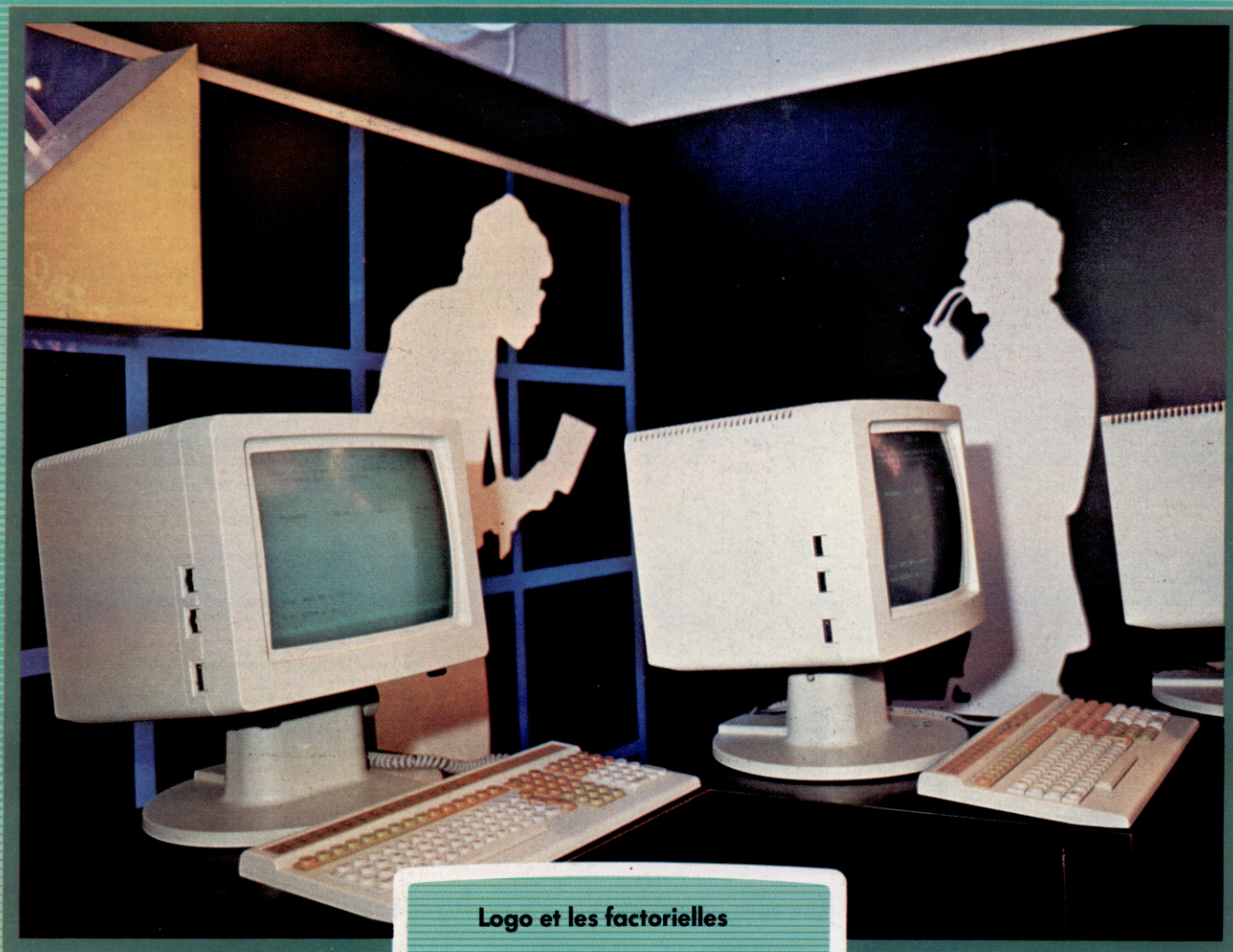


abc

N° 72

COURS
D'INFORMATIQUE
PRATIQUE
ET FAMILIALE

INFORMATIQUE



Logo et les factorielles

Un logiciel pour décideurs

Jeu Digitaya sur le C64

Le Memotech RS128

EDITIONS
ATLAS



Bien commencer

Nous allons voir les principes généraux que le programmeur doit observer dans la conception de programmes destinés à des enfants des écoles maternelles et primaires.

Les logiciels éducatifs étudiés ici sont d'origine américaine. Ils vont de la simple reconnaissance des formes et des couleurs à des notions de base pour la lecture et le calcul, et jusqu'à des expériences complexes d'éveil artistique de l'enfant.

Ces « didacticiels » ont une priorité absolue : l'enfant doit comprendre parfaitement ce que l'on attend de lui et recevoir des récompenses bien définies lorsqu'il est parvenu à maîtriser une démarche intellectuelle.

En outre, les programmes doivent être facilement utilisables. Un programme destiné à apprendre à lire ne peut pas commencer par une liste d'instructions de mise en œuvre. Le meilleur réduira les instructions à leur strict minimum.

Un programme éducatif doit également maintenir l'intérêt de l'enfant. Son importance et son enjeu final passent après son accessibilité et son intérêt pour l'enfant, faute de quoi, devenant répétitif et ennuyeux, il perdrait son attrait.

Finalement, le plus important est de savoir s'il apporte bien l'enseignement désiré. Cela peut paraître évident, mais en réalité, beaucoup d'éditeurs de didacticiels oublient le projet éducatif au profit du seul aspect divertissement.

Les logiciels étudiés ici sont produits par les firmes Spinnaker et Fisher-Price. Ils se présentaient à l'origine sous forme de cartouches, les jeunes enfants pouvant ainsi facilement apprendre à les manier. Le format cassette, dans lequel ils sont vendus parfois en Europe, suppose en revanche l'intervention d'un adulte.

Le programme le plus intéressant est probablement *Dance Fantasy* de Fisher-Price. Il est destiné aux enfants de quatre à huit ans. L'écran

Dance Fantasy



représente une scène où se tiennent deux silhouettes. L'utilisateur doit monter une chorégraphie. Avant de commencer, il doit préciser le sexe des danseurs : un garçon et une fille, deux garçons ou deux filles.

Au bas de l'écran, le programme affiche un ensemble de silhouettes dans différentes poses. Chacune représente une séquence de danse particulière, saut, gigue, etc. En déplaçant à l'aide de la manette de jeu un des personnages sur une des silhouettes, l'enfant sélectionne le pas et le programme correspondants. Il doit ensuite positionner le personnage sur la scène et faire exécuter le mouvement en appuyant sur le bouton d'action. Une chorégraphie est ainsi organisée en choisissant une suite de mouvements exécutés en différents points de l'écran; des mouvements d'enchaînements fournis par le programme assu-

Voyage d'Égée



rent l'harmonie de l'ensemble. Une fois la danse orchestrée, l'enfant la sauvegarde (SAVE), et peut ensuite visionner l'effet d'ensemble.

Comme vous l'avez probablement compris, l'intérêt de *Dance Fantasy* tient à ce qu'il présente, de manière très imaginative, une analogie avec un programme informatique. L'enfant est en mesure de créer sa propre danse (un programme) à l'aide d'un ensemble de routines (jeu de procédures). Le tout est ensuite sauvegardé (SAVE) et ensuite chargé à partir de la cassette (LOAD). L'enfant se familiarise ainsi sans peine avec ces deux notions.

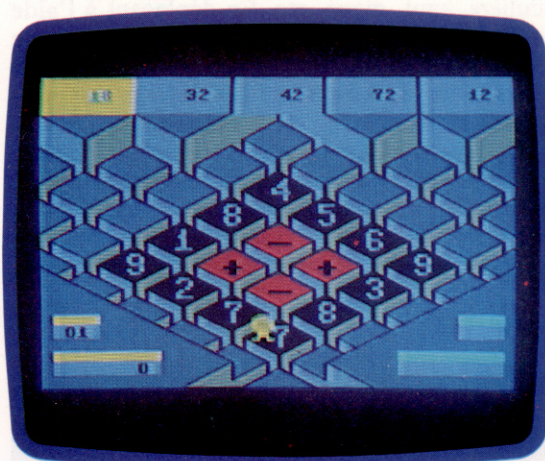
Voyage d'Égée (Spinnaker) est destiné à des enfants légèrement plus âgés. Il met en scène des personnages et des lieux appartenant à la mytho-



logie grecque, comme éléments d'un jeu d'aventures de type simple. L'objectif est de piloter un navire depuis Athènes jusqu'aux îles de la mer Égée, en évitant constamment les écueils et les tempêtes. En arrivant à bon port, le nom de l'île est affiché, et un message codé apparaît en bas de l'écran. Le joueur doit ensuite décider s'il va explorer l'île. Une décision correcte sera récompensée par la découverte d'un trésor tel que le bouclier d'Achille; une erreur dans l'interprétation de l'énigme, et le navire coulera sous les coups d'une créature mythique...

Les érudits seront peut-être agacés par les erreurs du jeu : par exemple, le Minotaure peut apparaître aussi bien à Délos qu'en Crète. Le jeu n'essaie pas d'expliquer les lieux et les noms; un enfant n'en retirera aucune connaissance sur la littérature classique. Et, en outre, il y a peu de

Number Tumblers

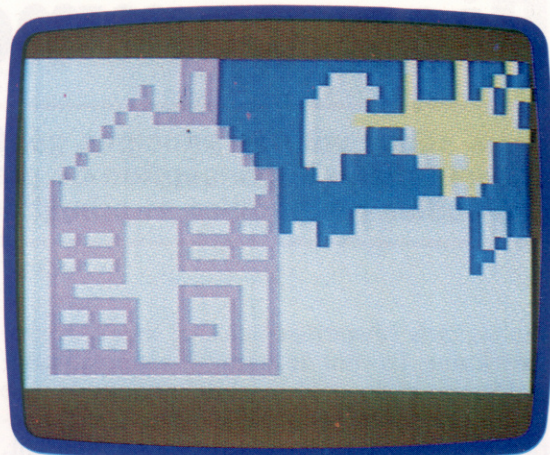


chances que le jeu parvienne à maintenir durablement l'intérêt de l'enfant, le graphisme et le format étant sans charme et répétitifs.

Number Tumblers (Fisher-Price) est destiné à développer le calcul mental chez les 8-12 ans; sa rapidité lui fait ressembler à un jeu d'aventures. Une suite de nombres s'affiche en haut de l'écran. Le joueur doit disposer des symboles numériques et arithmétiques sur les faces des dés, afin de créer une expression mathématique égale à l'un des nombres affichés. Le jeu est rapide et amusant, son graphisme est agréable et bien conçu; il devrait être très stimulant pour les exercices de calcul mental.

Kindercomp (Spinnaker) est destiné aux enfants de 3 à 8 ans. Son but est de familiariser les plus jeunes aux techniques de l'ordinateur en développant leur sens artistique. Plusieurs exercices sont proposés. Malheureusement, Kindercomp semble être principalement constitué de « trucs » informatiques, comme ceux que l'on crée en apprenant le BASIC et en découvrant les possibilités d'une machine. Par exemple, l'une des options est « noms ». L'utilisateur est invité à donner un nom ou une phrase courte allant jusqu'à quinze caractères. Celle-ci est ensuite exécutée à l'écran sous forme de couleurs assorties à des éléments géométriques de tailles variables.

Kindercomp



L'effet est très attrayant, et un enfant habitué à l'ordinateur sera complètement fasciné. Mais le jeu est de peu de valeur éducative; n'importe quel groupe de lettres donne le même effet et l'enfant ne découvre rien par lui-même.

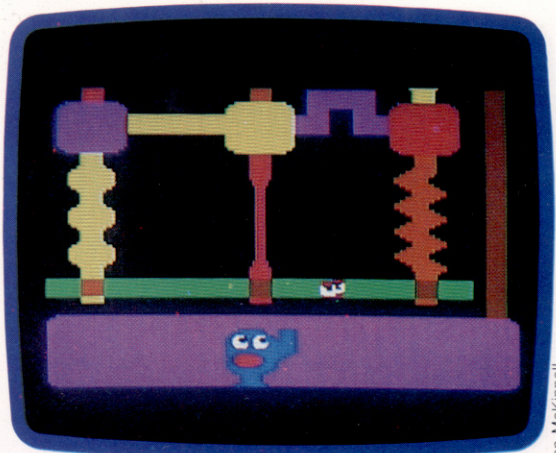
Il est peu probable que Kindercomp puisse intéresser longtemps un enfant. Les trucs de programmation sont amusants mais deviennent vite répétitifs. C'est le genre de progiciels qui occupent un enfant pendant trois jours environ et ne sont jamais plus utilisés par la suite.

Alf In The Colour Caves, pour les très jeunes enfants, représente un petit personnage amusant qui s'introduit dans des tubes de couleurs de formes différentes qui glissent en direction d'une cave. A l'aide de la manette de jeu ou du clavier, l'enfant les guide dans les diverses pièces.

Si le personnage est guidé à bon port en échappant à une paire d'yeux menaçants et rapides, le joueur est récompensé en voyant le spectacle d'un elfe (Alf) exécutant une danse pleine de charme. L'elfe est alors placé dans un tube pour remonter à la surface et ensuite, il recommence une autre descente.

Ces quelques exemples ne sont qu'une très faible représentation de ce qui se fait aujourd'hui en matière de didacticiels ou de programmes éducatifs destinés aux jeunes enfants. Nous aurons l'occasion d'en reparler, étant donné la place de plus en plus grande qu'ils prennent.

Alf In The Colour Caves



Ian McKinnell



Mesure pour mesure

Nous revenons à notre projet de robot pour concevoir un programme qui lui permettra de localiser et de mesurer précisément un côté d'un objet rectangulaire.

Pour permettre à notre robot de localiser et de mesurer le côté d'un objet, nous avons besoin d'un programme assez évolué. Le robot détectera l'objet à l'aide des micro-interrupteurs que nous avons posés précédemment. Voici une méthode possible pour accomplir cette tâche :

1. Trouver l'objet.
2. Trouver une extrémité du côté repéré.
3. Parcourir le côté de l'objet jusqu'à ce que l'autre extrémité soit atteinte.

La première étape peut être réalisée facilement si nous supposons que, lorsque le programme démarre, le robot se dirige vers le côté de l'objet que nous désirons mesurer. Le principal problème à résoudre est le risque que le robot touche une extrémité du côté avec un seul capteur au lieu des deux. Les diverses possibilités sont illustrées dans le diagramme. Cependant, si un seul capteur avant est fermé, il est possible de détecter s'il s'agit de celui de gauche ou de droite; cela nous permet de mettre au point une stratégie.

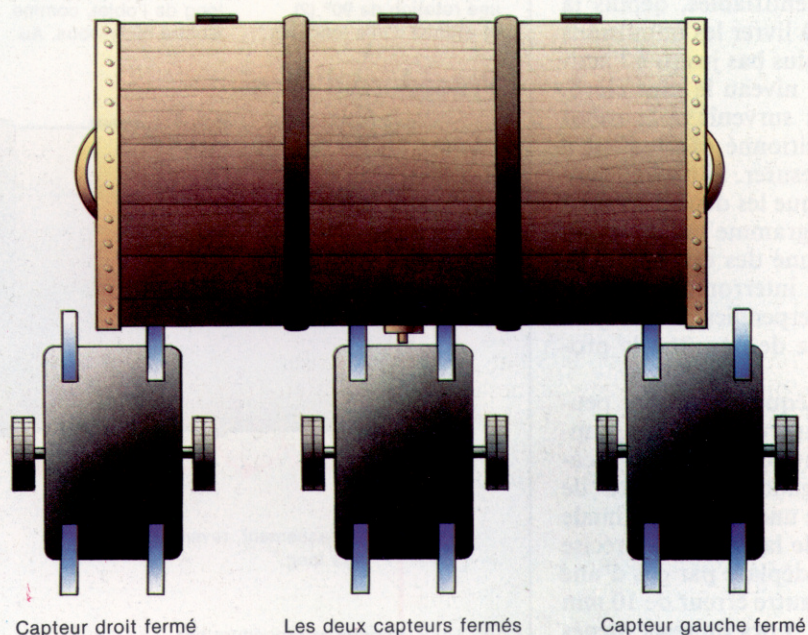
Nous devons également supposer que le robot est positionné initialement à 90° du côté de l'objet à mesurer. Nous pouvons ainsi ne pas tenir compte du cas — éventuel — où le robot rencontrerait l'objet de façon oblique.

La deuxième étape de notre méthode se trouve simplifiée car le robot se dirige toujours vers le côté droit de l'objet avant de commencer à le mesurer. Pour localiser l'extrémité de droite, le robot doit « sentir » sa progression le long du côté, en se déplaçant par étapes vers la droite jusqu'à ce que seul le contact de gauche (et non les deux) soit fermé. Pour se déplacer le long du côté, le robot doit effectuer une série de manœuvres compliquées — chacune implique cinq mouvements. En supposant que le robot soit initialement en contact avec le côté de l'objet, il doit d'abord reculer légèrement, effectuer une rotation de 90° , avancer d'une certaine distance, effectuer une rotation inverse de 90° et finalement avancer jusqu'à ce que ses capteurs rencontrent de nouveau l'objet. Le diagramme illustre toutes les étapes de la manœuvre. La « longueur de pas » (la distance entre deux points de contacts successifs) est l'équivalent de celle pendant laquelle le robot se déplace parallèlement au côté de l'objet.

Pour localiser précisément le côté droit de l'objet, le robot pourrait se déplacer le long de l'objet par pas de quelques millimètres. Mais ce serait trop lent. Nous utiliserons donc des pas

Détection

Ce schéma illustre les trois cas qui peuvent se présenter lorsque les capteurs entrent en contact avec le côté d'un objet. Lorsque le capteur droit seul est fermé, le robot a détecté l'extrémité gauche du côté; si les deux capteurs sont fermés, il « sait » qu'il se trouve quelque part au milieu; si le capteur gauche est seul fermé, il a atteint le côté droit de l'objet.
(Cl. Kevin Jones.)





plus longs jusqu'à ce que l'extrémité soit atteinte. Là, le robot reviendra en arrière en se déplaçant cette fois par pas plus petits pour localiser précisément l'extrémité droite de l'objet. Une longueur de pas appropriée sera égale à la distance qui sépare les deux capteurs avant — environ 60 mm — puisque tout dépassement sera détecté par la fermeture d'un seul capteur.

La troisième étape utilise une méthode similaire; mais, cette fois, le robot se déplace vers la gauche en comptant le nombre de pas jusqu'à ce que l'extrémité gauche soit atteinte. A la fin de cette étape, la longueur du côté de l'objet est mémorisée et peut être imprimée.

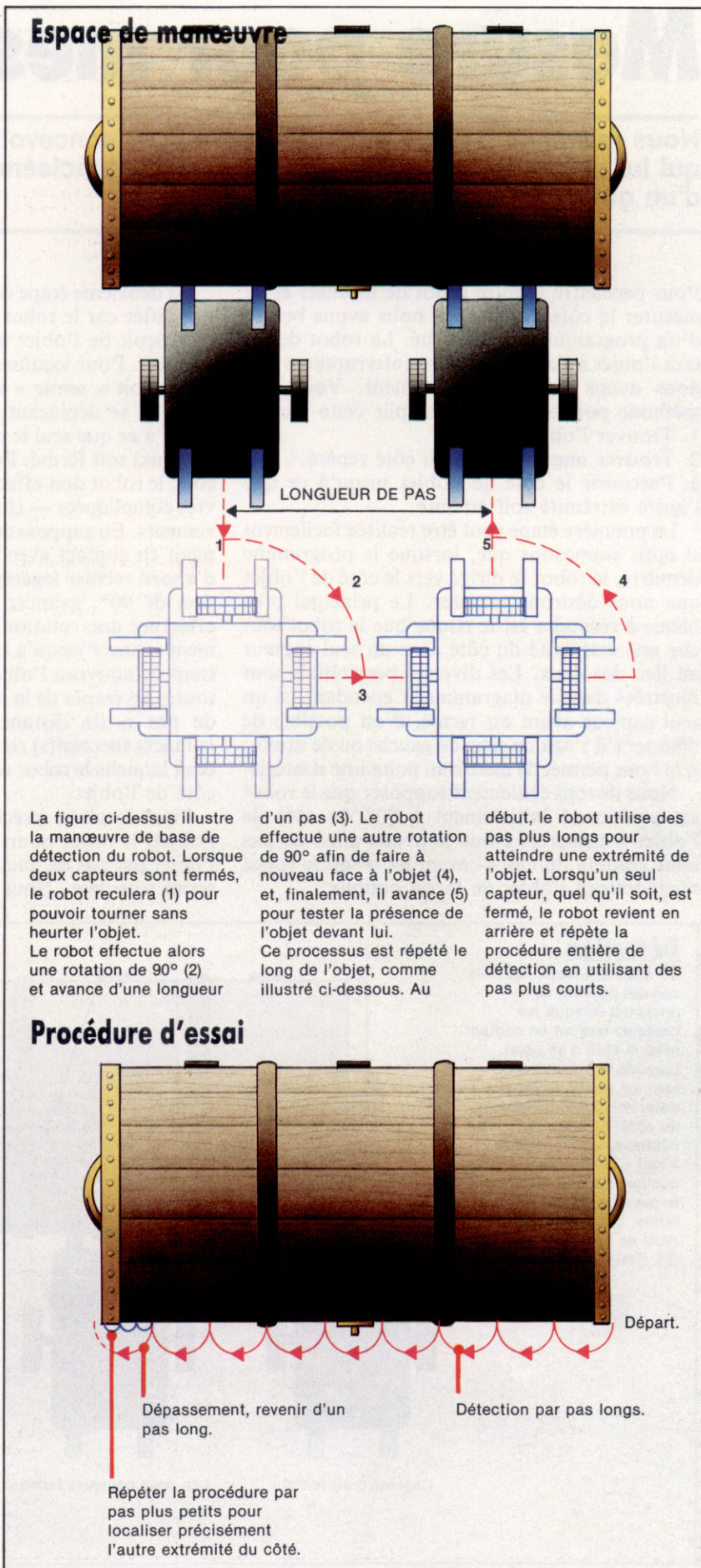
Programme de mesure

Nous donnons les listages pour le Commodore 64 et le BBC Micro. Les rapports impulsion/distance et impulsion/angle — trouvés de façon expérimentale auparavant — devraient être introduits pour votre propre robot. Les procédures employées par le BBC BASIC sont idéales pour écrire un programme de ce type. Nous pouvons adopter une approche très structurée pour ce problème, en contrôlant chaque mouvement du robot au moyen de procédures distinctes. Deux caractéristiques du BBC BASIC — noms de variables étendus et transfert de paramètres entre les procédures — signifient que le programme peut ressembler de très près à notre façon de penser. La version Commodore peut adopter la même approche structurée, mais notez à quel point la programmation structurée est plus difficile.

Après avoir isolé les principales tâches que le programme doit effectuer, nous pouvons concevoir des procédures individuelles pour déplacer le robot et combiner une série de manœuvres afin de créer une procédure de « détection ». Dans cette application, les différents niveaux de procédure sont facilement identifiables, depuis la simple procédure servant à livrer les impulsions aux moteurs au niveau le plus bas jusqu'à l'activité globale de mesure au niveau le plus élevé.

Des problèmes peuvent survenir si le robot n'est pas initialement positionné exactement à 90° du côté de l'objet à mesurer. Si un seul capteur établit le contact lorsque les deux devraient le faire, la logique du programme indiquera au robot qu'il se trouve à l'une des extrémités de l'objet. Si tel est le cas, interrompez le programme, alignez le robot perpendiculairement au côté à mesurer et exécutez de nouveau le programme depuis le début.

Plusieurs erreurs intrinsèques de mesure peuvent être identifiées. La largeur de chaque capteur, par exemple, est d'environ 5 mm. La localisation des extrémités gauche et droite de l'objet peut donc produire une erreur maximale de 10 mm. En plus, lors de la détection précise des extrémités, le robot se déplace par pas d'une longueur de 5 mm, et une autre erreur de 10 mm peut alors être introduite. Lors de tests menés avec notre robot prototype, la moyenne d'erreur en mesurant un objet de 410 mm de côté était d'environ 20 mm — une erreur de 5 % seulement.





Listage BBC Micro

```

1000 REM **** Mesure Robot BBC ****
1010 MODE 7
1020 PROCinitialise
1030 PROCmeasure
1040 PROCprintout
1050 END
1060 DEF PROCmeasure
1070 PROCfind
1080 REM ** Un seul capteur ? **
1090 PROCtest_bumpers
1100 REM **** Trouver extrémité ****
1110 REPEAT:PROCprobe(right,width)
1120 UNTIL (?DATREG AND 192)=left_bumper
1130 REM ** Retour en arrière **
1140 PROCprobe(left,width)
1150 REPEAT:PROCprobe(right,small_width)
1160 UNTIL (?DATREG AND 192)=left_bumper
1170 PRINT"FOUND RIGHT-HAND END"
1180 PRINT"STARTING TO MEASURE"
1190 REM ** Commencer à mesurer **
1200 count=width
1210 REPEAT:PROCprobe(left,width)
1220 count=count+width
1230 UNTIL (?DATREG AND 192)=right_bumper
1240 REM ** Retour en arrière **
1250 count=count-width
1260 PROCprobe(right,width)
1270 REPEAT:PROCprobe(left,small_width)
1280 count=count+small_width
1290 UNTIL (?DATREG AND 192)=right_bumper
1300 ?DATREG=0
1310 ENDPROC
1320 :
1330 DEF PROCprintout
1340 CLS
1350 PRINTTAB(5,12)"OBJECT SIDE MEASURED AT
    "count;" mm"
1360 ENDPROC
1370 :
1380 DEF PROCinitialise
1390 DDR=&FE62:DATREG=&FE60
1400 ?DDR=15:REM LINES 0-3 OUTPUT
1410 ?DATREG=1:REM TURN ON RESET BIT
1420 forwards=4:backwards=2:left=5:right=0
1430 pd_ratio=3,34446:pa_ratio=375/90
1440 right_bumper=128:left_bumper=64
1450 both_bumper=0:neither_bumpers=192
1460 width=60:small_width=5
1470 ENDPROC
1480 :
1490 DEF PROCsearch(sense)
1500 REPEAT:PROCprobe(sense,width)
1510 UNTIL (?DATREG AND 192)=both_bumpers
1520 ENDPROC
1530 :
1540 DEF PROCfind
1550 REPEAT:PROCmove(forwards,8)
1560 UNTIL (?DATREG AND 192)<neither_bumpers
1570 ENDPROC
1580 :
1590 DEF PROCtest_bumpers
1600 IF (?DATREG AND 192)=right_bumper THEN
    PROCsearch(right):ENDPROC
1610 IF (?DATREG AND 192)=left_bumper THEN
    PROCsearch(left):ENDPROC
1620 ENDPROC
1630 :
1640 DEF PROCprobe(way,step)
1650 IF way=right THEN opp_way=left ELSE
    opp_way=right
1660 PROCmove(backwards,30)
1670 PROCturn(way,90)
1680 PROCmove(forwards,step)
1690 PROCturn(opp_way,90)
1700 REPEAT:PROCmove(forwards,8)
1710 UNTIL (?DATREG AND 192)<neither_bumpers
1720 ENDPROC
1730 :
1740 DEF PROCmove(dir,distance)
1750 ?DATREG=(?DATREG AND 1)OR dir
1760 pulses=pd_ratio*distance
1770 FOR I=1 TO pulses:PROCpulse:NEXT I
1780 ENDPROC
1790 :
1800 DEF PROCturn(dir,angle)
1810 ?DATREG=(?DATREG AND 1)OR dir
1820 pulses=pa_ratio*angle
1830 FOR I=1 TO pulses:PROCpulse:NEXT I
1840 ENDPROC
1850 DEF PROCpulse
1860 ?DATREG=(?DATREG OR 8)
1870 ?DATREG=(?DATREG AND 247)
1880 ENDPROC

```

Listage Commodore 64

```

10 REM **** MESURE ROBOT CBM ****
20 GOSUB1000:REM INITIALISER
30 GOSUB2000:REM MESURER
40 GOSUB3000:REM AFFICHER
50 END
60 :
1000 REM **** INITIALISER ****
1010 DDR=56579:DATREG=56577
1020 POKE DDR,15:REM SORTIE DES LIGNES 0-3
1030 POKE DATREG,1:REM METTRE BIT DE REMISE A ZERO
1040 FW=4:BW=2:LF=6:RT=0
1050 PQ=3,34446:PA=375/90
1060 RB=128:LB=64:SB=0:NB=192
1070 WD=60:BW=5
1080 RETURN
1090 :
2000 REM **** MESURER ****
2010 GOSUB3500:REM TROUVER OBJET
2020 GOSUB4000:REM TESTER CAPTEURS
2030 REM ** TROUVER EXTREMITÉ **
2040 WY=RT:SP=WD:GOSUB6000:REM DETECTER
2050 IF(PEEK(DATREG)AND192)<LB THEN 2040
2060 REM ** GO BACK AND INCH TO END **
2070 DR=LF:DS=WD:GOSUB6000:REM DETECTER
2080 DR=RT:DS=SW:GOSUB6000:REM DETECTER
2090 IF(PEEK(DATREG)AND 192)<LB THEN 2080
2100 PRINT"FOUND RIGHT-HAND END"
2110 PRINT"STARTING TO MEASURE"
2120 REM ** DEBUT DES MESURES **
2130 CC=WD
2140 DR=LF:DS=WD:GOSUB6000:CC=CC+WD
2150 IF(PEEK(DATREG)AND192)<RB THEN 2140
2160 REM ** RETOUR EN ARRIERE **
2170 CC=CC-WD
2180 DR=RT:DS=WD:GOSUB6000:CC=CC+SW
2190 IF(PEEK(DATREG)AND192)<RB THEN 2180
2200 POKE DATREG,0
2210 RETURN
2220 :
3000 REM **** AFFICHER ****
3010 PRINTCHR$(147)
3020 PRINT"OBJECT MEASURED AT I:"CC;"MM"
3030 RETURN
3040 :
3500 REM **** TROUVER ****
3510 DR=FW:DS=5:GOSUB7000:REM DEPLACER
3520 IF(PEEK(DATREG)AND 192)=NB THEN 3510
3530 RETURN
3540 :
4000 REM **** TESTER CAPTEURS ****
4010 IF(PEEK(DATREG)AND 192)=RB THEN SS=RT:
    GOSUB5000:RETURN
4020 IF(PEEK(DATREG)AND 192)=LB THEN SS=LF:
    GOSUB5000:RETURN
4030 RETURN
4040 :
5000 REM **** CHERCHER (SS) ****
5010 DR=FW:DS=8:GOSUB7000:REM DEPLACER
5020 IF(PEEK(DATREG)AND192)=NB THEN 5010
5030 RETURN
5040 :
6000 REM **** DETECTER (WY,SP) ****
6010 IF WY=RT THEN DW=LF
6020 IF WY=LF THEN DW=RT
6030 DR=BW:DS=30:GOSUB7000:REM DEPLACER
6040 DR=WY:AG=90:GOSUB7500:REM TOURNER
6050 DR=FW:DS=SP:GOSUB7000:REM DEPLACER
6060 DR=DW:AG=90:GOSUB7500:REM TOURNER
6070 DR=FW:DS=8:GOSUB7000:REM DEPLACER
6080 IF(PEEK(DATREG)AND192)=NB THEN 6070
6090 RETURN
6100 :
7000 REM **** DEPLACER (DR,DS) ****
7010 POKE DATREG,(PEEK(DATREG)AND 1)OR DR
7020 PL=PD+DS
7030 FOR I=1 TO PL:GOSUB8000:NEXT I
7040 RETURN
7050 :
7500 REM **** TOURNER (DR,AG) ****
7510 POKE DATREG,(PEEK(DATREG)AND 1)OR DR
7520 PL=PA+AG
7530 FOR I=1 TO PL:GOSUB8000:NEXT I
7540 RETURN
7550 :
8000 REM **** IMPULSION ****
8010 POKE DATREG,PEEK(DATREG)OR 8
8020 POKE DATREG,PEEK(DATREG)AND 247
8030 RETURN

```



Sur le vif

Examinons un ensemble de programmes qui s'appuient sur l'évaluation psychologique des personnes pour aider les décideurs à adopter une stratégie dans leurs rapports professionnels.

Human Edge Software (« L'aspect humain »), progiciel d'un éditeur californien, constitue un nouveau pas vers l'intelligence artificielle. Les programmes traitent d'importantes quantités de données fournies par l'utilisateur et les évaluent selon des critères préalablement enregistrés. Ils concluent en fournissant la « conduite idéale » à suivre. Human Edge comprend quatre programmes — Communications Edge (« L'aspect communication »), Sales Edge (« L'aspect ventes »), Management Edge (« L'aspect gestion »), et Negotiation Edge (« L'aspect négociations ») — destinés à l'IBM PC et à ses compatibles. Le jeu complet de programmes revient à plus de 10 000 F et est destiné à améliorer la qualité professionnelle des individus dans les différents domaines précités. Une version simplifiée a été commercialisée pour le Commodore 64, l'Apple II et le Macintosh.

Les quatre programmes d'origine sont le résultat de plus de dix années de recherches sur le comportement humain. Les techniques les plus récentes, telles l'analyse des facteurs humains, la technologie des systèmes experts et les théories mathématiques de prises de décision, ont été utilisées.

Cette description qui peut sembler rébarbative ne doit pas repousser l'utilisateur potentiel. Les modules sont faciles d'utilisation et se révèlent pleinement exploitables après une heure de pratique. Ils sont pilotés par menus et construits sur la base de questionnaires. Ces derniers consistent en une suite d'affirmations très nuancées que l'utilisateur doit confirmer ou infirmer. Ces énoncés cherchent à cerner la personnalité de l'utilisateur. Ils lui demandent aussi d'indiquer ses perspectives de vente, ses clients courants, ses employés et supérieurs hiérarchiques, ainsi que l'aspect particulier des rapports professionnels qu'il désire voir se développer.

Les réponses sont évaluées par le programme qui fournit un rapport détaillé incluant une conduite à suivre. Il peut s'agir d'une proposition d'affaires à un nouveau client, d'une rupture d'affaires en raison de perspectives de ventes maussades, ou encore d'une stratégie à des fins de négociations avec ses subordonnés ou avec ses employeurs.

Chaque programme débute par un questionnaire où l'utilisateur doit s'auto-évaluer en répondant par oui ou par non à des affirmations du genre : « Dans les réunions, je prends souvent de nouvelles responsabilités », ou « Je suis assez véhément et engage des polémiques plus fréquem-

ment que mes collègues », ou encore « Je suis impulsif », etc. L'utilisateur doit se reconnaître parmi ces divers comportements. Cet outil d'auto-évaluation a été préparé avec soin, et de nombreuses questions se recoupent pour garantir la validité du programme. Ainsi les réponses « Je suis impulsif », et « Il m'arrive d'agir sans avoir réfléchi » sont confrontées pour s'assurer de la cohérence des propos. Lorsque cette partie du programme est finie, les réponses sont sauvegardées sur disque. Elles peuvent être mises à jour et réutilisées à tout moment.

Une fois l'auto-évaluation faite, une suite d'adjectifs est soumise à l'utilisateur qui doit retenir ceux qu'il pense le mieux se rapporter à l'objet de sa recherche. Des qualificatifs tels que « bavard », « inquiet », « indépendant », « performant », « ambitieux », « courtois », « crâneur », ou « énergique » sont proposés pour l'aider à apprécier le client, le supérieur ou le subordonné. Le sens et la nuance de chaque terme doivent être bien pesés pour ne pas induire le programme en erreur par une mauvaise appréciation de la « cible ».

L'utilisateur peut, à tout moment, faire défiler à l'écran la liste des adjectifs et modifier ces derniers. Comme pour l'auto-évaluation, la liste





est sauvegardée sur disque et peut être mise à jour ultérieurement.

L'auto-évaluation est effectuée une fois pour toutes; elle peut être, par la suite, mise en relation avec toute autre liste complémentaire. Huit listes complémentaires d'évaluation au maximum sont susceptibles d'être sauvegardées. Lorsque le programme dispose de deux listes complètes, il évalue leurs réponses et en tire un bilan sur les « qualités » du sujet et de sa « cible ».

Le bilan qui suit a été créé par Communication Edge et correspond à une enquête portant sur un utilisateur réel et sa contrepartie, ici le fils de l'utilisateur — un adolescent. Ce dernier est appelé par l'ordinateur « M. A... » (A pour adolescent). Le rapport se présente comme un discours en style direct adressé à l'utilisateur :

« Votre ouverture d'esprit et votre humeur égale seront bien nécessaires pour communiquer avec M. A... Ce dernier est très renfermé et préfère la solitude à la sociabilité et aux bavardages. Vous devez vous attendre de sa part à une attitude cynique ou soupçonneuse lorsque vous solliciterez son avis et son sentiment. Lorsque vous vous adressez à lui, ne pensez pas qu'il éprouve de la compréhension à votre égard. Soyez clair, concis et direct. »

« Contrairement à votre tempérament égal et paisible, M. A... se met vite en colère et peut même paraître hostile avant même que la conversation ne soit commencée. Il se peut qu'il essaie de vous imposer ses opinions. Restez aimable malgré son attitude. Soyez également prêt à affronter ses revirements d'humeur. Il est capable de parler de manière impulsive pendant un moment, puis de bien choisir et peser ses mots. Prenez l'initiative du déroulement de l'entretien. Paraphraser ses commentaires pour que votre propos lui soit plus clair et obtienne son assentiment. »

Les parents reconnaîtront probablement le portrait d'un adolescent, même si le module Communication Edge ne pose jamais la question de l'âge de la personne analysée (le sexe l'est, en revanche).

Le vocabulaire utilisé vous rappellera le style de certains magazines qui parlent de la personnalité ou vous proposent des tests pour « mieux vous connaître ». L'utilisateur est généralement décrit en termes positifs (« d'humeur égale », « souple », « stable »). La « cible » en revanche, n'est pas ménagée (« soupe au lait », « imprévisible », « cynique », « méfiant »). Cette disparité a probablement pour objet de renforcer la confiance de l'utilisateur. Elle correspond aussi à cette conception américaine selon laquelle le monde des affaires correspond à une guerre où, dans les prévisions de vente, les clients sont les ennemis dont il faut vaincre la résistance.

Cette mentalité devient encore plus manifeste dans le module « L'aspect négociations » qui suggère des stratégies du genre (affichée en lettres majuscules) :

UTILISEZ À VOTRE AVANTAGE LA CONNAISSANCE QUE VOUS AVEZ DE M. A.
 COMMENCEZ PAR SATISFAIRE M. A PAR DES CONCESSIONS PRÉMATURÉES.
 HABITUEZ M. A À VOUS RÉPONDRE PAR « OUI ».
 VOILEZ VOS MENACES.
 EXAGÉREZ VOS PROBLÈMES.
 MINIMISEZ VOS GAINS.
 CONCLUEZ DE MANIÈRE POSITIVE.

Il semble que les concepteurs de ces modules aient supposé qu'un utilisateur donné n'en acquerrait jamais qu'un seul. En effet, les quatre programmes supposent chacun de répéter l'épreuve d'auto-évaluation avec à peu près les mêmes questions dans un ordre légèrement différent. Il est effectivement probable, vu le prix, que l'utilisateur moyen fera un choix définitif sur l'un des modules, alors que ces produits ne conviendront pas aux grosses sociétés en tant qu'outils de gestion. Le même test devrait pouvoir servir aux quatre modules. Néanmoins, ces derniers sont tellement similaires qu'il suffit d'acheter le moins cher, « L'aspect communication », et d'adapter ses bilans aux diverses situations.

Le grave défaut de ces programmes est leur incapacité à tenir compte des expériences réalisées. Cela réduit sérieusement leur utilisation en tant qu'outils de gestion. En effet, ils ne prennent pas en considération les mises à jour personnelles de l'utilisateur relatives aux énoncés proposés. Il serait par exemple très utile de faire figurer au programme le résultat pratique d'une stratégie, afin que cette dernière puisse être modifiée en conséquence. Cela serait précieux tout particulièrement dans le cas où l'on connaît peu de choses du sujet étudié. En outre, il est difficile de répondre de manière tranchée, par oui ou par non.

Finalement, on croit ou non à ce genre d'approche des relations humaines. L'utilisation la plus profitable correspondrait peut-être à la préparation de futurs entretiens. L'utilisateur devra pourtant être attentif à ne pas prendre les conseils de Human Edge (« L'aspect humain ») trop au sérieux. Jusqu'à ce que l'ordinateur devienne vraiment une machine pensante...

HUMAN EDGE

Ensemble de quatre progiciels (pouvant être achetés séparément) pour machines MS-DOS et compatibles PC.

Auteurs

Human Edge Software California.

Format

Disque.

MIND PROBER

Pour Commodore 64, Apple II et Macintosh.

Auteurs

Les mêmes.

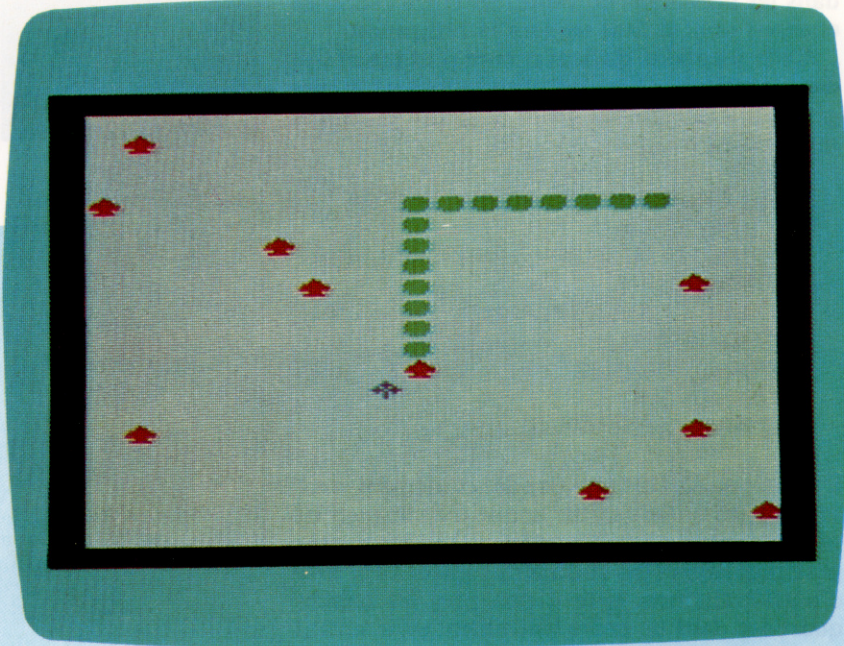
Format

Disque.



Micropede

A petits pas rapides, Pierre Monsaut a écrit ce jeu en BASIC pour l'ordinateur domestique Vic 20. Un conseil : enregistrez-le avant de le faire tourner.



Essayez de diriger votre mille-pattes robot le plus longtemps possible. Il doit se nourrir de fleurs bleues (les rouges sont empoisonnées), sans jamais sortir du cadre ou recouper son propre corps. La difficulté vient de ce que sa longueur augmente d'une unité à chaque repas, ce qui rend les déplacements de plus en plus délicats.

```

10 REM *****
20 REM * MICROPEDE *
30 REM *****
50 GOSUB 1000
100 GET X$
110 D1=(X$="A")-(X$="S")+22*((X$="W")-(X$="Z"))
120 IF D1<>0 THEN D0=D1
125 IF FL=1 THEN FL=0:GOTO 170
130 POKE A(0),CN
140 FOR I=0 TO L
150 A(I)=A(I+1)
160 NEXT I
170 A(L)=A(L-1)+D0
180 C=PEEK(A(L))
190 IF C=CB THEN 300
200 IF C<>32 AND C<>42 THEN 500
210 POKE A(L),MP
220 POKE A(L)+M,MC
230 GOTO 100
300 GOSUB 2000
320 POKE A(L),MP
330 POKE A(L)+M,MC
340 L=L+1
350 FL=1
360 GOTO 100
500 PRINT:PRINT:PRINT
505 GOSUB 700
510 PRINT TAB(5)"SCORE : "L*10-70
520 PRINT:PRINT:PRINT

```

```

530 PRINT TAB(5)"UNE AUTRE ?"
540 GET X$
550 IF X$<>" " THEN 540
560 GET X$
570 IF X$=" " THEN 560
580 IF X$<>"N" THEN RUN
590 PRINT CHR$(147);
600 END
700 FOR I=1 TO 6
710 POKE A(L),CN
730 FOR J=1 TO 200
740 NEXT J
750 POKE A(L),42
760 POKE A(L)+M,4
770 FOR J=1 TO 200
780 NEXT J
790 NEXT I
800 RETURN
1000 PRINT CHR$(147);
1010 GOSUB 1200
1020 GOSUB 1400
1030 GOSUB 1600
1040 GOSUB 1800
1050 GOSUB 2000
1190 RETURN
1200 CN=32
1210 CH=65
1220 HC=2
1230 BC=6
1240 DIM A(70)

```

```

1250 M=30720
1260 MP=81
1270 MC=5
1280 L=7
1290 D0=1
1300 CB=88
1390 RETURN
1400 FOR I=0 TO L
1410 A(I)=7923+I
1420 POKE A(I),MP
1430 POKE A(I)+M,MC
1440 NEXT
1450 RETURN
1600 FOR I=0 TO 21
1610 POKE 7680+I,160
1620 POKE 7680+I+M,0
1630 POKE 8164+I,160
1640 POKE 8164+I+M,0
1650 NEXT
1660 FOR I=1 TO 22
1670 POKE 7680+I*22,160
1680 POKE 7680+I*22+M,0
1690 POKE 7701+I*22,160
1700 POKE 7701+I*22+M,0
1705 NEXT
1710 FOR I=1 TO 10
1720 GOSUB 1800
1730 POKE P,CH
1740 POKE P+M,HC
1750 NEXT
1760 RETURN
1800 P=INT(RND(TI)*440)+7702
1810 IF PEEK(P)<>32 THEN 1800
1820 RETURN
2000 GOSUB 1800
2010 POKE P,CB
2020 POKE P+M,BC
2030 RETURN

```



Cartes supplémentaires

L'absence d'interfaces intégrées de lecteurs de disquettes avait compromis le succès de la série de micros Memotech 500. Memotech vient donc d'introduire le RS128 disposant de plusieurs interfaces.

Malgré une excellente réputation, la série de micro-ordinateurs Memotech 500 — MTX500 et MTX512 — a été un échec commercial relatif. Des caractéristiques séduisantes (les graphiques haute résolution, un assembleur intégré, un BASIC évolué et un langage unique de gestion de texte nommé NODDY) ne sont pas à l'origine de cette situation; le fait qu'elles n'aient pas réussi à provoquer un véritable succès populaire peut, d'une part, être attribué à la position bâtarde de ces machines entre deux segments du marché; d'autre part, ces machines sont un peu trop chères pour l'amateur de jeux qui peut estimer que les caractéristiques évoluées de cet ordinateur ne justifient pas son coût élevé. De plus, l'utilisateur « sérieux » (Memotech indique que cette série s'adresse à de petites applications de gestion) peut être déçu par l'absence d'interfaces intégrées qui lui auraient permis de connecter des lecteurs de disquettes. Ces interfaces étaient disponibles, mais, vendues en cartes séparées, elles devaient être branchées dans un connecteur plat situé à l'intérieur des machines. Ce comportement n'est pas surprenant de la part d'une société qui a bâti sa réputation sur la vente des cartes complémentaires pour le ZX81. Mais Memotech semble avoir reconnu son erreur et vient d'introduire le RS128, un ordinateur muni d'interfaces intégrées.

L'apparence de la machine

Au premier coup d'œil, le RS128 semble identique à la série 500 : il donne une impression d'élégance et de matériel haut de gamme. Comme chez ses frères, le boîtier est en aluminium, et non en plastique, ce qui rend la machine beaucoup plus lourde que la plupart des autres micros. L'ordinateur comporte un clavier QWERTY standard et un clavier numérique qui permet de solliciter certaines des commandes du langage de programmation NODDY. Il possède également huit touches de fonction programmables situées à la droite du clavier numérique. Le toucher du clavier est excellent.

La disposition des touches comporte certaines lacunes : la touche RETURN n'est pas beaucoup plus grosse que les autres et les dactylographes rapides auront peut-être du mal à la situer au départ; quant à la touche DELETE, elle n'est pas située sur le clavier lui-même, mais sur le clavier numérique. La touche de retour arrière est au coin supérieur droit, mais contrairement à ses concurrents, dont le retour arrière sert également



de touche d'effacement à gauche (retour arrière destructif), elle n'est qu'une simple touche de déplacement du curseur.

A l'arrière de la machine se trouvent plusieurs interfaces. Certaines étaient fournies avec la série 500, d'autres sont plus récentes. A l'extrême gauche, deux ports RS232 permettent de connecter des lecteurs de disquettes FDX à la machine. Ces ports peuvent aussi être utilisés à d'autres fins, comme connecter des imprimantes série ou établir un réseau de communication. A droite apparaît un jack vidéo composite et un jack hi-fi (ce dernier permet d'amplifier le son de l'ordinateur au moyen d'une chaîne hi-fi normale). Viennent ensuite la prise d'alimentation et un jack BF, suivis d'une interface de type Centronics. L'interface de cassette consiste en une paire de prises microjacks, « Écouteur » et « Micro ». On note finalement la présence d'une paire de ports de manche à balai de type Atari.

Les noms des ports d'interface sont inscrits en lettres blanches et peuvent être facilement lus à l'arrière de la machine. Cela aurait pu permettre de brancher les périphériques sans difficultés. Malheureusement, Memotech a placé les ports dans des renforcements qui obligent l'utilisateur à se pencher pour localiser les prises.

L'écran BASIC, de 24 par 40 caractères, est divisé en trois sections qui apparaissent

Beaucoup d'amélioration
Le Memotech RS128 est une version améliorée de la série MTX500. Ce nouveau modèle est muni de deux prises RS232 qui permettent à la machine d'utiliser les lecteurs de disquette FDX. Cela signifie que l'ordinateur est particulièrement intéressant pour l'utilisateur d'ordinateur domestique chevronné ou pour le petit utilisateur professionnel.
(Cl. Chris Stevens.)



RAM utilisateur

Le Memotech RS128 a 64 K de RAM que peut utiliser l'UC.

Modulateur BF

Ce dispositif produit un signal qui permet au RS128 de prendre en charge un écran de télévision.

Puce graphique

Ce dispositif est aussi utilisé dans les machines MSX.

Cartes d'extension

Ces cartes, qui sont optionnelles sur le 500 et sur le 512, font partie de l'équipement standard du RS128.

Interface cassette

Ces deux prises correspondent aux prises Ear et Mic d'un lecteur de cassette.

Ports de manche à balai

Ces ports permettent de connecter des manches à balai standard à l'ordinateur.

UC

Le RS128 utilise la puce Z80A de Zilog comme unité centrale de traitement.

RAM vidéo

Contrairement à de nombreux ordinateurs, les ordinateurs Memotech sont munis de leur propre RAM vidéo. Cela signifie que la RAM utilisateur n'est pas occupée par la mémoire écran.

Carte RS232

La carte RS232 gère les communications série de l'ordinateur. Elle permet à l'ordinateur d'utiliser le lecteur FDx et des modems.

Disquette silicium

Cette carte renferme 64 K de RAM supplémentaire. Cette RAM n'est pas directement accessible par l'UC (qui ne peut adresser plus de 64 K), mais agit comme une disquette externe. La vitesse d'accès est nettement plus grande.

Prise moniteur

Cette interface permet au RS128 de commander un moniteur vidéo composite.

clairement lors de la mise sous tension. Les dix-neuf lignes du haut forment l'écran principal où les listages défilent. Sous cet écran apparaît l'écran d'édition où les nouvelles lignes sont entrées. Au bas de l'écran, une ligne affiche les messages d'erreur. Les lignes de programme sont modifiées avec la commande EDIT. Ajoutons que le système d'exploitation n'autorise pas l'insertion d'une ligne dans le programme à partir de

l'écran EDIT si la ligne renferme une erreur de syntaxe. Le BASIC lui-même est très proche du BASIC MSX renfermant certaines commandes comme SOUND, PAPER, INK, et CIRCLE. Cependant, le BASIC renferme aussi certaines commandes très utiles non disponibles dans le BASIC MSX. Ces commandes sont dans l'ensemble reliées aux possibilités de gestion d'écran de la machine. A titre d'exemple, la commande CSRx,y positionne le

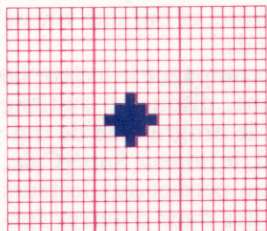
Grand écran

Nous avons déjà, pour notre jeu d'aventures Digitaya, créé des écrans graphiques illustrant des lieux traversés par le joueur. Voici une troisième version, destinée au Commodore 64.

Tuez-les tous !

Le port manette « tire » sur vous grâce à l'emploi d'un lutin qui sert de projectile. On POKE d'abord des zéros dans la zone définie, puis on crée le projectile proprement dit en POKant aux endroits désirés des données contenues en DATA.

PROJECTILE — LUTIN 1

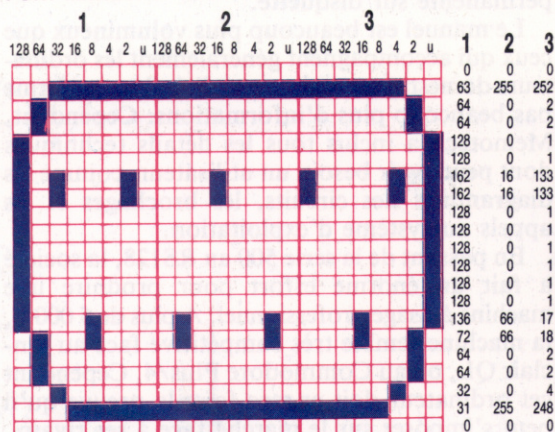


0	16	0
0	56	0
0	124	0
0	56	0
0	16	0

Extension de plan-objet

Une instruction READ lit les données contenues en DATA, qui sont ensuite POKées aux endroits indiqués. Le port manette paraît un peu serré; il est possible de l'étendre dans le sens horizontal (diagramme de droite) en changeant la valeur du registre qui contrôle cette extension.

PORT MANETTE — LUTIN 0



Les programmes conçus pour le Spectrum profitaient de la haute résolution et de commandes de formatage accessibles à partir du BASIC.

Pour le Commodore 64, il va falloir procéder d'une autre façon. Si lui aussi est capable de tracer des graphismes en haute résolution, celle-ci ne pourra être exploitée pleinement qu'en langage machine. Le BASIC « néolithique » du 64 ne comporte aucune commande spécifique pour obtenir de tels dessins; il faudrait procéder par accumulation de PEEKs et de POKes, ce qui est bien trop lent pour être d'une utilité quelconque. L'appareil offre en revanche certaines possibilités qu'il serait judicieux d'exploiter.

Il dispose, par exemple, de caractères graphiques à partir desquels on peut construire des lettres de grande taille, ou des éléments de décors. Il suffit, pour cela, de faire usage d'instructions PRINT, ou (mais cela est beaucoup moins rapide) de POKer à l'écran les codes correspondants. Par ailleurs, les plans-objets (ou lutins) peuvent se révéler très pratiques. Nous allons voir de plus près ces deux méthodes.

Le port manette

Les lignes 8020 à 8170 du listage consacré à l'illustration du port manette ont pour fonction de lire et de mettre en place les DATA des deux plans-objets utilisés dans cette routine. Le premier d'entre eux (plan-objet 0) est défini grâce aux soixante-trois nombres rassemblés dans les DATA compris entre les lignes 8450 et 8497. Il représente le port manette (voir diagramme). Ces DATA sont d'ordinaire placés très haut dans la zone de mémoire allouée aux programmes BASIC, mais, si ces derniers sont trop vastes, il est à craindre que nos données ne soient effacées. Il est par-

fois plus prudent de les stocker dans le tampon cassette (adresses 832 à 1022), qui peut gérer jusqu'à trois groupes de DATA affectés aux lutins. C'est ce que nous ferons ici.

Le plan-objet 0 est étiré dans le sens horizontal, jusqu'à atteindre deux fois sa longueur originale, en donnant au bit 0 du registre d'extension horizontale la valeur 1 (ligne 8170). Notez que tous les registres contrôlant les caractéristiques de ces lutins (couleur, position, extension) sont toujours définis en fonction de l'adresse de départ de la puce vidéo VIC. Il est bien plus facile de se souvenir de VIC+29 que de 53277. Certains attributs réclament un registre pour chaque plan-objet — ainsi ceux des coordonnées x et y —, mais la plupart du temps les huit bits d'un même octet contrôlent indépendamment la même fonction pour les huit plans-objets disponibles. Il suffit donc de les activer ou non, suivant l'effet recherché. Le plan-objet 1 est défini par les treize derniers nombres inclus dans les DATA, et il représente un projectile éjecté du port manette.

Il est en effet, dans sa partie visible, très petit, et il est donc plus simple de stocker les 63 octets qui le définissent en deux temps. On POKE tout d'abord 63 zéros dans la zone mémoire concernée, puis les quelques nombres qui définissent le projectile sont lus (instruction READ) et POKés au même endroit. C'est plus simple que de faire usage de tous les chiffres théoriquement indispensables.

Les lignes 8190 à 8220 définissent les chaînes de caractères graphiques dont nous aurons besoin. LES forme une ligne horizontale qui occupe toute la largeur de l'écran, par simple répétition d'un caractère accessible à partir de la touche C. DW\$ est une série de caractères « curseur bas » (*cursor down*). LS\$ et RS\$ sont tracés

à partir de diagonales (accessibles à partir des touches N et M) placées à l'avant-plan, de façon à donner à la scène une certaine profondeur.

La routine de tir qui commence ligne 8310 choisit de façon aléatoire un point en bas d'écran et y dirige le lutin 1; le processus se répète jusqu'à ce que le joueur appuie sur une touche. A ce moment, les couleurs d'écran reprennent leurs valeurs normales, l'écran est effacé et les lutins disparaissent. Puis l'on repasse au programme principal. Pour faire usage de cette routine dans Digitaya, il suffit d'insérer une seule ligne : 3845 GOSUB 8000 : REM IMAGE PORT MANETTE. L'autre listage est consacré à l'unité arithmétique et logique (ALU) et illustre différentes méthodes d'affichage de caractères à l'écran. Les lignes 7040 à 7090 lisent un certain nombre de DATA et POKent les valeurs correspondantes en mémoire écran, tandis que la zone correspondante en mémoire couleur est elle aussi POKée. Ici, par exemple, on y placera le code couleur 2, c'est-à-dire du rouge.

On peut faire défiler vers le bas les trois lettres ALU grâce à une petite astuce de programmation. Les codes écran affectés aux caractères graphiques des trois lettres sont POKés de la première ligne sur la deuxième de l'écran. On appelle

ensuite le sous-programme de la ligne 7680, ce qui provoque un déplacement de l'écran d'une ligne. La deuxième ligne de code est POKée à son tour, le sous-programme est appelé de nouveau, et ainsi de suite jusqu'à ce que les huit lignes qui composent les lettres ALU aient été mises en œuvre. Elles donnent alors l'impression de descendre à partir du haut de l'écran.

Deux autres méthodes pour la mise en place des caractères sont présentées ici. Elles peuvent être affichées directement grâce à des commandes PRINT, comme c'est le cas lignes 7130 et 7140. Elles peuvent aussi former une chaîne de caractères comme aux lignes 7170 et 7590 à 7670 (le point d'interrogation). Ce dernier procédé est beaucoup plus souple d'emploi.

Pour faire usage de cette routine, il faut insérer la ligne suivante :

4565 GOSUB 7000 : REM IMAGE ALU

Des chiffres et des lettres
Les trois lettres ALU sont créées à partir de caractères graphiques basse résolution. Elles semblent traverser l'écran de haut en bas, jusqu'à leur position définitive. (Cl. Liz Dixon)

	Caractères graphiques	Codes-écran	
		Normal	Inversé
		32	160
	105	233	
	45	223	

Écran ALU

```

7000 REM **** S/P IMAGE ALU ****
7010 VIC=52248:CS=55296:SC=1024
7020 PRINT CHR$(147):REM EFFACE ECRAN
7030 POKE VIC+32,0:POKE VIC+33,0:REM DEFINIR ECRAN ET BORDURE
7040 CC=0
7050 FOR J=1 TO 8:GOSUB 7680:REM DEFILEMENT
7060 FOR I=47 TO 72
7070 READ A:CC=CC+A:POKE SC+I,A:POKE CS+I,2
7080 NEXT I,J
7090 READ CS:IF CS<>CC THEN PRINT"CONTROLE ERREUR":STOP
7100 GOSUB 7680:REM DEFILEMENT
7110 PRINTCHR$(158):REM TEXTE EN JAUNE
7120 FOR I=1 TO 8:PRINT:NEXT I:REM DEPLACEMENT VERS LE BAS
7130 PRINTTAB(9)"AND",SPC(7)"DR":SPC(8)"NOT"
7140 PRINTTAB(10)"o":SPC(9)"o":SPC(9)"o"
7150 PRINTCHR$(28):REM ROUGE
7160 REM ** POINT D'INTERROGATION **
7170 FOR I=1 TO 9:READ Q:PRINTTAB(16)Q:NEXT I
7180 REM **** ATTEND ET APPUIE SUR TOUCHE ****
7190 GET A$:IF A$="" THEN 7190
7200 POKE VIC+32,14:POKE VIC+33,6:REM ECRAN ET BORDURE
7210 PRINTCHR$(154):REM TEXTE BLEU CLAIR
7220 PRINTCHR$(147):REM EFFACE ECRAN
7230 RETURN
7240 REM **** DONNEES ECRAN ****
7250 REM ** LIGNE 1 **
7260 DATA 160,32,32,32,32,160,32,32,32,32
7270 DATA 95:160,160,160,160,105
7280 DATA 32,32,32,32,95,160,160,160,105
7290 REM ** LIGNE 2 **
7300 DATA 160,32,32,32,32,160,32,32,32,32
7310 DATA 160,223,32,32,32,32,32,32,32,32
7320 DATA 160,223,32,32,233,160
7330 REM ** LIGNE 3 **
7340 DATA 160,32,32,32,32,160,32,32,32,32
7350 DATA 160,32,32,32,32,32,32,32,32,32
7360 DATA 160,32,32,32,160
7370 REM ** LIGNE 4 **
7380 DATA 160,160,160,160,160,160,32,32,32,32
7390 DATA 160,32,32,32,32,32,32,32,32,32

```

```

7400 DATA 160,32,32,32,32,160
7410 REM ** LIGNE 5 **
7420 DATA 160,32,32,32,32,160,32,32,32,32
7430 DATA 160,32,32,32,32,32,32,32,32,32
7440 DATA 160,32,32,32,160
7450 REM ** LIGNE 6 **
7460 DATA 233,105,32,32,95,223,32,32,32,32
7470 DATA 160,32,32,32,32,32,32,32,32,32
7480 DATA 160,32,32,32,160
7490 REM ** LIGNE 7 **
7500 DATA 32,233,105,95,223,32,32,32,32,32
7510 DATA 160,32,32,32,32,32,32,32,32,32
7520 DATA 160,32,32,32,160
7530 REM ** LIGNE 8 **
7540 DATA 32,32,233,223,32,32,32,32,32,32
7550 DATA 160,32,32,32,32,32,32,32,32,32
7560 DATA 160,32,32,32,160
7570 DATA 14463:REM TOTAL CONTROLE
7580 REM ** DATA POINT D'INTERROGATION **
7590 DATA " ??????"
7600 DATA "? ?"
7610 DATA "? ?"
7620 DATA "? ?"
7630 DATA "? ?"
7640 DATA "? ?"
7650 DATA "? ?"
7660 DATA "? ?"
7670 DATA "? ?"
7680 REM **** S/R DEFILEMENT ECRAN ****
7690 POKE 218,160
7700 PRINTCHR$(19):CHR$(17):CHR$(157):CHR$(148)
7710 RETURN

```

```

8060 FOR I=SI+25 TO S1+37:READ A:CC=CC+A:POKE I,A:NEXT I
8065 READ CS:IF CS<>CC THEN PRINT"CONTROLE ERREUR":STOP
8070 POKEVIC+33,0:POKEVIC+32,0:REM ECRAN ET BORDURE
8080 REM ** POINTEURS LUTIN **
8090 POKE 2040,80/64:POKE 2041,S1/64
8100 REM ** REGISTRES CONTROLE LUTIN **
8110 POKE VIC+39,7: COULEUR LUTIN 0
8120 POKE VIC+40,7: COULEUR LUTIN 1
8130 POKE VIC,65:COORD,X LUTIN 0
8140 POKE VIC+1,70:COORD,X LUTIN 0
8150 POKE VIC+2,74:COORD,X LUTIN 1
8160 POKE VIC+3,70:COORD,Y LUTIN 1
8170 POKE VIC+29,1:EXTENSION LUTIN 0
8190 LE$="" :FOR I=1 TO 40:LE$=LE$+CHR$(195):NEXT I
8200 DW$="" :FOR I=1 TO 25:DW$=DW$+CHR$(17):NEXT I
8210 LS$="" :FOR I=1 TO 11:LS$=LS$+CHR$(206)+" " :NEXT I
8220 RS$="" :FOR I=1 TO 11:RS$=RS$+CHR$(205)+" " :NEXT I
8230 PRINTCHR$(158):REM TEXTE JAUNE
8240 PRINTCHR$(19):PRINT TAB(2)"PORT MANETTE"
8250 PRINTCHR$(154):REM TEXTE BLEU CLAIR
8260 PRINTCHR$(19):LEFT$(DW$,17):LE$
8270 PRINT CHR$(145):
8280 PT$=LEFT$(LS$,19)+LEFT$(RS$,21)
8290 PT$=PT$+RIGHT$(LS$,19)+RIGHT$(RS$,21)
8300 FOR I=1 TO 3:PRINT PT$:NEXT I
8305 POKE VIC+21,3:REM APPARITION LUTINS 0 et 1
8310 REM ** TIR **
8320 YB=240:Y1=70
8330 X1=74:X=INT(RND(1)*150)+24
8340 G=2*(X-X1)/(YB-Y1)
8350 FOR Y=Y1 TO YB STEP 2
8360 X1=X1+G:POKE VIC+2,X1:POKE VIC+3,Y
8370 NEXT Y
8380 GET A$:IF A$="" THEN 8330
8390 REM ** CHANGEMENT ECRAN **
8400 POKE VIC+21,0:REM SUPPRIME PLANS-OBJETS
8410 POKE VIC+32,14:POKEVIC+33,6:REM ECRAN ET BORD COUL. INIT.
8420 PRINT CHR$(147): EFFACE ECRAN
8430 RETURN
8440 REM **** DATA LUTIN ****
8450 DATA 0,0,0,63,255,252,64,0,2,64,0,2
8460 DATA 128,0,1,128,0,1,152,16,133,162,16,133
8470 DATA 128,0,1,128,0,1,128,0,1,128,0,1
8480 DATA 128,0,1,128,0,1,136,66,17,72,66,18
8490 DATA 64,0,2,64,0,2,32,0,4,31,255,248
8495 DATA 0,0,0,16,0,0,56,0,0,124,0,0,56,0,0,16
8497 DATA 37101:REM TOTAL DE CONTROLE

```

Écran port manette

```

8000 REM **** S/R IMAGE PORT MANETTE ****
8010 PRINTCHR$(147):REM CLEAR SCREEN
8020 SB=832:SI=60:64:REM DATA DEBUT ADRESSE LUTIN
8030 CC=0:VIC=53248:REM DEBUT PUCE VIC
8040 FOR I=80 TO 80+62:READ A:CC=CC+A:POKE I,A:NEXT I
8050 FOR I=81 TO 81+62:POKE I,0:NEXT I

```

Les factorielles

Nous allons voir comment LOGO est utilisé pour le calcul factoriel, et par quels moyens certains résultats sont transformés en « arbres factorielles ».

Combien y a-t-il de manières pour disposer quatre personnes autour d'une table? La première peut s'asseoir n'importe où. Mais une fois assise, il ne reste que trois places pour la deuxième personne, puis deux pour la troisième, et enfin une seule pour la dernière. Aussi le nombre total de permutations différentes est $4 \times 3 \times 2 \times 1$. Ce qui s'écrit généralement « 4! », et se lit « factorielle 4 ». Les factorielles se rencontrent fréquemment dans les problèmes mathématiques de permutations, de combinaisons et de probabilités.

Une définition sur le calcul factoriel est simple. Il faut d'abord savoir que la factorielle de 0 est égale à 1. Factorielle de tout nombre positif différent de zéro, mettons x , est égale à factorielle de $x-1$ multipliée par x . Si l'on écrit cela sous forme de programme, nous obtenons :

```
POUR FACTORIELLE :X
  SI :X = 0 ALORS RÉSULTAT 1
  RÉSULTAT (FACTORIELLE :X - 1) * :X
FIN
```

Pour le vérifier, tapez AFFICHE FACTORIELLE 6 — le résultat doit être 720.

Cette procédure est valable jusqu'à 12, car au-delà les nombres sont trop grands pour figurer dans l'ordinateur sous forme d'entrées. Sur le Commodore 64, par exemple, AFFICHE FACTORIELLE 13 donne 6,22702E9, c'est-à-dire $6,22702 \times 10^9$. Cela n'est pas satisfaisant puisque les quatre derniers chiffres ont été perdus. De nombreuses raisons (dont la simple curiosité) font que l'on voudrait bien savoir ce qu'ils sont devenus. La première chose à faire est donc d'étendre les performances de calcul de LOGO de sorte qu'il puisse atteindre une précision supérieure à sept chiffres.

Pour simplifier les choses, limitons-nous aux entiers positifs. Nous représenterons les entiers sous forme de listes. Par exemple, 1.234.567 s'écrit [1 2 3 4 5 6 7]. Les deux procédures suivantes effectueront des additions sur de tels nombres. Essayez-les avec AFFICHE LONGUEADD[123][569] — le résultat devra être [692] :

```
POUR LONGUEADD :X :Y
  RÉSULTAT LONGUEADD1 :X :Y 0
FIN
```

```
POUR LONGUEADD1 :X :Y :RETENUE
  SI (TOUS (VIDE? :X) (VIDE? :Y)
  (:RETENUE = 0)) ALORS RÉSULTAT []
  TESTE VIDE? :Y
  SIVRAI SI :RETENUE = 0 ALORS RÉSULTAT :X SINON
  RÉSULTAT
```

```
LONGUEADD1 :X [1] 0
TESTE VIDE? :X
SIVRAI SI :RETENUE = 0 ALORS RÉSULTAT :Y SINON
RÉSULTAT
LONGUEADD1 [1] :Y 0
FAIS « SOMME (DERNIER :X) +
(DERNIER :Y) + :RETENUE
RÉSULTAT METSDANSLISTE RESTE
:SOMME 10 LONGUEADD1
SAUFDERNIER
::X SAUFDERNIER :Y QUOTIENT :SOMME 10
FIN
```

Ces procédures ressemblent à celles que nous utilisons pour additionner sur un papier, sans l'aide de machines.

La soustraction suit une démarche similaire. Nous avons cependant ajouté une routine qui retire les zéros de tête dans le résultat, ce qui empêche des résultats du genre [0 0 0 7 8].

```
POUR LONGUESOUSTR :X :Y
  RÉSULTAT ENLÈVEZÉROS
  LONGUESOUSTR1 :X :Y 0
FIN
```

```
POUR LONGUESOUSTR1 :X :Y :POSE
  SI (TOUS (VIDE? :X) (VIDE? :Y) (:POSE = 0)) ALORS
  RÉSULTAT [0]
  TESTE VIDE? :Y
  SIVRAI SI :POSE = 0 ALORS RÉSULTAT :X SINON
  RÉSULTAT LONGUESOUSTR1 :X [1] 0
  SI VIDE? :X ALORS AFFICHE (DÉSOLÉ, IMPOSSIBLE DE
  TRAITER UN RÉSULTAT NÉGATIF) PREMIERNIVEAU FAIS
  « DIFF (DERNIER :X) - (DERNIER :Y) - :POSE
  SI :DIFF < 0 ALORS RÉSULTAT METSDANSLISTE (10 +
  :DIFF)
  LONGUESOUSTR1 SAUFDERNIER :X SAUFDERNIER :Y 1
  RÉSULTAT METSDANSLISTE :DIFF
  LONGUESOUSTR1 SAUFDERNIER :X
  SAUFDERNIER :Y 0
FIN
```

```
POUR ENLÈVEZÉROS :X
  SI VIDE? :X ALORS RÉSULTAT [0]
  SI NON ((PREMIER :X) = 0) ALORS RÉSULTAT :X
  RÉSULTAT ENLÈVEZÉROS SAUFPREMIER :X
FIN
```

L'opération longue-multiplication est plus difficile. Pour la résoudre, nous utiliserons la méthode enseignée à l'école. Si nous voulons par exemple multiplier 123 par 338, le problème doit être divisé en trois phases. Nous multiplions d'abord 123 par 8, puis 123 par 330, et pour finir nous additionnons les deux résultats. Cette



0!	1
1!	1
2!	2
3!	6
4!	24
5!	120
6!	720
7!	5.040
8!	40.320
9!	362.880
10!	3.628.800
11!	39.916.800
12!	479.001.600
13!	6.227.020.800
14!	87.178.291.200
15!	1.308.674.368.000
16!	20.922.789.888.000
17!	355.687.428.096.000
18!	6.402.373.705.728.000
19!	121.645.100.408.832.000
20!	2.432.902.008.176.640.000
16!	20.922.789.888.000
17!	355.687.428.096.000
18!	6.402.373.705.728.000
19!	121.645.100.408.832.000
20!	2.432.902.008.176.640.000

Facteur explosif

Les valeurs des factorielles s'accroissent avec une rapidité étonnante, comme on peut le voir ici. C'est du reste parce que les nombres deviennent trop grands que la plupart des ordinateurs et calculateurs représentent en notation exponentielle les factorielles supérieures à 12. Ainsi la factorielle de 12 serait 4,79E8, ou $4,79 \times 10^8$. La précision s'accroît avec le nombre de chiffres significatifs. (Cl. Ian McKinnell.)

méthode suppose que la deuxième phase puisse elle-même se subdiviser en deux sous-étapes. Multiplier d'abord 123 par 33, puis ajouter un zéro à la fin du résultat. La multiplication d'un nombre par 33 implique l'utilisation de la récursion. La procédure LONGUEMULTI couvre tous ces aspects :

```

POUR LONGUEMULTI :X :Y
  SI VIDE? SAUFDERNIER :Y ALORS RÉSULTAT
  LONGUEMULTI1 :X :DERNIER :Y 0
  RÉSULTAT, LONGUEADD (LONGUEMULTI1 :X (DERNIER :Y)
  0) (METS DANSLISTE «0
  LONGUEMULTI :X SAUFDERNIER :Y)
  FIN

```

Les opérations de détail relatives à la multiplication d'une ligne par un chiffre sont effectuées par LONGUEMULTI1 :

```

POUR LONGUEMULTI1 :X :NOMB :RETENUE
  TESTE VIDE? :X
  SIVRAI SI :RETENUE = 0 ALORS RÉSULTAT [] SINON
  RÉSULTAT (LISTE :RETENUE)
  FAIS « PRODUIT (DERNIER :X) * :NOMB+ :RETENUE
  RÉSULTAT METS DANSLISTE RESTE :PRODUIT 10
  LONGUEMULTI1 SAUFDERNIER :X :NOMBQUOTIENT
  :PRODUIT 10
  FIN

```

Pour le calcul factoriel, nous n'avons pas besoin de procédures de division. Mais pourquoi ne pas les écrire vous-même ?

Nous disposons maintenant d'un jeu de base de calcul arithmétique pour aboutir à n'importe quel degré de précision. La seule limite de la taille des nombres dépend de la place mémoire disponible pour le programme.

Modifier

Nous pouvons maintenant modifier notre programme de calcul factoriel afin d'utiliser la nouvelle routine de longue-multiplication :

```

POUR FACT :X
  SI PREMIER :X = 0 ALORS RÉSULTAT [1]
  RÉSULTAT LONGUEMULTI (FACT LONGUEMULTI :X [1]) :X
  FIN

```

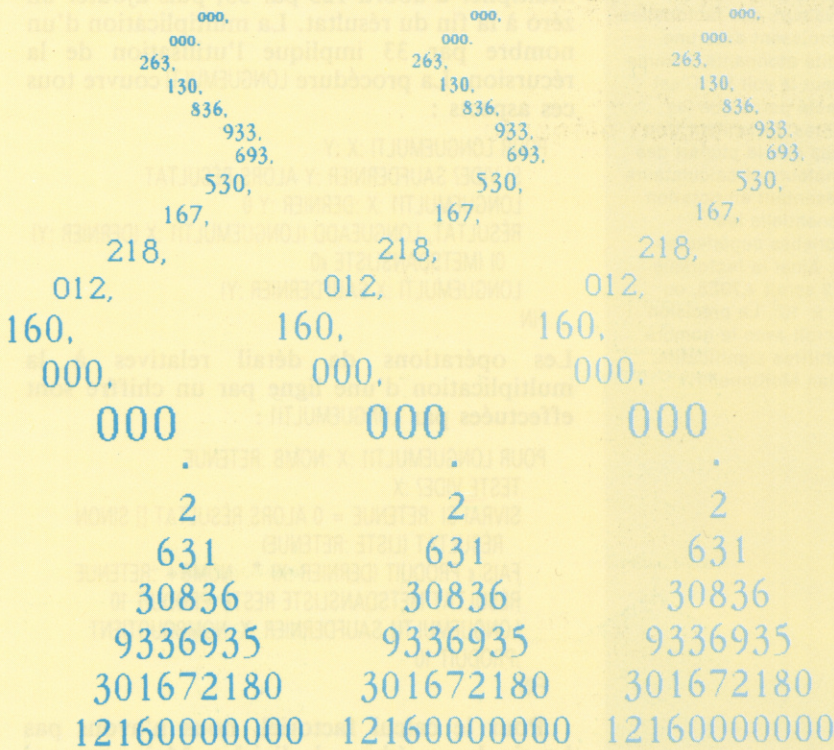
Pour l'essayer, tapez : FACT [1 3]; vous devez obtenir [6 2 2 7 0 2 0 8 0 0]. Pourtant des problèmes existent : le calcul est lent et, sur le Commodore 64, nous n'avons pu obtenir la valeur factorielle d'un nombre supérieur à 34. Cette dernière comprenait 39 chiffres avec beaucoup de temps pour le calcul.

L'expression de nombres importants sous forme de listes n'est pas familière, mais nous pouvons pallier cet inconvénient en modifiant le programme afin de passer librement de l'une à l'autre forme. Nous utilisons deux procédures, ÉTALE et RESSERRE : ÉTALE 123 donne [1 2 3], et RESSERRE [1 2 3] donne 123.

```

POUR ÉTALE :X
  SI VIDE? :X ALORS RÉSULTAT []
  RÉSULTAT (PHASE PREMIER :X ÉTALE
  SAUFPREMIER :X)
  FIN

```



Arbres à chiffres

On obtient les arborescences de valeurs factorielles à partir de leur chiffre le plus à gauche que l'on place sur le sommet de l'arbre. Ce diagramme représente la valeur factorielle 32 (32!). La valeur est plus lisible lorsque les chiffres sont disposés par groupe de trois, comme cela est montré ci-dessus.

```

POUR RESSERRE :X
SI VIDE? :X ALORS RÉSULTAT»
RÉSULTAT (MOT PREMIER :X RESSERRE)
SAUFPREMIER :X)
FIN
    
```

Ces procédures s'appuient sur un principe du LOGO, dans lequel les nombres sont traités comme des mots. Nous pouvons, en les utilisant, définir une procédure, F :

```

POUR F :X
AFFICHE RESSERRE FACT ÉTALE :X
FIN
    
```

Cette procédure calcule la valeur factorielle de 13 en réponse à F13.

Le résultat du calcul, 6227020800, n'est pas très lisible. C'est pourquoi on y insère des points (6.227.020.800) qui en améliorent la compréhension (des virgules pour l'ordinateur). Les procédures suivantes divisent le mot en deux groupes de trois chiffres et insèrent des points :

```

POUR AJOUTEPOINTS :X
SI ((COMPTE :X)<4) ALORS RÉSULTAT :X
RÉSULTAT (MOT AJOUTEPOINTS SAUFTROIS :X »,
TROISDERNIERS :X)
FIN
POUR SAUFTROIS :X
RÉSULTAT SAUFDERNIER SAUFDERNIER SAUFDERNIER :X
FIN
POUR TROISDERNIERS :X
RÉSULTAT (MOT (DERNIER SAUFDERNIER SAUFDERNIER
:X)
(DERNIER SAUFDERNIER :X) (DERNIER :X))
FIN
    
```

Nous devons également modifier F pour incorporer ces procédures :

```

POUR F :X
AFFICHE AJOUTEPOINTS RESSERRE FACT ÉTALE :X
FIN
    
```

L'utilisation de F pour afficher les vingt premières factorielles montre l'accroissement rapide de la taille des factorielles (les résultats figurent dans la table).

Ayant obtenu les valeurs factorielles de toute une gamme de nombres, commençons à classer leurs résultats. Un mathématicien a eu la brillante idée de disposer les très grands nombres factoriels qu'il obtenait dans ses calculs, sous la forme de structures arborescentes, comme des sapins de Noël, qu'il envoyait à ses amis ! Il faut bien sûr choisir ses factorielles car toutes n'ont pas le bon nombre de chiffres pour représenter un arbre, mais les procédures suivantes obtiendront l'effet voulu :

```

POUR ARBRE :L
ARBRE1 1 :L
FIN
POUR ARBRE1 :NOMB:L
SI VIDE? :L ALORS STOP
RÉPÈTE ANOUCHEAU (20 - :NOMB/2) [AFFICHE1 BLANC]
LIGNEAFFICHÉE :NOMB:L
ARBRE1 :NOMB+ 2 ÉLAGUE :NOM:L
FIN
POUR BLANC
RÉSULTAT CAR 32
FIN
POUR LIGNEAFFICHÉE :NOMB :L
SI :NOMB = 0 ALORS AFFICHE «STOP
AFFICHE1 PREMIER :L
LIGNEAFFICHÉE :NOMB - 1 SAUFPREMIER :L
FIN
POUR ÉLAGUE :NOMB :L
SI :NOMB=0 ALORS RÉSULTAT :L
RÉSULTAT ÉLAGUE :NOMB - 1 SAUFPREMIER :L
FIN
    
```

Il nous faut à nouveau modifier notre procédure de commande :

```

POUR F :X
ARBRE RESSERRE FACT ÉTALE :X
FIN
    
```

Variantes de logo

Pour toutes les versions LCS1, utilisez :

```

VIDEP pour VIDE?
ET pour TOUS
TAPE pour AFFICHE1
    
```

La syntaxe de SI est également différente. Par exemple :

```

SI :RETENUE = 0 [RÉSULTAT []] [RÉSULTAT (LISTE :RETENUE)]
    
```

Utilisez DIV :X :Y à la place de QUOTIENT :X :Y sur le Spectrum, et ROUND (:X / :Y) sur Atari.

Utilisez PH pour PHRASE sur Atari.

Boucles emboîtées

Les « boucles à l'intérieur d'autres boucles » ou « boucles emboîtées » sont utiles, pratiques et de conception élégante, mais délicates à traiter.

Nous donnons le nom de boucle emboîtée à la partie d'un organigramme dans laquelle une boucle contient un ou plusieurs cycles. Plutôt qu'un long discours, suivons les exemples fournis par les figures.

Premier exemple. Sur une éventuelle liste, on représentera pour chacun des travailleurs d'une entreprise les éléments suivants :

- a) lire les données d'identification des travailleurs;
- b) entrer les données des fichiers quotidiens de chaque travailleur;
- c) calculer le salaire brut, les retenues et les liquidités à percevoir selon le travail réalisé;
- d) imprimer le reçu correspondant (v. fig. 1).

On observe qu'il y a une boucle générale qui enveloppe toutes les opérations relatives à l'ensemble des travailleurs de l'entreprise. Mais il en existe une autre, intérieure, qui se charge de la lecture des fichiers quotidiens correspondant à chaque travailleur, car chacun d'entre eux a pu travailler une quantité différente de jours. Pour sortir de cette boucle intérieure, la question est de savoir s'il reste d'autres fichiers sur une même personne; si tel est le cas, les données d'un nouveau fichier seront entrées. Quand celles-ci seront complètes, le reste des opérations inscrites dans l'organigramme continuera jusqu'à la question : « Est-ce la dernière liste à réaliser ? » Si la réponse est affirmative, la boucle logée à l'intérieur de la boucle générale provoque un contrôle et une lecture des fichiers quotidiens. Elle répète le cycle une quantité variable de fois, qui dépend du nombre de fichiers, c'est-à-dire des jours travaillés par chacun des salariés.

Figure 1

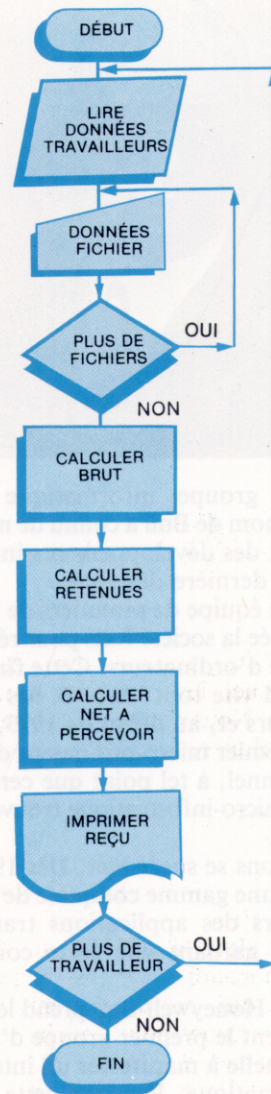
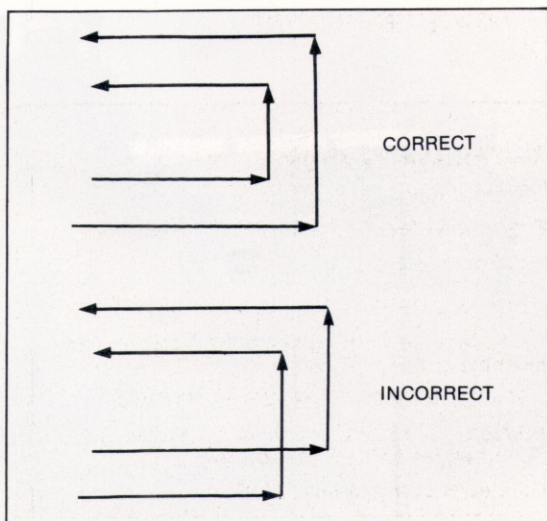
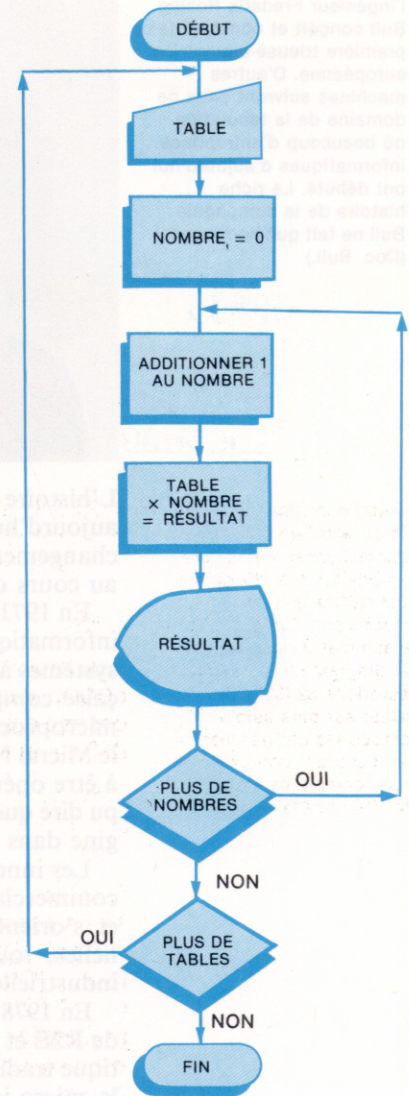


Figure 2



Deuxième exemple. Il s'agit de démontrer comment les tables de multiplication, que nous avons déjà expliquées, peuvent se développer au moyen de l'utilisation de ces deux boucles (v. fig. 2) : la « générale », qui demande si l'on souhaite ou non visualiser la dernière table après avoir imprimé la précédente, et l'« interne », logée à l'intérieur de l'autre, qui permet de confectionner la table avec autant de facteurs que nous le désirons.

Quelque chose qui s'oublie... Il est important d'observer que les boucles emboîtées sont celles qui contiennent une ou plusieurs boucles en totalité; ces dernières ne doivent jamais se croiser.



Dynamic Bull

Le premier micro-ordinateur du monde était français. L'activité Micral est aujourd'hui reprise par le groupe Bull. Celui-ci manifeste sa volonté de s'imposer sur le marché micro-informatique.

C'est en 1922 que l'ingénieur Frederik Rosing Bull conçoit et construit la première trieuse-tabulatrice européenne. D'autres machines suivront dans ce domaine de la tabulation où beaucoup d'entreprises informatiques d'aujourd'hui ont débuté. La riche histoire de la compagnie Bull ne fait que commencer. (Doc. Bull.)



L'histoire du groupe informatique portant aujourd'hui le nom de Bull a connu de nombreux changements et des développements importants au cours de la dernière décennie.

En 1971, une équipe de pionniers de la micro-informatique crée la société R2E pour réaliser des systèmes à base d'ordinateurs. Cette firme française comprend vite tout l'intérêt des premiers microprocesseurs et, au début de 1973, elle sort le Micral N, premier micro-ordinateur du monde à être opérationnel, à tel point que certains ont pu dire que la micro-informatique trouve son origine dans R2E.

Les innovations se succèdent. Dès 1975, R2E commercialise une gamme complète de systèmes et s'oriente vers des applications transactionnelles, tout en gardant une forte compétence industrielle.

En 1978, CII-Honeywell-Bull prend le contrôle de R2E et devient le premier groupe d'informatique traditionnelle à manifester un intérêt pour la micro-informatique. Pendant cette période, la gamme se complète et s'étend avec la série Micral 80 et, en 1982, R2E annonce son 16 bits, le Micral 90-50, complété ensuite par le Micral 90-20.

La maturation de l'entreprise pendant cette période, qui se traduit par la prise en compte du facteur industriel, des exigences de qualité, de grande diffusion, de communication et par le développement d'une « culture » micro-informatique, se poursuit en 1983, année de la création du groupe Bull.

La constitution du groupe Bull, qui couvre la plupart des besoins informatiques avec ses quatre entités-produits — Bull Systèmes, Bull Sems, Bull Transac, dont dépend Bull Micral, et Bull

Périphériques —, annonce une véritable stratégie micro-informatique destinée à tirer toutes les conséquences de l'évolution de ce marché pour mettre en place les conditions d'un développement dynamique.

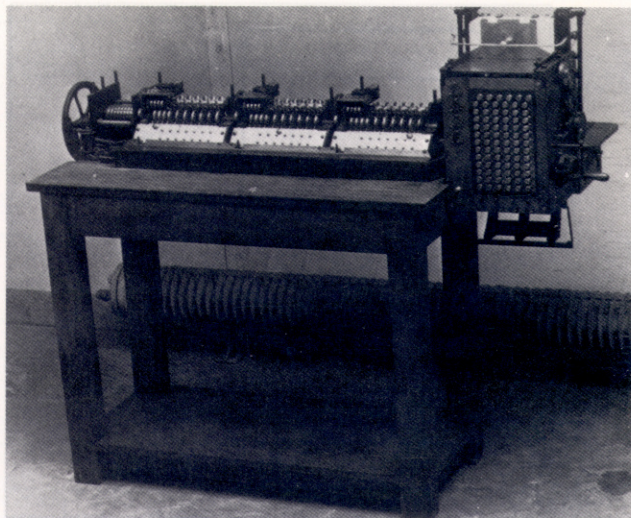
En effet, la période actuelle est marquée par le passage de systèmes d'information structurés et modélisables (comptabilité, stocks, paie, etc.) organisés autour d'une intelligence centrale et de grands fichiers, à des systèmes d'information souples et adaptables, capables d'assumer des tâches peu structurées (aide à la décision, traitement de texte, etc.).

Si le groupe Bull n'a pas été un acteur déterminant du développement récent du marché de la micro-informatique, la mise en place d'une politique destinée à replacer le groupe dans le peloton de tête des constructeurs s'est opérée dans des délais déterminés par le groupe.

Pour réussir, Bull s'est doté de l'ensemble des moyens nécessaires pour réaliser au mieux l'intégration de ses produits dans l'univers informatique, condition aujourd'hui essentielle de tout succès industriel et commercial. Parmi ces moyens, la micro-informatique joue un rôle considérable. Or, le groupe dispose en particulier de trois atouts essentiels :

— la taille du groupe permet d'éviter que la politique suivie en micro-informatique soit entièrement soumise aux variations à court terme du marché; elle est un facteur de pérennité;

— l'expérience irremplaçable acquise depuis plus de dix ans par R2E en matière de micro-informatique permet de maîtriser parfaitement la chaîne de conception d'un micro-ordinateur;





— d'emblée, la micro-informatique Bull se voit dotée d'une dimension internationale, condition indispensable pour s'implanter durablement sur le marché.

A l'origine, l'expansion du marché micro-informatique, à un rythme de 35 % par an environ, s'est traduite par une certaine anarchie. De nombreux constructeurs ont alors proposé leurs solutions aux besoins ponctuels du marché en espérant imposer leurs conceptions.

L'apparition de standards s'est effectuée sous la pression des utilisateurs, désireux d'instaurer une certaine compatibilité entre les différents matériels présents. C'est ainsi que l'IBM PC s'est imposé comme le standard de la micro-informatique professionnelle. Au cours des années 1983 et 1984, la compatibilité s'est imposée aux constructeurs comme une condition indispensable de tout succès sur le marché. L'annonce du dernier modèle, le Bull Micral 30, *totale*ment compatible à ce standard, s'inscrit dans cette logique. Parallèlement, le maintien en option du système d'exploitation Prologue adopté pour les précédents modèles de la gamme permettra au Bull Micral 30 de s'insérer harmonieusement parmi les autres Micral.

Dans ce contexte, Bull espère obtenir une part minimale de 5 à 7 % du marché micro-informatique en Europe. Pour faire face à la demande prévisible en 1985, Bull Micral dispose à Marcq-

en-Barœul (Nord) d'un outil industriel dont la capacité de production permet la livraison du produit dans les délais les plus brefs.

Capable d'innover et de créer, ce pôle Bull Micral, fonctionnant en petites équipes enthousiastes comme celles qui avaient permis à R2E de lancer un tout nouveau produit, devrait s'imposer sur un marché caractérisé par la mobilité.

Dès maintenant, la logistique Bull permet à son entité micro-informatique de concilier les exigences d'une production de grande diffusion du Bull Micral 30 avec les critères de qualité les plus sévères. Autant dire, des atouts essentiels aux yeux de l'utilisateur...

L'arrivée du Micral 30, dernier des micro-ordinateurs nés chez Bull, montre clairement les intentions de l'entreprise française : profiter de la forte croissance du marché et de la comptabilité IBM PC pour affirmer des ambitions européennes. (Doc. Bull.)

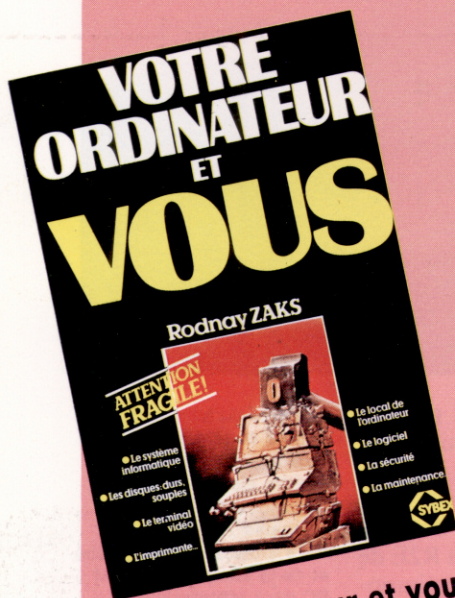


Le premier micro-ordinateur du monde à être opérationnel s'appelait le Micral N. C'était en 1973. Aujourd'hui ce clavier est celui du Micral 30.



Livres pour débiter

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.



Votre ordinateur et vous

Ce livre vous dit tout ce que vous devez savoir pour utiliser sûrement et simplement votre système informatique. Il présente les règles et les procédures, la plupart du temps sous la forme d'obligations et d'interdictions, pour chaque élément du système. Ces obligations et ces interdictions ont pour but d'assurer l'intégrité du matériel et du logiciel, aussi bien que la sécurité et la tranquillité d'esprit de l'opérateur.

Par Rodney Zaks.
236 pages. Format 16 x 22 cm.
Sybex.



Ainsi naquit l'informatique

L'évolution de l'informatique est si rapide qu'un type de matériel dure en général moins de dix ans, l'arrivée de nouveaux éléments conceptuels et technologiques opérant le passage de l'un à l'autre. Pourquoi dès lors ne pas envisager de s'initier à l'informatique, d'affiner ses connaissances ou d'anticiper son devenir en étudiant son histoire? Telle est la démarche attrayante et inédite proposée par René Moreau, éminent spécialiste, directeur du développement scientifique d'I.B.M. France et ancien président de la division informatique de l'A.F.C.E.T.

Par R. Moreau.
264 pages.
Dunod informatique.



Votre premier ordinateur

Si vous utilisez déjà un ordinateur personnel, si vous pensez en utiliser un ou en acheter un, ce livre vous sera indispensable. Véritable guide d'achat, il explique ce qu'est un micro-ordinateur, comment il fonctionne, ce qu'il est capable de faire dans ses applications tant familiales, personnelles que professionnelles. Un chapitre entier traite des critères de choix des différents éléments, des unités périphériques et des programmes.

Par Rodney Zaks.
294 pages. Format 14 x 21,5 cm.
Sybex.