

tremplin micro

Fonte pour AppleWorks

**Le nouveau Basic
du GS arrive !**

**Affichez
la Très Haute Résolution
sur votre GS !**

Le monde des C

Le Polyglotte

**Toujours
plus avec
votre Apple* !**



M 1631 - 17 - 33,00 F



N° 17 - Bimestriel - Troisième année
5 Novembre 1987 - 4 Janvier 1988
241 FB - 11 FS - **33 F**

tremplin micro 17

SOMMAIRE

Avec la collaboration de :

ARGOS, Claude AUBRY, Didier CHARRON, Maurice CHAVELLI,
Bruno CHEVALLIER, Marcel COTTINI, Rafi DERYEGHIYAN, Roland
JOST, Yvan KOENIG, NESTOR, Clément RENARD.

Apple et ProDOS (noms et logos) sont des marques déposées d'Apple Computer, Inc.

BIMESTRIEL

Le numéro : 33 F
Abonnement d'un an : 190 F
(6 numéros)

Tous nos prix sont indiqués TTC.

EDITIONS JIBENA

Direction-Rédaction :

Editions JIBENA

Guy-HACHETTE

La Petite Motte — Senillé
86100 CHÂTELLERAULT.

Téléphone :

49-93-66-66

PUBLICITÉ :

Raymond JULLIEN

(1) 45.75.41.81

Commission paritaire :

Les revues qui choisissent d'être réellement au service du Lecteur, en ne l'obligeant pas à glaner, dans plusieurs magazines, les renseignements concernant sa machine, ne bénéficient pas du numéro de Commission Paritaire, et pas davantage des tarifs postaux réduits.

TREMPIN MICRO — Bimestriel — C'est une publication des Editions JIBENA, 4, rue de la Cour-des-Neues, 75020 PARIS — S.A. au capital de 3 600 000 F — Imprimé par CITÉ-PRESS/PARIS — Service de vente : Presse-Promotion, tél. : 49.93.65.03 — Dépôt légal à la date de parution — Inscription à la Commission Paritaire des Publications et Agences de Presse : en cours — Directeur de la Publication : Guy-Clément COGNÉ — Diffusion N.M.P.P.

La Disquette TREMPIN MICRO contient tous les programmes du numéro, ainsi que les sources trop longs pour être publiés dans les colonnes de la revue.

Yvan KOENIG a visité Apple Expo	2
LE POLYGLOTTE avec Maurice CHAVELLI	3
WAR Un jeu d'aventure pour Apple	9
(nous ne publions pas ce long programme, mais vous recevrez la disquette de Rafi DERYEGHIYAN... pour 20 F).	
FONTE pour AppleWorks (Didier CHARRON)	11
CLASSEMENT suivant la longueur (Clément RENARD)	19
PLACE.DOS - DOS 3.3 pas mort (Marcel COTTINI)	21
PAVÉS ALÉATOIRES en Basic plus langage machine	24
LOUPE (Roland JOST)	25
UN X DEUX, initiation au langage machine (Nestor)	36
BLOLIB.A et BLOLIB.V, toujours de l'initiation (Nestor)	37
GS Basic: Il arrive... avec les outils	39
MÉLANGEUR, pour mêler aléatoirement N mots (Nestor)	40
QUATRE OPÉRATIONS, du Basic au LM	43
RND Ø ou 1 : comment obtenir aléatoirement des Ø et des 1 ...	44
ALLER-RETOUR jusqu'aux routines arithmétiques du 65CØ2 ...	45
TRI.B, trier une liste de nombre sur un octet	46

LE MONDE DES C

Claude AUBRY poursuit son initiation et la vôtre avec un vrai programme

47

MINI-DOC Encore un Startup sous ProDOS (Bruno CHEVALLIER) 51

Affichez la Très Haute Résolution sur votre GS

Yvan KOENIG vous livre le fruit de ses cogitations sur la Très Haute Résolution. Une disquette indispensable à celles et à ceux qui ont la chance de posséder un GS

57

MLI Petite initiation en langage machine (Argos)	61
C CONVERTI Un micro-programme en C (Nestor)	63
HGR MONOCHROME Réponse intéressante à un Lecteur	65
SuperMacroWorks La version française existe !	69
GRIBOUILLE PARLANT Le génie logiciel français	70
Yvan KOENIG se déchaîne dans un courrier spécial	73
TURBO PASCAL TUTOR Pour comprendre TURBO PASCAL ...	78
BIBLIOTHÈQUE INFORMATIQUE	20,66,68,74

YVAN KOENIG a visité Apple expo...

GS is well and alive

Je sais : certains vont supposer que cette fois les fruits de la passion m'ont frappé deux fois. C'est peut-être vrai, mais, à mon retour d'APPLE expo, je suis persuadé que le GS, malgré des débuts difficiles en France, décolle enfin.

Les raisons de ses difficultés initiales sont multiples : prix sans doute trop élevé, défauts dans les programmes livrés avec la machine, difficulté d'accès aux outils de développement, sans oublier, si j'en crois divers appels de lecteurs et des informations de source sûre, le comportement suicidaire de certains concessionnaires.

Les prix viennent de baisser.

Les programmes ont fait l'objet de révisions. Il convient de préciser que certains des problèmes rencontrés par les utilisateurs ne sont pas en totalité imputables aux auteurs, qui étaient (et sont encore) tributaires des outils de source APPLE (système d'exploitation encore évolutif, drivers disponibles de manière tardive). Attention ! je ne porte aucune accusation contre CUPERTINO ! L'équipe qui travaille sur la machine (nous avons rencontré trois de ses membres) semble tellement éprise de perfectionnisme qu'elle remanie sans cesse ce qu'elle vient de sortir.

Rich Williams nous a d'ailleurs apporté la version 3.1 du système d'exploitation, mais il y a fort à parier qu'il existera une version 3.1+ lorsque cette version 3.1 sera parvenue à vos machines. Il nous a, par ailleurs, présenté un éditeur d'icônes, et surtout un outil de génération de sons, tout à fait remarquables. Nous avons enregistré avec plaisir l'annonce pour 88, de la vraie version 16 bits du système d'exploitation.

Des nouveautés hard sont aussi à l'ordre du jour, mais le développeur que je suis (ou plutôt que j'étais encore) a promis la discrétion.

Je rappelle à ceux qui n'ont pu obtenir de leur

fournisseur les mises à jour 1.1 de GS.PAINT et 2.0 de GS.WRITE qu'ils peuvent les demander directement à Version Soft.

Après avoir tiré les leçons des difficultés d'accès aux outils de développement, APPLE a remanié la présentation de ceux-ci et a confié leur diffusion à une nouvelle structure, à savoir :

D.D.A. I. Development
"Technopole", rue Faraday
78180 Montigny le Bretonneux
Tél. : (1) 30.45.26.62.

J'ai rencontré de jeunes développeurs dont l'enthousiasme pour le GS tranchait avec le scepticisme affiché par certains (BORLAND par exemple). Ayant fait part de certains propos alarmistes aux représentants du team GS, j'ai été frappé par leur surprise et je crois pouvoir dire qu'ils travaillent dur pour que toute crainte se révèle vaine.

De son côté, J.-L. GASSEE a affirmé qu'APPLE gagne de l'argent avec le GS et que cela le conduisait à penser que l'APPLE II sera encore présent dans 10 ans, avec des modifications bien sûr. Il n'a rien voulu annoncer aux 200 personnes (au moins) qui l'écoutaient, mais nous sommes tous sortis convaincus de la mise en œuvre prochaine d'un système de communication entre les deux gammes APPLE.

Enfin, et je sais que vous n'allez pas me croire, le BASIC spécial GS existe. Je l'ai vu à l'œuvre, et il sera disponible chez D.D.A. lorsque vous lirez ces lignes. Aux U.S.A., A.P.D.A. le vend au prix de 50\$ et annonce déjà un compilateur pour ce BASIC. Avant de vous quitter, je tiens à mentionner ce qui restera mon meilleur souvenir d'APPLE expo, à savoir le sourire d'Alexandre FERAY (18 ans), lauréat, avec son langage Forth pour GS, du concours APPLE/Version Soft (que les 4 autres lauréats ne m'en veuillent pas, ma réaction n'enlevant rien aux mérites de leurs programmes).

Yvan KOENIG

LE POLYGLOTTE

Je ne suis pas un fervent adepte de la langue d'origine de mon Apple chéri et, un matin de nostalgie, je me suis mis en tête de lui apprendre ma langue maternelle. J'avais pas mal avancé mon projet lorsque je me suis aperçu que j'allais remplacer une contrainte linguistique par une autre alors que la grande qualité de l'informatique est plutôt la souplesse ! J'ai alors imaginé un Basic idéal, modifié à loisir, et qui me permette d'utiliser aussi bien le PRINT classique que IMPRIME, MARQUE, ECRIT... suivant l'humeur du moment !

C'est dans cette optique un rien surréaliste que s'est construit *Le Polyglotte*. Ce dernier permet de modifier à tout moment la syntaxe d'un mot réservé du Basic, sans pour autant corriger les programmes. Mais j'ai découvert en chemin une possibilité que je n'avais pas prévue au départ. Les mots-clés dans un programme Basic sont représentés par des codes (Tokens) dont le premier est \$80 pour END, le second \$81 pour FOR et ainsi de suite jusqu'à \$EA pour MID\$ (vous pouvez consulter utilement à ce sujet l'article de NESTOR à la page 6 de *Tremplin Micro* n°10). Il reste donc 21 codes disponibles de \$EB à \$FF !

En temps normal ces tokens ne sont pas utilisables, mais avec *Le Polyglotte* ils le deviennent, il est ainsi facile d'ajouter des commandes au Basic de la manière la plus naturelle qui soit.

Comment fonctionne Le Polyglotte ?

Le Basic est tout entier contenu en mémoire morte, il n'est donc pas possible de le modifier. Par contre, il utilise une routine située en mémoire vive pour lire les programmes et le buffer d'entrée en commande directe : c'est le fameux CHRGET situé en \$B1, ce dernier étant donc orienté, ou plutôt désorienté, sur la routine d'analyse du *Polyglotte*. Il est ainsi possible de capter et d'interpréter une instruction avant l'Applesoft. ProDos m'a également

donné beaucoup de soucis ! Lui aussi interprète tout au moyen du mode trace du Basic standard, et il se trouve pris au piège en essayant d'écrire le "£", ce qui l'envoie finalement dans la gestion de sortie de ProDos en \$9E2C.

Les mots-clés du Basic sont tous situés dans une table qui commence en \$D0D0. En version de base, *Le Polyglotte* comprend la même table, mais située en mémoire vive donc modifiable.

Les mots-clés du Basic apparaissent sur votre écran dans deux cas :

- lorsque vous entrez des instructions au clavier,
- lorsque vous utilisez la commande LIST.

Dans le premier cas, il peut s'agir d'une commande directe ou de l'enregistrement d'une ligne d'un programme. Dans les deux cas il faut tokéniser, c'est-à-dire remplacer les mots-clés par leur code. La routine de tokénisation standard se situe en \$D559. *Le Polyglotte* reprend cette routine mais l'oriente évidemment sur sa propre table. Dans le deuxième cas, le Basic utilise la routine LIST située en \$D6A5 ; *Le Polyglotte* comporte une routine identique, mais orientée sur sa propre table.

Comment utiliser Le Polyglotte ?

Le Polyglotte comporte une gestion complète et ne pose pas de problème particulier quant à son utilisation. Il convient toutefois d'être prudent dans le choix des mots-clés que vous affectez. Si vous remplacez FOR par DE, il ne faut pas ensuite remplacer WAIT par DELAI car les deux premières lettres de ce mot seront déjà considérées comme un mot-clé et vous déboucherez sur un SYNTAX ERROR, en anglais lui ! Soyez prudent également avec vos noms de variables et n'oubliez pas qu'ils doivent respecter les mots-clés. Pour limiter l'encombrement de la table, j'ai limité la longueur des mots-clés à 10 caractères.

(suite page 4)

D'autre part, *Le Polyglotte* est assemblé en \$8A00 mais il doit au préalable être chargé et lancé en \$4000. En effet, il doit d'abord "faire sa place" en décalant les buffers de ProDos vers le bas et ensuite se reloger. Vous pouvez donc calculer que *Le Polyglotte* occupe en tout 4K de \$8A00 à \$9A00. Lorsque vous êtes satisfait du *Polyglotte* que vous avez constitué, vous devez le sauver avec l'option correspondante du menu. Par la suite, pour le retrouver, il vous suffira de faire un BRUN, et le tour sera joué, même si vous aviez un autre *Polyglotte* en place. Cette manœuvre ne vous fait pas perdre le programme Basic en place, mais écrase 8 pages à partir de \$4000.

Le menu du *Polyglotte* s'obtient par pomme ouverte et le point d'interrogation actionnés en même temps !

Comment ajouter des commandes ?

Classiquement, il existe deux méthodes pour ajouter une commande au Basic. La première, la plus courante, utilise l'Ampersand (&) et vous avez de très nombreux exemples dans *Tremplin Micro*. La

deuxième, plus rarement employée, consiste à modifier CHRGET : c'est cette méthode qui a été retenue par *Le Polyglotte*. Mais le grand avantage de ce dernier réside dans l'utilisation de nouveaux tokens, ce qui simplifie considérablement le problème.

Le Polyglotte vous demande juste un nom pour votre commande et l'adresse de celle-ci. Il est bien entendu que le nom est modifiable par la suite. Le programme correspondant doit respecter le fonctionnement de CHRGET. *Le Polyglotte* empile l'adresse \$D7D1 pour le retour et place le token correspondant en \$6. Je vous propose, dans la disquette d'accompagnement de ce numéro 17, deux exemples d'illustration, le premier est le classique IF THEN ELSE, le second est inspiré du Pascal : c'est le REPEAT UNTIL. Les deux sont relogeables, les sources* comportent tous les renseignements nécessaires. J'espère que ces deux exemples vous inspireront. Vous pouvez très facilement adapter la routine SWAP parue dans *Tremplin Micro* n°4 ou le FILL de *Tremplin Micro* n°7, mais laissez plutôt s'exprimer votre imagination...

* EXEMPLE.SOURCE (un seul fichier).

POLY.OBJET ayant été lancé par BRUN POLY.OBJET,A\$4000 (ne pas oublier l'adresse) vous l'activerez, comme le précise Maurice Chavelli en appuyant simultanément sur POMME OUVERTE et le POINT D'INTERROGATION.

Un menu apparaîtra alors et vous proposera ces options :

- EDITER LES COMMANDES ACTUELLES.
- AJOUTER UNE COMMANDE.
- SUPPRIMER UNE COMMANDE.
- SAUVER LE POLY ACTUEL.

Vous utiliserez flèche droite ou flèche gauche pour effectuer votre choix. Personnellement, j'aurais préféré les deux autres flèches, plus logiques et je pense que notre ami YVAN KOENIG serait de mon avis.

EDITION Pas de problème particulier. On se promène dans la liste, toujours avec les flèches. Deux colonnes : le premier token est celui que pratique votre Apple. Le second est le même, sauf modification. A ce stade, pour modifier une commande, il suffit de presser la touche RETURN et de taper le mot (le vôtre) qui remplacera le mot-clef

actuel. Ainsi, à la place du classique END vous pourriez rentrer FIN. Moi, je préfère OUF ! Quand c'est fait : RETURN, ordre parfaitement compris par tout Apple, mais auquel les IBM et compatibles préfèrent ENTER.

AJOUTER Il y a ceux qui savent (Maurice)... et les autres. C'est vrai que l'emploi de l'option AJOUTER tombe sous le sens... quand on sait. Bien ! C'est quand même pas la mer à boire. Imaginons que vous puissiez installer une routine de COPIE D'ÉCRAN à l'adresse \$300... et qu'elle y soit effectivement. Pour l'automatiser par une commande IMP, il vous suffira de taper IMP (évidemment) quand POLYLOTTE vous le demandera puis 0300 quand il s'inquiétera de l'adresse. Tout ça est enfantin. Pour le reste, aucun problème particulier... sauf que Maurice Chavelli joue avec les buffers de ProDOS, mais il faut bien s'amuser, pas vrai.

LE SOURCE Vous le trouverez sur la disquette car il est bien trop long pour figurer dans la revue.

Les conseils du sage NESTOR

POLY OBJET

**A\$8A00
LS0C13**

Si vous n'avez jamais saisi une routine en langage machine, tapez d'abord un CALL - 151, puis sitôt l'apparition de l'astérisque (*), tapez gentiment chaque ligne, sans vous presser.

Attention ! ne tenez pas compte du total indiqué en bout de ligne. Il est destiné aux heureux possesseurs de notre disquette **SIGNATURE** et permet de vérifier la saisie.

Quand vous aurez terminé, sauvez la totalité de la routine par un indispensable :

**BSAVE POLY.OBJET,
A\$8A00, L\$C13**

8A00:	AD 88 BC C9 8B 90 13 A9 10 20 F5 BE 90 08 A9 0C	3BC1
8A10:	20 0C BE 4C D0 03 C9 8A D0 F4 A9 8A 85 43 A9 00	EAC4
8A20:	85 42 85 3C A9 40 85 3D A9 FF 85 3E A9 4F 85 3F	F65A
8A30:	A0 00 20 2C FE A9 4C 85 BA A9 69 85 BB A9 8A 85	9728
8A40:	BC A9 4C 8D F1 9E A9 DC 8D F2 9E A9 8A 8D F3 9E	A3C0
8A50:	A9 EE 8D C9 9A A9 8A 8D CA 9A 60 AE E2 92 AC E3	D7BC
8A60:	92 C9 3A B0 03 4C BE 00 60 8E E2 92 8C E3 92 BA	B46F
8A70:	BC 01 01 C0 4C F0 04 C0 1F F0 30 BC 02 01 C0 D4	9A10
8A80:	D0 D9 20 61 8A AA 68 68 8A D0 03 4C 3C D4 A2 FF	5888
8A90:	86 76 90 06 20 4E 91 4C 05 D8 A6 AF 86 69 A6 B0	DF54
8AA0:	86 6A 20 0C DA 20 4E 91 4C 6A D4 BC 02 01 C0 D8	9AD6
8AB0:	D0 A9 C9 BC D0 03 4C 03 92 C9 EB 90 9E AA 68 68	2D0E
8AC0:	8A E9 EB 0A A8 A9 D7 48 A9 D1 48 B9 5A 94 48 B9	7442
8AD0:	59 94 48 A0 00 B1 B8 85 06 4C B1 00 C9 BC D0 03	111E
8AE0:	4C 05 92 C9 EB B0 DA 85 33 A8 38 4C F4 9E 20 10	CBC7
8AF0:	FD 2C 61 C0 30 01 60 C9 BF D0 FB AD 1F C0 8D E4	0A2B
8B00:	92 10 05 A9 95 20 ED FD 20 39 FB 20 58 FC A9 82	81E2
8B10:	A0 93 20 3A DB 20 33 91 A9 0A 20 5B FB A9 00 85	EFA3
8B20:	24 A9 A2 A0 93 20 3A DB 20 FB DA A9 C5 A0 93 20	198D
8B30:	3A DB 20 FB DA A9 DE A0 93 20 3A DB 20 FB DA A9	8097
8B40:	F9 A0 93 20 3A DB A9 0A 20 5B FB 20 33 91 20 1E	88AC
8B50:	91 C9 9B D0 10 AD E4 92 10 05 A9 03 20 95 FE 20	918C
8B60:	58 FC 20 D0 03 C9 95 D0 17 A5 25 C9 0D D0 04 A9	CAA9
8B70:	09 85 25 E6 25 20 27 91 20 22 FC 20 33 91 30 CE	F4B6
8B80:	C9 88 D0 0E A5 25 C9 0A D0 04 A9 0E 85 25 C6 25	E3EC
8B90:	D0 E3 C9 8D F0 06 20 D0 92 4C 4E 8B A5 25 C9 0D	9A46
8BA0:	D0 03 4C 5A 8F C9 0C D0 03 4C 26 8E C9 0B D0 03	8657
8BB0:	4C 81 8D A9 17 85 23 20 5B FB A9 00 85 24 A9 0A	7C3D
8BC0:	A0 93 20 3A DB 20 33 91 20 3F 91 20 B5 90 20 1E	33DF
8BD0:	91 C9 95 F0 F6 C9 9B D0 03 4C 08 8B C9 8D F0 06	FA37
8BE0:	20 D0 92 4C CE 8B A9 17 20 5B FB A9 00 85 24 A9	5B58
8BF0:	32 A0 93 20 3A DB 20 33 91 A9 00 85 25 20 22 FC	B50F
8C00:	20 33 91 20 1E 91 C9 95 D0 19 A6 25 EC E5 92 F0	0718
8C10:	04 E0 16 D0 06 20 B5 90 4C F9 8B 20 27 91 E8 86	8B4B
8C20:	25 D0 DA C9 88 D0 13 A6 25 D0 06 20 D0 92 4C 03	D275
8C30:	8C 20 27 91 CA 86 25 4C FD 8B C9 9B D0 03 4C B3	7FE3
8C40:	8B C9 8D D0 E6 20 27 91 A5 25 48 A9 17 20 5B FB	A0B7
8C50:	A9 00 85 24 A9 5A A0 93 20 3A DB 20 33 91 68 20	6B29
8C60:	5B FB A9 12 85 24 20 9C FC 20 7D 90 8A D0 03 4C	FB48
8C70:	B3 8B 8E E2 92 CA F0 0C CA BD 00 02 29 7F 9D 00	A8D4
8C80:	02 CA 10 F5 A5 1C 48 A5 1D 48 A0 00 A6 25 F0 15	6554
8C90:	D0 06 E6 1C D0 02 E6 1D B1 1C 10 F6 CA D0 F3 E6	21F3
8CA0:	1C D0 02 E6 1D A5 1C 48 A5 1D 48 88 C8 A1 1C 30	E141
8CB0:	08 E6 1C D0 F7 E6 1D D0 F3 E6 1C D0 02 E6 1D C8	C136
8CC0:	8C E3 92 AD E2 92 38 ED E3 92 F0 48 B0 03 4C 49	9C3C
8CD0:	8D A8 A5 1C 85 1E A5 1D 85 1F D0 06 E6 1C D0 02	8BA9
8CE0:	E6 1D A1 1C D0 F6 F0 16 A5 1C C5 1E D0 06 A5 1D	66C8
8CF0:	C5 1F F0 11 C6 1C A5 1C C9 FF D0 02 C6 1D A1 1C	50C2
8D00:	91 1C 4C E8 8C 8C E3 92 A5 1A 18 6D E3 92 85 1A	76C6
8D10:	90 02 E6 1B 68 85 1D 68 85 1C A0 00 B9 00 02 91	F492
8D20:	1C C8 CC E2 92 D0 F5 68 85 1D 68 85 1C A5 25 48	9A0E
8D30:	A9 17 20 5B FB A9 00 85 24 A9 32 A0 93 20 3A DB	CCCB
8D40:	20 33 91 68 85 25 4C FD 8B 38 AD E3 92 ED E2 92	E185
8D50:	A8 C6 1C A6 1C E0 FF D0 02 C6 1D 88 D0 F3 A8 A2	D475

LE POLYGLOTTE (POLY.OBJET : suite)

8D60:	00 B1 1C 81 1C F0 08 E6 1C D0 F6 E6 1D D0 F2 8C	157B
8D70:	E3 92 A5 1A 38 ED E3 92 85 1A B0 02 C6 1B 4C 14	EF60
8D80:	8D AD E6 92 C9 2A D0 06 20 D0 92 4C 4E 8B 20 58	339A
8D90:	FC A9 05 85 24 A9 C5 A0 93 20 3A DB 20 33 91 A9	27B6
8DA0:	07 20 5B FB 20 FB DA A9 32 A0 94 20 3A DB 20 7D	9753
8DB0:	90 8A D0 03 4C 08 8B 8E E2 92 CA F0 0C CA BD 00	481B
8DC0:	02 29 7F 9D 00 02 CA 10 F5 20 3F 91 A0 00 E6 1A	7CA8
8DD0:	D0 02 E6 1B B1 1A D0 F6 B9 00 02 91 1A C8 CC E2	D940
8DE0:	92 D0 F5 A9 00 91 1A A9 0A 20 5B FB 20 FB DA A9	BB72
8DF0:	41 A0 94 20 3A DB A9 04 8D A3 90 20 7D 90 A9 0A	51F7
8E00:	8D A3 90 A0 00 20 A7 FF AE E6 92 C6 3E A5 3E 9D	CED0
8E10:	59 94 C9 FF D0 02 C6 3F A5 3F 9D 5A 94 EE E6 92	EA61
8E20:	EE E6 92 4C 08 8B AD E6 92 D0 06 20 D0 92 4C 4E	C35C
8E30:	8B 20 58 FC A9 04 85 24 A9 DE A0 93 20 3A DB 20	A464
8E40:	33 91 A9 16 20 5B FB 20 FB DA A9 32 A0 93 20 3A	AD56
8E50:	DB 20 33 91 A9 01 20 5B FB 20 FB DA 20 3F 91 A0	B864
8E60:	00 A2 6B E6 1A D0 02 E6 1B B1 1A 10 F6 CA D0 F3	9D3E
8E70:	E6 1A D0 02 E6 1B A5 1A 85 1C A5 1B 85 1D 84 24	773D
8E80:	B1 1A F0 37 48 09 80 20 ED FD E6 1A D0 02 E6 1B	5BA0
8E90:	68 10 ED A9 14 85 24 A9 A4 20 ED FD A4 25 88 88	9CFB
8EA0:	98 0A A8 BE 59 94 C8 B9 59 94 E8 D0 03 18 69 01	9BA0
8EB0:	20 41 F9 A0 00 20 FB DA 4C 80 8E C6 25 A5 25 8D	DE8B
8EC0:	E5 92 A9 01 85 25 E6 25 20 22 FC 20 33 91 20 1E	4336
8ED0:	91 C9 95 D0 0C 20 27 91 A5 25 CD E5 92 F0 E3 D0	7A54
8EE0:	E5 C9 88 D0 14 20 27 91 A5 25 C9 02 F0 04 C6 25	2C66
8EF0:	D0 D6 AD E5 92 85 25 D0 CF C9 9B D0 03 4C 08 8B	1329
8F00:	C9 8D F0 06 20 D0 92 4C CE 8E A6 25 CA CA 8A 48	2DA7
8F10:	C9 14 F0 0D 0A AA BD 5B 94 9D 59 94 E8 E0 28 D0	9D84
8F20:	F5 CE E6 92 CE E6 92 A0 00 68 AA F0 15 D0 06 E6	90F4
8F30:	1C D0 02 E6 1D B1 1C 10 F6 CA D0 F3 E6 1C D0 02	0D25
8F40:	E6 1D 88 C8 B1 1C 10 FB C8 B1 1C 81 1C F0 08 E6	D83B
8F50:	1C D0 F6 E6 1D D0 F2 4C 08 8B 20 58 FC A9 04 85	E62C
8F60:	24 A9 F9 A0 93 20 3A DB 20 33 91 20 FB DA A2 00	45A9
8F70:	BD EC 92 9D 00 02 E8 E0 06 D0 F5 A9 8D 9D 00 02	B542
8F80:	20 03 BE A0 27 A9 AD 91 28 88 10 F9 20 FB DA 20	0B5D
8F90:	FB DA 20 FB DA A9 13 A0 94 20 3A DB 20 0C FD C9	07E1
8FA0:	CF F0 4F 20 FB DA 20 FB DA 20 FB DA A9 EC A0 92	47B4
8FB0:	20 3A DB A9 28 8D A3 90 20 7D 90 A0 0A 8C A3 90	965C
8FC0:	8A D0 03 4C 08 8B A9 8D 9D 06 02 CA BD 00 02 9D	013D
8FD0:	06 02 CA 10 F7 E8 BD EC 92 9D 00 02 E8 E0 06 D0	2139
8FE0:	F5 20 FB DA 20 03 BE 90 06 20 0C BE 20 1E 91 4C	8266
8FF0:	5A 8F 20 FB DA 20 FB DA 20 FB DA A9 32 A0 94 20	13F7
9000:	3A DB A2 00 BD F6 92 9D 00 02 20 ED FD E8 E0 05	EF72
9010:	D0 F2 A9 14 8D A3 90 20 80 90 A9 0A 8D A3 90 E0	11C2
9020:	05 D0 03 4C 08 8B 8E E2 92 A0 00 B9 FB 92 9D 05	2D41
9030:	02 E8 C8 C9 8D D0 F4 AE E2 92 CA BD 00 02 9D 05	BA19
9040:	02 CA 10 F7 E8 BD E7 92 9D 00 02 E8 E0 05 D0 F5	5622
9050:	A0 00 84 3C 84 42 A9 40 85 43 A9 8A 85 3D A9 FF	2174
9060:	85 3E A9 99 85 3F 20 2C FE 20 FB DA 20 03 BE 90	B479
9070:	09 20 0C BE 20 1E 91 4C 5A 8F 4C 08 8B A2 01 CA	6B43

QUESTION

Lecteur de votre revue depuis le premier numéro, j'apprécie le nombre et la qualité des articles, et surtout leur caractère didactique.

Très modeste programmeur en Basic Applesoft, vous m'avez tiré sans le savoir de bien des tracas et je ne m'étendrai pas, car vous connaissez bien cela, sur la joie de voir enfin un petit programme tourner, même si le résultat est dû aux ficelles découvertes dans TREPLIN MICRO plutôt qu'à mes talents personnels.

Si je vous écris aujourd'hui, c'est que j'ai l'intention de changer mon "vieux" Apple IIC pour un Apple IIGS et que ce projet me plonge dans des abîmes de perplexité. Je m'explique; ma configuration actuelle est la suivante :

- 1 APPLE II C 128 KO.
- 1 LECTEUR EXTERNE MARQUE APPLE (5 pouces 1/4).
- 1 MONITEUR ZENITH MONOCHROME VERT
- 1 MODEM UNIVERSEL SECTRAD (AGRÉÉ APPLE).
- 1 IMPRIMANTE APPLE IMAGEWRITER I.
- 1 SOURIS.
- 1 JOYSTICK APPLE.
- 1 CÂBLE PÉRITEL "CHAT MAUVE" (Pour les jeux en couleurs).

Je travaille principalement sur APPLEWORKS V1.2 ainsi que sur VERSION TEL et VERSION CALC. Mes questions sont les suivantes :

- Si j'achète un IIGS, quels sont les périphériques que je pourrai conserver sans problème ?
- Pourrai-je récupérer mes (nombreux) fichiers Appleworks ?
- Mes logiciels tourneront-ils sur un IIGS ?

Si je viens vous ennuyer avec ces questions, c'est que malgré de nombreuses démarches auprès de différents revendeurs (y compris ceux qui m'ont vendu mon IIC il y a 3 ans) je n'ai jamais obtenu de réponses précises. L'un me dit que mon lecteur 5 pouces 1/4, bien que de marque Apple, ne sera pas compatible; un second le contredit; un troisième me prédit des problèmes sans nom entre le IIGS et l'IMAGEWRITER I; un autre m'assure que mon Modem est totalement incompatible; quand à APPLE FRANCE, ils n'ont jamais daigné répondre à mes lettres...

Voilà! vous êtes donc mon dernier recours, j'espère que vous pourrez m'éclairer et, dans l'attente de vous lire, je vous prie de recevoir mes cordiales salutations.

J.-F. P. (PARIS 13^e)

RÉPONSE

Voilà quelques mois que je signale à APPLE les comportements "CURIEUX" de certains de leurs revendeurs agréés, mais ça ne semble pas s'améliorer.

Je vais essayer d'éclairer votre lanterne :

- Votre lecteur 5" IIC est probablement un UNIDISK beige. Il est "normalement" compatible GS. Il se peut cependant, s'il s'agit d'un exemplaire "ancien" qu'il ait besoin d'une petite intervention (c'est le cas des premiers DUO-DISKS). Seul un essai pourrait vous renseigner. Si vous en avez l'occasion, il faudrait le brancher sur un GS et essayer le FINDER de la disquette système 3.008 du 10 juin 87. Si le lecteur vide se voit attribuer une icône, tout va bien. Sinon, écrivez au SAV en précisant le numéro de série de votre lecteur.
- Désolé mais le modem SECTRAD ne fonctionne sur aucune des listes de compatibilité que j'ai pu consulter.
- Votre souris sera inutilisable mais le GS est livré avec SA souris.
- Le Joystick sera utilisable.
- Le câble Péritel Chat Mauve est sans objet ici. Si vous achetez une unité centrale seule, il existe un câble qui vous permettra d'utiliser votre téléviseur équipé d'une prise "PÉRITEL" (résultat variant du correct à l'insupportable selon les téléviseurs).
- L'IMAGEWRITER 1 est compatible avec le GS mais ne l'est peut-être pas avec certains logiciels GS (tout est affaire de nuance).
- Je ne connais pas le moniteur ZENITH et je préfère ne rien dire sur ce point.

De toutes façons, j'envoie copie de votre lettre et de ma réponse à APPLE pour qu'ils puissent, éventuellement, compléter ou corriger.

Yvan KOENIG.



9080:	20	0C	FD	C9	95	D0	04	B1	28	D0	0F	C9	88	D0	07	8A	BCC5
9090:	F0	EE	C6	24	D0	E9	C9	8D	F0	15	C9	A1	90	E2	C9	E0	4561
90A0:	B0	DE	E0	0A	F0	DA	9D	00	02	20	ED	FD	E8	D0	D1	9D	7911
90B0:	00	02	4C	9C	FC	20	58	FC	A0	00	B1	1A	D0	03	20	3F	35F7
90C0:	91	A5	1A	85	1C	A5	1B	85	1D	A0	00	84	24	B1	18	F0	0654
90D0:	0F	48	09	80	20	ED	FD	E6	18	D0	02	E6	19	68	10	ED	781E
90E0:	B1	1A	F0	33	A9	0D	85	24	A9	AD	20	ED	FD	A9	BE	20	3E34
90F0:	ED	FD	E6	24	E6	24	E6	24	B1	1A	48	09	80	20	ED	FD	58AE
9100:	E6	1A	D0	02	E6	1B	68	10	EF	E6	25	A5	25	C9	17	F0	17DF
9110:	06	20	22	FC	4C	C9	90	A6	25	CA	8E	E5	92	60	AD	00	E790
9120:	C0	10	FB	8D	10	C0	60	A0	27	B1	28	09	80	91	28	88	E0F2
9130:	10	F7	60	A0	27	B1	28	29	3F	91	28	88	10	F7	60	A9	48C0
9140:	D0	85	18	85	19	A9	83	85	1A	A9	94	85	1B	60	A6	B8	D571
9150:	CA	A0	04	84	13	E8	BD	00	02	24	13	70	04	C9	20	F0	3930
9160:	F4	85	0E	C9	22	F0	74	70	4D	C9	3F	D0	04	A9	BA	D0	C9A2
9170:	45	C9	30	90	04	C9	3C	90	3D	84	AD	A9	83	85	9D	A9	36CC
9180:	93	85	9E	A0	00	84	0F	88	86	B8	CA	C8	D0	02	E6	9E	4497
9190:	E8	BD	00	02	C9	20	F0	F8	38	F1	9D	F0	EE	C9	80	D0	0235
91A0:	41	05	0F	C9	C5	D0	0D	BD	01	02	C9	4E	F0	34	C9	4F	F7D3
91B0:	F0	30	A9	C5	A4	AD	E8	C8	99	FB	01	B9	FB	01	F0	39	3002
91C0:	38	E9	3A	F0	04	C9	49	D0	02	85	13	38	E9	78	D0	86	8DBA
91D0:	85	0E	BD	00	02	F0	DF	C5	0E	F0	DB	C8	99	FB	01	E8	AA04
91E0:	D0	F0	A6	B8	E6	0F	B1	9D	C8	D0	02	E6	9E	0A	90	F6	250F
91F0:	B1	9D	D0	9D	BD	00	02	10	BB	99	FD	01	C6	B9	A9	FF	0C03
9200:	85	B8	60	68	68	A9	D8	48	A9	22	48	20	B1	00	90	0B	2DB5
9210:	F0	09	C9	C9	F0	05	C9	2C	F0	01	60	20	0C	DA	20	1A	D106
9220:	D6	20	B7	00	F0	10	C9	C9	F0	04	C9	2C	D0	EC	20	B1	8EB5
9230:	00	20	0C	DA	D0	E4	68	68	A5	50	05	51	D0	06	A9	FF	3153
9240:	85	50	85	51	A0	01	B1	9B	F0	44	20	58	D8	20	FB	DA	EA11
9250:	C8	B1	9B	AA	C8	B1	9B	C5	51	D0	04	E4	50	F0	02	B0	4792
9260:	2D	84	85	20	24	ED	A9	20	A4	85	29	7F	20	5C	DB	A5	50FD
9270:	24	C9	21	90	07	20	FB	DA	A9	05	85	24	C8	B1	9B	D0	9ED5
9280:	1D	A8	B1	9B	AA	C8	B1	9B	86	9B	85	9C	D0	B6	A9	0D	B84D
9290:	20	5C	DB	4C	D2	D7	C8	D0	02	E6	9E	B1	9D	60	10	CC	54F4
92A0:	38	E9	7F	AA	84	85	A0	83	84	9D	A0	93	84	9E	A0	FF	E78B
92B0:	CA	F0	07	20	96	92	10	FB	30	F6	A9	20	20	5C	DB	20	747A
92C0:	96	92	30	05	20	5C	DB	D0	F6	20	5C	DB	A9	20	D0	98	E202
92D0:	A0	A0	A2	F2	AD	30	C0	CA	EA	EA	D0	FB	88	EA	EA	D0	1706
92E0:	F1	60	00	00	00	00	00	C2	D3	C1	D6	C5	D0	D2	C5	C6	346F
92F0:	C9	D8	A0	BA	A0	00	D0	CF	CC	D9	AE	AC	C1	A4	B4	B0	1202
9300:	B0	B0	AC	CC	A4	B1	B0	B0	B0	8D	A0	A0	A0	A0	A0	A0	B58A
9310:	A8	D2	C5	D4	D5	D2	CE	A9	A0	A0	A0	A0	A0	A8	C5	D3	4791
9320:	C3	A9	A0	A0	A0	A0	A0	AD	BE	A0	A0	A0	A0	A0	A0	A0	D857
9330:	A0	00	A0	A0	A0	A0	A0	A8	D2	C5	D4	D5	D2	CE	A9		3E91
9340:	A0	A0	A0	A8	C5	D3	C3	A9	A0	A0	A0	BC	AD	A0	A0		EDB5
9350:	A0	AD	BE	A0	A0	A0	A0	A0	00	A0	A0	A0	A0	A0	A0		ED8B
9360:	A0	A0	A0	A8	D2	C5	D4	D5	D2	CE	A9	A0	A0	A0	A0	BC	6E4D
9370:	AD	A0	A0	A0	AD	BE	A0	A0	A0	A0	A0	A0	A0	A0	A0		4838
9380:	A0	00	A0	A0	A0	A0	A0	A0	CC	C5	D3	A0	CF	D5	D4		457C
9390:	C9	CC	D3	A0	C4	D5	A0	D0	CF	CC	D9	C7	CC	CF	D4	D4	F68F
93A0:	C5	00	A0	A0	A0	A0	C5	C4	C9	D4	C5	D2	A0	CC	C5	D3	AE06
93B0:	A0	C3	CF	CD	CD	C1	CE	C4	C5	D3	A0	C1	C3	D4	D5	C5	CB49
93C0:	CC	CC	C5	D3	00	A0	A0	A0	A0	C1	CA	CF	D5	D4	C5	D2	FC4A
93D0:	A0	D5	CE	C5	A0	C3	CF	CD	CD	C1	CE	C4	C5	00	A0	A0	8A2C

LE POLYGLOTTE

(POLY.OBJET : fin)

93E0:	A0 A0 D3 D5 D0 D0 D2 C9 CD C5 D2 A0 D5 CE C5 A0	D82F
93F0:	C3 CF CD CD C1 CE C4 C5 00 A0 A0 A0 A0 D3 C1 D5	F02D
9400:	D6 C5 D2 A0 CC C5 A0 D0 CF CC D9 A0 C1 C3 D4 D5	6C4F
9410:	C5 CC 00 A0 A0 A0 A0 CC E5 A0 D0 D2 C5 C6 C9 D8	3830
9420:	A0 E5 F3 F4 AD E9 EC A0 E3 EF F2 F2 E5 E3 F4 BF	8FBF
9430:	A0 00 A0 A0 A0 A0 D1 F5 E5 EC A0 EE EF ED BF A0	CC80
9440:	00 A0 A0 A0 A0 D1 F5 E5 EC EC E5 A0 E1 E4 F2 E5	1024
9450:	F3 F3 E5 BF A0 A0 A0 A4 00 00 00 00 00 00 00 00	B10E
9460:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0000
9470:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0000
9480:	00 00 00 45 4E C4 46 4F D2 4E 45 58 D4 44 41 54	5556
9490:	C1 49 4E 50 55 D4 44 45 CC 44 49 CD 52 45 41 C4	0F1C
94A0:	47 D2 54 45 58 D4 50 52 A3 49 4E A3 43 41 4C CC	37F9
94B0:	50 4C 4F D4 48 4C 49 CE 56 4C 49 CE 48 47 52 B2	81B6
94C0:	48 47 D2 48 43 4F 4C 4F 52 BD 48 50 4C 4F D4 44	8730
94D0:	52 41 D7 58 44 52 41 D7 48 54 41 C2 48 4F 4D C5	F1B8
94E0:	52 4F 54 BD 53 43 41 4C 45 BD 53 48 4C 4F 41 C4	B912
94F0:	54 52 41 43 C5 4E 4F 54 52 41 43 C5 4E 4F 52 4D	C9B7
9500:	41 CC 49 4E 56 45 52 53 C5 46 4C 41 53 C8 43 4F	E429
9510:	4C 4F 52 BD 50 4F D0 56 54 41 C2 48 49 4D 45 4D	A036
9520:	BA 4C 4F 4D 45 4D BA 4F 4E 45 52 D2 52 45 53 55	4533
9530:	4D C5 52 45 43 41 4C CC 53 54 4F 52 C5 53 50 45	F23A
9540:	45 44 BD 4C 45 D4 47 4F 54 CF 52 55 CE 49 C6 52	043A
9550:	45 53 54 4F 52 C5 A6 47 4F 53 55 C2 52 45 54 55	6F38
9560:	52 CE 52 45 CD 53 54 4F D0 4F CE 57 41 49 D4 4C	4768
9570:	4F 41 C4 53 41 56 C5 44 45 C6 50 4F 4B C5 50 52	74A3
9580:	49 4E D4 43 4F 4E D4 4C 49 53 D4 43 4C 45 41 D2	4FC2
9590:	47 45 D4 4E 45 D7 54 41 42 A8 54 CF 46 CE 53 50	9023
95A0:	43 A8 54 48 45 CE 41 D4 4E 4F D4 53 54 45 D0 AB	B387
95B0:	AD AA AF DE 41 4E C4 4F D2 BE BD BC 53 47 CE 49	0A40
95C0:	4E D4 41 42 D3 55 53 D2 46 52 C5 53 43 52 4E A8	022D
95D0:	50 44 CC 50 4F D3 53 51 D2 52 4E C4 4C 4F C7 45	1753
95E0:	58 D0 43 4F D3 53 49 CE 54 41 CE 41 54 CE 50 45	7052
95F0:	45 CB 4C 45 CE 53 54 52 A4 56 41 CC 41 53 C3 43	DC09
9600:	48 52 A4 4C 45 46 54 A4 52 49 47 48 54 A4 4D 49	79C5
9610:	44 A4 00	A2E8

BORLAND PRÉSENT À APPLE EXPO

A l'occasion du salon Apple Expo, BORLAND INTERNATIONAL a présenté toute sa gamme de logiciels dédiés Macintosh.

Le marché Macintosh étant en pleine croissance et représentant pour le constructeur son deuxième marché mondial après les Etats-Unis, il est normal que BORLAND INTERNATIONAL, fidèle à une politique commerciale consistant à proposer des produits aussi populaires que possible, suive l'évolution de ce secteur.

Devant le succès rencontré par ses logiciels pour PC, la société, soucieuse d'offrir les mêmes performances aux utilisateurs de Macintosh, propose donc toute une gamme (langage et bureautique) spécifique à ce type de produits.

Pour BORLAND INTERNATIONAL, c'est également la France qui constitue son deuxième marché après les Etats-Unis.

LOGICIELS EXPOSÉS :

Turbo Pascal pour le Mac.

Turbo Tutor pour le Mac.

Méthodes numériques pour le Mac.

Database toolbox pour le Mac.

Reflex pour le Mac.

Sidekick pour le Mac.

Eurêka, la solution pour le Mac.

L'ASSEMBLEUR ? mais c'est facile !

Commencez par :

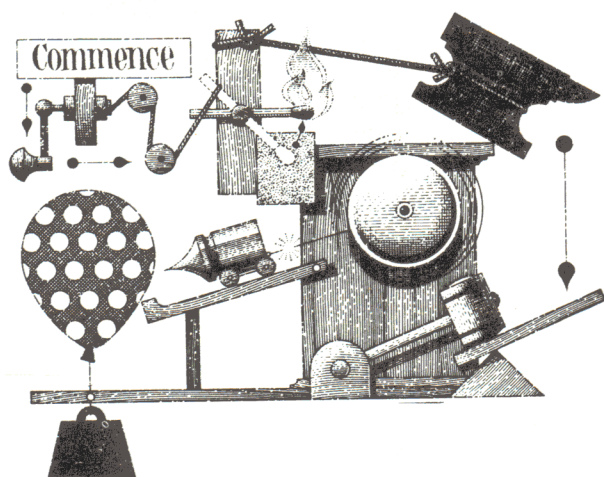
LE 6502 PAS À PAS

(ouvrage que nous devons à deux lecteurs de Tremplin Micro).

Offrez-vous nos bouquins de routines (avec disquette).

- CLINS D'OEIL AU 6502
- ROUTINES POUR 65C02 ET 6502
- NOUVELLES ROUTINES POUR LE 65C02

Utilisez le BULLETIN DE COMMANDE à la fin du journal.



DESCRIPTION DU PROGRAMME

WAR est un jeu d'aventure entièrement écrit en Langage Machine. Il comporte un véritable analyseur syntaxique. Toutes les explications sont contenues dans le programme : en voyageant à travers le temps et l'espace, vous devrez mettre fin à une guerre qui a dévasté la surface de la terre.

Un triple problème se pose donc à vous :

- trouver la machine à remonter le temps,
- en obtenir le fonctionnement,
- découvrir le moyen d'arrêter la guerre.

De plus, vous devez déterminer si la solution se trouve dans le passé ou le futur...

Comme dans tout jeu d'aventure, vous donnez vos ordres à l'aide d'un verbe à l'impératif suivi d'un nom. Le verbe peut être abrégé à trois lettres, et le nom à quatre.

Tapez E, O, N, S, pour vous déplacer, INV pour l'inventaire, QUI pour arrêter. A vous de découvrir le reste du vocabulaire.

UTILISATION

Le programme est composé :

- Du fichier binaire "TEXTES" contenant les codes ASCII des messages affichés à l'écran.
- Du fichier binaire "VECTEURS" contenant des adresses.
- Enfin, du programme lui-même : WAR.

WAR

Le source du programme (WAR.SOURCE et WAR1.SOURCE sur la disquette) a été écrit grâce à l'assembleur ProCODE. J'ai été obligé de le séparer en deux à cause de son importance.

Pour jouer, faites uniquement BRUN WAR. Le programme charge automatiquement les fichiers TEXTES et VECTEURS : ils doivent donc être sauves obligatoirement sous ces noms sur la disquette :

- BSAVE WAR, A\$4000, L\$1084
- BSAVE VECTEURS, A\$800, L\$118
- BSAVE TEXTES, A\$7000, L\$1FBE

ATTENTION : Le programme a été conçu sous le système d'exploitation ProDOS. Pour le faire fonctionner sous DOS 3.3, voici les étapes à suivre :

- Remplacer les trois premiers codes du programme WAR par : 4C E2 40.
- BLOAD VECTEURS.
- BLOAD TEXTES.
- BRUN WAR.

De plus, à chaque fois que le programme vous demandera si vous voulez rejouer, répondez non, rechargez le fichier TEXTES, puis faites BRUN WAR.

LA SOLUTION...

Je vous livre la solution pour vous prouver qu'elle existe (et par la même occasion que mon programme fonctionne !).

PRENDS VASE, CASSE VASE, REGARDE DEBRIS,
PRENDS PASSE, E, N, REGARDE MEUBLES,
PRENDS COUTEAU, S, POUSSE TAPIS, (suite page 10).

OUVRE TRAPPE, PASSE, VA TRAPPE, PRENDS ALLUMETTES, PRENDS PISTOLET, E, REGARDE CADRAN, POUSSE BOUTON, REGARDE CADRAN, TIRE LEVIER, E, S, TUE CHEVALIERS, PISTOLET, E, E, E, VA CHEMINEE, TOURNE CHENET, VA PORTE, PRENDS DIAMANTS, N, O, O, O, S, FOUILLE FOIN, PRENDS PELLE, POSE PISTOLET, N, O, N, O, POUSSE BOUTON, REGARDE CADRAN, POUSSE BOUTON, REGARDE CADRAN, TIRE LEVIER, E, E, DONNE DIAMANTS, ROBOT, PRENDS DESINTEGRATEUR, O, O, POUSSE BOUTON, REGARDE CADRAN, POUSSE BOUTON, REGARDE CADRAN, TIRE LEVIER, O, VA ECHELLE, E, E, E, REGARDE SOLDAT, TUE SOLDAT, DESINTEGRATEUR, POSE DESINTEGRATEUR, FOUILLE SOLDAT, PRENDS CLEF, S, OUVRE PORTE, CLEF, POSE CLEF, VA PORTE, PRENDS DYNAMITE, N, N, O, O, O, VA TRAPPE, E, POUSSE BOUTON, POUSSE BOUTON, POUSSE BOUTON,

BOUTON, TIRE LEVIER, E, ALLUME ALLUMETTES, REGARDE DYNAMITE, ALLUME MECHE, VA BRECHE, REGARDE CONDUIT, DEVISSE GRILLE, COUTEAU, VA CONDUIT, E, REGARDE PAN-CARTE (!) REGARDE ORDINATEUR, BRANCHE ORDINATEUR, ALLUME ECRAN, REGARDE ECRAN, O, O, O, O, POUSSE BOUTON, POUSSE BOUTON, POUSSE BOUTON, TIRE LEVIER, E, E, S, REGARDE OFFICIER, TUE OFFICIER, COUTEAU, REGARDE OFFICIER, PRENDS UNIFORME, ENTERRE OFFICIER, PELLE, S, E, KRIEG, E, TUE GENERAL, COUTEAU.

Le programme compte en tout 65 mots dans son vocabulaire. Bien entendu, j'ai essayé de programmer le plus de situations possibles : par exemple, je ne vous conseille pas de fouiller le soldat alors qu'il est encore vivant, ou de le tuer avec une autre arme que le désintégrateur ! Et ce n'est qu'un piège parmi tant d'autres... ■

Attention ! ce programme est beaucoup trop long pour être publié dans la revue, mais *TREMLIN MICRO* vous l'offre moyennant une faible participation : 20F.

Utilisez obligatoirement notre bulletin de commande à la fin du journal.

La disquette (ProDOS) comporte non seulement le programme, mais le source. Vous devrez par contre y copier ProDOS et BASIC.SYSTEM.



FONTE

pour
Apple IIe, IIc, IIGS
et APPLEWORKS

Cet utilitaire a pour objectif de vous permettre d'agrémenter vos textes, sous *APPLEWORKS*, avec des polices de caractères personnalisés. Il peut également être employé avec tout logiciel travaillant sous *PRODOS*. Il est utilisable sous *DOS 3.3* en modifiant certaines lignes du programme *FONTE*.

FONTE est divisé en 2 parties : le *CHARGEUR* et le *GÉNÉRATEUR*. Dans le *CHARGEUR* comme dans le *GÉNÉRATEUR*, le choix des rubriques se fait toujours avec *flèches haute ou basse*.

1. CHARGEUR

a. ...retour au sommaire

- permet de revenir au sommaire sans télécharger, ou après avoir chargé une fonte dans l'imprimante (taper 2 fois sur *RETOUR*).

b. charger une ancienne fonte

- permet de télécharger la fonte employée lors d'une précédente utilisation de l'utilitaire *FONTE* (allumer l'imprimante, taper sur *RETOUR*, attendre le chargement, puis taper *RETOUR*).

c. charger une nouvelle fonte

- permet de télécharger une fonte à partir d'un choix effectué d'après le catalogue des fontes disponibles (allumer l'imprimante, taper sur *RETOUR*, choisir une fonte à l'aide des flèches haute ou basse, valider en tapant sur *RETOUR*, attendre le chargement, et taper sur *RETOUR*).

d. supprimer une fonte

- permet d'effacer une fonte du disque (taper sur *RETOUR*, choisir la fonte à effacer à l'aide des flèches haute ou basse, valider en tapant sur *RETOUR*, répondre O (oui) pour confirmer).

e. créer une nouvelle fonte

- permet d'accéder au *GÉNÉRATEUR*. (suite page 12)

Ce logiciel comporte plusieurs volets :

- 1° **STARTUP** (écrit en Basic) est le menu de présentation. Il vous propose *APPLEWORKS — FONTE — MAILING — IMPRIMER FONTES*.
- 2° **VOIR** permet d'éditer toutes les fontes disponibles sur papier.
- 3° **FONTE** permet de charger une fonte dans la mémoire vive de l'imprimante et aussi de créer ses propres polices de caractères.

Nota : le *MAILING* n'est qu'une suggestion de programme. Vous pourrez avoir, sur la même disquette, tous les programmes *FONTE*, plus ceux d'*APPLEWORKS*. Vous pourrez aussi ajouter un *MAILING* éventuel !

Les lignes 105 et 215 de *STARTUP* doivent être modifiées pour une utilisation sur la disquette */APPLEWORKS*

ou la ligne 280 pourrait être réécrite de la façon suivante :

```
280 PRINT D$"/APPLEWORKS/APLWORKS.SYSTEM"
```

Il est à noter que pour le démarrage de *FONTE* sur la disquette */APPLEWORKS*, il est impératif que *BASIC.SYSTEM* soit devant *APLWORKS.SYSTEM* sur le catalogue.

Marche à suivre :

- Faire une copie des programmes *FONTE*.
- Transférer tous les fichiers de la face 1 d' *APPLEWORKS* sur cette nouvelle disquette, à l'aide de *MOUSEDESK* par exemple.
- Effectuer le même travail avec le *MAILING* (éventuel).
- Renommer la disquette */FONTE* en */APPLEWORKS* avec l'instruction *RENAME* ou avec *MOUSEDESK*.
- Recopier sur la face 2 de votre nouveau disque, la face 2 d'*APPLEWORKS*.

Les fichiers de police de caractères doivent toujours être sur la disquette où se trouve *FONTE*. Certaines polices de *TREMLIN MICRO* sont utilisables, mais les caractères devront être décalés... sinon on aura B au lieu de A, et ainsi de suite. ■

2. GÉNÉRATEUR

Il est organisé de la même façon que le chargeur.

a. utiliser une FONTE vierge

- permet de créer une fonte à partir d'une fonte vierge (taper RETOUR).

b. charger une ancienne fonte

- permet de créer une fonte à partir de la dernière fonte établie dans une utilisation précédente du logiciel (taper RETOUR).

c. charger une nouvelle fonte

- permet d'appeler en mémoire une fonte du catalogue pour travailler avec (taper RETOUR, choisir une fonte à l'aide des flèches haute ou basse, taper RETOUR).

d. sauvegarder la fonte en mémoire

- permet de sauver sur disque la fonte en mémoire (taper RETOUR, taper le nom désiré - pour effacer des caractères : utiliser flèche gauche ou delete -, taper RETOUR quand le nom convient).

3. GÉNÉRATION DES CARACTÈRES

Dans les rubriques a, b, c, du générateur vous appelez l'éditeur de caractères :

- A la question : **APPEL CAR?** vous pouvez répondre par n'importe quel caractère dont le code ASCII est entre 32 et 126 inclus. L'éditeur vous proposera dans une grille le caractère dessiné point par point. Si vous tapez RETOUR à la place d'une lettre, vous aurez une grille vierge. Si vous tapez ESC(escape), vous sortirez de l'édition.
- Le curseur est maintenant sur la grille, vous pouvez (re)définir votre caractère. Déplacez-vous avec les flèches. Pour *ajouter ou supprimer un point de la matrice* : tapez sur **ESPACE**. Pour *sortir de l'édition*, enfoncez la touche **ESC** (escape).
- A la question : **TYPE H ou h**, tapez H pour utiliser la matrice haute et h pour la matrice basse (lettre avec jambage...)
- A la question : **RANGE CAR?** vous tapez le caractère sous lequel celui que vous avez défini doit être rangé. Si vous tapez ESC(escape), il n'y aura pas de rangement ; vous sortirez alors de l'édition pour une sauvegarde éventuelle.

4. RÈGLES sur les touches à utiliser

- (RETOUR)** valide tous vos choix.
(ESC) vous ramène toujours à l'état précédent.

(?) permet d'avoir le catalogue des fontes disponibles.

(ESPACE) sert à définir la matrice d'un caractère.

5. NOTA

Ce logiciel fonctionne très bien avec les imprimantes DMP, IMAGEWRITER I et II. Sur les Apple IIe, la carte 80 colonnes est obligatoire, le Kit 65C02 sera aussi le bienvenu, mais n'est pas nécessaire.

Le chargement des fontes en mémoire s'effectue toujours à l'adresse \$4000 et 95 caractères sont disponibles (du code ASCII 32 au code ASCII 126 inclus). Ils se présentent sous la forme d'une matrice 8x8.

Pour faire travailler l'imprimante avec une fonte personnelle, il faut lui envoyer la séquence **ESCAPE ' (27 39)** et pour revenir à la fonte normale, c'est la séquence **ESCAPE \$ (27 36)**.

Sous APPLEWORKS, il faut définir une nouvelle imprimante :

- menu **AUTRES ACTIVITES**
- menu **DONNEES IMPRIMANTE** : choisir **AJOUTER IMPRIMANTE**
- menu **AJOUTER IMPRIMANTE** : choisir 12 et donner un nom
- menu **CODE IMPRESSION** : définir les codes de l'imprimante
- menu **SOULIGNEMENT** option imprimante à la commande START-STOP : répondre **ESCAPE ' ou ESCAPE ! ESCAPE '**
- menu **FIN SOULIGNEMENT** : répondre **ESCAPE \$ ou ESCAPE \$ ESCAPE "**

Ceci est un exemple, vous pouvez employer **GRAS** à la place de **SOULIGNEMENT**.

Dans le traitement de texte, lorsque vous voulez utiliser les *caractères spéciaux* : tapez **CONTROLE-S** ; pour revenir aux *caractères normaux* : tapez **CONTROLE-S**. Il faut savoir qu'après chaque retour chariot, le soulignement est annulé. Au moment de l'impression, demandez la *nouvelle imprimante*.

Il faudra taper **CONTROLE-G** à la place de **CONTROLE-S** si vous avez paramétré votre imprimante sur les caractères gras. L'utilisation de la fonte vierge permet d'écrire un texte en occultant tous les mots écrits en caractères personnalisés.

— CREE FONTE.FIN —

```
10 D$ = CHR$(4)
20 PRINT D$"CREATE FONTE.FIN,TTXT"
30 PRINT D$"OPEN FONTE.FIN,TTXT"
40 PRINT D$"WRITE FONTE.FIN"
50 PRINT "VIERGE"
60 PRINT "VIERGE"
70 PRINT D$"CLOSE"
```

Tapez et lancez ce court programme pour créer FONTE.FIN, indispensable lors de la première utilisation. ■

FONTE

ROUTINES

- 1** STARTUP : petit menu
- 2** VOIR : pour imprimer les fontes (IW/DMP)
- 3** FFTV : qui "fabrique" FT.VIERGE
- 4** FONTE : le gros morceau
- 5** FONTE.CODE : 51 octets en langage machine.

1 STARTUP

```

100 D$ = CHR$(4):B$ = "RUN"      884D
105 PRINT D$"PRÉ3": PRINT D$"PRE  FCAF
    FIX/TM17/FONTE"
110 REM MENU
115 DATA .....SOMMAIRE..      D18C
    .....
120 DATA * .....APPLEWORKS    389F
    .....*
125 DATA * .....FONTE .....   7AB3
    .....*
130 DATA * .....MAILING ..... 25F8
    .....*
135 DATA * .....Imprimer les fo 28CF
    ntes .....*                28CF
140 DATA &&
145 I = - 1: RESTORE : PRINT C    2A5B
    HR$(12): HOME
150 I = I + 1: READ A$: IF A$ <   B183
    "&&" THEN T$(I) = A$: GOTO    60E8
    150                          45A2
155 NBE = I - 1
160 HOME :HT = 23
165 HTAB HT: VTAB 8: INVERSE : P  951D
    RINT T$(0):; NORMAL
170 HTAB HT: VTAB 10: FOR G = 1   E994
    TO 34: PRINT CHR$(255):; N    0261
    EXT G: PRINT
175 FOR I = 1 TO NBE
180 VTAB 10 + I: HTAB HT: PRINT    3286
    T$(I)                        9DCB
185 NEXT I
190 HTAB HT: VTAB 10 + I: FOR G   16A5
    = 1 TO 34: PRINT CHR$(255)   634A
    ;: NEXT G: PRINT
195 I = 1
200 VTAB 10 + I: HTAB HT + 1: IN  9293
    VERSE : PRINT MID$(T$(I),2   FA26
    ,32):; NORMAL
205 HTAB HT: VTAB 14 + NBE: PRIN  E322
    T "Sélection :FLECHE HAUT ET
    BAS"
210 HTAB HT: VTAB 15 + NBE: PRIN
    T "Validation:RETOUR CHARIOT
    ";: GET R$
215 IF R$ = CHR$(27) THEN PRI

```

```

    NT D$"RUN/TM17/STARTUP"      AB4D
220 IF ASC (R$) = 13 THEN PRIN    438F
    T : GOTO 275                  32EB
225 IF ASC (R$) = 10 THEN 255
230 IF ASC (R$) < > 11 THEN 20    43B7
    5
235 VTAB 10 + I: HTAB HT: PRINT
    T$(I)                        3286
240 I = I - 1                    7A5C
245 IF I < 1 THEN I = NBE        CBA A
250 GOTO 200                      333D
255 VTAB 10 + I: HTAB HT: PRINT
    T$(I)                        3286
260 I = I + 1                    805B
265 IF I > NBE THEN I = 1       8AA8
270 GOTO 200                      333D
275 ON I GOTO 280,285,290,295   0BA0
280 PRINT D$"-APLWORKS.SYSTEM"  5119
285 PRINT D$ + B$ + "FONTE"     D5D8
290 GOTO 200                      333D
295 PRINT D$ + B$ + "VOIR"      FA9C

```

2 VOIR

```

100 B$ = " ..... "             0C1A
105 DIM C$(8,8)                  58DA
110 FOR F = 1 TO 8: FOR D = 1 TO  E8A6
    8:C$(D,F) = ".": NEXT D: NEX  B3A4
    T F                           CFDA
115 D$ = CHR$(4)                 47AC
120 E$ = CHR$(27)
125 PRINT D$"PREFIX": INPUT PF$
130 REM
135 TEXT : HOME : PRINT D$"PRÉ3" 1CE6
    : PRINT
140 DIM G$(50): PRINT D$"OPEN ";  FA3D
    PF$; ",TDIR": PRINT D$"READ "  4048
    ;PF$
145 G = 1                         582F
150 INPUT A$: INPUT A$: INPUT A$
155 INPUT G$(G):G$(G) = MID$(G   ABC2
    $(G),2,15)                    CF25
160 IF G$(G) = "" THEN 175
165 IF MID$(G$(G),1,3) = "FT."   0ED8
    THEN G$(G) = MID$(G$(G),4,
    12):G = G + 1

```

2 VOIR (suite et fin)

```
170 GOTO 155 2D46
175 G = G - 1: PRINT D$"CLOSE" 506E
180 PRINT D$"BLOAD FONTE.CODE" 0FAD
185 REM IMPRIMANTE
190 VTAB 18: HTAB 10: PRINT "All
  umer l'Imprimante (DMP/IW) "
  ;: INPUT A$ A7AE
195 VTAB 18: HTAB 10: PRINT "Edi
  tion des fontes disponibles
  "" 9807
200 PRINT D$"PR£1": PRINT E$"q";
  CHR$(9)"136N"; CHR$(15); C
  HR$(9)"Z" 0A49
205 PRINT E$"!" 0288
210 PRINT "EDITION DES FONTES":
  PRINT B0E9
215 PRINT "SYSTEME """"";: FOR L
  = 32 TO 126: PRINT CHR$(L)
  ;: NEXT L: PRINT BC48
220 FOR U = 1 TO G:K$ = G$(U) 1669
225 K$ = "FT." + K$ 7682
230 PRINT D$"BLOAD"K$,A$4000" 78D0
235 CALL 768 8331
240 PRINT G$(U);E$'':: FOR L =
  32 TO 126: PRINT CHR$(L);:
  NEXT L: PRINT E$"£" 662B
```

```
245 NEXT U A5D7
250 PRINT E$; CHR$(34) 8BFD
255 PRINT D$"RUN STARTUP" FFAE
```

3 FFTV

```
100 TEXT : NORMAL :D$ = CHR$(4
  ) CD3E
105 HOME : VTAB 12: INVERSE : PR
  INT "FT.VIERGE EN COURS": NO
  RMAL 80A6
110 A = 16384 9217
115 FOR I = 0 TO 3: READ R: POKE
  I + A,R: NEXT AD50
120 A = A + 3:B = 31:F = 72 2486
125 B = B + 1: POKE A + 1,B: POK
  E A + 2,F AD88
130 FOR I = A + 3 TO A + 10: POK
  E I,0: NEXT 2B55
135 A = A + 10: IF A < 17337 THE
  N 125 C0D5
140 POKE I,0 1E5E
145 PRINT D$"BSAVE FT.VIERGE,A$4
  000,L955": PRINT D$"RUN STAR
  TUP" F9D9
150 DATA 27,43,27,73 C2AA
```

4 FONTE

```
100 PRINT CHR$(4)"PR£3": PRINT ED56
105 VTAB 12: PRINT ".....
  .< Auteur >.....
  ....Didier..CHARRON.....
  ....." 9C9B
110 LOMEM: 20000:B$ = " ....."
  """:: FOR D = 0 TO 9: READ
  E: POKE 820 + D,E: NEXT : DA
  TA 104,168,104,166,223,154,7
  2,152,72,96: DIM C$(8,8): FO
  R F = 1 TO 8: FOR D = 1 TO 8
  :C$(D,F) = ".": NEXT D: NEXT
  F:D$ = CHR$(4):E$ = CHR$(
  27) B688
115 PRINT D$"PREFIX": INPUT PF$ 47AC
120 PRINT : DIM G$(50): PRINT D$
  "OPEN ";PF$;"TDIR": PRINT D
  $"READ";PF$:G = 1: INPUT A$:
  INPUT A$: INPUT A$ 5BFC
125 INPUT G$(G):G$(G) = MID$(G
  $(G),2,15): IF G$(G) = "" TH
  EN 140 D719
130 IF MID$(G$(G),1,3) = "FT."
```

```
THEN G$(G) = MID$(G$(G),4,
  12):G = G + 1 0ED8
135 GOTO 125 1B43
140 G = G - 1: PRINT D$"CLOSE":H
  = 0: IF PEEK(-1088) - 22
  4 < = 0 THEN H = 1 C1E3
145 PRINT D$"BLOAD FONTE.CODE":
  PRINT D$"OPENFONTE.FIN,TTXT"
  : PRINT D$"READ FONTE.FIN":
  INPUT H$: INPUT I$: PRINT D$
  "CLOSE":H$ = H$ + MID$(B$,
  1,12 - LEN(H$)):I$ = I$ +
  MID$(B$,1,12 - LEN(I$)):I
  = 6 C720
150 PRINT D$"PR£3": PRINT 4552
155 GOSUB 300: ONERR GOTO 290 6368
160 IF I > 14 THEN I = 6 323D
165 IF I < 6 THEN I = 14 9B3F
170 IF I = 10 THEN VTAB 1: HTAB
  35: PRINT " ....." CF41
175 VTAB 1: HTAB 3: PRINT "=>";:
  GOSUB 500: IF E = 10 THEN V
  TAB 1: HTAB 3: PRINT " ";:I
```

= I + 2: GOTO 160	B42F	"""; GOTO 160	1AF3
180 IF E = 11 THEN VTAB 1: HTAB			
3: PRINT " "; I = I - 2: GO	619C		
TO 160	6989	295 POKE 216,0: VTAB 18: POKE 14	
185 IF E = 13 THEN 195	4948	03,69: INVERSE : PRINT " ERR	
190 GOTO 175		EUR "; NORMAL : CALL 820: P	
195 ON (I / 2) - 2 GOTO 200,230,	9899	OKE 216,128: GOSUB 540: POKE	90FF
235,340,385		1403,69: PRINT " """"";	
200 VTAB 22: HTAB 39: GOSUB 500:		GOTO 605	2B4E
IF E < > 13 THEN 220	D749	300 IF H = 1 THEN 310	
205 IF J\$ < > "" THEN I\$ = J\$	896E	305 FOR D = 2 TO 23: VTAB D: HTA	
210 IF F\$ < > "" THEN H\$ = F\$	E465	B 1: PRINT "!"; POKE 1403,7	
215 PRINT D\$"OPENFONTE.FIN,TTXT"		9: PRINT "!"; NEXT D: VTAB	CA0E
: PRINT D\$"WRITE FONTE.FIN":	EC10	23: HTAB 2: FOR D = 2 TO 79:	
PRINT H\$: PRINT I\$: PRINT D\$	5E86	PRINT "_"; NEXT : GOTO 315	
"CLOSE": PRINT D\$"RUNSTARTUP	333D	310 PRINT E\$: INVERSE : FOR D =	
"	32A5	2 TO 23: VTAB D: HTAB 1: PRI	
220 IF E = 27 THEN 160	754B	NT "2"; HTAB 80: PRINT "_";	
225 GOTO 200	881E	: NEXT D: VTAB 24: HTAB 2: F	5CF3
230 F\$ = H\$: GOSUB 275: GOTO 200	E21C	OR D = 2 TO 79: PRINT "L";	
235 J = 1		NEXT D: PRINT CHR\$ (24); N	
240 IF J < 1 THEN J = G	7541	ORMAL	
245 IF J > G THEN J = 1	A8BF	315 VTAB 1: HTAB 2: FOR D = 1 TO	
250 VTAB 10: HTAB 35: INVERSE :	B129	78: PRINT "_"; NEXT D: VTAB	
PRINT " "; G\$(J); " "; NORMAL	2484	3: HTAB 5: PRINT "<?> Catalo	
: PRINT " """""; H	4B41	gue des fontes """"nombre d	
TAB 34: GOSUB 500: IF E = 11	AA85	e fontes disponibles: "; IN	
THEN J = J - 1: GOTO 240		VERSE : PRINT " "; G; " "; NO	
255 IF E = 10 THEN J = J + 1: GO		RMAL : PRINT " "; VTAB 4: H	
TO 240		TAB 2	C290
260 IF E = 13 THEN F\$ = G\$(J): G		320 FOR D = 1 TO 78: PRINT "_";	
OSUB 275: GOTO 200		NEXT D: VTAB 20: HTAB 2: FOR	
265 IF E = 27 THEN PRINT " """"		D = 1 TO 78: PRINT "_"; NEX	
"""""; GOTO 160		T D: VTAB 6: HTAB 6: PRINT "	
270 GOTO 240	Retour au SOMMAIRE"; I	
275 IF F\$ = "" THEN 240		F H\$ = "" THEN VTAB 8: HTAB	
280 K\$ = "FT." + F\$: PRINT D\$"BL		6: PRINT "Charger ancienne "	
OAD"K\$",A\$4000": VTAB 18: HT		FONTE"; GOTO 330	2CA1
AB 10: PRINT "Allumer l'Impr		325 VTAB 8: HTAB 6: PRINT "Charg	
imante (DMP/IW) "; GOSUB 50		er ancienne "FONTE"; HTAB 3	
0: PRINT D\$"PR&1": PRINT E\$"		5: INVERSE : PRINT " "; H\$;"	
N"; CHR\$ (9)"80N"; CHR\$ (15)		"; NORMAL	9A2C
; CHR\$ (9)"Z": CALL 768	3B84	330 VTAB 10: HTAB 6: PRINT "Char	
285 PRINT D\$"PR&3": PRINT : GOSU	F015	ger nouvelle "FONTE"; HTAB	
B 300: VTAB 19: HTAB 10: PRI		35: PRINT " """"";	
NT "Fonte "; INVERSE : PRIN		VTAB 12: HTAB 6: PRINT "Supp	
T " "F\$;" "; NORMAL : PRINT		rimier """"une "FONTE"; VTAB	
" installée "; RETURN		14: HTAB 6: PRINT "Créer ""n	
290 POKE 216,0: VTAB 22: HTAB 39		ouvelle "FONTE"; VTAB 15: H	
: PRINT "."; VTAB 18: POKE		TAB 2	A9D7
1403,69: INVERSE : PRINT " E		335 FOR D = 1 TO 78: PRINT "_";	
RREUR "; NORMAL : CALL 820:		NEXT D: VTAB 22: HTAB 16: PR	
POKE 216,128: POP : GOSUB 54		INT " "Sommaire <RETOUR>....	
0: POKE 1403,69: PRINT " """"	<ESC> changer de FONTE"	
		; RETURN	6C05

4 FONTE (suite)

340	VTAB 14: HTAB 35: PRINT " "				
	":J = 1	4617			
345	IF J < 1 THEN J = 6	881E			
350	IF J > 6 THEN J = 1	E21C			
355	VTAB 12: HTAB 35: INVERSE :				
	PRINT " ";G\$(J);" ";; NORMAL				
	: PRINT " ";; HT				
	AB 34: GOSUB 500: IF E = 27				
	THEN PRINT " "				
	": GOTO 160	C106			
360	IF E = 11 THEN J = J - 1: GO				
	TO 345	3CC7			
365	IF E = 10 THEN J = J + 1: GO				
	TO 345	D3C5			
370	IF E = 13 THEN HTAB 30: PRI				
	NT "O/N";: GOSUB 500: IF E =				
	79 THEN GOSUB 550: VTAB 12:				
	HTAB 30: PRINT " "				
	": GOTO 160	DBFC			
375	HTAB 30: PRINT " ";; IF E				
	= 78 THEN HTAB 35: PRINT " "				
	": GOTO				
	160	1A68			
380	GOTO 345	4347			
385	PRINT D\$"BLOADFT.VIERGE,A\$40				
	00"	74A7			
390	GOSUB 565: GOTO 580	CCD2			
395	IF H = 1 THEN 405	1D53			
400	FOR D = 6 TO 14: VTAB D: POK				
	E 1403,55: PRINT "!";: POKE				
	1403,64: PRINT "!";: NEXT D:				
	VTAB 14: POKE 1403,56: FOR D				
	= 1 TO 8: PRINT "_";: NEXT :				
	VTAB 5: POKE 1403,56: FOR D				
	= 1 TO 8: PRINT "_";: NEXT :				
	GOTO 410	D759			
405	PRINT E\$: INVERSE : FOR D =				
	6 TO 13: VTAB D: HTAB 56: PR				
	INT "Z";: POKE 1403,64: PRIN				
	T "_";: NEXT D: VTAB 14: HTA				
	B 57: FOR D = 1 TO 8: PRINT				
	"L";: NEXT D: PRINT CHR\$(2				
	4);: NORMAL : VTAB 5: HTAB 5				
	7: FOR D = 1 TO 8: PRINT "_"				
	": NEXT D	3461			
410	K = 1: IF E = 13 THEN K = 0	BDBB			
415	FOR D = 56 TO 63: FOR F = 6				
	TO 13: IF K = 1 THEN VTAB F				
	: POKE 1403,D: PRINT C\$(F -				
	5,D - 55);: GOTO 425	54AE			
420	VTAB F: POKE 1403,D: PRINT "				
	":C\$(F - 5,D - 55) = "."	96CF			
425	NEXT F: NEXT D:M = 2	5851			
430	VTAB 6: POKE 1403,56:L = 6:C				
	= 56: GOSUB 570	78F4			
435	IF E = 32 THEN GOSUB 820	B335			
440	IF E = 8 THEN C = C - 1: IF				
	C < 56 THEN C = 63	77B4			
445	IF E = 21 THEN C = C + 1: IF				
	C > 63 THEN C = 56	14DC			
450	IF E = 11 THEN L = L - 1: IF				
	L < 6 THEN L = 13	56C8			
455	IF E = 10 THEN L = L + 1: IF				
	L > 13 THEN L = 6	38C4			
460	IF E = 27 THEN 470	148A			
465	VTAB L: POKE 1403,C: PRINT C				
	\$(L - 5,C - 55); CHR\$(8);:				
	GOSUB 570: GOTO 435	369D			
470	VTAB 14: POKE 1403,34: PRINT				
	"TYPE H ou h ";; GOSUB 540:				
	VTAB 14: POKE 1403,34: PRINT				
	" ";; IF E = 72 0				
	R E = 104 THEN VL = E: GOTO				
	480	CC9C			
475	GOTO 470	F046			
480	VTAB 14: POKE 1403,34: PRINT				
	"RANGER CAR? ";; GOSUB 540:				
	VTAB 14: POKE 1403,34: PRINT				
	" ";; IF E = 27 T				
	HEN I = 12: POKE 1403,35: PR				
	INT " ";; GOTO 60				
	5	F148			
485	IF E < 32 OR E > 126 THEN 47				
	0	0C02			
490	IF E > 31 AND E < 127 THEN				
	GOSUB 830: GOSUB 800: GOTO 4				
	10	B9AD			
495	GOTO 430: GOTO 160	2DBE			
500	E = PEEK (- 16384): IF E <				
	128 THEN PRINT "_";: FOR F				
	= 1 TO 50: NEXT F: PRINT CH				
	R\$(8);" ";; FOR F = 1 TO 50				
	: NEXT F: PRINT CHR\$(8);:				
	GOTO 500	12B6			
505	POKE - 16368,0:E = E - 128:				
	IF E < > 63 THEN RETURN	4888			
510	VTAB 6: POKE 1403,55: INVERS				
	E : PRINT " CATALOGUE DES FO				
	NTES";: NORMAL :N = 0	4573			
515	L = 8: FOR L = 8 TO 13:N = N				
	+ 1: IF N > 6 THEN VTAB L:				
	POKE 1403,55: PRINT " "				
	": GOTO 530	96E5			
520	IF L = 13 AND N < = 6 THEN				
	VTAB L: POKE 1403,55: INVERS				

```

E : PRINT " ""SUITE """;;
NORMAL : GOSUB 540:N = N - 1
: GOTO 530
525 VTAB L: POKE 1403,55: PRINT
G$(N); SPC( 15 - LEN (G$(N)
));
530 NEXT L: IF N < = G THEN 515
535 VTAB 13: POKE 1403,55: INVER
SE : PRINT " ""FIN ""
""";: NORMAL : GOSUB 500:
FOR L = 6 TO 13: VTAB L: POK
E 1403,55: PRINT " ""
""";: NEXT : POP :
GOTO 160
540 E = PEEK ( - 16384): IF E <
128 THEN PRINT "_";: FOR F
= 1 TO 50: NEXT F: PRINT CH
R$(8);" ";: FOR F = 1 TO 50
: NEXT F: PRINT CHR$(8);:
GOTO 540
545 POKE - 16368,0:E = E - 128:
RETURN
550 IF I$ = G$(J) THEN I$ = ""
555 IF H$ = G$(J) THEN H$ = ""
560 PRINT D$"DELETE FT."G$(J): F
OR O = J TO G - 1:G$(O) = G$
(O + 1): NEXT O:G = G - 1: V
TAB 3: HTAB 5: PRINT "<?" Ca
talogue des fontes """"nomb
re de fontes disponibles :""
";: INVERSE : PRINT " ";G;" "
";: NORMAL : PRINT " ";: RETU
RN
565 VTAB 3: HTAB 5: PRINT " ""C
réation "de "fontes """"nom
bre de fontes disponibles :""
";: INVERSE : PRINT " ";G;" "
";: NORMAL : PRINT " ";: RET
URN
570 E = PEEK ( - 16384): IF E <
128 THEN INVERSE : PRINT C$
(L - 5,C - 55);: NORMAL : FO
R P = 1 TO 50: NEXT P: PRINT
CHR$(8);C$(L - 5,C - 55);:
FOR P = 1 TO 50: NEXT P: PRI
NT CHR$(8);: GOTO 570
575 POKE - 16368,0:E = E - 128:
RETURN
580 ONERR GOTO 295:F$ = ""
585 IF I$ = "" THEN VTAB 8: HTA
B 6: PRINT "Charger ancienne
" "FONTE";: HTAB 30: PRINT "
""";: GOTO 6
00
590 VTAB 8: HTAB 6: PRINT "Charg
er ancienne "FONTE";: HTAB 3
5: INVERSE : PRINT " ";I$;"
";: NORMAL : PRINT " ""
""";
595 VTAB 6: HTAB 6: PRINT "Utili
ser "FONTE "vierge";
600 VTAB 12: HTAB 6: PRINT "Sauv
egarder ""la "FONTE";: VTAB
22: HTAB 16: PRINT "<ESC> Po
ur retour au Menu Précédent
""";: I
= 6
605 IF I > 12 THEN I = 6
610 IF I < 6 THEN I = 12
615 VTAB 12: HTAB 30: PRINT " ""
""";: VTAB
10: HTAB 30: PRINT " ""
""";
620 VTAB I: HTAB 30: PRINT "<=";
: GOSUB 540: IF E = 11 THEN
VTAB I: HTAB 30: PRINT " """;
:I = I - 2: GOTO 605
625 IF E = 10 THEN VTAB I: HTAB
30: PRINT " """;: I = I + 2: G
OTO 605
630 IF E = 13 THEN 645
635 IF E = 27 THEN FOR P = 5 TO
14: VTAB P: HTAB 30: PRINT "
""";: NEXT P:I = 14: G
OTO 155
640 GOTO 620
645 VTAB I: HTAB 30: PRINT " ""
";: ON (I / 2) - 2 GOTO 650,6
55,665,705
650 J$ = "": VTAB 10: HTAB 35: P
RINT " """;: GOS
UB 800: GOTO 395
655 IF I$ = "" THEN 605
660 J$ = "": PRINT D$"BLOAD FT."
I$",A$4000": GOSUB 800: GOTO
395
665 J = 1
670 IF J < 1 THEN J = G
675 IF J > G THEN J = 1
680 VTAB 10: HTAB 35: INVERSE :
PRINT " ";G$(J);" ";: NORMAL
: PRINT SPC( 14 - LEN (G$(
J)));: HTAB 34: GOSUB 540: I
F E = 11 THEN J = J - 1: GOT
O 670
685 IF E = 10 THEN J = J + 1: GO
TO 670

```

4 FONTE (suite)

```

690 IF E = 27 THEN PRINT "
-----";: GOTO 605 BDA8
695 IF E = 13 THEN J$ = G$(J): P
RINT D$"BLOAD FT."J$",A$4000
": VTAB 10: HTAB 30: PRINT "
---";: GOSUB 800: GOTO 395 6F6A
700 GOTO 670 FC48
705 N = 0:L$ = "": IF J$ = " TH
EN 715 ODD6
710 L$ = J$: GOSUB 840:N = LEN
(L$): VTAB 12: HTAB 30: PRIN
T "=";: POKE 1403,34: PRINT
L$;: GOTO 735 B301
715 VTAB 12: HTAB 30: PRINT "="
;: HTAB 35 495E
720 GOSUB 540: IF E = 27 THEN 60
5 8C0D
725 IF E < 65 OR E > 90 THEN 720 63D6
730 PRINT CHR$(E);:N = N + 1:L
$ = L$ + CHR$(E) 3740
735 GOSUB 540: IF E = 27 THEN V
TAB 12: HTAB 30: PRINT "
-----";: GOTO 605 111D
740 IF E = 13 THEN M$ = L$ + MI
D$(B$,1,12 - LEN(L$)): GO
SUB 780: IF Q = 1 THEN 605 817E
745 IF Q = 2 THEN PRINT D$;"BSA
VE FT.";L$;"A$4000,L955"::Q
= 0: GOTO 605 6E9D
750 IF E = 13 THEN PRINT D$;"BS
AVE FT.";L$;"A$4000,L955":G
= G + 1:G$(G) = L$ + MID$(
B$,1,12 - LEN(L$)):J$ = L$
: GOSUB 565: GOTO 605 6F40
755 IF E = 8 OR E = 127 THEN PR
INT CHR$(8);" "; CHR$(8);:
N = N - 1: IF N < 1 THEN L$
= "": GOTO 715 248B
760 IF E = 8 OR E = 127 THEN L$
= MID$(L$,1, LEN(L$) - 1):
GOTO 735 34D1
765 IF (E < 48 OR E > 57) AND (E
< 65 OR E > 90) AND (E <
46) THEN 735 3E88
770 IF N > = 12 THEN N = 12: GO
TO 735 B6C6
775 PRINT CHR$(E);:N = N + 1:L
$ = L$ + CHR$(E): GOTO 735 F9C4
780 Q = 0: FOR R = 1 TO G: IF G$(
R) = M$ THEN Q = 1 54B3
785 NEXT R: IF Q = 0 THEN RETUR
N 2581

```

```

790 VTAB 17: HTAB 6: PRINT " La
fonte ";: INVERSE : PRINT "
";M$ " ";: NORMAL : PRINT " e
xiste déjà /Retour pour acc
epter ";: GOSUB 540: IF E =
13 THEN Q = 2 7843
795 VTAB 17: HTAB 6: PRINT SPC(
60);: RETURN 427C
800 VTAB 14: POKE 1403,34: PRINT
"APPEL "CAR? ";: GOSUB 540:
IF E = 27 THEN POKE 1403,34
: PRINT "
-----";:I
= 12: GOTO 605 E7F8
805 IF E = 13 THEN RETURN 799B
810 IF E < 32 OR E > 126 THEN 80
0 DEFF
815 PRINT CHR$(E);: FOR C = 1
TO 8:A = PEEK (16389 + (E -
32) * 10 + C): FOR L = 1 TO
8:S = 128 / 2 ^ (L - 1):C$(9
- L,C) = CHR$( - INT (A /
S) * 4 + 46):A = A - INT (A
/ S) * S: NEXT L: NEXT C: RE
TURN EA28
820 IF C$(L - 5,C - 55) = "*" TH
EN C$(L - 5,C - 55) = ".": R
ETURN 7224
825 IF C$(L - 5,C - 55) = "." TH
EN C$(L - 5,C - 55) = "*": R
ETURN 3224
830 POKE 16388 + (E - 32) * 10,E
: POKE 16389 + (E - 32) * 10
,VL: FOR C = 1 TO 8:B = 0: F
OR L = 1 TO 8: IF C$(L,C) =
"*" THEN B = B + 2 ^ (L - 1) 74F0
835 NEXT L: POKE 16389 + (E - 32
) * 10 + C,B: NEXT C: RETURN 18E6
840 A = 1 5342
845 IF A > LEN(L$) THEN 855 0BC7
850 IF MID$(L$,A,1) < > " " T
HEN A = A + 1: GOTO 845 73BB
855 L$ = MID$(L$,1,A - 1): RET
URN 129A

```

5 FONTE.CODE BSAVE FONTE.CODE A\$300,L\$33

```

*300.332
0300- A9 FF 85 06 A9 3F 85 07
0308- AD C8 BE 85 08 AD C9 BE
0310- 85 09 E6 06 A5 06 D0 02
0318- E6 07 C6 08 A5 08 C9 FF
0320- D0 06 A5 09 F0 0C C6 09
0328- A0 00 B1 06 20 ED FD 4C
0330- 12 03 60

```

CLASSEMENT

(alphabétique, mais suivant la longueur)

Ce micro-programme ne prétend pas être un modèle du genre. Il vous montre seulement comment trier une liste dans l'ordre alphabétique, mais en priorité suivant la longueur des mots. Vous noterez que le tri alphabétique est effectué en plusieurs fois, précisément par groupe de mots de n lettres.

```

100 TEXT : HOME : D$ = CHR$ (4)      7938
105 S = 20 : VTAB 23 : PRINT "TAPE
    R DES MOTS (RIEN POUR ARRETE
    R)"; : VTAB 1 : PRINT : VTAB 1  B05C
110 DIM M$(S + 1)                   8694
115 FOR N = 1 TO S : PRINT N " "; :
    INPUT "-> "; M$(N) : IF M$(N)
    = "" THEN 125                    B004
120 NEXT N                           9BD0
125 K = 0 : N = N - 1 : L1 = 30      F9D5
130 FOR I = 1 TO N : L = LEN (M$(
    I)) : IF L > L2 THEN L2 = L      E54D
135 IF L < L1 THEN L1 = L           BDA4
140 NEXT                             0582
145 FOR I = L1 TO L2                5056
150 K1 = K                           8697
155 FOR J = K + 1 TO N : IF LEN
    (M$(J)) < > I THEN 175           4D5F
156 IF J = K + 1 THEN 171           8C68
160 M$ = M$(J)                      3B4D
165 FOR X = J TO K + 2 STEP - 1
    : M$(X) = M$(X - 1) : NEXT X     5C06
170 M$(K + 1) = M$                  AE47
171 K = K + 1                       215F
175 NEXT J                          A3CC
180 IF K1 = K THEN 210              3C9B
185 IF K = K1 + 1 THEN 210          1A94
190 F = 0 : FOR Z = K1 + 1 TO K -
    1                                 F0A6
195 IF M$(Z) < = M$(Z + 1) THEN
    205                              49DA

```

PRINCIPALES VARIABLES

Dans l'ordre où elles apparaissent :

- S** : Nombre maximum de mots de la liste.
- M\$(** : Contiennent les mots.
- K** : Pointeur utilisé dans le classement suivant la longueur.
- L1** : Longueur du mot le plus court.
- L** : Longueur d'un mot en cours de tri.
- L2** : Longueur du mot le plus long.
- K1** : Pointeur-mémoire utilisé avec K.
- F** : Pointeur utilisé au cours du classement alphabétique.

```

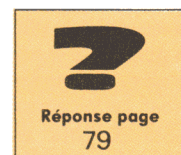
200 M$ = M$(Z) : M$(Z) = M$(Z + 1)
    : M$(Z + 1) = M$ : F = 1        0EA9
205 NEXT Z : IF F THEN 190          C067
210 NEXT I                          9DCB
215 FOR I = 1 TO N : VTAB I : HTAB
    L2 + 8 : PRINT M$(I) : NEXT      4308
220 VTAB 22 : PRINT : PRINT "(E)N
    CORE "(A)PPLESOFT "(M)ENU DI
    SK " ; : GET R$                 0A58
225 IF R$ = "M" OR R$ = "m" THEN
    PRINT D$"RUN /TM17/STARTUP"     6D02
230 IF R$ = "E" OR R$ = "e" THEN
    RUN                             5A89
235 IF R$ = "A" OR R$ = "a" THEN
    HOME : END                      5E26
240 GOTO 220                       273F

```

QUESTION

LUC n'est pas à sa place.
Qu'est-ce qui a pu se passer ?

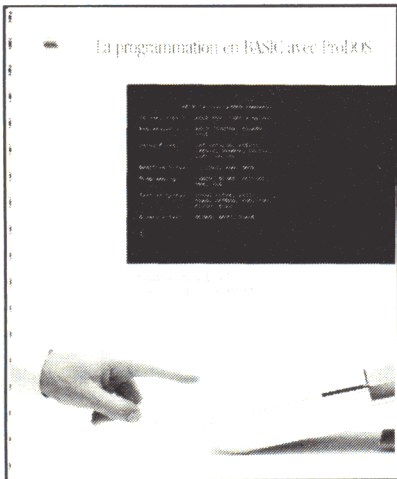
1. ADÈLE
2. BENOIT
3. GUY
4. RAOUL
5. LUC
6. ANNIE



1. GUY
2. ADÈLE
3. ANNIE
4. LUC
5. RAOUL
6. BENOIT

Votre bibliothèque INFORMATIQUE

par NESTOR



• LA PROGRAMMATION EN BASIC AVEC ProDOS (Apple)

Qui peut se vanter de bien connaître ProDOS ? Vous... alors, bravo ! Parcourez tout de même les 265 pages de ce manuel "officiel" et je parie que vous y apprendrez encore quelques détails ignorés. Je vous signale au passage qu'il est accompagné de sa disquette (ProDOS 1.1.1), laquelle comporte notamment le programme de renumérotage (et autres commodités), intitulé "APA".

INTER EDITIONS, 87 rue du Maine, 75014 PARIS
Tél. : 43.27.74.50

• INTRODUCTION AU BASIC (par Pierre LE BEUX)

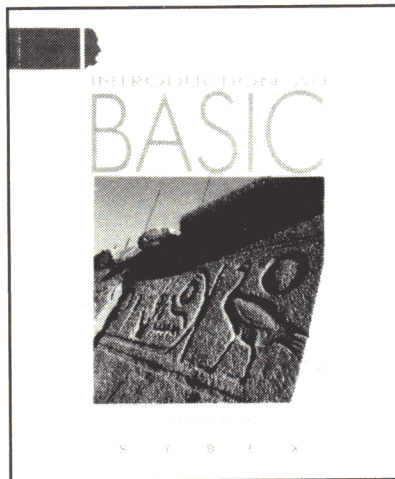
Si le MS BASIC — beaucoup plus structuré que notre bon vieil AppleSoft — n'a plus de secrets pour vous, c'est sans doute parce que vous chatouillez parfois le clavier d'un IBM PC ou de quelque clone (pas drôle du tout !).

Pierre LE BEUX vous propose, dans cette nouvelle édition, non pas un survol rapide du MS BASIC, mais bien une étude complète et méthodique de ce langage.

Je ne vous conseille pas d'escamoter le chapitre "introduction" et pas davantage le suivant, consacré aux "généralités du langage Basic". Ils vous procureront certainement une meilleure connaissance du Basic en général et du MS BASIC en particulier.

J'ai beaucoup apprécié la présentation claire et concise de ce gros manuel (500 pages), mais plus encore les nombreux exemples dont il est émaillé (l'auteur sait sortir des sentiers battus !): tris, carrés magiques, problème des huit reines, etc. Pierre LE BEUX, ingénieur de l'Ecole Centrale, docteur en informatique de l'Université de Californie est aussi professeur à l'Université de technologie de Compiègne. On se prend à envier ses élèves !

SYBEX, 6-8 impasse du Curé, 75018 PARIS, Tél. : 42.03.95.95



• TECHNIQUES DE PROGRAMMATION EN C LA GESTION DES DONNÉES (par R. A. RADCLIFFE et T. J. RAAB)

Cet ouvrage (traduit de l'américain par Yann Festal) se présente sous la forme d'un manuel de haut niveau destiné à la fois aux programmeurs expérimentés et aux développeurs. Il aborde deux aspects fondamentaux des logiciels d'application: l'interactivité et le besoin permanent de stocker des données. De nombreux exemples illustrent les fonctions d'entrée, de validation, d'affichage et de stockage.

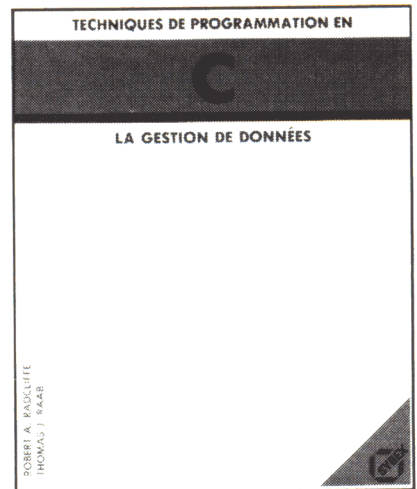
La première partie, consacrée au concept de structure de données, regroupe les techniques d'encodage, de décodage et de validation. La deuxième partie traite des différents types de données et comprend un chapitre entier sur le développement de types de données personnalisés. La troisième partie présente des exemples d'application correspondants aux parties 1 et 2.

Les auteurs :

- Robert A. Radcliffe est consultant en systèmes d'information. Il conçoit et met en place des logiciels informatiques ainsi que des applications pour micro-ordinateurs.
- Thomas J. Raab se spécialise dans la recherche micro-informatique et les technologies de pointe.

676 pages brochées.

SYBEX, 6-8 impasse du Curé 75018 PARIS.



• AMSTRAD PC1512, GUIDE DU GRAPHISME (par G. FAGOT-BARRALY)

Ce guide permet de tirer le meilleur parti des possibilités graphiques de l'Amstrad PC1512. Il s'adresse à la fois aux débutants et aux programmeurs confirmés. Il décrit, par un grand nombre de programmes, de dessins et de photographies d'écrans, toutes les instructions qui concernent le graphisme. Plusieurs chapitres traitent, entre autres :

Du multifenêtrage, caractéristique de l'ordinateur — des notions relatives au tracé des figures géométriques — de l'affichage alphanumérique — des commandes de dessin — du graphisme de type LOGO.

Le livre (305 pages au format 160 x 220) se termine par l'analyse des méthodes qui permettent de déplacer ou d'agrandir un motif ou encore de lui faire subir symétries et rotations.

L'auteur : Georges Fagot-Barraly, professeur de mathématiques et d'informatique, est spécialiste de l'utilisation des micro-ordinateurs dans l'enseignement. Il est l'auteur de nombreuses publications sur la programmation en BASIC et en Assembleur.

SYBEX, 6-8 impasse du Curé, 75018 PARIS, Tél. : 42.03.95.95

PLACE-DOS

Configuration requise : tout Apple II avec une RAM de 64 Ko ou plus, le DOS 3.3 Slave, et un lecteur de disquettes 5 pouces 1/4.

PLACE DOS est un programme écrit en Basic Applesoft, permettant de sauvegarder (Write) sur les pistes \$00-\$02 d'une disquette déjà initialisée, un système d'exploitation placé en mémoire centrale et patché par le programme PATCH DOS. Son fonctionnement convient tout aussi bien pour un DOS Slave autre que le DOS 3.3. Lorsque le programme est exécuté, il appelle un sous-programme en langage machine : RWTS.OBJ. Par la suite, HIMEM est déclaré à 16384. Le programme, d'une structure des plus simples, exploite certaines astuces liées au DOS 3.3. Un menu clair et précis renseigne l'utilisateur sur la poursuite éventuelle de l'exécution en cours, permettant un suivi des principales phases du programme. La touche ESC (Escapade) autorise un retour au menu précédent, ou l'abandon pur et simple du programme avec l'option du numéro du slot. ■

A PROPOS DU PROGRAMME

Les variables SL et DR mémorisent respectivement les numéros du slot et du drive où se trouve la disquette à traiter. Le numéro de la disquette actuelle est lu, puis mémorisé pour une application future.

Le nom du programme de boot, sauvegardé sur la piste \$01 secteur \$09 de la disquette, sera déclaré, puis formaté dans une chaîne de caractères limitée à 30 codes ASCII. Cette contrainte est dictée par le DOS 3.3 qui n'accepte qu'un maximum de 30 caractères pour un nom de fichier.

L'acceptation du nom du boot suppose certaines contraintes :

- Un maximum de 30 caractères est autorisé.
- Le premier caractère du nom sera un caractère alphabétique.
- Après le premier caractère alphabétique, tout caractère sera toléré.
- Le bit 7 sera mis à 1 (ligne 630).
- Les touches de déplacement sont tolérées.
- Les minuscules sont interdites.

PHASES ESSENTIELLES DU PROGRAMME

- | | |
|----------------|--|
| Lignes 170-200 | Affichage du menu à l'écran. |
| Lignes 210-250 | Mémorisation du numéro du slot. |
| Lignes 260-310 | Mémorisation du numéro du drive. |
| Ligne 350 | Annule une ancienne chaîne du même nom. |
| Lignes 360-480 | Composition de la chaîne de caractères avec le nom du boot. |
| Ligne 500 | Formate la chaîne de caractères avec les variables tableau. |
| Lignes 510-590 | On détermine le type de fichier booté : A = AppleSoft ; B = Binaire ; E = EXEC. |
| Lignes 610-630 | On poke le nom du programme de boot dans le tampon mémoire réservé au DOS. |
| Lignes 640-750 | On sauvegarde le DOS implanté en mémoire centrale sur les trois premières pistes de la disquette et dans un ordre déterminé. |
| Lignes 770-800 | Gestion d'une erreur éventuelle. |
| Lignes 810-820 | On quitte le programme. <i>(suite page 22)</i> |

**PLACE LE DOS
PAR MARCEL COTTINI
COPYRIGHT 86**

```

10 REM
20 REM
30 REM
40 REM
50 REM
60 REM

70 PRINT CHR$(4);"BRUN RWTS.OBJ"
80 HIMEM: 16384
90 DIM NO$(32)
100 ONERR GOTO 800
110 REM SLOT ET DRIVE

120 TEXT : HOME : NORMAL
130 SL = PEEK (43626): REM SLOT
140 DR = PEEK (43624): REM DRIVE
150 VOL = PEEK (46017): REM NUMERO DE
    LA DISQUETTE
160 REM MENU

170 TEXT : HOME : INVERSE : PRINT " PLACE LE
    DOS MODIFIE": NORMAL : PRINT : PRINT
    "SUR LES PISTES $00-$02 DE LA DISQUETTE"
180 PRINT
190 FOR X = 1 TO 40: PRINT "*":: NEXT X:
    PRINT
200 VTAB 8: HTAB 1: PRINT " - LA DISQUETTE SE
    TROUVE:"
210 REM SLOT NUMBER

220 VTAB 9: HTAB 1: PRINT "      - SLOT : ";;
    GET SL$
230 IF SL$ = CHR$(27) THEN GOTO 810
240 IF SL$ < "3" OR SL$ > "7" THEN 220
250 PRINT SL$
260 REM DRIVE NUMBER

270 VTAB 10: HTAB 1: PRINT "      - DRIVE: "
    ;; GET DR$
280 IF DR$ = CHR$(27) THEN VTAB 9: CALL -
    868: GOTO 200
290 IF DR$ < "1" OR DR$ > "2" THEN 270
300 PRINT DR$
310 SL = VAL (SL$):DR = VAL (DR$): REM
    REVECTORISE SLOT & DRIVE
320 REM BOOT NAME

330 VTAB 8: HTAB 1: CALL - 958: PRINT
    "BOOT: ";
340 REM ANNULE LA CHAINE

350 FOR NUL = 0 TO 30:NO$(NUL) = "":NO$
    = NO$(NUL): NEXT NUL: REM ANNULE
    L'ANCIENNE CHAINE
360 REM HELLO NAME

370 FOR HEL = 0 TO 30
380 VTAB 8: HTAB 7 + HEL: GET NO$(HEL)
390 IF HEL <> 0 AND NO$(HEL) = CHR$(13)

```

```

THEN NO$(HEL) = "": GOTO 500: REM
TESTE LA TOUCHE RETURN
400 IF NO$(0) = CHR$(27) THEN VTAB 8: HTAB
    1: CALL - 958: GOTO 270: REM TESTE LA
    TOUCHE ESCAPE
410 IF NO$(HEL) = CHR$(27) THEN VTAB 8:
    HTAB 7: CALL - 868: GOTO 340: REM
    TOUCHE ESC (RETOUR AU MENU)
420 IF HEL = 30 THEN HEL = 29: VTAB 8: HTAB
    7 + HEL: CALL - 868: GOSUB 790: GOTO
    380: REM TESTE LES 30 CARACTERES
    ADMIS
430 IF NO$(HEL) = CHR$(8) AND HEL > 0
    THEN NO$(HEL) = "":HEL = HEL - 1:
    VTAB 8: HTAB 7 + HEL: CALL - 868:
    GOTO 380: REM TESTE LA TOUCHE FLE
    CHE GAUCHE
440 IF HEL < 0 THEN HEL = 0: GOSUB 790:
    NO$(0) = "": VTAB 8: HTAB 7: CALL -868:
    GOTO 340: REM PREMIER CARACTERE ?
450 IF NO$(0) < CHR$(65) OR NO$(0) > CHR$(
    90) THEN GOSUB 790: GOTO 380: REM
    PREMIER CARACTERE ALPHABETIQUE ?
460 IF NO$(HEL) < CHR$(32) OR NO$(HEL) >
    CHR$(95) THEN GOSUB 790: GOTO 380:
    REM CARACTERES SUIVANTS ?
470 VTAB 8: HTAB 7 + HEL: PRINT NO$(HEL)::
    REM AFFICHE LE CARACTERE TAPE
480 NEXT HEL
490 REM FORMATE LA CHAINE

500 FOR CAR = 0 TO 29:NO$ = NO$ +
    NO$(CAR): NEXT CAR
510 REM FILE TYPE

520 VTAB 8: HTAB 1: CALL - 958: PRINT
    "(A)PPLESOFT          (B)INAIRE
    OU          (E)XEC —» ";
530 GET PK$
540 IF PK$ = CHR$(27) THEN GOTO 330
550 IF PK$ <> "A" AND PK$ <> "B" AND
    PK$ <> "E" THEN GOTO 530
560 VAR = 40514: REM TYPE DE FICHER
570 IF PK$ = "A" THEN PK$ = "APPLESOFT":
    POKE VAR,6: GOTO 610
580 IF PK$ = "B" THEN PK$ = "BINAIRE":
    POKE VAR,52: GOTO 610
590 IF PK$ = "E" THEN PK$ = "EXEC": POKE
    VAR,20: GOTO 610
600 REM POKE HELLO NAME

610 VTAB 8: HTAB 1: CALL - 958: FLASH : PRINT
    " WRITE DOS - - - -": NORMAL: PRINT :
    PRINT "BOOT NAME: ";NO$: PRINT
    "FICHER : ";PK$: PRINT "VOLUME
    :
    ";VOL
620 FOR NO = 1 TO 31: POKE 43636 +
    NO,160: NEXT NO: REM ANNULE HELLO

```

```

630 FOR I = 1 TO LEN (NO$): POKE 43636 + I,
    ASC ( MID$ (NO$,I)) + 128: NEXT I: REM
    POKE LE NOM NO$
640 REM WRITE DOS

650 & DISK(SL,DR,000)
660 BUFF = 46592
670 POKE 43615,0: REM ANNULE LA COM
    MANDE
680 POKE 47083,VOL: REM POKE LE NUMERO
    DU VOLUME ACTUEL
690 FOR I = 0 TO 9: & WSEC(00,I,BUFF): BUFF
    = BUFF + 256: NEXT I
700 BUFF = 39680
710 FOR I = 10 TO 15: & WSEC(00,I,BUFF):
    BUFF = BUFF + 256: NEXT I
720 BUFF = 41216
730 FOR I = 1 TO 1: & WTRK(I, BUFF): NEXT I

```

```

740 BUFF = 45312
750 FOR I = 0 TO 2: & WSEC(02,I,BUFF):BUFF
    = BUFF + 256: NEXT I
760 GOTO 120
770 REM GESTION ERREUR

780 HOME : VTAB 12: & BELL: & BELL: PRINT
    "I/O ERROR - LECT/ECRITURE IMPOSSI
    BLE.": PRINT "TAPEZ UNE TOUCHE POUR
    CONTINUER....": GET R$: RETURN
790 FOR BIP = 1 TO 10: & BEL1: NEXT BIP:
    RETURN
800 GOSUB 770: GOTO 170
810 REM QUITTER LE PROGRAMME

820 HOME : PRINT : PRINT "GOOD BYE.-": &
    BELL: END

```

Tapez ces codes
suivant le
processus
habituel et
sauvez votre
programme par
un **BSAVE**
RWTS.OBJ,
A\$9000,L\$01FA

Si vous possédez
notre disquette
SIGNATURE,
contrôlez
l'exactitude de la
saisie.

RWTS.OBJ,A\$9000,L\$01FA

COMPLÉMENT DE PLACE.DOS

9000:	A9 17 8D F6 03 A9 90 8D F7 03 AD E9 B7 8D D6 90	AB46
9010:	AD EA B7 8D D7 90 60 A2 00 86 19 A0 00 BD 4C 90	B21C
9020:	F0 1A C9 FF F0 08 D1 B8 D0 07 C8 E8 D0 EF 4C C9	95AE
9030:	DE E8 BD 4C 90 D0 FA E8 E6 19 D0 DF 20 98 D9 06	CB56
9040:	19 A6 19 BD 71 90 48 BD 70 90 48 60 44 49 53 48	486E
9050:	00 52 53 45 43 00 57 53 45 43 00 52 54 52 48 00	66A2
9060:	57 54 52 48 00 42 45 4C 4C 00 42 45 4C 31 00 FF	1B6A
9070:	F3 90 73 91 85 91 A8 91 AF 91 C3 91 DE 91 00 A9	9C82
9080:	00 85 3C 20 B7 00 C9 22 D0 2A E6 B8 D0 02 E6 B9	A18C
9090:	A5 B8 85 42 A5 B9 85 43 A0 00 B1 B8 C9 22 F0 03	C131
90A0:	C8 D0 F7 84 3C 18 98 65 B8 85 B8 A9 00 65 B9 85	B0A5
90B0:	B9 4C B1 00 20 E3 DF A5 81 30 04 A5 82 30 03 4C	0298
90C0:	76 DD A0 00 B1 83 85 3C C8 B1 83 48 C8 B1 83 85	4AAD
90D0:	43 68 85 42 60 01 60 01 00 00 00 E6 90 00 00 00	75AA
90E0:	00 00 00 00 60 01 00 01 EF D8 A0 D5 A9 90 4C D9	31FC
90F0:	03 00 00 00 20 BB DE 20 67 DD 20 FB E6 E0 08 90	E099
9100:	03 4C 99 E1 8E F1 90 8A 0A 0A 0A 0A 8D D6 90 20	D99D
9110:	BE DE 20 67 DD 20 FB E6 E0 03 B0 E5 8E F2 90 8E	0F17
9120:	D7 90 A9 00 8D F3 90 8D D8 90 20 B7 00 C9 2C F0	ABD1
9130:	03 4C B8 DE 20 BE DE 20 67 DD 20 FB E6 8E F3 90	6F17
9140:	8E D8 90 4C B8 DE 20 BB DE 20 67 DD 20 FB E6 8E	D884
9150:	D9 90 20 BE DE 20 67 DD 20 FB E6 8E DA 90 20 BE	EE60
9160:	DE 20 67 DD 20 52 E7 A5 50 8D DD 90 A5 51 8D DE	E0EB
9170:	90 4C B8 DE A9 01 8D E1 90 20 46 91 20 EA 90 90	993B
9180:	04 4C 12 D4 00 60 A9 02 8D E1 90 D0 EC A9 0F 8D	6840
9190:	DA 90 18 AD DE 90 69 0F 8D DE 90 20 EA 90 B0 E1	4C3B
91A0:	CE DE 90 CE DA 90 10 F3 60 A9 01 8D E1 90 D0 05	F154
91B0:	A9 02 8D E1 90 20 BB DE 20 55 91 AD DA 90 8D D9	ABE5
91C0:	90 4C 8D 91 A2 FF A9 5B A0 1B 88 D0 FD 2C 30 C0	73CB
91D0:	A8 88 D0 FD E9 01 F0 EE 2C 30 C0 CA D0 EA 60 A2	F167
91E0:	50 A9 22 A0 02 88 D0 FD 2C 30 C0 A8 88 D0 FD E9	5414
91F0:	01 F0 EE 2C 30 C0 CA D0 EA 60	8FDF

RWTS.
OBJ

HV

PAVÉS ALÉATOIRES

\$E6F5 transmet la valeur de H, puis celle de V.
\$FBC1 calcule l'adresse de base de la ligne V.
\$FC2F additionne H à cette adresse.
 Le pavé (mode inverse) est stocké.

Pomme ouverte (Basic) arrête tout.

0300-	20 F5 E6	JSR	\$E6F5
0303-	8A	TXA	
0304-	20 C1 FB	JSR	\$FBC1
0307-	20 F5 E6	JSR	\$E6F5
030A-	8A	TXA	
030B-	20 2F FC	JSR	\$FC2F
030E-	A9 20	LDA	£\$20
0310-	91 28	STA	(£28),Y
0312-	60	RTS	

Sur APPLE IIc exclusivement
(lire p. 79 pour Apple IIe)

10	TEXT : NORMAL : PRINT CHR\$(21) : HOME	9BC0
15	FOR I = 768 TO 786: READ R: POKE I,R: NEXT	AD2E
20	V = INT (RND (1) * 23)	AED6
25	H = INT (RND (1) * 39)	9FCF
30	CALL 768,V,H	9727
35	IF PEEK (49249) < 128 THEN 20	1D7E
40	VTAB 23: CALL - 868: PRINT "(M)ENU DE DISQUETTE ";: GET R\$: VTAB 22	A484
45	IF R\$ = "M" OR R\$ = "m" THEN PRINT CHR\$(4)"RUN STARTUP"	549F
50	HOME	2F97
55	DATA 32,245,230,138,32,193,251,32,245,230,138,32,47,252,169,32,145,40,96	1A72

QUESTION

Monsieur KOENIG, Abonné à Tremplin Micro, je me permets de vous écrire pour une question que j'ai déjà posée au Support Technique Apple (je suis membre du club Apple). La voici :

Lors du Boot de ProDOS*, une routine teste tous les slots (de 1 à 7) et reconnaît donc certaines cartes (Thunderclock, driver, souris (II^e), Super série...). Pour ce faire, ProDOS doit sans doute avoir une table de références. A quelle adresse se trouve-t-elle ?

Le Support Technique Apple m'a simplement conseillé de consulter ma documentation, car il se dit "support technique et non de développement" !

Je dois vous avouer que cette réponse me reste coincée dans la gorge !

Espérant que vous puissiez répondre à ma question, agréer monsieur, mes sincères salutations.

C. G. (LE BLANC MESNIL).

* ProDOS 1.0.2/1.1.1 utilisés.

RÉPONSE

reconnaissent la Thunderclock et 1 horloge //GS).

Puisque vous disposez de PRODOS 1.1.1, je vous suggère de faire BLOAD PRODOS.TSYS,AS\$2000, CALL - 151 puis 252A LLLL... jusqu'à 2674. C'est la zone d'identification des périphériques. Ce n'est pas trop difficile à interpréter. Je traduis ci-après une page de la DOC ProDOS APPLE. Durant sa mise en place, ProDOS recherche les dispositifs de stockage/lecture de blocs. Pour le connecteur n, ProDOS considère qu'il est en présence d'un "lecteur de disque" si la ROM associée à ce connecteur contient les identificateurs suivants :

\$Cn01 = \$20 \$Cn03 = \$00 \$Cn05 = \$03 l'octet Cn07 est présent dans la table de ProDOS, mais il n'est pas testé.

ProDOS contrôle ensuite le contenu de l'adresse \$CnFF :

- Si CnFF = \$00, ProDOS considère qu'il a trouvé un lecteur de type DISK II équipé des ROMs 16 secteurs et stocke, dans la table de "devices" intégrée s'il s'agit de ProDOS16, dans la table de "devices" en page globale s'il s'agit de ProDOS8, l'adresse du "Driver de DISK II". Ces routines de gestion de DISK II sont intégrées à ProDOS et supportent tout lecteur qui émule un lecteur DISK II APPLE (16 secteurs). Pour mémoire, cela suppose, entre autres, 280 blocs et un seul volume.
- Si CnFF = \$FF, ProDOS considère qu'il est en présence d'un DISK II équipé des ROMs 13 secteurs. Comme il ne sait pas manipuler ce format, il quitte ce connecteur et poursuit ses recherches.
- Si CnFF contient une valeur différente de \$00 ou \$FF, ProDOS considère qu'il a trouvé un "contrôleur de disque intelligent". Si en outre l'octet d'état en \$CnFE indique que le périphérique supporte les appels READ et STATUS, ProDOS range dans la table de devices (P8 ou P16) l'adresse du driver correspondant, à savoir : \$Cn pour l'octet haut et le contenu de \$CnFF pour l'octet bas. En d'autres termes, \$CnFF contient l'octet bas du driver correspondant au connecteur n°.....

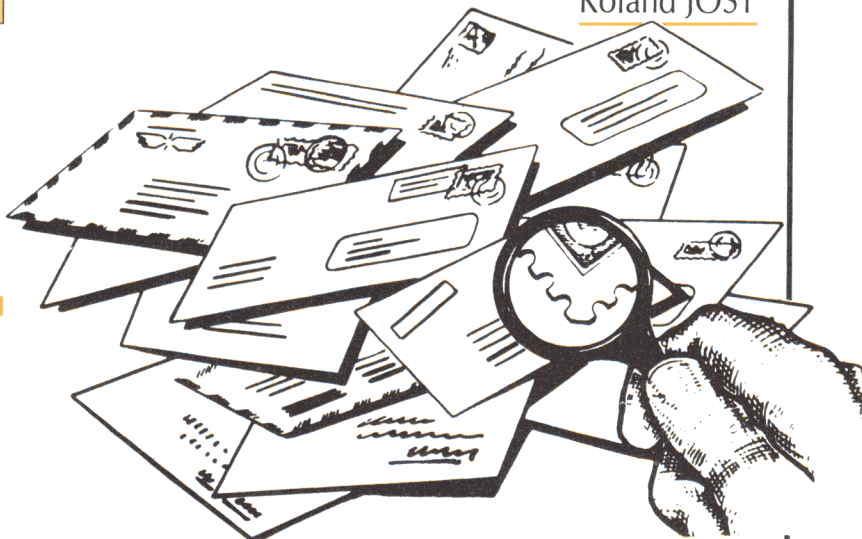
Les adresses spéciales de la ROM sont :

- \$CnFC/FD Nombre total de blocs sur ce périphérique. Information utile pour créer la table d'occupation du volume et l'en-tête du CATALOGUE principal après le formatage. Si la valeur stockée à cette adresse est nulle, le nombre de blocs doit être obtenu à l'aide d'un appel à la routine STATUS pour CE périphérique.
 - \$CnFE C'est l'octet "status" de ce périphérique. Les bit 0 et 1 doivent être à 1 pour que ProDOS installe le vecteur relatif à ce périphérique dans sa table de "devices".
- | | | | |
|--------|--|---|---|
| Bit n° | Signification du bit mis à 1 | 3 | périphérique supportant le Formatage. |
| 7 | support amovible | 2 | on peut écrire sur ce périphérique. |
| 6 | périphérique interruptible | 1 | on peut lire sur ce périphérique. |
| 5-4 | nombre de volumes de ce "device" (0 à 3) | 0 | on peut appeler STATUS sur ce périphérique. |
- \$CnFF Octet bas du point d'entrée du driver relatif à ce connecteur, que ProDOS placera dans sa table de "devices".

Je compléterai en signalant que la carte contrôleur de lecteurs 3"5 est un périphérique supportant 4 volumes tout comme le "SMARTPORT" du IIc ou du IIgs.

Yvan KOENIG.

LOUPE



Les lecteurs de *Tremplin Micro* sont certainement des familiers de la télématique en général, et du MINITEL en particulier. Ce dernier possède une touche nommée LOUPE qui permet un agrandissement de l'écran. Avec LOUPE, votre APPLE vaudra bien un MINITEL et même plus, puisque le déroulement d'écran et la copie sur imprimante vous sont offerts en sus.

Démonstration :

Si vous disposez de l'assembleur EDASM (Tool-kit) d'APPLE en version DOS ou ProDOS, entrez le source tel quel et compilez-le sous le nom de LOUPE.A. Si vous disposez d'un autre assembleur pour APPLE, tapez le source en faisant les modifications nécessaires. Sinon, passez en mode Moniteur par CALL - 151 et entrez le programme en hexadécimal :

8A00 : 20 FA 8A AD 00 C0 etc....

Sauvez sur une disquette par BSAVE LOUPE.A, A\$8A00,L\$202.

Pour accélérer le tracé HGR, LOUPE.A utilise des tables d'adresses. Pour éviter la saisie de ces tables, employez le programme CREATABHGR.

Création de LOUPE.0 :

BLOAD LOUPE.A

RUN CREATABHGR

BSAVE LOUPE.0,A\$8A00,L\$528

Tapez maintenant le programme BASIC DEMO.LOUPE et sauvez-le.

Si vous possédez la disquette IMAGEWRITER TOOLKIT (DOS 3.3), transférez le programme en langage machine GF de cette disquette sur celle qui contient LOUPE.0 et DEMO.LOUPE. Pour cela, faites : BLOAD GF puis, sur votre disquette, BSAVE GF,A\$9000,L\$200.

Ayez aussi sur la disquette une image graphique haute résolution de votre choix, puis faites RUN DEMO.LOUPE.

Dès que l'image HGR est affichée, vous pouvez utiliser les commandes suivantes :

- L pour passer en mode LOUPE.
- S pour voir l'autre moitié de l'image.
- → pour un déroulement d'image vers le haut.
- ← pour un déroulement d'image vers le bas.
- (RETURN) (touche ENTER) pour revenir à l'image normale.
- (ESC) pour retour en mode TEXT.

Si GF est présent sur la disquette, il vous est proposé une copie écran sur IMAGEWRITER. Répondez (O)ui pour obtenir une image 21 * 27.

A quoi peut servir LOUPE ?

Bien évidemment la majorité des lecteurs de *Tremplin Micro* peut se passer de LOUPE pour déchiffrer l'écran APPLE. Il existe heureusement d'autres utilisations :

1. Donner un look neuf à des images graphiques (portraits par exemple) en effectuant une dilatation dans le sens de la hauteur.
2. Ecrire des affiches 21 * 27 si vous disposez d'un utilitaire d'écriture de texte en page graphique. Dans ce domaine, EDIGRAPH (paru dans Pom's) me rend de grands services

(suite page 26)

3. Employer des sous-programmes de LOUPE pour vos propres applications :

- a. **sp. LOUPE** : programme de dilatation verticale d'une image HGR stockée en page 1. Les 91 premières lignes dilatées sont stockées en page HGR 2 (\$4000-\$5FFF), la deuxième moitié de l'image dilatée se trouve en page graphique 3 (\$6000-\$7FFF).
Appel par CALL 35433.
- b. **sp. SWAP** : ce programme échange le contenu de deux pages graphiques, en l'occurrence ici la page 2 (\$4000-\$5FFF) et la page 3 (\$6000-\$7FFF).
Appel par CALL 35551.
- c. **sp. SCROLLUP** : réalise un déroulement de la page 2 vers le haut, la page 3 alimente le bas de la page 2. Ce qui disparaît de la page 2 est réécrit en fin de page 3. On peut réaliser ainsi un déroulement sans fin. Appel par CALL 35328 (\$8A00) suivi de CALL 35614 (\$8B1E) pour dérouler l'écran d'une ligne vers le haut.

Exemple de programme de déroulement continu :

BLOAD IMAGE1,A\$4000 : charge 1^{ère} image en page 2

BLOAD IMAGE2,A\$6000 : charge 2^e image en page 3

400 CALL 35328 : REM initialise le mode déroulement

410 CALL 35614 : GOTO 410

- d. **sp. SCROLLDOWN** : action identique à SCROLLUP mais vers le bas. Appel par CALL 35328 suivi de CALL 35726.
4. Enfin le dernier intérêt de LOUPE, et non le moindre, est didactique. En effet, la gestion des pages graphiques de l'APPLE II n'est pas évidente a priori. L'étude du programme source permettra aux lecteurs de se familiariser avec la manipulation des octets en page graphique.

Quelques explications :

Une page graphique d'APPLE II comporte 192 lignes de 40 octets. Les 7 premiers bits d'un octet représentent chacun un point de l'écran (allumé si bit=1 ou éteint si bit=0). On retrouve ainsi les 280 points disponibles horizontalement en haute résolution.

Les octets de la page graphique ne sont malheureusement pas stockés en mémoire de façon séquentielle. Les adresses de début des 192 lignes ne peuvent être obtenues qu'après un calcul qui prend du temps s'il doit être répété de nombreuses fois, et ceci même en assembleur. C'est donc pour des raisons de rapidité que l'on utilise des tables contenant ces adresses de début de ligne. En réalité, il faut une table pour les octets bas des adresses (TAB1L) et une autre pour les octets hauts (TAB1H, TAB2H et TAB3H pour les pages 1, 2 et 3 respectivement).

Le principe de la loupe est de grossir l'image dans le sens de la hauteur, donc de dupliquer chaque ligne de la page 1 en l'écrivant deux fois en page 2 (pour les 91 premières lignes) et en page 3 (pour les 91 dernières lignes).

Le swapping (échange) des pages 2 et 3 consiste à échanger chaque octet d'une page avec l'octet correspondant de l'autre page.

Le déroulement d'écran (scrolling) est beaucoup plus compliqué. Prenons le cas du déroulement vers le haut et décomposons les différentes opérations :

- Sauvegarde des 40 octets de la première ligne (0) de la page 2 dans un tampon.
- Copie de la ligne 1 dans la ligne 0, ... de la ligne n dans la ligne n-1, ... de la dernière ligne (191) dans la ligne 190.
- Copie de la 1^{ère} ligne de la page 3 dans la dernière ligne de la page 2.
- Copie de la ligne 1 de la page 3 dans la ligne 0 de cette page, ... de la ligne n dans la ligne n-1, ... de la dernière ligne dans la ligne 190.
- Enfin copie du tampon dans la ligne 191 de la page 3.

Toutes ces opérations pour dérouler l'écran d'une ligne vers le haut ; pour x lignes, il faudra répéter toute la séquence x fois...

Le processus est inversé pour SCROLLDOWN qui effectue un déroulement vers le bas.

LOUPE est compatible DOS et ProDOS. Contrairement à certains écrits, le programme GF de la disquette IMAGEWRITER Toolkit (DOS 3.3) est apparemment utilisable sous ProDOS (servez-vous de CONVERT pour la conversion).

Les fichiers proposés sur la disquette :

LOUPE	: Programme source (Assembleur EDASM ProDOS).	LOUPE.Ø	: Programme assembleur = LOUPE.A + tables.
LOUPE.A	: Programme objet correspondant.	DEMO.LOUPE	: Driver pour LOUPE.Ø.
CREATABHGR	: Programme BASIC permettant de générer les tables d'adresses nécessaires.	LOGO.LOUPE	: Images HGR.
		IM.LOUPE	

Lisez page 35 comment utiliser la démo.

DEMO.LOUPE

```
100 REM DEMO.LOUPE
110 REM R.JOST (C) 1986
120 TEXT : HOME : PRINT "DEMONSTRATION DE LA LOUPE APPLE"
130 PRINT : PRINT
140 IMP = 1
150 D$ = CHR$(4)
160 PRINT D$"BLOAD LOUPE.Ø"
170 ONERR GOTO 190
180 PRINT D$"BLOAD GF": GOTO 200: REM GF EST LE PROGRAMME DE HARDCOPY DE LA DISQUETTE IMAGEWRITER TOOL KIT (DOS 3.3).
190 POKE 216,Ø:IMP = Ø
200 LOUPE = 35433:SWAP = 35551
210 INPUT "NOM DE L'IMAGE ";N$: IF N$ = "" THEN END
220 HGR : POKE - 16302,Ø
230 PRINT D$"BLOAD "N$",A$2000"
240 CALL 35328: REM INITIALISE CERTAINS POINTEURS
245 CALL 35367: REM APPEL COMMANDES
250 TEXT
260 IF IMP = Ø THEN 390
270 PRINT "VOULEZ-VOUS IMPRIMER L'IMAGE (O/N) ? ";; GET R$: PRINT R$: IF R$ < > "O" THEN 390
290 PRINT D$"PR£1"
300 PRINT CHR$(27) + CHR$(78);: REM CARACTERES PICA
310 PRINT CHR$(27) + CHR$(118);: REM POSITIONNE LE HAUT DE PAGE
320 PRINT CHR$(27) + CHR$(62);: REM MODE UNIDIRECTIONNEL
330 CALL LOUPE
340 POKE 6,1: POKE 7,7: CALL 36864: REM HARDCOPY : IMPRIMANTE EN SLOT 1, MODE DOUBLE GRANDEUR.
350 PRINT CHR$(27) + CHR$(114);: PRINT CHR$(31) + "1";: PRINT CHR$(27) + CHR$(102);: REM DEROULEMENT INVERSE + SAUT DE UNE LIGNE + RETOUR AU DEROULEMENT NORMAL
360 CALL SWAP
370 POKE 6,1: POKE 7,7: CALL 36864
380 PRINT : PRINT D$"PR£Ø"
390 TEXT : END
```

(suite de l'article page 28)

CREATABHGR

```

10 REM PROGRAMME DE CREATION DES
   TABLES DES ADRESSES HAUTE
   RESOLUTION
11 REM UTILISEES PAR LE PROGRAMME
   LOUPE.
15 FOR I = 0 TO 84: READ X: POKE 768 +
   I,X: NEXT
20 Z = 35880
30 POKE 230,32
40 FOR I = 0 TO 191
50 CALL 768,I
60 POKE Z + I, PEEK (6): REM OCTETS BAS
62 POKE Z + 192 + I, PEEK (7): REM
   OCTETS HAUTS PAGE 1

64 POKE Z + 384 + I, PEEK (7) + 32:
   REM OCTETS HAUTS PAGE 2
66 POKE Z + 576 + I, PEEK (7) + 64:
   REM OCTETS HAUTS PAGE 3
70 NEXT
2000 DATA 32, 245,230,32,11,3,133,6,132,7,
   96,138,74,74,74,168,185,34,3,72,138,
   41,7,10,10,24
2010 DATA 121,58,3,101,230,168,104,96,0,
   128,0,128,0,128,0,128,40,168,40,168,
   40,168,40,168
2020 DATA 80,208,80,208,80,208,80,208,0,
   0,1,1,2,2,3,3,0,0,1,1,2,2,3,3,0,0,1,1,2,
   2,3,3,0,0,0,0

```

LOUPE

PROGRAMME SOURCE

ASSEMBLÉ AVEC PROCODE à partir d'EDASM

```

1 *          14 X1          EQU $EB
2 * LOUPE    15 X2          EQU $EC
3 *          16 Y1          EQU $ED
4 * (C) 1986 R.Jost 17 Y2          EQU $EE
5 *          18 A1L         EQU $3C
6 *          19 A1H         EQU $3D
7 *          20 A2L         EQU $3E
8 Z1          EQU $06      21 A2H         EQU $3F
9 Z2          EQU $07      22 A3L         EQU $40
10 PAGE       EQU $09      23 A3H         EQU $41
11 DEPART     EQU $18      24 A4L         EQU $42
12 ARRIVEE    EQU $1A      25 A4H         EQU $43
13 ARRIVEE2   EQU $1C      26 *

27 TAB1L     EQU $8C28      ; octets bas des adr. des lignes
28 TAB1H     EQU $8CE8      ; octets hauts , page 1
29 TAB2H     EQU $8DA8      ; octets hauts , page 2
30 TAB3H     EQU $8E68      ; octets hauts , page 3
31 *
32 KBD       EQU $C000      ; teste le clavier
33 KBDSTRB   EQU $C010      ; STROBE
34 TXTCLR    EQU $C050      ; sélectionne mode graphique
35 MIXCLR    EQU $C052      ; plein écran

```

Avec EDASM de Apple, remplacez l'instruction DA de la ligne 323 par DW.

```

36 LOWSCR EQU $C054 ; page 1
37 MISCRC EQU $C055 ; page 2
38 HIRES EQU $C057 ; haute résolution
39 NEWSTT EQU $D7D2 ; retour au programme appelant
40 *
41 *
42 ORG $8A00
43 *
8A00: A2 00 44 LDX £0 ; octet bas des adresses HGR
8A02: BD 28 8C 45 LDA TAB1L,X
8A05: 85 3C 46 STA A1L
8A07: 85 40 47 STA A3L
8A09: BD A8 8D 48 LDA TAB2H,X
8A0C: 85 3D 49 STA A1H ; de la 1ère ligne page 2
8A0E: BD 68 8E 50 LDA TAB3H,X
8A11: 85 41 51 STA A3H ; et de la 1ère ligne page 3
52 ;
8A13: A2 BF 53 LDX £191 ; octet haut de l'adresse
8A15: BD 28 8C 54 LDA TAB1L,X
8A18: 85 3E 55 STA A2L
8A1A: 85 42 56 STA A4L
8A1C: BD A8 8D 57 LDA TAB2H,X
8A1E: 85 3F 58 STA A2H ; de la dernière ligne page 2
8A21: BD 68 8E 59 LDA TAB3H,X ; et de
8A24: 85 43 60 STA A4H ; la dernière ligne page 3
8A26: 60 61 RTS
62 *
63 *
64 * Commandes
65 *
8A27: AD 00 C0 66 TEST LDA KBD ; lit le clavier
8A2A: 10 FB 67 BPL TEST
8A2C: 2C 10 C0 68 BIT KBDSTRB
8A2F: C9 9B 69 CMP £$9B ; <ESC> interrompt le scrolling
8A31: D0 03 70 BNE HAUT
8A33: 4C D2 D7 71 JMP NEWSTT ; retour au pro. BASIC appelant
8A36: C9 95 72 HAUT CMP £$95 ; Flèche droite
8A38: D0 03 73 BNE BAS
8A3A: 20 1E 8B 74 JSR SCROLLUP
8A3D: C9 88 75 BAS CMP £$88 ; Flèche gauche
8A3F: D0 03 76 BNE GROS
8A41: 20 8E 8B 77 JSR SCROLLDOWN
8A44: C9 CC 78 GROS CMP £$CC ; L)oupe
8A46: D0 03 79 BNE ECH
8A48: 20 69 8A 80 JSR LOUPE
8A4B: C9 D3 81 ECH CMP £$D3 ; S)wap
8A4D: D0 03 82 BNE PAGE1
8A4F: 20 DF 8A 83 JSR SWAP
8A52: C9 8D 84 PAGE1 CMP £$8D ; RETURN
8A54: D0 03 85 BNE CONT
8A56: 20 5C 8A 86 JSR HGR
8A59: 18 87 CONT CLC ; retour obligé
8A5A: 90 CB 88 BCC TEST ; au début.
89 *

```

(suite page 30)

LOUPE (PROGRAMME SOURCE : suite)

```

90 * affiche la page 1 sans l'effacer
91 *
8A5C: 2C 50 C0 92 HGR BIT TXTCLR ; sélectionne le mode graphique
8A5F: 2C 52 C0 93 BIT MIXCLR ; pleine page
8A62: 2C 54 C0 94 BIT LOWSCR ; numéro 1
8A65: 2C 57 C0 95 BIT HIRES ; et haute résolution
8A68: 60 96 RTS
97 *
98 * grossit l'image page 1 (stockage entre $4000 et $7FFF)
99 *
8A69: 2C 50 C0 100 LOUPE BIT TXTCLR ; sélectionne le mode graphique
8A6C: 2C 52 C0 101 BIT MIXCLR ; plein écran
8A6F: 2C 55 C0 102 BIT MISCR ; page 2
8A72: 2C 57 C0 103 BIT HIRES ; haute résolution
8A75: A2 00 104 LDX £#00 ; initialisation
8A77: 86 EB 105 STX X1
8A79: 86 ED 106 STX Y1
8A7B: 86 06 107 STX Z1
8A7D: A2 C0 108 LDX £#C0 ; 192 lignes
8A7F: 86 07 109 STX Z2
8A81: 86 EE 110 STX Y2
8A83: A2 20 111 LDX £#20 ; page 1 d'abord
8A85: 86 09 112 STX PAGE
8A87: A2 28 113 LDX £#28 ; 40 colonnes
8A89: 86 EC 114 STX X2
8A8B: A6 ED 115 LOUPE1 LDX Y1 ; première ligne
8A8D: BD 28 8C 116 LDA TAB1L,X ; recherche son adresse
8A90: 85 18 117 STA DEPART
8A92: BD E8 8C 118 LDA TAB1H,X
8A95: 85 19 119 STA DEPART+1
8A97: A6 06 120 LDX Z1 ; première ligne de
8A99: BD 28 8C 121 LDA TAB1L,X ; l'image dilatée
8A9C: 85 1A 122 STA ARRIVEE
8A9E: BD E8 8C 123 LDA TAB1H,X
8AA1: 18 124 CLC
8AA2: 65 09 125 ADC PAGE ; en page 2
8AA4: 85 1B 126 STA ARRIVEE+1
8AA6: E8 127 INX
8AA7: 86 06 128 STX Z1
8AA9: BD 28 8C 129 LDA TAB1L,X ; la ligne est dédoublée
8AAC: 85 1C 130 STA ARRIVEE2
8AAE: BD E8 8C 131 LDA TAB1H,X
8AB1: 18 132 CLC
8AB2: 65 09 133 ADC PAGE
8AB4: 85 1D 134 STA ARRIVEE2+1
8AB6: A4 EB 135 LDY X1 ; 1ère colonne
8AB8: B1 18 136 LOUPE2 LDA (DEPART),Y ; on balaye la ligne
8ABA: 91 1A 137 STA (ARRIVEE),Y ;
8ABC: 91 1C 138 STA (ARRIVEE2),Y ;
8ABE: C8 139 INY ; octet suivant
8ABF: C4 EC 140 CPY X2 ; ligne complète ?
8AC1: D0 F5 141 BNE LOUPE2 ; non , on boucle

```

8AC3:	E6 06	142	INC	Z1	; oui , on passe
8AC5:	E6 ED	143	INC	Y1	; à la ligne suivante
8AC7:	A5 ED	144	LDA	Y1	
8AC9:	C9 60	145	CMP	£\$60	; moitié page 1 faite ?
8ACB:	D0 0B	146	BNE	LOUPE3	; non
8ACD:	A5 09	147	LDA	PAGE	; oui , il
8ACF:	18	148	CLC		; faut maintenant
8AD0:	69 20	149	ADC	£\$20	; transférer dans la
8AD2:	85 09	150	STA	PAGE	; page 3 (\$6000-7FFF).
8AD4:	A9 00	151	LDA	£\$00	; et on repart à
8AD6:	85 06	152	STA	Z1	; la première ligne
8AD8:	A5 ED	153	LOUPE3 LDA	Y1	
8ADA:	C5 EE	154	CMP	Y2	; toutes les lignes faites?
8ADC:	D0 AD	155	BNE	LOUPE1	; non
8ADE:	60	156	RTS		; terminé
		157	*		
		158	* échange les pages 2 et 3		
		159	*		
8ADF:	A2 00	160	SWAP LDX	£\$00	; initialise
8AE1:	86 EB	161	STX	X1	; 1ère colonne
8AE3:	86 ED	162	STX	Y1	; 1ère ligne
8AE5:	A2 28	163	LDX	£\$28	; 40=
8AE7:	86 EC	164	STX	X2	; dernière colonne+1
8AE9:	A2 C0	165	LDX	£\$C0	; 192 lignes=
8AEB:	86 EE	166	STX	Y2	; dernière ligne+1
8AED:	A2 40	167	LDX	£\$40	; page 2
8AEF:	86 09	168	STX	PAGE	
8AF1:	A6 ED	169	SWAP1 LDX	Y1	; on commence, ligne 0
8AF3:	BD 28 8C	170	LDA	TAB1L,X	; cherche l'adresse
8AF6:	85 18	171	STA	DEPART	
8AF8:	85 1A	172	STA	ARRIVEE	
8AFA:	BD A8 8D	173	LDA	TAB2H,X	
8AFD:	85 19	174	STA	DEPART+1	; octet de départ
8AFF:	BD 68 8E	175	LDA	TAB3H,X	
8B02:	85 1B	176	STA	ARRIVEE+1	; adresse d'arrivée
8B04:	A4 EB	177	LDY	X1	; première colonne
8B06:	B1 18	178	SWAP2 LDA	(DEPART),Y	; on balaye la ligne
8B08:	48	179	PHA		; sauve dans la pile
8B09:	B1 1A	180	LDA	(ARRIVEE),Y	; charge de la page arrivée
8B0B:	91 18	181	STA	(DEPART),Y	; stocke dans la page départ
8B0D:	68	182	PLA		; récupère de la pile
8B0E:	91 1A	183	STA	(ARRIVEE),Y	; stocke dans la page arrivée
8B10:	C8	184	INY		; on continue
8B11:	C4 EC	185	CPY	X2	; jusqu'à la dernière
8B13:	D0 F1	186	BNE	SWAP2	; colonne.
8B15:	E6 ED	187	INC	Y1	; ligne suivante ?
8B17:	A5 ED	188	LDA	Y1	
8B19:	C5 EE	189	CMP	Y2	
8B1B:	D0 D4	190	BNE	SWAP1	; oui , on n'a pas terminé
8B1D:	60	191	RTS		; c'est fini.
		192	*		
		193	*		
		194	* déroulement (scrolling) vers le haut		
		195	*		

(suite page 32)

LOUPE (PROGRAMME SOURCE : suite)

```

196 * on stocke la 1ère ligne de la page 2 dans le tampon
8B1E: A0 00 197 SCROLLUP LDY £0 ; on commence par la colonne 0
8B20: B1 3C 198 LIGNE1 LDA (A1L),Y ; charge un octet de la ligne
8B22: 99 FE 8B 199 STA TAMPON,Y ; et sauve dans le tampon
8B25: C8 200 INY ; balaye la ligne
8B26: C0 28 201 CPY £40 ; jusqu'au bout
8B28: D0 F6 202 BNE LIGNE1
203 ;
8B2A: A2 00 204 LDX £0 ; ligne 0
8B2C: BD 28 8C 205 ECRAN1 LDA TAB1L,X ; on copie la ligne n
8B2F: 85 1A 206 STA ARRIVEE
8B31: BD A8 8D 207 LDA TAB2H,X
8B34: 85 1B 208 STA ARRIVEE+1
8B36: E8 209 INX
8B37: BD 28 8C 210 LDA TAB1L,X ; dans la ligne n-1
8B3A: 85 1B 211 STA DEPART
8B3C: BD A8 8D 212 LDA TAB2H,X
8B3F: 85 19 213 STA DEPART+1
8B41: A0 00 214 LDY £0
8B43: B1 18 215 LIGNES1 LDA (DEPART),Y
8B45: 91 1A 216 STA (ARRIVEE),Y
8B47: C8 217 INY
8B48: C0 28 218 CPY £40
8B4A: D0 F7 219 BNE LIGNES1
8B4C: E0 BF 220 CPX £191 ; et ceci pour les 192
8B4E: D0 DC 221 BNE ECRAN1 ; lignes de la page 2
222 ;
8B50: A0 00 223 LDY £0 ; la 1ère ligne de la
8B52: B1 40 224 DLIGNE1 LDA (A3L),Y ; page 3 est copiée
8B54: 91 3E 225 STA (A2L),Y ; dans la dernière ligne
8B56: C8 226 INY ; de la page 2.
8B57: C0 28 227 CPY £40
8B59: D0 F7 228 BNE DLIGNE1
229 ;
8B5B: A2 00 230 LDX £0 ; on travaille maintenant
8B5D: BD 28 8C 231 ECRAN2 LDA TAB1L,X ; sur la page 3
8B60: 85 1A 232 STA ARRIVEE
8B62: BD 68 8E 233 LDA TAB3H,X
8B65: 85 1B 234 STA ARRIVEE+1
8B67: E8 235 INX
8B68: BD 28 8C 236 LDA TAB1L,X
8B6B: 85 1B 237 STA DEPART
8B6D: BD 68 8E 238 LDA TAB3H,X
8B70: 85 19 239 STA DEPART+1
8B72: A0 00 240 LDY £0
8B74: B1 18 241 LIGNES2 LDA (DEPART),Y
8B76: 91 1A 242 STA (ARRIVEE),Y
8B78: C8 243 INY
8B79: C0 28 244 CPY £40
8B7B: D0 F7 245 BNE LIGNES2
8B7D: E0 BF 246 CPX £191
8B7F: D0 DC 247 BNE ECRAN2
248

```

```

8B81: A0 00      249          LDY  £0          ; la ligne stockée dans
8B83: B9 FE 8B   250  DLIGNE2 LDA  TAMPON,Y  ; le tampon est copiée
8B86: 91 42      251          STA  (A4L),Y    ; dans la dernière ligne
8B88: C8         252          INY             ; de la page 3.
8B89: C0 28      253          CPY  £40
8B8B: D0 F6      254          BNE  DLIGNE2
8B8D: 60         255          RTS
                256 *
                257 * Déroulement vers le bas
                258 *
8B8E: A0 00      259  SCROLLDOWN LDY £0          ; la dernière ligne de
8B90: B1 3E      260  LIGNE191 LDA  (A2L),Y  ; la page 2 est copiée
8B92: 99 FE 8B   261          STA  TAMPON,Y  ; dans le tampon.
8B95: C8         262          INY
8B96: C0 28      263          CPY  £40
8B98: D0 F6      264          BNE  LIGNE191
                265
                266
8B9A: A2 BF      266          LDX  £191
8B9C: BD 28 8C   267  ECRAN1D LDA  TAB1L,X    ; la ligne n de la page 2
8B9F: 85 1A      268          STA  ARRIVEE   ; est copiée dans la ligne
8BA1: BD A8 8D   269          LDA  TAB2H,X    ; n+1.
8BA4: 85 1B      270          STA  ARRIVEE+1
8BA6: CA         271          DEX
8BA7: BD 28 8C   272          LDA  TAB1L,X
8BAA: 85 18      273          STA  DEPART
8BAC: BD A8 8D   274          LDA  TAB2H,X
8BAF: 85 19      275          STA  DEPART+1
8BB1: A0 00      276          LDY  £0
8BB3: B1 18      277  LIGNES1D LDA  (DEPART),Y
8BB5: 91 1A      278          STA  (ARRIVEE),Y
8BB7: C8         279          INY
8BB8: C0 28      280          CPY  £40
8BBA: D0 F7      281          BNE  LIGNES1D
8BBC: E0 00      282          CPX  £0
8BBE: D0 DC      283          BNE  ECRAN1D
                284
                285
8BC0: A0 00      285          LDY  £0          ; la dernière ligne de
8BC2: B1 42      286  DLIGNE1D LDA  (A4L),Y  ; la page 3 est copiée
8BC4: 91 3C      287          STA  (A1L),Y  ; dans la 1ère ligne
8BC6: C8         288          INY             ; de la page 2.
8BC7: C0 28      289          CPY  £40
8BC9: D0 F7      290          BNE  DLIGNE1D
                291
                292
8BCB: A2 BF      292          LDX  £191
8BCD: BD 28 8C   293  ECRAN2D LDA  TAB1L,X    ; la ligne n de la page 3
8BD0: 85 1A      294          STA  ARRIVEE   ; est copiée dans
8BD2: BD 68 8E   295          LDA  TAB3H,X    ; la ligne n+1.
8BD5: 85 1B      296          STA  ARRIVEE+1
8BD7: CA         297          DEX
8BD8: BD 28 8C   298          LDA  TAB1L,X
8BDB: 85 18      299          STA  DEPART
8BDD: BD 68 8E   300          LDA  TAB3H,X
8BE0: 85 19      301          STA  DEPART+1
8BE2: A0 00      302          LDY  £0
8BE4: B1 18      303  LIGNES2D LDA  (DEPART),Y

```

(suite page 34)

LOUPE (PROGRAMME SOURCE : suite)

```

8BE6: 91 1A      304          STA (ARRIVEE),Y
8BE8: C8        305          INY
8BE9: C0 28     306          CPY £40
8BEB: D0 F7     307          BNE LIGNES2D
8BED: E0 00     308          CPX £0
8BEF: D0 DC     309          BNE ECRAN2D
                        310
8BF1: A0 00     311          LDY £0 ; le tampon est recopié
8BF3: B9 FE 8B 312  DLIGNE2D LDA TAMPON,Y ; dans la 1ère ligne
8BF6: 91 40     313          STA (A3L),Y ; de la page 3.
8BF8: C8        314          INY
8BF9: C0 28     315          CPY £40
8BFB: D0 F6     316          BNE DLIGNE2D
                        317
8BFD: 60        318          RTS ;
                        319 *
                        320 *
                        321 * tampon
                        322 *
8BFE: 00 00 03 323 TAMPON DA $00,$03
      00        324 *
                        325 *
                        326 * c'est ici que commencent les tables HGR.

```

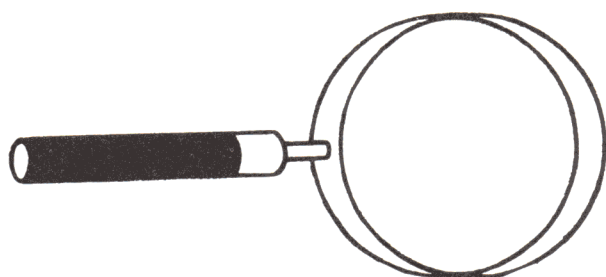


Table des symboles ordre alphabétique

-A1H.....\$003D	-A1L.....\$003C	-A2H.....\$003F	-A2L.....\$003E
-A3H.....\$0041	-A3L.....\$0040	-A4H.....\$0043	-A4L.....\$0042
-ARRIVEE...\$001A	-ARRIVEE2...\$001C	-BAS.....\$8A3D	-CONT.....\$8A59
-DEPART...\$0018	-DLIGNE1...\$8B52	-DLIGNE1D...\$8BC2	-DLIGNE2...\$8B83
-DLIGNE2D...\$8BF3	-ECH.....\$8A4B	-ECRAN1...\$8B2C	-ECRAN1D...\$8B9C
-ECRAN2...\$8B5D	-ECRAN2D...\$8BCD	-GROS.....\$8A44	-HAUT.....\$8A36
-HGR.....\$8A5C	-HIRES.....\$C057	-KBD.....\$C000	-KBDSTRB...\$C010
-LIGNE1...\$8B20	-LIGNE191...\$8B90	-LIGNES1...\$8B43	-LIGNES1D...\$8BB3
-LIGNES2...\$8B74	-LIGNES2D...\$8BE4	-LOUPE...\$8A69	-LOUPE1...\$8A8B
-LOUPE2...\$8AB8	-LOUPE3...\$8AD8	-LOWSCR...\$C054	-MISCR...\$C055
-MIXCLR...\$C052	-NEWSTT...\$D7D2	-PAGE.....\$0009	-PAGE1...\$8A52
-SCROLLDOW.\$8B8E	-SCROLLUP...\$8B1E	-SWAP.....\$8ADF	-SWAP1...\$8AF1
-SWAP2...\$8B06	-TAB1H...\$8CE8	-TAB1L...\$8C28	-TAB2H...\$8DA8
-TAB3H...\$8E68	-TAMPON...\$8BFE	-TEST.....\$8A27	-TXTCLR...\$C050
-X1.....\$00EB	-X2.....\$00EC	-Y1.....\$00ED	-Y2.....\$00EE
-Z1.....\$0006	-Z2.....\$0007		

LOUPE

/TM17/LOUPE/LOUPE.A,A#8A00,L#0202

Pour celles et ceux qui n'ont pas d'assembleur, mais possèdent la disquette **SIGNATURE** d'Yvan KOENIG.

Attention ! n'oubliez pas de compléter LOUPE.A avec la table générée par la routine Basic CREATABHGR (page 28) !

8A00:	A2 00 BD 28 8C 85 3C 85 40 BD A8 8D 85 3D BD 68	1572
8A10:	8E 85 41 A2 BF BD 28 8C 85 3E 85 42 BD A8 8D 85	3227
8A20:	3F BD 68 8E 85 43 60 AD 00 C0 10 FB 2C 10 C0 C9	0D57
8A30:	9B D0 03 4C D2 D7 C9 95 D0 03 20 1E 8B C9 88 D0	B57E
8A40:	03 20 8E 8B C9 CC D0 03 20 69 8A C9 D3 D0 03 20	6B46
8A50:	DF 8A C9 8D D0 03 20 5C 8A 18 90 CB 2C 50 C0 2C	B373
8A60:	52 C0 2C 54 C0 2C 57 C0 60 2C 50 C0 2C 52 C0 2C	409B
8A70:	55 C0 2C 57 C0 A2 00 86 EB 86 ED 86 06 A2 C0 86	EA52
8A80:	07 86 EE A2 20 86 09 A2 28 86 EC A6 ED BD 28 8C	8F0C
8A90:	85 18 BD E8 8C 85 19 A6 06 BD 28 8C 85 1A BD E8	82CD
8AA0:	8C 18 65 09 85 1B E8 86 06 BD 28 8C 85 1C BD E8	B9DD
8AB0:	8C 18 65 09 85 1D A4 EB B1 18 91 1A 91 1C C8 C4	D8F0
8AC0:	EC D0 F5 E6 06 E6 ED A5 ED C9 60 D0 0B A5 09 18	81CC
8AD0:	69 20 85 09 A9 00 85 06 A5 ED C5 EE D0 AD 60 A2	810F
8AE0:	00 86 EB 86 ED A2 28 86 EC A2 C0 86 EE A2 40 86	BF5E
8AF0:	09 A6 ED BD 28 8C 85 18 85 1A BD A8 8D 85 19 BD	7096
8B00:	68 8E 85 1B A4 EB B1 18 48 B1 1A 91 18 68 91 1A	F9BD
8B10:	C8 C4 EC D0 F1 E6 ED A5 ED C5 EE D0 D4 60 A0 00	3CF5
8B20:	B1 3C 99 FE 8B C8 C0 28 D0 F6 A2 00 BD 28 8C 85	9E1D
8B30:	1A BD A8 8D 85 1B E8 BD 28 8C 85 18 BD A8 8D 85	2119
8B40:	19 A0 00 B1 18 91 1A C8 C0 28 D0 F7 E0 BF D0 DC	F3EF
8B50:	A0 00 B1 40 91 3E C8 C0 28 D0 F7 A2 00 BD 28 8C	96EA
8B60:	85 1A BD 68 8E 85 1B E8 BD 28 8C 85 18 BD 68 8E	D69B
8B70:	85 19 A0 00 B1 18 91 1A C8 C0 28 D0 F7 E0 BF D0	F898
8B80:	DC A0 00 B9 FE 8B 91 42 C8 C0 28 D0 F6 60 A0 00	BD07
8B90:	B1 3E 99 FE 8B C8 C0 28 D0 F6 A2 BF BD 28 8C 85	A3DE
8BA0:	1A BD A8 8D 85 1B CA BD 28 8C 85 18 BD A8 8D 85	93FB
8BB0:	19 A0 00 B1 18 91 1A C8 C0 28 D0 F7 E0 00 D0 DC	9930
8BC0:	A0 00 B1 42 91 3C C8 C0 28 D0 F7 A2 BF BD 28 8C	64A9
8BD0:	85 1A BD 68 8E 85 1B CA BD 28 8C 85 18 BD 68 8E	6E7D
8BE0:	85 19 A0 00 B1 18 91 1A C8 C0 28 D0 F7 E0 00 D0	0ED9
8BF0:	DC A0 00 B9 FE 8B 91 40 C8 C0 28 D0 F6 60 00 00	9365
8C00:	03 00	1203

DEMO-LOUPE

Lorsque vous utiliserez la routine, à partir du STARTUP de notre disquette, choisissez l'option IM.LOUPE (c'est l'image ci-contre) et essayez les commandes indiquées par ce tableau !

COMMANDES POUR DEMO.LOUPE

L: MODE LOUPE
S: BASCULE ENTRE LES DEUX ECRANS
→: DERoule L'ECRAN VERS LE HAUT
←: DERoule L'ECRAN VERS LE BAS
ESC: RETOUR AU BASIC.
RETURN: RETOUR A L'ECRAN NORMAL

Impression sur IMAGEWRITER :

le programme GF de IMAGEWRITER TOOLKIT (DOS 3.3) doit être sur votre disquette.

UN x DEUX

Routine uniquement destinée à vous montrer comment il est possible de multiplier A par B... pour retrouver le résultat dans C, étant entendu que A B C peuvent être du type A(0) A(1) A(2) aussi bien que A(4) A(5) A(6)...

```

100 TEXT : NORMAL : HOME : GOSUB 200
110 VTAB 9: PRINT "RENTREZ DEUX CHIFFRES ": PRINT
120 INPUT "A(0) -> ":A(0)
130 INPUT "A(1) -> ":A(1): PRINT
140 CALL 768,A(0): PRINT : LIST 150
150 PRINT : PRINT A(2)
160 VTAB 22: PRINT "(E)ncore (M)enu (F)in ": GET R$
170 IF R$ = "E" OR R$ = "e" THEN HOME : GOTO 110

```

```

175 IF R$ = "M" OR R$ = "m" THEN PRINT CHR$(4)"RUN
    MENU"
180 IF R$ = "F" OR R$ = "f" THEN HOME : END
190 GOTO 160
200 FOR I = 768 TO 828: READ R: POKE I, R: NEXT : RETURN
210 DATA 32,190,222,165,184,72,165,185,72,32,227,223,133,6,
    132,7,104,133,185,104,133,184,32,103,221,32,99,235,164,
    7,165,6,24,105,5,144,1,200,32,249,234,32,130,233,
    164,7,165,6,24,105,10,170,144,1,200,32,45,235,76,46,237

```

300 :	20	BE	DE	JSR	\$DEBE] Teste la présence de la virgule.
303 :	A5	B8		LDA	\$B8	
305 :	48			PHA] Position de TXTPTR (adresse du dernier caractère obtenu par CHRGET) sauvegardée sur la pile.
306 :	A5	B9		LDA	\$B9	
308 :	48			PHA] PTRGET recherche la variable (ici A(0)).
309 :	20	E3	DF	JSR	\$DFE3	
30C :	85	06		STA	\$06] Son adresse est en A et Y. On la sauve en \$6.7.
30E :	84	07		STY	\$07	
310 :	68			PLA] Restauration de CHRGET.
311 :	85	B9		STA	\$B9	
313 :	68			PLA] FRNUM évalue la formule pointée par TXTPTR et la met dans FAC. FAC dans ARG.
314 :	85	B8		STA	\$B8	
316 :	20	67	DD	JSR	\$DD67] Partie haute de l'adresse de A(0) dans Y.
319 :	20	63	EB	JSR	\$EB63	
31C :	A4	07		LDY	\$07] Partie basse + 5 dans A pour adresse de A(1).
31E :	A5	06		LDA	\$06	
320 :	18			CLC] S'il y a retenue, la partie haute (Y) est incrémentée.
321 :	69	05		ADC	£\$05	
323 :	90	01		BCC	\$0326] MOVFM place le nombre pointé par Y, A dans FAC. FMULTT multiplie FAC par ARG.
325 :	C8			INY		
326 :	20	F9	EA	JSR	\$EAF9] Adresse haute dans Y.
329 :	20	82	E9	JSR	\$E982	
32C :	A4	07		LDY	\$07] Adresse basse majorée de \$A (deux fois 5) dans X via A, pour adresse de A(2).
32E :	A5	06		LDA	\$06	
330 :	18			CLC] Si retenue, on incrémente la partie haute.
331 :	69	0A		ADC	£\$0A	
333 :	AA			TAX] MOVMF transfère FAC vers le groupe de 5 octets pointé par X et Y. Affiche la valeur de FAC (résultat de la multiplication).
334 :	90	01		BCC	\$0337	
336 :	C8			INY		
337 :	20	2D	EB	JSR	\$EB2D	
33A :	4C	2E	ED	JMP	\$ED2E	

BLOLIB.A

Avec ProDOS, il suffit d'afficher le catalogue de la disquette pour connaître le monde de BLOCKS FREE... mais il peut se révéler intéressant, dans le cadre d'un programme, avant d'enregistrer des données, d'obtenir ce même renseignement sans pour autant passer par le catalogue... d'où cette routine.

300 :	20 00 BF	JSR \$BF00	Appel classique du MLI.
303 :	80		Commande READ BLOCK.
304 :	03 32		Adresse de la liste des paramètres.
306 :	A9 00	LDA £\$00] Nous allons utiliser deux adresses de la page 0 pour y écrire le nombre de blocs disponibles.
308 :	85 06	STA \$06	
30A :	85 07	STA \$07	
30C :	A0 C8	LDY £\$C8	Y prêt.
30E :	B9 FF 3F	LDA \$3FFF,Y	Lecture d'un octet.
311 :	A2 07	LDX £\$07] On passe les 8 bits en revue. Quand un bit est à 0, le bloc est occupé. Quand il est à 1, il est libre. ASL pousse chaque bit dans la retenue et il suffit d'incrémenter quand celle-ci est à 1.
313 :	0A	ASL	
314 :	90 06	BCC \$031C	
316 :	E6 06	INC \$06	
318 :	D0 02	BNE \$031C	
31A :	E6 07	INC \$07	
31C :	CA	DEX	
31D :	10 F4	BPL \$0313	
31F :	88	DEY	
320 :	D0 EC	BNE \$030E] Jusqu'à Y = 1 inclus.
322 :	A9 16	LDA £\$16	
324 :	20 5B FB	JSR \$FB5B] Curseur au début de la ligne choisie.
327 :	20 42 FC	JSR \$FC42	CLREOP efface jusqu'au bas de l'écran.
32A :	A6 06	LDX \$06	
32C :	A5 07	LDA \$07] LINPTR affiche les deux octets de X et A (mode décimal).
32E :	20 24 ED	JSR \$ED24	
331 :	60	RTS	Retour.
332 :	03] Table des paramètres, pour ProDOS... qui en attend 3 :
333 :	60		1. DRIVE et SLOT \$60 pour 6, \$50 pour 5, etc. (voir pour DRIVE la routine BLOLIB.V page suivante).
334 :	00 40		2. Adresse du Buffer : ici \$4000
336 :	06 00		3. Numéro du Bloc à lire : ici 6 (BIP.MAP).

BSAVE BLOLIB.A,A\$300,L\$38

BLOLIB.V

Appel par CALL 768, S, D, X... où S est le numéro du SLOT, D celui du DRIVE et X la variable qui, au retour, contiendra le nombre de BLOCKS FREE...

300 :	20	F5	E6	JSR	\$E6F5	GETBYTC transmet (dans X) le numéro du SLOT. On le passe dans A.
303 :	8A			TXA		
304 :	0A			ASL		Où 6 devient \$60, 5... \$50, etc.
305 :	0A			ASL		
306 :	0A			ASL		
307 :	0A			ASL		
308 :	48			PHA		SLOT sur la pile.
309 :	20	F5	E6	JSR	\$E6F5	Même opération pour le numéro de DRIVE.
30C :	8A			TXA		
30D :	4A			LSR		1 ou 0 tombe dans la retenue.
30E :	B0	04		BCS	\$0314	Si c'est un, rien à faire.
310 :	68			PLA		Si c'est 0, il faut forcer le bit 0 de la valeur SLOT à 1.
311 :	09	80		ORA	£\$80	
313 :	48			PHA		\$60 devient \$E0 (0110 0000... 1110 0000). Installation dans la liste des paramètres.
314 :	68			PLA		
315 :	8D	4C	03	STA	\$034C	Appel du MLI.
318 :	20	00	BF	JSR	\$BF00	\$80 : READ Block.
31B :	80					Adresse des paramètres.
31D :	4B	03				Préparation.
31E :	A9	00		LDA	£\$00	
320 :	85	06		STA	\$06	Y prêt.
322 :	85	07		STA	\$07	
324 :	A0	C8		LDY	£\$C8	Voir BLOLIB.A
326 :	B9	FF	3F	LDA	\$3FFF,Y	
329 :	A2	07		LDX	£\$07	GIVAYF rend flottant l'entier relatif dans A,Y (nbre de BLOCS). Résultat dans FAC.
32B :	0A			ASL		
32C :	90	06		BCC	\$0334	Teste la dernière virgule.
32E :	E6	06		INC	\$06	
330 :	D0	02		BNE	\$0334	Recherche la variable (son adresse est dans A,Y).
332 :	E6	07		INC	\$07	
334 :	CA			DEX		A passe dans X.
335 :	10	F4		BPL	\$032B	
337 :	88			DEY		MOVFM transfère FAC dans la variable pointée par X et Y.
338 :	D0	EC		BNE	\$0326	
33A :	A4	06		LDY	\$06	Liste des paramètres. (Voir BLOLIB.A page 37).
33C :	A5	07		LDA	\$07	
33E :	20	F2	E2	JSR	\$E2F2	
341 :	20	BE	DE	JSR	\$DEBE	
344 :	20	E3	DF	JSR	\$DFE3	
347 :	AA			TAX		
348 :	4C	2D	EB	JMP	\$EB2D	
34B :	03					
34C :	60					
34D :	00	40				
34F :	06	00				

BSAVE BLOLIB.V,A\$300,L\$51

GS Basic

IL ARRIVE... AVEC LES OUTILS

J' AURAIS aimé vous parler longuement du Basic qui, je l'espère fera bientôt le bonheur des possesseurs d'APPLE //GS. Hélas ! malgré une bonne vingtaine de coups de téléphone dans toutes les directions, malgré la bonne volonté manifestée par certains de mes interlocuteurs (merci tout de même, Xavier Schott), je n'ai pas réussi à dénicher la documentation (en anglais) qui accompagne normalement la disquette. Dommage ! L'événement justifiait, à mes yeux du moins, un petit retard dans la mise en vente de TREMLIN MICRO. A une époque où la Communication passe par une bonne Information des Médias, il ne semble pas que la Direction d'Apple France tienne réellement à utiliser cet excellent véhicule de l'information qu'est la Presse technique et spécialisée.

Hélas ! d'autres responsables français partagent ce point de vue. On se contente de réunir les représentants de la grande presse, à l'occasion de cocktails, petits déjeuners et autres manifestations bien parisiennes : petits fours, boissons fraîches (ou alcoolisées... speech de l'attachée de presse (si possible jeune et jolie) et discours du responsable des festivités. Tout cela aura peut-être l'honneur de la télévision, justifiera quelques lignes dans deux ou trois quotidiens nationaux... Bref, un truc certes bien ficelé, dont je ne nie pas

l'utilité, mais dont je conteste la valeur sur le plan "information du client potentiel"... notre lecteur. J'aurai sans doute la doc du GS Basic dans quelques jours, en la payant bien entendu, mais trop tard pour vous en parler dans ce numéro.

Premier constat

GS Basic fonctionne sous ProDOS 16, le futur standard de l'Apple //GS. Pour l'instant, sa durée de chargement reste exagérée, mais la prochaine nouvelle version de ProDOS 16 devrait améliorer cela.

L'éditeur paraît un peu lent, mais ce

n'est semble-t-il qu'une illusion. En réalité, il est difficile de perdre un caractère, même en accélérant la frappe. Comme cela se passe avec APW, il y a mémorisation des trois dernières lignes, qu'il est facile de rappeler en appuyant sur la flèche correspondant à cette option. On peut alors insérer et supprimer à volonté dans la ligne, ce qui est évidemment pratique. Nous n'envierons plus IBM sur ce point !

Visiblement, si ce nouveau Basic ressemble à notre bon vieil Apple-soft, il n'en a heureusement pas conservé les insuffisances les plus notoires. Nous en reparlerons en détail dans le prochain numéro de TREMLIN MICRO...

Par contre, avec lui, vous aurez enfin accès — et facilement, si j'en juge par l'exemple figurant sur la disquette —, aux outils de l'Apple //GS. A vous les fenêtres, le graphisme, et tout le reste ! Ceci dit, ne vous leurrez pas, ce n'est pas tout simple, et le Basic par lui-même se révélera relativement lent... pour s'accélérer quand la boîte à outils prendra le relais. Je vous mets l'eau à la bouche, pas vrai ? Croyez bien que je suis le premier à regretter de ne pas être en mesure de vous en dire davantage. Rendez-vous le 3 janvier ! **G.H.**

Apple Expo



Sous la Grande Halle de La Villette, durant 3 jours 1/2 Apple Expo a accueilli 44 136 visiteurs.

Dès 14 H le premier jour, puis dès 10 h les suivants, l'affluence était à son comble jusqu'à 19 h sur les stands des 170 exposants. Cette affluence record était d'ailleurs prévisible à la lecture des chiffres de consultation du serveur sur Minitel. Sur la totalité du mois de septembre, le serveur a été en connexion 5332 fois, et l'ensemble du temps de consultations représente 577 heures.

Pour la première fois le département des Ressources Humaines était présent à Apple Expo afin de favoriser la synergie entre les partenaires qui recrutent et les candidats potentiels. Plus de 70 offres d'emplois ont ainsi été affichées de la part du réseau Apple, plus de 100 candidatures ont été déposées pour Apple, ses concessionnaires et l'Apple Business School.

Parmi les chiffres records signalons également celui réalisé par la boutique des "goodies" Apple : 689 000 F, qui démontre que l'engouement pour les objets marqués à la pomme ne se dément pas, et ce d'autant plus qu'une collection entièrement nouvelle était présentée. Enfin, au détour des allées ce ne sont pas moins de 11 tonnes de pommes qui ont été croquées pendant 3 jours 1/2 !

MÉLANGEUR

Comment mélanger aléatoirement les N mots d'un fichier binaire (voir SAISIE — dans nos ROUTINES LM — et MULTIE.SAISIE — dans nos NOUVELLES ROUTINES ?

Par exemple en utilisant MÉLANGEUR.

PRÉCAUTION : Votre fichier devra impérativement se terminer par Lx2 zéros (L = longueur d'un mot, soit 8 zéros pour des mots de 4 lettres).

MEL.ESSAI

```

100 D$ = CHR$(4): PRINT D$"BLOOD
    MELANGEUR"
110 PRINT D$"PR£3": PRINT : HOME
120 PRINT D$"BLOOD ES5"
130 L = 5
140 HOME
150 CALL 768,L
160 A = L + 1 + ( PEEK (8) + PEEK
    (9) * 256)
170 LF = 1 + ( PEEK (25) + PEEK
    (26) * 256)
180 FOR I = A TO A + L - 1: M$ = M$ +
    CHR$( PEEK (I)): NEXT

```

```

190 N = N + 1: M$(N) = M$: M$ = "" : A
    = A + L: IF A < LF THEN 180
200 FOR I = 1 TO N: PRINT M$(I),: NEXT
210 PRINT : PRINT
220 N = 0: S = S + 1: IF S < 6 THEN 150
230 VTAB 23: PRINT "(M)ENU DE DIS
    QUETTE (F)IN "": GET R$: VTAB 22:
    PRINT
240 IF R$ = "M" OR R$ = "m" THEN
    PRINT D$"RUN MENU,S6"
250 IF R$ <> "F" AND R$ <> "f"
    THEN 230
260 HOME

```

MEL.BAS

```

100 D$ = CHR$(4): PRINT D$"BLOOD MELANGEUR"
110 PRINT D$"PR£3": PRINT : HOME
120 PRINT D$"CATALOG"
130 PRINT : INPUT "TITRE A TRAITER ? "; T$: IF T$ = "" THEN END
140 INPUT "LONGUEUR DES MOTS ? "; L: IF NOT L THEN END
150 PRINT D$"BLOOD" T$
160 CALL 768,L
170 A = L + 1 + ( PEEK (8) + PEEK (9) * 256)
180 LF = 1 + L * 2 + ( PEEK (25) + PEEK (26) * 256) - A
190 PRINT D$"BSAVE"; T$ + "M"; ",A"; "A"; "L"; LF
200 PRINT D$"CATALOG"
210 PRINT
220 VTAB 23: PRINT "(M)ENU DE DISQUETTE (F)IN "": GET R$:
    VTAB 22: PRINT
230 IF R$ = "M" OR R$ = "m" THEN PRINT D$"RUN MENU,S6"
240 IF R$ <> "F" AND R$ <> "f" THEN 220
250 HOME

```

MÉLANGEUR

POINTEURS : \$18 = Longueur d'un mot — \$08-09 = Adresse de base du fichier (moins la longueur d'un mot et moins un octet) — \$06-07 = Adresse du mot choisi aléatoirement — \$19-1A = Adresse du mot traité séquentiellement — \$1B-1C = Nombre de mots moins deux.

LONGUEUR DES MOTS

300 :	20	F5	E6	JSR	E6F5] GETBYTC saute un caractère, puis appelle GETBYT et CONINT. La valeur envoyée par CALL est dans X.
303 :	86	18		STX	18	

ADRESSE DU FICHIER

305 :	AD	B9	BE	LDA	BEB9] La partie basse est diminuée de la longueur d'un mot + 1 (on annule la retenue, comme pour une addition, ce qui diminue le résultat d'un octet). Stockage en \$8 et \$19.
308 :	18			CLC		
309 :	E5	18		SBC	18	
30B :	85	08		STA	08	
30D :	85	19		STA	19	

30F :	AD	BA	BE	LDA	BEBA] La partie haute va en \$9 et \$1A. Sous DOS 3.3, on utiliserait \$AA72 et \$AA73 au lieu de \$BEB9-BEBA.
312 :	E9	00		SBC	£00	
314 :	85	1A		STA	1A	
316 :	85	09		STA	09	

LONGUEUR DU FICHIER (réelle)

318 :	AC	C8	BE	LDY	BEC8] Longueur réelle du fichier dans Y et A. Sous DOS 3.3, remplacer \$BEC8-BEC9 par \$AA60-AA61. GIVAYF rend flottant l'entier relatif de Y,A. MOVAF transfère FAC dans ARG.
31B :	AD	C9	BE	LDA	BEC9	
31E :	20	F2	E2	JSR	E2F2	
321 :	20	63	EB	JSR	EB63	

NOMBRE DE MOTS (moins 3)

324 :	A4	18		LDY	18] Longueur du mot dans Y. A = 0.
326 :	A9	00		LDA	£00	
328 :	20	F2	E2	JSR	E2F2] L'entier relatif de Y,A devient flottant. FDIVT s'occupe de la division (résultat dans FAC). GETADR transforme FAC en 2 octets (dans Y,A).
32B :	20	69	EA	JSR	EA69	
32E :	20	52	E7	JSR	E752	

331 :	85	1C		STA	1C] Stockage de la partie haute dans \$1C. La partie basse passe dans A.
333 :	98			TYA		
334 :	38			SEC] Nombre de mots diminué de 3 unités.
335 :	E9	03		SBC	£03	
337 :	85	1B		STA	1B] Stockage de la partie basse dans \$1B. Partie haute décrétementée si nécessaire.
339 :	B0	02		BCS	033D	
33B :	C6	1C		DEC	1C	

ADRESSE DU MOT À TRAITER

33D :	18			CLC] Annulation de la retenue.
33E :	A5	18		LDA	18	
340 :	65	19		ADC	19] Pointeurs de l'adresse de traitement majorés de la longueur d'un mot. La partie haute n'est incrémentée que s'il y a retenue.
342 :	85	19		STA	19	
344 :	90	02		BCC	0348	
346 :	E6	1A		INC	1A	

NOMBRE ALÉATOIRE — MOT ALÉATOIRE

348 :	A9	01		LDA	£01] Entrée spéciale pour RND.
34A :	20	B1	EF	JSR	EFB1	
34D :	20	63	EB	JSR	EB63] MOVAF transfère FAC dans ARG.

(suite page 42)

350 :	A4 1B	LDY 1B] Nombre de mots (toujours moins 3) dans Y et A.
352 :	A5 1C	LDA 1C	
354 :	20 F2 E2	JSR E2F2	GIFAYF rend l'entier flottant.
357 :	20 82 E9	JSR E982	FMULTT se charge de multiplier l'argument par FAC.
35A :	20 63 EB	JSR EB63	MOVAF transfère FAC dans ARG.
35D :	20 52 E7	JSR E752	FAC transformé en entier sur 2 octets, dans Y et A.
360 :	C8	INY	Y = Y + 1 pour éviter le 0.
361 :	D0 01	BNE 0364] Si Y = 0, il faut incrémenter A (partie haute). (Attention ! pas de "1A" sur le vieux 6502 !).
363 :	1A	INC	
364 :	20 F2 E2	JSR E2F2	L'entier de Y, A devient (ou redevient ?) flottant.
367 :	20 63 EB	JSR EB63	Toujours MOVAF (FAC dans ARG).

ADRESSE DU MOT ALÉATOIREMENT CHOISI

36A :	A4 18	LDY 18] Longueur du mot dans Y,A.
36C :	A9 00	LDA 00	
36E :	20 01 E3	JSR E301	SGNFLT rend l'entier de Y flottant.
371 :	20 82 E9	JSR E982	FMULTT multiplie ARG par FAC.
374 :	20 52 E7	JSR E752	GETADR transforme FAC en entier sur 2 octets (Y,A).
377 :	48	PHA	A est empilé.
378 :	98	TYA	Y passe dans A (partie basse de l'adresse).
379 :	18	CLC] Addition et mise à jour de l'adresse du mot choisi aléatoirement.
37A :	65 08	ADC 08	
37C :	85 06	STA 06	
37E :	68	PLA	A est récupéré.
37F :	65 09	ADC 09] Même opération sur la partie haute de l'adresse.
381 :	85 07	STA 07	

LECTURE D'UN MOT (ordre séquentiel)

383 :	A4 18	LDY 18	Longueur du mot dans Y.
385 :	B1 19	LDA (19),Y	Lecture d'un caractère.
387 :	F0 12	BEQ 039B	Si c'est un zéro, le mélange est terminé.
389 :	48	PHA	Octet empilé.
38A :	88	DEY	Y = Y - 1 (jusqu'à 1 inclus).
38B :	D0 F8	BNE 0385	Quand Y = 0, on arrête.

PERMUTATION

38D :	C8	INY	Y = Y + 1 (au départ, Y = 0 + 1).
38E :	B1 06	LDA (06),Y	Lecture d'un caractère du mot choisi aléatoirement.
390 :	91 19	STA (19),Y	Ecriture de l'octet du mot traité.
392 :	68	PLA	Récupération sur la pile.
393 :	91 06	STA (06),Y	Remplacement.
395 :	C4 18	CPY 18] Si Y est égal à la longueur du mot, on passe au suivant. Sinon, on continue la permutation.
397 :	D0 F4	BNE 038D	
399 :	80 A2	BRA 033D	
39B :	60	RTS	Retour au Basic.

BSAVE MÉLANGEUR, A\$300, L\$9C

À PROPOS DE \$BEB9-BEBA

Sous ProDOS (voir la partie *Adresse du Fichier*), on lit l'adresse en question en FIAUX10 (\$BEB9-BEBA), mais ce renseignement sera inexact si l'on a chargé le fichier à une adresse différente de celle de l'enregistrement. Il est alors préférable de lire l'adresse d'un fichier chargé dans :

- RWDATA \$BED7-BED8 Pointeur pour lecture-écriture de données.
- RWCOUNT \$BED9-BEDA Fournit le nombre de bytes, comme \$BEC8-BEC9. ■

QUATRE OPÉRATIONS

Comment transmettre des données à une routine LM de calcul ? Voici une solution simple, facile à comprendre.

REMARQUE : On envoie $X = 0$ pour la soustraction, $X = 1$ pour l'addition, 2 pour la multiplication et 3 pour la division. On peut utiliser \times à la place de $*$ (multiplication) et $:$ à la place de $/$ (division).

```

100 TEXT : NORMAL :D$ = CHR$ (4): PRINT D$"PR&3": PRINT
110 FOR I = 768 TO 976: READ R: POKE I,R: IF R <> 0 THEN NEXT
120 HOME : PRINT "Q U A T R E   O P E R A T I O N S " :
      PRINT "_____ "
130 PRINT : VTAB 12: CALL - 958
140 PRINT "Opération (N1 - N2 ou N1 + N2 ou N1 * N2 ou N1 / N2)"
150 VTAB 14: INPUT "N1 -> ";R$:N1 = VAL (R$): IF NOT N1 THEN 230
160 VTAB 15: PRINT "SIGNE -> ";; GET R$: PRINT R$
170 FOR X = 0 TO 3: IF R$ <> MID$ ("- + x :",X + 1,1) THEN NEXT
180 IF X <> 4 THEN 200
190 FOR X = 0 TO 3: IF R$ <> MID$ ("- + * /",X + 1,1) THEN NEXT : IF X = 4 THEN
      160
200 VTAB 16: INPUT "N2 -> ";R$:N2 = VAL (R$): IF NOT N2 THEN 150
210 CALL 768,X,N1,N2
220 CALL - 198: POKE 49168,0: WAIT 49152,128: POKE 49168,0: IF PEEK (49152) <> 27
      THEN 130
230 VTAB 22: PRINT : PRINT "(E)NCORE (M)ENU DE DISQUETTE (A)PPLESOFT ";; GET R$
240 IF R$ = "E" OR R$ = "e" THEN 130
250 IF R$ = "M" OR R$ = "m" THEN PRINT D$"RUN MENU"
260 IF R$ = "A" OR R$ = "a" THEN HOME : END
270 GOTO 230
280 DATA 32,245,230,169,231,160,170,224,1,144,22,224,2,176,4,160,193,128,14,224,
      3,176,6,169,233,160,130,208,4,169,234,160,105,140,55,3,141,56,3,32,190,222,32,
      123,221,32,99,235,32,190,222,32,123,221,32,105,234,32,52,237,76,58,219,0

```

QUELLE OPÉRATION ?

300 :	20 F5 E6	JSR	\$E6F5	GETBYTC saute un caractère et appelle GETBYT, puis CONINT. Le résultat est dans X.
303 :	A9 E7	LDA	£\$E7] \$E7AA (FSUBT) : soustraction dans A et Y.
305 :	A0 AA	LDY	£\$AA	
307 :	E0 01	CPX	£\$01] Si X = 0, c'est bien la soustraction : saut.
309 :	90 16	BCC	\$0321	
30B :	E0 02	CPX	£\$02] Si X supérieur ou égal à 2, voir plus loin...
30D :	B0 04	BCS	\$0313	
30F :	A0 C1	LDY	£\$C1] X = 1, c'est l'addition. \$C1 dans Y pour FADDT (\$E7C1).
311 :	80 0E	BRA	\$0321	
313 :	E0 03	CPX	£\$03] Si X supérieur ou égal à 3, voir plus loin...
315 :	B0 06	BCS	\$031D	
317 :	A9 E9	LDA	£\$E9] \$E982 (FMULTT) : adresse de la routine de multiplication dans A et Y.
319 :	A0 82	LDY	£\$82	
31B :	D0 04	BNE	\$0321	Saut.
31D :	A9 EA	LDA	£\$EA] \$EA69 (FDIVT) : adresse de la routine de division.
31F :	A0 69	LDY	£\$69	
321 :	8C 37 03	STY	\$0337] Mise en place de l'adresse de la routine utilisée.
324 :	8D 38 03	STA	\$0338	
327 :	20 BE DE	JSR	\$DEBE	CHKCOM teste la virgule.
32A :	20 7B DD	JSR	\$DD7B	FRMEVL : évalue la formule. Résultat dans FAC.
32D :	20 63 EB	JSR	\$EB63	MOVAF : FAC dans ARG.
330 :	20 BE DE	JSR	\$DEBE] Même processus que ci-dessus pour N2.
333 :	20 7B DD	JSR	\$DD7B	
336 :	20 82 E9	JSR	\$E982	Routine choisie (adresse variable).
339 :	20 34 ED	JSR	\$ED34	FOUT convertit la valeur de FAC en chaîne de chiffres.
33C :	4C 3A DB	JMP	\$DB3A	STROUT affiche cette chaîne. ■

RND 0 OU 1 ?

Comment obtenir (aléatoirement bien sûr) une série de 0 ou de 1 ?

300 :	A9 01	LDA	£\$01] Entrée spéciale de RND.
302 :	20 B1 EF	JSR	\$EFB1	
305 :	20 63 EB	JSR	\$EB63	FAC vers ARG.
308 :	A0 02	LDY	£\$02] Y flottant dans FAC.
30A :	20 01 E3	JSR	\$E301	
30D :	20 82 E9	JSR	\$E982	FAC multiplié par ARG.
310 :	20 52 E7	JSR	\$E752	On a le nombre sur 2 octets dans Y et A.
313 :	98	TYA		La partie basse passe dans A.
314 :	09 B0	ORA	£\$B0	Bit 7 à 1.
316 :	20 ED FD	JSR	\$FDED	Affichage.
319 :	2C 10 C0	BIT	\$C010] On continue aussi longtemps que la touche ESCAPE n'a pas été pressée.
31C :	AD 00 C0	LDA	\$C000	
31F :	2C 10 C0	BIT	\$C010	
322 :	C9 1B	CMP	£\$1B	
324 :	D0 DA	BNE	\$0300] Retour.
326 :	60	RTS		

VOIR DÉMO SUR VOTRE DISQUETTE.

ALLER-RETOUR

Vous communiquez directement — ou presque ! — avec les routines arithmétiques du 65C02, mais comment récupérer le nombre obtenu dans une variable ?

Tout bêtement en fournissant au microprocesseur le nom de ladite variable — ici N3. Quand le calcul est effectué \$DFE3 (PTRGET) va rechercher cette variable et éventuellement la créer. Sa valeur sera pointée par VARPNT (\$83-84). MOVMF (\$EB2D) se chargera de transférer la valeur de FAC à cette adresse... si vous prenez la précaution — ô combien élémentaire — de mettre la partie basse de l'adresse en question dans X et sa partie haute dans Y.

```

100 TEXT : NORMAL :D$ = CHR$(4): PRINT D$"PR£3": PRINT
110 FOR I = 768 TO 976: READ R: POKE I,R: IF R <> 0 THEN NEXT
120 HOME : PRINT "A L L E R - R E T O U R ":
      PRINT " _____ "
130 PRINT : VTAB 12: CALL - 958
140 VTAB 14: INPUT "N1→» ";R$:N1 = VAL (R$): IF NOT N1 THEN 190
150 VTAB 16: INPUT "N2→» ";R$:N2 = VAL (R$): IF NOT N2 THEN 140
160 CALL 768,N1,N2,N3
170 PRINT N3
180 CALL - 198 : POKE 49168,0: WAIT 49152,128: POKE 49168,0: IF PEEK (49152) <> 27 THEN 130
190 VTAB 22: PRINT "(E)NCORE (M)ENU DE DISQUETTE (A)PPLESOFT "": GET R$
200 IF R$ = "E" OR R$ = "e" THEN 130
210 IF R$ = "M" OR R$ = "m" THEN PRINT D$"RUN MENU"
220 IF R$ = "A" OR R$ = "a" THEN HOME : END
230 GOTO 190
240 DATA 32,190,222,32,123,221,32,99,235,32,190,222,32,123,221,32,130,233,32,190,222,32,227,223,
      166,131,164,132,76,45,235,0

```

300 :	20 BE DE	JSR	\$DEBE] N1 dans FAC.
303 :	20 7B DD	JSR	\$DD7B	
306 :	20 63 EB	JSR	\$EB63] FAC dans ARG.
309 :	20 BE DE	JSR	\$DEBE	
30C :	20 7B DD	JSR	\$DD7B] N2 dans FAC.
30F :	20 82 E9	JSR	\$E982	
312 :	20 BE DE	JSR	\$DEBE] On divise, mais on pourrait faire n'importe quoi...
315 :	20 E3 DF	JSR	\$DFE3	
318 :	A6 83	LDX	\$83] Adresse de la valeur de N3 dans VARPNT.
31A :	A4 84	LDY	\$84	
31C :	4C 2D EB	JMP	\$EB2D	
] VARPNT dans X et Y.
] MOVMF transfère la valeur.

TRI.B

Comment trier rapidement une liste de nombres sur un octet ? Jusqu'à 255, voici une routine classique*, pouvant d'ailleurs servir de base à d'autres, plus complexes. Elle sous-entend que la liste à trier est installée à partir de l'adresse \$2000... et que le nombre d'éléments est contenu dans la case-mémoire \$1FFF.

Dans notre démonstration, les nombres sont aléatoirement mis en place par la ligne Basic 20. Nous nous sommes limités, ici, à 95 pour simplifier l'affichage, mais vous pouvez essayer avec d'autres valeurs (ne dépassez pas 255). La vérification (\$302 à \$308) pourrait être escamotée dans un programme où l'on aurait la certitude d'avoir plus d'un élément. Il n'est pas davantage interdit de placer le nombre d'éléments moins un (NE-1) dans \$1FFF, ce qui élimine le DEY de la ligne \$30C.

* Elle n'est certainement pas inédite, soit dans cette version, soit dans des versions voisines.

```

10 TEXT : NORMAL :D$ = CHR$(4): PRINT D$"PR&3": PRINT
   : PRINT D$"BLOAD TRI/TRI": HOME
20 FOR I = 8192 TO 8192 + 94: POKE I,1 + INT ( RND
   (1) * 95): NEXT : POKE 8191,95
30 A$ = "PRESSEZ UNE TOUCHE POUR ACTIVER LE TRI ":
   GOSUB 100
40 CALL 768: HOME
50 A$ = "VOTRE LISTE EST TRIEE (PAR COLONNE) ":
   GOSUB 100
60 A$ = "(TRI POUR UNE AUTRE LISTE ": GOSUB 110
70 VTAB 23: PRINT "(M)ENU DE DISQUETTE (A)PPL"

SOFT "": GET R$: VTAB 22: PRINT
80 IF R$ = "M" OR R$ = "m" THEN PRINT D$"RUN
   MENU,S6"
90 IF R$ = "A" OR R$ = "a" THEN HOME : END
95 GOTO 60
100 FOR I = 8192 TO 8192 + 18: FOR J = I TO I + 76 STEP
   19: PRINT PEEK (J),: NEXT : NEXT
110 VTAB 22: CALL - 958: PRINT : PRINT A$: GET R$: VTAB
   22: PRINT
120 IF R$ = "T" OR R$ = "t" THEN HOME : GOTO 20
130 RETURN

```

300 : A2 00

LDX £\$00

Registre X à 0.

PRÉCAUTION

302 : AC FF 1F

LDY \$1FFF

Nombre d'éléments à trier dans Y.

305 : C0 02

CPY £\$02

S'il n'y a pas de retenue, on n'a qu'un élément et le tri est inutile.

307 : 90 1F

BCC \$0328

TRI

309 : AC FF 1F

LDY \$1FFF

Nombre d'éléments moins un dans Y (NE - 1).

30C : 88

DEY

30D : B9 00 20

LDA \$2000,Y

Charger NE + Y dans l'Accumulateur et le comparer à NE - 2.

310 : D9 FF 1F

CMP \$1FFF,Y

313 : B0 0D

BCS \$0322

S'il n'y a pas de retenue, il est plus petit. S'il est plus grand, on saute. Puisqu'il y a échange, X est mis à 1.

315 : A2 01

LDX £\$01

317 : 48

PHA

A empilé (nombre à permuter).

318 : B9 FF 1F

LDA \$1FFF,Y

31B : 99 00 20

STA \$2000,Y

Le plus grand prend la place du plus petit.

31E : 68

PLA

31F : 99 FF 1F

STA \$1FFF,Y

On retrouve celui-ci sur la pile et il prend la place du plus grand.

322 : 88

DEY

323 : D0 E8

BNE \$030D

Y = Y - 1. On continue jusqu'à Y = 1 inclus.

325 : CA

DEX

326 : F0 E1

BEQ \$0309

X = X - 1. Si X = 0, il y a eu permutation : encore un tour !

328 : 60

RTS

LE MONDE DES C

J' ai rencontré un lecteur de *TREMLIN MICRO*, un individu comme vous et pas moi, le front plissé par l'effort, penché à une terrasse de bistrot sur **UN DE MES ARTICLES**. J'observai quelques instants le spécimen, tandis qu'il plongeait son sandwich jambon beurre dans sa bière pression, sous l'œil réprobateur de son fox-terrier à poils durs. Moi, vous connaissez, curieux, je m'approchai, prêt à jurer que je m'appelais Y. KOENIG si besoin était de me montrer discret. C'est beau un homme qui pense. Celui-là pensait, mangeait son sandwich trempé et parlait en même temps : un vrai régale esthétique.

De son monologue chuinté, il apparaissait que le langage **C**, c'est peut-être très simple, mais qu'on n'y comprend rien tant qu'on n'a pas vu un programme qui fait vraiment quelque chose...

Bon ! après tout, c'est encore un peu les vacances. Le clavier de mon GS fait encore CRRR, CRRR, CRRR à cause de tout le sable qui est resté coincé dedans. Je vais être gentil avec tous les courageux studieux qui n'achètent *TREMLIN MICRO* que dans le but de découper les pages de "**C**" très simples et de les porter deux mois sur leur cœur. Voici un vrai programme, un beau comme un camion, presque aussi gros que l'objet précité, à comprendre ou à taper comme un fou, pratiquement inutile si vous remplissez vos chèques pour *TREMLIN MICRO* à la main, mais indispensable autrement. On y retrouve le bon vieux **PRINTF** et le nouveau **SCANF**, qui est la forme évoluée de l'**INPUT** Basic. Vous retrouverez également les tableaux rencontrés dans le numéro précédent. En plus, et en prime, vous allez découvrir la structure de **rupture de séquence conditionnelle**, le...

```
if (condition) instruction; /* Ressemble furieusement au IF THEN basic */
else autre_instruction; /* instruction peut être un bloc d'instruction */
```

l'aiguillage (équivalent au case du Pascal)

```
switch (variable)
    é
    case valeur1: instruction1;break;
    case valeur2: instruction2;break;
    /* etc ..... */
    default: instruction_par_défaut;
    è
```

Si la variable vaut valeur1 on exécute instruction1, si elle vaut valeur2 on exécute instruction2, etc. Si elle ne prend aucune des valeurs définies on exécute instruction __ par __ défaut.

La boucle conditionnelle

```
do
    é
    instruction1;
    instruction2;
    /* etc ..... */
    è
while (condition);
```

Qui consiste à répéter un bloc d'instructions (ou une instruction) tant qu'une condition n'est pas remplie.

A vous de jouer, à vos claviers, je veux entendre CRRR, CRRR, CRRR...

(suite page 48)

NOMBRES EN LETTRES

Traduit un nombre entre 1 et 999999999 en lettres

```

=====
(c) 1987 CLAUDE AUBRY et TREMPIN-MICRO LANGAGE C POUR XT,AT,IIGS,MAC etc
Version IIGS compilée sur APWC 1.0B7
Adaptation pour IIGS Emile SCHWARZ
=====

```

```

#include <stdio.h>

```

```

/*=====
                               Traitement d'un nombre de trois chiffres
=====*/

```

```

void traduc3(nombre3,card)

```

```

unsigned int nombre3;
int card;

```

```

/* L'argument card 0 ou 1 selon que le mot suivant est nom ou adjectif
   il permet de respecter la syntaxe QUATRE-VINGTS, QUATRE-VINGT-MILLE
   QUATRE-VINGTS-MILLIONS (idem avec CENT) */

```

```

é

```

```

static char *unite°$=

```

```

é

```

```

" ","UN ","DEUX ","TROIS ","QUATRE ","CINQ ","SIX ","SEPT ","HUIT ",
"NEUF ","DIX ","ONZE ","DOUZE ","TREIZE ","QUATORZE ","QUINZE ",
"SEIZE ","DIX-SEPT ","DIX-HUIT ","DIX-NEUF "

```

```

è;

```

```

static char *dizaines°$=

```

```

é

```

```

"","","VINGT","TRENTE","QUARANTE","CINQUANTE","SOIXANTE","SEPTANTE",
"QUATRE-VINGT","NONANTE"

```

```

è;

```

```

/* SEPTANTE et NONANTE ne sont pas utilisés par le logiciel et ne figurent
   dans le tableau que pour faire plaisir à nos amis Suisses et Belges */

```

```

unsigned int nb_centaines,nb_dizaines,nb_unites;

```

```

/*=====
                               On calcule le nombre de centaines de dizaines et d'unités
=====*/

```

```

nb_centaines=nombre3/100;
nb_dizaines =(nombre3-(nb_centaines*100))/10;
nb_unites=nombre3-nb_centaines*100-nb_dizaines*10;

```

```

/*=====
                               On s'occupe des centaines:

```

```

Si pas de centaines on écrit rien ! Si une centaine on écrit CENT
Si supérieur à 1 on écrit le nombre de centaines suivi de CENT si le nom
est suivi d'un adjectif cardinal, suivi de CENTS s'il est suivi d'un nom
ou s'il est situé à la fin du nombre

```

```

if(nb_centaines > 0)

```

```

é

```

```

if (nb_centaines > 1) printf("%s",unite°nb_centaines$);

```

```

    if (card >= nb_dizaines >= nb_unites >= (nb_centaines==1)) printf("CENT ");
    else printf("CENTS ");
}

/*=====
On traite les dizaines et les unités:

Pour les valeurs entre 10 et 19 on augmente l'index du tableau des unités
de 10 de façon à écrire ONZE plutôt que DIX-UN !
On fait subir le même traitement pour SOIXANTE-DIX et QUATRE-VINGT-DIX
On traite le cas particulier de VINGT et VINGTS
On joint par ET les nombres des dizaines et le nombre UN de VINGT à SOIXANTE
inclus ainsi que pour SOIXANTE et ONZE au dessus on n'emploie pas de ET
On place un trait d'union entre les différentes parties de nombres
inférieurs à 100
=====*/

switch (nb_dizaines)
{
    case 0: printf("%s", unite°nb_unites$); break;

    case 1: printf("%s", unite°nb_unites+10$); break;

    case 7: é
        /* On écrit SOIXANTE */
        printf("%s", dizaines°6$);

        /* La liaison soit ET soit tiret */
        if (nb_unites == 1) printf(" ET ");
        else printf("-");

        printf("%s", unite°nb_unites+10$);
        é break;

    case 8: é
        printf("%s", dizaines°8$);

        /* Si pas d'unité ou pas de milliers S pluriel
        Si unité un tiret, si MILLE un espace */
        if (nb_unites == 0) if (card) printf(" ");
        else printf("S ");
        else printf("-");

        printf("%s", unite°nb_unites$);
        é break;

    case 9: é
        printf("%s-%s", dizaines°8$, unite°nb_unites+10$);
        é break;

    default: é
        printf("%s", dizaines°nb_dizaines$);

        if (nb_unites == 1) printf(" ET ");
        if (nb_unites > 1) printf("-");

        printf("%s", unite°nb_unites$);
        é break;
}

é

é /* fin de traduc3 */

```

```

/*=====
Conversion d'un nombre entre 1 et 999999999
=====*/
void traducn(long nombre)
{
    int nb_millions, nb_milliers, nb_unites;

    if (nombre < 1e9) /* On ne traite pas les milliards */
    {
        nb_millions = nombre / 1e6;
        nb_milliers = (nombre - (nb_millions * 1e6)) / 1e3;
        nb_unites = nombre - nb_millions * 1e6 - nb_milliers * 1e3;

        traduc3(nb_millions, 0); switch (nb_millions)
        {
            case 0: break;
            case 1: printf(" MILLION "); break;
            default: printf(" MILLIONS "); break;
        }

        switch (nb_milliers)
        {
            case 0: break;
            case 1: printf(" MILLE "); break;
            default: traduc3(nb_milliers, 1); printf(" MILLE ");
        }

        traduc3(nb_unites, 0);
        else printf("Nombre trop grand !!!");

        printf("\n");
    }
    /* fin de traducn() */
}

/*=====
PROGRAMME PRINCIPAL
=====*/

main()
{
    long nombre = 0;
    printf("Programme traducteur de nombres\nTapez 0 pour terminer\n");
    do
    {
        printf("NOMBRE :");
        scanf("%ld", &nombre);
        traducn(nombre);
    }
    while (nombre != 0);
}

```



*Ah ! mes amis ! Les vacances, le soleil, la plage...
Le bruit du groupe électrogène alimentant mon
APPLE GS et couvrant heureusement l'infernal bruit
des vagues et l'abominable cris des mouettes...
que tout cela est déjà loin...*

*Allez ! la prochaine fois, on travaille ! CRRR, CRRR,
CRRR, CRRR, CRRR...*

Claude AUBRY.

MINI-DOC

A l'affichage du catalogue, le préfixe du volume apparaît en haut à droite, en mode inverse.

Les titres des fichiers sont affichés sur 5 colonnes. Le nom de fichier inversé est celui concerné par les touches ESP et RETURN. ESP sélectionne les fichiers qui seront affectés par les commandes ultérieures. Une marque apparaît devant chaque fichier sélectionné.

Lorsque le nom de fichier inscrit en mode inverse est un sous-catalogue, RETURN envoie le catalogue de ce sous-volume. Lorsque ce nom inversé est un fichier binaire ou basique, l'exécution de ce dernier est lancée. Cette version ne comportant pas encore de gestion d'erreurs*, l'exécution de fichiers binaires non exécutables se traduira par des conséquences que je n'ose décrire...

TAB permet le retour au volume actuel.

La dernière ligne présente, de gauche à droite, les types de fichiers accessibles avec, en mode

inverse, les types sélectionnés, le statut d'accès du nom de fichier inversé et la taille de ce dernier.

Pomme ouverte T permet de modifier la sélection des TYPES : *flèches* déplacent, *ESP* (dé)sélectionne, *ESC* retourne à l'affichage du catalogue.

La première ligne décrit les commandes **Pomme ouverte** disponibles. Ces commandes sont les suivantes (du moins dans cette version) :

- **Pomme ouverte P** : Changement du préfixe. Le nouveau préfixe, indiqué par le chemin ou par sa localisation slot/drive révèle aussitôt ses secrets.
- **Pomme ouverte D** : Création d'un sous-volume dans le volume en cours.
- **Pomme ouverte V R et E** : Verrouillage, déverrouillage ou effacement des fichiers sélectionnés par ESP.

* Nous avons ajouté un **ONERR GOTO** à la ligne 506. Il évite un plantage en cas de changement de disquette.

```

100 GOTO 270                                     4544
105 WAIT - 16384,128,127: PRINT PEEK ( - 16384) - 128: POKE - 1636
    8,0: GOTO 105                                 F206
110 PRINT PEEK ( - 16287): HOME : GOTO 110       5C06
115 RE$ = "":RT = 0:PO = 1403                    8320
120 ON LEN (RE$) > 15 GOSUB 235: VTAB (1): POKE PO,RT + CT  7343
125 GET A$: PRINT ;:AA = ASC (A$): ON AA = 8 GOTO 175: ON AA = 21 GO
    TO 185: ON AA = 127 GOTO 195: ON AA = 13 GOTO 165:AA = AA * (1 -
    (RT = 0 AND AA > 46 AND AA < 58))           4464
130 IF AA < 46 OR AA = 47 OR (AA < 64 AND AA > 57) OR (AA < 97 AND AA
    > 90) OR AA > 122 THEN GOSUB 235: GOTO 120   E2FE
135 IF RT = LEN (RE$) THEN RE$ = RE$ + A$:RT = RT + 1: PRINT A$;; GO
    TO 120                                       32A4
140 IF RT = 0 AND LEN (RE$) = 1 THEN RE$ = RE$:RT = RT + 1: GOTO 160  95E5
145 IF RT = LEN (RE$) - 1 THEN RE$ = LEFT$ (RE$,RT) + A$:RT = RT +
    1: GOTO 160                                  CA19

```

MINI-DOC (suite)

```
150 IF RT = 0 THEN RE$ = A$ + RIGHT$ (
RE$, LEN (RE$) - RT - 1):RT = RT +
1: GOTO 160 9713
155 RE$ = LEFT$ (RE$,RT) + A$ + RIGHT$ (RE$, LEN (RE$) - RT - 1):RT
= RT + 1 B50E
160 VTAB (1): POKE 1403,CT: PRINT RE$;" "; GOTO 120 63DE
165 ON RE$ = "" GOTO 120:RE$ = LEFT$ (RE$,RT): RETURN 7337
170 GOSUB 235:RT = RT - 1:RE$ = LEFT$ (RE$, LEN (RE$) - 1): GOTO 160 EDE4
175 A$ = "": IF RE$ = "" OR RT = 0 THEN PRINT CHR$ (7); GOTO 120 1F43
180 RT = RT - 1: GOTO 120 7E8E
185 A$ = "": IF RT = LEN (RE$) THEN PRINT CHR$ (7); GOTO 120 4F65
190 RT = RT + 1: GOTO 120 248D
195 A$ = "": ON RT = 0 GOTO 175: IF RT = 1 AND LEN (RE$) = 1 THEN RE
$ = "":RT = 0: GOTO 160 2E8F
200 IF RT = LEN (RE$) THEN RE$ = LEFT$ (RE$,RT - 1):RT = RT - 1: GO
TO 160 BDED
205 IF RT = 1 AND LEN (RE$) > 1 THEN ER$ = RIGHT$ (RE$, LEN (RE$) -
1):AA = ASC ( LEFT$ (ER$,1)): ON AA > 46 AND AA < 58 GOTO 225:RE
$ = ER$:RT = RT - 1: GOTO 160 F843
210 RE$ = LEFT$ (RE$,RT - 1) + RIGHT$ (RE$, LEN (RE$) - RT):RT = RT
- 1: GOTO 160 ED5E
215 A$ = "":RT = RT + 1: IF RT = LEN (RE$) THEN PRINT CHR$ (7); 863C
220 GOTO 120 113E
225 GOSUB 235: GOTO 120 B4C2
230 GOTO 120 113E
235 VTAB (10): POKE PO,10: PRINT " CETTE IDENTIFICATION EST UN NOM DE
FICHER *PRODOS* " C371
240 VTAB (12): POKE PO,10: PRINT " TROIS REGLES D'OR SONT A RESPECTER
;....." 5758
245 VTAB (14): POKE PO,10: PRINT " - LE PREMIER CARACTERE EST UNE LET
TRE ! ....." F970
250 VTAB (15): POKE PO,10: PRINT " - LE NOM DE FICHER N'EXCEDE PAS 1
5 CARACTERES ! ....." 7506
255 VTAB (16): POKE PO,10: PRINT " - NE SONT VALIDES QUE LES LETTRES
LES CHIFFRES ET LE POINT !" 3952
260 FOR I = 5 TO 16: VTAB (I): POKE PO,10: PRINT " .....
.....": FOR J = 1 TO 1
00:SOUND = PEEK ( - 16336): NEXT : NEXT : RETURN CE71
265 PRINT "IDENTIFICATION DE L'ACQUISITION :"; GOSUB 115 36A3
270 A$ = "" EE79
275 D$ = CHR$ (4): PRINT D$"PR£3": PRINT BD30
280 REM *****
285 REM * TX$ : NOM DES FICHERS *
290 REM * BIP$: TYPE DE CHAQUE FICHER *
295 REM * TX : INDICE FICHER *
300 REM * TM : NOMBRE TOTAL DE FICHER DANS LE PREFIX *
305 REM * PL : PREMIERE LIGNE DU CATALOGUE *
310 REM * T$ : TYPES DES FICHERS AFFICHES *
315 REM * C$ : COMMAMNDES DISPONIBLES *
320 REM * AD : COMMUTAION AUXMAIN *
325 REM *****
330 : 003A
335 : 003A
340 REM *****
```

```

345 REM *   INITIALISATIONS   *
350 REM *****
355 :
360 TX = 110: DIM TX$(TX),BIP%(TX),LOCK%(TX),S(TX),BLOC%(TX):AD = -
      16300:ZERO = 0
365 PL = 2:DL = 21:HH = DL - PL + 1
370 T$(1) = "DIR":T$(2) = "BAS":T$(3) = "BIN":T$(4) = "TXT":T$(5) =
      "AWP":T$(6) = "USR":T$(7) = "SYS":T(1) = 1:T(2) = 1:T(3) = 1
375 C$(1) = "Préfixe":C$(2) = "Dossier":C$(3) = "BLoad":C$(4) = "Bsa
      ve":C$(5) = "":C$(6) = "":C$(7) = "Verrouiller":C$(8) = "dèveRrou
      iller":C$(9) = "Effacer"
380 LO$(0) = "Accès libre "":LO$(1) = "Accès protégé":
385 POKE 35,22: POKE 34,2: GOSUB 415: GOTO 435
390 :
395 REM *****
400 REM *   AFFICHAGES DES TYPES DE FICHIERS SELECTIONNES   *
405 REM *****
410 :
415 POKE 1024,65: VTAB (1): POKE 1403,2: PRINT "--commandes ";: FOR J
      = 1 TO 7: PRINT C$(J):;: PRINT " ";: NEXT
420 POKE 2000,65: VTAB (24): POKE 1403,2: PRINT "-Types :";: FOR J
      = 1 TO 7: GOSUB 425: PRINT T$(J):;: NORMAL: PRINT " ";: NEXT : RE
      TURN
425 IF T(J) THEN INVERSE : RETURN
430 RETURN
435 VTAB (23): POKE 1403,1: PRINT "Flèches déplacent; ESP sélectionne
      ; RETURN entre dans le dossier ou boote";
440 :
445 REM *****
450 REM *   LECTURE DU PREFIX   *
455 REM *****
460 :
465 PRINT D$"PREFIX": INPUT PR$: GOTO 475
470 ON LEN (PR$) = 1 GOTO 465: PRINT D$"PREFIX";PR$
475 VTAB (2): CALL - 868: POKE 1403,80 - LEN (PR$): INVERSE : PRINT
      PR$: NORMAL
480 :
485 REM *****
490 REM *   LECTURE DU CATALOGUE SAUVEGARDE DES TX$ TYPE$   *
495 REM *****
500 HOME
505 FOR TX = 1 TO TM:TX$(TX) = "":S(TX) = 0:BIP%(TX) = 0:LOCK%(TX) =
      0:BLOC%(TX) = 0: NEXT :TX = 0:TM = 0
506 ONERR GOTO 905
510 PRINT D$"OPEN"PR$",TDIR": PRINT D$"READ"PR$: INPUT L1$: INPUT L2$
      : INPUT L3$
515 INPUT L4$: IF L4$ = "" THEN 525
520 TYP$ = MID$ (L4$,18,3): GOSUB 1030:TX$ = MID$ (L4$,2,16): ON TX
      $ = "STARTUP " GOTO 515:TX = TX + 1:LOCK%(TX) = ( LEFT$ (
      L4$,1) = "*" ):BLOC = VAL ( MID$ (L4$,25,4)):BLOC%(TX) = BLOC:BIP
      %(TX) = 1:TX$(TX) = TX$: GOTO 515
525 TM = TX: PRINT D$"CLOSE"PR$
530 :
535 REM *****

```

(suite page 54)

MINI-DOC (suite)

```
540 REM * AFFICHAGE DES TX$ SUR 5 COLONNES
      *
545 REM *****
      *****
550 : 003A
555 FOR TX = 1 TO TM AE90
560 L = PL + TX - HH * ( INT ((TX - 1) / HH)):C = 16 * INT ((TX - 1)
      / HH) + 1: POKE 36,C: VTAB (L): PRINT TX$(TX); 8C43
565 NEXT TX A72E
570 : 003A
575 REM *****
580 REM * DEPLACEMENT ET SELECTION PRISE DES POMMES OUVERTES*
585 REM *****
590 : 003A
595 POKE 1403,1: VTAB (2):C = 1:L = PL + 1: IF TM = 0 THEN VTAB (15)
      : PRINT " Pas de fichier des types selectionnés dans ce volume":
      TX = 1:BIP%(TX) = 1: GOTO 605 F382
600 GOSUB 665 2F51
605 : WAIT - 16384,128,127:A = PEEK ( - 16384) - 128: POKE - 16368
      ,0: ON PEEK ( - 16287) > 127 GOTO 760: ON A < 0 GOTO 605: ON A =
      32 GOTO 1085: GOSUB 670 D199
610 IF A = 9 THEN PR$ = LEFT$ (PR$, LEN (PR$) - 1): ON RIGHT$ (PR$,
      1) = "/" GOTO 470: GOTO 610 32C0
615 IF A = 8 THEN C = C - 1:C = (C < 1) + C 13F8
620 IF A = 21 THEN C = C + 1:C = - (C > 5) * 1 + C: IF HH * (C - 1)
      + (L - PL) > TM THEN C = C - 1 C505
625 IF A = 10 THEN L = L + 1:L = (L = DL + 2) * (PL + 1) + (L < DL +
      2) * L: IF HH * (C - 1) + (L - PL) > TM THEN L = PL + 1 2995
630 IF A = 11 THEN L = L - 1:L = (L = PL) * (DL + 1) + (L > PL) * L:
      IF HH * (C - 1) + (L - PL) > TM THEN L = PL + TM - INT (TM / HH)
      * HH 1060
635 ON A = 13 GOTO 705: GOTO 600 50EB
640 : 003A
645 REM *****
650 REM * AFFICHAGE INVERSE NORMAL INVERSE NORMAL INVERSE...*
655 REM *****
660 : 003A
665 TX = (HH * (C - 1)) + L - PL: POKE 2003 + 2 * BIP%(TX),85: VTAB (
      24): POKE 1403,50: PRINT LO$(LOCK%(TX));: PRINT " BLOCS :";BLOC%
      (TX);: CALL - 868: INVERSE : GOTO 675 A63A
670 NORMAL : POKE 2003 + 2 * BIP%(TX),160 4ED9
675 POKE 1403,1 + (16 * (C - 1)): VTAB (L): PRINT LEFT$ (TX$(TX),15)
      ;: NORMAL : RETURN B7B8
680 : 003A
685 REM *****
690 REM * REALISATION DE LA COMMANDE SUR LES FICHIERS SELECTIONNES *
695 REM *****
700 : 003A
705 TX = (HH * (C - 1)) + L - PL:TX$ = TX$(TX) 85D5
710 FOR I = 16 TO 1 STEP - 1: ON MID$ (TX$,I,1) < > " " GOTO 715:T
      X$(TX) = LEFT$ (TX$,I - 1) E51A
715 NEXT I 9DCB
720 IF BIP%(TX) = 3 THEN POKE 48984,0: PRINT D$"-";TX$(TX) 0B6C
725 IF BIP%(TX) = 2 THEN PRINT D$"RUN";TX$(TX) D9D3
730 IF BIP%(TX) = 1 THEN PR$ = PR$ + TX$(TX) + "/": GOTO 470 771B
```

```

735 END 0180
740 REM *****
745 REM * ANALYSE ET EXECUTION DES POMMES OUVERTES *
750 REM *****
755 : 003A
760 IF A = 84 THEN C$ = " ESP sélectionne (-) déplace ESC sort "; VTA
    B (24): POKE 1403,79 - LEN (C$): PRINT C$;: GOTO 810 B9F0
765 IF A = 80 THEN 905: REM P F588
770 IF A = 68 THEN 945: REM D 0A92
775 IF A = 86 THEN B$ = "LOCK":BF$ = "Verrouill": GOTO 955: REM V FAD9
780 IF A = 82 THEN B$ = "UNLOCK":BF$ = "déveRrouill": GOTO 955: REM R 5437

785 IF A = 69 THEN B$ = "DELETE":BF$ = "Effaç": GOTO 955: REM E 040E
790 GOTO 760 FC48

795 REM *****
800 REM * SELECTION DES TYPES DE FICHER A AFFICHER *
805 REM *****

810 GOSUB 670 3B4D
815 I = 1:K = 1 11D0
820 VTAB (24) 3D59
825 POKE 2003 + 2 * I,85 2F24
830 WAIT - 16384,128,127:A = PEEK ( - 16384) - 128: POKE - 16368,0
    : ON A = 32 GOTO 860 398F
835 IF A = 8 THEN K = I - (I > 1): GOTO 855 8408
840 IF A = 21 THEN K = I + (I < 7): GOTO 855 523A
845 IF A = 27 THEN NORMAL :C$ = " .....
    """: VTAB (24): POKE 1403,79 - LEN (C$): PRINT C$;: POKE 2003 +
    2 * I,160: POKE 34,2: HOME : GOTO 500 CF6C
850 GOTO 830 2146
855 POKE 2003 + 2 * I,160:I = K: GOTO 825 3B70
860 T(I) = ABS (T(I) - 1): IF T(I) = 0 THEN NORMAL : GOTO 870 B985
865 INVERSE 3D9E
870 POKE 1403,8 + 4 * I: PRINT T$(I); 5435
875 K = I + (I < 7): GOTO 855 5155
880 : 003A
885 REM *****
890 REM * POMME p *
895 REM *****
900 : 003A
905 VTAB (1): POKE 1403,3: PRINT C$(1);" P pour prefixe , S pour port
    et lecteur ";: CALL - 868 5EAA
910 GET A$: PRINT ;: ON A$ = "P" GOTO 915: ON A$ = "S" GOTO 920: GOTO
    910 B7D2
915 HOME : VTAB (1): POKE 1403,1: PRINT "Nouveau préfixe :/ .....
    """:;: CALL - 868:CT = 19:: GOSUB 115:PR$ = "/" + RE$ +
    "/": GOSUB 415: GOTO 475 673A
920 HOME : VTAB (1): POKE 1403,1: PRINT "Nouveau port :";" "Nouveau
    drive :";:PR$ = "": CALL - 868: POKE 1403,16 B10F
925 GET A: PRINT A;:PR$ = PR$ + ",S" + STR$ (A) 12CE
930 POKE 1403,33 3213
935 GET A: PRINT A;:PR$ = PR$ + ",D" + STR$ (A) 72BF
940 GOSUB 415: PRINT D$"PREFIX";PR$: GOTO 465 DB3D
945 : 003A ▶

```


MINI-DOC (fin)

950 HOME : VTAB (1): POKE 1403,1: PRINT "Sous catalogue à créer :";: CALL - 868:CT = 31: GOSUB 115: PRINT D\$; "CREATE";RE\$: GOSUB 415:: GOTO 475	58A3 2F97 072D 3605 019F
955 HOME 960 FOR I = 1 TO TM 965 IF S(I) = 0 THEN 995 970 A\$ = TX\$(I) 975 FOR J = 16 TO 1 STEP - 1: ON MID\$(A\$,J,1) < > " " GOTO 980:A\$ = LEFT\$(A\$,J - 1) 980 NEXT J 985 VTAB (15): POKE 1403,15: PRINT BF\$;"age de ";A\$;: CALL - 868 990 PRINT D\$;B\$;A\$: PRINT ;:S(I) = 0 995 NEXT I 1000 GOSUB 670: GOTO 500 1005 : 1010 REM ***** 1015 REM * NE CHARGE QUE LES FICHIERS DE TYPE SELECTIONNE * 1020 REM ***** 1025 : 1030 I = 1 1035 IF T(I) = 0 THEN I = I + 1: ON I = 8 GOTO 1050: GOTO 1035 1040 IF T\$(I) = TYP\$ THEN RETURN 1045 I = I + 1 1050 IF I = 8 THEN POP : GOTO 515 1055 GOTO 1035 1060 : 1065 REM ***** 1070 REM * ESP SELECTIONNE DES FICHIERS * 1075 REM ***** 1080 : 1085 TX = (HH * (C - 1)) + L - PL:TX\$ = TX\$(TX) 1090 IF S(TX) THEN POKE 1403,(16 * (C - 1)): VTAB (L): PRINT CHR\$(1 60);:S(TX) = 0: GOTO 605 1095 POKE AD + 1,0:F = PEEK (41) * 256 + PEEK (40): POKE F + 8 * (C - 1),68: POKE AD,1 1100 S(TX) = 1: GOTO 605	D7E3 A3CC 696B 6BB9 9DCB 28C7 003A 003A 634A BD18 FC25 805B E1E3 4874 003A 003A 85D5 A9FF 4D41 E6D1

- Vous pouvez recopier tous les programmes de **TREMLIN MICRO...**
- Vous pouvez également vous offrir notre **disquette n°17...**
- Vous disposerez ainsi de toutes vos routines (ProDOS et DOS)... mais aussi de la **disquette gratuite** que vous offrent Yvan KOENIG (pour le contenu) et TREMLIN MICRO (pour le support).
- Vous apprendrez par ailleurs tout ce qu'il faut savoir sur cette seconde disquette (**gratuite** pour celles et ceux qui commanderont la **disquette n°17**).

Affichez

par Yvan KOENIG

la TRÈS HAUTE RÉOLUTION sur votre GS

Suite au courrier d'un lecteur de *Tremplin Micro*, j'ai été amené à étudier l'affichage Très Haute Résolution du GS... alors que le graphisme informatique est loin de me passionner. Je vous livre le fruit de mes cogitations. L'utilitaire que tous les possesseurs de GS connaissent est GS.PAINT. La version 1.1 de ce programme (demandez-la à votre concessionnaire si ce n'est déjà fait) est capable de créer 3 types de fichiers images.

1 Fichier ECRAN

Fichier ECRAN contient, sous forme brute, la totalité du tampon THGR, c'est-à-dire la zone mémoire \$E12000-\$E19FFF. GS.PAINT crée ce fichier sous le type \$C1. C'est à ma connaissance le seul type de fichier créé par DELUXE.PAINT.

Il comporte 32756 octets. Les 200 * 160 = 32000 premiers correspondent aux 64000 pixels de l'écran (mode 320). Viennent ensuite 200 octets *pointeurs*, 56 octets *vides* puis 16 * 32 octets définissant les palettes (une seule est utilisée avec GS.PAINT).

Si vous souhaitez récupérer une image THGR incluse dans l'un des *innombrables* programmes affichant dans ce mode afin de la manipuler avec GS.PAINT par exemple, ou de l'afficher, depuis le BASIC, à l'aide de THGS, voici la procédure à suivre :

- Ctrl Reset pour sortir du programme original.
- CALL - 151 pour passer en mode moniteur.
- C029:81 pour forcer l'affichage de l'image, ce qui permet de vérifier son intégrité.
- C029:21 pour rétablir le mode texte.
- Booter une disquette ProDOS 8 bits (ProDOS 1.1.1, ProDOS 2.0, P8) par 5 Ctrl P.
- Puisque nous nous retrouvons sous APPLESOFT, CALL - 151 pour accès au Moniteur.

- C029:81 affiche l'image ; on peut ainsi contrôler une nouvelle fois son intégrité.
- C029:61 permet de revenir en mode texte avec le mode *adressage linéaire* actif ce qui est indispensable pour pouvoir déplacer le bloc image (cf description du contenu de NEW-VIDEO).
- 00/1000 > E1/2000.E1/9FFFFM transfère vers \$00/1000 le buffer THGR situé de \$E1/2000 jusqu'à \$E1/9FFF.
- Q ramène au BASIC.
- BSAVE Pathname, A\$1000,E\$8FFF crée un fichier BINaire que GS.PAINT pourra exploiter.

On pourrait plus simplement utiliser ma commande externe PACK monimagé qui sauverait le tampon THGR sous forme d'un fichier Ecran Compressé (voir plus loin), mais GS.PAINT ne sait pas, à ce jour, exploiter ce type de fichier.

2 Fichier type PAINT

Fichier type PAINT contient, sous forme compressée, les informations correspondant à une page écran complète. Les 32 premiers octets sont la définition de LA palette retenue... (suite page 58)

(oui, GS.PAINT n'utilise qu'une palette par dessin). Nous avons ensuite 512 octets correspondant à la définition des motifs. Ces premiers octets sont stockés tels quels. Ensuite, nous trouvons les octets image à proprement parler, compressés à l'aide d'un outil contenu dans la ROM du GS, j'ai nommé PackBytes, outil n°\$26 de la famille n°3 : Miscellaneous Tools. Ces fichiers PAINT sont de type \$C0, AuxType \$0000.

3 Fichier Ecran Compressé

Type \$C0, AuxType \$0001. Je ne connais pas de programme commercial utilisant ce type de fichier qui est le plus compact de tous. C'est l'ensemble du tampon THGR (200 lignes image, pointeurs et palettes) qui est compacté par __PackBytes. La commande externe PACK a été écrite pour générer ce type de fichiers. Le fichier source se compose de K.PACK.S, RELOPRO.S et PACK.S.

4 Fichier type APPLE

Il s'agit d'un fichier de type \$C0, AuxType \$0002 et il a été défini par APPLE pour devenir le mode standard de stockage car sa structure modulaire ouvre des perspectives très souples.

Il est constitué de plusieurs blocs, MAIN impératif, PATS et SCIB facultatifs (GS.PAINT 1.1 n'utilise pas de bloc SCIB).

Pour suivre les informations ci-dessous, je vous suggère de charger le fichier IAPPLE par BLOAD IAPPLE,\$C0,\$2000.

- 2000.2003 longueur du bloc MAIN sur 4 octets (ici \$0000114B).
- 2004.2008 Str 'MAIN'.
- 2009.200A "master mode" 2 octets définis par QuickDraw (ici 0000).
- 200B.200C nombre de points lumineux (pixels) par ligne (ici 320).
- 200D.200E nombre de palettes (ici 16 bien que seule la 1^{ère} soit utilisée par GS.PAINT).
- 200F-220E NbPalettes de 32 octets.

- 200F.2210 nombre de lignes stockées (396 pour GS.PAINT). Bien entendu, on ne peut en afficher que 200 à la fois.
- 2211-2840 NbLignes fois 4 octets descripteurs de lignes, les 2 premiers octets de chaque groupe représentant la longueur de la ligne compactée.
- 2841-314A les octets définissant le contenu de l'image, comprimé ligne par ligne par l'outil __PackBytes.

Vient ensuite le bloc PATS :

- 314B.314E longueur du bloc "PATS" (ici \$0000020B).
- 314F.3153 Str "PATS"
- 3154.3155 nombre de motifs sur 2 octets (ici 16).
- 3156.3355 16 motifs de 32 octets (ce que vous voyez en bas de l'écran de travail de GS.PAINT).

Pourrait venir ensuite le bloc SCIB :

- Longueur du bloc sur 4 octets.
- Str "SCIB".
- Données motifs 1^{er} plan.
- Données motifs fond.
- Données motifs cadre (FRAME).

Je n'ai hélas aucun détail sur ces données que je n'ai encore jamais rencontrées.

La richesse des informations contenues dans ce type de fichier permet toutes les acrobaties. La commande THGS en est un exemple, qui permet, entre autres choses, d'afficher, depuis le Basic, une image de ce type en commençant à une ligne redéfinie par la simple commande THGS monimage, B begline (l'option (.A0) permet de bénéficier d'une routine de pause suivie de la restauration du mode de fonctionnement initial).

La documentation technique du GS donne les informations essentielles sur le mode de codage lors de la compression de l'image (dans les fichiers PAINT ou APPLE, l'image est compactée ligne par ligne).

Nous avons des octets *descripteurs* et des octets *objets*.

Il existe quatre types de descripteurs :

- 00xxxxxx (avec xxxxxx de 0 à 63) indique qu'il faut afficher les xxxxxx + 1 octets qui suivent.
- 01xxxxxx (avec xxxxxx = 2,4,5,6) indique qu'il faut afficher xxxxxx + 1 fois l'octet qui suit.

- 10xxxxxx (avec xxxxxx de 0 à 63) indique qu'il faut afficher xxxxxx + 1 fois le groupe de 4 octets qui suit.
- 11xxxxxx (avec xxxxxx de 0 à 63) indique qu'il faut afficher (xxxxxx + 1) * 4 fois l'octet qui suit.

Ajoutons à ces informations quelques détails sur un commutateur essentiel : NEWVIDEO \$C029.

- bit7 1 = déconnecte mode vidéo *lle*, connecte mode Super Haute Résolution.
- bit6 1 = adressage linéaire du tampon vidéo par opposition au mode "entrelacé" bien connu des habitués du mode HGR.
- bit5 0 = DHGR couleur, 1 = DHGR monochrome.
- bit4,3,2,1 OBLIGATOIREMENT mis à 0.
- bit0 manipulation INTERDITE à l'utilisateur. Seul le MONITOR peut le faire.

Pour afficher un écran de type non compressé, pas de problème. On lit 256 octets du fichier, on les expédie dans le tampon THGR et on recommence jusqu'à la fin du fichier.

Si l'on est en présence d'un Ecran compressé (\$C0.0001), il suffit de lire et de décompresser en continu.

Pour les deux autres modes compressés traités, ça se complique. Dans un premier temps, il faut lire et transférer la palette. Ensuite, on utilise Set_Mark pour avancer dans le fichier jusqu'à la zone "octets compressés" qui nous intéresse. Dans le cas des fichiers APPLE, on récupère dans le fichier lui-même le nombre de palettes et le nombre de lignes sauvées afin d'exploiter correctement les informations disponibles. Pour décompresser, on utilise l'outil UnPackBytes \$2703 (\$27 dans miscellaneous tools).

Après ce décompactage, il nous reste à garnir la zone \$E19D00-E19DFF qui contient les 200 descripteurs de lignes et 56 octets de remplissage qu'APPLE demande de mettre à 0 (tout au moins pour la version actuelle du GS). Pour des images de la 1^{ère} génération, il nous suffit de mettre toute cette zone à 0 car nous ne faisons pas appel aux possibilités évoluées de la machine (palettes multiples ou changement de palette par l'intervention des interruptions).

J'ai mis tout cela en œuvre sous forme d'une commande externe ProDOS qui donne un remarquable confort d'utilisation.

En mode direct, THGS monimage,A0 charge et affiche l'image indiquée et attend une frappe du clavier pour revenir au mode d'affichage initial. Cette commande fonctionne aussi en mode programme : PRINT CHR\$(4)"THGS monimage,A0".

Si l'on s'abstient de spécifier le paramètre (,A0), il n'y a pas de pause ni de restauration du mode affichage initial qui reste alors à votre diligence.

Le fichier source, sous MERLIN.PRO, comporte les trois modules désormais classiques :

- K.THGS.S = module d'appel.
- RELOPRO.S = module relogeur.
- THGS.S source de la partie active qui s'attribue quatre pages mémoire.

Le programme DEMO.THGS sauve l'état de NEW-VIDEO, affiche une image avec pause puis, après votre intervention au clavier, affiche en séquence plusieurs images. Au passage, on utilise CHGPALETTE, une petite routine implantée en page 3 qui *récupère* la palette de l'image active pour l'ajouter aux 2 palettes intégrées d'origine dans la routine. Ensuite, une boucle affiche alternativement ces 3 palettes.

En outre, il est aussi fait appel à THGS pour l'affichage d'une image APPLE à partir d'un numéro de ligne non nul, et à PACK pour créer un fichier Ecran compacté (\$C0 0001).

Cette courte routine démo met en évidence un inconvénient des fichiers de type écran. Qu'ils soient Normaux ou Compactés, le bloc palettes est chargé en dernier. Si l'on est déjà en mode visualisation pendant le chargement, on voit apparaître l'image dans une gamme de couleurs qui provient de l'image précédente. Il y a également de ce fait risque de problèmes si les pointeurs prévoient des interruptions.

Comme à l'accoutumée, la mise en palce de THGS et PACK est faite par le programme STARTUP.

Je me permets d'attirer votre attention sur les solutions adoptées pour l'écriture du programme. RELOPRO ne sachant pas reloger du code 65C816, tout ce qui est spécifique à ce microprocesseur a été regroupé dans une courte routine implantée dans la zone data du programme qui n'est pas manipulée par RELOPRO.

Le catalogue de BASIC.SYSTEM n'affichait pas AuxType pour les fichiers de type "\$xx", avec un petit patch ; cette lacune est comblée et vous pouvez identifier les fichiers images lors du CATALOG.

BEE0: JSR A612, LDY E4E, LDA E08, RTS A4F2: JSR BEE0, BNE A516

Il était impossible de faire figurer les sources, les objets et la démo sur la disquette *Tremplin Micro* n°17. Il a été décidé de réaliser une disquette spéciale.

Plutôt que de remplir celle-ci en multipliant les images, j'ai ajouté la commande TRANS qui permet de recoder une image HGR ou DHGR préalablement chargée en une image THGR noir et blanc en mode 640. Le fichier généré est alors un écran compacté (type \$C0.0001). La syntaxe d'accès à la commande est simple :

- TRANS monimage (,Sslot)(,Ddrive) crée le fichier correspondant à une image HGR.
- TRANS monimage, titrebidon (,Sslot)(,Ddrive)

crée le fichier correspondant à une image DHGR (voir le programme ESSAI).

La disquette double face est pleine et il ne reste pas de place pour PRODOS et BASIC.SYSTEM. Il vous faudra démarrer à partir d'une autre disquette, faire PREFIX THGS1 puis RUN STARTUP.

Je remercie Xavier SCHOTT qui m'a procuré les spécifications officielles des divers formats image, et Alexandra CLEMENT qui m'a transmis les images ASTRONAUTS, COASTLINE et WINDYCOVE. IAPPLE est une facétie de mon fils qui connaît mon allergie à un groupe de trois lettres bien connu, et les autres images ont été extraites de diverses disquettes APPLE ou VERSION.SOFT. **Yvan KOENIG.**

THGS

ATTENTION !

DISQUETTE TREMLIN MICRO YVAN KOENIG

est adressée automatiquement et **GRATUITEMENT** à tous les lecteurs **abonnés** aux **DISQUETTES TREMLIN MICRO**, mais n'oubliez pas qu'elle est spécifique GS. Elle sera par ailleurs jointe à toute commande de la disquette d'accompagnement n°17 (inutile de le préciser lors de votre commande).

Vous pouvez aussi obtenir **THGS** contre une modeste participation aux frais (voir votre bon de commande, à la fin du journal).

POURQUOI N'AVONS-NOUS PAS PUBLIÉ CETTE BATTERIE DE PROGRAMMES ?

Pour deux raisons faciles à comprendre :

- Ils auraient occupé un nombre de pages impressionnant. Ne présenter que les codes se serait révélé indigeste. Publier les sources n'aurait rien apporté aux possesseurs d'Apple II, IIe et IIc.
- Ils sont destinés aux seuls utilisateurs de GS, machine à laquelle *TREMLIN MICRO* a déjà consacré bon nombre de pages.

ALLONS-NOUS CONTINUER DANS CETTE VOIE ?

Il suffit de feuilleter le magazine pour constater que *TREMLIN MICRO* ne se contente pas de proposer des programmes sur disquettes, mais consacre la presque totalité de la revue à de nombreux programmes et routines.

Un grand sacrifié : **le courrier des Lecteurs...** pour lequel un numéro entier ne suffirait pas !

G.-H.

Cette routine ne présenterait qu'un intérêt secondaire si elle n'aidait pas le néophyte à comprendre comment utiliser le MLI (Machine Language Interface), partie centrale de PRODOS.SYSTEM. On remarquera que l'affichage des éventuelles erreurs n'y est pas traité.

MLI

PRÉPARATION

```

300 : 20 58 FC JSR FC58
303 : A9 5C LDA $5C
305 : A0 03 LDY £03
307 : 20 3A DB JSR DB3A
30A : 20 6F FD JSR FD6F
30D : 8A TXA
30E : F0 F0 BEQ 0300
310 : 8D 9A 03 STA 039A
313 : BD 00 02 LDA 0200,X
316 : 29 7F AND £7F
318 : 9D 9B 03 STA 039B,X
31B : CA DEX
31C : 10 F5 BPL 0313
  
```

HOME efface l'écran.

Adresse de la chaîne "TITRE: ".

STROUT affiche cette chaîne.

Entrée de GETLIN sans astérisque.

Le nombre de caractères est dans X. On le passe dans A.

S'il est nul, retour à la saisie.

Longueur du mot stockée (voir plus loin).

Chaîne installée où le MLI saura la lire (grâce à la liste des paramètres).

GET FILE INFO (\$C4)

```

31E : 20 00 BF JSR BF00
321 : C4 C4
322 : 78 03 0378
324 : B0 DA BCS 0300
326 : A9 64 LDA £64
328 : A0 03 LDY £03
32A : 20 3A DB JSR DB3A
32D : AE 7D 03 LDX 037D
330 : AD 7E 03 LDA 037E
333 : 20 24 ED JSR ED24
336 : A9 6E LDA £6E
338 : A0 03 LDY £03
33A : 20 3A DB JSR DB3A
33D : AE 80 03 LDX 0380
340 : AD 81 03 LDA 0381
343 : 20 24 ED JSR ED24
  
```

Saut au point d'entrée du MLI.

\$C4 : code de GET FILE INFO.

Adresse de la liste des paramètres.

Si retenue à 1, erreur et retour au point de départ.

Adresse de la chaîne... "Ad/Long : "

Aussitôt affichée par STROUT.

On lit ici l'adresse de chargement d'un fichier binaire (longueur si c'est un fichier texte).

LINPTR affiche cette adresse ou longueur en (déc.).

Adresse de la chaîne "Blocs: ".

Aussitôt affichée par STROUT.

Nombre de blocs retrouvé

dans la liste des paramètres... et affiché par LINPTR.

OPEN (\$C8)

```

346 : 20 00 BF JSR BF00
349 : C8 C8
34A : 8A 03 038A
34C : 90 01 BCC 034F
34E : 60 RTS
  
```

Nouvel appel au MLI.

\$C8 est le code de OPEN (ouvrir le fichier).

Adresse de la liste des paramètres d'OPEN (3).

Si retenue à 0 (pas d'erreur), on saute.

Terminé.

(suite page 62)

34F : 20 00 BF JSR BF00
352 : CA CA
353 : 90 03 0390

READ (\$CA)

Nouvel appel au MLI.
\$CA est le code de READ (on va lire 512 octets maxi).
Adresse de la liste des paramètres de READ (4).

355 : 20 00 BF JSR BF00
358 : CC CC
359 : 98 03 0398
35B : 60 RTS

CLOSE (\$CC)

Dernier appel au MLI.
\$CC est le code de CLOSE (fermeture de fichier).
Adresse de l'unique paramètre.
Terminé.

35C : 54 49 54 52 45 3A 20 00 : TITRE: •
364 : 41 64 2F 4C 6F 6E 67 3A 20 00 : Ad/Long: •
36E : 20 20 42 6C 6F 63 73 3A 20 00 : ..Blocs: •

CHAÎNES

378 : 0A
379 : 9A 03
37B : 00
37C : 00
37D : 00 00
37F : 00
380 : 00 00
382 : 00 00
384 : 00 00
386 : 00 00
388 : 00 00

PARAMÈTRES GET FILE INFO

Nombre de paramètres (10).
Nom d'accès (adresse).
On lira ici le code d'accès.
Le type de fichier.
Le type auxiliaire (adresse de chargement ou longueur).
Le type sauvegarde.
Nombre de blocs occupés.
Date modifiée.
Heure modifiée.
Date de création.
Heure de création.

38A : 03
38B : 9A 03
38D : 00 20
38F : 00

PARAMÈTRES OPEN

Nombre de paramètres (3).
Adresse du nom d'accès.
Buffer (ici \$2000).
Numéro de référence.

390 : 04
391 : 01
392 : 00 20
394 : 00 00
396 : 00 00

PARAMÈTRES READ

Nombre de paramètres (4).
Numéro de référence.
Adresse du buffer.
Nombre d'octets à lire.
Nombre d'octets lus.

398 : 01 01

PARAMÈTRES CLOSE

Un seul paramètre : le numéro de référence.

39A : 00 00 etc. jusqu'à \$3CF

STOCKAGE DE LA CHAÎNE-TITRE

Le premier octet contiendra la longueur de la chaîne, les suivants ses caractères ASCII (bit 7 à 0).

Pour essayer cette routine, cherchez un titre dans votre catalogue de disquette (MLI par exemple) et tapez-le à la suite de titre. Normalement, si vous faites ensuite CALL - 151 puis 2000L, vous obtiendrez la liste du programme MLI. Vous pourrez ensuite essayer avec un fichier TXT et, à la même adresse, vous aurez les 512 premiers octets de ce fichier... Après utilisation, amusez-vous à contrôler les résultats dans la liste des paramètres (Adresse ou longueur en \$37D.37E, nombre de blocs en \$380.381... par exemple). ■

C converti

Convertir un nombre décimal en hexa, puis en octal n'est pas d'une grande simplicité en Basic. En langage **C**, cela devient tellement simple qu'on aurait tort de s'en priver. Parcourez plutôt les lignes 44 et 45 de ce programme... et vous aurez tout compris :

- %x retourne la valeur de la variable en hexa
- %o retourne cette même valeur en octal.

```

1 main ()
2 é
3 int c, s=20;
4 long x;
5
6 /* TRAVAILLONS AVEC UN ECRAN VIERGE */
7
8 CL:
9     printf("çf");
10
11 /* ET SEPARONS CHAQUE PHASE PAR UNE LIGNE */
12
13 LG:
14     --s;
15     if(s > 0)é
16         printf("_____");
17         goto LG;
18
19 /* DEUX PETITS RETOURS LIGNE */
20
21     è
22     printf("çnçn");
23     s = 20;
24
25 /* SAISIE DE VOTRE NOMBRE */
26
27     SA:
28     printf("TAPEZ LE NOMBRE A CONVERTIR: ");
29     scanf("%1d",&x);
30     if(x >= 65536)é
31         printf("PAS PLUS DE 65535 S'IL
32         VOUS PLAIT!çn");
33         goto SA;
34 /* SI C'EST UN ZERO, C'EST FINI */
35
36     è
37     if(x == 0)é
38         printf("MERCI DE VOTRE ATTEN
39         TION! A BIENTOT!çn");
40         exit(0);
41 /* CONVERSION EN HEXA, PUIS OCTAL ET RETOUR */
42
43     è
44     printf("Ce qui fait, en hexadécimal : $%xçn",
45     x);
46     printf("ou encore en octal : %oçn", x);
47     goto LG;

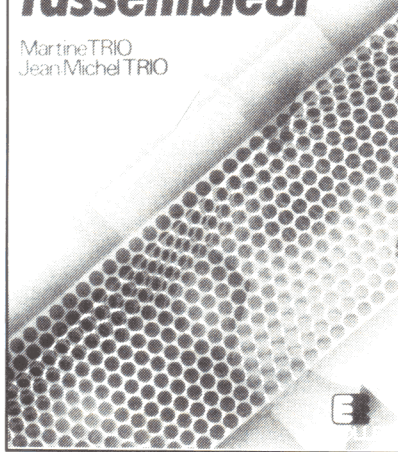
```


Votre bibliothèque INFORMATIQUE

par NESTOR

Utiliser MS-DOS à l'aide de l'assembleur

Martine TRIO
Jean-Michel TRIO



• UTILISER MS-DOS À L'AIDE DE L'ASSEMBLEUR (Martine et J.-Michel TRIO)

Qui n'a pas déploré son incapacité à allouer dynamiquement de la mémoire ou, plus simplement, à ne pas pouvoir laisser certaines parties d'un programme d'application en mémoire de masse ? Vous... peut-être. S'il vous arrive de travailler avec MS.DOS, le livre de Martine et de Jean-Michel TRIO saura éclairer votre lanterne sur bien des points. Vous apprendrez tout sur la structure d'un programme *.EXE sur celle d'un programme *.COM, sur le chargement et l'exécution d'un programme ou d'un overlay, etc.

Quant à la gestion des volumes disquettes/disque dur, elle deviendra pour vous un jeu d'enfant.

Chaque thème est développé de façon méthodique et pratique, sans fioritures inutiles, mais avec de bons exemples (plus de 130 pages !). Un seul regret : sauf erreur de ma part, il ne semble pas qu'une disquette d'accompagnement soit prévue. Dommage ! j'aurais bien aimé tester tout cela sans passer par de fastidieuses heures de saisie !

ÉYROLLES - 61, boulevard St Germain 75005 PARIS
Tél. : 46.34.21.99

• LE LANGAGE MACHINE SUR PC (DATA BECKER, Schieb)

La préface de ce gros bouquin (590 pages) devrait faire plaisir à Guy-Hachette qui, dans des termes voisins, rappelle régulièrement, aux lecteurs de *Tremplin Micro* que le langage machine n'est pas aussi compliqué que l'affirment... les professionnels. Du 65C02 et du 65C816 au 8088 et au 80286, il n'y a que... le premier pas qui coûte ! Si vous avez la possibilité de "jouer" avec un PC, ne manquez pas, bouquin ouvert, d'en découvrir tous les secrets ! Je vous promets de longues heures de veille, mais une inégalable jouissance intérieure (à mon âge, on fait avec ça qu'on a, pas vrai?).

EDITIONS MICRO APPLICATION
13, rue Sainte-Cécile 75009 PARIS.



• DISQUETTE ET DISQUE DUR SUR PC (DATA BECKER, Bomanns)

J'avoue connaître (assez bien) mon Apple chéri et sa disquette (DOS ou ProDOS). Pour ce qui est de son disque dur, c'est encore au-dessus de mes modestes moyens. Par contre, même s'il m'arrive assez souvent de pianoter sur un compatible PC, j'avoue que — avant la lecture, pardon le survol, de ce livre — j'ignorais à peu près tout des sujets clairement traités par l'auteur, exemples à l'appui. Cela va de la des-

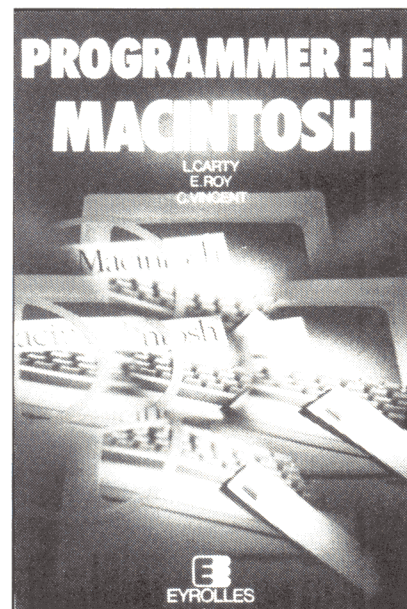
cription des supports (sans surprise pour moi) à la protection des données (récupérer un fichier abîmé), en passant par l'organisation de la mémoire de masse... entre autres choses. Fait intéressant : une disquette est jointe au livre et comprise dans son prix de vente (269 F pour 260 pages), ce qui m'a paru très pratique.

EDITIONS MICRO APPLICATION
13, rue Sainte-Cécile 75009 PARIS.

• PROGRAMMER EN MACINTOSH (L. Carty, E. Roy, C. Vincent)

Macintosh, enfant chéri d'Apple, n'a pas été conçu à l'intention des "petits" programmeurs de ma génération. C'est du prêt-à-porter informatique de bonne facture, fait pour être animé par quelque obéissante souris. A me lire, on croirait que j'ai une dent contre cette pomme-là. C'est faux. Foi de Nestor, j'en ai même un spécimen dans mon bureau, payé de mes propres deniers... mais je ne le programme pas... ou peu. Pourtant, Borland et des bouquins comme celui-ci me donnent envie de me remettre à l'ouvrage. Le Mac vaut bien un GS, pas vrai ? En tout cas, je suis certain que *Programmer en Macintosh* donnera des idées à celles (je ne vous oublie pas, jolies lectrices) et à ceux (je vous aime bien aussi) qui, utilisateurs d'un Mac au bureau, disposent d'un GS pour meubler leurs longues soirées d'hiver. J'ai dévoré le chapitre sur le C (merci de m'envoyer un autre bouquin pour mon dîner), mais le gestionnaire de mémoire a stoppé mon élan. Je reviendrai dessus dès que ma pauvre cervelle aura retrouvé son équilibre... et je n'en serai qu'à la page 45 (le livre en compte 300). Un bon ouvrage de programmation pour tous les courageux.

ÉYROLLES - 61, boulevard St Germain 75005 PARIS
Tél. : 46.34.21.99



HGR MONOCHROME

Un lecteur de Sophia Antipolis m'a appelé pour me faire part de ses problèmes avec l'émulation du mode HGR, sur le GS. Lorsqu'il affiche des caractères ou des lignes, il lui est impossible d'obtenir un aspect identique à celui dont il bénéficiait en monochrome sur son *Ile*. Disons, pour être bref, que selon la parité de leur colonne, les points font 1/3 ou 2/3 du point habituel. La lisibilité des affichages s'en trouve évidemment perturbée. Essayez par exemple AUX.DRAW de *Tremplin Micro n°12* sur un GS !

Ne trouvant rien sur ce point dans ma documentation, et surpris par la réponse du Support Technique Apple : bug de la carte mère, j'ai interrogé le responsable du Support Développeur qui m'a aussitôt envoyé un document très clair, permettant de régler le problème.

Les possesseurs d'un GS savent qu'il leur est possible de sélectionner le mode monochrome à partir du tableau de bord, méthode parfaite pour l'affichage DHGR, mais insuffisante pour le mode HGR.

Le tableau de bord manipule l'octet \$C029 (NEWVIDEO pour les intimes) dont la structure est la suivante :

- bit7 = 1 active le mode THGR (le nouveau mode spécifique au GS).
- bit6 = 1 *linéarise* (indispensable pour envoyer ou récupérer des infos dans le buffer THGR en \$E12000).
- bit5 = 0 pour affichage hires couleur, = 1 pour affichage monochrome
- bit4,3,2,1 = 0 IMPERATIVEMENT.
- bit0 = manipulé par le moniteur (accès INTERDIT aux utilisateurs).

Lorsqu'on lui demande de passer en monochrome, le tableau de bord met le bit5 à 1.

Pour que le mode HGR passe effectivement en monochrome, il faut également positionner AN3

sur OFF par un accès à \$C05E (attention, les labels APPLE sont curieux : \$C05E s'appelle SETAN3 et il est commenté *clear annunciator3* tandis que \$C05F s'appelle CLAN3 et qu'il est commenté *set annunciator3*).

La routine MONOCH permet de forcer le mode monochrome et de rétablir le mode initial si nécessaire. Dans le programme TEST.AUX.DRAW, il suffira d'ajouter la ligne 25 PRINT D\$"-MONOCH" et d'ajouter, en fin de programme CALL771.

Un point m'intriguait dans la réponse de M.Schott (Support Développeur). On pouvait y lire que de nombreux programmes HGR *Ile* étaient incomplets parce qu'ils omettaient de manipuler \$C05E/\$C05F alors que je n'avais jamais rencontré la moindre mention du rôle de ces switches sur le mode HGR (en dehors de l'activation du mode DHGR).

Je me suis replongé dans mes archives et, surprise ! bien au chaud dans le classeur FORUM des DEVELOPPEURS, à la fin de *APPLE Ile technote £3*, j'ai trouvé le paragraphe traduit ci-dessous :

Si AN3 est en position OFF en présence d'une carte 80 colonnes étendue (permettant l'affichage DHGR), le bit le plus significatif en HGR standard sera ignoré. C'est-à-dire que toute image HGR simple sera affichée en utilisant exclusivement Noir, Blanc, Violet ou Vert. Si l'image contenait du bleu ou de l'orange, ces couleurs seront converties en violet ou en vert.

Ce qui me console un peu, c'est que je ne suis apparemment pas le seul à ne pas avoir saisi les implications de ce paragraphe.

Activer \$C05E lorsque l'on emploie le mode HGR force en fait le mode Monochrome sur les *Ile* et *Ilc*.

Il est, de ce fait, convenable de prévoir un BIT \$C05E chaque fois que l'on a des messages à afficher et de faire BIT \$C05F lorsque, la réponse attendue étant obtenue, on souhaite rétablir l'affichage en couleurs.

(suite page 66)

En résumé :

- sur Ile et Iic HGR mono par BIT \$C05E
HGR couleur par BIT \$C05F
- sur GS HGR mono par BIT \$C05E
HGR couleur et bit5 = 1 sur \$C029
par BIT \$C05F
et bit5 = 0 sur \$C029

MONOCH fait tout cela pour vous.

- CALL768** — appelle la routine officielle d'identification pour savoir si l'on se trouve sur un GS. Si c'est le cas, on sauve le contenu de \$C029 pour pouvoir le rétablir et on met le bit5 à 1 pour forcer le mode monochrome.
- GS ou pas, on lit l'indicateur d'état de AN3, on le sauve et on force le mode monochrome.

- CALL771** — appelle la routine officielle d'identification. Si l'on a affaire à un GS, on rétablit \$C029 dans son état initial.
- GS ou pas, on récupère l'état initial de AN3 ce qui nous permet de savoir s'il nous faut rétablir le mode couleur par BIT \$C05F.

QUESTION

Ayant fabriqué un petit programme *STARTUP*, j'ai réussi à obtenir des pauses de mes catalogues *PRODOS* par l'emploi de *CTRL-S*. Rien de bien extraordinaire, me direz-vous et, de plus, cela ne me satisfait guère. Ma question est la suivante :

- Existe-t-il sous *ProDOS* des Pokes semblables à ceux du *DOS* permettant d'avoir des pauses automatiques (*Poke 44452,22* etc.) ?

J. D. (95300 PONTOISE)

RÉPONSE

NON, NON et NON ! par pitié, ne bidouillez pas *BASIC.SYSTEM* ! Je me suis laissé aller à le faire il y a plusieurs mois pour avoir une pause plus commode que *Ctrl-S*, (c'était sur le disque *SIGNATURE*) mais, c'était dans une zone qui n'a pas bougé de *BI 1.0* à *BI 1.1* (pardon, *BI* c'est *BASIC.SYSTEM*).

Il n'y a pas beaucoup de place disponible et celle-ci est nécessaire pour corriger les VRAIS bugs de *BI*.

N'oubliez pas que l'on peut interroger un *CATalog* en cours par *Ctrl Q* et qu'il est possible de faire un *CATalog* sélectif par *CATalog.Txxx*.

Bien entendu, je ne peux vous interdire de mettre en place une routine de comptage des lignes comme cela existait dans le *DOS 3.3* mais à l'époque les utilisateurs demandaient un patch *SUPPRIMANT* cette pause.

Yvan KOENIG

IMAGEWRITER II

Manuel de référence

Bibliothèque Apple,
en coédition avec
Apple Computer France

Un ouvrage
de base pour votre
Bibliothèque informatique

L'ImageWriter II fonctionne aussi bien avec les ordinateurs Macintosh qu'avec ceux de la gamme Apple II.

Aux utilisateurs chevronnés, ainsi qu'aux responsables des équipes-système, ce manuel offre la clé qui leur permettra de connaître et de mieux maîtriser les caractéristiques de cette imprimante.

Les programmeurs pourront ainsi :

- soit utiliser l'ImageWriter II à partir de leurs propres programmes.
- soit configurer un programme d'application général pour l'imprimante.

Ils seront en mesure de piloter l'ImageWriter à partir du langage de programmation de leur choix, de contrôler la vitesse de transmission, d'utiliser des caractères étrangers, de moduler le style d'impression, la disposition du texte sur la page, de créer et d'utiliser des caractères personnalisés, de concevoir et d'imprimer des graphiques en couleurs, etc.

Des annexes techniques fournissent en outre une table ASCII, la liste des commandes de l'imprimante, les motifs point-par-point de tous ses caractères, sa fiche technique ainsi que son interface série.

Spirale, 195 × 230, 244 pages, 320 F

InterEditions 87, avenue du Maine 75014 PARIS — Tél. : 43.27.74.50

```

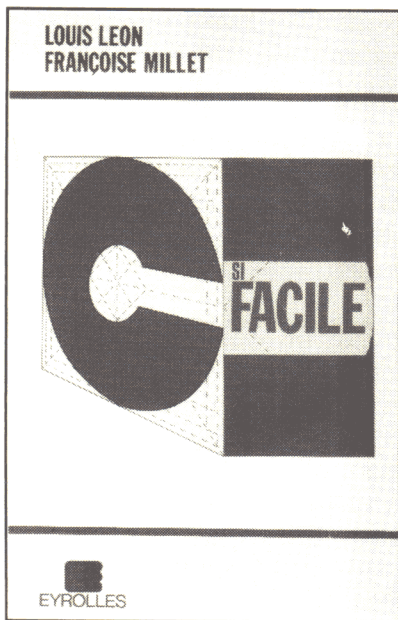
1
2 *****
3 * HGR MONOCHROME
4 *
5 * Si vous êtes déçu par l'émulation HGR de votre GS
6 * ou lassé par les 'facéties' du mode HGR //e, //c
7 * sur un moniteur couleur utilisez MONOCH.
8 *
9 * APPLE nous avait caché l'existence d'un 'switch'
10 * permettant de forcer l'affichage monochrome.
11 *
12 * M. SCHOTT m'ayant révélé celui-ci, tout s'arrange.
13 *
14 * Essayez par exemple TEST.AUX.DRAW de Tremplin 13
15 * en ajoutant la ligne 25 PRINT D$"-MONOCH"
16 *
17 * Yvan KOENIG                                     25/05/87
18 *****
19
20 IDROUTINE =    $FE1F
21
22 *-----*
23             ORG    $300
24 *-----*
25
0300: 4C 1A 03 26 CALL768 JMP    DOMONO    ;Force affichage monochrome
0303: 38          27 CALL771 =    *          ;Rétablit mode initial
0304: 20 1F FE 28          SEC
0307: B0 06      29          JSR    IDROUTINE ;Quelle machine ?
0309: AD 35 03 30          BCS    :1          =>Pas un GS
030C: 8D 29 C0 31          LDA    GREG
030F: AD 36 03 32          STA    $C029    ;Rétablit valeur initiale.
0312: 29 DF      33 :1     LDA    ETAT_AN3
0314: D0 03      34          AND    £%11011111 ;teste bit5
0316: 2C 5F C0 35          BNE    :2          ;on était en mono, on y reste
0319: 60          36          BIT    $C05F    ;rétablit mode couleur
031A: 38          37 :2     RTS
031B: 20 1F FE 38          DOMONO SEC
031E: B0 0B      39          JSR    IDROUTINE ;Quelle machine ?
0320: AD 29 C0 40          BCS    :1          =>Pas un GS
0323: 8D 35 03 41          LDA    $C029    ;Le bit5 contient le mode TABLEAU
0326: 09 20      42          STA    deBORD
0328: 8D 29 C0 43          STA    GREG      ;Sauvons le
032B: AD 46 C0 44          ORA    £%00100000 ;Force mode MonoChrome
032E: 8D 36 03 45          STA    $C029
0331: 2C 5E C0 46          :1     LDA    $C046    ;Le bit5 contient l'état C05E/C05F
0334: 60          47          STA    ETAT_AN3 ;Sauvons le
0335: 00          48          BIT    $C05E    ;Force mono sur TOUTES machines
0336: 00          49          RTS
0337: 00          50          DS    1
0338: 00          51          DS    1
0339: 00          52          DS    1
0340: 00          53          DS    1

```

BSAVE MONOCH,A\$300,L\$37

Votre bibliothèque INFORMATIQUE

par NESTOR



• C'EST SI FACILE (L. LÉON, F. MILLET)

Je me suis amusé à taper plusieurs des exemples de ce livre sur mon GS... et ça a marché. En fait, ce livre présente le **C** d'aujourd'hui, celui qui respecte un certain nombre de normes, mondialement reconnues et passe par les fonctions d'une bibliothèque standard.

Si vous recherchez une bonne initiation, n'attendez pas davantage.

Les 316 pages de **C'EST SI FACILE** vous donneront envie d'aller plus loin, mais vous en saurez déjà assez pour créer de petites routines sur votre Apple IIGS... ou sur n'importe quelle machine acceptant le **C**.

EYROLLES - 61, boulevard St Germain 75005 PARIS

• PROGRAMMATION EN C DE L'IBM PC (Gérard LEBLANC)

Un de plus vont penser les ennemis jurés du **C**. C'est vrai que la littérature sur icelui commence à devenir abondante, mais on ne peut que s'en réjouir.

Le **C** deviendra probablement le langage de développement par excellence. Il n'est pas sûr qu'il parvienne à séduire les actuels adeptes du Basic, mais j'avoue souhaiter

qu'un grand nombre de programmeurs "amateurs" s'y intéressent.

Ici — à *Tremplin Micro* —, on nous pose souvent des questions sur la portabilité réelle du **C**. Réponse claire et nette : s'il s'agit du **C** de l'IBM PC du même nom, aucun problème. Le **C** est généralement portable. L'APW.C de l'Apple IIGS fera probablement son affaire du petit programme que votre copain a pondu sur un compatible PC, mais ne lui demandez tout de même pas l'impossible.

En attendant, notamment si vous êtes cinquante pour cent IBM et Apple pour l'autre moitié, étudiez le bouquin de Gérard LEBLANC.

EYROLLES - 61, boulevard St Germain 75005 PARIS

• DÉVELOPPER EN LANGAGE C (Jean-Pierre LAGRANGE)

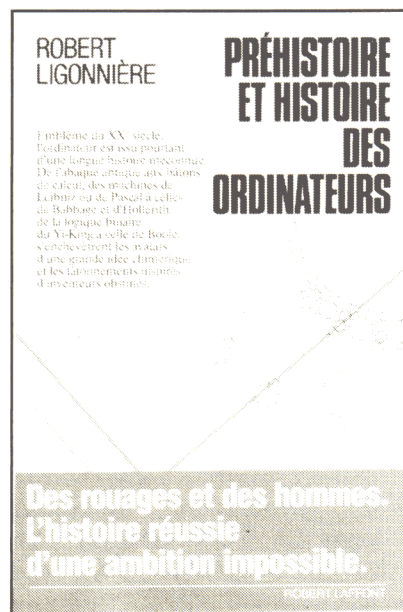
L'auteur nous rappelle — comme je l'ai fait plus haut — que le langage **C** est portable... dans certaines limites. Les fonctions données dans son livre concernent en priorité l'IBM PC (et les compatibles), mais pourront presque toujours être adaptées par d'autres utilisateurs, notamment en ce qui concerne les fichiers.

Les Applemaniaques apprendront avec plaisir que les routines de Jean-Pierre LAGRANGE passent très bien dans le LAT-



TICE.C... ce qui leur donnera de sérieux espoirs pour les compiler avec l'APW.C de l'Apple IIGS. Pas davantage de problèmes avec le TURBO **C** de Borland ou le MICRO-SOFT V4.0. En réalité, les programmes du livre (la disquette y est jointe, bravo !) sont donnés pour l'excellent compilateur de Borland. Gageons que certains d'entre eux (je pense notamment à la gestion des fichiers, mais aussi aux diverses méthodes de tri), feront le bonheur des programmeurs, avertis ou non.

EDITIONS MICRO APPLICATION
13, rue Sainte-Cécile 75009 PARIS.



• PRÉHISTOIRE ET HISTOIRE DES ORDINATEURS (Robert LIGONNIÈRE)

Sortez un peu de la programmation et revivez la longue histoire — ô combien méconnue ! — de l'ordinateur, sans doute la plus géniale création de notre XX^e siècle.

L'Homme a toujours été attiré par les chiffres qu'il n'a jamais cessé d'additionner, puis de multiplier à l'infini, en inventant pour cela des machines de plus en plus ingénieuses. Mais l'ordinateur n'aurait sans doute jamais vu le jour — sous sa forme actuelle en tout cas — sans la logique. Il n'y a pas si loin de la carte-programme du métier à tisser, qui préfigure la mécanographie, à la "machine analytique" de Charles Babbage. Mais ce n'est encore qu'une solution *mécanique* à un problème que (seule) *l'électronique* sait résoudre... un peu plus tard. Hier... aujourd'hui, la réalité dépasse la fiction. Quelle belle histoire !

EDITIONS ROBERT LAFFONT

SuperMacroWorks

APPLEWORKS + LA SOURIS + LES MACROS

SuperMacroWorks, de Randy Brandt, est un produit Beagle Bros.Inc, maintenant disponible en France, et avec un manuel complet en Français. Le traducteur et adaptateur est Dimitri Geystor*. Vous pourrez vous procurer cet excellent logiciel américain auprès de lui.

Mais nous n'en aurions peut-être pas parlé dans Tremplin Micro si Dimitri, applemaniaque convaincu, s'était contenté de commercialiser l'un des best-sellers des Beagle Brothers. Il a fait mieux que cela puisqu'il vous accorde, en prime, une disquette d'exemples commentés (en français bien sûr), contenant notamment une macro inédite, construisant sous vos yeux un agenda annuel.

Mais revenons à SuperMacroWorks !

• Qu'est-ce qu'une macro ?

Une série de commandes compilées pour en faire une routine pratique et rapide, actionnée par une seule commande du clavier. On peut, de cette manière, imaginer des choses invraisemblables, depuis la mise en page élémentaire d'un en-tête de lettre, jusqu'à celle, plus complexe, d'une facture... l'une et l'autre n'étant exécutées qu'à partir d'une seule touche.

Aux Etats-Unis, on parle de plus en plus des macros pour Appleworks et les revues de programmation leur consacrent de plus en plus de place. Nos confrères américains, qui connaissent bien Autoworks, KeyPlayer et SuperMacroWorks, ont accordé la première place à ce dernier. Fait intéressant, c'est aussi le seul qui soit compatible avec la version française 1.4 d'AppleWorks.

• Les plus de SuperMacroWorks

D'abord et surtout, mais cela reste facultatif,

l'usage de la souris. Par ailleurs un ensemble cohérent de nouvelles commandes, utilisant toutes les ressources de POMME OUVERTE, POMME FERMÉE et CONTRÔLE.

Elles vous permettent par exemple d'imprimer l'heure et la date, de lister directement les fichiers dans une base de données, etc.

D'autre part, des fonctions logiques permettent, en les incorporant à des macros, de construire des macro-programmes. Mieux encore, vous pourrez, avec SuperMacroWorks, créer des macros sans quitter AppleWorks. De quoi rêver !

(*)

Adaptation en français et auteur du manuel français :

Dimitri Geystor : "La Chenaz", Saint-Blaise
74350 CRUSEILLES
Tél. : 50.44.21.70

Vous pouvez d'ores et déjà passer commande auprès de :

Société Bréjoux : 29 rue de Montriblond
69009 LYON
Tél. : 78.36.52.69

S.I.G.E.A. : 21 avenue des Hirondelles
74000 ANNECY
Tél. : 50.57.02.80

Si vous possédez le logiciel américain, et que vous souhaitez vous procurer le manuel français + la disquette d'applications, adressez un chèque de 50 F à Dimitri Geystor, en joignant une enveloppe format A.5 affranchie à votre adresse, une disquette vierge format 5'1/4, et **obligatoirement** le coin du dos de la couverture du manuel anglais qui porte la mention "Proof of purchase" (preuve d'achat). ■

GRIBOUILLE PARLANT

Une série de réalisations françaises pour les handicapés visuels

C'est la France qui a reçu, cette année, la conférence internationale de l'Union Européenne des Aveugles (au Palais de l'U.N.E.S.C.O., du 17 au 19 septembre).

A cette occasion, des postes de travail informatiques pour handicapés visuels étaient réunis et présentés aux visiteurs. Une série de réalisations met, en effet, notre pays en bonne position dans ce domaine.

Les cartes françaises **Valentine** et **Sonolect**, couramment utilisées pour sonoriser des bases de données, tableurs, jeux etc., viennent de recevoir deux applications majeures, qui ont été présentées pour la première fois à la conférence internationale de l'Union Européenne des Aveugles :

MINITEL PARLANT

Valentine et **Sonolect** rendent parlant l'Appletel (de la société française Hello). Dès lors, les déficients visuels peuvent consulter l'annuaire électronique et les bases de données du minitel, exécuter les locations, achats, appels de taxis, ordres bancaires et autres opérations qu'il permet.

GRIBOUILLE PARLANT

Le très puissant traitement de texte français *Gribouille* a été adapté par son auteur Madeleine HODÉ, et par Philippe LÉON, auteur de **Sonolect**, à la synthèse vocale. Les non-voyants ont maintenant pleinement accès à un logiciel professionnel dont chaque détail a été mis au point pour s'adapter à leur mode de perception et répondre à leurs besoins. Ils peuvent ainsi relire et corriger les textes qu'ils dactylographient, et en contrôler la mise en page avant impression. En outre, *Gribouille* comporte des commandes de calcul qui le transforment, pour les handicapés visuels, en une "calculatrice parlante".

Il faudrait plusieurs pages de *Tremplin Micro*

pour énumérer les difficultés rencontrées par la dynamique Madeleine HODÉ et par l'éminent Philippe LÉON lors de l'adaptation de *Gribouille*. Que d'efforts ! Mais le résultat est là, extraordinaire. Ce n'est pas trahir un secret en disant qu'il a été unanimement apprécié par celles et ceux qui ont pu le tester.

Peut-être aurons-nous un jour l'occasion de revenir sur ce *Gribouille parlant*, très bel exemple de la réussite du génie logiciel français.

POUR EN SAVOIR PLUS :

Un peu de technique : Valentine et Sonolect.

Valentine, carte de synthèse vocale (société FERMA, 14, rue de l'Abbé Carton, 75014 PARIS) et **Sonolect** (éditeur vocal réalisé par un étudiant marseillais aveugle, Monsieur Philippe Léon) se conjuguent pour sonoriser les logiciels : l'utilisateur entend, dans une prononciation française correcte, ce qui s'inscrit à l'écran. Il peut ainsi lire ce qu'il tape lui-même au clavier, aussi bien que les menus et messages du logiciel.

Correspondant permanent :

Association Valentin Haüy pour le bien des aveugles

5 rue Duroc, 75007 PARIS

Tél. : (16-1) 47.34.07.90.

Cette association, par son Centre de Formation et de Réadaptation Professionnelle, organise depuis plusieurs années des cours professionnels et notamment en informatique.

Elle commercialise la carte **Sonolect** et la version sonorisée du logiciel *Gribouille*. Pour l'ensemble : 1770 F hors taxe, soit 2100 F T.T.C.

La carte **Valentine** peut être achetée soit à la société FERMA soit à l'association Valentin Haüy : 2530 F hors taxe, soit 3000 F T.T.C.

(Tél. : (16-1) 47.34.07.90, Melle LANO ou M. JACQUIN)

On peut tester logiciel et carte sur rendez-vous, à : Club Micro-Son

3, rue Berthe 92370 CHAVILLE

Tél. : (16-1) 47.50.16.49

LES CARRÉS MAGIQUES

de Jean PERROT

Dans le n°14 de *Tremplin Micro* a paru un programme appelé CARRÉS MAGIQUES. Il permet d'obtenir des carrés magiques, quel que soit leur ordre, mais il ne donne aucune explication concernant les méthodes qui ont permis l'élaboration du logiciel. Aussi Jean PERROT propose-t-il aux lecteurs de *Tremplin Micro* qui seraient intéressés, de leur faire parvenir une copie de sa disquette personnelle.

Elle contient 10 programmes :

- 2 de présentation : MENU et CAR.HISTO.
- 4 de méthodes : CAR.IMP, CAR.PAIR2, CAR.PAIR4, CAR.PROD.
- Le programme principal : CARRES.MAG.
- 3 programmes (en langage machine) qui permettent l'écriture en page HGR ou le transfert d'un écran en mémoire auxiliaire à savoir : TEXT/HGR, TRANS80.LM, FAIRE.F1

Envoi effectué contre un chèque de 60 F + 10 F de participation aux frais d'envoi.

Jean PERROT 27 rue du Clos BEAUMOIS 14000 CAEN.

VERSION SOFT À APPLE EXPO

Version Soft, éditeur leader sur Apple, a présenté plusieurs nouveautés pour l'Apple II/GS lors de la dernière édition d'APPLE EXPO :

- **VS/File**, un logiciel de gestion de fichier qui récupère les fichiers issus de CLICKWORKS et d'APPLEWORKS.
- **VS/Com**, un logiciel de communication qui pourra être couplé avec la carte modem Appel-Tell-GS de Hello Informatique, produit pour lequel Version Soft vient de signer un accord de distribution exclusif.
- **Grapholic**, une disquette de 300 motifs et dessins pouvant être utilisés avec les logiciels GS/Paint, VS/Draw et GS/Write nouvelle version.
- **Movie Studio**, logiciel permettant de réaliser des films sonorisés.

Rappelons que Version Soft, avec plus de 280 000 logiciels vendus dans le monde dont près de 100 000 aux Etats-Unis, notamment par le biais d'Activision, est aujourd'hui le numéro 1 mondial du logiciel sur Apple II/GS.

Nous reviendrons, dans ces colonnes, sur les logiciels cités plus haut. ■

UN SYSTÈME EXPERT POUR LE PRIX D'UN JEU :

PROELEC 1

Pour Apple IIc

Conçu et réalisé par des professeurs agrégés, ce logiciel **ouvert et interactif** permet la construction et l'étude de circuits électriques en courant continu. A partir de la classe de seconde.

Prix : 290 F T.T.C. et port payé

Comprenant : • 2 disquettes

• 1 notice de 9 pages

• 1 suivi client

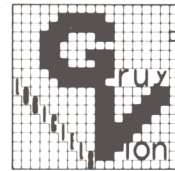
Bon de commande à découper et à envoyer à :

Frédéric GRUY

35 rue du Pilat

42400 SAINT-CHAMOND

Tél. 77 31 74 93



*Veillez m'adresser par retour du courrier le logiciel **PROELEC 1** pour Apple IIc.*

NOM : _____

Prénom : _____

Adresse : _____

Tél : _____

Ci-joint mon règlement de 290 F par :

chèque

mandat

TURBO TUTOR

pour le Macintosh

Comment va votre Pascal ? Ah ! vous travaillez sur Apple II. Vous dites que la compilation vous fait bouillir d'impatience ? Bien sûr... je sais, mais il faut vous offrir un GS, ce sera déjà plus rapide !

Moi, du Pascal, j'en fais sur Macintosh, et avec le *TURBO* de Borland. C'est quand même autre chose. Vous vou-

lez la vérité ? par rapport à ce que je connaissais déjà, sur Apple IIe, c'est réellement extra.

Ceci étant, j'ai eu entre les mains, pour un essai — il ne faut rien cacher à un lecteur de *Tremplin Micro* —, le *TURBO PASCAL TUTOR* de Borland. Toujours lui !

Je ne suis pas un fana de la

langue de Shakespeare et le gros bouquin (près de 700 pages) qui accompagne la disquette est en anglais. Je n'ai donc pas tiré un grand profit de l'historique (d'ailleurs court) du PASCAL, mais comme il y a toujours des exemples dans les manuels de la société Borland, je ne m'en tire pas trop mal.

Celui-ci permet réellement, disquette à l'appui, à un programmeur débutant de commencer à écrire des programmes en *TURBO PASCAL*. Naturellement, le débutant devra d'abord assimiler des notions aussi élémentaires que les définitions et la terminologie de base, mais ça ne devrait pas détériorer ses neurones.

Tout ce qui peut poser problème est passé en revue dans cet ouvrage et on peut dire qu'il s'agit effectivement d'un excellent guide du programmeur. Il traite avec précision, mais en anglais, de la structure des programmes en *TURBO PASCAL* : procédures et fonctions, chaînes, pointeurs, fichiers... tout y passe !

Un bon outil d'apprentissage !

Clément RENARD.

Philippe KAHN 34 ans, Président Fondateur de BORLAND INTERNATIONAL.

Cet ancien professeur de mathématiques décide, en 1982, de quitter l'éducation nationale et les Alpes Maritimes. Il s'envole à 30 ans pour la Californie poussé par sa passion de la micro-informatique, avec l'intention de devenir éditeur de logiciels.

Une compétence acquise à l'âge de 20 ans auprès de Niklaus Wirth, auteur du langage Pascal, lui permet de découvrir une version très puissante de Pascal. Il la baptise *Turbo Pascal*, nouveau langage de programmation dix fois plus rapide que les autres compilateurs et qu'il veut vendre six à dix fois moins cher que les produits concurrents. Son produit exceptionnel en main, Philippe Kahn cherche à le commercialiser. Il effectue des démarches auprès des "Venture-Capitalists" qui demeurent vaines. Il décide enfin de financer lui-même son projet et, en mai 1983, fonde en moins de deux heures la société américaine **BORLAND INTERNATIONAL** à Scotts Valley en Californie. Pour lancer son produit, Philippe Kahn utilise deux vecteurs; la vente par correspondance et la publicité dans les revues spécialisées. Le premier mois de vente rapporte 150 000 dollars de chiffre d'affaires, 200 000 le deuxième mois.

BORLAND INTERNATIONAL est lancé.

Sa créativité et sa façon de travailler font de Philippe KAHN un personnage peu conforme au modèle traditionnel des managers américains. Son emploi du temps, extrêmement chargé en tant que Président d'une société internationale, ne l'empêche pas de se consacrer parfois à ses deux principaux loisirs : le saxophone et la voile. ■

En réponse à de nombreuses lettres

PROBLÈMES AVEC L'IMAGewriter II

De nombreux utilisateurs de l'imprimante IMWII apprennent à leurs dépens que la grammaire française a changé. En effet, il faut savoir que le pluriel d'"un ruban", s'écrit désormais "des problèmes" et que parfois, il ne s'imprime pas.

L'Echo des APPLE a déjà signalé quelques problèmes avec le côté gauche des cartouches, pourtant purement "passif". Le côté droit, qui contient les éléments mobiles, nous réserve d'autres surprises. Le SAV APPLE a rédigé une fiche qui note les différents modèles disponibles. Son meilleur choix, PELIKAN, semble difficile à trouver chez les concessionnaires APPLE. Il sera plus facile de se les procurer chez des fournisseurs d'accessoires de bureau.

Voici les importateurs de ces cartouches :

- **NOBLET**, 178 rue du Temple, 75003 PARIS
Tél. : (16-1) 42.77.11.34.
- **JACMO-FRANCE**, Rubis 20, Z.I. 06510 CARROS
Tél. : 93.08.29.00.

Je précise que chez ce dernier, le tarif unitaire est de 29 F HT (par boîte de 5 cartouches).

Certaines cartouches forcent légèrement, ce qui pose des problèmes au moteur de l'IMWII. J'ai tendance à penser que la cause réelle est un sous-dimensionnement du mécanisme de l'imprimante (un même ruban qui freine l'IMWII passe parfaitement sur une DMP ou une IMW1). APPLE précise d'ailleurs que les rubans qu'ils fournissent actuellement aux concessionnaires sont des rubans IMW1.

Quel que soit le responsable, il y a un remède "aisé".

Divers modèles disponibles compor-

tent, sous la molette d'entraînement, un coulisseau généralement gris clair, maintenu par un ressort plat qui apparaît sur la tranche inférieure de la cartouche. Ouvrez-la, détendez légèrement le ressort et remettez tout en place. Vous aurez alors un entraînement facile et souple.

Il existe un modèle de cartouche que je déconseille vraiment : celui sans coulisseau, sans molette striée, au tubulaire de gros diamètre, sans méplat, aux engrenages à petites dents triangulaires. J'en ai essayé deux exemplaires avec des résultats identiques : deux blocages complets du ruban sans blocage de l'imprimante. *Conséquences* : une belle boutonnière dans chaque ruban et 3 pages imprimées blanc sur blanc.

Pour être complet, précisons que le SAV fait état de problèmes liés à la

nature de l'encrage de la plupart des rubans. Encore une fois, le meilleur choix serait PELIKAN.

LES CLEFS POUR LE IIGS

Nicole BRÉAUD-POULIQUEN a des problèmes avec *les clefs pour le IIGS*. On y trouve souvent le (/) au lieu du (ç) dans les commandes moniteur, mais aussi (') au lieu de ("), ce qui réserve quelques surprises aux utilisateurs. Ainsi, l'exemple de recherche de chaîne de la page 77 fonctionne mieux lorsqu'il est rédigé ainsi :

*ç"Check"ç FF/0000.FFFFP.

En utilisant le délimiteur ('), comme le fait l'ouvrage, on recherche la chaîne Check STOCKÉE À L'ENVERS (cad: KcehC) que l'on ne trouve évidemment pas.

Il s'agit bien entendu de coquilles et non d'erreurs de l'auteur.

L'APPLE IIGS ET LES CARTES D'EXTENSION

La situation de l'acquéreur français d'un GS est différente de celle de son confrère américain puisqu'APPLE France livre systématiquement la carte APPLE avec la machine, ce qui n'est pas le cas aux U.S.A. De ce fait, en dehors de quelques cas précis, il n'y a pas lieu de se précipiter sur les offres des divers constructeurs indépendants.

On voit apparaître de multiples cartes extension mémoire avec ou sans batterie. Il me semble préférable d'attendre. Pour la plupart, ces cartes ne sont rien d'autre que des RAMdisk (éventuellement plus ou moins permanents). Elles ne tirent pas parti d'une caractéristique du GS, à savoir la possibilité de manipuler un ROMdisk de 500k et une extension de la boîte à outils de près d'un demi méga. Avec un tel ROMdisk en place, il suffirait de démarrer sur lui pour mettre en place quasi instantanément son contenu (système d'exploitation et finder par exemple) et aussi (surtout), mettre en service les outils présents dans la carte SANS avoir à effectuer une quelconque opération de chargement, exactement comme pour les outils déjà présents sur la carte mère.

Avec un peu de chance, la table des messages français qui occupe actuellement une partie de la RAM (\$020300... avec la disquette Système) pourrait être installée elle aussi dans cette extension ROM. Il est inutile d'être devin pour prévoir l'apparition, lorsque Système d'exploitation et outils seront stabilisés...

(suite page 74)

(la quasi totalité des outils, ROM ou RAM a été modifiée au cours du 1^{er} trimestre 87), d'une carte made in APPLEland qui dotera le GS d'une mise en route enfin rapide.

Malgré la différence de situation initiale, je conseille à tout lecteur tenté par l'acquisition d'une extension mémoire, la lecture du numéro d'Août de Call APPLE. Le constructeur le plus prodigue en pages de publicité (et de ce fait, le plus connu) en ressort doté d'un magnifique "costard" dont je n'ai pu vérifier le bien-fondé.

Pendant que je parle des cartes extension, il me faut rappeler deux détails. Il n'est pas indispensable d'acheter les RAMs complémentaires dans le circuit commercial APPLE, mais il est nécessaire de faire attention au paramètre "temps d'accès". Les puces soudées sur la carte APPLE sont des 15usec, et il est déconseillé de les accoupler à des puces 12usec (cette remarque s'adresse surtout aux jeunots, les anciens se souviennent que le non-respect de ce vieux principe leur avait créé quelques problèmes lorsqu'ils ont porté leur APPLE II de 16 à 48K).

Par ailleurs, seules les configurations 256, 512 et 1024 K sont garanties par APPLE. La configuration 768K est physiquement réalisable mais elle semble mal digérée par le MEMORY MANAGER et vous ferait perdre le bénéfice de la garantie constructeur.

MISE À JOUR DE VOS DISQUETTES

Je répète que "votre" concessionnaire a, entre autres obligations, celle de vous mettre à jour certaines disquettes (c'est bien entendu à vous d'en formuler poliment la demande). En ce qui concerne le GS, citons les disquettes : Mouse-Desk (9 décembre 87), GS.WRITE 2.0, GS.PAINT 1.1. Il s'agit de versions améliorées et améliorables. L'impression est toujours désespérément lente et il y a encore quelques bugs. PATIENCE... je sais, c'est désagréable MAIS, chez APPLE comme chez VERSION. SOFT, on a le courage et l'honnêteté de reconnaître l'existence de ces problèmes. J'en connais d'autres dont les réactions sont à la limite de l'injure lorsqu'on leur signale des bugs.

PITIÉ POUR L'UNITÉ CENTRALE !

Je rappelle qu'il est DANGEREUX pour l'unité centrale de brancher/débrancher un lecteur sur une machine sous tension. Pour déconnecter quoi que ce soit d'un GS, il faut éteindre par l'interrupteur et attendre environ 1 minute.

S'il s'agit d'une opération interne, il faut en outre toucher le boîtier alimentation pour décharger votre éventuelle électricité statique et seulement alors, effectuer l'opération. Par ailleurs, le SAV APPLE a vu revenir des cartes mères qui ne présentaient aucune anomalie. Il semble que le pré-diagnostic ef-

fectué par les concessionnaires soit parfois erroné (il est parfois inexistant). Pour améliorer le fonctionnement du SAV, il est fortement conseillé de donner à votre concessionnaire, lorsque c'est possible, le numéro de téléphone qui permettra, si nécessaire, au SAV de vous joindre pour vous demander un éventuel complément d'information qui servira parfois à corriger le pré-diagnostic et (ou) à configurer correctement votre équipement en fonction du (des) logiciel(s) employé(s).

Afin de vous éviter des pertes de temps, je vous fais part de mon

expérience. Lorsque votre GS refuse de lire tout ou partie d'une disquette neuve, ce n'est pas nécessairement la disquette ou le lecteur qui sont en cause. Le problème peut provenir de la carte mère. Essayez donc, si cela vous arrive, de lire votre disquette à problème sur un autre lecteur puis, sur une autre unité afin d'isoler la source réelle du phénomène. N'oubliez pas également l'existence d'un autotest exécuté par Ctrl Option Pomme Reset qui permet de savoir si tout va bien au niveau de la carte mère. Bien entendu, si un connecteur de liaison au monde extérieur est dessoudé, ce test ne le verra pas.

Yvan KOENIG

OFFRE SPÉCIALE

- CLINS D'OEIL AU 6502 DE L'APPLE
- ROUTINES LM POUR 65C02 ET 6502
- NOUVELLES ROUTINES LM (65C02 et 6502)

Profitez de la fin de l'année pour vous attaquer sérieusement au langage machine et compléter votre collection de **TREMPIN MICRO**.

Un numéro gratuit pour tout exemplaire des routines commandé avant le 31 décembre 1987. ■

INFOS APPLE //GS

Chers lecteurs ! voici les dernières nouvelles concernant l'Apple //GS :

- **Système :** Le système 3.1 serait bientôt disponible, ainsi que le FINDER GS, tournant sous ProDOS 16 et Icon Editor, programme permettant de dessiner vos propres icônes système. Le FINDER est un produit extraordinaire permettant des copies, des sauvegardes, l'organisation des disks, le lancement d'applications, tout ceci uniquement avec la souris. C'est d'une simplicité déconcertante.
- **Hardware :** Un nouveau VGC et une nouvelle ROM contenant les derniers outils devraient être proposés.
- **Langages de programmation :**
 - Le *Pascal GS* est terminé ; il existe sous deux versions : une version TML Pascal avec éditeur, menus déroulants et souris (compilation en mémoire prenant au maximum 10 secondes) et une version compatible APW, ORCA/PASCAL.
 - L'*Assembleur* est disponible en version définitive, numéro 1.0.
 - Le **C**, qui sera certainement le langage le plus puissant sur le GS, est disponible en version 1.0 B7. Je vous conseille de vous lancer dans l'apprentissage de ce langage simple. Dans les prochains numéros de *TREMLIN MICRO*, un *Guide d'installation de l'APWC* vous sera présenté, ainsi que de nombreuses pages d'apprentissage du langage **C**. Gageons que ce sera le langage du futur.
 - Pour les inconditionnels du Basic, un *Basic GS* est disponible chez DDA, avec un éditeur texte nettement plus performant. Ce Basic tourne sous ProDOS 16 et fait appel à toutes les fonctions des outils. Il permet

ainsi une programmation aisée, enfin puissante et rapide.

J'ai personnellement testé tous les logiciels précités ainsi que ceux de la liste qui suit.

Tous ces programmes tournent, bien sûr, sous ProDOS 16, en mode natif. Le dernier système est beaucoup plus rapide et très fiable. Le FINDER ressemble à celui disponible sur Macintosh (il a évidemment la couleur en plus !). Les drivers d'imprimantes ont été réécrits. Ils impriment trois fois plus vite en qualité brouillon et deux fois plus vite en qualité courrier.

J'espère que toutes ces informations feront plaisir aux utilisateurs d'*APPLE //GS*. A très bientôt dans le monde de la Pomme, et sur GS !

Adresse de DDA : Technopole rue M. Faraday
78180 MONTIGNY LE BRETONNEUX
Tél. : (16-1) 30.45.26.62.

LISTE DES LOGICIELS

UTILITAIRES :

Sous Dos 3.3

UNIDOS . utilisation des disquettes 3'5 sous Dos 3.3.
UNIMATE idem.

Sous ProDOS 8

BEAGLE BROS COMPILER : compilateur Basic Applesoft.

BIG.U : utilitaires de patch pour ProDOS ; patches de l'Applesoft.

PRO-BYTER : utilitaires sous ProDOS. Indispensable !

DIVERSY-COPY : copieur disks 3'5 très rapide.

Copy II+7.4 : copieur disks 3'5 et 5'25. Sauvegarde, récupération fichiers, transfert Dos 3.3/ProDOS...

BAGS OF TRICKS 2 : réparation de disquettes 3'5 et 5'25, récupération de fichiers écrasés. (suite page 76)

DIVERSY-KEY, DIVERSY-CACHE : accélération Boot, très utile.
SOFTSWITCH : switcher entre applications (3 applications en même temps).

Sous ProDOS 16

MOUSE-DESK 2.0 : le mouse desk bien connu.
* (FINDER 1.0 : le dernier finder Apple. Absolument indispensable !
* ICON EDITOR : fabrication d'icônes pour le FINDER), bientôt disponibles.

MUSIQUE :

MUSIC CONSTRUCTION SET : créez vos propres musiques.
MUSIQUE STUDIO : idem mais plus puissant, reconnaît l'interface MIDI.
INSTANT MUSIC : logiciel musical.
CRISTAL (+ carte) : digitalisation sonore + parole (à venir).

TRAITEMENTS DE TEXTE :

GRAPHIC WRITER : très bien, avec possibilités d'insertion d'images.
MULTISCRIBE : très bon traitement de texte (en anglais).
WORD-PERFECT (+ vérification orthographe + dictionnaire).
GRIBOUILLE (ProDOS 8) : très bien fait.

BASES DE DONNÉES — TABLEURS :

APPLEWORKS 1.4 (ProDOS 8) : traitement de texte + tableur + base de données.
SOFTWOOD GSFILE : très beau logiciel.
NOTES & FILES : idem !
VISUALIZER : récupération de fichiers Appleworks/graphismes.
V.I.P. : tableur avec graphiques.
VS/FILE : gestion de fichiers.

COMMUNICATION :

VS/COM : le seul logiciel de communication en mode natif, très beau !

DESSIN :

816 PAINT : dessins en mode 320 ou 640.
DELUXE-PAINT II : très beau logiciel de dessin + animations.
TOP DRAW : dessins.
USDRAW (architectes) : excellent logiciel (320 et 640) en mode natif.

CLIP ART GALLERY : collection d'images.
GRAPHOLIE : collection d'images pour GSWRITE & GSPAIN.

LANGAGES — PROGRAMMATION :

BASIC GS : Basic en mode natif ; il n'est plus limité par la mémoire.
GSFORTH : Forth en mode natif.
LOGO : langage graphique (disks 5'1/4).
TML PASCAL : très bon logiciel pour programmer en Pascal.
ORCA/PASCAL : Pascal compatible environnement APW.
APW 1.0 ASSEMBLEUR : le dernier assembleur 65C816 disponible.
APWC 1.0 B7 : LE MEILLEUR LANGAGE.
CROSS COMPILATEUR MEGAMAX C + PASSEPORT : transfert de fichiers entre Macintosh et Apple //GS.

JEUX :

COMPUTER MAD LIBS : création de courtes histoires + musique + paroles.
COLOR MIND : master mind.
SHANGAI : jeu de réflexion datant du XI^e siècle.
BARD'S TALE : jeu d'aventures, très beau.
HACKER II : sauvez les U.S.A. en pilotant un robot !
TASSTIMES IN TONETOWN : jeu d'aventures, vraiment bien aussi !
MEAN 18 : très beau jeu de GOLF (+ disks avec différents parcours).
BASKET-BALL : comme son nom l'indique !
RENDEZ-VOUS EN U.R.S.S. : logiciel du domaine public, jeu d'aventures.
WORLDGAMES : Jeux olympiques.

(VERSION SOFT) :

GS/WRITE !
GS/PAINT : création graphique.

ÉDUCATION :

KID'S TIME II : programme destiné à l'enseignement.

À venir :

CHAIRMAN : création d'exposés et animations.
PROGRAMME AUTEUR SUR APPLE //GS : destiné à l'enseignement.
MELIES : création de dessins animés sonores. Génial !
ATLAS 2000 : logiciel de géographie.

LISTE DES LIVRES APPLE IIGS

DOCUMENTATION APPLE : 8 classeurs comprenant plus de 4000 pages sur l'Apple IIGS (en anglais !). Indispensable à tout programmeur chevronné ! (disponible chez DDA).

CLASSEURS :

- ROM TOOLS : les outils en Rom.
- RAM TOOLS : les outils chargés en Ram.
- FIRMWARE : toutes les cartes et moniteur.
- LANGUAGES : tout sur l'assembleur et le **C**.
- INTRODUCTION : introduction, Shell.
- UTI.PRODOS :
- HARDWARE :

Tous ces volumes peuvent être complétés par plus de 20 disquettes de démo comprenant les différents langages (**C** et APW.A) ainsi que des exemples.

APPLE IIGS PRODOS 16 RÉFÉRENCE :

(Addison Wesley, éditeur). Livre en anglais comprenant une disquette EXERCISER GS. Tout ce qui concerne la gestion des fichiers et des volumes sous ProDOS 16.

THE APPLE IIGS TOOLBOX REVEALED :

(Bantam Computer Books). Livre en anglais de Danny Goodman. Tout ce qui concerne les appels à la toolbox de l'Apple IIGS.

TECHNICAL INTRODUCTION TO THE APPLE IIGS :

(Addison Wesley). Introduction à l'Apple IIGS (en anglais).

INSIDE APPLE IIGS :

(Sybex). Programmer en assembleur sur le GS, par Gary Bond (en anglais).

APPLE IIGS FIRMWARE RÉFÉRENCE :

(Addison Wesley). Très utile. Présentation complète du GS et de la programmation en assembleur (livre en anglais).

APPLE IIGS TECHNICAL REFERENCE :

(Mac Graw Hill). Livre en anglais de Michel Fisher. INDISPENSABLE !

CLÉS POUR APPLE IIGS (2^e édition) :

(PSI). Livre en FRANÇAIS de Nicole Bréaud-Pouliquen. Résumé de tous les appels à la Toolbox et présentation des différents langages, éditeur et astuces pour l'Apple IIGS.

BOÎTE À OUTILS DE L'APPLE IIGS :

(PSI). Livre en FRANÇAIS de Jean-Pierre Curcio. Programmation en **C** sur l'Apple IIGS, illustré de très nombreux exemples de programmation.

LES 2 DERNIERS LIVRES CITÉS DOIVENT FAIRE PARTIE DE VOTRE BIBLIOTHÈQUE.

To Guidi Scapsky with my best regards and deep thanks.

L'ÉDITION SUR MACINTOSH

Par A. CHICHERY, J.-P. DRIEUX et A. JARLAUD.

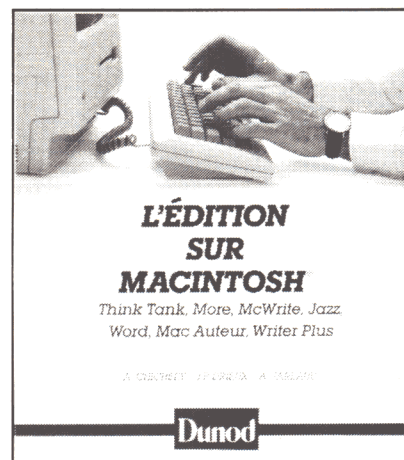
Think Tank, More,
McWrite, Jazz, Word,
Mac Auteur, Writer Plus

De la rédaction à l'impression d'un texte, qu'il s'agisse d'une lettre, d'une circulaire, d'un rapport ou d'un livre, Macintosh se révèle un outil précieux servi par des logiciels aussi variés qu'efficaces.

Dans cet ouvrage, les auteurs étudient le mode d'utilisation de sept logiciels — traitements de textes plus ou moins évolués — permettant de traiter du texte et fonctionnant sur l'ordinateur Macintosh : Think Tank, More, McWrite, Jazz, Word, Mac Auteur, Writer Plus.

Ils présentent chacun d'entre eux de la façon la plus complète et la plus logique possible, afin que le lecteur qui ne le connaissait pas jusque-là puisse très rapidement l'utiliser. Le livre est organisé en chapitres indépendants, chacun d'eux étant consacré à un seul logiciel ; le lecteur trouve donc plus aisément ce qu'il cherche à propos d'un logiciel précis, puisque tout ce qui concerne celui-ci est regroupé dans un même chapitre.

Dans une dernière partie, sont étudiés les transferts de texte entre logiciels, ce qui permet la comparaison de ces logiciels, l'étude des associations possible en vue de perfectionnement des applications et les interfaçages éventuels pour faire passer des fichiers de l'un à l'autre. ■



Coll. "Micro-informatique professionnelle" DUNOD, 1987, 208 pages, broché, 145 F TTC.

PROGRAMMATION DU 65816

par William Labiak (SYBEX*)

Si vous possédez un Apple IIGS, vous programmerez tôt ou tard son 65816, version 16 bits du 6502... et il est certain que le manuel de William Labiak deviendra alors votre livre de chevet.

Il vous apprendra tout ce qu'il faut connaître sur l'organisation et le jeu d'instructions du nouveau microprocesseur d'Apple.

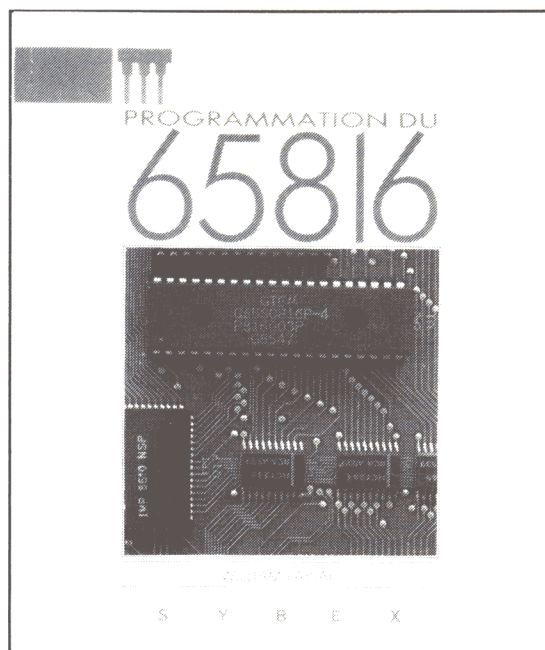
Il vous enseignera l'essentiel des techniques de programmation.

Je vous conseille de le lire de la première à la dernière ligne et de taper vous-même, sur le clavier de votre GS, les multiples exemples qui font de cet ouvrage un vrai manuel d'initiation.

De l'organisation interne du 65816 jusqu'au traitement des structures de données complexes, en passant par les modes opération 8 et 16 bits, les techniques d'émulation du 6502, les techniques d'adressage et la gestion des entrées/sorties, vous saurez (presque) tout sur la bête !

Attention ! il n'est aucunement question, dans ce livre, de l'Apple IIGS, mais seulement du microprocesseur dont il est doté. Ne vous attendez donc pas à y trouver des routines vous permettant de programmer LE GS en particulier. Il s'agit d'un ouvrage de base... mais tout à fait indispensable. **NESTOR**

* SYBEX — 6-8 impasse du Curé — 75018 PARIS (400 pages — 248 F TTC)



L'auteur est titulaire d'un diplôme d'ingénierie de l'Université de l'Illinois. Il participe au développement de réseaux informatiques utilisés pour le contrôle industriel et les applications scientifiques.

INTERFACE MIDI (TREMPIN MICRO N°16)
Notre ami Jean-Paul VERPEAUX a mal revu sa copie. Il n'est pas possible d'adapter son interface MIDI sur un Apple IIc, ce que tous les possesseurs de IIc savent bien. Nous le condamnons à pianoter sur un clavier de IIc jusqu'à ce que musique s'en suive !

DOCUMENTATION GS

A la demande d'APPLE France, D.D.A. met à la disposition des utilisateurs la documentation développeurs relative à cette machine. Il ne s'agit en aucun cas de logiciels, et surtout pas de notices grand-public. Ces documents, très utiles, sont en langue anglaise et sont tout, sauf agréables à manipuler.

Documents papier :

vol 1	ROM tools	300 F TTC
vol 2	RAM tools	320 F TTC
vol 3	Firmware	300 F TTC
vol 4	Hardware	300 F TTC
vol 5	APW (shell, A et C)	520 F TTC
vol 6	65816, introduction	490 F TTC
vol 7	ProDOS	380 F TTC

vol 8 Périphériques 230 F TTC

Disquettes 3P 1/2 :

APW	144 F TTC
APW A et C	144 F TTC
Samples	
Systèmes + outils	108 F TTC
Utilitaires	144 F TTC
Basic GS (interprété)	
notice + disquette	386 F TTC

APPLE indique volontiers que, pour utiliser les outils de développement, la configuration minimale comporte deux lecteurs 3P 1/2 et une carte extension équipée de 1 mégaoctet de RAM. Avec cet équipement, on se sentira très vite à l'étroit.

Dans ce numéro

HV PAVÉS (page 24) ALÉATOIRES

Nous nous sommes plantés, sur le //e et sur le GS avec la routine \$FC2F (qui fonctionne sur notre vieil Apple //c). Remplacez donc la ligne de data par ces quelques octets :

0300-	20 F5 E6	JSR	\$E6F5
0303-	8A	TXA	
0304-	20 C1 FB	JSR	\$FBC1
0307-	20 F5 E6	JSR	\$E6F5
030A-	8A	TXA	
030B-	18	CLC	
030C-	65 28	ADC	\$28
030E-	85 28	STA	\$28
0310-	A9 20	LDA	\$20
0312-	91 28	STA	(\$28),Y
0314-	60	RTS	

CLASSEMENT (p. 19)

Tel que le programme se présente, la simple présence d'un malheureux contrôle dans un mot fausse évidemment le classement. C'est ce qui s'est produit ici... d'où nécessité de soigner la saisie d'une liste de mots !

Une nouvelle imprimante Apple :

L'ImageWriter LQ

On la dit capable de se mesurer à la LaserWriter... et c'est évidemment exagéré. Cela revient à comparer un Apple //e à un Macintosh.

Reste que c'est une imprimante prometteuse, un haut de gamme. Sa tête d'impression comporte 27 aiguilles, ce qui lui permet de nous offrir une définition maximale de 216 points par pouce... contre 300 pour la Laser.

Ses vitesses d'impression sont variables, suivant la qualité demandée :

- **LQ** (Letter Quality) : 115 cps à 216 points par pouce (impression proportionnelle réellement soignée).
- **NLQ** (Near Letter Quality) : 140 cps à 144 pts par pouce (impression proportionnelle très acceptable).
- **DRAFT** : 250 caractères-seconde à 10 caractères par pouce (ça suffit pour un listage).

Naturellement, l'ImageWriter LQ est compatible avec tous les MAC et avec les Apple //GS et //c.

Elle accepte tous les types de papier (un chargeur feuille à feuille est prévu en option). Elle utilise des rubans de type continu (en noir ou en quatre couleurs) et ceux-ci sont vendus pour quatre millions de caractères. Enfin une bonne imprimante, comparable (soyons raisonnables), sur le plan de la qualité d'impression, à une machine à marguerite... avec la rapidité en plus, entre autres avantages.

Disponibilité à la fin de cette année à un prix sans doute voisin de 12000 F (non fixé actuellement).

NESTOR

LOGIQUE ÉLÉMENTAIRE (N°16 page 34) Une ligne de ce programme a été amputée lors de la mise en page. Nous vous la restituons... complète :

```
2690 IF A$(L,1) = A$(I,1) AND A$(L,2) = A$(I,2) AND A$(L,3) < > A$(I,3  
    ) AND A$(L,4) = A$(I,4) AND A$(L,5) = A$(I,5) THEN 2740
```

• **ASPECTS AVANCÉS DU LANGAGE C** (Ron PERSON)

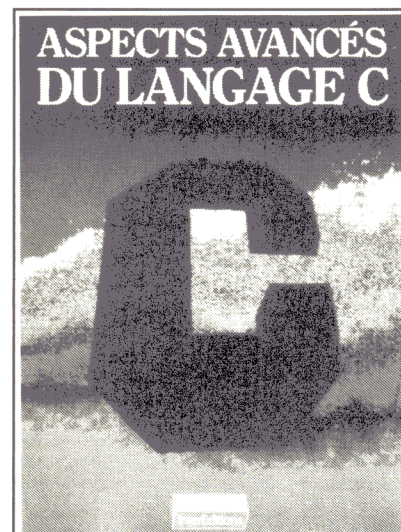
Cet ouvrage (traduit par Dimitri Stoquart) s'adresse à des programmeurs connaissant déjà les bases du langage C, mais désirant le maîtriser pour en tirer la quintessence.

Après une première partie consacrée aux principaux aspects de la programmation avancée en C, on examine de façon méthodique les techniques relatives aux listes chaînées et aux arborescences.

La dernière partie du livre traite des applications de génération de graphiques, de la création de menus et de la construction d'interfaces. Plus de 300 pages agréablement présentées sous une reliure souple très pratique, mais — soulignons-le — près de 100 pages dédiées au programme source du système de segmentation présenté au chapitre 7 : "Les graphismes d'application", puis, dans une autre annexe, le source des graphismes de l'IBM PC.

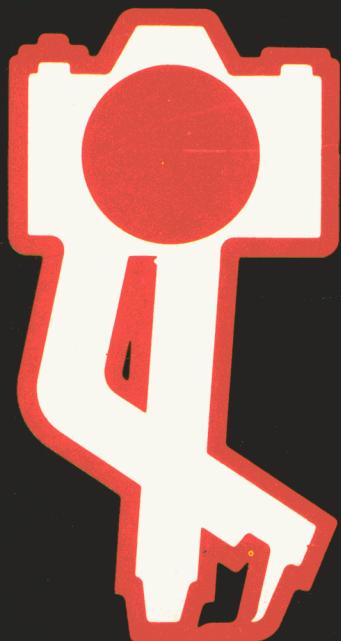
D'une manière générale, on peut dire que ce bouquin fourmille d'exemples. Très bonne édition à laquelle il ne manque qu'une disquette.

InterEditions, 87 avenue du Maine — 75014 PARIS.



Chasseur d'Images

**Chaque mois,
le meilleur
de la
technique
et de la
pratique
photo !**



**SPECIAL
SALON de la PHOTO**

Chasseur d'Images



N° 97
Novembre 1987
Distribution année
Canada 4,25 \$
Espagne 560 PTA
France 26 DM
Suisse 6 FS
Belgique 180 FB
FRANCE 20 F

**EXCLUSIF: un hologramme de collection
CAMESCOPIES : lequel choisir
Les kits pour développer vos diapos
Dossier nouvelles images**

**Chez votre
marchand de journaux !**