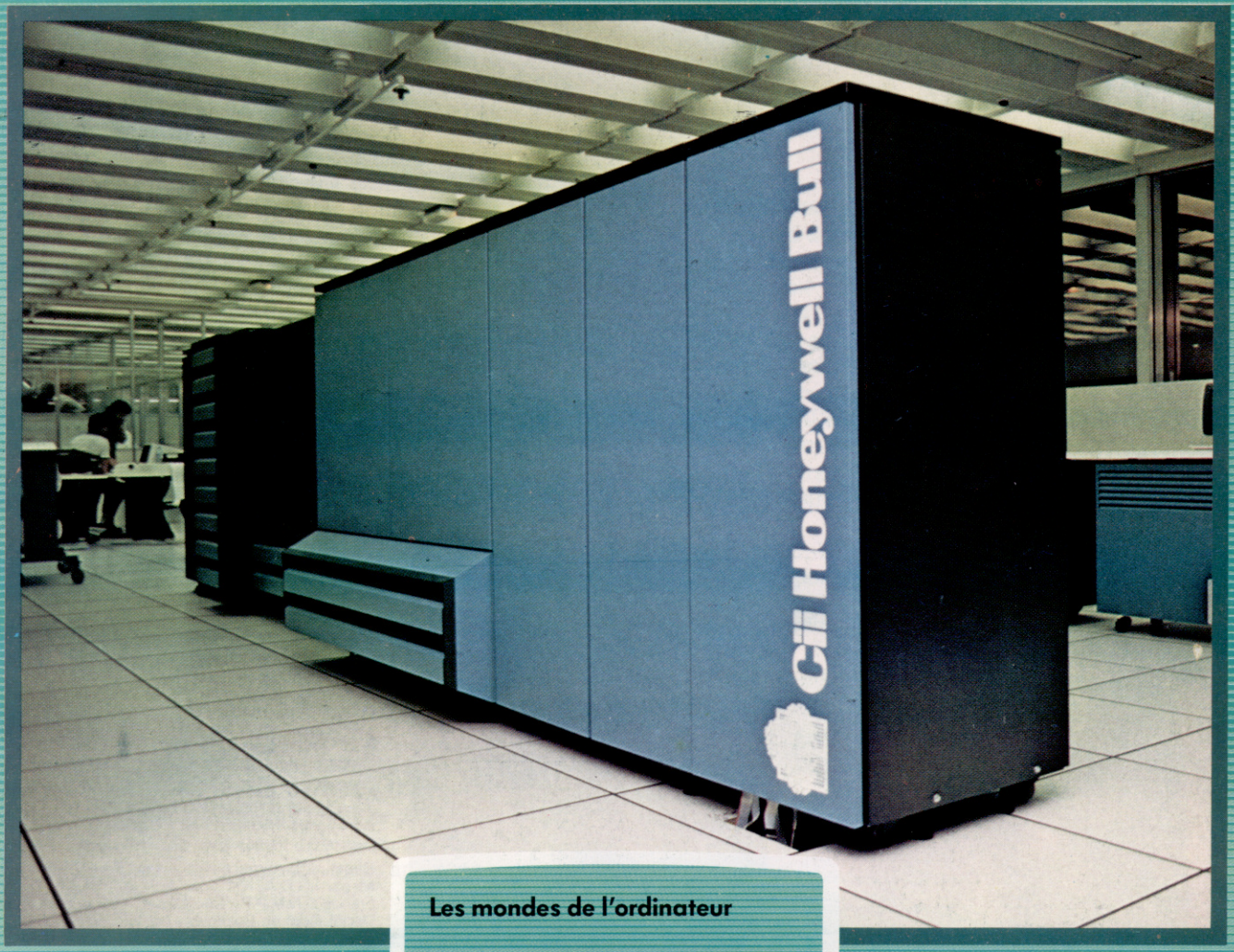
ABC INFORMATIQUE

N° 78

COURS D'INFORMATIQUE PRATIQUE ET FAMILIALE



Les mondes de l'ordinateur

Des programmes à la ferme

Des livres sur les micros

Zaxxon est partout !

EDITIONS ATLAS

M 6062-78-12,50 F 90 FB - 3,80 FS - $ 1.95

Les mondes de l'ordinateur

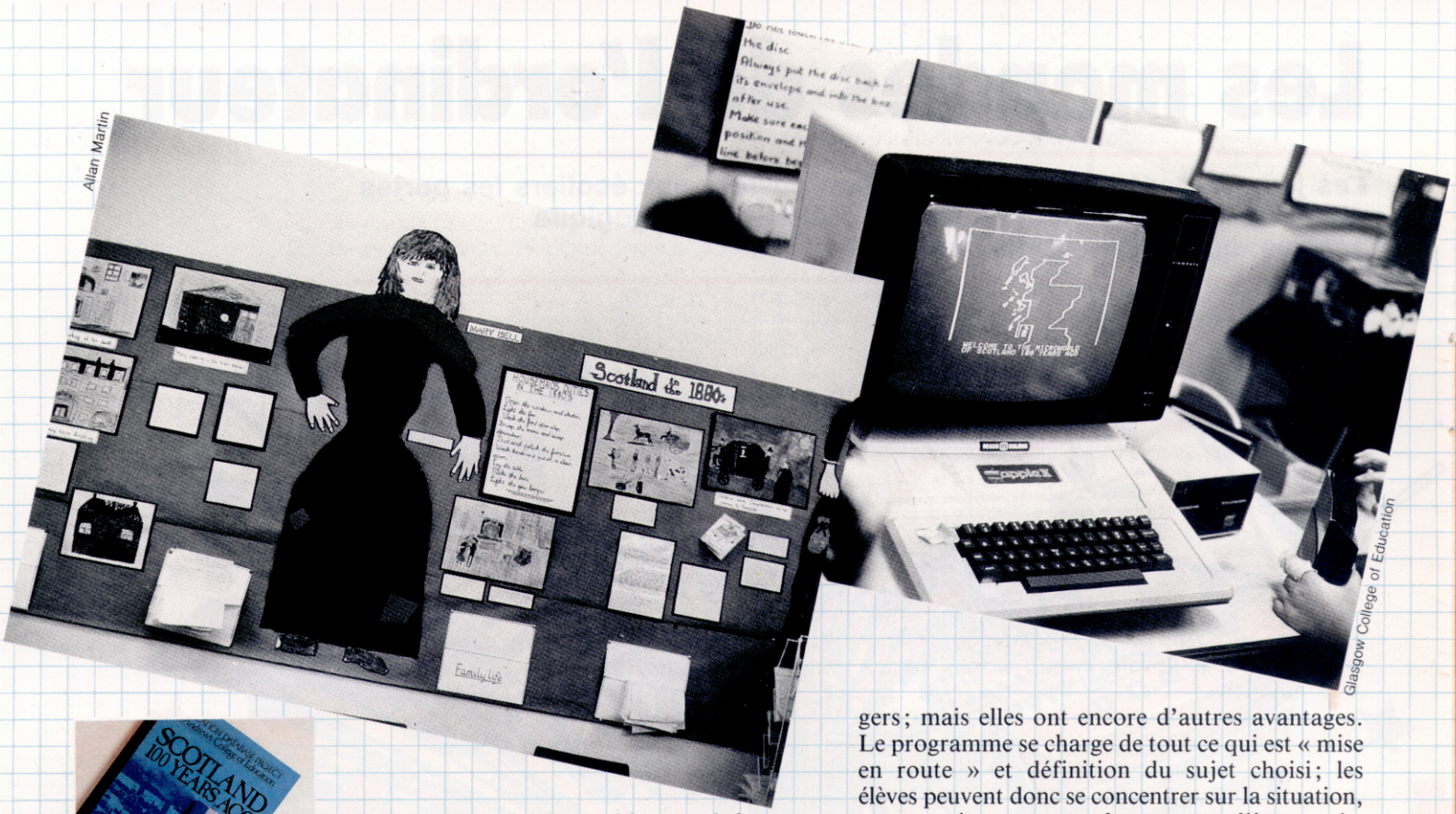
Les jeux d'aventures et de simulation ouvrent aux écoliers les portes d'univers inaccessibles; ils ont un potentiel pédagogique énorme, qu'il convient d'examiner.

Jouer : un gros travail
Les jeux d'aventures et de simulation permettent à l'enfant d'entrer dans des mondes inaccessibles, parfois même dangereux, sans jamais quitter la salle de classe. Le jeu n'est pas une perte de temps, mais un moyen très sûr d'apprendre, sans jamais s'ennuyer.
(Cl. Terry Kennett.)

On joue pour le plaisir. Tous les lycéens, et sans doute beaucoup d'enseignants, seront sûrement d'accord pour dire qu'apprendre devrait aussi être agréable. Les recherches récentes en psychologie infantile (Piaget, par exemple) ont insisté sur le caractère formateur du jeu, notamment lorsque l'enfant doit jouer un rôle. L'ordinateur a, de ce point de vue, une importance qu'on ne saurait sous-estimer.

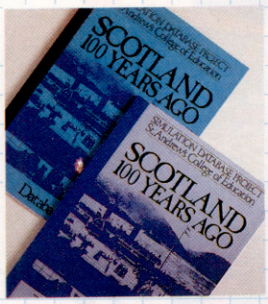
Il y eut d'abord les jeux de café de type Space Invaders; puis les jeux d'aventures et de simulation firent peu à peu leur apparition. D'un côté action pure, de l'autre réflexion et stratégie : mais cette distinction si pratique est de plus en plus difficile à maintenir, bien que, évidemment, chaque logique se rattache plus ou moins nettement à l'un des deux genres.

Les jeux de type « combat dans l'espace », si populaires qu'ils soient, ne semblent pas avoir beaucoup influencé les pédagogues... L'opinion des psychologues à leur sujet n'est pas très favorable. Il est vrai pourtant qu'ils exigent une grande rapidité de réflexes, et une excellente coordination psychomotrice, ce qui implique qu'ils pourraient avoir une certaine utilité dans le traitement de certaines déficiences mentales, ou de troubles psychologiques.



Allan Martin

Glasgow College of Education



Voyage dans le temps
Les responsables du St. Andrews' College of Education de Glasgow ont mis au point une « base de données de simulation » grâce à laquelle l'ordinateur vient prêter main-forte aux autres activités pédagogiques, sans jamais les supplanter. Le logiciel est en trois parties — un jeu d'aventures, un ensemble de textes fournissant des informations supplémentaires et une autre base de données contenant d'autres références. La classe est divisée en groupes, et chacun d'eux prend l'identité d'un personnage particulier qui entreprend un long voyage dans l'Écosse du XIXe siècle.
(Cl. Ian McKinnell.)

Certains didacticiels ont bien tenté de recourir à des scénarios de ce genre : par exemple, dans Maths Invaders, il faut lutter contre les envahisseurs venus de l'espace en « tirant » sur eux les réponses correctes à des problèmes d'arithmétique. Mais de tels programmes ont un double défaut : ils sont infiniment moins passionnants que les originaux, et il n'est pas sûr qu'ils aient une véritable valeur pédagogique.

Les jeux d'aventures et de simulation semblent avoir un potentiel pédagogique bien plus prometteur. L'enfant a ainsi accès à des univers imaginaires (aventures) ou réalistes (simulation) qui lui sont inaccessibles en temps normal. Son imagination peut se déployer librement par rapport à une sorte de « structure d'accueil » à la fois souple et bien dessinée.

Des avantages multiples

Ballooning est un programme de Hill McGibbon destiné au Spectrum et au Commodore 64. C'est un jeu de simulation typique : les enfants se voient offrir l'occasion d'apprendre quelque chose en se familiarisant avec une technologie peu connue. En bas de l'écran, le programme affiche un altimètre, un thermomètre et un indicateur de combustible, ainsi que des renseignements sur la façon dont le ballon monte et descend. Toute manipulation de la valve de gaz ou du brûleur a des résultats immédiats. Lorsque l'engin est redescendu à une certaine altitude, le terrain d'atterrissage apparaît ; il sera parfois nécessaire de se poser pour se réapprovisionner en gaz, sans bien sûr mettre en danger la vie de l'équipage ! Les simulations sont peu coûteuses et sans dangers ; mais elles ont encore d'autres avantages. Le programme se charge de tout ce qui est « mise en route » et définition du sujet choisi ; les élèves peuvent donc se concentrer sur la situation, sur ce qui se passe, et les moyens d'intervenir. Des projets assez ambitieux et détaillés peuvent ainsi être menés à bien. Le temps n'est plus un obstacle, et des expériences de laboratoire qui demandent plusieurs jours peuvent être reproduites en quelques secondes.

Un modèle informatisé permet aussi une simplification des problèmes trop complexes, tandis que les graphismes rendent plus faciles la compréhension. De surcroît, il est possible de répéter indéfiniment tout élément particulier, ou de modifier les conditions générales. Les enfants sont donc tout naturellement amenés à faire l'expérience, et, comme ils contrôlent en fait le déroulement du programme, ils en comprennent mieux les tenants et les aboutissants. Ils ont sur les questions une vision globale et non partielle.

Naturellement, les simulations ne constituent pas une panacée. Un ordinateur ne pourra jamais garder en mémoire qu'un nombre limité de variables, et il donnera de son sujet une vision par trop simplifiée. Il lui est possible de recréer des situations réalistes, mais ce qui est un point fort peut aussi devenir une faiblesse. Car c'est surtout l'imprévu qui nous stimule et nous fait réaliser que tout ne se passe pas comme on le croit d'habitude. Un programme comme Ballooning, pour être utilisé avec profit, devra sans doute être intégré dans un cadre plus vaste — il peut très bien servir à illustrer un moment de l'histoire de l'aviation. Les jeux de simulation ne sauraient se substituer à la chose étudiée : c'est un aspect qu'on néglige trop souvent. Leur énorme intérêt vient du fait qu'ils stimulent la curiosité sans fournir de réponses toutes faites, ni se restreindre à une simple « illustration ». Il existe par exemple un

programme scientifique tournant sur Apple qui simule les différentes façons de procéder à des montages simples de piles électriques et d'ampoules. Il a sans doute été écrit pour les lycées qui ont assez d'argent pour s'acheter un ordinateur, mais qui, pour des raisons obscures, ne peuvent se permettre de faire l'acquisition de piles et d'ampoules...

L'imaginaire à portée de la main

Tout le monde, enfants comme adultes, aime à se plonger dans l'imaginaire. Des livres comme *The Hobbit, Le Seigneur des annneaux, 2001 l'Odyssée de l'espace* ont connu récemment un succès qui ne s'explique pas autrement. Les pédagogues ont vu très vite le potentiel énorme de l'ordinateur à ce sujet, encore renforcé par ses possibilités sonores et graphiques. Des jeux de rôles du type « Donjons et Dragons » se prêtent parfaitement à une mise en œuvre informatisée et répondent parfaitement aux attentes de ceux qui veulent promouvoir des méthodes d'éducation interactives.

Il semblerait, par exemple, qu'il n'y ait pas de meilleur moyen d'apprendre l'histoire que de faire voyager dans le temps et donner à résoudre un jeu d'aventures. L'enfant aurait bien sûr besoin de renseignements, qui lui seraient fournis par une base de données très facilement accessible à partir du clavier. Un projet de ce type a par exemple été mis au point en Grande-Bretagne par Allan Martin, du St. Andrews College of Education de Glasgow. Il s'agissait « d'étudier la façon dont les ordinateurs peuvent contribuer réellement au travail à l'école primaire, en créant un objet pédagogique spécifique, la base de données de simulation ». L'enfant entre dans un monde historiquement situé, traverse certains lieux, fait face à certaines situations. Des textes donnent des précisions utiles sur tel ou tel aspect de ce monde, et sont accessibles à tout moment. Le programme lui-même s'insère dans un projet bien plus vaste; l'acquisition des connaissances se fait pour l'essentiel sans recours à l'ordinateur, qui n'est jamais que l'un des nombreux outils pédagogiques disponibles.

Une seconde base de données renferme des références à utiliser par la suite : livres, films, émissions télévisées. Le logiciel se veut intégré à l'ensemble des activités scolaires. Il est ainsi accompagné d'une carte, d'images, de lettres. Rédigé en LOGO, il peut être adapté à chaque cas particulier : on peut y ajouter ou y supprimer des données, des zones géographiques entières, pour satisfaire tel ou tel objectif précis propre à une classe.

Le décor est installé en Écosse aux alentours de 1880. Les élèves d'une classe se répartissent en six groupes, dont chacun joue le rôle d'un personnage fictif (mais très réaliste) qui représente à chaque fois une couche sociale bien définie, de l'aristocrate à la femme de chambre. Chaque groupe a pour tâche de voyager à travers l'Écosse. Il faut préparer le périple et rejoindre la région, pour se heurter aussitôt aux premiers incidents. Chacun joue à son tour, et les enfants, en se servant des bases de données et des références fournies par le professeur, se livrent à des enquêtes. Il leur faut tracer des cartes, rédiger des articles de journaux, ou tenir un journal de bord. Tout cela mène ultérieurement à des pièces de théâtre, à des « interviews » enregistrées avec les personnages, à des visites aux lieux traversés, à des conférences. Le « travail » fut l'occasion de discussions passionnées, de peintures et de dessins; les élèves comme les enseignants trouvèrent l'expérience passionnante.

Ce n'est là, bien sûr, qu'un exemple parmi d'autres, mais il montre bien les ressources énormes que peut révéler un jeu sur ordinateur d'un point de vue pédagogique. Le même groupe d'éducateurs prépare, par exemple, un ensemble sur la Palestine au temps du Christ, et un autre sur les rivières qui se veut à la fois scientifique et centré sur les questions d'environnement. Le jeu que nous avons décrit fait simplement usage, de façon créative, des possibilités de l'informatique. L'ordinateur cesse d'être un outil même très performant (ainsi pour ce qui est du traitement de texte) pour devenir un véritable moyen d'expression et de communication, grâce auquel tout enfant peut, de façon très libre, explorer un sujet donné et en tirer un profit certain.

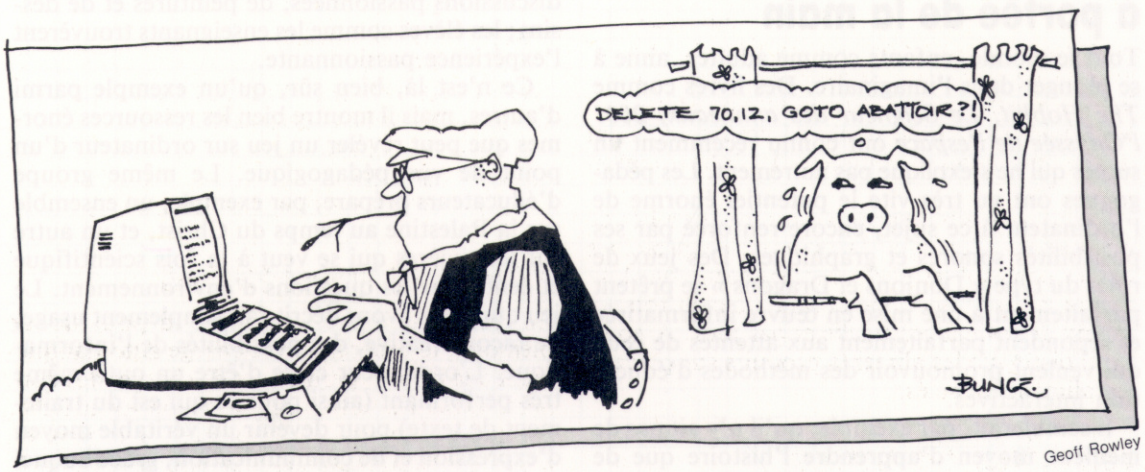
L'homme, ce joueur

Bien des enseignants et des parents s'inquiètent souvent de voir les enfants « jouer au lieu d'apprendre ». Ils ont tort. Des enfants peuvent très bien suivre les évolutions d'une tortue-robot sur un plancher, ce n'est pas une perte de temps; même si l'exercice reste très informel, ils apprendront quelque chose. Le jeu est un excellent outil pédagogique, peut-être même le seul qui vaille. De telles idées ont par exemple inspiré Piaget et Papert.
Dès 1944, J. Huizinga — professeur à l'université de Leyde en Hollande — avait écrit un ouvrage intitulé *Homo Ludens* (« L'Homme joueur »), dans lequel il montrait que le jeu n'est pas une activité gratuite, mais qu'il a ses propres valeurs. Ses arguments étaient les suivants :

- Le jeu a un sens, c'est une activité importante.
- Tous les jeux veulent dire quelque chose.
- Le jeu est à la fois instinctif et réfléchi. Les enfants jouent d'abord pour connaître le monde qui les entoure, mais aussi parce qu'ils y prennent plaisir.
- La civilisation moderne souffre de la trop grande structuration de l'activité ludique.
- Elle est mieux guidée mais beaucoup moins amusante.

A la ferme

L'agriculture utilise des micros depuis plusieurs années déjà. L'ensemble des programmes Farmfax, sur cartouches destinées au Dragon, couvre les principaux aspects modernes de l'agriculture.



Geoff Rowley

Si l'on considère que l'usage de l'ordinateur s'est répandu dans tous les secteurs de l'économie, la présentation sur le marché d'une suite de programmes de gestion agricole n'a rien en soi de remarquable. Cependant, l'ordinateur auquel ce progiciel est destiné, le Dragon, constitue une surprise, même si ce micro connaît, selon son fabriquant, un intérêt chez les agriculteurs.

Farmfax est dû à Geoffrey Paterson qui est lui-même fermier. Cette suite de programmes est divisée selon des travaux agricoles spécifiques. Les directives de base ont été traduites en langage d'assemblage dans l'optique de rendre les programmes accessibles aux non-informaticiens. Ils sont sur cartouches ROM prêtes à l'emploi aussitôt enfichées. Certaines d'entre elles disposent d'une RAM supplémentaire permettant le stockage rapide de données, même si le progiciel conseille aux utilisateurs de garder des bandes de sauvegarde.

On peut citer l'exemple d'une exploitation agricole où les réponses précises apportées par l'informatique furent, par exemple, le taux de vêlage, ou calcul du temps pris pour le vêlage. L'intérêt pour le fermier est, dans ce cas, de pouvoir calculer les rendements laitiers. En effet, explique-t-on « si une vache ne vêle pas au moins une fois par an, vous perdez de l'argent (du fait d'un moindre rendement), environ 30 F par jour en moyenne, au-delà d'une année sans vêlage. Sans ordinateur, il est difficile de tenir l'historique des vêlages, leur espacement ayant souvent été jusqu'à atteindre 390 jours au pire ».

La pleine utilisation d'un cheptel a pu être atteinte par l'informatique, avec 366 jours par bête entre deux vêlages. C'est pratiquement le mieux que l'on puisse faire.

Enregistrements par tête de bétail

Pour donner une indication de l'étendue d'un programme, les enregistrements concernant chaque vache en particulier peuvent couvrir une population totale de 240 têtes, sur une seule cartouche. Une version moins coûteuse de Farmfax effectue le stockage sur bande magnétique. Les versions procèdent de la même manière, en recopiant en mémoire centrale les données, avant qu'elles ne soient à nouveau sauvegardées (SAVE). Il n'y a donc pas de différence préjudiciable entre ces deux versions.

Même si le temps de sauvegarde sur bande est relativement lent, cela n'a rien à voir avec l'ancien mode de calcul sans ordinateur. Le fermier devait apprécier à partir de quel moment une vache cessait d'être rentable en comparant son coût en nourriture et ce qu'elle rapporte en lait. Dans l'exemple précédent, on rapporte qu'avant l'informatisation, le fermier devait payer l'organisme central de collecte du lait pour connaître ses rendements. Cet organisme lui prenait environ 500 F par mois pour tenir une gestion informatisée du rendement de ses vaches. Le fermier lui envoyait les éléments à chaque fin de mois pour recevoir, trois semaines après, les états résultant du traitement central. C'était trop tard pour remédier dans ce laps de temps à une situation éventuellement dégradée. Le fermier avait longtemps hésité avant d'acquérir son propre ordinateur, car le coût de la plupart des systèmes le retenait : il ne pouvait mettre la somme normale pour ce type de matériel, de l'ordre de 30 000 F.

Le Dragon était dans ses moyens, avec l'imprimante et le programme Farmfax, pour moins de 8 000 F. Il est rapidement rentré dans ses frais. En outre, il peut accéder aux informations en temps réel.

A la fin de chaque mois, il passe environ une heure à réorganiser ses informations. Il commence par exécuter le programme de gestion laitière, qui lui donne les résultats financiers, la marge bénéficiaire du lait par rapport aux dépenses fourragères et autres, son revenu brut, le nombre de litres de fuel nécessaires pour aboutir à un litre de lait, la production totale de lait, et d'autres paramètres appropriés. Il exécute ensuite le programme de calcul du rendement en lait de son troupeau. Cet utilisateur estime que le progiciel est utile dans la mesure où il lui donne une analyse complète de l'alimentation fourragère.

En fournissant au programme les quantités de mégajoules d'énergie, la quantité de protéines brutes comestibles et de matière sèche par kilo de nourriture, il peut déterminer le coût total et le prix de revient par vache et par jour. A noter qu'il distingue les aliments produits sur place et les aliments achetés.

Farmfax en action

En fournissant au programme les rendements journaliers en lait de chaque vache, on obtient des récapitulatifs très précis. Lorsqu'un des facteurs est modifié, l'ordinateur recalcule instantanément l'ensemble des états. Le temps gagné est égal aux deux tiers du temps nécessaire pour effectuer ces calculs à la main. Cette gestion permet également au fermier de connaître sa situation bancaire exacte (découverts éventuellement acceptés par la banque, etc.).

Chaque module de Farmfax est orienté vers l'utilisateur par sa simplicité d'emploi. Des brochures très simples les accompagnent; elles ne nécessitent que quelques minutes d'étude. Chaque module s'ouvre sur un menu très accessible. Le programme de sélection bovine, qui permet au fermier de prendre des décisions concernant les vaches dont le rendement est passé en dessous d'un certain seuil, commence par demander :

PAGE 1 – DÉTAILS DES PRIX
PAGE 2 – ACHAT/VENTE

Avec la page 1, le fermier doit d'abord donner le mois et l'année concernés, ensuite le taux d'intérêt applicable, le prix du lait, le rendement en litres par jour et par vache, et le coût mensuel. Ces données sont alors aussitôt transférées à toutes les autres données en entrée, ce qui permet de gagner du temps de frappe. Ces saisies peuvent néanmoins être modifiées par intervention de l'utilisateur lorsque ce dernier estime, par exemple, que les taux d'intérêt sont susceptibles de croître au cours des mois à venir, ce qui est plus souvent le cas que l'inverse, malheureusement.

En page 2, le fermier se voit demander s'il vend ou achète du bétail, le nombre concerné par rapport à l'ensemble du troupeau, le prix auquel il s'attend, et le mois de la transaction. L'ordinateur compare ensuite les chiffres et indique le meilleur parti pour la vente par un astérisque. La touche « Break » renvoie l'utilisateur à la table des matières (l'index), pour modifier éventuellement des données.

Les autres programmes fonctionnent de manière similaire, même si le menu devient plus complexe avec les programmes plus sophistiqués. Par exemple, le programme déterminant les rations journalières propose :

PAGE 1 – TABLE DES ALIMENTS
PAGE 2 – MINÉRAUX
PAGE 3 – ÉLÉMENTS DE COMPARAISON
PAGE 4 – DESCRIPTION DES GROUPES
PAGE 5 – COMPOSITION
PAGE 6 – QUANTITÉS
PAGE 7 – SAUVEGARDE/CHARGEMENT DE DONNÉES

Pour des spécialistes

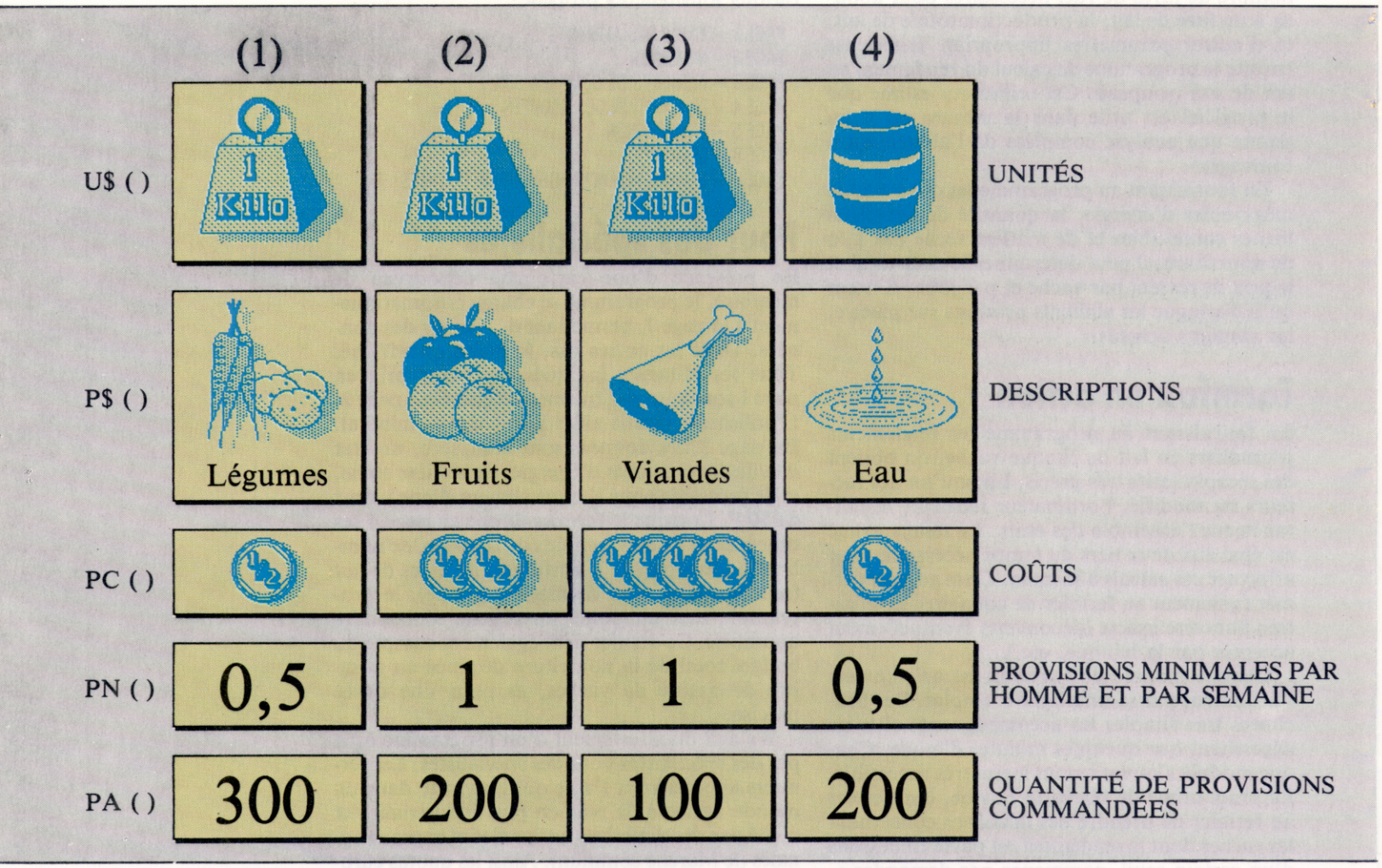
En présence d'une cartouche dépourvue de mémoire, le programme se charge automatiquement à la page 7, permettant de charger des données. Dans les autres cas, le menu est affiché. Tous les aliments indiqués avec leur prix en page 1 sont automatiquement reportés en page 2. L'utilisateur donne alors leur contenu minéral. En page 3, les données sont analysées; un état détaille les quantités d'énergie, de matière sèche, et de protéines pour chaque aliment d'une valeur de 10 F. Lorsque le fermier donne en page 4 ses objectifs en termes de poids de la vache, de nombre de jours où elle « portera », de litres de lait (rendement et taux de matière grasse), le programme met au point un régime alimentaire approprié. Ce dernier tient également compte du budget total de la nourriture destinée au nombre déterminé de vaches, et pour une durée précise.

Il s'agit manifestement d'un programme écrit par des spécialistes pour des spécialistes. Les fermiers apprécieront l'aide qu'il apporte dans un monde agricole de plus en plus industrialisé et cerné par des directives nationales et supranationales (le Marché commun). Ainsi les quotas européens obligent-ils les agriculteurs à gérer leur exploitation au plus près. Et c'est en quoi l'ordinateur est utile. Dans tous les cas, si cela ne fait pas gagner de l'argent aux fermiers, cela leur évite d'en gaspiller! N'est-ce pas un des arguments auquel ils sont le plus sensible?

Farmfax Ensemble : Pour le Dragon.
Noms des programmes : AchatVente, SélectionBovine, AchatEngrais, LiquiditésEngrais, PlanificationLiquidités, GestionLaitière, PrévisionsLaitières, RotationsLaitières, RelevésChamps, RelevésIndividuels.
Distributeur : Farmfax Computer Systems.
Format : Cartouche.

Provisions de bouche

Voici le second module de notre jeu de simulation. Il nous permettra d'accumuler des provisions pour l'équipage, en vue de la traversée de l'Atlantique.



Provisions de bouche
Le joueur peut faire l'acquisition de quatre catégories différentes de provisions : légumes, fruits, viande, eau. Leur description est contenue dans le tableau P$( ), tandis que U$( ) gère les unités correspondantes. Pour faire ses choix, il aura bien sûr besoin d'informations : prix de chaque marchandise, ration hebdomadaire minimale pour chaque membre d'équipage. Les données nécessaires sont contenues dans les tableaux PC( ) et PN( ). (Cl. Ian McKinnell.)

Jusqu'ici, le joueur n'a fait encore que recruter un équipage. Il se voit alors confronté au message APPUYEZ SUR UNE TOUCHE POUR CONTINUER. Passons maintenant à la seconde phase des événements : le voyage prendra environ huit semaines, et il faut donc acheter de la nourriture en quantités suffisantes, et se procurer de l'eau.

Le programme calcule les rations sur huit semaines et s'assure que les réserves sont suffisantes. Si ce n'est pas le cas, il rappelle que les hommes peuvent avoir faim ou soif, et offre au joueur la possibilité de recommencer les opérations de ravitaillement. Il se peut que les fonds nécessaires soient alors insuffisants ; dans ce cas, il faudra faire une troisième tentative.

Tout comme lors de l'embauche de l'équipage, nous aurons besoin de créer un certain nombre de tableaux pour l'étape des approvisionnements :

```
19 REM ***** PROVISIONS *****
20  DIMU$(4):U$(1)="KILO:U$(2)="KILO":
U$(3)="KILO":U$(4)="BARRIQUE"
21  DIMP$(4):P$(1)="LEG":P$(2)="FRUITS":
P$(3)="VIANDE":P$(4)="EAU"
22 DIMPA(4)
23  DIMPC(4):PC(1)=.5:PC(2)=1:PC(3)=2:PC(4)=.5
24  DIMPN(4):PN(1)=2:PN(2)=1:PN(3)=1:PN(4)=.5
```

Toutes les provisions sont calculées en kilos pour les légumes, les fruits et la viande, tonneaux pour l'eau. La ligne 20 crée un tableau, avec quatre éléments pour gérer le tout. Il est préférable de conserver ces valeurs dans un tableau plutôt que de les imprimer continuellement, cela économise de l'espace mémoire. Il est de surcroît plus facile d'insérer d'autres articles — par exemple des sacs de farine.

Les quatre éléments du tableau P$( ), ligne 21, constituent la description des quatre catégories

de provisions. Un tableau PA(), ligne 22, gère le poids de chacune d'elles. Un kilo de légumes vaut une demi-pièce d'or, un kilo de fruits une pièce, un kilo de viande deux, et un tonneau d'eau douce une demi-pièce. Ces informations sont stockées ligne 23. L'ordre des provisions n'est pas fixé, mais défini par défaut. Là encore, rien de plus simple que d'ajouter tel ou tel article.

Chaque membre de l'équipage, s'il veut rester en bonne santé, doit avoir droit à une certaine quantité de nourriture. Ces exigences sont fixées par le tableau PN(), ligne 24. Par exemple, chaque matelot a besoin de deux kilos de légumes pour huit jours, et la valeur du premier élément du tableau, PN1, est donc fixée à 2.

La ligne 500 du premier module renvoie à un sous-programme en ligne 1000 lors de l'embauche de l'équipage. De la même façon, pour acheter nos provisions, nous aurons besoin d'un GOSUB après la ligne 500 :

```
550 GOSUB 2000 : REM ACHAT DES PROVISIONS
```

Le programme affiche alors une page de texte, qui explique la marche à suivre pour acheter le ravitaillement. CN est une variable qui gère le nombre d'hommes d'équipage. Elle est présentée ligne 2040 pour que le joueur sache quelles sont les exigences à respecter.

La boucle principale de notre programme est parcourue quatre fois (une fois par catégorie de provisions). T est augmentée de 1 à chaque fois (ligne 2050). L'ordinateur affiche les quantités indispensables à chacun, ce qui est calculé à partir des éléments du tableau PN().

Comme nous l'avons vu, l'eau est calculée en tonneaux et la nourriture en kilos. Pour cela, il faut examiner le tableau U$(), défini ligne 20. S'il y a plus d'un kilo, la ligne 2080 se charge d'ajouter un S à « kilo », la ligne 2075 imprimant un espace dans le cas contraire. Les trois lignes suivantes donnent les précisions nécessaires.

Calculs des rations

Les lignes 2100 et 2110 demandent combien d'unités de chaque type de provision le joueur veut acheter. Les réponses sont intégrées dans un tableau ligne 2140. Il se peut qu'il ne veuille rien acheter; cette possibilité est prévue ligne 2160. Ensuite, le programme fait ses calculs sur une semaine, à partir des informations contenues dans le tableau PA, et se sert notamment de la formule CN*8*PN(T). CN est le nombre d'hommes d'équipages, PN(T) représente telle quantité nécessaire de provisions, déterminés par T (ligne 2170).

Si les estimations sont insuffisantes, le programme affiche à l'écran un message d'avertissement. Il vérifie également s'il parcourt la boucle principale pour la quatrième fois, et, si oui, affiche SOIF plutôt que FAIM.

Le joueur étant averti qu'il risque des ennuis, la ligne 2230 lui demande DÉSIREZ-VOUS RECOMMENCER? (O/N). Si vous désirez modifier vos estimations, la valeur correspondante du tableau PA() est remise à zéro, et le programme revient au choix de tel ou tel type de nourriture. Il suffit pour cela qu'il déduise 1 du compteur de boucle, ligne 2250 par T=T-1.

Si, en revanche, vous estimez que tout est bien, vous aurez droit à l'affichage de l'argent qui vous reste et des provisions déjà achetées. Il se peut que cela pose des problèmes (il ne reste plus assez d'argent pour les salaires, par exemple). La ligne 2260 vérifie si c'est le cas ou non. Si oui, les lignes 2270-2290 en avertissent le joueur, et lui demandent de procéder à de nouvelles estimations. Si cette fois les achats sont satisfaisants, leur montant est soustrait de MO (somme d'argent totale dont vous disposez).

La liste des provisions acquises est affichée ligne 2410, par l'intermédiaire d'une boucle, contrôlée par la variable TT, et qui se sert des variables U$(), PA() et P$(). La ligne 2480 donne le nombre total des pièces d'or encore accessibles. Le joueur doit désormais APPUYER SUR UNE TOUCHE POUR CONTINUER. Il repasse alors en programme principal.

Module Deux

```
2000 REM **** PROVISIONS ****
2005 PRINTCHR$(147):REM EFFACER ECRAN
2010 S$="    PHASE 2 - LES PROVISIONS "
2015 GOSUB9100
2020 S$=" ......  --- -------- "
2025 GOSUB9100
2030 GOSUB9200:PRINT
2040 PRINT "VOTRE EQUIPAGE EST DE "; CN;"HOMMES"
2045 GOSUB9200:GOSUB9200
2050 FORT=1TO4
2055 PRINT
2060 PRINT "CHACUN AURA BESOIN "
2070 PRINT "D'AU MOINS ";PN(T);" ";U$(T);
2075 IFPN(T)=1THENPRINT"   ";:GOTO2085
2080 PRINT"S";
2085 PRINT"   ";P$(T)
2086 PRINT"A";PC(T); "PIECES D'OR PAR "; U$(T).
2090 PRINT"PAR SEMAINE DE VOYAGE".
2095 GOSUB9200:PRINT:GOSUB9200
2100 PRINT"COMBIEN DE"; U$T ; "S DE" ; P$(T)
2110 S$="VOULEZ-VOUS ACHETER ?":GOSUB9100
2120 PRINT
2130 INPUTP$
2140 PA(T)=VAL(P$):GOSUB9200
2150 IFPA(T)>((CN*8*PN(T))-1) THEN2260
2160 IFPA(T)=0THENPRINT"SI VOUS N'ACHETEZ PAS DE":GOTO
2180
2170 PRINT"SI VOUS N'ACHETEZ QUE ";PA(T);U$(T)
2175 PRINT"DE S"
2180 PRINTP$(T);",":GOSUB9200
2190 PRINT"VOUS RISQUEZ D'AVOIR ";
2200 S$="FAIM"
2210 IFT=4THENS$="SOIF"
2220 PRINTS$;"!":GOSUB9200
2230 S$= "VOULEZ-VOUS RECOMMENCER (O/N)":GOSUB9100
2240 INPUTP$:P$=LEFT$(P$,1)
2242 IF P$  "o" AND P$<>"N"THEN 2230
2245 IF P$="N"THEN2400
2250 PA(T)=0:T=T-1:GOTO2410
2260 IFPA(T)*PC(T)>MOTHEN2270
2265 GOTO2400
2270 S$="VOUS N'AVEZ PAS ASSEZ D'ARGENT POUR":GOSUB9100
2280 PRINTPA(T)
2290 PRINTU$(T);" SE DE ";P$(T):GOSUB9200
2300 S$="RECOMMENCEZ":GOSUB9100:PA(T)=0:T=T-1:GOTO2410
2400 MO=MO-(PA(T)*PC(T))
2410 PRINT:S$="VOS PROVISIONS:":GOSUB9100
2412 GOSUB9200
2415 FORTT=1TO4
2420 PRINTPA(TT);U$(TT);
2430 IFPA(TT)=1THENPRINT"  DE ";:GOTO2440
2435 PRINT "S DE ";
2440 PRINTP$(TT)
2450 GOSUB9200
2460 NEXT
2480 PRINT "IL VOUS RESTE ";MO;" PIECES D'OR":GOSUB9100:
GOSUB9200
2510 PRINT: S$=K$:GOSUB9100:PRINT:    GOSUB9200
2520 GETP$:IFP$=""THEN2520
2999 RETURN
```

Du cœur au ventre
Ce sous-programme gère l'achat des provisions pour le voyage et devra venir s'adjoindre au premier module.
Il comporte une boucle principale parcourue quatre fois — une pour chaque catégorie de nourriture.

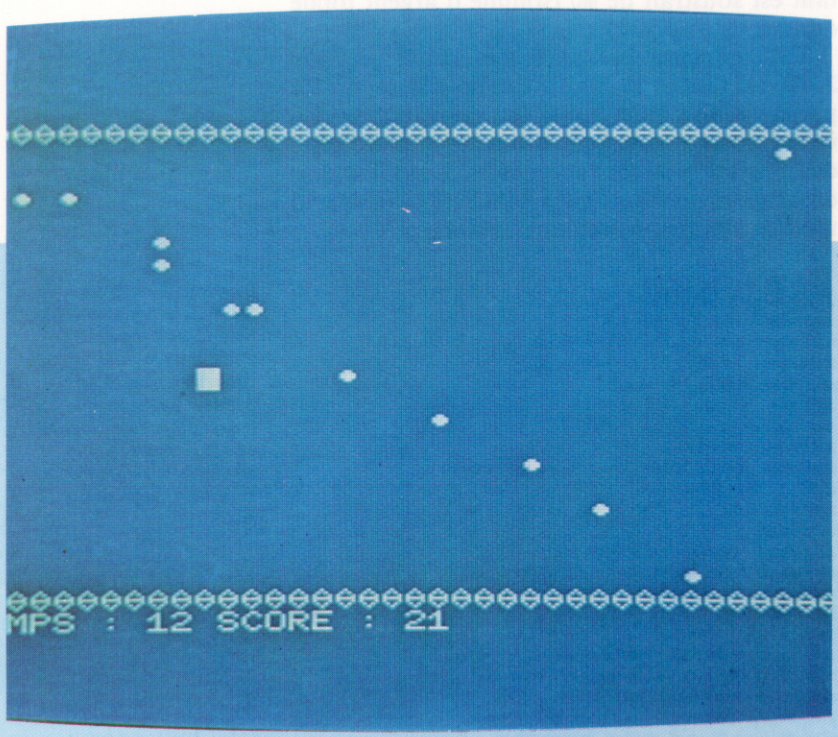
Variantes de basic

Spectrum :
Procédez aux modifications suivantes :
2005 CLS
2240 INPUT P$: LET P$ = P$ (1 TO 1)

BBC Micro
Procédez aux modifications suivantes :
2005 CLS

Ramasse-miettes

Vous avez peut-être déjà rencontré ce titre de jeu. Mais attention, le programme que nous vous proposons ici est uniquement destiné aux ordinateurs ayant la norme MSX.



Voici une curieuse façon d'utiliser un ordinateur : vous devez vous efforcer de ramasser le plus vite possible les miettes qui jonchent la nappe. Vous disposez de 30 secondes pour un nettoyage complet. Les miettes sont représentées par des points blancs. Les touches du curseur vous permettent de contrôler votre ramasse-miettes.

```
10 REM ********************
20 REM * RAMASSE-MIETTES *
30 REM ********************
40 KEY OFF
50 GOSUB 260
60 LOCATE 0,23,0
70 PRINT "TEMPS :";INT(Z);"SCORE :";S;
80 IF Z<1 THEN 580
90 K=STICK(0)
100 D1=(K=7)-(K=3)
110 D2=(K=1)-(K=5)
120 IF D1<>0 THEN DX=D1:DY=0
130 IF D2<>0 THEN DY=D2:DX=0
140 XP=PX+DX
150 YP=PY+DY
160 CR=VPEEK(XP+YP*40)
170 IF CR=J OR CR=B THEN XP=PX:YP=PY
180 IF CR=M THEN S=S+1:BEEP:X=X+1
190 VPOKE PX+PY*40,N
200 VPOKE XP+YP*40,J
210 PX=XP
220 PY=YP
230 Z=Z-.1
240 IF X=NM THEN 720
250 GOTO 60
260 SCREEN 0,0
270 COLOR 15,4,5
280 DEFINT A-Y
290 B=233
300 M=249
310 N=32
320 S=0
330 NM=10
340 X=0
350 FOR PX=0 TO 39
360 VPOKE PX+40,B
370 VPOKE 880+PX,B
380 NEXT PX
390 FOR PY=2 TO 21
400 VPOKE PY*40,B
410 VPOKE PY*40+39,B
420 NEXT PY
430 FOR I=1 TO NM
440 PX=INT(RND(-TIME)*37)+1
450 PY=INT(RND(-TIME)*21)+2
460 IF VPEEK(PX+PY*40)<>32 THEN 440
470 VPOKE PX+PY*40,M
480 NEXT I
490 PX=INT(RND(-TIME)*37)+1
500 PY=INT(RND(-TIME)*21)+2
510 IF VPEEK(PX+PY*40)<>32 THEN 490
520 VPOKE PX+PY*40,J
530 Z=30
540 DX=0
550 DY=0
560 J=219
570 RETURN
580 FOR I=1 TO 500
590 NEXT I
600 IF INKEY$<>"" THEN 600
610 IF S>R THEN R=S
620 LOCATE 13,10
630 PRINT "RECORD :";R;
640 LOCATE 13,16
650 PRINT "UNE AUTRE ?";
660 D$=INKEY$
670 IF D$="" THEN 660
680 IF D$<>"N" AND D$<>"n" THEN 50
690 CLS
700 LOCATE 0,0,1
710 END
720 NM=NM+1
730 VPOKE XP+YP*40,N
740 GOSUB 340
750 GOTO 60
```

Un périphérique Acorn

L'Acorn Electron a eu un succès limité en raison de son faible potentiel d'extension dû au manque d'interfaces. Le module d'extension Plus 3 lui donne maintenant une image plus sérieuse.

Une des principales critiques formulées à l'égard de l'Electron concernait le manque d'interfaces. Le fait que l'Electron ne possédait qu'une paire de prises moniteur, un jack HF et une interface cassette (pas même de port de manche à balai) signifiait que le potentiel d'extension de la machine était sévèrement limité. Acorn a toujours promis de proposer des modules d'extension, mais l'arrivée de ces dispositifs complémentaires s'est fait attendre.

Le premier module d'extension fut le Plus 1 qui contribua à éliminer des critiques. Mais bien qu'il ait fourni de nombreuses interfaces non disponibles sur l'Electron d'origine, le Plus 1 ne comprenait pas d'interface d'unité de disquette qui aurait permis d'utiliser un dispositif de stockage de masse d'accès facile.

Un plus

La situation s'est améliorée avec l'arrivée du Plus 3, un système d'unité de disquette autonome. Le module prend la forme d'un L, le contrôleur de disquette est connecté à l'arrière de l'ordinateur, et l'unité de disquette prend place sur le côté droit — cette unité couvre malheureusement la prise d'alimentation de l'Electron. Comme le Plus 1, le Plus 3 se glisse dans le connecteur plat situé à l'arrière de l'Electron, et pour interdire tout mouvement de l'unité qui pourrait endommager le connecteur plat, deux vis fixent le Plus 3 solidement à la base. Dès qu'ils sont assemblés, l'Electron et le Plus 3 forment une unité robuste, difficile à endommager. Contrairement au Plus 1, le nouveau modèle est aussi muni de son propre connecteur plat à l'arrière, ce qui permet de connecter le Plus 1 — et d'autres unités devant éventuellement faire leur apparition.

Chose surprenante, Acorn a choisi d'utiliser une unité de disquette Sony 3 $\frac{1}{2}$ pouces et non le format d'unité de disquette 5 $\frac{1}{4}$ habituel. On retrouve également un deuxième connecteur plat derrière l'unité de disquette pour permettre de connecter une seconde unité; selon Acorn, ce connecteur pourra accueillir non seulement des unités 3$\frac{1}{2}$ pouces, mais aussi des unités 3 pouces et des unités 5 $\frac{1}{4}$ pouces. Le dispositif du Plus 3

Assemblage de blocs
Le dispositif complémentaire Plus 3 d'Acorn destiné à l'Electron est conçu de façon à pouvoir accepter le Plus 1. La présence de ces interfaces permet à l'utilisateur de l'Electron d'élever sa machine au rang d'un bon micro.
(Cl. Chris Stevens.)

est divisé en quatre parties. A l'extrême gauche, on aperçoit les circuits d'alimentation caractérisés par la présence de gros condensateurs. Très près, apparaissent les bus venant de l'Electron; certains de ces bus sont acheminés vers l'unité de disquette, d'autres sortent directement vers le connecteur plat qui relie l'unité au Plus 1. Viennent ensuite les circuits du contrôleur de disquette. Ces puces renferment les commandes stockées en ROM qui vous permettent de « communiquer » avec l'unité de disquette et avec les puces de gestion de la disquette. Finalement, à l'extrême droite, on retrouve l'unité de disquette elle-même, dans un boîtier métallique. Ce boîtier remplit deux rôles. Il sert d'abord de dissipateur thermique pour éviter qu'une température trop élevée soit atteinte à l'intérieur du module, mais il sert également de cage de Faraday pour empêcher que des champs magnétiques compromettent le bon fonctionnement du système.

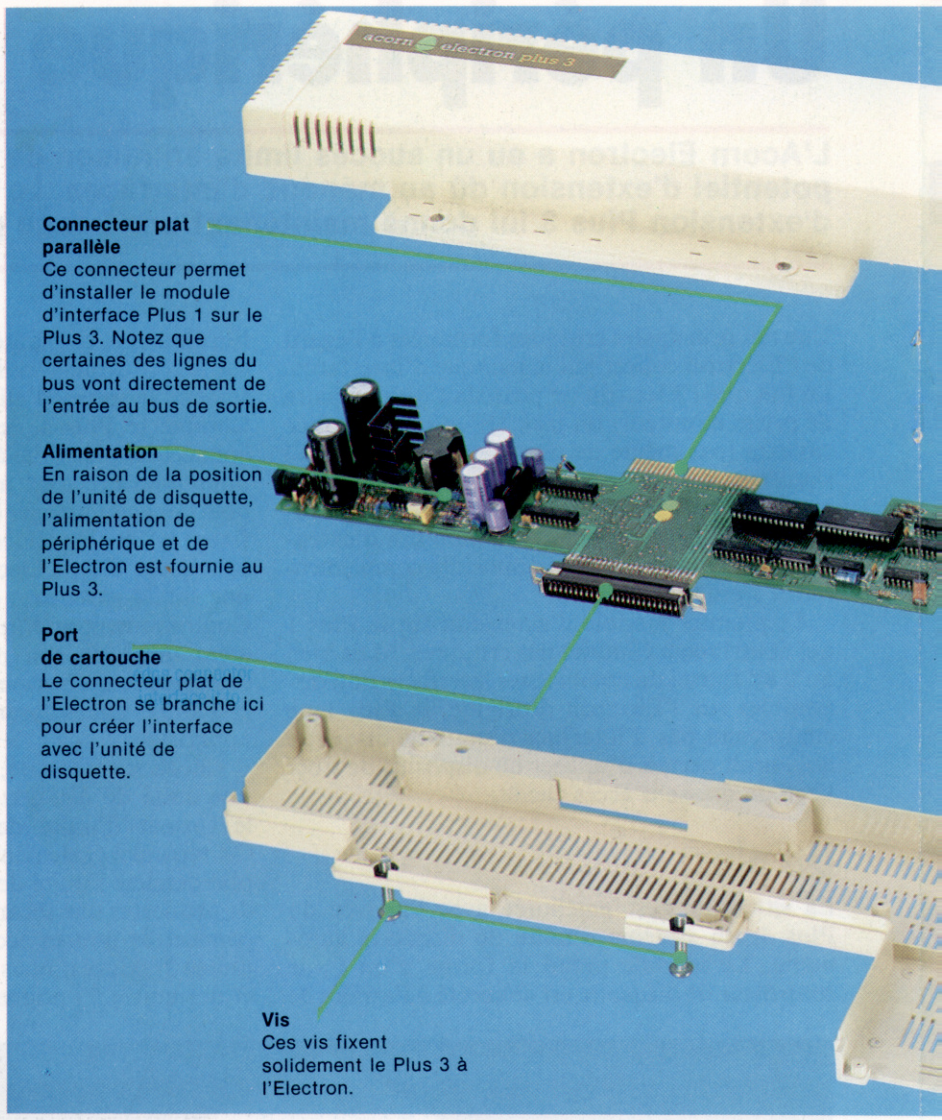
Système ADFS

Les disquettes Sony sont protégées par des enveloppes en plastique rigide munies d'une languette protectrice au-dessus de la fenêtre de lecture/écriture. Lorsqu'elle est formatée, la disquette est divisée en 80 pistes de 16 secteurs. Puisque chaque secteur peut renfermer jusqu'à 256 octets d'information, la capacité totale d'une disquette simple face est donc de 320 K.

Lors de la mise sous tension, lorsque le Plus 3 est installé, le message ACORN ADFS est affiché en plus des messages habituels. Ces lettres sont l'abréviation de Advanced Disk Filing System. Vendue avec le Plus 3, une disquette d'introduction présente des programmes de démonstration semblables à ceux enregistrés sur la cassette fournie avec l'Electron.

La disquette renferme aussi de nombreux utilitaires indispensables pour utiliser les disquettes. Parmi ces utilitaires, BACKUP permet de copier des disquettes entières, DIRCOPY permet de copier uniquement certains fichiers d'une disquette sur une autre, BUILD permet de construire des routines d'amorçage sur la disquette, et DUMP permet d'afficher le contenu d'un fichier en notations hexadécimales et ASCII. Les commandes de disquettes de l'Electron sont précédées du préfixe *; il en va ainsi des commandes du système d'exploitation *LOAD, *RUN et *DELETE. La commande *CAT produira un affichage où apparaissent le titre de la disquette, le numéro de l'unité de disquette, les options, le nom du répertoire en cours, le nom de la bibliothèque en cours et finalement la liste des fichiers.

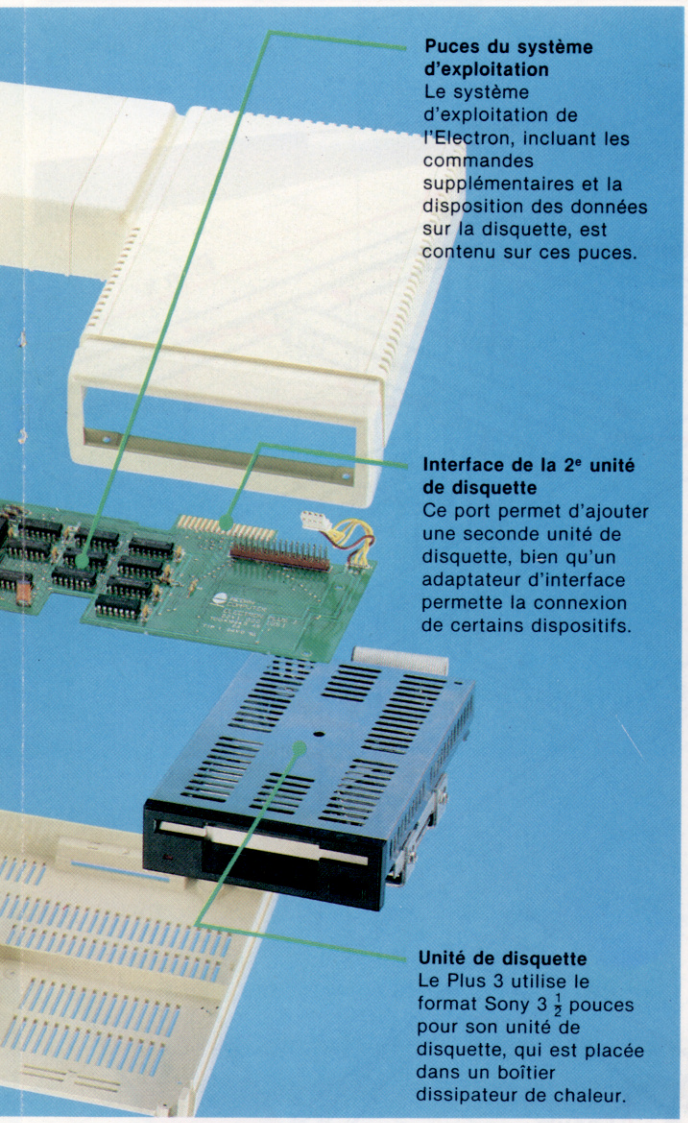
Il est intéressant de s'arrêter sur le système d'exploitation qui présente des particularités quant à la méthode de chargement de nouvelles disquettes. Lorsque la disquette est insérée dans l'unité de disquette Plus 3, l'utilisateur doit en informer le système d'exploitation en tapant la commande *MOUNT — cela a pour effet de charger en mémoire le répertoire (CDS) et la bibliothèque en cours (CSL).



Lorsque l'utilisateur désire changer de disquette, la commande *DISMOUNT est entrée. DISMOUNT a pour effet de fermer tous les fichiers séquentiels pouvant être encore ouverts, et d'effacer les CSD et CSL. Après avoir changé de disquette, la commande *MOUNT doit de nouveau être tapée.

Le système d'exploitation gère les fichiers par voie hiérarchique. Cela autorise un accès simple et rapide aux fichiers grâce à une série de noms d'accès. Au sommet de la hiérarchie on retrouve le répertoire en cours, qui, lors de la mise sous tension, est le répertoire de base implicite $. Pour quitter le CSD et pour adopter tout autre répertoire présent sur la disquette insérée dans l'unité, l'utilisateur tape simplement la commande *DIR suivie du nom d'accès choisi. De nombreux fichiers ou sous-répertoires (fichiers bibliothèques) peuvent être placés à l'intérieur d'un répertoire. Puisque le CSD ne peut contenir que quarante-sept fichiers à la fois, vous devez les disposer en bibliothèques si vous désirez sauvegarder un très grand nombre de fichiers sur la disquette.

Il se peut que ces sous-répertoires renferment un nombre de fichiers qui n'auraient pu être logés

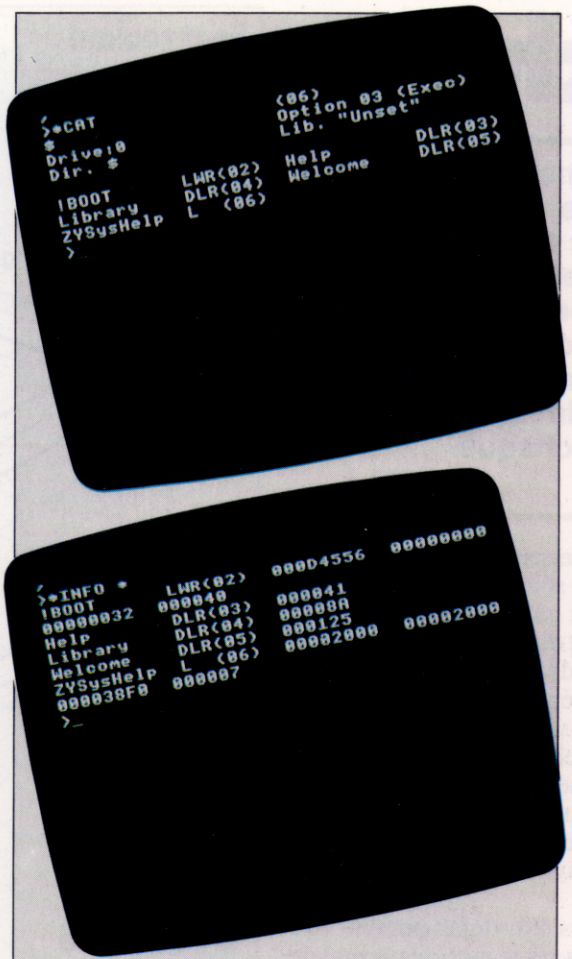


Puces du système d'exploitation
Le système d'exploitation de l'Electron, incluant les commandes supplémentaires et la disposition des données sur la disquette, est contenu sur ces puces.

Interface de la 2e unité de disquette
Ce port permet d'ajouter une seconde unité de disquette, bien qu'un adaptateur d'interface permette la connexion de certains dispositifs.

Unité de disquette
Le Plus 3 utilise le format Sony 3 $\frac{1}{2}$ pouces pour son unité de disquette, qui est placée dans un boîtier dissipateur de chaleur.

Chris Stevens



Écrans du système d'exploitation Acorn
L'écran du haut présente l'affichage produit par une commande *CAT. *INFO affiche le nom d'un fichier, ses conditions d'accès, ses adresses de chargement, d'exécution et de disquette ainsi que la longueur du fichier.

Acorn Electron Plus 3

PRIX

★★ avec écran.

DIMENSIONS

365 × 200 × 50 mm (unité de disquette); 365 × 90 × 50 mm (interface)

CAPACITÉ

320 K pour une disquette simple face.

INTERFACES

Jack d'alimentation, bus d'entrée cartouche, bus de sortie du connecteur plat.

DOCUMENTATION

Le manuel accompagnant le Plus 3 est complet et propose au débutant une approche progressive à l'utilisation de l'unité. Nous avons cependant noté certaines lacunes qui devraient être corrigées.

FORCES

A ce prix, l'interface représente un bon achat et l'utilisateur se voit à la fois doté d'un système d'exploitation et d'une unité de disquette.

FAIBLESSES

Il n'y a à l'heure actuelle que peu de logiciels offerts pour cette machine. De plus, pour utiliser au maximum l'unité de disquette, on a besoin de dispositifs complémentaires comme une interface d'imprimante, qui n'est disponible que sur le plus 1.

sur le répertoire principal; ainsi, pour avoir accès à un fichier nommé Fred sur une disquette renfermant plusieurs répertoires comportant des bibliothèques, le format d'appel serait le suivant : $.biblio1.fred.

Bien que cette méthode puisse engendrer une perte de temps, elle offre l'avantage de pouvoir sauvergarder dans chaque bibliothèque un fichier nommé Fred. Si l'utilisateur demeure dans le bon répertoire, sauvegarder le fichier de nouveau n'écrasera pas les autres fichiers Fred contenus sur la disquette.

Plus sérieux

Le système comporte de nombreux utilitaires. On dispose d'une série de commandes permettant, par exemple, de nommer et de renommer des disquettes, des répertoires et des fichiers. Il y a aussi plusieurs commandes servant à gérer les conditions d'accès aux fichiers. C'est possible grâce à la commande *ACCESS, qui limite l'accessibilité d'un fichier.

Par exemple, entrer la commande *ACCESS fichier1 RL « verrouille » le fichier, ce qui le met à l'abri de tout effacement accidentel, mais qui autorise sa lecture. D'autre part, *ACCESS fichier1 WR autorise à la fois la lecture et l'écriture du fichier. Pour interdire à quiconque de copier un fichier code machine sur la disquette, l'utilisation du suffixe E dans le nom du fichier limite le nombre de commandes pouvant être utilisées sur ce fichier, et limite également les diverses méthodes d'effacement du fichier.

Le manuel fourni avec le Plus 3 est complet, comporte des exercices pratiques élaborés et se situe à un niveau de qualité auquel Acorn a habitué ses utilisateurs. Il y a quand même certains défauts dans le manuel, qui, bien que mineurs, sont gênants. Par exemple, les abréviations CSD et CSL sont mentionnées à une page du guide, mais ne sont expliquées que plus loin dans le manuel.

Le peu d'interfaces, particulièrement l'absence de port d'unité de disquette, donnait à l'Electron une réputation de machine de jeux qui ne pouvait être considérée comme un outil véritablement sérieux. Cependant, suite à la diminution du prix de l'Electron et à l'arrivée des interfaces Plus 1 et Plus 3, la machine offre un rapport qualité/prix intéressant. L'Electron peut maintenant se tailler une part du marché de sa catégorie.

Ébauche

Nous commençons un nouveau projet : construire une petite machine capable de saisir et de déplacer des objets avec précision. Elle sera commandée à partir du Commodore 64 ou du Spectrum. Nous mettrons également au point un logiciel de commande pour chaque ordinateur.

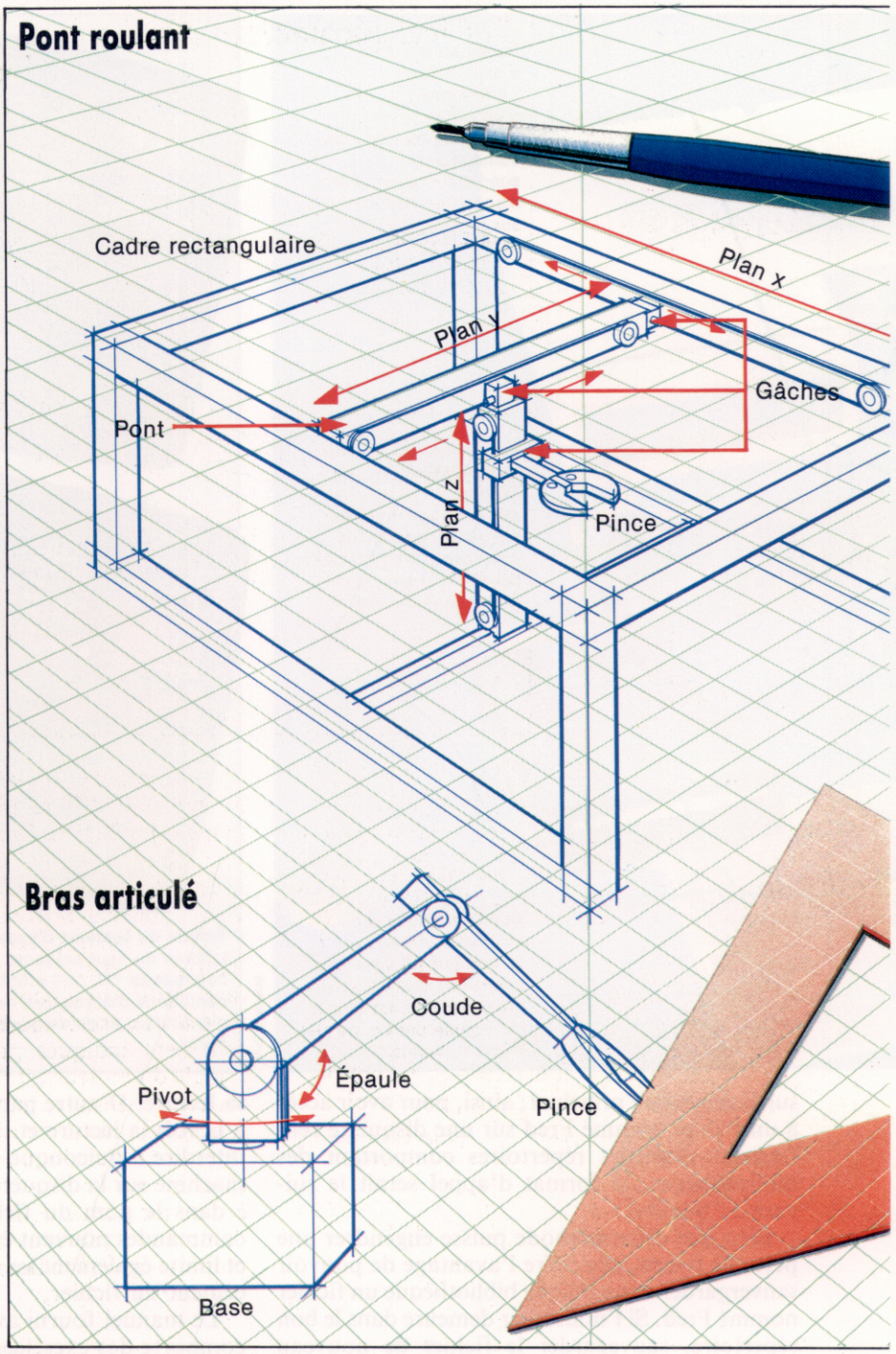
Pour ce projet, la tâche essentielle consiste à faire en sorte que la machine soit capable de se déplacer d'un point de départ quelconque, de saisir un objet (par exemple de la dimension d'une bobine de fil) et de le mettre dans une boîte avant de regagner son point de départ. Nous supposerons que toutes les positions sont connues à l'avance.

De quel type de machine avons-nous besoin pour effectuer cette tâche? Il est évident que nous aurons besoin de plusieurs moteurs. Ces moteurs peuvent être hydrauliques ou pneumatiques, mais nous opterons pour un système de moteurs entièrement électriques.

Trois types de moteurs électriques peuvent être utilisés; le plus simple étant le moteur normal de type CC. Bien que de tels moteurs soient peu coûteux, ils sont difficiles à commander avec précision à partir d'un dispositif numérique, comme un ordinateur domestique. Pour obtenir le niveau de précision désiré, nous devrions ajouter des codeurs sur les axes des moteurs. Chaque moteur nécessiterait alors une logique de décodage, soit par matériel, soit par logiciel, pour permettre un positionnement précis.

Les moteurs pas à pas et les servomoteurs CC sont plus faciles à commander par ordinateur. Les servomoteurs sont cependant coûteux et nécessitent divers types de circuits de commande; ils sont de dimensions généralement trop grandes pour nos besoins. Pour un petit système comme celui qui est envisagé, les servomoteurs présentent deux avantages importants : ils font l'objet d'une production en série pour le marché du modélisme et sont donc largement diffusés et relativement bon marché.

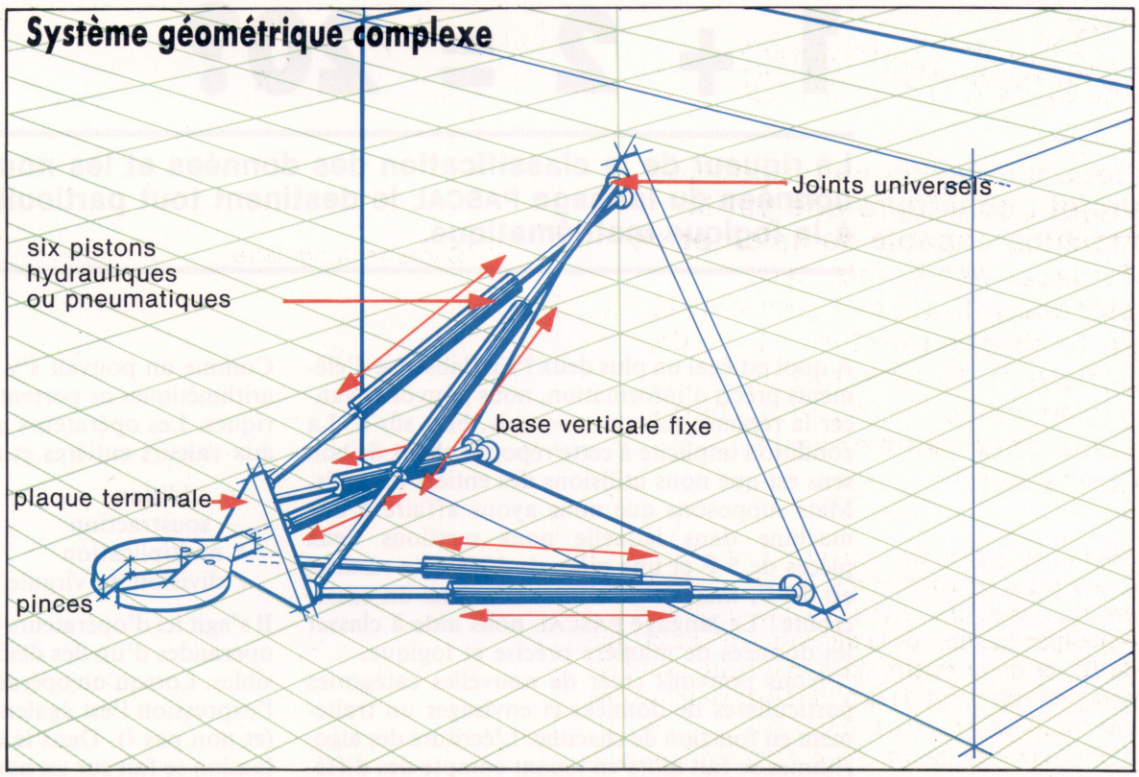
Nous avons déjà expliqué comment fonctionne un servomoteur; pour récapituler brièvement, un servomoteur est composé d'un petit moteur de 5 v CC, commandé par une boucle à retour numérique. La position angulaire de l'axe du moteur est contrôlée par un potentiomètre miniature et comparée avec le signal de commande produit par la puce intégrée. Le signal de commande est une impulsion de 5 v d'une durée comprise entre une et deux millisecondes; cette durée d'impulsion détermine l'angle de rotation de l'axe du moteur. Les impulsions de commande peuvent être répétées à un rythme de 50 Hz (cycle par seconde). Nous avons déjà décrit le logiciel de commande pour les servomoteurs (simples ou multiples). Le programme est piloté au moyen d'interruptions, ce qui permet l'exécution du logiciel BASIC en arrière-plan et son interruption à intervalles réguliers pour envoyer des groupes d'impulsions aux moteurs.



Prototypes possibles
Voici deux types de machines pouvant saisir une bobine de fil et la placer dans une boîte ou dans une tasse. Le pont roulant ne peut effectuer les tâches compliquées dont peut se charger le bras-robot. Il offre l'avantage d'une géométrie de mouvement très simple; les moteurs du bras-robot requièrent une commande beaucoup plus complexe pour exécuter un simple mouvement linéaire de la pince.

Géométrie de la machine

Nous devons à présent envisager les diverses contraintes qui sont imposées. La saisie d'un objet, comme une bobine de fil, implique l'utilisation d'un mécanisme à pince. Ce mécanisme nécessite déjà un moteur. Ensuite, la pince devra



Géométrie complexe Grâce à la commande par ordinateur de moteurs hydrauliques ou pneumatiques, il est possible de concevoir des machines dotées d'une grande souplesse. L'interaction des divers moteurs servant à mouvoir les sections de la machine peut être gérée par un logiciel évolué. Le fonctionnement de ce logiciel est, dans une grande mesure, incompréhensible au niveau de l'utilisateur. (Cl. Kevin Jones.)

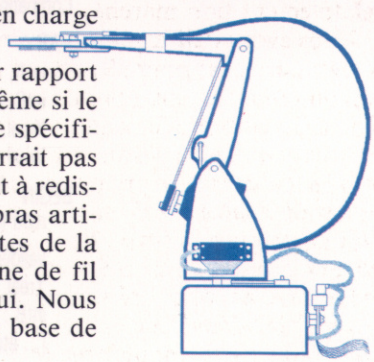
pouvoir être déplacée dans trois plans. La machine doit être capable de se déplacer d'au moins 10 cm de gauche à droite (le plan x), de quelques centimètres de l'arrière vers l'avant (le plan y) et d'au moins 10 cm de haut en bas (le plan z). Chaque plan de mouvement nécessite au moins la présence d'un moteur, et la pince doit être manœuvrée en combinant les mouvements de ces trois moteurs avec une précision d'au moins 2,5 mm pour lui permettre de regagner son point de départ. Elle doit être capable de saisir un objet pesant au moins 20 g, et sa mâchoire doit s'ouvrir d'au moins 3 cm.

L'un des modèles qui répondraient à ces exigences intégrerait un système de pont roulant reposant sur un cadre rectangulaire. Le pont roulant et la pince doivent être déplacés en les reliant, au moyen de fils métalliques et de gâches, à des poulies qui doivent être connectées à des moteurs pas à pas ou à des servomoteurs. Ce système nécessiterait quatre moteurs, trois pour commander le mouvement et un pour actionner la pince.

Une deuxième possibilité, plus près du modèle humain, consiste en un bras articulé. La rotation de chaque articulation serait commandée par un moteur. Les servomoteurs numériques sont parfaitement indiqués pour cette application, et l'angle du moteur est facilement commandé par logiciel, permettant ainsi de commander de façon précise les membres du bras. Déplacer la pince au moyen d'un bras-robot implique un logiciel de commande plus complexe que pour le déplacer au moyen du pont roulant. Par exemple, le déplacement de la pince dans un seul plan — le plan x — nécessite la commande simultanée d'au moins deux moteurs avec le système du bras articulé; ce mouvement pouvait être obtenu en utilisant un seul moteur avec le système à pont roulant. Cependant, puisque le système sera commandé par ordinateur, des séquences de commandes complexes ne posent pas réellement de problèmes; elles peuvent être prises en charge par un logiciel bien conçu.

Le bras articulé offre un avantage par rapport au pont roulant : il est plus souple. Même si le pont roulant pouvait effectuer la tâche spécifique que nous avons définie, il ne pourrait pas effectuer d'autres tâches sans que l'on ait à redisposer les composants du système. Le bras articulé peut effectuer des tâches différentes de la tâche définie, comme placer une bobine de fil dans une boîte qui se trouve près de lui. Nous retiendrons donc le bras-robot comme base de notre projet de construction.

Repères en couleur Dans chaque article de notre projet, nous utiliserons la couleur correspondante de ce dessin pour indiquer de quelle partie du bras-robot il est question. (Cl. Ian McKinnell.)



1 + 2 = 20?

La rigueur de la classification des données et les nombreux types de données du langage PASCAL le destinent tout particulièrement à la logique mathématique.

A quoi est égal un plus deux ? En l'absence d'éléments précis d'information, nous pouvons avancer la réponse suivante : « trois, bien sûr ». La condition implicite à cette réponse pleine de bon sens est que nous utilisions des entiers naturels. Mais supposons que nous ayons affaire à une machine dans laquelle nous mettions deux pièces de 5 F et une pièce de 10 F ? Le résultat est 20 F, mais en fait nous recevons un casse-croûte ! Le langage PASCAL nous aide à classer les données de manière précise et logique.

Nous pouvons créer de nouvelles catégories particulières de données et envisager un traitement en fonction de chacune. L'écriture des algorithmes se fait alors en tenant compte des différents types de données. En cas de faute de syntaxe, comme par exemple lire au clavier une valeur booléenne vraie, le compilateur PASCAL débusque immédiatement l'erreur logique. Vous pouvez imaginer le temps gagné, lorsque le compilateur n'exécute que des programmes sources totalement exempts d'erreurs. L'incompatibilité entre les différents types de données et la souplesse de leur syntaxe constituent un des atouts majeurs du PASCAL. En fait, loin de nous restreindre, une description rigoureuse deviendra un avantage. Le programmeur imposera de lui-même des contraintes supplémentaires à ses variables.

Après avoir vu les types simples de variables (ceux qui n'acceptent qu'une valeur), nous pouvons résumer les règles qui président à leur emploi. Les grandeurs scalaires ne sont aucunement compatibles entre elles, même si elles peuvent avoir des caractéristiques communes et parfois cohabiter. Un nombre réel peut constituer une approximation d'un nombre entier. Il est donc possible d'affecter des valeurs entières à des réels, mais non l'inverse. En voici quelques exemples :

```
Compatibilité d'un programme (entrées, sorties) ;
VAR
   intA,
   intB :integer ;
   Xreal,
   Yreal : real ;
BEGIN
   read (intA, Xreal) ; {lecture d'un entier, puis de tout nombre
   valide}
   Yreal : = intA ; {réel : = entier, OK}
   intB : = Xreal ; {**ERREUR : illégal**}
   {... etc.
```

Comme on pouvait s'y attendre, les opérations arithmétiques ne portent que sur des types numériques. Les opérateurs usuels s'appliquent donc aux valeurs entières et aux valeurs réelles :

+ addition
− soustraction
* multiplication
/ division en virgule flottante

Il s'agit ici d'opérateurs binaires nécessitant deux opérandes d'un des deux types numériques possibles. Lorsqu'un opérande est réel, le résultat de l'expression l'est également : 2 + 2.0 est égal à 4.0 (et non pas 4). Dans le cas de la division, l'évaluation se fait sur un nombre réel, même lorsque les deux opérandes sont des entiers : 3/5 donne 0.6, 8/4 = 2.0.

La division de deux entiers sous forme d'un résultat réel est souvent dépourvu de sens (répartir douze chocolats entre dix personnes par exemple). Le résultat entier et le reste peuvent être obtenus par les deux opérateurs de division entre entiers, DIV et MOD (les deux étant des mots réservés en PASCAL). 15DIV5 = 3, 15MOD5 = 0 ; 31DIV7 = 4 et 31MOD7 = 3. Attention, si vos données peuvent être négatives, la division entière ne s'applique pas aux valeurs négatives de dénominateurs. Tant pour la division entière que réelle, diviser par zéro provoque une erreur à l'exécution.

Les multiplications et divisions sont prioritaires par rapport aux additions et soustractions. Mais les parenthèses permettent d'inverser cet ordre de préséance. Par exemple (8 + 4) DIV 2 = 6, mais 8 + 4 DIV 2 = 10.

Fonctions transfert

L'assignation directe d'une valeur réelle à une valeur entière n'est pas possible, mais ce « tablage » peut cependant avoir lieu par l'intermédiaire de fonctions très utiles dites de transposition, les identificateurs trunc et round. Trunc effectue la troncature d'un nombre réel, sans tenir compte de sa partie fractionnaire (et quelle qu'elle soit). Ainsi, trunc (3.999) = 3 et trunc (− 123.456) = − 123. La fonction round arrondit les nombres de manière intelligente, vers zéro lorsque la partie fractionnaire est inférieure à 0,5 ; à partir de zéro dans les autres cas. Ainsi round(1234.5) = 1235 et round(− 0.49237) = 0. Bien sûr, un argument réel donnant un résultat entier hors champs par rapport à l'entier-type donnerait une erreur pour les deux fonctions (entre −Maxint et Maxint). C'est pourquoi

vous aurez intérêt à toujours utiliser préalablement une instruction IF. En outre, l'argument doit être réel et non entier (après tout, il y a déjà eu troncature et approximation).

La fonction INT présente un raffinement qui sera sûrement agréable à ceux qui ont souffert de sa version BASIC. Le PASCAL comporte une fonction prédéfinie qui restitue un résultat booléen : faux pour un argument pair et vrai pour un argument impair. L'identificateur utilisé s'appelle odd.

```
IF odd (N)
   THEN
     WriteLn ('nombre impair')
   ELSE
     WriteLn ('nombre pair')
```

ou encore :

```
IF odd (N) THEN
   IF N MOD 2 = 0 THEN
     WriteLn ('Quel compilateur !')
```

La table donne toutes les fonctions arithmétiques PASCAL. Elles peuvent toutes recevoir un seul argument numérique, réel ou entier. Les deux premières fonctions, abs et sqr, restituent une valeur compatible avec leur argument. Ainsi, abs(–19.372) donne 19. 372, et abs(255), 255. Les autres fonctions donnent *toujours* un résultat réel ; ainsi sqr(16) donne 4,0, et non pas 4. Cela est dû au fait que les algorithmes servant à les évaluer sont construits sur la base d'une sommation d'une série de termes, tous fractionnaires.

Il y a là une bonne leçon de style par rapport aux réels. Le résultat d'une fonction arithmétique portant sur des nombres réels n'est pas exprimé en termes d'égalité (par le signe égal), mais d'équivalence (telle opération « donne » tel résultat).

Les nombres réels ne peuvent être strictement égaux à un résultat. La moindre erreur de calcul signifiera que deux nombres réels dits « équivalents » pourront très bien varier entre eux d'une valeur égale, à 1.0 E-27 par exemple. C'est, bien sûr, une différence sans signification, mais l'ordinateur ne le sait pas. Au lieu d'utiliser IF X = Y THEN... nous pouvons utiliser la fonction PASCAL prédéfinie abs :

```
IF abs (X-Y) < PlusPetiteValeur Significative THEN {etc.}
```

Nous apprécions ainsi la différence entre X et Y. Une définition de constante (CONST) au début du programme donnera cette plus petite valeur significative. Les logarithmes sont donnés en base naturelle (e), et non décimale (10) ; exp élevant e à la puissance de son argument, c'est-à-dire calculant son exponentielle.

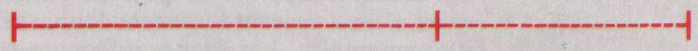
Wirth n'a pas voulu qu'il y ait à proprement parler d'opérateur PASCAL de calcul exponentiel. Cela évite des incohérences de programmation du style BASIC :

```
500 LET D = B ^ 2 + 4 * A * C
```

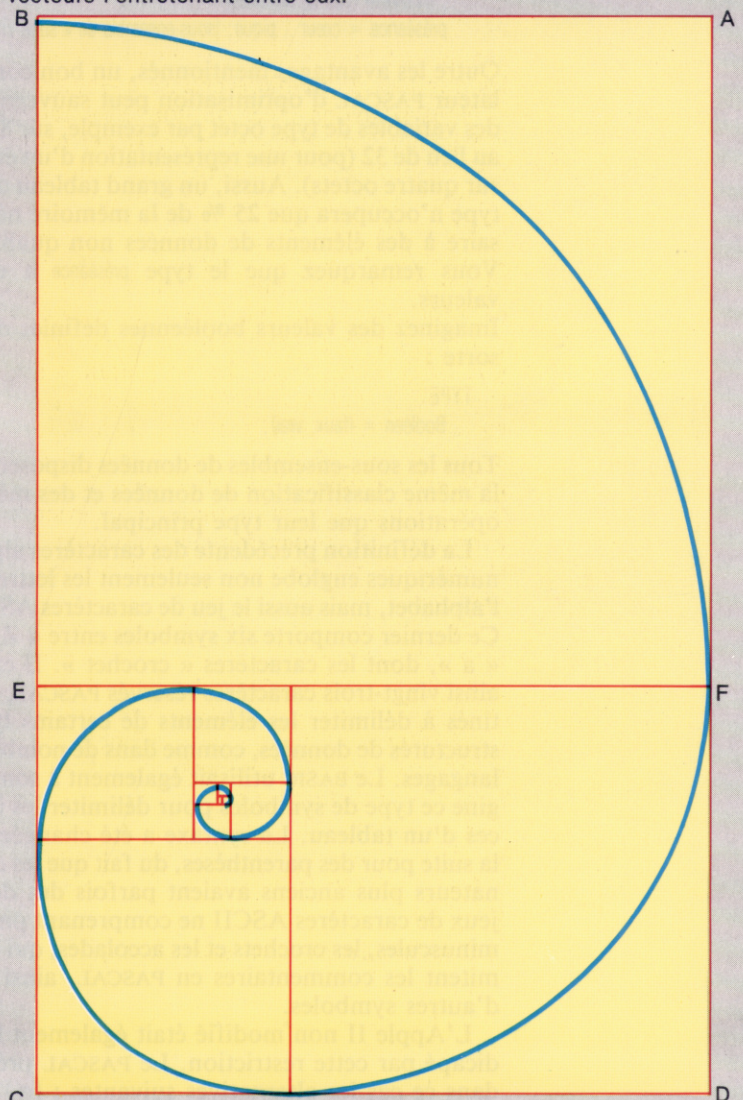
qui est la méthode la plus lente et la moins appropriée pour calculer le carré d'un nombre. B * B est évidemment bien mieux. Vous remarquez que la fonction PASCAL sqr(N) donne un véritable entier

Le nombre d'or

Il n'existe qu'un seul point sur une ligne qui la croise, de telle sorte que le rapport de la longueur de la plus petite section de la droite par rapport à la plus grande soit le même que celui entre la plus grande et la totalité de la droite. Ce rapport est appelé le nombre d'or ; comme le nombre π, c'est un nombre irrationnel comportant un nombre infini de chiffres.



Le nombre d'or fut utilisé par les artistes et architectes grecs, notamment pour le Parthénon. Il revient souvent dans toute l'histoire de l'architecture au cours des diverses théories des proportions. Une de ses manifestations les plus familières se trouve néanmoins dans la nature : la spirale d'un coquillage respecte en effet ce rapport « sacré », ses rayons-vecteurs l'entretenant entre eux.



(Cl. Liz Dixon)

Les côtés du rectangle ABCD sont dans un rapport mutuel égal au nombre d'or. Un carré peut y être tracé, ABEF, de telle sorte qu'un rectangle entretenant le même rapport entre ses côtés soit formé (EFDC). Le processus peut être poursuivi à l'infini, les angles des carrés successifs étant des points situés sur la spirale.

directement comme carré de N (en supposant que N soit un entier); sqr(X/3) donne une valeur réelle pour résultat. Sachez également qu'en PASCAL la racine carrée d'un nombre s'écrit sqrt(nombre), et non sqr(nombre) comme en BASIC.

Types sous-ensembles

Les scalaires sont définis pour un ensemble de valeurs, mais peuvent également donner lieu à des sous-ensembles. Les limites inférieure et supérieure du sous-ensemble sont séparées par le signe (..) dans la définition de l'identificateur du sous-ensemble. Ainsi,

```
TYPE
   octet = 0 .. 255 {sous-ensemble de l'entier}
   alpha = 'A' .. 'Z'; {sous-ensemble du caractère}
   suite = (trèfle, carreau, cœur, pique)
   ; {nouveau type de données}
   préséance = cœur .. pique; {sous-ensemble de « suite »}
```

Outre les avantages mentionnés, un bon compilateur PASCAL d'optimisation peut sauvegarder des variables de type octet par exemple, sur 8 bits au lieu de 32 (pour une représentation d'un entier sur quatre octets). Aussi, un grand tableau de ce type n'occupera que 25 % de la mémoire nécessaire à des éléments de données non qualifiés. Vous remarquez que le type préséance a deux valeurs.
Imaginez des valeurs booléennes définies de la sorte :

```
TYPE
   Booléen = (faux, vrai);
```

Tous les sous-ensembles de données disposent de la même classification de données et des mêmes opérations que leur type principal.

La définition précédente des caractères alphanumériques englobe non seulement les lettres de l'alphabet, mais aussi le jeu de caractères ASCII. Ce dernier comporte six symboles entre « Z » et « a », dont les caractères « crochet ». Il existe ainsi vingt-trois caractères réservés PASCAL, destinés à délimiter les éléments de certains types structurés de données, comme dans de nombreux langages. Le BASIC utilisait également à son origine ce type de symboles pour délimiter les indices d'un tableau. La syntaxe a été changée par la suite pour des parenthèses, du fait que les ordinateurs plus anciens avaient parfois des demi-jeux de caractères ASCII ne comprenant pas les minuscules, les crochets et les accolades, qui délimitent les commentaires en PASCAL, ainsi que d'autres symboles.

L'Apple II non modifié était également handicapé par cette restriction. Le PASCAL prévoit dans ce cas les alternatives suivantes : (.et.) à la place de [et]; (*et*) à la place de {et}. La deuxième alternative, concernant les délimiteurs de commentaires, est assez répandue. Seuls les compilateurs de nombres ISO acceptent l'alternative aux crochets. L'une ou l'autre des alternatives est possible, mais aussi les deux ensemble : (*commentaire valide}.

Fonction	Valeur résultante
abs (K)	valeur absolue de K
sqr (K)	carré de K
sqrt (K)	racine carrée de K
sin (A)	sinus de A radians
cos (A)	cosinus de A radians
arctan (T)	angle en radians, avec tangente T
ln (K)	logarithme naturel de K
exp (L)	e élevé à la puissance L

Bande d'or

Le programme Or, donné ci-dessous, génère des termes de Fibonacci, et les affiche avec leurs rapports mutuels. Il permet de voir que l'une des propriétés des séries de Fibonacci (1, 1, 2, 3, 5, 8, 13, etc.) est que les rapports successifs des chiffres, pris deux à deux, vont croissant et tendent vers le rapport idéal du nombre d'or.
Le programme met en œuvre certains principes déjà traités; il donne notamment des exemples plus poussés de la structure syntaxique PASCAL, REPEAT. Essayez comme exercice de modifier la condition finale de la boucle, de sorte qu'elle prenne fin lorsque le prochain terme à calculer dans une suite de Fibonacci excède Maxint.

```
Program  Nombre d'or (sorties);
CONST
    Epsilon = 1,0E-08;
TYPE
    Fibonacci = 1 .. Maxint;
VAR
    première,
    deuxième,
    suivante  : Fibonacci;
    rapport,
    Nombre d'Or   : réel;
    compteur   : entier;
BEGIN
    WriteLn  ( 'Le rapport du nombre d'or'
    : 30);
    WriteLn  ( 'série de Fibonacci :');
    premier :1;  par définition
    deuxième := 1;
    rapport := premier/deuxième;
    compteur := 0;

REPEAT
    IF compteur MOD 10 = 0 THEN
       BEGIN {En-têtes toutes les 10 lignes}
          WriteLn;
          WriteLn ( 'Premier' : 10, 'deuxième'  : 10,
             'Rapport'  : 14 );
          WriteLn
       END

     WriteLn (premier : 10, deuxième : 10,
              rapport : 15 : 8 );
     compteur := compteur + 1;
     Nombre d'Or  :=  rapport;  souvenez-vous de
          l'ancien rapport
     suivant := premier + deuxième;
     premier := deuxième; {avancez..}
     deuxième := suivant;  la série
     rapport := premier / deuxième
UNTIL abs (rapport - Nombre d'Or)  Epsilon ;

     WriteLn;
     WriteLn ( 'La moyenne d'Or est : ',
           100 * rapport : 10 : 5, '%')
  END
```

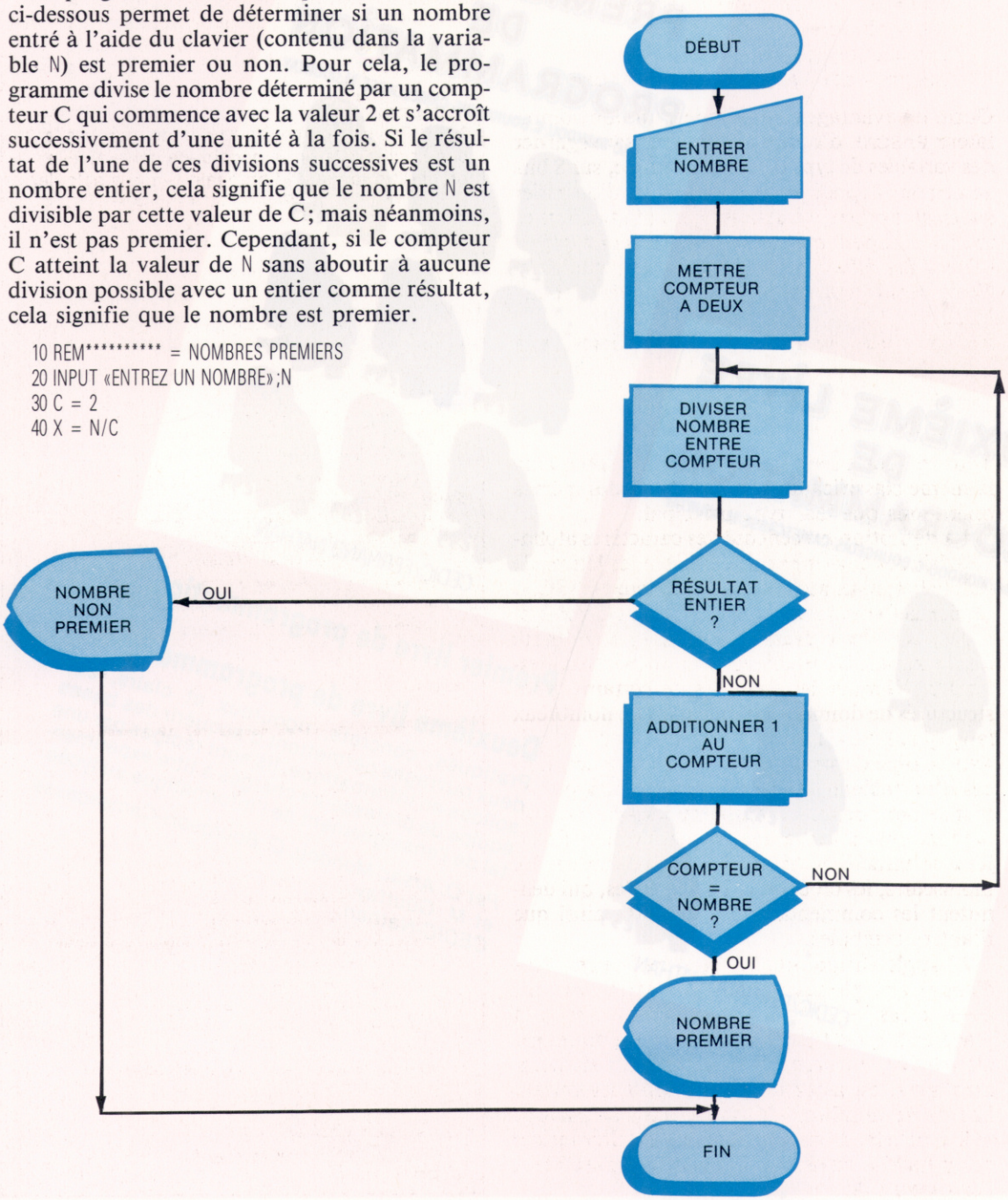
Nombres premiers

Après avoir généré des nombres aléatoires, il est logique de revenir à l'éternelle question des nombres entiers. Mais comment savoir si un nombre est premier ou non.

Très souvent, lorsqu'il s'agit de savoir si un nombre est premier ou non, le doute peut apparaître. Il est alors nécessaire de faire appel à une série d'opérations qui peuvent se prolonger selon la complexité du nombre en question.

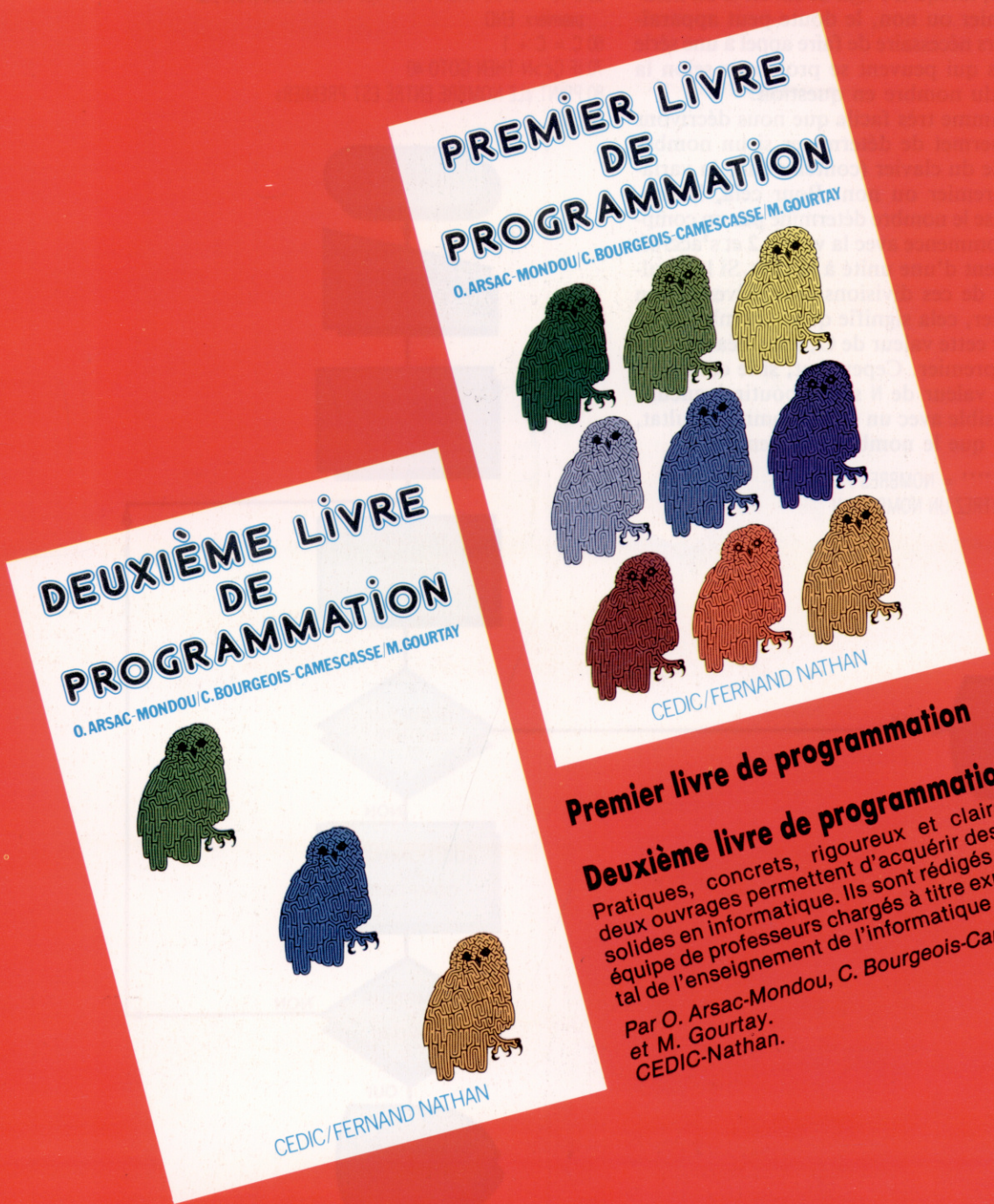
Le programme très facile que nous décrivons ci-dessous permet de déterminer si un nombre entré à l'aide du clavier (contenu dans la variable N) est premier ou non. Pour cela, le programme divise le nombre déterminé par un compteur C qui commence avec la valeur 2 et s'accroît successivement d'une unité à la fois. Si le résultat de l'une de ces divisions successives est un nombre entier, cela signifie que le nombre N est divisible par cette valeur de C; mais néanmoins, il n'est pas premier. Cependant, si le compteur C atteint la valeur de N sans aboutir à aucune division possible avec un entier comme résultat, cela signifie que le nombre est premier.

```
10 REM********** = NOMBRES PREMIERS
20 INPUT «ENTREZ UN NOMBRE»;N
30 C = 2
40 X = N/C
50 IF X = INT(X) THEN PRINT «Le nombre entré n'est pas
   premier»: END
60 C = C + 1
70 IF C<>N THEN GOTO 40
80 PRINT «LE NOMBRE ENTRÉ EST PREMIER»
90 END
```



Livres pour débuter

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

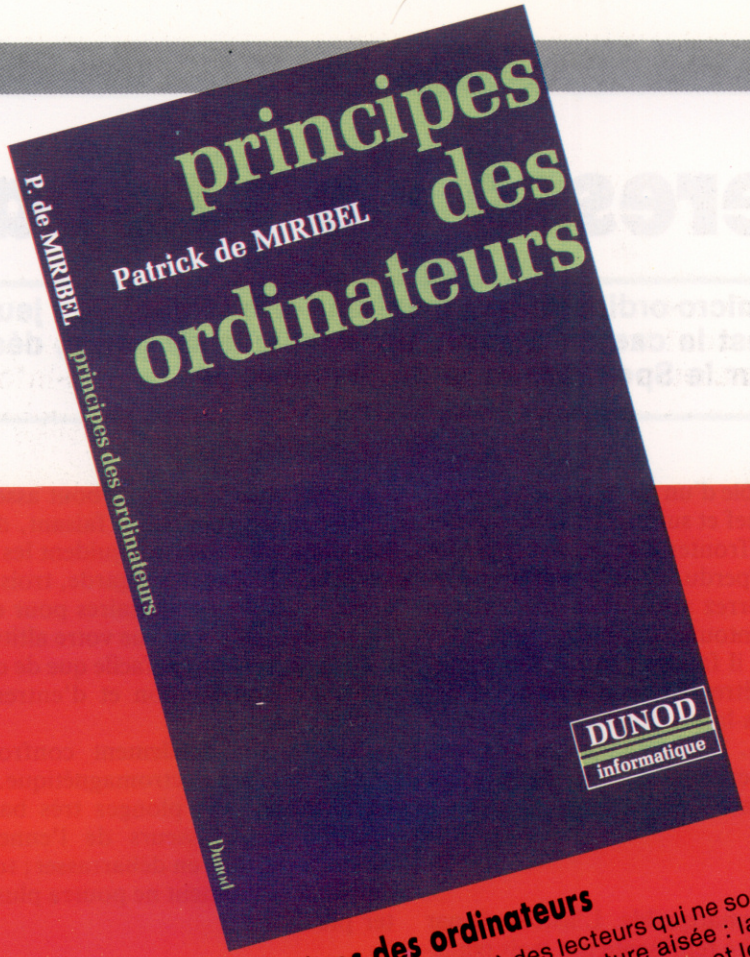


Premier livre de programmation

Deuxième livre de programmation

Pratiques, concrets, rigoureux et clairs, ces deux ouvrages permettent d'acquérir des bases solides en informatique. Ils sont rédigés par une équipe de professeurs chargés à titre expérimental de l'enseignement de l'informatique au lycée.

Par O. Arsac-Mondou, C. Bourgeois-Camescasse et M. Gourtay.
CEDIC-Nathan.



Principes des ordinateurs

« Cet ouvrage, destiné à des lecteurs qui ne sont pas tous des techniciens spécialistes, est de lecture aisée : la compréhension est facilitée par des tableaux et des figures, et le désir d'être accessible à tous n'a pas empêché P. de Miribel de fournir, en annexe, à ceux qui le désirent, des précisions scientifiques sur les circuits électroniques, les systèmes de numération et les procédés de calcul numérique... »

Par P. de Miribel.
144 pages.
Dunod informatique.

Choisir son micro-ordinateur

Apprendre? Jouer? Programmer? Travailler? Pour la maison ou le bureau? Face à la multitude des modèles présents et à venir, comment le consommateur doit-il se déterminer? Grâce à ce guide pratique, vous pourrez définir vos besoins et les caractéristiques du micro-ordinateur. Avant d'entamer la tournée des magasins de micro-informatique, il vous sera possible d'établir la fiche signalétique qui vous permettra de choisir « le micro-ordinateur qu'il vous faut » en diminuant les risques d'erreurs.

Par Yvon Dargery
Collection Micro monde
CEDIC-Nathan

Forteresse de l'espace

Les jeux sur micro-ordinateurs s'inspirent étroitement des jeux d'arcades. C'est le cas de Zaxxon, un classique du genre, désormais disponible pour le Spectrum et le Commodore 64.

Vous allez jouer le rôle d'un pilote lancé aux commandes d'un astronef et surmonter toutes sortes de périls avant d'affronter Zaxxon le robot. En début de partie, perdu dans l'espace intergalactique, vous verrez apparaître une « forteresse de l'espace » entourée d'un haut mur. Pour franchir l'obstacle, il faudra profiter des moindres fissures, sans être victime des missiles, des radars, des canons-laser, tous représentés de façon très réaliste... Il faut également bombarder les dépôts de carburant, afin de pouvoir continuer plus avant.

L'ombre

Le graphisme est remarquable. L'astronef, par exemple, est en réalité représenté par trois plans-objets : un pour le vol à l'horizontale, un pour la plongée, un pour la montée en chandelle. Petite touche réaliste : ces lutins sont dilatés ou réduits, suivant les cas, pour donner à la scène plus de présence.

Et le vaisseau spatial a une ombre, qui augmente ou diminue suivant l'altitude de l'engin, ce qui permet d'ailleurs de négocier plus subtilement tous les passages.

Si votre appareil est détruit par l'adversaire, vous devrez recommencer depuis le début. C'est pourtant, avec un peu de pratique, une phase de jeu dont vous viendrez à bout assez facilement. Il vaut la peine, chemin faisant, de détruire les installations radar, qui guident les missiles droit sur vous ! Mais prenez garde, trois coups au but consécutifs sont nécessaires pour en triompher. Ne perdez jamais de vue votre altitude — rien de plus dangereusement facile que de commettre des erreurs d'appréciation et d'entrer en collision avec un missile.

Vous serez finalement confronté avec un champ de force électromagnétique. Grimpez très en hauteur, puis plongez très bas de façon à détruire les chasseurs de l'ennemi. C'est le moment de vous en débarrasser ; tout adversaire détruit à ce moment ne pourra plus vous affronter plus tard.

Zaxxon en trois dimensions

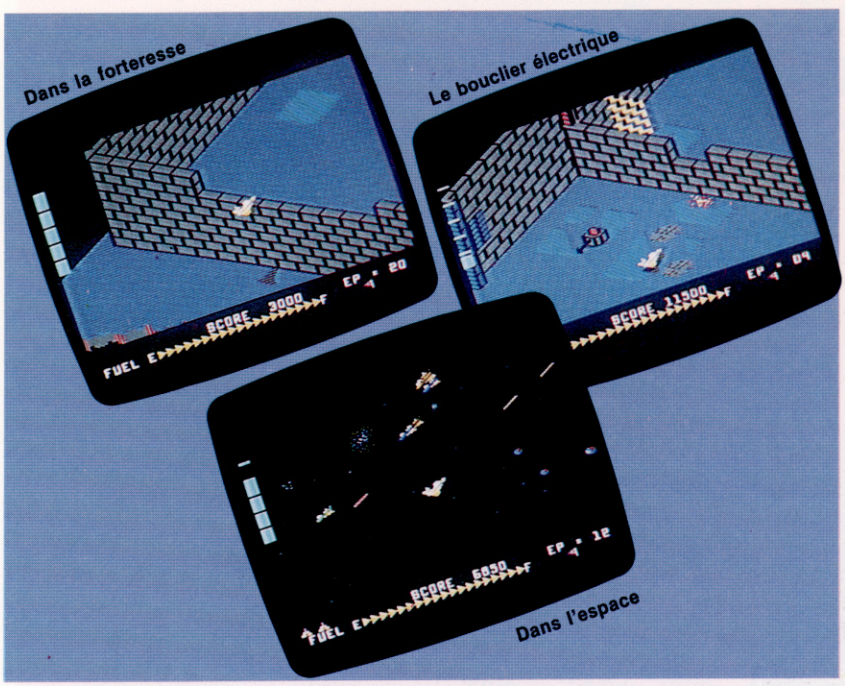
Après cette séquence, votre vaisseau se retrouve de nouveau dans l'espace, et doit faire face à des attaquants multiples. La difficulté est ici de combattre en trois dimensions sur un écran qui n'en comporte que deux.

Pour attaquer l'ennemi, il faut monter ou descendre à la bonne altitude, ce qui est assez compliqué ; la meilleure solution consiste à procéder à des estimations, et par exemple à s'attaquer à des astronefs qui ont, à peu de chose près, la même taille que votre vaisseau. Inutile de dire qu'il vous faudra des réflexes rapides ! C'est le moins que l'on puisse dire pour ce genre de jeu.

Vous revenez ensuite à la forteresse de l'espace. Cette fois, la fissure par laquelle vous pourrez vous infiltrer est bien plus réduite... et les cibles infiniment plus coriaces !

Pour finir, vous aurez à combattre Zaxxon le robot. Votre vaisseau s'arrête et la seule solution est désormais d'être plus rapide que votre opposant. Trois coups au but sur ses lance-missiles vous seront nécessaires pour en venir à bout. Et pendant tout ce temps, Zaxxon avance : s'il vous détruit, vous serez renvoyé en début de partie.

Un « combat dans l'espace » tout à fait classique, mais remarquablement conçu et très soigné tant au niveau des images qu'à celui de la richesse des différentes phases du jeu.



Par-delà l'espace
On voit ici différents moments de la partie. Le joueur entre d'abord, aux commandes d'un astronef, dans la forteresse spatiale. Pour franchir le mur, il faudra que l'engin soit à la bonne altitude. Il est possible de marquer des points en détruisant les défenses antiaériennes de l'adversaire. Une fois la forteresse détruite, l'astronef se retrouve dans l'espace, où il doit faire face à d'innombrables assaillants dont le dernier est Zaxxon le robot.
(Cl. Ian McKinnell.)

Zaxxon : Spectrum et Commodore 64.
Éditeur : US GOLD.
Auteur : Peter Adams.
Manche à balai : indispensable.
Format : cassette (Spectrum), cassette et disquette (Commodore 64).