

# pom's

La revue francophone des utilisateurs de l'Apple

**Disquette mixte DOS/ProDOS**

**Gestion de fenêtres**

**Aldo Reset a la parole...**

**Hard Copy de la page HGR étendue**

**Rendez le DOS transparent**

**Cahier Macintosh**



NUMERO 18 • PRIX 40 F

ISSN : 0294-6068



## PRECISION™ : LES DISQUES SOUPLES XIDEX

UNE NOUVELLE GAMME DE DISQUETTES 8" ET 5" 1/4 SPECIALEMENT DEVELOPPEE POUR LES APPLICATIONS HAUTE DENSITE

UN NIVEAU DE CERTIFICATION ELEVE (65 %)

RESEAU DE DISTRIBUTION REGIONAL EN COURS DE CONSTITUTION

XIDEX : 537 RUE HELENE BOUCHER - ZI 78530 BUC. TEL. : (3) 956.22.23

## Sommaire

	Page	Langage*	Matériel
<b>Editorial</b> par Hervé Thiriez	5		
<b>Disquette mixte DOS/ProDOS</b> par François Sermier	7	B-A	// +, //e, //c
<b>Saisie de variables par écran</b> par Gérard Michel	12	B-A	//e, //c
<b>Le porte-parole</b> par Olivier Herz	20		
<b>Gestion de fenêtres</b> par Georges Zwingelstein	24	B-A	// +, //e-//c
<b>Graphiques aléatoires</b> par Thierry Han	30	B-A	// +, //e-//c
<b>Récupérez les incônes du système</b> par Marianne Sutz	35	B-A	Macintosh
<b>Modifications des informations pour le Finder</b> par Jean-Luc Bazanegue	37	B-A	Macintosh
<b>Les curseurs du système</b> par Julien Thomas	45	B-A-	Macintosh
<b>Micro-journal</b> par Hervé Thiriez	46		
<b>Les pirates ont la parole...</b> par Pom's	47		
<b>Hard-copy de la page HGR étendue</b> par Marc Debuigne	49	B-A	//e
<b>Les pointeurs, suite...</b> par Olivier Herz	54	P	// +, //e, //c-Mac
<b>Tirage au sort !</b>	55		
<b>Rendez le DOS transparent</b> par Jean-Pierre Januel	59	B	//e-//c
<b>Recherche dans un tableau</b> par Patrice Neveu	67	B-A	// +, //e-//c
<b>Micro-informations</b> par Jean-Michel Gourévitch	70		
<b>Bibliographie</b> par Alexandre Duback	73		
<b>Sommaire des précédents numéros</b>	74		
<b>Courrier des lecteurs</b> par Olivier Herz	75		

Langage : B(asic)    A(ssembleur)    P(ascal)

## Les annonceurs

Adefi : p. 61 / Apple : p. 56-57 / Club Apple : p. 25 / Computer 3 : p. 70 / Controle X : p. 21 / Hello : p. 4 / IEF : p. 6 / List : p. 58 / MBDC : p. 69 / PSI : p. 80 / Télécompo : p. 34 / Version Soft : p. 22-23 / Vidéo Technologie : p. 79 / Xidex : p. 2

Apple, est une marque déposée de la société Apple Computer Inc.

**Éditions MEV - 49, rue Lamartine - 78000 Versailles**

Directeur de la publication : Hervé Thiriez.

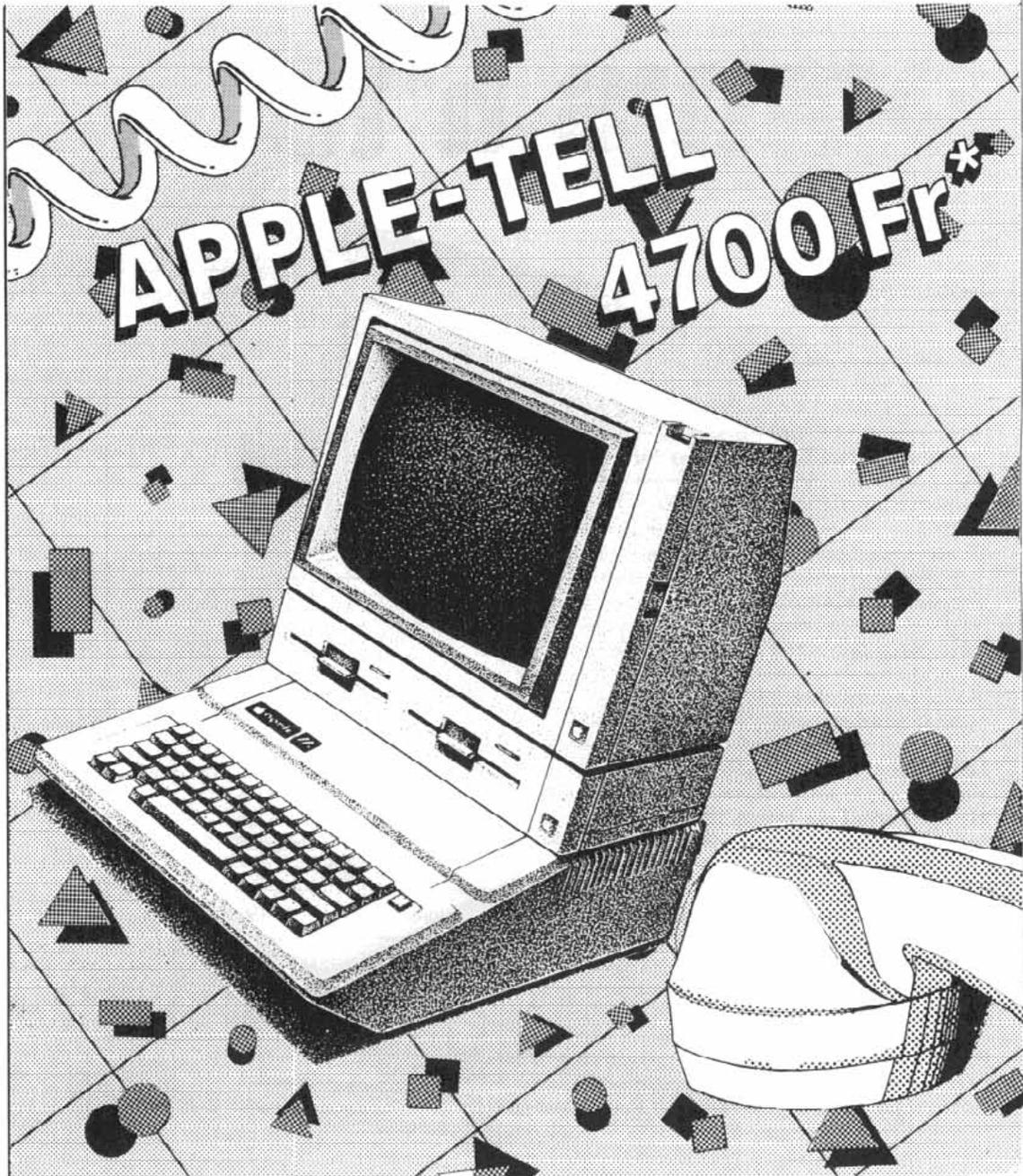
C'est pour pouvoir faire face à une demande plus de dix fois supérieure à ce qui avait été prévu qu'Hello a décidé de lancer une nouvelle fabrication de 2 000 cartes Apple-Tell. Avantage de la grande série : de très sensibles réductions de coût sur le prix des composants LSI et VLSI, permettant de faire chuter de plus de 20 % le prix de la carte Apple-Tell.

Le modem vedette de votre Apple, déjà couronné Pomme d'Or 1983, a suscité en 1984 trois nouvelles Pommes d'Or, récompensant trois des (très nombreux) logiciels que ses utilisateurs ont déjà dédiés à Apple-Tell :

- MICRO-KIDS : serveur monovoie pour les établissements d'enseignement.
- VASA : outil de composition de pages vidéotex incluant un serveur arborescent complet.
- TÉLÉBASIC : Basic télécom et vidéotex, destiné à la création de serveurs et de terminaux automatiques.

Parmi les autres logiciels créés pour Apple-Tell :

- TELEPOM : enrichissement du Basic (60 instructions nouvelles) permettant de créer soi-même des serveurs (Ascii et vidéotex), des messageries ou toutes applications télématiques.
- ASCII EXPRESS : outil général de télécom destiné à permettre l'impression et le stockage des données reçues, incluant un émulateur universel de terminaux.
- PROTEXT : éditeur-souris, permettant la composition (texte et dessins) de pages Télétel destinées à des serveurs.
- FAKIR : serveur Ascii (messagerie électronique intégrée, incluant panneau d'annonces, annuaire des abonnés, téléchargement, horloge temps réel, mois de passe, console opérateur, etc.).
- DISCOBOLE : copie de disquettes par téléphone (Dos, MemDos, Pascal, CP/M).
- TRANSTEXT : convertisseur d'écrans (Haute Résolution Télétel).
- BASI : module-système permettant la commande du modem depuis le Basic, sous ProDos.
- PROSPECTOR : générateur d'étiquettes-adresses par consultation automatique de l'annuaire électronique selon des choix socioprofessionnels et géographiques.
- CALVA-KIT : consultation



\* Prix utilisateur I.L.T. conseillé au 1<sup>er</sup> février 1985.

automatisée du centre serveur "Calvados".

→ PELAGIE : serveur spécialisé, permettant la consultation d'un Apple à partir d'un réseau de minitels. (Deux versions : Basic Dos 3.5 et MemDos.)

→ NESTOR4 : serveur à QUATRE ACCÈS SIMULTANÉS, disque dur 10 Mo, gestion automatique des touches de fonctions du minitel, arborescence et mots clés intégrés.

→ NESTORI : serveur monovoie (même caractéristiques que Nestor4).

... Les serveurs sont servis !

#### APPLE-TELL COMPREND :

- une carte Modem (avec terminal Ascii intégré) incluant un décodeur Télétel.
  - Un logiciel d'émulation de terminal Minitel enrichi des fameuses trois fonctions dont l'absence est une tragédie quotidienne pour les utilisateurs de Télétel :
- IMPRESSION (sur l'imprimante de l'Apple),

STOCKAGE sur disquettes des pages consultées (formats Télétel ou Ascii),  
 AUTOMATISME : interrogation automatique des serveurs (appel téléphonique, orientation Transpac, identification, choix successifs),  
 enregistrement des données consultées, puis traitement et incorporation des données dans l'application. (Ces procédures d'interrogation sont créées par l'utilisateur, sans aucun langage de programmation, grâce au mode APPRENTISSAGE d'Apple-Tell.)  
 Enfin et surtout, une jonction est possible avec les outils de travail habituels (Apple-Writer, Visicalc, Multiplan, PFS, Quick File, compta facturation, fichier, etc.)

#### CARACTÉRISTIQUES GÉNÉRALES :

- modem 1 200 75 (Ceit) et 300 full (standards Ceit et Bell), à numérotation automatique,
- auto-connexion, permettant la création de serveurs 300 ou 1 200 bauds,
- sorties : video composite (N & B) et Peritel couleurs,
- enfichable dans n importe quel slot libre de votre Apple 2<sup>e</sup> ou 2<sup>e</sup> (+48 K, une disquette),
- transparence totale vis-à-vis du système.

HELLO  
 Informatique  
 1, rue de Metz  
 75010 PARIS  
 Tél. : (1) 523.30.34  
 Telex : FLASH 210 500 F

Nom .....

Société .....

Adresse .....

Ville .....

Code postal ..... Tél. ....

Souhaite recevoir une documentation sur le système Apple-Tell.

Vous êtes nombreux, déjà plusieurs centaines, à avoir répondu au questionnaire publié dans le numéro 17 de Pom's. La liste des gagnants se trouve en page 55. Les lecteurs qui ne l'ont pas encore rempli peuvent toujours le faire. En effet, une revue comme la nôtre est en permanence à l'écoute de son lectorat, et ce sondage est le meilleur moyen de connaître vos besoins. Nous ne pouvons pas, bien entendu, satisfaire toutes vos requêtes, dans la mesure où certaines d'entre elles sont contradictoires, mais nous tiendrons compte de toutes les remarques. Les résultats détaillés de ce sondage paraîtront dans le numéro 19.

Nous tenons à rendre hommage à la revue Golden pour son bon goût : elle apprécie tellement Pom's qu'elle a publié dans un récent numéro, sous la plume de Joël Bernard, la transposition en assembleur du programme "Un analyseur de syntaxe" écrit par Olivier Herz et publié dans Pom's 6. En désassemblant cette contribution "originale", on retrouve l'algorithme d'Olivier, avec à peine quelques petites améliorations. Pour ne pas priver les fidèles lecteurs de Pom's, nous leurs offrirons dans le prochain numéro une version en assembleur de ce programme par Carlos Sacré ; celle-ci permet, en outre, d'analyser de façon automatique tous les fichiers Basic d'une disquette.

Puisque nous parlons de contributions, rappelons que nous demandons aux auteurs de nous envoyer sur une disquette les fichiers programme et le texte de l'article sous forme de fichier TEXT (de préférence AppleWriter ou fichier ASCII sur Macintosh, mais ce n'est pas très important). Nous souhaitons que, de façon générale, les articles suivent le plan suivant : objet du programme, mode d'emploi, explications techniques. Nous essayons de généraliser ce plan dans les articles de Pom's, afin de faciliter au lecteur non spécialiste l'utilisation des merveilleux programmes de votre revue préférée. Dans la même ligne d'idée, vous verrez avant la fin de l'année un livre regroupant les objets et modes d'emploi de tous les programmes de Pom's depuis le péché de gourmandise d'Adam.

Dans ce numéro, vous trouverez, comme à l'habitude les rubriques *Courier des lecteurs*, *Micro-informations* et *Bibliographie* que nous devons à **Olivier Herz**, **Jean-Michel Gourévitch**, et **Alexandre Duback**. Un sommaire des anciens numéros de Pom's vous est proposé, il vous permettra de trouver rapidement l'article qui vous intéresse, afin de compléter votre collection.

**Pom's** a rencontré le célèbre pirateur de logiciel Aldo Reset... **François Sermier** nous offre la possibilité de sauver, sur la même disquette, des fichiers DOS 3.3 et ProDOS, tandis que **Gérard Michel** nous propose une saisie de variables par écran.

**Olivier Herz** teste pour vous le logiciel Porte-Parole et illustre, grâce à un programme de **Régis Lardennois**, l'article sur les pointeurs en Pascal UCSD publié dans le précédent numéro.

Les fenêtres du Macintosh ont inspiré **Georges Zwingelstein** ; il a écrit un programme de gestion de fenêtres pour les Apple II.

Dans chaque numéro de Pom's (ou presque), plusieurs pages sont consacrées au graphisme ; ainsi, **Thierry Han** nous propose des "Graphiques Aléatoires" et **Marc Debuigne** une hard copy de la page HGR étendue.

**Jean-Pierre Januel** nous offre la possibilité d'utiliser le DOS de façon tout à fait transparente, tandis que **Patrice Neveu** recherche et trouve instantanément des mots ou groupe de mots dans un tableau de chaînes alphanumériques. Dans le Cahier Mac, **Marianne Sutz** explique comment récupérer les icônes du système depuis un programme Basic, tandis que **Jean-Luc Bazanegue** modifie les informations pour le Finder. Enfin, **Julien Thomas** analyse les curseurs du système.

Nous nous retrouverons enfin tous lors du grand délire d'Apple Expo, du 14 au 16 juin où, outre le stand de Pom's, vous pourrez apprécier de nombreuses manifestations. Nous recommandons tout particulièrement le débat passionnant que notre collègue Jean-Louis Courleux (France Inter) organise le dimanche après-midi, mettant face à face des auteurs de logiciels bestsellers et les plus prestigieux pirateurs de logiciel français.

Il est important de préciser que l'entrée à Apple Expo est gratuite pour les membres du Club Apple. Les lecteurs de Pom's, quant à eux, bénéficieront d'une remise de 50% sur présentation du dernier numéro.

**Hervé Thiriez**

**Ont collaboré à ce numéro** : Jean Luc Bazanegue — Alexandre Duback — Marc Debuigne — Jean-Michel Gourévitch — Thierry Han — Olivier Herz — Jean-Pierre Januel — Régis Lardennois — Gérard Michel — Patrice Neveu — François Sermier — Martanne Sutz — Hervé Thiriez — Julien Thomas — Georges Zwingelstein. **Rédacteurs** : Alexandre Aviane — Olivier Herz — Laurence Tichkowsky. **Directeur de la publication, rédacteur en chef** : Hervé Thiriez. **Dessins** : Laurent Bidot.

**Siège social** : Editions MEV — 64/70 rue des Chantiers — 78000 Versailles — Tél. : (3) 951.24.43.

**Publicité** : Consulter les Editions MEV.

**Diffusion N.M.P.P.** : Consulter les Editions MEV.

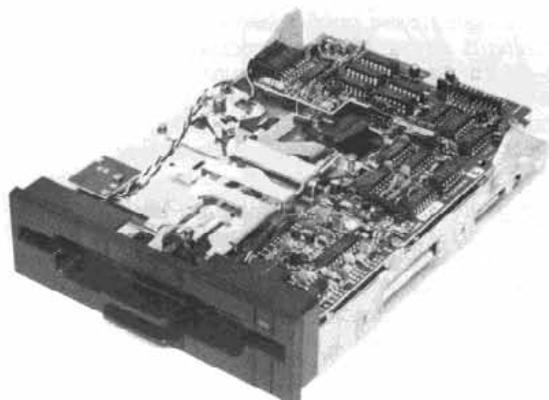
**Composition** : Télécompo — 13/15 avenue du Petit Parc — 94300 Vincennes — Tél. : 328.18.63.

**Impression** : Rosay — 47 avenue de Paris — 94300 Vincennes — Tél. : 328.18.63.

# DRIVE CHINON

**5"1/4** COMPATIBLE APPLE II+,IIe,IIc

NOUVEAU: Câble pour IIc



- 1/2 HAUTEUR
- 40 PISTES (48 T.P.I)
- 143 Ko SOUS DOS 3.3
- 160 Ko SOUS DOS MODIFIE
- DETECTION DE PISTE 0 ET PROTECTION
- ECRITURE PAR OPTO-ELECTRONIQUE
- MOTEUR A ENTRAÎNEMENT DIRECT
- SYSTEME DE FERMETURE BREVETE

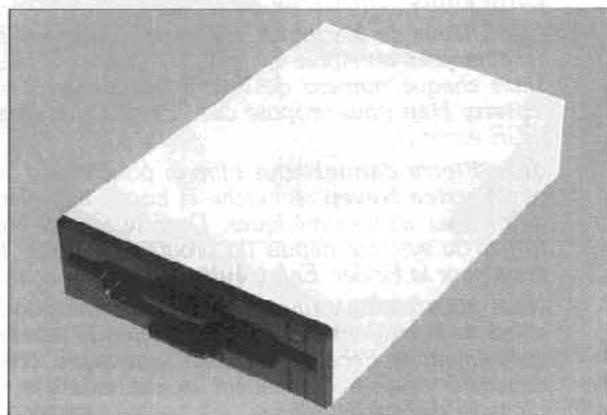
## LE MEILLEUR RAPPORT QUALITE/PRIX DU MARCHÉ

**PROMOTION**

Floppy simple 1 256 Frs H.T 1 490 Frs T.T.C  
Floppy double 2 521 Frs H.T 2 990 Frs T.T.C  
(prix par quantité sur demande)

*Ce lecteur de disquette a été homologué par la FRAMIF,  
l'Aérospatiale, le club Thomson, etc...*

- HAUTE FIABILITE
- SILENCIEUX
- FAIBLE CONSOMMATION
- GARANTIE 1 AN



**CE MODELE FAIT PARTIE D'UNE GAMME COMPLETE DE LECTEURS 5"1/4 & 3"1/2 AVEC INTERFACE SASI.**  
Autres modèles disponibles pour tous micros (nous contacter)

**PROMO**

- Carte 80 colonnes étendue 990 T.T.C
- Disquettes 5"1/4 11,90 Frs H.T par boîte de 10. (13,90 Frs T.T.C)
- Moniteurs 12" haute résolution vert ou orange 990 Frs T.T.C

MICRO-DISPO  
58, rue Blomet

Tél.: 566 57 17  
75015 PARIS

P.I.E.D  
42, bd Magenta

Tél.: 249 16 50  
75010 PARIS

NOM:..... PRENOM:..... Tél.:.....  
ADRESSE:.....  
SOCIETE:..... PROFESSION:.....

Veuillez m'envoyer une documentation sur vos produits.

Je suis intéressé à titre personnel.....  .....professionnel

# Disquette mixte ProDOS/DOS 3.3

François Sermier

## Principe

Vous venez de découvrir ProDOS, particulièrement la grande facilité de l'accès disque au niveau du bloc à partir du Basic (nous allons y revenir...), mais vous aimez bien DOS 3.3 et d'ailleurs beaucoup de vos programmes style ZAP ne sont pas compatibles ProDOS...

Ou encore, vous échangez des disquettes de programmes avec des amis et il vous faut toujours 2 disquettes différentes (ou 2 faces), une pour DOS 3.3, une pour ProDOS; cela finit par poser des problèmes (exemple : disquette Pom's).

Vous aimeriez bien avoir tout cela sur une même disquette. Et bien c'est dorénavant possible avec le jeu de programmes présentés ici.

La démarche à suivre est (relative-ment) naturelle :

- Initialiser une disquette ProDOS à l'aide du `FILER`
- Modifier à la main, à l'aide d'un utilitaire de type ZAP, le Volume Bit Map (équivalent ProDOS du VTOC 3.3), pour limiter la partie utilisée par ProDOS
- Créer à la main une piste `CATALOG 3.3` et un VTOC s'attribuant la partie non ProDOS.

Le programme `INITPRO3.3` ci-après se charge automatiquement des deux dernières étapes. Il vous demande seulement à partir de quelle piste vous désirez que commence la partie DOS 3.3 (par défaut, il coupera la disquette en 2 parties égales, au milieu de la piste 17). Il vous demande également si vous désirez raccourcir le Directory ProDOS : comme la disquette contient moins de blocs, vous pouvez ramener le Directory de 4 à 2 blocs et récupérer ainsi 1K octets de mémoire disque (cela vous laisse tout de même 25 entrées, dont d'éventuels SubDirectory au `CATALOG`, au lieu de 51). Même problème pour la partie DOS 3.3, où la taille du `CATALOG` peut être fixée de 4 secteurs minimum à 15 maximum.

**Remarque importante :** normalement, tronquer le répertoire est sans danger... mais, on peut avoir des surprises : ainsi, `CONVERT` lit d'autorité les 15 secteurs de la piste 17 sans s'occuper de leur chaînage ni s'arrêter à la première entrée vide, il signale une erreur mais on peut poursuivre et transférer tous les fichiers que l'on veut. Pire encore, la dis-

quette "Utilitaires //c" refuse de reconnaître un volume ProDOS dont le répertoire est tronqué pour l'opération Identifier / Lister Volume, mais accepte tout autre opération (copier fichier, transférer) ... bizarre ! Notez également qu'elle refuse obstinément de reconnaître la partie DOS 3.3 d'une disquette mixte; on ne peut donc pas convertir sur la même disquette avec Utilitaires //c. Par contre, cela est parfaitement possible avec `CONVERT`.

La disquette produite par `INITPRO3.3` est lisible par ProDOS et DOS 3.3. Attention, un même programme ou fichier ne peut être lu que par l'un des deux DOS; il faut toujours recourir à `CONVERT` si l'on veut le faire passer d'une moitié à l'autre. Par contre la disquette n'est pas bootable ni en ProDOS ni en DOS 3.3. Ceci est normal en ProDOS où il faut la rendre explicitement bootable en y transférant `PRODOS` et `BASIC.SYSTEM` (ou tout autre programme de type `SYSTEM`). On peut enfin simuler un boot DOS 3.3 en copiant DOS 3.3 sur la partie ProDOS et en appelant `HELLO.OBJ` qui se charge du reste.

Ceci n'est que le premier pas vers un système totalement mixte où le Basic Applesoft pourra au choix utiliser DOS ou ProDOS permettant de `CONVERTIR` très élégamment ces programmes Basic ou binaire. Mais n'anticipons pas...

## Secteurs, Blocs et MLI

L'ensemble de l'opération est réalisable car tous les systèmes d'exploitation (y compris CP/M et Pascal UCSD, voir programme `INITPAS3.3` dans le prochain numéro) utilisant le `DISK II` d'Apple s'appuient sur une même sectorisation physique de la disquette. Les différences viennent du fait que, pour diverses raisons, la sectorisation logique de la disquette n'est pas identique, c'est ce qu'on appelle l'entrelacement (interleaving). De plus en ProDOS, l'accès disque le plus élémentaire s'effectue non plus secteur par secteur mais par groupe de 2 secteurs appelé bloc. A noter que l'idée de piste n'existe plus en ProDOS.

Les deux tables suivantes permettent de voir la correspondance entre les différentes notions de sectorisation pour la piste 0.

TABLE 1

emplacement physique :  
0 1 2 3 4 5 6 7 8 9 A B C D E  
secteur logique DOS 3.3 :  
0 7 E 6 D 5 C 4 B 3 A 2 9 1 8  
bloc logique ProDOS :  
0A 4A 0B 4B 1A 5A 1B 5B 2A 6A  
2B 6B 3A 7A 3B 7B

TABLE 2

blocs ProDOS :  
0 1 2 3 4 5 6 7  
secteurs DOS :  
0-E D-C B-A 9-8 7-6 5-4 3-2 1-F

Pour les autres pistes, ces tables restent valables et relient le reste de la division par 8 du numéro de bloc et les secteurs DOS correspondants. Le numéro de piste est le quotient de la division par 8 du numéro de bloc.

Exemple :

piste 17, secteur 7 : début du bloc numéro  $8 \times 17 + 4$  soit 140  
réciproquement, bloc 100 = piste numéro  $(100/8)$  soit 12, reste 4 donc secteurs 7 et 6.

On pourrait réaliser la conversion par les routines Basic suivantes :

### BLOC --> PISTESECTEUR

```
PISTE = INT (BLOC / 8);  
RESTE = BLOC - 8 * PISTE;  
S1 = 15 - 2 * RESTE; S2 = S1 - 1  
IF S1=0 OR S1=15 THEN  
S1=15-S1
```

### PISTESECTEUR --> BLOC

```
IF S=0 OR S=15 THEN S=15-S  
RANG = 0 : IF S = 2 * INT(S/2)  
THEN RANG = 1 (ou simplement :  
RANG = (S = 2 * INT(S / 2))  
BLOC = 8 * PISTE + (7 - INT(S/2))
```

L'accès disque au niveau du bloc est extrêmement simple par l'intermédiaire de MLI (Machine Language Interface) qui regroupe un ensemble de 26 sous-routines ProDOS repérées par un code et appelées de manière standardisée. `MLICALL` montre comment utiliser cet appel MLI (voir listing `MLICALL.SCE`) : l'appel de MLI doit être suivi du code de la fonction appelée et de l'adresse de la table des paramètres de la fonction. Le retour de MLI se fait en sautant ces 3 octets. L'indicateur de retenue (Carry) est mis à 1 s'il s'est produit une erreur; le code de l'erreur est contenu dans l'accumulateur. On peut alors faire un traitement d'erreur. Cependant, il y a beaucoup d'erreurs possibles (29), aussi le plus

efficace consiste à utiliser celui qui est fourni par le Basic : JSR BADCALL (\$BE8B) analyse les 29 erreurs MLI en 19 erreurs reconnues par l'interpréteur Basic (BI) et JSR ERROUT (\$BE09) entre dans le traitement d'erreur Basic (cette méthode très simple n'est pas très documentée). La liste des paramètres dépend de la fonction appelée et est extrêmement bien documentée (voir la bibliographie).

## La structure d'un volume ProDOS

Un volume ProDOS est donc simplement une succession de blocs, normalement 280, sur une disquette 35 pistes. Cependant tous ne sont pas disponibles, en effet le début du volume est occupé par 7 blocs spéciaux que nous allons analyser.

- blocs 0-1 :  
Programme en langage machine exécuté automatiquement lors du boot. Il cherche sur la disquette un fichier du nom de PRODOS; s'il ne le trouve pas, il affiche UNABLE TO LOAD PRODOS et se plante (cas d'une disquette de données), sinon il le charge en \$2000 puis saute en \$2000. Voir la suite des aventures de ProDOS lors du boot dans Beneath Apple ProDOS par exemple.
- blocs 2-5 :  
Répertoire (directory) du volume. Chacun des 4 blocs est composé des éléments suivants :
  - 2 octets réalisant le chaînage arrière des blocs du répertoire. Ils contiennent le numéro du bloc précédent ou 0 s'il n'y a pas de précédent.
  - 2 octets réalisant le chaînage avant ... numéro du bloc suivant ou 0 si ...
  - 13 groupes de 39 octets, chaque groupe décrivant un fichier du volume sauf le tout premier des 52 qui décrit le Volume lui-même. Le bloc 2 qui contient ce groupe crucial pour l'accès au disque a reçu le nom de bloc clé.
- bloc 6 :  
Image logique de l'occupation de la disquette (Volume Bit Map). A chaque bloc du volume est associé un bit dans ce secteur; si le bloc est utilisé, le bit correspondant est à 0, s'il est libre, le bit est mis à 1. Le premier octet décrit l'utilisation des blocs 0 à 7. Pour une disquette fraîchement initialisée (pas d'autre opération), il doit valoir 1. En effet, on vient de voir que les blocs 0 à 6 sont occupés, d'où en représentant cet octet sur les 8 bits :

(voir schéma)

Remarquez que seuls les blocs 0 à 2 sont à un emplacement rigide, puisque l'accès aux autres se fait par l'intermédiaire du bloc 2 (chaînage des blocs du répertoire et pointeur sur le bloc VBM). On peut donc déplacer les autres ou même les supprimer (sauf évidemment le VBM !!!).

(voir tableau 3)

## Le programme INIT PRO3.3

Sa structure est très simple, elle est purement linéaire. On a essayé de rendre aussi lisible que possible les lectures / écritures de blocs et d'accepter les options par défaut par un simple RETURN.

*Lignes 10 à 70* : initialisation et vérification du volume. On pourrait tester si l'opération de mixage n'a pas déjà été effectuée en testant DOS=1 en ligne 80. Ce morceau appelle un sous programme en 3000 qui assure diverses tâches d'intendance : le Bip ProDOS tel qu'il est décrit par Apple (ProDOS Tech. Man.), mise en place de MLCALL (notez que les lignes 3090 à 3110 mettent en évidence la structure d'un appel MLI et permettent en modifiant les lignes 3040-3050 de placer le bout de code machine n'importe où ailleurs que dans l'archi-usée page 3...).

*Lignes 100 à 130* : lecture du bloc clé (premier bloc du Directory). On y trouve successivement :

- le nom du volume (reconstitué par le sous programme en 900),
- le pointeur vers le Volume Bit Map (VBM, normalement = 6 sauf si l'on a déjà tronqué le Directory auquel cas VBM=4),
- le nombre de blocs du volume (NB, normalement = 280 sauf si la disquette a déjà été trafiquée auquel cas la valeur de NB est imprévisible...), on en déduit le numéro de la dernière piste ProDOS (PM, normalement = 34).

Remarquez que le volume à modifier doit se trouver dans le lecteur 2, on peut le mettre ailleurs mais il faut modifier la valeur de UNIT en ligne 3040.

*Lignes 140 à 230* : vous entrez vos options. Par défaut, le roi Salomon tranche au milieu de la disquette. Si le Directory est déjà tronqué, vous n'avez pas le choix, il le restera.

*Lignes 240 à 360* : mise à jour du Volume Bit Map. Si l'on rajoute de la place ProDOS sur une disquette déjà mixte, on libère les blocs correspondants (il se peut que l'on perde des fichiers DOS 3.3)(ligne 260). En 270-280, on vérifie que la place consacrée au DOS 3.3 (c'est-à-dire

les pistes de P1 à P2 plus la piste répertoire) n'est pas déjà occupée par du ProDOS. Le sous-programme en 800 examine l'occupation de chaque piste par ProDOS; à la première qu'il trouve non vide, il vous le signale et vous laisse la possibilité de ne toucher à rien. Si vous persistez, en 840, on marque la piste comme occupée. C'est indispensable, bien sûr pour la piste 17, mais aussi pour les pistes situées dans la partie DOS, car leurs numéros de bloc dépassent la valeur maximale prévue dans le bloc clé du volume et si MLI y trouve un octet non nul (c'est-à-dire un bloc libre), il considère que le Volume Bit Map a été détérioré d'où erreur, traduite par BI en une inexplicable I/O ERROR.

Le test de la *ligne 830*, autorise à tenter l'expérience sur une disquette contenant déjà des fichiers. Il n'y a aucune difficulté si aucun bloc n'est occupé après le bloc numéro 135. Mais si des fichiers existaient sur la partie détruite, ils seront détruits physiquement sur la disquette, mais pas dans le répertoire (il faudrait pour cela analyser l'intégralité du répertoire et des listes de blocs des fichiers). De plus, si on essaie malgré tout d'y accéder, y compris pour les DELETEr, on provoquera une I/O ERROR, car le driver devra aller lire un bloc situé au-delà de son dernier bloc. Il vaut mieux éviter de poursuivre (option par défaut...). S'il faut tronquer le Directory, on libère 2 blocs en piste 0 (si ce n'est pas déjà fait... ligne 290) et on réécrit VBM à son nouvel emplacement.

*Lignes 310 à 360* : mise à jour du bloc clé, c'est-à-dire du nombre de blocs, du pointeur sur VBM; et mise à jour du chaînage entre blocs du Directory (ligne 360 si nécessaire).

*Lignes 500 à fin* : écriture du Directory 3.3, VTOC en \$11/0. Catalog de \$11/F à \$11/1 au plus. On vérifiera que, quelle que soit la valeur de NSC<4 le programme créera un minimum de 4 secteurs.

## Booter ProDOS et DOS 3.3

La disquette produite n'est qu'une disquette de données, elle ne peut booter sous aucun système.

**Booter en ProDOS** est très simple, il suffit de transférer sur la partie ProDOS les fichiers ProDOS, BASIC.SYSTEM et, si l'on veut, un programme STARTUP à l'aide du FILER. On peut réaliser cela dès l'initialisation de la disquette, puisque la transformation conserve les premiers blocs du volume. Ceci permet de diminuer le nombre de manipulations en ne recourant qu'une fois au FILER.

**Booter en DOS 3.3** est bien sûr impossible stricto sensu, mais on peut le simuler. Pour cela, il est nécessaire d'avoir un fichier ProDOS contenant l'image du DOS 3.3. Si cela est aisé à l'écriture, on conçoit aisément que cela l'est moins à la lecture. Il faut donc le transférer à un autre emplacement mémoire; d'où les opérations suivantes :

- Booter une disquette DOS 3.3 normale,
- CALL-151
- 3000<9D00.BFFFM (déplace le DOS)
- 3D0G (ou CTRL-C pour revenir en Basic)
- BSAVE DOS 3.3,A\$3000,L\$2300 (sur la disquette DOS 3.3)
- Employer l'utilitaire CONVERT pour transférer DOS 3.3 sur la disquette ProDOS
- Faire les opérations inverses sous ProDOS n'est pas possible directement (la machine se plante car on n'entre à aucun moment dans le DOS 3.3 et la page 3 (\$3D0 à \$3FF) n'est pas en place. Il faut utiliser le petit utilitaire HELLO.OBJ (à implanter sur la partie ProDOS).
- Sous ProDOS : BLOAD DOS 3.3
- Puis BRUN HELLO.OBJ et nous y sommes !

Voyons le programme HELLO.OBJ

– de \$300 à \$317 : retransfert du DOS à sa position initiale en \$9D00. Ces lignes sont équivalentes à la commande moniteur :

\*9D00<3000.5300M

– en \$318 et \$31A : on supprime le mode TRACE mis discrètement par ProDOS pour ses besoins internes

– \$31C à \$325 : on implante la chaîne de caractères HELLO dans le tampon contenant les paramètres des commandes DOS.

– \$327 et fin : on rentre à froid dans le DOS (DOS coldstart), après lui avoir fait croire que la dernière commande exécutée (en \$AA5F) par lui était INIT ce qui est son état normal lorsqu'il vient d'une disquette normalement initialisée. Ayant vérifié la chose, le démarrage à froid du DOS met en place les vecteurs de la page 3 et cherche à exécuter un programme du nom de HELLO (programme situé évidemment sur la partie DOS 3.3 de la disquette). S'il n'y a pas de HELLO, on obtient le classique FILE NOT FOUND, mais on est quand même en DOS 3.3.

On pourra trouver un exemple de boot DOS 3.3 dans le court programme (ProDOS) HELLO joint.

## Bibliographie

Pour la description des formats des secteurs (respectivement blocs) des

## Source Toolkit Programme MLI CALL.SCE

SOURCE FILE: MLI CALL.SCE

```

0000:      1      LST NOA
0000:      2 *****
0000:      3 *  Routine d'appel de MLI à partir du BASIC
0000:      4 *****
----- NEXT OBJECT FILE NAME IS MLI CALL.SCE.OBJ
0300:      5      ORG $300
0300:      6 *
0080:      7 LIT    EQU $80
0081:      8 ECRIT  EQU $81
0300:      9 *
0003:     10 LEN    EQU $3
00E0:     11 UNIT   EQU $E0      (drive 2; pour drive 1:$60)
2000:     12 BUFFER EQU $2000
0002:     13 BLOC   EQU $2
BEBB:     14 BADCALL EQU $BEBB
BE09:     15 ERROUT EQU $BE09
BF00:     16 MLI    EQU $BF00
0300:     17 *
0300:20 00 BF   18      JSR MLI
0303:80      19 CODE  DFB LIT
0304:10 03     20      DW  PARLST
0306:90 06     21      BCC OK
0308:20 8B BE  22      JSR BADCALL
0308:20 09 BE  23      JSR ERROUT
030E:60      24 OK    RTS
030F:00      25      BRK
0310:03      26 PARLST DFB LEN
0311:E0      27      DFB UNIT
0312:00 20     28      DW  BUFFER
0314:02 00     29      DW  BLOC

```

\*\*\* SUCCESSFUL ASSEMBLY: NO ERRORS

## Schéma

octet n°	0	1 et les 33 suivants
		jusqu'au n°34 inclus
bit:	0000 0001	1111 1111
bloc:	7654 3210	7654 3210
	0123 4567	89AB CDEF

## TABLE 3

TABLE 3:		Contenu d'un groupe de 39 octets.	
rang de l'octet dans le groupe		description d'un fichier	d'un volume (1er groupe bloc 2)
0		type de stockage, longueur du nom	nom de l'élément décrit
1-15			
16	type du fichier	I	
17-18	bloc clé du fichier	I	inutilisé (8 fois 0)
19-20	nombre de blocs utilisés	I	
21-23	nombre d'octets dans le fichier	I	
24-27		date-heure de création (cf. DATE ProDOS, POM'S)	
28		n° version ProDOS utilisé lors de la création	
29		n° version ProDOS minimal requis pour ce fichier	
30		accès autorisés (lecture, écriture, destruction)	
31	I adresse de chargement	I	longueur des groupes (39)
32	I ou longueur d'enregistrement	I	nbe groupes par bloc(13)
33-34	date dernière modification	I	nombre de groupes actifs
35-36	heure dernière modification	I	N° bloc UBM (normt 6)
37-38	bloc clé du répertoire	I	nombre total de blocs

## Source Toolkit Programme HELLO

```
SOURCE FILE: HELLO.SCE
0000:      1      LST NOA
0000:      2 *****
0000:      3 * coexistence de DOS3.3 et Pro-DOS
0000:      4 *****
----- NEXT OBJECT FILE NAME IS HELLO.SCE.OBJO
0300:      5      ORG $300
0300:      6 *
003C:      7 A1     EQU $3C
003E:      8 A2     EQU $3E
0042:      9 A4     EQU $42
00F2:     10 TRACE EQU $F2
0300:     11 *
9DEA:     12 DOSBOOT EQU $9DEA
AA5F:     13 CMD     EQU $AA5F
AA75:     14 NAME    EQU $AA75
FE2C:     15 MOVE    EQU $FE2C
0300:     16 *
0300:A2 00 17      LDX £0
0302:86 3C 18      STX A1
0304:86 42 19      STX A4
0306:CA    20      DEX
0307:86 3E 21      STX A2
0309:A9 30 22      LDA £$30
030B:85 3D 23      STA A1+1
030D:A9 52 24      LDA £$52
030F:85 3F 25      STA A2+1
0311:A9 9D 26      LDA £$9D
0313:85 43 27      STA A4+1
```

```
0315:20 2C FE 28      JSR MOVE
0318:      29 *
0318:A2 00 30      LDX £0
031A:86 F2 31      STX TRACE
031C:8D 2F 03 32 LOOP LDA HELLO,X
031F:9D 75 AA 33      STA NAME,X
0322:E8    34      INX
0323:C9 A0 35      CMP £$A0
0325:D0 F5 36      BNE LOOP
0327:A9 00 37      LDA £0
0329:8D 5F AA 38      STA CMD
032C:20 EA 9D 39      JSR DOSBOOT
032F:      40 *
032F:      41      MSB ON
032F:C8 C5 CC 42 HELLO ASC 'HELLO
0332:CC CF A0
```

\*\*\* SUCCESSFUL ASSEMBLY: NO ERRORS

## Programme DEBUT

```
100 D$ = CHR$(4): PRINT D$;"BLOAD DOS3.
    3"
110 ONERR GOTO 200
120 PRINT D$;"BRUN HELLO.OBJ"
200 IF PEEK(222) = 6 THEN PRINT CHR$(
    7): PRINT "Pas de programme HELL
    O !"
210 PRINT : PRINT "Vous etes en DOS 3.3"
220 NEW
```

## Programme INITPRO3.3

```
10 HOME :00 = 256:FF = 00 - 1
20 DEF FN HI(X) = INT (X / 00): DEF F
    N LO(X) = X - 00 * INT (X / 00)
30 DEF FN W(X) = PEEK (X) + 00 * PEEK
    (X + 1)
40 DEF FN PISTE(X) = BU + 56 + 4 * X
50 DIM TY$(2):TY$(0) = "DOS3.3":TY$(1) =
    "ProDOS":TY$(2) = "Pascal"
60 GOSUB 3000: REM Initialisation
70 IF TYPE < > 1 THEN PRINT : PRINT "C
    e n'est pas une disquette ProDOS":
    CALL BELL: GOSUB 3120: GOTO 70
```

```
100 REM ----- Directory ProD
    OS -----
102 REM
104 REM ----- Lecture Key-Blo
    ck
110 POKE CODE,LIT: POKE BLOC,2: CALL MLI
120 R = BU + 4: GOSUB 900:VOL$ = N$ + "/"
130 VBM = FN W(BU + 39):NB = FN W(BU +
    41):PM = INT ((NB - 1) / 8)
140 PRINT : PRINT "Vous allez transforme
    r ";VOL$
150 PRINT " en mixte ProDOS-DOS 3.3"
160 PRINT : PRINT "1 )re piste DOS 3.3 e
    n n[ : 17"; HTAB 27: INPUT R$:P1 =
    VAL (R$): IF R$ = "" THEN P1 = 1
    7
```

```

170 VTAB PEEK (37): HTAB 28: PRINT P1
180 IF P1 < 0 OR P1 > 34 THEN CALL BELL
    : PRINT "nI piste hors limite !":
    GOTO 160
190 P2 = 34: IF PM > P2 THEN P2 = PM
200 IF VBM < 6 THEN TR = 1: GOTO 220
210 PRINT : PRINT "Tronquer le Directory
    ProDOS ?N":; HTAB 31: GET R$: PRI
    NT R$: IF R$ = "0" THEN TR = 1
220 PRINT : PRINT "Nbe Secteurs Director
    y DOS 3.3: 15":; HTAB 32: INPUT R$
    :NSC = VAL (R$): IF R$ = "" THEN
    NSC = 15
230 VTAB PEEK (37): HTAB 33: PRINT NSC
239 REM
240 REM ----- Mise 2 Jour Vol
    ume-Bit-Map
250 POKE BLOC,VBM: CALL MLI
260 IF PM < P1 THEN FOR P = PM TO P1 -
    1: POKE BU + P,FF: NEXT
270 FOR P = P1 TO P2: GOSUB 800: NEXT
280 P = 17: GOSUB 800
290 IF TR THEN POKE BU, PEEK (BU) + 6 *
    (VBM = 6):VBM = 4
300 POKE BLOC,VBM: POKE CODE,ECRIT: CALL
    MLI
309 REM
310 REM ----- Mise 2 Jour Key
    -Block
320 POKE CODE,LIT: POKE BLOC,2: CALL MLI
330 POKE BU + 39,VBM
340 NB = 8 * P1: POKE BU + 41, FN LO(NB):
    POKE BU + 42, FN HI(NB)
350 POKE CODE,ECRIT: CALL MLI
360 :: IF TR THEN POKE BLOC, FN W(BU + 2
    ): POKE CODE,LIT: CALL MLI: POKE B
    U + 2,0: POKE BU + 3,0: POKE CODE,
    ECRIT: CALL MLI
500 REM ----- Directory DOS 3
    .3 -----
502 REM
504 REM ----- Ecriture secte
    urs Directory
510 FOR I = 0 TO 511: POKE BU + I,0: NEX
    T
520 POKE CODE,ECRIT
530 POKE BU + 257,17: POKE BU + 258,14:
    POKE BLOC,143: CALL MLI
540 IMIN = 1:OC = 0:F = 0
550 IF NS < 15 THEN IMIN = 7 - INT ((NS
    - 1) / 2):OC = 16382:F = 12288
560 FOR I = 6 TO IMIN STEP - 1
570 ::OC = OC - F:F = F / 4
580 :::FIN = (I < = IMIN) AND (NSC < 15)
590 :: POKE BU + 1,17: POKE BU + 257,17 *
    NOT FIN
600 :: POKE BU + 2,2 * I: POKE BU + 258,(
    2 * I - 1) * NOT FIN
610 :: POKE BLOC,143 - I: CALL MLI
620 NEXT
630 REM ----- Cr(ation VTOC
640 POKE BU + 2,15: POKE BU + 257,17: PO
    KE BU + 258,13
650 POKE BU + 3,3: POKE BU + 6,FF - 1
660 POKE BU + 39,122: POKE BU + 48,16: P
    OKE BU + 49,1

```

```

670 POKE BU + 52,P2 + 1: POKE BU + 53,16
    : POKE BU + 55,1
680 FOR I = P1 TO P2
690 :: POKE FN PISTE(I),FF: POKE FN PIS
    TE(I) + 1,FF
700 NEXT
710 POKE FN PISTE(17), FN HI(OC): POKE
    FN PISTE(17) + 1, FN LO(OC)
720 POKE BLOC,136: CALL MLI
730 END
799 REM ----- Bar de la derni}re chance
    ...
800 IF ( PEEK (BU + P) = FF) OR OK THEN
    GOTO 840
810 :: PRINT : PRINT : INVERSE : PRINT "A
    TTENTION":; NORMAL : CALL BELL: PR
    INT " Vous risquez de d}truire de
    s donn}es de ";VOL$: PRINT
820 :: PRINT "Voulez-vous continuer ?N":;
    HTAB 24: GET R$: PRINT R$
830 ::OK = 1: IF R$ < > "0" THEN END
840 POKE BU + P,0
850 RETURN
899 REM ----- champ Nom
900 L = PEEK (R):ST = INT (L / 16):L =
    L - 16 * ST
910 N$ = "": IF ST = 15 THEN N$ = "/"
920 FOR J = 1 TO L:N$ = N$ + CHR$ ( PEE
    K (R + J)): NEXT
930 RETURN
3000 REM ----- Bip-Bip ProDOS ! --
-----
3010 BELL = 816: FOR I = 0 TO 21: READ X:
    POKE BELL + I,X: NEXT
3020 DATA 162,32,169,2,32,168,252,141,
    48,192,169,36,32,168,252,141,48,19
    2,202,208,237,96
3030 REM ----- Initialisati
    on Appel MLI -----
3040 MLI = 768:BUFFER = 8192:UNIT = 224:
    REM Drive 1:96 --- Drive 2:224
3050 PA = 784:BLOC = PA + 4
3060 CODE = MLI + 3:LIT = 128:ECRIT = 129
3070 FOR I = 0 TO 21: READ X: POKE MLI +
    I,X: NEXT
3080 DATA 32,0,191,128,16,3,144,6,32,139
    ,190,32,9,190,96,0,3,224,0,32,2,0
3090 POKE MLI + 4, FN LO(PA): POKE MLI +
    5, FN HI(PA)
3100 POKE PA,3: POKE PA + 1,UNIT: POKE P
    A + 5,0
3110 POKE PA + 2, FN LO(BUFFER): POKE PA
    + 3, FN HI(BUFFER)
3120 REM ----- Identification Dis
    quette -----
3130 PRINT "Disquette --> Drive 2. OK?":
    : GET R$: PRINT " OK"
3140 POKE BLOC,136: CALL MLI
3150 DOS = ( PEEK (BU + 3) = 3)
3160 POKE BLOC,2: CALL MLI
3170 TYPE = ( PEEK (BU + 2) AND NOT PEE
    K (BU + 3)) * ( NOT PEEK (BU + 4)
    + 1)
3180 RETURN

```

# Saisie de variables par écran

Gérard Michel

La saisie de données reste souvent un sujet de préoccupation pour le programmeur, notamment en raison des insuffisances de l'instruction INPUT du Basic. Pom's vous a d'ailleurs déjà proposé, sous des formes plus ou moins complètes et complexes, de nombreuses routines de saisie-clavier.

Ces différentes routines, en Applesoft ou en assembleur, reposaient généralement sur un même principe de fonctionnement : analyse et contrôle des caractères entrés au clavier, puis affectation au contenu d'une variable à l'issue d'une session de frappe, marquée, par exemple, par un RETURN.

Elles pouvaient également comporter une gestion de la position du curseur à l'écran, en fonction des variables à saisir, au moyen de HTAB et VTAB, ou de traitements équivalents en langage machine.

La solution présentée ici procède d'une autre démarche, articulée en trois étapes :

- Saisie de caractères au clavier, avec rejet des éventuels "parasites", caractères de contrôle en particulier.
- Affichage pur et simple des caractères valides à l'écran.
- A l'issue de la saisie, détectée lors de la frappe de la touche ESC, analyse du contenu de l'écran pour y reconstituer les différentes données et les affecter aux variables.

Ce mode de saisie peut présenter certains avantages, tant pour le programmeur (pas de gestion complexe d'écran à définir et à réviser en fonction des "erreurs et omissions"), que pour l'utilisateur (saisie, sous réserve du respect de certaines règles, en n'importe quel endroit libre de l'écran, possibilité de se déplacer en permanence sur tout l'écran pour y modifier des données, sans risque pour les variables...).

## Principes d'utilisation

Il vous faudra tout d'abord définir les libellés devant apparaître en regard des données à fournir. A chaque variable doit correspondre obligatoirement un libellé, aussi bref soit-il, mais il peut y avoir en revanche plus de libellés que de variables (pour un titre d'écran, par exemple).

Ces libellés seront ensuite affichés sur un écran vierge par tous moyens à votre convenance, tels les simples ordres PRINT utilisés dans notre programme d'illustration.

```

1 *****
2 *
3 * SAISIE ECRAN APPLE //E-C *
4 * (BIG MAC) *
5 * *
6 *****
7
8
9 ORG $9000
10 CMP $5D ;CARACTERE "5" ?
11 BEQ S0
12 CMP $3A ;TRAITEMENT CHRGET STANDARD
13 BCC S1
14 RTS
15 S1 JMP $8E
16 S0 LDA $0 ;RECHERCHE DES LIBELLES A L'ECRAN
17 STA $25
18 S4 JSR $FBC1 ;CALCUL ADRESSE BASE DE LA LIGNE
19 LDY $0
20 S3 LDA ($28),Y
21 CMP $A0 ;"ESPACE" ?
22 BEQ S2 ;OUI --> PAS DE CHANGEMENT
23 CMP $DF ;"_" ?
24 BEQ S2
25 CMP $40 ;CARACTERE DEJA EN INVERSE ?
26 BCC S2 ;OUI --> PAS DE CHANGEMENT
27 AND $3F ;TRANSFORME NORMAL EN INVERSE
28 STA ($28),Y
29 S2 INY
30 CPY $28 ;FIN DE LIGNE ?
31 BNE S3
32 INC $25 ;LIGNE SUIVANTE
33 LDA $25
34 CMP $18 ;SI L'ECRAN N'EST PAS TERMINE
35 BNE S4
36 JSR $B1 ;OCTET SUIVANT $ DANS LE PROGRAMME
37 CMP $'S'
38 BEQ S5
39 CMP $'D'
40 BEQ S5
41 S6 JSR $F399
42 JMP $DEC9 ;"SYNTAX ERROR"
43 S5 STA $6 ;SAUVEGARDE OCTET COMMANDE
44 JSR $B1 ;OCTET SUIVANT
45 JSR $F7D9 ;RECHERCHE DU TABLEAU DE VARIABLES
46 LDA $9B
47 STA $7 ;SAUVEGARDE DE SON ADRESSE
48 LDA $9C
49 STA $8
50 LDY $0
51 LDA ($7),Y ;1ERE LETTRE DU NOM DU TABLEAU
52 BMI S6 ;DOIT ETRE EN ASCII POSITIF
53 INY
54 LDA ($7),Y ;2EME LETTRE
55 BPL S6 ;DOIT ETRE EN ASCII NEGATIF
56 LDY $4
57 LDA ($7),Y
58 CMP $1 ;TABLEAU A 1 SEULE DIMENSION
59 BNE S6 ;SINON --> ERREUR
60 INY

```

Source Big Mac

Programme IPE2.S

Concernant chaque variable, il faudra en outre en préciser :

- la longueur (si cette dernière n'est pas pertinente pour vos traitements, fixez-la à 255 caractères);
- le type : 0 pour des données alphanumériques, 1 pour des données numériques.

Les longueurs seront stockées dans un tableau d'entiers à une seule dimension (baptisons-le TL%), et les types dans un autre tableau de même nature (soit TT%, par exemple). Quant aux données elles-mêmes, elles seront affectées dans un tableau de chaînes de caractères (TV\$, par exemple).

Ces trois tableaux doivent avoir un nombre d'éléments identique et il est préférable de les dimensionner en conséquence avant toute opération de saisie.

Une fois les différents tableaux définis, les longueurs et types chargés (par l'intermédiaire de DATAs ou par tout autre moyen) et les libellés affichés, vous disposez de deux instructions pour réaliser la saisie :

- :JSTV\$.TL%,TT%
- :JDTV\$.TL%,TT%

L'instruction :JS... permet une saisie pure et simple : le tableau de variables est considéré comme "vide" et il faut entrer autant de données qu'il y a d'indices dans ce tableau.

L'instruction :JD... en revanche, affiche d'abord le contenu du tableau avant de passer à la phase de saisie. Seuls les éléments à modifier nécessitent alors une nouvelle frappe. Cette instruction peut d'ailleurs servir dans un but de consultation des données, sans altération de leurs valeurs.

Le caractère "]" constitue l'identificateur de nos ordres de saisie; il est obligatoire et doit toujours être précédé de ":", même s'il vient en tout début de ligne ou après un THEN.

### Fonctionnement de la saisie

Dès qu'elle rencontre un caractère "]", la routine commence par explorer l'écran pour y trouver les libellés et les transformer de NORMAL en INVERSE. Les données étant ensuite affichées en NORMAL, cette différence de mode vidéo permettra de les distinguer des libellés.

Il est préférable de ne pas mettre d'espaces à l'intérieur des libellés et de les remplacer, par exemple, par des points.

Si vous le souhaitez, vous pouvez indiquer à l'écran la longueur maximale autorisée pour une donnée en affichant, à la suite du libellé correspondant, un nombre égal de caractères "souligné" (code clavier #DF). Ces caractères seront traités comme des espaces par la routine et resteront en mode NORMAL.

61	LDA (\$7),Y	
62	STA \$1A	;SAUVEGARDE NOMBRE D'ELEMENTS
63	INY	
64	LDA (\$7),Y	
65	STA \$19	;DANS LA DIMENSION
66	CLC	
67	LDA \$7	
68	ADC \$7	
69	STA \$7	;\$7-\$8 POINTE SUR LE 1ER
70	BCC \$8	
71	INC \$8	;ELEMENT DU TABLEAU
72 S8	LDY \$0	
73	LDA (\$88),Y	;TESTE ", " AVANT LE
74	CMP \$','	;DEUXIEME TABLEAU
75	BNE \$6	;SINON --> ERREUR
76	JSR K00	
77	JMP K2	
78 K00	JSR \$B1	;TRAITEMENT DES TABLEAUX
79	JSR \$F7D9	;DE "LONGUEURS" ET "TYPES"
80	LDY \$0	
81	LDA (\$9B),Y	
82	BMI K0	;TABLEAU ENTIER --> LES DEUX
83 K1	PLA	;LETTRES DOIVENT ETRE EN ASCII NEGATIF
84	PLA	
85	JMP \$6	;SINON --> ERREUR
86 K0	INY	
87	LDA (\$9B),Y	
88	BPL K1	
89	LDY \$4	
90	LDA (\$9B),Y	
91	CMP \$1	;TABLEAU A 1 DIMENSION SEULEMENT
92	BNE K1	
93	INY	
94	LDA (\$9B),Y	
95	CMP \$1A	;LE NOMBRE D'ELEMENTS DOIT ETRE
96	BNE K1	;LE MEME QUE POUR LE TABLEAU
97	INY	;DE VARIABLES A SAISIS
98	LDA (\$9B),Y	
99	CMP \$19	
100	BNE K1	
101	RTS	
102 K2	LDA \$9B	
103	CLC	
104	ADC \$8	;\$EB-\$EC POINTERA SUR LE POIDS FAIBLE
105	STA \$EB	;DE LA VALEUR DU PREMIER ELEMENT
106	LDA \$9C	;DU TABLEAU DES LONGUEURS
107	ADC \$0	
108	STA \$EC	
109	LDY \$0	
110	LDA (\$88),Y	
111	CMP \$','	;TESTE ", " AVANT LE TABLEAU DES "TYPES"
112	BEQ K3	
113	JMP \$6	
114 K3	JSR K00	;MEME TRAITEMENT QUE POUR LES LONGUEURS
115	LDA \$9B	
116	CLC	
117	ADC \$8	;\$ED-\$EE POINTERA SUR LE POIDS FAIBLE
118	STA \$ED	;DE LA PREMIERE VALEUR
119	LDA \$9C	;DU TABLEAU DES TYPES
120	ADC \$0	
121	STA \$EE	
122	LDA \$6	

Après inversion des libellés, la main est donnée à l'utilisateur pour la saisie proprement dite. Le curseur est initialement placé en haut à gauche de l'écran; il est possible de le déplacer en tout point avec les quatre flèches de direction.

Au niveau des caractères entrés, la routine procède aux contrôles suivants :

- Pas d'affichage autorisé sur les libellés : seules les parties d'écran en NORMAL (vierges ou non) sont admises pour la saisie / affichage.

- Pas de caractères de contrôle, ni de ":", ";", " " ou guillemets, les virgules étant pour leur part remplacées par des points (problème des caractères séparateurs et des guillemets avec les ordres PRINT).

Chaque donnée doit se terminer par un RETURN. La frappe de cette touche se traduit à l'écran par l'affichage d'une barre oblique (dans le sens inverse de "/" - code clavier # \$DC). En clavier QWERTY, vous pourriez utiliser la touche correspondant à cette barre oblique directement, en lieu et place de RETURN.

Il s'agit là d'un simple séparateur de données : il s'affiche mais ne provoque aucun autre déplacement du curseur que celui occasionné par les autres touches.

Il n'est nullement indispensable de saisir les valeurs exactement en regard des libellés qui leur sont liés, sauf si vous avez besoin de vous placer sur les éventuels "soulignés" pour être sûr de ne pas dépasser les longueurs prévues.

Ainsi, si vous disposez "quelque part" d'un nombre suffisant de positions libres contiguës à l'écran, vous pouvez saisir toutes les données requises en continu (à la suite les unes des autres), à condition de terminer chacune d'elles par un RETURN et de bien taper autant de RETURN que de données attendues (nombre d'éléments dans le tableau des variables). A noter que, comme à l'habitude, une donnée "vide" se traduit par la frappe d'un simple RETURN. Par exemple :

D1 puis RETURN et RETURN

donnera "D1" comme valeur pour le premier élément du tableau et "" pour le second (chaîne vide).

En résumé, dès lors que vous respectez l'ordre des données (entrée dans l'ordre des indices croissants du tableau) et que vous terminez chacune par un RETURN, vous pouvez les saisir en n'importe quel endroit autorisé de l'écran, individuellement, en un seul bloc, ou en sous-ensembles continus.

La frappe de la touche ESC marque la fin de la phase de saisie et déclenche l'analyse de l'écran par la routine, qui comporte les contrôles suivants :

123	CMP	£'S'	;REPRISE DU TRAITEMENT DES COMMANDES
124	BEQ	S7	
125	JMP	SD	
126 S7	LDA	£0	
127	TAY		
128	STA	\$24	
129	STA	\$25	
130	JSR	\$FBC1	;ON COMMENCE EN HAUT DE L'ECRAN
131 A1	LDA	(\$28),Y	
132	STA	\$9	;SAUVEGARDE DE L'OCTET SOUS LE CURSEUR
133 AA1	JSR	\$FDC0	;SAISIE AU CLAVIER
134	LDY	\$24	;Y=POSITION HORIZONTALE
135	LDX	\$25	;X=POSITION VERTICALE
136	CMP	£\$88	;FLECHE VERS LE HAUT ?
137	BNE	A0	
138	CPX	£0	;VERIFIER SI L'ON PEUT "REMONTRE"
139	BEQ	A1	
140 A10	DEC	\$25	
141 A3	JSR	\$FC22	; VTAB
142	JMP	A1	
143 A0	CMP	£\$8A	;FLECHE VERS LE BAS ?
144	BNE	A2	
145	CPX	£\$17	;VERIFIER SI ON PEUT "DESCENDRE"
146	BEQ	A1	
147 A6	INC	\$25	
148	BNE	A3	
149 A2	CMP	£\$95	;FLECHE VERS LA DROITE ?
150	BNE	A4	
151 AA2	CPY	£\$27	;PEUT-ON ALLER A DROITE ?
152	BNE	A5	
153	CPX	£\$17	
154 A9	BEQ	A1	
155	LDY	£0	
156	STY	\$24	
157	BEQ	A6	
158 A5	INC	\$24	
159	INY		
160	JMP	A1	
161 A4	CMP	£\$88	;FLECHE VERS LA GAUCHE ?
162	BNE	A7	
163	CPY	£0	;PEUT-ON ALLER A GAUCHE ?
164	BNE	A8	
165	CPX	£0	
166	BEQ	A9	
167	LDY	£\$27	
168	STY	\$24	
169	BNE	A10	
170 A8	DEC	\$24	
171	DEY		
172	JMP	A1	
173 A7	CMP	£\$9B	; "ESC" ?
174	BEQ	E0	; OUI --> SORTIE POUR ANALYSE DE L'ECRAN
175	PHA		
176	LDA	\$9	
177	CMP	£\$40	
178	BCS	A11	
179	PLA		
180	JMP	AA1	; PAS D'ECRITURE SUR LES LIBELLES EN INVERSE
181 A11	PLA		
182	CMP	£\$8D	
183	BNE	A12	
184	LDA	£\$DC	; SI "RETURN" --> AFFICHER "c" (£\$DC)

- Vérification de la longueur des données par rapport aux longueurs maximales déclarées dans le tableau adéquat; si une donnée "déborde", un crochet en FLASH est affiché à sa suite (il écrase le RETURN) et la routine reprend une phase de saisie pour correction.

A noter que tous les espaces compris entre les caractères d'une même donnée comptent dans sa longueur, alors que ceux qui se trouvent à l'extérieur (entre un RETURN et le premier octet de la valeur suivante, par exemple), sont ignorés.

- Pour les données de type numérique, les caractères autorisés sont: les chiffres, ".", "-", "+" et "/". Aucun espace, ni "souligné", n'est autorisé à l'intérieur d'une valeur numérique. Si on trouve un caractère non conforme, il est écrasé par une "\*" en FLASH et la saisie reprend.

- Jamais de données comptant plus de 255 caractères; dans le cas contraire, un "J" en FLASH est affiché à la fin de la donnée trop longue et la saisie reprend dans les mêmes conditions que ci-dessus.

- Exactement autant de RETURNS que de variables attendues, ni plus, ni moins. Sinon, reprise de la saisie, accompagnée d'un beep.

Si tous ces contrôles sont satisfaits, la routine procède directement à l'affectation des chaînes de caractères au tableau de variables défini en entrée, à partir de l'élément d'indice 0. Les données sont alors utilisables par toutes instructions Basic ou DOS.

### Particularités du pré-affichage

Lors d'une instruction :JD T\$,... , la routine affiche le contenu des variables de T\$ avant de passer à la saisie. Pour ce faire, elle cherche à localiser des libellés en partant de la fin de l'écran (en bas à droite) et s'arrête dès qu'elle atteint le début de l'écran ou qu'elle trouve autant de libellés que de données à afficher.

De ce fait, il convient de placer les autres textes éventuels en haut d'écran et de n'afficher ensuite que des libellés "utiles".

Les données sont restituées dans l'ordre croissant des indices, en regard et à la suite des libellés correspondants. Le RETURN final est automatiquement ajouté à la fin des caractères.

Si vous n'avez pas mis assez de textes à l'écran, vous obtiendrez une SYNTAX ERROR lorsque la routine tentera d'afficher les valeurs.

La saisie est identique pour :JS et :JD. Pour cette dernière, toutefois, il peut être bon d'entrer les nouvelles valeurs "sur" les anciennes, en les écrasant, plutôt que de les mettre dans une partie vierge de l'écran, et ce afin de ne pas laisser à l'affichage plus de données qu'il n'en faut.

185	AA11	STA (\$28),Y	;AFFICHE LE CARACTERE SAISI
186		JMP AA2	
187	A12	CMP E\$A0	
188		BCS A13	
189	AA12	JMP AA1	;PAS DE CARACTERES DE CONTROLE
190	A13	CMP E\$A2	
191		BEQ AA12	;PAS DE "
192		CMP E\$BA	;PAS DE :
193		BEQ AA12	
194		CMP E\$BB	;PAS DE ;
195		BEQ AA12	
196		CMP E\$AC	
197		BNE AA11	
198		LDA E\$AE	;REPLACE LES VIRGULES PAR DES POINTS
199		BNE AA11	
200	E0	LDA \$7	;ANALYSE DE L'ECRAN
201		STA \$1D	;RECOPIE DE TOUS LES PARAMETRES
202		LDA \$8	;DANS DES ZONES DE TRAVAIL
203		STA \$1E	
204		LDA \$EB	
205		STA \$F9	
206		LDA \$EC	
207		STA \$FA	
208		LDA \$ED	
209		STA \$FB	
210		LDA \$EE	
211		STA \$FC	
212		LDY E0	
213		LDA (\$FB),Y	;MEMORISE LE TYPE DE LA PREMIERE DONNEE
214		STA \$FF	
215		LDA E0	
216		STA \$1B	;NOMBRE DE DONNEES TROUVEES SUR L'ECRAN
217		STA \$1C	
218	E00	LDA E\$A0	;CONTROLERA L'USAGE DES ESPACES
219		STA \$18	
220		LDA E\$DF	;CONTROLERA L'USAGE DES " _ "
221		STA \$FE	
222		LDX E0	
223		TXA	
224		STA \$25	
225	E4	JSR \$FBC1	
226		LDY E0	
227	E3	LDA (\$28),Y	
228		CMP E\$40	
229		BCC E1	;LES CARACTERES INVERSE NE SONT PAS DES DONNEES
230		CMP E\$0C	; "RETURN" A L'ECRAN ?
231		BEQ E2	
232		CPX E\$FF	;255 CARACTERES DEJA LUS POUR UNE MEME DONNEE ?
233		BEQ E2	
234		CMP E\$A0	; "ESPACE" ?
235		BEQ B0	
236		CMP E\$DF	; " _ " ?
237		BEQ B01	
238		STA \$18	
239		STA \$FE	
240		LDA \$FF	
241		BEQ E01	;TYPE ALPHANUMERIQUE --> OK
242		LDA \$18	
243		CMP E\$AB	;ACCEPTTE LE +
244		BEQ E01	
245		CMP E\$AD	
246		BCS E010	

Par ailleurs, si vous souhaitez inciter à une saisie initiale en regard des libellés et non en continu, vous pouvez initialiser votre tableau de variables en ne lui affectant que des chaînes vides et utiliser une instruction :JD plutôt que :JS. Le pré-affichage de tous les RETURNS juste après les libellés correspondants constituera une contrainte de positionnement pour l'utilisateur.

## Mode d'emploi

Il convient de placer les instructions suivantes en tête de votre programme Basic d'exploitation :

- BLOAD IPE2 : chargement de la routine de saisie en \$9000.
- BLOAD INTER : mise en place dans CHRGET (\$B1) d'un aiguillage sur notre routine IPE2, afin de détecter les "]" d'identification éventuels dans le programme Applesoft. Ce second BLOAD ne doit jamais venir avant celui de IPE2, sous peine de résultats imprévisibles...

Pour revenir au Basic standard, il suffit de charger par un BLOAD le petit fichier baptisé CHRGET.

Telle qu'elle vous est présentée ici, cette routine de saisie d'écran est susceptible de nombreux perfectionnements. On peut songer, par exemple, à un alignement automatique sur le libellé suivant lors de la frappe d'un RETURN, ou à une adaptation pour carte 80 colonnes, à l'emploi de touches de fonction pour détruire ou insérer des caractères... Si le sujet vous intéresse, à vous de jouer...

## INTER

\*BA.BD

00BA- 4C 00 90

## CHRGET

\*BA.BE

00BA- C9 3A B0 0A

## IPE2

\*9000.93D5

9000- C9 50 F0 08 C9 3A 90 01  
 9008- 60 4C BE 00 A9 00 85 25  
 9010- 20 C1 FB A0 00 B1 28 C9  
 9018- A0 F0 0C C9 DF F0 08 C9  
 9020- 40 90 04 29 3F 91 28 C8  
 9028- C0 28 D0 E9 E6 25 A5 25  
 9030- C9 18 D0 DC 20 B1 00 C9  
 9038- 53 F0 0A C9 44 F0 06 20  
 9040- 99 F3 4C C9 DE 85 06 20  
 9048- B1 00 20 D9 F7 A5 9B 85

247 E011	LDA	£\$6A	;RIEN EN DESSOUS DU "--"
248	STA	(\$28),Y	;ERREUR DE TYPE --> AFFICHE UNE * EN FLASH
249	JSR	\$FBDD	;BEEP !
250	JMP	S7	;REPREND LA SAISIE AU DEBUT
251 E010	CMP	£\$BA	
252	BCS	E011	;AU DESSUS DE ":" --> ERREUR DE TYPE
253 E01	LDA	\$18	
254	BNE	B1	
255 B0	CMP	\$18	
256	BEQ	E1	;ON NE PREND PAS LES "ESPACES" HORS DES DONNEES
257	LDA	\$FF	
258	BNE	E011	;PAS D'ESPACE A L'INTERIEUR DES NOMBRES
259	LDA	£\$A0	
260	BNE	B1	
261 B01	CMP	\$FE	;ON NE PREND PAS LES "_" HORS DES DONNEES
262	BEQ	E1	
263	LDA	\$FF	
264	BNE	E011	;PAS DE "_" DANS LES NOMBRES
265	LDA	£\$DF	
266 B1	STA	\$200,X	;STOCKE LE CARACTERE DANS LE BUFFER DE DONNEES
267	INX		
268 E1	INY		
269	CPY	£\$28	;PASSAGE AU CARACTERE SUIVANT SUR L'ECRAN
270	BNE	E3	
271	INC	\$25	
272	LDA	\$25	
273	CMP	£\$18	
274	BNE	E4	
275	JMP	C0	;L'ANALYSE DE L'ECRAN EST TERMINEE...
276 E2	LDA	\$1C	
277	CMP	\$1A	; \$1B-\$1C DOIT TOUJOURS ETRE INFERIEUR
278	BCC	E5	;A \$19-\$1A PUISQUE L'ON COMPTE LES
279	LDA	\$18	;DONNEES A L'ECRAN A PARTIR DE 0
280	CMP	\$19	
281	BCC	E5	
282 E11	JSR	\$FBDD	;PLUS DE DONNEES QUE DE VARIABLES --> ERREUR
283	JMP	S7	
284 E5	CPX	£\$FF	
285	BNE	E6	
286 E50	LDA	£\$50	;DONNEE TROP LONGUE --> AFFICHE UN \$ EN FLASH
287	STA	(\$28),Y	
288	JSR	\$FBDD	
289	JMP	S7	;ET REPREND LA SAISIE
290 E6	STY	\$1F	;SAUVEGARDE LA VALEUR DE Y
291	LDY	£0	
292	TXA		
293	CMP	(\$F9),Y	;CONTROLE "LONGUEUR" LUE PAR RAPPORT
294	BCC	E60	;A "LONGUEUR MAXI" POUR LA DONNEE
295	BEQ	E60	
296	LDY	\$1F	
297	BCS	E50	;SIGNALA LE DEPASSEMENT ET REPREND LA SAISIE
298 E60	PHA		
299	CLC		
300	LDA	\$F9	
301	ADC	£2	;PASSE A LA LONGUEUR DE LA DONNEE SUIVANTE
302	STA	\$F9	
303	BCC	E61	
304	INC	\$FA	
305 E61	CLC		
306	LDA	\$FB	;PASSE AU TYPE DE LA DONNEE SUIVANTE
307	ADC	£2	
308	STA	\$FB	

9050- 07 A5 9C 85 08 A0 00 B1  
 9058- 07 30 E4 C8 B1 07 10 DF  
 9060- A0 04 B1 07 C9 01 D0 D7  
 9068- C8 B1 07 85 1A C8 B1 07  
 9070- 85 19 18 A5 07 69 07 85  
 9078- 07 90 02 E6 08 A0 00 B1  
 9080- B8 C9 2C D0 BA 20 88 90  
 9088- 4C B8 90 20 B1 00 20 D9  
 9090- F7 A0 00 B1 9B 30 05 68  
 9098- 68 4C 3F 90 C8 B1 9B 10  
 90A0- F6 A0 04 B1 9B C9 01 D0  
 90A8- EE C8 B1 9B C5 1A D0 E7  
 90B0- C8 B1 9B C5 19 D0 E0 60  
 90B8- A5 9B 18 69 08 85 EB A5  
 90C0- 9C 69 00 85 EC A0 00 B1  
 90C8- B8 C9 2C F0 03 4C 3F 90  
 90D0- 20 8B 90 A5 9B 18 69 08  
 90D8- 85 ED A5 9C 69 00 85 EE  
 90E0- A5 06 C9 53 F0 03 4C C5  
 90E8- 92 A9 00 A8 85 24 85 25  
 90F0- 20 C1 FB B1 28 85 09 20  
 90F8- 0C FD A4 24 A6 25 C9 8B  
 9100- D0 0C E0 00 F0 ED C6 25  
 9108- 20 22 FC 4C F3 90 C9 8A  
 9110- D0 08 E0 17 F0 D0 E6 25  
 9118- D0 EE C9 95 D0 14 C0 27  
 9120- D0 0A E0 17 F0 CD A0 00  
 9128- 84 24 F0 EA E6 24 C8 4C  
 9130- F3 90 C9 88 D0 14 C0 00  
 9138- D0 0A E0 00 F0 E6 A0 27  
 9140- 84 24 D0 C2 C6 24 88 4C  
 9148- F3 90 C9 9B F0 32 48 A5  
 9150- 09 C9 40 80 04 68 4C F7  
 9158- 90 68 C9 8D 00 07 A9 DC  
 9160- 91 28 4C 1E 91 C9 A0 80  
 9168- 03 4C F7 90 C9 A2 F0 F9  
 9170- C9 BA F0 E5 C2 BB F0 F1  
 9178- C9 AC D0 E4 A9 AE D0 E0  
 9180- A5 07 85 1D A5 08 85 1E  
 9188- A5 EB 85 F9 A5 EC 85 FA  
 9190- A5 ED 85 FB A5 EE 85 FC  
 9198- A0 00 B1 FB 85 FF A9 00  
 91A0- 85 18 85 1C A9 A0 85 18  
 91A8- A9 DF 85 FE A2 00 8A 85  
 91B0- 25 20 C1 FB A0 00 B1 28  
 91B8- C9 40 90 4E C9 DC F0 5A  
 91C0- E0 FF F0 56 C9 A0 F0 28  
 91C8- C9 DF F0 30 85 18 85 FE  
 91D0- A5 FF F0 18 A5 18 C9 AB  
 91D8- F0 12 C9 AD B0 0A A9 6A  
 91E0- 91 28 20 D0 FB 4C E9 90  
 91E8- C9 BA B0 F2 A5 18 D0 16  
 91F0- C5 18 F0 16 A5 FF D0 E6  
 91F8- A9 A0 D0 0A C5 FE F0 0A  
 9200- A5 FF D0 DA A9 DF 9D 00  
 9208- 02 E8 C8 C0 28 D0 A7 E6  
 9210- 25 A5 25 C9 18 D0 9A 4C  
 9218- B3 92 A5 1C C5 1A 90 0C  
 9220- A5 1B C5 19 90 06 20 D0  
 9228- FB 4C E9 90 E0 FF D0 0A  
 9230- A9 5D 91 28 20 D0 FB 4C  
 9238- E9 90 84 1F A0 00 8A D1  
 9240- F9 90 06 F0 04 A4 1F 80  
 9248- E7 48 18 A5 F9 69 02 85  
 9250- F9 90 02 E6 FA 18 A5 FB  
 9258- 69 02 85 FB 90 02 E6 FC  
 9260- B1 FB 85 FF 68 85 06 91  
 9268- 1D F0 28 38 A5 6F F1 1D  
 9270- 85 6F B0 02 C6 70 C8 A5  
 9278- 6F 91 1D C8 A5 70 91 1D  
 9280- A2 00 A0 00 B0 00 02 29  
 9288- 7F 91 6F E8 E4 06 F0 03  
 9290- C8 D0 F1 E6 18 D0 02 E6  
 9298- 1C A4 1F A2 00 A9 A0 85  
 92A0- 18 A9 DF 85 FE 18 A5 1D  
 92A8- 69 03 85 1D 90 02 E6 1E  
 92B0- 4C 0A 92 A5 1C C5 1A F0  
 92B8- 03 4C 26 92 A5 1B C5 19  
 92C0- D0 F7 4C B7 00 A9 00 85  
 92C8- 1B 85 1C 85 06 A9 17 85  
 92D0- 25 20 C1 FB A0 27 B1 28  
 92D8- C9 A0 F0 0C C9 DF F0 08  
 92E0- A6 06 D0 1C E6 06 D0 18  
 92E8- A6 06 F0 14 C6 06 E6 1B  
 92F0- D0 02 E6 1C A5 1C C5 1A  
 92F8- D0 06 A5 1B C5 19 F0 11  
 9300- 89 10 D3 C6 25 A5 25 10  
 9308- C8 A9 00 85 24 85 25 F0  
 9310- 08 C8 C0 28 D0 04 A0 00

309 BCC E62  
 310 INC %FC  
 311 E62 LDA (%FB),Y  
 312 STA %FF ;STOCKE LE TYPE  
 313 PLA  
 314 STA %6 ;SAUVEGARDE LONGUEUR REELLE  
 315 STA (%1D),Y ;LA STOCKE DANS LA ZONE DES TABLEUX  
 316 BEQ E7 ;LONGUEUR = 0 --> PAS DE "CHAINE" A STOCKER  
 317 SEC  
 318 LDA %6F ;MISE A JOUR DU POINTEUR DE FIN DE LA ZONE  
 319 SBC (%1D),Y ;DE STOCKAGE DES CHAINES DE CARACTERES  
 320 STA %6F  
 321 BCS E8  
 322 DEC %70  
 323 E8 INY  
 324 LDA %6F ;STOCKE L'ADRESSE DE LA CHAINE AU NIVEAU DE  
 325 STA (%1D),Y ;L'INDICE CONCERNE DANS LA ZONE DES TABLEUX  
 326 INY  
 327 LDA %70  
 328 STA (%1D),Y  
 329 LDX E0  
 330 LDY E0  
 331 E9 LDA \$200,X  
 332 AND %7F ;TRANSFERT DE LA CHAINE DANS LA ZONE DE  
 333 STA (%6F),Y ;STOCKAGE APRES REMISE EN ASCII POSITIF  
 334 INX  
 335 CPX %6  
 336 BEQ E7  
 337 INY  
 338 BNE E9  
 339 E7 INC %1B ;MISE A JOUR DU NOMBRE DE DONNEES LUES  
 340 BNE E10  
 341 INC %1C  
 342 E10 LDY %1F  
 343 LDX E0  
 344 LDA %A0 ;REMISE A LA VALEUR INITIALE DES PARAMETRES  
 345 STA %18  
 346 LDA %DF  
 347 STA %FE  
 348 CLC  
 349 LDA %1D  
 350 ADC E3 ;ADRESSE DE L'ELEMENT SUIVANT DU TABLEAU  
 351 STA %1D ;S'IL EN RESTE A SAISIR  
 352 BCC E100  
 353 INC %1E  
 354 E100 JMP E1 ;PASSAGE AU CARACTERE SUIVANT SUR L'ECRAN  
 355 C0 LDA %1C  
 356 CMP %1A  
 357 BEQ C1 ;A CE NIVEAU, LE COMPTEUR DE VALEURS LUES  
 358 C2 JMP E11  
 359 C1 LDA %1B ;EST MAJORE DE 1 ET DOIT DONC ETRE EGAL  
 360 CMP %19 ;AU NOMBRE DE DONNEES ATTENDUES  
 361 BNE C2  
 362 JMP %B7 ;REND LA MAIN AU BASIC  
 363 SD LDA E0 ;REPRISE POUR AFFICHAGE ET SAISIE D'UN TABLEAU  
 364 STA %1B  
 365 STA %1C  
 366 STA %6  
 367 LDA %17 ;RECHERCHE DES LIBELLES A PARTIR  
 368 STA %25 ;DU BAS DE L'ECRAN  
 369 SD5 JSR %FBC1  
 370 LDY %27

```

9318- E6 25 84 24 A9 00 85 18
9320- 85 1C A5 07 85 1D A5 08
9328- 85 1E A0 00 A2 00 B1 1D
9330- 85 06 F0 18 C8 B1 1D 85
9338- CE C8 B1 1D 85 CF A0 00
9340- B1 CE 09 80 9D 00 02 E8
9348- E4 06 F0 03 C8 D0 F1 A9
9350- DC 9D 00 02 E6 06 A2 00
9358- A5 25 20 C1 FB A4 24 B1
9360- 28 C9 40 90 1E BD 00 02
9368- 91 28 E8 E4 06 F0 2A C8
9370- C0 28 D0 F1 E6 25 A5 25
9378- C9 18 F0 14 20 C1 FB A0
9380- 00 F0 E2 C8 C0 28 D0 D7
9388- E6 25 A5 25 C9 18 D0 03
9390- 4C 3F 90 A0 00 84 24 F0
9398- C1 E6 18 D0 02 E6 1C A5
93A0- 1C C5 1A D0 09 A5 18 C5
93A8- 19 D0 03 4C E9 90 C8 C0
93B0- 28 D0 0D E6 25 A5 25 C9
93B8- 18 F0 D3 20 C1 FB A0 00
93C0- B1 28 C9 40 B0 E8 84 24
93C8- A5 1D 69 03 85 1D 90 02
93D0- E6 1E 4C 2A 93

```



## Trucs et Astuces

### Un bug du Filer de ProDOS 1.0

Si l'on formate (Volume Commands, Option F) une disquette dont le nom comporte des minuscules, la disquette ne sera pas formatée correctement.

- Un CAT ne fonctionne pas ;
- Certaines commandes du Filer, par exemple D)etect bad blocks, B)lock allocation, R)ename a volume..., entraînent un "VOLUME NOT FOUND" tandis que L)ist volumes indique l'existence de ce volume.

**Conclusion** : puisque le Filer ne met pas automatiquement le nom de la disquette en majuscules, faites-le vous-même !

```

371 SD4 LDA ($28),Y
372 CMP #$A0
373 BEQ SD0
374 CMP #$DF ;LAISSÉ LES ESPACES ET LES " _ "
375 BEQ SD0
376 LDX $6 ;DRAPEAU POUR SAVOIR SI L'ON EST DANS UN MEME
377 BNE SD1 ;LIBELLE OU SI L'ON EN COMMENCE UN NOUVEAU
378 INC $6
379 BNE SD1

380 SD0 LDX $6
381 BEQ SD1
382 DEC $6
383 INC $18 ;QUAND ON RETROUVE UN ESPACE OU UN " _ " APRES
384 BNE SD2 ;ANALYSE D'UN LIBELLE EN INVERSE, IL Y AURA
385 INC $1C ;UNE DONNÉE A METTRE EN REGARD DE CE LIBELLE
386 SD2 LDA $1C
387 CMP $1A
388 BNE SD1
389 LDA $18 ;ON DOIT LOCALISER AUTANT DE LIBELLES QUE
390 CMP $19 ;DE DONNÉES A AFFICHER
391 BEQ SD3
392 SD1 DEY
393 BPL SD4
394 DEC $25
395 LDA $25
396 BPL SD5
397 LDA $0
398 STA $24
399 STA $25
400 BEQ SD6
401 SD3 INY
402 CPY $28 ;DEPART DE L'AFFICHAGE DES DONNÉES
403 BNE SD7
404 LDY $0
405 INC $25
406 SD7 STY $24
407 SD6 LDA $0
408 STA $18
409 STA $1C
410 LDA $7
411 STA $1D
412 LDA $8
413 STA $1E
414 SD21 LDY $0
415 LDX $0
416 LDA ($10),Y ;RECUPERE LA LONGUEUR DE LA CHAÎNE
417 STA $6
418 BEQ SD80 ;LONGUEUR = 0 --> PAS DE CHAÎNE A AFFICHER
419 INY
420 LDA ($10),Y
421 STA $CE ;RECUPERE L'ADRESSE DE LA CHAÎNE
422 INY
423 LDA ($10),Y
424 STA $CF
425 LDY $0
426 SD9 LDA ($CE),Y ;TRANSFERT DE LA CHAÎNE DANS LE BUFFER
427 ORA $80 ;APRES REMISE EN ASCII NEGATIF
428 STA $200,X
429 INX
430 CPX $6
431 BEQ SD80
432 INY

```

433	BNE	SD9	
434 SD80	LDA	£\$DC	;PLACE UN "PSEUDO RETURN" EN FIN DE DONNEE
435	STA	\$200,X	
436	INC	\$6	
437 SD8	LDX	£0	
438	LDA	\$25	
439 SD15	JSR	\$FBC1	
440	LDY	\$24	
441 SD13	LDA	(\$28),Y	
442	CMP	£\$40	
443	BCC	SD10	;PROTEGE LES DONNEES EN INVERSE
444 SD12	LDA	\$200,X	;RECOPIE LA CHAINE SUR L'ECRAN
445	STA	(\$28),Y	
446	INX		
447	CPX	\$6	;CHAINE TERMINEE ?
448	BEQ	SD11	
449	INY		
450	CPY	£\$28	;PASSAGE A POSITION SUIVANTE SUR L'ECRAN
451	BNE	SD12	
452	INC	\$25	
453	LDA	\$25	
454	CMP	£\$18	
455	BEQ	SD16	;ON NE DEVRAIT JAMAIS TOUCHER LA FIN DE L'ECRAN
456	JSR	\$FBC1	
457	LDY	£0	
458	BEQ	SD12	;SUITE DE L'AFFICHAGE
459 SD10	INY		;SAUTE LES "INVERSE" AVANT DE COMMENCER
460	CPY	£\$28	;A AFFICHER LA CHAINE
461	BNE	SD13	
462	INC	\$25	
463	LDA	\$25	
464	CMP	£\$18	
465	BNE	SD14	
466 SD16	JMP	\$6	;FIN DE L'ECRAN SANS FIN D'AFFICHAGE --> ERREUR
467 SD14	LDY	£0	
468	STY	\$24	
469	BEQ	SD15	
470 SD11	INC	\$18	;MISE A JOUR DU NOMBRE DE CHAINES AFFICHEES
471	BNE	SD17	
472	INC	\$1C	
473 SD17	LDA	\$1C	
474	CMP	\$1A	
475	BNE	SD18	;TOUTES LES CHAINES NE SONT PAS ENCORE AFFICHEES
476	LDA	\$1B	
477	CMP	\$19	
478	BNE	SD18	;IDEM
479	JMP	\$7	;SINON --> REPREND UNE SAISIE
480 SD18	INY		;ON VEUT PASSER A LA POSITION SUIVANTE A L'ECRAN
481	CPY	£\$28	
482	BNE	SD19	
483	INC	\$25	
484	LDA	\$25	
485	CMP	£\$18	
486	BEQ	SD16	;JAMAIS LE BAS DE L'ECRAN NE DOIT ETRE ATTEINT
487	JSR	\$FBC1	
488	LDY	£0	
489 SD19	LDA	(\$28),Y	
490	CMP	£\$40	
491	BCS	SD18	;IL FAUT RELOCALISER LE LIBELLE SUIVANT
492	STY	\$24	
493	LDA	\$1D	
494	ADC	£3	;PASSE A L'INDICE SUIVANT DANS LE TABLEAU

**Pom's**

**Apple  
Expo !**

**C'est du  
14 au 16  
Juin**

**Venez  
Nous  
Voir . . .**

**Pom's  
Pom's**

```

495 STA $1D
496 BCC SD20
497 INC $1E
498 SD20 JMP SD21

```

;REPRISE DE LA PROCEDURE D'AFFICHAGE

## TEST.IPE2

```

1 HIMEM: 9 * 4096 - 1
5 D$ = CHR$(4): PRINT D$"BLOAD IPE2": P
  RINT D$"BLOAD INTER
10 TEXT : HOME
15 PO$ = "
  "
20 DIM D1$(9),L1%(9),T1%(9): FOR I = 0 T
  O 9: READ L1%(I),T1%(I): NEXT
22 DATA 10,0,5,1,20,0,2,1,2,1,5,0,10,0,
  4,1,6,1,30,0
25 DIM D2$(5),L2%(5),T2%(5): FOR I = 0 T
  O 5: READ L2%(I),T2%(I): NEXT : DA
  TA 10,0,6,1,6,1,20,0,1,1,6,0
27 GOTO 100
30 HOME :Z$ = "ECRAN NO 1": HTAB 20 - L
  EN (Z$) / 2: PRINT Z$: PRINT
33 FOR I = 0 TO 9: PRINT : PRINT "DONNEE
  ."I".(T1%(I)).:" LEFT$ (PO$,L1%(
  I)): NEXT
35 RETURN
40 HOME :Z$ = "ECRAN 1 MODIFIE": HTAB 20
  - LEN (Z$) / 2: PRINT Z$: PRINT
  : PRINT : PRINT "DON.0.:" SPC( 15)

```

```

"DON.1.:" : PRINT : PRINT "DON.2.:"
  SPC( 15)"DON.3.:"
45 FOR I = 4 TO 9: PRINT : HTAB 10: PRIN
  T "DON."I".:" LEFT$ (PO$,L1%(I)):
  NEXT : RETURN
50 HOME :Z$ = "ECRAN 2": HTAB 20 - LEN
  (Z$) / 2: PRINT Z$: FOR I = 0 TO 5
  : PRINT : PRINT "DONNEE."I".:" LEF
  T$ (PO$,L2%(I)): NEXT : RETURN
60 HOME : PRINT "ECRAN 2 MODIFIE":
  FOR I = 0 TO 5: PRINT : HTAB 10: P
  RINT "DONNEE."I".:" LEFT$ (PO$,L2%
  (I)): NEXT : RETURN
100 GOSUB 30:$SD1$,L1%,T1%: GOSUB 30:$DD
  1$,L1%,T1%: GOSUB 40:$DD1$,L1%,T1%
  : GOSUB 50:$SD2$,L2%,T2%: GOSUB 60
  :$DD2$,L2%,T2%
110 HOME : FOR I = 0 TO 9: PRINT D1$(I):
  NEXT : PRINT : FOR I = 0 TO 5: PR
  INT D2$(I): NEXT : GET Z$
120 FOR I = 0 TO 9:D1$(I) = "": NEXT : F
  OR I = 0 TO 5:D2$(I) = "": NEXT
130 GOSUB 30:$DD1$,L1%,T1%: GOSUB 50:$DD
  2$,L2%,T2%: HOME : FOR I = 0 TO 9:
  PRINT D1$(I): NEXT : PRINT : FOR
  I = 0 TO 5: PRINT D2$(I): NEXT

```

# Le porte-parole

Olivier Herz

Ediciel (Matra & Hachette) nous propose un synthétiseur de parole qui parle français, sous forme d'une carte (à brancher dans un slot, le 4 de préférence) et d'une disquette.

## La carte

Elle comporte :

- Un chip synthétiseur de parole MEA 8000 permettant la synthèse de parole et la génération de sons musicaux.

- Un amplificateur basse fréquence pour amplifier le signal vers un haut-parleur.

- Des emplacements libres pour recevoir des RAMs ou des PROMs, et les décodeurs pour y accéder.

- Une sortie vers le haut-parleur de l'Apple, que l'on doit alors débrancher de la carte mère (ce n'est pas très commode sur un II+ !).

**Inconvénients :** le haut-parleur est déconnecté pour les usages classiques, mais fait du bruit à cause de parasites électriques quand les drives tournent ! On aurait aimé une possibilité de 'switcher' sans avoir à ouvrir le capot et rebrancher les fils.

- Une sortie vers un haut-parleur externe, de type prise de chaîne Hi-Fi, que l'on peut relier à un amplificateur. J'aurais préféré une petite

prise 'jack' pour relier à un magnétophone, mais c'est personnel (mon Apple est à 2 mètres de ma chaîne, à vol d'oiseau).

## Le logiciel de synthèse de texte

### Première face de la disquette

Malgré de nombreux défauts ergonomiques (par exemple, pas d'éditeur de lignes des phrases entrées), ce programme est un magnifique bébé né de l'union du hardware et du software.

Il permet de créer des fichiers de messages parlés, que l'on peut intégrer à des programmes Basic ou assembleur, voire au célèbre EDI-LOGO.

Les deux grandes possibilités sont :

- L'édition en français : on rentre une phrase (avec des codes spéciaux pour les accents...), le programme la traduit en phonèmes et peut la prononcer.

Et cela marche ! En français intelligible et agréable. On peut même mettre un point d'interrogation, et l'intonation changera. De plus, on peut indiquer les groupes verbaux et nominaux avec des marqueurs spé-

ciaux, de façon à se rapprocher d'une élocution naturelle.

Toutefois, ce programme n'a pas la science infuse et commet des erreurs. Que pensez-vous des phrases "Les poules du couvent couvent" et "Nous portions nos portions" ?

- C'est pourquoi on peut aussi éditer en phonétique, en indiquant les phonèmes. Pour des raisons scientifiques, la solution employée ici pour reconstituer la parole est celle retenue par le CNET, qui part des couples de phonèmes successifs.

C'est tellement bon que je l'ai fait parler en anglais, en allemand et ... en japonais. Son accent français est tout simplement délicieux ("Maille teil'lor iz ritche").

## Mise au point graphique

### Deuxième face de la disquette

Ce programme permet de créer des notes de musiques et des effets spéciaux sonores en faisant varier l'amplitude du signal, la fréquence et la largeur des bandes et en visualisant le tout. Il peut créer ses propres fichiers de bruits ou bien modifier des messages créés par le synthétiseur de parole.



# VERSION SOFT, LES FRUITS

## L'édition française

### **PROCEDURE**

L'OUTIL INDISPENSABLE DU DÉVELOPPEUR.

Premier assembleur éditeur pleine page,  
pour APPLE IIe et APPLE IIc.



Si vous avez la programmation dans la peau, initiés ou débutants, Procure est le logiciel qu'il vous faut. Il vous permet de développer en un temps record grâce à ses fonctions d'éditeur pleine page 80 colonnes. Étudié pour ProDOS 64 k ou 128 k, Procure peut assembler sans accès disque jusqu'à 3 sources en mémoire. Normal, on travaille entre Pros!

### **BUDGET FAMILIAL**

LE PLAISIR DE LA GESTION.

Gestionnaire de talent pour le plaisir de rentrer dans ses comptes.



Budget Familial vous en donne pour votre argent! Il vous permet d'enregistrer tous les mouvements de votre compte bancaire et d'avoir l'œil sur vos dépenses. Avec lui, plus de problèmes de gestion, vous êtes sûr d'y trouver votre compte. Même en ignorant tout de la gestion, vous pouvez très vite devenir un super gestionnaire. Avec le plaisir en plus...

LES LOGICIELS VERSION SOFT UTILISENT LE SYSTÈME D'EXPLOITATION PRODOS  
LES MENUS DÉROULANTS PERMETTENT D'ACCÉDER DIRECTEMENT AUX DIFFÉRENTES FONCTIONS.

# D'UNE PASSION de logiciels

Rendez-vous  
Apple expo

## EPISTOLE

### L'ÉCRITURE SOURIS.

Best-seller incontesté  
du traitement de texte français.

## VERSION CALC

### LE CALCUL SOURIS.

Premier tableur graphique  
multi-fonctions utilisant la souris.

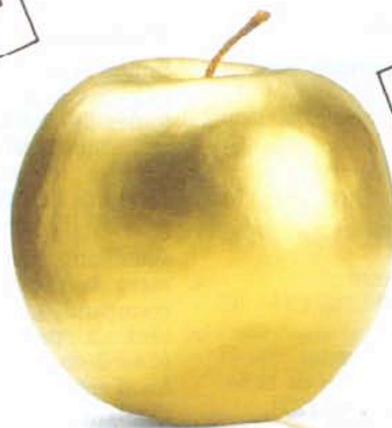


NOUVELLE FONCTION  
COMMUNICATION

Partagez le plaisir  
d'écrire avec Épistole.  
Modèle de simplicité et  
de puissance, ce logiciel  
est l'outil idéal pour tous  
vos traitements de texte  
français.

En un clin d'œil, Épistole  
rédige toutes vos lettres, vos  
rapports et vos mailings — avec  
le souci en moins.

Vous voulez effacer un mot, changer  
une phrase, déplacer un paragraphe?  
Rien de plus simple, l'écriture  
souris vous laisse les mains libres!  
Épistole vous offre l'écriture facile.  
A vous l'imagination!



POMME D'OR  
DU MEILLEUR  
LOGICIEL 84

Voici enfin le logiciel  
adapté à toute votre  
gestion quotidienne. Il  
suffit d'une simple  
pression du doigt avec la  
souris pour comprendre que  
toutes vos opérations de prévisions,  
de synthèses ou d'analyses,  
deviennent alors un jeu  
d'enfant.

Vous souhaitez une représen-  
tation graphique? Facile! Ouvrez  
une fenêtre, cliquez et faites le plein  
d'images.

De mémoire de logiciel, on n'avait encore  
jamais vu une telle facilité alliant puissance  
et rapidité.



19, RUE GANNERON 75018 PARIS

DÉMONSTRATION ET VENTE CHEZ VOTRE CONCESSIONNAIRE AGRÉÉ APPLE.



P 18

# Gestion de fenêtres texte

Georges Zwingelstein

La mode étant au Macintosh, sa souris et ses fenêtres, je n'ai pu m'empêcher de succomber à la tentation de donner une de ces particularités à l'Apple II : les fenêtres. J'ai choisi de gérer les fenêtres en écran texte plutôt qu'en haute résolution, car elles ont l'avantage d'occuper moins de place mémoire, d'avoir un temps de réponse acceptable, de ne pas nécessiter une gestion de curseur particulière, et surtout parce qu'elles servent, le plus souvent, à afficher des menus sur le Macintosh.

Ce programme permet à l'aide de l'ampersand (&) de créer, d'ouvrir, de fermer et d'échanger à volonté des fenêtres, pouvant se superposer les unes aux autres, sans perte d'informations.

## Les commandes

Les principales commandes concernent :

- la création
- l'ouverture
- la fermeture
- l'échange.

### La création

La syntaxe est la suivante :  
& C NF,MD,MG,MH,MB,CH (C pour Créer)

avec les conventions suivantes :

- NF représente le numéro de la fenêtre (compris entre 1 et 127).
- MD définit la marge droite.
- MG définit la marge gauche.
- MH délimite la marge haut.
- MB délimite la marge bas.
- CH est le code écran (optionnel) du caractère délimitant les bords de la fenêtre, les marges étant données par rapport à ceux-ci.

Pour obtenir, par exemple, une fenêtre remplissant tout l'écran, il suffit de donner l'ordre :

& C1,1,40,1,24 (en 40 colonnes)  
& C1,1,80,1,24 (en 80 colonnes)

Ce logiciel fonctionne également en 80 colonnes sur Apple //e. Quelques remarques sont néanmoins nécessaires sur une utilisation en 80 colonnes, les commandes d'ouverture et de fermeture de fenêtre utilisant les adresses standard, pour la protéger (\$20-\$23 ou 32 à 35 en décimal) :

- les colonnes étant numérotées de 1 à 80, on ne peut protéger une fenêtre qu'entre une colonne impaire (à gauche) et une colonne paire (à droite).
- la marge gauche est donc arrondie au premier nombre impair inférieur.
- la marge droite est, de la même façon, arrondie au premier nombre pair supérieur.

Par exemple, pour une fenêtre sur les deux colonnes 65 et 66, les commandes suivantes sont équivalentes :

&C1,65,65,...  
&C1,66,65,...  
&C1,65,66,...  
&C1,66,66,...

### L'ouverture

& O NF (O comme Ouvrir)  
où NF représente le numéro de la fenêtre que l'on désire ouvrir. C'est tout.

### La fermeture

La commande de fermeture est encore plus simple, puisqu'on ne peut fermer que la fenêtre se trouvant au-dessus de toutes les autres :  
& F (F comme Fermer)

### L'échange

Enfin, la commande permettant de faire passer une fenêtre au-dessus de toutes les autres est légèrement plus compliquée, le numéro NN de la fenêtre à remonter ne pouvant être connu de l'ordinateur :  
& E NF (E comme Echanger)

Attention : afin d'améliorer notablement la rapidité du transfert, cette commande crée dans le sens décroissant, à partir de l'adresse \$2FF (c'est-à-dire sur un morceau du buffer d'entrée), une table des adresses des fenêtres concernées par le transfert; il y a donc risque de recouvrement d'une commande préalablement entrée au clavier, si celle-ci fait sensiblement plus de 255 - 2 \* (nombre d'adresses) caractères.

Dois-je vous rappeler, au passage, la commande qu'il ne faut pas oublier sous peine de perdre votre sang froid ? L'initialisation. Il s'agit de :  
& I (I comme ???)

Cette commande est automatiquement réalisée lorsqu'on lance le programme : elle identifie le type de l'Apple, ainsi que l'activité d'une éventuelle carte 80 colonnes sur Apple //e, crée une pseudo fenêtre de numéro 0 juste sous le début du programme, puis efface les variables (CLEAR) et enfin positionne HIMEM sous la première fenêtre.

### Le positionnement

&P F,MG,MH (P comme Positionner)

permet de déplacer une fenêtre; ce qui autorisera une mise en place plus facile des différentes fenêtres utilisées dans un programme. NF est comme à l'habitude le numéro de la fenêtre, MG et MH les nouvelles marges gauche et haut de la future position de cette même fenêtre.

Attention : comme pour l'échange, une table est créée dans le buffer d'entrée.

J'ai d'autre part activé, pour vous soulager, la touche de tabulation, celle de l'Applesoft ne fonctionnant pas correctement en 80 colonnes, et j'ai également repris, en la modifiant un peu, la routine de menu proposée par Denis Sureau dans l'article "La programmation facilitée" du Pom's 5. Ces commandes travaillent à l'intérieur de la fenêtre, quelle que soit la position de celle-ci. Par exemple, pour positionner le curseur sur la première ligne et la première colonne de la fenêtre, on utilisera :

& VTAB 1:& HTAB 1

Ces commandes ne seront pas toujours nécessaires, le programme se chargeant de mémoriser et de rappeler la position du curseur, lorsque celui-ci change de fenêtre.

Pour appeler la routine de menu, la commande sera :

& LET NV,VT,LN,NI,(IL)

avec les conventions suivantes :

- NV = nom d'une variable
- VT = tabulation verticale dans la fenêtre
- LN = longueur de la barre en inverse
- NI = nombre d'items
- IL = interligne (optionnelle, l'interligne initiale étant de 1).

La routine vérifie d'abord que l'ensemble du menu rentre à l'intérieur de la fenêtre, utilise non plus des chiffres mais des lettres à partir de "A" pour la sélection, ce qui autorise un nombre d'items compris entre 1 et 26 et tient compte des flèches verticales de l'Apple //e. De plus, la barre de sélection n'inverse que les caractères inversables.

J'ai enfin transformé un sous-programme qui me permettait de déplacer HIMEM pour créer une fenêtre sans perdre les variables chaînes, en une commande :

& HIMEM : NNNN

où NNNN représente la nouvelle valeur non pas de HIMEM, mais du début de la zone libre entre les fenêtres et le programme, vous permettant d'y charger vos propres routines (le début du programme se trouve en \$8A00 ou 35328 en décimal).

Désirant, de plus, pouvoir récupérer la place d'une fenêtre devenue inutile, il m'a été nécessaire de rajouter la commande :

&D NF (D comme Détruire)

avec laquelle on peut détruire la fenêtre de numéro NF à condition que celle-ci soit fermée, ce qui est évident si elle est devenue inutile.

Enfin, comme il est beaucoup plus agréable, dans un programme Basic, de n'avoir qu'à charger une fenêtre préalablement créée et sauvée sur

# Applemania

## Symptômes.

- |   | OUI                      | NON                      |   | OUI                      | NON                      |
|---|--------------------------|--------------------------|---|--------------------------|--------------------------|
| 1. Je veux connaître PRODOS sans être un pro du DOS   | <input type="checkbox"/> | <input type="checkbox"/> | 5. Pianoter des nuits entières, c'est planant. Mais planer tout seul, c'est frustrant.    | <input type="checkbox"/> | <input type="checkbox"/> |
| 2. Ami des bêtes, je cherche un chat à puces pour colorier ma souris en mauve                                       | <input type="checkbox"/> | <input type="checkbox"/> | 6. Je veux relier mon lave-vaisselle à mon Apple II.                                      | <input type="checkbox"/> | <input type="checkbox"/> |
| 3. Devenir à l'aise comme Blaise en Pascal, quel pied!  | <input type="checkbox"/> | <input type="checkbox"/> | 7. Appeler un S.O.S. informatique quand la technique coince, ça dépanne.                  | <input type="checkbox"/> | <input type="checkbox"/> |
| 4. Mon baud à 7 bits cherche un 8 <sup>e</sup> pour compléter son protocole X-On X-Off en vue d'une liaison durable | <input type="checkbox"/> | <input type="checkbox"/> | 8. Je veux apprendre à mieux gérer mon bureau en le jetant par les fenêtres de Macintosh. | <input type="checkbox"/> | <input type="checkbox"/> |

3 OUI et MOINS : vous êtes en pleine incubation, prenez des mesures!

4 à 7 OUI : vous êtes contagieux, vous serez bien entouré au Club Apple.

8 à 10 OUI : bravo, venez d'urgence cultiver votre passion au Club Apple.

## Remède: REJOIGNEZ LE CLUB APPLE

Plein, plein d'avantages vous y attendent :

- une messagerie électronique gratuite;
- un numéro de téléphone pour consulter les ingénieurs Apple 7 jours sur 7;
- tous les mois, le journal du Club : L'Echo des Apple;
- des conférences, réunions d'information, projections privées et rencontres avec des "pros";
- des stages d'initiation et de perfectionnement animés par les "gourous" du Club;
- des voyages organisés en Californie;
- une librairie spéciale Club;
- un Apple Check pour bénéficier de conditions de crédit préférentielles;
- des rencontres organisées avec d'autres Applemaniaques.

EROMIC 2000

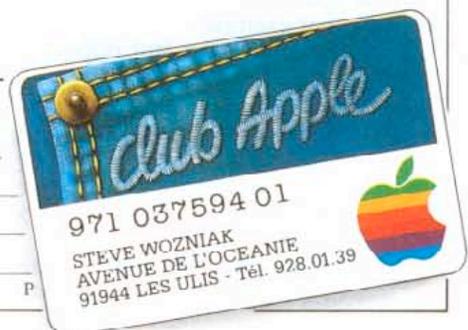
### BULLETIN-REPOSE

Je désire recevoir, sans engagement de ma part, une documentation complète sur le Club Apple et votre formulaire d'inscription.

NOM \_\_\_\_\_

ADRESSE \_\_\_\_\_

CODE POSTAL \_\_\_\_\_



disquette, on disposera des quatre commandes suivantes :

**&SAVE "NOM",F1**

pour sauver la fenêtre numéro F1 à la suite de celles précédemment sau-  
vées dans un fichier d'un type nou-  
veau. Celui-ci apparaît comme bi-  
naire dans le catalogue.

**&LOAD "NOM",F1,N1**

pour charger la N1 ième fenêtre du  
fichier en lui donnant le numéro F1.

**&STORE "NOM",F1,N1**

pour sauver les marges et le contenu  
de la fenêtre F1 à la place de la N1  
ième fenêtre du fichier (il faut que les  
dimensions soient identiques).

**&RECALL "NOM",N1,F1**

pour effectuer la réciproque de  
&STORE sans rappeler les marges;  
"NOM" représente une chaîne de  
caractères contenant le nom du fi-  
chier, dans lequel on ne peut sauve-  
garder plus de 127 fenêtres (ce qui  
est largement suffisant dans la plu-  
part des cas, surtout qu'il n'est pas  
possible d'avoir un catalogue des fe-  
nêtres), accompagné si nécessaire  
des numéros de drive et de slot sui-  
vant la syntaxe du DOS.

Pour SAVE et LOAD :

- si N1 n'est pas précisé, il est pris  
égal à F1
- si ni F1, ni N1 sont donnés, alors  
le programme charge toutes les fe-  
nêtres en leur donnant un numéro  
croissant (sauve toutes les fenêtres  
par numéro croissant à la suite du  
fichier).

Pour STORE et RECALL :

- les valeurs par défaut de F1 et N1  
sont :  
N1 = F1 si N1 n'est pas donné, et  
N1 = F1 = 1 si F1 et N1 sont  
inexistants.

### Remarques

Chaque fois qu'une fenêtre est créée  
ou chargée, la routine FRE(0) est ap-  
pelée. Les commandes de gestion de  
fichiers de fenêtres ne vérifient pas la  
fermeture de la fenêtre. Il sera donc  
possible, par exemple, de fusionner  
des fenêtres ou de récupérer un mor-  
ceau d'une fenêtre en ouvrant la  
nouvelle par dessus puis en la sau-  
vant avant de la fermer.

Il est possible de faire ressortir les fi-  
chiers de fenêtres en POKant dans la  
table des caractères des types de fi-  
chiers, le code voulu à l'affichage du  
catalogue, par exemple :

POKE 45998, 198 pour leur donner  
le type "F".

Tous les paramètres peuvent être  
donnés sous forme d'expression, le  
programme utilise les routines d'éva-  
luation de nombre ou chaîne de ca-  
ractères de l'Applesoft.

### Les messages d'erreur

Les messages d'erreur utilisés ont été

détournés de leurs significations origi-  
nelles.

– OUT OF MEMORY indique que la  
place nécessaire à la création d'une  
fenêtre est insuffisante ou que le dé-  
placement de HIMEM entraînerait un  
recouvrement soit des variables, soit  
du programme de gestion des fe-  
nêtres.

– UNDEF'D STATEMENT signale  
une opération sur une fenêtre inexis-  
tante.

– REDIM'D ARRAY souligne le fait  
que l'on veut créer ou charger une  
fenêtre dont le numéro a déjà été at-  
tribué.

– OVERFLOW indique une opéra-  
tion entraînant un dépassement de  
capacité (ouvrir une fenêtre déjà ou-  
verte, en fermer une quand il n'y en  
a plus à fermer, ou RECALL ou  
STORE sur des fenêtres qui n'ont  
pas les mêmes dimensions).

– ILLEGAL QUANTITY sert pour  
toutes les valeurs interdites : mauvai-  
ses marges, mauvais numéro de fe-  
nêtre, valeurs impossibles d'un ou  
des paramètres du menu, destruction  
d'une fenêtre ouverte...

Enfin, deux messages du DOS utili-  
sés par le programme :

– OUT OF DATA signale que le fi-  
chier a moins de N fenêtres, si l'on  
veut accéder à la N ième fenêtre.

– DISK FULL indique soit une dis-  
quette pleine, soit un fichier conte-  
nant 127 fenêtres, lorsque l'on veut  
sauver une fenêtre.

### La structure du programme

Pour les utilisateurs de l'assembleur  
et les autres, voici quelques informa-  
tions sur la structure du programme.

Le programme se charge en \$89EE,  
commence par brancher l'ampersand  
(\$89EE à \$89FF), puis saute à la

routine d'initialisation.

Celle-ci, créée juste sous l'adresse  
contenue en \$8A00 – \$8A01 (le  
contenu initial est \$8A00), une pre-  
mière fenêtre de numéro 0 corres-  
pondant à tout l'écran et ne possé-  
dant pas de buffer. HIMEM pointe  
alors sur cette fenêtre.

Lorsqu'une fenêtre est créée ou  
chargée, les chaînes ainsi que  
HIMEM sont déplacées vers le bas  
grâce à une partie de &HIMEM,  
pour laisser la place à la nouvelle fe-  
nêtre, qui y est alors copiée.

Lorsque le programme cherche une  
fenêtre, il part de celle pointée par  
HIMEM et remonte jusqu'à la fenêtre  
de numéro 0.

Vous pouvez vous libérer de la place  
pour vos propres routines : soit de  
manière temporaire avec &HIMEM  
(&I replace la fenêtre 0 juste sous  
\$8A00), soit de manière définitive en  
modifiant l'adresse \$8A00 – \$8A01  
et en réinitialisant. L'interpréteur se  
termine en \$8A0F par un saut à la  
routine d'erreur de syntaxe du Basic,  
que vous pouvez donc détourner.

Une fenêtre est mémorisée sous  
forme de 2 parties : les données la  
caractérisant et le buffer qui contient  
ce qui sera affiché à l'écran.

– Les données comprennent :

- le numéro de la fenêtre
- le numéro de la fenêtre ouverte  
suivante (nul si c'est la dernière fe-  
nêtre ouverte, c'est-à-dire celle qui  
est au-dessus des autres).
- les 2 octets de mémorisation des  
tabulations verticales puis horizon-  
tales (tabulations dans la fenêtre).
- les 4 marges de la fenêtre (dans le  
format qui permet une recopie di-  
recte dans les mémoires \$20-\$23).

La gestion des fichiers sur disquette  
utilise quelques routines du DOS  
pour l'ouverture et la fermeture des

## Max : Le moniteur étendu

Jacques Supernant

Apple II+, IIe, IIc

*Ce moniteur autorise un contrôle de l'exécution des  
routines en langage machine.*

- *Un mode Trace et Pas à Pas très évolués et  
sélectifs sont complétés par un accès direct  
aux registres du 6502 (ou 6502E). Il est ainsi  
possible de les initialiser à son gré ou à l'aide de la  
routine de hasard.*
- *La gestion des fenêtres d'écran simplifie le  
mode Trace.*
- *Une routine permet la recherche de suites  
d'octets.*
- *Un mini-assembleur très souple fait partie de  
MAX.*
- *Une ligne de commande peut devenir une  
boucle avec l'ordre JUMP.*

Les fichiers source sont sur la disquette.

**Disquette et documentation : 150,00 F TTC franco  
(bon de commande page 78).**

fichiers puis le File Manager pour gérer le contenu qui est organisé de la manière suivante :

Le premier octet contient le nombre de fenêtres dans le fichier. Puis on trouve les fenêtres enregistrées de manière séquentielle (de même que pour l'organisation en mémoire, un

enregistrement contient deux zones : la zone des données qui ne comprend que les 4 marges, puis la zone du buffer qui contient le contenu de la fenêtre).

Je termine enfin, en vous souhaitant plein de petites fenêtres !

**N.D.L.R.** : Le listing du programme Basic a été quelque peu modifié pour les lecteurs n'ayant pas la disquette d'accompagnement à leur disposition. La version originale (sur la disquette) charge les fenêtres FGEN, FPRE et FPOM'S 14.

## Programme ESSAI

```
10 TEXT : HOME
20 PRINT CHR$(4)"BRUN WINDOW"
30 NN = 9:COL = 40: REM METTRE 80 POUR ESSAI EN 80 COLONNES, ET PRINT CHR$(4)"PRÉ3" AVANT LA LIGNE 20
100 & VTAB 10: & HTAB 5: PRINT "EXEMPLE D'UTILISATION DES FENETRES"
110 & VTAB 13: & HTAB 7: PRINT "PATIENCE, JE CREE DES FENETRES"
200 FOR I = 1 TO NN
210 MG = COL * RND(1) + 1:MD = COL * RND(1) + 1: IF MD - MG < 12 THEN 210
220 MH = 24 * RND(1) + 1:MB = 24 * RND(1) + 1: IF MB - MH < 4 THEN 220
230 & CI,MG,MD,MH,MB,256 * RND(1)
240 NEXT I
245 SPEED= 100
```

```
250 & C127,1,COL,20,24,32: & O127: GOSUB 1000
260 & VTAB 3: & HTAB 3: PRINT "OUVERTURE DES FENETRES";: GOSUB 1000
300 FOR I = 1 TO NN: & OI
310 & VTAB 3: & HTAB 3: PRINT "FENETRE";I;: GOSUB 1000
320 NEXT I
330 & E127: GOSUB 1000: & HTAB 3: PRINT "PUIS ECHANGE DES FENETRES";: GOSUB 1000
400 FOR I = 1 TO NN: & EI: GOSUB 1000: NEXT I
410 & E127: GOSUB 1000: & HTAB 3: PRINT "ENFIN FERMETURE DES FENETRES";: GOSUB 1000: & F: GOSUB 1000
500 FOR I = 1 TO NN: & F: GOSUB 1000: NEXT I
999 END
1000 FOR J = 1 TO 1000: NEXT J: RETURN
```

## Source Big MAC

### Programme WINDOW.S

```
1 *****
2 *
3 * PROGRAMME DE GESTION DE
4 * FENETRES TEXTES
5 *
6 * PAR G. ZWINGELSTEIN
7 *
8 *****
9
10 LISTING KBD ;DEMANDE AFFICHAGE LISTING
11 DO LISTING
12 ELSE
13 LST OFF
14 FIN
15 SAVEOBJ KBD ;DEMANDE SAUVEGARDE DU CODE OBJET
16 DO SAVEOBJ
17 DSK WINDOW
18 FIN
19 EXP OFF ;IMPRIME SEULEMENT LE CODE MACRO
20 TR ON ;3 BYTES PAR LIGNES
21
22 * SOUS-PROGRAMME DEPENDANT DE LA CARTE 80 COL. :
23 * VTAB
24 * HTAB
25 * (GETPARAM,GETNEW,CHKMARGE)
26 * SAVECURS
27 * SCREENIT
28 * INVERT
29 * CALCBAS
30 * APPLEID
31
32 ORG $89EE
33
34 ESSAIECH = 1 ;ASSEMBLAG
```

```
E MODIF. ECHANGE.
35 ESSAIMOV = 1 ;ASSEMBLAG
E MODIF. DEPLACE
36 ESSAI80 = 0 ;ASSEMBLAG
E MODIF. SCREENIT
37 ; POUR 80 COLONNES
38 PAG
39 *
40 * COMMANDES MACROS
41 DO 0
42
43 INCR MAC
44 INC $1
45 BNE NI
46 INC $1+1
47 NI <<<
48
49 DECR MAC
50 LDX $1
51 BNE ND
52 DEC $1+1
53 ND DEC $1
54 <<<
55
56 CMPR MAC
57 LDA $1
58 CMP $2
59 LDA $1+1
60 SBC $2+1
61 <<<
62
63 TR MAC
64 LDA $1
65 STA $2
66 <<<
67
68 TRDB MAC
69 >>> TR.$1;$2
70 >>> TR.$1+1;$2+1
71 <<<
72
73 TRAX MAC
74 LDA $1
```

```

75      LDX  $1+1
76      STA  $2
77      STX  $2+1
78      <<<
79
80 AD    MAC
81      LDA  $1
82      ADC  $2
83      STA  $3
84      <<<
85
86 ADD   MAC
87      CLC
88      >>> AD.$1;$2;$3
89      >>> AD.$1+1;$2+1;$3+1
90      <<<
91
92 SB    MAC
93      LDA  $1
94      SBC  $2
95      STA  $3
96      <<<
97
98 SUB   MAC
99      SEC
100     >>> SB.$1;$2;$3
101     >>> SB.$1+1;$2+1;$3+1
102     <<<
103
104 BUMP  MAC
105     ADC  $1
106     STA  $1
107     BCC  BPD
108     INC  $1+1
109 BPD   <<<
110
111      FIN
112     PAG
113 *
114 * CONSTANTES
115 *
116
117 * VALEUR DES TOKENS
118 HTAB: = $96
119 VTAB: = $A2
120 HIMEM: = $A3
121 RECALL: = $A7
122 STORE: = $A8
123 LET: = $AA
124 LOAD: = $B6
125 SAVE: = $B7
126
127 *
128 * VARIABLES
129 *
130
131 YSAV1 = $1F ;SAUVEGARD
    E DU REGISTRE Y
132 WNDLFT = $20 ;TABLE FEN
    ETRE DU MONITEUR
133 WNDWDTH = $21
134 WNDTOP = $22
135 WNDBTM = $23
136 CH = $24
137 CV = $25
138 BASL = $28
139 PROMPT = $33
140 A1L = $3C
141 A2L = $3E
142 A3L = $40
143 A4L = $42
144 LINNUM = $50
145 TEMPPT = $52
146 TEMPST = $55 ;POINTE VE

```

```

RS TEMP STRING DESCR.
147 INDEX = $5E
148 VARTAB = $69
149 ARYTAB = $6B
150 STREND = $6D
151 FRETOP = $6F
152 FRESPEC = $71
153 MEMSIZ = $73 ;HIMEM DU
    BASIC
154 CURLIN = $75 ;NUMERO LI
    GNE EN COURS
155 ; (CURLIN+1 = $FF SI MODE DIRECT)
156 FORPNT = $85 ;POINTEUR
    SUR UNE VARIABLE
157 DSCLEN = $8F ;LONGUEUR
    DU DESCRIPTeur DE LA VAR.
158 ARYPNT = $94
159 IN = $0200
160 AMPER = $03F5
161 OLDCH = $047B
162 OURCH = $057B
163 OURCV = $05FB
164 KBD = $C000
165 RDALTCAR = $C01E ;LIT SWITC
    H ALTERNATE CHARACTER SET (APPLE I
    IE)
166 KBDSTRB = $C010
167 CHRGET = $B1 ;RECUPERE
    UN CODE D'UNE LIGNE BASIC
168 CHRGET = $B7 ;IDEM SANS
    INCR. TXTPTR
169 MEMERR = $D410 ;OUT OF ME
    MORY ERROR
170 ERROR = $D412 ;AFFICHE E
    RREUR POINTE PAR X
171 RESTART = $D43C ;DEPART A
    CHAUD DU BASIC
172 CLEARC = $D66C ;COMMANDE
    CLEAR
173 NEWSTT = $D7D2 ;RETOURNE
    AU BASIC
174 UNDERR = $D97C ;UNDEF. ST
    ATEMENT ERROR
175 PRINT = $DAD5 ;COMMANDE
    PRINT
176 FRMNUM = $DD67 ;RECUPERE
    UN NOMBRE
177 FRMEVL = $DD7B ;EVALUE LA
    FORMULE POINTEE PAR TXTPTR
178 SYNERR = $DEC9 ;SYNTAX ER
    ROR
179 CHKCOM = $DEBE ;VERIFIE L
    A PRESENCE D'UNE VIRGULE
180 PTRGET = $DFE3 ;TROUVE L'
    ADRESSE DE LA VARIABLE POINTEE PAR
    TXTPTR
181 IQERR = $E199 ;ILLEGAL Q
    UANTITY ERROR
182 GIVAYF = $E2F2 ;TRANSFORM
    E A,Y EN FLOTTANT DANS FAC
183 GARBAG = $E484 ;EFFECTUE
    UN FRE(0)
184 ; ET LONGUEUR DANS ACCU
185 FRESTR = $E5FD ;CHECK STR
    ING AND GET POINTER TO
186 ; STRING IN INDEX AND LENGTH IN ACC
    U
187 GETBYTC = $E6F5 ;SAUTE 1 C
    ARACTERE ET GETBYT
188 GETBYT = $E6F8 ;RECUPERE
    UN NOMBRE <256
189 GETADR = $E752 ;CONVERTI
    EN ENTIER DANS LINNUM
190 OVERFLOW = $E8D5 ;OVERFLOW

```

```

ERROR
191 SETFOR = $EB27 ;TRANSFERT
    FAC DANS VARIABLE TROUVE PAR PTRG
    ET
192 SETTXT = $FB39
193 PRNTAX = $F941 ;ENVOIE A,
    X EN HEXA
194 BASCALC = $FBC1 ;CALCUL AD
    RS LIGNE TEXTE
195 COUT = $FDED
196 MOVE = $FE2C ;DEPLACE (
    A1L) A (A2L) EN (A4L)
197 PAG
198 *
199 * INITIALISE POINTEUR DE L'AMPERSAN
    D
200 DEPART LDA $4C ;CODE DE J
    MP
201 STA AMPER
202 LDA $<DEBUT
203 STA AMPER+1
204 LDA $>DEBUT
205 STA AMPER+2
206 JMP INIT
207
208 DEBUTPRM = *
209 FRSTUND DA DEBUTPRM ;ADRESSE D
    EBUT PLACE LIBRE POUR FENETRES
210
211 *****
212 * INTEPRETEUR DE L'AMPERSAND *
213 *****
214
215 *
216 * RECHERCHE CODE DE LA COMMANDE
217 DEBUT PHA
218 JSR CHRGET ;AJUSTE TX
    TPTR
219 PLA
220 LDX $TBCMDADR-TBCMD
221 NEXTCMD CMP TBCMD-1,X
222 BEQ CMDFOUND
223 DEX
224 BNE NEXTCMD
225 JMP SYNERR
226 *
227 CMDFOUND DEX
228 TXA
229 ASL
230 TAX
231 LDA TBCMDADR+1,X
232 PHA
233 LDA TBCMDADR,X
234 PHA
235 RTS ;RETOUR PAR LA COMMAN
    DE
236
237 TBCMD DFB UTAB:,HTAB:,LET:,LOAD
    :,SAVE:,STORE:,RECALL:,HIMEM:
238 DFB 'I','C','O','F','E',
    'D','P'
239 TBCMDADR DA UTAB-1,HTAB-1,MENU-1,
    LOAD-1,SAVE-1,STORE-1,RECALL-1,HIM
    EM-1

```

```

240 DA INIT-1,CREE-1,OUVRE-1
    ,FERME-1,ECHANGE-1,DETRUIS-1
241 DA DEPLACE-1
242 PAG
243 PUT WINDOW1 ;GESTION F
    ENETRES
244 PAG
245 PUT WINDOW2 ;GESTION F
    ICHIERS FENETRES
246 ERR */$9600 ;FIN PROGR
    AMME EN DESSOUS $9600?
247 FINPRMG = *

```

## HCT.OBJ

```

1 *****
2 *
3 *      HARDCOPY POM'S
4 *
5 *****
6
7 CR> = $8D
8
9 CV = $25
10 BASL = $28
11
12 BASCALC = $FBC1
13 COUT = $FDED
14
15 ORG $0300
16
17 LDA $00
18 STA CV
19 H0304 LDY $00
20 JSR BASCALC
21 H0309 LDA (BASL),Y
22 CMP $40
23 BCS H031C ;FLASHING
    ET NORMAL
24 CMP $20
25 BCS H0319 ;SPECIAL C
    HARACTERS INVERSE
26 CLC ;UPPERCASE LETTERS IN
    VERSE
27 ADC $C0
28 JMP H031C
29 * PROBLEME AVEC UPPERCASE LETTERS
    CODE $80-$9F (CODE CONTROLE)
30 *
31 H0319 CLC
32 ADC $80
33 H031C JSR COUT
34 INY
35 CPY $40
36 BNE H0309
37 LDA $CR>
38 JSR COUT
39 INC CV
40 LDA CV
41 CMP $24
42 BNE H0304
43 RTS

```

## HCT.OBJ

\*300.331

```

0300- A9 00 85 25 A0 00 20 C1
0308- FB B1 28 C9 40 B0 0D C9
0310- 20 B0 06 18 69 C0 4C 1C
0318- 03 18 69 80 20 ED FD C8
0320- C0 28 D0 E5 A9 8D 20 ED

```

```

0328- FD E6 25 A5 25 C9 18 D0
0330- D3 60

```

## WINDOW

\*89EE.8A21

```

89EE- A9 4C
89F0- 8D F5 03 A9 02 8D F4 03

```

```

89F8- A9 8A 8D F7 03 4C 59 8A
8A00- 00 8A 48 20 B1 00 68 A2
8A08- 0F D0 20 8A F0 06 CA D0
8A10- F8 4C C9 DE CA 8A 0A AA
8A18- BD 31 8A 48 BD 30 8A 48
8A20- 60 A2

```

# Graphiques aléatoires

Thierry Han

Je vous propose un intéressant exercice de style avec quatre petits programmes d'animation graphique aléatoire. Trois sont en Basic et nous permettent de bien saisir le déroulement de l'animation. Le quatrième - le plus beau - suit la même démarche, mais en assembleur, démontrant en chemin comment réutiliser quelques routines mathématiques et graphiques de la ROM Applesoft.

Le principe commun (sauf CARRE) consiste à faire "rebondir" une ligne sur les limites d'un rectangle.

Un segment de droite se caractérise par ses extrémités, c'est-à-dire par les coordonnées de 2 points, soit X0, Y0 et X1, Y1.

Chaque boucle leur ajoute un déplacement horizontal et vertical, positif ou négatif (AX, AY et BX, BY), décalant une ligne de longueur variable sur l'écran.

Le "rebond" survient quand une extrémité de la ligne atteint la limite du rectangle alloué, ceci électivement pour le sens de déplacement rendant le point "hors-limites" (d'où modification de rotation).

En commun également le tirage au sort par RND(1) des extrémités de la droite, des déplacements et de HCOLOR (0 à 4).

Suivons pas à pas le programme LIGNES :

- 50 : haute résolution page 2
- 60-65 : tirage des coordonnées de 2 points
- 70-75 : tirage de leurs déplacements
- 80 : stockage de leur valeur absolue; servira à tester le dépassement
- 85 : on démarre en blanc
- 100 : traçage de la ligne définie en 60-65
- 110 : ajout des déplacements (positifs ou négatifs)
- 120-150 : si un point doit se retrouver hors-limite à l'itération suivante, alors inversion du déplacement et nouvelle couleur
- 200 : teste le clavier : une touche a été frappée si PEEK(-16384) > 127 (\$C000 >= \$80)

Si oui : lit le caractère frappé remet le clavier en attente (adresse -16368 = \$C010)

- si "E" (CHR\$(197)) : efface et recommence
- si ESC (CHR\$(155)) : fin
- si quelconque sauf SPACE : nouveaux points
- si SPACE (CHR\$(160)) : attend

qu'une touche ait été frappée

- 210-230 : boucle à la ligne 100 299 fois sur 300, sinon efface et recommence une fois sur 1200, sinon boucle en 60 avec de nouveaux points
- 240 : mode texte et fin

QUATRE choisit une ligne dans le quart supérieur gauche de l'écran (60-65) et la reproduit "en miroir" sur les 3 autres quarts (ligne 100).

CARRE utilise une seule variable X et trace à chaque itération 4 lignes, soit une ligne à une distance X de chaque bord d'un carré.

HUIT est en assembleur. Certes ! Mais il procède du même esprit et a exactement la même structure. De plus, par rapport à QUATRE, il construit dans chaque quart d'écran une image supplémentaire par retournement sur une diagonale et, à des fins de symétrie totale, considère l'écran comme étant carré avant de l'étirer aux dimensions de 280 \* 192 qui sont celles de la page graphique de ce cher Apple. (Notons que le rapport 280/192 peut se réduire à 35/24, valeurs apparaissant dans le programme.)

Le programme commence donc par fabriquer le nombre flottant stocké dans F sur cinq octets valant 35/24; vous le verrez réutilisé pour ajuster les abscisses.

Vous trouverez ci-après une brève description des principales adresses et routines utilisées :

- PAGE1 ou \$C054 sélectionne la page 1 (texte ou graphique)
- TEXTE ou \$C051 sélectionne le mode texte (page 1 ou 2)
- CLAV ou \$C000, si une touche a été frappée, est >= \$80 et contient le code de la touche frappée
- STRB ou \$C010 rend \$C000 < \$80
- HGR2 en \$F3D8 identique au Basic et de même pour RND (\$EFAE)
- HCOLOR en \$F6F0, appelé avec le code (0..7) de la couleur dans X, ajuste l'octet \$E4 utilisé par HPLT et HLINE. Le point d'entrée habituel, en \$F6EC, vérifie que X < 8
- HPLT en \$F457 trace un point; l'abscisse (0..279) est codée dans 2 octets, on charge l'octet haut dans Y, l'octet bas dans X et l'ordonnée dans A avant d'appeler HPLT



- HLINE en \$F53A comme en BASIC, octet haut de l'abscisse dans X, bas dans A et ordonnée dans Y

Un nombre réel est représenté par Applesoft sur 5 octets (forme compactée) ou, en page 0 pour être "mouliné", étendu sur 6 dans FAC (\$9D-A2) ou ARG (\$A5-AA).

- YTOFAC alias SGNFLT ou \$E301 convertit un entier de 0..255 chargé en Y en réel dans FAC
- INTTOFAC en \$E2F2 convertit un entier (-32768..+32767) chargé en A (partie basse) et Y (partie haute) en réel dans FAC
- FACTOINT en \$E10C aurait pu faire l'inverse. Eh bien non ! Il tronque, si possible, FAC en entier laissé dans 2 octets de FAC : \$A0 et \$A1
- FACTOMEM en \$EB2B compacte le réel dans FAC (6 octets) et le transfère à l'adresse pointée par Y (partie haute) et X (partie basse)
- FACTOARG en \$EB63 transfère FAC dans ARG
- FMULTM en \$E97F transfère un réel en mémoire pointé par Y (adresse haute) et A (adresse basse) dans ARG et appelle FMULT1
- FMULT1 en \$E982 multiplie FAC par ARG et laisse le résultat dans FAC - cette routine "prise en marche" demande l'exécution préalable du sous-programme GETSGN (à la fin de HUIT)
- FDIV en \$EA69 divise ARG par FAC et laisse le résultat dans FAC - nécessite GETSGN (même remarque que pour FMULT1)
- INTL et INTH sont les deux octets de FAC où FACTOINT laisse son résultat

Nous savons charger un nombre dans FAC, multiplier, diviser, tronquer, appeler RND et les routines graphiques. HUIT calcule 35/24, choisit ses points, ses vitesses de déplacement (1 à 4) et ses couleurs de façon identique aux programmes Basic, lit le clavier et boucle de façon analogue.

## Programme QUATRE

```
10 REM QUATRE
50 HGR2
60 X0 = RND (1) * 136:Y0 = RND (1) * 93
65 X1 = RND (1) * 136:Y1 = RND (1) * 93
70 AY = RND (1) * 2 + 1:AX = RND (1) *
  3 + 1
75 BY = RND (1) * 2 + 1:BX = RND (1) *
  3 + 1
80 QX = ABS (AX):QY = ABS (AY):RX = AB
  S (BX):RY = ABS (BY)
85 HCOLOR= 3
100 HPLLOT X0,Y0 TO X1,Y1: HPLLOT 279 - X0
  ,Y0 TO 279 - X1,Y1: HPLLOT X0,191 -
  Y0 TO X1,191 - Y1: HPLLOT 279 - X0
  ,191 - Y0 TO 279 - X1,191 - Y1
110 X0 = X0 + AX:Y0 = Y0 + AY:X1 = X1 + B
  X:Y1 = Y1 + BY
120 IF X0 < QX OR X0 > 139 - QX THEN AX
  = - AX: HCOLOR= RND (1) * 4
130 IF Y0 < QY OR Y0 > 95 - QY THEN AY =
  - AY: HCOLOR= RND (1) * 4
140 IF X1 < RX OR X1 > 139 - RX THEN BX
  = - BX: HCOLOR= RND (1) * 4
150 IF Y1 < RY OR Y1 > 95 - RY THEN BY =
  - BY: HCOLOR= RND (1) * 4
200 IF PEEK ( - 16384) > 127 THEN X =
  PEEK ( - 16384): POKE - 16368,0:
  ON X = 197 GOTO 50: ON X = 155 GOT
  O 240: ON X < > 160 GOTO 60: WAIT
  - 16384,128: POKE - 16368,0
210 IF RND (1) * 300 > 1 THEN 100
220 IF RND (0) * 1200 < 1 THEN 50
230 GOTO 60
240 TEXT : END
```

## Programme CARRE

```
10 REM CARRE
50 HGR2
60 HCOLOR= 3: GOTO 80
70 HCOLOR= RND (1) * 4
80 X = INT ( RND (1) * 48) * 2
100 HPLLOT X,0 TO X,191: HPLLOT X + 1,0 TO
  X + 1,191
110 HPLLOT 190 - X,0 TO 190 - X,191: HPLLO
  T 191 - X,0 TO 191 - X,191
```

## Source Big Mac

### Programme HUIT.S

```
1 *****
2 *GRAPHISMES ALEATOIRES
3 *PAR THIERRY HAN
4 *DATE: 21/7/84
5 *****
6 ;
7 PAGE1 EQU %C054
8 TEXTE EQU %C051
9 CLAV EQU %C000
10 STRB EQU %C010
11 ;
12 HGR2 EQU %F3D8
13 HCOLOR EQU %F6F0
14 HPLLOT EQU %F457
15 HLINE EQU %F53A
16 ;
17 FACTOMEM EQU %EB2B ;FAC =>
  MEM (Y,X)
```

```
120 HPLLOT 0,X TO 191,X: HPLLOT 0,X + 1 TO
  191,X + 1
130 HPLLOT 0,190 - X TO 191,190 - X: HPLLO
  T 0,191 - X TO 191,191 - X
200 IF PEEK ( - 16384) < 128 THEN 70
210 X = PEEK ( - 16384): POKE - 16368,0
  : ON X = 197 GOTO 50: ON X = 155 G
  OTO 230: ON X < > 160 GOTO 70
220 WAIT - 16384,128: POKE - 16368,0:
  GOTO 70
230 TEXT : END
```

## Programme LIGNES

```
10 REM LIGNES
50 HGR2
60 X0 = RND (1) * 276:Y0 = RND (1) * 18
  9
65 X1 = RND (1) * 276:Y1 = RND (1) * 18
  9
70 AY = RND (1) * 2 + 1:AX = RND (1) *
  3 + 1
75 BY = RND (1) * 2 + 1:BX = RND (1) *
  3 + 1
80 QX = ABS (AX):QY = ABS (AY):RX = AB
  S (BX):RY = ABS (BY)
85 HCOLOR= 3
100 HPLLOT X0,Y0 TO X1,Y1
110 X0 = X0 + AX:Y0 = Y0 + AY:X1 = X1 + B
  X:Y1 = Y1 + BY
120 IF X0 < QX OR X0 > 279 - QX THEN AX
  = - AX: HCOLOR= RND (1) * 4
130 IF Y0 < QY OR Y0 > 191 - QY THEN AY
  = - AY: HCOLOR= RND (1) * 4
140 IF X1 < RX OR X1 > 279 - RX THEN BX
  = - BX: HCOLOR= RND (1) * 4
150 IF Y1 < RY OR Y1 > 191 - RY THEN BY
  = - BY: HCOLOR= RND (1) * 4
200 IF PEEK ( - 16384) > 127 THEN X =
  PEEK ( - 16384): POKE - 16368,0:
  ON X = 197 GOTO 50: ON X = 155 GOT
  O 240: ON X < > 160 GOTO 60: WAIT
  - 16384,128: POKE - 16368,0
210 IF RND (1) * 300 > 1 THEN 100
220 IF RND (0) * 1200 < 1 THEN 50
230 GOTO 60
240 TEXT : END
```

```
18 YTOFAC EQU %E301 ;YREG =
  > FAC
19 FACTOINT EQU %E10C ;INT<FA
  C> => INTL/H
20 RND EQU %EFAE ;RND<FA
  C> => FAC
21 FACTOARG EQU %EB63 ;FAC =>
  ARG
22 FMULTM EQU %E97F ;MEM <Y
  ,A> => ARG: ARG*FAC => FAC
23 FMULT1 EQU %E982 ;ARG*FA
  C => FAC (NECESSITE GETSGN)
24 FDIV EQU %EA69 ;ARG/FA
  C => FAC (NECESSITE GETSGN)
25 INTTOFAC EQU %E2F2 ;Y/A =>
  FAC
26 ;
27 INTL EQU %A1
28 INTH EQU %A0
29 ;
30 ORG %F00 ;ZONE D
  E STOCKAGE
31 X0 DFS 1
```

```

32 Y0      DFS 1
33 X1      DFS 1
34 Y1      DFS 1
35 XX0     DFS 1
36 YY0     DFS 1
37 XX1     DFS 1
38 YY1     DFS 1
39 AX      DFS 1
40 AY      DFS 1
41 BX      DFS 1
42 BY      DFS 1
43 QX      DFS 4
44 YREG    DFS 1
45 XREG    DFS 1
46 F       DFS 5
          NTE 35/24
          ;CONSTA
47 ;
48          ORG $803
49 ;F=35/24
50          LDY #35
51          JSR YTOFAC
52          JSR FACTOARG
53          LDY #24
54          JSR YTOFAC
55          JSR GETSGN
56          JSR FDIV
57          LDX #F
58          LDY /F
59          JSR FACTOMEM
60 ;
61 ;DEBUT DU PROGRAMME
62 ;
63 INIT     LDX #3          ;HCOLOR
          =3
64          JSR HCOLOR
65          JSR HGR2
66 ;
67 ;CHOIX DES COORDONNEES (0-91)
68 ;
69 TIRAGE   JSR RND92
70          STX X0
71          JSR RND92
72          STX Y0
73          JSR RND92
74          STX X1
75          JSR RND92
76          STX Y1
77 ;
78 ;CHOIX DES VITESSES (1-4)
79 ;
80          JSR RND3
81          STX AX
82          JSR RND3
83          STX AY
84          JSR RND3
85          STX BX
86          JSR RND3
87          STX BY
88 ;
89 ;
90 ;VALEURS ABSOLUES
91 ;
92          LDX #0
93          CLC
94 ABS      LDA AX,X
95          BPL ABS.NEXT
96          EOR #$FF          ;INVERS
          ION SIGNE
97          ADC #1
98 ABS.NEXT STA QX,X
99          INX
100         CPX #4
101         BNE ABS
102 ;
103 ;
104 ;PLOTS DES 8 SEGMENTS
105 ;
106 TRACES   JSR TRACE4          ;QUATRE
          COINS
107         JSR RETOURNE          ;SYMETR
          IQUES
108         JSR TRACE4          ;PLOTS
109         JSR RETOURNE          ;REMISE
          EN PLACE
110 ;
111 ;
112 ;DEPLACEMENT DES POINTS
113 ;
114         LDX #0
115 DEPL     LDA X0,X
116         CLC
117         ADC AX,X          ;ADDITI
          ON
118         STA X0,X
119         INX
120         CPX #4
121         BNE DEPL
122 ;
123 ;
124 ;TEST DE DEPASSEMENT
125 ;
126         LDX #0
127 TEST     LDA #95
128         SEC
129         SBC QX,X
130         CMP X0,X
131         BCC INVERSE          ;X0>95-
          AX
132         LDA X0,X
133         CMP QX,X
134         BCS T.NEXT          ;X0>AX
135 INVERSE  LDA #0
136         SEC
137         SBC AX,X          ;INVERS
          E SIGNE
138         STA AX,X
139         STX XREG
140         JSR NEWCOL
141         LDX XREG
142 T.NEXT   INX
143         CPX #4
144         BNE TEST
145 ;
146 ;
147 ;TEST CLAVIER
148 ;
149         LDA CLAV
150         BPL DD          ;RIEN
151         BIT STRB
152         CMP #"E"          ;EFFACE
153         BNE ESCAPE
154 RET.I    JMP INIT
155 ESCAPE   CMP #$9B          ; = ESC
          ?
156         BEQ FIN
157         CMP #$A0          ;ESPACE
158         BNE NEWDIR          ;AUTRE
159 WAIT     LDA CLAV          ;ATTENT
          E
160         BPL WAIT
161         LDA STRB
162 ;
163 ;SI RND(1)*512=0, NOUVEAUX POINTS
164 ;
165 DO       LDY #1
166         JSR YTOFAC
167         JSR RND

```

```

168 JSR FACTOARG
169 LDA #2
170 LDY #0
171 JSR INTTOFAC ;512
172 JSR FMULT
173 JSR FACTOINT
174 LDA INTL
175 BEQ NEWDIR ;=0, NO
176 UVEAU JMP TRACES ;(<>0, B
    OUCLE
177 ;
178 ;SI RND(0)*2048<1, EFFACE
179 ;
180 NEWDIR LDY #0
181 JSR YTOFAC
182 JSR RND
183 JSR FACTOARG
184 LDA #8
185 LDY #0
186 JSR INTTOFAC ;2048
187 JSR FMULT
188 JSR FACTOINT
189 LDA INTL
190 BEQ RET.I ;=0, IN
    IT (EFFACE)
191 JSR NEWCOL ;(<>0, N
    OUVELLE COULEUR
192 JMP TIRAGE ;NOUVEA
    UX POINTS
193 ;
194 FIN BIT TEXTE
195 BIT PAGE1
196 RTS
197 ;
198 ;
199 TRACE4 LDA X0 ;HAUT-G
    AUCHE
200 STA XX0
201 LDA Y0
202 STA YY0
203 LDA X1
204 STA XX1
205 LDA Y1
206 STA YY1
207 JSR TRACE
208 ;
209 ;
210 ROITE SEC ;HAUT-D
211 LDA #191
212 SBC X0
213 STA XX0
214 LDA Y0
215 STA YY0
216 LDA #191
217 SBC X1
218 STA XX1
219 LDA Y1
220 STA YY1
221 JSR TRACE
222 ;
223 ;
224 UCHE SEC ;BAS-GA
225 LDA X0
226 STA XX0
227 LDA #191
228 SBC Y0
229 STA YY0
230 LDA X1
231 STA XX1
232 LDA #191
233 SBC Y1

```

```

234 STA YY1
235 JSR TRACE
236 ;
237 ;
238 SEC ;BAS-DR
    OITE
239 LDA #191
240 SBC X0
241 STA XX0
242 LDA #191
243 SBC Y0
244 STA YY0
245 LDA #191
246 SBC X1
247 STA XX1
248 LDA #191
249 SBC Y1
250 STA YY1
251 ;
252 ;
253 TRACE LDY XX0
254 JSR MULTF ;*F
255 LDA YY0 ;Y
256 LDX INTL ;X-LSB
257 LDY INTH ;X-MSB
258 JSR HPLLOT
259 LDY XX1
260 JSR MULTF ;*F
261 LDA INTL ;X-LSB
262 LDX INTH ;X-MSB
263 LDY YY1 ;Y
264 JMP HLINE
265 ;
266 ;
267 RETOURNE LDA X0 ;INVERS
    ION X-Y
268 LDX Y0
269 STA Y0
270 STX X0
271 LDA X1
272 LDX Y1
273 STA Y1
274 STX X1
275 RTS
276 ;
277 ;
278 ;Y=INT(RND(1)*92)
279 RND92 LDY #92
280 BNE RAND
281 ;Y=INT(RND(1)*3)+1
282 RND3 LDY #3
283 JSR RAND
284 INX
285 RTS
286 ;Y=INT(RND(1)*Y)
287 RAND STY YREG
288 LDY #1
289 JSR YTOFAC
290 JSR RND ;RND(1)
291 JSR FACTOARG
292 LDY YREG
293 JSR YTOFAC
294 JSR FMULT ;RND(1)
    *Y
295 JSR FACTOINT
296 LDX INTL
297 RTS
298 ;
299 ;FAC=INT(Y*35/24)
300 ;
301 MULTF JSR YTOFAC
302 LDA #F ;35/24
303 LDY /F

```

```

304 JSR FMULTM
305 JMP FACTOINT
306 ;
307 FMULT JSR GETSGN ;NECESS
      AIRE POUR FMULT1
308 JMP FMULT1
309 ;
310 GETSGN LDA $AA
311 EOR $A2

```

```

312 STA $AB
313 LDA $9D
314 RTS
315 ;
316 NEWCOL LDY #4
317 JSR RAND
318 JMP HCOLOR
319 ;
320 END

```

## HUIT

\*803.A20

```

0803- A0 23 20 01 E3
0808- 20 63 EB A0 18 20 01 E3
0810- 20 10 0A 20 69 EA A2 12
0818- A0 0F 20 28 EB A2 03 20
0820- F0 F6 20 D8 F3 20 D4 09
0828- 8E 00 0F 20 D4 09 8E 01
0830- 0F 20 D4 09 8E 02 0F 20
0838- D4 09 8E 03 0F 20 D8 09
0840- 8E 08 0F 20 D8 09 8E 09
0848- 0F 20 D8 09 8E 0A 0F 20
0850- D8 09 8E 0B 0F A2 00 18
0858- BD 08 0F 10 04 49 FF 69
0860- 01 9D 0C 0F E8 E0 04 D0
0868- EF 20 18 09 20 BB 09 20
0870- 1B 09 20 BB 09 A2 00 BD
0878- 00 0F 18 7D 08 0F 9D 00
0880- 0F E8 E0 04 D0 F1 A2 00
0888- A9 5F 38 FD 0C 0F DD 00
0890- 0F 90 08 BD 00 0F DD 0C
0898- 0F 80 12 A9 00 38 FD 08
08A0- 0F 9D 08 0F 8E 11 0F 20

```

```

08A8- 19 0A AE 11 0F E8 E0 04
08B0- D0 D4 AD 00 C0 10 1A 2C
08B8- 10 C0 C9 C5 D0 03 4C 1D
08C0- 08 C9 98 F0 4F C9 A0 D0
08C8- 28 AD 00 C0 10 FB AD 10
08D0- C0 A0 01 20 01 E3 20 AE
08D8- EF 20 63 EB A9 02 A0 00
08E0- 20 F2 E2 20 0A 0A 20 0C
08E8- E1 AD A1 00 F0 03 4C 69
08F0- 08 A0 00 20 01 E3 20 AE
08F8- EF 20 63 EB A9 08 A0 00
0900- 20 F2 E2 20 0A 0A 20 0C
0908- E1 AD A1 00 F0 00 20 19
0910- 0A 4C 25 08 2C 51 C0 2C
0918- 54 C0 60 AD 00 0F 8D 04
0920- 0F AD 01 0F 8D 05 0F AD
0928- 02 0F 8D 06 0F AD 03 0F
0930- 8D 07 0F 20 97 09 38 A9
0938- BF ED 00 0F 8D 04 0F AD
0940- 01 0F 8D 05 0F A9 BF ED
0948- 02 0F 8D 06 0F AD 03 0F
0950- 8D 07 0F 20 97 09 38 AD
0958- 00 0F 8D 04 0F A9 BF ED
0960- 01 0F 8D 05 0F AD 02 0F

```

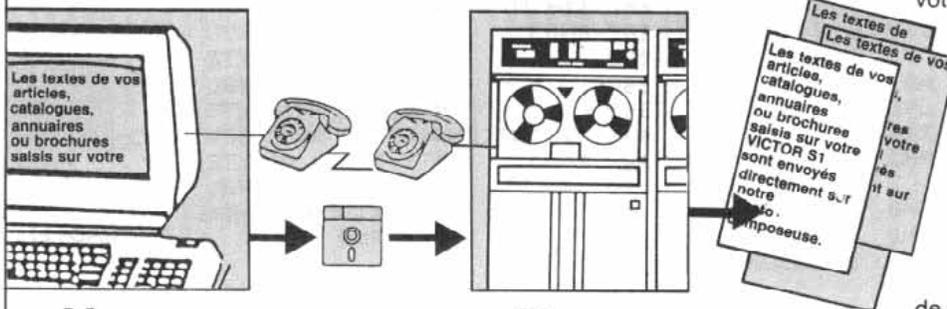
```

0968- 8D 06 0F A9 BF ED 03 0F
0970- 8D 07 0F 20 97 09 38 A9
0978- BF ED 00 0F 8D 04 0F A9
0980- BF ED 01 0F 8D 05 0F A9
0988- BF ED 02 0F 8D 06 0F A9
0990- BF ED 03 0F 8D 07 0F AC
0998- 04 0F 20 FD 09 AD 05 0F
09A0- AE A1 00 AC A0 00 20 57
09A8- F4 AC 06 0F 20 FD 09 AD
09B0- A1 00 AE A0 00 AC 07 0F
09B8- 4C 3A F5 AD 00 0F AE 01
09C0- 0F 8D 01 0F 8E 00 0F AD
09C8- 02 0F AE 03 0F 8D 03 0F
09D0- 8E 02 0F 60 A0 5C D0 07
09D8- A0 03 20 DF 09 E8 60 8C
09E0- 10 0F A0 01 20 01 E3 20
09E8- AE EF 20 63 EB AC 10 0F
09F0- 20 01 E3 20 0A 0A 20 0C
09F8- E1 AE A1 00 60 20 01 E3
0A00- A9 12 A0 0F 20 7F E9 4C
0A08- 0C E1 20 10 0A 4C 82 E9
0A10- A5 AA 45 A2 85 AB A5 9D
0A18- 60 A0 04 20 DF 09 4C F0
0A20- F6

```

# Vos textes en direct de votre ordinateur à nos photocomposeuses

## Gain de temps et économie



**Vous**

**Nous**

Les textes de vos articles, catalogues, annuaires ou brochures saisis sur votre micro-ordinateur sont envoyés directement sur notre photocomposeuse.

Nous vous évitons ainsi, le coût et le temps de la saisie supplémentaire que nécessite le traitement traditionnel de la photocomposition avant l'impression des documents.

Si vous le désirez nous pouvons également nous charger de l'impression et du brochage.

# TELECOMPO (1) 328.18.63

PHOTOCOMPOSITION - BUREAUTIQUE - TRANSMISSION DE DONNÉES - GESTION DE FICHIERS - MATÉRIELS DE TRAITEMENT DE TEXTES  
13 et 15, avenue du Petit-Parc - 94300 VINCENNES

Une référence : la revue  
**pom's**



# Récupérez les icônes du système

Marianne Sutz

Le fichier "système", situé sur une disquette d'amorçage, comporte quatre icônes fréquemment utilisés pour accompagner les messages d'erreur. Voici une petite routine en langage machine, qui autorise la récupération de ces icônes depuis un programme Basic.

## Utilisation de la routine

Le code objet doit, comme à l'accoutumée, être placé dans un tableau de

variables. Ensuite, il faut prévoir un tableau de 66 variables entières pour la définition de l'icône à récupérer (un icône est constitué de 32 lignes de 32 pixels; il faut donc 64 mots de 16 bits pour la définition, plus les deux mots indiquant la taille, puisque l'icône sera affiché avec l'instruction PUT). Pour le passage des paramètres depuis le Basic vers la routine, il faut utiliser :

```
CALL ADR!(ID%,VARPTR(T%(0)))
```

avec la première version du Basic Microsoft; ADR! ID%,VARPTR(T%(0)) avec la version actuelle.

ADR! représente l'adresse de la routine.

ID% représente l'identificateur de l'icône (-6047, 0, 1 ou 2).

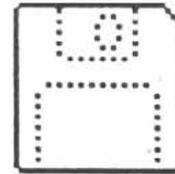
VARPTR(T%(0)) passe à la routine l'adresse de base du tableau de variables destiné à recevoir l'icône.

## 10 ' PROGRAMME DE DEMONSTRATION

```

20 '
30 CLS:DEFINT A-Z:DIM C(18),T(65)
40 '
50 ' Code objet à implanter en mémoire
60 '
70 DATA &h4E56,0,&h42A7,&h3F2E,&hC,&hA9BB
80 DATA &h2057,&h2250,&h246E,&h8,&h24FC,32
90 DATA 32,&H701F,&H24D9,&H51C8,&HFFFC
100 DATA &H4E5E,&H4E75
110 '
120 ' Identificateurs des icônes
130 '
140 DATA 0,1,2,-6047
150 '
160 ' Implantation du code objet en mémoire
170 '
180 FOR I=0 TO 18:READ C(i):NEXT
190 D=0:GOSUB 170
200 RESTORE 80
210 D=100:GOSUB 170
220 GOTO 150
230 ' S.p. chargement et/ou affichage des icônes
240 FOR P=50 TO 200 STEP 50:READ ID
250 A!=VARPTR(C(0)):CALL A!(ID,VARPTR(T(0)))
260 PUT(P+D,P),T:NEXT
270 RETURN

```



ID=-6047



ID=0



ID=1



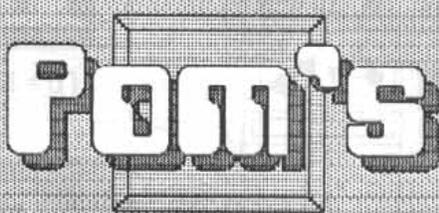
ID=2

0000  
0000  
0000

; Cette routine charge un icône appartenant au "fichier système" (ressource) en mémoire (si ce n'est déjà fait) et transfère sa définition depuis la zone de mémoire choisie par le système vers un tableau de variables.

0000					
0000		.Trap	_GetIcon	\$A9BB	
0000					
0000	4E56	0000	LINK	A6,*0	
0004	42A7		CLR.L	-(SP)	; Réserve 4 octets au sommet de la pile.
0006		:			; L'adresse du pointeur sur la définition
0006		:			; de l'icône sera retournée à cet endroit.
0006	3F2E	000C	MOVE	12(A6),-(SP)	; L'identifiant de l'icône, passé depuis
000A		:			; le programme Basic, est empilé (2
000A		:			; octets).
000A	A9BB		_GetIcon		; Chargement en mémoire de la définition
000C		:			; de l'icône et retour de l'adresse du
000C		:			; pointeur à l'emplacement réservé
000C		:			; précédemment.
000C	2057		MOVEA.L	(SP),A0	; L'adresse du pointeur est placée dans
000E		:			; le registre d'adresse A0.
000E	2250		MOVEA.L	(A0),A1	; Le registre A0 contient l'adresse où se
0010		:			; trouve l'adresse du premier des octets
0010		:			; formant la définition de l'icône
0010		:			; (indirection). Après cette instruction,
0010		:			; l'adresse du premier octet se trouve
0010		:			; dans le registre d'adresse A1.
0010	246E	0008	MOVEA.L	8(A6),A2	; L'adresse de base du tableau de 66
0014		:			; variables entières est placée dans A2.
0014	24FC	00200020	MOVE.L	*\$00200020,(A2)+	; Les deux premiers éléments du tableau
001A		:			; de variables sont initialisés à 32 (\$20)
001A		:			; car les icônes font 32*32 pixels. Ceci
001A		:			; permet l'utilisation de l'instruction
001A		:			; PUT dès le retour au programme Basic.
001A	701F		MOVEQ	*31,D0	; 128 octets, soit 32 mots de 32 bits
001C	24D9	Boucle	MOVE.L	(A1)+,(A2)+	; sont nécessaires à la représentation
001E	51C8	FFFC	DBRA	D0,Boucle	; d'un icône. Le compteur de boucle D0
0022	4E5E		UNLK	A6	; est initialisé à 31 (la sortie de boucle
0024	4E75		RTS		; a lieu lorsque D0 est égal à -1).

Basic Pascal Assembleur Initiation Informations...



**Pom's**

La revue francophone  
des utilisateurs du  
Macintosh

Sur la disquette Macintosh  
numéro 15/16

vous trouverez les programmes Basic "Editeur", "Mac/AppleII", "Paint/Basic", "Paint/Start", et d'autres programmes "exemples" employés pour l'illustration des articles publiés dans les numéros 15 et 16 de Pom's.

Les polices de Caractères "Los angeles" 24 et 12 points, "CAIRO", "Mos Elsley" 24 et 12 points, "Hollywood" et "Manhattan".

Les programmes :

- LOCALIZER : configuration du clavier AZERTY pour la France (ou un autre pays).
- Copie de disque : pour effectuer des copies de disquettes avec un seul lecteur sur un Macintosh 128Ko, en seulement quatre passages.

150,00 F TTC franco

(bon de commande page 78)



# Modification des informations pour le "Finder"

Jean-Luc Bazanegue

Dans le précédent numéro de Pom's, nous vous proposons un programme permettant d'afficher ou de lister les informations sur les fichiers. Nous avons pensé qu'il pourrait être intéressant de pouvoir manipuler directement ces informations; c'est l'objet de ce programme. On ne peut bien sûr pas tout modifier, puisqu'il serait dangereux de toucher aux indications sur la position physique ou logique des fichiers, mais ce programme autorise la modification du nom, du type, du nom de créateur ainsi que de divers paramètres liés à l'apparence et à la position de l'icône représentant le fichier.

## Mode d'emploi

Ce programme fonctionne avec la version 2 du Basic Microsoft, et avec toutes les configurations (Macintosh 128 et 512Ko, avec ou sans lecteur de disquettes externe). Toutefois, avec un Mac 128Ko, il est préférable de ne pas entrer les REMARQUES; ceci autorise une vitesse d'exécution accrue (le programme situé sur la disquette Mac numéro 18 ne comporte pas de remarques). Pour ceux d'entre-vous qui ne disposent pas encore de la nouvelle version du Basic, signalons que les routines en langage machine fonctionnent aussi avec l'an-

cienne version; le programme est donc facilement adaptable.

### Modification du nom de fichier

Un nom de fichier peut contenir jusqu'à 63 caractères et, bien sûr, ne doit pas être de longueur nulle. Enfin, il ne peut pas comporter le caractère ":", qui est utilisé par le système comme séparateur entre le nom du volume et le nom de fichier. Le programme intercepte toutes les erreurs possibles. D'une manière générale, il vaut mieux s'abstenir de modifier les noms des fichiers "systèmes", afin de ne pas perturber le fonctionnement du système d'exploitation. Si cela devait arriver, il n'y aurait rien de grave puisque ce que fait le programme n'est pas irréversible.

### Modification du type de fichier

Le type de fichier est constitué de quatre caractères ASCII. Si vous entrez une chaîne comportant moins de quatre caractères, les emplacements laissés libres seront remplacés par un octet nul. Dans la zone de saisie, un octet nul est remplacé par un point afin de visualiser sa présence.

### Modification du nom de créateur

Les remarques au sujet du type de fichier sont aussi valables ici. Il

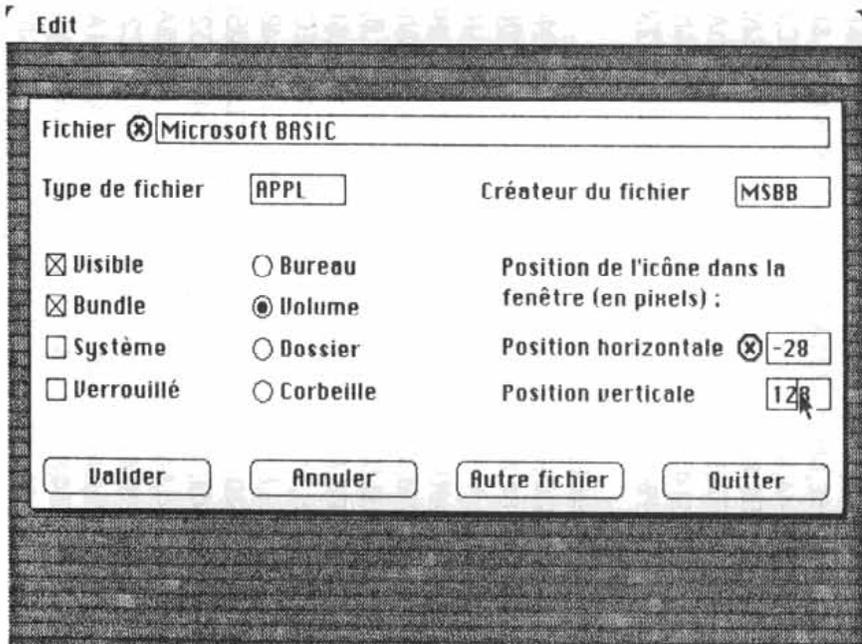
convient de signaler que le nom du créateur est utilisé par le "Finder" pour déterminer le programme d'application à partir duquel est issu le fichier. Par exemple, un fichier de caractères issu du "Font Mover" a, comme nom de créateur, "FMOV". Si l'on modifie cette chaîne, une tentative d'ouverture du fichier provoquera l'affichage du message "Il n'y a pas d'application pour ouvrir ce document".

### Position horizontale et verticale de l'icône

Si l'icône du fichier se trouve sur le bureau, la position, en pixels, est relative à l'angle supérieur gauche de l'écran. Dans le cas où l'icône est situé dans la fenêtre d'un dossier (ou la fenêtre du volume), la position est relative à l'angle supérieur gauche de la fenêtre au moment de sa création.

### Paramètres divers

Il est aussi possible de rendre un fichier invisible, de le placer sur le bureau alors qu'il se trouvait dans une fenêtre, etc... Nous ne nous étendons pas sur ces points, expliqués en détails dans le précédent numéro de Pom's.



## Pour reprendre la main...

Il vous est certainement arrivé, lors de l'écriture ou de la mise au point d'un programme Basic, et alors que vous aviez supprimé des menus, de ne pas pouvoir reprendre la main. Ceci est fort désagréable lorsque l'on a fait de nombreuses modifications, car il n'est pas possible de les sauvegarder. Si cette mésaventure devait vous arriver, sachez qu'il existe une autre solution que le "Reset". Il faut appuyer sur la touche "Interruption"; ceci a pour effet d'afficher le célèbre message "Une grave erreur système...". Cliquez ensuite sur "Reprendre" (ou "Resume"); la fenêtre de sortie s'affiche mais l'on ne peut toujours pas reprendre la main. Si l'on renouvelle l'opération, on obtient l'affichage de la fenêtre de commande. Il suffit maintenant d'entrer la commande "MENU RESET" pour que tout rentre dans l'ordre.

## \* Modifications des informations pour le "Finder"

```

WINDOW CLOSE 1:DEFINT A-Z:ON ERROR GOTO Er
DIM TA(16),TB(16),TC(184),TG(3543),AN(4),TCO(229):N=0
ON DIALOG GOSUB Dial:ON MOUSE GOSUB Mous
LI=119: Point d'entrée dans le tableau TC pour la routine L:Info
EI=149: Point d'entrée pour la routine E:Info
ME=177: Point d'entrée pour la routine E:Menu

' Data pour le code objet
DATA &h226E,8,&h723E,&h14E9,1,&h2429,1,&h282,&hFF,&hFFFF,&h2642
DATA &h14DB,&h51C9,&hFFFC,&h4E75,&h4E56,0,&h41FA,&hFFFC,&h2248
DATA &h7013,&h4299,&h51C8,&hFFFC,&h45FA,&hFF4E,&h214A,&h12
DATA &h226E,&hC,&h721A,&h61C6,&h5368,&h1C,&hA007,&h45FA,&hFF38
DATA &h214A,&h12,&h61B0,&h4268,&h1C,&hA00C,&h4E5E,&h4E75
DATA &h4E56,0,&h41FA,&hFED0,&h43FA,&hFF1C,&h2149,&h12,&hA00D
DATA &h45FA,&hFF52,&h214A,&h001C,&h618A,&hA00B,&hA035,&h363C
DATA 1,&h3143,&h16,&hA00F,&h363C,2,&h3828,&h10,&h66F0,&h4E5E
DATA &h4E75,&h4E56,0,&h3F2E,8,&hA936,&hA937,&h4E5E,&h4E75

' Data pour boutons style Multiplan
DATA &hF,&hF,&hFE0,&h1FF0,&h3838,&h701C,&hE44E,&hCEE6,&hC7C6
DATA &hC386,&hC7C6,&hCEE6,&hE44E,&h701C,&h3838,&h1FF0,&hFE0

' Data pour petit insecte utile, car il se nourrit de pucerons
DATA 64,57,0,&h1000,&h10,0,0,&h800,&h20,0,0,&h400,&h40,0,0,&h400
DATA &h40,0,&h0,&h400,&h40,0,0,&h40F,&hE040,0,&h78,&h630,&h18C0
DATA &h3C00,&h86,&h5C0,&h740,&hC200,&h101,&h241,&h481,&h100
DATA &h138,&h8111,&h1102,&h3900,&h124,&h4E0D,&h66E4,&h4900
DATA &h122,&h3233,&h9898,&h8900,&h121,&h4409,&h2045,&h900,&h91
DATA &h841D,&h7043,&h1200,&h92,&h341D,&h7056,&h9200,&h94,&h40F
DATA &hE040,&h5200,&h3C95,&h418,&h3041,&h5278,&h4A98,&h5220
DATA &h894,&h32A4,&hB590,&h220,&h880,&h335A,&h9CDD,&h1A0,&hB00
DATA &h1672,&h9B04,&h1E0,&hF00,&h81B2,&h8C21,&h40F0,&h1E04
DATA &h862,&hB060,&h147F,&hF890,&hC1A,&h81A0,&hF,&hE000,&h4B02
DATA &h7E24,&h100,1,&h8FC,&h21,&h4A,&h4814,&h800,&h20,&h4000
DATA &h40,&h4800,&h20,&h1400,&h100,&h800,&h14,&h155,&h5400

```

```

DATA &h9000,&h31,0,&hA,&h1800,&h58,&h5000,&h150,&h3400,&hA8
DATA &h555,&h5400,&h2A00,&h144,0,&h2A,&h6500,&hA6,&hA000,&h80
DATA &hEA00,&h57,&hA00,&h201,&hD400,&hAAAAB,&h8CAA,&hA81F
DATA &hEAAA,&h5555,&hF000,&h83,&hD554,&hA554,&hE208,&h1011
DATA &hAAAA,&h5515,&h47E2,&hA7F8,&hD154,&h2A2A,&h8FF0,&hFF4
DATA &hA8A8,&h15,&h17FF,&hFFE4,&hD000,&hB,&h2BFE,&hFFC3,&hA000
DATA 5,&h35FF,&h7FD1,&h4000,3,&h1AFE,&hB022,&hA000,1,&h8F80
DATA &h61C5,&h4000,0,&hC702,&hC00A,&h8000,1,&h6041,&h40B5,0,0
DATA &hBF81,&h6042,&h8000,1,&h6203,&h7022,0,0,&h4406,&h3801,0,0
DATA &h800C,&h1E01,0,0,&h8038,&hF01,0,1,&hF0,&h7C1,0,1,&h1E0
DATA &hF2,0,1,&hF80,&h7C,0,0,&hFE00,0,0,0,&h7800,0,0

```

```

' Data pour annulation des menus
DATA 1,2,4,5,6

```

```

' Initialisation des tableaux

```

```

FOR I=104 TO 184:READ TC(I):NEXT:FOR I=0 TO 16:READ TB(I):NEXT
FOR I=0 TO 229:READ TCO(I):NEXT
TA(0)=15:TA(1)=15:FOR I=2 TO 16:TA(I)=0:NEXT

```

```

' Annulation des menus (routine E:Menu)

```

```

AI=VARPTR(TC(0)):FOR I=0 TO 4:READ N:A:I:NEXT

```

```

' Choix du premier fichier et initialisation de la fenêtre de sortie

```

```

GOSUB Cfich

```

```

WINDOW 1,(14,50)-(498,267),4:TEXTFONT 0

```

```

MOVETO 7,20:PRINT "Fichier":MOVETO 7,52:PRINT "Type de fichier"

```

```

MOVETO 268,52:PRINT "Créateur du fichier"

```

```

MOVETO 280,92:PRINT "Position de l'icône dans la"

```

```

MOVETO 280,108:PRINT "fenêtre (en pixels):"

```

```

MOVETO 280,134:PRINT "Position horizontale"

```

```

MOVETO 280,158:PRINT "Position verticale"

```

```

BUTTON 9,0,"Valider",(7,190)-(107,210)

```

```

BUTTON 10,0,"Annuler",(130,190)-(230,210)

```

```

BUTTON 11,1,"Autre fichier",(253,190)-(353,210)

```

```

BUTTON 12,1,"Quitter",(376,190)-(476,210)

```

```

GOSUB Linf:GOSUB Init:IALOG ON:MOUSE ON

```

```

10 GOTO 10

```

```

Dial:

```

```

' Sous programme pour événement de type "Dialog"

```

```

MOUSE OFF:DIALOG OFF
D0=DIALOG(0):IF D0=1 THEN GOSUB Bout:GOTO Rep
IF D0=2 THEN GOSUB ChDi:SA=SA2:GOTO Rep
IF D0=6 OR D0=7 THEN GOSUB Retu
Rep:
DIALOG ON:MOUSE ON
RETURN

Mous:
' Sous programme pour événement de type "Mouse" (bouton style Multiplan)
DIALOG OFF:MOUSE OFF:J=MOUSE(0):MX=MOUSE(1):MY=MOUSE(2)
IF MX>58 AND MY<73 AND MY>8 AND MY<23 THEN MA=0:GOTO Dmous
IF MX>114 AND MY<129 AND MY>40 AND MY<55 THEN MA=1:GOTO Dmous
IF MX>403 AND MY<418 AND MY>40 AND MY<55 THEN MA=2:GOTO Dmous
IF MX>421 AND MY<436 AND MY>122 AND MY<137 THEN MA=3:GOTO Dmous
IF MX>421 AND MY<436 AND MY>146 AND MY<161 THEN MA=4:GOTO Dmous
GOTO Rmous
Dmous:
ON MA+1 GOSUB Mous0,Mous1,Mous2,Mous3,Mous4
Rmous:
DIALOG ON:MOUSE ON
RETURN

Bout:
' Sous programme utilisé en cas d'action sur un bouton
D1=DIALOG(1)
IF D1<9 THEN GOSUB BoDi:GOSUB ChDi2:GOSUB Bval:RETURN
ON D1-8 GOSUB Vali,Ann,Aut,Qui
RETURN

Ann:
' Annulation des modifications apportées au fichier (avant validation)
PUT(58,8),TA,PSET:PUT(114,40),TA,PSET:PUT(403,40),TA,PSET
PUT(421,122),TA,PSET:PUT(421,146),TA,PSET
FOR I=0 TO 4:AN(I)=0:NEXT:GOSUB Init
RETURN

BoDi:
' Boutons autres que "Valider", "Annuler", "Autre fichier" et "Quitter"

```

```

IF D1<5 THEN BoDi2
IF D1<7 THEN BoDi1
IF BUTTON(7)=2 THEN RETURN
IF BI THEN GOSUB win2:GOSUB ErD:RETURN
BoDi1:
BUTTON B,1:BUTTON D1,2:B=D1:RETURN
BoDi2:
IF BUTTON(D1)=1 THEN BUTTON D1,2 ELSE BUTTON D1,1
RETURN

Aut:
' Autre fichier
GET(42,8)-(405,161),TG:GOSUB Cfich:PUT(42,8),TG,OR:GOSUB Ann
RETURN

Vali:
' Validation et mise à jour (routines L:Info et E:Info)
DM=0:GOSUB D11:IF DM THEN RETURN
DM=0:GOSUB D12:IF DM THEN RETURN
DM=0:GOSUB D13:IF DM THEN RETURN
DM=0:GOSUB D14:IF DM THEN RETURN
DM=0:GOSUB D15:IF DM THEN RETURN
AI=VARPTR(TC(16)):FOR I=0 TO 3:POKE AI+I,0:NEXT
FOR I=1 TO LEN(T2$):POKE AI+I-1,ASC(MID$(T2$,I,1)):NEXT
AI=VARPTR(TC(18)):FOR I=0 TO 3:POKE AI+I,0:NEXT
FOR I=1 TO LEN(C2$):POKE AI+I-1,ASC(MID$(C2$,I,1)):NEXT
TC(22)=PH:TC(21)=PV
IF B=8 THEN TC(23)=-3 ELSE IF B=5 THEN TC(23)=-2
IF B=6 THEN TC(23)=0
TC(20)=TC(20) AND &hfff
IF BUTTON(1)=1 THEN TC(20)=TC(20) OR &h4000
IF BUTTON(2)=2 THEN TC(20)=TC(20) OR &h2000
IF BUTTON(3)=2 THEN TC(20)=TC(20) OR &h1000
IF BUTTON(4)=2 THEN TC(20)=TC(20) OR &h8000
Mj2:
AI=VARPTR(TC(EI)):AI VARPTR(F2$)
IF TC(8) THEN Mj2
F$=F2$
Mj:

```

```

AI=VARPTR(TC(L1)):AI VARPTR(V$),VARPTR(F$)
IF TC(8) THEN MJ ELSE GOSUB Linf:GOSUB Ann
RETURN

Retu:
' Pression sur la touche "Return" ou "Tab"
GOSUB ChDi2
IF SA=5 THEN SA=1 ELSE SA=SA+1
EDIT FIELD SA
RETURN

Cfich:
' Lecture des informations sur un fichier (routine L.Info)
F$=FILE$(1):IF F$="" THEN Qui
L=LEN(F$):FOR I=1 TO L:IF MID$(F$,I,1)=":" THEN N=1
NEXT:V$=LEFT$(F$,N):F$=RIGHT$(F$,L-N)
Cfich2:
AI=VARPTR(TC(L1)):AI VARPTR(V$),VARPTR(F$)
IF TC(8) THEN Cfich2 ELSE GOSUB Linf
RETURN

ChDi:
' Passage d'une zone de saisie à l'autre (contrôle de validité)
SA2=DIALOG(2)
ChDi2:
ON SA GOSUB Di1,Di2,Di3,Di4,Di5:RETURN
Di1:
' Contrôle de la zone "Nom de fichier"
F2$=EDIT$(1):L=LEN(F2$)
IF L=0 OR L>63 THEN GOSUB Win2:GOSUB ErE1:DM=-1:RETURN
DC=0:FOR I=1 TO LEN(F2$):IF MID$(F2$,I,1)=":" THEN DC=-1
NEXT:IF DC THEN GOSUB Win2:GOSUB ErC:DM=-1:RETURN
IF F2$<>F$ THEN PUT(58,8),TB,PSET:AN(0)=-1:GOSUB Bval
RETURN
Di2:
' Contrôle de la zone "Type de fichier"
T2$=EDIT$(2):L=LEN(T2$)
IF L>4 THEN GOSUB Win2:GOSUB ErE2:DM=-1:RETURN
IF T2$<>T$ THEN PUT(114,40),TB,PSET:AN(1)=-1:GOSUB Bval

```

---

```

RETURN
Di3:
' Contrôle de la zone "Créateur du fichier"
C2$=EDIT$(3):L=LEN(C2$)
IF L>4 THEN GOSUB Win2:GOSUB ErE3:DM=-1:RETURN
IF C2$<>C$ THEN PUT(403,40),TB,PSET:AN(2)=-1:GOSUB Bval
RETURN
Di4:
' Contrôle de la zone "Position horizontale"
4 PH2$=EDIT$(4):PH=VAL(PH2$)
DE=0:FOR I=2 TO LEN(PH2$):M$=MID$(PH2$,I,1)
IF M$<"0" OR M$>"9" THEN DE=-1
NEXT:M$=LEFT$(PH2$,1)
IF M$<"-" AND M$<" " THEN IF M$<"0" OR M$>"9" THEN DE=-1
IF DE THEN GOSUB Win2:GOSUB ErE4:DM=-1:RETURN
IF PH2$<>PH$ THEN PUT(421,122),TB,PSET:AN(3)=-1:GOSUB Bval
Di42:
RETURN
Di5:
' Contrôle de la zone "Position verticale"
5 PV2$=EDIT$(5):PV=VAL(PV2$)
DE=0:FOR I=2 TO LEN(PV2$):M$=MID$(PV2$,I,1)
IF M$<"0" OR M$>"9" THEN DE=-1
NEXT:M$=LEFT$(PV2$,1)
IF M$<"-" AND M$<" " THEN IF M$<"0" OR M$>"9" THEN DE=-1
IF DE THEN GOSUB Win2:GOSUB ErE5:DM=-1:RETURN
IF PV2$<>PV$ THEN PUT(421,146),TB,PSET:AN(4)=-1:GOSUB Bval
Di52:
RETURN
Linf:
' Lecture et mise en forme des informations retournées par L.Info
T$=MKI$(TC(16))+MKI$(TC(17))
FOR I=1 TO 4:IF ASC(MID$(T$,I,1))=0 THEN MID$(T$,I,1)="-":
NEXT
C$=MKI$(TC(18))+MKI$(TC(19))
FOR I=1 TO 4:IF ASC(MID$(C$,I,1))=0 THEN MID$(C$,I,1)="-":
NEXT
PH$=STR$(TC(22)):IF TC(22)>-1 THEN PH$=RIGHT$(PH$,LEN(PH$))-1

```

```
PV$=STR$(TC(21)):IF IC(21)>-1 THEN PV$=RIGHT$(PV$,LEN(PV$)-1)
RETURN
```

Init:

```
' Initialisation des zones de saisie et boutons
EDIT FIELD 1,F$(77,8)-(475,22)
EDIT FIELD 2,T$(133,40)-(186,54)
EDIT FIELD 3,C$(422,40)-(475,54)
EDIT FIELD 4,PH$(440,122)-(475,136)
EDIT FIELD 5,PV$(440,146)-(475,160)
BUTTON 1,2+((&H4000 AND TC(20))=&H4000),"Visible",(7,80)-(67,95),2
BUTTON 2,2+((&H2000 AND TC(20))=0),"Bundle",(7,102)-(58,117),2
BUTTON 3,2+((&H1000 AND TC(20))=0),"Système",(7,124)-(80,139),2
BUTTON 4,2+((&H8000 AND TC(20))=0),"Verrouillé",(7,146)-(88,161),2
BUTTON 5,1,"Bureau",(130,80)-(193,95),3
BUTTON 6,1,"Volume",(130,102)-(195,117),3
BUTTON 7,1,"Dossier",(130,124)-(196,139),3
BUTTON 8,1,"Corbeille",(130,146)-(205,161),3
BI=-1
IF TC(23)=-3 THEN BUTTON 8,2:B=8:GOTO Init2
IF TC(23)=-2 THEN BUTTON 5,2:B=5:GOTO Init2
IF TC(23)=0 THEN BUTTON 6,2:B=6 ELSE BUTTON 7,2:B=7:BI=0
```

Init2:

```
EDIT FIELD 1:SA=1:BUTTON 9,0:BUTTON 10,0:DV=0
RETURN
```

Mous0:

```
' Action sur le bouton d'annulation de la zone "Nom du fichier"
IF NOT AN(0) THEN RETURN
AN(0)=0:EDIT FIELD 1,F$(77,8)-(475,22):PUT(58,8),TA,PSET:SA=1
RETURN
```

Mous1:

```
' Action sur le bouton d'annulation de la zone "Type de fichier"
IF NOT AN(1) THEN RETURN
AN(1)=0:EDIT FIELD 2,T$(133,40)-(186,54):PUT(114,40),TA,PSET
SA=2
RETURN
```

Mous2:

```
' Action sur le bouton d'annulation de la zone "Créateur du fichier"
```

```
IF NOT AN(2) THEN RETURN
AN(2)=0:EDIT FIELD 3,C$(422,40)-(475,54):PUT(403,40),TA,PSET
SA=3
RETURN
```

Mous3:

```
' Action sur le bouton d'annulation de la zone "Position horizontale"
IF NOT AN(3) THEN RETURN
AN(3)=0:EDIT FIELD 4,PH$(440,122)-(475,136):PUT(421,122),TA,PSET
SA=4
RETURN
```

Mous4:

```
' Action sur le bouton d'annulation de la zone "Position verticale"
IF NOT AN(4) THEN RETURN
AN(4)=0:EDIT FIELD 5,PV$(440,146)-(475,160)
PUT(421,146),TA,PSET:SA=5
RETURN
```

Bval:

```
' Validation des boutons "Valider" et "Annuler"
IF NOT DV THEN BUTTON 9,1:BUTTON 10,1:DV=-1
RETURN
```

Er:

```
' En cas de position horizontale ou verticale < à -32768 ou > à 32767
GOSUB win2
IF ERL=4 THEN GOSUB ErE4:RESUME Di42
GOSUB ErE5:RESUME Di52
```

Win2:

```
' Affichage de la fenêtre utilisée pour les messages d'erreurs
GET(42,8)-(405,161),TG:WINDOW 2,(100,90)-(412,190),2
TEXTFONT 0:BUTTON 13,1,"OK",(238,70)-(302,90)
RETURN
```

ErE1:

```
' Message pour erreur dans la saisie du nom de fichier
' (plus de 63 ou 0 caractères)
MOVETO 10,35:PRINT "Le nom de fichier doit contenir
MOVETO 10,55:PRINT "au minimum 1 et au
```

```

MOVETO 10,75:PRINT "maximum 63 caractères...
GOSUB Aok:EDIT FIELD 1,F$(,77,8)-(475,22):PUT(58,8),TA,PSET
SA=1:AN(0)=0
RETURN

```

ErE2:

```

' Message pour erreur dans la saisie du Type de fichier
MOVETO 10,35:PRINT "Pas plus de 4 caractères
MOVETO 10,55:PRINT "pour le type de fichier...
GOSUB Aok:EDIT FIELD 2,T$(,133,40)-(186,54):PUT(114,40),TA,PSET
SA=2:AN(1)=0
RETURN

```

ErE3:

```

' Message pour erreur dans la saisie du créateur du fichier
MOVETO 10,35:PRINT "Pas plus de 4 caractères
MOVETO 10,55:PRINT "pour le nom du créateur...
GOSUB Aok:EDIT FIELD 3,C$(,422,40)-(475,54):PUT(403,40),TA,PSET
SA=3:AN(2)=0
RETURN

```

ErE4:

```

' Message pour erreur dans la saisie de la position horizontale
MOVETO 10,35:PRINT "Entrée incorrecte dans la
MOVETO 10,55:PRINT "zone de saisie de la position
MOVETO 10,75:PRINT "horizontale de l'icône...
GOSUB Aok:EDIT FIELD 4,PH$(,440,122)-(475,136)
PUT(421,122),TA,PSET:SA=4:AN(3)=0
RETURN

```

ErE5:

```

' Message pour erreur dans la saisie de la position verticale
MOVETO 10,35:PRINT "Entrée incorrecte dans la

```

```

MOVETO 10,55:PRINT "zone de saisie de la position
MOVETO 10,75:PRINT "verticale de l'icône
GOSUB Aok:EDIT FIELD 5,PV$(,440,146)-(475,160)
PUT(421,146),TA,PSET:SA=5:AN(4)=0
RETURN

```

ErD:

```

' Message pour tentative de passage du volume ou bureau vers dossier
MOVETO 10,25:PRINT "Les numéros de dossiers sont
MOVETO 10,45:PRINT "alloués par le 'Finder', cette
MOVETO 10,65:PRINT "opération est donc impossible...
GOSUB Aok
RETURN

```

ErC:

```

' Message pour erreur dans la saisie du nom de fichier
' (caractère ":" dans la chaîne)
MOVETO 10,25:PRINT "Le caractère ':' ne doit
MOVETO 10,45:PRINT "pas apparaître dans
MOVETO 10,65:PRINT "le nom de fichier...
GOSUB Aok:EDIT FIELD 1,F$(,77,8)-(475,22):PUT(58,8),TA,PSET
SA=1:AN(0)=0
RETURN

```

Aok:

```

' Attente d'une action sur le bouton "OK" (message d'erreur)
PUT(238,10),TCO:WHILE DIALOG(0)<>1:WEND
WINDOW CLOSE 2:PUT(42,8),TG,OR
RETURN

```

Qui:

```

SYSTEM

```

## Routines utilisées par le programme Basic

0000		.Trap	_GetVolInfo	\$A007	; Retaourne les informations sur le volume.
0000		.Trap	_ReName	\$A00B	; Change le nom du fichier et/ou du volume.
0000		.Trap	_GetFileInfo	\$A00C	; Retaourne les informations sur le fichier.
0000		.Trap	_SetFileInfo	\$A00D	; Modifie les informations sur le fichier.
0000		.Trap	_MountVol	\$A00F	; Charge en mémoire les informations sur la disquette.
0000		.Trap	_OffLine	\$A035	; Libère la mémoire utilisée par le volume.
0000		.Trap	_DeleteMenu	\$A936	; Annule un menu (en mémoire).
0000		.Trap	_DrawMenuBar	\$A937	; Réaffiche la barre des menus.
0000	xxxx xxxx	Tampon	DCB.L	20,0	; Tampon de 80 octets pour les ent/sorties.
0050	xxxx xxxx	Nom	DCB.L	16,0	; Tampon de 64 octets pour le nom de volume ou le nom
0090					; de fichier.
0090	xxxx xxxx	N.Nom	DCB.L	16,0	; Tampon de 64 octets pour le nouveau nom de fichier.
00FFFF		Masque	EQU	\$00FFFFFF	; Constante pour adresses sur 24 bits.
00D0					; Sous-programme commun aux deux routines. Il est utilisé pour transférer les
00D0					; caractères du nom de volume ou du nom de fichier depuis le programme Basic
00D0					; vers le tampon "Nom" ou le tampon "N.Nom".
00D0					; Le premier segment (Sp.N.F) est utilisé uniquement pour les noms de fichier.
00D0	226E 0000	Sp.N.F	MOVEA.L	8(A6),A1	; L'adresse du descripteur de la chaîne, placée dans la
00D4					; pile, est transférée dans A1.
00D4	72 3E		MOVEQ	*62,D1	; D1 est utilisé comme compteur de boucles.
00D6	14E9 0001	Sp.N.V	MOVE.B	1(A1),(A2)+	; Place le nombre de caractères formant le nom dans le
00DA					; premier octet du tampon Nom ou N.Nom.
00DA	2429 0001		MOVE.L	1(A1),D2	; L'adresse de la chaîne est placée dans D2.
00DE	0282 00FFFFFF		ANDI.L	*Masque,D2	; Supprime l'octet de poids fort pour obtenir une adresse
00E4					; valide sur 24 bits.
00E4	2642		MOVEA.L	D2,A3	; L'adresse obtenue est placée dans A3.
00E6	14DB	Tr.Nom	MOVE.B	(A3)+,(A2)+	; Boucle de transfert de la chaîne.
00E8	51C9 FFFC		DBRA	D1,Tr.Nom	; D1 est décrémenté de 1 et comparé à -1. Si différent,
00EC					; branchement sur "Tr.Nom".
00EC	4E75		RTS		; Retour à la routine appelante.
00EE					; Routine utilisée pour lire les informations sur le volume et le fichier à traiter.
00EE					; Pour le volume, seul le numéro de référence est retenu. Les informations
00EE					; obtenues sont passées au programme Basic par l'intermédiaire du tampon
00EE					; de 80 octets "Tampon" situé, pour le programme Basic, dans les 40 premiers
00EE					; éléments du tableau de variables entières TC.
00EE	4E56 0000	L.Info	LINK	A6,*0	
00F2	41FA FF0C		LEA	Tampon,A0	; L'adresse de base du tampon (correspondant à
00F6					; VARPTR(TC(0)) ) est placée dans A0.
00F6	2248		MOVEA.L	A0,A1	; Copie de A0 dans A1.
00F8					; Réinitialisation du tampon (mise à zéro).
00F8	70 13		MOVEQ	*19,D0	; D0 est un compteur de boucles. On charge la valeur 19
00FA	4299	T.Zero	CLR.L	(A1)+	; dans le compteur. Comme l'instruction DBRA provoque
00FC	51C8 FFFC		DBRA	D0,T.Zero	; une sortie de la boucle lorsque le compteur est égal à
0100					; -1, on met à zéro 20 mots de 32 bits, soit 80 octets.
0100	45FA FF4E		LEA	Nom,A2	; Place l'adresse du tampon "Nom" dans A2.
0104	214A 0012		MOVE.L	A2,18(A0)	; Adresse du tampon "Nom" dans A0+18 (voir Pom's
0108					; numéro 17," Catalogue sur Imp.").
0108	226E 000C		MOVEA.L	12(A6),A1	; L'adresse du descripteur de la chaîne contenant le nom
010C					; du volume, placée dans la pile au moment du "CALL",
010C					; est transférée dans le registre d'adresse A1.
010C	72 1A		MOVEQ	*26,D1	; Compteur de boucles pour le sous-programme Sp.N.V.
010E					; Un nom de volume contient au maximum 27 caractères.

010E	61 C6		BSR.S	Sp.N.V	; Branchement vers le sous-programme "Sp.N.V".
0110	5368 001C		SUBQ	*1,28(A0)	; Le mot de 16 bits situé à l'adresse A0+28, mis à zéro
0114					; par la boucle T.Zero, est décrémenté de 1 et devient
0114					; ainsi négatif. Cela indique à la routine _GetVolInfo
0114					; qu'elle doit utiliser le pointeur (A0+18) sur le nom
0114					; de fichier.
0114	A007			_GetVolInfo	
0116	45FA FF38		LEA	Nom,A2	; On effectue les mêmes opérations que pour la routine
011A	214A 0012		MOVE.L	A2,18(A0)	; précédente mise à part l'adresse A0+28 qui est remise
011E	61 B0		BSR.S	Sp.N.F	; à zéro pour indiquer à la routine _GetFileInfo qu'elle
0120	4268 001C		CLR	28(A0)	; doit utiliser, pour déterminer le volume, le numéro de
0124	A00C			_GetFileInfo	; référence placé dans A0+22 par _GetVolInfo.
0126	4E5E		UNLK	A6	; Retour au
0128	4E75		RTS		; programme Basic.
012A					
012A					; Routine pour modifier les informations utilisées par le "Finder" et
012A					; modifier le nom de fichier.
012A					
012A	4E56 0000	E.Info	LINK	A6,*0	; Passage du nom de fichier initial aux routines système.
012E	41FA FED0		LEA	Tampon,A0	; La méthode utilisée est la même que pour les routines
0132	43FA FF1C		LEA	Nom,A1	; précédentes.
0136	2149 0012		MOVE.L	A1,18(A0)	
013A	A00D			_SetFileInfo	; Modification des informations.
013C	45FA FF52		LEA	N.Nom,A2	; On passe maintenant le nouveau nom du fichier. Un
0140	214A 001C		MOVE.L	A2,28(A0)	; pointeur sur ce nouveau nom doit se trouver dans
0144	61 8A		BSR.S	Sp.N.F	; l'adresse A0+28.
0146	A00B			_ReName	
0148	A035			_OffLine	; On vide la mémoire des informations sur le volume,
014A	363C 0001		MOVE	*1,D3	; puis on les recharge avec la routine système
014E	3143 0016	Er.Drive	MOVE	D3,22(A0)	; _MountVol. Ceci permet d'avoir toujours en mémoire
0152	A00F			_MountVol	; des informations à jour. La routine _MountVol utilise
0154	363C 0002		MOVE	*2,D3	; le numéro du drive; pour déterminer ce numéro, on
0158	3828 0010		MOVE	16(A0),D4	; utilise le code d'erreur retourné dans A0+16. On essaie
015C	66 F0		BNE.S	Er.Drive	; d'abord sur le drive 1; si un code d'erreur est
015E	4E5E		UNLK	A6	; retourné (A0+16<>0), on recommence sur le drive 2.
0160	4E75		RTS		
0162					
0162					; Routine pour supprimer certains menus (voir Pom's numéro 17).
0162					
0162	4E56 0000	E.Menu	LINK	A6,*0	
0166	3F2E 0008		MOVE	8(A6),-(SP)	
016A	A936			_DeleteMenu	
016C	A937			_DrawMenuBar	
016E	4E5E		UNLK	A6	
0170	4E75		RTS		

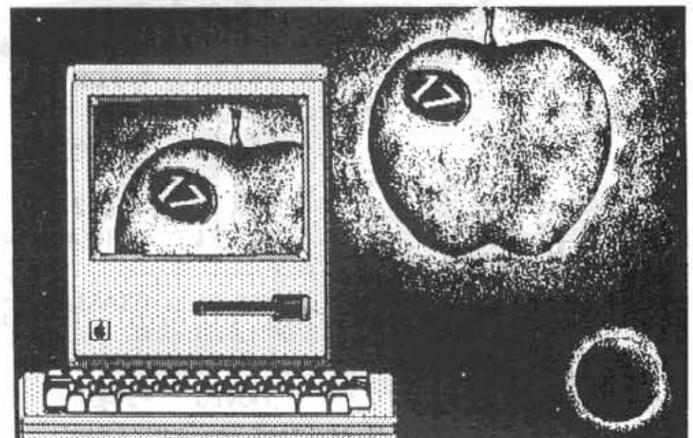
## Disquette Mac 17

### Catalogue sur imprimante

Pour obtenir toutes les informations sur les fichiers (type, créateur, index, position du premier bloc pour la partie "Données", position du premier bloc pour la partie "Ressource", longueurs logiques et physiques, fichiers protégés ou non, icônes invisibles, etc...) sur papier ou à l'écran.

### Routines "BLOAD" et "BSAVE"

Similaires aux routines "BLOAD" et "BSAVE" de l'Apple //, elles autorisent des sauvegardes et chargements rapides de fichiers binaires (données graphiques, tableaux de variables, tables de références, etc...)



# Les curseurs du système

Julien Thomas

Le Basic Microsoft offre la possibilité de redéfinir la forme du curseur en plaçant des codes dans un tableau de variables. Ceci est très intéressant mais, s'il on se penche quelques minutes sur le problème, on s'aperçoit qu'il est possible d'utiliser cinq curseurs prédéfinis. Les quatre premiers font partie du dossier système alors que le dernier est issu du Basic Microsoft.

Une très courte routine est utilisée pour le chargement et l'affichage du

curseur désiré. On utilise deux "Traps"; le premier charge en mémoire la définition du curseur (curseur, masque et position du "point chaud") et retourne l'adresse où se trouve le pointeur sur la structure en question; la seconde est l'équivalent de l'instruction CALL SETCURSOR.

Le passage de l'identificateur (numéro du curseur) se fait comme d'habitude par la pile, avec CALL ROUTINE!(ID%) (ou ROUTINE! ID% avec la seconde version du Basic Microsoft).

## 10 ' PROGRAMME DE DEMONSTRATION

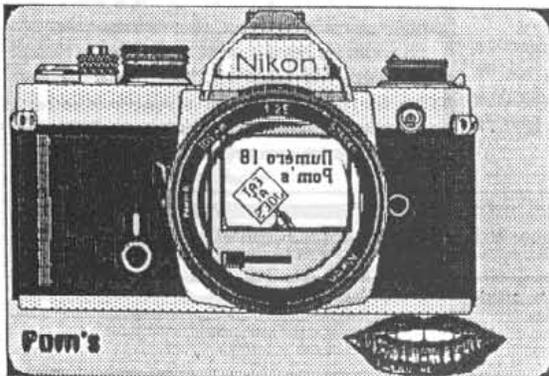
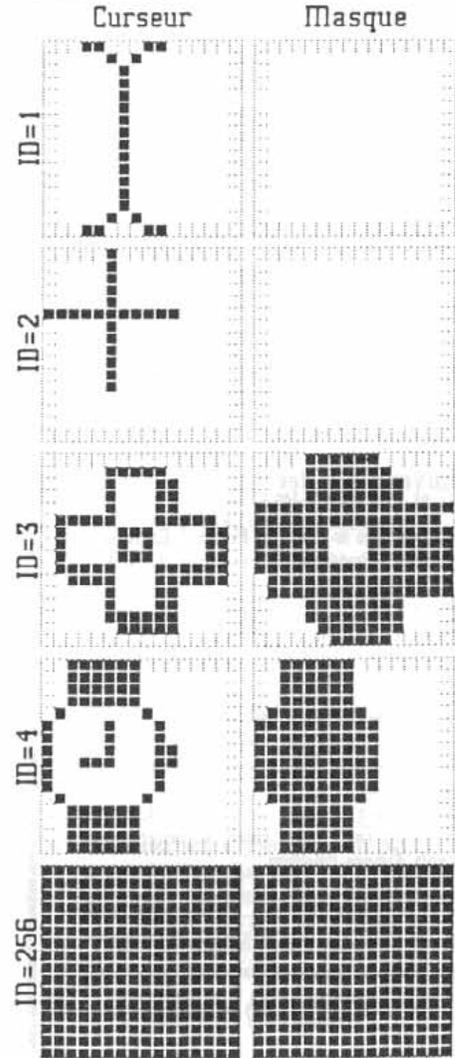
```

20 DEFINT A-Z:CLS
30 ' Data pour code objet
40 DATA &H4E56,0,&h42A7,&h3F2E,8
50 DATA &hA9B9,&h2057,&h2E90
60 DATA &HA851,&H4E5E,&H4E75
70 FOR N=0 TO 10:READ C(N):NEXT
80 A!=VARPTR(C(0))
90 INPUT "Numéro du curseur ? ",N
100 CALL SHOWCURSOR
110 CALL A!(N):GOTO 90
    
```



```

0000 ; Récupération des curseurs prédéfinis.
0000
0000 .Trap _GetCursor $A9B9
0000 .Trap _SetCursor $A851
0000
0000 4E56 0000 LINK A6,*0
0004 42A7 CLR.L -(SP)
0006 3F2E 0006 MOVE 0(A6),-(SP)
000A A9B9 _GetCursor
000C 2057 MOVEA.L (SP),A0
000E 2E90 MOVE.L (A0),(SP)
0010 A851 _SetCursor
0012 4E5E UNLK A6
0014 4E75 RTS
    
```



## Disquette Mac 18

Cette nouvelle disquette comporte le nouveau fichier "système" ainsi que la version 4.1 du "finder", beaucoup plus rapide et complète que la précédente. Le programme "System update" autorise l'implantation des nouveaux "finder" et "système" sur vos anciennes disquettes. "BR DEMO" est un programme d'application qui permet l'affichage temporisé d'écrans créés avec la séquence de touches "Commande-Shift-Majuscule-3". "Font/DA Mover" est une version nettement améliorée de "Font Mover". De plus, il permet de manipuler les accessoires de bureau. Enfin, outre les programmes publiés dans ce numéro, vous trouverez les polices de caractères de l'imprimante Laser Writer "Times", "Courier", "Symbol" et "Helvetica" (utilisables avec l'imprimante ImageWriter).

## Logiciels de troisième génération

La première génération des logiciels sur Mac était conçue pour 128K de mémoire.

La seconde génération, née avec le Finder 1.1g, était destinée aux utilisateurs de Macs 512K, même si ce Finder fonctionnait en 128K. Certains programmes de cette génération exigent 512K de mémoire centrale, d'autres, à l'instar de Think Tank, ont une version 512K plus puissante.

La troisième génération vient d'arriver, avec le Finder 4.1, la version 4.5 de MacWrite et la version 1.5 de MacPaint. Le nouveau système, outre le Finder 4.1 dont nous parlons plus bas, ajoute l'accessoire Sélection d'imprimante, met à jour le fichier System et remplace le fichier ImageWriter par une nouvelle version autorisant l'impression à 50% de réduction et fonctionnant sur les ImageWriter 12" et 15".

## Le nouveau Finder

Qu'apporte de plus le nouveau Finder ? En résumé :

- la possibilité d'imprimer le catalogue de la disquette;
- l'affichage d'un cadenas pour les fichiers verrouillés, avec les catalogues sous forme de texte;
- la possibilité de copier des fichiers sur un disque ouvert;
- la possibilité de mettre un disque dans la poubelle (éjection et disparition de l'icône du disque);
- une option du menu Fichier pour la création d'un nouveau dossier;
- un Mini-Finder, dans le menu Spécial, permet de se déplacer rapidement entre des programmes et documents d'usage fréquent.

Et nous n'avons pas tout dit ... Nous vous laissons découvrir le reste, y compris le gain en vitesse.

### Pom's recrute encore !

Pour faire face à sa croissance continue, Pom's recherche un collaborateur ou une collaboratrice à temps plein.

-----  
Connaissance parfaite de l'assembleur, de l'Apple II, du DOS et du ProDOS requise. Il faut aimer le travail en équipe, être capable de s'exprimer clairement et avoir un niveau acceptable en orthographe.

## Gestion de fichiers sur Mac

Sans le moindre souci d'exhaustivité, nous faisons ici le point sur les logiciels permettant de faire de la gestion de fichiers sur Macintosh, même s'ils possèdent parfois des possibilités supplémentaires. Lire la rubrique des Micro-Informations pour les informations générales en matière de logiciels sur Apple.

**Hélix** est un programme de base de données relationnelle fonctionnant exclusivement avec 512K. Nous avons été impressionnés par ses possibilités, qui se payent quand même par un certain effort d'apprentissage. Ce programme puissant sera évoqué plus en détail dans un Pom's ultérieur. Il est vendu par Softsel - Tél (1) 768.59.50 au prix de 4.763 FF HT.

**CX MacBase**, avec sa version M20, résout la grande majorité des bugs encore présents dans la version M10. Contacter votre revendeur ou la société Controle X pour la version M20.

De son côté, **ABC Base** est maintenant lui aussi disponible dans sa nouvelle version, où les bugs antérieurs sont aussi en grande partie résolus. De plus, la documentation est sensiblement améliorée.

Enfin, depuis le temps qu'on l'attendait, **File** est maintenant vendu par Microsoft. Nous devons souligner sa grande facilité d'usage, sa rapidité de traitement, son absence apparente de bugs, et son parfait fonctionnement en 128K. En échange de quoi, ses fonctionnalités sont certes moins riches que celles des précédents : ainsi, les possibilités de calcul sont tout-à-fait sommaires, et il n'est pas question de réaliser des traitements reliant entre eux deux fichiers ou plus. On peut réaliser des fiches présentant joliment, mais au prix de certaines manipulations.

Nous attendons donc avec une grande impatience le futur intégré de Microsoft qui a tout de même réussi la performance notable en micro-informatique de placer la plupart de ses produits dans la série des bestsellers.

En ce qui concerne **Jazz**, à la date de rédaction de cet article, on ne sait pas encore s'il ne faudra pas avoir, outre 512K, un disque dur pour pouvoir l'utiliser dans des conditions pour le moins acceptables.

## Disquette N°18 du cahier Macintosh

La disquette Macintosh de ce numéro de Pom's comporte tous les programmes parus dans le cahier Mac, mais elle vous réserve en outre des cadeaux surprise :

- le **System Update**, qui transforme vos vieux fichiers Système en remplaçant le Finder 1.1g par la version 4.1 et l'ancien fichier ImageWriter par le nouveau.
- le **Font/DA Mover**, pour déplacer des polices (fonts) ou des accessoires de bureau d'une disquette à une autre. Les polices peuvent provenir d'un système ou d'un fichier quelconque. Un excellent programme, valorisé par le nombre croissant d'accessoires de bureau qui traînent partout aujourd'hui.
- le programme **Demo**, dû à Bruno Rives, qui vous permet de monter une séance de présentation avec plusieurs écrans.
- les quatre polices de caractères utilisés à ce jour par la LaserWriter.

Pour transformer toutes vos disquettes comportant l'ancien système, il suffit de cliquer sur l'icône System Update.



System Update



Font/DA Mover

La mise à jour du système d'une disquette ne modifie en rien ses caractéristiques spécifiques telles que le clavier ou les polices qui y sont intégrées. Aucun risque donc ...

Le seul effet négatif que nous ayons pu constater est que les dossiers réalisés avec l'ancien système et relus par le nouveau perdaient leur nom. Par contre, leur structure de dossier et leur contenu restent inchangés, que le système ayant servi à les définir soit français ou américain.

Les nouvelles versions de MacWrite et MacPaint se trouveront, pour leur part, sur la disquette Macintosh du N°19 de Pom's. Nous profiterons certainement de l'occasion pour joindre encore à cette disquette d'autres nouveautés relevant du domaine public, en plus des programmes publiés dans Pom's.

# Aldo Reset a la parole

Pom's

Beaucoup a été dit sur les "pirateurs", et Pom's a trouvé juste de leur donner la parole, afin qu'ils aient la possibilité de dire ce qu'ils pensent sans que leurs propos ne soient abusivement interprétés ou même, comme cela leur est déjà arrivé avec une revue spécialisée dans le choc des clichés, purement inventés. Nous avons donc rencontré Marc, plus connu sous le nom de Aldo Reset, membre fondateur du Clean Crack Band. Quel est le possesseur d'un Apple II Plus, //e ou //c qui n'a eu entre les mains des disquettes signées du Clean Crack Band, ou de ses membres les plus éminents : Aldo Reset, Tony NMI, Laurent Rueil ou Johnny Disquette ?

La première impression que donne Marc, appelons-le plutôt Aldo pour respecter son nom de guerre, c'est sa jeunesse. Il compte 20 printemps et, quand on sait qu'il œuvre sur son Apple II+ depuis plusieurs années déjà, on voit qu'il a démarré tôt.

*NDLR : Pom's n'a pas pu retranscrire l'interview in extenso, la discussion ayant duré plus d'une heure et demie. Nous avons donc extrait des questions et réponses typiques, avec un grand souci de ne pas dénaturer les propos échangés.*

P - Le piratage qui a déjà eu lieu vous pose-t-il un problème, légalement ?

A - Je ne vois pas très bien qui m'en voudrait ...

P - Dans la mesure où vous avez déjà signé un certain nombre de disquettes Apple II, les éditeurs ne peuvent-ils vous ennuyer ?

A - Les éditeurs américains ? A mon avis, ils feraient déjà quelque chose contre ceux qui existent aux Etats-Unis, avant de venir nous embêter. Les éditeurs français, sans vouloir être méchant pour eux, disons qu'ils n'ont pas le même niveau.

P - Comment signez-vous les disquettes que vous "déplombez" ?

A - Avant, je signais Aldo Reset ou CCB; maintenant, je ne fais plus grand chose. CCB est un groupe; on a toujours été plusieurs dedans. On a commencé, on était au départ vraiment deux, Tony NMI et puis moi. Ensuite, Laurent Rueil qu'on a rencontré. Au fur et à mesure, on mettait tous les copains dedans. Ça fait toujours plaisir de faire partie de quelque chose.

Alors, on s'est retrouvé, on a bien été une quinzaine, pendant un moment. Et puis, l'année dernière, ça n'allait plus. Je n'arrivais plus à savoir qui était du CCB; j'entendais parler d'untel qui était du CCB et que je ne connaissais même pas. Alors on a fait un petit nettoyage. On s'est retrouvé à cinq. C'est à peu près l'époque où on a laissé tomber, de toute façon.

P - Vous ne déplombez plus ?

A - Non, pas vraiment. De temps en temps, parce que d'abord ça nous plaît. C'est surtout ça.

P - Est-ce que cela a été une industrie à un moment donné ?

A - Mettons que je faisais cela tous les jours. Pendant une bonne partie de la journée et de la nuit. Cinq par jour pendant un moment.

P - Cela a-t-il été une activité commerciale ?

A - On n'a jamais gagné d'argent. On n'est jamais passé au stade industriel, c'est-à-dire déplomber des trucs exprès. C'était pour l'amour de l'art, et surtout pour avoir les programmes. S'il fallait acheter tous les trucs qui sortent, moi je pourrais travailler encore pendant dix ans, je ne pourrais pas les avoir. Ça coûte quand même cher, et puis c'était un moyen facile et agréable. On n'allait pas se gêner.

P - Y a-t-il d'autres groupes que le Clean Crack Band ?

A - Il y en a plein. Ça a réellement commencé l'année dernière. Il y a plein de monde. Maintenant, cela ne vaut plus tellement la peine. A la limite, ça ne me dérangerait pas de continuer, mais ça prend du temps, et ça ne rapporte pas.

P - Comment déplombe-t-on ?

A - Il y a plusieurs méthodes pour déplomber. Il y a la méthode sale, qui est malheureusement trop répandue, avec les cartes d'interruption, les choses comme cela. Non seulement, ça ne marche pas toujours, mais c'est rien, c'est même pas drôle. Moi, j'ai une carte comme ça; je m'en sers, mais pas pour déplomber. Je m'en sers quand je suis en train de jouer, quand je veux absolument voir la fin du jeu; dès que je me fais tuer, je reviens au stade précédent avec ce que j'ai sauvé de la carte.

Dans la méthode propre, il faut suivre tout, depuis le début, à la main, sans rien de tout en hard, juste à la ruse. On suit pas à pas. Si on veut faire quelque chose de bien, il faut aller jusqu'au bout, suivre et bien comprendre ce qui se passe. Mais quand en veut ensuite faire (de la disquette déplombée) quelque chose de standard, il faut aussi faire proprement.

P - Est-ce que, pour un système comme ProLock, vous avez un mode de déplombage systématique, ou est-ce que ce travail est personnalisé ?

A - ProLock, c'est même pas la peine de s'énerver avec ça. C'est tellement ridicule que (rires).

P - Il paraît que certains pirates déclarent pouvoir déplomber n'importe quel logiciel en une demi-heure, montre en main. Est-ce vrai ?

A - C'est des bêtises. Une demi-heure, ce n'est pas standard. Il y a 90% des machins qui sont faisables en cinq minutes. Faut le dire. Maintenant, dans les 10% restant, il y en a qui prennent deux heures, d'autres plus. Omnis 3, j'ai passé une semaine dessus, à bosser nuit et jour; je ne dis pas ça pour être vache avec Ka, mais c'était bien. C'est pas que la protection était une des meilleures, mais elle était hors du commun, différente de tout ce qui se faisait. Sun Dog, c'est encore un bon exemple; une semaine aussi. Une semaine, ça a été le maximum. Pas du matin au soir, mais en y pensant tout le temps, la nuit aussi. Sun Dog et Omnis 3, c'était très dur, mais on pouvait faire plus dur encore; ils auraient pu rajouter des trucs. Ils se sont dit "C'est pas la peine"; à mon avis, ils avaient raison.

P - Peut-on faire un programme qu'on ne puisse pas déplomber ?

A - Pas possible ... on passera le temps qu'il faudra, mais on l'aura.

Cette interview a aussi été réalisée avec Ready-Set-Go et l'imprimante LaserWriter - On peut aussi utiliser pour cela les programmes ...

P - Vous travaillez aussi sur le Macintosh ?

A - Pas vraiment, je commence un peu. Le 68000, c'est pas mal, mais le Macintosh, il traîne un peu, je trouve. Je m'attendais à mieux. Tant qu'il n'y aura pas une vitesse plus rapide ... A 10 MHz, on peut commencer à travailler. Ceci dit, je commencerais à travailler dessus si on m'en donnait un. Je ne peux pas encore en acheter un maintenant.

P - Les protections semblent meilleures sur le Macintosh.

A - C'est normal. D'abord, il y a 128K ou 512K au lieu de 64K; fouiller dans 64K, à la limite, c'est faisable. Dans 128 ou 512, c'est plus dur. En plus, dans le 68000, il y a des registres à ne pas savoir où on en est. Moi, j'ai encore rien déplombé sur Mac; un truc, quoi, mais c'était vraiment ridicule.

**Autre pirateur** - Les grands spécialistes du piratage sur Macintosh, c'est du côté du soleil. On parle de Nice et Marseille. On n'est même pas sûr des villes. La raison fondamentale de ce piratage-là, c'est le chômage.

P - Les protections d'ACI ne sont pas mal, n'est-ce pas ?

A - Laurent Ribardière (créateur d'ABC Base, diffusé par ACI), c'est quelqu'un qui n'est pas négligeable.

P - Que faisiez-vous au début du CCB ?

A - Moi, j'étais au lycée. Laurent venait d'avoir eu son bac; Tony était aussi au lycée. On faisait ça pour s'amuser. Moi, je suis toujours étudiant, je fais des études en biologie. Les autres, il y en a qui travaillent, à écrire des logiciels pour des boîtes connues.

**Autre pirateur** - Aldo et un certain nombre d'autres pirates du CCB travaillent maintenant, sous des formes variées, avec la société IRD, qui part du principe que les bons pirates sont de bons programmeurs.

A - A partir du moment où on déplombe bien, on sait programmer bien. Si on déplombe mal, on ne sait pas programmer.

Il y a ceux qui savent suivre un programme, regarder une routine en langage machine, comprendre ce qu'elle fait, programmer ce qu'il faut pour qu'elle fasse la même chose, mais en DOS 3.3 normal.

Il y a les autres qui mettent quinze fois plus de temps à comprendre, qui se trompent quinze fois de suite et font des trucs qui ont l'air dégueulasses à l'écran. Des bouts de machins qui traînent dans tous les coins.

Nous, quand on signe, on essaye de faire ça discrètement, pour rien enlever à la façon dont les gens auraient voulu que ce soit présenté originellement. Moi, quand je vois une page qui arrive, et où on ne voit plus ce que c'est, parce qu'un abruti a mis son nom en plein milieu et en gros, et qu'en plus les couleurs bavent, ça me plaît pas, et je suis sûr que ce type-là ne sait pas programmer. C'est une insulte aux gens qui ont fait le programme au départ.

P - En fait, vous voulez dire que vous avez un certain respect pour les auteurs de programmes.

A - Cela paraît bizarre, mais c'est vrai. On s'est appelé Clean Crack Band parce qu'on essaye, à chaque fois, de faire du beau travail. Tous les programmes qu'on a déplombés et qui portent notre signature quelque part, c'est bien fait, ça marche à la même vitesse ou plus rapidement que l'original, ça conserve l'esprit du logiciel. C'est bien fait, quoi. C'est discret, et bien.

P - Vous êtes en quelque sorte des esthètes.

A - On peut dire ça. On peut, on peut ... Ça fait un peu prétentieux, mais (rires). Mais c'est vrai, je vois ça comme ça. C'est pour cela que ça me met hors de moi quand je vois des gens qui ne le font pas.

La majorité des gens qui écrivent des protections pour les logiciels font des trucs bien; je ne parle pas de ceux où il faut cinq minutes pour déplomber, quoique ce n'est pas une raison pour les massacrer. Prenons par exemple Bill Bashan, qui a fait Dogfight et la protection de tous les IDSI (Point 5, Trick Shot), c'est un type que j'admire vraiment. Quand on regarde une page prise n'importe où dans un listing qu'il a écrit, on reconnaît tout de suite que c'est lui qui l'a fait. Il a une espèce de style.

P - En général, c'est la même personne qui crée les logiciels et réalise la protection ?

**Autre pirateur** - Non, mais celui qui fait la protection intercale son code dans celui de l'auteur.

P - Mais, alors, comment reconnaît-on le code original du code de protection ?

A - C'est ça qui le trahit. Pour déplomber un truc, souvent, je commence comme normalement, puis ensuite je regarde dans la mémoire les zones où je pense qu'il pourrait y avoir quelque chose. Je regarde au hasard. Et souvent, je tombe sur un endroit où je me dis "Tiens, ça, ce n'est pas écrit comme le reste"; il y a alors de fortes chances pour que la protection, ce soit ce truc-là.

Y'a des gens qui utilisent plus certains registres que d'autres. Ce sont des trucs comme ça. C'est subtil, et c'est bien.

**Autre pirateur** - A ce niveau-là, quelqu'un comme Aldo - surtout lui, parce que c'est un des meilleurs dans le genre - reconnaît le style d'un auteur de logiciel comme on reconnaît celui d'un écrivain. Signe distinctif très caractéristique à mon avis, Aldo n'utilise pratiquement pas d'assembleur, il travaille directement avec les codes mnémoniques, en hexa directement.

P - Savez-vous ainsi identifier plusieurs "protecteurs" américains ?

A - J'en connais plusieurs. Il y a toujours dans un coin le nom de celui qui a fait la protection; on ne peut pas y échapper. C'est pas forcément leur nom en clair; il y en a qui mettent leurs nombres favoris, des trucs comme ça. On les retrouve. Quand je vois dans Dogfight, à un moment dans le programme, il y a deux/trois machins, et puis ensuite, je retrouve les mêmes codes dans une protection, et plus tard je retrouve que c'est le même type (à cause de cette signature) dans d'autres logiciels. Il suffit de faire attention aux bons trucs.

Moi, j'ai arrêté de déplomber aussi parce qu'une de mes raisons, c'était de récupérer tous les softs qui sortaient, pour être au courant. Maintenant, j'arrive à tout avoir par les amis, mal déplombé, on n'y peut rien, mais je les ai quand même. C'est quand même triste de voir que c'est mal fait ... C'est dommage.

P - Quel genre de bêtises a-t-on dit sur vous ?

**Autre pirateur** - D'après Paris-Match, Aldo aurait travaillé sur le fichier d'une banque. Or, il ne fait pas du tout de communication, il a horreur de cela. C'est Tony qui est "branché".

A - J'aurais piraté le Crédit Agricole par Minitel. Faut quand même pas exagérer. J'ai fait d'autres choses. Ceci dit, les auteurs de logiciels, j'en connais peu qui ne piratent pas. Si un auteur veut faire un bon logiciel, il faut qu'il voie ce que font tous les concurrents.

P - Une morale à tout cela ?

A - C'est pas forcément une mauvaise chose que les logiciels se diffusent par piratage. Que les programmes circulent permet d'amener plus de gens à un certain niveau, de voir ce qu'est un bon programme, ça aide à mon avis la micro-informatique et l'informatique en général; et ça, je trouve que c'est une bonne chose. Les avis sont partagés là-dessus. Enfin comme les programmes circulent de toute façon déplombés, il vaut mieux qu'ils circulent déplombés de façon propre que sale. Mal déplombé, cela peut donner l'impression que le programme marche mal, cela fait du tort. C'est pour cela que les gens qui déplombent pas bien, ils font du mal doublement. Le bon déplombage fait progresser l'informatique.

... MacPublisher, Page Maker (non encore disponible) et, pourquoi pas, tout simplement le génial Mac Draw, encore trop méconnu ...

# Hard copy de la page HGR étendue

Marc Debuigne

Cet article s'adresse uniquement aux propriétaires d'un Apple //e, équipé d'une carte 80 colonnes étendue, ou d'un Apple //c (la carte Chat Mauve est intégrée en standard). Vous savez certainement que ces cartes additionnelles améliorent non seulement l'affichage en mode TEXT de l'Apple, mais aussi ses capacités graphiques (jusqu'à 560 x 192 points), en permettant l'utilisation des pages haute résolution de 16 Ko, d'où une meilleure qualité des graphismes obtenus. Il était jusqu'à présent possible de créer et de sauvegarder des graphismes 16 Ko à l'aide d'utilitaires très puissants tels que Purplesoft, ou à partir du Basic (voir suite de l'article), mais pas d'en obtenir une reproduction sur papier. Le programme en langage machine proposé ici, associé au programme Basic COPIE comble cette lacune, la qualité de la reproduction obtenue n'étant toutefois optimale que lors de la recopie d'une image 560 x 192 en noir et blanc.

## Commutateurs mémoire et graphismes 16 Ko

Les modes graphiques 16 Ko page 1 des cartes 80 colonnes étendues utilisent les 8 Ko de la page graphique haute résolution 1 et 8 Ko de la mémoire auxiliaire situés aux mêmes adresses. Leur utilisation suppose donc lecture et écriture dans les 64 Ko additionnels.

L'écriture et la lecture de données dans la mémoire auxiliaire peut s'effectuer de deux manières : soit en utilisant la routine AUXMOVE (voir Manuel de référence //e ou de la carte 80 colonnes), soit en jonglant avec les commutateurs mémoire 80STORE, PAGE2 etc...

C'est cette deuxième méthode qu'utilise le programme de hard copie. Afin de permettre son emploi à partir d'un programme Basic, seules les deux zones mémoire correspondant aux pages TEXT1 et HGR1 sont commutées.

Ceux qui ont déjà essayé d'utiliser les 64 Ko de la mémoire auxiliaire sans passer par AUXMOVE se sont certainement rendu compte de la complexité du problème, et ont pu constater que la séquence CTRL-Pomme Ouverte-RESET est parfois incapable de remettre de l'ordre dans tous les pointeurs. En Basic, cette mémoire additionnelle n'est absolument pas utilisable directement et ne peut être pleinement qu'en assembleur (l'Applesoft n'est malheureusement

pas un Basic 128 Ko). Les commutations de pages mémoire effectuées pour utiliser le mode HGR 16 Ko sont les seules exploitables sans grand danger en Basic. Il suffit alors de protéger la page HGR1 par un LOMEM ou un HIMEM approprié et de veiller à ce qu'aucun pointeur ne soit perdu lors d'une opération d'Entrée / Sortie.

**Attention:** certaines interfaces imprimante, telles que la SSC, utilisent les "trous" de la page TEXT1 pour y loger des pointeurs. Il faut donc veiller, lors de l'envoi d'un caractère à l'imprimante par un PRINT ou un JSR COUT, à ce que la page TEXT1 de la mémoire principale soit bien accessible (l'auteur de ces quelques lignes en a fait la triste expérience).

## Précisions sur les cartes 64 Ko

Si vous avez essayé de comprendre l'effet des différents commutateurs mémoire vous avez pu constater que les explications fournies par les manuels ne sont pas d'un abord facile; les quelques lignes qui suivent vous permettront certainement d'y voir plus clair :

- Le commutateur mémoire 80STORE a priorité sur RAMWRT et RAMRD, ce qui rend parfois AUXMOVE inutilisable.
- Les commutateurs RAMWRT et RAMRD spécifient la Bank utilisée en lecture ou en écriture.

### Action des commutateurs

1) Si ALTZP est OFF :

Les commutations s'effectuent obligatoirement entre les pages 2 et \$BF incluses, deux cas de figure sont alors possibles :

– Si 80STORE est OFF, RAMWRT et RAMRD commutent entièrement de la page 2 à la page \$BF.

– Si 80STORE est ON, il est prioritaire par rapport à RAMWRT et RAMRD qui ne commutent plus, contrairement aux pages 2 à 4, 8 à \$20 et \$40 à \$BF. Les pages TEXT1 et HGR1 commutent alors indépendamment des autres si HIRES est ON. PAGE2 effectue leur commutation.

2) Si ALTZP est ON :

Les commutations affectent la carte langage, la page 0 et la page 1. Elles sont effectuées avec RAMWRT et RAMRD, les autres commutateurs devant être positionnés exactement comme l'indiquent les manuels.

### Attention :

– Pour pouvoir encore afficher du texte à l'écran il est nécessaire que TEXT1 ne soit pas commuté.

– Après un appel d'une routine du moniteur, il faut faire très attention à la nouvelle position des commutateurs.

– S'il est nécessaire de commuter la mémoire auxiliaire aussi bien en lecture qu'en écriture, il semble que seules les séquences RAMWRT ON et RAMRD ON, puis RAMRD OFF et RAMWRT OFF évitent les "plantages".

– Les seules commutations possibles sont celles indiquées dans les manuels, il faut alors s'assurer du bon positionnement de tous les commutateurs afin d'éviter un "plantage".

– Après certains "plantages", la seule manière de tout remettre en ordre est d'éteindre et de rallumer l'U.C.

**Dernières précisions :** dans certains cas la routine AUXMOVE ne peut être utilisée. Il existe deux commutateurs mémoire assez peu connus SLOTCXROM et SLOT3ROM qui permettent d'accéder aux ROM spécifiques du //e-//c, ce qui oblige à effectuer une écriture en \$C00A (49162) avant toute utilisation de AUXMOVE. Etant donné que AUXMOVE n'utilise que RAMRD et RAMWRT, il faut que 80STORE soit OFF pour qu'elle puisse accéder à toute la mémoire.

## Visualisation de la page 1

Si vous avez un //e, la haute résolution étendue ne peut être visualisée qu'avec des cartes mères révision B ou postérieures. Une seconde condition doit être remplie : les deux Jumpers des cartes d'origine Apple doivent être reliés (ce problème n'existe pas avec la carte Chat Mauve). Les commutations à effectuer sont :

- AN3 OFF : \$C05E - 49246 (lecture)
- GR ON : \$C050 - 49232 (lecture)
- HIRES ON : \$C057 - 49239 (lecture)
- FULL ON (ou MIXED) : \$C052 - 49234 (lecture)
- 80COL ON : \$C00D - 49165 (écriture)

Pour utiliser à nouveau la haute résolution normale il suffit que le commutateur AN3 soit ON (lire en \$C05F - 49247).

Il existe aussi une page 2 en haute résolution 16 Ko. Pour la visualiser il faut également que 80STORE soit OFF. Cette deuxième page est difficilement utilisable, à moins d'écrire d'autres routines d'affichage en 80 colonnes, car le firmware 80 colonnes active toujours 80STORE.

## Création et sauvegarde

Pour créer un graphisme haute résolution 16 Ko en Basic, sans risquer un "plantage", il faut activer les commutateurs suivants afin de ne commuter que les pages TEXT1 et HGR1 :

- 80STORE ON : \$C001 - 49153 (écriture : modifie l'action de PAGE2)
- HIRES ON : \$C057 - 49239 (lecture)
- RAMWRT OFF : \$C004 - 49156 (écriture)
- RAMRD OFF : \$C002 - 49154 (écriture)
- PAGE2 ON : \$C055 - 49237 (écriture : commute la mémoire auxiliaire) ou PAGE2 OFF : \$C054 - 49236 (écriture : commute la mémoire principale)

Lorsque 80STORE est OFF, PAGE2 commute l'affichage entre les pages TEXT et HGR 1 ou 2. Lorsque 80STORE est ON, il permet de commuter soit la mémoire principale, soit la mémoire auxiliaire (pages TEXT1 et HGR1). Pour afficher un point en Basic il faut déterminer s'il se trouve en mémoire principale ou auxiliaire, effectuer la commutation adéquate, modifier son abscisse en

conséquence et utiliser le HPLLOT de l'Applesoft (structure des pages haute résolution : voir manuels). Pour sauvegarder et réutiliser l'image obtenue, la méthode la plus simple est d'utiliser les commandes BSAVE et BLOAD du DOS 3.3. Une image est enregistrée en deux parties : la première correspondant au contenu de la page HGR1 de la mémoire principale, et la seconde à celui de la page HGR1 de la mémoire auxiliaire. Pour comprendre parfaitement les opérations à effectuer, il est bon d'analyser le programme CREATION CERCLE HAUT qui crée une série de cercles concentriques.

Pour qu'une image puisse être chargée par le programme COPIE associé à l'utilitaire de Hard Copie, il faut que les fichiers associés se terminent par : " 1" (mémoire principale), " 2" (mémoire auxiliaire). Par exemple : CERCLE HAUT 1 et CERCLE HAUT 2.

**Conseils :** pour éviter les "plantages", utiliser HGR pour effacer la mémoire principale et CALL 62450 pour effacer la mémoire auxiliaire. En outre, il faut protéger la HGR1 par un HIMEM ou un LOMEM.

## Programme de Hard Copie

Le programme Basic COPIE sert seulement à charger une image en mémoire, à la visualiser, et à appeler la routine en assembleur après avoir positionné certains pointeurs. Cette routine tourne sur les configurations

suivantes : Apple //c ou //e et imprimante Image Writer. Les listings sont suffisamment commentés pour permettre une adaptation rapide à d'autres configurations. Il a les mêmes fonctionnalités qu'un programme de Hard Copie classique : impression de l'image telle qu'elle apparaît à l'écran ou en inverse, taille normale 560 x 192 points ou double 1120 x 384 points.

La routine est implantée à l'adresse \$4000 (16384), mais elle est parfaitement relogeable (elle utilise la page 3).

Une précision importante pour les possesseurs de //c : avant d'envoyer des informations à l'imprimante en langage machine par un JSR COUT, il est nécessaire d'effectuer un SEI.

La taille de l'image imprimée doit être spécifiée via l'adresse \$300. Seul l'état du 8ème bit est testé. Pour obtenir une taille normale : POKE 768,N avec N strictement inférieur à 128; pour obtenir une taille double, N doit être supérieur à 128.

L'aspect de cette image est spécifié via l'adresse \$301 par un POKE 769,N. Si N est strictement inférieur à 127 l'image est telle qu'elle apparaît à l'écran et si N est supérieur à 128, l'image est en "inverse".

Pour appeler la routine, il suffit, après avoir positionné les deux précédents pointeurs, de charger une image en mémoire, de mettre en marche l'imprimante et d'effectuer un CALL 16432 (ou CALL adresse de début d'assemblage + 48).

## Programme COPIE

```

10 REM IMPRESSION SUR DMP D'UNE IMAGE T
   HGR
20 D$ = CHR$(4) : POKE - 16368,0
30 PRINT D$;"PR#3": REM COMMUTATION 80C
   OLONNES
40 PRINT
50 HIMEM: 8100: REM PROTEGE LA ROUTINE
   ASSEMBLEUR
60 D$ = CHR$(4)
70 PRINT D$;"BLOAD HARD"
80 HOME
90 INVERSE
100 VTAB 8: HTAB 35: PRINT "IMPRESSION"
110 VTAB 11: HTAB 34: PRINT "D'UNE IMAGE
   "
120 VTAB 14: HTAB 28: PRINT "HAUTE RESOL
   UTION ETENDUE";
130 GET A$: REM ATTENTE D'UN CR POUR
   CONTINUER
140 NORMAL : HOME
150 VTAB 7: HTAB 28: PRINT "1) CHARGER U
   NE IMAGE THGR"
160 CALL 1002: REM POUR BIEN CONNECTER
   LE DOS
170 VTAB 10: HTAB 28: PRINT "2) VISUALIS
   ER UNE IMAGE THGR"

```

```

180 VTAB 13: HTAB 28: PRINT "3) IMPRIMER
   UNE IMAGE THGR"
190 VTAB 16: HTAB 28: PRINT "4) ARRET DU
   PROGRAMME
200 VTAB 23: INPUT "VOTRE REponse ( 1,2,
   3 OU 4 ) ";A$
210 REM VERIFICATION DE LA VALIDITE DE
   LA REponse
220 IF LEN(A$) < > 1 THEN 140
230 IF ASC(A$) > 52 OR ASC(A$) < 49
   GOTO 140
240 IF A$ = "4" THEN HOME : END
250 A = VAL(A$): ON A GOSUB 270,650,680
260 GOTO 140
270 HOME : VTAB 22: PRINT "POUR EFFECTUE
   R UN CATALOG REpondRE ? "
280 VTAB 23: PRINT "POUR RETOURNER A L'E
   CRAN PRECEDENT CR "
290 VTAB 12: HTAB 10: INPUT "NOM GENERIQ
   UE DE L'IMAGE A CHARGER ";A$
300 IF LEN(A$) = 0 THEN RETURN
310 IF (ASC(A$) < 65 OR ASC(A$) > 90
   ) AND A$ < > "?" THEN RETURN
320 IF A$ < > "?" THEN 410
330 HOME
340 REM CONNECTE LE DOS QUI PEUT ETRE D
   ECONNECTE PAR UN PR#3
350 REM SOUS DOS 3.3 ( PAS DE PROBLEME
   AVEC PRODOS )

```

```

360 CALL 1002
370 PRINT D$;"CATALOG"
380 PRINT : INPUT "PRESSER LA TOUCHE RET
URN POUR CONTINUER ";A$
390 GOTO 270
400 REM AFFICHAGE DE LA PAGE THGR
410 GOSUB 870
420 REM CHARGE LA PAGE HGR DE LA MEMOIR
E PRINCIPALE
430 POKE 49153,0: REM 80STORE ON
440 POKE 49236,0: REM PAGE2 OFF
450 POKE 49156,0: REM RAMWRT OFF
460 POKE 49154,0: REM RAMRD OFF
470 REM POSITIONNE LES COMMULATEURS ET
EFFACE LA THGR
480 REM UTILISER HGR POUR LA MEMOIRE
490 REM PRICIPALE ET CALL 62450 POUR LA
MEMOIRE AUXILIAIRE
500 HOME : HGR : POKE 49234,0: POKE 4923
7,0: CALL 62450
510 POKE 49236,0: REM COMMUTE LA MEMOIR
E PRINCIPALE
520 ONERR GOTO 600
530 PRINT D$;"BLOAD ";A$;" 1"
540 POKE 49237,0: REM COMMUTE LA MEMOIR
E AUXILIAIRE
550 PRINT D$;"BLOAD ";A$;" 2"
560 POKE 49236,0
570 POKE 216,0
580 HOME : TEXT
590 RETURN
600 REM ERREUR DISQUE
610 TEXT : HOME : VTAB 11: HTAB 20: INVE
RSE
620 PRINT "ACCES A CE FICHER IMPOSSIBLE
": NORMAL
630 VTAB 23: PRINT "APPUYEZ SUR UNE TOUC
HE POUR CONTINUER ": GET A$
640 POKE 216,0: GOTO 140
650 REM VISUALISATION D'UNE IMAGE
660 GOSUB 870
670 INPUT A$: HOME : TEXT : RETURN
680 REM IMPRESSION D'UNE IMAGE
690 HOME
700 VTAB 7: HTAB 28: PRINT "1) TAILLE 56
0*192, NORMALE"
710 VTAB 10: HTAB 28: PRINT "2) TAILLE 5
60*192, INVERSE"
720 VTAB 13: HTAB 28: PRINT "3) TAILLE 1
120*384, NORMALE"
730 VTAB 16: HTAB 28: PRINT "4) TAILLE 1
120*384, INVERSE"
740 VTAB 19: HTAB 28: PRINT "5) RETOUR A
U MENU GENERAL"
750 VTAB 23: INPUT "VOTRE REPONSE (1..5)
";A$
760 IF LEN (A$) < > 1 THEN 690
770 IF ASC (A$) < 49 OR ASC (A$) > 53
THEN 690
780 IF A$ = "5" THEN RETURN
790 A1 = 1: IF A$ = "3" OR A$ = "4" THEN
A1 = 2
800 A2 = 1: IF A$ = "2" OR A$ = "4" THEN
A2 = 2
810 REM EN 768 ET 769 SEULE LA VALEUR D
U 7EME BIT EST TESTEE

```

```

820 REM ( VALEUR ) OU ( A 128 )
830 POKE 768,0: IF A1 = 2 THEN POKE 768
,200
840 POKE 769,00: IF A2 = 2 THEN POKE 76
9,200
850 CALL 16432
860 RETURN
870 REM AFFICHAGE DE LA PAGE THGR
880 POKE 49160,0: REM ALTZP OFF
890 A = PEEK (49246): REM AN3 OFF
900 A = PEEK (49232): REM GR ON
910 A = PEEK (49239): REM HIRES ON
920 A = PEEK (49234): REM HIRES ENTIERE
930 POKE 49165,0: REM 80 COL ON
940 RETURN

```

## Programme CREATION CERCLE HAUT

```

10 REM TRACE DE 3 CERCLE EN HAUTE RESOL
UTION
20 D$ = CHR$ (4): PRINT D$;"PR#3": PRINT
30 REM AFFICHAGE DE LA TRES HAUTE RESOL
UTION
40 POKE 49160,0:A = PEEK (49246):A = P
EEK (49232):A = PEEK (49239)
50 A = PEEK (49165)
60 REM POSITIONNEMENT DES COMMULATEURS
70 POKE 49234,0: POKE 49153,0: POKE 4915
6,0: POKE 49154,0: POKE 49236,0
80 REM EFFACE LA THGR
90 HGR : POKE 49237,0: CALL 62450: POKE
49236,0: POKE 49234,0
100 HCOLOR= 3
110 FOR I = 1 TO 90 STEP 5
120 FOR J = 0 TO 6.29 STEP 0.5 / I
130 X = 270 + 2 * I * COS (J):B = 96 + I
* SIN (J)
140 GOSUB 250
150 NEXT J
160 NEXT I
170 REM SAUVEGARDE DE L'IMAGE CREE
180 PRINT D$;"BSAVE CERCLE HAUT 1,A$2000
,L$2000"
190 POKE 49237,0: PRINT D$;"BSAVE CERCLE
HAUT 2,A$2000,L$2000"
200 POKE 49236,0
210 TEXT
220 END
230 REM
240 REM ANALYSE DES COORDONNEES THGR :
PAGE1 OU 2 ET 560 ->280
250 XX = INT (X / 7):PG = XX - 2 * INT
(XX / 2)
260 XX = INT (XX / 2) * 7 + X - 7 * XX
270 POKE 49237,0: IF PG THEN POKE 49236
,0
280 IF XX > 279 THEN RETURN
290 REM AFFICHAGE DU POINT AVEC HPLLOT D
ANS LA BONNE PAGE MEMOIRE
300 HPLLOT XX,B
310 POKE 49236,0
320 RETURN

```

## Programme HARD.S Source Big Mac

```

1 *****
2 *
3 *
4 * TABLE DES LIGNES HGR *

```

```

5 *
6 *
7 *****
8 ORG $4000
9 ;ORGANISATION MEMOIRE

```

```

10 ;300 : Taille
11 ;301 : Inverse ou Normal
12 ;302 a 309 : Buffer Caracteres
13 ;30A : Ordre Ligne
14 ;30B : Ordre Octet
15 ;18 : octet a aller chercher
16 ;310 Partie basse ou haute ligne
17 ;311 : Pointeur de repetition horiz
    ontale
18      ORG    $4000
19 DEBTAB  HEX    00,20
20      HEX    80,20
21      HEX    00,21
22      HEX    80,21
23      HEX    00,22
24      HEX    80,22
25      HEX    00,23
26      HEX    80,23
27      HEX    28,20
28      HEX    A8,20
29      HEX    28,21
30      HEX    A8,21
31      HEX    28,22
32      HEX    A8,22
33      HEX    28,23
34      HEX    A8,23
35      HEX    50,20
36      HEX    D0,20
37      HEX    50,21
38      HEX    D0,21
39      HEX    50,22
40      HEX    D0,22
41      HEX    50,23
42      HEX    D0,23
43 FINTAB  ;Fin table HGR
44 *
45 *****
46 *
47 *
48 *      PROGRAMME PRINCIPAL
49 *
50 *
51 *****
52 ; Table des commutateurs memoire
53 AN3     EQU    $C05E
54 GR      EQU    $C050
55 HIRES   EQU    $C057
56 FULL    EQU    $C052
57 COL80   EQU    $C000
58 ON80ST  EQU    $C001
59 OFF80ST EQU    $C000
60 ONPG2   EQU    $C055
61 OFFPG2  EQU    $C054
62 OFFWT   EQU    $C004
63 OFFRD   EQU    $C002
64 ONTEXT  EQU    $C051
65 ; ADDRESSES SYSTEME
66 COUT    EQU    $FDED
67 ; POINTEURS PROGRAMME
68 ORLIG   EQU    $30A
69 OROCT   EQU    $30B
70 ADR     EQU    $18
71 INDIC   EQU    $30C
72 TYPE    EQU    $301
73 TAILLE  EQU    $300
74 HBLIG   EQU    $310
75 POINTE  EQU    $311
76 OUTPORT EQU    $FE95      ; SELECTIO
    N DES PERIPHERIQUES
77 *
    DE
78      CLD
79      SEI
80      STA  OFFPG2      ;Evite les

```

```

81      LDA    #$01      ; MISE EN
    ROUTE IMPR
82      JSR    OUTPORT
83      LDA    #$89
84      JSR    COUT
85      LDA    #"Z"
86      JSR    COUT
87      LDA    #$8D
88      JSR    COUT
89      LDA    #$9B      ; SELECTIO
    N DE L'INTERLIGNE
90      JSR    COUT
91      LDA    #$D4      ; CHARACTER
    E T
92      JSR    COUT
93      LDA    #$B1      ; NOMBRE 1
    6
94      JSR    COUT
95      LDA    #$B6
96      JSR    COUT
97      LDA    AN3      ; AFFICHAG
    E DE LA PAGE THGR
98      LDA    GR
99      LDA    HIRES
100     LDA    FULL
101     STA    COL80
102     STA    ON80ST      ;PREPARATI
    ON COMMUTATION
103     STA    OFFWT
104     STA    OFFRD
105     LDA    #$00      ; INITIALI
    SATION N LIGNE
106     STA    ORLIG
107     STA    POINTE
108     LDA    #$FF
109     STA    HBLIG
110 AUTRE BIT TAILLE
111     BPL    A1
112     LDA    HBLIG
113     EOR    #$FF
114     STA    HBLIG
115     BIT    HBLIG
116     BPL    A1
117     DEC    ORLIG
118 A1    LDA    #$8D
119     JSR    COUT
120     LDA    #$9B
121     JSR    COUT
122     LDA    #$F0
123     JSR    COUT
124     LDA    #$FF
125     STA    OROCT
126     LDA    #$00
127     STA    INDIC      ; INDICATE
    UR DE PAGE
128     LDA    #$9B      ; SELECT
    ION UNE LIGNE HGR
129     JSR    COUT
130     LDA    #$C7      ; CHARACTER
    E G
131     JSR    COUT
132     LDA    #$B1
133     JSR    COUT
134     LDA    #$B1
135     JSR    COUT
136     LDA    #$B2
137     JSR    COUT
138     LDA    #$B0
139     JSR    COUT
140     JMP    COMPL
141 COMPL LDA    #$FF      ;CHANGEMEN
    T DE PAGE
142     INC    OROCT      ; INCREMEN

```

```

TATION ORDRE OCTET
143      EOR  INDIC
144      STA  INDIC
145      STA  ONPG2
146      BNE  SELEC
147      STA  OFFPG2
148      DEC  OROCT      ;PAS INCRE
      MENTATION SI GOTO PG2
149 SELEC LDA  ORLIG      ; CALCUL D
      E L'ADRESSE A CHARGER
150      ASL
151      TAX
152      LDA  DEBTAB,X
153      STA  ADR
154      LDA  DEBTAB+1,X
155      BIT  TAILLE
156      BPL  STORE
157      BIT  HBLIG
158      BPL  STORE
159      CLC
160      ADC  #$10
161 STORE STA  ADR+1
162      LDX  #$0
163      LDY  OROCT
164      LDA  #$00
165      STA  POINTE
166 CHRG  LDA  (ADR),Y
167      STA  $302,X
168      INX
169      CPX  #$8
170      BEQ  FINCH
171      BIT  TAILLE
172      BPL  NON
173      LDA  POINTE
174      EOR  #$FF
175      STA  POINTE
176      BNE  CHRG
177 NON   LDA  #$04
178      CLC
179      ADC  ADR+1
180      STA  ADR+1
181      BNE  CHRG
182 FINCH LDY  #$07      ; MISE EN
      FORME DES COLONNES
183 L1   LDX  #$08
184 L2   LSR  $301,X
185      ROL
186      DEX

```

```

187      BNE  L2
188      STA  OFFPG2
189      BIT  TYPE
190      BPL  NOINV
191      EOR  #$FF
192 NOINV JSR  COUT
193      BIT  TAILLE
194      BPL  NORME
195      JSR  COUT
196 NORME DEY
197      BNE  L1
198      LDA  #$27
199      CMP  OROCT
200      BEQ  N1
201      JMP  COMPL
202 N1    LDA  #$00
203      CMP  INDIC
204      BEQ  N2
205      JMP  COMPL
206 N2    INC  ORLIG      ; INCREMENT
      TATION ORDRE LIGNE
207      LDA  #$03
208      STA  $FF
209      LDA  #$30
210      STA  $FE
211 LLL   LDA  #$00
212      JSR  COUT
213      DEC  $FE
214      BNE  LLL
215      DEC  $FF
216      BNE  LLL
217      LDA  #$18
218      CMP  ORLIG
219      BEQ  FIN
220      JMP  AUTRE
221 FIN   STA  OFFPG2      ; Position
      normale des switches
222      STA  ONTEXT
223      STA  OFF80ST
224      LDA  #$80
225      JSR  COUT
226      LDA  #$03      ; PR#3
227      JSR  OUTPORT
228      LDA  #$80
229      JSR  COUT      ; POUR BIE
      N EFFECTUER PR#3
230      RTS

```

## HARD

\*4000.41A0

```

4000- 00 20 80 20 00 21 80 21
4008- 00 22 80 22 00 23 80 23
4010- 28 20 A8 20 28 21 A8 21
4018- 28 22 A8 22 28 23 A8 23
4020- 50 20 00 20 50 21 00 21
4028- 50 22 00 22 50 23 00 23
4030- D8 78 80 54 C0 A9 01 20
4038- 95 FE A9 89 20 ED FD A9
4040- DA 20 ED FD A9 8D 20 ED
4048- FD A9 9B 20 ED FD A9 D4
4050- 20 ED FD A9 B1 20 ED FD

```

```

4058- A9 B6 20 ED FD AD 5E C0
4060- AD 50 C0 AD 57 C0 AD 52
4068- C0 8D 00 C0 8D 01 C0 8D
4070- 04 C0 8D 02 C0 A9 00 8D
4078- 0A 03 8D 11 03 A9 FF 8D
4080- 10 03 2C 00 03 10 10 AD
4088- 10 03 49 FF 8D 10 03 2C
4090- 10 03 10 03 CE 0A 03 A9
4098- 8D 20 ED FD A9 9B 20 ED
40A0- FD A9 F0 20 ED FD A9 FF
40A8- 8D 08 03 A9 00 8D 0C 03
40B0- A9 9B 20 ED FD A9 C7 20
40B8- ED FD A9 B1 20 ED FD A9
40C0- B1 20 ED FD A9 B2 20 ED

```

```

40C8- FD A9 80 20 ED FD 4C D1
40D0- 40 A9 FF EE 0B 03 4D 0C
40D8- 03 8D 0C 03 8D 55 C0 D0
40E0- 03 8D 54 C0 CE 0B 03 AD
40E8- 0A 03 0A AA 8D 00 40 95
40F0- 18 8D 01 40 2C 00 03 10
40F8- 08 2C 10 03 10 03 18 69
4100- 10 35 19 A2 00 AC 0B 03
4108- A9 00 8D 11 03 B1 18 9D
4110- 02 03 E8 ED 08 F0 18 2C
4118- 00 03 10 0A AD 11 03 49
4120- FF 8D 11 03 D0 E7 A9 04
4128- 18 65 19 85 19 D0 DE A0
4130- 07 A2 03 5E 01 03 2A CA
4138- D0 F9 8D 54 C0 2C 01 03
4140- 10 02 49 FF 20 ED FD 2C
4148- 00 03 10 03 20 ED FD 88
4150- 00 0F A9 27 CD 0B 03 F0
4158- 03 4C D1 40 A9 00 CD 0C
4160- 03 F0 03 4C D1 40 EE 0A
4168- 03 A9 03 85 FF A9 30 85
4170- FE A9 00 20 ED FD C6 FE
4178- 00 F7 C6 FF D0 F3 A9 18
4180- CD 0A 03 F0 03 4C 82 40
4188- 8D 54 C0 8D 51 C0 8D 00
4190- C0 A9 8D 20 ED FD A9 03
4198- 20 95 FE A9 8D 20 ED FD
41A0- 60

```

## Editeur plein écran : EPE

Le Pacha

Apple II+, //e, //c

- o Listez vos programmes Basic en avant et en arrière.
- o Modifiez, insérez, effacez des caractères en plein écran sans relire les lignes.
- o Recherchez toute chaîne de caractères.
- o Choisissez vous-même les codes de contrôle d'EPE.
- o Modifiez EPE : le fichier source est sur la disquette.

150,00 F TTC franco (bon de commande page 78).

# Les pointeurs, suite...

Olivier Herz

Pour illustrer et compléter l'article du Pom's 17 sur les pointeurs, nous vous offrons un petit programme de démonstration de Régis Lardennois. Celui-ci vous permet de bien comprendre la gestion des variables dynamiques et de visualiser les adresses mémoires utilisées.

Le lecteur pourra aussi se reporter avec profit à deux autres articles du même auteur, qui sont une application un peu plus sophistiquée des notions décrites ci-dessous :

"Tableaux de taille déclarée en Pascal" (Pom's 6) et :

"Effacement de directory en Pascal" (Pom's 7).

Citons également pour mémoire les "PEEKs et POKEs en Pascal" de Jacques PINO (Pom's 12); c'est une sorte de "magouille" dont la compréhension n'est pas immédiate.

## Utilisation

Il suffit d'exécuter le programme et de répondre par un chiffre de 1 à 7 au menu.

- 1- destruction tableaux 1 et 2
- 2- création tableau 1
- 3- initialisation tableau 1
- 4- création tableau 2
- 5- initialisation tableau 2
- 6- visualisation tableau 1
- 7- visualisation tableau 2

## Explications

Laurence Tichkowsky nous a présenté les pointeurs comme étant des variables dynamiques; la place pour les stocker n'est pas réservée de façon immuable par la compilation, mais au contraire cette place est dévolue à l'exécution au moment où l'on crée un nouvel élément du pointeur (instruction NEW).

Nous allons maintenant voir que le Pascal UCSD permet à un programme de se débarrasser en cours d'exécution des variables dynamiques dont il n'a plus besoin. Il utilise pour cela, outre NEW, deux procédures du système UCSD : MARK et RELEASE.

Le Pascal utilise une sorte de pile pour gérer ses pointeurs; la procédure NEW(P) crée P et le place au sommet de la pile.

La PROCEDURE MARK(VAR PILEPTR: ^INTEGER) va permettre à l'Apple de "mémoriser" l'adresse du sommet de la pile, en faisant un MARK(PTR) où PTR est de type ^INTEGER.

Inversement, la PROCEDURE RELEASE(VAR PILEPTR: ^INTEGER)

```
( LE PROGRAMME CI APRES EST UNE APPLICATION TRES )
( SIMPLE DES POINTEURS AVEC VISUALISATION DES )
( DES ADRESSES MEMOIRE UTILISEES )
( IL PERMET DE BIEN COMPRENDRE LE FONCTIONNEMENT )
( DES PROCEDURES PASCAL UCSD DE GESTION DES )
( VARIABLES DYNAMIQUES NEW MARK et RELEASE )
```

```
(*R-*)
```

```
PROGRAM TESTPOINTEUR;
```

```
TYPE ARTICLE = RECORD NUMERO:INTEGER;
                  NOM :STRING ;
                END;
TABLEAU = ARRAY[0..9] OF ARTICLE;
POINTEUR = ^TABLEAU;
PTI = ^INTEGER;
```

```
VAR ART: ARTICLE;
    HEAP1,HEAP2 :PTI;
    POINT1,POINT2 :POINTEUR;
    CL : CHAR;
    I : INTEGER;
```

```
FUNCTION PTARTICLE (PT:POINTEUR):INTEGER;
VAR X:RECORD CASE BOOLEAN OF
    TRUE : (PTR:POINTEUR);
    FALSE : (INT:INTEGER);
END;
```

```
BEGIN
    X.PTR:=PT;
    PTARTICLE:=X.INT;
END;
```

```
FUNCTION PTINTEGER (PT:PTI):INTEGER;
VAR X:RECORD CASE BOOLEAN OF
    TRUE : (PTR:PTI);
    FALSE : (INT:INTEGER);
END;
```

```
BEGIN
    X.PTR:=PT;
    PTINTEGER:=X.INT;
END;
```

```
PROCEDURE MENU;
```

```
BEGIN
    PAGE(OUTPUT);WRITE(CHR(28)); ( COMPATIBILITE APPLE II ET /// )
    WRITELN('PROGRAMME DE DEMONSTRATION DE L'USAGE DES POINTEURS');
    WRITELN;
    WRITELN('MEMOIRE DISPO',MEMAVAIL);
    WRITELN;
    WRITE (' HEAP1 = ',PTINTEGER(HEAP1),
           ' HEAP2 = ',PTINTEGER(HEAP2));
    WRITELN(' POINT1 = ',PTARTICLE(POINT1),
           ' POINT2 = ',PTARTICLE(POINT2));
    WRITELN;
    WRITELN('1 DESTRUCTION TABLEAUX 1 ET 2');
    WRITELN('2 CREATION TABLEAU 1');
    WRITELN('3 INITIAL TABLEAU 1');
    WRITELN('4 CREATION TABLEAU 2');
    WRITELN('5 INITIAL TABLEAU 2');
    WRITELN('6 VISUALIS TABLEAU 1');
    WRITELN('7 VISUALIS TABLEAU 2');
    WRITELN('ESC QUITTER');
    READ(CL);
END;
```

```
BEGIN
MARK(HEAP1);HEAP2:=HEAP1;POINT1:=NIL;POINT2:=NIL;
REPEAT MENU;
CASE CL OF
    '1':BEGIN RELEASE(HEAP1);HEAP2:=HEAP1;
            END;
    '2':BEGIN NEW(POINT1);MARK(HEAP2);END;
    '3':FOR I:=0 TO 9 DO
            BEGIN POINT1[I].NUMERO:=I;
                  POINT1[I].NOM :='TAB1';
```



# Puisque nous ne vous aviez besoin, nous

Apple et le logo Apple sont des marques déposées d'Apple Computer, Inc.



Version Calc

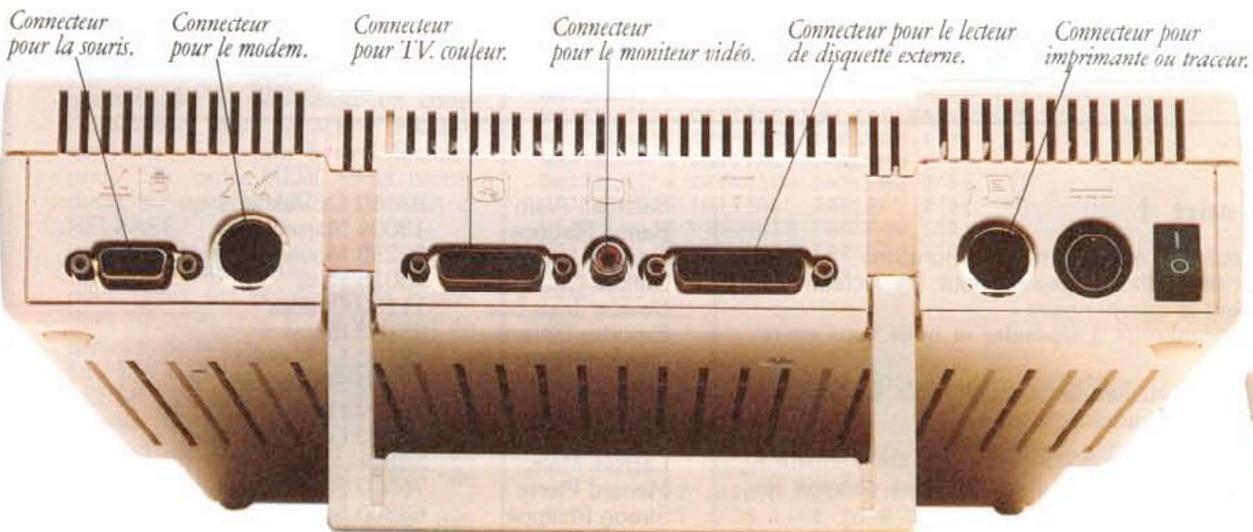


MousePaint

Une des plus grandes bibliothèques de logiciels programmes compatibles avec l'Apple IIe : jeu, gestion de base de données, analyse financière



Un clavier 63 touches type AZERTY comprenant une accentuation complète et des caractères majuscules/minuscules intégrés.



Connecteur pour la souris.

Connecteur pour le modem.

Connecteur pour TV. couleur.

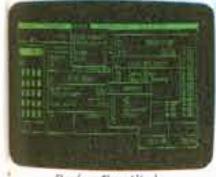
Connecteur pour le moniteur vidéo.

Connecteur pour le lecteur de disquette externe.

Connecteur pour imprimante ou traceur.

Voici comment  
l'écran de haute  
très facilement  
Si nous vous  
autant de mémoire  
en un seul appareil  
était indispensable

# savons pas de quoi vous avons tout donné.



Budget Familial



Apple Writer



Sorcellerie



Apple Works



Omnis II

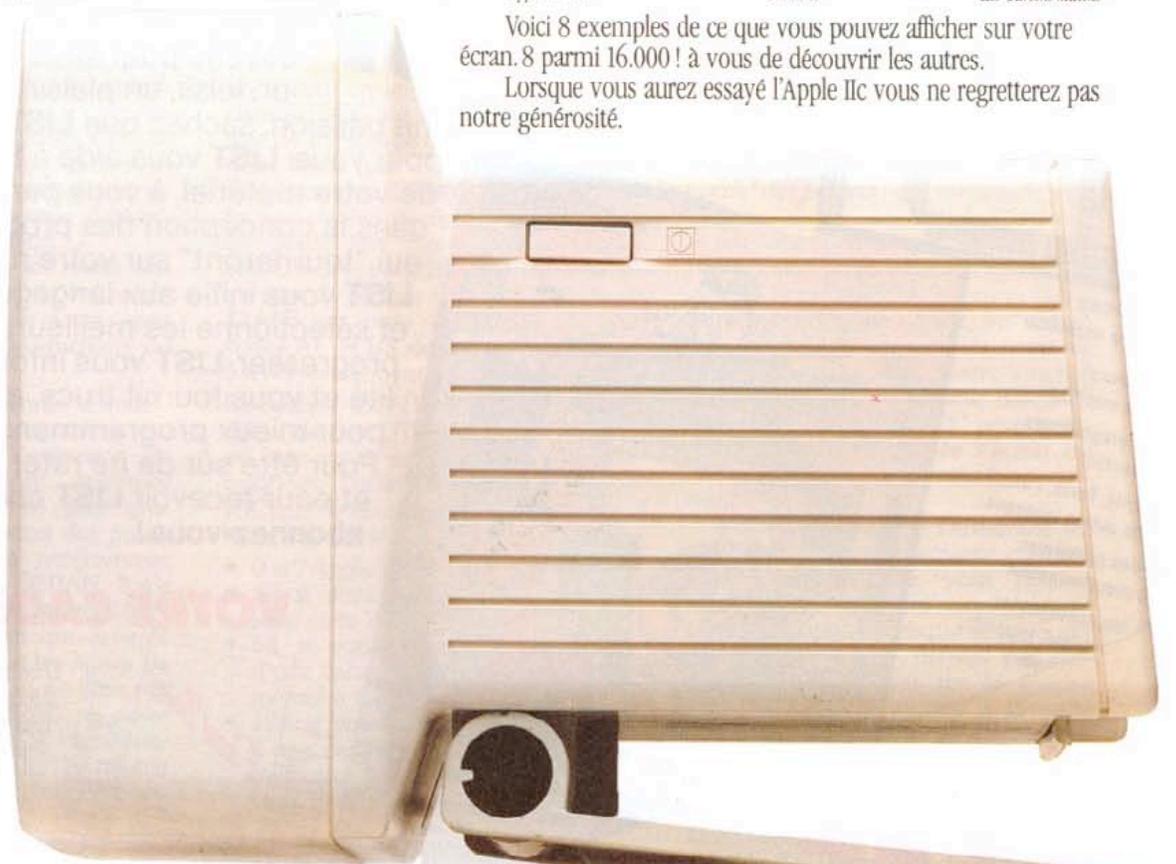


Les Oursins malins

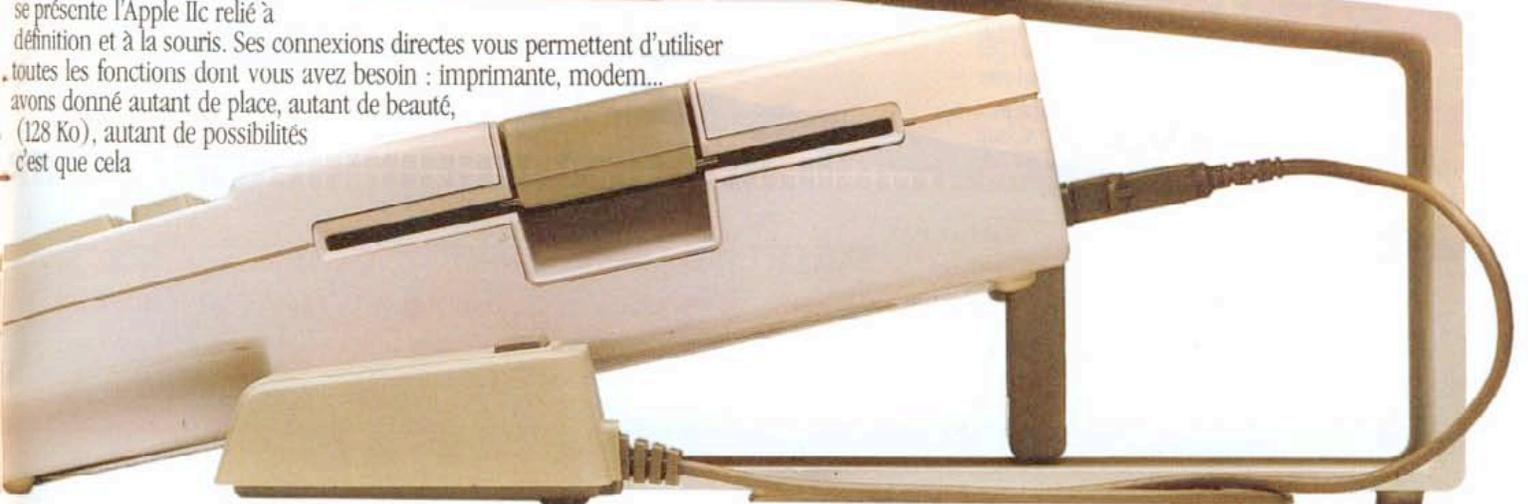
au monde; 16.000  
traitement de texte,  
planification.

Voici 8 exemples de ce que vous pouvez afficher sur votre  
écran. 8 parmi 16.000 ! à vous de découvrir les autres.

Lorsque vous aurez essayé l'Apple IIc vous ne regretterez pas  
notre générosité.



se présente l'Apple IIc relié à  
définition et à la souris. Ses connexions directes vous permettent d'utiliser  
toutes les fonctions dont vous avez besoin : imprimante, modem...  
avons donné autant de place, autant de beauté,  
(128 Ko), autant de possibilités  
c'est que cela



Apple présente l'Apple IIc.



# LIST

## LE JOURNAL DES AMATEURS DE PROGRAMMATION

### LE JOURNAL DES AMATEURS DE PROGRAMMATION

# Si

programmer un ordinateur est devenu pour vous un loisir, un plaisir...

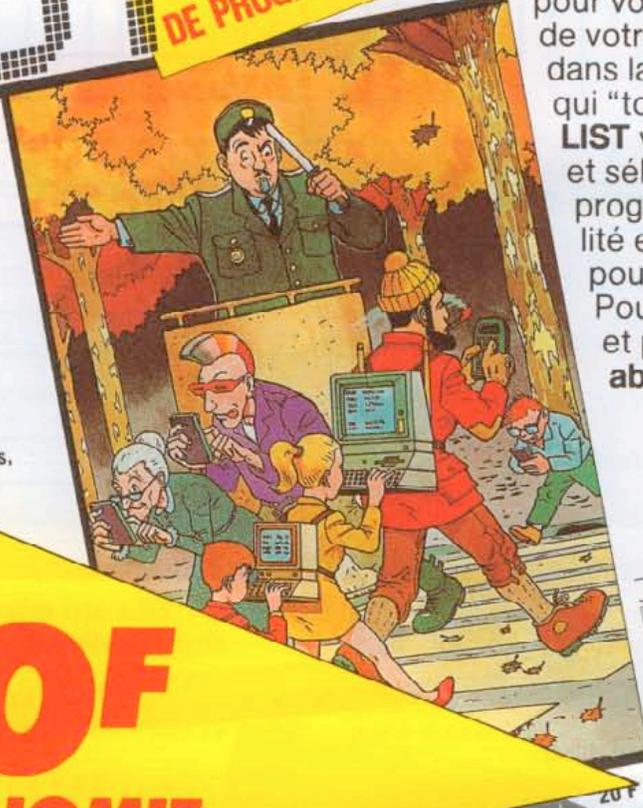
une passion, sachez que **LIST** a été créé pour vous. **LIST** vous aide à tirer davantage de votre matériel, à vous perfectionner dans la conception des programmes qui "tourneront" sur votre machine.

**LIST** vous initie aux langages informatiques et sélectionne les meilleurs livres pour progresser. **LIST** vous informe de l'actualité et vous fournit trucs, astuces et idées pour mieux programmer...

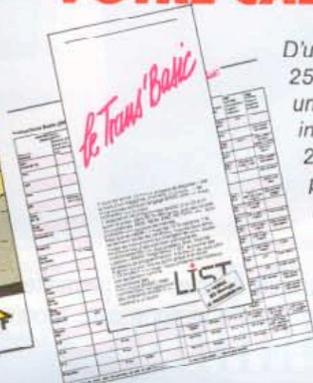
Pour être sûr de ne rater aucun numéro et pour recevoir **LIST** chez vous, **abonnez-vous!**

#### Dans chaque numéro :

- les "trucs" de votre machine.
- des programmes commentés.
- Basic, Forth, Pascal et les autres langages.
- des problèmes de programmation.
- les nouveautés : matériels, logiciels, programmes.



### VOTRE CADEAU!



D'un format pratique (11,5 x 25 cm), le **TRANS'BASIC** est une table de conversion des instructions **BASIC** des 21 micro-ordinateurs les plus courants. Ce livret de 24 pages vous permettra d'adapter sur votre ordinateur la plupart des programmes conçus pour les autres machines.

# 40F D'ECONOMIE

### BULLETIN D'ABONNEMENT

(à retourner à LIST - Service Abonnements - 5, place du Colonel Fabien - 75491 Paris Cedex 10)

Nom \_\_\_\_\_ Prénom \_\_\_\_\_

Adresse \_\_\_\_\_

Ville \_\_\_\_\_ Code postal | | | | | Pays \_\_\_\_\_

Veuillez m'abonner pour 10 numéros au prix avantageux de 160 F\* au lieu de 200 F - prix total au numéro - et m'adresser en cadeau le **TRANS'BASIC**. Je joins mon règlement indispensable libellé à l'ordre de **LIST**.

\* Belgique: 1330 FB; Suisse: 50 FS; Canada: 30 \$C; autres pays: 210 FF. Par avion: Afrique Francophone: 245 FF; Amérique, autre Afrique, Océanie: 305 FF; Asie: 355 FF. Belgique: Soumillion, 28, av. Massenet, 1190 Bruxelles. Versement Société Générale 2100405 835-39. Suisse: 18, route du Grand-Mont, CH 1052, Le Mont-sur-Lausanne, versement Caisse d'Epargne et de Crédit, 10-2418 Le Mont CH 1052, compte courant n° 650156-7. Canada: LMPI, 9345, rue de Meaux, St Léonard (Québec), H1R 3H3, Canada. Autres pays: 5, place du Colonel Fabien, 75491 Paris Cedex 10.

# Rendez le DOS transparent

Jean-Pierre Januel

Les programmes décrits dans cet article ne fonctionnent qu'avec un Apple //e muni d'une carte 80 colonnes.

Ayant réalisé quelques programmes qui n'utilisent l'Apple qu'avec des logiciels standards, j'ai été amené à rechercher une transparence maximale du logiciel et en particulier du DOS. Ainsi les utilisateurs ne sont pas désorientés et retrouvent le style de présentation auquel ils sont habitués (80 colonnes, utilisation des touches "Pomme", menus...) et n'ont quasiment rien à apprendre avant d'employer mes programmes.

Les programmes Basic MENU, CAT PLEIN ÉCRAN et EDITCAT permettent à l'utilisateur :

- De sélectionner un fichier dans le catalogue de sa disquette, sans avoir à taper son nom, pour traitement ultérieur par un programme (lecture, mise à jour, transformation...).

- D'afficher à l'écran les noms des fichiers du catalogue et de les modifier directement en plein écran, sans avoir à connaître les instructions du DOS.

SALUT est un programme d'initialisation; MENU est le programme qui vous guidera lors de l'utilisation (ils pourront vous donner quelques idées) et DEMO est un petit programme de démonstration des possibilités offertes par les programmes MENU et CAT PLEIN ÉCRAN. Il n'a pas la prétention d'être un modèle (il utilise un INPUT élargi qui accepte les virgules, mais refuse les lignes de plus de 256 caractères et ne filtre pas les caractères de contrôle). A noter que la ligne 300 teste le commutateur 80 colonnes afin de le rétablir s'il a été annulé pendant l'impression.

## Programme d'initialisation

Certaines routines en langage machine peuvent être exploitées par plusieurs programmes, ici c'est le cas de :

- La routine de reconstitution du contexte en cas d'erreur (voir le manuel Applesoft), indispensable dans tout programme qui s'adresse à des non informaticiens : un message d'erreur, c'est la honte !

- La routine RWTS et ses paramètres (voir le manuel du DOS, p.90).

- La routine Catalog --> Tableau publiée dans Pom's 9.

- La routine de comptage du nombre de secteurs libres (inspirée de celle parue dans Pom's 10).

- La routine d'INPUT élargi (tirée de Pom's 5).

Il est avantageux d'implanter, en une seule fois, toutes ces routines en mémoire (par exemple en page 3) par le biais du programme de salutation; ainsi, on gagne du temps et de la place (programmes plus courts).

Pour rendre le programme plus lisible et plus facile à modifier, la méthode d'implantation par S.H.LAM (publiée entre autres dans Pom's 2) peut être adoptée (comme ici pour la routine de comptage des secteurs libres) mais elle est lente. On gagne quelques secondes en chargeant (par BLOAD) un fichier binaire contenant toutes les routines (ici ROUTINES) ou encore en implantant directement dans le DOS de la disquette la routine de comptage du nombre de secteurs libres (par exemple en utilisant le programme DIRAC de Pom's 10; ne pas oublier de forcer à zéro le bit de poids fort des adresses : ainsi \$ADC3 devient \$2DC3 et se trouve en piste 1, secteur C; \$BA6B devient \$3A6B en piste 0, secteur 4; les codes AE et B3 des octets \$3A6D et \$3A79 deviennent : 2E et 33).

La routine IDENT.APPLE, lancée au début de SALUT, vérifie que l'Apple est bien un //e équipé d'une carte 80 colonnes; l'octet résultant stocké en \$3CF (975 en décimal) prend les valeurs suivantes :

- 0 si l'Apple utilisé n'est pas un //e.
- 32 si vous travaillez avec un //e sans carte 80 colonnes.
- 64 si votre Apple //e est muni d'une carte 80 colonnes mais sans mémoire auxiliaire.
- 128 si votre Apple //e est équipé d'une carte 80 colonnes et d'une mémoire auxiliaire (Source : Traux Pratiques Applesoft, p.200).

Remarque : ROUTINES réutilise les zones mémoires précédemment occupées par IDENT.APPLE.

## Système d'aiguillage de la disquette

Pour les utilisateurs non initiés, la disquette programme doit représenter un ensemble fermé et cohérent : le "boot" débouche sur un menu d'aiguillage auquel renverront également tous les programmes lorsqu'ils auront accompli leurs tâches. Le programme MENU assure cette fonction; il signale le moment d'introduction de la disquette, demande le numéro du lecteur, avertit clairement en cas de fausse manoeuvre, et (en ligne 350) POKE dans les adresses 923 à 953 le nom du programme (précédé de

la longueur de ce nom) choisi par l'utilisateur lorsque ce programme nécessite un passage préalable par le sous-programme CAT PLEIN ÉCRAN.

MENU utilise la technique du menu à barres en inversion vidéo; c'est la plus reposante pour l'utilisateur en l'absence de souris !

## Choix sur catalogue en 80 colonnes

CAT PLEIN ÉCRAN offre les possibilités suivantes :

- Il affiche le catalogue de votre disquette en utilisant tout l'espace disponible en 80 colonnes : 2 fichiers sur la même ligne (un enregistrement fichier occupe 37 caractères), soit 44 noms de fichiers au maximum sur l'écran.

- Il vous permet de travailler en plein écran : vous pouvez déplacer le curseur (une barre en inversion vidéo est positionnée, au départ, sur le premier nom de la liste) sur les noms de fichiers en utilisant les quatre touches de déplacement du curseur (flèches), ou encore en tapant "D" pour revenir au premier nom de l'écran ou "F" pour aller directement au dernier.

- Vous avez accès à l'ensemble du catalogue : si votre disquette contient plus de 44 fichiers, vous pouvez consulter la suite du catalogue (3 écrans maximum puisque le DOS accepte au plus 105 fichiers par disquette) en appuyant sur la touche "S" (Suite), puis revenir à la page précédente en tapant "P" (Précédent écran).

- Vous pouvez limiter l'affichage à un seul type de fichiers : les fichiers TEXT dans l'exemple de DEMO; la variable TS (ligne 370 de MENU) signale cette option à CAT PLEIN ÉCRAN, qui sélectionne les fichiers voulus (ligne 320).

Lorsque vous avez amené le curseur sur le nom de votre fichier, appuyez sur RETURN pour le sélectionner. Le programme qui va utiliser le fichier choisi est alors lancé; il PEEKera dans les adresses 923 et suite le nom du fichier sélectionné, que CAT PLEIN ÉCRAN y aura POKé.

La dernière ligne de l'écran est un menu aide mémoire qui rappelle la liste des touches utilisables et leur signification :

- Flèches (pour déplacer le curseur dans les 4 directions).
- D (pour début de l'écran).
- F (pour fin de l'écran).
- S (suite du catalogue, s'il occupe

- plus d'un écran).
- P (précédent écran s'il existe).
- RETURN pour le choix du fichier.

### Nom des principales variables

- PG\$: nom du programme sur lequel il faudra enchaîner.
- A\$(I) : enregistrements du catalogue (37 caractères chacun suivi d'un retour chariot).
- F\$(I) : nom de fichier (= MID\$(A\$(I),8,30))
- NF : Nombre de fichiers dans le catalogue.
- KT : Nombre de fichiers sélectionnés (si l'on désire seulement les fichiers textes par exemple).
- SL : Nombre de secteurs libres.
- NL : Numéro du fichier dans sa colonne.
- IO : Nombre de fichiers dans les colonnes précédentes (y compris écrans précédents).
- NN : Numéro du fichier dans son écran.
- X : code ASCII (+ 128) de la touche frappée par l'utilisateur.
- MV : déplacement (nombre de lignes).
- PH : position horizontale.
- PV : position verticale.

Remarque : la ligne 250 simule le ONERR GOTO (non actif avec RWTS) en récupérant le code erreur de RWTS en \$321 et en le "traduisant" en code erreur du DOS : POKE 222,8.

### Variante

On pourrait aussi utiliser ceci comme un programme de HELLO : il ne s'agit plus alors de sélectionner un fichier à faire traiter par un programme, mais un programme à lancer : CAT PLEIN ECRAN doit donc être intégré au programme SALUT (Attention ! la routine Catalog --> Tableau ne fonctionne que si le tableau A\$(N) est le premier déclaré et initialisé), et MENU n'a plus de raison d'être, pas plus que les lignes 180 et 850 à 870 de CAT PLEIN ECRAN (PEEK et POKE du nom de fichier); les routines de traitement d'erreur (lignes 30 à 70 et 240 à 260) deviennent également inutiles; supprimer aussi le suffixe 'D'LE' après CATALOG en ligne 290; la fin de programme devient :

```
840 PRINT : TEXT : HOME
850 IF MID$(A$(NN),2,1) = "A"
THEN PRINT D$ "RUN" A$(NN)
860 IF MID$(A$(NN),2,1) = "B"
THEN PRINT D$ "BRUN" A$(NN)
870 IF MID$(A$(NN),2,1) = "T"
THEN PRINT D$ "EXEC" A$(NN)
```

Ces deux dernières lignes supposent que vous connaissiez le contenu des fichiers : à vos risques et périls... (le programme ainsi transformé ne s'adresse évidemment plus aux non

initiés); vous pouvez cependant remettre en début de programme un petit sous-programme de traitement d'erreur du genre :

```
30 CALL 768 : POKE 216,0 :
Y = PEEK(222)
40 IF Y = 13 THEN VTAB12 :
PRINT "Erreur sur le type de fichier.
Ret --> autre essai ou
Esc --> Applesoft"; : GET Z$:
PRINT : IF Z$ = CHR$(13) THEN
280
50 END
```

Vous pouvez même prévoir que RETURN lance un RUN ou BRUN et "Espace" (par exemple) un LOAD ou BLOAD.

### Edition plein écran sur catalogue

Visicalc et Applewriter offrent des possibilités d'action assez limitées sur le catalogue d'une disquette. FID n'est intéressant que pour des opérations en série sur des fichiers; quoi qu'il en soit, l'instruction la plus malaisée à utiliser est le RENAME, qui oblige à taper en entier le nom de l'ancien fichier et le nouveau.

Le programme EDITCAT remédie à ces lacunes :

- Il offre toutes les fonctions de déplacement de CAT PLEIN ECRAN.

- Vous pouvez non seulement effectuer les opérations classiques d'effacement, de verrouillage et déverrouillage de fichiers, mais aussi aller modifier directement le nom d'un fichier comme si vous travailliez avec un traitement de textes de type Applewriter.

- De nombreuses sécurités sont prévues : le programme refuse d'obéir si vous essayez de franchir les limites de l'écran ou du catalogue, il vous avertit si vous avez oublié de fermer le lecteur, si votre disquette n'est pas formatée sous DOS 3.3, si vous essayez d'effacer ou de modifier un fichier verrouillé et même si vous essayez d'intervenir sur le "trou" laissé par un fichier effacé, ou si vous avez, par mégarde, rebaptisé un fichier avec un nom déjà attribué.

Quand vous l'appellez, le programme commence par vérifier que vous avez bien mis une disquette DOS 3.3 dans le lecteur choisi et refermé la porte (pour ce faire, il va lire le VTOC de la disquette en utilisant RWTS et vérifie le numéro de la version du DOS : 4 ème octet du VTOC), puis il charge le catalogue de la disquette et affiche les 44 premiers noms de fichiers.

Sur la première ligne de l'écran, il précise le nombre total de fichiers de la disquette et le nombre de secteurs libres.

La dernière ligne rappelle la liste des touches que vous pouvez utiliser

pour accomplir les actions désirées :

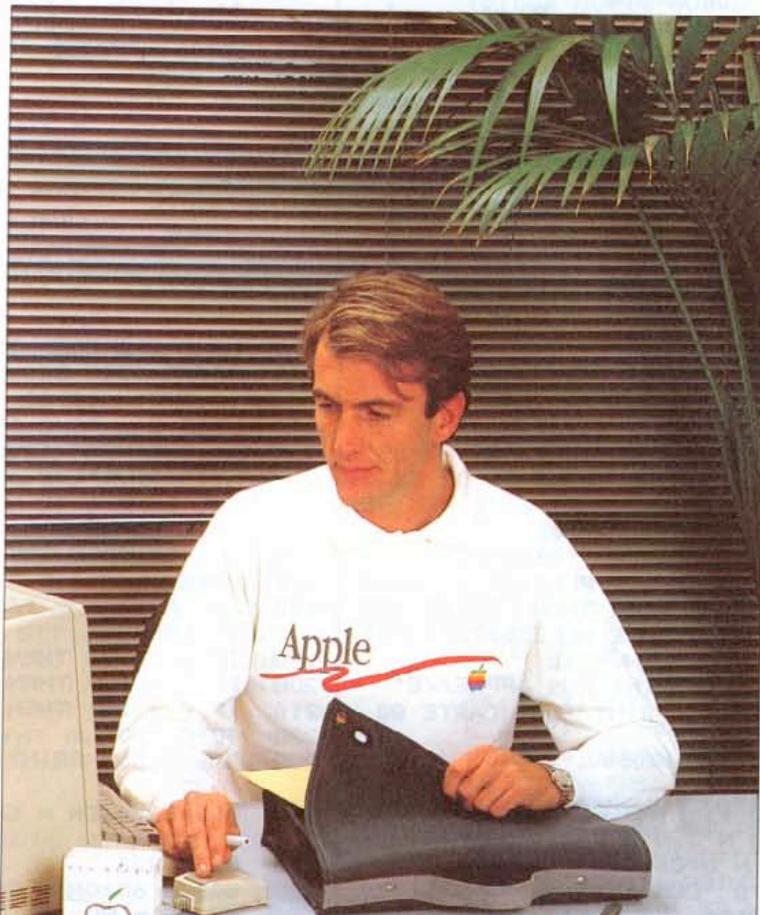
- Flèches (pour le déplacement du curseur dans les 4 directions).
- D (début : appuyer sur la touche D ou d pour aller au premier nom en haut à gauche de l'écran).
- F (fin : pour aller au dernier nom de l'écran).
- S (suite : pour aller à l'écran suivant).
- P (pour revenir à l'écran précédent).
- M (pour modifier le nom du fichier sur lequel se trouve le curseur (voir plus loin le sous-menu de cette option).
- E (pour effacer le fichier : un RENAME est d'abord tenté pour vérifier que le fichier n'est pas verrouillé; il vous est ensuite demandé, par message en haut d'écran, confirmation ou annulation : O (oui) / N(on)).
- V (pour verrouiller un fichier).
- W (pour le déverrouiller).
- C (pour compacter l'affichage, éliminer les "trous" laissés par les fichiers effacés, et remettre à jour les compteurs de nombre de fichiers et de secteurs libres : en fait, il s'agit tout simplement de recharger le catalogue mis à jour).
- RETURN (Retour chariot) indique la fin de travail sur ce catalogue : le programme demande si vous désirez passer à une autre disquette (RETURN) ou retourner au menu général de la disquette (par la touche ESC).

### Quelques précisions sur la modification

Cette fonction affiche dans le coin supérieur gauche de l'écran le mot "Modification", remplace le curseur barre par un curseur caractère et affiche sur la dernière ligne un nouveau menu :

- Flèches : déplacement du curseur à l'intérieur du nom de fichier.
- Pomme Ouverte-X : remplace le caractère sous le curseur par le caractère tapé (ici X).
- Pomme Pleine-Y : insère le caractère Y avant le curseur; cette option ne fonctionne pas sur les 29 ème et 30 ème caractères : utiliser le remplacement dans ce cas.
- Del : la touche DELETE vous permet d'effacer le caractère qui se trouve sous le curseur.
- ESC : en appuyant sur cette touche, vous annulez toutes les modifications effectuées sur ce nom de fichier; l'ancien nom est réaffiché et vous retournez au mode "Déplacement".
- RETURN permet de confirmer le nouveau nom de fichier (RENAME, après avoir vérifié que le nouveau nom n'est pas déjà attribué); vous êtes alors ramené au mode "Déplacement".

# Signe de reconnaissance



**Sac sport :** toile enduite marine, biais gris.  
Sangles + bandoulière amovible, ton sur ton.  
Grand poche extérieure fermant par glissière.  
Fond caoutchouc isolant.  
Longueur totale : 70 cm.  
Largeur : 30 cm.

**Sac de voyage :** caoutchouc souple "high tec", gris doux. Sangles ton sur ton. Fermeture par glissière. Grande poche intérieure. Marquage bagage.

**Attaché-case :** Assorti au sac de voyage, souple et confortable. Une poche intérieure compartimentée. Fermeture par deux boutons noirs. Marquage bagage.

Les « goodies » dans le jargon Apple, ce sont tous les accessoires à s'offrir ou à offrir pour (se) faire plaisir. La collection 85 vous propose une gamme sport et voyage. Nous espérons que ces nouveautés sauront vous séduire. Elles ont été créées dans l'esprit Apple, pour vous qui partagez avec nous une certaine façon de vivre, de travailler, de bouger...

Désignation	Quantité	Prix unitaire	Prix total
Sac sport bleu		285,00 F	
Sac voyage gris		285,00 F	
Attaché-case gris		230,00 F	

Montant total de la commande. Frais de port inclus \_\_\_\_\_ T.T.C.

Règlement par chèque joint à l'ordre de APPLE SEEDRIN, et adressé à :  
« Boutique Apple » Avenue de l'Océanie ZA de Courtabœuf - BP 131 - 91944 Les Ulis Cedex

Cette commande est à envoyer à :

Nom : \_\_\_\_\_ Adresse : \_\_\_\_\_

Des conditions particulières sont réservées aux adhérents du Club Apple. Souhaitez-vous recevoir des informations sur le Club Apple : OUI  NON



Apple

Le programme contrôle le bon respect des règles du DOS concernant les noms de fichiers : il transforme les minuscules en majuscules, refuse les caractères non alphabétiques comme première lettre du nom, les virgules et même les caractères de contrôle qui poseraient des problèmes à l'affichage.

Note : de nombreux messages ont été prévus pour prendre en compte le travail avec un seul lecteur de disquette : si vous en avez deux, vous pouvez alléger notablement les programmes en affectant le premier à la disquette programme et le second à la disquette fichiers. Vous pouvez alors éliminer les messages du style

"Insérez la disquette programme (ou : fichier)..." ou "Erreur ! Ce n'est pas la disquette programme !", etc.

## Utilisation des pages 0 à 3

### Routines en langage machine

IDENT.APPLE : A\$2D4,L\$FB  
 ROUTINES : A\$300,L\$9B  
 - \$300 (768) : reconstitution du contexte en cas d'erreur.  
 - \$30B (779) : lancement de RWTS.  
 - \$314 à \$328 : paramètres RWTS.  
 - \$316 (790) : numéro de lecteur;  
 - \$321 (801) : code erreur  
 - \$32A (810) : routine Catalog -->

### Tableau

- \$340 (832) : INPUT élargi

### Paramètres divers

- \$06-\$07 : nombre de secteurs libres  
 - \$08 : numéro de lecteur  
 - \$09 : mode d'affichage (40 ou 80)  
 - \$10 : numéro de choix dans le menu.  
 - \$11 : option "fichiers textes uniquement" pour CAT PLEIN ECRAN

### Nom de programme ou de fichier

- \$39B (923) : longueur du nom  
 - \$39C à \$3BA : nom du fichier ■

## Programme SALUT

```
10 REM          PROGRAMME D'INITIALISATION
20 Y$ = "ADC3:20 6B BA N BA6B:20 2F AE A9
    00 85 06 85 07 A0 C8 18 B9 F2 B3
    F0 0E 0A 90 FB 48 E6 06 D0 02 E6 0
    7 68 18 90 F0 88 D0 E9 60": REM C
    omptage secteurs libres
30 D$ = CHR$(4): REM      D$=CTRL-D
40 PRINT D$"BRUN IDENT.APPLE":AP = PEEK
    (975): IF AP < 64 THEN PRINT "DE
    SOLE: CE PROGRAMME NECESSITE": PRI
    NT TAB(10): INVERSE
50 IF AP = 0 THEN PRINT "UN APPLE//E"
60 IF AP = 32 THEN PRINT "UNE CARTE 80
    COLONNES"
70 IF AP < 64 THEN NORMAL : END
80 PRINT D$"PRÉ3"
90 PRINT : VTAB 1: HTAB 32: INVERSE : PR
    INT " B O N J O U R ";: NORMAL
100 Y$ = Y$ + " N D9C6G"
110 FOR J = 1 TO LEN (Y$): POKE 511 + J
    , ASC ( MID$ (Y$,J,1)) + 128: NEXT
    J: POKE 72,0: CALL - 144
120 PRINT : PRINT D$"BLOAD ROUTINES": CA
    LL 832
130 PRINT : PRINT D$"RUN MENU"
140 END
```

## Programme MENU

```
10 REM          --- MENU ---
20 GOTO 90
30 MA = INT ((78 - LEN (T$)) / 2): PRIN
    T : GOSUB 40: PRINT TAB(MA);IV$*
    "T$" "NO$"
40 PRINT TAB(MA): INVERSE : FOR I = 1
    TO LEN (T$) + 4: PRINT " " : NEX
    T : NORMAL : PRINT : RETURN
50 VTAB 2 * CH + 7: POKE 1403, INT ((80
    - LEN (C$(CH))) / 2): RETURN
60 GOSUB 50: PRINT C$(CH): RETURN
70 CALL 768: POKE 216,0: ONERR GOTO 70
80 Y = PEEK (222): IF Y = 8 THEN PRINT
    G$IV$"ATTENTION!"NO$" Porte lecteu
    r disq.ouverte ou disq.vierge ou a
    bimée.: PRINT "Fermez la porte ou
    changez la disquette puis appuyez
    sur une touche.": GET Z$: PRINT
    : HOME : RESUME
90 D$ = CHR$(4): REM Ctrl-D
100 R$ = CHR$(13): G$ = CHR$(7): REM
    Return & Ctrl-G
110 IV$ = CHR$(15): NO$ = CHR$(14): RE
    M INVERSE et NORMAL
```

```
120 HOME : ONERR GOTO 70
130 GOSUB 450: GOSUB 30: PRINT TAB(MA)
    ;IV$* "ST$" "NO$": GOSUB 40
140 VTAB 24: POKE 1403,18: PRINT IV$*FIè
    ches"NO$" "IV$*Choix par Return"NO
    $" "IV$*Esc --> Menu Général"NO$;
150 FOR CH = 1 TO NC: GOSUB 50: PRINT C$
    (CH): NEXT
160 CH = 1: GOTO 230
170 POKE - 16368,0
180 C = PEEK (- 16384): IF C < 128 THEN
    180
190 IF C = 138 THEN 240: REM      Bas
200 IF C = 139 THEN 260: REM      Haut
210 IF C = 141 THEN 280: REM      Return
220 GOTO 170
230 GOSUB 50: PRINT IV$C$(CH)NO$: GOTO 1
    70
240 GOSUB 60: CH = CH + 1: IF CH > NC THE
    N CH = 1
250 GOTO 230
260 GOSUB 60: CH = CH - 1: IF CH = 0 THEN
    CH = NC
270 GOTO 230
280 IF CH < 1 OR CH > NC THEN PRINT G$:
    GOTO 170
290 POKE - 16368,0
300 PRINT : HOME : POKE 10,CH:TS = 0
310 ON CH GOTO 320,330,380,320,380,440
320 T$ = "AFFICHAGE FICHER TEXTE": GOTO
    340
330 T$ = "IMPRESSION FICHER TEXTE"
340 PG$ = "DEMO": HOME : GOSUB 30
350 LP = LEN (PG$): POKE 923,LP: FOR I =
    1 TO LP: POKE 923 + I, ASC ( MID$
    (PG$,I,1)): NEXT
360 COL = 80: IF CH = 4 THEN COL = 40
370 POKE 9,COL:TS = 1: POKE 11,TS: PRINT
    D$"RUN CAT PLEIN ECRAN,D1": REM
    TS=Option "fichiers TEXT seuls"
380 PRINT : VTAB 10: PRINT "Insérez votr
    e disquette fichier puis donnez le
    n" de lecteur.": GET Z$:NL = VA
    L (Z$): IF NL < > 1 AND NL < > 2
    THEN PRINT G$: GOTO 380
390 IF CH = 3 THEN POKE 8,NL: PRINT : P
    RINT D$"RUN EDITCAT,D1"
400 PRINT : HOME : PRINT TAB(26)IV$*CA
    TALOGUE DE VOTRE DISQUETTE"NO$"
    Tapez 'Retour chariot'"
410 PRINT TAB(64)"pour continuer."
420 POKE 34,2
430 PRINT D$;"CATALOG,D*NL: VTAB 2: POKE
    1403,31: PRINT PEEK (6) + 256 *
    PEEK (7);" Secteurs libres": GET
```

```

Z$: TEXT : HOME : GOTO 130
440 END
450 T$ = "DISQUETTE":ST$ = "
-M e n u- " :NC = 6
460 C$(1) = "Afficher un fichier texte "
470 C$(2) = "Imprimer un fichier texte"
480 C$(3) = "Remanier le catalogue d'une
disquette"
490 C$(4) = C$(1) + "en 40 col.":C$(1) =
C$(1) + "en 80 col."
500 C$(5) = "Voir le catalogue de votre d
isquette"
510 C$(6) = "Arrêter et passer en Basic":
RETURN

```

## Programme EDICAT

```

10 REM --- EDITEUR DE CATALOGUE
PLEIN-ECRAN ---
20 GOTO 360
30 REM --- Traitement erreurs ---
40 CALL 768: POKE 216,0:Y = PEEK (222):
POKE - 16368,0: ONERR GOTO 40
50 IF Y = 8 THEN HOME : PRINT B$B$B$"At
tention! Porte lecteur disq.ouvert
e ou disq.vierge ou abimée.": PRIN
T "Fermez la porte ou changez la d
isquette puis appuyez sur une touc
he.": GET Z$: GOTO 100
60 IF Y = 4 THEN HOME : VTAB 12: PRINT
B$B$"Disquette protégée. Enlevez l
'étiquette.": GET Z$: GOTO 100
70 IF Y = 10 THEN GOSUB 180: PRINT B$B$
B$IV$"Attention! Fichier verrouill
é": GOSUB 110: RETURN
80 IF Y = 6 THEN PRINT B$B$"Erreur! Ce
n'est pas la disquette programme!":
GOTO 1170
90 GOTO 450
100 PRINT : HOME : ON LC GOTO 400,530,11
70
110 FOR I = 1 TO 1000: NEXT : GOSUB 150:
RETURN
120 REM --- Déplacements curseur ---
130 GOSUB 180
140 PRINT F$(NN + I0);;NN = NN + MV: IF
NN = NI THEN VTAB PV: POKE 1403,P
H: GOTO 160
150 GOSUB 180
160 PRINT IV$F$(NN + I0)NO$:
170 RETURN
180 IF NN < 23 THEN VTAB NN + 1: HTAB 8
190 IF NN > 22 THEN VTAB NN - 21: POKE
1403,47
200 RETURN
210 IF NN < 23 THEN VTAB NN + 1: HTAB 1
220 IF NN > 22 THEN VTAB NN - 21: POKE
1403,40
230 RETURN
240 REM --- Ss-programmes de vérifi
cation & réinitialn ---
250 VTAB 20: PRINT R$D$"RENAME "F$(NN +
I0)", "F$(NN + I0): RETURN
260 IF F$(NN + I0) = "
THEN GOSUB 180: PRI
NT B$B$B$IV$"Fichier effacé"NO$:
GOSUB 110: POP
270 VTAB 20: RETURN
280 VTAB 24: POKE 1403,0: PRINT IV$"Flêc
hes"NO$ "IV$"Début"NO$ "IV$"F<i
n"NO$ "IV$"Suite"NO$ "IV$"P(réc
"NO$ "IV$"M(odif"NO$ "IV$"E(fficac
"NO$ "IV$"V(ern"NO$ "IV$"W(déver
r"NO$ "IV$"C(omp"NO$ "IV$"Ret'-'

```

```

>menu"NO$;: RETURN
290 REM --- Vérification pour éviter d
ouble attribution ---
300 FOR I = 1 TO NF
310 IF FF$ < > F$(I) THEN 330
320 GOSUB 180: PRINT B$B$IV$"Nom déjà at
tribué"NO$;: GOSUB 110: POP : GOTO
1250
330 NEXT
340 RETURN
350 REM --- PROGRAMME PRINCIPAL ---
360 IV$ = CHR$(15):NO$ = CHR$(14):B$
= CHR$(7):R$ = CHR$(13)
370 D$ = CHR$(4): REM Ctrl-D
380 LC = 1: ONERR GOTO 40
390 N = 105: DIM A$(N),F$(N)
400 ER = 0:KL = PEEK (8): POKE 790,KL: C
ALL 779
410 ER = PEEK (801): IF ER = 64 THEN PO
KE 222,8: GOTO 40
420 IF ER = 16 THEN POKE 222,4: GOTO 40
430 DOS = PEEK (8195): IF DOS < > 2 AND
DOS < > 3 THEN VTAB 12: PRINT B
$B$"SEULES LES DISQUETTES FORMATTE
ES SOUS "IV$"DOS 3.3 ou 3.2"NO$" S
ONT ACCEPTEES.": FOR I = 1 TO 2000
: NEXT : GOTO 1180
440 REM --- Affichage catalogue
---
450 HOME :A = FRE (0):A$ = "": FOR I =
1 TO 19:A$ = A$ + " ": NEXT : FOR
I = N TO 1 STEP - 1:A$(I) = A$ +
A$: NEXT :A$(0) = A$ + "":A = PEE
K (107) + PEEK (108) * 256 + 8: P
OKE 254, PEEK (A): POKE 255, PEEK
(A + 1)
460 PRINT R$D$"NOMONCIO": POKE 54,42: PO
KE 55,3: POKE 56,60: POKE 57,3: CA
LL 1002: PRINT D$"CATALOG,D"KL: PR
INT D$"PRE0": PRINT D$"INÉ0"
470 NF = 0: FOR I = 1 TO N: IF MID$(A$(
I),2,1) < > " " THEN NF = NF + 1:
NEXT
480 SL = PEEK (6) + 256 * PEEK (7)
490 PRINT D$"PRE3":LC = 2
500 PRINT TAB(28)IV$" CATALOGUE "NO$"
"IV$NF;NO$" fichiers "IV$S
L;NO$" secteurs libres": PRINT : P
OKE 34,1
510 GOSUB 280
520 POKE 35,23: HOME
530 I0 = 0:NN = 1
540 NL = NF - I0: IF NL > 22 THEN NL = 22
550 FOR I = 1 TO NL:II = I + I0: VTAB I
+ 1
560 POKE 1403,0: IF I0 = 22 OR I0 = 66 T
HEN POKE 1403,40
570 PRINT LEFT$(A$(II),7);
580 F$(II) = MID$(A$(II),8,30): PRINT F
$(II);
590 NEXT
600 IF I0 = 22 OR I0 = 66 THEN I0 = I0 -
22: GOTO 640
610 IF NF - I0 < 23 THEN 640
620 NL = NF - I0 - NL: IF NL > 22 THEN NL
= 22
630 I0 = I0 + 22: VTAB 2: GOTO 550
640 NN = 1: VTAB 2: HTAB 8: PRINT IV$F$(N
+ I0)NO$
650 REM --- Choix utilisateur ---
660 POKE - 16368,0
670 X = PEEK ( - 16384): IF X < 128 THEN
670

```

```

680 IF X = 136 THEN GOSUB 890
690 IF X = 138 THEN GOSUB 860
700 IF X = 139 THEN GOSUB 840
710 IF X = 141 THEN 1100
720 IF X = 149 THEN GOSUB 910
730 IF X > 224 AND X < 251 THEN X = X -
    32
740 IF X = 195 THEN 450
750 IF X = 196 THEN GOSUB 950
760 IF X = 197 THEN GOSUB 960
770 IF X = 198 THEN GOSUB 1030
780 IF X = 205 THEN GOSUB 1220
790 IF X = 208 AND I0 > 0 THEN 1080
800 IF X = 211 AND NF > 10 + 44 THEN 105
    0
810 IF X = 214 THEN GOSUB 1010
820 IF X = 215 THEN GOSUB 1020
830 GOTO 660
840 IF NN = 1 THEN 1070
850 MV = - 1; NI = 22; PV = 23; PH = 7: GOS
    UB 130: RETURN
860 IF NN = NF - 10 THEN RETURN
870 IF NN = 44 THEN X = 211: RETURN
880 MV = 1; NI = 23; PV = 2; PH = 47: GOSUB
    130: RETURN
890 IF NN < 23 THEN RETURN
900 MV = - 22; NI = 0: GOSUB 130: RETURN
910 IF NN > 22 THEN RETURN
920 IF NF - 10 < 23 THEN RETURN
930 MV = 22: IF NN + MV > NF - 10 THEN R
    ETURN
940 NI = 0: GOSUB 130: RETURN
950 MV = 1 - NN; NI = 0: GOSUB 130: RETURN

960 GOSUB 260: Y = 0: GOSUB 250: IF Y = 1
    0 THEN RETURN
970 VTAB 1: POKE 1403,0: PRINT IV$"Effac
    er ? Confirmez (O/N)"NO$; GET Z$:
    IF Z$ < > "0" AND Z$ < > "N" TH
    EN 970
980 VTAB 1: POKE 1403,0: PRINT "
    ";
990 IF Z$ = "N" THEN RETURN
1000 PRINT R$D$"DELETE "F$(NN + I0):F$(N
    N + I0) = "
    "; GOSUB 210: PRINT "
    "IV$F$(NN + I0)NO$; RETURN
1010 GOSUB 260: PRINT R$D$"LOCK "F$(NN +
    I0): GOSUB 210: PRINT "*";: RETUR
    N
1020 GOSUB 260: PRINT R$D$"UNLOCK "F$(NN
    + I0): GOSUB 210: PRINT " ";: RET
    URN
1030 MV = NF - NN - I0: IF NF > 44 + I0 T
    HEN MV = 44 - NN
1040 NI = 0: GOSUB 130: RETURN
1050 IF I0 > 66 THEN 660
1060 I0 = I0 + 44: HOME : GOTO 540
1070 IF I0 = 0 THEN RETURN
1080 I0 = I0 - 44: HOME : GOTO 540
1090 REM --- Fin traitement ---
1100 HOME : VTAB 10: HTAB 1: PRINT "Voul
    ez-vous passer à une autre disquet
    te ou retourner au menu général ?
    "; VTAB 24: POKE 1403,0: CALL - 8
    68: PRINT IV$"Return" = autre dis
    quette"NO$" "IV$"Esc" = menu"NO$;
1110 POKE - 16368,0
1120 X = PEEK ( - 16384)
1130 IF X < 128 THEN 1120
1140 IF X < > 141 AND X < > 155 THEN 1
    110
1150 POKE - 16368,0: PRINT : TEXT : HOM
    E

```

```

1160 IF X = 141 THEN 1190
1170 LC = 3: IF KL = 1 THEN VTAB 10: PRI
    NT "Insérez la disquette programme
    en D1 puis faites 'Return'";: GET
    Z$
1180 PRINT R$D$"RUN MENU,D1"
1190 VTAB 10: PRINT "Insérez votre disqu
    ette et donnez le n° du lecteur";:
    GET Z$: PRINT : KL = VAL (Z$): IF
    KL < > 1 AND KL < > 2 THEN 1190
1200 POKE 8, KL: CLEAR : GOTO 360
1210 REM --- Modification nom fic
    hier ---
1220 GOSUB 260: Y = 0: GOSUB 250: IF Y =
    10 THEN RETURN
1230 VTAB 1: HTAB 1: PRINT IV$"Modificat
    ion"NO$;
1240 VTAB 24: POKE 1403,0: PRINT IV$"-->
    "NO$" "IV$"<--"NO$" "IV$"PomOuv=Re
    mplacer"NO$" "IV$"PomPline=Insérer"
    NO$" "IV$"Del=Effacer"NO$" "IV$"Es
    c=Annuler"NO$" "IV$"Ret=Confirmer
    "NO$;
1250 FF$ = F$(NN + I0): PH = 7
1260 Y$ = "": IF PH > 7 THEN Y$ = LEFT$
    (FF$, PH - 7)
1270 Z$ = "": IF PH < 36 THEN Z$ = RIGHT
    $ (FF$, 36 - PH)
1280 X$ = MID$ (FF$, PH - 6, 1): GOSUB 180
    : PRINT Y$IV$X$NO$Z$;
1290 POKE - 16368,0
1300 X = PEEK ( - 16384): PO = PEEK ( -
    16287): PP = PEEK ( - 16286)
1310 IF X < 128 THEN 1300
1320 IF X = 136 AND PH > 7 THEN PH = PH
    - 1: GOTO 1260
1330 IF X = 149 THEN 1440
1340 IF X = 155 THEN 1460
1350 IF X = 141 THEN GOSUB 300: VTAB 20
    : PRINT R$D$"RENAME "F$(NN + I0)",
    "FF$:F$(NN + I0) = FF$: GOTO 1460
1360 IF X < 160 THEN 1290
1370 IF X = 255 THEN FF$ = Y$ + Z$ + CH
    R$ (160): GOTO 1260
1380 IF X > 224 AND X < 251 THEN X = X -
    32
1390 IF X = 172 THEN 1290
1400 IF PH = 7 AND (X > 218 OR X < 193)
    THEN GOSUB 180: PRINT B$IV$"Comme
    ncez par une lettre!"NO$;: GOSUB 1
    10: GOTO 1260
1410 IF PO > 127 THEN FF$ = Y$ + CHR$ (
    X) + Z$: GOTO 1440
1420 IF PP > 127 AND PH < 35 THEN FF$ =
    Y$ + CHR$ (X) + X$ + LEFT$ (Z$,
    LEN (Z$) - 1): GOTO 1440
1430 GOTO 1290
1440 IF PH < 36 THEN PH = PH + 1
1450 GOTO 1260
1460 VTAB 1: POKE 1403,0: PRINT "
    ";: GOSUB 280: GOSUB 150: RET
    URN

```

## Programme CAT PLEIN ECRAN

```

10 REM --- CATALOGUE PLEIN-ECRAN
20 GOTO 180
30 CALL 768: POKE 216,0: ONERR GOTO 30:
    POKE - 16368,0
40 Y = PEEK (222): IF Y = 8 THEN PRINT
    B$B$B$"Attention! Porte lecteur di
    sq.ouverte ou disq.vierge ou abimé

```

```

e.": PRINT "Fermez la porte ou cha
ngez la disquette puis appuyez sur
une touche.": GET Z$: PRINT : HO
ME : IF ER = 64 THEN 230
50 IF Y = 8 THEN RESUME
60 IF Y = 6 THEN VTAB 15: PRINT B*B*"Er
reur! Ce n'est pas la disquette pr
ogramme!": GOTO 880
70 GOTO 280
80 REM --- Déplacements curseur --
-
90 GOSUB 140
100 PRINT F*(NN + I0);:NN = NN + MV: IF
NN = NI THEN VTAB PV: POKE 1403,P
H: GOTO 120
110 GOSUB 140
120 PRINT IV*F*(NN + I0)NO$;
130 RETURN
140 IF NN < 23 THEN VTAB NN + 1: HTAB 8
150 IF NN > 22 THEN VTAB NN - 21: POKE
1403,47
160 RETURN
170 REM --- PROGRAMME PRINCIPAL -
---
180 LP = PEEK (923): FOR I = 1 TO LP:X =
PEEK (923 + I):PG$ = PG$ + CHR$(
X): NEXT
190 ONERR GOTO 30
200 IV$ = CHR$(15):NO$ = CHR$(14):B$
= CHR$(7):R$ = CHR$(13)
210 D$ = CHR$(4):REM Ctrl-D
220 N = 105: DIM A$(N),F$(N)
230 VTAB 10: PRINT "Insérez votre disque
tte fichier puis donnez le n" de l
ecteur.": GET Z$:LE = VAL(Z$):
IF LE < 1 OR LE > 2 THEN PRINT B
$: GOTO 230
240 ER = 0: POKE 8,LE: POKE 790,LE: CALL
779: REM Lecture du UTOC par RWTS
250 ER = PEEK (801): POKE 801,0: IF ER =
64 THEN PRINT : POKE 222,8: GOTO
30
260 DOS = PEEK (8195): IF DOS < > 3 THE
N VTAB 15: PRINT B*B*"SEULES LES
DISQUETTES FORMATTEES SOUS "IV*"DO
S 3.3"NO$ SONT ACCEPTEES.": FOR I
= 1 TO 2000: NEXT : TEXT : HOME :
PRINT D$"RUN MENU,D1"
270 REM --- Affichage catalogue ---
280 HOME :A = FRE (0):A$ = "": FOR I =
1 TO 19:A$ = A$ + " ": NEXT : FOR
I = N TO 1 STEP - 1:A$(I) = A$ +
A$: NEXT :A$(0) = A$ + "":A = PEE
K (107) + PEEK (108) * 256 + 8: P
OKE 254, PEEK (A): POKE 255, PEEK
(A + 1)
290 PRINT R*D$"NOMONCIO": POKE 54,42: PO
KE 55,3: POKE 56,60: POKE 57,3: CA
LL 1002: PRINT D$"CATALOG,D"LE: PR
INT D$"PR£0": PRINT D$"IN£0"
300 FOR I = 1 TO N: IF MID$(A$(I),2,1)
< > " " THEN NF = NF + 1: NEXT
310 KT = NF:TS = PEEK (11): IF TS < > 1
THEN 340
320 KT = 0: FOR I = 1 TO NF: IF MID$(A$(
I),2,1) = CHR$(212) THEN KT = K
T + 1:A$(KT) = A$(I)
330 NEXT
340 SL = PEEK (6) + 256 * PEEK (7)
350 PRINT D$"PR£3"
360 IF KT = 0 THEN PRINT B*"Erreur: il
n'y a pas de fichier texte sur cet
te disquette.": GOTO 230
370 PRINT TAB(28)IV$ "CATALOGUE "NO$

```

```

"IV$NF;NO$" fichiers "IV$S
L;NO$" secteurs libres": PRINT : P
OKE 34,1
380 VTAB 24: PRINT IV$* -->, <--, D(ébut
, F(in, S(uite, P(récédent écran
- choix fichier : 'Return'"NO$;
390 POKE 35,23: HOME
400 IO = 0:NI = 1:NF = KT
410 NL = NF - IO: IF NL > 22 THEN NL = 22
420 FOR I = 1 TO NL:II = I + IO: VTAB I
+ 1
430 POKE 1403,0: IF IO = 22 OR IO = 66 T
HEN POKE 1403,40
440 PRINT LEFT$(A$(II),7);
450 F$(II) = MID$(A$(II),8,30): PRINT F
$(II);
460 NEXT
470 IF IO = 22 OR IO = 66 THEN IO = IO -
22: GOTO 510
480 IF NF - IO < 23 THEN 510
490 NL = NF - IO - NL: IF NL > 22 THEN NL
= 22
500 IO = IO + 22: VTAB 2: GOTO 420
510 NI = 1: VTAB 2: HTAB 8: PRINT IV*F*(N
N + I0)NO$
520 REM --- Choix utilisateur ---
530 POKE - 16368,0
540 X = PEEK (- 16384): IF X < 128 THEN
540
550 IF X = 136 THEN GOSUB 710
560 IF X = 138 THEN GOSUB 680
570 IF X = 139 THEN GOSUB 660
580 IF X = 141 THEN 840
590 IF X = 149 THEN GOSUB 730
600 IF X > 224 AND X < 251 THEN X = X -
32
610 IF X = 196 THEN GOSUB 770
620 IF X = 198 THEN GOSUB 780
630 IF X = 208 AND IO > 0 THEN 830
640 IF X = 211 AND NF > IO + 44 THEN 800
650 GOTO 530
660 IF NI = 1 THEN 820
670 MV = - 1:NI = 22:PV = 23:PH = 7: GOS
UB 90: RETURN
680 IF NI = NF - IO THEN RETURN
690 IF NI = 44 THEN X = 211: RETURN
700 MV = 1:NI = 23:PV = 2:PH = 47: GOSUB
90: RETURN
710 IF NI < 23 THEN RETURN
720 MV = - 22:NI = 0: GOSUB 90: RETURN
730 IF NI > 22 THEN RETURN
740 IF NF - IO < 23 THEN RETURN
750 MV = 22: IF NI + MV > NF - IO THEN R
ETURN
760 NI = 0: GOSUB 90: RETURN
770 MV = 1 - NI:NI = 0: GOSUB 90: RETURN
780 MV = NF - NI - IO: IF NF > 44 + IO TH
EN MV = 44 - NI
790 NI = 0: GOSUB 90: RETURN
800 IF IO > 66 THEN 530
810 IO = IO + 44: HOME : GOTO 410
820 IF IO = 0 THEN RETURN
830 IO = IO - 44: HOME : GOTO 410
840 NI = NI + IO: POKE - 16368,0: TEXT :
HOME
850 FOR I = LEN (F$(NI)) TO 1 STEP - 1
: IF MID$(F$(NI),I,1) < > CHR$(
160) THEN LP = 1: POKE 923,LP: G
OTO 870
860 NEXT
870 FOR I = 1 TO LP: POKE 923 + I, ASC (
MID$(F$(NI),I,1)): NEXT
880 IF LE = 1 THEN PRINT : VTAB 12: PRI
NT "Insérez la disquette programme

```

```

en D1 puis faites 'Return'; GET
Z$
890 PRINT R$D$"RUN "PG$","D1"

```

## Programme DEMO

```

10 D$ = CHR$(4):B$ = CHR$(7)
20 ONERR GOTO 40
30 GOTO 120
40 CALL 768: POKE 216,0
50 Y = PEEK(222): IF IP = 2 THEN PRINT
D$"PR£0"
60 PRINT D$"CLOSE "FI$: ONERR GOTO 40
70 IF Y = 5 THEN INVERSE: PRINT "FIN D
E FICHER": NORMAL:F = 1: GOTO 28
0
80 IF Y = 6 AND F = 1 THEN HOME: PRINT
B$"Erreur: ce n'est pas la disque
tte programme!": PRINT:NL = 1: GO
TO 280
90 IF Y = 6 THEN PRINT B$"Erreur: le fi
chier "FI$: PRINT "n'est pas sur c
ette disquette.": GOTO 160
100 IF Y = 8 THEN HOME: PRINT B$"Ferme
z la porte du lecteur de disq.SVP!
"; GET Z$: PRINT: RESUME
110 HOME: GOTO 280
120 LP = PEEK(923): FOR I = 1 TO LP:FI$
= FI$ + CHR$(PEEK(923 + I)):
NEXT
130 COL = PEEK(9):IP = PEEK(10): IF I
P = 2 THEN COL = 40
140 IF COL = 40 THEN PRINT CHR$(12);
CHR$(21): REM Désactive carte
80 col.
150 HOME: HTAB INT((COL - 9 - LP) / 2
): INVERSE: PRINT " FICHER "FI$:
NORMAL: POKE 34,1
160 NL = PEEK(8): PRINT: IF NL = 1 THE
N PRINT "Insérez la disquette fic
hier en D"NL: PRINT "puis appuyez
sur 'return': GET Z$: PRINT
170 PRINT D$"VERIFY"FI$,"D"NL
180 IF IP = 2 THEN PRINT "Imprimante pr
ête?": GET Z$: PRINT: PRINT D$"
PR£1"
190 PRINT: HOME: PRINT D$"OPEN"FI$,"D"
NL
200 PRINT D$"READ"FI$
210 & INPUT A$: PRINT A$: IF PEEK(37)
< 22 OR IP = 2 THEN 210
220 PRINT D$: INVERSE: PRINT "'RETURN'=
CONTINUER OU 'ESC'=ARRETER: "; NO
RMAL
230 POKE - 16368,0
240 Z = PEEK(- 16384): IF Z < 128 THEN
240
250 IF Z = 155 THEN PRINT: F = 1: Y = 0:
GOTO 60
260 IF Z = 141 THEN PRINT: HOME: GOTO
200
270 GOTO 230
280 POKE - 16368,0: IF NL = 1 THEN PRI
NT "Insérez disquette programme en
D1 puis "; IF COL = 40 THEN PRI
NT
290 PRINT "'RETURN' pour revenir au menu
"; GET Z$: PRINT
300 IF PEEK(- 16353) < 128 THEN PRIN
T D$"PR£3": REM Si 80COL pas e
n fonction, le commuter
310 TEXT: PRINT D$"RUN MENU,D1"

```

## IDENT.APPLE

```

02D4- 08 78 AD 00
02D8- E0 8D D0 02 AD 00 D0 8D
02E0- D1 02 AD 00 D4 8D D2 02
02E8- AD 00 D8 8D D3 02 AD 81
02F0- C0 AD 81 C0 AD B3 FB C9
02F8- 06 D0 49 AD 17 C0 30 3C
0300- AD 13 C0 30 27 AD 16 C0
0308- 30 22 A0 2A BE A2 03 B9
0310- 00 00 96 00 99 A2 03 88
0318- D0 F2 4C 01 00 08 A0 2A
0320- B9 A2 03 99 00 00 88 D0
0328- F7 68 B0 08 A9 80 8D CF
0330- 03 4C 49 03 A9 40 8D CF
0338- 03 4C 49 03 A9 20 8D CF
0340- 03 4C 49 03 A9 00 8D CF
0348- 03 AD 00 E0 CD D0 02 D0
0350- 18 AD 00 D0 CD D1 02 D0
0358- 10 AD 00 D4 CD D2 02 D0
0360- 08 AD 00 D8 CD D3 02 F0
0368- 38 AD 88 C0 AD 00 E0 CD
0370- D0 02 F0 06 AD 80 C0 4C
0378- A1 03 AD 00 D0 CD D1 02
0380- F0 06 AD 80 C0 4C A1 03
0388- AD 00 D4 CD D2 02 F0 06
0390- AD 80 C0 4C A1 03 AD 00
0398- D8 CD D3 02 F0 03 AD 80
03A0- C0 28 60 A9 EE 8D 05 C0
03A8- 8D 03 C0 8D 00 08 AD 00
03B0- 0C C9 EE D0 0E 0E 00 0C
03B8- AD 00 08 CD 00 0C D0 03
03C0- 38 B0 01 18 8D 04 C0 8D
03C8- 02 C0 4C 1D 03 EA 00 39

```

## ROUTINES

\*300.39B

```

0300- 68 A8 68 A6 DF 9A 48 98
0308- 48 60 00 A9 03 A0 14 20
0310- D9 03 60 00 01 60 02 00
0318- 11 00 25 03 00 20 00 00
0320- 01 00 FE 60 02 00 01 EF
0328- D8 00 38 48 84 FD A0 00
0330- 91 FE E6 FE D0 02 E6 FF
0338- 68 A4 FD 60 A9 A0 60 00
0340- A9 4C 8D F5 03 A9 50 8D
0348- F6 03 A9 03 8D F7 03 60
0350- C9 84 F0 03 4C 91 03 20
0358- B1 00 20 E3 DF 20 6C DD
0360- 85 85 84 86 20 6F FD 8A
0368- F0 1E BD FF 01 C9 83 F0
0370- 1A BD FF 01 29 7F 9D FF
0378- 01 CA D0 F5 A9 00 A0 02
0380- A2 8D 20 E9 E3 20 9A DA
0388- A2 00 60 20 3A FF 4C D0
0390- 03 20 84 FE 20 2D FF 4C
0398- D0 03 FF 02

```

# Recherche dans un tableau

Jean et Patrice Neveu

Voici un petit programme écrit en assembleur, d'une grande utilité, qui vous intéressera sans doute. Il permet de rechercher dans un tableau de chaînes, les chaînes comportant un mot ou groupe de signes alpha-numériques particulier. La réponse est fournie "instantanément".

## Programme de démonstration

- Il charge le sous programme en langage machine RECHERCHE OBJ.  
- Il crée un tableau A\$ de 500 groupes de 5 lettres choisies aléatoirement, précédant les numéros d'ordre (ce tableau peut être plus impor-

tant et occuper toute la mémoire disponible. Dans ce cas il serait bon d'ajouter le "FREE" rapide, proposé dans Pom's 2, afin d'accélérer le chargement du tableau).

- Il demande l'expression à rechercher. Cette expression peut être plus grande que les chaînes stockées et peut comporter des signes "=" pour remplacer les lettres inconnues.

- Il stocke à partir de l'adresse \$9500 l'expression à rechercher.

- Il stocke à l'adresse \$0A la longueur de l'expression à rechercher.

- Il appelle la routine assembleur située en 768 (\$300). Cette routine affiche alors toutes les chaînes contenant l'expression.

- Pour une nouvelle demande appuyer sur une touche.

- Pour sortir, appuyer sur RETURN lorsqu'une nouvelle expression est demandée.

Le tableau A\$ a été choisi arbitrairement. Si pour des raisons spécifiques, la recherche se faisait dans d'autres tableaux, il suffirait de POKER les adresses \$301 et \$305 pour initialiser le nom du tableau concerné (code ASCII de la première lettre en \$301, code de la deuxième lettre + \$80 en \$305).

La routine Recherche Text a été assemblée à l'aide de LISA 2.5. Les notes figurant sur le listing en expliquent le déroulement. ■

## Programme RECHERCHE DEMO

```
1 REM RECHERCHE DEMO
2 REM LE 16 AOUT 1984
3 HIMEM: 38144
4 TEXT : POKE 33,40: HOME : INVERSE : HT
  AB 14: PRINT "RECHERCHE DEMO": NOR
  MAL : POKE 34,5
6 PRINT CHR$(4)"BLOAD RECHERCHE OBJ"
10 PRINT : PRINT "PATIENTEZ QUELQUES INS
  TANTS S.V.P, JE CREE 500 CHAINES
  ..."
100 DIM A$(500): FOR I = 0 TO 500: FOR J
  = 1 TO 5:A$ = A$ + CHR$(65 + R
  ND (1) * 26): NEXT :A$(I) = A$ + "
  " + STR$(I):X% = PEEK (- 1633
```

```
6) + PEEK (- 16336):A$ = "": NEX
T
105 VTAB 15: HTAB 34: PRINT " 'CR'
  POUR
  ARRETER": POKE 33,33
110 HOME
310 INPUT "EXPRESSION A RECHERCHER ? ":A
  $
315 IF A$ = "" THEN 400
320 FOR I = 1 TO LEN (A$): POKE 38143 +
  I, ASC ( MID$( A$,I,1)): NEXT
330 POKE 10, LEN (A$)
340 CALL 768
350 GET Z$: PRINT : GOTO 310
360 END
```

## Source Lisa 2.5

### Programme RECHERCHE TEXT

```
1 ;
2 ORG $300
3 NOM1 EPZ $6 ;1ERE L
  ETRE DE LA VARIABLE A RECHERCHER
4 NOM2 EPZ $7 ;2EME L
  ETRE DE LA VARIABLE A RECHERCHER
5 ADTABB EPZ $8 ;ADRESS
  E BASSE DU TABLEAU
6 ADTABH EPZ $9 ;ADRESS
  E HAUTE DU TABLEAU
7 LONG EPZ $A ;LONGEU
  R DE LA CHAINE RECHERCHEE
8 ARYTAB EPZ $6B ;EMPLAC
  EMENT DE L'ADRESSE DU TABLEAU
9 LTB EPZ $18 ;LONGUE
  UR TABLEAU BAS
10 LTH EPZ $19 ;LONGUE
  UR TABLEAU HAUT
11 NELEMH EPZ $E8 ;NOMBRE
  D'ELEMENTS DANS TABLEAU HAUT
12 NELEMB EPZ $EC ;NOMBRE
```

```
D'ELEMENTS DANS TABLEAU BAS
13 LGELEM EPZ $1D ;LONGUE
  UR DE L'ELEMENT
14 ADELEB EPZ $1E ;ADRESS
  E DE DEBUT DE L'ELEMENT BAS
15 ADELEH EPZ $1F ;ADRESS
  E DU DEBUT DE L'ELEMENT HAUT
16 CELEM EPZ $CF ;COMPTE
  UR ELEMENT
17 CLET EPZ $CE ;COMPTE
  UR DE LETTRE
18 LGELM1 EPZ $ED
19 ADELB1 EPZ $EE
20 ADELH1 EPZ $EF
21 INVFLG EPZ $32
22 COUT EQU $FDED ;AFFICH
  AGE DES LETTRES
23 CROUT EQU $FD8E ;RETURN
24 STOCK EQU $9500 ;CHAINE
  A RECHERCHER STOCKEE EN $9500
25 ;
26 ;
27 ;
28 ;
29 ;PROGRAMME DE RECHERCHE
```

```

30 ;POUR UN TABLEAU DE CHAINE A$
31 ;AVEC SEULEMENT 1 DIMENSION
32 ;
33     LDA #$41
34     STA NOM1
35     LDA #$80
36     STA NOM2
37 ;
38 ;ADRESSE DU DEBUT DES TABLEUX
39 ;
40     LDA ARYTAB
41     STA ADTABB
42     LDA ARYTAB+1
43     STA ADTABH
44 ;
45 ;VERIFICATION DU NOM
46 ;
47 NOM     LDX #0
48         LDA (ADTABB,X)
49         CMP NOM1
50         BNE LGTAB
51         LDY #$1
52         LDA (ADTABB),Y
53         CMP NOM2
54         BEQ NBELEM
55 ;
56 ;LONGUEUR DU TABLEAU
57 ;
58 LGTAB   INY
59         LDA (ADTABB),Y
60         STA LTB
61         INY
62         LDA (ADTABB),Y
63         STA LTH
64 ;
65 ;CALCULE ADRESSE NOUVEAU TABLEAU
66 ;
67         CLC
68         LDA ADTABB
69         ADC LTB
70         STA ADTABB
71         LDA ADTABH
72         ADC LTH
73         STA ADTABH
74         JMP NOM
75 ;
76 ;NOMBRE D'ELEMENTS DANS TABLEAU
77 ;
78 NBELEM  LDY #$5
79         LDA (ADTABB),Y
80         STA NELEMH
81         INY
82         LDA (ADTABB),Y
83         STA NELEMB
84 ;
85 ;ENLEVE 1 AU COMPTEUR CAR VA
86 ;JUSQU'AU NUMERO 0...
87 ;
88         CLC
89         DEC NELEMB
90         BCC CONT
91         CLC
92         DEC NELEMH
93 ;
94 CONT    EQU *
95 ;
96 ;DEPLACE L'ADRESSE TABLEAU
97 ;AU DEBUT DES ELEMENTS
98 ;
99         CLC
100        LDA ADTABB
101        ADC #$7
102        STA ADTABB

```

```

103 ;
104 ;LONGUEUR DE L'ELEMENT
105 ;
106 LONGE   LDY #$0
107 LONGEL  LDA (ADTABB),Y
108         STA LGELEM
109 ;
110 ;ADRESSE DEBUT DE L'ELEMENT
111 ;
112         INY
113         LDA (ADTABB),Y
114         STA ADELEB
115         INY
116         LDA (ADTABB),Y
117         STA ADELEH
118 ;
119         INY
120         LDA (ADTABB),Y
121         STA LGELM1
122 ;
123         INY
124         LDA (ADTABB),Y
125         STA ADELBI
126 ;
127         INY
128         LDA (ADTABB),Y
129         STA ADELHI
130 ;
131 ;DEPLACE ADRESSE DEBUT ELEMENT
132 ;
133         CLC
134         LDA ADTABB
135         ADC #$3
136         STA ADTABB
137         BCC CONT1
138         INC ADTABH
139 ;
140 ;LECTURE DES ELEMENTS
141 ;
142 CONT1   EQU *
143 ;
144 ;VERIFIE SI L'ELEMENT EXISTE
145 ;
146         LDA LGELEM
147         BEQ ELEM
148 ;
149 ;RECHERCHE LA CHAINE STOCKEE EN
150 ;$300 DANS LA CHAINE EN QUESTION
151 ;
152         LDY #$000 ;COMPTE
153         UR DANS LE MOT CIBLE
154         LDX #$000 ;COMPTE
155         UR DANS LE MOT RECHERCHE
156         LDA STOCK,X
157         CMP /= ;CARACT
158         BNE PLULOIN
159         INX
160         CPX LONG
161         BCS EDIT1
162         INY
163         CPY LGELEM
164         BCS ELEM
165         JMP COMP
166 ;
167 ;
168 PLULOIN INY
169         CPY LGELEM
170         BCS ELEM
171         LDX #$00
172         JMP COMP

```

```

173 ;
174 ;STOCKE LE COMPTEUR DE LETTRES
175 ;
176 EDIT1 LDY #800
177 EDIT STY CLET
178 ;
179 ;LIT LA LETTRE
180 ;
181 LDA (ADELEB),Y
182 ;
183 ;AJOUTE #80 AU CODE ASCII
184 ;POUR EVITER LE FLASH
185 ;
186 CLC
187 ADC #80
188 ;
189 ;ENVOIE AU SOUS-PROGRAMME AFFICHAGE
190 ;
191 JSR COUT
192 ;
193 ;INCREMENTE LE COMPTEUR DE LETTRE
194 ;

```

```

195 LDY CLET
196 INY
197 ;
198 ;VERIFIE SI DERNIERE LETTRE
199 ;
200 CPY LGELEM
201 BEQ ELEM1
202 JMP EDIT
203 ;
204 ;PASSE A L'ELEMENT SUIVANT
205 ;
206 ELEM1 JSR CROUT
207 ELEM LDA NELEMB
208 BNE LGEL
209 LDA NELEMH
210 BEQ FIN
211 DEC NELEMH
212 LGEL DEC NELEMB
213 JMP LONGE
214 FIN RTS
215 END

```

## RECHERCHE OBJ

\*300.300

```

0300- A9 41 85 06 A9 80 85 07
0303- A5 68 85 08 A5 5C 85 09
0310- A2 00 A1 08 C5 06 D0 08
0318- A0 01 B1 08 C5 07 F0 1A
0320- C9 B1 08 85 12 C9 B1 09

```

```

0328- 85 19 13 A5 08 65 18 85
0330- 08 A5 09 65 19 85 09 4C
0338- 10 03 A0 05 B1 08 85 EB
0340- C9 B1 08 95 EC 18 C6 EC
0348- 90 03 18 C6 EB 18 A5 08
0350- 69 07 85 08 A0 00 B1 08
0358- 85 10 C9 B1 08 85 1E C8
0360- B1 08 85 1F C9 B1 08 85
0368- ED C9 B1 08 85 EE C9 B1
0370- 08 85 EF 18 A5 08 69 03
0378- 85 08 90 02 E3 09 A5 1D

```

```

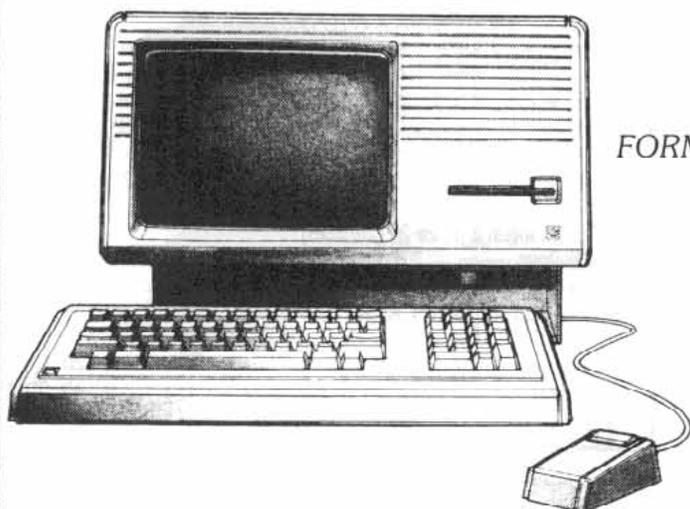
0380- F0 3F A0 00 A2 00 80 00
0388- 95 C9 3D F0 04 D1 1E D0
0390- 0D E2 E4 0A 80 12 C8 C4
0398- 1D 80 26 4C 86 03 C8 C4
03A0- 1D 80 1E A2 00 4C 86 03
03A8- A0 00 84 CE B1 1E 18 69
03B0- 80 20 ED FD A4 CE C8 C4
03B8- 1D F0 03 4C AA 03 20 3E
03C0- FD A5 EC D0 06 A5 EB F0
03C8- 07 C6 EB C6 EC 4C 54 03
03D0- 60

```



**m.b.d.c. informatique**

**TEL. 20.74.84.00 CALVADOS 1061**



• **32, RUE LEPELLETIER. LILLE**

PARKING «LA TREILLE»

ETUDES, CONSEILS, MISES EN PLACE  
FORMATION SUR MATÉRIELS ET LOGICIELS

S.A.V. ET LOCATION

CONCESSIONNAIRE



• **21, RUE BASSE. LILLE**

PARKING «LA TREILLE»

TOUTES LES FOURNITURES ET  
ACCESSOIRES POUR VOS MATÉRIELS

OUVERT DU MARDI AU SAMEDI DE 9H à 12H et de 14H à 19H

# Micro-Informations

Jean-Michel Gourévitch

L'Apple //e devient un Apple //c. Comme Pom's l'avait annoncé à ses lecteurs dans son numéro 16, le 65C02 devient dorénavant le standard de la famille Apple //. A partir du mois de mai, les nouveaux Apple //e vendus, intégreront le processeur 65C02, qui équipait déjà le //c, et des ROMs supplémentaires (notamment un nouveau jeu de caractères comprenant les caractères souris et un moniteur assurant la gestion des fenêtres et du curseur). Un kit vendu moins de 1000 francs permettra aux possesseurs d'anciens //e de les mettre à niveau. A première vue, il s'agit donc bien d'un abandon de ce qui était jusqu'alors le standard de l'Apple //e. Devant le dilemme du progrès, la firme de Cupertino a dû abandonner un peu de terrain au chapitre de la compatibilité. Ainsi, les programmes qui tournaient difficilement (ou pas du tout) sur le //c ne tourneront, hélas, pas mieux sur le nouveau //e; aucun "switch"

ne permet de sélectionner un mode ou l'autre.

Toutefois, on précise chez Apple que les indéracinables du //e pourront laisser en place l'ancienne ROM de caractères, tout en changeant les autres; rien n'empêche de programmer une ROM contenant les deux jeux de caractères qui seraient alors modifiables par logiciel.

La seconde nouveauté concernant l'Apple //e était annoncée dans le numéro 17 de Pom's. Il s'agit du changement du clavier. Le nouveau //e bénéficie donc d'un clavier comparable à celui du //c. Ainsi, lorsqu'on enclenche la touche "Majuscules", on peut, comme sur une machine à écrire standard, taper directement les chiffres. Techniquement, le shift lock est remplacé par un caps lock.

D'autre part, le nouveau clavier du //e n'est plus QWERTY. Sous le clavier de l'ancien //e, un interrupteur permettait d'obtenir à gauche, un

clavier français aux touches AZERTY, et à droite, un clavier QWERTY. Cet interrupteur existe toujours sur le nouveau //e. Mais lorsqu'on le bascule vers la droite, on obtient seulement les caractères du genre "[ " ou "]" , impossibles à obtenir en français. Les amateurs de jeux américains dont les commandes sont disposées au clavier, selon un ordre pratique, apprécieront.

*Un exemple :*

Sur le simulateur Flight Simulator II, la touche "<" permet de régler la valeur d'un instrument à la baisse, et ">" d'augmenter la valeur. Sur le clavier américain, ces touches sont voisines. Sur un clavier AZERTY, il faut à chaque fois actionner la touche majuscule pour passer du "<" au ">".

Pourquoi donc ces modifications ? Parce qu'Apple souhaite la standardisation, et la firme de Cupertino voudrait encourager les auteurs de

## Computer 3

3, rue Papillon 75009 Paris - Tél. 523.51.15



## BON de COMMANDE

Alimentation .....	600	Port	30	<input type="checkbox"/>
Boîtier et clavier .....	755		35	<input type="checkbox"/>
Clavier détachable (type MAK 2) .....	1 200		30	<input type="checkbox"/>
Clavier détachable (STAFF) .....	1 100		30	<input type="checkbox"/>
Moniteur 12" vert .....	950		100	<input type="checkbox"/>
Moniteur couleur .....	2 900		100	<input type="checkbox"/>
Base orientable pour moniteur .....	199		30	<input type="checkbox"/>
Lecteur disquettes 5' 1/4 distar .....	1 450		45	<input type="checkbox"/>
Contrôleur de drive .....	370		18	<input type="checkbox"/>
Carte langage 16K RAM .....	400		18	<input type="checkbox"/>
Carte 128 K RAM .....	1 550		20	<input type="checkbox"/>
Carte Z 80 CP/M .....	320		18	<input type="checkbox"/>
Carte 80 colonnes (Inverses et majuscules) .....	640		18	<input type="checkbox"/>
Carte couleur RVB (2 +) .....	950		18	<input type="checkbox"/>
Carte couleur RVB (2e) .....	1 380		18	<input type="checkbox"/>
Carte 80 col + 64 K (2c) .....	600		18	<input type="checkbox"/>

Carte imprimante .....	380	Port	20	<input type="checkbox"/>
Carte Grappler .....	475		20	<input type="checkbox"/>
Carte Grappler + Buffer 16K .....	1 350		20	<input type="checkbox"/>
Interface série .....	495		20	<input type="checkbox"/>
Interface supersérie .....	950		25	<input type="checkbox"/>
Carte Modem intégré .....	1 085		30	<input type="checkbox"/>
Wildcard .....	400		20	<input type="checkbox"/>
Speechcard .....	320		20	<input type="checkbox"/>
Carte Horloge .....	500		20	<input type="checkbox"/>
Joystick Métal 2 +, 2e-2c (préciser) .....	165		23	<input type="checkbox"/>
Ventilateur .....	280		23	<input type="checkbox"/>
Pince à disquette .....	50		18	<input type="checkbox"/>
Ruban impr. Dot Matrix Apple, NEC 8 .....	49		18	<input type="checkbox"/>
Disquettes SF/DD la boîte de (10) .....	89		20	<input type="checkbox"/>
Disquettes SF/DD 10 boîtes de (10) la boîte .....	85		40	<input type="checkbox"/>
Boîte de rangement pour 100 disquettes .....	199		20	<input type="checkbox"/>

NOM: .....  
 PRÉNOM: .....  
 ADRESSE: .....  
 .....  
 CODE POSTAL: ..... TÉLÉPHONE: .....

RÈGLEMENT PAR: CCP   
 CHÈQUE BANCAIRE   
 CARTE BLEUE   
 N° .....  
 DATE D'EXPIRATION .....

logiciels à recourir à l'arsenal des menus déroulants, fenêtres, et icônes en vigueur sur le Macintosh. Les nouvelles ROMs et le 65C02 permettront aux utilisateurs de booter depuis un disque dur, d'écrire des programmes Basic en minuscules et majuscules, d'utiliser un code plus serré, tout en utilisant les multiples fenêtres, et un meilleur contrôle du curseur. Reste à voir si cela sera suffisant pour dissuader les développeurs d'abandonner le //e pour le Mac. En attendant, les nouveaux programmes ne sont pas légions.

### Une vieille connaissance

C'est l'AppleWriter // qui nous revient de loin. Fonctionnant sous ProDOS, il peut désormais bénéficier du disque dur. Il permet d'écrire un texte dans n'importe quel ordre, de le mémoriser par paragraphe et de l'imprimer dans l'ordre choisi. Grâce à un module de communications, on peut recevoir des données d'une messagerie, ou d'un autre Apple, et les intégrer dans un texte. Il est distribué par Apple pour moins de 1800 francs.

Toujours distribué par Apple, le ruban à trois couleurs superposables offre à l'imprimante Scribe, une impression en couleurs avec notamment les logiciels Dazzle Draw, Koala Pad, ... Un ruban devrait suffire pour imprimer une dizaine de pages en mode graphique. Son prix est de 72 francs.

### Devine quoi dîner

*Random Dining* de Solitaire génère une série de menus (3 repas par jour pendant un maximum de 31 jours), et fournit la liste des ingrédients. Un seul petit problème : c'est au goût américain !

Un autre programme : *The Ultimate Shopping Machine* est fourni en prime et permet d'obtenir une liste imprimée des courses à effectuer au supermarché. Vous pouvez vous procurer l'ensemble pour 50 dollars.

### Un copieur conforme

C'est la version 5.0 de *Copy II plus* de Central Point. Un bitcopieur automatique capable de surmonter quelques systèmes de protection. Les paramètres sont stockés sur le disque, il suffit de taper le nom du programme pour lui adapter les paramètres. Combien de temps faudra-t-il pour que les logiciels lui résistent et que Central Point sorte une nouvelle révision de *Copy II plus* ? Cette nouvelle version vaut 40 dollars.

### CP/M pour le //c

C'est aujourd'hui possible grâce au *Z80c* d'Applied Engineering. Un module qui s'installe à l'intérieur d'un

ordinateur qu'on croyait pourtant "fermé". Il peut profiter des 128 Ko de l'ordinateur, ou n'en utiliser que 64, en réservant le reste comme RAM disk. Il est livré avec la version 4.0 de CP/M, le Z80c, coûte, outre Atlantique 160 dollars.

### De la mémoire

Combien vous en faut-il ? Un mégaoctet vous suffirait-il ? Alors, la carte *RamWorks* d'Applied Engineering est pour vous. Celle-ci sera notamment reconnue par Appleworks, Advanced Visicalc, FlashCalc, et bien d'autres. Elle accélère la vitesse d'exécution des programmes. Pour 1200 dollars, on en attendait pas moins !

### Un jeu de cartes.

Trois cartes en une, qui dit mieux ? C'est la *Business Card* de Street Electronics. Elle regroupe deux interfaces séries (pour disposer à la fois de l'imprimante et du modem), une carte horloge, et des possibilités de copie automatique d'écran graphique. Elle est compatible avec AppleWorks, Apple Access//, et surtout avec MousePaint (les spécialistes apprécieront : c'est le programme le plus difficile à interfacer avec une imprimante), la *Business Card* coûte moins de 200 dollars.

### Un vrai clavier.

Pas besoin d'acheter un nouvel Apple //e pour avoir le clavier vendu par Pentasonic. 78 touches dont un vrai pavé numérique. Les majuscules et minuscules comme sur une machine à écrire (la touche CAPS LOCK dispose du voyant LED qui fait tellement défaut sur l'IBM), les fonctions Pomme Ouverte et Fermée, et l'auto-test propres aux Apples : quel luxe ! Et en plus, il est inclinable pour un prix de 1273 francs.

### Et le Mac ?

Compatible avec l'Apple //, compatible avec l'IBM, accédant, grâce au menu pomme, au multi-tasking c'est vraiment la fête de Macintosh.

### Mac = Apple //

Pendant que certains développeurs essaient de transformer l'Apple II en Macintosh, David Hemmo a réussi à transformer le Mac en un Apple. Un Apple II+ pour le moment. Supportant donc seulement 48 Ko de mémoire et des programmes en DOS 3.3. Grâce à David Hemmo, le processeur 68000 du Mac, émule le 6502 de son cadet. Une prochaine version permettra de bénéficier du ProDOS et des caractéristiques (et surtout des programmes) du //e. Im-

aginez qu'on peut récupérer les disquettes du // sur les disquettes 3,5 pouces du Mac. C'est assez simple : on charge le logiciel dans l'Apple II, on relie le Macintosh à l'Apple II par la sortie RS 232 (série) reliée à l'entrée Modem du Mac, puis on transfère les programmes. Il ne reste plus qu'à les enregistrer sur une disquette de 3,5 pouces. Ensuite, on pourra "jouer" ces logiciels sur le Macintosh. Il paraît que Steve Jobs avait juré que la chose était impossible...

### Mac = IBM

Cette fois, c'est en IBM que le Mac se transforme grâce à *Mac Charlie* de Dayna Communications. "Étonnant ce qui entre dans Macintosh ces temps-ci. Les logiciels IBM PC, par exemple" proclame, non sans humour, la publicité de ce produit.

En bref, il s'agit d'un complément "hard" au Macintosh. Une base et surtout un prolongement de la partie droite de l'Apple, parfaitement dans le style et la couleur du Mac. La partie droite qui élargit le Macintosh d'un bon tiers contient les processeurs, 256 Ko de RAM extensibles à 640, et un drive pour disquettes de 5 pouces 1/4 (on peut rajouter un second lecteur). Oui, diront les spécialistes, tout ça est bien, mais l'IBM PC dispose d'un clavier bien spécifique, qui n'a rien à voir avec celui du Macintosh. Vous n'avez encore rien entendu, car *Mac Charlie* contient aussi un clavier, dans lequel on emboîte celui du Mac. On dispose alors à gauche des touches de fonctions, et à droite d'un pavé numérique (également utilisable en mode Macintosh). C'est bien le moins, car *Mac Charlie* reviendrait à quelques 1895 dollars. Soit le prix aux Etats Unis d'un compatible bien équipé. Dommage qu'on n'ait pas encore pensé au gadget transformant un IBM PC en Macintosh. Chez Apple, qui a dû procéder en février à un chômage technique pour cause de stocks trop importants, on apprécierait...

### Des programmes comme s'il en pleuvait

Il est bien fini le temps où les logiciels pour Macintosh étaient aussi rares qu'un rayon de soleil, à Paris.

A remarquer d'abord, *le Switcher*. Ce programme est développé par Apple et permet de faire tourner simultanément jusqu'à quatre applications sur un seul Macintosh 512 Ko. Comment ? Le logiciel partitionne la mémoire en quatre, on peut définir la mémoire nécessaire à chaque application. Ainsi, en se limitant à trois, il est possible d'avoir une application de 256 Ko, si elle requiert beaucoup de RAM. On choisit, au début de la séance, la mémoire utile à chaque

application puis on charge l'application en mémoire.

Une double flèche s'installe alors dans le coin supérieur droit de l'écran. Un clic sur la flèche et on bascule sur l'application suivante. La version finale permettra de transférer la presse papier d'une application à l'autre. Le prix de ce programme ? Il devrait être gratuit, distribué avec les Macintosh comme le Finder, et offert aux possesseurs actuels. Ce programme exige 512 Ko.

Aux Etats Unis, Centa System propose la transformation d'un Mac 128 Ko en 512 Ko pour 350 dollars. C'est environ trois fois moins cher qu'en France pour la greffe officielle de mémoire. Un scandale non ?

Mass Tech propose la même transformation pour 379 dollars. Mais, offre surtout une greffe portant la mémoire à 1 mégaoctet pour 995 dollars. Un méga, de quoi rêver non ! Autant que le Lisa (pardon Macintosh XL) mais avec la compatibilité en plus (actuellement, Macintosh XL n'est pas parfaitement compatible avec ses petits frères, et déforme notamment l'image de l'écran, en refusant d'absorber certains programmes en émulation avec MacWorks). Le Mac à un méga est déjà là, et ce n'est qu'un début...

### Deux traitements de textes

Jusqu'à maintenant, les utilisateurs de Macintosh devaient se contenter de MacWrite. Un logiciel facile, permettant une vision exacte à l'écran de ce qui sera imprimé, mais limité, notamment par ses possibilités d'impression. Voici donc deux bonnes nouvelles.

D'abord, la sortie aux Etats Unis (et bientôt chez nous) de la version 4.3 de MacWrite. Elle permet d'imprimer sur un Mac 512 Ko quelques 250 pages (à condition que ce document n'incorpore pas plus de 2047 passages à la ligne par la touche Return, et qu'il y ait suffisamment de place sur la disquette pour l'accueillir). Le texte ne réside plus en mémoire, mais est chargé depuis la disquette au fur et à mesure des besoins. En plus des perfectionnements annoncés dans le numéro 17 de Pom's, le nouveau Macwrite offre la possibilité d'imprimer un document en le réduisant de 50%; et un nouvel accessoire de bureau permet de choisir son imprimante, adapté au réseau local Appletalk et à l'imprimante Laserwriter.

L'autre nouveauté, c'est la sortie officielle de la version finale de Word de Microsoft. Un traitement de textes qui répondra aux besoins de tous les professionnels; on peut :

- disposer simultanément d'un maximum de quatre fenêtres,

contenant chacune un document  
• coller sans problème des morceaux de textes entre documents. Il existe un glossaire, et la taille des documents est illimitée

On peut aussi réaliser des notes de bas de pages, automatiquement numérotées, ou écrire en colonnes. Toutes les commandes (même la taille des caractères) sont doublées par des séquences de touches au clavier. Les documents MacWrite sont utilisables dans Word. Ajoutons que le programme dispose en plus de la possibilité d'opérer fusions de lettres et mailings. Très très impressionnant. Un must pour les utilisateurs sérieux.

### Journaux électroniques

Il n'a pas fallu longtemps pour constater que le Macintosh, particulièrement dans son association avec l'imprimante Laserwriter, permettait de réaliser des journaux et lettres périodiques sans problème. Dans le précédent numéro nous vous présentions MacPublisher. Voici encore *Ready Set Go* de Manhattan Graphics, basé sur une habile utilisation de MacDraw. Très facile à utiliser pour 125 dollars.

### Un Macintosh en couleurs ?

Sur papier, seulement. Grâce au logiciel *Color Print* importé par BIP. Un jeu de trois rubans (rouge jaune et bleu) se substituent au ruban original de l'imprimante Imagewriter. La couleur à partir du Macintosh pour 710 francs.

### Des pommes au menu

A la différence de Lisa, en théorie, le Macintosh ne permet pas d'effectuer simultanément plusieurs tâches. Voici ce handicap partiellement surmonté. Grâce à des accessoires qui s'installent dans le menu Pomme. Vous avez déjà, en standard, sur tous les logiciels l'heure, une calculatrice bien rudimentaire, l'album,... Voici que de nouveaux accessoires peuvent s'y installer facilement. Comment ? Grâce au *Desk Accessory Mover (DAM)* qui permet de rajouter, d'enlever, ou de débaptiser des accessoires aussi facilement que de changer un jeu de caractères avec le Font Mover. Ce DAM conçu par CE Software est vendu d'une façon inhabituelle sous le système du Mac Honor. Les utilisateurs sont invités à en faire des copies pour leurs amis. Et ceux ci, s'ils souhaitent le conserver sont invités, à leur tour, à envoyer au créateur la somme de 25 dollars, ce qui donne droit aux perfectionnements et aux documentations.

Que peut-on installer dans le menu Pomme ? CE propose trois outils bien utiles : le *Mockwrite*, un mini

traitement de textes dont on peut disposer à tout moment (même sur le bureau) permet d'écrire un texte, de le sauver, de le retrouver. En outre, il possède même une fonction de recherche.

Le *MockTerminal*, est un programme de communication. C'est peut-être le plus utile. Vous remplissez une fiche de CX MacBase, lorsque vous avez soudain besoin d'une information disponible sur votre banque de données. Sans quitter le programme, vous cliquez sur *MockTerminal* dans le menu Pomme, vous appelez par le Modem la banque de données et vous recevez l'information. Un coup de copier-coller, et elle est incorporée à votre fiche.

*Mock Printer* permet d'imprimer un document à n'importe quel moment, même sans disposer d'un traitement de textes.

Un agenda permet de noter des rendez-vous. Une calculatrice scientifique ou financière peut remplacer avantageusement la calculatrice standard. Pour les programmeurs, il existe une calculatrice hexadécimale. Un utilitaire permet de connaître en permanence la place encore disponible sur un disque, ou dans la mémoire centrale de Macintosh. Un "Executive Decision Maker" remplace la pile ou face, pour vous donner un avis (hasardeux) sur une décision à prendre. Une loupe permet d'agrandir un morceau de l'écran. Un accessoire enfin permet de provoquer l'éjection du (ou des) disque(s), et de provoquer un Reset par logiciel.

Tous ces accessoires ont l'avantage de pouvoir être "invoqués" tout en utilisant un programme, sans interrompre son exécution. Et ce n'est qu'un début.

T/Maker Graphics propose *ClickOn*. Rien de moins qu'un mini-tableau et ses graphiques, cachés dans le menu Pomme. L'avantage est de pouvoir l'utiliser tout en écrivant un document avec MacWrite, et coller dans le texte le tableau et le graphique obtenu. Il en coûtera 80 dollars.

Controle X pourrait commercialiser en France *Art Grabber*. Un accessoire (toujours utilisé dans le menu Pomme) qui permet simplement d'ouvrir (et de retoucher) un document MacPaint, sans avoir à ouvrir MacPaint quel que soit le programme où on se trouve, et surtout, d'utiliser dans tous ces environnements les dessins tout faits de Click Art.

Un seul pépin dans cette pomme : une fois "gonflée" de tous ces accessoires, le système de la disquette occupe une place considérable (fréquemment 300 Ko). La solution

consiste donc à travailler avec une disquette ne contenant que le système, et une autre contenant fichiers et applications. Ce n'est pas facile quand l'application est protégée. Quoi qu'il en soit, il vaut mieux disposer de 512 Ko et de deux lecteurs de disquettes.

Les utilisateurs du Macintosh savent déjà que deux disquettes, c'est pour cet ordinateur un vrai minimum.

## Adresses

**Solitaire** 746 West 113th Street  
Bloomington - Minnesota 55438

**Applied Engineering** PO Box 798  
- Carrollton - Texas 75006

**Central Point Software** 9700 S.W.  
Capitol Highway #100 Portland -  
OR 97219

**Street Electronics Corp.** 1140  
Mark Av. Carpinteria CA.

**Pentasonic** 34 rue de Turin -  
75008 Paris - Tél : 293 41 33

**Dayna Communications** 50  
S.Main - Salt Lake City Utah 84144

**Centa Systems** 5308 Derry Av.

Suite C Agoura Hills - CA 91301

**Mass Tech** 451 Boston Road - Gro-  
ton - MA 01450

**JRD Orléans** 11 place Ste Croix -  
45000 Orléans - Tél : (38) 54 83 30

**CE Software** 801 73rd Street - Des  
Moines - Iowa 50312

**T/Maker Graphics** 2115 Landings  
Drive Mountain View - CA 94043

**BIP** 13 rue le Duc - 75018 Paris -  
Tél : 255 44 63

**Controle X** Tour Maine Montpar-  
nasse - 33 avenue du Maine - 75755  
Paris Cedex 15

## Pour l'Apple //

**Techniques de programmation sur l'Apple II** de René Belle, Editions du PSI - 164 pages - 95 FF

Ce livre démontre, à l'aide d'un exemple complet, l'illustration de la programmation "structurée" en Basic Applesoft. Le découpage en chapitres est assez clair : généralités, saisie des données, édition, tri et recherche. Un peu léger peut-être sur le contrôle de sécurité : une routine d'INPUT généralisé travaillant en GET aurait été intéressante. Dans l'ensemble, bon pour le fond et mauvais pour la forme : ce n'est même pas du MacWrite, mais du traitement de textes sur Apple II avec imprimante matricielle...

**Apple Logo activities** par Stephen Dewitt, Reston - Prentice Hall - 186 pages - \$22.05

**The Apple Logo workbook** de M.J. Winter, Reston - Prentice Hall - 230 pages - \$16.85

Deux livres pratiques sur le Logo auxquels le seul reproche que l'on puisse faire est qu'ils sont rédigés en anglais et font référence au Logo anglais. Ceci dit, une simple table de correspondance permettra une transposition rapide des exemples.

Le premier s'articule principalement autour de questions posées à l'utilisateur en indiquant des références à des explications se situant dans d'autres pages. Le second, présenté en 46 leçons bien programmées, alterne les explications et les exercices. Les deux livres sont intéressants, le second étant un excellent guide pour un professeur souhaitant organiser des cours de Logo.

## Pour le Macintosh

**Connaissez-vous Macintosh** de Pierre Courbier, ETSF - 143 pages - 80 FF

Le énième livre en français sur l'utilisation du Macintosh, et qui ne ferait pas vingt pages s'il était composé et

si l'on enlevait les copies d'écran... Encore une fois, aucune astuce d'utilisation. Mieux vaut assister à une bonne démo de 30 minutes sur le Macintosh.

**L'outil Macintosh** de Gérard Lévy, Eyrolles - 288 pages - 260 FF

Il y a déjà là plus de contenu que dans les livres semblables au précédent. Mais, quand on "présente" plus de 30 logiciels, sans parler du Macintosh lui-même, en si peu de place, on est forcément superficiel. En outre, la présentation ballote complètement le lecteur : on passe d'un logiciel à l'autre sans indication claire de changement; il se dégage hélas une impression de fouillis.

**Macintosh, outils, progiciels, applications** de Xavier Gaucherand, Edimicro - 237 pages - 148 F.

Un des meilleurs ouvrages dans le genre frelaté du livre d'initiation à Macintosh et à ses logiciels. Au moins, celui-là est composé... car même si MacWrite est un programme sympathique et l'ImageWriter une imprimante de qualité, un lecteur objectif peut difficilement trouver plaisant à lire un ouvrage rédigé en MacWrite. Signalons un effort de classement et un souci pédagogique rare dans la logorrhée ambiante du Macintosh.

**Découverte et pratique de CX MacBase** par Hervé Thiriez, Editions Controle X - 170 pages - 125 FF. Disquette d'accompagnement : 175 FF.

A la fois un excellent guide pédagogique sur l'utilisation de CX MacBase et un recueil d'applications avec de nombreux tableaux expliqués dans le livre et prêts à l'emploi sur la dis-

quette d'accompagnement. Les divers exemples illustrent les nombreuses possibilités du programme. Le livre fait référence aux versions M10 et M20 du programme.

La disquette comporte une version bridée du programme, interdisant l'impression ou la création de nouvelles fiches, mais permettant de mettre en oeuvre tout le reste du programme : un moyen économique de tester CX MacBase avant l'achat. A noter tout particulièrement : UniFiche, un modèle menant à des utilisations multiples.

## La communication

**Le guide marabout de la télématique** de Michel Pouillet - 221 pages.

Un ouvrage clair, mais uniquement destiné à ceux qui souhaitent enrichir leur culture générale. Certains sujets, tels que la monnaie électronique ou la bureautique, n'intéressent directement ni l'Appleomane, ni le fana de la micro. De plus, on n'y trouve rien de spécifique à l'Apple.

**La télé-informatique personnelle** de Jacques Bessières, Hachette Informatique - 217 pages - 145 FF.

Voici un excellent livre sur la communication. En outre, comme l'indique le sous-titre, il s'intéresse aux "Applications aux ordinateurs Apple". Clair et plein d'informations, cet ouvrage couvre les aspects hard et soft ainsi que les services. Il présente de multiples recettes d'utilisation. Même le célèbre Roland Moreno nous dit que c'est, pour ses collaborateurs, un ouvrage de référence. C'est tout dire... Un must pour les branchés.

## Bibliographie

Alexandre Dubaïk



Numéro 14

Initiation à l'Assembleur (4)	B-A
Procédure &	A
Disque virtuel 64K	A
Lecture/Ecriture rapide de tableaux	B
INPUT généralisé miniaturation	B
Amper-Interpréteur ICARE	B-A
Les essais du Macintosh	/
De MacWrite à Apple Writer //e	B
Les tokens du Basic Microsoft	(B)
Gestion de masques améliorée	B-A
Lève toi et brille	/
217 fichiers par disquette	B
Conversion minuscules/MAJUSCULES	A
Un catalogue général	B-A
Koala Pad à l'essai	/
Appleworks à l'essai	/
Visicalc Advanced Version	/
Jane : l'Apple II sauce Macintosh	/

Numéro 15

Initiation à l'Assembleur (5)	A
Mobby Disk	A
Disk Check-Up	A
Fleuves de France	B
HPGRAPH	P
Ecriture en page haute résolution	B-A
Mouse Paint	/
Les caractères programmables sur imprimante Apple	/
Gestion de fichiers par RWTS et DOS 3.3	B-A
Les routines en ROM du Macintosh	(B)
Un éditeur de formes et curseurs	B
Omnis 2 à l'essai	/

Numéro 16

Initiation à l'Assembleur (6)	A
Pot-pourri ProDOS	/
Gérer la date ProDOS	B-A
Signature	B-A
Votre Epson en mode proportionnel	B-A
Accès direct aux disquettes	B
L'Apple en multitâches	B-A
Personnalisez vos disquettes Macintosh	B
Call : un exemple d'application MacPaint et le Basic	B-A
Appel des routines en ROM	A
Entrée analogique améliorée	A
Un désassembleur en Applesoft et WPL	A-WPL
Les codes ASCII du //e	B
STARTUP et saisie de la date en Pascal	P
PLE+CRAE+APA	B-A
Conversion minuscules/MAJUSCULES	B-A
CALLs à gogo	(B)

Numéro 17

Initiation à l'Assembleur (7)	A
Transfert rapide de tableaux	B-A
Les pointeurs en Pascal UCSD	P
Animations graphiques	B-A
Visualisation d'une distribution normale	B
Copie d'écran sur imprimante Epson	A
Catalogue sur imprimante BSAVF et BLOAD avec le Basic Microsoft	B-A
Switch vidéo pour carte 80 colonnes	/
Mini-éditeur Basic	B-A
Transformez votre //e en II+	A
Conversion chiffres-lettres	B

A(ssembleur) - B(asic) - P(ascal) - (B/A) relatif au Basic ou à l'Assembleur  
Les articles dont les titres sont en Italiques, concernent le Macintosh.

# Courrier des lecteurs

Olivier Herz

Je dispose d'un Apple //e muni d'un lecteur de disquettes, d'une carte Chat Mauve et d'une carte "souris". Le programme "Disque Virtuel 16K", publié dans Pom's 12, fonctionne très bien, mais rien ne va plus avec le "Disque Virtuel 64K", paru dans Pom's 14. L'auteur parle de carte 80 colonnes étendue, sans préciser laquelle. La carte Chat Mauve serait-elle incompatible ?

D'autre part, une petite suggestion pour les forts en assembleur; est-il possible de réaliser un patch du programme COPY du DOS 3.3 qui permette de faire une copie de disquette en une seule passe ou deux (avec un seul lecteur de disquettes et une carte 64K). Ce n'est pas évident, mais je ne m'adresse pas à n'importe qui, que diable !

Régis Ringenbach - 30 Villa Chaptal - 92300 Levallois

En effet, le "Disque Virtuel 64K" pose un problème, il ne protège que la première moitié de la page texte de la mémoire auxiliaire (pages 4-5, les pages 6-7 étant oubliées). Par conséquent, il faut apporter des modifications au programme pour son bon déroulement.

Patch de AUTO RWAUXINIT.OBJ :  
BLOAD AUTO RWAUXINIT.OBJ  
4F47: 0C  
4FCB: 6C  
5065: 10 07 06 00 01 04 05 06 07  
5094: 65  
50AD: 05  
50AF: 68  
BSAVE AUTO RWAUXINIT.OBJ

Ces modifications ont été précisées dans Pom's 16 (p. 66). Il est à noter que le "Disque Virtuel 64K" ne fait, en réalité, que 50.5K (4K pour la piste du catalogue et 46.5K pour les données).

En ce qui concerne la page graphique haute résolution de la mémoire auxiliaire, elle est correctement protégée par le disque virtuel (inhibition des pistes 6 et 7). En revanche, ce disque virtuel est incompatible avec le programme PURPLESOFT de la carte EVE car ils utilisent tous les deux l'équivalent carte langage de la mémoire auxiliaire (adresses \$D000 à \$FFFF). Il serait possible de patcher le disque virtuel de la même façon que ci-dessus pour inhiber les pistes correspondantes (\$11, 12 et 13) si la piste du catalogue (\$11) n'était précisément à cet endroit-là (\$D000 à \$E000 plus exactement), ce qui oblige à remanier le disque virtuel.

Je ne pense pas qu'il suffise d'un simple patch pour que COPY utilise les 64K de la mémoire auxiliaire et fasse une copie en 2 passes. C'est faisable, mais il s'agit d'un gros travail, voire même d'une refonte du programme. En revanche, il existe des programmes de copie qui en tiennent compte actuellement, comme Locksmith 5.0.

Le programme Gesmask Modifié, publié dans Pom's 14, m'a intéressé. Malheureusement, j'ai été obligé de changer quelque peu HCT.OBJ, pour obtenir une copie d'écran correcte. J'ai une OKI 82A avec "Universal parallel printer interface PRX12 (E.E.P.I) et j'avais les problèmes suivants :

1-je n'obtenais qu'une ligne d'écran sur quatre;

2-les deux dernières lignes écran qui, par un appel à \$FDÉD, génèrent un "plantage sans fin".

La solution que je propose, n'est pas très élégante (scrolling des deux dernières lignes) mais a l'avantage de fonctionner. Ci-dessous, le listing de la routine HCT.OBJ modifiée.

```

9465- A7 00 LDA $800
9467- 05 25 STA $25
9469- A0 00 LDY $800
946B- 20 C1 FB JSR $FBC1
946E- B1 2R LDA ($2B),Y
9470- C9 40 CMP $40
9472- B0 00 BCS $9481
9474- C9 20 CMP $20
9476- B0 06 BCS $947E
9478- 18 CLC
9479- 69 C0 ADC $C0
947B- 4C B1 94 JMP $9481
947E- 18 CLC
947F- 69 80 ADC $80
9481- 20 ED FD JSR $FDÉD
9484- C8 INY
9485- C0 2B CPY $2B
9487- D0 E5 BNE $948E
9489- A9 8D LDA $8D
948B- 20 ED FD JSR $FDÉD
948E- C6 25 DEC $25
9490- A5 25 LDA $25
9492- C9 16 CMP $16
9494- D0 D3 BNE $9469
9496- A0 55 LDY $55
9498- C4 CE CPY $CE
949A- F0 0F BEQ $94AB
949C- 20 70 FC JSR $FC70
949E- 20 70 FC JSR $FC70
94A2- A9 14 LDA $14
94A4- A0 55 LDY $55
94A6- 84 CE STY $CE
94A8- 4C 67 94 JMP $9467
94AB- A9 00 LDA $00
94AD- 85 CE STA $CE
94AF- 60 RTS
    
```

Olivier Fages - 61 rue Victor Hugo - 94700 Maison Alfort.

N'ayant pas ce matériel, il nous a été impossible de tester les modifications que vous apportez à la routine.

L'article STARTUP et saisie de la date en Pascal, publié dans Pom's

16, m'a inspiré et je vous propose d'obtenir le même résultat par une méthode beaucoup plus simple.

1-Créer un fichier EXEC appelé DATEUR :

Depuis le niveau global taper <M> pour Make Exec File, le système vous demande NEW EXEC NAME? Il faut répondre DATEUR <RTN>. Puis le système demande TERMINATOR = %, CHANGE IT; en répondant N, on retourne au niveau global. Tout ce qui est tapé sera enregistré dans le fichier EXEC. Ensuite, taper F (Filer) et D (Date) et deux fois % pour terminer. Le fichier EXEC est alors enregistré.

2-Créer un programme appelé STARTUP :

```
PROGRAM STARTUP;
USES CHAINSTUFF;
BEGIN
SET ('EXEC/DATEUR');
END.
```

Le fichier EXEC et le code du programme STARTUP doivent se trouver sur la disquette APPLE0 ou APPLE1 sous les noms respectifs de DATEUR et SYSTEM.STARTUP. A chaque boot, le programme SYSTEM.STARTUP sera exécuté et lancera les commandes contenues dans le fichier EXEC. Ainsi, il suffit de rentrer la date.

Jean Philip - Chemin St Roch - 84490 St Saturnin d'Apt

Cette méthode ne peut fonctionner qu'avec la version 1.1 du Pascal UCSD. En effet, les fichiers EXEC ne sont disponibles que sur cette version.

Il m'est impossible de faire fonctionner le programme "Gérer la date ProDOS", paru dans Pom's 16. Mon ordinateur me répond toujours NO BUFFERS AVAILABLE.

Monsieur Rambaud - 11 rue E. Roux - 15000 Aurillac

Il y a effectivement un problème avec le programme de DATE en ProDOS. En effet, si on fait, après avoir transféré DATE sur une disquette ProDOS, un CATALOG de cette disquette, on voit que le SUBTYPE du fichier DATE correspond à une adresse de chargement A\$0000. Or, ProDOS refuse de charger un fichier en page 0 et émet donc un NO BUFFERS AVAILABLE après l'essai d'un -DATE.

La solution est de remplacer dans le programme Applesoft (STARTUP par exemple) l'instruction PRINT CHR\$(4)"-DATE" par PRINT CHR\$(4)"BRUN DATE,A\$300"

Convertir des programmes DOS 3.3 en ProDOS avec les programmes ac-

compagnant le Pom's 16 pose quelques problèmes. En effet, le programme me répond que je ne suis pas sous ProDOS.

Monsieur Blin - 37 rue Gue d'Orient - 95470 St Witz

Afin de résoudre le problème, il faut modifier le début de la ligne 117 du programme Basic CQFD. Celle-ci devient :

```
117 IF PEEK (48961) <> 67 THEN...
```

C'est le c du copyright Apple dans la page globale ProDOS (\$BF00 - \$BFFF)

Je possède une imprimante Epson avec une interface Shinshu Seiki Co. Ltd. dont je suis tout à fait satisfait. Malheureusement, cette imprimante se refuse à imprimer correctement les menus des disquettes Pom's. C'est un problème soulevé par plusieurs lecteurs et auquel vous avez donné une réponse dans le numéro 15 en conseillant d'envoyer vers l'imprimante le code permettant de supprimer l'affichage sur l'écran. Malheureusement, je n'ai pas trouvé dans la notice de l'interface le code en question.

Aussi me suis-je attaqué au problème à sa source. Le saut des lignes vient du fait que lorsque le 40ème caractère est affiché, il se produit un retour ligne automatique sur l'écran. Il est suivi d'un deuxième retour ligne envoyé par le programme. Ce dernier passant ensuite à la ligne suivante se retrouve en fait sur la quatrième ! Pour passer à la ligne suivante, il faudrait donc que le programme revienne d'une ligne en arrière, dans le cas où l'interface affiche sur l'écran. Comme il n'est pas question d'avoir deux versions du programme, j'ai modifié le programme HCT.OBJ pour que le numéro de la ligne suivante ne soit plus calculé à partir du numéro de ligne fourni par le moniteur mais d'après un numéro tenu dans le programme. Et tout rentre dans l'ordre. J'ai également modifié l'instruction 70 du programme Basic MENU pour la débarrasser de tout ce qui était devenu superflu.

Je n'ai pu faire le test qu'avec l'interface dont je dispose. Il faudrait vérifier que tout se passe bien avec un interface qui n'affiche pas en même temps sur l'écran.

### HCT.OBJ modifié

\*9465.9496

```
9465- 20 BE FD
9468- A7 00 85 25 85 96 20 C1
9470- FB A0 00 B1 28 C9 40 B0
```

```
9478- 0A C9 20 B0 04 49 C0 D0
9480- 02 49 B0 20 ED FD CB C0
9488- 28 D0 EB 20 BE FD E6 96
9490- A5 96 C9 18 D0 D4 60
```

\*  
Ligne 70 modifiée :

```
70 VTAB 1: PRINT D1$"PR#" S:
CALL 37989 : PRINT D1$"PR#0"
```

C. Leyo - 32 allée de la meute - 78110 Le Vésinet

Suite à ma lettre, publiée dans Pom's 12, concernant les incompatibilités que je constatais, sous Pascal ucSD, entre la carte interface Epson et la carte Eve Chat Mauve, j'ai le plaisir de vous annoncer que j'ai enfin trouvé la solution de ce problème.

Il s'agissait d'interférences entre les deux cartes à travers les circuits de la carte mère. On élimine ces interférences en soudant un condensateur céramique de 220 picofarads 50V non polarisé (0.9 Frs) entre les broches 3 et 7 de la puce 6A sur la carte interface Epson; on dispose alors à nouveau de l'intégralité des fonctions des deux cartes. Cette solution a été trouvée, dans le courrier des lecteurs (octobre 84) de Call Apple, par un lecteur assidu de Pom's (M. Koenig).

Jean-Pierre Januel - 5 rue Duperré - 75009 Paris

J'ai acquis, il y a quelques mois, la souris Apple Mouse II et son interface pour mon Apple II+. Programmant assez souvent en Pascal, j'ai donc voulu l'intégrer à mes programmes par le biais de menus à la Macintosh. Une première série de Units a vu le jour et m'a donné satisfaction, dans un premier temps. Dans un premier temps seulement, car la souris peut fonctionner en deux modes. En mode passif, le plus simple, qui oblige donc le programme utilisateur à l'interroger régulièrement pour détecter les éventuels mouvements et déplacer le curseur en conséquence. C'est sur ce principe que mes premières réalisations ont été faites. Cela fonctionne très bien, mais si l'intervalle qui sépare les interrogations est trop long (à cause des traitements qu'effectue le programme) le mouvement du curseur est saccadé.

La solution est d'utiliser le deuxième mode de fonctionnement, à savoir le mode mettant en jeu les interruptions. Pour ceux qui ne connaissent pas le problème, je précise rapidement que l'interface de la souris envoie alors au 6502 un signal d'interruption lorsque la position de celle-ci a changé, ou lorsque le bouton a été appuyé (ou les deux). Il suffit de mettre en place une routine de prise en charge de cette interruption, dont le rôle sera de lire les coordonnées

de la souris et de déplacer le curseur en conséquence. L'avantage immédiat est que, quelle que soit la longueur des traitements du programme utilisateur, le curseur suivra toujours régulièrement les interruptions dans ce système.

J'ai expérimenté ce mode de fonctionnement, avec succès, sous DOS 3.3 et j'ai décidé de l'appliquer en Pascal. C'est là que les choses se corsent, car je ne sais pas, de façon précise, comment on peut gérer les interruptions dans ce système. Je lance donc un appel auprès des lecteurs qui auraient des renseignements à me fournir, quitte ensuite à centraliser tous les éléments qui m'auront été communiqués pour en faire un article synthétique si le sujet intéresse d'autres lecteurs. Les données qui me manquent sont, entre autres :

1-l'adresse du vecteur d'interruption (il semblerait que ça se passe ailleurs qu'en \$FFFF - \$FFFF).

2-Les précautions à prendre (doit on sauvegarder des informations particulières autres que les registres généraux).

3-et tout ce à quoi je ne pense pas ici, mais qui a son importance dans la manipulation.

Merci d'avance à tous ceux qui se pencheront sur le problème. Pour les autres (qui manqueraient d'informations mais que le problème intéresse) j'ai l'intention, si le courage ne m'a pas quitté d'ici là, d'envoyer à notre revue préférée le résultat de mes cogitations et nuits blanches.

Eric Pascual - 29 rue Foucher Lepelletier - 92130 Issy Les Moulineaux.

Le programme de Startup avec saisie de date, publié dans Pom's 16, est fort intéressant. Toutefois, un problème se pose avec la version 1.2 du Pascal UCSD car l'adresse de la date n'est pas la même que dans les versions précédentes. J'ai trouvé qu'elle était égale à -21252, et indique aux lecteurs intéressés la démarche à suivre pour arriver à ce résultat.

D'après la déclaration du type DATE, une telle variable occupe deux cases mémoire ADDR et ADDR+1 remplies comme suit :

Année (7 bits), jour (5 bits), mois (4 bits).

Ainsi, pour la date du 1er janvier 1985, les 16 bits de ADDR et ADDR+1 seront :

0000001000010001 soit la valeur 2 dans ADDR+1 et la valeur 17 dans ADDR.

Il suffit d'enregistrer cette date grâce à la commande D(ate) du F(iler) puis de chercher deux adresses consécutives contenant les valeurs 17 et 2. Il est inutile d'explorer les 64K de RAM, la date se trouvant aux alen-

tours de -2000 à -2500, dans les autres versions.

Christian Bellot - 21 bd Dahdah - 13004 Marseille

Je possède un Apple //e et la carte Chat Mauve et désire utiliser le programme Visicalc dans les meilleures conditions. Or aujourd'hui, j'obtiens un écran où les colonnes paires (mémoire auxiliaire) restent noires. Que dois-je faire pour résoudre ce problème ?

Dominique Dubois - BP 410 - 210 Antsirana - Madagascar

La version de Visicalc que je possède date de plusieurs années et est donc incapable de reconnaître et d'utiliser la carte 80 colonnes de l'Apple //e. Si l'on boote sur Visicalc à l'allumage, il fonctionnera en 40 colonnes. Si l'on fait PR#3, puis PR#6 pour booter Visicalc, il y aura un affichage des 40 colonnes de Visicalc sur un écran de 80 colonnes, donc une colonne sur deux. Je ne sais pas si les versions récentes de Visicalc sont adaptées au //e.

En revanche, les nouveaux tableurs (Magicalc, Appleworks, Multiplan, Visicalc advanced version...) connaissent le //e et fonctionnent en 80 colonnes.

Je me pose de nombreuses questions et je vous serais reconnaissant de bien vouloir me répondre :

1-Est-il possible de rendre compatible Magic Window à la police de caractères offerte par l'Epson ? Celle-ci étant composée de "ESC" ou de "CHR\$(27) +..." alors que Magic Window permet seulement les "CTRL ...".

2-L'assembleur Big Mac est allergique au "OBJ"; pourriez-vous expliquer le rôle et l'utilité de "OBJ" (je commence à peine et c'est dur, dur...).

3-Y a-t-il dans Big Mac une commande remplaçant le "EPZ" du Lisa; à moins que Big Mac fasse la distinction suivant le "\$adresse" ?

4-Dans tous les listings que j'ai pu lire, il n'y a pas une seule routine de revectorisation de l'ampersand (&) qui soit semblable. N'existe-t-il pas une routine standard ?

Pascal Bordais - BP 45 - 94350 Villiers sur Marne

1 N'ayant pas d'Epson, il m'est difficile de vous répondre. Toutefois, je vous propose de faire un essai sachant que "ESC" n'est autre que "CTRL + [".

2-Sur Lisa 2.5, le code "OBJ" indique l'adresse de début du programme objet qu'il faut sauver après assemblage par un BSAVE, soit au clavier, soit dans une commande DCM dans le fichier source. Ce code

n'existe pas sur Big Mac, qui met d'autorité le programme objet en \$8000 et permet de le sauver par le biais de la commande "O" du menu. Cette commande inscrit sur la disquette la vraie valeur du début du programme (donnée par ORG), et non \$8000 comme le ferait un BSAVE.

3-Vous avez vous-même répondu à la question; Big Mac fait la distinction suivant l'adresse. En cela, il est plus intelligent que Lisa, celui-ci ayant besoin d'un EPZ pour la page zéro.

4-Il n'existe pas de routine standard de revectorisation de l'ampersand (&). Pour les commandes d'une seule lettre, je vous renvoie à "Amper Jump & TSort" paru dans Nibble Express (volume 2).

En mode texte 80 colonnes des phénomènes étranges se produisent avec la routine de "La programmation facilitée" (Pom's 5).

& LET donne une barre inverse hachurée,

& INPUT A\$,Lx ne positionne pas le curseur en début de ligne de pointillé mais sur la ligne au-dessous et à une distance du bord droit de la fenêtre égale à la longueur spécifiée pour le pointillé.

Grâce aux articles d'initiation à l'assembleur, j'ai pu remédier à ces problèmes.

Pour l'option LET: il faut tester le mode d'affichage; si l'on est en 80 colonnes alors on divise par deux les adresses \$6 et \$7 qui sont respectivement la marge et la longueur + marge. Ainsi, en 80 colonnes la routine est de nouveau exécutée après commutation des pages textes de la carte 80 colonnes.

Pour l'option INPUT: le saut à la sous routine du moniteur (JSR \$FC10) est modifié pour aller en \$9594 où l'on teste le mode d'affichage. Si le mode 80 colonnes est enclenché, le saut à \$FC10 est remplacé par un saut \$CBDB (routine du moniteur pour la carte 80 colonnes).

Marche à suivre :

CALL-151

BLOAD PROG48K

93C9: 20 94 95

9550: taper la récapitulation ci-après BSAVE PROG48K,A\$9300,L\$2A2

9550- AD 18 C0 0A 90 05 46 06

9558- 46 07 18 A4 06 C4 07 D0

9560- 06 AD 96 94 91 28 60 B1

9568- 28 09 80 25 32 91 28 AD

9570- 18 C0 0A 90 0E 8D 55 C0

9578- B1 28 09 80 25 32 91 28

9580- 8D 54 C0 C8 C4 07 90 DF

9588- AD 18 C0 0A 90 04 06 06

9590- 06 07 38 60 AD 18 C0 0A

9598- 90 04 20 DB CB 60 20 10

95A0- FC 60

Frédéric Ivšic - 203 bis ave Napoléon Bonaparte - 92500 Rueil Malmaison

# pom's

## DISQUETTES (sauf précision, toutes les disquettes fonctionnent sous DOS 3.3 sur Apple II +, //e ou //c)

HAIFA .....	(cf. Pom's n° 5)	à	55,00 F	.....
H-BASIC .....	(cf. Pom's n° 8)	à	150,00 F	.....
MUSIC .....	(cf. Pom's n° 10)	à	80,00 F	.....
DISK-MANAGER .....	(cf. Pom's n° 11)	à	450,00 F	.....
DBSTAG (CP/M) .....	(cf. Pom's n° 11)	à	450,00 F	.....
JEUX A .....	(cf. Pom's n° 12)	à	80,00 F	.....
JEUX B .....	(cf. Pom's n° 12)	à	80,00 F	.....
BASICIUM .....	(cf. Pom's n° 13)	à	150,00 F	.....
E.P.E. ....	(cf. Pom's n° 15)	à	150,00 F	.....
DEMO MAX THE GOLBE TROTTER	(cf. Pom's 17)	à	55,00 F	.....
MAX (moniteur étendu) .....	(Cf. ce numéro)	à	150,00 F	.....
MACINTOSH N° 14-15-16 .....		à	150,00 F	.....
MACINTOSH N° 17 .....		à	80,00 F	.....
MACINTOSH N° 18 .....		à	80,00 F	.....
PASCAL .....	(cf. Pom's n° 15)	à	80,00 F	.....

## RECUEILS

N° 1, recueil des revues 1 à 4 .....	à	140,00 F	.....
Disquettes d'accompagnement des numéros 1 à 4 .....	à	150,00 F	.....
N° 2, recueil des revues 5 à 8 .....	à	140,00 F	.....
Disquettes d'accompagnement des numéros 5 à 8 .....	à	190,00 F	.....

## ANCIENS NUMEROS

REVUES	<input type="checkbox"/> 4	<input type="checkbox"/> 7	<input type="checkbox"/> 8	.....	à	35,00 F	.....		
REVUES	<input type="checkbox"/> 9	<input type="checkbox"/> 10	<input type="checkbox"/> 11	<input type="checkbox"/> 12	<input type="checkbox"/> 13	.....	à	40,00 F	.....
	<input type="checkbox"/> 14	<input type="checkbox"/> 15	<input type="checkbox"/> 16	<input type="checkbox"/> 17	<input type="checkbox"/> 18				
DISQUETTES	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	.....	à	55,00 F	.....
	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9	<input type="checkbox"/> 10				
	<input type="checkbox"/> 11	<input type="checkbox"/> 12	<input type="checkbox"/> 13	<input type="checkbox"/> 14	<input type="checkbox"/> 15				
	<input type="checkbox"/> 16	<input type="checkbox"/> 17	<input type="checkbox"/> 18						

## ABONNEMENTS

POUR 6 NUMEROS à partir du n .....	
ABONNEMENT SANS DISQUETTES .....	à 200,00 F
ABONNEMENT AVEC DISQUETTES (Apple II +, //e, //c) .....	à 480,00 F

TOTAL TTC .....

Supplément expédition  
par avion à l'étranger .....

MONTANT  
DU REGLEMENT .....

**Ces tarifs comprennent l'envoi postal en France métropolitaine, CEE et Suisse  
Supplément avion hors CEE : 15 F par numéro et/ou disquette**

Envoyez ce bon et votre règlement à (Abonnés : n'oubliez pas de joindre l'étiquette-adresse).

**Editions MEV - 64-70, rue des Chantiers - 78000 VERSAILLES**

Nom .....

Adresse .....

# Le nouveau LASER 3000

Nouveau système  
d'exploitation  
DOS 3.3/PRODOS/CPM



Enfin un « PRO » complet  
avec lecteur de disquettes à : **4.980<sup>F</sup>** TTC



## **COMPAREZ !**

### **Processeur :**

6502 A / 2 Mhz + en option Z80A disponible.

### **Système d'exploitation :**

Dos 3.3 + en option CPM.

### **Mémoire morte :**

32 K ROM.

### **Mémoire vive :**

64 K extensible à 192 K RAM.

### **Affichage :**

24 lignes.

Carte 40 et 80 colonnes incorporées.

### **Graphisme :**

280x192 et 560x192 points.

Carte couleur incorporée.

### **Clavier :**

81 touches ergonomiques, antifatigue.

Qwerty et Azerty au choix, avec pavé numérique

+ 24 touches de fonction.

### **Son :**

4 canaux / 6 octaves.

### **Entrée/sortie :**

Centronics parallèle + en option RS 232 disponible.

## **DES MILLIERS DE LOGICIELS AU BOUT DES DOIGTS !**

Polyvalent grâce à ses différents microprocesseurs 6502 A et Z80, le LASER 3000 offre le plaisir d'utiliser les meilleurs logiciels de langages, de gestion, d'organisation, de calculs scientifiques, de musique, soit des milliers de programmes :

- PRACTICALC,
- MULTIPLAN,
- SAARI,
- ZAXXON...

Ex : tableau Practicale : 890 F TTC

ENCORE UNE RÉALISATION



**VIDEO TECHNOLOGIE FRANCE**

19, rue Luisant - 91310 Monthéry  
Tél. (6)901.93.40 - Télex : 692 169 F

Documentation couleur et liste des revendeurs, sur simple demande.

Livré complet avec unité centrale, clavier Azerty accentué + pomme ouverte/pomme fermée, contrôleur et lecteur de disquettes + manuel technique en français + câble...

Demandez le catalogue gratuit  
chez votre revendeur habituel  
ou chez :  
PSI diffusion BP 86  
77402 Lagny s/ Marne Cedex  
Tél. (6) 006.44.35

# APPLE A LIVRE OUVERT

## DES LIVRES ET DES DISQUETTES POUR TOUS

### POUR DÉBUTER

Apple pour tous ; la découverte de l'Apple ;  
36 programmes Apple pour tous ; exercices pour  
Apple ; 102 programmes pour Apple ; la pratique  
de l'Apple tomes 1 et 2 ; le Basic et l'école tomes  
1 et 2 ; Macintosh le magnifique ; programmer  
le Macintosh ; multiplan pour Macintosh.

### POUR GÉRER

La nouvelle comptabilité sur  
Apple tomes 1 et 2 ; La factu-  
ration et ses annexes ; outils  
financiers et comptables pour  
l'entreprise ; la paie et ses annexes.

### POUR PROGRESSER

Clefs pour Apple II ; clefs pour Apple // c ;  
la pratique de l'Apple tome 3 ; l'Apple et ses  
fichiers ; gestion de fichiers et de périphériques  
pour AppleII/Pascal ; Microbook, base de données  
pour Apple ; Pascal UCSD sur Apple tomes 1 et 2 ;  
du Logo pour Apple ; ProDOS sur Apple ;  
Pangraphe ; modèles d'expression graphique.

### POUR DÉCIDER

Modèles pratiques de décision  
tomes 1 et 2 ; Visicalc sur Apple ;  
Multiplan pour Apple II ; bases  
de données sur Apple II ;  
bibliothèque scientifique en  
Pascal ; tests statistiques  
usuels.

