N° 20 - MAI - JUIN 1988

ISSN 0753 2864

tremplin micro

Duel (pacifique) avec votre Apple

Carte stéréo sur GS

Caractères personnalisés sur imprimantes Apple et Epson

Routines faciles en Basic

Le GS Basic est arrivé !

Et toujours le langage C ...

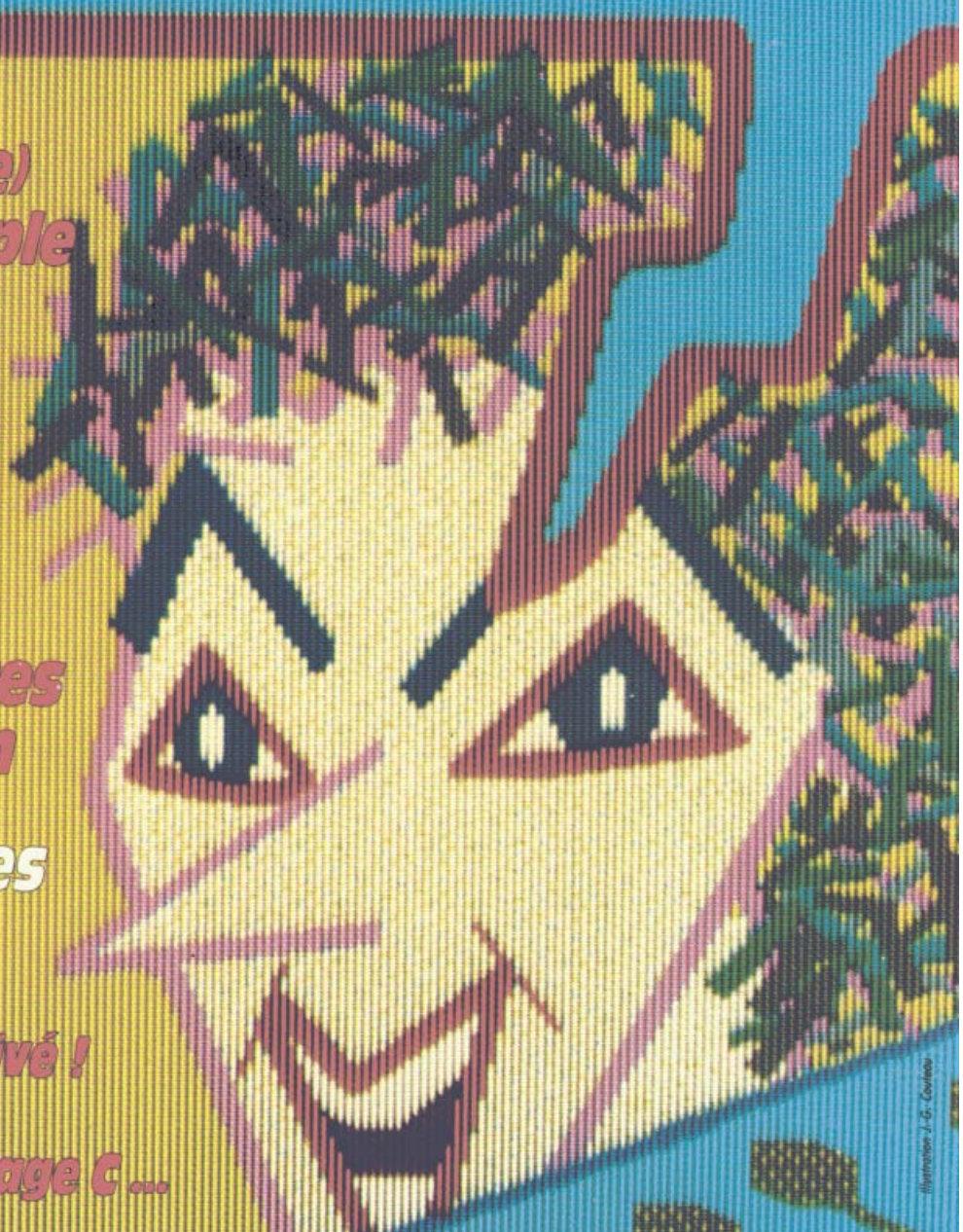


Illustration J.-G. Cuvilliez

M 1631 - 20 - 33,00 F

3791631033006 00200

N° 20 - Bimestriel - Quatrième année
5 Mai - 4 Juillet 1988
241 FB - 11 FS - 33 F

tremplin micro 20

SOMMAIRE

Avec la collaboration de :

Claude AUBRY, Marc FREFOYE, Yvan KŒNIG, Jean PERROT, Jacques REY, Emile SCHWARZ.

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 d'avantage des tarifs postaux réduits.

TREMPLIN MICRO — Bimestriel — C'est une publication des Editions JIBENA, 4, rue de la Cour-des-Noues, 75020 PARIS — S.A. au capital de 3600000 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 TREMPLIN MICRO contient tous les programmes du numéro, ainsi que les sources trop longs pour être publiés dans les colonnes de la revue.

Amusez-vous avec votre Apple II

La programmation est un jeu. Je ne cesse de le répéter depuis la création de TREMPLIN MICRO. Un jeu captivant en diable, mais qui exige une concentration d'esprit peu commune. Quand on programme, tout est dans la pensée, de la première à la dernière instruction. Logique et mémoire sont sollicitées à chaque instant. Bref, on ne pense plus qu'à l'objectif numéro un : obtenir, lors du **run** *final, le résultat attendu.*

La programmation passe évidemment par l'étude d'un langage, voire par celle de plusieurs langages. Nul ne me démentira si je prétends que cette étude constitue elle-même un jeu extraordinaire. Et puis, par l'intermédiaire de ces langages, n'avons-nous pas la possibilité de communiquer avec des partenaires du monde entier... grâce à un nombre limité de mots-clefs ?

Nouveaux langages

Je sais que bon nombre de nos Lectrices et Lecteurs s'intéressent de plus en plus au langage **C** *. Nous allons évidemment poursuivre notre patiente initiation, mais nous rencontrons encore des difficultés lorsqu'il s'agit d'adapter le* **C** *des IBM et clones sur notre GS. Souhaitons que les nouvelles versions de l'APWC soient plus compatibles. Il reste que le* **C** *est un langage portable par excellence et que son apprentissage est actuellement conseillé sur toutes les machines.*

Le GS Basic nous donne satisfaction, même s'il comporte encore quelques points noirs. Vous le découvrirez plus facilement avec l'excellent guide de notre ami Emile SCHWARZ (voir notre bulletin de commande). Ce livre ne vous sera utile que si vous disposez déjà de la documentation américaine et du logiciel (diffuseur : DDA).

On ne saurait concevoir de programmation sérieuse sur le GS sans passer par un langage au moins aussi performant que le GS Basic (qu'il sera possible de compiler).

Applesoft et Assembleur

Le bon vieil Applesoft reste le Basic facile par excellence. Son adaptation ne pose aucun problème sur une autre machine... si l'on se dispense d'utiliser les accès directs à la mémoire.

Il autorise moult fantaisies et permet de programmer n'importe quelle application. On peut lui reprocher sa lenteur (le GS Basic ne fait pas mieux !), mais il est facile de l'accélérer en utilisant ici et là quelques petites routines en Assembleur.

D'où la nécessité de **comprendre** *le* **langage machine.**

Ici, à **Tremplin Micro**, *nous avons déployé des trésors d'imagination pour aider les néophytes (mais aussi ceux qui n'en sont plus) à résoudre des dizaines de petits problèmes en langage machine.*

Nos recueils de routines (avec disquettes associées) constituent une base sérieuse et même indispensable à tout programmeur en Basic... soucieux de donner des ailes à ses programmes.

Toutes ces routines tournent sur Apple II avec 65C02, et la majorité d'entre elles fonctionnent aussi sur le GS. A quelques rares instructions près, elles s'accommodent également du 6502.

Je conseille vivement nos recueils à tous ceux qui désirent aller plus loin et mieux. Bonne programmation ! **G.-H.**

Un jeu simple et amusant : DUEL

Ce programme est en Basic Applesoft et ne comporte pas de difficultés particulières. Jean Perrot aurait peut-être pu élaborer une routine de traitement d'erreurs, mais libre à vous de détecter les possibles fautes de manipulation, puis de les traiter comme bon vous semblera.

- **LANCE.DUEL** explique la règle du jeu et n'est pas indispensable : vous pouvez en effet la lire ci-contre. Nous vous conseillons néanmoins de taper ce module car il comporte un écran de présentation agréable (voir page suivante).
- **DUEL** est le programme principal.
- **SON** (A$300,L24) porte bien son nom !
- **FONTE.G** est un tableau de formes (A$6000,L$2F2) contenant les numéros de 0 à 9 et les caractères...
- **PION3** (A$62F8,L159) est réservé aux pions noirs et blancs ainsi qu'au carré blanc.

REGLE DU JEU

Ce duel est un combat singulier entre 2 PIONS, évoluant dans le cadre d'une grille 9×9.

Le joueur utilise le pion noir et l'ordinateur joue avec le pion blanc.

Un pion ne peut se déplacer que d'une seule case tout autour de lui (à la manière du ROI aux échecs, c'est-à-dire dans toutes les directions, y compris en diagonale).

Celui qui joue doit accomplir deux tâches successives :

- *Déplacer son pion.*
- *Détruire une case inoccupée.*

Un pion ne peut pas jouer sur une case occupée par le concurrent... ou détruite.

BUT DU JEU :

Encercler l'adversaire pour qu'il ne puisse plus se déplacer. L'ordinateur tient les comptes et indiquera le nombre de coups nécessaires pour immobiliser l'adversaire.

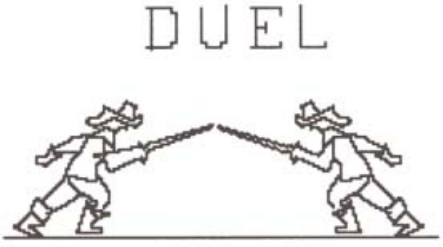
POUR DÉPLACER UN PION, entrer le nombre de 2 chiffres dont le chiffre de gauche est le numéro de la ligne et celui de droite le numéro de la colonne.

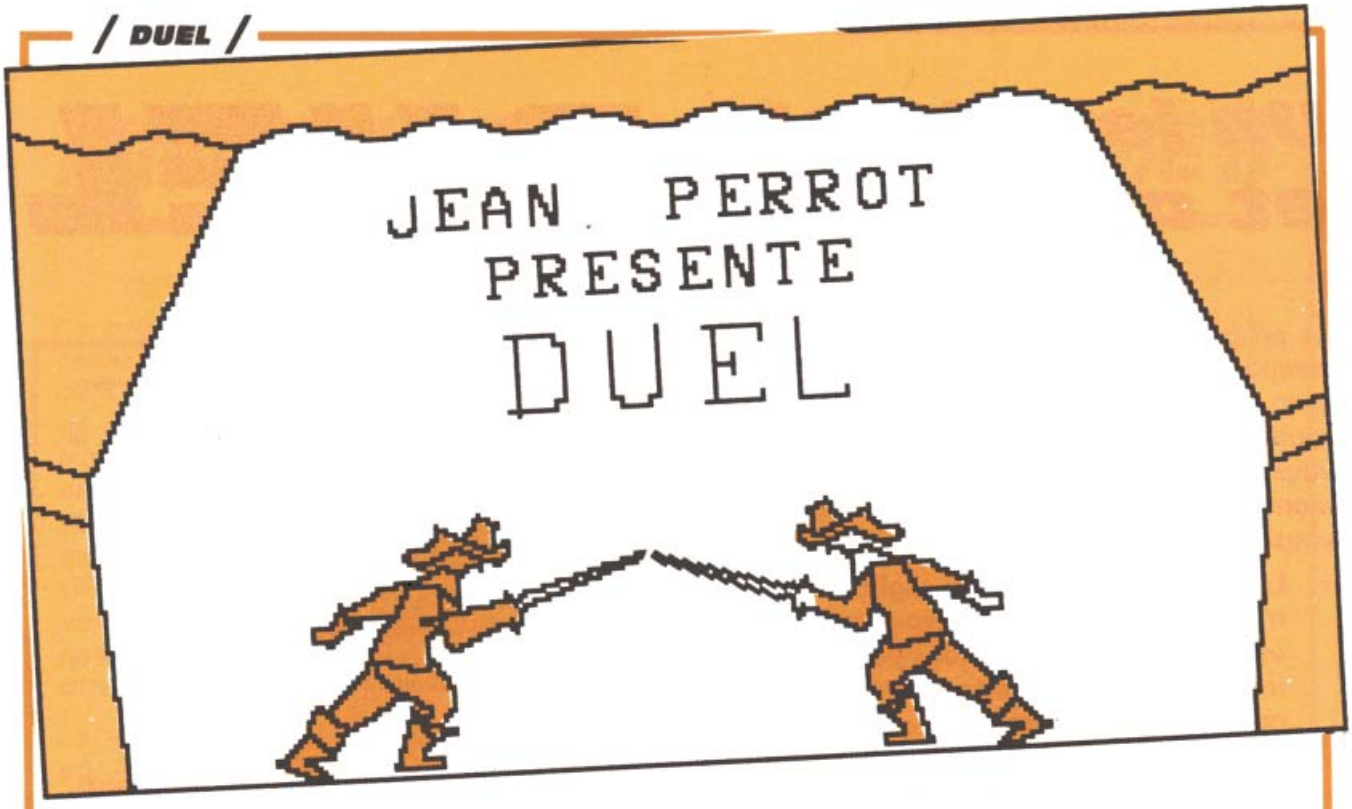
Si on entre 0 on va directement à la fin de jeu.

N.D.L.R. : Attention ! les erreurs peuvent avoir des conséquences fatales !

LANCE.DUEL

```
100 PRINT  CHR$ (17): HOME                                              252B
105 POKE 232,0: POKE 233,96: DIM F(24),X(97)                            8ED4
110 PRINT  CHR$ (4)"BLOAD SON"                                          A6DC
115 PRINT  CHR$ (4)"BLOAD FONTE.G"                                      27DD
120 HGR                                                                 2391
125 HCOLOR= 3: POKE 8,1                                                 2F4D
130 SCALE= 1: ROT= 0                                                    E3CC
135 GOSUB 385                                                           F150
140 FOR I = 1 TO 24: READ F(I): NEXT                                    784F
145 T = 14: HOME : POKE 769,5                                           C754
150 FOR I = 1 TO 12                                                     78EF
155 IF F(I) = 0 THEN 170                                                94E9
160 DRAW F(I) AT 66 + T,40                                              C651
165 FOR II = 1 TO 5:Z =  PEEK (49200): NEXT                             A25C
170 T = T + 10                                                          33A1
175 NEXT                                                                0582
180 T = 14                                                              A889
185 FOR I = 13 TO 20: DRAW F(I) AT 87 + T,55:T = T + 10                 1990
190 FOR II = 1 TO 10:Z =  PEEK (49200): NEXT II: NEXT I                 4F1F
```





LANCE.DUEL

```
195 SCALE= 2:T = 14                                                          378E
200 FOR I = 21 TO 24: DRAW F(I) AT 90 + T,80:T = T + 20                      218C
205 FOR II = 1 TO 10:Z =  PEEK (49200): NEXT II: NEXT I                      4F1F
210 FOR J = 1 TO 2                                                           40BF
215 FOR I = 1 TO 97: READ X(I): NEXT : GOSUB 420                             3EEB
220 NEXT J                                                                   A3CC
225 FOR J = 1 TO 10                                                          4EEE
230 GOSUB 500                                                                1B45
235 HCOLOR= 0: GOSUB 480: HCOLOR= 3: GOSUB 485                               E9D2
240 FOR I = 1 TO 20: NEXT                                                    48AA
245 HCOLOR= 0: GOSUB 490: HCOLOR= 3: GOSUB 495                               0FD4
250 GOSUB 505                                                                154A
255 HCOLOR= 0: GOSUB 485: HCOLOR= 3: GOSUB 480                               0FD2
260 FOR I = 1 TO 20: NEXT                                                    48AA
265 HCOLOR= 0: GOSUB 495: HCOLOR= 3: GOSUB 490                               FFD4
270 NEXT                                                                     0582
275 VTAB 22: HTAB 1: PRINT "Appuyer 'R' pour avoir la REGLE du JEU "         6B45
280 HTAB 4: PRINT "Une autre touche relance le duel ";                       D0E3
285 GET H$: PRINT : IF H$ = "R" THEN  HOME : TEXT : GOTO 515                 9975
290 HOME : GOTO 225                                                          4215
295 DATA 21,16,12,25,0,0,27,16,29,29,26,31                                   4ECD
300 DATA 27,29,16,30,16,25,31,16,15,32,16,23                                 A735
305 :                                                                        003A
310 DATA 58,54,51,60,62,57,63,58,60,66,63,64,72,78,75,73,72                  E54B
315 DATA 69,62,78,81,84,83,82,80,77,76,86,86,82,84,89,83                     82D8
320 DATA 80,84,87,89,89,84,84,72,75,78,81,74,67,66,64,60                     EDD5
325 DATA 60,63,66,70,67,78,84,84,87,87,81,87,91,91,87,87,89                  C572
330 DATA 102,103,105,132,105,103,101,102,83,91,91,93,96,96                   67EA
335 DATA 99,102,102,99,100,96,93,87,87,85,81,80,83,90,96,99,93               28F6
340 :                                                                        003A
345 DATA 208,212,215,206,204,209,203,208,206,200,203,202,194,188             B230
```

```
350 DATA 191,193,194,197,204,188,185,182,183,184,186,189,190,180          5973
355 DATA 180,184,182,179,183,186,182,179,177,177,182,182,194,191          A055
360 DATA 188,185,192,199,200,202,206,206,203,200,196,199,188,182          4339
365 DATA 182,179,179,185,179,175,175,179,179,177,164,163,161,134          9D5B
370 DATA 161,163,165,164,173,175,175,173,170,170,167,164,164,167          9B5A
375 DATA 166,170,173,179,179,181,185,186,183,176,170,167,173              1187
380 END                                                                   0180
385 HPLOT 50,20 TO 14,90 TO 14,100 TO 20,158 TO 1,158 TO 1,1 TO 278,1
    TO 278,158 TO 259,158 TO 265,100 TO 265,90 TO 228,20                  902B
390 HPLOT 14,100 TO 1,95 TO 1,85 TO 14,90: HPLOT 278,95 TO 265,100 TO
    265,90 TO 278,85                                                      F398
395 HPLOT 1,15 TO 3,15: FOR X = 3 TO 270 STEP 30: GOSUB 410: NEXT : H
    PLOT 274,15 TO 278,15                                                 2CA1
400 HPLOT 1,158 TO 278,158                                                BCBA
405 RETURN                                                                63B1
410 HPLOT X,16 TO X + 2,17 TO X + 8,18 TO X + 10,19 TO X + 13,20 TO X
    + 17,20 TO X + 20,19 TO X + 22,18 TO X + 24,17 TO X + 28,16 TO X
    + 30,16                                                               6898
415 RETURN                                                                63B1
420 HPLOT X(1),147 TO X(2),152 TO X(3),154 TO X(4),158 TO X(5),157 TO
    X(6),153 TO X(7),152 TO X(8),147 TO X(9),142 TO X(10),147 TO X(11
    ),152                                                                 1385
425 HPLOT X(12),148 TO X(13),144 TO X(14),140 TO X(15),137 TO X(16),1
    35 TO X(17),132 TO X(18),137 TO X(19),144                             A57F
430 HPLOT X(20),140 TO X(21),142 TO X(22),144 TO X(23),146 TO X(24),1
    45 TO X(25),143 TO X(26),153 TO X(27),156 TO X(28),156 TO X(29),1
    54 TO X(30),153 TO X(31),150 TO X(32),148 TO X(33),146                09FD
435 HPLOT X(34),143 TO X(35),150                                          F6FB
440 HPLOT X(36),146 TO X(37),143 TO X(38),140 TO X(39),132 TO X(40),1
    35 TO X(41),132 TO X(42),125 TO X(43),122 TO X(44),116 TO X(45),1
    18                                                                    69AB
445 HPLOT  TO X(46),122 TO X(47),120 TO X(48),125 TO X(49),126 TO X(5
    0),128 TO X(51),129 TO X(52),126 TO X(53),124 TO X(54),123 TO X(5
    5),122                                                                B875
450 HPLOT X(56),135 TO X(57),128 TO X(58),125 TO X(59),117 TO X(60),1
    16 TO X(61),113 TO X(62),115 TO X(63),122 TO X(64),117                7929
455 HPLOT X(65),125 TO X(66),131 TO X(67),125 TO X(68),128 TO X(69),1
    22 TO X(70),112 TO X(71),120 TO X(72),122 TO X(73),119 TO X(74),1
    21 TO X(75),125 TO X(76),122                                          CEEE
460 HPLOT X(77),115 TO X(78),113 TO X(79),114 TO X(80),112 TO X(81),1
    13 TO X(82),110 TO X(83),108 TO X(84),110 TO X(85),104 TO X(86),1
    02 TO X(87),107 TO X(88),104 TO X(89),107                             B38F
465 HPLOT  TO X(90),109 TO X(91),109 TO X(92),111 TO X(93),113 TO X(9
    4),109 TO X(95),112                                                   FE26
470 HPLOT X(96),110 TO X(97),107                                          5507
475 RETURN                                                                63B1
480 HPLOT 105,122 TO 132,112 TO 105,120: RETURN                           8702
485 HPLOT 105,122 TO 134,126 TO 105,120: RETURN                           0609
490 HPLOT 161,122 TO 134,112 TO 161,120: RETURN                           6F08
495 HPLOT 161,122 TO 132,126 TO 161,120: RETURN                           760B
500 POKE 768,14: POKE 769,30: CALL 770: RETURN                            8F66
505 POKE 768,17: POKE 769,15: CALL 770: RETURN                            C66C
510 REM ----------  REGLE DU JEU  ------------
515 HOME : PRINT  CHR$ (17)                                               C82B
```

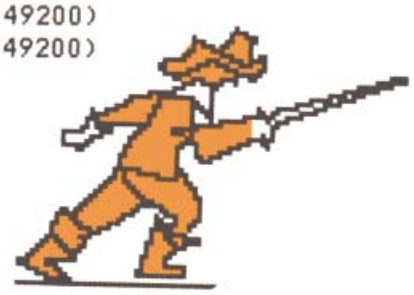
```
520 INVERSE : HTAB 15: PRINT " D U E L ": NORMAL : PRINT                    DAA1
525 PRINT " """Ce duel est un combat singulier """entre 2 PIONS évol
    uant dans le cadre"                                                    F91A
530 PRINT "d'une grille 9 x 9"                                             A25A
535 PRINT : PRINT " """"Le joueur joue avec le pion noir etAPPLE avec
     le blanc"                                                             0FA9
540 PRINT : PRINT " """"Un pion ne peut se déplacer que """d'une seul
    e case tout autour de lui """"(à la maniere du ROI aux échecs)"        2B92
545 PRINT : PRINT " """"Celui qui joue devra accomplir deuxtaches suc
    cessives :"                                                            188A
550 HTAB 5: INVERSE : PRINT "(A)";: NORMAL : PRINT " "DEPLACER SON PI
    ON"                                                                    5F9D
555 HTAB 5: INVERSE : PRINT "(B)";: NORMAL : PRINT " "DETRUIRE UNE CA
    SE INOCCUPEE"                                                          A35B
560 PRINT : PRINT " """"Un pion ne peut pas jouer sur une "case occup
    ée ou détruite"                                                        F39D
565 HTAB 28: PRINT "(RET) -->";                                            1790
570 GET H$: HOME                                                           61FB
575 HTAB 1: INVERSE : PRINT " BUT DU JEU ": PRINT : NORMAL : PRINT "
    """"Entourer l'adversaire pour qu'il nepuisse plus se déplacer."       69A3
580 PRINT : PRINT " """Il est interdit a un pion de jouer "sur une ca
    se détruite ou occupée"                                                E7B6
585 PRINT : PRINT " """"APPLE tient les comptes et indique le nombre
    de coups qu'il aura fallu pourimmobiliser l'adversaire"                1830
590 PRINT : PRINT " """POUR DEPLACER UN PION :entrer le nombre de 2 c
    hiffres dont le chiffre de gauche est le numéro de la ligne et ce
    lui de droite le numéro de la colonne"                                 41F1
595 PRINT : PRINT " """Si on entre 0 on va directement à la fin du je
    u"                                                                     DA48
600 HTAB 28: VTAB 22: PRINT "(RET) -->";                                   77D0
605 GET H$: HOME                                                           61FB
610 INVERSE : HTAB 7: PRINT "RENSEIGNEMENTS TECHNIQUES": NORMAL : PRI
    NT                                                                     7DE8
615 INVERSE : PRINT "LANCE.DUEL";: NORMAL : PRINT " lance le prog.pri
    ncip. ";: INVERSE : PRINT "DUEL": NORMAL                               C244
620 PRINT : HTAB 2: PRINT "Trois programmes L.M. sont utilisés :": PR
    INT                                                                    CE31
625 INVERSE : PRINT "SON";: NORMAL : PRINT " "(A$300,L24)": PRINT          2E6B
630 INVERSE : PRINT "FONTE.G";: NORMAL : PRINT " Tableau de formes (A
    $6000,L$2F2) "qui contient,dans l'ordre :"",0,1,2...9 puis A,B,C.
    ..X,Y,Z,-,ù,."                                                         AB15
635 PRINT : INVERSE : PRINT "PION3";: NORMAL : PRINT " (A$62F8,L159)
    qui contient 1 pion noir, 1 pion blanc et 1 carré blanc"               A682
640 HTAB 15: VTAB 22: INVERSE : PRINT "RETURN POUR COMMENCER";: NORMA
    L                                                                      2E5D
645 GET H$: HGR : HOME : TEXT : VTAB 10: HTAB 8: PRINT "B O N N E "C
    H A N C E !": VTAB 22                                                  05FB
650 PRINT  CHR$ (4)"RUN DUEL"                                              A8A9
```

SON (A$300,L24)

```
0300: 02 02 AD 30 C0 88 D0 05 CE 01 03 F0 09 CA D0 F5   2858
0310: AE 00 03 4C 02 03 60 00                           CC62
```

DUEL

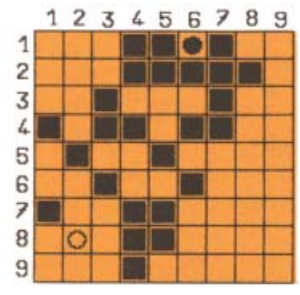
```
115 PRINT  CHR$ (4)"BLOAD SON"                                          A6DC
120 PRINT  CHR$ (4)"BLOAD FONTE.G"                                      27DD
125 PRINT  CHR$ (4)"BLOAD PION3"                                        6855
130 PRINT  CHR$ (4)"PR£3": PRINT  CHR$ (17)                             0AF6
135 HOME                                                                2F97
140 DIM C(121),D(8)                                                     A2A7
145 IV$ =  CHR$ (15):NR$ =  CHR$ (14)                                   449C

150 REM ===== INITIALISATION =====
155 GOSUB 795                                                           0F55
160 INVERSE : HTAB 12: PRINT "TIRAGE AU SORT": NORMAL : PRINT           97AE
165 PRINT "Je lance un DE et si j'obtiens un :'                         07B8
170 PRINT : PRINT IV$" 1 "NR$" c'est vous qui comencerez..."            ABB5
175 PRINT : PRINT IV$" 2 "NR$" c'est moi..."                            3D34
180 FOR I = 1 TO 2000: NEXT                                             290A
185 INVERSE : FOR I = 10 TO 12: VTAB I: HTAB 18: PRINT " "": NEXT :
    NORMAL                                                              5C80
190 FOR I = 1 TO 20:K =  INT ( RND (1) * 2) + 1                         55B9
195 HTAB 19: VTAB 11: PRINT K                                           447D
200 FOR J = 1 TO 12:Z =  PEEK (49200): NEXT J,I                         6F01
205 VTAB 15                                                             6408
210 IF K = 1 THEN  PRINT "C'EST DONC VOUS QUI COMMENCEZ"                5592
215 IF K = 2 THEN  PRINT "C'EST DONC MOI QUI COMMENCE"                  AFD1
220 VTAB 21: PRINT "PATIENTEZ QUELQUES INSTANTS...": FOR I = 1 TO 200
    0: NEXT                                                             9CE8

225 REM ======  LE CADRE  ======
230 SCALE= 1: ROT= 0                                                    E3CC
235 POKE 232,0: POKE 233,96                                             10D2
240 HGR : HCOLOR= 3                                                     F190
245 FOR I = 18 TO 153 STEP 15                                           458A
250 HPLOT 80,I TO 215,I:Z =  PEEK (49200)                               4DD4
255 HPLOT 62 + I,18 TO 62 + I,153:Z =  PEEK (49200)                     4F36
260 NEXT                                                                0582
265 FOR I = 1 TO 9                                                      0CC5
270 DRAW I + 2 AT 70 + I * 15,14:Z =  PEEK (49200)                      016B
275 DRAW I + 2 AT 70,15 * I + 14:Z =  PEEK (49200)                      946B
280 NEXT                                                                0582
285 SCALE= 2                                                            35CB
290 DRAW 15 AT 15,40: DRAW 32 AT 15,70                                  7BA6
295 DRAW 16 AT 15,100: DRAW 23 AT 15,130                                7B01
300 SCALE= 1                                                            37CA
305 POKE 232,248: POKE 233,98                                           8542
310 REM -- POSE DES PIONS ==
315 IF K = 2 THEN 345                                                   5E5A
320 HOME : VTAB 22: PRINT "Posez votre pion...Dans quelle case ";       F6D5
325 INPUT N                                                             97D2
330 GOSUB 670: IF L < 1 OR L > 9 OR C < 1 OR C > 9 THEN  GOSUB 775: C
    ALL - 198: GOTO 320                                                 A092
335 POKE 6,N:A = A + 1: GOSUB 765                                       A67A
340 DRAW 1 AT X,Y:C(NN) = 5                                             21D6
345 K = 2: FOR I = 1 TO 2000: NEXT                                      3391
350 :                                                                   003A
355 N = 55: GOSUB 670                                                   040F
```



```
360 IF C(NN) = 5 THEN L = 4:C =  INT ( RND (1) * 3) + 4:N = 40 + C:NN
    = N +  INT (N / 10) + 1: GOSUB 670                                435E
365 :                                                                 003A
370 POKE 7,N:B = B + 1: GOSUB 770                                     D979
375 DRAW 2 AT X,Y:C(NN) = 5                                           DAD7

380 REM --- DEROULEMENT DU JEU ---
385 REM =======( LE JOUEUR )========

390 K = 1: HOME : VTAB 22: PRINT "A vous de jouer... Quelle case ";   C69A
395 M =  PEEK (6)                                                     C786
400 N = M: GOSUB 670:X1 = X:Y1 = Y:L1 = L:C1 = C:N1 = NN              B183
405 INPUT M2: IF M2 = 0 THEN 610                                      4EC4
410 HCOLOR= 0: DRAW 1 AT X1,Y1:C(N1) = 0                              2712
415 N = M2: GOSUB 670:X2 = X:Y2 = Y:L2 = L:C2 = C:N2 = NN             AEBA
420 IF A = 0 THEN 430                                                 4E49
425 IF  ABS (L1 - L2) > 1 OR  ABS (C1 - C2) > 1 OR C(N2) = 5 THEN  GO
    SUB 775: GOTO 390                                                 09BA
430 IF L2 < 1 OR L2 > 9 OR C2 < 1 OR C2 > 9 THEN 390                  DB71
435 HCOLOR= 3: DRAW 1 AT X2,Y2:C(N2) = 5                              F81D
440 POKE 6,N:A = A + 1: GOSUB 765                                     A67A
445 REM --- CASE A DETRUIRE ---
450 PRINT : HOME : VTAB 22: PRINT "CASE à detruire..Laquelle ";       DF0F
455 INPUT M: IF M = 0 THEN 610                                        0C60
460 N = M: GOSUB 670:LD = L:CD = C:XD = X:YD = Y:ND = NN              5BE2
465 :                                                                 003A
470 IF LD < 1 OR LD > 9 OR CD < 1 OR CD > 9 OR C(NN) = 5 THEN  GOSUB
    775: GOTO 450                                                     38F1
475 GOSUB 730                                                         F94A

480 REM === TEST DE FIN DE PARTIE ===
485 N =  PEEK (7):NN = N +  INT (N / 10) + 1                          1BDB
490 FOR I = 1 TO 8                                                    0EC4
495 IF C(NN + D(I)) <  > 5 AND C(NN + D(I)) <  > 9 THEN 510            6A2E
500 NEXT I                                                            9DCB
505 K = 1: GOTO 610                                                   45C8

510 REM ===  JEU DU MICRO  ===
515 K = 2                                                             454D
520 HOME : VTAB 22: PRINT "Je joue... Patience !...";                 59EA
525 M =  PEEK (7)                                                     CD87
530 N = M: GOSUB 670:X1 = X:Y1 = Y:N1 = NN                            86EF
535 HCOLOR= 0: DRAW 2 AT X1,Y1:C(N1) = 0                              5213
540 :                                                                 003A
545 N = M: GOSUB 685: GOSUB 700                                       4B79
550 GOSUB 670:X2 = X:Y2 = Y:N2 = NN                                   164D
555 HCOLOR= 3: DRAW 2 AT X,Y:C(N2) = 5                                F9BA
560 POKE 7,N:B = B + 1: GOSUB 770                                     D979
565 N =  PEEK (6): GOSUB 685                                          2C14
570 GOSUB 700: GOSUB 670:XD = X:YD = Y:ND = NN                        6504
575 GOSUB 730                                                         F94A
580 :                                                                 003A
585 N =  PEEK (6):NN = N +  INT (N / 10) + 1                          27DA
590 FOR I = 1 TO 8                                                    0EC4
595 IF C(NN + D(I)) <  > 5 AND C(NN + D(I)) <  > 9 THEN 380            4C33
600 NEXT I                                                            9DCB
605 K = 2: GOTO 610                                                   33C9
```

```
610 REM === FIN DE PARTIE ===                                          A3DC
615 HOME : VTAB 22: PRINT "C'EST FINI !..LA PARTIE EST TERMINEE !.."   62BA
620 VTAB 23: PRINT "(RETURN) pour obtenir le score";                   9A90
625 FOR J = 1 TO 3: GOSUB 765: GOSUB 770: NEXT                        33BE
630 GET H$: HOME : TEXT                                               3F08
635 VTAB 12: HTAB 3                                                   598F
640 IF K = 1 THEN  PRINT "BRAVO !.. Vous avez gagne en ";A;" COUPS"   3863
645 IF K = 2 THEN  PRINT "DESOLE !.. Je suis le plus fort !..": PRINT
    "J'ai gagne en ";B;" coups"                                       6F65
650 VTAB 20: PRINT "UNE AUTRE PARTIE ? (O/N) ";: GET H$: PRINT H$     92EC
655 IF H$ = "O" THEN  RUN                                             B94E
660 TEXT : HOME : PRINT : PRINT  CHR$ (4)"RUN /T20/STARTUP"           0180
665 END
670 REM === TRAITEMENT DES CASES ===                                  2A95
675 L =  INT (N / 10):C = N - 10 * L                                  0DDD
680 X = 15 * C + 67:Y = 15 * L + 15                                   5919
685 NN = N +  INT (N / 10) + 1                                        63B1
690 RETURN                                                            C407
695 N = NN -  INT (N / 11) - 1: RETURN                                5095
700 I =  INT ( RND (1) * 8) + 1                                       360E
705 E = E + 1: IF E > 50 THEN 610                                     3F90
710 IF C(NN + D(I)) = 5 OR C(NN + D(I)) = 9 THEN 700                  8286
715 N = NN + D(I) -  INT ((NN + D(I)) / 11) - 1:E = 0                 63B1
720 RETURN                                                            003A
725 :                                                                 680F
730 REM === EFFACEMENT CASE DETRUITE ===                              3E0C
735 IF K = 1 THEN  GOSUB 765                                          51C5
740 IF K = 2 THEN  GOSUB 770                                          A0EA
745 HCOLOR= 3                                                         CE2B
750 DRAW 3 AT XD,YD + 1                                               63B1
755 C(ND) = 5                                                         77A2
760 RETURN                                                            5C95
765 POKE 768,255: POKE 769,80: CALL 770: RETURN                       185C
770 POKE 768,100: POKE 769,60: CALL 770: RETURN                       290A
775 HOME : VTAB 21: CALL  - 198: PRINT "ERREUR !... RECOMMENCEZ !...
    ":                                                                63B1
780 FOR I = 1 TO 2000: NEXT                                           003A
785 RETURN                                                            DF7A
790 :                                                                 931F
795 FOR I = 1 TO 121: READ C(I): NEXT                                 63B1
800 FOR I = 1 TO 8: READ D(I): NEXT                                   2DAE
805 RETURN                                                            7B5D
810 DATA 9,9,9,9,9,9,9,9,9,9,9                                        7B5D
815 DATA 9,0,0,0,0,0,0,0,0,0,9                                        7B5D
820 DATA 9,0,0,0,0,0,0,0,0,0,9                                        7B5D
825 DATA 9,0,0,0,0,0,0,0,0,0,9                                        7B5D
830 DATA 9,0,0,0,0,0,0,0,0,0,9                                        7B5D
835 DATA 9,0,0,0,0,0,0,0,0,0,9                                        7B5D
840 DATA 9,0,0,0,0,0,0,0,0,0,9                                        7B5D
845 DATA 9,0,0,0,0,0,0,0,0,0,9                                        7B5D
850 DATA 9,0,0,0,0,0,0,0,0,0,9                                        7B5D
855 DATA 9,0,0,0,0,0,0,0,0,0,9                                        2DAE
860 DATA 9,9,9,9,9,9,9,9,9,9,9                                        7819
865 DATA 1,12,11,10,-1,-12,-11,-10
```

DUEL



FONTE.G (A$6000,L$2F2)

```
6000: 28 00 52 00 56 00 6B 00 77 00 89 00 9A 00 A9 00  937E
6010: BB 00 CE 00 DD 00 F2 00 05 01 16 01 2C 01 3D 01  3DE0
6020: 4E 01 63 01 73 01 8A 01 9B 01 A5 01 B1 01 C6 01  3E6D
6030: D3 01 E8 01 F9 01 09 02 19 02 2D 02 41 02 53 02  4AA4
6040: 62 02 74 02 86 02 9D 02 B3 02 C6 02 D8 02 E0 02  2E3A
6050: EC 02 41 41 01 00 09 3C 24 24 24 25 2D 35 35 36  B014
6060: 36 3E 3E 3F 24 2C 2C 2C 24 04 00 49 29 3D 24 24  AEB8
6070: 24 24 37 37 37 06 00 29 2D 2D 3F 3F 27 25 25 25  438A
6080: 25 25 3C 3C 3F 3E 36 00 00 01 34 35 2D 25 25 3C  FA92
6090: 3C 27 25 25 25 3F 3F 3F 00 00 49 09 24 34 2E 3D  D6A4
60A0: 3F 3F 24 25 2C 24 25 04 00 01 34 35 2D 25 25 24  C545
60B0: 27 3F 37 27 24 2C 2D 2D 36 00 00 01 24 25 2D 35  3550
60C0: 35 3E 3E 3F 3C 24 24 24 25 2D 35 35 06 00 21 25  68A0
60D0: 25 25 3F 2D 3D 2C 2C 24 3F 3F 37 06 00 09 3C 24  C893
60E0: 25 2D 25 25 3C 3C 3F 3E 36 35 2D 35 35 3E 3E 3F  324E
60F0: 07 00 01 34 35 2D 25 25 24 24 3C 3C 3F 3E 36 35  A290
6100: 2D 25 2D 00 00 21 24 24 25 2C 2C 2E 36 35 36 36  CD6A
6110: 24 3C 3F 3F 07 00 2D 2D 25 25 3C 24 25 3C 3C 3F  38C5
6120: 3F 35 36 36 36 26 24 2C 2D 2D 00 00 09 2D 25 25  7866
6130: 3E 3E 3F 3C 24 24 24 25 2D 35 35 06 00 2D 2D 25  8EA4
6140: 25 24 24 3C 3C 3F 3F 35 36 36 36 36 00 00 29 2D  DBC6
6150: 2D 25 3E 3F 3F 24 24 2D 34 26 3F 24 24 2F 2D 2D  A6ED
6160: 35 06 00 2D 27 24 2C 25 36 3C 27 24 3C 2D 2D 2D  AE84
6170: 36 00 00 09 2D 25 25 24 3F 37 2C 2D 36 3E 3E 3F  859A
6180: 3C 24 24 24 25 2D 35 35 06 00 21 24 24 24 34 36  8561
6190: 2E 2D 2D 24 24 36 36 36 36 06 00 09 2D 27 24 24  A953
61A0: 24 2C 3F 07 00 01 24 36 35 2D 25 25 24 24 24 04  BB0D
61B0: 00 21 24 24 24 34 36 2E 2C 2C 2C 2C 37 37 37 37  23B1
61C0: 2E 2E 2E 2E 05 00 29 2D 2D 34 3F 3F 27 24 24 24  7485
61D0: 24 00 00 24 24 24 24 35 36 35 36 35 26 2C 24 2C  8561
61E0: 24 2C 36 36 36 36 06 00 21 24 24 24 2C 36 35 2E  B380
61F0: 36 35 2E 24 24 24 24 04 00 09 3C 24 24 24 25 2D  0B30
6200: 35 35 36 36 3E 3E 3F 07 00 2D 27 24 24 24 3C 2D  ACC1
6210: 2D 35 35 3E 3E 3F 3F 00 00 09 3C 24 24 24 25 2D  F294
6220: 35 35 36 36 3E 36 3C 3F 2D 24 3F 00 00 2D 27 24  7ECD
6230: 24 24 3C 2D 2D 35 35 3E 3E 3F 2F 2E 2E 2E 2E 05  BCEF
6240: 00 09 3C 34 35 2D 25 25 3C 3C 3F 3C 24 25 2D 35  6EC3
6250: 35 06 00 09 2D 27 24 24 24 2C 2D 26 3F 3F 3F 36  CE76
6260: 00 00 09 3C 24 24 24 34 36 36 36 35 2D 25 25 24  7A57
6270: 24 24 04 00 08 08 24 3C 24 34 36 35 36 35 2E 24  BB3C
6280: 25 24 25 24 04 00 21 24 3C 24 24 36 36 35 36 2E  8564
6290: 24 2C 24 36 35 36 25 24 2C 24 24 04 00 21 2C 2C  F04F
62A0: 2C 2C 2C 24 36 37 37 27 27 27 34 2E 2E 2E 2E 2E  A3DB
62B0: 36 06 00 49 21 24 3C 24 27 34 2E 36 35 36 2E 24  0BA6
62C0: 24 25 2C 24 04 00 29 2D 2D 34 3F 3F 27 2C 2C 2C  9B7D
62D0: 2C 2C 24 3F 3F 37 06 00 08 08 08 28 3F 3F 00 00  C9F5
62E0: 49 21 24 24 24 34 36 36 36 36 06 00 49 29 3C 2E  D8C4
62F0: 00 00                                            0000
```

PION3 (A$62F8,L159)

```
62F8: 03 00 08 00 1D 00 4B 00                          AE73
6300: 49 29 2D 2C 25 2C 24 3C 24 3F 3C 3F 3E 37 3E 36  7743
6310: 2E 36 2D 06 00 49 29 2D 2C 3D 3F 3F 3F 2C 2D 2D  84E2
6320: 2D 2C 3F 3F 3F 3F 27 2D 2D 2D 2D 25 3F 3F 3F 3F  7651
6330: 27 2D 2D 2D 2D 3D 3C 3F 3F 3F 2C 2D 2D 2D 3F 3C  453F
6340: 3F 00 00 2D 2D 2D 2D 2D 25 3F 3F 3F 3F 3F 27 2D  7CD4
6350: 2D 2D 2D 2D 25 3F 3F 3F 3F 3F 27 2D 2D 2D 2D 2D  4A1C
6360: 25 3F 3F 3F 3F 3F 27 2D 2D 2D 2D 2D 25 3F 3F 3F  554A
6370: 3F 3F 27 2D 2D 2D 2D 2D 25 3F 3F 3F 3F 3F 27 2D  523A
6380: 2D 2D 2D 2D 25 3F 3F 3F 3F 3F 3F 00 00 3F 3F 27  90F8
6390: 2D 2D 2D 2D 2D 2D 00                             EA0E
```

VÉRIF-COMPTE

Vous arrive-t-il de contrôler vos relevés de compte bancaire ? Sans aucun doute. Savez-vous pour autant où vous en êtes exactement ? Pas sûr ! Il est rare que le dernier relevé bancaire tienne compte des chèques récents.

De plus, certains débiteurs ne sont pas pressés de présenter leurs chèques à l'encaissement... d'où bon nombre de "problèmes".

Le petit programme de Jean Perrot n'a pas la prétention de gérer plusieurs comptes et ce n'est pas davantage un logiciel "professionnel". Néanmoins, tel qu'il se présente, il pourra rendre service à bien des Français moyens. Vous peut-être ?

Notez que la partie INSTRUCTIONS peut être négligée.

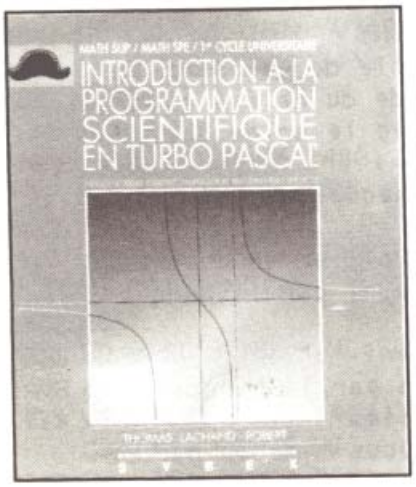
```
100 HOME : DIM ND(30),ND$(30),NC(10),NC$(10)                          C135
105 IV$ =  CHR$ (15):NR$ =  CHR$ (14):D$ =  CHR$ (4)                 187A
110 PRINT  CHR$ (4)"PR£3": PRINT :ND = 0:NC = 0:SB = 0:SC = 0         788C
115 PRINT D$"BLOAD SIRENE"                                            1BAE
120 VTAB 5: POKE 1403,35: PRINT IV$" "MENU "NR$                       079A
125 VTAB 8: POKE 1403,33: PRINT "Voulez-vous :": PRINT                C40C
130 PRINT IV$" 1 "NR$" Consulter les instructions ?": PRINT           CD45
135 PRINT IV$" 2 "NR$" Voir un exemple du tableau final ?": PRINT     4942
140 PRINT IV$" 3 "NR$" Commencer immédiatement ?": PRINT              BD61
145 PRINT IV$" 4 "NR$" Quitter le programme ?"                        1F65
150 PRINT : POKE 1403,33: PRINT "VOTRE CHOIX :"";                     781F
155 GET H$:H =  VAL (H$): IF  NOT H OR H < 1 OR H > 4 THEN 155        C6A3
160 ON H GOTO 580,670,175,165                                         729E
165 POKE 6,0: HOME : PRINT : PRINT : PRINT  CHR$ (4)"RUN STARTUP"     97F0
170 REM ========================================
175 HOME : PRINT "Retraits non enregistrés (taper '0' pour ARRETER)"  220F
180 FOR I = 1 TO 30: INPUT ND(I):P = ND(I): GOSUB 540:ND(I) = P:ND =
    ND + ND(I)                                                        C98A
185 ND$(I) =  STR$ ( INT (ND(I))):ND$ =  STR$ ( INT (ND))            4F50
190 CALL  - 998: PRINT " "Numéro ";I;" --> "";ND(I): GOSUB 560        C68A
195 IF ND(I) = 0 THEN D = I: GOTO 205                                 1A76
200 NEXT                                                              0582
205 GOSUB 545                                                         ED4E
210 P = ND: GOSUB 540:ND = P                                          3F21
215 :                                                                 003A
220 PRINT "Sommes non créditées (taper '0' pour ARRETER)"             BD27
225 FOR I = 1 TO 10: INPUT NC(I):P = NC(I): GOSUB 540:NC(I) = P:NC =
    NC + NC(I)                                                        BD82
230 NC$(I) =  STR$ ( INT (NC(I))):NC$ =  STR$ ( INT (NC))            4C4C
235 CALL  - 998: PRINT " "Numéro ";I;" --> "";NC(I): GOSUB 560        A289
240 IF NC(I) = 0 THEN C = I: GOTO 250                                 1974
245 NEXT                                                              0582
```

```
250 GOSUB 545                                                                    ED4E
255 P = NC: GOSUB 540:NC = P                                                     3F1F
260 :                                                                            003A
265 PRINT : INPUT "Solde de mon carnet :";SC:SC$ =  STR$ ( INT (SC)):            75C2
    P = SC: GOSUB 540:SC = P: GOSUB 560: CALL  - 998
270 POKE 1403,40: INPUT "Solde du relevé bancaire :";SB:SB$ =  STR$ (            5F72
    INT (SB)):P = SB: GOSUB 540:SB = P: GOSUB 560
275 PRINT : HTAB 9: PRINT "Voulez-vous modifier quelque chose ? (O/N)            97CC
    ";: GET H$: PRINT
280 IF H$ <  > "O" THEN 350                                                      0AA8
285 PRINT : PRINT "Taper (R) pour un retrait (V) pour un versement"               90B0
290 PRINT " (C) pour corriger le solde du carnet """(B) pour le rele             E07D
    vé bancaire"
295 PRINT : POKE 1403,25: PRINT IV$" R "NR$" ou "IV$" V "NR$" OU "IV$            B2C7
    " C "NR$" ou "IV$" B "NR$" ": GET Z$                                         107C
300 IF Z$ = "C" OR Z$ = "B" THEN 310                                             99DC
305 INPUT "Numéro ? ";N                                                          6682
310 INPUT "Nouveau montant ? ";MO
315 IF Z$ = "R" THEN NZ = ND(N):ND(N) = MO:ND = ND - NZ + ND(N):P = N            54C8
    D: GOSUB 540:ND = P
320 IF Z$ = "V" THEN NZ = NC(N):NC(N) = MO:NC = NC - NZ + NC(N):P = N            6FC5
    C: GOSUB 540:NC = P                                                          8448
325 IF Z$ = "C" THEN SC = MO                                                     E946
330 IF Z$ = "B" THEN SB = MO                                                     AEAB
335 PRINT : PRINT "AUTRE CORRECTION ? (O/N) ": GET H$                            64E0
340 IF H$ = "O" THEN 295

345 REM ======================================================
350 REM =======  AFFICHAGE DES RESULTATS ET CONCLUSION  =======
355 REM ======================================================
360 HOME                                                                         2F97
365 VTAB 1: GOSUB 545: VTAB 3: GOSUB 545: VTAB 19: GOSUB 545: VTAB 21           0B87
    : GOSUB 545                                                                  93CC
370 H = 0: GOSUB 550
375 VTAB 2: HTAB 5: PRINT "RETRAITS NON DEBITES";: HTAB 32: PRINT "VE            BD60
    RS.NON CRED.";: POKE 1403,57: PRINT "VERIFICATION"                           2DCA
380 FOR I = 1 TO D - 1
385 IF I < 16 THEN  VTAB I + 3: HTAB 2: PRINT I;: POKE 1403,10 -  LEN            F68D
    (ND$(I)): PRINT ND(I): GOTO 395
390 IF I > 15 THEN  VTAB (I - 12): POKE 1403,16: PRINT I;: POKE 1403,            F5D7
    24 -  LEN (ND$(I)): PRINT ND(I)                                              0582
395 NEXT                                                                         044F
400 GOSUB 555                                                                    C279
405 VTAB 20: POKE 1403,2: PRINT IV$" R "NR$;: POKE 1403,12: PRINT ND             5A07
410 H = 29: GOSUB 550                                                            FEC9
415 FOR I = 1 TO C - 1
420 VTAB (I + 3): POKE 1403,30: PRINT I;: POKE 1403,40 -  LEN (NC$(I)            21AF
    ): PRINT NC(I)                                                               0582
425 NEXT
430 SO = SC + ND - NC:SO$ =  STR$ ( INT (SO)):P = SO: GOSUB 540:SO =             9D02
    P
435 VTAB 20: POKE 1403,30: PRINT IV$" V "NR$;: POKE 1403,37 -  LEN (N            C800
    C$(I)): PRINT NC                                                             3305
440 H = 45: GOSUB 550
```

```
445 REM ==============================================
450 REM ============   VERIFICATION   ==============
455 REM ==============================================
460 VTAB 6: POKE 1403,47: PRINT "SOLDE CARNET ¨¨-->";: POKE 1403,74 -
    LEN (SC$): PRINT SC                                                  557C
465 VTAB 8: POKE 1403,51: PRINT IV$" R "NR$;" ¨¨¨--> ¨¨¨+";: POKE 140
    3,74 -  LEN (ND$): PRINT ND                                          B280
470 VTAB 10: POKE 1403,51: PRINT IV$" V "NR$;" ¨¨¨--> ¨¨¨-";: POKE 14
    03,74 -  LEN (NC$): PRINT NC                                         68AD
475 VTAB 11: POKE 1403,68: PRINT "__________"                            A547
480 VTAB 13: POKE 1403,54: PRINT "TOTAL ¨-->";: POKE 1403,74 -  LEN (
    SO$): PRINT SO                                                       6AD0
485 VTAB 16: POKE 1403,47: PRINT "SOLDE REL.BANC. -->";: POKE 1403,74
    -  LEN (SB$): PRINT SB                                               C901
490 H = 79: GOSUB 550                                                    3F0C
495 IF SO = SB THEN  VTAB 20: POKE 1403,55: PRINT IV$"TOUT EST CORREC
    T"NR$: GOSUB 565: GOTO 505                                           1B1F
500 IF SO <  > SB THEN  VTAB 18: POKE 1403,52: PRINT "DIFFERENCE -->
    ";(SO - SB): VTAB 20: POKE 1403,53: PRINT IV$" IL Y A UN PROBLEME
     "NR$: GOSUB 570: GOSUB 570: GOSUB 570                               23CE
505 VTAB 22: HTAB 4: PRINT IV$" D "NR$" =DEBUT" SPC( 10)IV$" C "NR$"
    = CORRIGER (noter les numéros)" SPC( 9)IV$" F "NR$" = FIN"           12AB
510 AA =  PEEK ( - 16384): POKE  - 16368,0                               9174
515 IF AA <  > 68 AND AA <  > 70 AND AA <  > 67 THEN 510                  5C49
520 IF AA = 68 THEN  RUN                                                 7ADD
525 IF AA = 67 THEN  HOME : GOTO 285                                     2B4B
530 IF AA = 70 THEN 165                                                  A8C6
535 :                                                                    003A
540 P =  INT (P * 100 + .5) / 100: RETURN                                5C61
545 FOR J = 1 TO 10: PRINT "--------";: NEXT : RETURN                    F970
550 FOR J = 1 TO 21: VTAB J: POKE 1403,H: PRINT "!": NEXT : RETURN       6E45
555 FOR J = 4 TO 18: VTAB J: POKE 1403,15: PRINT "!": NEXT : RETURN      306C
560 FOR II = 1 TO 30:ZZ =  PEEK (49200): NEXT : RETURN                   9CCF
565 FOR II = 1 TO 2: CALL 781: NEXT : RETURN                             ED14
570 FOR II = 1 TO 20:BZ =  PEEK (49200): NEXT : RETURN                   FBB6

575 REM ------------------------------------
580 REM ---------  INSTRUCTIONS  ------------
585 REM ------------------------------------
590 HOME                                                                 2F97

595 POKE 1403,33: PRINT IV$" INSTRUCTIONS "NR$: PRINT                    51BB
600 PRINT " ¨¨¨¨¨¨¨Chacun sait combien il est pénible de verifier la
    gestion d'un compte": PRINT "bancaire . Le solde du relevé envoyé
     par la banque ne coincide jamais, hélas! ¨¨avec le solde que vou
    s avez calcul2."                                                     9BA1
605 PRINT " ¨¨¨¨¨¨¨En effet, il y a toujours des chèques non débités
    et des versements non crédités"                                      A345
610 PRINT " ¨¨¨¨¨¨¨Le but de ce programme est de vérifier qu'aucune e
    rreur n'a été commise": PRINT "ni par vous-meme ni par la banque"
    : PRINT                                                              5DF3
615 PRINT " ¨¨¨¨¨¨¨Il suffira d'entrer successivement ¨:"                0CE9
620 PRINT "-1 les montants des retraits non débités par la banque"       B9A1
625 PRINT "-2 les montants des versements non crédités"                  4107
630 PRINT "-3 le solde que vous avez calculé (par vous-meme ou par or
    dinateur)"                                                           3FB9
```

```
635 PRINT "-4 le solde du relevé bancaire"                                  E5CE
640 PRINT : PRINT "       ""Le programme permet de corriger les erreur
    s soit à la fin de la SAISIE": PRINT "soit après l'affichage du t
    ableau final. Dans un cas comme dans l'autre il"                        A8C3
645 PRINT "faut bien noter les NUMEROS des sommes enregistrées."            2465
650 VTAB 22: POKE 1403,40: PRINT IV$" T.S.V.P. "--> "NR$                    2D2D
655 GET H$: HOME                                                            61FB
660 GOTO 120                                                                113E
665 END                                                                     0180
670 REM ========  EXEMPLE  ========                                         44BA
675 FOR I = 1 TO 20: READ ND(I):ND$(I) =  STR$ ( INT (ND(I)))               9F43
680 ND = ND + ND(I):P = ND: GOSUB 540:ND = P                                3519
685 NEXT :D = I                                                             1071
690 ND$ =  STR$ ( INT (ND))                                                 448A
695 FOR I = 1 TO 5: READ NC(I):NC$(I) =  STR$ ( INT (NC(I)))               7572
700 NC = NC + NC(I):P = NC(I): GOSUB 540:NC(I) = P                          CE18
705 NEXT :C = I                                                             A16F
710 NC$ =  STR$ ( INT (NC))                                                 617B
715 SB = 2440.49:SB$ =  STR$ ( INT (SB))                                    D683
720 SC = 7341.85:SC$ =  STR$ ( INT (SC))
725 SO = SC + ND - SC:P = SO: GOSUB 540:SO = P:SO$ =  STR$ ( INT (SO)
    )                                                                       CE07
730 RESTORE                                                                 5DAE
735 DATA 123.5,254.61,1281.75,1500,264.25,32.3,385.65,53.25,2763.51         C8CB
740 DATA 693.37,259.2,82.9,164.15,281.5,390.27,345.67,289.34,1200,876
    .36,872.5                                                               6418
745 DATA 12953.67,795.36,410.2,343.51,2512.7                                9960
750 GOTO 350                                                                3F43
755 GET H$:H =  VAL (H$): HOME : IF  NOT H OR H < 1 OR H > 3 THEN 155       CE73
```

```
SIRENE,A$0300,L$0054

0300: A9 01 A2 FF A0 FF 4C 1F 03 A9 01 A2 FF A0 01 4C   AA90
0310: 1F 03 A9 80 A2 06 A0 FF 4C 1F 03 00 1F 9D 42 8D   CE8B
0320: 1B 03 8C 1C 03 8E 1D 03 A9 00 8D 1E 03 08 78 AE   36FC
0330: 1D 03 AD 1E 03 6D 1C 03 8D 1E 03 A8 88 D0 FD AD   E2D2
0340: 30 C0 8A A8 88 D0 FD AD 30 C0 CA D0 E5 CE 1B 03   AA7F
0350: D0 DD 28 60                                       3F35
```



• INTRODUCTION À LA PROGRAMMATION SCIENTIFIQUE EN TURBO PASCAL (par Thomas Lachand-Robert)

Cette introduction à la programmation scientifique et technique en Turbo Pascal se destine plus particulièrement aux étudiants des deux premières années d'université et aux élèves des classes préparatoires de mathématiques supérieures et spéciales.

L'auteur, Thomas Lachand-Robert, actuellement élève à l'Ecole Polytechnique, a voulu apporter à tout jeune étudiant une première approche de la programmation scientifique sans toutefois proposer un cours. Le lecteur y trouvera des chapitres traditionnels de mathématiques abordant, entre autres, l'algèbre linéaire, le calcul différentiel et intégral, et des exemples tirés de la physique et de nombreux programmes graphiques tels, les tracés de courbes, les surfaces, les champs différentiels.

Les nombreux programmes et modules proposés sont tous accompagnés de commentaires qui donneront au lecteur des indications sur la nature d'un certain nombre de problèmes et certaines solutions que l'on peut y apporter. L'utilisateur, grâce à cet ouvrage, pourra se contruire une première couche technique à partir de laquelle un travail bien structuré sera immédiatement profitable.

Livre broché, format 190 X 230 — 288 pages — 178 F **SYBEX**, *6-8 Impasse du Curé, 75018 PARIS — Tél. : (16-1) 42.03.95.95.*

J'AI MIS UN PALMIER dans mon Apple GS

En fait, il y a deux sortes de programmeurs, ceux qui font de l'informatique pour s'amuser ou pour épater les copains (on les appelle des amateurs), et puis les autres, les pros, tous ceux qui gagnent leur vie à la sueur de leurs doigts manucurés.

Les premiers ont tout leur temps — quand on aime on ne compte pas —, ils programment en **Basic**, en **Pascal**, en **C** ou mieux (ou pire) en **Assembleur**. Les autres ont toujours une semaine de retard sur leur planning et utilisent des outils de programmation puissants tels que **DBASE** ou **MEMSOFT**.

Je me souviens du temps archaïque où l'importateur de Commodore vendait ses **CBM 8032** comme des grille-pain : quelques pages de doc et une disquette système. Pour cinq ou six mille francs de plus, on avait droit à une gestion de fichier en Basic (bugguée), une compta (archi-bugguée), un traitement de texte (en américain) et un jeu du pendu en suédois.

Pour les développeurs assoiffés dans ce désert, **MEMSOFT** arrivait comme une oasis : pensez donc, le Basic du **CBM** gagnait le séquentiel indexé, les masques de saisie et surtout, on pouvait utiliser des disques durs de capacité quasi-illimitée à une époque où les plus courageux échangeaient leurs lecteurs de cassettes contre des disquettes 360 K.

Cela dit, le **CBM** n'était pas vraiment un marché porteur et une deuxième version de **MEMDOS** allait apparaître, sur **ITT2020** d'abord (le premier clône Apple) puis sur l'**Europlus**, le **IIE**, le **IIC**, l'**IBM** et puis tout récemment, sur l'**APPLE GS**.

Pour éviter la redoutable pompe à logiciel qui, comme chacun feint de l'ignorer, fonctionne à plein rendement sur l'Apple, les concepteurs futés allaient distribuer leur produit sous la forme d'une carte pleine de puces, moins facile à copier qu'une disquette 5 pouces. Voici comment, grâce à leur ingéniosité commerciale et à la qualité de leur logiciel, ils sont encore là, à se faire bronzer le clavier, au soleil des Alpes-Maritimes.

UN DESIGN D'ENFER

Il y a sept ans, lorsque j'ai acheté mon premier **MEMDOS**, j'avais eu droit à une carte imprimée entourée de scotch et de plastique pustuleux avec une disquette du même acabit. Aujourd'hui, je dois dire, ils font fort ! Je suis ressorti de chez mon fournisseur préféré avec une sorte de petite mallette noire (style pistolet de collection) : sur l'emballage, le sigle **MEMSOFT** et un palmier (genre Paris-Dakar) ; dans la boîte, la carte pleine de puces précitée, une disquette 3.5 pouces, et un bouquin de 487 pages décrivant le BASIC **MEMSOFT** en détail.

Si vous ne savez pas quoi offrir à votre PDG, faites-lui cadeau de **MEMSOFT** ; même s'il n'utilise pas le logiciel, il pourra garder la boîte, c'est classe !

Y-A-T'IL UN ROULEAU DE SCOTCH DANS LA BOÎTE ?

Avant tout, ouvrir l'Apple GS. Opération facile si on dispose de trois mains et que l'on est premier prix de conservatoire de flûte à quinze trous. Voilà, c'est fait. Je constate au passage que ma carte d'extension mémoire est toujours là (condition sine qua non au fonctionnement de MENDOS) ; introduction de la carte dans le slot 4 comme il est dit à la page 17 de la doc (dont les pages 1 à 29 viennent de se détacher...). En plus de la mallette, votre PDG aura un puzzle...

Terminé. Fermeture de l'Apple dans un bruit de noisette écrasée. Introduction de la disquette système dans le lecteur 1.

Savez-vous pourquoi j'aime bien le GS ? Eh bien ! c'est la seule machine actuelle à ne pas avoir de ventilateur ! *(suite page 16).*

J'allume donc silencieusement la bête qui sommeillait ; elle avale la jolie petite disquette blanche avec prodos, la jolie petite disquette bleue (avec palmier) dans son deuxième lecteur... MIAMM, SLURRRP, RAAACLLLLE, BLURRRRP...

UN RÉGAL POUR MA POMME

Heureusement les gens de **MEMSOFT** sont plus doués pour écrire des logiciels que pour relier des bouquins. Le temps de cliquer sur l'icône "**MENU**" et je suis convaincu que MEMDOS n'est pas une nouvelle version de **LOAD RUNNER** mais bien un outil de programmation, et je sélectionne **MEMSOFT**.

Trois-quarts de tour de disquette plus loin, je me retrouve en paysage familier : ça ressemble à **MEM BASIC IBM** comme deux gouttes d'eau.

Alors voilà, **MEMSOFT** ça se compose de trois parties principales. La première, c'est **MEMBASIC** : un Basic puissant et structuré, un beau Basic de gestion avec une précision de 14 chiffres significatifs des instructions comme **DO..LOOP**, **SELECT..CASE**, **IF..THEN..ELSE** multilignes avec la possibilité d'utiliser des étiquettes de branchement.

Pour le debbugage, **MEMBASIC** permet de tracer le programme dans une fenêtre indépendante... un vrai plaisir.

Mais **MEMSOFT**, ce n'est pas seulement un **BASIC** puissant ; c'est aussi, et surtout, un gestionnaire de fichiers qui vous libère de la programmation laborieuse, pénible et répétitive des classiques créations, recherches, effacements de fiches.

MEM-FILE structure ses données en arbres B, système présentant un bon compromis entre les organisations arborescentes et les organisations séquentielles. Si vous avez déjà écrit une gestion de fichiers en arbres B, vous savez que l'écriture des modules de base n'est pas à la portée du premier basicois venu. **MEMSOFT** permet à ce même basicois d'utiliser ces structures avec presque autant de facilité qu'il a écrit, il y a une demi-heure et pour la première fois, **PRINT "BONJOUR"**.

C'en est écœurant : les applications de haut niveau en gestion compta etc. deviennent d'une telle facilité que l'on se dit que pour continuer à programmer dans un autre langage, il faut soit être masochiste, soit ne pas avoir lu cet article ; d'autant que les gens de MEMDOS ont crée un autre utilitaire : **MEM-SCREEN** qui est, à la saisie, ce que **MEM-FILE** est au fichier : vous définissez une fenêtre, des textes à afficher, des zones de saisie, des formules de calcul, des contrôles sur les saisies et des formats d'affichage ; **MEM-SCREEN** se débrouille pour gérer tout cela.

NON JE NE SUIS PAS VENDU À MEMSOFT

Pour ne rien vous cacher, MEMSOFT, j'aime bien, comme j'aime bien tous les bons produits qui nous facilitent la vie à nous autres, pauvres informaticiens. C'est vrai quoi, à force de ne penser qu'à l'utilisateur final et à son confort, on oublie un peu que les développeurs passent tout de même pas mal de temps devant leur clavier et ont des taux d'adrénaline inversement proportionnels à la vitesse de leurs compilateurs.

Alors vive **BORLAND**, bravo **MEMSOFT** !
Eh oui ! En France, on a plein d'autres bonnes choses, en dehors du Camembert et du Bourgogne !

RANG

Comment déterminer quel est le plus grand nombre d'une série ?

Cette routine-exemple construit d'abord une série de 45 nombres aléatoires, plus petits que 999 (ligne 125). Notez que la virgule de la ligne 130, à droite de N(I), permet un affichage automatique sur 3 colonnes. Pour obtenir un bon alignement, on augmente la valeur de SP (SPC) quand le numéro du rang est inférieur à 10. La recherche est effectuée dans la boucle des lignes 135-150. Le cas de plusieurs nombres identiques n'est pas prévu.

```
100 TEXT : PRINT  CHR$ (17): HOME                                          B0EE
105 DIM N(45)                                                              578E
110 INVERSE : PRINT " ** LE PLUS GRAND NOMBRE D'UNE SERIE ** "             596D
115 NORMAL : VTAB 3                                                        29AC
120 FOR I = 1 TO 45                                                        5CF5
125 N(I) = 1 +  INT ( RND (1) * 998):SP = 2: IF I < 10 THEN SP = 3         FB51
130 PRINT I; SPC( SP);N(I),: NEXT                                          3CD8
135 FOR I = 2 TO 45                                                        4AF6
140 IF N(I) <  = N(CHERCHE) THEN 150                                       EA21
145 CHERCHE = I                                                            E40B
150 NEXT                                                                   0582
155 VTAB 20: INVERSE : PRINT N(CH);: NORMAL                                4D0C
160 PRINT " est le plus grand - position: ";: INVERSE : PRINT CH: NOR
    MAL                                                                    B950
165 VTAB 22: PRINT "PRESSEZ UNE TOUCHE ";: GET R$: VTAB 2: CALL  - 95
    8: LIST 120,160                                                        92B2
170 PRINT : VTAB 22: PRINT "(E)ncore (B)asic (S)tartup disquette:ctG
    ";: GET R$                                                             AC98
175 IF R$ = "E" OR R$ = "e" THEN  RUN                                      5A89
180 IF R$ = "B" OR R$ = "b" THEN  HOME : END                               3728
185 IF R$ = "S" OR R$ = "s" THEN  PRINT  CHR$ (4);"RUN /T20/STARTUP"       92FA
190 GOTO 170                                                               3F43
```

ALEA

Mélange aléatoire d'une série de nombres

Le procédé est classique. Il fonctionne de la même manière avec des variables alphanumériques. Vous l'utiliserez dans bien des cas : anagramme d'une phrase, mélange de mots... Notez que la liste de base est détruite, ce qui est ici sans importance, mais pourrait devenir gênant dans bien des cas. Il conviendrait donc de la sauvegarder éventuellement dans une autre variable. A la ligne 155, la variable incrémentée à chaque tour de boucle, est utilisée pour créer un saut de ligne supplémentaire. On obtiendrait le même résultat avec IF 1/3 = INT (1/3) THEN PRINT.

```
100 TEXT : PRINT  CHR$ (17): HOME                                          B0EE
105 NOMBRE = 24: DIM NBASE(NO),NMELE(NO)                                   689B
110 INVERSE : PRINT " ** LE MELANGE ALEATOIRE D'UNE SERIE ** "             4469
```

```
115 NORMAL : VTAB 3                                                    29AC
120 FOR I = 1 TO NO:NBASE(I) = I: NEXT                                 C93B
125 FOR I = NOMBRE TO 1 STEP  - 1                                      9D10
130 MELE = 1 +  INT ( RND (1) * I)                                     E380
135 NMELE(I) = NBASE(MELE)                                             7FB8
140 NBASE(MELE) = NBASE(I): REM Attention! liste de base détruite      16B0
145 NEXT                                                               0582
150 FOR I = 1 TO NOMBRE:E$ = " ": IF I > 9 THEN E$ = ""                119F
155 PRINT E$;: INVERSE : PRINT I;: NORMAL : PRINT " "NMELE(I),:J = J
    + 1: IF J = 3 THEN  PRINT :J = 0                                   1EE1
160 NEXT                                                               0582
165 VTAB 22: PRINT "PRESSEZ UNE TOUCHE ";: GET R$: VTAB 2: CALL  - 95
    8: LIST 120,160                                                    92B2
170 PRINT : VTAB 22: PRINT "(E)ncore (B)asic (S)tartup disquette:ctG
    ";: GET R$                                                         AC98
175 IF R$ = "E" OR R$ = "e" THEN  RUN                                  5A89
180 IF R$ = "B" OR R$ = "b" THEN  HOME : END                           3728
185 IF R$ = "S" OR R$ = "s" THEN  PRINT  CHR$ (4);"RUN /T20/STARTUP"   92FA
190 GOTO 170                                                           3F43
```

TOTAUX

Double totalisation des nombres d'une table

Là encore, il s'agit d'un exemple. Les nombres sont aléatoires, mais pourraient naturellement provenir du clavier. Le nombre de colonnes peut être augmenté, mais il serait alors souhaitable, voire nécessaire, de passer en mode 80 colonnes. Vous adopterez facilement cette routine pour vous programmer un mini-tableur !

```
100 TEXT : PRINT  CHR$ (17): HOME                                      B0EE
105 LIGNES = 15:COL = 4: DIM TABLE(LIGNES,COL),T(LIGNES),V(COL + 1)    9E96
110 INVERSE : PRINT " ** DOUBLE TOTALISATION D'UNE TABLE "** "          898F
115 NORMAL : VTAB 3:NALEA = 99998: REM Nbre maximum-1                  1D33
120 FOR I = 1 TO LIGNES                                                5B4E
125 FOR J = 1 TO COL                                                   1D6B
130 TABLE(I,J) = 1 +  INT ( RND (1) * NA)                              ED1B
135 HTAB (J * 7) + 1 -  LEN ( STR$ (TABLE(I,J))): PRINT TABLE(I,J);    827C
140 T(I) = T(I) + TABLE(I,J):V(J) = V(J) + TABLE(I,J)                  8518
145 NEXT :V(COL + 1) = V(COL + 1) + T(I)                               1D3E
150 PRINT " > "; RIGHT$ ((" "+ STR$ (T(I))),7): NEXT                   E970
155 FOR I = 1 TO 10: PRINT "----";: NEXT                               60D0
160 FOR I = 1 TO COL + 1:AF$ =  RIGHT$ ((" "+ STR$ (V(I))),7)
    : IF I = COL + 1 THEN  PRINT " > ";                                B869
165 PRINT AF$;: NEXT : PRINT                                           5450
170 PRINT : VTAB 22: PRINT "(E)ncore (B)asic (S)tartup disquette:ctG
    ";: GET R$                                                         AC98
175 IF R$ = "E" OR R$ = "e" THEN  RUN                                  5A89
180 IF R$ = "B" OR R$ = "b" THEN  HOME : END                           3728
185 IF R$ = "S" OR R$ = "s" THEN  PRINT  CHR$ (4);"RUN /T20/STARTUP"   92FA
190 GOTO 170                                                           3F43
```

/ Claude AUBRY /

DEUS EX MACHINA

Il y a quelque temps Dieu, ayant décidé d'informatiser le paradis, fit venir un bataillon d'archanges-analystes pour définir un projet à la dimension de sa grandeur.

Il s'agissait de calculer la surface de nuages dont disposait chaque élu et de vérifier que cette surface était bien conforme à ses mérites, mesurés en norme ANSY. Le directeur du projet, un ange écossais du nom de MAC-INTOSH proposa de stocker en mémoire éthérée (nous, nous dirions plutôt vive), et sous forme de tableau, un nombre fixe de bienheureux, repérables grâce à une icône en forme de paire d'ailes. Dieu pourrait ainsi étudier le cas de chacun en le désignant à l'aide de la souris placée à sa droite.

***DIEU** qui avait d'autres choses en tête dit : "Que cela soit". Et cela fut.*

Or, il advint que MAC-INTOSH, soucieux d'économiser la mémoire éthérée, n'avait réservé de place que pour 8 milliards de candidats, et que le 8 milliard unième était justement le pape.

Vu l'impossibilité d'expliquer à un pape que son admission au paradis était impossible à cause d'un Bug de l'Ordinateur central, Dieu fut mis au courant, et entra dans une grande colère.

Il convoqua MAC-INTOSH et le renomma MAC II, puis il dit très fort afin que chacun entende Sa Parole : "LA GESTION DE LA MÉMOIRE ÉTHÉRÉE DEVRA ÊTRE DYNAMIQUE CHAQUE FOIS QUE LE NOMBRE DE DONNÉES NE SERA PAS PRÉVISIBLE !".

Puis Dieu créa les listes chaînées et les arbres binaires, et Dieu vit que cela était bon et retourna se coucher. ■

LA GESTION DYNAMIQUE DES DONNÉES

Un problème angoissant se pose à tout programmeur consciencieux lorsqu'il désire mémoriser un certain nombre de données, nombre à priori indéterminé au moment de l'écriture du programme.

En Basic APPLESOFT, la seule structure existante étant le tableau dimensionné à l'avance par l'instruction DIM, trois solutions sont possibles :

1. On surdimensionne le tableau et l'on gaspille ainsi de la place mémoire.
2. On écrit une bidouille en langage machine qui permettra de redimensionner le tableau au fur et à mesure des besoins.
3. On écrit un programme en langage machine qui va réserver de la place à une adresse mémoire à laquelle Basic n'a pas accès, stocker les données, retourner au programme appelant l'adresse à laquelle sont stockées ces données.

La première approche est coûteuse en emplacements mémoire et dangereuse : on n'est jamais sûr que l'on a prévu suffisamment de place.

La deuxième technique est une approche peu digne de la gestion dynamique de l'espace mémoire (je n'aime pas beaucoup les instructions modifiées).

La troisième méthode peut vous apporter richesse et célébrité si vous commercialisez votre programme (attention ! je prends 10%) et c'est très exactement ce qui se passe dans les autres langages, moins handicapés moteurs.

MALLOC ET CALLOC SONT DANS UN BATEAU

Pascal et **C** ont quelques points communs en ce qui concerne la gestion de la mémoire. L'un et l'autre disposent d'un espace mémoire disponible pour l'allocation dynamique. Il s'agit de la zone appelée le **heap** (en français le tas).
Faites un petit effort de mémoire : *(suite page 20).*

quand je vous avais parlé des pointeurs, je vous avais mis en garde contre les créations de pointeurs sauvages, vous disant que la déclaration suivante : **char *bidule;** créait un pointeur vers un objet nommé bidule, que ledit objet devait être de type caractère, mais que pour l'instant il n'existait pas et n'avait pas de place réservée en mémoire.

Sournoisement, je ne m'étais pas tellement étendu sur ce qu'il fallait faire pour réserver de la place pour l'objet, et je suppose que certains **scanf("%s",bidule)** ont dû faire des dégâts.

C propose deux instructions essentielles pour utiliser le **heap**. Il s'agit de **malloc** (memory allocate) et de **calloc** (contiguous allocate).

Malloc sert à réserver de la place pour un seul objet et **calloc** pour plusieurs objets de même type.

Malloc admet, comme argument, la taille (**unsigned**) de l'objet pour lequel on désire réserver de la place et retourne un pointeur vers la mémoire réservée, en considérant par convention que ledit objet est de type **CHAR**.

L'utilisation de **malloc** passera donc par le calcul de la taille de l'objet à mémoriser et par un changement de type (cast) si l'objet en question n'est pas de type **char**.

En pratique, le calcul de la taille est effectué par l'opérateur "**sizeof**" et le changement de type s'effectue par :

nouveau_type = (nom_de_type) ancien_type.

Si, par exemple, vous voulez réserver de la place pour un objet de type **float**, la procédure normale est :

```
float *reserve;  /* on réserve de la place pour le
pointeur, mais pas pour l'entier */
reserve = (float *) malloc (sizeof(float));  /* Cette
fois de la place pour l'entier */
```

Calloc fonctionne de la même façon, mais un paramètre "**n**" permet de réserver de la place pour **n** objets de même type.

reserve = (float *) calloc(100,sizeof(float))
réservera la place nécessaire au stockage de 100 objets de type **float**.

Je vois des Pascaliens froncer méchamment les sourcils. Je sais que les changements de type font dresser tous les poils des adeptes de Saint NIKLAUS... mais, que voulez-vous, l'efficacité est à ce prix, et puis vous pouvez toujours écrire :

```
£define MALLOC(x)  ((x *) malloc (sizeof(x)))
£define CALLOC(n,x)  ((x *) calloc(n,sizeof(x)))

reserve = CALLOC(100,float); /*pratique, non ? */
```

Pour une raison qui m'est inconnue, **calloc** et **malloc** font toujours peur aux débutants en **C** . Il faut absolument se débarrasser de cette phobie ridicule : moi qui les fréquente régulièrement, je peux vous l'affirmer **calloc** et **malloc** sont tout à fait adorables.

LA LIBÉRATION DES ESPACES RÉSERVÉS

Bien que gérée dynamiquement, la zone de **heap** n'en est pas pour autant infinie. Il faut donc pouvoir libérer un espace que l'on n'utilise plus et c'est le rôle de l'instruction **free(bloc)** où bloc est un pointeur vers une zone réservée par **malloc** ou **calloc**.

Dans l'exemple précédent **free(reserve)** restitue au **heap** l'espace de stockage des 100 objets de type **float**.

UN EXEMPLE SIMPLE DE GESTION DYNAMIQUE : LA LISTE CHAÎNÉE

Avant de partir explorer le monde infini des allocations dynamiques, je vous propose un petit exemple très simple : la liste chaînée.

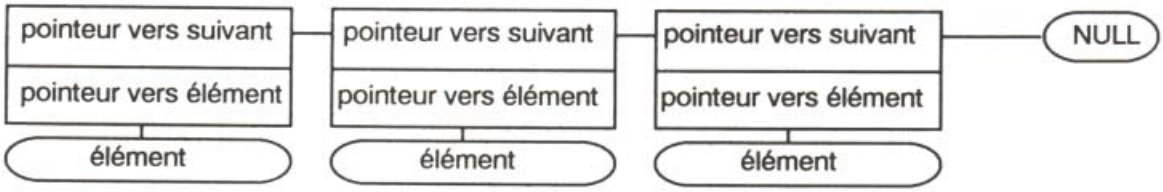
Vous pourrez, partant de cet exemple, construire des structures de plus en plus complexes, et gagner 500 000$ en faisant **le** traitement de texte ou **le** tableur qui manque le plus au GS.

Une liste chaînée est une suite ordonnée d'objets appelés **nœuds**. Un **nœud** est une structure comportant un objet (ou un pointeur vers un objet) et un pointeur vers le **nœud** suivant.

En **C**, une telle structure stockant des entiers peut être fournie par :

```
struct noeud
                {
                int objet;
                struct noeud *suivant;
                }
```

La représentation graphique d'une telle structure est donnée ci-dessous



Deux nœuds de la liste sont un peu particuliers : le nœud terminal, facile à identifier puisqu'il ne pointe vers aucun autre, et le nœud de départ.

Le plus souvent, on localise le nœud de départ à l'aide d'une structure particulière, la tête de liste, qui peut, en plus du pointeur vers le premier élément, comporter des informations sur la liste considérée dans sa globalité : nombre d'éléments, pointeur vers le dernier élément.

Le petit programme que je vous propose, vous offre les outils de création de liste chaînée. A vous de l'améliorer, de créer par exemple un algorithme de suppression d'éléments, de recherche, d'insertion, de modifier la liste en en créant une double (pointeur vers l'élément suivant et précédent), de faire une liste des listes, etc.

Vous en avez bien pour deux mois de travail... bon courage !

N.D.L.R. : Ce programme a été écrit en TURBO C. A nos lecteurs de l'adapter sur leur GS !

```
/*-------------------------------------------------------------------------------

////////////// L'INFERNAL DOCTEUR MALLOC ///////////////
Compilateur TURBO-C et IBM AT2 à adapter pour GS
Claude Aubry 1988

-------------------------------------------------------------------------------*/

#include <stdio.h>

#define MALLOC(x) ((x *) malloc(sizeof (x)))
#define CALLOC(n,x) ((x *)calloc(n,sizeof(x)))
#define VRAI 1
#define FAUX 0
#define BOOLEAN int

/*
** DEFINITION DES STRUCTURES
*/

/*-------------------------------------------------------------------------------
un noeud de la liste comporte
                un pointeur vers une chaine de caractères
                un pointeur vers le noeud suivant
-------------------------------------------------------------------------------*/
```

```
struct noeud
 {
 char        *pointeur_element;
 struct noeud  *suivant;
 };
typedef struct noeud type_noeud;

/*-------------------------------------------------------------------------
 La tete de liste comporte:

                    un entier correspondant au nombre d'items dans la liste
                    un pointeur vers le premier élément
                    un pointeur vers le dernier
-------------------------------------------------------------------------*/
struct tete
 {
 int nb_items;
 type_noeud *premier,*dernier;
 };
 typedef struct tete type_tete;

 /*
 ** DEFINITION DES FONCTIONS DE MANIPULATION DE LISTES
 */

 /*-------------------------------------------------------------------------*/
 /*
 ** FONCTION initialise_tete()
 **
 ** Réserve place mémoire pour structure tête de liste
 **    Initialise nb_items à 0
 **       fait pointer premier et dernier vers NULL
 **
 ** renvoie un pointeur vers la tête de liste
 */
 /*-------------------------------------------------------------------------*/

 type_tete *initialise_tete()

 {
 type_tete *nouveau_noeud;
 if ((nouveau_noeud=MALLOC(type_tete))!=0)
  {
  nouveau_noeud->nb_items=0;
  nouveau_noeud->premier=nouveau_noeud->dernier=NULL;
  }
 return(nouveau_noeud);
 }

 /*-------------------------------------------------------------------------*/
 /*
 ** FONCTION fabrique_noeud(val,ptr)
 ** Reserve espace pour un nouveau noeud (seulement pointeurs)
 ** l'allocation mémoire pour l'élément à insérer doit être effectuée
 ** val est un pointeur vers une variable de type char
```

```
** ptr est de type pointeur vers structure noeud
** Retourne un pointeur vers structure
*/
/*-------------------------------------------------------------------------
*/

type_noeud *fabrique_noeud(pointeur_donnees,pointeur_suivant)

char   *pointeur_donnees;
type_noeud   *pointeur_suivant;
{
type_noeud *nouveau_noeud;
if ((nouveau_noeud=MALLOC(type_noeud))!=0)  /* si mémoire suffisante */
   {
   nouveau_noeud->pointeur_element=pointeur_donnees;
   nouveau_noeud->suivant=pointeur_suivant;
   }
return(nouveau_noeud);
}

/*-------------------------------------------------------------------------*/
/*
** FONCTION ajoute_noeud(donnees,liste);
**
** ajoute_noeud un noeud en fin de liste
*/
/*-------------------------------------------------------------------------*/

BOOLEAN ajoute_noeud(pointeur_donnees,liste)

char   *pointeur_donnees;
type_tete    *liste;

{
 type_noeud *nouveau_noeud;
   /* si allocation possible pour le nouveau noeud */
 if((nouveau_noeud=fabrique_noeud(pointeur_donnees,NULL))!=0)
      {

       /* s'il y a déjà des enregistrements on chaine */

      if(liste->nb_items) liste->dernier->suivant=nouveau_noeud;
            else          liste->premier=nouveau_noeud;       /* sinon pointeur sur le premier */

      liste->dernier=nouveau_noeud;
      liste->nb_items++;
      return(VRAI);
      }
  else
  return(FAUX);
}
```

(suite du programme au verso).

Depuis le numéro 14 de *Tremplin Micro,* Claude Aubry accomplit de louables efforts pour initier nos lecteurs au langage C.

Notre courrier nous prouve chaque jour davantage qu'il est sur la bonne voie.

```
/*
//////////////          PROGRAMME PRINCIPAL          ///////////////
*/


type_tete  *liste_items;
type_noeud *parcours_liste;

main()
{
char *x,*buf;
int i,random;

/* reserve place pour buffer de données */
buf=CALLOC(80,char);
liste_items=initialise_tete();
/* On mémorise un nombre aléatoire de données */
random=rand();
for (i=0;i<=random;i++)

            {
            /* On génère un nouvel élément (un nombre aléatoire en base 2) */
            itoa(rand(),buf,2);
            /* On alloue de la place mémoire au nouvel élément */
            if((x=CALLOC(strlen(buf)+1,char))!=0)
                        {
 /* si la place peut être réservée, on ajoute le pointeur vers l'élément à la liste */

                        strcpy(x,buf);
                        ajoute_noeud(x,liste_items);
                        }
            else printf(«\nPlus de mémoire !!!»);
            }

/* On affiche le contenu de la liste */

parcours_liste=liste_items->premier;

i=0;

do
   {
    printf(«\nITEM numéro %3d : %8s»,i++,parcours_liste->pointeur_element);
    parcours_liste=parcours_liste->suivant;
   }
while(parcours_liste);

}
```


Excusez moi, on m'appelle au téléphone...
Allo, oui, Claude Aubry, qui ça ?...
MULTINATIONALE PARADIS PERE ET FILS...surtout ne quittez pas...
Amis lecteurs à dans deux mois...travaillez bien...

STÉRÉOPHONIE

sur votre Apple IIGS

Les capacités sonores de l'Apple *IIGS* sont exceptionnelles, mais le haut-parleur interne étant incapable de prouesses acoustiques, elles restent coincées dans la boîte ; quant à la prise de sortie disponible à l'arrière de la machine, elle est monophonique et particulièrement bruyante lors des accès disque.

Apple a prévu un petit connecteur qui permet d'accéder directement au synthétiseur Ensoniq et, pour quelques composants de plus, nous allons voir qu'il est possible de disposer d'une sortie stéréo permettant de raccorder notre machine favorite sur une chaîne de qualité Hi-Fi.

L'effet est saisissant et l'utilisation d'un logiciel comme *Musique Studio* devient très attrayante. Certains jeux sont aussi particulièrement réussis au niveau sonore.

Examinons le connecteur de l'Ensoniq que l'on trouve en bas de la carte mère, à gauche du slot d'extension mémoire.

Nous trouvons : (du bas vers le haut)

broche n°1 : entrée de l'Ensoniq (permettant la digitalisation des sons).
broche n°2 : masse du circuit sonore.
broche n°3 : sortie du synthétiseur.
broche n°4 : bit Ø d'adresse de voie sonore (AØ)
broche n°5 : bit 1 d'adresse de voie sonore (A1).
broche n°6 : Strobe d'adressage (STR).
broche n°7 : bit 2 d'adresse de voie sonore (A2).

Comment pouvons-nous disposer de deux signaux sonores indépendants sur l'unique sortie du synthétiseur ?

En fait, le signal analogique présent sur la broche n°3 est découpé en tranches représentant chacune une fraction du signal sonore de la voie de droite ou de gauche : on dit que le signal est multiplexé. Il suffit alors d'aiguiller chaque tranche de signal respectivement à droite ou à gauche ; un interrupteur électronique synchronisé sur la fréquence de multiplexage remplira ce rôle. Un filtre passe-bas éliminant toute fréquence supérieure à 15 kHz restitue ensuite au signal sonore sa forme originale.

La figure n°1 illustre ces explications.

La réalité est un peu plus complexe car le signal est en fait découpé en 16 tranches correspondant à chacune des 16 voies disponibles ; en utilisant le bit Ø de sélection, qui change d'état à chaque incrémentation du numéro de voie, nous aiguillons toutes les voies paires à droite et les impaires à gauche, ce qui nous donne bien deux signaux distincts.

La figure n°2 nous donne le schéma de principe détaillé de la carte stéréo :

- L'ampli IC1 conditionne le signal de sortie de l'Ensoniq. Celui-ci est ensuite dirigé sur les entrées XØ et Y1 de IC2, qui est un démultiplexeur 2 fois 4 voies vers 2 (les 6 entrées inutilisées sont reliées à la masse), lequel aiguille le signal sonore vers ses sorties SX ou SY en fonction de l'état de la broche B (bit d'adresse de voies AØ).
- Le circuit IC3 ne sert qu'à transformer la logique 5 volts en logique 12 volts, le reste des circuits étant alimenté en + et — 12 V pour une meilleure dynamique du signal sonore.
- Nous trouvons ensuite le filtre passe-bas 24 dB par octave construit autour du double ampli IC4 (IC6 pour la voie de gauche), sa fréquence de coupure se situant aux environs de 15 kHz.
- L'ampli IC5 adapte le niveau de sortie à une valeur suffisante pour attaquer n'importe quelle entrée 'Auxiliaire' d'une chaîne Hi-Fi. Le gain peut être adapté aux besoins, en modifiant la valeur de R15 (et R16) suivant la formule G = R15/R9.
- Un petit interrupteur (facultatif) permet de relier les deux sorties en cas de son monophonique pour qu'il soit tout de même présent sur les enceintes droite et gauche.
- La carte prélève son alimentation + et — 12 V de l'un des slots... *(suite page 26).*

- (le slot 4 par exemple car la souris étant un organe indispensable dans tous les logiciels du GS, il y a peu de chance qu'une autre carte vienne s'y loger), deux résistances et deux condensateurs filtrent les bruits présents sur les tensions d'alimentation du GS; les condensateurs de 100 nF placés près des circuits intégrés ont aussi pour but de supprimer tout parasite indésirable.
- Le + 5 V de IC4 est fabriqué à l'aide d'une diode Zéner; l'utilisation du 5 V de l'ordinateur aurait conduit à la fabrication d'un circuit imprimé double-face plus difficile à réaliser par un amateur.

La figure n°3 représente le circuit imprimé à l'échelle 1; il est assez simple pour être réalisé à la main directement sur la plaque cuivrée. L'auteur peut fournir un circuit gravé et percé à ceux que ce travail rebuterait.

Pour le montage des divers composants, se reporter au schéma n°4 qui est plus explicite qu'un long discours (ne pas oublier les ponts S1 à S4); quatre fils suffisent pour relier la carte au connecteur de l'Ensoniq; bien respecter le sens de montage représenté.

La carte ne nécessite aucun réglage et doit fonctionner du premier coup (ne pas oublier de couper l'alimentation de l'ordinateur pour toute manipulation de mise en place, un incident est trop vite arrivé !).

Les composants sont classiques et disponibles dans toutes les boutiques d'éléctronique.

Nomenclature des composants :

R1	=	10 K Ohms
R2 à R14	=	5,6 K Ohms
R15, R16	=	22 K Ohms
R17, R18	=	100 Ohms
R19 à R21	=	220 Ohms 1 Watt
C1	=	68 pF mylar ou céramique
C2, C3	=	2,7 nF mylar ou céramique
C4, C5	=	1,5 nF "
C6, C7	=	2,7 nF "
C8, C9	=	1,5 nF "
C10, C11	=	120 pF "
C12	=	4,7 micro F Tantale 10 V
C13, C14	=	47 micro F Chimiques 16 V verticaux
C15 à C22	=	100 nF mylar ou Tantale 16 V (respecter alors le sens de montage)
IC1	=	LM 318 ou LM 741
IC2	=	MC 14052
IC3	=	7407
IC4 à IC6	=	TL 072 ou TL 082

1 prise de sortie Jack stéréo 3,5 mm
1 connecteur Molex 7 trous
1 mini-interrupteur.

Nous disposons d'un lot limité de circuits imprimés (y compris les plaquettes annexes pour ceux qui retiendront la formule "prise et interrupteur sur façade arrière"). Vous pourrez obtenir ce support en utilisant le bulletin de commande, à la fin de la revue (il ne s'agit que d'une participation aux frais de port et d'emballage). Attention ! notre stock est limité et ne sera pas renouvelé !
Vous trouverez les composants chez les revendeurs spécialisés. Leur coût total ne devrait pas dépasser 100 F.
Le prototype de la carte de Jacques REY a été testé par Emile SCHWARZ et Yvan KŒNIG.

Note d'Yvan KŒNIG

Les schémas électroniques ont été réalisés à l'aide de VS.DRAW qui est, à mes yeux, le meilleur logiciel GS de VersionSOFT. Je profite de l'occasion pour donner quelques détails sur l'utilisation de ce programme lors des impressions.

Comme toujours, il faut s'assurer de la présence dans le 'directory' FONT de la (ou des) police(s) de caractères employée(s) dans le document. La figure n°2 utilise la police STUTTGART et, en son absence, les caractères sont imprimés en SHASTON ce qui provoque des 'chevauchements'.

Pour obtenir une impression correcte il faut sélectionner :

- *mode réduction 50% (il n'y a pas de carte Orange)*
- *impression couleur*
- *option couleur*

Le non respect de ces dernières indications conduit à l'absence sur papier d'un des brins du câble de liaison au connecteur J25.

Figure n°1

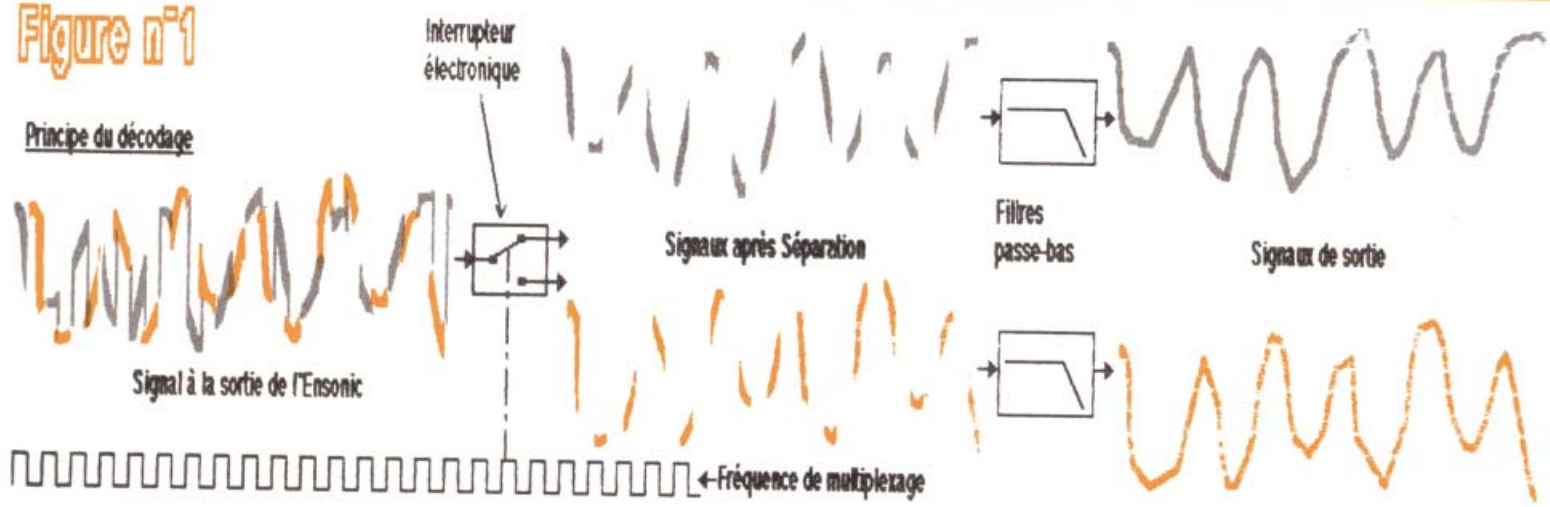


Figure n°1

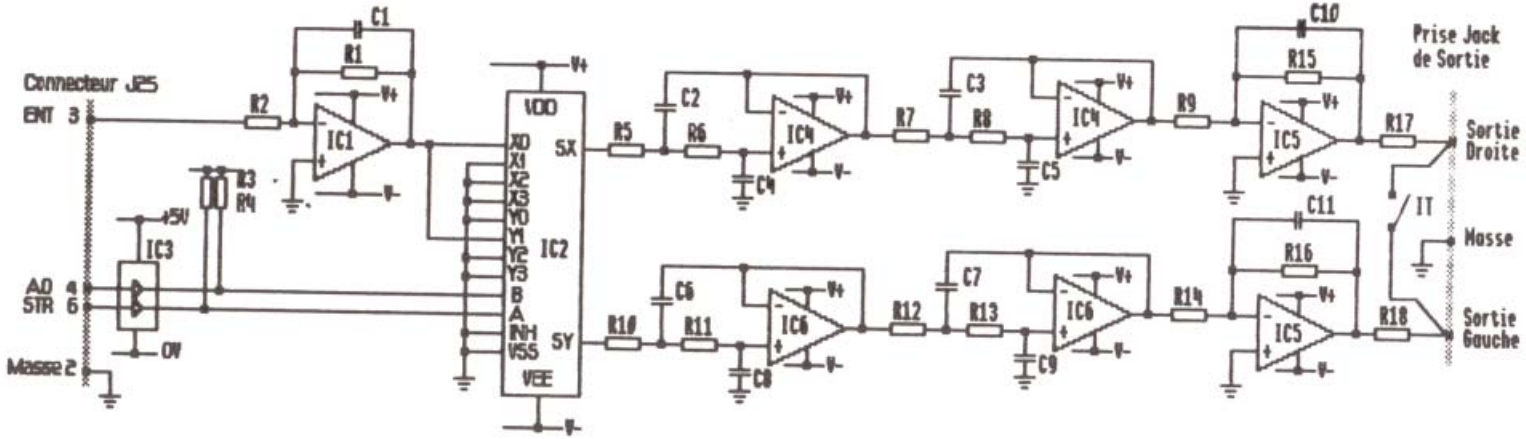


Figure n° 2

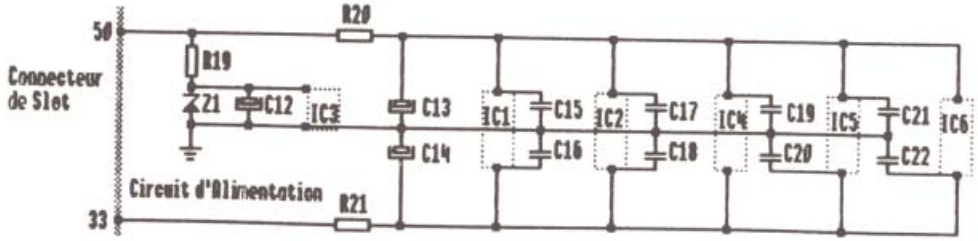
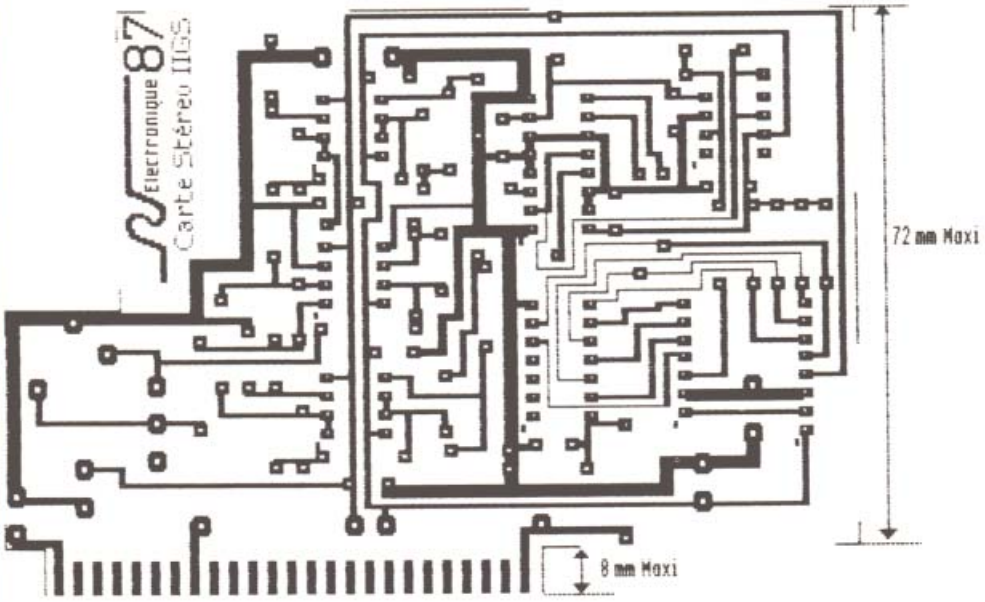


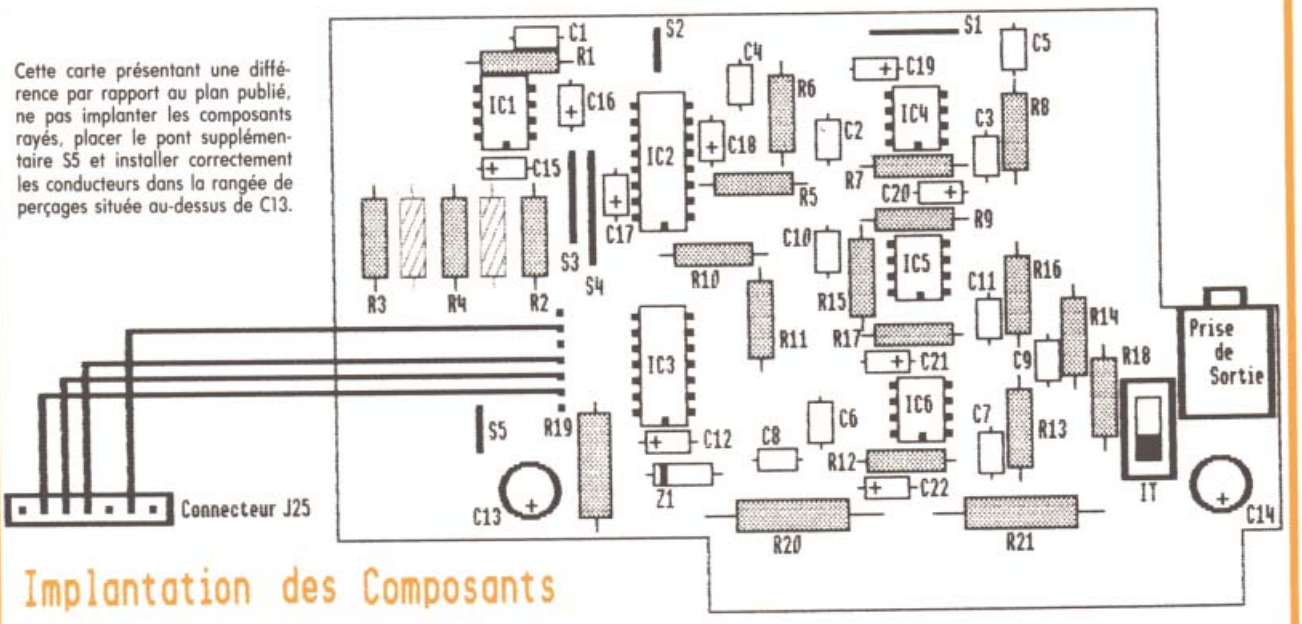
Figure n° 2



Vue côté cuivre

Découper la plaque en tenant compte des côtes maximum indiquées.

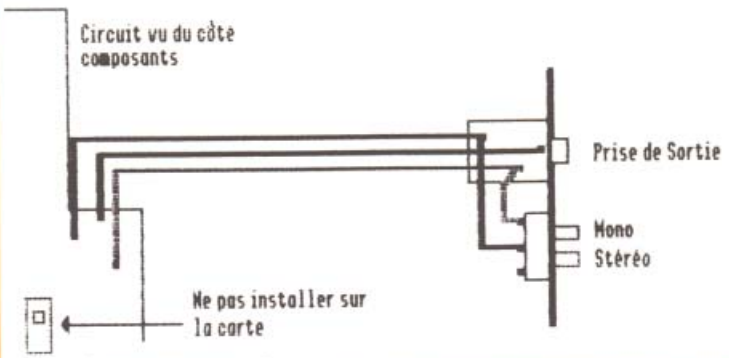
Cette carte présentant une différence par rapport au plan publié, ne pas implanter les composants rayés, placer le pont supplémentaire S5 et installer correctement les conducteurs dans la rangée de perçages située au-dessus de C13.

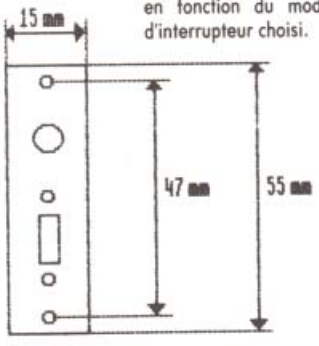


Implantation des Composants

Variante de montage de la prise de Sortie

La plaquette support sera fixée à l'aide de deux vis de 3 mm dans l'une des ouvertures situées à l'arrière du IIGS.
Les perçages non cotés seront réalisés en fonction du modèle de prise et d'interrupteur choisi.





QUATTRO (Borland)

Bien sûr que ça fonctionne sur l'Apple IIGS, équipé d'un PC Transporter !

Vous savez, depuis le dernier numéro de Tremplin Micro, qu'il est possible d'utiliser les logiciels PC sur l'Apple IIGS, grâce à une petite merveille d'Applied Engineering (diffusée par BRÉJOUX) : ***le PC Transporter.*** *Comme cela, par curiosité et pour le plaisir, mais grâce à la bienveillance de Borland International, j'ai donc testé le fameux QUATTRO sur le GS de* ***Tremplin Micro.***

Première (agréable) surprise

Qu'est-ce que **Quattro** ? un tableur. Borland prétend que c'est le meilleur choix et c'est vrai que l'on est d'emblée étonné par sa facilité de prise en main et par ses possibilités.

Personnellement, l'ayant déjà longuement essayé (en version 5'1/4) sur compatible monochrome, je désirais surtout voir comment allait réagir son graphisme sur mon GS couleur.

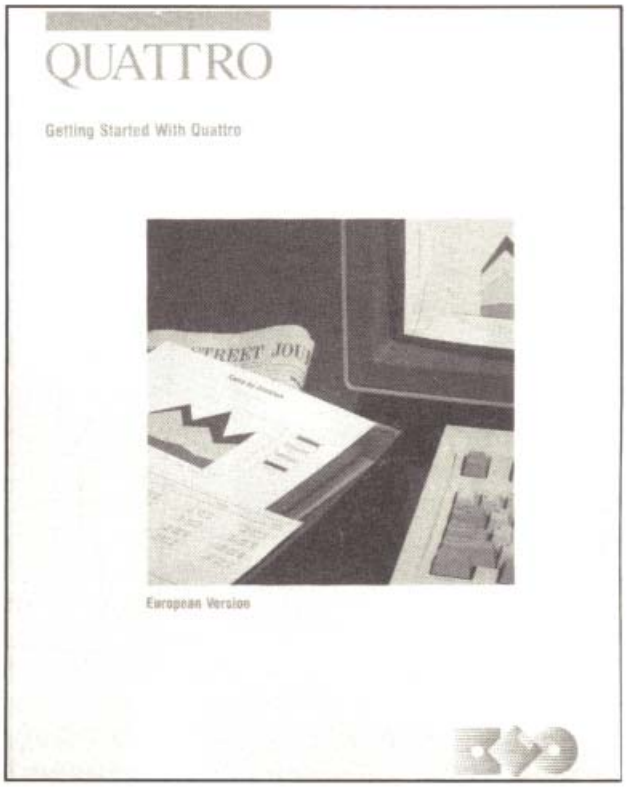
Pas de déception ! Quattro dispose en standard de onze types de graphiques et je vous assure que c'est joli tout plein !

Mieux : ça sort sur imprimante IMAGE WRITER II et le résultat vaut celui que j'avais obtenu sur l'EPSON LQ2500 reliée à mon PC. Vous en doutez ? essayez...

Sachez que ces graphiques peuvent aussi sortir via un traceur ou une photocomposeuse (interface Postcript supportée)... mais je n'ai pas tenté ces deux expériences.

Calcul rapide

Quattro est un tableur rapide, voire très rapide. On dit qu'il doit cette célérité à la manière intelligente dont il traite la feuille de calcul. En fait, le gain de temps provient du fait qu'il calcule en traitant spécifiquement les seules cellules dont on a modifié les paramètres.



Les macro-commandes

Je ne suis pas rompu au maniement des macro-commandes, mais il m'a semblé que se familiariser avec celles de ce tableau ne relève pas du tour de force. Le déboguage facilite d'ailleurs l'identification et la correction des erreurs.

Des trucs sympas

Quattro identifie automatiquement le type de la machine et de l'écran utilisés. Evidemment, sur Apple, il a été abusé — comme il convient — par le PC Transporter, et son identification s'est révélée exacte puisqu'il n'y a pas eu de plantage !

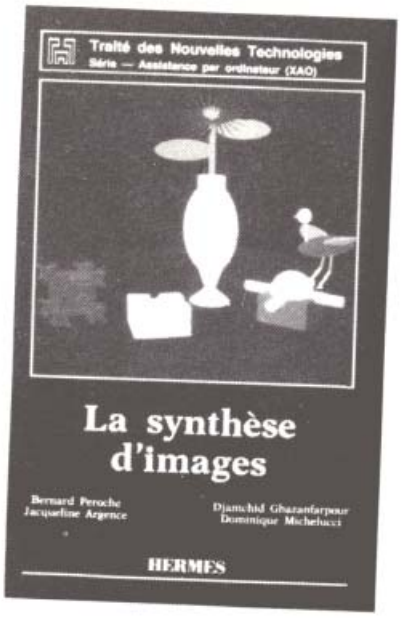
Le logiciel de Borland lit (et écrit) directement des fichiers utilisables par **Lotus 1.2.3, Symphony, dBase, Paradox,** ou **ASCII.**

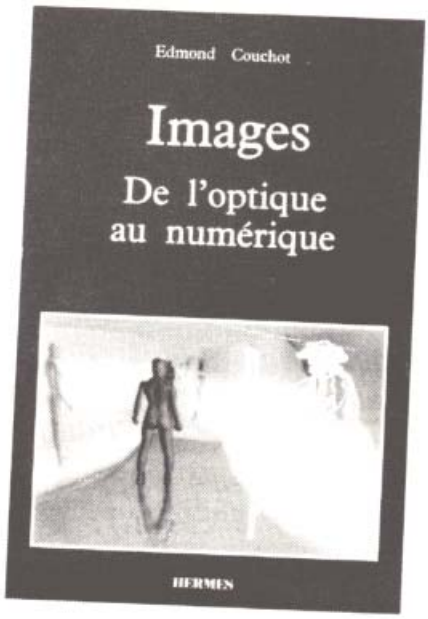
Notez au passage que la sauvegarde automatique des données (Franscript) est fort utile.

Bref, un tableur de qualité professionnelle, facile d'emploi, et aux performances remarquables.

Marc FREFOYE

Borland International *— 65, rue de la Garenne 92318 Sèvres Cedex.*





• LA SYNTHÈSE D'IMAGES
par Bernard PEROCHE (collectif)

La synthèse d'images est probablement née en 1963 avec le premier logiciel graphique digne de ce nom : Sketchpad, développé par I. Sutherland.

Depuis, les progrès n'ont jamais cessé et cet essor remarquable a permis à la synthèse d'images de conquérir de nombreux marchés : ceux de la CAO, du graphique d'affaires, de la simulation ou de l'audiovisuel.

Le but de cet ouvrage collectif est de faire le point sur les techniques, les méthodes et les algorithmes qui ont été développés au cours des vingt-cinq dernières années pour produire les images de synthèse d'un grand réalisme.

Il présente les méthodes et les algorithmes utilisés pour la synthèse d'images réalistes tridimensionnelles. Il décrit les divers modèles permettant de représenter les scènes 3D en machine et expose les algorithmes permettant d'obtenir des images d'un grand réalisme (élimination des parties cachées, lissage, ombres portées, anti-lissage...)

295 pages — relié — 320 F TTC.

***Editions HERMES**, 51 rue Rennequin ,75017 PARIS.*
Tél. : (16-1) 43.80.95.71.

• NOUVEAU MEMENTO TURBO PASCAL
par Dominique SCHMIT

Ce livre s'adresse au programmeur en Turbo Pascal. Il traite de Turbo Pascal version 3 sous MS-DOS/PC-DOS, et Turbo Pascal versions 2.00 et suivantes sur compatible PC.

C'est un excellent memento, présenté avec sobriété, sous une reliure très résistante. On le compulsera avec plaisir. Chaque chapitre couvre un thème spécifique. Le livre se divise comme suit :

1. Généralités sur le langage.
2. Divers types de données.
3. Diverses procédures et fonctions classées par utilisation.
4. Recouvrement et chaînage du programme.
5. Structure interne d'un programme avec les appels système et les sous-programmes externes.
6. Utilisation du système Turbo Pascal.
7. Liste des codes d'erreurs renvoyés par le programme.
8. Liste des codes renvoyés par les touches du clavier.

Relié — 304 pages — 100 F TTC.

***SYBEX**, 6-8 Impasse du Curé, 75018 PARIS*
Tél : (16-1) 42.03.95.95.

• IMAGES DE L'OPTIQUE AU NUMÉRIQUE
par Edmond COUCHOT

Avec l'ordinateur sont apparus de nouveaux processus automatiques de génération et socialisation de l'image. L'image numérique, en effet, n'est plus l'enregistrement d'un objet qui laisse sa trace sur un support chimique ou magnétique, mais une synthèse : le résultat d'un calcul où le nombre et le langage de la programmation se sont substitués à la lumière.

L'auteur analyse dans cette étude le passage de l'optique au numérique et les relations que les techniques de figuration entretiennent avec les arts visuels et la culture. Alors que les premières nous poussaient à représenter le réel, les dernières nous invitent à le simuler.

L'image numérique s'ouvre sur un univers virtuel, en puissance, à vivre et à revivre indéfiniment, jamais totalement actualisable. Elle bouleverse radicalement l'économie symbolique de notre système de figuration du monde. De la capacité à comprendre cette nouvelle image, à l'expérimenter et à la rêver, dépend désormais notre culture.

248 pages — 8 pages couleur
Broché — 240 F TTC.

***Editions HERMES**, 51 rue Rennequin, 75017 PARIS.*

CHARGEMENT

de caractères personnalisés pour Image Writer II et GSBasic

Voici un programme qui vous permettra de télécharger des fontes sur votre Image Writer II. Sur la disquette d'accompagnement, vous trouverez quatre polices de caractères. Deux difficultés majeures pour élaborer ce programme :

1. _SetOutGlobals, utilisé avec les masques appropriés, nous permet d'envoyer les données sans forcer le bit 8 à 1.
2. Inhiber la reconnaissance de la commande Control-I — CHR$ (9) — par le port série.

Depuis le n°19 de *Tremplin Micro*, j'ai utilisé _SetOutGlobals, mais le CHR$ (9) me posait encore problème. J'ai donc été amené à utiliser PEEK et POKE, tout en sachant que ce n'est pas recommandé.

Procédures CtrlIZon et CtrlIZoff :

On commence par mettre en place le caractère de contrôle port série imprimante, soit CHR$ (9), par un POKE 1273,9. Ensuite, PEEK(1529) va lire le contenu de cet emplacement et, suivant que la valeur est supérieure à 127 ou inférieure à 128, nous en déduirons que le port série va filtrer le caractère de commande contenu en 1273 ou non. Enfin, si cette valeur est inférieure à 128, on envoie la commande de non reconnaissance du Control-I.

Avant de quitter le programme, il faut remettre en place la reconnaissance du Control-I ; pour ce faire, on change le masque contenu en 1529 en effectuant la soustraction à la ligne 62020.

Le programme est commenté et n'appelle pas d'explications supplémentaires.

CARAC.PERSO.GSB

```
  5    REM  CARAC.PERSO.GSB  Charge une fonte et l'envoie à l'imprimante
 10    E$=CHR$(27):a=19:P$="TM20/FONTES/FAPL/"
 20    DIM Fonte!(255):REM                          Faux tableau...
 30    PROC tdf:REM                 Vérifie la présence du fichier texttool.tdf
 35    PROC CtrlIZOff:REM                           Ctrl-I Z plus reconnu
 40    PROC presente:REM                            Trace un cadre
 50    LOCATE 3,10:INVERSE
 60    PRINT "Utilitaire de chargement de fontes de caractères personalisés";
 70    NORMAL:PRINT
 80    LOCATE 5,a:PRINT "Fontes présentes sur la disquette /TM20/ :"
 90    LOCATE 7,a:PRINT P$"COUNT":LOCATE 9,a:PRINT P$"EPSDROITE"
100    LOCATE 11,a:PRINT P$"FONETIC":LOCATE 13,a:PRINT P$"ITALEPS"
110    LOCATE 17,a:PRINT "(c) Avril 1988 Tremplin Micro & Emile Schwarz"
115    REM
120    ON ERR  GOTO EPRODOS:REM                     Si erreur ProDOS...
130    LOCATE 20:INPUT "Nom ProDOS COMPLET de la fonte à charger: ";Fonte$
140    x%=FILE(Fonte$):IF NOT x% THEN erreur:REM Si x% = 0: fichier inconnu
150    OPEN Fonte$, FILTYP=6 FOR INPUT AS£2:REM Nous ouvrons le fichier
160    y%=EOFMARK(2):IF y%>4000 THEN fin:REM    Fichier trop grand --> fin
170    ERASE Fonte!():DIM Fonte!(y%):REM        Dimensionne le tableau
180    GET£2,y%,0;Fonte!(0):REM                 Charge le fichier
190    CLOSE£2:REM                              Ferme le fichier
```

```
195     REM
200     IF Fonte!(0)<>27 AND Fonte!(1)<>43 OR Fonte!(1)<>45 AND Fonte!(2)<>27
         AND Fonte!(3)<>73 THEN fin
210     ON ERR  GOTO EIWII:REM                          A tout hasard
215     REM
220     OPEN ".PRINTER", FOR OUTPUT AS£1:REM            Ouvre l'imprimante
230     _setoutglobals(255,0):REM                       Fixe la nouvelle valeur 'OR'
240     PRINT£1;E$"Z "CHR$(0);:REM                      Pas de LF si ligne pleine
250     PRINT£1;E$"11";:REM                             Pas de CR avant LF et FF
260     PRINT£1;E$"Z"CHR$(128)CHR$(0);:REM              Pas de LF après CR
270     PRINT£1;E$"Z"CHR$(0)" ";:REM                    8° bit reconnu
280     FOR i%=0 TO y%-1:REM                            Envoie la fonte
290         PRINT£1;CHR$(Fonte!(i%));
300     NEXT
310     PRINT£1;E$"D "CHR$(0);:REM                      LF si ligne complète
320     PRINT£1;E$"10";:REM                             Ajoute CR avant LF et FF
330     PRINT£1;E$"D"CHR$(0)" ";:REM                    Huitième bit ignoré
340     PRINT£1;"":REM                                  Retour en début de ligne
345     REM
350     REM ------------- Ecrit la fonte personnalisée -------------------
355     REM
360     PROC CtrlIZon:PROC recup:REM Active la reconnaissance du CHR$(9)"Z"
370     PRINT£1;E$"!";:REM                              On passe en caractères gras
380     IF INSTR(Fonte$,"EPS") THEN PRINT£1;E$"q";:REM Compressé si 'EPS'
390     PRINT£1;E$"'";:REM                              Début des caractères perso.
400     PRINT£1;"  ! "CHR$(34)" £ $ % & ' ( ) * + , -";
410     PRINT£1;" . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?"
420     PRINT£1;"à A B C D E F G H I J K L M N O P";
430     PRINT£1;" Q R S T U V W X Y Z ° ç § ^ _ `"
440     PRINT£1;"  a b c d e f g h i j k l m n o";
450     PRINT£1;" p q r s t u v w x y z é ù è ¨"
460     PRINT£1;E$"$";:REM                              Fin des caractères perso.
470     PRINT£1;E$CHR$(34);:REM                         Fin des caractères gras
480     PRINT£1;E$"E";:REM                              Elite
490     PRINT£1;"":REM                                  Retour à la ligne
500     GOTO EIWII:REM                                  Le programme est fini
505     REM
510     REM ------------- Traitement des erreurs et fin ------------------
515     REM
520 ERREUR:OFF ERR:REM Retour au traitement d'erreur standard
530     TEXT:HOME:INVERSE:PRINT "Le fichier "Fonte$" est inconnu!":NORMAL
540     PRINT:PRINT "<1> pour CATALOG ":PRINT:PRINT "<2> pour Continuer"
550     PRINT:PRINT "<3> pour Fin "
560     PRINT:PRINT "Votre Choix: ";:GET$x$:PRINT x$
570     x=VAL(CHR$(ASC(x$)-128))
580     ON x>3 OR x<1 GOTO erreur
590     ON x GOTO 600,40,fin
600     INPUT "Nom du volume: ";Volume$:ON ERR  GOTO EPRODOS
610     PRINT "Appuyez sur la barre d'espace pour arrêter le défilement"
620     PRINT "-----------    Ctrl-C         pour stopper ------------"
630     CATALOG Volume$
640     PRINT "Pressez une touche pour continuer";:GET$x$:PRINT x$
650     TEXT:HOME:GOTO 40
660 EPRODOS:OFF ERR:REM                 Retour au traitement d'erreur standard
670     LOCATE,33:INVERSE:PRINT CHR$(7)"Nom incorrect":NORMAL
```

```
  680     FOR i%=0 TO 1000
  690     NEXT:REM                                Boucle d'attente
  700     GOTO 40
  710 EIWII:OFF ERR:REM                           Retour au traitement d'erreur
  720     _setoutglobals(255,128):REM             et aux valeurs standards
  730 FIN:END:REM                                 Terminus!
  740     REM
49980     REM                          Procédures
60080     REM
60090     REM -- Charge éventuellement 'texttool.tdf' et enlève 'AUTO LF' --
60100 TDF:DEF PROC tdf
60110     LIBFIND "SetOutGlobals",o1%,f2%,x3%
60120     IF o1% THEN 60160:REM Si le fichier a déjà été chargé alors...
60130     INVERSE:PRINT "Un instant SVP.":NORMAL
60140     PRINT "Je charge le fichier /gsb/tdfs/texttool.tdf"
60150     LIBRARY APPEND "/gsb/tdfs/texttool.tdf"
60160     ASSIGN ".PRINTER",-1:ASSIGN ".PRINTER",1:REM Pas d'auto LF
60170     END PROC tdf
60190     REM --      Affiche un rectangle de présentation à l'écran      --
60200 PRESENTE:DEF PROC presente
60210     TEXT:HOME:PRINT REP$("-",80)"!";
60220     FOR I%=0 TO 15
60230         HPOS=80:PRINT "!!";
60240     NEXT
60250     HPOS=80:PRINT "!"REP$("-",80);
60260     END PROC presente
61990     REM -- Permet la reconnaissance du CHR$(9)"Z" par le port série --
62000 IZON:DEF PROC CtrlIZon:REM      Rétablit la reconnaissance du Ctrl-I "Z"
62010     CtrlIZ%=PEEK(1529):REM     par le port série
62020     POKE 1529,CtrlIZ%-128
62030     END PROC CtrlIZon
62090     REM -- Inhibe la reconnaissance du CHR$(9)"Z" par le port série --
62100 IZOFF:DEF PROC CtrlIZoff:REM   Supprime la reconnaissance du Ctrl-I "Z"
62110     POKE 1273,9:REM            par le port série
62120     IF PEEK(1529)>127
62130         THEN OPEN ".PRINTER", AS£1:PRINT£1;CHR$(9)"Z";:CLOSE£1
62140     END PROC CtrlIZoff
62990     REM -- Récupère le nom de la fonte et l'envoie sur l'imprimante --
63000 RECUP:DEF PROC recup
63010     LOCAL a%,b%:REM            Ce sont des variables locales
63020     a%=1:b%=1:REM              qui ne servent qu'ici!
63030     WHILE b%<>0
63040         b%=INSTR(Fonte$,"/",a%+1):REM Recherche la présence de "/"
63050         IF b% THEN SWAP a%,b%
63060     UNTIL
63070     PRINT£1;"Test de la fonte ";MID$(Fonte$,a%+1)
63080     END PROC recup
```

DEMO.CAR.PERSO

```
   5      REM DEMO.CAR.PERSO
  10 DEBUT:ON ERR  GOTO fin:REM              A toutes fins utiles...
  20       HOME:LOCATE 12,15:E$=CHR$(27)
  30       PRINT "Démonstration des caractères personalisés £à°ç§éùè¨"
  40       OPEN ".PRINTER", AS£1:REM         Ouvre le port imprimante
```

```
 50       _setoutglobals(255,0):REM          Nouvelle valeur de 'OR'
 60       PRINT£1;E$"Z "CHR$(0);:REM         Pas de LF si ligne pleine
 70       PRINT£1;E$"l1";:REM                Pas de CR avant LF et FF
 80       PRINT£1;E$"Z"CHR$(128)CHR$(0);:REM Pas de LF après CR
 90       PRINT£1;E$"Z"CHR$(0)" ";:REM       8° bit reconnu
100 BOUCLE:FOR A%=0 TO 94:REM               Boucle de lecture/
110           READ C%:PRINT£1;CHR$(C%);
120       NEXT:REM                          écriture des datas
130       PRINT£1;E$"D "CHR$(0);:REM        LF si ligne pleine
140       PRINT£1;E$"l0";:REM               Ajoute CR avant LF et FF
150       PRINT£1;E$"D"CHR$(0)" ";:REM      8° bit ignoré
160       PRINT£1;"":REM                    Passe à la ligne
170       CLOSE£1:REM                       Ferme provisoirement
180       OPEN ".PRINTER", AS£1:REM         Pour le rouvrir aussitot!
190       PRINT£1;"Démonstration des caractères: £à°ç§éùè¨";
200       PRINT£1;E$"!":REM                 Début des caractères gras
210       PRINT£1;E$"!";"£à°ç§éùè¨";E$"$";
220       PRINT£1;E$CHR$(34);:REM           Fin des caractères gras
230       PRINT£1;"":REM                    Passe à la ligne
240 FIN:  _setoutglobals(255,128):REM       Fixe les valeurs standards
250       CLOSE£1:END:REM                   C'est terminé!
260       REM ---------------------------------------
270       REM
280       REM     PROL pour Prologue,
290       REM     CAR1 à CAR9 pour chaque caractère,
300       REM     EPIL pour Epilogue.
310       REM
320       REM ---------------------------------------
330 PROL: DATA 027,045,027,073
340 CAR1: DATA 035,072,020,127,020,020,127,020,000,000
350 CAR2: DATA 064,072,028,034,065,008,085,000,085,014
360 CAR3: DATA 091,072,000,000,000,127,065,065,000,000
370 CAR4: DATA 092,072,000,001,002,004,008,016,032,064
380 CAR5: DATA 093,072,000,000,000,065,065,127,000,000
390 CAR6: DATA 123,072,000,000,008,054,065,065,000,000
400 CAR7: DATA 124,072,000,000,000,000,127,000,000,000
410 CAR8: DATA 125,072,000,000,065,065,054,008,000,000
420 CAR9: DATA 126,072,002,001,001,002,002,001,000,000
430 EPIL: DATA 004
440       REM (c) Avril 1988 Tremplin Micro & Emile SCHWARZ
49970     REM-----------------------------------------------------------
49980     REM                        Procédures
49990     REM-----------------------------------------------------------
60100 TDF:  DEF PROC tdf
60110       LIBFIND "SetOutGlobals",o1%,f2%,x3%
60120       IF o1% THEN 60160:REM Si le fichier a déjà été chargé alors...
60130       INVERSE:PRINT "Un instant SVP, ";
60140       PRINT "je charge le fichier /gsb/tdfs/texttool.tdf":NORMAL
60150       LIBRARY APPEND "/gsb/tdfs/texttool.tdf"
60160       ASSIGN ".PRINTER",-1:ASSIGN ".PRINTER",1:REM Pas d'auto LF
60170       END PROC tdf
```

QuickDraw Démo Gsb

Voici comment utiliser certains outils de QuickDraw. Pour une plus grande rapidité d'exécution, j'ai défini des procédures d'accès aux outils à la place de sous-programmes "GOSUB / RETURN".

Chaque appel à un outil se présente comme suit :

Pile avant appel

Contenu avant appel
Espace pour résultat — S'il y a lieu
Paramètre 1 — Description du paramètre
Paramètre 2 — Description du paramètre
(éventuellement)

← Pointeur de pile

Pile après appel

Contenu après appel
Résultat — S'il y a lieu

← Pointeur de pile

Chaque paramètre peut être une valeur, l'adresse d'une table de paramètres ou l'adresse d'une routine de traitement.

Dans les programmes **QD.DEMO1** et **QD.DEMO2**, je vous propose deux façons différentes d'utiliser une table de paramètres.

Dans chaque procédure, j'ai incorporé un paramètre **c%** pour numéro de couleur. Vous pouvez constater qu'une procédure peut en appeler une autre... (ou en cacher une autre !).

Pour QuickDraw, une figure simple se définit par les coordonnées des points haut/gauche et bas/droit d'un rectangle fictif. Les noms de variables de définition de ces points sont **V1,H1,V2,H2** ou, si vous préférez, **y,x,v,h.**

- **y,x** définissent le point haut/gauche,
- **v,h** le point bas/droit du rectangle.

J'ai employé plusieurs types de variables dans les boucles, chacun correspondant à une volonté de **lenteur** ou de **rapidité**. En effet, une variable entière indice (suffixe %) de boucle demandera un temps d'exécution plus rapide qu'une variable entière longue (suffixe **&**).

_SetForeColor (sfc%)	Fixe la couleur de l'écriture.
_SetBackColor (sbc%)	Fixe la couleur du fond.
_MoveTo (y%,x%)	Place le crayon à ces coordonnées.
_LineTo (y%,x%)	Dessine une ligne depuis la position actuelle du crayon jusqu'en y%,x%.

Les appels outils suivants ont l'adresse des paramètres en argument :

_DrawString (VARPTR (msg! (0)))	Dessine le texte stocké dans le tableau msg! (0).
_invertrect (VARPTR (rect! (0)))	Inverse les couleurs dans le rectangle.
_FrameRect (VARPTR (rect! (0)))	Dessine le contour d'un rectangle.
_FrameOval (VARPTR (rect! (0)))	Dessine le contour d'un ovale.
_FrameRRect (VARPTR (rect! (0)))	Dessine le contour d'un rectangle arrondi.
_FrameArc (VARPTR (rect! (0)))	Dessine le contour d'une portion de cercle.
_PaintRect (VARPTR (rect! (0)))	Dessine le contenu d'un rectangle.
_PaintOval (VARPTR (rect! (0)))	Dessine le contenu d'un ovale.
_PaintRRect (VARPTR (rect! (0)))	Dessine le contenu d'un rectangle arrondi.
_PaintArc (VARPTR (rect! (0)))	Dessine le contenu d'une portion de cercle.

(suite page 36).

Vous pouvez diminuer la valeur de la ligne 20 si votre GS n'a que 512 K. Dans ce cas, 65536 est une valeur correcte. Les boucles FOR... NEXT sont sur plusieurs lignes pour plus de clarté et pour contourner le Bug de l'indention.

Voir d'autre part les programmes de patchs de PICS en ce qui concerne l'affichage des caractères accentués.

---

1 - QD.DEMO1

```
  5 REM QD.DEMO1
 10 DIM rect!(7):REM              Paramètres d'un rectangle
 15 PROC outil:REM                Charge le fichier quickdraw.tdf
 20 GRAF INIT 320:GRAF ON:REM     320 points/ligne + affichage écran SHGR
 30 y%=10:x%=10:v%=20:h%=30:REM   Ce rectangle contiendra 200 points
 40 SET(rect!(0),2)=y%:REM y%,x% = localisation V1/H1 de l'angle haut/droit
 50 SET(rect!(2),2)=x%:REM y% = position verticale, x% = position horizontale
 60 SET(rect!(4),2)=v%:REM v%,h% = localisation V2/H2 de l'angle bas/droit
 70 SET(rect!(6),2)=h%:REM v% = position verticale, h% = position horizontale
 80 ptrà=VARPTR(rect!(0)):REM     ptrà = adresse de la table rect!(0)
 90 _SetSolidPenPat(15):REM       Le crayon écrira en blanc
100 _PaintRect(ptrà):REM Dessine 1 rectangle en utilisant le tableau rect!(0)
110 GET$A$:GRAF OFF:END:REM       Attente frappe clavier, puis fin
63100 DEF PROC outil
63110 LIBFIND "QDStartUp",o1%,f2%,x3%
63120 IF o1% THEN 63140
63130 LIBRARY APPEND "/gsb/tdfs/quickdraw.tdf"
63140 END PROC outil
```

2 - QD.DEMO2

```
  5 REM QD.DEMO2
 10 DIM msg!(255):REM                    Nombre de caractères maximum
 15 PROC outil:REM                       Charge éventuellement quickdraw.tdf
 20 GRAF INIT 320:GRAF ON:REM            320 points/ligne + affichage écran SHGR
 30 a$="Vive le GS Basic!":REM           Chaine à afficher
 40 SET(msg!(0))=^a$:REM Stocke la chaine a$ à la Pascal: 1° octet = longueur
 45 _MoveTo(10,10):REM                   Début de l'écriture en 10,10
 50 _SetBackColor(15):REM                Couleur du fond: Blanc
 60 _SetForeColor(1):REM                 Couleur de l'écriture: Noir
 70 _DrawString(VARPTR(msg!(0))):REM Ecrit le message
 80 GET$A$:GRAF OFF:REM                  Attente clavier, puis fin
 90 END
63100 DEF PROC outil
63110 LIBFIND "QDStartUp",o1%,f2%,x3%
63120 IF o1% THEN 63140
63130 LIBRARY APPEND "/gsb/tdfs/quickdraw.tdf"
63140 END PROC outil
```

POURQUOI LE GSBASIC ?

Personne ne vous interdit de programmer en APPLESOFT, mais vous n'aurez vraiment accès aux outils du GS qu'en utilisant le GSBasic, l'une des plus récentes moutures du Basic... ▪

3 - QUICKDRAW.GSB

```
5           REM QUICKDRAW.GSB
10 DEBUT:    HOME:ON ERR  GOTO erreur
20           CLEAR 165536
30           b$=" Tremplin Micro et Emile Schwarz "
40           c$="Pressez une touche pour continuer"
50           a$=" IIGS Basic: Quick DrawII Demo ":j%=1
60           DIM msg!(256),rect!(7)
70           LIBFIND "QdStartUp",q1%,q2%,q3%
80           IF q1% THEN saute
90           LIBRARY APPEND "/GSB/TDFS/quickdraw.tdf"
100 SAUTE:   GRAF INIT 320:GRAF ON
110 IMAGE1:  y%=0:x%=0:v%=200:h%=320
120 BOUCLE1: FOR I%=0 TO 15
130              PROC framrect(i%,y%,x%,v%,h%)
140              y%=y%+10:x%=x%+10:v%=v%-10:h%=h%-10
150          NEXT
160          PROC msg(51,193,5,15,a$)
170          PROC msg(46,14,6,15,b$)
180          PROC pintrect(15,91,91,109,229)
190 BOUCLE2: FOR i=15 TO 0 STEP-1
200              PROC msg(51,193,15,i,a$)
210              PROC msg(46,14,15,i,b$)
220              PROC msg(102,103,i,15,"(c)  2 Mars 1988")
230          NEXT
240 BOUCLE3: FOR A&=0 TO 5
250              PROC invrect(0,160,200,320)
260              PROC invrect(0,0,200,160)
270              PROC invrect(100,0,200,320)
280              PROC invrect(0,0,100,320)
290          NEXT
300          PROC invrect(0,0,200,360)
310          PROC msg(38,103,0,15,c$)
320          GET$x$
330 IMAGE2:  GRAF INIT 320:GRAF ON
340          _setsolidpenpat(15):REM Couleur du crayon: Blanc
350          PROC boite(0,0,200,320)
360          PROC pintrect(9,10,10,30,30)
370          PROC msg(5,40,0,15,"paintrect")
380          PROC pintarc(8,10,80,30,240,-90,-90)
390          PROC msg(100,40,1,14,"paintarc")
400          PROC framarc(15,10,80,30,240,0,180)
410          PROC msg(160,40,2,13,"framearc")
420          PROC framrect(3,10,290,30,310)
430          PROC msg(245,40,3,12,"framerect")
440          PROC pintRrect(5,55,10,90,110,15,15)
450          PROC msg(11,100,0,14,"paintRrect")
460          PROC framRrect(10,55,210,90,310,15,15)
470          PROC msg(228,100,0,14,"frameRrect")
480          PROC pintoval(4,120,10,170,110)
490          PROC msg(25,180,5,9,"paintoval")
500          PROC framoval(2,120,210,170,310)
510          PROC msg(228,180,4,11,"frameoval")
520 BOUCLE4: FOR i%=0 TO 13
```

```
  530                PROC pintarc(j%,80,110,150,209,k%,15)
  540                j%=j%+1:k%=k%+30:IF k%>180 THEN k%=-k%
  550             NEXT
  560             PROC msg(125,180,6,7,"paintarc")
  570             PROC msg(38,195,0,15,c$)
  580             PROC invrect(0,0,200,320)
  590             GET$x$:GOTO fin
  600 ERREUR:     OFF ERR:PRINT "?"ERRTXT$(ERR)" error in line "ERRLIN
  610 FIN:        GRAF OFF:END
61997             REM-------------------------------------------------------
61998             REM                          Procédures
61999             REM-------------------------------------------------------
62000 BOITE:      DEF PROC boite(y%,x%,v%,h%)
62001             _moveto(y%,x%)
62002             _lineto(y%+h%-1,x%)
62003             _lineto(y%+h%-1,x%+v%-1)
62004             _lineto(y%,x%+v%-1)
62005             _lineto(y%,x%)
62006             _moveto(y%+1,x%+1)
62007             _lineto(y%+1,x%+v%-2)
62008             _moveto(y%+h%-2,x%+1)
62009             _lineto(y%+h%-2,x%+v%-2)
62010             END PROC boite
62020 MSG:        DEF PROC msg(y%,x%,sfc%,sbc%,a$)
62021             SET(msg!(0))=^a$
62022             _setforecolor(sfc%)
62023             _setbackcolor(sbc%)
62024             _moveto(y%,x%)
62025             _drawstring(VARPTR(msg!(0)))
62026             END PROC msg
62030 INVERT:     DEF PROC invrect(y%,x%,v%,h%)
62031             PROC rect(y%,x%,v%,h%)
62032             _invertrect(ptrà)
62033             END PROC invrect
62040 CORECT:     DEF PROC rect(y%,x%,v%,h%)
62041             SET(rect!(0),2)=y%
62042             SET(rect!(2),2)=x%
62043             SET(rect!(4),2)=v%
62044             SET(rect!(6),2)=h%
62045             ptrà=VARPTR(rect!(0))
62046             END PROC rect
62050 FRAMERECT:DEF PROC framrect(c%,y%,x%,v%,h%)
62051             PROC rect(y%,x%,v%,h%)
62052             _setsolidpenpat(c%)
62053             _framerect(ptrà)
62054             END PROC framrect
62060 PINTOVAL:  DEF PROC pintoval(c%,y%,x%,v%,h%)
62061             PROC rect(y%,x%,v%,h%)
62062             _setsolidpenpat(c%)
62063             _paintoval(ptrà)
62064             END PROC pintoval
62070 FRAMOVAL:  DEF PROC framoval(c%,y%,x%,v%,h%)
62071             PROC rect(y%,x%,v%,h%)
62072             _setsolidpenpat(c%)
```

```
62073            _frameoval(ptrà)
62074            END PROC framoval
62080 PINTARC:   DEF PROC pintarc(c%,y%,x%,v%,h%,ad%,va%)
62081            PROC rect(y%,x%,v%,h%)
62082            _setsolidpenpat(c%)
62083            _paintarc(ptrà,ad%,va%)
62084            END PROC pintarc
62090 PINTRECT:  DEF PROC pintrect(c%,y%,x%,v%,h%)
62091            PROC rect(y%,x%,v%,h%)
62092            _setsolidpenpat(c%)
62093            _paintrect(ptrà)
62094            END PROC pintrect
62100 PINTRRECT:DEF PROC pintrrect(c%,y%,x%,v%,h%,lo%,ho%)
62101            PROC rect(y%,x%,v%,h%)
62102            _setsolidpenpat(c%)
62103            _paintrrect(ptrà,lo%,ho%)
62104            END PROC pintrrect
62110 FRAMRRECT:DEF PROC framrrect(c%,y%,x%,v%,h%,lo%,ho%)
62111            PROC rect(y%,x%,v%,h%)
62112            _setsolidpenpat(c%)
62113            _framerrect(ptrà,lo%,ho%)
62114            END PROC framrrect
62120 FRAMARC:   DEF PROC framarc(c%,y%,x%,v%,h%,ad%,va%)
62121            PROC rect(y%,x%,v%,h%)
62122            _setsolidpenpat(c%)
62123            _framearc(ptrà,ad%,va%)
62124            END PROC framarc
```

GSB.HELLO AMELIORE

```
 10         REM  GSB.HELLO amélioré!
 20 DEBUT:  ASSIGN ".PRINTER",-1:REM Supprime .PRINTER
 30         ASSIGN ".PRINTER",1:REM  Installe .PRINTER sans AUTO-LF
 40         CLEAR 65535:REM          Réserve $FFFF de mémoire
 50         PREFIX 6,"/gsb/tdfs"
 60         PRINT "Un instant SVP, je charge les fichiers .tdf"
 70         LIBRARY "6/locator.tdf"
 80         LIBRARY APPEND "6/memory.tdf"
 90         LIBRARY APPEND "6/misctool.tdf"
100         LIBRARY APPEND "6/quickdraw.tdf"
110         LIBRARY APPEND "6/desk.tdf"
120         LIBRARY APPEND "6/event.tdf"
130         LIBRARY APPEND "6/scheduler.tdf"
140         LIBRARY APPEND "6/sound.tdf"
150         LIBRARY APPEND "6/adb.tdf"
160         LIBRARY APPEND "6/sane.tdf"
170         LIBRARY APPEND "6/intmath.tdf"
180         LIBRARY APPEND "6/texttool.tdf"
190         ON ERR  GOTO erreur:REM Branchement sur ERREUR s'il y a lieu...
200         OPEN ".PRINTER", AS£1:OUTPUT£1
210         _setoutglobals(255,0):REM Fixe les masques AND et OR
220         PRINT CHR$(27)"D"CHR$(0)CHR$(1):REM Zéros barrés.
230         CLOSE£1
240 ERREUR:OFF ERR:REM Rétablit le mode standard de traitement d'erreur
250         _setoutglobals(255,128):REM Valeurs standards des masques.
260 FIN:    NEW
```

Par suite d'une mauvaise manipulation, le programme qui figurait dans *Tremplin Micro n°19* n'était qu'une... ébauche comportant quelques erreurs.

Voici le *Hello* définitif (qui figurait sur la disquette d'accompagnement).

Avec toutes les excuses d'Emile SCHWARZ.

Image Writer II

Corrigez votre manuel de référence

Vous avez sûrement lu l'annonce de la sortie du manuel de référence de l'*Image Writer II* en Français, paru chez *Inter Editions*. Je l'ai acheté.

1. Page 73

Le texte a simplement été traduit, ainsi que l'erreur relevée dans l'édition américaine.

Il n'est pas possible d'envoyer plusieurs 'backspaces' consécutifs sur l'Image Writer II. A l'origine, on devait pouvoir en envoyer deux pour un bon positionnement des accents circonflexes sur le 'i' et sur les autres lettres. Ce n'est pas le cas : cette possibilité a été malencontreusement *oubliée*.

2. Pages 111, 112, 153

Ainsi que sur la carte de référence :

Alors qu'Apple avait été informé de ce problème (par un collaborateur de *Tremplin Micro*) et avait publié une fiche technique IMW 1 sur ce point (sous le nom : Custom Font Selection), aucune mention n'est faite du bug interdisant le retour au banc 1 des caractères personnalisés par **ESC '** lorsque l'on a utilisé le banc 2 par **ESC ***.

Le remède consiste à envoyer les commandes suivantes :

Caractères	Code décimal	Code hexa	Signification
ESC D Ctrl-à Espace	27 68 Ø 32	1B 44 00 20	Bit 8 ingnoré
ESC Z Ctrl-à Espace	27 90 Ø 32	1B 5A 00 20	Bit 8 reconnu

3. Carte de référence

Caractères personnalisés, il faut lire :

ESC I 27 73

et non : 27 43 comme imprimé.

4. Carte de référence

Divers, il faut lire :

ESC D Ctrl-à Espace 27 68 Ø 32 1B 44 00 20

5. Une mise en garde

Il faut lire :

Esc I Ø pour Insertion de CR avant LF et FF

Esc I 1 pour Pas d'insertion de CR avant LF et FF

soit escape petit I zéro = 27 108 48
escape petit I un = 27 108 49

En effet, la police utilisée pour l'impression peut laisser un doute : est-ce ESC 1 Ø ou ESC I Ø ? Si vous lisez les codes en décimal ou hexadécimal, aucun doute n'est permis, mais... errare humanum est !

Les points 1, 2 et 5 sont communs aux deux éditions. Les points 3 et 4 n'existent que dans l'édition française.

Si vous ne possédez pas encore ce manuel, vous pouvez vous le procurer au rayon librairie de votre concessionnaire ou chez :

Inter Editions, 87 Avenue du Maine, 75014 PARIS
Tél : (16-1) 43.27.74.50

Pierre MÉTIVIER

nommé Responsable Communication Technique

Pour assurer une meilleure information sur les problèmes techniques concernant les logiciels BORLAND, Pierre MÉTIVIER a été nommé Responsable Communication Technique, à destination de la presse et des entreprises.

Pierre MÉTIVIER, 30 ans, est entré chez BORLAND INTERNATIONAL en décembre 1986 pour y créer le Support Technique. C'est également lui qui a mis en place le service télématique "Forum des langages" accessible par le 36.14.

Ingénieur Electronicien diplômé de l'ESME, Pierre MÉTIVIER est titulaire d'un DESS de gestion de l'IAE et d'un Master of Computer Science de BGSU (Ohio). Ingénieur informaticien chez Micro Application de 1982 à 1985, il a ensuite été Responsable du Support Technique chez Apricot avant de rejoindre la société BORLAND INTERNATIONAL. BORLAND INTERNATIONAL

Trois patches pour PICS

Voici trois patches pour le programme PICS. Pourquoi ces patches ? Examinons-les un par un.

1. RDTBL.PICS.GSB

Comme nous avons vu dans le n°18 de *Tremplin Micro* (Janvier), il n'est pas possible d'utiliser en mode graphique SHGR les caractères accentués lorsqu'ils sont pris directement depuis le clavier. Pour contourner cette impossibilité, il faut recoder ces caractères. Ce patch est là pour ça !

Syntaxe d'une ligne de data comprenant un (ou plusieurs) caractères accentués :

```
1000 DATA " * ç LC'est çç136ç quel sujet?çN257V* Aa"
1010 DATA " * ç LC'est çç136ççç136ç quel sujet?çN257V*Aa"
```

La ligne 1000 comprend un 'à' ; la ligne 1010 en comprend deux. On constate la présence d'un 'ç' après l'astérisque en début de data ; puis deux 'ç' suivis du code ASCII étendu du caractère désiré (ici : 'à') et enfin un 'ç' terminateur. D'autre part, en fin de ligne, j'ai ajouté un équivalent clavier soit : '* Aa' (Pomme-A ou Pomme-a) ; cet équivalent clavier doit commencer par un astérisque et être suivi par deux caractères :

- soit deux caractères identiques,
- soit un caractère suivi d'un espace,
- soit enfin une majuscule et une minuscule.

2. PUT.PICS.GSB

En Basic Applesoft, nous avons **STORE** et **RESTORE** qui sauvent sur disquette les variables employées dans le programme et qui les chargent lorsqu'on se sert à nouveau de ce même programme.

En GSBasic, on peut considérer que **PUT £n** et **GET £n** sont dans ce cas. Nous avions vu (dans *Tremplin Micro* n°19) une première utilisation de **PUT £n** et **GET £n**. En fait, ces instructions permettent de charger dans un tableau d'octets (suffixe '!') ou de sauver à partir du même type de tableau des données. Présentement, il s'agit des datas de définition des fenêtres, des menus et autres messages...

On doit : soit saisir le programme à partir de la revue, soit EXECuter le fichier **PUT.PICS.TXT** figurant sur la disquette d'accompagnement, puis lancer le programme.

3. GET.PICS.GSB

Une fois les opérations 1 et 2 effectuées, il ne nous reste plus qu'à mettre en place la récupération des datas à partir de la disquette.

Procédons de la même façon que pour **PUT.PICS.GSB**, puis tapons : **DEL 15800,24100** pour enlever les routines de reconnaissance des datas et les datas eux-mêmes (dont nous n'avons plus besoin).

Et maintenant à nos chronomètres ! On ne peut que constater la lenteur d'exécution de notre PICS original par rapport à ce PICS turbo !

1 - RDTBL.PICS.GSB

```
16800 DEFDOSET:DEF FN doset%(destà,vstr$)
16900          LOCAL a%,b%,c%,d%,e%,f%,h%,l%,x%,a$
17000 GOBBLE:  IF MID$(vstr$,1,1)=" " THEN vstr$=MID$(vstr$,2):GOTO gobble
17100          x%=INSTR("!%£à$^*&",MID$(vstr$,1,1))
17105          a%=INSTR("ç",MID$(vstr$,2,1))
17110          ON a% AND x% GOTO french
```

```
17200           IF x% THEN vstr$=MID$(vstr$,2)
17300           IF MID$(vstr$,1,1)="0" THEN vstr$=STR$(TEN(vstr$))
17400           ON x% GOTO doint,doint,dbl,doint,bstr,xstr,xstr,doint
17500           SET(t!(0),4)=CONV(vstr$):l%=4:GOTO done
17600 DBL:      SET(t!(0),8)=CONV£(vstr$):l%=8:GOTO done
17700 DOINT:    IF MID$(vstr$,1,1)="0" THEN vstr$=STR$(TEN(vstr$))
17800           t&=CONV&(vstr$):l%=x%:IF t&=0 THEN GOTO dozero
17900           ELSE SET(t!(0),x%)=t&:GOTO done
18000 XSTR:     IF x%=6 THEN vstr$=CHR$(LEN(vstr$))+vstr$
18100               ELSE vstr$=vstr$+CHR$(0)
18200 BSTR:     l%=LEN(vstr$):SET(t!(0))=*VARPTR$(vstr$),l%:GOTO done
18210 FRENCH:   vstr$=MID$(vstr$,3)+CHR$(0)
18215           c%=INSTR(vstr$,"çç",1)
18220           SET(t!(0))=MID$(vstr$,1,c%-1)
18225           d%=INSTR(vstr$,"ç",c%+2)
18230           b%=VAL(MID$(vstr$,c%+2,(d%-c%-2)))
18235           SET(t!(c%-1))=b%
18240 TEST:     e%=INSTR(vstr$,"çç",d%+1):f%=f%+1
18245           IF e% THEN h%=d%:GOTO suite
18250               ELSE SET(t!(c%))=MID$(vstr$,d%+1)
18255           l%=LEN(vstr$)-(LEN(STR$(b%))+2)*f%:GOTO done
18260 SUITE:    d%=INSTR(vstr$,"ç",e%+3)
18270           IF e%>(h%+1) THEN a$=MID$(vstr$,h%+1,e%-(h%+1)):SET(t!(c%))=A$:
                c%=c%+LEN(A$)
18300           SET(t!(c%))=VAL(MID$(vstr$,e%+2,d%-(e%+2))):c%=c%+1:GOTO test
18400 DONE:     topà=destà+l%-1:x%=0
18500           FOR pkà=destà TO topà
18600               z%=t!(x%)
18700               POKE pkà,z%
18800               x%=x%+1
18900           NEXT
19000 DOZERO:   FN doset%=l%
19100           END FN doset%
```

2 - PUT.PICS.GSB

```
14910 TEXT:HOME:PRINT "Un instant, je sauve le contenu des tableaux"
14920 OPEN "PICS.256.F2", FILTYP=242 AS£1,256
14930 PUT£1,256,0;menu1!(0)
14940 PUT£1,256,1;menu2!(0)
14950 PUT£1,256,2;menu3!(0)
14960 PUT£1,256,3;message!(0)
14970 PUT£1,256,4;msg1!(0)
14980 PUT£1,256,5;dtitle!(0)
14990 PUT£1,256,6;w1title!(0)
15000 PUT£1,256,7;w2title!(0)
15010 PUT£1,256,8;w3title!(0)
15020 PUT£1,256,9;w4title!(0)
15030 CLOSE£1
15050 OPEN "PICS.78.F3", FILTYP=243 AS£1,78
15060 PUT£1,78,0;dpb!(0)
15070 PUT£1,78,1;w1pb!(0)
15080 PUT£1,78,2;w2pb!(0)
15090 PUT£1,78,3;w3pb!(0)
15100 PUT£1,78,4;w4pb!(0)
```

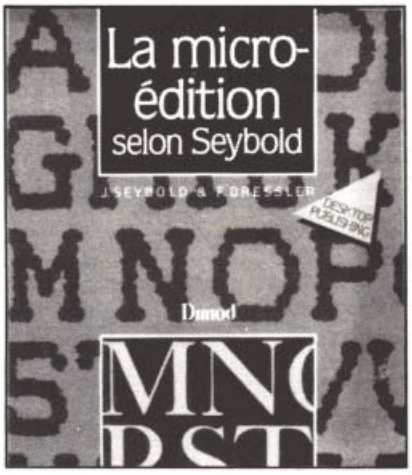
```
15110 PUT£1,78,5;slinfo!(0)
15120 PUT£1,78,6;SrcRect!(0)
15130 CLOSE£1
15140 END
```

3 - GET.PICS.GSB

```
2000            LIBFIND "WindStartUp",q1%,q2%,q3%
2100            IF q1%=0 THEN LIBRARY "6/quickdraw.tdf"
2200            PROC please.wait
2300            OPEN "PICS.256.F2", FILTYP=242 FOR INPUT AS£1,256
2400            GET£1,256,0;menu1!(0)
2500            GET£1,256,1;menu2!(0)
2600            GET£1,256,2;menu3!(0)
2700            GET£1,256,3;message!(0)
2800            GET£1,256,4;msg1!(0)
2900            GET£1,256,5;dtitle!(0)
3000            GET£1,256,6;w1title!(0)
3100            GET£1,256,7;w2title!(0)
3200            GET£1,256,8;w3title!(0)
3250            GET£1,256,9;w4title!(0)
3300            CLOSE£1
3310            PROC msg("'Nother Moment Please ...")
3400            OPEN "PICS.78.F3", FILTYP=243 FOR INPUT AS£1,78
3500            GET£1,78,0;dpb!(0):GET£1,78,1;w1pb!(0)
3600            GET£1,78,2;w2pb!(0):GET£1,78,3;w3pb!(0)
3650            GET£1,78,4;w4pb!(0):GET£1,78,5;slinfo!(0)
3700            GET£1,78,6;SrcRect!(0)
3710            CLOSE£1
3800            PROC msg("'Nother 'Nother Moment Please ...")
```

DUNOD, 17 rue Rémy-Dumoncel, B.P. 50, 75661 PARIS Cedex 14 — Tél. : (16-1) 43.20.15.50.

18×21, 1987, 320 pages, broché, 220 F TTC



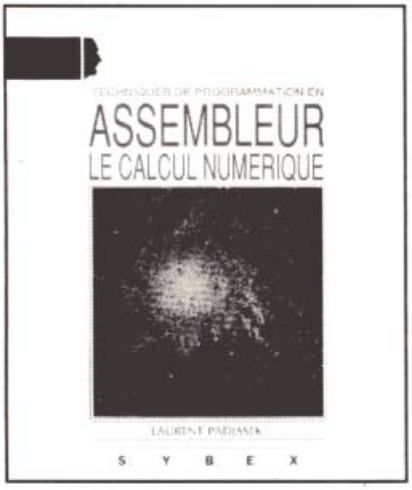
• LA MICRO-ÉDITION SELON SEYBOLD par John Seybold et Fritz Dressler

Micro-édition ou "Desktop Publishing", édition électronique, publication assistée par ordinateur (PAO), édition personnelle, informatique éditoriale, autant de mots pour traduire un phénomène qui bouleverse le monde de la communication écrite en multipliant les accès directs à l'édition.

Considérée aujourd'hui par les spécialistes comme l'un des plus importants vecteurs de développement de la micro-informatique dans la société, la micro-édition est née il y a quelques années de la disponibilité simultanée de micro-ordinateurs graphiques, de logiciels de traitement de texte et de mise en page et d'imprimantes laser de qualité.

Face à ces innovations techniques, l'ouvrage de John Seybold et Fritz Dressler est une référence tant pour les professionnels de l'imprimerie, des arts graphiques, de l'édition... que pour tous ceux qui ont à produire des documents imprimés, en particulier dans les entreprises.

Du dessin des caractères à l'image numérique, de la saisie du texte à la mise en page, de la composition à l'impression laser, les auteurs montrent comment, en respectant les règles traditionnelles de la typographie, on peut atteindre une qualité professionnelle sur les matériels de la micro-édition.



Livre broché, format 19×23, 384 pages — 298 F

SYBEX, 6-8 Impasse du Curé, 75018 PARIS
Tél. : (16-1) 42.03.95.95.

• TECHNIQUES DE PROGRAMMATION EN ASSEMBLEUR : LE CALCUL NUMÉRIQUE par Laurent Padjasek

Cet ouvrage présente les bases de calcul informatique. Il propose l'ensemble des algorithmes permettant d'évaluer et de compiler les expressions mathématiques, de la plus simple à la plus complexe.

Le livre s'adresse à la fois aux concepteurs et aux utilisateurs de logiciels et bien sûr à tous ceux qui souhaitent construire leur propre calculateur.

Cinq thèmes y sont traités : la représentation binaire des nombres entiers suivie de l'exposé des algorithmes permettant de simuler les opérations mathématiques ; la représentation binaire des nombres entiers signés suivie de l'exposé des algorithmes permettant de simuler les opérations mathématiques ; la représentation binaire des nombres réels et l'exposé des algorithmes permettant de simuler les opérations mathématiques ainsi que les fonctions telles que, cos, sin, tan, Ln, etc. ; l'évaluation d'une expression mathématique ; la compilation d'une expression mathématique.

Chacune des parties théoriques exposée est suivie d'une application concrète dont les routines sont utilisées pour la conception de quatre logiciels : un vérificateur syntaxique, deux évaluateurs d'expressions et un traceur de fonctions. Toute la programmation a été réalisée sur IBM PC à l'aide du logiciel MASM de Microsoft.

SPRINT VERSION 1.01 Mise à jour technique

La version 1.Ø1 de SPRINT est une mise à jour technique caractérisée par des fonctionnalités supplémentaires et des améliorations ergonomiques.

LES FONCTIONNALITÉS Parmi les fonctionnalités supplémentaires, SPRINT 1.Ø1 intègre : l'interface utilisateur WordPerfect ; le support de la souris (Microsoft PC Mouse, Logi Mouse et compatibles) ; le support des claviers PS/2.

SPRINT possède également des outils de conversions permettant de transférer : les fichiers saisis sous Word 3.Ø vers SPRINT ou inversement ; les fichiers WordPerfect 4.2, Textor 3.Ø, Wordstar 3.4 et Wordstar 2ØØØ vers SPRINT.

LES AMÉLIORATIONS Parmi les améliorations ergonomiques, SPRINT est encore plus simple à installer (utilitaires batch) et possède de nouvelles commandes de : création de Dictionnaire, création de Glossaire, visualisation des textes sous Infomerge.

SPRINT possède de nouveaux fichiers d'exemples pour les manipulations de tableaux et d'Infomerge. Les perfectionnements interviennent aussi au niveau de l'éditeur, du formateur et de l'Infomerge de SPRINT. La version 1.Ø1 associe à la documentation de SPRINT un addenda de vingt pages, qui complète les manuels actuels et explique les nouvelles fonctionnalités. Un kit de mise à jour SPRINT est envoyé gratuitement à tous les utilisateurs ayant retourné leur licence.

BORLAND INTERNATIONAL

FONTES

Commande permettant de charger une fonte dans votre imprimante.

Yvan KŒNIG vous explique d'abord comment utiliser le module RELOPRO. Ne négligez surtout pas la lecture de la première partie de son article.

RELOPRO encore une fois

Il semble, à en croire les réactions de nombreux lecteurs de *Tremplin Micro,* que le module relogeur qui accompagnait plusieurs de mes programmes pose parfois problème.

Il faisait sa première apparition dans une version préliminaire, avec PRO.TYPC, dans *T.M.* n°9. Il revenait en complément de la routine de recopie d'écran DHGR de P. Quettier dans *T.M.* n°10, avec SLOADC dans *T.M.* n°11. Ensuite, dans *T.M.* n°14, il accompagnait SONSIR (le source figurait dans la revue). Enfin, il était également à l'œuvre dans les commandes de la disquette THGS.

Comme, entre temps, je l'avais envoyé à de nombreux lecteurs pour mettre en place des commandes externes ProDOS écrites sur mesure, et ce, sans aucun problème, il ne me semblait pas utile de m'apesantir sur un module donnant apparemment satisfaction.

Je me trompais. Revenons donc au RELOPRO...

Toutes les versions officiellement commercialisées de BASIC.SYSTEM contiennent un point d'entrée appelé GETBUFR, implanté en $BEF5 (il avait été omis dans le ProDOS Technical Manual, mais existait dans le programme). Si (A) est le contenu de l'accumulateur, la routine dans laquelle on entre en GETBUFR descend proprement de (A) pages, les tampons (buffers) du BASIC.SYSTEM en compagnie des variables chaînes implantées juste au-dessous de ceux-ci. Cela permet, entre autres choses, de créer une zone-mémoire dans laquelle on peut installer, en toute sûreté (ou presque) une routine ou un groupe de données.

Certains auteurs en sont restés aux habitudes héritées du DOS 3.3 et se bornent à manipuler HIMEM. C'est une formule *cavalière,* mais qui peut fonctionner si on l'utilise correctement. Pour ce faire, il convient de fixer HIMEM sur une limite de page-mémoire et de se souvenir que BASIC.SYSTEM utilisera au premier accès disque, un tampon de 4 pages qu'il croira, en toute candeur, implanté juste au-dessus de HIMEM. En outre, à chaque ouverture de fichier, il baladera joyeusement ce qui se trouvera en-dessous de HIMEM. Par contre, si on descend HIMEM de 4 pages, par exemple, on peut être certain d'être tranquille en s'installant dans les 4 pages qui vont de $9600 à $99FF puisque le tampon à tout faire sera désormais de $9200 à $95FF.

RELOPRO est un outil destiné à permettre d'utiliser correctement GETBUFR.

Si vous disposez de l'assembleur MERLIN, prévoyez d'écrire vos programmes en 3 parties : un en-tête que j'appelle généralement K.Routine.S, le source de la routine proprement dite que j'appelle Routine.S et le fichier source RELOPRO.S que l'on écrit une fois pour toutes.

L'en-tête se borne à appeler les deux autres modules et à en organiser l'assemblage en fonction de 5 variables utilisées par RELOPRO :

- HASADRS signale la présence éventuelle d'une table d'adresses dans le module actif ;
- HASDATAS signale la présence éventuelle d'une zone de simples données ;
- EXTERNE indique si l'on est en présence d'une commande externe ProDOS ou d'un module d'un autre type ;
- PROTEGE précise si l'on doit ajuster la BITMAP, ceci afin d'indiquer au BASIC.SYSTEM qu'il lui est interdit de charger quoi que ce soit dans la zone libérée.
- INIROUT permet de demander que l'on saute au point bas de la zone libérée pour initialiser une routine Ampersand par exemple.

Pour que tout cela fonctionne, il suffit de respecter quelques règles simples : *(suite page 46).*

La partie programme doit commencer par les instructions :

```
DOSENTRY    jmp    PARSE
NEXTCMD
DOSEXIT     jmp    XRETURN    ($BE9E)
AMPEXIT     jmp    IORTS      ($FF58)
.............
PARSE       analyse de la commande
```

elle doit se terminer par le label ***ENDPROG*** *suivi par les éventuelles adresses ; ensuite, label impératif ENDADRS suivi par d'éventuels 'datas' avec en clôture :*

```
LAST   = * - 1
ENDALL = *
```

Si ces règles sont respectées, RELOPRO se chargera d'abaisser les tampons du BASIC.SYSTEM et d'installer votre module actif dans la zone libérée en en modifiant les adresses internes en fonction de la position réelle de cette zone.

Les appels au MLI seront relogés correctement. Attention ! si une instruction manipule le seul octet haut d'une adresse interne au module actif, il faudra prendre des précautions : soit récupérer cet octet haut à partir d'une adresse stockée dans la zone d'adresses, soit utiliser l'octet haut d'une adresse dans une instruction si l'on est sûr que le résultat sera fiable. Je choisis, en règle générale, cette dernière formule (cf. le label HIBIT dans FLOAD).

On peut même (que ceux qui ne veulent pas entendre parler du GS passent au paragraphe suivant !), si l'on fait très attention, écrire des routines contenant du code 65C816 bien que le module 'relogeur' de RELOPRO ne connaisse que les opcodes 65CØ2 (ceux qui ont acheté THGS ont pu s'en rendre compte).

Je tiens à préciser que les premières instructions du module actif sont indispensables afin que les routines puissent être aisément intégrées, comme je le signale systématiquement, à l'utilitaire ProCMD de Glen BREDON.

La commande FLOAD

Cette fois, je vous ai resservi RELOPRO pour installer une commande qui devrait réjouir nombre de lecteurs qui m'ont interrogé sur ce thème : FLOAD permet de charger une fonte dans votre imprimante.

Le programme est présenté dans une version correspondant à l'équipement **standard** : une unité centrale accompagnée d'un dispositif de liaison série carte SSC si *IIe*, port série si *IIc* ou *IIGS*. Il est cependant prévu des conditionnels permettant de traiter, après réassemblage, d'autres configurations : bonne vieille DMP à liaison parallèle ou machines de la galaxie EPSON, en présicant cependant que certaines d'entre elles, comme la LX8ØØ, ne permettent pas de charger une fonte, mais seulement quelques caractères.

L'en-tête K.FLOAD.S contient en prime un rappel de la structure des fichiers fontes utilisables. En ce qui concerne les imprimantes APPLE, les fontes générées par CARAC.1.Ø de *T.M.* n°6 (repris en catimini dans la disquette UTIL A Ø19 de VIF) feront l'affaire. Ceux qui se seront procurés mes extensions AppleWriter (Graphiques et Fontes) en auront quelques autres. La disquette de *T.M.* n°20 en présente plusieurs, pour IMAGEWRITER et EPSON.

Dans mes fontes, je ne définis pas de caractères $7F pour éviter quelques désagréments. En effet, ce caractère est accepté par l'IMW2 (lorsqu'elle n'est pas en panne) mais ne l'est pas sur l'increvable IMW1. Je m'abstiens également d'utiliser le second banc de caractères personnalisés, celui que l'on appelle par ESC "* ", à cause d'un bug de l'IMW2 (dont je vous ai entretenu par ailleurs — le manuel français n'en parle toujours pas). Je me suis arraché suffisamment de cheveux, lors de la mise au point de la routine Hard-Copie du Bureau de MINIE (je précise pour éviter toute interprétation malveillante que je devais bien cela à Claude AUBRY qui est à l'origine de RELOPRO).

Comme toujours avec mes commandes externes, je conseille de les mettre en place à partir d'une configuration **propre** à la suite d'un boot ou d'un appel à PRO.FP. Le programme utilisateur considérera que la commande a été mise à sa disposition.

Allez, je vous laisse, le téléphone sonne. J'espère que ce n'est pas pour un problème de fontes...

Y. K.

```
                    1
                2   ************************************************************
                3   *                                                          *
                4   *        K.FLOAD.S   utilise RELOPRO.S et FLOAD.S          *
                5   *                                                          *
                6   *           FLOAD fonte (,A slotimp, S £, D £)             *
                7   *                                                          *
                8   *                    Slot 1  par défaut                    *
                9   *                                                          *
               10   ************************************************************
               11   * Yvan KOENIG                          VALLAURIS le 23/01/88 *
               12   ************************************************************
               13   * Cette commande est conçue pour intégration éventuelle    *
               14   * à ProCMD de Glen BREDON.     (Editeur Call A.P.P.L.E.)   *
               15   ************************************************************
=0000          16   HASADRS  =     0          ;1 = table adresses,   0 sinon *
=0001          17   HASDATAS =     1          ;1 = table DATAs,      0 sinon *
=0001          18   EXTERNE  =     1          ;1 = commande externe, 0 sinon *
=0000          19   PROTEGE  =     0          ;1 = on protège la routine     *
=0000          20   INIROUT  =     0          ;1 = on saute dans la routine  *
               21   ************************************************************
=0000          22   DISK     KBD   "DISK    (1=obj/disque   0=Non  ) "
               23   *********************************
               24   *        Structure Fonte APPLE
               25   *
               26   *        DFB   $1B
               27   *        DFB   '-'            si aucun carac n'a plus de 8 cols
               28   *       (DFB   '+')           si au moins 1 carac > 8cols
               29   *                             mais limSup=16
               30   *
               31   *        DFB   $1B,'I'        début de chargement
               32   *
               33   *------------------------------
               34   *        pour chaque caractère
               35   *        DFB   'char'
               36   *        DFB   64+Nbcol+(32)     (32) si on emploie les
               37   *                                8 aiguilles basses
               38   *        DFB   p1
               39   *        DFB   p2
               40   *        ........
               41   *        DFB   pNbcol
               42   *------------------------------
               43   *
               44   *        DFB   4              marque fin de fonte
               45   *
               46   *********************************
               47   *        Structure Fonte EPSON
               48   *
               49   *        DFB   $1B,"&",0
               50   *        DFB   1er code ASCII
               51   *        DFB   dernier code ASCII
               52   *
               53   *------------------------------
               54   *        pour chaque caractère
               55   *        DFB   attribut        Veuillez consulter votre manuel
               56   *        DFB   p1
               57   *        DFB   p2
               58   *        ........
               59   *        DFB   p11
               60   *********************************
```

```
                61            TR    ADR
                63            PUT   RELOPRO
               >1
               >2    ************************************************
               >3    *  Module Relogeur  destiné à placer une routine *
               >4    *          entre ProDOS et ses buffers.          *
               >5    *                                                *
               >6    *   La routine peut etre une commande externe,   *
               >7    * une simple routine (CALL) ou une routine AMPER *
               >8    *                                                *
               >9    *    Prendre garde  à la position d'une table    *
               >10   *    d'adresses et (ou) d'une table de données   *
               >11   *                                                *
               >12   * Yvan KOENIG                          le 25-04-86 *
               >13   ************************************************
               >14   *                                                *
               >15   *HASADRS  = 1     ;1 si table adresses, 0 sinon   *
               >16   *HASDATAS = 1     ;1 si table DATAs, 0 sinon      *
               >17   *EXTERNE  = 1     ;1 si commande externe, 0 sinon *
               >18   *PROTEGE  = 0     ;1 si on protège la routine     *
               >19   *INIROUT  = 0     ;1 si on saute dans la routine  *
               >20   *                                                *
               >21   ************************************************
               >22   * On demande à GETBUFR de nous allouer (^) pages *
               >23   * puis le relogeur copie la routine              *
               >24   * dans la zone qui nous est attribuée.           *
               >25   * Le relogeur utilise sa propre routine pour     *
               >26   * identifier les instructions afin de traiter    *
               >27   * correctement les opérateurs spécifiques  65c02 *
               >28   * On traitera correctement les appels au MLI     *
               >29   * Attention cependant, les tables MLI devront    *
               >30   * etre placées en zone DATAs et                  *
               >31   * initialisées par le programme                  *
               >32   ************************************************
               >33
=0002          >34   R_OFFSET =     2          ;&3
=0004          >35   R_ZBEG   =     4          ;&5
=0006          >36   R_ZEND   =     6          ;&7
=003C          >37   R_A1     =     $3C        ;Début du bloc à déplacer
=003E          >38   R_A2     =     $3E        ;Fin du bloc pour MOVE
=0042          >39   R_A4     =     $42        ;Début zone dest. pour MOVE
=00FD          >40   R_INDIR  =     $FD        ;&FE
               >41
               >42   * Adresses ProDOS
               >43
=BE00          >44   CI_ENTRY =     $BE00
=BE06          >45   EXTRNCMD =     CI_ENTRY+6
=BEF5          >46   GETBUFR  =     $BEF5
=BF00          >47   MLI      =     $BF00
=BF58          >48   BITMAP   =     $BF58
               >49
               >50   * Adresses MONITEUR
               >51
=FCB4          >52   R_NXTA4  =     $FCB4      ;Avance R_A4 puis tombe en NXTA1
=FDED          >53   R_COUT   =     $FDED      ;Affiche le caractère (A)
=FE2C          >54   R_MOVE   =     $FE2C      ;Déplace (R_A1 L/H--R_A2 L/H)
               >55                              ; vers R_A4 L/H
               >56
               >57   *--------------------------------------------------
               >58           ORG   $6080
               >59   *--------------------------------------------------
```

```
                    >60
                    >61   *-------------------------------------------------------
                    >62
6080: A9 02         >63           LDA   £LAST-PROGRAM/$100+1
6082: 20 F5 BE      >64           JSR   GETBUFR     ;Libère (A)pages et revient avec
                    >65                             ;partie haute adresse dans (A)
6085: 90 65 =60EC   >66           BCC   PAGEFND
6087: A0 16         >67           LDY   £CANTEND-CANTRELO-1
6089: B9 95 60      >68   :loop   LDA   CANTRELO,Y
608C: 20 ED FD      >69           JSR   R_COUT
608F: 88            >70           DEY
6090: 10 F7 =6089   >71           BPL   :loop
6092: 4C 00 BE      >72           JMP   CI_ENTRY
                    >73
                    >74   *******************************************************
6095: 8D 87         >75   CANTRELO HEX   8D,87
6097: C5 C3 C1 CC   >76           REV   "IL N'Y A PAS DE PLACE"
609B: D0 A0 C5 C4 A0 D3 C1 D0 A0 C1 A0 D9 A7 CE A0 CC
60AB: C9
                    >77   CANTEND
                    >78   *******************************************************
                    >79
60AC: 04 15 04 2A   >80   OPTABLE  DFB   %00000100,%00010101,%00000100,%00101010
60B0: 05 14 08 2A   >81           DFB   %00000101,%00010100,%00001000,%00101010
60B4: 16 15 04 2A   >82           DFB   %00010110,%00010101,%00000100,%00101010
60B8: 15 15 08 2A   >83           DFB   %00010101,%00010101,%00001000,%00101010
60BC: 04 15 04 2A   >84           DFB   %00000100,%00010101,%00000100,%00101010
60C0: 15 14 08 28   >85           DFB   %00010101,%00010100,%00001000,%00101000
60C4: 04 15 04 2A   >86           DFB   %00000100,%00010101,%00000100,%00101010
60C8: 15 15 08 2A   >87           DFB   %00010101,%00010101,%00001000,%00101010
60CC: 15 15 00 2A   >88           DFB   %00010101,%00010101,%00000000,%00101010
60D0: 15 15 08 2A   >89           DFB   %00010101,%00010101,%00001000,%00101010
60D4: 15 15 04 2A   >90           DFB   %00010101,%00010101,%00000100,%00101010
60D8: 05 15 08 2A   >91           DFB   %00000101,%00010101,%00001000,%00101010
60DC: 05 15 04 2A   >92           DFB   %00000101,%00010101,%00000100,%00101010
60E0: 15 14 08 28   >93           DFB   %00010101,%00010100,%00001000,%00101000
60E4: 05 15 04 2A   >94           DFB   %00000101,%00010101,%00000100,%00101010
60E8: 15 14 08 28   >95           DFB   %00010101,%00010100,%00001000,%00101000
                    >96
                    >97   *******************************************************
                    >98
60EC: 85 43         >99   PAGEFND  STA   R_A4+1      ;partie haute adr. zone allouée
                    >100
                    >101  *-------------------------------------------------------
                    >102          DO    EXTERNE
60EE: AE 08 BE      >103          LDX   EXTRNCMD+2 ;sauve l'ancienne adresse pour
60F1: 8E 05 62      >104          STX   NEXTCMD+2  ;chainer les commandes
60F4: 8D 08 BE      >105          STA   EXTRNCMD+2 ;met en place
60F7: AE 07 BE      >106          LDX   EXTRNCMD+1 ;la nouvelle adresse
60FA: 8E 04 62      >107          STX   NEXTCMD+1
                    >108  *
                    >109  *NEXTCMD JMP   $0000 ;On doit trouver ce label
                    >110  *        et cette instruction dans toute
                    >111  *        commande externe pour assurer
                    >112  *        le chainage des diverses commandes
                    >113  *    cf. PRO.TYPC dans Tremplin MICRO n°9
                    >114  *
                    >115          ELSE
                    >116
                    >117          STA   R_INDIR+1  ;Octet haut adresse routine
```

```
                    >118            LDA   £0          ; relogée
                    >119            STA   R_INDIR     ; octet bas dito
                    >120            DS    9,$EA       ;pour fixer la suite
                    >121
                    >122            FIN
                    >123  *---------------------------------------------------
                    >124
60FD: A9 00         >125            LDA   £<PROGRAM   ;Début de zone à déplacer
60FF: 85 3C         >126            STA   R_A1
6101: 85 04         >127            STA   R_ZBEG
6103: A9 62         >128            LDA   £>PROGRAM
6105: 85 3D         >129            STA   R_A1+1
6107: 85 05         >130            STA   R_ZBEG+1
                    >131
6109: A9 8B         >132            LDA   £<ENDPROG-1 ;Fin de la partie programme
610B: 85 3E         >133            STA   R_A2
610D: A9 63         >134            LDA   £>ENDPROG-1
610F: 85 3F         >135            STA   R_A2+1
                    >136
6111: A9 C6         >137            LDA   £<ENDALL-1
6113: 85 06         >138            STA   R_ZEND
6115: A9 63         >139            LDA   £>ENDALL-1
6117: 85 07         >140            STA   R_ZEND+1
                    >141
6119: D8            >142            CLD               ;Au cas où
                    >143
611A: A9 00         >144            LDA   £0          ;Place PROGRAM sur début de page
611C: 85 42         >145            STA   R_A4
                    >146
                    >147  *---------------------------------------------------
                    >148            DO    EXTERNE
611E: 8D 07 BE      >149            STA   EXTRNCMD+1
                    >150            ELSE
                    >151            DS    3,$EA
                    >152            FIN
                    >153  *---------------------------------------------------
                    >154
6121: 38            >155            SEC
6122: E5 04         >156            SBC   R_ZBEG      ;Calcule offset
6124: 85 02         >157            STA   R_OFFSET    ; que l'on ajoutera
6126: A5 43         >158            LDA   R_A4+1      ; aux adresses
6128: E5 05         >159            SBC   R_ZBEG+1    ; devant être 'relogées'
612A: 85 03         >160            STA   R_OFFSET+1
                    >161
612C: A0 00         >162  :reloc1   LDY   £0
612E: B1 3C         >163            LDA   (R_A1),Y    ;Opcode
6130: C9 20         >164            CMP   £$20        ;est-ce JSR ?
6132: D0 17 =614B   >165            BNE   :notMLI
6134: C8            >166            INY
6135: B1 3C         >167            LDA   (R_A1),Y    ;lobyte
6137: D0 12 =614B   >168            BNE   :notMLI     ;ce n'est pas <MLI
6139: C8            >169            INY
613A: B1 3C         >170            LDA   (R_A1),Y
613C: C9 BF         >171            CMP   £>MLI
613E: D0 0B =614B   >172            BNE   :notMLI
                    >173
6140: A0 00         >174            LDY   £0
6142: A2 02         >175            LDX   £3-1        ; Pour
6144: 20 C3 61      >176            JSR   COPIE       ; copier JSR MLI
6147: A9 02         >177            LDA   £3-1        ;Trompe la suite pour
```

```
6149: D0 16 =6161 >178           BNE   :opdone     ;reloger l'adresse table
                  >179
                  >180                             ;Se souvenir qu'après JSR MLI
                  >181                             ;on doit trouver:
                  >182                             ;   DFB code à exécuter
                  >183                             ;   DA  adresse table paramètres
614B: A0 00       >184  :notMLI  LDY   £0
614D: B1 3C       >185           LDA   (R_A1),Y    ;Opcode
614F: 48          >186           PHA
6150: 29 03       >187           AND   £%00000011  ;A= 0 1 2 3 ; Offset
6152: A8          >188           TAY               ; dans l'octet de la table
6153: 68          >189           PLA
6154: 4A          >190           LSR               ; A=OPC/2
6155: 4A          >191           LSR               ; A=OPC/4
6156: AA          >192           TAX
6157: BD AC 60    >193           LDA   OPTABLE,X   ;contient longueur de 4 opcodes
615A: 88          >194  :oplup   DEY
615B: 30 04 =6161 >195           BMI   :opdone     ;On a poussé à droite les 2 bits
615D: 4A          >196           LSR               ;représentant la longueur
615E: 4A          >197           LSR               ;de l'instruction
615F: D0 F9 =615A >198           BNE   :oplup      ;Si A=0, ne branche pas
                  >199                             ;Opérande de longueur nulle
                  >200                             ; Attention, Y est indéterminé
6161: 29 03       >201  :opdone  AND   £%00000011  ;Isole la longueur
6163: AA          >202           TAX               ; X= 2 1 0
6164: A0 00       >203           LDY   £0          ;Lève l'indétermination
6166: E0 02       >204           CPX   £3-1
6168: 90 05 =616F >205           BCC   :movit1     ;on ne change pas les
                  >206                             ;instructions de 1/2 octets
616A: C8          >207           INY               ; -> Y=1
616B: 20 9C 61    >208           JSR   CHANGIT?
616E: 88          >209           DEY               ; -> Y=0
                  >210  :movit1                    ;Ici X=Longueur opérande (0,1,2)
616F: 20 C3 61    >211           JSR   COPIE       ;Copie l'instruction
6172: 90 B8 =612C >212           BCC   :reloc1     ;Si (R_A1)<(R_A2)
                  >213
                  >214  *-------------------------------
                  >215  :mov2?   DO    HASADRS
                  >216           BCC   :mov3?      ;Ne branche pas si table Adresses
                  >217           ELSE
6174: B0 12 =6188 >218           BCS   :mov3?      ;Branche si pas de table Adresses
                  >219           FIN
                  >220  *-------------------------------
                  >221
6176: A9 8B       >222           LDA   £<ENDADRS-1 ;Fin de table d'adresses
6178: 85 3E       >223           STA   R_A2
617A: A9 63       >224           LDA   £>ENDADRS-1
617C: 85 3F       >225           STA   R_A2+1
                  >226
                  >227  :reloc2                    ;Ici Y=0
617E: 20 9C 61    >228           JSR   CHANGIT?
                  >229
6181: A2 01       >230  :movit2  LDX   £2-1        ;2 octets par adresse
6183: 20 C3 61    >231           JSR   COPIE
6186: 90 F6 =617E >232           BCC   :reloc2     ;si l'on n'a pas fini la table
                  >233
                  >234  *-------------------------------
                  >235  :mov3?   DO    HASDATAS
6188: 90 0B =6195 >236           BCC   :exit       ;Ne branche pas si table DATAs
                  >237           ELSE
```

```
                    >238            BCS   :exit       ;Branche si pas de table DATAs
                    >239            FIN
                    >240  *-------------------------------
                    >241
618A: A5 06         >242            LDA   R_ZEND      ;Fin de table de valeurs
618C: 85 3E         >243            STA   R_A2
618E: A5 07         >244            LDA   R_ZEND+1
6190: 85 3F         >245            STA   R_A2+1
                    >246
6192: 20 2C FE      >247            JSR   R_MOVE
6195: 20 CE 61      >248  :exit     JSR   PROTECT
                    >249
                    >250  *-------------------------------
                    >251            DO    INIROUT
                    >252            NOP
                    >253            ELSE
6198: 60            >254            RTS
                    >255            FIN
                    >256  *-------------------------------
                    >257
6199: 6C FD 00      >258            JMP   (R_INDIR)   ;Exécute si INIROUT = 1
                    >259
                    >260  *=============================================================
                    >261  *
                    >262  *                                                  C'est relogé
                    >263  *
                    >264  *=============================================================
                    >265
                    >266  CHANGIT?
619C: A5 06         >267            LDA   R_ZEND
619E: D1 3C         >268            CMP   (R_A1),Y
61A0: C8            >269            INY
61A1: A5 07         >270            LDA   R_ZEND+1
61A3: F1 3C         >271            SBC   (R_A1),Y
61A5: 90 1A =61C1   >272            BCC   :end        ;Pas de changement
                    >273
61A7: 88            >274            DEY
61A8: A5 04         >275            LDA   R_ZBEG
61AA: D1 3C         >276            CMP   (R_A1),Y
61AC: C8            >277            INY
61AD: A5 05         >278            LDA   R_ZBEG+1
61AF: F1 3C         >279            SBC   (R_A1),Y
61B1: B0 0E =61C1   >280            BCS   :end        ;Pas de changement
                    >281                              ;Ici C=0
61B3: 88            >282            DEY
61B4: B1 3C         >283            LDA   (R_A1),Y
61B6: 65 02         >284            ADC   R_OFFSET    ;Ajuste
61B8: 91 3C         >285            STA   (R_A1),Y    ; l'adresse
61BA: C8            >286            INY
61BB: B1 3C         >287            LDA   (R_A1),Y
61BD: 65 03         >288            ADC   R_OFFSET+1
61BF: 91 3C         >289            STA   (R_A1),Y
                    >290
61C1: 88            >291  :end      DEY
61C2: 60            >292            RTS
                    >293
                    >294  ************************************************************
                    >295
61C3: B1 3C         >296  COPIE     LDA   (R_A1),Y    ;Copie X+1 octets
61C5: 91 42         >297            STA   (R_A4),Y    ;à leur destination
```

```
61C7: 20 B4 FC     >298           JSR   R_NXTA4
61CA: CA           >299           DEX
61CB: 10 F6 =61C3  >300           BPL   COPIE
61CD: 60           >301           RTS
                   >302
                   >303  ************************************************************
                   >304
                   >305  PROTECT  DO    PROTEGE
                   >306           NOP                ;On protègera
                   >307           ELSE
61CE: 60           >308           RTS                ;On ne protègera pas
                   >309           FIN
                   >310
61CF: A5 05        >311           LDA   R_ZBEG+1
61D1: 18           >312           CLC
61D2: 65 02        >313           ADC   R_OFFSET
61D4: AA           >314           TAX                ;(Haut) plus basse page utilisée
61D5: 18           >315           CLC
61D6: 69 02        >316           ADC   £LAST-PROGRAM/$100+1 ;Nombre de pages
61D8: 85 43        >317           STA   R_A4+1       ;(Haut) page suivant la routine
61DA: 8A           >318  :loop     TXA
61DB: 48           >319           PHA
61DC: 4A           >320           LSR
61DD: 4A           >321           LSR
61DE: 4A           >322           LSR
61DF: A8           >323           TAY
61E0: 8A           >324           TXA
61E1: 29 07        >325           AND   £7
61E3: AA           >326           TAX
61E4: A9 00        >327           LDA   £0
61E6: 38           >328           SEC
61E7: 6A           >329  :L       ROR
61E8: CA           >330           DEX
61E9: 10 FC =61E7  >331           BPL   :L
61EB: 19 58 BF     >332           ORA   BITMAP,Y
61EE: 99 58 BF     >333           STA   BITMAP,Y
61F1: 68           >334           PLA
61F2: AA           >335           TAX
61F3: E8           >336           INX
61F4: E4 43        >337           CPX   R_A4+1
61F6: 90 E2 =61DA  >338           BCC   :loop
61F8: 60           >339           RTS
                   >340
                   >341           ERR   *-1/$6200
61F9: 00 00 00 00  >342           DS    $6200-*      ;Remplissage
61FD: 00 00 00
                   >343
                   >344  *------------------------------------------------------------
                   >345  PROGRAM                     ;Controler début de page
                    65            PUT   FLOAD
                   >1                                ; ORG $9800
                   >2    ********************************
                   >3    *                              *
                   >4    *            FLOAD.S           *
                   >5    *                              *
                   >6    * Yvan KOENIG      le 23/01/88 *
                   >7    ********************************
                   >8            TR    ADR
         =001B     >9    ESC     =     $1B
                   >10
```

```
          =0006   >11   Ptr      =     $06        ;&07 Pointe /tampon fonte
          =0008   >12   SavY     =     $08
          =0009   >13   CntH     =     $09
                  >14
          =0036   >15   CSW      =     $36        ;&37
          =006D   >16   STREND   =     $6D        ;&6E Haut zone variables
          =006F   >17   FRETOP   =     $6F        ;&70 Bas zone chaine
          =0073   >18   HIMEM    =     $73        ;&74
          =00B7   >19   CHRGOT   =     $B7
                  >20
                  >21   ************************************************
          =0001   >22   SERIAL   KBD   "SERIAL  (1=Série   0=Parallèle ) "
          =0001   >23   APPLE    KBD   "APPLE   (1=Apple   0=Epson     ) "
                  >24   ************************************************
                  >25
6200: 4C 1E 62    >26   DOSENTRY JMP   PARSE
                  >27   NEXTCMD
6203: 4C 9E BE    >28   DOSEXIT  JMP   XRETURN
6206: 4C 58 FF    >29   AMPEXIT  JMP   IORTS
                  >30
6209: D0 FB =6206 >31   AMPENTRY BNE   AMPEXIT
620B: A0 00       >32            LDY   £0
620D: B9 AE 63    >33   :1       LDA   CMDNAME,Y
6210: 20 ED FD    >34            JSR   COUT
6213: C8          >35            INY
6214: C0 19       >36            CPY   £AMPEND-CMDNAME+1
6216: 90 F5 =620D >37            BCC   :1
6218: 20 B7 00    >38            JSR   CHRGOT
621B: F0 E9 =6206 >39            BEQ   AMPEXIT
                  >40
621D: EA          >41   IDBYTE   HEX   EA         ; Provisoire
                  >42
                  >43   ********************************
                  >44
621E: A2 05       >45   PARSE    LDX   £CMDEND-CMDNAME
6220: BD FF 01    >46   :1       LDA   IN-1,X
6223: 29 DF       >47            AND   £$DF
6225: 5D AD 63    >48            EOR   CMDNAME-1,X
6228: 38          >49            SEC
6229: D0 D8 =6203 >50            BNE   DOSEXIT
622B: CA          >51            DEX
622C: D0 F2 =6220 >52            BNE   :1
622E: 8E 53 BE    >53            STX   XCNUM      ; X=0
6231: A9 04       >54            LDA   £CMDEND-CMDNAME-1 ; Longueur 'commande'
6233: 8D 52 BE    >55            STA   XLEN
6236: A9 81       >56            LDA   £%10000001
6238: 8D 54 BE    >57            STA   PBITS      ; Il faut 1 pathname
623B: A9 84       >58            LDA   £%10000100
623D: 8D 55 BE    >59            STA   PBITS+1    ; Autorise A, S, D
6240: A9 4D       >60            LDA   £<TrueCMD
6242: 8D 50 BE    >61            STA   XTRNADDR
6245: AD 47 62    >62            LDA   HIBIT
        =6247     >63   HIBIT    =     *-1        ; C'est l'octet haut de TrueCMD
6248: 8D 51 BE    >64            STA   XTRNADDR+1
624B: 18          >65            CLC
624C: 60          >66            RTS
                  >67
                  >68   ************************************************************
                  >69
                  >70   TrueCMD
```

```
624D: AD 56 BE     >71             LDA   FBITS       ; A-t-on spécifié
6250: 4A           >72             LSR               ; un nom de fichier ?
6251: 90 0B =625E  >73             BCC   Invalide    ; NON
6253: A5 70        >74             LDA   FRETOP+1
6255: E5 6E        >75             SBC   STREND+1    ; OK, C=1
6257: E9 09        >76             SBC   £7+1+1      ; Il faut au moins 7 pages libres
                   >77                               ; 7+1 car STREND pas garanti XX00
                   >78                               ; 7+1+1 pour simplifier le test
6259: B0 07 =6262  >79             BCS   GOGET       ; assez de place
                   >80
625B: A9 0E        >81             LDA   £$0E        ; PROGRAM TOO LARGE
625D: 2C           >82             HEX   2C          ; Saute instruction suivante
625E: A9 0B        >83   Invalide  LDA   £$0B        ; INVALID PARAMETER
6260: 38           >84             SEC
6261: 60           >85             RTS
                   >86
6262: A9 0A        >87   GOGET     LDA   £$0A
6264: 8D B4 BE     >88             STA   SSGINFO
6267: A9 C4        >89             LDA   £$C4        ; GET INFO
6269: 20 70 BE     >90             JSR   GOSYSTEM    ; =============== GET INFO font
626C: B0 4B =62B9  >91             BCS   ClosERR     ; Erreur ProDOS
                   >92
626E: AD B8 BE     >93             LDA   FIFILID
6271: C9 06        >94             CMP   £$06        ; BIN ?
6273: D0 42 =62B7  >95             BNE   typeMISS
                   >96
6275: A2 01        >97             LDX   £1          ; Slot 1 par défaut
6277: AD 57 BE     >98             LDA   FBITS+1
627A: 10 08 =6284  >99             BPL   :settype    ; On n'a pas donné A (notre slotimp)
                   >100
627C: AD 58 BE     >101  :getadr   LDA   VADDR       ; Récupère
627F: 29 07        >102            AND   £$07
6281: F0 01 =6284  >103            BEQ   :settype    ; si 0 , slot 1
6283: AA           >104            TAX
                   >105
6284: 8E 8C 63     >106  :settype  STX   OurSlot
                   >107
6287: A4 74        >108  :setbuf   LDY   HIMEM+1
6289: 8C CF BE     >109            STY   OSYSBUF+1   ; utiliser GP buffer
628C: A9 C8        >110            LDA   £$C8        ; OPEN
628E: 20 70 BE     >111            JSR   GOSYSTEM    ; ================== OPEN file
6291: B0 26 =62B9  >112            BCS   ClosERR     ; erreur
6293: AD D0 BE     >113            LDA   OREFNUM     ; num ref file1
6296: 8D D6 BE     >114            STA   RWREFNUM
6299: 8D DE BE     >115            STA   CFREFNUM
629C: 8D C7 BE     >116            STA   SUNITNUM    ; pour get eof
629F: A9 D1        >117            LDA   £$D1        ; GET EOF
62A1: 20 70 BE     >118            JSR   GOSYSTEM    ; ================ GET EOF fonte
62A4: B0 13 =62B9  >119            BCS   ClosERR     ; erreur
                   >120
62A6: AD CA BE     >121            LDA   SEOF+2
62A9: D0 0C =62B7  >122            BNE   typeMISS    ; ce n'est pas pour nous
62AB: AD C8 BE     >123            LDA   SEOF
62AE: C9 C6        >124            CMP   £<1734
62B0: AD C9 BE     >125            LDA   SEOF+1
62B3: E9 06        >126            SBC   £>1734
62B5: 90 09 =62C0  >127            BCC   fileok
                   >128
62B7: A9 0D        >129  typeMISS  LDA   £$0D        ; FILE TYPE MISMATCH
62B9: 48           >130  ClosERR   PHA
```

```
62BA: 20 67 63    >131            JSR   DOCLOSE
62BD: 68          >132            PLA
62BE: 38          >133            SEC
62BF: 60          >134            RTS
                  >135
62C0: AD C8 BE    >136  fileok    LDA   SEOF
62C3: 8D D9 BE    >137            STA   RWCOUNT
62C6: AD C9 BE    >138            LDA   SEOF+1
62C9: 8D DA BE    >139            STA   RWCOUNT+1
                  >140
62CC: A4 6E       >141            LDY   STREND+1
62CE: A5 6D       >142            LDA   STREND
62D0: F0 01 =62D3 >143            BEQ   :noinc
62D2: C8          >144            INY                 ; Implante tampon lecture
62D3: 8C D8 BE    >145  :noinc    STY   RWDATA+1      ; au-dessus zone des variables
62D6: 84 07       >146            STY   Ptr+1
62D8: A9 00       >147            LDA   £0
62DA: 8D D7 BE    >148            STA   RWDATA
62DD: 85 06       >149            STA   Ptr
62DF: 85 09       >150            STA   CntH
                  >151
62E1: A9 CA       >152            LDA   £$CA          ; READ  file
62E3: 20 70 BE    >153            JSR   GOSYSTEM
62E6: B0 D1 =62B9 >154            BCS   ClosERR
62E8: 20 67 63    >155            JSR   DOCLOSE
                  >156
                  >157  * Prépare routine de Chargement
                  >158
62EB: AD 8C 63    >159            LDA   OurSlot
62EE: 09 C0       >160            ORA   £$C0          ; A= Cs
                  >161
                  >162  *===============================
                  >163            DO    SERIAL
                  >164  *-------------------------------
62F0: 8D 80 63    >165            STA   GOCARD+2
62F3: 8D 8B 63    >166            STA   vector+2
62F6: 8D 11 63    >167            STA   which1+2
62F9: 8D 56 63    >168            STA   which2+2
62FC: 0A          >169            ASL
62FD: 0A          >170            ASL
62FE: 0A          >171            ASL
62FF: 0A          >172            ASL
6300: 8D 88 63    >173            STA   Yslot16+1     ; A= s0
6303: 18          >174            CLC
6304: 69 8A       >175            ADC   £$8A
6306: 8D 5E 63    >176            STA   reset+1
                  >177  *-------------------------------
                  >178            ELSE
                  >179  *-------------------------------
                  >180            STA   wait+2
                  >181            ASL
                  >182            ASL
                  >183            ASL
                  >184            ASL
                  >185            CLC
                  >186            ADC   £$80
                  >187            STA   sendit+1
                  >188
                  >189            JMP   :suite
                  >190            DS    9,$EA         ; Remplissage
```

```
                    >191  :suite
                    >192  *------------------------------
                    >193          FIN
                    >194  *==============================
                    >195
6309: A5 36         >196          LDA   CSW
630B: 48            >197          PHA
630C: A5 37         >198          LDA   CSW+1
630E: 48            >199          PHA
                    >200
                    >201  *==============================
                    >202          DO    SERIAL
                    >203  *------------------------------
630F: AD 11 C1      >204  which1  LDA   $C111       ; adresse ajustée
6312: C9 85         >205          CMP   £$85        ; est-ce une SSC ?
6314: D0 08 =631E   >206          BNE   init
6316: A9 09         >207          LDA   £9          ; restaure Ctrl-I au cas où...
6318: AC 8C 63      >208          LDY   OurSlot
631B: 99 F8 05      >209          STA   $5F8,Y
631E: A0 0D         >210  init     LDY   £$0D
6320: 20 7E 63      >211          JSR   GOCARD      ; initialise Firmware"
                    >212  *------------------------------
                    >213          ELSE
                    >214  *------------------------------
                    >215          JMP   :suit2
                    >216          DS    17,$EA
                    >217  :suit2
                    >218  *------------------------------
                    >219          FIN
                    >220  *==============================
                    >221
                    >222  *-------- ENVOYER PROLOGUE
                    >223
6323: A0 EA         >224          LDY   £prolog-prolend
6325: B9 A3 62      >225  :p1      LDA   prolend-$100,Y
6328: 20 6F 63      >226          JSR   KOUT
632B: C8            >227          INY
632C: D0 F7 =6325   >228          BNE   :p1
                    >229
                    >230  *-------- ENVOYER FONTE
                    >231
632E: A0 00         >232  :f1      LDY   £0
6330: B1 06         >233          LDA   (Ptr),Y
6332: 20 6F 63      >234          JSR   KOUT
6335: E6 06         >235          INC   Ptr
6337: D0 04 =633D   >236          BNE   :f2
6339: E6 07         >237          INC   Ptr+1
633B: E6 09         >238          INC   CntH
633D: A5 06         >239  :f2      LDA   Ptr
633F: CD C8 BE      >240          CMP   SEOF
6342: A5 09         >241          LDA   CntH
6344: ED C9 BE      >242          SBC   SEOF+1
6347: 90 E5 =632E   >243          BCC   :f1
                    >244
                    >245  *-------- ENVOYER EPILOGUE
                    >246
6349: A0 F5         >247          LDY   £epilog-epilend
634B: B9 AE 62      >248  :e1      LDA   epilend-$100,Y
634E: 20 6F 63      >249          JSR   KOUT
6351: C8            >250          INY
```

```
6352: D0 F7 =634B >251          BNE   :e1
                  >252
                  >253  *===============================
                  >254          DO    SERIAL
                  >255  *-------------------------------
6354: AD 11 C1    >256  which2  LDA   $C111       ; adresse ajustée
6357: C9 85       >257          CMP   £$85        ; est-ce une SSC ?
6359: D0 05 =6360 >258          BNE   exit        ; NON
635B: A9 00       >259          LDA   £0          ; faire croire à Reset Interface
635D: 8D 9A C0    >260  reset   STA   $C08A+$10   ; adresse ajustée
                  >261  *-------------------------------
                  >262          ELSE
                  >263  *-------------------------------
                  >264          JMP   :suit3
                  >265          DS    9,$EA
                  >266  :suit3
                  >267  *-------------------------------
                  >268          FIN
                  >269  *===============================
                  >270
6360: 68          >271  exit    PLA
6361: 85 37       >272          STA   CSW+1
6363: 68          >273          PLA
6364: 85 36       >274          STA   CSW
6366: 60          >275          RTS
                  >276
                  >277  *===============================
                  >278
6367: 8D DE BE    >279  DOCLOSE STA   CFREFNUM
636A: A9 CC       >280          LDA   £$CC        ; CLOSE
636C: 4C 70 BE    >281          JMP   GOSYSTEM
                  >282
                  >283  *===============================
                  >284
636F: 08          >285  KOUT    PHP
6370: 78          >286          SEI
                  >287  *===============================
                  >288          DO    SERIAL
                  >289  *-------------------------------
6371: 84 08       >290          STY   SavY
6373: 48          >291          PHA
6374: A0 0F       >292          LDY   £$0F        ; WRITE
6376: 20 7E 63    >293          JSR   GOCARD
6379: 68          >294          PLA
637A: A4 08       >295          LDY   SavY
                  >296  *-------------------------------
                  >297          ELSE
                  >298  *-------------------------------
                  >299  wait    BIT   $C1C1       ; Octet haut ajusté
                  >300          BMI   wait
                  >301  sendit  STA   $C090       ; Octet bas ajusté
                  >302          DS    3,$EA
                  >303  *-------------------------------
                  >304          FIN
                  >305  *===============================
637C: 28          >306          PLP
637D: 60          >307          RTS
                  >308
                  >309  *===============================
                  >310
```

```
637E: BE 00 C1     >311  GOCARD   LDX   $C100,Y    ; adresse ajustée
6381: 8E 8A 63     >312          STX   vector+1
6384: AE 80 63     >313          LDX   GOCARD+2   ; Cslot
6387: A0 10        >314  Yslot16  LDY   £$10       ; Slot16
6389: 4C 00 C1     >315  vector   JMP   $C100      ; adresse ajustée
                   >316
                   >317  *================================
                   >318  ENDPROG
                   >319  *================================
                   >320  ENDADRS
                   >321  *================================
                   >322
638C: 00           >323  OurSlot  DS    1
                   >324
638D: 8D           >325  prolog   HEX   8D
638E: 09           >326          DFB   9
638F: 32 35 34 4E  >327          ASC   '254N'     ;    pas d'écho
                   >328  *================================
                   >329          DO    APPLE
                   >330  *--------------------------------
6393: 1B 5A 00 20  >331          DFB   ESC,'Z',0,32 ;  reconnait bit8
6397: 1B 5A 80 00  >332          DFB   ESC,'Z',$80,0 ; pas de LF après CR
639B: 09 5A        >333          DFB   9,'Z'      ;    ZAP cmd
                   >334  *--------------------------------
                   >335          ELSE
                   >336  *--------------------------------
                   >337          DFB   ESC,'£'    ;    annule forçage du bit8
                   >338          DFB   ESC,':',0,0,0 ; copie le jeu en RAM
                   >339          DFB   ESC,'&',0  ;    début chargement
                   >340  *--------------------------------
                   >341          FIN
                   >342  *================================
639D: 00 00 00 00  >343          DS    6          ; Pour adaptations
63A1: 00 00
                   >344  prolend                   ;-----
                   >345
                   >346  *================================
                   >347  epilog   DO    APPLE
                   >348  *--------------------------------
63A3: 1B 44 00 20  >349          DFB   ESC,'D',0,32 ; ignore bit8
63A7: 8D           >350          HEX   8D
                   >351  *--------------------------------
                   >352          ELSE
                   >353          DS    5
                   >354  *--------------------------------
                   >355          FIN
                   >356  *================================
63A8: 00 00 00 00  >357          DS    6          ; Pour adaptations
63AC: 00 00
                   >358  epilend                   ;-----
                   >359
                   >360  *================================
                   >361
63AE: C6 CC CF C1  >362  CMDNAME  ASC   "FLOAD"
63B2: C4
63B3: A0 F0 E1 F4  >363  CMDEND   ASC   " path, <,A slotimp>"
63B7: E8 AC A0 BC AC C1 A0 F3 EC EF F4 E9 ED F0 BE
63C6: 8D           >364  AMPEND   HEX   8D
         =63C6     >365  LAST     =     *-1
         =63C7     >366  ENDALL   =     *
```

```
          >367
          >368  *================================
          >369
=0200     >370  IN        =     $0200
          >371
=BE50     >372  XTRNADDR  =     $BE50       ;&BE51
=BE52     >373  XLEN      =     $BE52
=BE53     >374  XCNUM     =     $BE53
=BE54     >375  PBITS     =     $BE54       ;&BE55
=BE56     >376  FBITS     =     $BE56       ;&BE57
=BE58     >377  VADDR     =     $BE58       ;&BE59
=BE70     >378  GOSYSTEM  =     $BE70
=BE9E     >379  XRETURN   =     $BE9E
=BEB4     >380  SSGINFO   =     $BEB4
=BEB8     >381  FIFILID   =     $BEB8
=BEC7     >382  SUNITNUM  =     $BEC7
=BEC8     >383  SEOF      =     $BEC8       ;&BEC9,BECA
=BECE     >384  OSYSBUF   =     $BECE       ;&BECF
=BED0     >385  OREFNUM   =     $BED0
=BED6     >386  RWREFNUM  =     $BED6
=BED7     >387  RWDATA    =     $BED7       ;&BED8
=BED9     >388  RWCOUNT   =     $BED9       ;&BEDA
=BEDE     >389  CFREFNUM  =     $BEDE
          >390
=FDED     >391  COUT      =     $FDED
=FF58     >392  IORTS     =     $FF58       ; RTS garanti APPLE
           67   *********************************
           68              DO    DISK
           69              SAV   FLOAD
           70              FIN
           71   *********************************
```

End Merlin-16 assembly,

839 bytes, errors: 0 ,

symbol table: $1800-$1D28

FLOAD,A$6080,L$0347

```
6080: A9 02 20 F5 BE 90 65 A0 16 B9 95 60 20 ED FD 88 10 F7 4C 00 BE 8D 87 C5 C3 C1 CC D0 A0 C5 C4 A0   783C
60A0: D3 C1 D0 A0 C1 A0 D9 A7 CE A0 CC C9 04 15 04 2A 05 14 08 2A 16 15 04 2A 15 15 08 2A 04 15 04 2A   6276
60C0: 15 14 08 28 04 15 04 2A 15 15 08 2A 15 15 00 2A 15 15 08 2A 15 15 04 2A 05 15 08 2A 05 15 04 2A   9E98
60E0: 15 14 08 28 05 15 04 2A 15 14 08 28 85 43 AE 08 BE 8E 05 62 8D 08 BE AE 07 BE 8E 04 62 A9 00 85   AF13
6100: 3C 85 04 A9 62 85 3D 85 05 A9 8B 85 3E A9 63 85 3F A9 C6 85 06 A9 63 85 07 D8 A9 00 85 42 8D 07   7551
6120: BE 38 E5 04 85 02 A5 43 E5 05 85 03 A0 00 B1 3C C9 20 D0 17 C8 B1 3C D0 12 C8 B1 3C C9 BF D0 0B   37CC
6140: A0 00 A2 02 20 C3 61 A9 02 D0 16 A0 00 B1 3C 48 29 03 A8 68 4A 4A AA BD AC 60 88 30 04 4A 4A D0   9A51
6160: F9 29 03 AA A0 00 E0 02 90 05 C8 20 9C 61 88 20 C3 61 90 B8 B0 12 A9 8B 85 3E A9 63 85 3F 20 9C   CE24
6180: 61 A2 01 20 C3 61 90 F6 90 0B A5 06 85 3E A5 07 85 3F 20 2C FE 20 CE 61 60 6C FD 00 A5 06 D1 3C   EE61
61A0: C8 A5 07 F1 3C 90 1A 88 A5 04 D1 3C C8 A5 05 F1 3C B0 0E 88 B1 3C 65 02 91 3C C8 B1 3C 65 03 91   043D
61C0: 3C 88 60 B1 3C 91 42 20 B4 FC CA 10 F6 60 60 A5 05 18 65 02 AA 18 69 02 85 43 8A 48 4A 4A 4A A8   FABA
61E0: 8A 29 07 AA A9 00 38 6A CA 10 FC 19 58 BF 99 58 BF 68 AA E8 E4 43 90 E2 60 00 00 00 00 00 00 00   EA58
6200: 4C 1E 62 4C 9E BE 4C 58 FF D0 FB A0 00 B9 AE 63 20 ED FD C8 C0 19 90 F5 20 B7 00 F0 E9 EA A2 05   78BD
6220: BD FF 01 29 DF 5D AD 63 38 D0 D8 CA D0 F2 8E 53 BE A9 04 8D 52 BE A9 81 8D 54 BE A9 84 8D 55 BE   7C1D
6240: A9 4D 8D 50 BE AD 47 62 8D 51 BE 18 60 AD 56 BE 4A 90 0B A5 70 E5 6E E9 09 B0 07 A9 0E 2C A9 0B   1949
6260: 38 60 A9 0A 8D B4 BE A9 C4 20 70 BE B0 4B AD B8 BE C9 06 D0 42 A2 01 AD 57 BE 10 08 AD 58 BE 29   BC6D
6280: 07 F0 01 AA 8E 8C 63 A4 74 8C CF BE A9 C8 20 70 BE B0 26 AD D0 BE 8D D6 BE 8D DE BE 8D C7 BE A9   8425
62A0: D1 20 70 BE B0 13 AD CA BE D0 0C AD C8 BE C9 C6 AD C9 BE E9 06 90 09 A9 0D 48 20 67 63 68 38 60   FC59
62C0: AD C8 BE 8D D9 BE AD C9 BE 8D DA BE A4 6E A5 6D F0 01 C8 8C D8 BE 84 07 A9 00 8D D7 BE 85 06 85   D715
62E0: 09 A9 CA 20 70 BE B0 D1 20 67 63 AD 8C 63 09 C0 8D 80 63 8D 8B 63 8D 11 63 8D 56 63 0A 0A 0A 0A   57F4
6300: 8D 88 63 18 69 8A 8D 5E 63 A5 36 48 A5 37 48 AD 11 C1 C9 85 D0 08 A9 09 AC 8C 63 99 F8 05 A0 0D   1F4D
6320: 20 7E 63 A0 EA B9 A3 62 20 6F 63 C8 D0 F7 A0 00 B1 06 20 6F 63 E6 06 D0 04 E6 07 E6 09 A5 06 CD   C327
6340: C8 BE A5 09 ED C9 BE 90 E5 A0 F5 B9 AE 62 20 6F 63 C8 D0 F7 AD 11 C1 C9 85 D0 05 A9 00 8D 9A C0   402E
6360: 68 85 37 68 85 36 60 8D DE BE A9 CC 4C 70 BE 08 78 84 08 48 A0 0F 20 7E 63 68 A4 0B 28 60 BE 00   031D
6380: C1 8E 8A 63 AE 80 63 A0 10 4C 00 C1 00 8D 09 32 33 32 4E 1B 5A 00 20 1B 5A 80 00 09 5A 00 00 00   A8F2
63A0: 00 00 00 1B 44 00 20 8D 00 00 00 00 00 00 C6 CC CF C1 C4 A0 F0 E1 F4 E8 AC A0 BC AC C1 A0 F3 EC   EC33
63C0: EF F4 E9 ED F0 BE 8D                                                                           F4F4
```

Sur l'Apple IIGS

Depuis huit mois, je me pose bon nombre de questions sur l'Apple *IIGS*. J'ai lu beaucoup d'articles dans diverses revues *(Tremplin Micro,* bien sûr, mais aussi dans *POM'S* et *l'Ordinateur Individuel),* mais je ne suis pas tellement plus avancé.

Dans un article paru dans l'O.I., N. Bréaud-Pouliquen écrit : «On pourrait envisager de le *fer*mer (il s'agit du GS) comme le *IIc* et de le rendre beaucoup plus rapide».

Première question :

— Comment fermer mon Apple *IIGS* ?

L'auteur du même article évoque ensuite l'existence de cartes permettant l'extension jusqu'à 8 Mo de RAM et *l'implantation de modules de ROM éventuellement reprogrammables.*

Deuxième question :

— Où me procurer ces accessoires (mon concessionnaire, interrogé, me répond que j'aurais dû m'offrir un MAC) ?

Dans la foulée...

Troisième question :

— Au cours de mes lectures, j'ai remarqué que vos programmes chargeaient des fichiers binaires de dessin. Comment les créer ?

Quatrième question :

— Peut-on installer des programmes dans la RAM obtenue par «Cat, S3,D2» ?

Un drogué du GS — M. N. (51350 Cormontreuil)

RÉPONSE

Je ne devrais pas vous répondre puisque vous avez omis de joindre un timbre à votre lettre.

Cependant, votre flot de questions me donne l'occasion de donner mon avis sur des points sensibles :

1. J'ai expliqué dans *T.M. n°17* comment sauver un écran SHGR dans un fichier BIN. Je n'y reviens pas. Il faut savoir que c'est un mode de stockage 'déconseillé'. Il est préférable d'utiliser les formats officiels définis par APPLE (cf. article cité). La disquette qui accompagnait *T.M. n°17* contenait en prime un programme permettant de créer, à partir d'images HGR ou DHGR, des images SHGR noir et blanc.

2. Lorsqu'on travaille sous P8, on peut effectivement stocker des fichiers dans le disque virtuel /RAM en S3,D2. Il peut être utile de protéger certaines zones, mais j'ai déjà traité la question dans *Tremplin Micro.*

   Attention ! certains auteurs emploient le banc Ø1 comme tampon lors du chargement d'images SHGR, ce qui est fort désagréable pour /RAM qui utilise justement ce banc.

   Sous P16, je ne me suis jamais servi de /RAM car je pense que le banc Ø1 est utilisé par ce système. N'oubliez pas que les bancs Ø et 1 sont réalisés en RAM rapide, alors que le reste de la mémoire est en RAM lente.

3. Dans *l'Ordinateur Individuel,* Nicole Bréaud-Pouliquen souhaite un GS fermé. Personnellement, je ne crains pas les courants d'air.

   Dans l'article que vous citez, elle décrivait ce qui, selon elle, serait souhaitable pour une future version de la machine. Celle-ci existera sans doute, mais j'espère qu'elle ne correspondra pas aux souhaits de Nicole...

   Quelques fabricants américains proposent des cartes d'extension permettant déjà l'installation de ROMs, mais le système d'exploitation étant en perpétuelle évolution, je ne saurais conseiller cette formule.

   Je vous orienterais plus volontiers vers le trio suivant :

   - carte extension APPLE 1 méga
   - carte extension APPLIED 1,5 méga ou plus
   - carte 'alimentation permanente' RAM-KEEPER sur laquelle seront branchées les cartes susnommées.

   Vous auriez ainsi la possibilité de travailler avec une mémoire importante, un disque virtuel 'permanent', mais reprogrammable, ce qui vous permettrait de suivre l'évolution du système d'exploitation et de vos besoins. Bien entendu, ce n'est pas un ROMdisk au sens défini par les concepteurs du GS, mais, qui s'amuserait à les mettre en ROM, tant que les outils ne sont pas stabilisés ? N'oubliez pas que certains outils de la nouvelle Rom sont réécrits pour moitié en Ram lors du boot.

   APPLE ouvre les MACS, Nicole veut fermer les GS, au secours ! Marthe Richard revient !

Yvan KŒNIG — 6 Avril 1988

Bug dans système Pascal Le Pascal 1.3 présente une incompatibilité avec la nouvelle ROM du GS. Un appel à un outil pendant une opération critique provoque une erreur lors de l'écriture sur disque. Si vous utilisez dans ce langage des programmes traitant des données importantes (comptabilité, par exemple), contactez d'urgence votre revendeur. Il trouvera confirmation du problème dans CALL APPLE d'avril, page 15. Avec un peu de chance, il pourra obtenir d'Apple France les informations lui permettant de mettre en place le PATCH indispensable : un octet à changer dans SYSTEM.APPLE. Apple U.S. devrait diffuser un Pascal corrigé fin avril. ***Y. K.***

PRODOS.YEAR

De nombreux utilisateurs des 'vieux' modèles APPLE 2 utilisent encore des vieilles versions de PRODOS. S'ils disposent d'une carte horloge THUNDERCLOCK ou assimilée, ils ont eu la surprise de remarquer que depuis le 1er janvier 1988 leurs fichiers sont datés 1983 au lieu de 1988.

Les lecteurs de Call A.P.P.L.E. étaient prévenus depuis janvier 86, les autres ont eu l'impression de rajeunir de 5 ans.

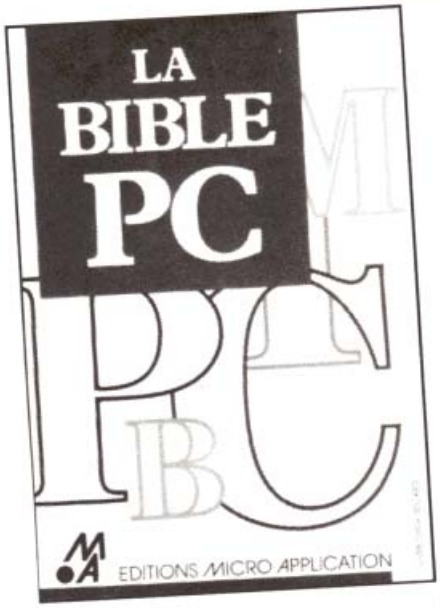
Le petit programme BASIC joint (PRODOS.YEAR) vous donne la possibilité d'actualiser votre fichier PRODOS afin qu'il date correctement jusqu'au 31/12/90.

Si vous utilisez toujours un 'vieux' PRODOS en 1990, il vous suffira de remplacer les lignes 200 et 210 par des REMs pour qu'une nouvelle exécution de PRODOS.YEAR vous permette de repartir correctement jusqu'au 31/12/95.

Je tiens cependant à rappeler qu'il est souhaitable d'utiliser la version 1.4 qui est en train d'être remplacée par la version 1.5.

Yvan KŒNIG — 3 Avril 1988.

```
  1 REM ***************
  2 REM * PRODOS.YEAR *
  3 REM ***************ctJ                                        1C5A
100 TEXT : HOME                                                   3FAC
110 TI$ = "PRODOS": REM "peut être P8                             B3A4
120 D$ =  CHR$ (4)                                                18F3
130 PRINT D$"UNLOCK"TI$                                           AB9F
140 PRINT D$"BLOAD"TI$",TSYS,A$2000"                              73DA
150 ADR = 4 * 4096 + 118
160 IF  PEEK (ADR) = 84 AND  PEEK (ADR + 1) = 84 AND  PE
    EK (ADR + 2) = 83 AND  PEEK (ADR + 3) = 82 AND  PEEK
    (ADR + 4) = 87 AND  PEEK (ADR + 5) = 86 AND  PEEK (A
    DR + 6) = 85 GOTO 190                                         4C92
170 ADR = ADR + 256: IF ADR > 6 * 4096 + 118 GOTO 210             2AE3
180 GOTO 160                                                      3942
190 POKE ADR,90: POKE ADR + 1,89: POKE ADR + 2,88: POKE
    ADR + 3,88: POKE ADR + 4,87: POKE ADR + 5,86: POKE A
    DR + 6,85                                                     6169
200 GOTO 270: REM "mettre REM en 1990                             4544
210 GOTO 310: REM "mettre REM en 1990                             273F
220 ADR = 4 * 4096 + 118                                          73DA
230 IF  PEEK (ADR) = 90 AND  PEEK (ADR + 1) = 89 AND  PE
    EK (ADR + 2) = 88 AND  PEEK (ADR + 3) = 88 AND  PEEK
    (ADR + 4) = 87 AND  PEEK (ADR + 5) = 86 AND  PEEK (A
    DR + 6) = 85 GOTO 260                                         849D
240 ADR = ADR + 256: IF ADR > 6 * 4096 + 118 GOTO 310             57E4
250 GOTO 230                                                      2D40
260 POKE ADR,90: POKE ADR + 1,95: POKE ADR + 2,94: POKE
    ADR + 3,93: POKE ADR + 4,92: POKE ADR + 5,92: POKE A
    DR + 6,91                                                     0955
270 REM                                                           2DAE
280 PRINT D$"BSAVE"TI$",TSYS,A$2000"                              C650
290 PRINT D$"LOCK"TI$                                             E2F8
300 PRINT : PRINT "Modification effectuée": PRINT : END           DFEF
310 PRINT : PRINT "Je n'ai rien modifié": PRINT : END
```



• LA BIBLE PC (Tischer)

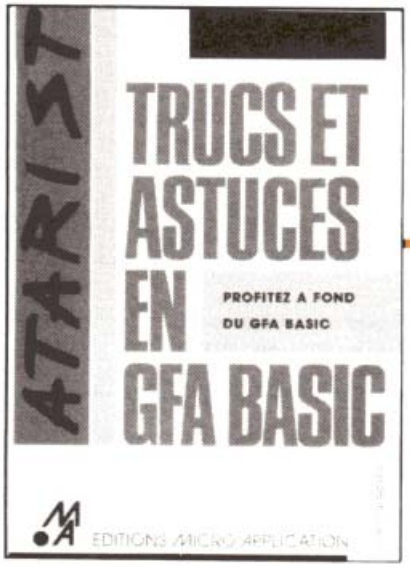
Avec le PCTransporteur (voir le numéro 19 de *T.M.*), vous pouvez mettre un compatible PC dans votre GS (et aussi dans un Apple *II*, d'ailleurs).

Parfait ! mais que savez-vous du hardware, des interruptions, du MS-DOS, du BIOS... des différents langages ?

Quelle est la signification de chaque bit des codes-clavier ? A quelle adresse se trouve la mémoire vidéo de votre carte ?

Cette grosse *BIBLE PC* répondra à toutes ces questions et à beaucoup d'autres. C'est un ouvrage de référence fondamental de 760 pages, sous couverture cartonnée (299 F TTC).

***MICRO APPLICATION**, 13 rue Sainte Cécile, 75009 PARIS*



• APPRENDRE À PROGRAMMER EN TURBO C par C. Delannoy

L'objectif de ce livre est de vous conduire à la maîtrise du Turbo C. Pour ce faire, il vous en propose un apprentissage progressif largement illustré de programmes complets, accompagnés d'exemples d'exécution.

A la fin de la plupart des chapitres, vous trouverez à la fois :

- des exercices vous permettant d'appliquer et d'intégrer les connaissances acquises,
- des manipulations destinées à développer en vous le "savoir-faire" indispensable au développement et à la mise au point de vos applications personnelles.

Cet ouvrage aborde l'ensemble des possibilités du Turbo C, y compris celles qui sont considérées comme les plus pointues et les plus professionnelles (pointeurs, gestion de fichiers, modèles mémoire, récursivité, allocation dynamique, appels systèmes...).

416 pages — 250 F ***Editions EYROLLES,***
61, boulevard Saint-Germain, 75240 PARIS CEDEX 05.

• PROGRAMMER EN QUICK BASIC par P. Bihan

Le *QUICK BASIC* (TM) est un outil de haut niveau qui allie toutes les facilités de programmation d'un Basic classique à la rigueur d'un langage structuré.

Ce livre vous propose un apprentissage "express' et très efficace : la première partie introduit toutes les options de Quick Basic, y compris son éditeur ; les commandes principales du langage sont ensuite présentées de façon synthétique mais détaillée, convenant aussi bien au débutant qu'au programmeur pressé.

Plus de 60 programmes commentés, des sections séparées traitant de la gestion de fichiers, du graphisme, des commandes musicales, des directives de compilation... complètent l'ouvrage pour en faire un vrai guide pratique.

216 pages — 180 F ***Editions EYROLLES,***
61, boulevard Saint-Germain, 75240 PARIS CEDEX 05.

• TRUCS ET ASTUCES EN GFA BASIC

Ce livre s'adresse aussi bien aux informaticiens confirmés qu'aux néophytes.

MICRO APPLICATION présente dans cet ouvrage les meilleurs trucs et astuces pour programmer graphismes, fenêtres et fichiers RSC en GFA BASIC.

Le but de ce livre est d'aider l'utilisateur de ST à exploiter au mieux ce langage, XBIOS, les routines GEMDOS, BIOS.

Principaux sujets traités : Création d'un bureau personnalisé (redessinez les icônes du Desktop !) ; les fonctions du TOS et le GFA BASIC, XBIOS, GEMDOS, BIOS, VDISYS, GEMSYS ; comment programmer les ressources en GFA BASIC ; déplacez, agrandissez, modifiez les fenêtres... ; routines de graphisme (dessins en trois dimensions, représentation de diagrammes Manhattan) ; un moniteur de disque entièrement en GFA BASIC ! ; espionnez la mémoire de votre ATARI ST et voyez directement à l'écran ce qui s'y passe ; comment fonctionne la commande Monitor (). ***MICRO APPLICATION***
372 pages — livre et disquette d'utilitaires — 269 F

/ BULLETIN DE COMMANDE ET D'ABONNEMENT — N°20 /

A retourner à : TREMPLIN MICRO — Senillé — 86100 CHATELLERAULT

Nom ..... Prénom .....

Adresse complète .....

Code postal |_|_|_|_|_| Ville .....

LES DISQUETTES FONCTIONNENT SOUS DOS ET ProDOS (à condition de posséder une version Apple de ce SYSTEME D'EXPLOITATION)

Qté.	DÉSIGNATION	PRIX UNITAIRE	TOTAL
	Abonnement à 6 numéros (un an) : FRANCE à partir du numéro : _____	190 F	
	Abonnement à 6 disquettes (un an) : FRANCE à partir du numéro : _____	600 F	
	MINIE LE BUREAU DE MINIE LES SOURCES DU BUREAU — Logiciels de Claude AUBRY	180 F 150 F 50 F	
	LIVRE : APPLE // PRODOS, GUIDE DU PROGRAMMEUR APPLESOFT	120 F	
	LIVRE : LE 65Ø2 PAS à PAS	120 F	
	LIVRE + DISQUETTE : CLINS D'ŒIL AU 65Ø2	160 F	
	LIVRE + DISQUETTE : ROUTINES LM POUR 65CØ2 ET 65Ø2	160 F	
	LIVRE + DISQUETTE : NOUVELLES ROUTINES POUR LE 65CØ2	160 F	
	LIVRE + DISQUETTE : PROGRAMMES LM 65CØ2 & 65C816	160 F	
	LIVRE + DISQUETTE : A LA DÉCOUVERTE DU GS BASIC	300 F	
	DISQUETTE DE JEU WAR	20 F	
	DISQUETTE DE JEU "Le Monde des Sorciers"	20 F	
	DISQUETTE N°18 + jeu "Le Monde des Sorciers"	105 F	
	DISQUETTE-INDEX (N°1 à 12)	30 F	
	SIGNATURE (CHECK-LIST)	30 F	
	RELIURE-ÉCRIN	40 F	
	SUPPORT POUR MONTAGE CARTE STÉRÉO (quantité limitée)	30 F	
	DISQUETTE DE TREMPLIN MICRO : * *(IIe, IIc, GS)* ☐ N°1 ☐ N°4 ☐ N°7 ☐ N°10 ☐ N°13 ☐ N°16 ☐ N°19 ☐ N°2 ☐ N°5 ☐ N°8 ☐ N°11 ☐ N°14 ☐ N°17 ☐ N°20 ☐ N°3 ☐ N°6 ☐ N°9 ☐ N°12 ☐ N°15 ☐ N°18	105 F	
	NUMÉROS DE TREMPLIN MICRO : * ☐ N°1 ☐ N°2 ☐ N°3 ☐ N°4 ☐ N°5 ☐ N°6 ☐ N°7 ☐ N°8 ☐ N°9 ☐ N°10 ☐ N°11 ☐ N°12 ☐ N°13 ☐ N°14 ☐ N°15 ☐ N°16 ☐ N°17 ☐ N°18 ☐ N°19 ☐ N°20	30 F 33 F	

* *Cochez la (ou les) case(s) de votre choix.*

Merci de libeller votre règlement à l'ordre de TREMPLIN MICRO / Editions JIBENA.

Supplément pour disquette 3,1/2 + 20 F
Participation aux frais d'envoi (gratuit pour les abonnés) + 10 F
Envoi en paquet-poste recommandé à partir de 400 F

Numéro d'abonné ou de client : _____ **Total à payer** _____ F

Mode de règlement choisi :

☐ Chèque
☐ Mandat-lettre
☐ Carte Bleue (complétez ci-contre)

VISA

N° de votre Carte Bleue |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_|

Date d'expiration |_|_|_|_| **Signature (obligatoire)**

Montant à débiter _____ F

Chasseur d'Images

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

Chez votre marchand de journaux !