme la couche ou la « soupe » magnétique.

Par un procédé spécial, cette dernière est étalée sur un support plastique flexible. Le support utilisé aujourd'hui est le polyester, qui, même s'il est plus coûteux, présente une résistance et une stabilité exceptionnelles (voir encadré).

Les solvants, les liants, les plastifiants, les lubrifiants, les tensio-actifs, constituent l'ensemble des substances qui, associées à la dispersion des cristaux magnétiques en forme d'aiguilles, fournissent le « vernis » ou la « soupe » magnétique dont sera enduit le support en polyester. Le broyage de ces éléments avec l'oxyde de fer représente la première étape importante de ce délicat processus de fabrication. Un broyage incomplet ou peu soigné peut en effet compromettre la bonne qualité du produit fini. La couche magnétique doit être homogène, et doit permettre également le maintien des caractéristiques de ses différents composants.

Pendant cette phase de travail, le liant est soumis à des contrôles continus et très attentifs, car c'est lui qui déterminera la qualité et la fidélité d'enregistrement du produit fini.

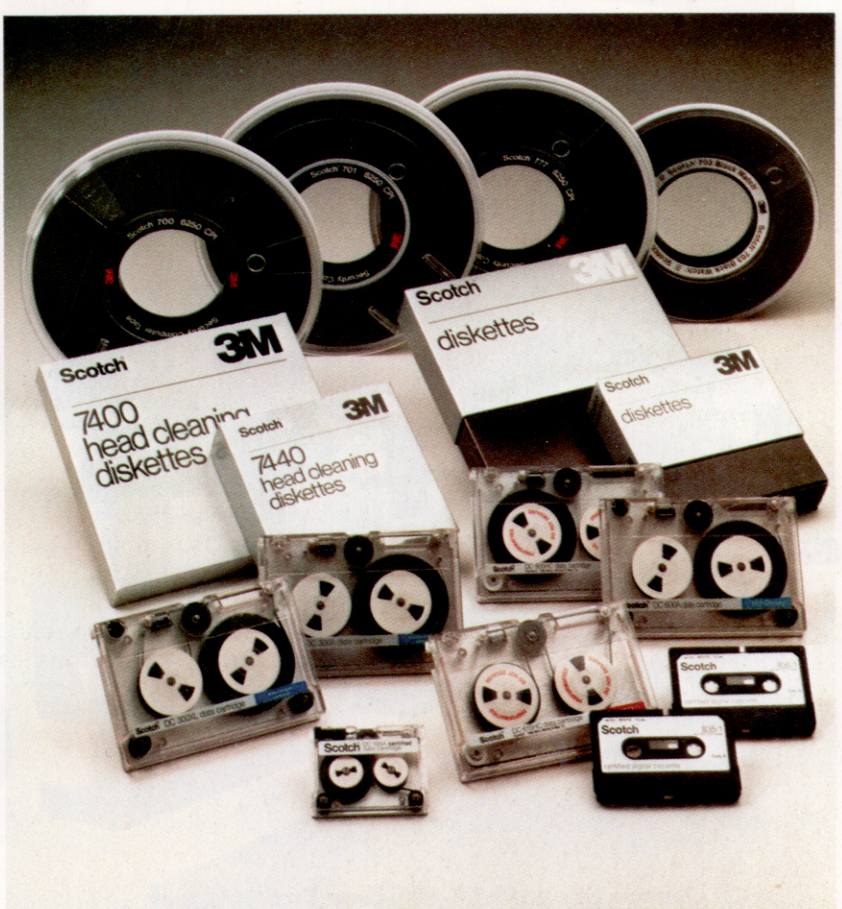
La phase de filtrage est destinée à vérifier l'absence dans le produit de corps étrangers ou de grumeaux du vernis qui se seraient éventuellement créés pendant le processus de broyage. D'autres contrôles seront également effectués afin de vérifier les propriétés de stabilité de la couche magnétique, dont la préparation est réalisée dans des locaux spéciaux, qui ont des caractéristiques de température et un degré hydrométrique constants.

Tout comme dans le cas des pellicules photographiques, le procédé d'enduction sur le ruban magnétique représente l'étape déterminante de tout le cycle de production.

A l'usine de San Marco Evangelista en Italie, où sont produits intégralement les supports magnétiques, les installations modernes, équipées de dispositifs de contrôle électronique et d'appareils de mesure de sensibilité, vérifient la production point par point. Pendant cette phase, les contrôles concernent tant l'uniformité du composé enduit sur son support que l'orientation des cristaux d'oxyde de fer dans la bonne direction longitudinale, afin d'amplifier de façon optimale ses caractéristiques magnétiques : haute sensibilité, stabilité aux hautes et basses fréquences, faible distorsion, faible effet de copie, etc. Ces termes, qui font désormais partie du langage des passionnés de hi-fi, désignent les caractéristiques qui opèrent dans le domaine du son (radios, maisons d'édition de disques, cinéma...).

Après cette opération, le produit est enroulé autour de grandes bobines, ou « jumbos », qui sont ensuite stockées, pendant une période prédéterminée, dans des locaux spéciaux, équipés d'une installation de climatisation; de là, elles sont envoyées aux ateliers de découpe, où des appareillages très modernes les façonnent aux formats désirés; cette opération est très délicate, car la surface de la bande ne doit pas être endommagée et son bord ne doit absolument pas s'effranger.

Compte tenu de l'épaisseur de la bande (de l'ordre de quelques millièmes de millimètre), il est évident que des appareillages de haute technologie s'imposent, même si le travail apparaît



Dès 1953, la société 3M créait la première bande magnétique pour ordinateurs. Un nom qui aujourd'hui ne fait pas penser qu'aux rubans adhésifs.
(Cl. 3M.)

L'expansion des supports magnétiques

Le marché des disquettes est en pleine évolution. Il est lié à l'expansion des ventes des mini-ordinateurs, micro-ordinateurs professionnels et domestiques et des machines de traitement de texte.

Le taux de croissance moyen pour les cinq années à venir est évalué à environ 30 %, avec des pointes plus élevées pour 1984 et 1985. Pour 1988, on prévoit la vente de 500 millions de disquettes. Les disquettes 8 pouces représentaient, en 1982, 60 % des ventes environ contre 40 % pour les disquettes 5 1/4 pouces. Ce rapport s'est déjà inversé en 1983, puisqu'il est passé à 44 % pour les 8 pouces et à 55 % pour les 5 1/4 pouces.

L'apparition depuis cette année des nouveaux formats de disquettes inférieurs à 5 pouces fera encore évoluer cette tendance. En 1984, les prévisions étaient de 29 % pour les 8 pouces, 68 % pour les 5 1/2 pouces et 3 % pour les disquettes inférieures à 5 1/4 pouces.

On prévoit que ces chiffres deviendront en 1988 : 6,5 % pour les 8 pouces; 74,5 % pour les 5 1/4 pouces; 19 % pour les 3 pouces.

On peut constater que la part des disquettes 5 1/4 pouces va rester majoritaire malgré l'apparition des disquettes 3 pouces, et ce probablement pour deux raisons :

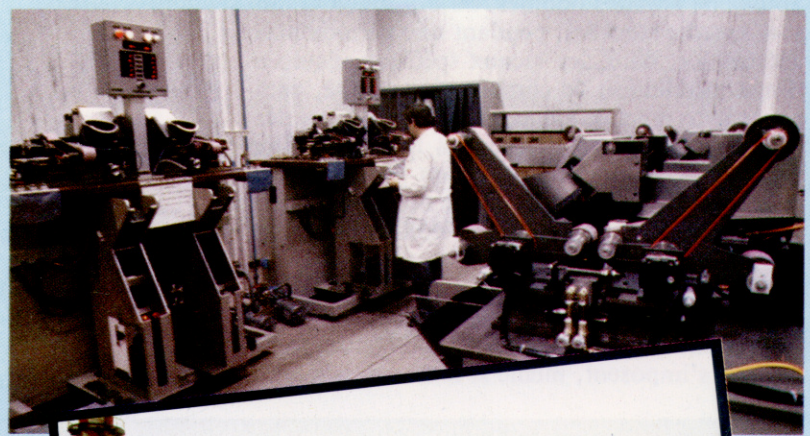
— le parc déjà installé d'ordinateurs;

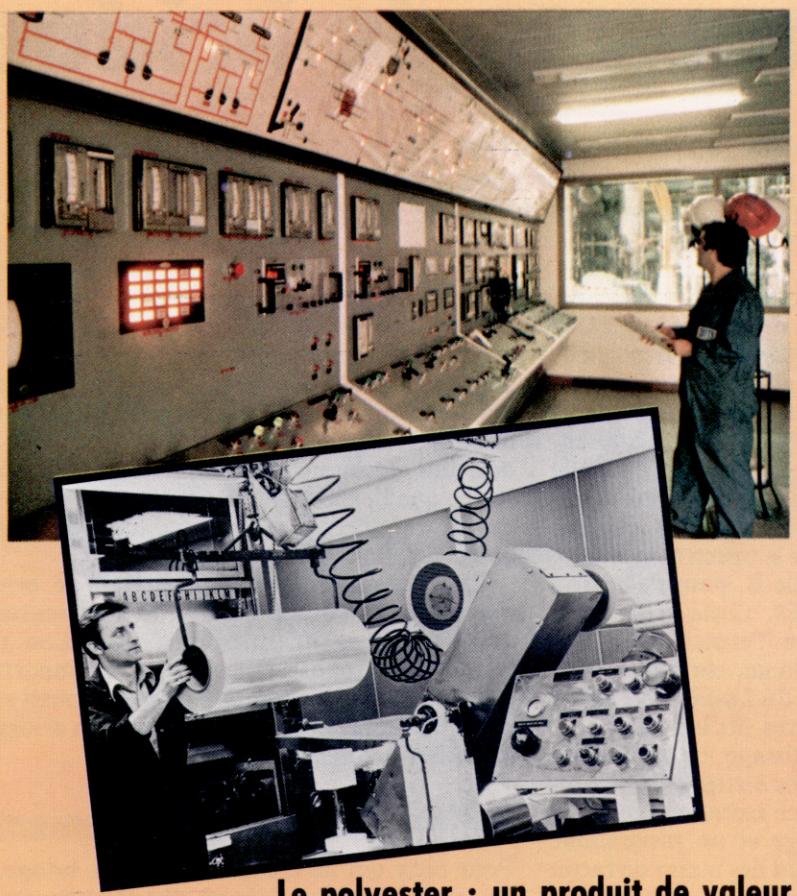
— l'évolution technologique des 5 1/4 pouces : disquette haute densité et haute coercitivité, chargeurs de cinq disques 5 1/4 pouces loméga, qui permettent une capacité de stockage très élevée.

De son côté, le marché des bandes croît à un taux nettement plus réduit, mais plus soutenu que ne le pensent de nombreux professionnels du secteur. En effet, sa croissance moyenne varie entre 5 et 10 % par an et atteindra, en 1986, une pointe de 10 millions de bandes.

La raison principale de cette croissance provient des nouvelles exigences de mémoires auxiliaires (« back up ») des utilisateurs de mini-ordinateurs et de la multiplication des unités à disque fixe, qui augmentent la nécessité de mémoires auxiliaires sur bandes dans le secteur des gros ordinateurs.

L'usine 3M de Caserta, en Italie, approvisionne l'ensemble du marché européen.





Le support polyester est indispensable dans de nombreux usages, allant des mémoires informatiques jusqu'aux emballages industriels. (Cl. 3M.)

Le polyester : un produit de valeur

Le polyester est un produit dont l'apparition est assez récente. En effet, c'est en 1955 qu'on retrouve les premières applications de ce composé polymère aux États-Unis et en Europe, et qu'on étudie les installations spécifiquement adaptées à sa fabrication. Plusieurs des grandes firmes les plus connues se sont engagées dès le début dans son développement, compte tenu des investissements importants nécessaires pour ce type de production. Les sociétés les plus importantes opérant dans le secteur des polyesters se trouvent implantées actuellement : cinq aux États-Unis (dont 3M), huit en Europe (dont l'une est 3M Italie) et cinq dans les pays asiatiques.

La production du polyester dans l'usine de San Marco Evangelista a débuté en 1972, et l'installation couvre actuellement 19 200 m² environ. Le travail est réparti comme suit : d'une part, le travail des résines, de l'autre, la fabrication et la découpe du film. Une installation spéciale s'occupe de la régénération des déchets, le polyester étant l'un des rares produits dont les déchets, traités de façon opportune, peuvent être recyclés dans le processus de fabrication, avec des caractéristiques techniques identiques à la matière première de base.

La production du film polyester commence par le travail de la résine, un processus chimique et physique très long et délicat, qui a comme composants de base le *diméthyltéréphtalate* et le *glycol éthylénique.* A travers les réactions d'estérification et de polymérisation à chaud, on obtient la résine polyester qui, refroidie, est broyée et réduite en granulés. Pendant ces phases de travail, on ajoute des parcelles d'additifs spéciaux, qui apportent les qualités requises au produit fini.

La résine est ensuite déposée dans un séchoir spécial, afin d'enlever aux granulés toute trace d'humidité. Puis, on effectue un premier réchauffage. Pendant la phase suivante, l'extrusion, la résine est fondue et immédiatement filtrée, afin d'éviter toute impureté. Arrivé à ce stade, le matériel obtenu est coulé sur un rouleau en inox (laminage) à l'aide d'un appareillage spécial. Grâce à une longue série de rouleaux, de pressions ainsi qu'à une combinaison savante de températures et de tensions diverses, le film prend progressivement la largeur et l'épaisseur souhaitées. Cette étape est appelée « phase d'orientation » et elle est très délicate, car le film de polyester (dont l'épaisseur n'est que de quelques millièmes de millimètre) est entraîné dans un parcours en continu, long de 130 m, qui n'est jamais interrompu. A la fin, le film ainsi travaillé est découpé et enroulé autour de bobines dont la largeur dépend de l'usage auquel le produit est destiné.

Le polyester est utilisé comme support pour les pellicules radiographiques, pour les bandes magnétiques, les pellicules utilisées dans le domaine des arts graphiques, les rubans adhésifs, les pellicules utilisées dans le cadre de la production du matériel électrique, ainsi que pour les emballages industriels.

Page ci-contre : avant d'être découpées à des dimensions convenant aux utilisateurs, les bandes magnétiques sont emmagasinées sous forme de rouleaux dits « jumbo » (Cl. 3M.)

simple en lui-même. La découpe des bobines s'effectue en bandes de 1/8 de pouce pour les cassettes, de 1/4 de pouce, de 1/2 pouce, etc., pour les bandes audio-vidéo, pour l'informatique et l'instrumentation.

La phase de conditionnement complète l'itinéraire de production : cassettes, bobines, pièces en matière plastique, emballages particuliers achèvent le travail en fonction de l'utilisation finale de la bande.

Les pièces en matière plastique sont également produites par l'usine, grâce à une installation moderne qui a été mise en œuvre récemment.

Même si des contrôles rigoureux et constants sont effectués tout le long de la production et à la fin des différents cycles de fabrication, le matériel est encore soumis à un dernier contrôle, très sévère, avant d'être commercialisé. Si le produit magnétique, ou même une partie de ce dernier, ne correspond pas aux caractéristiques de qualité requises par des normes très spécifiques, il est automatiquement écarté.

C'est la dernière étape d'un parcours de fabrication très intense qui se termine dans les entrepôts. De là, les produits partent pour un nouveau voyage en direction des utilisateurs.

Programmes de bridge

Le bridge, tout comme les échecs, est un jeu passionnant, quel que soit le niveau atteint. Il convient parfaitement à une adaptation sur ordinateur, car il est ainsi possible de jouer sans partenaire.

Les programmes de bridge offerts sur le marché sont répartis en deux catégories distinctes; certains servent à apprendre le jeu, d'autres ne sont que des programmes de jeu proprement dits.

Il existe de nombreux programmes d'apprentissage. Parmi les meilleurs, en termes de conception et de présentation, mentionnons la série Bridgemaster. Des versions de ce programme sont offertes pour le Spectrum, pour le ZX81, pour le BBC modèle B, pour l'Electron et pour le Commodore 64. Cette série a été conçue par le Britannique Terence Reese, un ancien champion international, chroniqueur de bridge de *The Observer* et du *London Standard* et auteur de nombreux livres sur le bridge.

Dans les jeux d'apprentissage, on soumet au débutant une série de mains écrites dans le programme. C'est la différence essentielle entre un programme d'apprentissage et un programme de jeu, ce dernier produisant des mains de façon purement aléatoire. Puisque le programme d'apprentissage « connaît » la composition de chaque main, il est en mesure de guider le joueur étape par étape, lui faisant ainsi lentement découvrir les règles et conventions qui font du bridge un jeu agréable et intelligent.

La série Bridgemaster illustre parfaitement les possibilités d'un jeu d'apprentissage. Le cours est constitué de deux programmes qui s'adressent au débutant et au joueur moyen. Le premier se nomme Complete Learning Package For The Beginner At Bridge, le second, Expert Bridge. L'ensemble proposé au débutant est composé de deux cassettes de programme, de deux bandes de commentaires, d'un fascicule d'instruction et d'un livre de poche donnant des notions élémentaires de bridge. Ce dernier ne doit pas nécessairement être utilisé en même temps que l'exécution du programme. Les bandes de commentaires donnent tous les renseignements requis par le débutant.

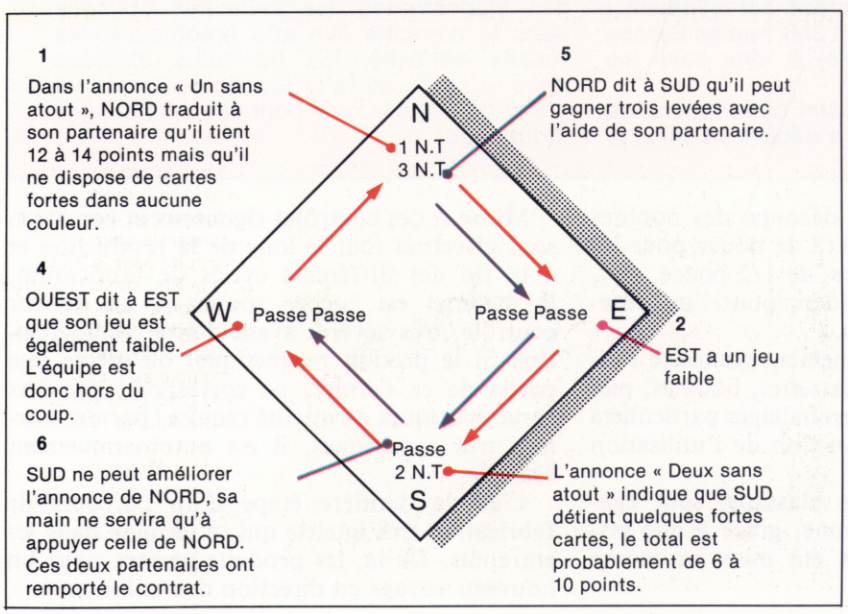
Les programmes d'apprentissage de bridge qui n'offrent pas de bandes de commentaires doivent inclure un manuel d'instruction très complet ou proposer des affichages comportant d'excellentes explications. Ils peuvent aussi supposer une connaissance élémentaire du bridge et se borner à améliorer votre jeu.

Initiation et perfectionnement

Pour ceux qui ne connaissent pas le bridge, le jeu de cartes est d'abord distribué à quatre joueurs qui forment deux équipes de deux. En bridge sur ordinateur, on adopte généralement la convention qui veut que les quatre joueurs soient nommés d'après leur position (nord, sud, est et ouest). Avant chaque jeu, chaque joueur doit annoncer la force de son jeu sans montrer ses cartes. La force d'une main est fonction du nombre de figures ou du nombre de cartes ayant la même couleur. Nous illustrons une annonce dans notre diagramme. L'annonce la plus forte détermine le contrat. On sélectionne ainsi l'atout et le nombre de levées que doit faire la paire gagnante. Après cette annonce, la main est jouée. Au cours du jeu, le joueur qui a remporté l'annonce devient le « déclarant » et son partenaire est le « mort ». Les cartes du joueur sont affichées afin d'être vues par tous. Le déclarant perdra ou gagnera des points, selon qu'il fera ou non le nombre de levées spécifié dans le contrat.

Tous les programmes d'apprentissage de bridge effectuent des annonces, qu'ils laissent ou non le joueur remporter. Avec Bridgemaster, le joueur peut faire sa propre annonce, après avoir vu les cartes affichées à l'écran. L'ordinateur fait ensuite l'annonce appropriée. Celle-ci peut être différente de celle du joueur. Le programme de Reese n'accepte que l'annonce ou le jeu que le concepteur du programme désire que vous choisissiez. Reese admet que cela puisse parfois être frustrant pour le débutant dont le choix est rejeté en faveur d'une tactique prédéterminée.

Annonce
Les valeurs des cartes pour l'annonce sont les suivantes : as-4; roi-3; reine-2; valet-1.
Les caractères rouges concernent la première annonce des joueurs, et les caractères bleus leur seconde annonce.



En plus d'afficher votre main et l'annonce, le programme doit gérer l'exécution du jeu. La méthode d'affichage la plus fréquemment utilisée est un carré au centre de l'écran dont les quatre côtés sont identifiés N., E., S., O. Les mains nord et sud sont alors entièrement affichées. Généralement, les cartes sont groupées par couleurs et affichées dans l'ordre suivant : pique, cœur, carreau et trèfle.

Le programme suppose que le sud est le joueur, et que le sud ou le nord sera le mort. Dans les deux cas, le sud joue toujours les cartes pour les deux joueurs. En d'autres mots, le sud joue toujours la main, que l'annonce ait été remportée par le sud ou par le nord. Cela diffère bien sûr du véritable jeu de bridge dans lequel vous pouvez jouer une soirée entière sans jamais définir le contrat.

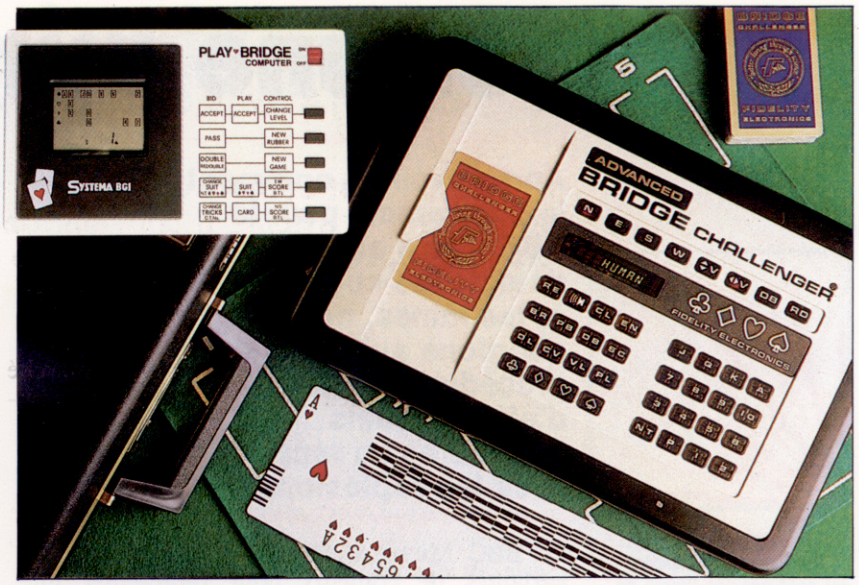
Dans un programme d'apprentissage, cette différence n'a aucune importance, sauf dans le domaine important du jeu défensif. Cette expérience est nécessaire pour empêcher vos adversaires de remplir le contrat lorsqu'ils ont remporté l'annonce. Dans un programme de jeu, les choses sont plus sérieuses, et nous ne connaissons pas de programme de jeu où est et ouest gagnent l'annonce. Si vous décidez de vous tenir à l'écart en passant toujours, en d'autres mots si vous laissez délibérément l'est ou l'ouest être le déclarant, l'excellent programme de jeu de CP Software, destiné au Spectrum 48 K, affichera un message vous disant qu'il n'a pas été conçu pour jouer dans de telles circonstances et il vous demandera d'appuyer sur la touche R pour générer une nouvelle main.

Lorsque l'annonce est terminée, le jeu de la main est affiché. Selon le vainqueur de l'annonce, l'ordinateur (jouant pour E. et O.) ou le joueur (pour S. ou N.) sélectionne une carte. Les cartes E. et O. apparaissent à l'écran une à la fois, comme sur une table pendant un jeu normal.

Les programmes sur ordinateur offrent de nombreuses fonctions utiles. Bridgemaster, par exemple, permet au joueur d'effectuer un choix parmi quatre options avant chaque partie, afin de déterminer le niveau des difficultés. Il permet de jouer toutes cartes affichées, le joueur humain jouant pour N. et pour S. (sous Reese); avec A pour « autoplay », vous surveillez le jeu exécuté automatiquement; H affiche les quatre mains à l'écran, tandis que D appelle une nouvelle donne. Cet ensemble d'options est généralement offert dans la plupart des programmes de bridge.

La fonction « autoplay » est particulièrement utile avec les commentaires des programmes Reeve, puisque le joueur n'a plus à être distrait par la sélection de cartes. Les programmes de jeu offrent eux aussi ces fonctions mais, ici, elles ne génèrent plus d'exercices répétitifs mais des jeux purement aléatoires.

Le programme Expert Bridge, dans la série Bridgemaster, constitue un cours avancé et très bien présenté qui examine certains aspects insolites du bridge.



Le programme de jeu créé par Alligata Software intéressera le joueur plus expérimenté. Des versions sont offertes pour l'Oric, pour le BBC modèle B, pour l'Electron et pour le Commodore 64. Une bonne connaissance du bridge est nécessaire pour utiliser ce programme, et aucun manuel n'est fourni. Les joueurs n'ayant pas de mémoire devront se préparer à prendre des notes.

Le programme de CP Software est un programme très agréable. Il ne sera jamais déclarant, il n'est donc pas possible d'exercer votre jeu défensif. Cependant, puisqu'une vaste majorité de joueurs de bridge préfère être déclarant, cela ne représente pas un grand inconvénient.

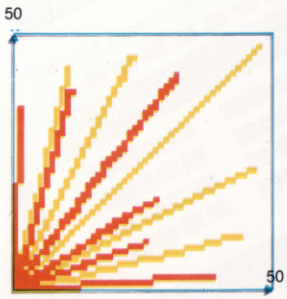
Les programmes de bridge, de jeu comme d'apprentissage, permettent d'apprendre et d'améliorer vos points faibles facilement. Le bridge nécessite de nombreuses qualités : posséder une bonne mémoire et un bon jugement, pouvoir travailler efficacement avec un partenaire, savoir jouer en prenant des risques tout en étant prudent, ne rien laisser transparaître et toujours garder son sang-froid. Les programmes dont nous venons de parler ne simuleront pas parfaitement une partie de bridge impliquant quatre joueurs, mais ils constituent un exercice valable et stimulant.

Machines dédiées
Les deux machines ci-dessus sont des ordinateurs dédiés au jeu de bridge. Le premier est un appareil de table nommé Advanced Bridge Challenger et a été créé par Fidelity Electronics. Bridge Challenger est recommandé aux joueurs de bridge expérimentés, puisqu'il nécessite beaucoup de maîtrise et de concentration. Un inconvénient majeur de Bridge Challenger est son minuscule écran de 8 caractères. La seconde machine à gauche est l'ordinateur Systema Play-Bridge. Il fonctionne à l'aide de piles. Il suffit de quelques minutes pour apprendre à utiliser Play-Bridge; il ne comporte pas de fonctions évoluées, mais il permet de jouer très agréablement. (Cl. Ian McKinnell.)

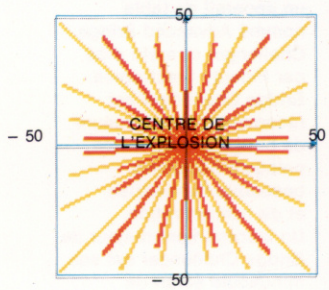
Choix d'un programme
De gauche à droite, les jeux Bridgemaster, disponibles sur cassette pour le Spectrum, pour le ZX81, pour le BBC modèle B, pour l'Electron et pour le Commodore 64; Bridge par Alligata Software disponible sur cassette pour l'Oric, pour le BBC modèle B, pour l'Electron et pour le Commodore 64; et Bridge Player de CP Software disponible sur cassette pour le Spectrum 48 K. (Cl. Ian McKinnell.)

Sur une mine

Après avoir vu comment le programme pour le BBC Micro et l'Electron détecte les collisions et les mines, nous abordons aujourd'hui les procédures de création d'effets visuels et sonores en rapport avec les explosions.



Un jeu explosif
Le graphisme de l'explosion est créé par des lignes de longueur aléatoire entre 50 et – 50 unités ; ces lignes divergent du centre de la déflagration. L'explosion est dessinée en quadrants.



Le BBC Micro et l'Electron autorisent seize variantes de couleurs pour représenter une explosion. Il n'y a en réalité que huit couleurs distinctes, les huit autres variantes étant dues à des effets de clignotement entre deux des huit couleurs de base. Le clignotement se fait toutes les demi-secondes, mais la cadence peut être modifiée par deux commandes FX.

*FX9 donne le temps d'affichage pour la première couleur du couple des couleurs en clignotement. Le temps est mesuré en cinquièmes de seconde. Ainsi, *FX9,20 modifie le temps d'affichage de la première couleur pour le mettre à 2/5 de seconde. L'autre commande FX, *FX10, est utilisée de la même manière, mais pour la deuxième couleur du couple.

Pour rehausser l'effet visuel de l'explosion, la couleur des mines ainsi que celle des lettres qui apparaissent pour afficher le score peuvent être mises en clignotement rapide. Comme ces dernières étaient originellement affichées (PRINT) dans la couleur logique 2 (que nous avons attribuée au vert), nous pouvons utiliser la commande suivante pour assigner la couleur logique 2 à une autre couleur prise de manière aléatoire parmi seize combinaisons.

```
VDU19,2,RND(15),0,0,0
```

RND(15) choisit un nombre entier compris entre 1 et 15. Cela signifie que la couleur 0 n'est jamais retenue.

Les couleurs utilisées pour tracer les lignes de déflagration peuvent également être prises au hasard en utilisant :

```
GCOL 0,RND(3)
```

Nous mettons ensuite en œuvre une boucle, pour tracer les segments selon un enchaînement de couleurs aléatoires. Nous avons jusqu'à présent envisagé les commandes haute résolution MOVE et DRAW. DRAW(X,Y) trace une ligne jusqu'au point de coordonnées *x, y*. Il existe cependant une autre famille de commandes haute résolution de dessin qui permettent, entre autres choses, de spécifier des points relativement les uns aux autres. La commande PLOT K,X,Y peut être utilisée pour tracer des lignes entre des points de coordonnées relatives ou absolues selon la valeur de K.

Variantes possibles avec PLOT

K = 0	Déplacement relatif en fonction du dernier point.
K = 1	Tirer une ligne jusqu'à la position relative en couleur d'affichage.
K = 2	Tirer une ligne jusqu'à la position relative dans la couleur logique inverse.
K = 3	Tirer une ligne jusqu'à la position relative dans la couleur du fond.
K = 4	Aller sur la position absolue.
K = 5	Tirer une ligne jusqu'à la position absolue dans la couleur d'affichage.
K = 6	Tirer une ligne jusqu'à la position absolue dans la couleur logique inverse.
K = 7	Tirer une ligne jusqu'à la position absolue dans la couleur du fond.

PLOT4 et PLOT5 sont entièrement équivalents aux commandes MOVE et DRAW. Pour notre effet d'explosion, nous utiliserons PLOT1 pour tracer des lignes relatives au centre de la déflagration.

Dans un article précédent, lorsque nous avons appelé PROC une version fictive de la procédure explosion, nous avons attribué les valeurs de xgraph et ygraph à x-explosion et y-explosion. Ces valeurs ne jouaient aucun rôle dans la procédure virtuelle, mais elles seront utilisées maintenant pour spécifier le centre de l'explosion. Si cette procédure est appelée par la ligne 3390 de la procédure PROCdéplacement, xgraph et ygraph seront les coordonnées graphiques du centre de la position où se trouve la mine. Ainsi, MOVEx-explosion,y-explosion déplacera le curseur graphique sur le centre de la position que nous voulons affecter à l'explosion.

Si nous voulons passer les coordonnées sur d'autres variables pour la procédure explosion, au lieu d'utiliser simplement xgraph et ygraph, c'est parce que cette procédure sera également appelée depuis une autre partie du programme dans laquelle les coordonnées du centre de l'explosion seront spécifiées par d'autres variables. Une fois que nous nous sommes positionnés sur le centre de l'explosion, nous pouvons tracer une ligne dans une direction aléatoire, d'une longueur maximale de 50 unités par exemple :

```
PLOT 1,RND(50),RND(50)
```

Cette instruction ne conviendra pas réellement puisqu'elle ne spécifie pas de coordonnées négatives (une coordonnée négative *x* suscite une ligne vers la gauche, alors qu'une coordonnée négative *y* permet de tracer une droite vers le bas). Aussi l'utilisation répétée de cette commande ne remplira qu'un quart de l'espace avoisinant la déflagration.

Il nous faut donc introduire des valeurs négatives (et donc tracer des lignes dans toutes les directions depuis le centre de l'impact). Pour cela, nous remplacerons RND(50) par RND(100)-50. La valeur maximale que peut prendre RND(100) est

bien sûr 100, ce qui fait que la valeur maximale pour RND(100)-50 est 50.

La valeur minimale de RND(100) est par ailleurs 1, ce qui signifie que RND peut prendre d'aussi petites valeurs que 1-50 = −49.

Voyons maintenant comment produire un effet sonore d'accompagnement de la déflagration à l'écran. Nous pouvons créer des sons intéressants et complexes avec les commandes BASIC du BBC, SOUND et ENVELOPE.

SOUND peut soit générer des notes de hauteur déterminée, soit produire des bruits pour des effets sonores. La commande ENVELOPE est utilisée en conjonction avec SOUND pour mettre en forme le son en simulant les instruments de musique les plus divers. SOUND comporte quatre paramètres associés qui commandent les caractéristiques de la tonalité produite :

SOUND C,A,P,D

— C est le numéro du canal, compris entre 0 et 3. Les canaux 1 et 3 produisent des sons à hauteur déterminée, tandis que le canal 0 est destiné aux effets sonores spéciaux. C'est le canal que nous utiliserons ici.

— A représente l'amplitude du son. Le volume maximal est pour A= −15. Les valeurs positives de A déterminent la mise en forme du son et sont utilisées conjointement avec la commande ENVELOPE.

— P est la hauteur *(pitch)*.

— D est la durée de la note. D varie de 0 à 254 et permet de tenir des notes dont la durée est mesurée en vingtièmes de seconde. Quand D a la valeur −1, le son sera ininterrompu jusqu'à ce que vous interveniez. L'effet sonore pour l'explosion doit être d'intensité sonore élevée pour s'approcher le plus possible de ce que l'on imagine être le bruit d'une explosion.

Notre commande sera donc d'une amplitude maximale, avec une valeur pour A de −15.

La valeur pour la hauteur du son est plus complexe. Utilisé avec le canal 0, le paramètre P prend des valeurs comprises entre 0 et 7 selon la table suivante :

P = 0	Impulsion de haute fréquence.
P = 1	Impulsion de fréquence moyenne.
P = 2	Impulsion de basse fréquence.
P = 3	Impulsion de fréquence déterminée par le canal 1.
P = 4	Son de haute fréquence.
P = 5	Son de moyenne fréquence.
P = 6	Son de basse fréquence.
P = 7	Fréquence déterminée par le canal 1.

La commande SOUND utilisée dans le programme est la suivante :

SOUND 0,−15,4,40

Cette commande produit un sifflement tout à fait surprenant — strident, de grande hauteur (extrêmement aigu); ce bruit a une durée de deux secondes (40/20) avec un volume maximal.

Nous vous présentons ici le listage complet de la procédure :

```
3550 DEF PROCexplosion(x-explosion,y-explosion)
3560 REM ** EFFET SONORE **
3570 SOUND 0,-15,6,50
3580 REM ** DETERMINER CADENCE CLIGNOTEMENT **
3590 * FX9,20
3600 * FX10,50
3610 FOR I=1 TO 100
3620 MOVE x-explosion,y-explosion
3630 VDU19,2,RND(15),0,0,0
3640 GCOL 0,RND(3)
3650 PLOT 1,RND(100)-50,RND(100)-50
3660 NEXT I
3670 PROCré-initialisation
3680 ENDPROC
```

La procédure « ré-initialisation »

Il y a plusieurs choses à faire après une explosion :

— réduire le nombre des vies qui restent et vérifier si toutes les vies ont été utilisées;

— vider l'écran des séquelles de l'explosion;

— repositionner le détecteur et l'assistant à leur position de départ.

Après avoir testé la fin du jeu, il nous faut nettoyer l'écran.

Plusieurs méthodes pourraient être utilisées, par exemple afficher PRINT des blancs sur la zone de l'explosion. Mais la méthode que nous utilisons ici repositionne les mines et sauvegarde l'information. Pour être tout à fait juste, il faudrait repositionner autant de mines qu'il y en avait avant l'explosion. Cela peut être facilement calculé à partir du score. Puisque chaque mine vaut 150 points au départ, il est facile de calculer le nombre de mines qui restent et de passer ce paramètre à la procédure placer des mines.

Le détecteur de mines et l'assistant sont repositionnés en ré-initialisant leurs coordonnées, et en appelant ensuite la procédure de positionnement de figures graphiques.

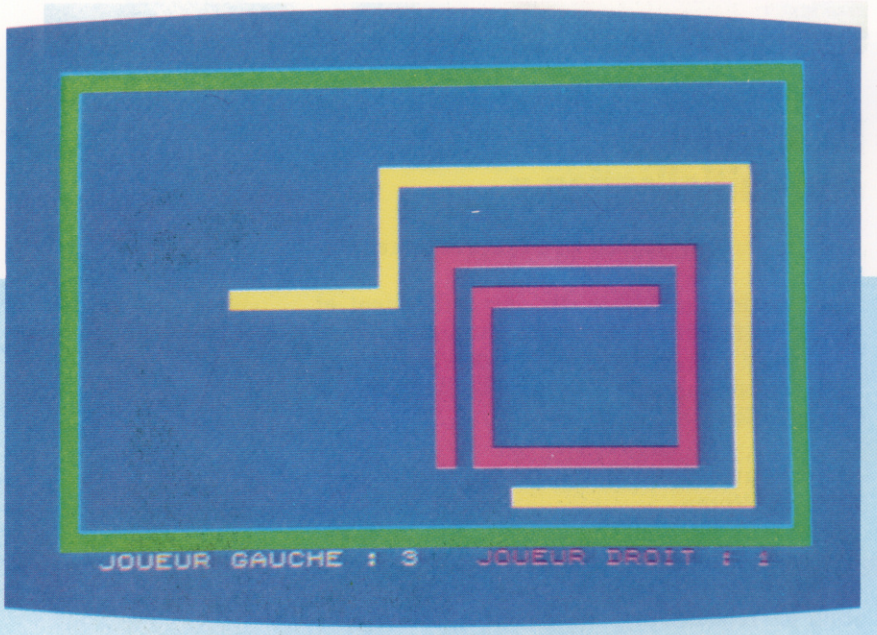
Le listage complet de la procédure RÉ-INITIALISATION est donné ci-dessous. Ajoutez-le à votre programme avec la procédure explosion.

```
3880 DEF PROCré-initialisation
3890 total=total+1
3900 IF total>4 THEN drapeau de fin=1:PROCFIN
3910 CLS
3920 VDU19,2,2,0,0,0
3930 COULEUR 2
3940 PROCinitialisation-variables
3950 mines_gauche =50-score/150
3960 PROCplacer-des-mines(mines-à-gauche)
3970 PROCtracer-la-bordure
3980 PRINTTAB(2,27);"Temps"
3990 PRINTTAB(2,28)"Score"
4000 PRINTTAB(11,28) score $
4010 PRINTTAB(2,29)"Hi score"
4020 PRINTTAB(11,29) hi_score $
4030 hommes-restants$=LEFT$(homme$-total)
4040 COULEUR 1
4050 PRINTTAB(2,30);hommes-restants$;" "
4060 COULEUR 2
4070 PROCpositionnement
4080 ENDPROC
```

Kevin Jones

Trace

Voici un jeu d'action que l'on retrouve sur de nombreux ordinateurs familiaux. Pierre Monsaut a écrit cette version pour le MO5 de Thomson. Il est également valable pour le TO 7.



Deux joueurs s'affrontent pour se partager l'espace vital. Chacun doit s'efforcer, tout en se déplaçant, de ne jamais recouper sa trace ou celle de son adversaire, et de ne pas sortir du rectangle dessiné sur l'écran. Utilisez les manches à balai ou les touches suivantes :

— Joueur de droite : P, L, M et @.
— Joueur de gauche : Z, Q, S et W.

```
10 REM *********
20 REM * TRACE *
30 REM *********
40 CLEAR ,,1
50 GOSUB 750
60 ON JK GOTO 130
70 D$=INKEY$
80 HB=(D$="L")-(D$="M")
90 VB=(D$="P")-(D$="@")
100 HA=(D$="Q")-(D$="S")
110 VA=(D$="Z")-(D$="W")
120 GOTO 170
130 HA=(STICK(0)=7)-(STICK(0)=3)
140 VA=(STICK(0)=1)-(STICK(0)=5)
150 HB=(STICK(1)=7)-(STICK(1)=3)
160 VB=(STICK(1)=1)-(STICK(1)=5)
170 IF HA<>0 THEN H1=HA:V1=0
180 IF VA<>0 THEN V1=VA:H1=0
190 IF HB<>0 THEN H2=HB:V2=0
200 IF VB<>0 THEN V2=VB:H2=0
210 X1=X1+H1
220 Y1=Y1+V1
230 IF SCREEN(X1,Y1)=127 THEN 350
240 LOCATE X1,Y1
250 COLOR 3
260 PRINT N$;
270 X2=X2+H2
280 Y2=Y2+V2
290 IF SCREEN(X2,Y2)=127 THEN 410
300 LOCATE X2,Y2
310 COLOR 5
320 PRINT N$;
330 BEEP
340 GOTO 60
350 F2=F2+1
360 GOSUB 690
370 IF F2=10 THEN 470
380 IF INKEY$<>"" THEN 380
390 GOSUB 870
400 GOTO 60
410 F1=F1+1
420 GOSUB 690
430 IF F1=10 THEN 540
440 IF INKEY$<>"" THEN 440
450 GOSUB 870
460 GOTO 60
470 CLS
480 COLOR 1,7
490 LOCATE 9,5
500 PRINT "LE JOUEUR DROIT GAGNE";
510 LOCATE 15,10
520 PRINT F2;"A";F1;
530 GOTO 600
540 CLS
550 COLOR 1,7
560 LOCATE 9,5
570 PRINT "LE JOUEUR GAUCHE GAGNE";
580 LOCATE 15,10
590 PRINT F1;"A";F2;
600 LOCATE 14,15
610 PRINT "UNE AUTRE ?";
620 IF INKEY$<>"" THEN 620
630 D$=INKEY$
640 IF D$="" THEN 630
650 IF D$<>"N" THEN RUN
660 CLS
670 SCREEN 4,6,6
680 END
690 FOR I=1 TO 5
700 BEEP
710 FOR J=1 TO 100
720 NEXT J
730 NEXT I
740 RETURN
750 CLS
760 SCREEN 2,4,4
770 DEFINT A-Z
780 ATTRB 1,1
790 LOCATE 6,10,0
800 PRINT "JOYSTICKS ?";
810 R$=INKEY$
820 IF R$="" THEN 810
830 JK=-(R$="O")
840 ATTRB 0,0
850 DEFGR$(0)=255,255,255,255,255,255,25
5,255
860 N$=GR$(0)
870 CLS
880 COLOR 2
890 FOR X1=0 TO 39
900 LOCATE X1,0:PRINT N$;
910 LOCATE X1,23:PRINT N$;
920 NEXT X1
930 FOR Y1=1 TO 22
940 LOCATE 0,Y1:PRINT N$;
950 LOCATE 39,Y1:PRINT N$;
960 NEXT Y1
970 LOCATE 2,24:COLOR 3
980 PRINT "JOUEUR GAUCHE :";F1;
990 LOCATE 22,24:COLOR 5
1000 PRINT "JOUEUR DROIT :";F2;
1010 X1=8:Y1=11:X2=32:Y2=11:H1=1:V1=0
1020 H2=-1:V2=0:S1=0:S2=0:RETURN
```

Des ailes pour l'Electron

La nouvelle interface Plus 1, de prix raisonnable, permet enfin à l'Electron de tenir le rôle pour lequel il a été conçu : des qualités du BBC Micro à un prix réduit.

Les concepteurs de l'Electron voulaient en faire une sorte de version modèle réduit du BBC Micro, dont ils conservèrent le système d'exploitation et le BASIC, tous deux excellents ; mais ils ne jugèrent pas utile de le pourvoir des nombreuses interfaces qui rendaient l'original si puissant. L'Electron ne dispose que d'un port cassette, deux prises moniteur (RVB ou vidéo composite), un modulateur (pour le raccord sur un poste de télévision) et une prise d'alimentation.

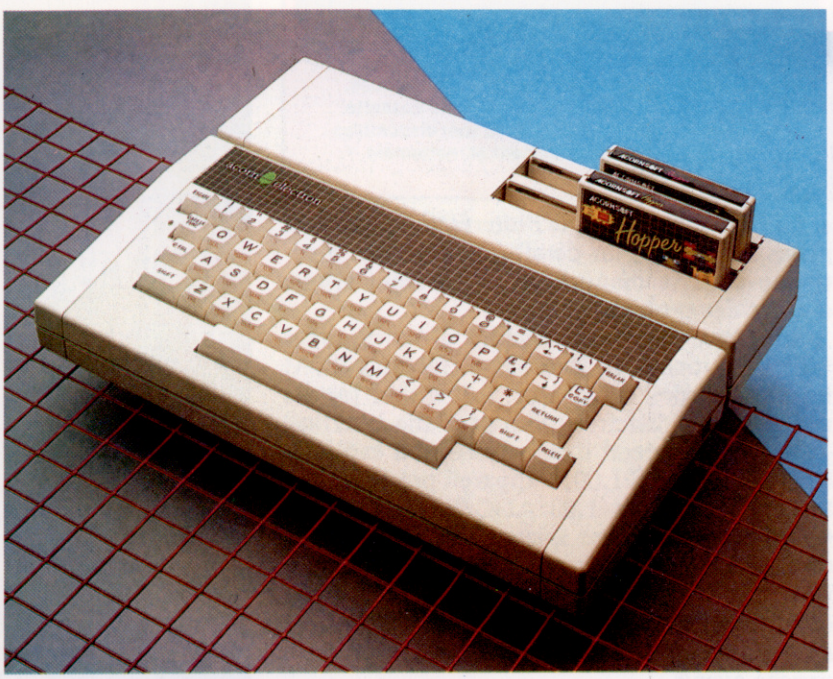
L'interface Plus 1 ajoute à cela une interface imprimante parallèle, une prise pour manche à balai, et deux autres pour cartouches ROM. Acorn espère que le potentiel de vente de l'appareil s'en trouvera accru. Cartouches et manche à balai permettent d'utiliser l'Electron comme appareil de jeu, tandis que l'interface imprimante assure la connexion de cet engin, et rend possible le traitement de texte.

La mise en place est très simple : il suffit de l'insérer sur le connecteur plat situé à l'arrière de l'Electron. Deux boulons sont alors poussés à travers le boîtier de l'interface Plus 1, et fixés sur les prises installées à l'intérieur de l'ordinateur. Alimentation et données circulent par l'intermédiaire du connecteur plat. Une puce ROM mise en place à l'intérieur du Plus 1 gère toutes les opérations où le dispositif intervient.

Le port imprimante est de type Centronics parallèle standard. Pour activer l'imprimante, l'usager recourt à la commande VDU2, et à VDU3 pour la désactiver. Ce sont les mêmes commandes que sur le BBC Micro, tout comme VDU1, qui permet d'envoyer un caractère à l'imprimante.

Un port analogique accueille les manches à balai. Il est constitué d'une prise à 15 broches, de type D ; là encore les connections sont les mêmes que sur le BBC Micro. Toutefois l'interface Plus 1 ne peut mesurer que quatre tensions, en provenance de quatre poignées ou de deux manches à balai (ceux-ci produisant en effet deux signaux : l'un pour les déplacements verticaux, l'autre pour les mouvements horizontaux).

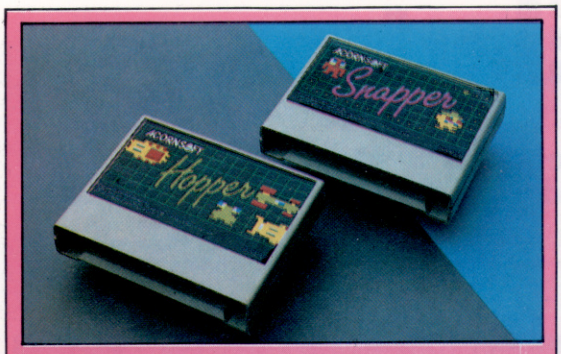
Ces tensions sont ensuite transformées en signaux numériques par un convertisseur analogique/numérique, la puce ADC0844. C'est un dispositif bien plus simple que celui dont le BBC Micro fait usage ; ce qui veut dire que l'Electron lit les mouvements du manche à balai avec beaucoup moins de précision. Pour la plupart des jeux, ce n'est pas un problème, mais avec les logiciels graphiques — qui permettent par exemple de dessiner librement sur l'écran — c'est plus ennuyeux : le déplacement devient assez saccadé, et le dessin est moins détaillé.



Espace vital
L'Electron a été conçu comme étant une version moins coûteuse du BBC Micro. Il lui manque les interfaces, mais il a gardé le basic et les excellentes possibilités graphiques de l'original, pour un prix moitié moindre. L'interface Plus 1, récemment mise en vente par Acorn, permet d'assurer les connections les plus importantes, pour un prix raisonnable. (Cl. Chris Stevens.)

Lorsque l'interface est mise en place, la cartouche ROM, installée sur la prise à l'avant, est automatiquement activée : pour l'interrompre, il faut appuyer sur ESCAPE. Le système de fichiers fonctionne à peu près (mais plus rapidement !) comme la version cassette, et recourt aux mêmes commandes *CAT, LOAD et CHAIN. Si, précisément, vous voulez charger un logiciel installé sur cassette, il faudra d'abord faire *TAPE, ou ôter la cartouche — en prenant bien soin de mettre d'abord l'ordinateur hors tension ! En dehors des jeux, certains langages de programmation seront bientôt disponibles sur cartouche ; il suffit de les connecter pour qu'ils prennent la place du BASIC implanté en ROM.

Le gros problème des périphériques est qu'ils ont parfois des effets secondaires gênants. Lorsque l'interface Plus 1 est en place, il peut se produire par exemple des erreurs de chargement, surtout lorsque le programme considéré comporte des fichiers de données : c'est ainsi que deux des programmes de la cassette de présentation de l'Electron refusent tout simplement d'être chargés en mémoire. Il est vrai qu'on peut toujours duper le système d'exploitation,



Munitions
Les cartouches destinées à l'Electron ont le gros avantage d'être chargées en une ou deux secondes, alors qu'il faut plusieurs minutes avec une cassette. Pour l'instant, seules six d'entre elles sont disponibles : quatre jeux, un programme éducatif et, nettement plus cher, le langage LISP.

et lui faire « croire » que l'interface n'est pas en place. Mais c'est un détail qui n'est pas mentionné dans le manuel, et le débutant risque d'avoir des problèmes.

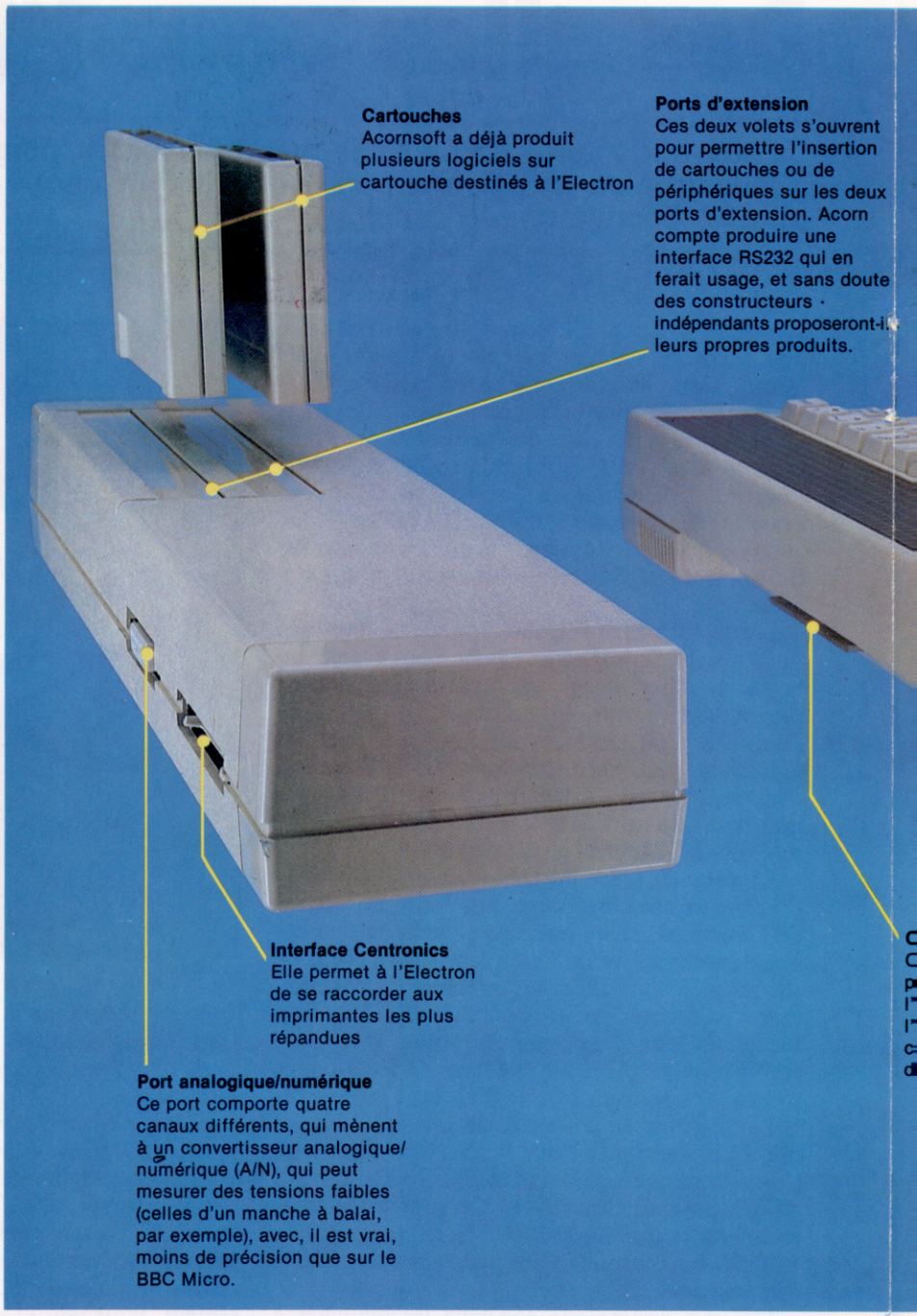
Sur n'importe quelle machine d'Acorn, taper *HELP vous donne la liste des différentes ROM implantées sur l'ordinateur. Lorsqu'on fait de même sur un Electron pourvu d'une interface Plus 1, on apprend que le système d'exploitation ROM est OS 1.00; l'appareil précise également : Expansion 1.00 ADC/Printer/RS243. C'est tout simplement le système d'exploitation de l'extension elle-même. RS243 fait référence à une interface série standard, dont l'Electron comme le Plus 1 sont actuellement dépourvus, mais Acorn songe à en installer une dans un proche avenir.

La principale différence entre le BASIC du BBC Micro et celui de l'Electron est que ce dernier est dépourvu du Mode 7 (mode graphique de type Vidéotext), qui, sur le BBC, sert essentiellement pour les génériques et les instructions, car il permet d'économiser de l'espace mémoire.

Sur l'Electron, les éléments correspondants sont illisibles. Mais la plupart des jeux ont recours à d'autres modes, et il devrait donc être facile de charger un programme destiné primitivement au BBC; il n'y aura plus de problèmes une fois passée l'introduction.

Par ailleurs, les possibilités sonores sont très différentes. L'Electron n'a qu'un canal sonore, contre quatre au BBC, et sa commande ENVELOPE ne règle que la hauteur du son, et non son volume. Pourtant, les instructions sont compatibles, et le même logiciel pourra tourner sur les deux machines, même si le son se révèle très différent sur l'Electron.

Celui-ci a un clavier plus agréable que le BBC, mais il comporte moins de touches, ce qui veut dire qu'elles ont parfois, comme sur le Spectrum, trois ou quatre fonctions différentes. La touche L donne une majuscule ou une minuscule, suivant que la commande SHIFT a été pressée ou non; mais employée en conjonction avec la touche FUNCTION, elle produit l'instruction BASIC LIST. CONTROL et L, enfoncées simultanément, permettent de vider l'écran. Il



Cartouches
Acornsoft a déjà produit plusieurs logiciels sur cartouche destinés à l'Electron

Ports d'extension
Ces deux volets s'ouvrent pour permettre l'insertion de cartouches ou de périphériques sur les deux ports d'extension. Acorn compte produire une interface RS232 qui en ferait usage, et sans doute des constructeurs indépendants proposeront-ils leurs propres produits.

Interface Centronics
Elle permet à l'Electron de se raccorder aux imprimantes les plus répandues

Port analogique/numérique
Ce port comporte quatre canaux différents, qui mènent à un convertisseur analogique/numérique (A/N), qui peut mesurer des tensions faibles (celles d'un manche à balai, par exemple), avec, il est vrai, moins de précision que sur le BBC Micro.

aurait sans doute été plus judicieux de conserver les dix touches de fonction de couleur rouge installées sur le BBC Micro. L'Electron en possède aussi, mais ce sont en fait les touches numériques, employées avec la touche FUNCTION.

L'autre grosse différence entre les deux machines tient à leur gamme d'interfaces. Le BBC Micro en a plus que n'importe quel micro-ordinateur domestique, ce qui veut dire que de très nombreux périphériques — imprimantes, modems, lecteurs de disquettes, second microprocesseurs, robots articulés — peuvent y être connectés. Il faut naturellement pour cela ajouter des puces et des prises, ce qui augmente sensiblement les coûts de production. D'autre part, s'en dispenser permet de proposer un ordinateur plus petit, plus modeste et, surtout, meilleur

A vous de jouer
Snapper est une très bonne version du légendaire Pacman. Contrôlé au clavier ou au manche à balai, il est disponible sur cartouche et, à un prix moins élevé, sur cassette.

Connecteur d'extension
Ce connecteur plat permet le raccord de l'interface Plus 1 sur l'Electron, dont il constitue le seul port d'extension.

marché. Ce raisonnement, fondé sur des arguments commerciaux évidents, a présidé à la création de l'Electron. L'apparition de l'interface Plus 1 laisse penser qu'Acorn était allé trop loin dans cette voie.

Même équipé de cette extension, l'Electron reste infiniment moins puissant que le BBC; quoi qu'il en soit, il offre des possibilités que bien des usagers n'ont pas souvent l'occasion de mettre en œuvre.

C'est pourquoi on peut dire, en conclusion, que l'ensemble Electron/interface Plus 1, pour un prix bien moins élevé, est d'un rapport qualité-prix très satisfaisant, d'autant plus qu'il dispose d'un BASIC structuré, de capacités graphiques et sonores intéressantes, et de logiciels adaptés de plus en plus nombreux.

BBC Micro Modèle B

PRIX
***

DIMENSIONS
75 × 340 × 410 mm.

UC
6502, 1,8 MHz.

MÉMOIRE
32 K RAM, 32 K ROM.

AFFICHAGE ÉCRAN
8 modes d'affichage. Résolution maximale : 80 × 32 (texte), 640 × 256 (graphisme). Jusqu'à huit couleurs (stables ou clignotantes). Mode d'affichage Vidéotext. Caractères redéfinissables.

INTERFACES
UHF (télévision), RVB et vidéo composite (moniteurs), port cassette, interfaces RS243 et Centronics, port analogique (manches à balai, etc.), prises ROM (pour logiciels), bus 1 MHz, interface disquette (en option), interface pour réseau Econet (en option), port utilisateur, source d'alimentation extérieure (pour lecteur de disquettes, etc.).

LANGAGES DISPONIBLES
BBC basic (intégré), Assembleur du 6502 (intégré), lisp, forth, BCPL, pascal.

CLAVIER
72 touches type machine à écrire, dont 10 touches de fonction programmables.

DOCUMENTATION
Le manuel rendra de grands services aux programmeurs expérimentés, mais n'aidera guère le débutant.

POINTS FORTS
Une des meilleures versions de basic, énormément d'interfaces, clavier, sons et graphismes très bons.

POINTS FAIBLES
Il est difficile au débutant d'en tirer réellement parti.

Acorn Electron Plus 1

PRIX
**

DIMENSIONS
65 × 260 × 340 mm (interface en place).

UC
6502, 1,8 MHz.

MÉMOIRE
32 K RAM, 32 K ROM.

AFFICHAGE ÉCRAN
7 modes d'affichage. Résolution maximale : 80 × 32 (texte), 640 × 256 (graphisme). Jusqu'à huit couleurs (stables ou clignotantes, Caractères redéfinissables.

INTERFACES
UHF (télévision), RVB et vidéo composite (moniteurs), port cassette, port parallèle imprimante, port analogique (pour manches à balai, etc.), deux prises pour cartouches ROM.

LANGAGES DISPONIBLES
Basic, assembleur du 6502 (intégrés), forth, s-pascal, lisp (aussi disponible en cartouche).

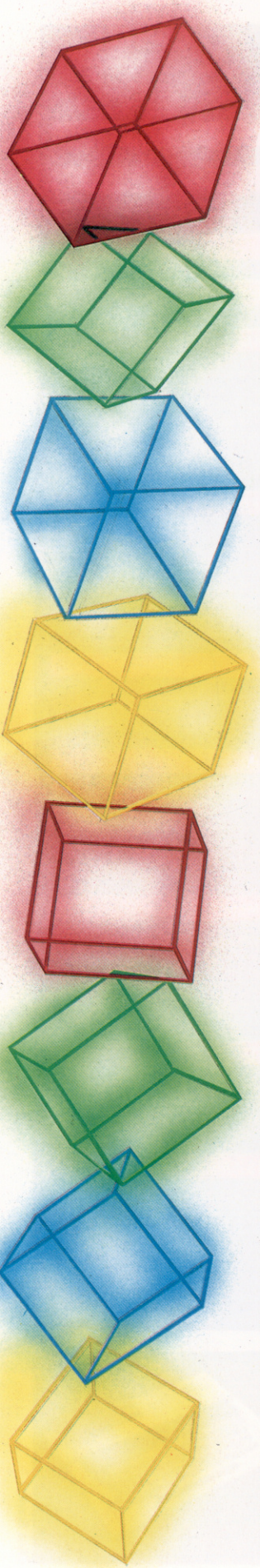
CLAVIER
56 touches type machine à écrire, dont 10 programmables. Les touches de fonction permettent d'entrer des commandes basic par appui d'une seule touche.

DOCUMENTATION
Le manuel de l'utilisateur est bien conçu et facile à lire. C'est une présentation exhaustive du basic de l'appareil et, ce qui est plus inhabituel, de l'assembleur du 6502.

POINTS FORTS
Le graphisme est bon, et les images sont très claires. Le clavier est d'excellente qualité. Bonne version du basic.

POINTS FAIBLES
Les touches multi-fonctions sont parfois difficiles à utiliser. Il n'y a qu'un canal sonore. Mémoire assez faible. Pas de mode vidéotext. Absence d'un port communications.

Chris Stevens



Révolution cubiste

Nous poursuivons notre série d'articles avec un programme permettant de tracer des formes géométriques en trois dimensions, et de les faire tourner.

Le programme que nous présentons ici utilise des principes géométriques élémentaires pour créer des projections d'objets en perspective susceptibles d'être visionnées et représentées sous n'importe quel angle et à n'importe quelle distance.

Les données des objets graphiques figurent au programme dans des instructions DATA. Les données sont constituées de coordonnées en trois dimensions pour chaque extrémité des lignes. A chaque point ainsi défini est également associé un nombre qui indique si la ligne à tracer doit partir de lui ou se terminer par lui.

La transcription de ces coordonnées tridimensionnelles en valeurs à deux dimensions, représentant des points à l'écran, est une opération mathématique simple mais longue. Les coordonnées (X,Y) de chaque point (au plan de l'écran) sont divisées par un facteur représentant la distance entre l'observateur et l'objet. Les coordonnées résultantes sont en outre corrigées par un facteur propre au système de coordonnées du micro. En modifiant le facteur de distance, l'objet peut varier en taille, rétrécissant ou s'agrandissant au fur et à mesure que l'on s'éloigne ou que l'on s'approche.

Une troisième constante peut être utilisée pour modifier l'effet de la projection de l'objet en perspective. En accroissant la valeur de cette dernière, la vue en perspective de l'objet est graduellement accentuée, allant jusqu'à déformer l'angle de vision comme le fait un objectif de type « fish-eye ». En diminuant cette valeur, la vision s'aplatit comme avec un téléobjectif.

En plus de la vue en perspective d'objets graphiques, le programme permet de les tourner et de les représenter sous n'importe quel angle, par le biais d'opérations trigonométriques simples. Les axes sont tournés selon l'angle voulu de sorte que lorsque la projection en perspective est demandée, l'image à l'écran apparaît comme ayant été tournée.

Cette rotation de l'objet peut se faire soit selon l'axe des ordonnées Y (le point de vue semble alors avoir tourné autour de l'objet), soit selon l'axe des abscisses X (le point d'observation semble alors s'être élevé au-dessus de l'objet ou être passé en dessous).

Le programme est particulièrement simple d'utilisation. On fait varier le point de vue en tapant un nombre de 1 à 8. On obtiendra successivement : mouvement vers la gauche du point d'observation, vers la droite, vers le haut, vers le bas, vers l'objet et, s'en éloignant, l'accroissement de l'effet de perspective (« fish-eye »), et sa réduction (téléobjectif).

Le changement de perspective a lieu dans le sous-programme qui trace l'image. Ce dernier prend en compte toutes les coordonnées en trois dimensions, les transcrit en coordonnées en deux dimensions et les affiche à l'écran.

Le calcul des données d'un objet graphique est long mais relativement simple. Les données sont sauvegardées dans des instructions à la fin du programme, à raison de quatre valeurs pour chaque point. Le nombre total de points figure à la première ligne du programme. Pour créer vos propres données (et votre propre dessin à l'écran), il vous faudra les modifier. Les données que nous indiquons ici sont destinées à représenter un cube dont chaque face comportera une ligne en diagonale. Chaque ensemble de quatre valeurs figure selon l'ordre : valeur de tracé, coordonnée X, coordonnée Y, coordonnée Z.

Les valeurs sont calculées en traçant abstraitement les contours de l'objet en trois dimensions.

Avec un crayon imaginaire, passez en revue les différents sommets de l'objet. Si vous vous déplacez vers un point sans tracer de ligne, 4 est utilisé comme première valeur. La valeur 5 indiquera qu'il convient de tracer une ligne en partant du point précédemment représenté. Ces valeurs spécifiques sont utilisées pour simplifier le programme qui est destiné au BBC Micro.

Un tracé réaliste

L'origine des coordonnées (0,0) est au centre de l'écran. Il est préférable que vous en fassiez également le centre de votre objet. L'axe des X est l'axe horizontal, les valeurs positives allant vers le haut. L'axe des Z représente les mouvements d'approche et d'éloignement de l'écran. Le sens positif sur cet axe est celui qui s'approche de l'écran.

Efforcez-vous de maintenir les valeurs X, Y et Z aussi petites que possible. La spécification initiale de l'effet de perspective et la distance à l'objet devront prendre en compte le fait qu'une largeur de 10 unités remplira tout l'écran. Lorsque vous saurez parfaitement trouver les coordonnées d'objets simples et pleins (une pyramide ou un cube), vous pourrez passer à des formes plus complexes. Vous pouvez également essayer d'écrire une routine qui escamote à la vue les arêtes non visibles des objets graphiques.

Cette vision rend le tracé beaucoup plus réaliste. Il faut cependant savoir que cette suppression des parties cachées est une affaire compliquée qui implique des opérations mathématiques complexes et ralentit le programme. Cependant, tel qu'il est, ce programme est suffisamment riche pour vous donner beaucoup à apprendre et à expérimenter.

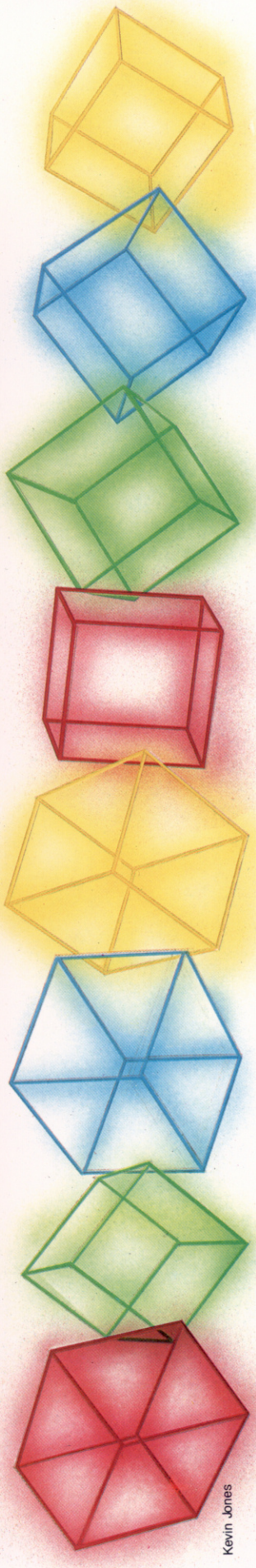
Vous pourrez obtenir des résultats en rapport avec votre habileté à manier intellectuellement des figures graphiques et des formes géométriques, du spectaculaire ou du beau travail.

Version Spectrum

```
 10 LET N=16
 20 DIM P(50)
 21 DIM X(50)
 22 DIM Y(50)
 23 DIM Z(50)
 24 DIM A(50)
 25 DIM B(50)
 40 LET D=10: LET P=0.5
 50 LET SI=SIN 0.09: LET CO=COS 0.09
 60 FOR I=1 TO N
 70 READ P(I),X(I),Y(I),Z(I)
 80 NEXT I
 90:
200 INVERSE 0: GO SUB 300
210 IF INKEY$<>"" THEN GO TO 210
211 IF INKEY$="" THEN GO TO 211
212 LET I$=INKEY$
230 INVERSE 1: GO SUB 300
240 IF I$="1" THEN GO SUB 1000
241 IF I$="2" THEN GO SUB 2000
242 IF I$="3" THEN GO SUB 3000
243 IF I$="4" THEN GO SUB 4000
244 IF I$="5" THEN GO SUB 5000
245 IF I$="6" THEN GO SUB 6000
246 IF I$="7" THEN GO SUB 7000
247 IF I$="8" THEN GO SUB 8000
250 GO TO 200
260:
300 FOR I=1 TO N
310 LET A(I)=X(I)*300/(P+Z(I)+D): LET B(I)=Y(
I)*300/(P+Z(I)+D)
320 NEXT I
330 FOR I=1 TO N
340 IF P(I)=4 THEN PLOT A(I)+128,B(I)+85
345 IF P(I)=5 THEN DRAW A(I)-A(I-1),B(I)-B(I
-1)
350 NEXT I
360 RETURN
370:
1000 FOR I=1 TO N
1010 LET X=X(I)*CO-Z(I)*SI
1020 LET Z=Z(I)*CO+X(I)*SI
1030 LET X(I)=X: LET Z(I)=Z
1040 NEXT I
1050 RETURN
1060:
2000 FOR I=1 TO N
2010 LET X=X(I)*CO+Z(I)*SI
2020 LET Z=Z(I)*CO_Z(I)*SI
2030 LET X(I)=X: LET Z(I)=Z
2040 NEXT I
2050 RETURN
2060:
3000 FOR I=1 TO N
3010 LET Y=Y(I)*CO+Z(I)*SI
3020 LET Z=Z(I)*CO-Y(I)*SI
3030 LET Y(I)=Y: LET Z(I)=Z
3040 NEXT I
3050 RETURN
3060:
4000 FOR I=1 TO N
4010 LET Y=Y(I)*CO-Z(I)*SI
4020 LET Z=Z(I)*CO+Y(I)*SI
4030 LET Y(I)=Y: LET Z(I)=Z
4040 NEXT I
4050 RETURN
4060:
5000 LET D=D*0.9
5010 RETURN
5020:
6000 LET D=D/0.9
6010 RETURN
6020:
7000 LET P=P/0.9
7010 RETURN
7020:
8000 LET P=P*0.9
8010 RETURN
8020:
9000 DATA 4,1,1,1, 5,1,1,-1, 5,-1,1,-1, 5,-1,1
,1,5,1,1,1
9010 DATA 5,1,-1,1, 5,1,-1,-1, 5,-1,-1,-1, 5,-
1,-1,1, 5,1,-1,1
9020 DATA 4,1,-1,-1, 5,1,1,-1, 5,-1,-1,-1, 5,-
1,1-1
9030 DATA 4,-1,1,1, 5,-1,-1,1
```

Version BBC

```
 10 N=16
 20 DIM P(50),X(50),Y(50),Z(50),A(50),B(50)
 30 MODE 0:VDU29,640;512;5
 40 D=10:P=0.5
 50 SI=SIN(0.09):CO=COS(0.09)
 60 FOR I=1 TO N
 70   READ P(I),X(I),Y(I),Z(I)
 80   NEXT I
 90 :
200 GCOL0,3:GOSUB 300
210 I$=GET$
220 V=VAL(I$)
230 GCOL0,0:GOSUB 300
240 ON V GOSUB 1000,2000,3000,4000,5000,6000
,7000,8000 ELSE 150
250 GOTO 200
260 :
300 FOR I=1TO N
310   A(I)=X(I)*1000/(P+Z(I)+D):B(I)=Y(I)*
1000/(P+Z(I)+D)
320   NEXT I
330 FOR I=1 TO N
340   PLOT P(I),A(I),B(I)
350   NEXT I
360 RETURN
370 :
1000 FOR I=1 TO N
1010   X=X(I)*CO-Z(I)*SI
1020   Z=Z(I)*CO+X(I)*SI
1030   X(I)=X:Z(I)=Z
1040   NEXT I
1050 RETURN
1060 :
2000 FOR I=1 TO N
2010   X=X(I)*CO+Z(I)*SI
2020   Z=Z(I)*CO-X(I)*SI
2030   X(I)=X:Z(I)=Z
2040   NEXT I
2050 RETURN
2060 :
3000 FOR I=1 TO N
3010   Y=Y(I)*CO+Z(I)*SI
3020   Z=Z(I)*CO-Y(I)*SI
3030   Y(I)=Y:Z(I)=Z
3040   NEXT I
3050 RETURN
3060 :
4000 FOR I=1 TO N
4010   Y=Y(I)*CO-Z(I)*SI
4020   Z=Z(I)*CO+Y(I)*SI
4030   Y(I)=Y:Z(I)=Z
4040   NEXT I
4050 RETURN
4060 :
5000 D=D*.9
5010 RETURN
5020 :
6000 D=D/.9
6010 RETURN
6020 :
7000 P=P/.9
7010 RETURN
7020 :
8000 P=P*.9
8010 RETURN
8020 :
10000 DATA 4,1,1,1, 5,1,1,-1, 5,-1,1,-1, 5,-1,
1,1,5,1,1,1
10010 DATA 5,1,-1,1, 5,1,-1,-1, 5,-1,-1,-1, 5,
-1,-1,1, 5,1,-1,1
10020 DATA 4,1,-1,-1, 5,1,1,-1, 5,-1,-1,-1, 5,
-1,1,-1
10030 DATA 4,-1,1,1, 5,-1,-1,1
```



Kevin Jones

Pièces du puzzle

La manière la plus efficace de créer des programmes est d'utiliser une structure modulaire. Le pascal encourage cette approche, le basic exige une plus grande discipline.

Un module est une portion de code qui effectue une fonction particulière. Les points d'entrée et de sortie dits « interfaces » de module doivent être définis précisément et les processus qui surviennent entre ces interfaces doivent être entièrement indépendants du reste du programme. Dès qu'un module a été écrit, il peut être traité comme une entité autonome. Les données peuvent entrer et sortir du module, celui-ci demeure toujours seul responsable du traitement interne.

Les modules peuvent être assemblés pour construire un programme sans que le programmeur ait à se demander comment ils exécutent leurs diverses tâches. Un programmeur peut écrire une série de modules afin de les utiliser lorsqu'il en a besoin et il est même possible d'utiliser dans ses programmes un module écrit par un autre.

Une règle très simple doit être suivie : tous les modules doivent avoir un seul point d'entrée et un seul point de sortie. Ce qui signifie que le déroulement des opérations à l'intérieur d'un module doit être planifié de façon à ce qu'il commence à un endroit et que, quel que soit le nombre de boucles et d'embranchements empruntés, tous les cheminements aboutissent à la même sortie.

Les modules correspondent aux algorithmes. Les langages structurés, comme le PASCAL, permettent au programmeur de créer des sous-programmes qui peuvent être appelés par leurs noms, et qui utilisent leurs propres variables. De tels langages encouragent les programmeurs à entrer dans une routine (procédure) et à en sortir par des points d'entrée et de sortie uniques.

En BASIC, à l'aide de la combinaison GOSUB ... RETURN, un sous-programme peut être appelé à partir du programme principal et, après l'exécution du sous-programme, le déroulement du programme se poursuit à la ligne qui suit immédiatement la commande GOSUB. Cependant, il n'existe aucune restriction quant à la destination d'une commande GOSUB. Deux instructions GOSUB différentes peuvent diriger l'exécution du programme vers des lignes différentes d'un sous-programme comportant un seul RETURN, et le résultat peut être complètement différent dans chaque cas. De façon similaire, il n'y a aucune restriction quant au nombre d'instructions RETURN pouvant être utilisées dans un sous-programme.

Cela signifie que le programmeur BASIC doit faire preuve d'auto-discipline. Vous devez d'abord vérifier que toutes les instructions GOSUB concernant un même sous-programme désignent le même numéro de ligne, et que chaque sous-programme ne comporte qu'une seule instruction RETURN. Il est préférable de prendre l'habitude de commencer la première ligne de chaque sous-programme par une instruction REM lui donnant un titre, et d'utiliser cette ligne comme point d'entrée. La ligne RETURN doit être la dernière ligne du sous-programme. Ce n'est pas essentiel, mais tout est ainsi plus clair.

La règle Goto

Une attention particulière doit être exercée avec la commande GOTO, car celle-ci peut compromettre la structure d'un programme. Voici la règle : utilisez uniquement une instruction GOTO pour diriger le déroulement vers une ligne comprise dans le même sous-programme. Vous éliminez ainsi le risque de passer un RETURN ou de diriger le déroulement vers le mauvais RETURN. Il est parfois nécessaire de quitter une routine sans exécuter toutes les lignes. Dans un tel cas, vous n'avez qu'à diriger le déroulement vers la ligne qui renferme l'instruction RETURN.

L'utilisation de GOTO à l'intérieur de boucles est encore plus dangereux. Si le déroulement saute en dehors d'une boucle, BASIC ne peut pas le savoir et suppose que le reste du programme est le corps de cette boucle ! La règle de sécurité est la suivante : à l'intérieur du corps d'une boucle, ne jamais utiliser une instruction GOTO dirigée vers une ligne située à l'extérieur de cette boucle. Si une boucle doit être terminée prématurément, mettre le compteur de boucle ou la variable de test à la valeur finale et utiliser une instruction GOTO désignant la ligne de test (la ligne comportant une instruction NEXT ou WHILE). Tout comme l'instruction RETURN, mettez toujours les instructions NEXT ou WHILE seules sur une ligne pour faciliter les choses.

Les modules peuvent être utilisés indépendamment les uns des autres, vous devez les concevoir de façon à ce que la seule influence qu'ils exercent entre eux s'effectue par l'intermédiaire des données échangées. Le programme principal transmet des données à un module et, à la fin de l'exécution du module, le résultat du traitement est retransmis.

Les données circulent dans des programmes au moyen de variables et la liberté de mouvement d'une variable est nommée son « champ ». De nombreux langages de programmation peu-

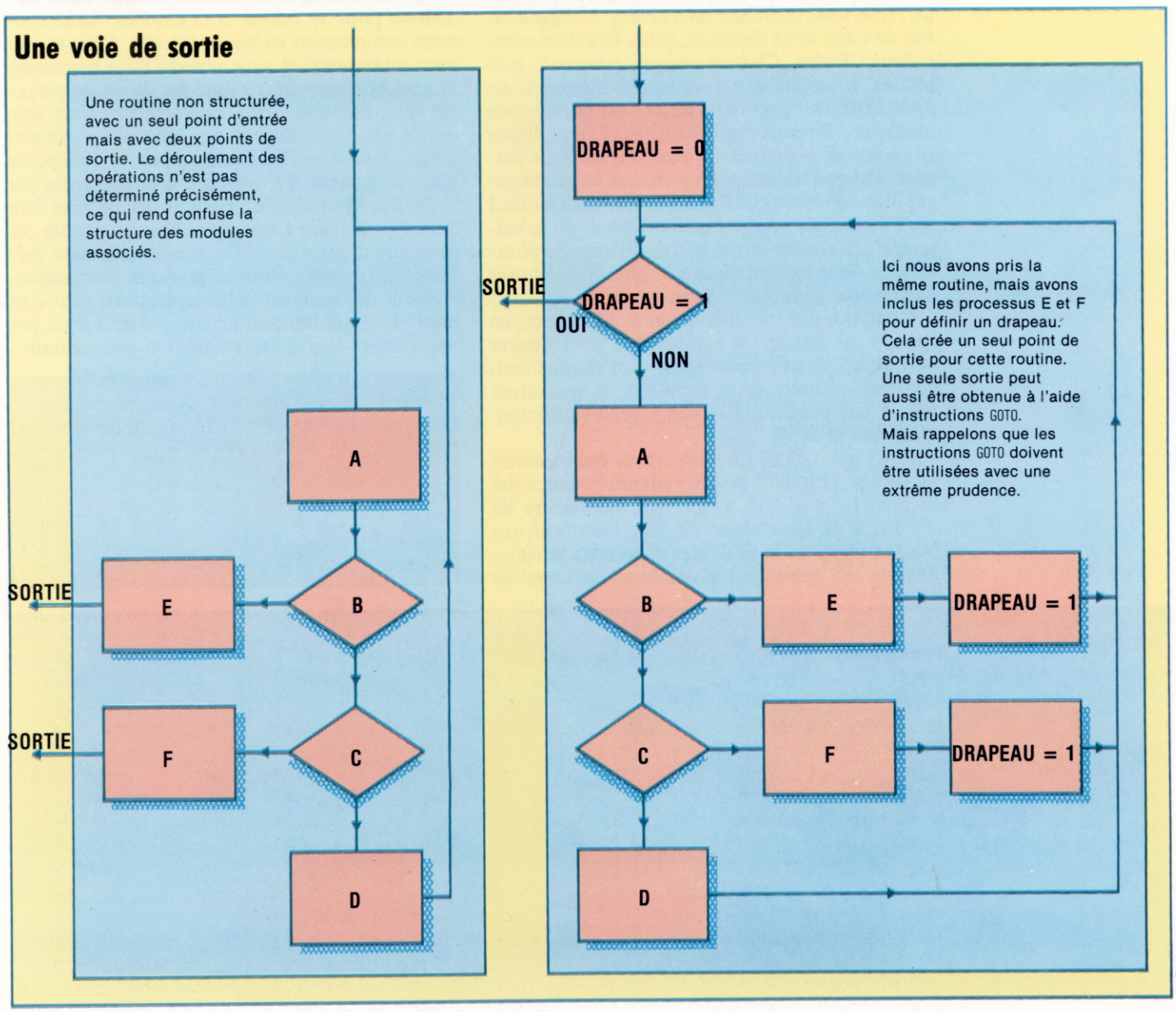
vent restreindre le champ d'une variable à des sous-programmes particuliers. En PASCAL, les variables utilisées dans un sous-programme particulier (procédure) doivent être « déclarées » pour cette procédure. Les variables pour le programme principal sont dites globales et peuvent être utilisées ailleurs à tout endroit du programme. Les variables déclarées à l'intérieur d'une procédure particulière, cependant, sont dites locales dans cette procédure et ne peuvent être utilisées qu'à cet endroit.

Les variables locales peuvent avoir les mêmes noms que les variables globales et utiliser l'une ne modifie pas la valeur de l'autre. L'utilisation d'un langage qui prend en charge des variables locales nous permet d'écrire des sous-programmes sans avoir à se soucier de l'effet qu'auront ces variables dans d'autres routines. Malheureusement, très peu de versions du langage BASIC ont des variables locales, ce qui signifie que si nous désirons écrire des sous-programmes indépendants nous devons simuler l'existence de variables locales.

La manière la plus simple de le faire est d'adopter des conventions d'attribution de noms qui distinguent les variables effectuant des tâches différentes. Certaines conventions existent déjà et sont largement utilisées par les programmeurs. L'utilisation de I, J et K comme compteurs de boucle est très répandue.

Après avoir décrit un programme avec un organigramme, il est très simple de numéroter les sous-programmes impliqués, ou de leur donner un autre type de code. Toute variable globale qui doit devenir locale dans un sous-programme particulier peut alors adopter ce code comme suffixe pour la rendre unique. Ainsi, la routine numéro 5 utilise les variables locales SOM5 et TOTAL5 pour les distinguer des variables SOM12 et TOTAL12 de la routine numéro 12. Mais assurez-vous que le BASIC que vous utilisez ne tient pas uniquement compte des deux premiers caractères ! Les variables qui sont utilisées pour transmettre des valeurs entre des sous-programmes et celles utilisées uniquement dans le programme principal n'ont pas à être codées.

Une voie de sortie



Liz Dixon

Le plaisir des dieux

Valhalla est un jeu d'aventures rempli de références à la mythologie nordique. Destiné au Spectrum et au Commodore 64, il a connu un énorme succès grâce à des graphismes spectaculaires.

Vive le jeu d'aventures ! Par jeu d'aventures, il faut entendre jeu conversationnel : le joueur qui communique avec le programme par l'intermédiaire du clavier, en tapant un verbe suivi d'un nom ou d'une indication de direction, est le héros d'une histoire cohérente (scénario). Il n'incarne pas un personnage particulier, il est lui-même.

Le jeu que nous vous présentons ne comporte pas moins de 81 lieux différents, dont 16 à Asgard (dont le Valhalla proprement dit, véritable paradis, que vous cherchez sans cesse). Midgard est fait de vingt lieux distincts, et les 45 autres constituent l'Enfer. Chaque endroit comporte huit sorties, bien que certaines soient bloquées, ou nécessitent le recours à un objet doté de pouvoirs magiques. Si par exemple vous avez à votre disposition un certain anneau, vous pouvez même passer d'un lieu à un autre. Il y a en tout 36 personnages, bons ou mauvais : il y en a toujours au moins 3 sur l'écran, à n'importe quel moment. Ils se battent, s'offrent du vin ou de la nourriture, se poursuivent, se cherchent ou encore jettent des choses à la tête des gens qu'ils détestent.

Vous pouvez très bien vous mêler à eux, en entrant au clavier les actions que vous désirez accomplir. Vous pouvez aussi, plus simplement, contempler toute cette agitation en spectateur tout à fait passif, mais vous y perdrez immanquablement la vie.

Il n'y a pas de route toute tracée qui mènerait au succès ; le joueur peut s'y prendre de bien des façons. Il vous faut avant tout convaincre les « bons » de vous aider, en leur montrant que vous le méritez. Si vous êtes d'humeur morose, il vous est également possible de changer de camp, et de vous faire assister par les méchants...

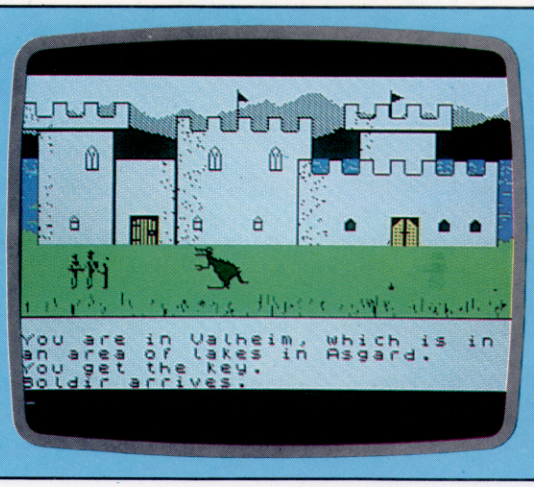
Le graphisme est réussi, surtout dans la version Commodore, cet appareil ayant des possibilités supérieures à celles du Spectrum. Le décor de chaque scène est d'abord tracé (en couleurs vives sur le Spectrum, en teintes pastel sur le Commodore 64), puis les personnages font leur apparition. Enfin c'est le tour des objets : nourriture, vin, joyaux, clés, armes.

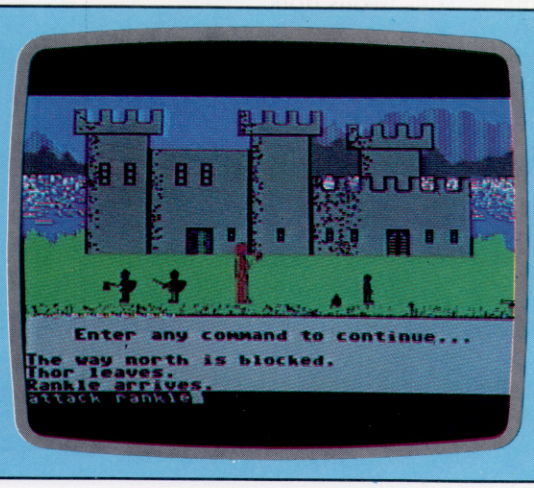
Les personnages se lancent ensuite dans des actions pour le moins compliquées, heureusement commentées en bas de l'écran. C'est là que vous intervenez, et vous pouvez faire beaucoup de choses : convaincre l'un des dieux de partager son trésor avec vous, ou l'attaquer pour voir quelle est votre force. Au bout d'un certain temps, vous chercherez sans doute à quitter les lieux et à partir à l'aventure à la recherche du Valhalla. Mais c'est un paradis qui ne peut être atteint que grâce à certains objets magiques, et plusieurs d'entre eux sont abominablement difficiles à trouver. Vous connaîtrez immanquablement des moments d'exaspération, peut-être inutiles, et qui limitent un peu l'attrait d'un jeu par ailleurs tout à fait original et passionnant.

Valhalla : Spectrum 48 K.
Commodore 64.
Éditeurs : Legend, Freepost.
Auteurs : Graham Asher, Richard Edwards, Charles Goodwind, James Learmont, Jan Ostler, Andrew Owen, John Peel.
Manche à balai : inutile.
Format : cassette.

Le jour et la nuit

Valhalla est l'un de ces jeux récemment publiés, dans lesquels le passage du temps est souligné graphiquement par un assombrissement de l'écran, ce qui permet de montrer aussi bien le jour que la nuit. (Cl. Liz Heaney.)





Le cycle de l'anneau

Jusqu'à présent, nous avons construit des sous-programmes utilisant les capacités de haute résolution du Commodore 64. Dans ce numéro, nous verrons une routine pour dessiner des cercles.

Il est impossible de produire un cercle dessiné avec précision sur un ordinateur domestique. Le degré de précision que nous pouvons obtenir dépend de la méthode utilisée, de la longueur et de la complexité de la routine finale. Le dessin de cercles à partir du BASIC implique habituellement des calculs de fonctions sinus et cosinus, ou de carrés et racines carrées, pour obtenir les coordonnées de points sur la circonférence du cercle à dessiner. Ces deux méthodes amènent cependant des difficultés lorsque nous essayons de les appliquer en langage machine; aussi en verrons-nous une autre particulièrement adaptée à une solution en langage machine.

Cette méthode considère que le diamètre d'un cercle doit être divisé en un nombre égal de parties, chacune de largeur W. A chaque division nous pouvons penser à un segment qui atteint un point C à la verticale, sur la circonférence du cercle. Le diagramme montre un de ces segments, à N divisions de l'extrémité gauche du diamètre AB. En joignant A et B à C, nous formons deux triangles rectangles ACD et BCD, comme indiqué ci-contre.

A l'aide du théorème de Pythagore, nous pouvons écrire les expressions suivantes à partir de ce diagramme :

$$AC^2 = AD^2 + CD^2$$
$$CB^2 = DB^2 + CD^2$$

Si nous additionnons ces équations, nous obtenons :

$$AC^2 + CB^2 = AD^2 + DB^2 + 2CD^2$$

Mais il y a une propriété particulière des cercles, selon laquelle le triangle ABC est aussi rectangle. Nous avons donc :

$$AC^2 + CB^2 = AB^2$$

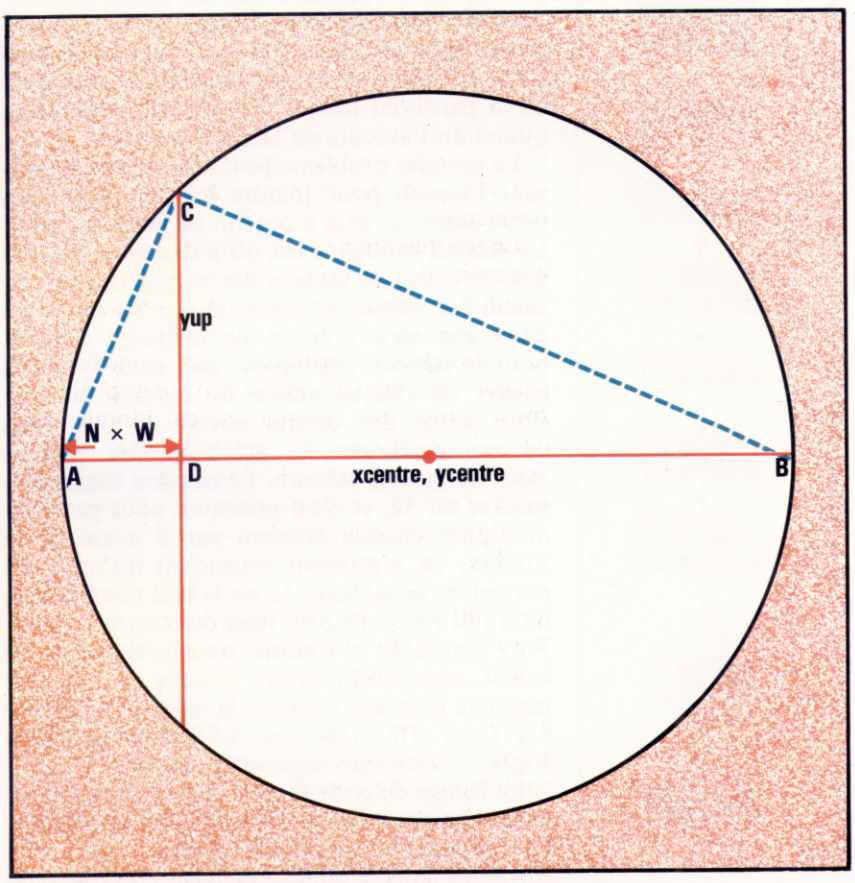
En mettant cela dans le membre de gauche de l'équation précédente, nous obtenons :

$$AB^2 = AD^2 + DB^2 + 2CD^2$$

CD correspond à « yup » — distance du diamètre à la circonférence. AD est N × W et AB est 2 × R, ou R est le rayon du cercle. En substituant ces valeurs dans l'équation et en réarrangeant ces facteurs, on trouve :

$$2 \times yup^2 =$$
$$(2 \times R)^2 - (N \times W)^2 - (2 \times R - N \times W)^2$$
$$yup^2 = 2 \times R \times N \times W - N^2 \times W^2$$

Si nous décidons de diviser le diamètre en 64 parties égales, alors W = 2 × R/64, ce qui se



Liz Dixon

réduit à W = R/32. En substituant cela dans l'équation, nous avons :

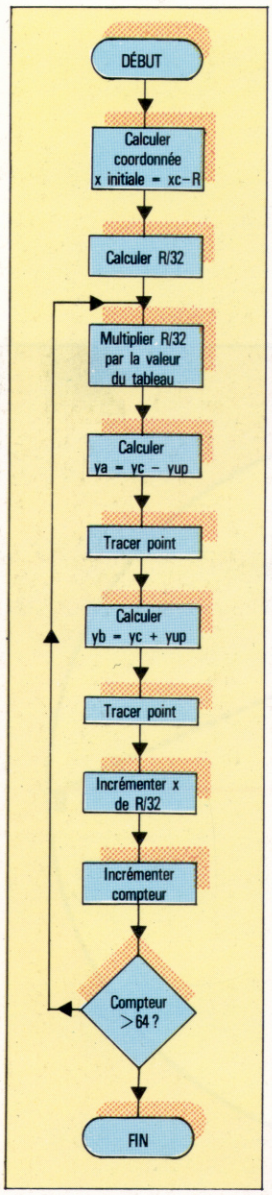
$$yup^2 = 2 \times R \times N \times R/32 - N^2 \times R^2/32^2$$
$$yup^2 = R^2/32^2 \times (64 \times N - N^2)$$

et en prenant la racine carrée des deux membres :

$$yup = R/32 \times SQR\ (64 \times N - N^2)$$

Si nous partons de x = xcentre-R et l'incrémentons de 64 pas égaux, alors, à chaque incrémentation, la distance à la circonférence (yup) sera donnée par la formule à laquelle nous sommes parvenus, où N est le nombre de pas. Quoique ce résultat inclue une racine carrée, il faut noter que l'expression dont on prend la racine est indépendante des coordonnées du centre ou du rayon du cercle. Nous pouvons donc calculer une table de valeurs pour la fonction racine carrée, qui donne une solution pour chaque valeur de N de 0 à 64. La coordonnée absolue y pour chaque incrémentation de x est :

ya = ycentre - yup



Nous pouvons aussi utiliser la symétrie du cercle pour calculer la coordonnée y correspondante dans la moitié inférieure du cercle :

yb = ycentre + yup

L'organigramme montre comment chaque point de la circonférence peut être calculé. Nous voyons que la routine devrait être assez rapide puisqu'elle ne doit effectuer qu'une seule multiplication pour chaque point tracé. Cependant il y a deux problèmes à ce stade de la routine. *Primo,* on n'obtiendra pas un cercle continu, mais une série de points autour de la circonférence. *Secundo,* bien que cette technique produise des cercles bien définis lorqu'elle est utilisée à partir du BASIC, il y a des inexactitudes quand on l'exécute en langage machine.

Le premier problème peut être résolu en utilisant Linesub pour joindre les points par des petits segments afin d'obtenir un cercle continu. Le second problème est dû à des inexactitudes dans le calcul du tableau des racines carrées. Le calcul des valeurs en BASIC et leur entrée (POKE) dans une série d'octets de mémoire réservés pour le tableau impliquent que seule la partie entière de chaque valeur est stockée en fait. Pour avoir des cercles mieux définis, nous devons améliorer la précision des valeurs stockées dans le tableau. Le nombre maximal à stocker est 32, et c'est pourquoi nous pouvons multiplier chaque nombre par 8 avant de le stocker, en n'utilisant cependant qu'un octet par entrée de tableau. 32 est le seul nombre qui, multiplié par 8, ne peut tenir dans un seul octet. Pour simplifier la routine, nous prendrons une valeur approchée. Notre routine en langage machine peut alors diviser la valeur du tableau par 8, en effectuant trois LSR (Logical Shifts Right = décalages logiques à droite).

Le listage de code source réserve 65 octets au début du programme pour stocker le tableau, dont le premier octet porte un label. Les entrées suivantes dans le tableau peuvent être obtenues par adressage indexé. Le listage du code source peut être entré et assemblé en mémoire comme d'habitude. Avant de sauvegarder (SAVE) le code assemblé, le programme BASIC suivant doit être entré et exécuté (RUN). Le programme calcule les 65 valeurs du tableau requises par notre programme Circsub, multiplie ces valeurs par 8 et POKE le résultat dans la zone de mémoire réservée dans votre programme en langage machine. Une fois que le programme de création de tableau a été exécuté (RUN), le langage machine peut être sauvegardé (SAVE), mais assurez-vous que vous sauvegardez la zone de tableau avec le code objet. Le tableau commence en $C500. Cela fait, le tableau sera chargé automatiquement chaque fois que vous chargerez Circsub.

```
5 REM***** N CREER TABLEAU CIRCSUB *****
10 FORN=0TO64
20 X=SQR(64*N-N12)*8
25 X%=X:DF=X-X%
27 IFRE>=.5THENX%=X%+1
28 IFX%>255THENX%=255
29 POKE50432+N,X%
30 NEXT
```

Le programme suivant montre comment on peut utiliser Circsub à partir d'un programme BASIC. Il suffit de spécifier les coordonnées du centre et le rayon. Le sous-programme à la ligne 2000 sépare la coordonnée x en octet LO et octet HI, puis POKE les valeurs spécifiées dans Circsub, et fait l'appel SYS approprié. Notez que, tandis que Circsub utilise Linesub qui, à son tour, utilise Plotsub, les trois sous-programmes sont chargés au début du programme. Celui-ci dessine des cercles de rayon croissant sur l'écran.

```
3 REM***** PROGRAMME TEST CIRCSUB *****
5 DN=8:REM POUR CASSETTE DN=1
10 IFA=0THENA=1:LOAD"PLOTSUB.HEX",DN,1
15 IFA=1THENA=2:LOAD"LINESUB.HEX",DN,1
20 IFA=2THENA=3:LOAD"CIRCSUB.HEX",DN,1
100 GOSUB1000
110 YC=100:R=2
120 FORXC=30TO210STEP7
130 R=R+3
140 GOSUB2000
150 NEXT
160 GETA$:IFA$=""THEN160
170 GOSUB3000
180 END
1000 REM ***** METTRE HIRES *****
1010 POKE 49408,1:POKE49409,1
1020 POKE49410,3
1030 SYS49422
1040 RETURN
1050 :
2000 REM ***** ENTRER CIRCSUB *****
2010 CHI=INT(XC/256):CLO=XC-256*CHI
2020 POKE50497,CLO:POKE50498,CHI
2030 POKE50499,YC
2040 POKE50500,R
2050 SYS50521
2060 RETURN
2070 :
3000 REM ***** ENLEVER HIRES *****
3005 RESTORE
3007 PRINTCHR$(147):REM EFFACER ECRAN
3008 PRINTTAB(10)"CIRCSUB VARIABLES"
3009 PRINT
3010 POKE49408,0:SYS49422
3030 FORI=50497TO50497+23STEP2
3035 READA$
3040 PRINTTAB(2):A$,PEEK(I),
3045 READA$
3047 PRINTA$,PEEK(I+1)
3050 NEXT
3060 RETURN
3070 :
5000 DATAXCTRLO,XCTRHI,YCTR,RADIUS,INTR
5010 DATAREMR,TOTX,RESREM,RESLO,RESHI,OLDXLO
5020 DATAOLDXHI,NEWXLO,NEWXHI,OLDYA
5030 DATAOLDYB,NEWYA,NEWYB,XFLAG,YFLAG,OLDY,NEWY
5040 DATAINTTAB,REMTAB
```

Le langage machine peut être entré comme une série d'instructions DATA si vous n'avez pas d'assembleur. Le listage suivant doit être tapé et exécuté pour charger Circsub en mémoire. Les chargeurs BASIC pour Linesub et Circsub devaient également être chargés et exécutés avant de charger le programme de démonstration. Comme les trois routines seront en mémoire, les lignes 10, 15 et 20 du programme Demo doivent être omises. Le chargeur BASIC pour Circsub contient déjà les données du tableau, et il n'est donc pas nécessaire d'exécuter « Créer tableau » dans ce cas.

Dans un prochain numéro

La rubrique « langage machine » d'ABC Informatique s'est jusqu'à présent consacrée aux ordinateurs familiaux basés sur les microprocesseurs Z80 et 6502. Mais il existe un troisième microprocesseur assez répandu, le 6809. Nous allons bientôt nous intéresser à cette puce et apprendre aux possesseurs de 6809 à programmer eux aussi en langage machine.

Programme chargeur basic

```
10 REM**** CHARGEUR BASIC POUR CIRCUB ****
20 FORI=50432TO50432+498
30 READA:POKEI,A
40 CC=CC+A
50 NEXT I
60 READA:IFA<>CCTHENPRINT"CHECKSUM ERROR"
100 DATA0,63,89,108,123,137,149,159
110 DATA169,177,185,193,199,205,211
120 DATA216,221,226,230,233,237,240
130 DATA243,245,247,249,251,252,253
140 DATA254,255,255,255,255,255,254
150 DATA253,252,251,249,247,245,243
160 DATA240,237,233,230,226,221,216
170 DATA211,205,199,193,185,177,169
180 DATA159,149,137,123,108,89,63,0
190 DATA205,0,100,80,2,16,16,0,0,0,29
200 DATA1,31,1,100,100,100,100,0,0,120
210 DATA100,0,0,72,138,72,152,72,173
220 DATA68,197,41,31,141,70,197,173,68
230 DATA197,160,5,74,136,208,252,141
240 DATA69,197,173,65,197,56,237,68,
250 DATA197,141,77,197,141,75,197,173
260 DATA66,197,233,0,141,78,197,141,76
270 DATA197,173,67,197,141,79,197,141
280 DATA80,197,169,0,170,141,71,197
290 DATA189,0,197,160,3,74,136,208,252
300 DATA141,87,197,189,0,197,41,7,141
310 DATA88,197,172,68,197,169,0,141,72
320 DATA197,141,73,197,141,74,197,173
330 DATA72,197,24,109,88,197,141,72
340 DATA197,201,8,144,23,56,233,8,141
350 DATA72,197,173,73,197,24,105,1,141
360 DATA73,197,173,74,197,105,0,141,74
370 DATA197,173,73,197,24,109,87,197
380 DATA141,73,197,173,74,197,105,0
390 DATA141,74,197,136,208,198,160,5
400 DATA78,74,197,110,73,197,110,72
410 DATA197,136,208,244,173,72,197,201
420 DATA16,144,9,173,73,197,24,105,1
430 DATA141,73,197,173,67,197,56,237
440 DATA73,197,141,81,197,173,81,197
450 DATA141,86,197,173,79,197,141,85
460 DATA197,32,165,198,173,67,197,24
470 DATA109,73,197,141,82,197,173,82
480 DATA197,141,86,197,173,80,197,141
490 DATA85,197,32,165,198,173,77,197
500 DATA141,75,197,173,78,197,141,76
510 DATA197,173,81,197,141,79,197,173
520 DATA82,197,141,80,197,173,71,197
530 DATA24,109,70,197,141,71,197,201
540 DATA32,144,26,173,71,197,56,233,32
550 DATA141,71,197,173,77,197,24,105,1
560 DATA141,77,197,173,76,197,105,0
570 DATA141,78,197,173,77,197,24,109
580 DATA69,197,141,77,197,173,78,197
590 DATA105,0,141,78,197,232,224,65
600 DATA240,3,76,153,197,104,168,104
610 DATA170,104,96,169,0,141,83,197
620 DATA141,84,197,173,75,197,205,77
630 DATA197,208,3,238,83,197,173,85
640 DATA197,205,86,197,208,3,238,84
650 DATA197,173,83,197,45,84,197,208
660 DATA39,173,75,197,141,0,195,173,76
670 DATA197,141,1,195,173,85,197,141,4
680 DATA195,173,77,197,141,2,195,173
690 DATA78,197,141,3,195,173,86,197
700 DATA141,5,195,32,14,195,96
710 DATA67223:REM*CHECKSUM*
```

Programme Circsub

```
                ;++++++++++++++++++
                ;++++++++++++++++++
                ;++             ++
                ;++  CIRCSUB 64  ++
                ;++             ++
                ;++++++++++++++++++
                ;++++++++++++++++++
                ;++++ VARIABLES LINESUB ++++
                ;
                LINSUB = $C30E
                X1LO   = $C300
                X1HI   = $C301
                X2LO   = $C302
                X2HI   = $C303
                Y1     = $C304
                Y2     = $C305
                *      = $C500
                ;
                ;++++ VARIABLES CIRCSUB ++++
                ;
                TABLE  ****65          ; LOOK UP TABLE
                XCTRLO ****1
                XCTRHI ****1
                YCTR   ****1
                RADIUS ****1
                INTR   ****1
                REMR   ****1
                TOTX   ****1
                RESREM ****1
                RESLO  ****1
                RESHI  ****1
                OLDXLO ****1
                OLDXHI ****1
                NEWXLO ****1
                NEWXHI ****1
                OLDYA  ****1
                OLDYB  ****1
                NEWYA  ****1
                NEWYB  ****1
                XFLAG  ****1
                YFLAG  ****1
                OLDY   ****1
                NEWY   ****1
                INTTAB ****1
                REMTAB ****1
                ;
                ;++++ ENTRER REGISTRES SUR PILE ++++
                ;
                       PHA
                       TXA
                       PHA
                       TYA
                       PHA
                ;
                ;++++ CALC INT(R/32) ET REMR ++++
                ;
AD 44 C5        LOOP1  LDA RADIUS
29 1F                  AND #$1F
8D 46 C5               STA REMR
AD 44 C5               LDA RADIUS
A0 05                  LDY #$05
4A              LOOP1  LSR A
88                     DEY
D0 FC                  BNE LOOP1
8D 45 C5               STA INTR
                ;
                ;++++ CALC COORDONNEE X INITIALE ++++
                ;
AD 41 C5               LDA XCTRLO
38                     SEC
ED 44 C5               SBC RADIUS
8D 4D C5               STA NEWXLO
8D 4B C5               STA OLDXLO
AD 42 C5               LDA XCTRHI
E9 00                  SBC #$00
8D 4E C5               STA NEWXHI
8D 4C C5               STA OLDXHI
                ;
                ;++++ METTRE EN VALEURS INITIALES DE Y ++++
                ;
AD 43 C5               LDA YCTR
8D 4F C5               STA OLDYA
8D 50 C5               STA OLDYB
                ;
                ;++++ COMPTEUR DE ZERO ET TOTX ++++
                ;
A9 00                  LDA #$00
AA                     TAX
8D 47 C5               STA TOTX
                ;
                ;++++ CALC INTTAB ET REMTAB ++++
                ;
                NEXTPT
BD 00 C5               LDA TABLE,X
A0 03                  LDY #$03
                LOOP2
4A                     LSR A
88                     DEY
D0 FC                  BNE LOOP2
8D 57 C5               STA INTTAB
                ;
BD 00 C5               LDA TABLE,X
29 07                  AND #$07
8D 58 C5               STA REMTAB
                ;
                ;++++ MULTIPLIER TABLEAU PAR RAYON ++++
                ;
AC 44 C5               LDY RADIUS
A9 00                  LDA #$00
8D 48 C5               STA RESREM
8D 49 C5               STA RESLO       ;ZERO ANSWER
8D 4A C5               STA RESHI
                ;
AD 48 C5        AGAIN  LDA RESREM
18                     CLC
6D 58 C5               ADC REMTAB
8D 48 C5               STA RESREM
C9 08                  CMP #$08        ;>= 8?
90 17                  BBC NOCARR
38                     SEC
E9 08                  SBC #$08        ;RESET REM
8D 48 C5               STA RESREM
AD 49 C5               LDA RESLO
18                     CLC
69 01                  ADC #$01        ;INC RESULT
8D 49 C5               STA RESLO
AD 4A C5               LDA RESHI
69 00                  ADC #$00
8D 4A C5               STA RESHI
                NOCARR
AD 49 C5               LDA RESLO
18                     CLC
6D 57 C5               ADC INTTAB    ; ADD INTEGER PART
8D 49 C5               STA RESLO
AD 4A C5               LDA RESHI
69 00                  ADC #$00
8D 4A C5               STA RESHI
88                     DEY
D0 C6                  BNE AGAIN
                ;++++ DIVISER RESULTAT PAR 32 ++++
                ;
A0 05                  LDY #$05
                LOOP3
4E 4A C5               LSR RESHI
6E 49 C5               ROR RESLO
6E 48 C5               ROR RESREM
88                     DEY
D0 F4                  BNE LOOP3
                ;
AD 48 C5               LDA RESREM
C9 10                  CMP #$10        ; REM >=16?
90 09                  BBC NOCRRY
AD 49 C5               LDA RESLO
18                     CLC
69 01                  ADD #$01        ; ADD 1
8D 49 C5               STA RESLO
                ;
                ;++++ CALC 1RE COORDONNEE Y ++++
                ;
                NOCRRY
AD 43 C5               LDA YCTR
38                     SEC
ED 49 C5               SBC RESLO
8D 51 C5               STA NEWYA
                ;
                ;++++ DESSINER LIGNE ++++
                ;
AD 51 C5               LDA NEWYA
8D 56 C5               STA NEWY
AD 4F C5               LDA OLDYA
8D 55 C5               STA OLDY
20 A5 C6               JSR DRAW
                ;
                ;++++ CALC 2E COORDONNEE Y ++++
                ;
AD 43 C5               LDA YCTR
18                     CLC
6D 49 C5               ADC RESLO
8D 52 C5               STA NEWYB
                ;
                ;++++ DESSINER LIGNE ++++
                ;
AD 52 C5               LDA NEWYB
8D 56 C5               STA NEWY
AD 50 C5               LDA OLDYB
8D 55 C5               STA OLDY
20 A5 C6               JSR DRAW
                ;
                ;++++ CALC 2E COORDONNEE Y ++++
                ;
AD 43 C5               LDA YCTR
18                     CLC
6D 49 C5               ADC RESLO
8D 52 C5               STA NEWYB
                ;
                ;++++ DESSINER LIGNE ++++
                ;
AD 52 C5               LDA NEWYB
8D 56 C5               STA NEWY
AD 50 C5               LDA OLDYB
8D 55 C5               STA OLDY
20 A5 C6               JSR DRAW
                ;
                ;++++ INTERVERTIR ANC. ET NOUV. ++++
                ;
AD 4D C5               LDA NEWXLO
8D 4B C5               STA OLDXLO
AD 4E C5               LDA NEWXHI
8D 4C C5               STA OLDXHI
AD 51 C5               LDA NEWYA
8D 4F C5               STA OLDYA
AD 52 C5               LDA NEWYB
8D 50 C5               STA OLDYB
                ;
                ;++++ MODIF. COORD. Y ++++
                ;
AD 47 C5               LDA TOTX
18                     CLC
6D 46 C5               ADC REMR
8D 47 C5               STA TOTX        ;ADD REMR TO TOTAL
C9 20                  CMP #$20        ;>=32?
90 1A                  BCC NOINC
AD 47 C5               LDA TOTX
38                     SEC
E9 20                  SBC #$20        ;RESET TOTX
8D 47 C5               STA TOTX
AD 4D C5               LDA NEWXLO
18                     CLC
69 01                  ADC #$01        ; INC X COORD
8D 4D C5               STA NEWXLO
AD 4C C5               LDA OLDXHI
69 00                  ADC #$00
8D 4E C5               STA NEWXHI
                NOINC
AD 4D C5               LDA NEWXLO
18                     CLC
6D 45 C5               ADC INTR
8D 4D C5               STA NEWXLO
AD 4E C5               LDA NEWXHI
69 00                  ADC #$00
8D 4E C5               STA NEWXHI
                ;
                ;++++ INCREMENTEUR COMPTEUR ++++
                ;
E8                     INX
E0 41                  CPX #$41
F0 03                  BEQ FINISH
4C 99 C5               JMP NEXTPT
                ;
                ;++++ SORTIR REGISTRES DE PILE ++++
                ;
                FINISH
68                     PLA
A8                     TAY
68                     PLA
AA                     TAX
68                     PLA
60                     RTS
                ;
                ;++++ VERIF. POINT ++++
                ;
                DRAW
A9 00                  LDA #$00
8D 53 C5               STA XFLAG       ;ZERO FLAGS
8D 54 C5               STA YFLAG
                ;
AD 4B C5               LDA OLDXLO
CD 4D C5               CMP NEWXLO
D0 03                  BNE NEXFLG
                NOXFLG
AD 55 C5               LDA OLDY
CD 56 C5               CMP NEWY
D0 03                  BNE NOYFLG
EE 54 C5               INC YFLAG
                NOYFLG
AD 53 C5               LDA XFLAG
2D 54 C5               AND YFLAG
D0 27                  BNE NODRAW
                ;
                ;++++ DESSINER LIGNE ++++
                ;
AD 4B C5               LDA OLDXLO
8D 00 C3               STA X1LO
AD 4C C5               LDA OLDXHI
8D 01 C3               STA X1HI
AD 55 C5               LDA OLDY
8D 04 C3               STA Y1
AD 4D C5               LDA NEWXLO
8D 02 C3               STA X2LO
AD 4E C5               LDA NEWXHI
8D 03 C3               STA X2HI
AD 56 C5               LDA NEWY
8D 05 C3               STA Y2
                ;
20 0E C3               JSR LINSUB
                ;
                NODRAW
60                     RTS
```

Par la bande

Audiogenic s'est fait une bonne réputation par ses programmes destinés aux ordinateurs Commodore. Cette société distribue aujourd'hui aussi bien des logiciels que des périphériques.

Un peu d'air
Les bureaux d'Audiogenic sont situés à Sutton Park, en dehors de Reading. La compagnie s'est installée là en avril 1984; ses anciens locaux, au centre de Reading, n'étant plus suffisants.

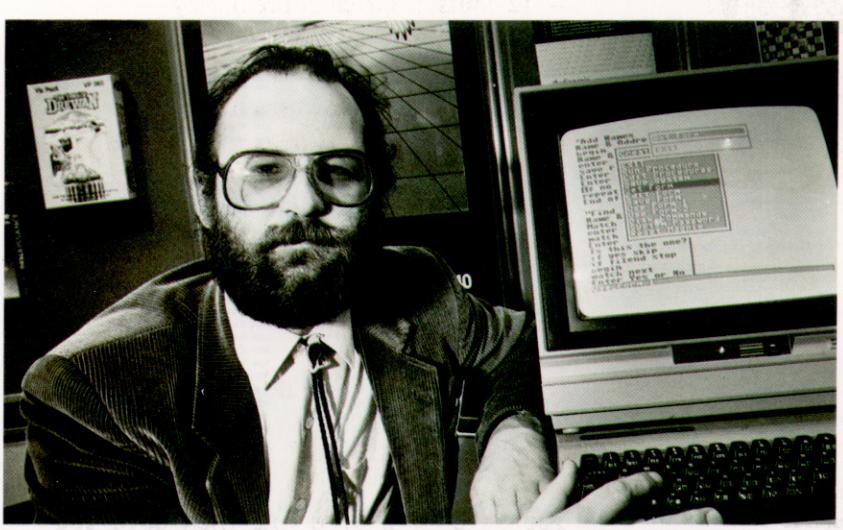
Martin Maynard, fondateur et responsable du management d'Audiogenic, décrit sa compagnie comme étant « une firme de production et de distribution ». Lui-même travailla d'abord pour l'industrie du disque, et, vers le début des années soixante-dix, créa à Reading un studio d'enregistrement, ainsi qu'une petite installation permettant la duplication des cassettes.

Tout vint de là; en 1978, Audiogenic se vit demander par les responsables régionaux de l'électricité britannique la livraison de copies d'un logiciel pour ordinateur. La firme acquit un nouvel équipement pour satisfaire à la demande, et signa peu après un contrat avec Commodore, afin d'assurer la duplication des programmes destinés au micro-ordinateur PET.

Audiogenic prit ensuite en main la distribution et la mise en vente du catalogue Commodore, vendant aussi des livres et des revues spécialisées. Après le lancement du Vic-20 en 1981, Maynard décida d'acquérir les droits des programmes réalisés pour cet appareil par des firmes américaines. Aujourd'hui encore, ils représentent 80 % du catalogue de la firme, et entre 85 et 90 % de son chiffre d'affaires annuel, qui atteint près de vingt millions de francs.

Maynard estime qu'Audiogenic a d'ores et déjà produit plus d'un million de cassettes. « Notre point fort, c'est l'étendue de notre catalogue. De cette façon nous sommes mieux à même de comprendre l'évolution du marché. » Et il ajoute : « Cela fait six ans que nous sommes de la partie, et nous avons eu le temps de savoir ce qui se passe. Si l'on prend l'ensemble des programmes distribués l'année dernière, 20 à 30 pour cent d'entre eux encombrent encore les rayons. Les auteurs perdent peu à peu le contact avec le public. »

C'est là une approche prudente, qui ne met en avant que des produits testés et bien au point, en opposition complète avec la politique du « tout ou rien » chère à bien des firmes. David Smithson, responsable des produits, remarque : « Nous ne sommes pas le genre à nous acheter des Porsche. Il y a vraiment des compagnies qui font tout pour faire la culbute. »



Le créateur
Martin Maynard a fondé Audiogenic au début des années soixante-dix. Ce n'était alors qu'un simple studio d'enregistrement.

Des activités diversifiées

Audiogenic emploie actuellement vingt-cinq personnes; mais Dave Middleton, le principal programmeur de la firme (auteur de Magpie, une base de données remarquable), ne travaille qu'au contrat. Les programmes édités par la compagnie sont essentiellement des utilitaires. Maynard explique : « Les ordinateurs auront toujours un côté ludique, mais en ce moment l'intérêt pour les jeux disparaît peu à peu, et les logiciels seront bientôt des outils bien plus utiles. Quand un produit se vend à deux cents ou trois cents exemplaires par mois, on peut penser que ce n'est pas beaucoup. Mais il continuera à se vendre au même rythme pendant une année entière. »

Cela ne veut pas dire qu'Audiogenic néglige entièrement le marché des jeux. Motor Mania a été un très gros succès, et récemment la firme a lancé Alice in Videoland, un programme d'origine américaine et destiné au Commodore 64, et qui n'occupe pas moins de 90 K!

Audiogenic s'est récemment installé dans des bureaux plus vastes, et a également renouvelé son matériel de duplication. Un programme est ainsi reproduit à de nombreuses reprises sur une bande magnétique continue, qui est ensuite découpée et placée à l'intérieur d'une cassette. C'est un procédé bien plus rapide et efficace : auparavant il fallait dupliquer chaque cassette séparément, d'où des problèmes de stockage et d'encombrement.

Tout en continuant à distribuer des produits réalisés par d'autres compagnies, qu'elles soient britanniques ou américaines, la société Audiogenic entend diversifier ses activités, c'est ainsi qu'elle a mis en vente Koala Pad, une tablette graphique mise au point aux États-Unis. Elle prévoit de la même façon le lancement des logiciels pour les ordinateurs MSX, ainsi que pour le futur Commodore 16.

PROGRAMME N° 29

SPÉCIAL HÔTEL

Nous avons introduit un programme de gestion de comptes clients. Pourquoi ne pas aller un peu plus loin, en nous attardant sur une application particulière : SPÉCIAL HÔTEL ?

Ce petit programme peut, moyennant quelques menues transformations, être adapté à des locations de particuliers, qu'il s'agisse de location d'appartement ou de location de voiture. En effet, ce programme présente une base modulable d'un calcul du coût d'un séjour, c'est-à-dire, d'une manière générale, d'une prestation de service.

Pour rester compatible avec la mémoire de tous les micros du marché, nous avons sciemment limité le nombre de chambres (NC) à douze, ainsi que les paramètres s'y reportant :

total des impulsions téléphoniques

coût des services annexes

coût d'une chambre

Rien ne vous empêche, si vous disposez d'une machine et d'une mémoire adéquates, d'augmenter le nombre de ces données.

```
10 REM Calcul du coût d'un séjour
20 REM d'un client dans un hôtel (par ex
   emple)
30 REM Variables utilisées
50 REM NM$ nom de l'occupant ou de l'o
   ccupante
60 REM C nombre de chambre par client
70 REM NC Numéro de la chambre
80 REM K Catégorie d'une chambre
90 REM NN Nombre de nuitées
100 REM TI Nombre d'impulsions téléphoni
    ques
110 REM SA Coût des services annexes
120 REM CC Coût de location d'une chamb
    re
130 REM CT Coût total du séjour
140 REM
145 DIM NC(12),K(12),NN(12),TI(12),
    SA(12) CC(12)
```

Pour chaque client de l'hôtel, on peut saisir les paramètres concernant au plus douze chambres :
— 12 totaux d'impulsions téléphoniques (TI(12)) ;
— 12 coûts de services annexes (SA(12)) ;
— 12 coûts d'une chambre (CC(12)).

Vous pouvez éventuellement augmenter ce nombre en fonction de la mémoire du micro dont vous disposez.

```
150 INPUT "NOM DE L'OCCUPANT:":NM$
```

On tapera FIN pour stopper le déroulement du programme.

```
160 IF NM$ = "FIN" GOTO 460
170 CT = 0
```

On initialise la variable coût total du séjour CT à zéro.

```
180 ? "NOMBRE DE CHAMBRES RETENUES:"
190 INPUT C
200 IF C  = 0 or C  12 THEN GOTO 180
```

Si C est plus petit ou égal à 0, ou plus grand que 12, retour au début en 180, car alors C est incorrect.

```
210 FOR I = 1 TO C
220 INPUT "NUMERO DE LA CHAMBRE:";NC(I)
```

Pour I allant de 1 à C, on saisit le numéro de la chambre pour l'occupant NM$.

```
230 IF NC(I) 0 or NC(I) 100
     THEN GOTO 220
```

On considère que les numéros de chambre NC sont compris entre 0 et 100 à l'extérieur de ces deux valeurs, retour en 220.

```
240 INPUT "CATEGORIE";K(I)
```

On décide arbitrairement qu'il n'y a que cinq catégories : 0, 1, 2, 3, 4.

```
250 IF K(I 0 or K(I) 4 THEN GOTO 240
260 INPUT "NOMBRE DE NUITEES:";NN(I)
```

Le nombre de nuitées ne peut être inférieur à 0.

```
270 IF NN(I) 0 THEN GOTO 260
280 INPUT "Nombre d'impulsions téléphoni
   ques:";TI(I)
```

Ici encore, le nombre d'impulsions téléphoniques ne peut être inférieur à 0.

```
290 IF TI(I) 0 THEN GOTO 280
300 INPUT "SERVICES ANNEXES :"; SA(I)
```

Le coût des services annexes ne peut être négatif.

```
310 IF SA(I) 0 THEN GOTO 300
320 CC(I) = SA(I) + TI(I) * 0.6
```

On calcule le coût d'une chambre CC (une impulsion téléphonique vaut 0,60 F, chaque unité de catégorie K vaut 100 F). On peut éventuellement réviser ces tarifs.

```
330 CC(I) = CC(I) + NN(I). K(I). 100
340 NEXT I
```

On passe à la chambre suivante?

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

Affichage du nom de l'occupant.

```
360 ?"NOMBRE DE CHAMBRES:";C
```

Affichage du nombre des chambres retenues.

```
370 ?"NO: CAT: NUITS: TEL: SA: PRIX:
```

Affichage des titres des colonnes des tableaux. Répartir harmonieusement ces six rubriques sur la largeur de votre écran (40 ou 80 colonnes).

```
380 FOR I = 1 TO C
```

Pour chacune des chambres retenues on affiche :
— le numéro de la chambre (NC(I));
— le numéro de la catégorie (K(I));
— le nombre d'impulsions téléphoniques (TI(I));
— les services annexes (SA(I));
— le coût de la chambre (CC(I)).

```
390 ? NC(I);K(I);NN(I);TI(I);SA(I);CC(I)
```

La tabulation sur la ligne dépend de l'écran sur lequel vous travaillez. Il faut simplement aligner horizontalement chaque valeur avec son titre ou en-tête.

```
420 CT = CT + CC(I)
```

Cumul dans le prix total du séjour CT; on ajoute le prix de la chambre en cours de traitement au prix total du séjour.

```
430 NEXT I
440 ? : ? "COUT TOTAL : UCT
```

Affichage du coût total du séjour CT.

```
450 GOTO 150
```

Retour à la lecture et au traitement d'un autre client.

```
460 END
```

Après avoir tapé le programme, taper RUN au clavier. L'écran affiche NOM DE L'OCCUPANT : taper le nom désiré, ou FIN pour arrêter. Dans le premier cas, la machine affiche une succession de questions auxquelles il faut répondre. Le libellé des questions est contenu dans les guillemets des lignes 180 à 300.

Ensuite, le programme réaffichera vos données avec les coûts totaux correspondants.