

LE TELESTRAT PLUS QU'UN NOUVEL ORIC

Fabrice BROCHE

Annoncé depuis longtemps, le TELESTRAT arrive enfin dans les foyers: suivons ensemble quelques Instants de vie de l'homo oricus qui pianote sur son TELESTRAT.

Sa première action est d'afficher et de mettre à l'heure L'horloge (TIME 1,34:CLOCKSET # 8880). En effet, le temps passe si vite sur le TELESTRAT...

Il charge ensuite le programme qu'il est en train d'élaborer, une gestion de base de données. La rapidité de l'Hyper-basic lui permet en effet de réaliser ce logiciel complexe en Basic. Il tape SGBD pour charger son programme, songeant un instant qu'il aurait dû taper 'LOAD "SGBD.BAS" sur d'autres ordinateurs, et s'étonnant que des opérations aussi courantes nécessitent autant de caractères...

Ne se rappelant plus trop où il en était, il lance une impression de son programme (LLIST). Grâce à son buffer imprimante intégré, le TELESTRAT lui rend tout de suite la main, et il continue l'édition de son programme. C'est, il est vrai, facile, grâce à cet analyseur de syntaxe qui va jusqu'à vous dire où placer la virgule. Et puis, cet éditeur de pleine page, quel régal !

Mais le téléphone retentit. Un collègue graphiste lui téléphone et veut absolument lui montrer sa dernière création sur TELESTRAT. HIRES:MLOAD, il est

prêt. Son ami connecte son TELESTRAT, et l'homo oricus appuie sur CONNEXION/FIN. La transmission de la page débute. Il jette un oeil amusé sur l'imprimante qui continue à tirer son listing. La transmission est finie. Son ami lui signale à tout hasard le dernier serveur à la mode sur le 3615, lui souhaite bonne nuit. Il sauve l'écran sur la disquette (ESAVE -ECRAN-) et décide d'en faire immédiatement une copie graphique sur l'imprimante, qui a, entre-temps, fini le listing. (HCOPY). Il revient en mode texte pour continuer son programme. Déjà 2 heures, lui signale l'horloge.

Se ravisant, il va consulter le serveur conseillé par son ami. APLIC1 : il lance l'émulation Vidéotex. Il se connecte, consulte et décide de recopier quelques jolis dessins ; il lui suffit de taper E et le nom de la page, et le tour est joué. Sa collection s'enrichit.

Il revient ensuite au Basic. Son programme est toujours là. Sa base de données commence à

tourner, il est vrai que l'accès direct façon TELESTRAT est d'une facilité d'emploi incroyable. Son programme devenant long, il bénit l'utilisation des labels pour ses routines, qui documentent son programme en même temps qu'elle facilite son écriture

Dans sa tête surgit alors une idée un peu folle : pourquoi ne pas rendre sa base de données accessible par Minitel ?

Il réfléchit deux secondes et se lance. APLIC3: il saute dans l'édition d'arborescence et écrit un mini-serveur pour tester son application : une ou deux pages de présentation (il appuie 2 fois sur P, et c'est tout), suivi d'un sommaire (il appuie sur S). Ensuite, une page d'accès Basic (il presse H). Pour l'instant, ça suffit, on signolera après. Il teste immédiatement le fonctionnement correct de son arborescence, même sans avoir dessiné les pages (test sans accès disque). Il se dit qu'une petite boîte aux lettres ne ferait pas de mal, c'est si facile. Et puis, ça peut fidéliser les correspondants... Il retourne dans l'arborescence, se déplace avec la souris sur le SOMMAIRE et appuie sur B. La boîte aux lettres est créée. Bien sûr, quand il aura le temps, il l'écrira en Basic avec des anima

tions sympa.

Il retourne sous Basic (la touche ESC). Tiens ? Son programme est toujours là. Il le sauve (SAVEU "SGBD"). Pratique, cette sauvegarde qui évite toujours les deux dernières versions du programme. Pas d'accidents à craindre...

Pendant qu'il réfléchit à sa base de données, il décide de dessiner la page de présentation de son serveur. APLIC 2 : il est dans l'éditeur de page. Il tape G pour passer en mode graphique. C'est un réel plaisir que de dessiner avec la souris en mode graphique. Il a sommeil. Bon, un dernier titre en double taille et il coupe la machine.

En s'endormant, il songe à ce que lui aurait coûté une machine avec un Basic compilé, un buffer imprimante, une console de composition Vidéotex, un logiciel de serveur, un modem... Vive le TELESTRAT.

LE CONCEPT TELESTRAT

Le TELESTRAT apporte plusieurs mini-révolutions au sein de la micro-informatique.

- L'existence d'un logiciel télématique intégré, utilisant le MINITEL distribué gratuitement. Ce logiciel est d'une extraordinaire facilité d'emploi par rapport à tout ce qui s'est fait dans ce domaine. Notamment, quel logiciel permet de dessiner l'arborescence de son serveur directement à l'écran, ou de dessiner des pages graphiques pavé par pavé, sans jongler avec les barbares jeux de caractères Vidéotex ?

Aucune connaissance préliminaire de la télématique n'est nécessaire...

- Un basic compilé, le plus complet du marché. Outre sa rapidité, inédite sur un micro, il peut se targuer d'être le plus interactif des Basics, allant même jusqu'à supprimer le classique "Syntax Error".

- Un grand nombre d'extensions, dont la très rare MIDI, sont intégrées. La communication n'est pas un vain mot !

- Enfin, une facilité d'emploi hors du commun, liée surtout à une très grande rapidité (accès dis

que, imprimante bufférisée, clavier bufférisé, le Basic lui-même), à l'éditeur pleine page qui autorise le défilement dans tout le texte Basic. Et, en général, tout a été conçu pour que l'utilisateur n'ait à se préoccuper que de la résolution de son problème, et que les tâches "basses", du type problèmes syntaxiques, écriture disques, édition... prennent le moins de temps possible.

Un autre aspect de la conception est à signaler, même s'il est moins apparent : c'est l'aspect modulaire du logiciel.

En effet, le TELESTRAT est fourni avec pas moins de 40 Ko de ROM, plus 11 Ko de DOS, sans compter une foule d'utilitaires sur la disquette système.

ORGANISATION DU LOGICIEL

Le principal souci lors de la conception a été de rendre les évolutions du logiciel faciles, tout en assurant la compatibilité avec les versions précédentes. Il était aussi utile de grouper toutes les routines de base en un seul logiciel, toujours présent, quelle que soit la configuration de la machine, ou les logiciels et langages présents.

Donc, le logiciel s'articule autour du TELEMON, sorte de système d'exploitation du TELESTRAT. Lorsqu'il s'agit de gérer les accès disque, c'est le STRATSED.

Ainsi, la cartouche TELEMATIQUE n'a besoin que du TELEMON, et éventuellement du STRATSED.

RESUME DES FONCTIONS DU TELEMON

Le TELEMON assure en premier lieu l'initialisation de la machine. Il termine la configuration, c'est-à-dire la présence des lecteurs, de l'imprimante, de la capacité RAM disponible, de savoir s'il faut se servir du Minitel comme terminal de visualisation. Ensuite, il initialise les entrées / sorties, puis lance enfin le programme d'application ou le langage selon la configuration.

Mais aussi, et surtout, le TELEMON est un réservoir de routines. Ainsi, les langages ou applica

tions n'ont pas à se préoccuper de tous les problèmes de gestion du clavier, affichage, etc. Tout ce qu'il faut, et même plus, existe dans le TELEMON.

Ainsi, on trouve des routines classiques, telles que conversions binaire-décimal-hexa-flottant, déplacement de blocs mémoire, drivers pour toutes les entrées/sorties (4 fenêtres d'écran, clavier, souris, joystick, Minitel, RS 232, imprimante, émulation Vidéotex).

Il y a aussi toutes les routines de calcul en flottant, tout ce qui concerne la haute résolution, la possibilité d'échange via le Minitel, la détection d'appel.

Comble de raffinement, l'éditeur pleine page est aussi une routine du TELEMON, ainsi que la gestion des menus déroulants que l'on trouve dans la télématique.

Ainsi, les langages n'ont à se préoccuper que de l'analyse de syntaxe et du passage de paramètres au TELEMON.

Cette construction a pour avantage sa simplicité d'usage pour les développeurs, amateurs ou professionnels, mais aussi sa compacité : le logiciel TELEMATIQUE du TELESTRAT occupe seulement 8 Ko. Etonnant, non ?

LE STRATSED

Le STRATSED assure la même fonction que le TELEMON, à savoir être un réservoir de routines. Mais, on s'en doute, il gère uniquement les disquettes. Il contient toutes les primitives pour sauver, lire, copier, changer de nom, protéger, déprotéger un programme. Ainsi que les routines d'initialisation, de copie de disquette, etc.

Bien entendu, la gestion de fichier est accessible, les accès direct et séquentiel étant gérés par le STRATSED.

Avant de passer à la description de l'Hyper-Basic, puis de la TELEMATIQUE, voici un bref résumé des caractéristiques du Hard du TELESTRAT.

QU'EST CE QU'IL Y A A L'INTERIEUR

Le TELESTRAT est construit

autour d'un 6502. La gestion des périphériques est assurée par deux VIA 6522, une ACIA 6551, un FDC 1793 (pour les lecteurs de disques). La capacité RAM intégrée est de 64 Ko..

Le logiciel est implanté sur deux ports cartouches. Chaque port peut recevoir 64 Ko de mémoire. Ce peut être de la ROM, de la RAM ou n'importe quoi d'autre (extensions diverses ...). La ROM est partitionnée en banques de 16 Ko. A la mise sous tension du système, la cartouche contenue dans le port de gauche est lancée. On aura donc intérêt à laisser un TELEMOM, qui assurera l'initialisation. Après, toutes les fantaisies sont permises : 96 Ko de ROM ou de RAM supplémentaires, ou un mélange des deux... Si les cartouches ROM ou EPROM ont un format reconnu par le TELESTRAT, elles seront automatiquement lancées.

L'espace mémoire du TELESTRAT est donc séparé entre la zone # 0000- # BFFF, la RAM, et la zone #COOO-#FFFF qui est paginée 8 fois : la banque 0 contient de la RAM, et le STRATSED. La banque 7 contient le TELEMOM, la banque 6 l'HyperBasic. La cartouche TELEMATIC occupe la banque 3.

Que d'extensions !

il suffit de faire le tour du TELESTRAT pour remarquer le nombre incroyable de connecteurs.

- Un bus d'extension complet, ou on retrouve tous les signaux du bus 6502. Ce bus est de plus directement compatible avec celui de l'ATMOS.

- Un connecteur pour lecteur de disquettes. Le contrôleur de disques est intégré au TELESTRAT et permet de gérer 4 lecteurs de disquettes.

Ces lecteurs doivent être au format Shugart, et donc n'importe quel lecteur 3", 3.5", 5" 1/4 convient, qu'il soit double ou simple face. Le STRATSED gère une capacité formatée de 960 Ko. Le lecteur livré avec le TELESTRAT est un 3" double densité, double face, assurant une capacité formatée de 360 à 400 Ko selon formatage.

- Un connecteur imprimante

parallèle, type Centronics, compatible avec la plupart des imprimantes du marché. La gestion de l'imprimante est bufférisée par le TELESTRAT (2 Ko), ce qui assure une grande rapidité dans les impressions.

- Un connecteur RS 232, permettant d'accéder à une foule de périphériques, type modems, tables traçantes, ainsi que de dialoguer directement avec la plupart des ordinateurs. La liaison peut se faire en FULL DUPLEX jusqu'à 19200 bauds, et bien entendu l'Hyper-Basic permet de gérer facilement la sortie série.

- Un connecteur permettant de se relier à un Minitel pour les applications TELEMATIQUES. De plus, pour assurer la détection d'appel, un joncteur téléphonique PTT, peut se brancher sur le réseau PTT.

- 2 prises joystick, la prise de droite étant particulièrement prévue pour gérer une souris. D'ailleurs, une souris (port de droite) et un joystick (port de gauche) sont gérés simultanément au clavier : la souris peut aussi bien être utilisée pour faire de l'édition en Basic que pour éditer une page Vidéotex.

- Enfin, une extension MIDI, à laquelle vient se raccorder une boîte comportant les prises standard MIDI (cette boîte ne contient que de la connectique, l'interface est intégrée au TELESTRAT, mais il était impossible de rajouter deux connecteurs DIN et deux jacks sur le TELESTRAT ...).

- Bien entendu, une sortie PERITEL (avec son) pour attaquer les télévisions et la plupart des moniteurs.

- Enfin, pour lire les programmes de l'ATMOS, une interface cassette est disponible, aux mêmes normes que l'ATMOS.

Ouf, j'aurais été exhaustif en ajoutant un bouton de RESET, qui permet de se sortir de toutes les mauvaises situations en perdant un minimum de données.

L'HYPER-BASIC COMPILE ? QUID ?

L'Hyper-Basic est un langage compilé, ce qui signifie en bref

que lorsque vous tapez RUN, au lieu de lancer un interpréteur qui lira les instructions de votre programme une à une, vous lancez tout simplement un programme en langage machine. La compilation est le fait de passer d'un texte source (le programme Basic en clair) à un programme objet, en langage machine.

Généralement, les langages compilés ne tournent qu'à grands renforts de moyens et sont horriblement peu souples d'emploi : la démarche était la suivante (avant l'ère Hyper-Basic, n'ayons pas peur des mots) : charger un éditeur de texte, écrire le programme. Sortir, charger un compilateur, compiler. Sortir, charger un éditeur de lien, l'exécuter.

A ce stade, on peut enfin voir ce que donne le programme. Mais il est probable que vous en ayez oublié un ; (point-virgule) ou une , (virgule) : on repart depuis le début. Toute personne sensée trouverait cela insupportable. Le seul avantage : le programme produit est, lui, très rapide.

En revanche, les avantages d'un langage interprété sont grands : l'édition et l'exécution étant des phases intimement liées, plus de manipulations longues. En revanche, le programme résultat, lui, s'exécute on ne peut plus lentement. Donc, le cahier des charges était simple : offrir les avantages d'un langage interprété (facilité d'édition, etc.) et ceux du langage compilé (rapidité d'exécution). Je pense y être parvenu.

L'Hyper-Basic se comporte en fait comme un langage interprété sauf... que le RUN exécute bel et bien du langage machine. Un utilisateur non averti ne le saura pas; si ce n'est la rapidité d'exécution.

Exemple : la boucle -classique (FOR I = 1 TO 10000/NEXT) s'exécute en moins de 4 secondes. Une même boucle s'exécute même en seulement 1/10e de seconde (vous avez bien tu) avec COUNT/UNCOUNT.

D'une manière générale, toutes les structures de contrôle (FOR/NEXT, GOTO, GOSUB, IFTHEN ...) prennent avec l'HyperBasic un temps tout à fait négligé

de 4 à ... 200 fois plus rapide qu'un Basic "normal".

Par exemple, un GOTO ne prend que... 3 micro-secondes (300000 GOTO à la seconde !), un GOSUB/RETURN environ 30, une boucle COUNT/UNCOUNT .environ 14 micro-secondes (70000 boucles par seconde !). D'une manière générale, un programme s'exécutera de 2 à 4 fois plus vite.

Et surtout, le temps d'exécution ne dépend plus de la taille du programme. Savez-vous qu'avec un Basic non compilé, un simple GOTO avec un programme de 10000 lignes peut prendre 40 ms (25 GOTO par seconde!) ? Avec l'Hyper-Basic, la vitesse d'exécution ne dépend en aucun cas de la taille du programme.

ADIEU REFLEXES !

L'Hyper-Basic invite donc à perdre les vieux réflexes, du style éviter les REM qui ralentissent (avec l'Hyper-Basic, elles ne font pas perdre la moindre microseconde), placer les sous-programmes en début de programme pour accélérer les GOSUB, abuser des variables pour éviter des conversions décimales coûteuses (avec un Basic normal, si vous effectuez 1000 fois la boucle $I = A + 10000$, le nombre 10000 sera converti 1000 fois en binaire, sachant que la conversion prend au moins 1 ms ...), alors que l'Hyper-Basic effectue toutes les conversions à l'interprétation, une fois pour toutes.

De plus, l'analyse d'un programme montre certaines instructions très fréquentes. Elles ont donc été particulièrement optimisées. Ainsi, les boucles FOR/NEXT sans STEP sont les plus utilisées. Un algorithme spécial a donc été conçu, quatre fois plus rapide. Il est aussi de nombreuses boucles qui servent uniquement à compter, là où la variable de boucle n'a aucune importance. Dans ce cas, une nouvelle structure de contrôle a été inventée, COUNT/UNCOUNT, qui effectue une boucle en 14 micro-secondes.

été particulièrement optimisés. Or, l'expérience montre que ces instructions, avec l'affectation de variable ($A = ...$) forment 80 % des instructions d'un programme !

Faire le tour des optimisations serait long. En résumé, l'HyperBasic est le plus rapide des Basic. Il est même aussi rapide que le Forth, à machine égale.

Une expérience intéressante faire tourner le même programme sur un TELESTRAT et un ATMOS (le programme ANI de la disquette master, par exemple). Edifiant...

OÙ L'ON CAUSE DE L'ÉDITEUR

Le programme s'exécute facilement, certes. Mais encore faut-il pouvoir l'écrire. Là encore, le TELESTRAT innove et vous rend la tâche facile : l'éditeur est de type pleine page, et l'analyse de syntaxe se fait à l'entrée de la ligne.

L'éditeur de programme du TELESTRAT est d'une simplicité à toute épreuve : vous vous déplacez sur la ligne à modifier (clavier, souris, joystick), vous modifiez (insertion automatique, effacement de tout ou partie de la ligne, suppression en insertion, etc.), vous validez (retum ou souris ou joystick), c'est tout.

En fait, la simplicité d'usage n'apporte que peu de commentaires, sinon qu'il sera difficile de vous en passer!

Des éditeurs pleine page, ce n'est pas nouveau, à part sur des machines de ce prix. Mais celui-ci possède une caractéristique supplémentaire : le déplacement le long du listing. Exemple: votre programme fait 100 lignes. Evidemment, il ne tient pas sur l'écran. Avec tout autre ordinateur, vous vous déplacerez à grands coups de LIST, ^C, zut, c'est pas celle que je voulais, LIST, cette fois je l'ai eue, mais j'avais oublié de modifier la 34, LIST... Avec le TELESTRAT, rien de tout cela, vous avez à l'écran les lignes 50 à 200 ? Vous voulez la ligne 210 ? Il suffit de des-

) et la ligne 210 apparaît. Bien sûr, la ligne 50 a disparu. Il suffit de remonter pour la faire réapparaître. L'écran est donc une fenêtre sur le texte Basic, fenêtre dans laquelle on se déplace à volonté. Attention ! si vous touchez cet éditeur, vous n'en voudrez plus d'autre...

J'ANALYSE SYNTAXIQUE

Voilà pour l'éditeur. Mais qu'est-ce au juste, cet analyseur syntaxique ? C'est méchant ?

Sous ce nom se cache une réalité agréable : vous ne pouvez pas faire d'erreur de syntaxe.

En effet, l'analyse de syntaxe est effectuée dès l'entrée de la ligne : impossible de valider une ligne dont la syntaxe n'est pas correcte. C'est déjà pratique : inutile de relancer les 3000 instructions de votre programme pour que votre modification s'achève sur un "syntax error".

Mais le TELESTRAT ne s'arrête pas là, il ne délivre pas un message barbare (et frustrant pour les débutants) du style "syntax error". Il se paie le luxe de déplacer le curseur sur le caractère qui n'a pas pu être interprété correctement. Par exemple, si vous tapez "POKE 3000A + B, le curseur se positionne sur le A. Votre esprit vif diagnostiquera qu'il manque une virgule. Vous vous dites, c'est incroyable, jusqu'où vont-ils aller ?

Toujours plus loin. En effet, non content de placer le curseur sur l'erreur, le TELESTRAT va vous dire ce qu'il attendait. Dans le cas qui nous concerne, le message "Attention: ',' " va s'afficher, vous invitant à placer une virgule. Une vingtaine de messages de ce type vous guideront. Vous pouvez jouer les novices, en tapant, par exemple FOR, puis en validant un à un les paramètres. Vous verrez apparaître tour à tour "variable numérique', '=, ' expression numérique', 'TO', ,expression numérique'.

Encore une innovation dont il sera difficile de se passer...

Vous voyez que le monde se sépare : il y a l'Hyper-Basic et les autres.



Allez, au passage, quelques facilités supplémentaires, qui feront le lien avec un aperçu des commandes disponibles !

L'Hyper-Basic invite à la structuration, grâce à ces nombreuses structure de contrôle. Rien n'est plus beau qu'un programme dont les boucles ressortent bien. Mais rien de plus fastidieux que de créer des indentations à grands coups d'espaces et de tabulations. Mais l'Hyper-Basic est là ! Les indentations sont générées 'automatiquement' par le TELESTRAT. Un FOR, REPEAT ou un COUNT, et le texte se déplace vers la droite. Un NEXT, un UNTIL... et le texte se replace. Facile ! Evidemment, les motsclés séparés par des espaces, c'est joli et facilite la lecture. Mais ça prend du temps, de la place mémoire,... Stop. Le TELESTRAT génère les espaces où il faut. Inutile de les taper. La preuve.

```
10FOR I = 1 TOCOS(I 2)
```

```
20A=A+1 AND B
```

```
30NEXT
```

List et hop

```
10 FOR 1 = 1 TO COS(I 2)
```

```
20 A=A+ 1 AND B
```

```
30 NEXT
```

Inutile de préciser que tout ceci ne prend ni de temps d'exécution, ni de la place en mémoire. Au fait, vous auriez pu taper tout ceci en minuscules, le résultat aurait été le même

JE STRUCTURE DE CONTROLE

Tout est fait pour écrire de manière structurée, c'est-à-dire de manière claire :

- Les structures FOR/NEXT, COUNT/UNCOUNT, REPEAT/UNTIL, WHILE/WEND, GOSUB/RETURN. Il reste tout de même GOTO, IF THEN ELSE !

- La substitution totale des numéros de ligne par des labels. Labels qui peuvent avoir, comme les variables d'ailleurs, autant de caractères (tous significatifs) que vous voulez.

Pour définir un label, il suffit de commencer ligne par]label Ainsi, 10]ESSAI suffit à définir le label essai. Les commandes RUN 10 et RUN ESSAI, GOTO ESSAI et GOTO 10, GOSUB 10 et GOSUB ESSAI, etc. sont équivalentes. Plus la peine de se souvenir de la signification de la ligne 8700, qui va d'ailleurs changer au premier RENUM : on l'appelle par son nom. Que de temps gagné ! Inutile de préciser que la lisibilité est incroyablement facilitée.

Autre possibilité - et non des moindres - c'est la suppression (facultative) du mot-clé GOSUB. Ainsi, un sous-programme peut

être appelé par son nom, sans taper GOSUB

```
100]SAUVE
```

```
110 SAVEU "PROGRAMME",  
AUTO
```

```
120 RETURN
```

Pour appeler le sous-programme, il suffit de frapper au clavier (ou d'inclure dans un programme) SAUVE. En d'autres termes, le vocabulaire s'enrichit à l'infini. De plus, il n'y a aucune perte de temps à utiliser cette méthode. Un programme peut s'écrire

```
10 ]DEBUT
```

```
20 INI
```

```
30 EXE
```

```
40 GOTO DEBUT
```

```
60 ]EXE
```

```
70 REPEAT
```

```
80 VALEUR
```

```
90 TEST
```

```
100 ACTION
```

```
110 UNTIL VRAI
```

```
120 RETURN
```

Donc, vive les structures.

JE FACILITE

Pour faciliter la vie du programmeur, il est bon de lui éviter des conversions incessantes et

autres. Voici, en vrac (et entre autres) quelques facilités :

- Mode degré, pour tous les calculs trigonométriques (mots-clés DEGRE/RADIAN).

- 4 types de constantes numériques -décimale, hexadécimale, binaire, et Ascii.

65 = # 41 = % 1000001 ='A'

- Les chaînes admettent directement les codes de couleur, sans les classiques CHR\$(27). De plus, les couleurs apparaissent dans le listing.

- Les entrées-sorties sont banalisées. Une simple instruction pour changer de fenêtre d'écran, pour que les PRINT se dirigent vers la RS 232 au lieu de l'écran, ou pour se servir du Minitel comme clavier. Cet aspect nécessite un article à lui tout seul, nous le développerons ultérieurement.

JE BASIQUE

La liste des mots-clés serait fastidieuse. Disons simplement qu'ils couvrent, outre les structures de contrôle, toutes les interfaces (RING pour détecter une sonnerie, SLOAD pour charger un programme par la RS 232, MPRINT pour écrire sur le Minitel, LLIST pour lister sur l'imprimante, WINDOW pour définir les fenêtres d'écrans, etc.).

Une partie du vocabulaire est destinée à la TELEMATIQUE, et est développée dans ce chapitre. Bien entendu, le graphisme et les sons ne sont pas oubliés.

Attardons-nous sur la gestion de fichier, complément indispensable à toute application sérieuse. L'accès séquentiel est on ne peut plus simple à utiliser : on ouvre un fichier, on écrit les données, on les lit séquentiellement... Il faut tout de même noter la présence d'instructions permettant de se déplacer dans le fichier (REWIND pour aller au début, APPEND pour se placer à la fin, JUMP pour sauter des enregistrements).

L'accès direct, quant à lui, frappe par sa simplicité d'emploi et sa rapidité. La puissance n'est pas oubliée, voici quelques chiffres

65535 enregistrements,

512 octets maximum pour enre-

gistrement, 25 champs maximum par enregistrement,

0,2 secondes pour accéder à une fiche,

occupation minimale de la disquette (aucun séparateur, les secteurs sont remplis au maximum).

La simplicité d'utilisation a été le maître-mot pendant la conception.

Une fois le fichier créé avec CREATE (édition pleine page des champs, avec leur nom, le type, etc.), il suffit d'un OPEN, et le

tour est joué. Finies les manipulations lourdes avec les FIELD, LSET, RSET : les champs sont des variables comme les autres. Par exemple : 10 OPEN 1,1"ESSAI" 20 TAKE -1, 12 30 INPUT NOM\$ 40 INPUT 1, 12 et le champ NOM\$ est modifié Les champs peuvent être aussi bien numériques qu'alphanumériques, pour éviter les fastidieuses conversions. Enfin, 16 fichiers peuvent être ouverts simultanément.

TELEMATIQUE INTÉGRÉE

La force du TELESTRAT est, plus encore que l'HyperBasic, la présence d'un logiciel télématique entièrement intégré. C'est déjà *0 une révolution.

Voyons quelles sont ses caractéristiques.

Ce logiciel peut être abordé sous deux angles :

- e Application à part entière, ne nécessitant aucune connaissance de l'informatique ni du monde Vidéotex. Cet aspect permet de se constituer un serveur en un temps record, dont les performances n'ont pas à rougir face à la concurrence, bien au contraire. Ne serait-ce que par la rapidité d'accès aux pages, etc..

- e Application interactive avec le Basic, permettant les applications les plus complexes et les plus personnalisées.

LE SERVEUR CLEF EN MAIN

La force de la télématique à la manière TELESTRAT est sa facilité d'emploi. Ceux qui débutent avec le TELESTRAT trouvent ceci normal, et ils ont raison. Mais tous ceux qui ont tâté d'autres matériels sont stupéfaits.

Passons rapidement sur la possibilité de consulter un serveur, de sauver sur disque les pages Vidéotex, de les imprimer rapide-

ment (mode texte) ou avec les graphisme. Ces possibilités sont classiques. Bien que la possibilité de faire en quelques secondes un mini-serveur pour les consulter facilement soit vraiment très pratique.

Qu'est-ce qu'un serveur ? Une suite de pages Vidéotex dans lesquelles on se déplace en validant des choix ou grâce à des mots-clés tapés sur le Minitel. La logique d'affichage des pages (leur ordre, leur nom) s'appelle une arborescence. Il faut ensuite que le serveur sache détecter une sonnerie, lancer l'arborescence, raccrocher (éventuellement tout seul si le correspondant ne se manifeste plus), etc.

Ainsi, la télématique, outre l'émulation, se divise en trois parties : - la composition des pages Vidéotex, - la création de l'arborescence, - le lancement du serveur.

LA COMPOSITION DES PAGES VIDEOTEX

C'est la partie dans laquelle le concepteur du serveur passe le

plus de temps (avec le TELESTRAT). En effet, même si vous êtes un as du monde Vidéotex, écrire des pages demande du temps et... des qualités artistiques indubitables. Toutefois, si vous n'êtes pas artiste, les pages Vidéotex en mode texte sont très faciles à composer.

Il y a plusieurs types de commande. D, abord, modifier les attributs taille, couleur des caractères, etc. Ensuite, déplacer des blocs de pages (couper/coller), les effacer. Il est aussi facile de dessiner directement en mode graphique, sans avoir à se soucier des difficultés du mode graphique Vidéotex : l'écran a une résolution de 0 par 72, et c'est tout ! L'usage de la souris est un véritable régal.

Et, le fin du fin pour vos animations, il est enfantin de préciser la taille de la page (qui ne remplit pas forcément tout l'écran), la position du curseur, l'affichage du curseur...

Bref, l'éditeur de page n'a aucune limite, si ce n'est les vôtres.

JE SERBORNETESTE ?

Une fois les pages créées, l'arborescence dessinée, il ne reste plus qu'à exploiter le serveur.

Le moyen le plus classique est de relier le Minitel à une ligne téléphonique (c'est sûrement déjà fait !) et d'attendre qu'on vous appelle. Dans ce cas, l'écran du TELESTRAT, véritable espion, vous permettra de suivre la communication. Sachez que dans ce mode, tout ce que vous tapez sur le Minitel sera reçu par le correspondant. Très drôle si vous avez l'esprit farceur !

Evidemment, le TELESTRAT offre des moyens de tests du serveur, en utilisant le Minitel local comme s'il était à l'autre bout de la ligne. Ceci ne coûte rien en communication, mais permet d'être certain du fonctionnement du serveur, avant même le premier appel. Rappelons tout de même qu'un serveur ne vous coûte rien en communication puisqu'on vous appelle.

Très puissant est le mode "Borne de communication". Il permet d'utiliser un serveur en mode local, avec affichage sur le moniteur du TELESTRAT. Idéal pour faire une démo dans la vitrine d'un magasin, pour consulter le programme d'une exposition...

De plus, ce mode permet, grâce aux accès Basic, de mixer des dessins haute résolution à l'arborescence, de commander des périphériques, de lancer des spots, une sono, un instrument Midi, grâce aux cartes d'entrée/sortie.

Sans compter les innombrables applications de téléchargement... Mais, n'oubliez pas qu'un dessin haute résolution, s'il peut être transmis par Minitel, ne peut pas être affiché sur celui-ci

HYPER-BASIC et TELEMATIQUE NE FONT QU'UN

La télématique et l'Hyper-Basic sont deux logiciels séparés mais qui communiquent, c'est normal ! Ainsi, le Basic comporte un vocabulaire télématique fourni, dont voici un aperçu :

MSAVE, MLOAD pour transmettre des programmes, zone

mémoire... MPRINT pour écrire sur le Minitel,

MOUT pour envoyer un caractère (MOUT 12 efface l'écran),

GET] pour saisir des caractères, RING pour détecter une sonnerie, CONNECT/UNCONNECT pour prendre ou libérer la ligne,

WCXFIN pour attendre la connexion du correspondant,

SERVEUR pour lancer un serveur, PAGE\$ pour savoir la page où l'on se trouve,

MINITEL pour relancer la télématique après un traitement par le Basic,

APPLIC pour accéder directement aux menus de la télématique à partir du Basic.

Difficile d'imaginer un outil plus puissant pour les applications télématiques. Sa force est de ne pas disperser inutilement les efforts (et donc le temps de développement) sur des points triviaux tels que la gestion de l'arborescence.

Sans écrire une ligne de Basic, il est possible de faire un très beau serveur, permettant la prise de rendez-vous, par exemple, ou l'utilisation en simple répondeur. Grâce -au Basic, on a accès à la gestion de fichiers, aux calculs sur les données entrées par les correspondants.

Du garagiste au courtier en assurance, de l'agence de voyage au VRP, tout le monde a besoin de prendre des commandes, des rendez-vous, ou même de renseigner sur ses services en dehors des heures de bureau.

Evidemment, le TELESTRAT peut se transformer en automate pour se déplacer à travers un serveur. Et, pour conclure, sachez qu'il est facile de programmer le TELESTRAT à distance, à travers un simple clavier de MINITEL : appuyer sur SHIFT pendant l'initialisation, pour signaler de se servir du Minitel comme écran de visualisation. Taper ensuite RING:CONNECT et aller appeler avec un Minitel, le tour est joué. Il suffit de taper SHIFT ENVOI au lieu de RETURN. Pourquoi ne pas rentabiliser votre TELESTRAT en initiant à domicile les gens à la programmation ?