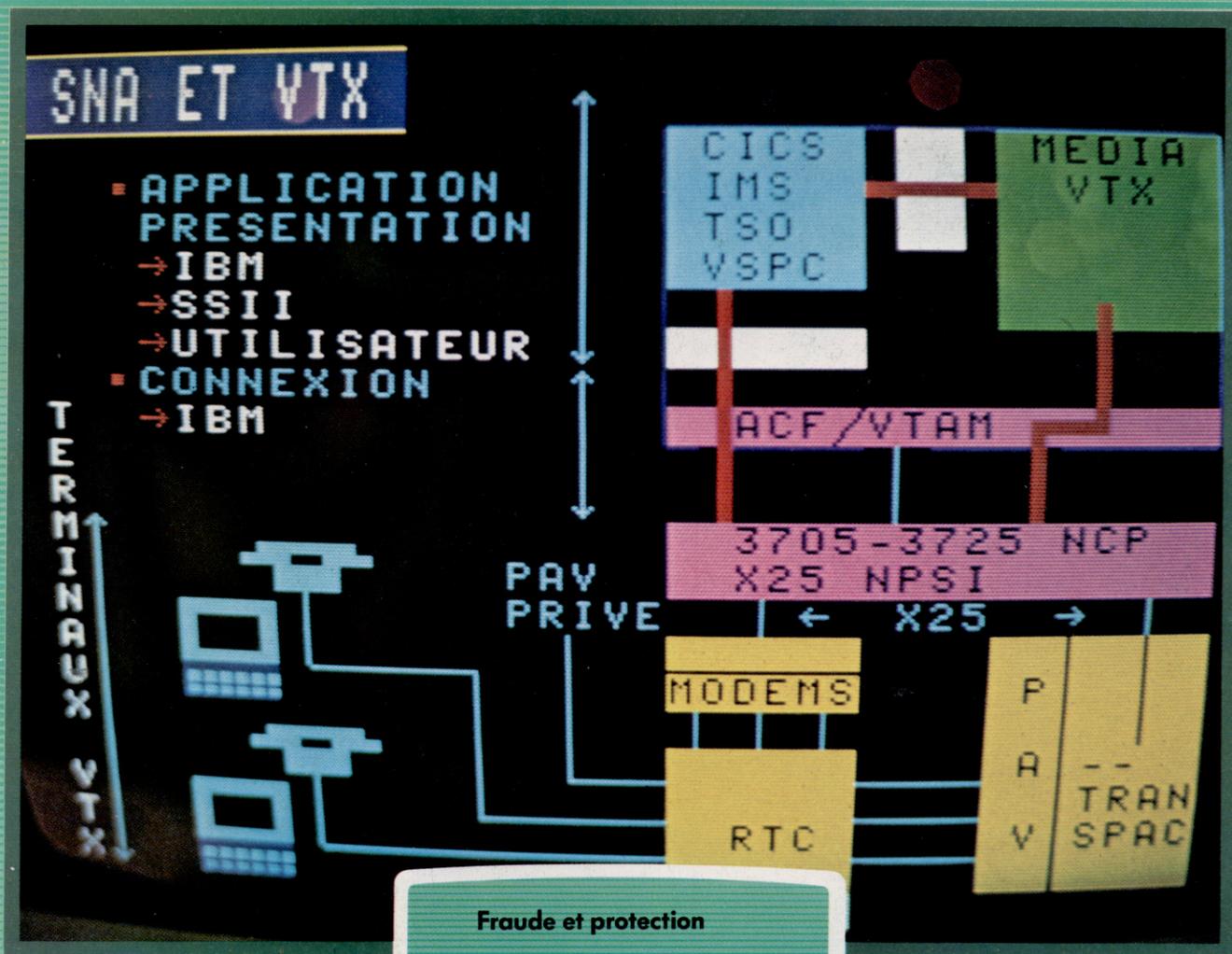


abc

N° 73

COURS
D'INFORMATIQUE
PRATIQUE
ET FAMILIALE

INFORMATIQUE



Fraude et protection

Le Bull Micral 30

Music Maker sur le C64

Le pavage Logo

EDITIONS
ATLAS



Du bout des doigts

Music Maker est un ensemble qui fait usage d'un clavier à deux octaves installé sur un Commodore 64. Le but? Tirer un meilleur parti des possibilités sonores du micro-ordinateur.

Les ordinateurs personnels sont souvent pourvus de possibilités techniques qui laissent véritablement espérer la création de périphériques spécialisés. C'est particulièrement vrai quant aux capacités graphiques et sonores de ces appareils pour lesquels de nombreux « additifs » ont déjà été construits. On sait que le Commodore 64, pourtant équipé d'une puce sonore très sophistiquée, ne dispose d'aucune commande BASIC qui permette d'en tirer parti! On a donc vu apparaître de très nombreux logiciels de création musicale, souvent excellents, mais qui doivent être mis en œuvre à partir du clavier de l'appareil, de type machine à écrire. Il serait très intéressant, et plus pratique, de se servir du 64 comme d'un piano. C'est précisément ce que Music Maker entend rendre possible.

Il comprend d'abord un cache en plastique, qu'il faut installer sur le corps de l'appareil, autour du clavier de base et des touches de fonction. Ce cache comporte, en sa partie centrale, des touches de piano qui viennent s'insérer sur les touches alphanumériques, grâce à une série d'ergots. Toute pression est donc répercutée jusqu'à l'unité centrale du 64. C'est une méthode assez primitive, mais qui a l'avantage d'être peu coûteuse et très fiable : il est rare de manquer une note, même en jouant très vite. Ce clavier piano se limite cependant à deux octaves.

Le logiciel d'accompagnement

Il existe en version cassette ou disquette. Le programme est commandé par menu, et met à contribution les touches de fonction. L'utilisateur peut modifier les sons produits en changeant sa hauteur, ou sa forme d'onde. Un rythme de basse et des percussions peuvent servir d'accompagnement; un séquenceur permet de programmer des mélodies, puis de les exécuter. Tout comme les sons eux-mêmes, elles peuvent être stockées sur cassette ou disquette, puis chargées en mémoire à la demande.

Chacune des voix peut être définie de façon à produire tel ou tel effet sonore. L'option F6 détermine celle qui sera modifiée. Après quoi, il faut sélectionner les paramètres (entre 0 et 15) attribués à l'attaque, au déclin, au soutien et à la relâche : ils définissent le timbre particulier d'un son. Le programme vous demandera ensuite si vous comptez faire usage des filtres (qui suppriment certaines fréquences sonores). En cas de réponse positive, il voudra aussi savoir quels sont les réglages nécessaires.



Croque-notes

Destiné au Commodore 64, Music Maker comprend un clavier en plastique qui se fixe sur l'ordinateur de façon à être en contact avec les touches alphanumériques, et d'un logiciel qui gère la production des sons et des notes. Les touches de fonction restent accessibles, car ce sont elles qui permettent la programmation proprement dite. Les résultats sont d'une qualité étonnante. (Cl. Ian McKinnell.)

Le séquenceur est un élément important dans la création de musique sur ordinateur. Pourtant bien des logiciels musicaux le laissent de côté. Il joue une séquence de notes qu'il répète; sa programmation se fait en deux temps. Il faut d'abord entrer toutes les notes qui composent l'ensemble de la mélodie à l'aide du clavier du piano; ensuite, il est nécessaire de fixer la durée de chacune d'elles (donc le rythme de l'ensemble) en appuyant sur la touche F5.



MUSIC MAKER

PRIX

★★

LOGICIEL

Disponible sur cassette ou sur disquette.

DOCUMENTATION

Music Maker est accompagné de deux manuels. Le premier donne tous les conseils nécessaires à une bonne mise en route, mais, avec un peu d'expérience, il arrive un moment où l'on aurait besoin de plus de précisions. Le second se réduit à un ensemble de partitions de chansons à succès.

FORCES

D'un prix raisonnable, Music Maker est un ensemble très intéressant qui permet d'utiliser le Commodore 64 comme un véritable instrument de musique.

FAIBLESSES

L'éventail des options reste très limité, surtout au niveau de la section rythmique (basse et percussion).

Les trois rythmes de percussion disponibles sont un peu limités mais assez bien choisis. Les rythmes de la basse suivent ceux des percussions, bien qu'on puisse en modifier la hauteur, ou la supprimer complètement. En gros, selon le rythme choisi, le décalage varie d'une quinte ou d'une octave entière. Les touches curseur accélèrent ou ralentissent le tempo suivant le sens désiré.

Malheureusement, bien des options sont incompatibles les unes avec les autres. Si basse et percussion jouent simultanément, il n'est possible de se servir du clavier qu'en mode monophonique. Il est donc impossible de jouer des accords avec le rythme en fond sonore.

Cela est aussi vrai pour le séquenceur. Bien des groupes New Wave ont renoncé à se doter d'une section rythmique (batterie), ou, plutôt, confient ce soin à un appareil électronique. Cela n'est pas envisageable avec Music Maker — le séquenceur ne peut être accompagné de la basse et de la percussion. C'est ici le matériel qui est en cause. La puce SID (*Sound Interface Device*) ne dispose que de trois voix (plus un canal pour le bruit blanc, qui ne peut d'ailleurs être activé en même temps). L'appui simultané d'une touche et de la barre d'espace permet des effets de glissando — passage progressif d'une note à une autre — qui n'est pas non plus compatible avec la section rythmique; mais cette fois, cela est sans doute dû au fait que le son produit est de type numérique. Obtenir de la basse et de la batterie un changement suffisamment petit pour fournir un « glissement » très doux semble hors de portée des capacités du processeur. Actuellement, bien entendu. Mais beaucoup plus grave, il n'est pas

possible de reprogrammer la basse et la percussion, afin de les adapter à des besoins particuliers. Pourtant, cela n'aurait demandé que le recours à des procédés qui sont précisément utilisés par le séquenceur; une programmation des rythmes selon les besoins aurait donné au logiciel une richesse bien plus grande.

Enfin, lorsqu'on joue en « temps réel », on ne peut modifier la voix ou l'octave choisie au départ. Dans les faits, l'utilisateur est donc effectivement limité à deux octaves.

Facilité d'emploi

Music Maker est vendu avec deux manuels. Le guide de l'utilisateur explique comment charger le programme et présente très succinctement les différentes fonctions. Il n'est pas très riche de précisions, mais suffira sans doute pour qui débute. Le programme est en effet commandé par menu et présente des instructions très claires, qui permettent de tirer un profit maximal des possibilités de l'ensemble. Le second fascicule explique brièvement les principes de la notation musicale, montre comment jouer au clavier, et contient les partitions de vingt-huit chansons à succès.

Music Maker est un logiciel intéressant, qui sait tirer parti des immenses possibilités sonores du Commodore 64. Les débutants apprécieront tout particulièrement sa facilité d'emploi. Les autres estimeront peut-être qu'il manque de souplesse, et que le clavier se prête mal à la création d'effets un tant soit peu complexes. Il représente pourtant un investissement profitable pour quiconque se sent une âme de compositeur.

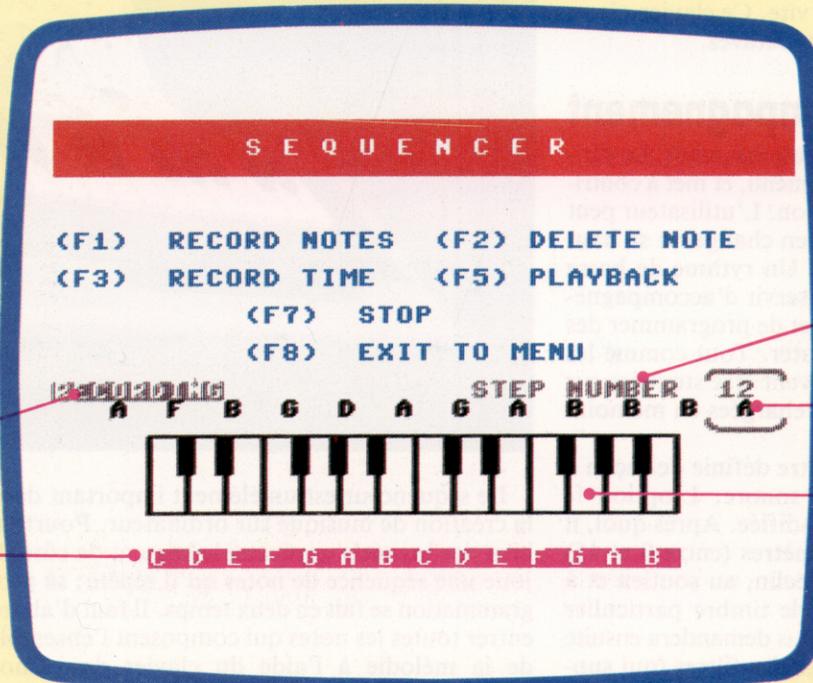
De la musique au menu

Voici l'option séquenceur de Music Maker, telle qu'elle est présentée à l'écran.

Le menu affiche les six options disponibles. Notez qu'il faut entrer séparément les notes à jouer et leur durée.

L'utilisateur a sélectionné ici le mode enregistrement, et non le mode playback.

Voici (en notation anglaise : C = do) le nom de chacune des notes du clavier musical.



L'indication Step Number précise combien de notes ont déjà été mises en mémoire par le séquenceur.

Nom des notes déjà mises en mémoire (A = la).

Un curseur bleu se déplace sur le clavier pour indiquer quelle est la note jouée à chaque exécution.



Fraude et protection

La multiplication des traitements informatisés, leur aspect de plus en plus décentralisé et leur ouverture à une population plus nombreuse ont fait naître une forme de criminalité : la fraude informatique.

L'ordinateur a transformé la vie des entreprises, comme il bouleverse déjà celle des individus. Il a aussi entraîné une nouvelle forme de criminalité, que l'on qualifie souvent de « délinquance en col blanc ».

La fraude informatique a, en effet, connu ces dernières années un important développement, l'imagination et le savoir-faire des fraudeurs évoluant presque aussi rapidement que les techniques. Qu'elle porte sur des escroqueries ou sur des détournements d'informations de grande valeur, la fraude informatique peut avoir de très graves conséquences pour les utilisateurs de systèmes informatiques et, en première ligne, pour les entreprises.

C'est pourquoi celles-ci se sont intéressées de plus en plus près à ce qu'on appelle la « sécurité informatique » et à toutes les techniques de protection, aussi bien du matériel que du logiciel, afin de mettre à l'abri des biens qui représentent la plupart du temps un très lourd investissement en temps, en argent et en connaissances.

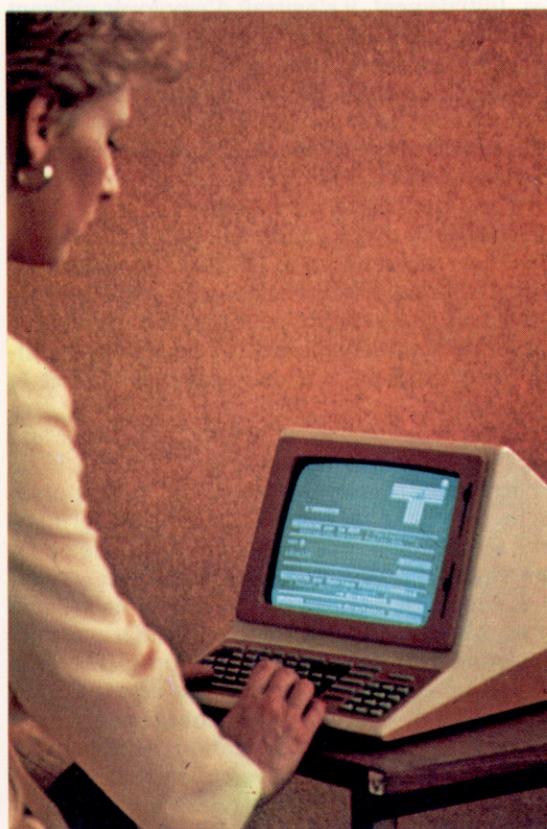
A l'instar d'un appartement ou de tout autre bien mobilier ou immobilier, le système informatique doit donc être protégé contre l'effraction. Pourtant, cette protection n'est pas aussi simple à réaliser qu'une porte blindée ou une alarme antivol dans un appartement. Beaucoup de personnes doivent avoir souvent accès à l'ordinateur et, par le biais des télécommunications, l'ensemble des individus concernés par un système informatique donné est fort difficile à circonscrire.

Par ailleurs, la sécurité informatique est encore auréolée d'un certain flou provenant de l'absence d'une véritable définition de la fraude informatique en tant que concept. La relation entre les deux domaines distincts que sont l'informatique et la délinquance est, en effet, peu aisée à définir.

De plus, le fraudeur informatique jouit, aux yeux de l'homme de la rue, d'un certain prestige : il est souvent considéré comme un petit malin, sinon comme le héros des temps modernes, tel qu'il est incarné dans le film *War Games*. Cette représentation peut s'expliquer par une certaine inquiétude vis-à-vis de l'informatique, dont le développement est si rapide que la technologie est souvent mal maîtrisée et ses conséquences mal appréhendées : certains craignent d'être dépassés par l'informatique, d'autres sentent peser une menace sur leurs libertés individuelles... Enfin, les victimes, elles-mêmes, de la délinquance informatique semblent relativement démotivées : elles paraissent avoir parfois plus à perdre qu'à gagner en dénonçant publiquement les fraudeurs.

Sur le plan juridique, le champ d'étude s'avère donc particulièrement étendu. En premier lieu, il y a la falsification de données. Il s'agit de fraudes, généralement financières, commises par altération des données numériques composant des fichiers ou des programmes. La falsification peut être simple, notamment lorsque le fraudeur se contente d'utiliser le système existant en le vouant à un objet illicite. Mais elle peut être complexe lorsque le fraudeur crée lui-même des programmes ou des fichiers. Ces fraudes nécessitent une très bonne connaissance du système informatique et, plus généralement, du système d'information et du contrôle interne de l'organisation visée. De nombreux experts considèrent que seules ces fraudes peuvent être qualifiées d'informatiques.

Une autre catégorie porte sur la fraude assistée par ordinateur. Il s'agit là de fraudes « classiques », perfectionnées grâce à l'ordinateur, instrument performant dont l'utilité ne saurait échapper aux délinquants. Entrent notamment dans cette catégorie la tenue de fausses comptabilités par des dirigeants d'entreprises et la création de faux documents par la délinquance



Matra

La protection des logiciels

Deux attitudes extrêmes s'affrontent sur le terrain de la protection des logiciels : celle de Bill Gates, de Microsoft, qui, jusqu'au début de 1983, ne voyait aucun inconvénient aux recopies de logiciel, car cela avait, selon lui, pour effet d'élargir le marché dans des secteurs jugés non solvables (en avril 1983, Microsoft n'avait engagé qu'un seul procès au Japon); et l'attitude de Micropro, qui consistait à multiplier les procès lesquels, d'ailleurs, se terminaient souvent par des non-lieux.



Fait divers

Piratage informatique et football

Deux étudiants d'un collège américain, Caltech, ont eu l'aventureuse idée de pirater, à l'aide d'un ordinateur portable (en l'occurrence un Epson HX20) et d'un modem, les panneaux d'affichage d'un stade de football. Ils ont pu ainsi faire passer divers messages « humoristiques » avant de choisir en définitive de changer les noms des équipes en compétition. Ainsi, les spectateurs ébahis ont pu voir « UCLA contre Illinois » se transformer en « Caltech contre MIT » (MIT étant l'université rivale de celle des deux pirates).

Pour couronner le tout et savourer leur victoire, les jeunes gens se sont offert le luxe d'organiser un séminaire intitulé : « Comment venir à bout des problèmes de transmission et commander des panneaux d'affichage ». A quand « Sorbonne contre Jussieu » au Parc des Princes ?

(D'après Sequoia Press — Bulletin du 20.4.84.)

organisée. La diffusion des connaissances et des matériels informatiques aura pour conséquence de banaliser ce type de fraudes.

En troisième lieu, on distingue le vol de services ou d'informations techniques ou commerciales. Initialement, il s'agissait d'employés utilisant à des fins personnelles les ressources informatiques de leur entreprise. Le vol de services s'est maintenant étendu, du fait de la mise à la disposition du public de services informatisés toujours plus nombreux et plus accessibles. Certaines fraudes sont, en effet, désormais à la portée d'un enfant : interrogation non autorisée de banques de données à partir de réseaux publics, et singulièrement de terminaux Minitel, utilisation de réseaux ou de « temps-machine » sur le compte d'un tiers. Cela résulte pour partie d'une confusion entre l'informatique ludique et l'informatique professionnelle, entre la convivialité et la permissivité.

Enfin intervient le sabotage. Dans la mesure où l'informatique constitue un élément clé de l'entreprise, elle sera toujours une cible potentielle pour le sabotage. Cependant, l'informatique semble déjà moins vulnérable qu'elle ne l'a été vis-à-vis de ce type de délinquance.

La copie de fichiers ou de programmes constitue, quant à elle, une fraude informatique déjà ancienne. Elle a d'abord posé le problème de la définition et de la protection des données dites « sensibles », notamment les données nominatives et les secrets de fabrication. Elle connaît aujourd'hui un grand développement à travers la copie de produits commercialisés, et en particulier des progiciels pour micro-ordinateurs.

L'enjeu se situe actuellement au niveau juridique. Dans ce domaine, la situation est encore incertaine et préoccupe les éditeurs de logiciels. En l'état actuel des choses, il semblerait que seule la protection par le droit d'auteur soit efficace. En effet, le droit des brevets ne peut pas s'appliquer : le droit français l'exclut expressément (article 6 de la loi du 2 février 1968, modifié par la loi du 13 juillet 1978). Seul, le droit d'auteur peut être invoqué comme moyen de protection du logiciel au plan international. En France, à la suite de récentes décisions des tribunaux, la tendance est d'estimer que les logiciels, de par leur originalité, doivent être considérés comme des œuvres protégées par le droit d'auteur.

Mais il n'est pas évident que cette protection soit adéquate. Le Syndicat national de l'édition (S.N.E.) estime que la production des logiciels grand public (jeux, didacticiels...) ne se développera qu'à l'abri d'une protection juridique adéquate, nécessitant donc une modification de la loi actuelle sur le droit d'auteur.

Le système informatique a donc globalement trois relations possibles avec la fraude : il peut être un but, un environnement ou un moyen.





L'ordinateur devient un but lorsque le délinquant cherche à s'approprier matériels, programmes, fichiers ou temps de traitement pour les convertir en gains financiers.

L'environnement, quant à lui, est important pour ceux dont la fraude s'appuie sur des connaissances et des droits d'accès privilégiés, donc le plus souvent les employés d'un service. Néanmoins, des groupes externes ont souvent profité de l'environnement pour atteindre leurs buts (étudiants, terroristes, etc.).

Enfin, l'informatique comme moyen de perpétrer une fraude est comparable à une clé, et permet d'entrer dans un système sans laisser la moindre trace. Sont particulièrement visées les organisations dont l'information se rapporte à l'argent, les banques notamment, où les modifications illicites de fichiers sont parmi les plus fréquentes.

Pour faire face à la fraude informatique, il est nécessaire, surtout aux entreprises, pour lesquelles l'enjeu est vital, de bien connaître les risques, d'être informé des protections juridiques, de ne pas négliger les possibilités d'assurance et, surtout, de se tenir au fait des techniques disponi-

bles pour décourager, freiner ou détecter le processus frauduleux.

Tout d'abord, il faut faire un diagnostic sûr de la sécurité des installations et applications informatiques. Puis, le cas échéant, il faut définir des mesures appropriées pour améliorer cette sécurité, ainsi que pour prévenir et détecter des actes frauduleux. Enfin, il faudra organiser ces mesures en plan d'action et mettre en place le suivi de ce plan. Pour tout cela, deux éléments s'avèrent essentiels : une méthodologie et des moyens humains adéquats.

La méthodologie commence par la définition claire du périmètre et des objectifs de la protection. Elle porte ensuite sur l'évaluation de la sécurité de l'environnement informatique général, le contrôle des accès et celui des applications.

L'aspect humain consiste, pour une entreprise, à faire appel à des auditeurs-conseils. Leur action est plus souvent orientée vers la prévention des risques que vers la détection des fraudes. Mais, du fait de leurs acquis techniques et méthodologiques, ils se positionnent aussi comme de très précieux auxiliaires lorsqu'il s'agit d'intervenir après l'« accident ».



Le film War Games a rendu le fraudeur informatique sympathique. Les petits malins sont devenus les héros des temps modernes. (Coll. G. Troussier.)

Nage ou meurs

La programmation de notre jeu d'aventures touche à sa fin. Voyons comment traiter les deux derniers lieux spéciaux de la Forêt hantée, le marais et le village, et achevons le listage de Digitaya.

Il y a dans la Forêt hantée deux endroits qui restent à analyser : le village et le marais. Intéressons-nous d'abord à ce dernier. N'oubliez pas qu'avant toute programmation proprement dite il faut décider d'une intrigue précise, qui intégrera tous les lieux dont elle a besoin. Il n'est pas nécessaire qu'elle soit très complexe, mais chaque endroit, si on veut l'exploiter à fond, exigera beaucoup d'efforts — et consommera énormément d'espace mémoire !

Le marais

Les règles de base sont les suivantes : dès que le joueur pénètre dans le marais, il s'y enfonce. Il peut toujours faire usage des commandes normales ; mais il lui est impossible de sortir de là. Une seule solution — nager. Mais cela n'est possible que si l'on transporte avec soi moins de deux objets. Pour échapper à la noyade, il sera donc parfois obligatoire de se débarrasser de l'un d'eux. Chaque objet abandonné dans le marais est, dans ce cas, définitivement perdu.

```
5085 SN*="VOUS COULEZ ET VOUS VOUS NOYEZ AVEC ELEGANCE" GOSUB
5500
5090 END
```

Les lignes 4870 à 4917 permettent un recours aux commandes classiques et font usage de routines déjà définies. Si, dès l'entrée dans le marais, le joueur décide de se séparer d'un objet, la routine LAISSER s'en chargera. Mais elle le replacera dans l'inventaire principal, IV\$(,), en se servant de la valeur actuelle de P (le pointeur de position). Cela veut dire que, pour le programme, l'objet se trouve désormais dans le marais. Si nous voulons qu'il disparaisse à jamais, nous allons devoir modifier l'entrée correspondante en IV\$(,).

IV\$(F,2) garde en mémoire la position de l'objet F. Il contient généralement le numéro attribué à l'endroit en question ; il peut être aussi d'une valeur de — 1 si l'objet est transporté par le joueur. Pour que celui-ci ne puisse plus jamais le retrouver, il faut que IV\$(F,2) ne soit plus interprété de cette manière. La ligne 4915 lui affecte donc la valeur — 2. F a déjà été spécifié grâce à la routine LAISSER.

Le joueur peut aussi décider de NAGER, et le programme passe alors à la ligne 4930. Le programme vérifie d'abord quels sont les objets qu'il emmène avec lui. S'il en a moins de deux, il lui est aussitôt possible de nager et de quitter le marais. Si ce n'est pas le cas, il a encore la ressource d'en abandonner un ; s'il néglige de le faire, il coule et se noie.

La routine NAGER permet au joueur de s'en sortir, et de préciser dans quelle direction il veut aller. A l'origine, le chiffre contenu en DATA est 00000000, indiquant qu'il n'existe aucune sortie. La ligne 5050 change tout cela et renvoie à la routine chargée d'indiquer les directions possibles. Le malheureux peut donc faire un choix parmi ces issues et quitter le marais. La ligne 5087 remet tout à zéro ; en cas de passage ultérieur dans le marais, il n'aura plus de moyen de s'en tirer.

Le village

C'est en le traversant que le joueur pourra quitter la Forêt hantée ; mais il n'est pas encore au bout de ses peines. En effet, les règles sont les suivantes :

- le village est entouré d'un mur apparemment infranchissable ;
- la seule porte qui existe est surveillée par un garde ;
- celui-ci devra être tué avec le fusil si le joueur veut entrer.

```
4870 REM **** S/P MARAIS ****
4875 SF=1
4880 SN*="VOUS VOUS ENFONCEZ DANS LE MARAIS":GOSUB
5500
4885 PRINT:INPUT"INSTRUCTIONS":IS#
4890 GOSUB2500:REM ANALYSE REPONSES
4895 IF F=0 THEN 4885:REM NON VALABLE
4900 GOSUB3000:REM COMMANDES NORMALES
4910 IF VB*="REGARDER" THEN GOSUB2000:GOTO4885
4915 IF VB*="LAISSER" THEN IV*(F,2)="-2":REM OBJET PERDU
POUR DE BON
4917 IF VF=1 THEN 4885:REM COMMANDE NORMALE
4920 REM ** NOUVELLES COMMANDES **
4925 IF VB* "NAGER" THEN SN*="JE NE COMPRENDS PAS":
GOSUB5500:GOTO4885
4930 REM ** NAGER **
4932 F=0
4935 FOR I=1 TO 2
4940 IF IC*(I) "" THEN F=F+1
4950 NEXT I
4955 IF F 2 THEN GOSUB5035:RETURN:REM FUITE A LA NAGE
4960 GOSUB 5000:RETURN:REM POSSESSION 2 OBJETS
5000 REM **** S/P 2 OBJETS ****
5010 SN*="LES OBJETS VOUS ALOURDISSENT ET VOUS COULEZ
PEU A PEU":GOSUB5500
5012 PRINT:INPUT"INSTRUCTIONS":IS#
5015 GOSUB2500:REM ANALYSE REPONSE
5020 IF VB* "LAISSER" THEN GOSUB5080:REM ON COULE !
5025 GOSUB3900:IV*(F,2)="-2":REM LAISSER OBJET
5030 IF HF=0 OR F=0 THEN 5080:REM ON COULE !
5035 REM **** FUITE A LA NAGE ****
5040 SN*="VOUS NAGEZ A TRAVERS LE MARAIS. QU VOULEZ-
VOUS ALLER ?":GOSUB5500
5050 EX*(2)="00000605":GOSUB2300:REM LES SORTIES POSSIBLES
5055 PRINT:INPUT"INSTRUCTIONS":IS#
5060 GOSUB2500:REM ANALYSE REPONSE
5062 IF F=0 THEN 5055:REM NON VALABLE
5065 GOSUB3500:REM DEPLACEMENT
5067 EX*(2)="00000000":REM AUCUNE SORTIE
5070 RETURN
5075 :
5080 REM **** S/P ON COULE ! ****
```

Une fois l'opération réalisée, il sera nécessaire d'ouvrir la porte avec la clé. Il faut donc distinguer deux parties dans le programme « Village » : d'abord, il s'agit de trouver une solution au danger que représente le garde; ensuite, il est nécessaire que la porte soit ouverte. On peut très bien envisager un scénario dans lequel le joueur, muni du fusil, arrive au village et tue le garde. Mais il se rend alors compte qu'une clé est nécessaire pour ouvrir la porte. Si le joueur en question ne dispose pas de la clé, il devra alors faire demi-tour pour partir à sa recherche.

Une seule fois!

Si, au retour du joueur, la description du village est la même qu'auparavant — c'est-à-dire que le garde est encore présent — cela n'aurait plus aucun sens.

```

5100 REM **** S/P VILLAGE ****
5102 SF=1
5105 SN$="LE VILLAGE EST ENTOURE D'UN GRAND
MUR.":GOSUB5500
5106 IF GF 0 THEN GOSUB5190:RETURN:REM PORTE
5107 SN$="IL Y A UN GARDE A LA PORTE D'ENTREE":GOSUB5500
5115 PRINT:INPUT"INSTRUCTIONS":IS$
5120 GOSUB2500:IF F=0 THEN 5115:REM NON VALABLE
5125 GOSUB3000:REM INSTRUCTIONS NORMALES
5130 IF VB$="REGARDER" THEN GOSUB2000:REM DESCRIPTION
5135 IF VB$="ALLER" AND MF=1 THEN RETURN
5140 IF VF=1 THEN 5115:REM INSTRUCTION SUIVANTE
5145 IF VB$ "TUER" THEN SN$="JE NE COMPRENDS PAS":
GOSUB5500:GOTO5115
5150 REM ** TUER **
5155 SN$="TUER LE GARDE ? AVEC QUOI ?":GOSUB5500
5160 SN$="ENTREZ UN OBJET OU I COMME INSTRUCTION":
GOSUB5500
5162 INPUT IS$:IF IS$="I" THEN 5115
5165 GOSUB2500:REM ANALYSE
5167 IF F=0 THEN 5160:REM NON VALABLE
5170 GOSUB3000:IF F=0 THEN SN$="JE N'EN VOIS PAS":
GOSUB5500:GOTO5160
5172 OV=F:GOSUB5450:REM OBJET POSSEDE ?
5174 IF HF=0 THEN SN$="VOUS N'EN N'AVEZ PAS":GOSUB5500:
GOTO5160
5175 IF F 1 THEN SN$="AUCUN INTERET !"
:GOSUB5500:GOTO5160
5180 SN$="VOUS TUEZ LE GARDE":GOSUB5500:GF=1
5185 :
5190 REM **** S/P PORTE FERMEE ****
5195 SN$="VOUS AVANCEZ ET ESSAYEZ D'OUVRIR LA PORTE"
5200 SN$=SN$+"MAIS ELLE EST FERMEE ET REFUSE DE S'OUVRIR":
GOSUB5500
5205 PRINT:INPUT"INSTRUCTIONS":IS$
5210 GOSUB2500:IF F=0 THEN 5205:REM NON VALABLE
5215 GOSUB3000:REM INSTRUCTIONS NORMALES
5220 IF VB$="REGARDER" THEN GOSUB2000:REM DESCRIPTION
5225 IF VB$="ALLER" AND MF=1 THEN RETURN
5230 IF VF=1 THEN 5205:REM INSTRUCTION SUIVANTE
5232 IF VB$="UTILISER" THEN 5240
5234 IF VB$="OUVRIR" THEN SN$="ET COMMENT ?":GOSUB5500:
GOTO5205
5235 SN$="JE NE COMPRENDS PAS":GOSUB5500:GOTO5205
5240 GOSUB3000:REM OBJET VALABLE
5242 OV=F:GOSUB5450:REM OBJET TRANSPORTE ?
5244 IF F=0 THEN SN$="IL N'Y EN A PAS !":GOSUB5500:
GOTO5205
5246 IF HF=0 THEN SN$="VOUS NE L'AVEZ PAS !":GOSUB5500:
GOTO5205
5248 IF F 3 THEN SN$="AUCUN INTERET PRATIQUE...":GOSUB
5500:GOTO5205
5250 REM ** SAUVE ! **
5255 SN$="VOUS OUVREZ LA PORTE, ET, REVETEZ DES HABITS
DU GARDE"
5260 SN$=SN$+"VOUS TRAVERSEZ LE VILLAGE":GOSUB5500
5270 END
    
```

Un aspect particulier du jeu consiste donc à faire en sorte que les différents éléments du jeu ne se reproduisent qu'une seule fois. Si le garde a été tué, cette « caractéristique » doit être enregistrée une fois pour toute dans la description du village.

La difficulté est plus apparente que réelle. Il suffit d'utiliser une variable qui précisera si le garde est ou non vivant. Appelons-là GF, et donnons-lui une valeur de 0 s'il est vivant, de 1 s'il est tué. Quand le joueur entrera dans le village, il faudra donc d'abord que le programme prenne connaissance de cette valeur.

La suite des événements présente moins de problèmes. Le joueur se dirige vers la porte. S'il a la clé et s'en sert pour l'ouvrir, il a réussi et quitte l'univers périlleux de la Forêt hantée. Sinon, rien n'est encore joué.

Pour mettre en œuvre nos deux dernières routines, il faut modifier, ligne 2720, la routine qui s'assure qu'un lieu présente ou non quelque chose de spécial. Une seule ligne suffit :

2720 ON P GOSUB 4590, 4870, 5100, 4590

Variantes de basic

Spectrum :

Dans la Forêt hantée, remplacez SN\$ par S\$, IS\$ par T\$, VB\$ par B\$, IV\$(,) par V\$(,), EX\$(I) par X\$(I) et IC\$(I) par I\$(I).

Procédez aux changements de lignes suivants :

```

2720 IF P=1 THEN GOSUB 4590
2722 IF P=2 THEN GOSUB 4870
2724 IF P=3 THEN GOSUB 5100
2726 IF P=4 THEN GOSUB 4590
    
```

```

4937 LET A$=IC$(I):GOSUB7000
4940 IF A$<>«» THEN LET F=F+1
    
```

```

5175 IF F<>1 THEN LET S$=«CELA»
:LET A$=V$(F,1):GOSUB7000
5177 IF F<>1 THEN LET S$=S$+«NE SERVIRA À RIEN»
:GOSUB5500:GOTO5160
    
```

```

5248 IF F<>3 THEN LET S$=«CELA»
:LET A$=V$(F,1):GOSUB7000
5249 IF F<>3 THEN LET S$=S$+«NE SERVIRA À RIEN»
:GOSUB5500:GOTO5205
    
```

Dans Digitaya, remplacez SN\$ par S\$, IS\$ par I\$, VB\$ par B\$, IV\$(,) par V\$(,) et IC\$(I) par I\$(I).

Changez ces lignes :

```

3650 LET S$=«CELA»:LET A$=V$(F,1):
GOSUB 7000
3655 LET S$=S$+«NE SERT À RIEN, LA FORCE AUGMENTE»
4520 LET V$(4,2)=STR$(INT(RND(1) *40+8))
    
```

BBC Micro :

Insérez cette ligne dans la Forêt hantée :

```
190 LET GF=0
```

Insérez cette ligne dans Digitaya :

```
4520 IV$(4,2)=RND(40)+8
```



Listage Digitaya

```

2960 REM ***** S/P PORT UTILISATEUR *****
2970 SF=1
2980 SN$="VOUS ETES PRESQUE TIRE D'AFFAIRE MAIS UN
EMPLOYE"
2990 SN$=SN$+"VOUS BARRE LA ROUTE ET DIT QU'IL A POUR
INSTRUCTIONS"
3000 SN$=SN$+"DE N'ACCEPTER QUE DES ENTREES. MAIS IL
ACCEPTE"
3010 SN$=SN$+"TOUTES LES CARTES DE CREDIT"
3020 GOSUB5880:REM FORMATAGE
3030 :
3040 PRINT:INPUT"INSTRUCTIONS";IS#
3050 GOSUB1700:REM ANALYSE REPONSE
3060 GOSUB1900:REM ACTIONS NORMALES
3070 IF MF=1 THEN RETURN:REM DEMI-TOUR
3080 IF VF=1 THEN3040:REM INSTRUCTION SUIVANTE
3090 IF VB# "DONNER" THENPRINT "JE NE COMPRENDS PAS":
GOTO3040
3100 :
3110 REM **: INSTRUCTION DONNER **
3120 GOSUB5730:REM OBJET ACCEPTABLE ?
3130 IFF=0THENPRINT "IL N'Y EN N'A PAS":GOTO3040:REM
INSTRUCTION SUIVANTE
3140 :
3150 REM **: CARTE CREDIT ? **
3160 IF F 5THENPRINT"IL N'ACCEPTÉ QUE LES CARTES
DE CREDIT":GOTO3040
3170 :
3180 REM **: CARTE POSSEDEE ? **
3190 OV=5:GOSUB5830
3200 IFHF=0THENPRINT"VOUS N'EN N'AVEZ PAS !":GOTO 3040
3210 :
3220 SN$="L'EMPLOYE PREND LA CARTE ET VOUS REMERCIE"
3230 GOSUB5880:REM FORMATAGE
3240 SN$="VOUS PASSEZ LA BARRIERE ET ENTREZ DANS LE
PORT UTILISATEUR"
3250 GOSUB5880:REM FORMATAGE
3260 :
3270 REM **: DIGITAYA ? **
3280 OV=6:GOSUB5830
3290 IF HF=1 THEN 3380:REM GAGNE
3300 :
3310 REM **: PERDU **
3320 SN$="VOUS ETES SORTI VIVANT DES ENTRAILLES DE
LA MACHINE"
3330 SN$=SN$+"MAIS VOUS AVEZ ECHOUÉ"
3340 SN$=SN$+"PUISQUE VOUS NE RAMENEZ PAS DIGITAYA"
3350 GOSUB5880:REM FORMATAGE
3360 END
3370 :
3380 REM **: GAGNE **
3390 SN$="FELICITATIONS, VOUS AVEZ MENE A BIEN VOTRE
MISSION"
3400 SN$=SN$+ "ET EXTIRPE DIGITAYA"
3410 SN$=SN$+"DES ENTRAILLES DE LA MACHINE"
3420 GOSUB5880:REM FORMATAGE
3430 END
3440 :
3450 REM ***** S/P PORT CASSETTE *****
3460 SF=1
3470 SN$="UNE FORCE IRRESISTIBLE"
3480 SN$=SN$+"VOUS SUBMERGE PEU-A-PEU"
3490 GOSUB5880:REM FORMATAGE
3500 NS=0:REM DENOMBRE LES INSTRUCTIONS
3510 REM **: INSTRUCTIONS **
3520 NS=NS+1:IFNS 3THEN3770:REM ASPIRE !
3530 PRINT:INPUT"INSTRUCTIONS";IS#
3540 GOSUB1700:REM ANALYSE INSTRUCTIONS
3550 GOSUB1900:REM ACTIONS NORMALES
3560 IF MF=1THENMF=0:PRINT"VOUS NE POUVEZ
BOUGER":GOTO3510
3570 IF VF=1THEN3510:REM INSTRUCTION SUIVANTE
3580 IF VB# "UTILISER" THENPRINT "JE NE COMPRENDS
PAS":GOTO3510
3590 REM **: INSTRUCTION : UTILISER **
3600 GOSUB5730:REM OBJET VALABLE ?
3610 IFF=0THENPRINT"IL N'Y EN A PAS !":GOTO3510
3620 :
3630 REM **: OBJET : ACTIVATEUR TAMPON ? **
3640 IF F=8 THEN3680:REM OK
3650 SN$="CELA NE SERT A RIEN, LA FORCE AUGMENTE"
3660 GOSUB 5880:GOTO3510:REM INSTRUCTION SUIVANTE
3670 :
3680 OV=8:GOSUB5830:REM ACTIVATEUR TAMPON ?
3690 IFHF=0THENSN$="VOUS N'EN N'AVEZ PAS !":GOSUB5880:
GOTO3510
3700 :
3710 REM **: SAUVE ! **
3720 SN$="L'ACTIVATEUR TAMPON S'OPPOSE A LA FORCE"
3730 SN$=SN$+"QUI DISPARAIT PEU A PEU"
3740 GOSUB5880:REM FORMATAGE
3750 RETURN
3760 :
3770 REM **: ASPIRE VIVANT ! **
3780 SN$="LA FORCE EST TROP FORTE ET VOUS ETES
ASPIRE"
3790 SN$=SN$+"A TRAVERS LE PORT CASSETTE DISPARAISANT
DANS LE NEANT MAGNETIQUE"
3800 GOSUB 5880:REM FORMATAGE
3810 END
4180 REM ***** S/P APPAREIL TROIS ETATS *****
4190 SF=1
4200 SN$="UNE PANCARTE ANNONCE E/S MAIS COMME
VOUS VOUS Y RENDEZ"
4210 SN$=SN$+"UN CONTROLEUR HURLE TICKET SIOUPLAIT"
4220 GOSUB5880:REM FORMATAGE
4230 :
4240 REM **: INSTRUCTIONS **
4250 PRINT:INPUT"INSTRUCTIONS";IS#
4260 GOSUB1700:GOSUB1900:REM ANALYSE
4270 IFMF=1 THEN RETURN
4280 IFVF=1 THEN4240:REM "INSTRUCTION SUIVANTE
4290 IFVB# "DONNER"ANDVB# "OFFRIR"THENPRINT"JE NE
COMPRENDS PAS":GOTO4240
4300 REM **: INSTRUCTION : DONNER **
4310 GOSUB5730:REM OBJET VALABLE ?
4320 IFF=0THENPRINT"IL N'Y EN A PAS":GOTO4240:REM
INSTRUCTION SUIVANTE
4330 :
4340 REM **: OBJET : TICKET ? **
4350 IF F=4 THEN4400:REM OK
4360 SN$="LE CONTROLEUR SECQUE LA TETE ET DIT : "
4370 SN$=SN$+"JE NE PEUX ACCEPTER CA"
4380 GOSUB5880:GOTO4240:REM INSTRUCTION SUIVANTE
4390 :
4400 OV=4:GOSUB5830:REM TICKET POSSEDE ?
4410 IFHF=0THENPRINT "VOUS N'AVEZ PAS LE TICKET":
GOTO4240
4420 :
4430 REM **: OK **
4440 SN$="LE CONTROLEUR ACCEPTE TICKET ET VOUS PERMET"
4450 SN$=SN$+"DE FRANCHIR LA BARRIERE"
4460 GOSUB5880:REM FORMATAGE
4470 REM **: TICKET SUPPRIME INVENTAIRE **
4480 F=0
4490 FORJ=1TO4
4500 IF IC$(J)=IV$(4,1)THENIC$(J)="":J=4
4510 NEXT J
4520 IV$(4,2)=STR$(INT(RND(TI)*40+8)):REM TICKET REPLACE
AILLEURS
4530 P=15:MF=1:RETURN

```



Bull au standard

Compatible avec les grands standards internationaux, le Bull Micral 30 concurrence l'IBM PC sur son propre terrain, tout en étant connectable aux autres ordinateurs de Bull.

Dès l'origine, Bull Micral 30 a été conçu pour répondre aux attentes du marché européen et mondial, tant il est vrai que Bull entend assumer sa responsabilité dans la compétition micro-informatique à l'échelle internationale.

Dans cet esprit, ce produit a été pensé en vue de son implantation dans un environnement européen. A titre d'exemple, il répond aux normes de la République fédérale d'Allemagne. Il dispose également de claviers en six versions différentes (française, italienne, espagnole, allemande, etc.), que le micro-ordinateur est capable de reconnaître immédiatement sans intervention du logiciel.

Conforme aux standards actuels du marché au travers du système d'exploitation MS-DOS, du microprocesseur Intel 8088 (celui-là même de l'IBM PC), du bus, des interfaces, des mémoires de masse, du clavier et des codes internes, le Bull Micral 30 répond à la demande la plus large en faisant bénéficier l'utilisateur d'une part, d'un maximum d'équipements diffusés par Bull, tels les adaptateurs de communication, la carte à mémoire CP8... et, d'autre part, des produits du marché développés par des tiers.

Bull Micral 30 se définit comme « un matériel compatible aux configurations multiples ». Ainsi, à partir d'une même architecture (cabinet, alimentation, bus) sont constituées cinq versions de base différentes de l'unité centrale. Ces dernières diffèrent par le choix des éléments (mémoire de masse et mémoire centrale) qui leur sont affectés. La version à 128 K est disponible avec un ou deux lecteurs de disquettes de 360 K. La version intermédiaire, de 256 K de mémoire, dispose d'un lecteur de disquette et d'un disque dur (10 méga-octets), ou de deux lecteurs de disquette. C'est la configuration maximale qui est retenue pour la version haut de gamme, avec 384 K de mémoire.

Chaque unité centrale Bull Micral 30 est composée en standard de quatre modules : la carte unité centrale, la carte mémoire vive, la carte écran/imprimante, et la carte communication/mémoires de masse.

La carte unité centrale est équipée d'un microprocesseur 8088 d'Intel, d'une horloge 4,77 MHz et de 64 K de mémoire morte (ROM) pour le BIOS (logiciel d'environnement entrées-sorties). Elle peut disposer en option du coprocesseur arithmétique 8087 et offre en standard l'interface clavier et un générateur acoustique.

La carte mémoire vive se présente sous la forme d'un module électronique de 128, 256 ou



Un pionnier

Jacques Stern, l'actuel Président de Bull, est une des grandes figures de l'informatique française. Il fut à l'origine d'une des premières sociétés de services en informatique, la SESA. (Doc. Bull.)

384 K. L'augmentation de la capacité mémoire centrale se fait soit par substitution de la carte mémoire d'origine par une carte ayant une capacité supérieure, soit par addition d'une seconde carte mémoire (maximum 256 K) dès lors que la première carte est à capacité maximale (384 K). La capacité maximale utilisable est de 640 K (384 K + 256 K).

La carte écran/imprimante permet la connexion d'un écran monochrome en mode alphanumérique et semi-graphique, et d'une imprimante en liaison parallèle type Centronics.

La carte communication/mémoire de masse sert à la connexion d'une ou deux unités de disquettes intégrées dans le coffret unité centrale et d'un disque dur de 10 méga-octets lui aussi intégré dans l'unité centrale. Un bus au standard SCSI permet de raccorder un coffret disque externe de 10 M. Enfin, une liaison asynchrone est fournie en standard sur cette carte (V24-RS232C).

Trois modèles d'écrans professionnels peuvent être connectés sur les unités centrales Bull Micral 30 : soit monochrome à affichage vert ou ambre sur fond noir, soit couleur permettant de traiter texte et graphiques avec 16 couleurs disponibles. En mode texte, ces écrans antireflets affichent 25 lignes de 80 caractères avec un certain nombre d'attributs : clignotement, secret, inverse vidéo, souligné, demi-intensité. Ils disposent d'un registre de 256 caractères alphanumériques et semi-graphiques. Les images graphiques peuvent être traitées par l'adjonction d'une carte du commerce; la définition est alors de 350 x 720 points.

L'écran couleur est connecté sur l'interface RVB. Il admet deux résolutions en mode graphique : 200 x 320 points et 200 x 640 points. Le mode couleur nécessite l'adjonction de la carte graphique couleur (option). La taille de l'écran, couleur ou monochrome, est de 30 cm (12").



BULL MICRAL 30

PRIX

De 21 460 F à 45 220 F H.-T., selon la configuration.

UC

8088 à 4,77 MHz, coprocesseur arithmétique 8087 (option).

MÉMOIRE

RAM : 128 K, 256 K ou 384 K selon la version.
ROM (BIOS) : 64 K extensible jusqu'à 640 K.

AFFICHAGE ÉCRAN

Monochrome :
25 lignes
× 80 colonnes
texte; caractères
7 × 9 points dans
une fenêtre 9 × 14;
définition graphique :
350 × 720 points.
Couleur : 25 lignes
× 80 colonnes
texte; caractères
7 × 7 points dans
une fenêtre 8 × 8;
mode graphique :
200 × 320 points
ou 200 × 640 points,
16 couleurs.
Dimensions : 30 cm
diagonale (12").

LANGAGES DISPONIBLES

BASIC, GW-BASIC,
Macro-assembleur,
PASCAL, COBOL,
FORTRAN, SOUS
MS-DOS.

FORCES

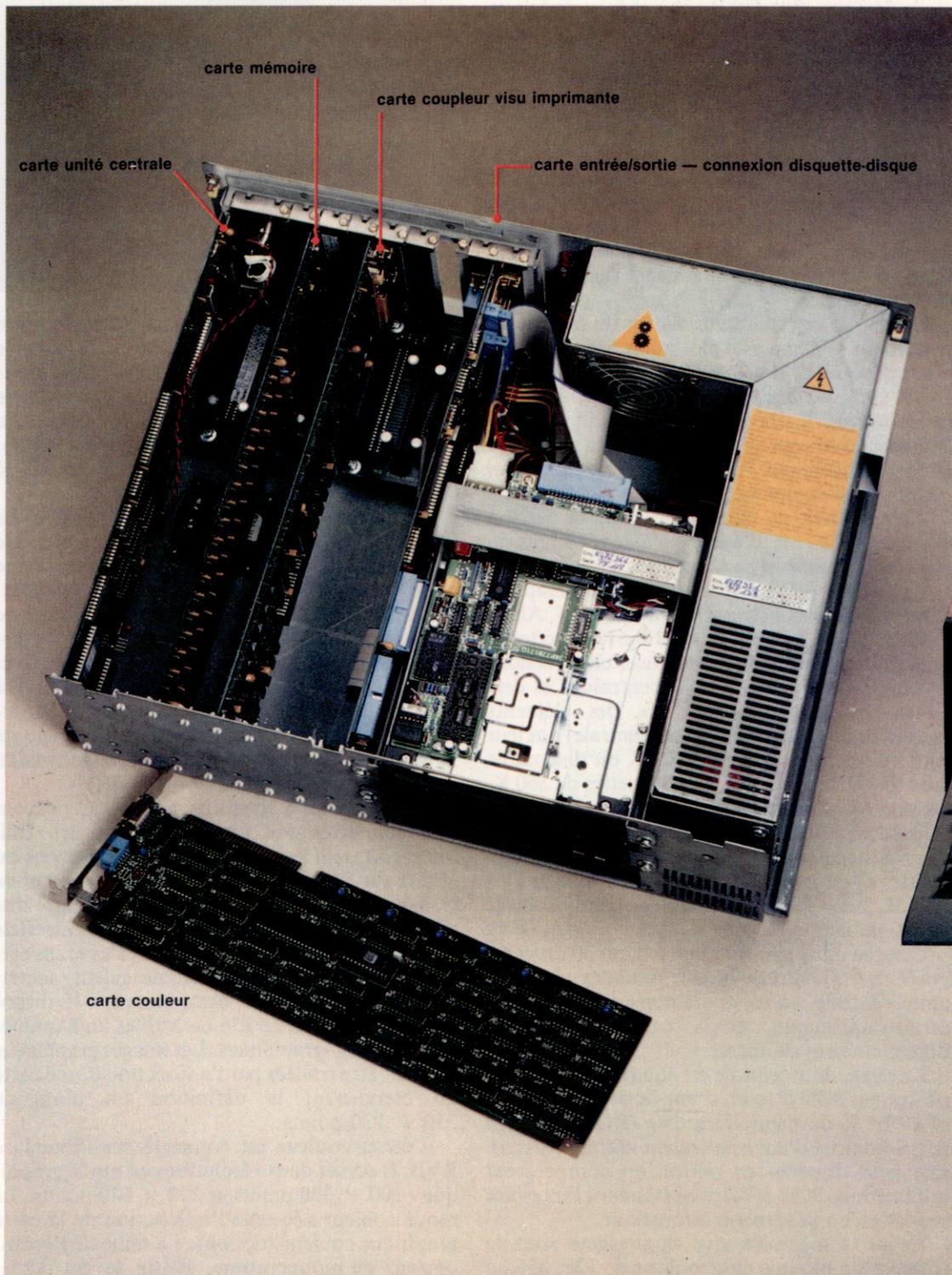
Véritablement compatible IBM PC, conforme aux standards européens, ouvert à la télématique (avec notamment l'utilisation possible de la carte à mémoire), vaste catalogue de logiciels, connectable aux ordinateurs universels de Bull ainsi qu'aux grands systèmes IBM.

Le clavier du micro-ordinateur Bull Micral 30 est un clavier standard (QWERTY ou AZERTY) de 83 touches sur lequel on distingue un bloc central de 58 touches alphanumériques, de signes, une touche « CAPS LOCK » équipée d'un voyant repère, une touche « ALT » pour générer les mots clés du langage BASIC; un bloc de 15 touches sur la droite regroupant le clavier numérique et les commandes curseur, des fonctions spécifiques et une touche « NUM LOCK » équipée d'un voyant repère; un bloc de 10 touches de fonctions programmables sur la gauche.

Double « univers logiciel »

Le clavier est séparé de l'unité centrale, à laquelle il est relié par un cordon spiralé.

En plus de ses caractéristiques de micro-ordinateur professionnel, Bull Micral 30 offre aux utilisateurs un double « univers logiciel ». MS-DOS, développé par Microsoft pour les micro-ordinateurs 16 bits, s'est imposé comme un standard et a entraîné le développement d'une très importante bibliothèque de logiciels, progiciels et applications. Prologue est le système





Le réseau

Pour mieux se lancer sur le marché de la micro-informatique, Bull a dû mettre sur pied un réseau de revendeurs qualifiés représentant environ 150 points de vente en France. (Cl. Bull.)



d'exploitation bien connu des utilisateurs des matériels Micral. Par ses puissants moyens de gestion et sa compatibilité 8 et 16 bits, Prologue a permis le développement de nombreuses applications sectorielles et fonctionnelles.

Les fonctionnalités de communications — accès aux bases de données, aux réseaux locaux, à la télématique avec, notamment, la carte à mémoire — sont rendues possibles par la mise à disposition de cartes options, procédures et émulations logicielles. Déjà dans sa configuration de base, le micro-ordinateur Bull Micral 30 permet de communiquer en mode asynchrone grâce

à l'interface V24-RS232C présente sur la carte communication/mémoires de masse.

Parallèlement à l'annonce de Bull Micral 30, le groupe publie un catalogue de logiciels qui propose une sélection de cent soixante-sept produits parmi les deux mille logiciels les plus réputés. Les logiciels sélectionnés ont été soumis à des tests standards d'évaluation destinés à vérifier entre autres leur fonctionnement sur Bull Micral 30. Le catalogue comprend notamment des progiciels diffusés sous la marque Bull; citons dBase III, Multiplan, Chart, Framework et Dialogue 2.

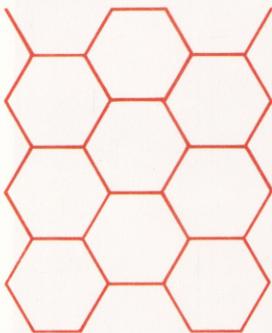
Le Bull Micral 30, fabriqué actuellement à Marcq-en-Barœul, dans l'unité de production de Bull Transac, capable de répondre à des montées en fabrication rapide, est livré aux vendeurs agréés de Bull depuis le mois de janvier 1985. Bull a, en effet, mis sur pied un réseau de revendeurs qualifiés, qui permettra une distribution segmentée et structurée, et un quadrillage précis du marché. En janvier 1985, une cinquantaine de revendeurs représentant cent-vingt à cent-cinquante points de vente doivent avoir reçu en France l'agrément de commercialisation du produit Bull Micral 30. La maintenance sera assurée soit par les revendeurs, soit par les services de maintenance Bull.

Tout cela permet à la politique de distribution élaborée par Bull de prendre en considération non seulement la réalité du marché actuel, mais aussi celle de son évolution à court et long termes.

Au bout des lignes

Le pavage revient à couvrir totalement une surface plane de motifs identiques. C'est le cas pour les carreaux des salles d'eau et des cuisines. Avec LOGO, assemblons diverses formes.

Trois pavages constitués



de tuiles hexagonales

Certains polygones réguliers peuvent donner lieu à des pavages : les carrés, les triangles équilatéraux et les hexagones (voir ci-contre). Les autres polygones ne le permettent pas. Pour le comprendre, prenons un exemple, le pentagone. Un pentagone régulier a des angles de 108° . Si vous réunissez trois pentagones autour d'un point central, vous n'avez utilisé que 324° , et il reste donc un vide. Si vous ajoutez maintenant un quatrième pentagone, l'angle total obtenu est de 432° , ce qui signifie que les formes se chevauchent. Les carrés, les triangles équilatéraux et les hexagones s'assemblent parfaitement parce que leurs angles (respectivement 90° , 60° et 120°) sont des quotients exacts de 360° .

L'écriture de procédures LOGO de pavage est très rapide. La meilleure approche est probablement de tracer le motif à partir du centre pour revenir vers le centre. Nous pouvons alors utiliser cette procédure de base dans une « superprocédure » qui déplace simplement la tortue au centre de la nouvelle figure à dessiner.

Les procédures suivantes dessinent un simple motif de carrés. L'assemblage de triangles équilatéraux et d'hexagones se ferait de manière similaire.

```
POUR PAVAGE
  LÈVEPLUME DONNEXY (-100) (90) POSEPLUME
  RÉPÈTE 5 [LIGNEVERTI DONNEXY (-100) (COORY - 40)]
FIN
```

```
POUR LIGNEVERTI
  RÉPÈTE 5 [CARRÉ 40 DONNEX COORX + 40]
FIN
```

```
POUR CARRÉ :C
  LÈVEPLUME GAUCHE 45
  AVANCE :C * (RACINECARRÉE) / 2
  DROITE 135 POSEPLUME
  AVANCE :C DROITE 90 AVANCE :C DROITE 90
  AVANCE :C DROITE 90 AVANCE :C DROITE 90
  LÈVEPLUME GAUCHE 135
  RECULE :C * (RACINECARRÉE) / 2 DROITE 45
FIN
```

Les pavages peuvent être obtenus à partir de figures bien plus complexes que des polygones réguliers. Les assemblages fondés sur des polygones réguliers offrent cependant de multiples ressources. De nombreux dessins de M.C. Escher sont des variations sur des pavages réguliers.

Pour construire des diagrammes plus complexes, une méthode directe modifie CARRÉ en remplaçant, par une procédure, la commande

AVANCE qui trace les côtés du carré. La règle d'or à observer est de faire correspondre à toute modification concernant le côté supérieur du carré une modification équivalente pour la base. De même, toute modification pour le côté droit doit avoir un équivalent à gauche, et inversement. Cela permet aux formes de pouvoir effectivement s'assembler.

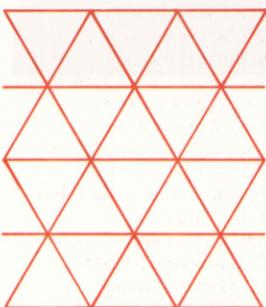
Pour tracer la forme élémentaire, il suffit de remplacer les commandes qui tracent les côtés du carré par des procédures traçant les nouvelles lignes. Le nouveau motif de pavage peut alors être dessiné en appelant PAVAGE. Voici le listage complet :

```
POUR CARRÉ :C
  LÈVEPLUME GAUCHE 45
  AVANCE :C * (RACINECARRÉE) / 2
  DROITE 135 POSEPLUME
  CÔTÉA :C DROITE 90 CÔTÉB :C DROITE 90
  CÔTÉC :C DROITE 90 CÔTÉD :C DROITE 90
  LÈVEPLUME GAUCHE 135
  RECULE :C * (RACINECARRÉE) / 2
  DROITE 45
FIN
```

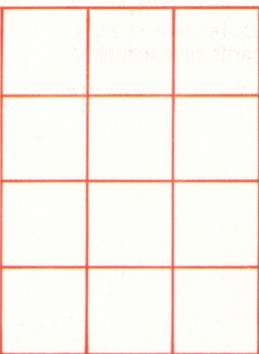
```
POUR CÔTÉA :C
  AVANCE :C / 4 GAUCHE 90
  RÉPÈTE 19 [AVANCE2 * 3,1416 * :C / 144 DROITE 10]
  GAUCHE 100 AVANCE :C / 4
FIN
```

```
POUR CÔTÉB :C
  GAUCHE 45 AVANCE :C / 2
  DROITE 90 AVANCE :C / 2
  GAUCHE 45 AVANCE :C - :C *
  (RACINECARRÉE) / 2
FIN
```

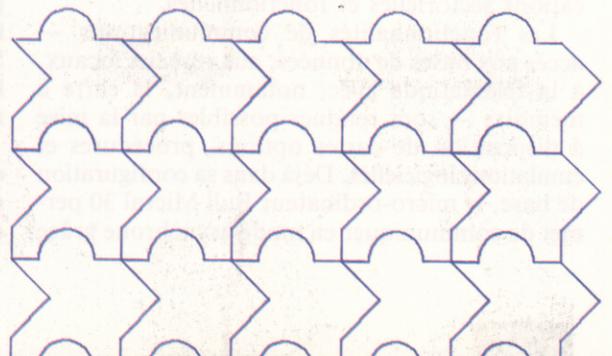
```
POUR CÔTÉC :C
  AVANCE :C / 4 DROITE 90
  RÉPÈTE 19 [AVANCE2 * 3,1416 * :C / 144 GAUCHE 10]
  DROITE 100 AVANCE :C / 4
FIN
```



de tuiles en forme de triangles équilatéraux



de tuiles carrées



POUR CÔTÉD :C
 AVANCE :C - :C * (RACINECARRÉE) /2 DROITE 45
 AVANCE :C / 2 GAUCHE 90
 AVANCE :C / 2 DROITE 45
 FIN

La technique d'Escher était bien sûr plus complexe, puisqu'il modifiait également les formes au fur et à mesure qu'elles « traversaient » le pavage. Une procédure LOGO capable de cela serait la bienvenue...

Avec de la colle

Une approche différente de la question est donnée dans l'excellent livre de Harold Abelson et Andrea di Sessa, *Turtle Geometry*. On peut la résumer de la sorte : soit une forme dessinée dans un carré, le dessin d'une tuile par exemple; assemblons-en quatre pour faire un plus grand carré. A nouveau, assemblons plusieurs de ces grands carrés, et ainsi de suite...

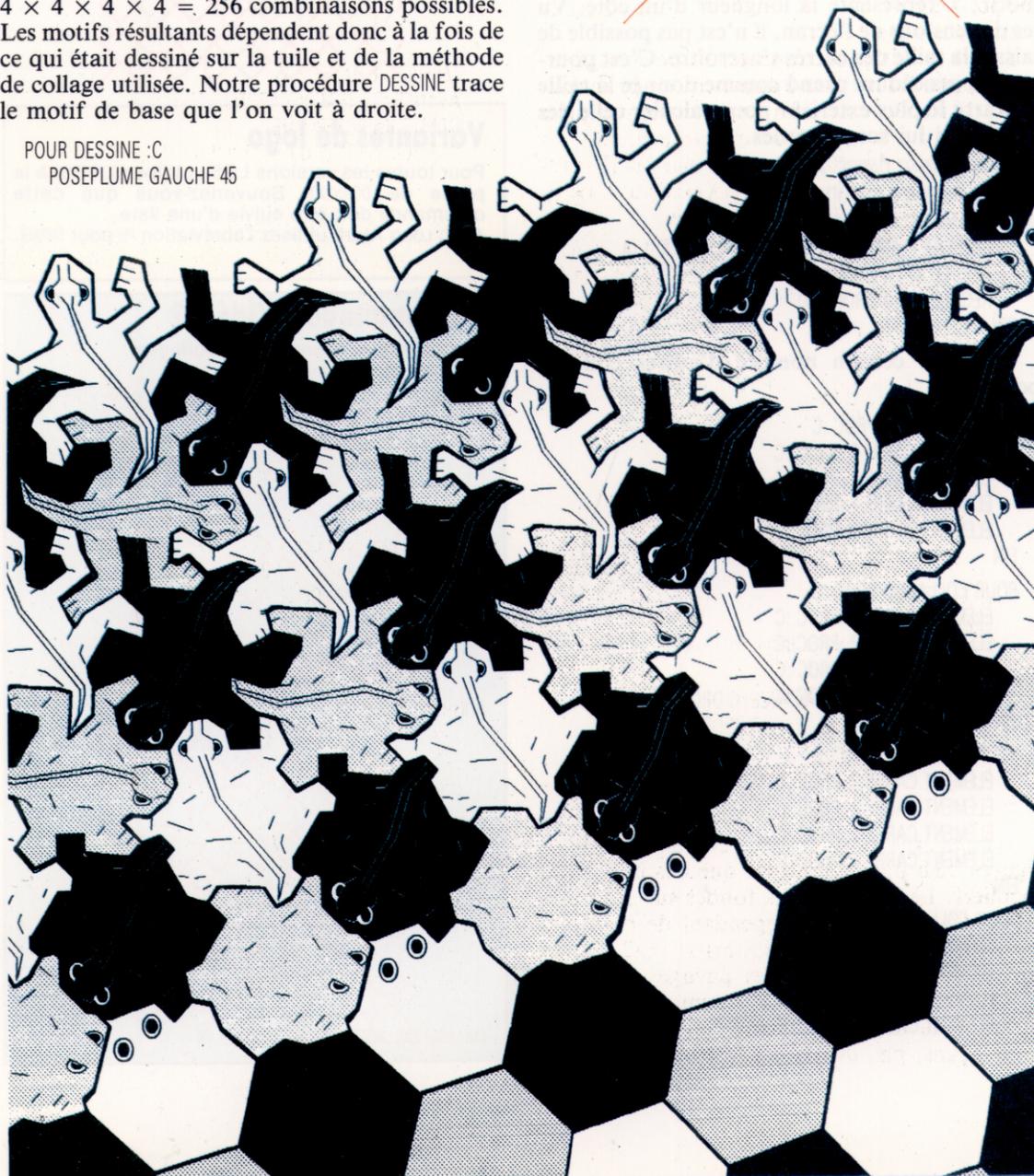
Il existe plusieurs méthodes de « collage ». Plus précisément, puisque chacun des quatre éléments peut être orienté selon quatre directions, il y a $4 \times 4 \times 4 \times 4 = 256$ combinaisons possibles. Les motifs résultants dépendent donc à la fois de ce qui était dessiné sur la tuile et de la méthode de collage utilisée. Notre procédure DESSINE trace le motif de base que l'on voit à droite.

POUR DESSINE :C
 POSEPLUME GAUCHE 45

AVANCE :C * (RACINECARRÉE) / 2
 RECULE :C * (RACINECARRÉE) / 2
 GAUCHE 90 AVANCE :C * (RACINECARRÉE) / 2 LÈVEPLUME
 RECULE :C * (RACINECARRÉE) / 4 GAUCHE 45
 AVANCE :C / 2 GAUCHE 45
 AVANCE :C * (RACINECARRÉE) / 4
 LÈVEPLUME
 RECULE :C (RACINECARRÉE) / 2 GAUCHE 135

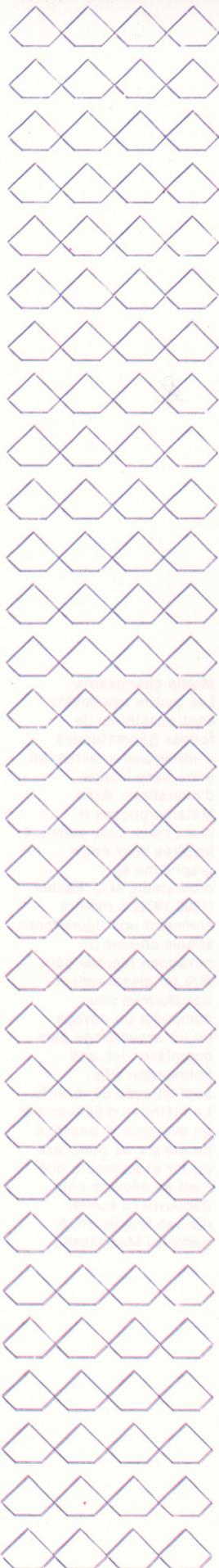
FIN

Voici une méthode possible de collage, COLLAGÉ1. Pour l'exécuter, tapez DESSINE LÈVEPLUME COLLAGÉ1 [DESSIN] 100. Le motif obtenu est indiqué ci-dessous.

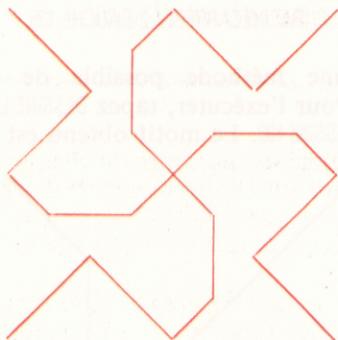


Motifs changeants

Les motifs assemblés sont constitués de formes géométriques s'imbriquant exactement, donc sans laisser d'interstices. Avec *Métamorphoses II* (dont nous nous sommes inspirés pour notre graphisme sur ordinateur), M.C. Escher (1898-1972) a montré comment une figure-motif simple comme un hexagone régulier peut être progressivement transformée pour composer un pavage ou un assemblage plus complexe : ici, une imbrication très sophistiquée de lézards. La méthode d'Escher met en évidence le passage du simple au complexe au fur et à mesure que l'œil se déplace pour découvrir la scène. (Cl. Ian McKinnell, à partir du Macintosh.)



```
POUR COLLAGE1 :PROC :C
  ÉLÉMENT.CARRÉ 0 :PROC :C
  ÉLÉMENT.CARRÉ 0 :PROC :C
  ÉLÉMENT.CARRÉ 0 :PROC :C
  ÉLÉMENT.CARRÉ 0 :PROC :C
FIN
```



ÉLÉMENT.CARRÉ trace l'un des éléments carrés constitutifs du grand carré. Il accepte trois données en entrée. :A détermine l'orientation du carré; :PROC est le nom de la procédure qui trace le motif; :C représente la longueur d'un côté. Vu les dimensions de l'écran, il n'est pas possible de laisser la taille des carrés s'accroître. C'est pourquoi la procédure prend comme donnée la taille du carré le plus extérieur pour calculer celle des carrés qui lui sont internes.

```
POUR ÉLÉMENT.CARRÉ :A :PROC :C
  AVANCE :C / 4 DROITE 90
  AVANCE :C / 4 DROITE 90 * :A
  EXÉCUTE PHRASE :PROC :C / 2 GAUCHE 90 * :A
  RECULE :C / 4 GAUCHE 90
  RECULE :C / 4 DROITE 90
FIN
```

Voici un certain nombre d'autres collages possibles :

```
POUR COLLAGE2 :PROC :C
  ÉLÉMENT.CARRÉ 0 :PROC :C
  ÉLÉMENT.CARRÉ 1 :PROC :C
  ÉLÉMENT.CARRÉ 2 :PROC :C
  ÉLÉMENT.CARRÉ 3 :PROC :C
FIN
```

```
POUR COLLAGE3 :PROC :C
  ÉLÉMENT.CARRÉ 0 :PROC :C
  ÉLÉMENT.CARRÉ 2 :PROC :C
  ÉLÉMENT.CARRÉ 3 :PROC :C
  ÉLÉMENT.CARRÉ 1 :PROC :C
FIN
```

```
POUR COLLAGE4 :PROC :C
  ÉLÉMENT.CARRÉ 3 :PROC :C
  ÉLÉMENT.CARRÉ 2 :PROC :C
  ÉLÉMENT.CARRÉ 1 :PROC :C
  ÉLÉMENT.CARRÉ 0 :PROC :C
FIN
```

```
POUR COLLAGE5 :PROC :C
  ÉLÉMENT.CARRÉ 2 :PROC :C
  ÉLÉMENT.CARRÉ 1 :PROC :C
  ÉLÉMENT.CARRÉ 0 :PROC :C
  ÉLÉMENT.CARRÉ 3 :PROC :C
FIN
```

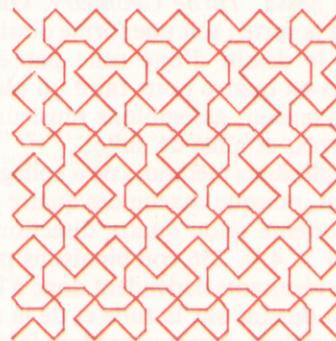
DESSIN est le nom d'une commande qui trace une figure à une seule donnée. Nous l'avons assemblé par collage en écrivant COLLAGE1 [DESSIN] 100. En étudiant cette ligne de commande, on voit que la syntaxe COLLAGE1 [DESSIN] peut être considérée à elle seule comme une commande qui nécessite un seul nombre pour tracer une figure. Nous pouvons donc utiliser [COLLAGE1 [DESSINE]] en entrée à une autre commande COLLAGE. Par exemple :

```
COLLAGE2 [COLLAGE1 [DESSIN]] 100
```

Et cela ne fait que commencer! Ainsi :

```
COLLAGE3 [COLLAGE2 [COLLAGE1 [DESSIN]]] 100
```

Nous avons alors $256 \times 256 \times 256$ combinaisons de collages à trois niveaux. Et pourquoi s'arrêter à trois?



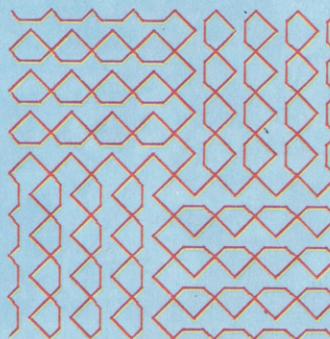
Variantes de logo

Pour toutes les versions LCS1, utilisez DONNEPOS à la place de DONNEXY. Souvenez-vous que cette commande doit être suivie d'une liste.

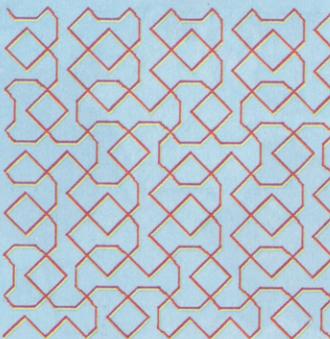
Avec LOGO Atari, utilisez l'abréviation PH pour PHRASE.

Vous manquez d'idées?

Voici d'autres exemples de collages :



COLLAGE1 [COLLAGE4 [COLLAGE5 [DESSIN]]] 100



COLLAGE4 [COLLAGE3 [COLLAGE3 [DESSIN]]] 100



Pour voir...

Nous allons concevoir un programme qui permettra à notre robot d'explorer une zone donnée et d'afficher la forme d'un objet trouvé à l'intérieur de la zone.

Pour permettre à notre robot d'explorer un espace déterminé, il est nécessaire de mettre au point un parcours efficace de reconnaissance. Nous devons d'abord penser à une analyse afin que le robot puisse avancer ou reculer dans la zone de recherche de l'objet. Lors de la localisation de l'objet, nous pourrions décider de faire le tour de ses côtés avant de balayer le reste de la zone. Un tel algorithme est cependant difficile à programmer et peut entraîner des problèmes si la zone à explorer renferme plus d'un objet. Il serait également possible de contourner simplement l'objet et de poursuivre l'exploration. Les étapes d'un tel algorithme seraient les suivantes :

RÉPÉTER

RÉPÉTER

Rechercher un objet vers l'avant

JUSQU'À la rencontre de la bordure de la zone définie ou la découverte de l'objet

Monter d'une « unité de largeur »

Tourner

JUSQU'À la rencontre de la bordure supérieure de la zone définie

À la suite de cette analyse horizontale, le robot n'aura pas exploré toute la zone. En utilisant ce parcours de reconnaissance, la zone située derrière tout objet — la « zone aveugle » — ne sera pas explorée. Un balayage vertical (c'est-à-dire à 90° du premier) est au moins nécessaire.

Avant que notre robot puisse effectuer avec succès un balayage, une modification mineure doit être apportée aux deux capteurs avant. Puisque les roues du robot dépassent sur les côtés, elles risquent de heurter un objet. Pour résoudre ce problème, nous devons ajouter un bouclier à l'avant de chaque capteur pour protéger les roues. Chaque bouclier doit être de forme rectangulaire, environ 95 par 25 mm, et doit être fait d'un matériau léger (plastique rigide ou métal). Veillez à ce que le poids du matériau ne soit pas trop lourd, ce qui mettrait le capteur en position fermée. Les boucliers doivent être montés sur les capteurs au moyen de ruban adhésif ou de colle, et doivent être positionnés afin de couvrir complètement la trajectoire des roues. Vérifiez le bon fonctionnement des capteurs ; ils doivent toujours s'ouvrir et se fermer.

Le programme qui permet au robot d'effectuer une exploration de zone a été écrit en utilisant un parcours de reconnaissance horizontal et un parcours de reconnaissance vertical. Lorsque le

robot a terminé son parcours sur chaque ligne de balayage, une zone correspondante de l'écran de l'ordinateur est remplie de couleur. À la fin, une représentation picturale de l'objet est affichée à l'écran.

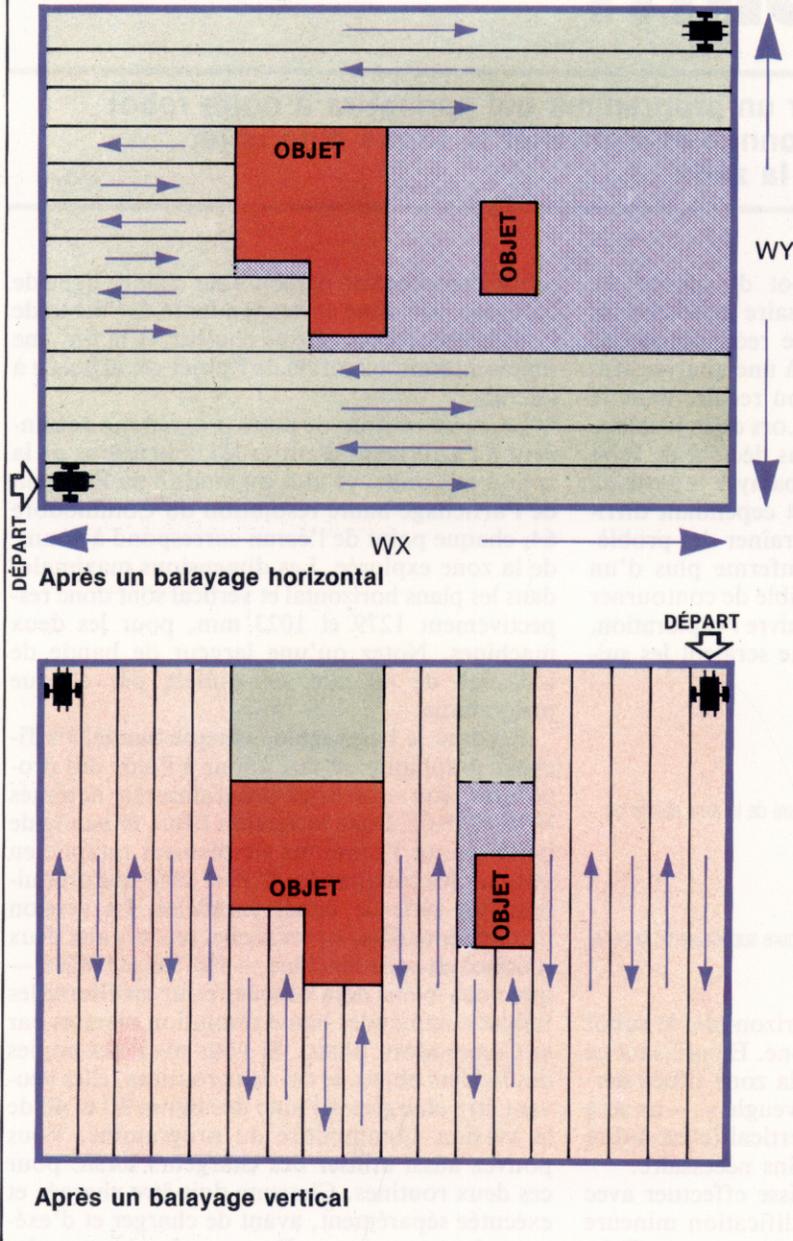
Les deux versions de notre programme demandent à l'utilisateur d'entrer les dimensions de la zone à explorer. À l'aide du mode 4 du BBC, ou de l'affichage haute résolution du Commodore 64, chaque point de l'écran correspond à 4 mm² de la zone explorée. Les dimensions maximales dans les plans horizontal et vertical sont donc respectivement 1279 et 1023 mm, pour les deux machines. Notez qu'une largeur de bande de balayage de 40 mm est utilisée par chaque programme.

Pendant le balayage de chaque bande, l'affichage graphique est mis à jour à l'aide des procédures (ou des sous-programmes) nommés XPLOT et YPLOT. Dans la version BBC, la bande de balayage sur l'écran est simplement remplie en utilisant les commandes MOVE et DRAW qui produisent une série de lignes parallèles. La version Commodore 64 a, en revanche, recours aux deux routines en code machine — PLOTSUB et LINESUB — que nous avons déjà conçues pour améliorer les faibles commandes haute résolution offertes par le Commodore BASIC. Si vous avez des copies des fichiers objets de ces deux routines, elles peuvent être chargées à l'aide des lignes 30 et 40 de la version Commodore du programme. Vous pouvez aussi utiliser des chargeurs BASIC pour ces deux routines. Chacune doit être chargée et exécutée séparément, avant de charger et d'exécuter le programme. Dans ce dernier cas, les lignes 30 et 40 peuvent être supprimées.

Outre cette zone aveugle, subsistant parfois après deux balayages, plusieurs autres difficultés peuvent être rencontrées. Parmi celles-ci, soulignons que l'efficacité du programme varie selon la forme de l'objet et son orientation. Il sera plus efficace avec des formes rectangulaires, orientées le long des axes horizontal et vertical. Les formes triangulaire et circulaire produiront des zones aveugles plus importantes puisque le parcours de reconnaissance aura tendance à « oublier » les irrégularités des objets rencontrés. À titre d'exercice, modifiez le programme donné de façon à effectuer un autre balayage vertical. Le robot peut aussi avoir des difficultés s'il rencontre un objet pendant la manœuvre qui lui permet de commencer une nouvelle ligne. Afin de simplifier les choses, l'état des capteurs n'est pas vérifié pendant cette manœuvre.



Zones aveugles



Touche :

Zone balayée horizontalement

Zone balayée verticalement

Zone non balayée

La façon la plus simple pour balayer une zone rectangulaire consiste à faire deux balayages orientés à 90° l'un par rapport à l'autre. Lorsqu'un objet est rencontré ou lorsque la bordure de la zone est atteinte, le robot monte d'une bande et continue le balayage dans la direction opposée. Avec cette méthode, des « zones aveugles » suivront tout objet trouvé. Cependant, le second balayage explorera la majeure partie de la zone qui n'avait pas encore été vue. Des imprécisions subsisteront toujours en raison de la largeur de la bande de balayage utilisée et de l'orientation respective des objets. (Cl. Kevin Jones.)

Listage BBC Micro

```

10REM **** RECONNAISSANCE DE FORME BBC ****
20MODE 4
30PROCinitialise
40PROCscan dimensions
50PROCchoriz scan
60PROCvert scan
70END
80DEF PROCchoriz scan
90 target=wx:sense=left:xstart=0
100REPEAT
110REPEAT
120PROCmove(forwards,dx):x=x+dx
130UNTIL(?DATREG AND 192) neither bumpers OR x=target
140PROCxplot
150dx=-dx:y=y+dy:xstart=x
160IF target=0 THEN target=wx ELSE target=0
170IF y<wy THEN PROCnext strip(sense)
180IF sense=right THEN sense=left ELSE sense=right
190UNTIL y =wy
200ENDPROC
210:

```

```

220DEF PROCvert scan
230IF x=0 THEN sense=left:dx=width ELSE sense=right:dx=-width
240dy=-dw:target=0:ystart=y
250PROCturn(sense,90)
260REPEAT
270REPEAT
280PROCmove(forwards,dy):y=y+dy
290UNTIL(?DATREG AND 192) neither bumpers OR y=target
300PROCyplot
310dy=-dy:k=k+dx:ystart=y

```

```

100 GOSUB5100:REM QUITTER MODE HAUTE RESOLUTION
110 END
120 :
1000 REM **** S.P. INITIALISER ****
1010 DDR=56579:DATREG=56577
1020 POKE DDR,15:POKE DATREG,1
1030 FW=4:BW=2:LF=6:RT=0
1040 PD=3,34446:PA=375/90
1050 RB=128:LB=64:BB=0:NB=132
1060 WD=40:DW=WD/10:X=0:Y=0:DX=DW:DY=WD
1070 REM ** M/C ADRESSES DE DEPART **
1080 HIRES=49422:LINESUB=49934
1090 RETURN
1100 :
2000 REM **** DIMENSIONS BALAYAGE ****
2010 INPUT"X DIMENSIONS EN MM":WX
2020 IF WX/4>319 THEN 2010
2030 INPUT"Y DIMENSIONS EN MM":WY
2040 IF WY/4>199 THEN 2030
2050 WX=WD*INT(WX/WD):WY=WD*INT(WY/WD)
2060 RETURN
2070 :
3000 REM **** BALAYAGE HORIZONTAL ****

```

Listage Commodore 64

```

10 REM **** RECONNAISSANCE DE FORME CBM ****
20 DN=0:REM IF CASS THEN DN=1
30 IF A=0 THEN A=1:LOAD"PLOTSUB.HEX",DN,1
40 IF A=1 THEN A=2:LOAD"LINESUB.HEX",DN,1
50 GOSUB1000:REM INITIALISER
60 GOSUB2000:REM DIMENSIONS DE BALAYAGE
70 GOSUB5000:REM PASSER AU MODE HAUTE RESOLUTION
80 GOSUB3000:REM BALAYAGE HORIZONTAL
90 GOSUB4000:REM BALAYAGE VERTICAL

```

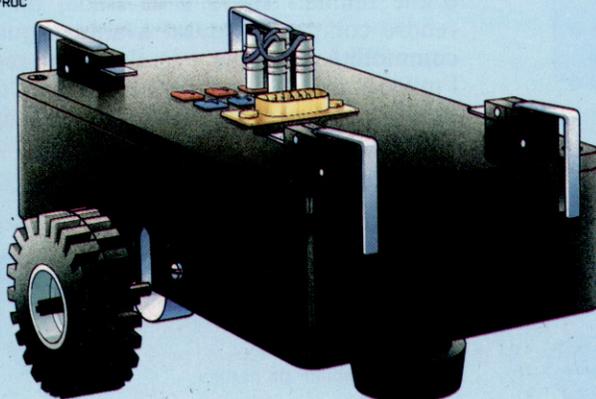


Suite du listage du BBC Micro

```

320IF target=0 THEN target=wx ELSE target=0
330PROCnext strip(sense)
340IF sense=right THEN sense=left ELSE sense=right
350UNTIL x =wx OR x =0
360ENDPROC
370:
380DEF PROCnext strip(way)
390PROCmove(backwards,30)
400PROCturn(way,90)
410PROCmove(forwards,width)
420PROCturn(way,90)
430ENDPROC
440:
450DEF PROCscan dimensions
460INPUT"X DIMENSION IN MM"iwx
470 IF wx 1279 THEN 460
480INPUT"Y DIMENSION IN MM"iwy
490 IF wy 1023 THEN 480
500wx=width*(wx DIV width)
510wy=width*(wy DIV width)
520CLS
530ENDPROC
540:
550DEF PROCinitialise
560DDR=&FE62:DATREG=&FE60
570?DDR=15:REM LINES 0-3 OUTPUT
580?DATREG=1:REM TURN ON RESET BIT
590forwards=4:backwards=2:left=5:right=0
600pd ratio=3.34446:pa ratio=375/90
610right bumper=120:left bumper=64
620both bumpers=0:neither bumpers=192
630width=40:dwidth=width/10
640 x=0:y=0:dx=dwidth:dy=width
650MOVE 0,0
660ENDPROC
670:
680DEF PROCmove(dir,distance)
690?DATREG=(?DATREG AND 1)OR dir
700pulses=pd ratio*ABS(distance)
710FOR I=1 TO pulses:PROCpulse:NEXT I
720ENDPROC
730:
740DEF PROCturn(dir,angle)
750?DATREG=(?DATREG AND 1)OR dir
760pulses=pa ratio*angle
770FOR I=1 TO pulses:PROCpulse:NEXT I
780ENDPROC
790:
800DEF PROCpulse
810?DATREG=(?DATREG OR 8)
820?DATREG=(?DATREG AND 247)
830ENDPROC
840DEF PROCxplot
850FOR I=0 TO width STEP 4
860MOVExstart,y+I
870DRAWx,y+I
880NEXT I
890ENDPROC
900:
910DEF PROCyplot
920FOR I=0 TO width STEP 4
930MOVEx+I,ystart
940DRAWx+I,y
950NEXT I
960ENDPROC

```



Suite du listage Commodore 64

```

3010 TG=wx:SE=LF:XS=0
3020 DR=FW:DS=DX:GOSUB7000:REM DEPLACER
3030 X=X+DX
3040 IF(PEEK(DATREG)AND 192)=NB AND X<>TG THEN 3020
3050 GOSUB9000:REM TRACE X
3060 DX=-DX:Y=Y+DY:XS=X
3070 IF TG=0 THEN TG=wx:GOTO 3090
3080 TG=0
3090 IF Y<WY THEN WA=SE:GOSUB8000:REM NEXT STRIP
3100 IF SE=RT THEN SE=LF:GOTO 3120
3110 SE=RT
3120 IF Y<WY THEN 3020
3130 RETURN
3140 :
4000 REM **** BALAYAGE VERTICAL ****
4010 IF X=0 THEN SE=LF:DX=WD:GOTO 4030
4020 SE=RT:DX=-WD
4030 DY=-DY:TG=0:YS=Y
4040 DR=SE:AG=90:GOSUB7100:REM TOURNER
4050 DR=FW:DS=DY:GOSUB7000:REM DEPLACER
4060 Y=Y+DY
4070 IF(PEEK(DATREG)AND 192)=NB AND Y<>TG THEN 4050
4080 GOSUB9100:REM TRACE Y
4090 DY=-DY:X=X+DX:YS=Y
4100 IF TG=0 THEN TG=WY:GOTO 4120
4110 TG=0
4120 WA=SE:GOSUB8000:REM BANDE SUIVANTE
4130 IF SE=RT THEN SE=LF:GOTO 4150
4140 SE=RT
4150 IF X<WX AND X>0 THEN 4050:REM REPETER
4160 RETURN
4170 :
5000 REM **** ENTRER HAUTE RESOLUTION ****
5010 POKE 49408,1:POKE 49409,1
5020 POKE 49410,1:SYS HIRES:RETURN
5030 :
5100 REM **** QUITTER HAUTE RESOLUTION ****
5110 POKE 49408,0:POKE 49409,0
5120 POKE 49410,1:SYS HIRES:RETURN
5130 :
6000 REM **** ENTRER LIGNE ****
6010 MHI=INT(X1/256):MLO=X1-256*MHI
6020 NHI=INT(X2/256):NLO=X2-256*NHI
6030 POKE 49920,MLO:POKE 49921,MHI
6040 POKE 49922,NLO:POKE 49923,NHI
6050 POKE 49924,Y1:POKE 49925,Y2
6060 SYS LINESUB:RETURN
6070 :
7000 REM **** DEPLACER (DR,DS) ****
7010 POKE DATREG,(PEEK(DATREG)AND 1)OR DR
7020 PL=PD*DS
7030 FOR I=1 TO PL:GOSUB7200:NEXT I
7040 RETURN
7050 :
7100 REM **** TOURNER (DR,AG) ****
7110 POKE DATREG,(PEEK(DATREG)AND 1)OR DR
7120 PL=PA*DS
7130 FOR I=1 TO PL:GOSUB7200:NEXT I
7140 RETURN
7150 :
7200 REM **** IMPULSION ****
7210 POKE DATREG,PEEK(DATREG)OR 8
7220 POKE DATREG,PEEK(DATREG)AND 247
7230 RETURN
7240 :
8000 REM **** BANDE SUIVANTE ****
8010 DR=BW:DS=30:GOSUB7000:REM DEPLACER
8020 DR=WA:AG=90:GOSUB7100:REM TOURNER
8030 DR=FW:DS=WD:GOSUB7000:REM DEPLACER
8040 DR=WA:AG=90:GOSUB7100:REM TOURNER
8050 RETURN
9000 REM **** TRACE X ****
9010 FOR I=0 TO WD
9020 X1=XS/4:Y1=(Y+I)/4:X2=X/4:Y2=Y1
9030 GOSUB6000:REM ENTRER LIGNE
9035 NEXT I
9040 RETURN
9100 REM **** TRACE Y ****
9110 FOR I=0 TO WD
9120 X1=(X+I)/4:Y1=YS/4:X2=X1:Y2=Y/4
9130 GOSUB6000:REM ENTRER LIGNE
9135 NEXT I
9140 RETURN

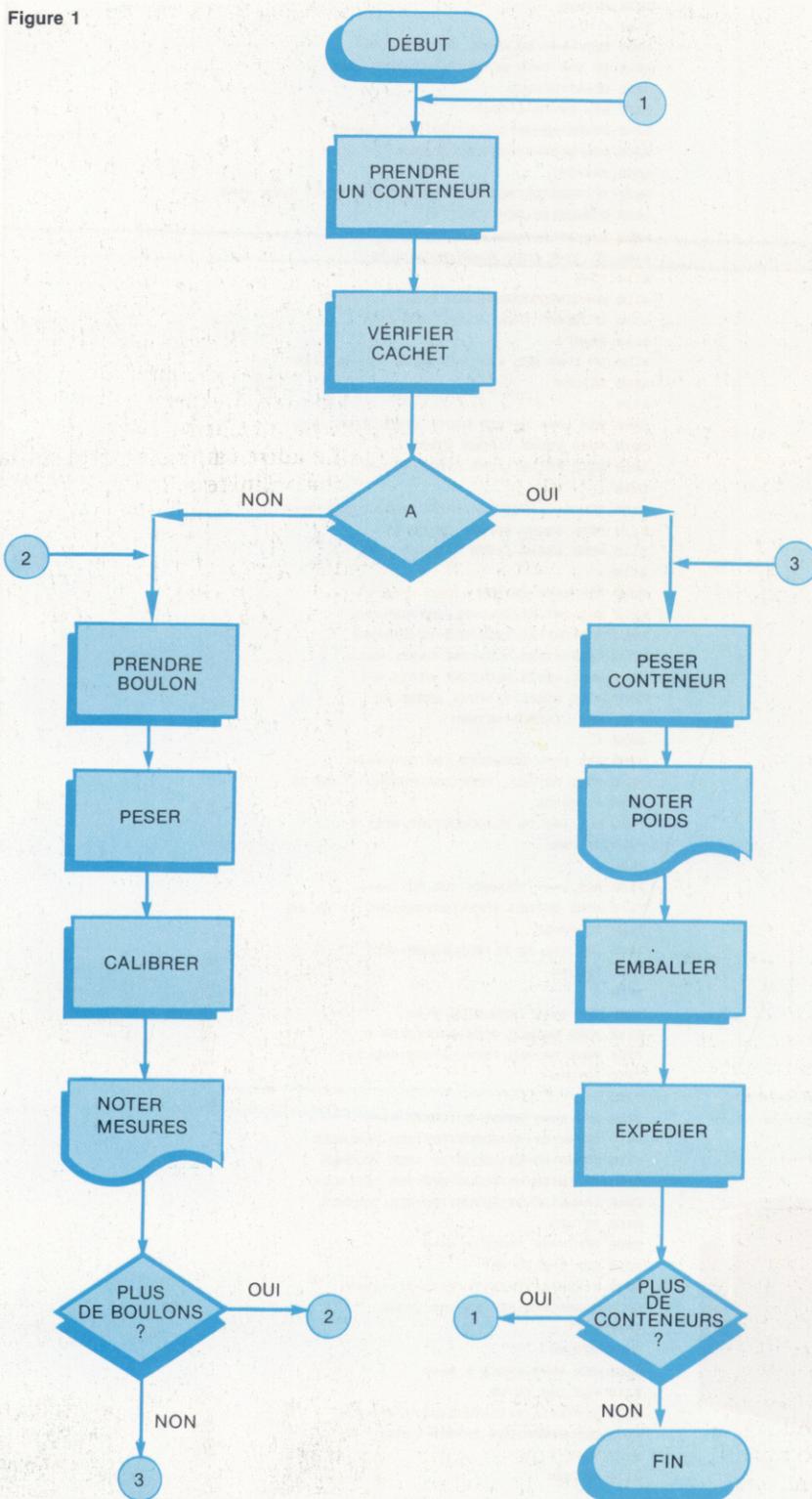
```



Connecteurs

Dans les organigrammes longs et complexes, il est nécessaire d'introduire un maximum de clarté. Nous allons parler de deux symboles qui s'avèrent indispensables.

Figure 1

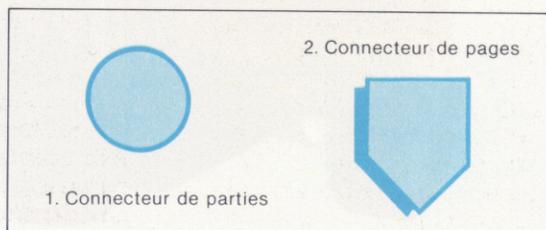


Ces symboles se nomment « connecteur de parties » et « connecteur de pages ». Pour illustrer le premier, prenons un exemple. Un opérateur, situé à la fin d'une chaîne de production, reçoit deux types de conteneurs — A et B — contenant un nombre indéterminé de boulons. Sa tâche se réduit à la chose suivante : il prend un conteneur et, si celui-ci appartient à la classe A, il doit le peser, noter son poids, l'emballer et l'expédier. En revanche, si le conteneur appartient au groupe B, il prendra un boulon, le pèsera, le calibrera et notera ses mesures. Il continuera ainsi jusqu'au dernier boulon, puis il pèsera le conteneur, notera son poids et l'emballera.

Observez, dans cet exemple, l'utilisation des symboles connecteurs de parties (fig. 1). Ils réalisent la même fonction que les lignes de flux, mais avec un avantage : plus l'organigramme est compliqué et plus le nombre de lignes augmente, ce qui peut provoquer un grand nombre d'erreurs ; à l'aide de ces petits cercles, non seulement on simplifie l'écriture, mais on accède également au point marqué par le connecteur d'entrée, dont l'indication écrite à l'intérieur (une lettre ou un numéro de préférence) coïncide avec la figure placée dans le connecteur de sortie. Le connecteur qui marque l'entrée de la séquence doit être unique ; cela signifie qu'il ne peut exister deux connecteurs d'entrée possédant une même indication. Mais il n'y a pas de limite aux connecteurs de sortie.

En comparant ce cas à celui des lignes de flux, on voit qu'il est possible d'utiliser autant de connecteurs que nécessaire dans un même organigramme, et ils peuvent s'alterner indistinctement avec les lignes de flux dans l'organigramme.

Le connecteur de pages (fig. 2), en revanche, est de caractère obligatoire, en particulier lorsqu'un même organigramme occupe plus d'une feuille. Arrivé à ce stade, on peut se rendre compte du grand avantage quant à la commodité et surtout à la clarté que représente l'utilisation d'un connecteur à la place d'une ligne de flux pour relier les points d'un organigramme comprenant différentes pages.

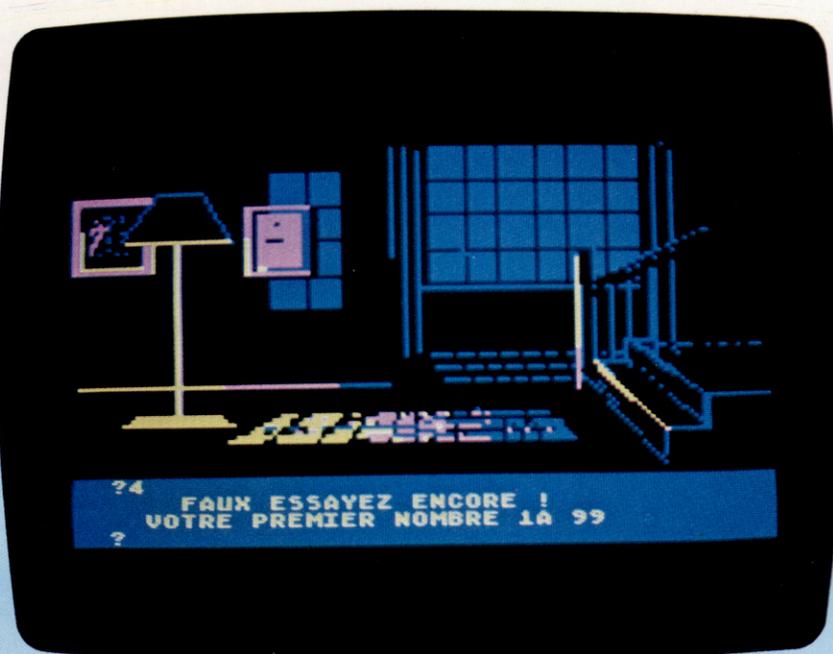




Cambriolage

Ce jeu, écrit par Terry Davies, utilise 24 K de mémoire de votre ordinateur Atari. Le but est de retirer le plus d'argent possible contenu dans le coffre-fort.

Pour ouvrir le coffre, il faut trouver trois bons numéros; le programme vous prévient lorsque vous êtes à plus ou moins cinq du numéro recherché. Vous avez droit à dix essais par numéro avant que l'alarme ne se déclenche. Lorsque vous trouvez le premier numéro, vous essayez le second, puis le troisième. Si vous trouvez les trois numéros (une probabilité de 1 sur 970 299), le coffre s'ouvre et vous pouvez dérober l'argent. Vous essayez ensuite dans une pièce différente. La chance tient une place importante dans ce jeu, mais la technique aussi. Après avoir expérimenté ce jeu, vous réfléchirez à deux fois avant de vous lancer dans un véritable cambriolage.



```

1 REM *****
2 REM ** CAMBRIOLAGE **
3 REM ** ECRIT PAR TERRY DAVIES **
4 REM *****
5 GRAPHICS 2:FLASH=0:P=10:DIM A$(5),D$(3
200),C(4),PT$(62):TT=0
10 POKE 708,P+10:POKE 709,145:POKE 752,8
:POKE 712,145:POKE 710,145
15 ? " TAPEZ start "
20 POSITION 4,4: ? #6; " CAMBRIOLAGE"
25 POSITION 4,6: ? #6; " CAMBRIOLAGE":160
SUB 6000
30 IF PEEK(53279)=6 THEN 1000
35 P=P+0,1:FLASH=FLASH+1:IF FLASH>50 THE
W POKE 709,P:IF FLASH=50 THEN POKE 709,1
45:FLASH=20:POKE 53279,0
40 IF FLASH=0 THEN POKE 709,145:IF FLASH
=0 THEN POKE 708,P:POKE 53279,0
50 IF P<200 THEN 30
1000 GRAPHICS 7:PX=4:LC=40:X=0:Y=0
1005 GOSUB 5100
1007 ? "UN INSTANT."
1010 GOSUB 30000
1020 FOR I=0 TO 4:POKE 708+I,C(I):NEXT I
1030 GOSUB 5199
1040 POKE 752,1:POKE 709,0: ? " VO
ICI LE COFFRE...CHUT!!!!!!":FOR W=1 TO 1
500:NEXT W: ? CHR$(125)
1043 ? " VOUS FEREZ MIEUX D'ELABORER": ?
" VOUS ETES SEUL...TAPEZ START":POKE 532
79,0
1045 IF PEEK(53279)>6 THEN 1045
1050 POKE 709,202: ? CHR$(125):"VOUS UOYE
Z MIEUX MAINTENANT": ? "DANS LA PIECE":FO
R W=1 TO 2000:NEXT W
1055 ? CHR$(125):GOTO 2000
1060 GOSUB 6000
1070 ? CHR$(125):"TAPEZ START POUR UN AU
TRE CAMBRIOLAGE"
1071 POKE 710,30:FOR W=1 TO 200:NEXT W:P
OKE 710,138:FOR W=1 TO 99:NEXT W:POKE 71
0,148:FOR W=1 TO 250:POKE 710,138
1072 FOR W=1 TO 99:NEXT W:POKE 710,148:FO
R W=1 TO 250:NEXT W:POKE 710,138:FOR W=
1 TO 100:NEXT W:POKE 710,148
1073 FOR W=1 TO 250:NEXT W:POKE 710,138:
FOR W=1 TO 100:NEXT W:POKE 710,148
1074 POKE 53279,0:IF PEEK(53279)>6 THEN
1072
1075 GOTO 1020
2000 A=INT(RND(1)*81)+10:B=INT(RND(1)*81
)+10:C=INT(RND(1)*81)+10:J=0
2003 J=J+1: ? " VOTRE PREMIER NOMBRE 1A
99 ":INPUT N:IF N>99 OR N<0 THEN 2003
2005 IF J=11 THEN GOSUB 3950:GOTO 1060
2006 IF N<A THEN 2006
2007 GOSUB 3850:GOTO 2499
2008 IF N>A-5 AND N<A+5 THEN GOSUB 3900:
GOTO 2003
2010 IF N>A+5 OR N<A-5 THEN GOSUB 4000:G
OTO 2003
2011 GOTO 2003
2499 J=0
2500 J=J+1: ? " VOTRE DEUXIEME NOMBRE":INP
UT N:IF N>99 OR N<0 THEN 2500
2510 IF J=11 THEN GOSUB 3950:GOTO 1060
2520 IF N<B THEN 2520
2525 GOSUB 3850:GOTO 2999
2530 IF N>B-5 AND N<B+5 THEN GOSUB 3900:
GOTO 2500

```

```

2535 IF N>B+5 OR N<B-5 THEN GOSUB 4000:G
OTO 2500
2540 GOTO 2500
2999 J=0
3000 ? " VOTRE TROISIEME NOMBRE":INPUT
N:IF N>99 OR N<0 THEN 3000
3005 J=J+1
3010 IF J=11 THEN GOSUB 3950:GOTO 1060
3020 IF N<C THEN 3020
3025 GOSUB 3850: ? "LE COFFRE EST OUVERT"
:J=0:COLOR 8:PLOT 41,30:DRABTO 39,16
3026 J=J+1:IF J>9 THEN 3028
3027 DRABTO 41+J,16:DRABTO 41+J,30:GOTO
3026
3028 FR=INT(RND(0)*200)+100: ? CHR$(125):
"BIEN JOUE , IL Y AVAIT " :FR: " FRAN S DA
NS": ? "CE COFFRE!!!!!!"
3029 TT=TT+FR: ? "ESSEYER CELUI CI.....":
FOR W=1 TO 1000:NEXT W:GOTO 1020
3030 IF N<C-5 AND N<C+5 THEN GOSUB 3900:
GOTO 3000
3035 IF N>C+5 OR N<C-5 THEN GOSUB 4000:G
OTO 3000
3040 GOTO 3000
3050 FOR T=1 TO 100:SOUND 0,T,10,8:NEXT
T: ? "C'EST BON":SOUND 0,0,0,0:RETURN
3900 FOR S=20 TO 15 STEP -4:NEXT S:SOUND
0,0,14:FOR W=1 TO 3:NEXT W:SOUND 0,0
,0,0:RETURN
3950 ? "MALHEUR !!! L'ALARME S'EST DECLENC
HEE": ? "LES POLICIERS ARRIVENT " :R=0
3960 ? "VOUS RETIREZ ":TT: " FRANCS DES C
OFFRES": ? TT=0
3998 FOR N=20 TO 15 STEP -4:NEXT N:SOUND
0,0,14:FOR W=1 TO 33:NEXT W:SOUND 0
,0,0,0:R=R+1:IF R=20 THEN RETURN
3999 GOTO 3998
4000 ? " FAUX ESSEYER ENCORE !":RETUR
N
5100 DL=PEEK(560)+PEEK(561)*256:MEM=PEEK
(106)*256:DAT=PEEK(DL+4)+256*PEEK(DL+5)
5108 RESTORE 5200:FOR P=1 TO 62:READ A:P
T(P,)=CHR$(A):NEXT P
5110 ST=DAT+V*LC+INT(X/PX):RETURN
5199 GOSUB 5110:A=USR(ADR(PT*),ST,ADR(D#
),BYT,LIN,LC):RETURN
5200 DATA 104,104,133,204,104,133,203,10
4,133,206,104,133,205,104,104,133,207,10
4,104
5210 DATA 170,104,104,133,208,164,207,13
6,177,205
5220 DATA 145,203,136,192,255,208,247,24
,165
5230 DATA 203,101,208,133,203,144,2,230,
204,24,165,205,101,207,133,205,144,2,230

```

```

5240 DATA 206,202,208,219,96
6000 FOR J=1 TO 8:SOUND 0,47,10,8:FOR L=
1 TO 100:NEXT L: ? SOUND 0,64,10,8:FOR L=1
TO 100:NEXT L:NEXT J
6001 SOUND 0,0,0,0:RETURN
20000 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0
20001 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
20002 DATA 0,0,0,12,0,0,0,0,0,0,0,0,0,0,0
,0
20003 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,195
,0
20004 DATA 63,255,207,255,63,255,207,255
,63,255,207,255,192,192,0
20005 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
20006 DATA 0,0,0,0,0,0,0,0,195,12,63,255
,207,255,63
20007 DATA 255,207,255,63,255,207,255,19
2,192,192,0,0,0,0,0
20008 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
20009 DATA 0,0,0,195,12,63,255,207,255,6
3,255,207,255,63,255
20010 DATA 207,255,192,192,192,0,0,0,0,0
,0,0,0,0,0
20011 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,195
,12
20012 DATA 63,255,207,255,63,255,207,255
,63,255,207,255,192,192,192
20013 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
20014 DATA 0,0,0,0,0,0,0,0,195,12,63,255
,207,255,63
20015 DATA 255,207,255,63,255,207,255,19
2,192,192,0,0,0,0,0
20016 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
20017 DATA 0,0,0,195,12,63,255,207,255,6
3,255,207,255,63,255
20018 DATA 207,255,192,192,192,0,0,0,0,0
,0,0,0,0,0
20019 DATA 0,0,0,0,0,0,63,255,207,255,19
2,0,0,195,12
20020 DATA 63,255,207,255,63,255,207,255
,63,255,207,255,192,192
20021 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
20022 DATA 0,63,255,207,255,192,0,0,195
,12,63,255,207,255,63
20023 DATA 255,207,255,63,255,207,255,19
2,192,192,0,0,0,0,0
20024 DATA 0,0,0,0,0,0,0,0,0,0,0,63,255
,207,255
20025 DATA 192,0,0,195,12,0,0,0,0,0,0,0,0
,0,0,0
20026 DATA 0,0,0,192,192,0,0,0,0,0,0,0,0
,0,0
20027 DATA 0,0,0,0,0,0,63,255,207,255,19
2,0,0,195,12

```



```

20028 DATA 63,255,207,255,63,255,207,255
,63,255,207,255,192,192,192
20029 DATA 0,0,0,0,0,0,0,0,3,255,255,2
40,0,0
20030 DATA 0,63,255,207,255,192,0,0,195,
12,63,255,207,255,63
20031 DATA 255,207,255,63,255,207,255,19
2,192,192,0,0,0,0,0
20032 DATA 85,85,85,85,92,0,0,12,0,0,0,6
3,255,207,255
20033 DATA 192,0,0,195,12,63,255,207,255
,63,255,207,255,192,192,192,0,0,0,0,0
,85,85,85,85,92
20034 DATA 207,255,192,192,192,0,0,0,0,0
,85,85,85,85,92
20035 DATA 0,0,12,0,0,0,63,255,207,255,1
92,0,0,195,12
20036 DATA 63,255,207,255,63,255,207,255
,63,255,207,255,192,192,192
20037 DATA 0,0,0,0,0,0,0,0,0,48,0,0,3,0
,0
20038 DATA 0,63,255,207,255,192,0,0,195,
12,63,255,207,255,63
20039 DATA 255,207,255,63,255,207,255,19
2,192,192,0,0,0,0,0
20040 DATA 80,0,0,0,192,0,0,3,0,1,85,85,
85,80,0
20041 DATA 0,0,0,195,12,63,255,207,255,6
3,255,207,255,63,255
20042 DATA 207,255,192,192,192,0,0,0,0,0
,0,3,195,3,0
20043 DATA 0,0,0,192,1,64,63,255,240,255
,192,0,0,195,12
20044 DATA 63,255,207,255,63,255,207,255
,63,255,207,255,192,192,192
20045 DATA 0,0,0,0,0,0,16,51,3,0,0,0,0,
192,1
20046 DATA 69,85,85,240,255,192,0,0,195,
12,0,0,0,0,0
20047 DATA 0,0,0,0,0,0,0,192,192,0,0,0,
0,0
20048 DATA 80,4,192,12,0,0,0,48,1,69,8
5,85,240,255
20049 DATA 192,0,0,195,12,63,255,207,255
,63,255,207,255,63,255
20050 DATA 207,255,192,192,192,0,0,0,0,0
,80,85,192,48,0
20051 DATA 0,0,0,12,1,69,81,85,240,255,1
92,0,0,195,12
20052 DATA 63,255,207,255,63,255,207,255
,63,255,207,255,192,192,192
20053 DATA 0,0,0,0,0,88,68,3,192,0,0,0,0
,12,1
20054 DATA 69,85,85,240,255,192,0,0,195,
12,63,255,207,255,63
20055 DATA 255,207,255,63,255,207,255,19
2,192,192,0,0,0,0,0
20056 DATA 88,20,192,192,0,0,0,0,3,1,69,
85,85,240,255
20057 DATA 192,0,0,195,12,63,255,207,255
,63,255,207,255,63,255
20058 DATA 207,255,192,192,192,3,192,0,0
,0,88,16,195,0,0
20059 DATA 0,0,0,3,1,69,85,85,240,255,19
2,0,0,195,12
20060 DATA 63,255,207,255,63,255,207,255
,63,255,207,255,192,192,192
20061 DATA 48,0,0,0,0,88,80,0,170,170,17
0,170,170,194
20062 DATA 69,85,240,0,0,0,0,195,12,63
,255,207,255,63
20063 DATA 255,207,255,63,255,207,255,19
2,192,195,0,0,0,0,0
20064 DATA 88,64,192,0,88,2,64,0,0,2,69,
85,85,240,255
20065 DATA 192,0,0,195,12,63,255,207,255
,63,255,207,255,63,255
20066 DATA 207,255,192,192,240,0,0,0,0,0
,88,76,12,192,88
20067 DATA 2,64,0,0,2,69,85,85,240,255,1
92,0,0,195,12
20068 DATA 63,255,207,255,63,255,207,255
,63,255,207,255,192,192,192
20069 DATA 0,0,0,0,0,88,0,0,88,2,64,0,
0,2
20070 DATA 69,85,85,240,255,192,0,0,195,
12,0,0,192,0,0
20071 DATA 0,0,0,0,12,0,0,0,240,192,0,0,
0,0,0
20072 DATA 88,0,0,0,88,2,64,0,0,2,69,85,
85,240,255
20073 DATA 192,0,0,195,12,63,255,255,255
,63,255,207,255,60,15
20074 DATA 207,255,207,0,0,0,0,0,0,88,0
,0,0,88
20075 DATA 2,64,0,0,2,69,85,85,240,255,1
92,0,0,3,12
20076 DATA 63,255,207,255,63,255,207,255
,56,15,207,255,240,192,0
20077 DATA 0,0,0,0,0,0,0,170,170,170,88,2
,64,0,0,2
20078 DATA 69,85,85,240,255,192,0,0,3,12
,63,255,207,255,63
20079 DATA 255,207,255,56,15,207,255,0,0
,192,0,0,0,0,0
20080 DATA 85,85,85,85,88,2,64,0,0,2,74,
255,255,240,255
20081 DATA 192,0,0,3,12,63,255,207,255,6
3,255,207,255,56,15
20082 DATA 207,240,192,0,192,0,0,0,0,0,0
,0,0,0,0
20083 DATA 2,64,0,0,2,85,85,85,80,255,19
2,0,0,3,12
20084 DATA 63,255,207,255,63,255,207,255
,56,15,207,12,3,192,192
20085 DATA 0,0,0,0,0,0,0,0,0,2,64,0,0,0
,0
20086 DATA 0,0,0,0,0,0,0,3,12,63,255,2
07,255,63
20087 DATA 255,207,255,56,15,192,0,3,192
,0,0,0,0,0

```

```

,192,0,0,0,0,0
20088 DATA 0,0,0,0,0,2,64,0,0,0,0,63,255
,207,255
20089 DATA 192,0,0,3,12,63,255,207,255,6
3,255,207,255,56,12
20090 DATA 0,0,195,192,192,0,0,0,0,0,0,0
,0,0,0
20091 DATA 2,64,0,0,0,0,63,255,207,255,1
92,0,0,3,12
20092 DATA 0,0,0,0,0,0,0,0,0,12,3,19
2,192
20093 DATA 0,0,0,0,0,0,0,0,0,2,64,0,0,0
,0
20094 DATA 0,63,255,207,255,192,0,0,3,15
,255,255,255,255,255
20095 DATA 255,255,255,251,12,60,255,195
,192,192,0,0,0,0,0
20096 DATA 0,0,0,0,0,2,64,0,0,0,0,63,255
,207,255
20097 DATA 192,0,0,3,12,0,0,0,0,0,0,0,0,
11,0
20098 DATA 12,12,3,192,192,0,0,0,0,0,0,0
,0,0,0
20099 DATA 2,64,0,0,0,0,63,255,207,255,1
92,0,0,3,12
20100 DATA 0,0,0,0,0,0,0,11,0,12,12,3,
192,192
20101 DATA 0,0,0,0,0,0,0,0,0,2,64,0,0,0
,0
20102 DATA 0,63,255,207,255,192,0,0,3,12
,0,0,0,0,0
20103 DATA 0,0,0,11,12,12,12,3,192,192,0
,0,0,0,0
20104 DATA 0,0,0,0,0,2,64,0,0,0,0,63,255
,207,255
20105 DATA 192,0,0,3,12,0,0,0,0,0,0,0,0,
11,12
20106 DATA 12,12,3,192,192,0,0,0,0,0,0,0
,0,0,0
20107 DATA 2,64,0,0,0,0,0,0,0,0,0,0,0,3,
12
20108 DATA 0,0,0,0,0,0,0,11,12,12,12,3,
192,192
20109 DATA 0,0,0,0,0,0,0,0,0,2,64,0,0,0
,0
20110 DATA 0,0,0,0,0,0,0,3,12,0,0,0,0,0
,0
20111 DATA 0,0,0,11,12,12,12,3,192,192,0
,0,0,0,0
20112 DATA 0,0,0,0,0,2,64,0,0,0,0,0,0,0,0
,0
20113 DATA 0,0,0,3,12,0,0,0,0,0,0,0,11
,12
20114 DATA 12,12,3,192,192,0,0,0,0,0,0,0
,0,0,0
20115 DATA 2,64,0,0,0,0,0,0,0,0,0,0,0,3,
12
20116 DATA 0,0,0,0,0,0,0,11,12,12,12,3,
192,192
20117 DATA 0,0,0,0,0,0,0,0,0,2,64,0,0,0
,0
20118 DATA 0,0,0,0,0,0,0,3,12,0,0,0,0,0
,0
20119 DATA 0,0,0,11,12,12,12,3,192,192,0
,0,0,0,0
20120 DATA 0,0,0,0,0,2,64,0,0,0,0,0,0,0,0
,0
20121 DATA 0,0,0,3,12,0,0,0,0,0,0,0,0,11
,12
20122 DATA 12,12,3,192,192,0,0,0,0,0,0,0
,0,0,0,0
20123 DATA 2,64,0,0,0,0,0,0,0,0,0,0,0,3,
12
20124 DATA 0,0,0,0,0,0,0,11,12,12,12,3,
192,192
20125 DATA 0,0,0,0,0,0,0,0,0,2,64,0,0,0
,0
20126 DATA 0,0,0,0,0,0,0,3,12,0,0,0,0,0
,0
20127 DATA 0,0,0,11,12,12,12,3,192,192,0
,0,0,0,0
20128 DATA 0,0,0,0,0,2,64,0,0,0,0,0,0,0,0
,0
20129 DATA 0,0,0,3,12,0,0,0,0,0,0,0,0,11
,12
20130 DATA 12,63,255,192,192,0,0,0,0,0,0
,0,0,0,0
20131 DATA 2,64,0,0,0,0,0,0,0,0,0,0,0,3,
12
20132 DATA 0,0,0,0,0,0,0,11,12,12,51,1
92,252,63
20133 DATA 192,195,195,15,192,0,0,0,0,0
,2,64,0,0,0
20134 DATA 0,0,0,0,0,0,0,0,3,12,0,0,0,0,0
,0
20135 DATA 0,0,0,7,12,12,48,252,192,192,
0,0,0,0,0
20136 DATA 0,0,0,0,0,2,64,0,0,0,0,0,0,0,0
,0
20137 DATA 0,0,0,3,0,63,243,252,255,63,2
07,243,240,247,12
20138 DATA 207,240,15,192,48,0,0,0,0,0,0,0
,0,0,0,0
20139 DATA 2,64,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20140 DATA 0,0,0,0,0,0,0,0,7,12,12,48,3,
192,0
20141 DATA 0,0,0,0,0,0,0,0,0,0,2,64,0,0,0
,0
20142 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20143 DATA 0,0,0,7,11,255,252,0,240,0,0,0
,0,0,0,0
20144 DATA 0,0,0,0,0,2,64,0,0,0,0,0,0,0,0
,0
20145 DATA 0,0,0,0,0,15,255,63,207,207,2
43,252,255,55,2
20146 DATA 192,12,0,63,0,0,0,0,0,0,0,0,0
,0,0
20147 DATA 2,64,0,0,0,0,0,0,0,0,0,0,0,0,0
,0

```

```

0
20148 DATA 0,0,0,0,0,0,0,0,4,2,176,0,48,
15,192
20149 DATA 0,0,0,0,0,0,0,0,0,0,2,64,0,0,0
,0
20150 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0
20151 DATA 0,0,0,7,0,172,0,0,0,240,0,0,0,0
,0,0
20152 DATA 0,0,0,0,0,2,64,0,0,0,0,0,0,0,0,0
,0
20153 DATA 0,0,0,0,0,0,255,243,252,252,2
52,255,63,195,192
20154 DATA 40,0,3,0,12,0,0,0,0,0,42,170,
170,170,170
20155 DATA 170,170,170,165,85,85,85,85,8
5,85,255,255,255,0,0
20156 DATA 0,0,0,0,0,0,0,0,4,51,10,0,0,0,0
,3
20157 DATA 0,0,0,0,0,0,0,0,0,0,2,64,0,0,0
,0
20158 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20159 DATA 0,0,0,4,12,194,128,0,48,0,240
,0,0,0,0
20160 DATA 0,0,0,0,0,2,64,0,0,0,0,0,0,0,0,0
,0
20161 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,3
20162 DATA 0,160,0,0,0,12,0,0,0,0,0,0,0,0,0
,0,0
20163 DATA 2,64,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20164 DATA 0,0,0,0,0,0,0,0,0,48,40,0,3
,0
20165 DATA 3,255,255,255,252,0,0,0,0,0,2
,64,0,0,0
20166 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20167 DATA 0,0,0,0,0,12,10,0,0,192,3,192
,0,0,0
20168 DATA 0,0,0,0,0,2,64,0,0,0,0,0,0,0,0,0
,0
20169 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20170 DATA 51,0,128,0,48,3,192,0,0,0,0,0,0
,0,0,0
20171 DATA 2,64,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20172 DATA 0,0,3,0,0,0,0,0,0,12,0,32,0
,15
20173 DATA 3,192,0,0,0,0,0,0,10,170,170,170
,170,160,0
20174 DATA 0,2,168,10,170,130,128,21,72,
70,5,143,197,223,255
20175 DATA 243,255,0,0,0,3,0,8,0,0,195,1
92,0,0,0
20176 DATA 0,0,0,42,170,170,170,170,168,
0,0,42,128,154,106
20177 DATA 0,5,85,69,65,127,0,0,0,0,0,0,0,0
,0,0,0
20178 DATA 0,192,2,0,0,51,192,0,0,0,0,0,0,0
,2,170,170
20179 DATA 170,170,170,170,128,2,170,130
,170,168,6,138,21,21,71
20180 DATA 213,255,255,223,207,240,255,2
52,0,0,0,48,0,128,0
20181 DATA 15,192,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0
20182 DATA 43,248,60,42,195,254,63,255,2
55,245,89,81,85,255,204
20183 DATA 243,63,252,0,0,0,12,0,32,0,3,
192,0,0,0
20184 DATA 0,0,0,0,0,0,0,0,0,0,168,32,5,
22,2
20185 DATA 168,42,170,21,79,245,0,0,0,4,
4,1,63,0,0
20186 DATA 0,3,0,11,255,255,192,0,0,0,0,0
,0,0,0,0
20187 DATA 0,0,0,0,0,42,170,138,170,170,42
,168,165,85,255,245
20188 DATA 127,241,63,255,207,207,207,25
5,192,0,0,0,0,192,12,0
20189 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,2,1
70
20190 DATA 168,10,170,168,170,160,1,84,1
65,70,5,81,85,95,207
20191 DATA 207,207,255,240,0,0,0,0,12,0,0
,0,0,0,0
20192 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20193 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20194 DATA 0,0,48,12,0,0,0,0,0,0,0,0,0,0,0,0
,0
20195 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20196 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,12,12
,0,0
20197 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20198 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20199 DATA 0,0,0,0,0,0,0,3,12,0,0,0,0,0,0,0
,0
20200 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20201 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20202 DATA 0,0,0,204,0,0,0,0,0,0,0,0,0,0,0,0
,0
20203 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20204 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,48,
0,0
20205 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20206 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20207 DATA 0,0,0,0,0,0,0,0,0,12,0,0,0,0,0,0
,0
20208 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20209 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0
20210 DATA 0,0,0,0,0,12,0,0,0,0,0,0,0,0,0,0
,0
30000 BYT=40:LIN=79:GR=7: C(0)=6:C(1)=0:C
(2)=148:C(3)=70:C(4)=0
30010 RESTORE 20000:FOR P=1 TO 3143
30020 READ A:D*(P,P)=CHR$(A)
30030 NEXT P
30040 RETURN

```