

A PROPOS DU CLUB

Le Bureau
J. Belin
J. Belin

Editorial 1
Un nouveau partenaire pour le Club 2
Bruits et rumeurs 2

HP28

G. Toublanc
G. Toublanc
G. Toublanc

Input HP28S simplifié 4
Anti Memory Lost 5
Suppression de délimiteurs de programmes 6

HP48

G. Toublanc
G. Toublanc
L. V. Huynh
G. Toublanc
G. Toublanc

Le petit train de PPC 8
Voyeur de Grobs 10
Réduction des programmes 11
Suppression de délimiteurs de programmes 12
Ensemble de Mandelbrot (Correctif) 13

HP95 / HP100

R. Amram
J. Belin
J. Belin

Réflexions et carte Fax EXP 16
Conflit entre 100BUDDY et HELV100 19
Création d'animations (1ere partie) 19

Le coin des codes 30

EDITORIAL

Tout d'abord, excusez nous du retard dans la parution de ce JPC. Il semble que, une fois tous les six mois, une avalanche de contretemps, changements de programme et autres incidents surviennent sans prévenir au moment même où nous devons nous occuper du journal !

Pour ne pas entrer dans les détails, disons que cela a commencé avec l'annulation très tardive (moins de dix jours avant la date prévue !) de la conférence d'Amsterdam, qui non seulement nous priva du plaisir de retrouver nos amis des clubs étrangers, mais aussi déjoua nos prévisions d'y consacrer une grande partie de ce numéro... Quand à la dernière raison ayant retardé la mise en page du journal, il s'agit d'une cascade de problèmes survenus lors des dernières étapes du développement d'un programme. Gênant lorsque le programme en question est le but final d'un article d'une vingtaine de pages devant paraître sur deux numéros...

Une fois de plus, nous tenons à signaler que ces problèmes n'auraient probablement eu aucune incidence sur la date de sortie du journal, si nous avions eu un stock d'articles suffisant pour alimenter ce numéro... A propos, nous encourageons tous les lecteurs (y compris les utilisateurs de HP28/48 de lire la première partie de l'article de Robert Amram, que nous remercions en passant pour son excellent article...

Pour revenir brièvement sur la conférence d'Amsterdam, sachez qu'elle a été annulée par faute d'un nombre suffisant de participants, et que Prompt a proposé de la reporter au premier octobre, date à laquelle ils espèrent un plus grand nombre de personnes disponibles. Suite à cette annulation, le club Anglais HPCC a décidé d'organiser le 13 aout une mini conférence afin de fêter les dix ans (déjà !) du HP-71. Puisque nous parlons de conférences, nous pouvons ajouter que Craig Finseth (l'auteur de Freyja et Memutil), se propose d'organiser l'année prochaine (en juillet ou en aout) une conférence aux USA (dans le Minesota).

Maintenant une mauvaise nouvelle. Nous avons le regret de vous annoncer que la collaboration que nous avons depuis de très longues années avec le magasin *Maubert Electronic* fait maintenant partie du passé. Les raisons exactes seraient trop longues à expliquer, mais elles sont dûes principalement à de profondes divergences d'opinions, à lesquelles on peut ajouter plusieurs atteintes à notre liberté d'expression. Dans tous les cas, la perte n'est pas très grande, puisque nous nous étions rendu compte depuis longtemps que le club n'avait plus rien à attendre de cette boutique...

Pour finir, nous tenons à rappeler aux nouveaux adhérents (et aux distraits) que nos réunions mensuelles s'interrompent pendant les vacances scolaires, le Centre Jean Verdiar étant fermé pendant cette période. Notez donc le prochain rendez-vous : le premier octobre.

UN NOUVEAU PARTENAIRE

En remplacement de Maubert Electronic, nous avons le plaisir de vous présenter le nouveau partenaire du club. Il s'agit d'Infopoint, qui est un des principaux grossistes HP. De nombreux utilisateurs ont déjà eu des contacts (souvent sans le savoir) avec cette société, puisque c'est elle qui assure le support technique calculatrices depuis le début de l'année.

Pour vous, cette collaboration se formalise bien sûr par la possibilité d'obtenir des réductions substantielles sur votre matériel préféré. Cependant, pour des raisons contractuelles existant entre HP et Infopoint (sans aucun rapport avec le club), les taux de réduction qui nous sont accordés varient en fonction du type de matériel acheté :

- 35% de réduction pour toute la branche calculatrices (HP48 et accessoires).
- 22% de réduction pour toute la branche palmtops (HP95/100 et accessoires).

Le prix de base étant le "prix catalogue" HP (TTC).

A titre d'exemple, voici quelques prix indicatifs :

- HP48GX : 1700 F TTC.
- Carte 1Mo pour HP48GX : 2230 F TTC.
- HP100 1Mo : 3960 F TTC.
- Carte Flash (type SunDisk) 5Mo : 2660 F TTC.

Notez bien qu'il s'agit de vente par correspondance uniquement (pas d'enlèvement sur place). Le port est gratuit, mais la commande minimum est fixée à 800 F. Toute commande doit être accompagnée du chèque et d'une photocopie de la carte de membre.

Pour tous renseignements supplémentaires, vous pouvez vous adresser à :

INFOPOINT
Joelle Le Ginche
1, rue de Terre Neuve
Z.A. de Courtaboeuf
B.P. 62
91942 LES ULIS CEDEX
Tel : (1) 69 18 20 65

Notez enfin que si la demande est suffisante, des possibilités d'achats groupés d'un même matériel peuvent être étudiées. Faites nous connaître vos souhaits !

Jacques Belin (123)

ELLES COURENT, LES RUMEURS...

Certains d'entre vous ont probablement eu entendu parler des nombreuses rumeurs qui ont circulé sur Internet et CompuServe à propos d'un nouveau palmtop. Celles-ci sont apparues suite à l'apparition d'une référence "HP200LX2MXP" dans un catalogue (à paraître) d'un distributeur américain. Bien sûr, cela n'est peut être qu'une erreur de typographie... Or, depuis, d'autres distributeurs US auraient confirmé l'information et annonceraient sa sortie pour le mois d'aout, pendant que d'autres (généralement aussi bien informés) ont démenti. Bref, allez savoir... Dans tous les cas, si cette machine existait, et malgré le fait que la machine pourrait porter un nouveau numéro, aucune mention n'est faite sur un changement du hardware (microprocesseur, écran...). En fait, d'après les descriptions données, il ne s'agirait que d'un HP100 2Mo sur lequel on aurait remplacé le logiciel de messagerie *cc:Mail* par *Quicken*, un logiciel de gestion financière très répandu aux USA. Il est vrai que les développeurs de Corvallis ont probablement remarqué (ne serait-ce qu'en lisant les messages sur CompuServe) que *Quicken* est très utilisé sur le HP100, alors que *cc:Mail* semble être totalement ignoré...

Autre chose. Certains d'entre vous ont peut-être lu dans un récent *Décision Micro*, une interview de Philippe Germond, le responsable de la division micro de HP en France. Dans celle ci, il annonçait entre autres : *"..Et nous travaillons actuellement sur une nouvelle technologie de batterie qui nous assurera une bonne longueur d'avance pendant quelque temps."* En fait, il pourrait s'agir d'un accord de participation entre HP et une société appelée Valance, autour de batteries de type "Lithium-Polymère" qui sembleraient avoir une plus grande capacité que les batteries NiMH alimentant les portables actuels. La production de ces batteries devrait débuter au début de l'année prochaine.

Jacques Belin (123)

HP28

G. Toublanc
G. Toublanc
G. Toublanc

Input HP28S simplifié	4
Anti Memory Lost	5
Supression de délimiteurs de programmes	6

Le coin des codes	30
--------------------------	-----------

INPUT HP28S SIMPLIFIE

La commande INPUT des HP48 fait défaut sur nos HP28. Cela m'avait posé des problèmes pour réaliser le programme de saisie des codes ASSC0028. Pour les besoins de la cause j'avais simulé INPUT mais pour les seules nécessités de ASSC0028, donc cet INPUT est très spécifique. Dans le livre de William Wickes *HP28 Insights* on trouve des programmes PROMPT pour simuler cette commande HP-41 via HALT, mais pas de INPUT. Pour pallier à cette lacune je vous propose un INPUT28 simplifié qui n'aura donc pas toutes les fonctionnalités de la commande HP48 mais qui peut, malgré ses limites, être utile aux programmeurs.

Syntaxe :

niveau 1 : 1 chaîne "question"

INPUT28R → l'objet réponse

Fonctionnement du programme:

dans un programme on trouvera la séquence suivante:

```
« ..... "chaîne question" INPUT28R ..... »
```

la chaîne question aura de une à trois lignes avec les NEWLINES éventuels. Elle sera affichée pendant que la réponse sera saisie jusqu'à la validation avec ENTER. Vous pouvez interrompre la saisie et le programme avec ATTN. La réponse est affichée au fur et à mesure de son entrée. Après validation la réponse est analysée et transformée en objet résultat si cela est correct, sinon arrêt avec message Syntax Error. Les corrections peuvent se faire avec la touche [=] (BACK).

Exemple:

```
« .....  
"borne inférieure  
de la fonction ?"  
INPUT28R  
.....  
"borne supérieure ?"  
INPUT28R  
.....  
"pas d'incrémentation"  
INPUT28R  
..... »
```

A la 1ère question, vous entrez par exemple :

-5.6 ou -.00125 ou -1.25E-3

Si vous avez une donnée numérique stockée dans la variable TOTO, vous pouvez très bien avoir une réponse du type :

```
TOTO 5 *
```

Donc il faut toujours respecter la syntaxe Rpn et se limiter à des données numériques.

Ce que l'on ne peut pas faire :

- entrer des expressions algébriques.
- utiliser la touche [shift] pour avoir accès aux commandes telles % ^ etc...
- entrer des chaînes.

INPUT28R

160 octets cksum # 461Dh

```
::  
CK1NOLASTWD  
Dispatch1  
THREE  
::  
DOCLLCD  
%1 xDISP  
NULL$  
BEGIN  
DUP DISPROW4  
BEGIN  
CKOATTNABORT  
xKEY %0=  
WHILE  
REPEAT  
DUP $_BACK EQUAL  
ITE :: DROP ONE OVER LENS #1- SUB$ NULL$ TRUE ;  
:: DUP $_ENTER EQUAL NOT ;  
WHILE  
&$  
REPEAT  
xCLMF  
DROP  
palparse  
ITE EVAL :: 2DROP CKONOLASTWD  
CODE  
ad0ex  
d0=(5) #c015f  
c=0 a  
dat0=c a  
ad0ex  
lchex 106  
govlng =ErrjmpC  
ENDCODE  
;  
;  
;
```

Remarques:

- Je n'ai pas utilisé toutes les possibilités offertes par `palparse` pour rester compatible avec les limites fournies par le système de saisie des réponses. Pour réaliser un `INPUT` polyvalent comme celui des HP48 il faudrait aller beaucoup plus loin en programmation système et aussi en encombrement mémoire, ce dont HP lui-même s'est abstenu.

- Vous pouvez rebaptiser ce programme en `INPUT` tout simplement mais j'ai tenu à rappeler que c'est un `INPUT HP48` réduit d'où `INPUT28R`.

Vous trouverez la liste des codes dans *le coin des codes*.

Guy Toublanc (276)

AMNESIE BIENFAITRICE

Si le message `Memory Lost` de nos HP28 est tant redouté pour la plupart, il peut apparaître parfois comme l'annonce d'une remise en ordre de la mémoire mettant fin à certaines incohérences provoquées par une utilisation anormale de la machine (Sysevals mal contrôlés, programmes en System Rpl ou assembleur au fonctionnement anormal). Lors de l'apparition des premiers symptômes de ces maux on aimerait bien faire cette remise en ordre mais comment ne pas perdre nos fichiers péniblement rentrés ? Il y a une solution que certains connaissent déjà: le programme `AML` pour *Anti-Memory-Lost*. Je ne connais pas l'auteur de ce programme mais je le remercie de l'avoir créé et je pense que `AML` doit être connu d'un plus grand nombre d'utilisateurs.

Utilisation et limites de `AML`.

Après avoir tapé et socké le programme, lancer `AML`. Celui-ci va stocker, au début du répertoire `HOME`, une chaîne dans la variable `SAV`.

Lors de l'exécution de `AML`, celui-ci affiche le message `# CFB7Dh SYSEVAL` pour vous rappeler le `Syseval` à exécuter.

A partir de maintenant vous pouvez faire un `RESET` avec les trois touches `[ON]` `[INS]` `[RIGHT]` appuyées simultanément ce qui provoquera un *Memory Lost*. Pour retrouver la mémoire il faut taper :

```
# CFB7Dh SYSEVAL
```

puis faire un arrêt système avec les touches `[ON]` et `[UP]` appuyées simultanément. Vous devez retrouver vos chers programmes.

Si la partie *code* (voir ci-dessous) de `SAV` n'a pas été altérée le `# CFB7Dh SYSEVAL` est réutilisable.

Attention :

- Pour que `AML` soit opérationnel il faut que `SAV` soit absolument la première variable du répertoire `HOME`.

- En général `AML` n'est efficace que pour des `Memory Lost` provoqués par un `RESET` effectué comme ci-dessus et non par un programme qui plante la machine. Dans ce dernier cas vous pouvez tenter la récupération de la mémoire mais cela ne marche pas souvent. Si un `POKE` a été effectué par vous ou un programme dans l'espace mémoire occupé par une ou plusieurs des variables de votre HP28, ne comptez pas sur `AML` pour vous restituer l'état initial de ces variables (`SAV` comprise).

Voici donc le programme guérisseur :

`AML`

```
377 octets cksum # 7D93h
```

```
«
[ 16 187 183 207 163 184 42 16 0 2 32 26 8 81 18
 25 102 31 35 177 135 0 28 4 129 25 4 216 0 0 0 ]
1
DO
  GETI CHR 3 ROLLD
UNTIL
  DUP 1 - NOT
END
SWAP SIZE 1 GET 1 -
START
+
NEXT
"#CFB7Dh SYSEVAL"
CLLCD 2 DISP LCD+ +
HOME 'SAV' DUP PURGE STO
»
```

Que fait ce programme `AML` ?

Je ne peux donner que quelques explications car je n'ai pas été très loin dans mes investigations. Le programme génère une chaîne de 31 caractères qui correspond au code suivant :

```
01BB7BFC3A8BA201002002A18015219
166F1321B7800C1401891408D000000
```

Puis il concatène une chaîne de 548 caractères générés par LCD→ dont le contenu importe peu, seul l'espace ainsi réservé est primordial.

Ainsi notre chaîne de 579 caractères est stockée dans SAV au début du répertoire HOME pour être à une adresse bien déterminée: # CFB70h, c'est-à-dire :

D0000h - 1158 - 10

où :

D0000h après la fin de la Ram

1158 quartets : espace pour les 579 caractères

10 quartets pour prologue chaîne et longueur

Lorsque l'on fait # CFB7D SYSEVAL on exécute le code à cette adresse et contenu au début de la chaîne SAV d'où la nécessité absolue que SAV soit en première position dans HOME.

Après un RESET on constate que le code contenu dans la partie correspondant aux 31 premiers caractères n'est pas modifié, seul celui correspondant aux 548 caractères suivants est modifié et donc cette partie de SAV sert de zone tampon.

Pour aller plus loin dans les explications il faudrait désassembler le code de la Rom correspondant à la fois aux routines générant un *Memory Lost* et un arrêt système. A ce propos vous pouvez exécuter un RESET avec # 1FD6h SYSEVAL au lieu de l'appui simultané sur les touches [ON] [INS] [RIGHT]. L'effet sera le même.

Traduction pour radin

Pour ma part j'utilise une transcription en System Rpl du programme AML, quatre fois moins encombrante que l'original. La voici :

AML2

92.5 octets # 739Ah

```
::
$ "\10\bb\b7\cf\3\b8\2a\10\00\02\20\1a\08\51\12
  \19\66\1f\23\b1\87\00\1c\04\81\19\04\d8\00\00\00"
$ "#CFB7Dh SYSEVAL"
DOCLLCD
%2 xDISP
DOLCD>
&$
xHOME
,
ID SAV
DUP xPURGE xSTO
;
```

dont vous trouverez la liste des codes dans *le coin des codes*.

Dernières recommandations. Notez en lieu sûr le :

CFB7Dh SYSEVAL

et ayez toujours en mémoire les limites de cet *Anti-Memory-Lost* qui ne peut pas remédier à toutes les agressions subies par la vulnérable mémoire de votre chère HP28. Donc ne tentez pas le diable, surtout si votre machine n'est pas équipée de connecteurs pour permettre les sauvegardes.

Guy Toublanc (276)

P.S : Si quelqu'un a d'autres informations sur AML, qu'il se manifeste, il sera le bienvenu.

SUPPRIMEZ LES EXTREMES

Dans l'article *Allègements en douceur* (JPC 96) nous avons vu comment réduire l'encombrement de programmes comprenant des Sysevals et par la même occasion aussi supprimer les délimiteurs '«' et '»'. Peut-être que vous n'utiliserez pas de programmes avec Sysevals, mais vous aimeriez gagner ici et là 5 octets en éliminant les délimiteurs et pour cela avoir un mini-programme pour le faire. Dans ce cas SHRINK est fait pour vous et pour le prix de 30 octets. Son emploi est simple puisqu'il suffit de mettre le programme sur la pile et de lancer SHRINK pour obtenir une copie allégée de 5 octets. Le nouveau venu ne sera pas éditable pour le modifier, pour cela il faudra avoir recours à PRG→ et →PRG (voir JPC 96 et 87).

SHRINK28

30 octets cksum # 57ACh

```
:: CK1NOLASTWD Dispatch1 EIGHT
:: INNERCOMP SWAPDROP
#2- ::N SWAPDROP
;
;
```

Voir la liste des codes dans *le coin des codes*.

Exemple d'utilisation :

Pour remonter dans le répertoire HOME, je mets en début de sous-répertoire le programme: « HOME » sans les délimiteurs et dans la variable '←' obtenue en faisant (←) LIST→ DROP STO. Cela est économique car ne prenant que 2,5 octets pour le contenu de la variable '←'.

Guy Toublanc (276)

HP48

G. Toublanc
G. Toublanc
L. V. Huynh
G. Toublanc
G. Toublanc

Le petit train de PPC	8
Voyeur de Grobs	10
Réduction des programmes	11
Supression de délimiteurs de programmes	12
Ensemble de Mandelbrot (Correctif)	13
Le coin des codes	30


```
* correspondant à l'écran *
*****
```

```
BEGIN
  3GETLAM
  #0<>
```

```
*****
* déroulement du défilement tant que son compteur *
* est positif ( valeur initiale : 226 = largeur du *
* grob - 1 *
*****
```

```
WHILE
  3GETLAM SIX
  4GETLAM
  XYGROBDISP
  %RAN % .35 %>
```

```
*****
* affichage du grob du train à : <6d> , <compteur> *
* l'avancement se faisant de droite à gauche au fur *
* et à mesure que le compteur décroît *
* les coordonnées des pixels de fumées sont dans *
* une liste *
* si le nombre aléatoire actuel est > 0,35 alors *
* s'il y a des listes de coordonnées de pixels *
* de fumées (déjà affichés) sur la pile alors *
* on prend la liste de n° = nombre de *
* listes x nombre aléatoire actuel *
* et on efface le pixel correspondant *
*****
```

```
IT
  :: DEPTH 5GETLAM #-DUP
  #0=ITE DROP
  :: UNCOERCE %RAN %* XROLL
  INCOMPDROP PIXOFF3
  ;
  ;
  3GETLAM #1- 3PUTLAM 2GETLAM
```

```
*****
* le compteur d'avancement est décrémenté *
* si le booléen de contrôle de génération des *
* fumées est TRUE alors *
* il bascule à FALSE *
* un 1er pixel de fumée est affiché *
* un 2ième pixel et un 3ième pixel sont *
* affichés en conservant leurs coordonnées *
* sur la pile pour l'effaçage ultérieur *
*****
```

```
IT
  :: FALSE 2PUTLAM
  3GETLAM #3+ SIX PIXON3
  3GETLAM #5+ FOUR 2DUP PIXON3
  TWO{N} 3GETLAM #6+ FOUR 2DUP PIXON3
```

```
TWO{N}
;
%RAN % .97 %> 3GETLAM 96 #> AND
```

```
*****
* si le nombre aléatoire actuel est > 0,97 et si le *
* compteur est > 96 (donc la locomotive ne *
* commence pas à disparaître à gauche alors *
* on effectue 5 fois une boucle avec : *
* émission d'un 1er sifflement *
* le compteur est décrémenté et le train *
* avance d'un pixel *
* un 2ième sifflement est émis *
*****
```

```
IT
  :: FIVE #1+_ONE_DO (DO)
  2.E3 %.1 2DUP DOBEEP
  3GETLAM ONE #MAX #1-DUP 3PUTLAM SIX
  4GETLAM XYGROBDISP DOBEEP
  LOOP
  ;
  %RAN % .75 %>
```

```
*****
* si le nombre aléatoire actuel est > 0,75 alors *
* le booléen de contrôle d'émission de fumées *
* devient TRUE *
* un 1er pixel de fumée est affiché *
* un 2ième pixel et un 3ième pixel sont *
* affichés en conservant leurs coordonnées *
* sur la pile pour l'effaçage ultérieur *
*****
```

```
IT
  ::
  TRUE 2PUTLAM 3GETLAM #3+
  DUP SIX PIXON3
  FIVE 2DUP PIXON3 TWO{N}
  3GETLAM #4+ FIVE 2DUP PIXON3 TWO{N}
  ;
  %.1 dowait
  REPEAT
```

```
*****
* le tour de boucle se termine par une temporisation*
* pour régulariser la vitesse d'avancement du train *
* entre séquences avec émission de sifflements ou *
* non émission *
* ----- *
* après sortie de la boucle on élimine les listes *
* de coordonnées de pixels de fumées non effacés *
* on repasse en mode texte et on restaure *
* l'affichage initial avant l'animation graphique *
* l'environnement des variables locales est détruit *
*****
```

```
DEPTH 5GETLAM #- NDROP
TOADISP 1GETLAM GROB>GDISP
ABND
;
```

La liste des codes se trouvera dans *le coin des codes*.

Bon amusement.

Guy Toublanc (276)

VOYEUR DE GROBS

A propos d'un article publié dans 48Sxtant (no 27) et donnant quelques recettes pour optimiser, sans trop se fatiguer, un programme en Sytem Rpl et assembleur, j'avais créé une version plus *correcte* du programme SCROLLER paru dans *Voyage Au Centre De La HP48G/GX*. Ce programme me semblant plus utile aux possesseurs de HP48 S/SX qu'à ceux de HP48G/GX j'ai pensé que les lecteurs de JPC pourraient aussi en bénéficier. Je ne reprendrai pas toutes les explications de mon article antérieur, le but étant de fournir un utilitaire. Je signalerai seulement les *plus* de ma version:

- Elle ne plante pas la machine pour des grobs de dimensions inférieures à 131 x 64.
- Elle occupe 129 octets contre 175.
- Elle n'utilise pour le System Rpl que des points d'entrée supportés donc un avenir plus assuré pour les éventuelles nouvelles versions de HP48.
- la logique élémentaire de programmation y est un peu plus respectée. On ne trouve pas par exemple :

```
#0= ITE TRUE FALSE UNTIL
      au lieu de #0= UNTIL !!!!!
```

ce qui fera au moins sourire les programmeurs en System Rpl et aussi les autres programmeurs en User Rpl quand il sauront que dans leur langage préféré cela peut se traduire par:

```
UNTIL 0 == IF THEN 1 ELSE 0 END END
      au lieu de UNTIL NOT END !!!!!
```

(Alors, ceux qui sont complexés et n'osent pas publier leurs programmes, je vous en prie soyez plus osés, car vous ne programmerez certainement pas avec plus d'inélégance et PPC Paris est un club ouvert à tous les programmeurs, dans tous les langages pour HP48 et

pour tout motif de programmation. Donc pas de sectarisme et laissez vos complexes à la porte du club).

Voici donc le listing de la version que vous pourrez comparer à celle qui a servi de *modèle*.

GSCROLL

129 octets cksum # F57h

```
*****
* fichier source compilable sur PC par RPLCOMP de HP*
* ou sur HP48 avec ->RPL de D. Mueller (en mettant *
* en majuscules les mnémoniques de l'assembleur) *
*****
```

```
::
CK1NoBlame CK&DISPATCH1 TWELVE
::
GROB>GDISP ZEROZERO WINDOW#
BEGIN
CODE
=OUT<C>IN equ #01eec
      gosbvl =SAVPTR
      a=0    a
      st=0   15
loop   lchex 080
      gosub  out<c>in
      ?cbit=1 1
      goyes  up
      lchex  40
      gosub  out<c>in
      ?cbit=1 0
      goyes  right
      ?cbit=1 1
      goyes  down
      ?cbit=1 2
      goyes  left
      lchex  10
      gosub  out<c>in
      ?cbit=1 0
      goyes  out
      lchex  1FF
      out=c
      shutdn
      gonc   loop
left   a=a+1  a
right  a=a+1  a
down   a=a+1  a
up     a=a+1  a
out    govlng =PUSH#ALOOP
out<c>in govlng =OUT<C>IN
ENDCODE
::
ONE OVER#=case WINDOWUP
TWO OVER#=case WINDOWDOWN
THREE OVER#=case WINDOWRIGHT
```

```

FOUR OVER#=case WINDOWLEFT
;
#0= UNTIL
FLUSHKEYS
;
;

```

Voir la liste des codes dans *le coin des codes*.

Mode d'emploi :

Le programme permet de visionner rapidement des objets graphiques en faisant défiler l'affichage avec les quatre flèches de déplacement. La sortie s'effectue avec la touche Backspace (␣). Ayez le doigt léger car c'est très rapide et répétitif.

Ce programme est donc utile pour visionner des objets graphiques dont les dimensions dépassent celles de l'écran HP48.

Il suffit de mettre sur la pile un objet graphique, par exemple les triangles de Sierpinski (JPC 89), et de lancer GSCROLL.

Les possesseurs de HP48S/SX apprécieront plus ce programme que ceux de HP48G/GX. En effet ce n'est qu'avec la version J des HP48S/SX que la vitesse de défilement de l'affichage a été améliorée notablement pour devenir satisfaisante avec les HP48G/GX.

Je dois remercier notre ami Robert Pulluard qui m'a gentiment autorisé à réutiliser le travail que j'avais fourni pour son journal 48Sxtant.

Guy Toublanc (276)

REDUCTIONS ET SUPPRESSIONS

STRIP est un petit utilitaire pour User-RPL. Il permet de réduire la taille et augmenter la vitesse d'exécution d'un programme en supprimant ou en remplaçant les commandes inutiles.

STRIP supprime :

```

' [']
DO [UNTIL] END
[CASE] ... [END]
[IF] THEN ELSE [END]
IFERR THEN ELSE [END]

```

STRIP remplace :

```

x'          par      '
xSILENT'    par      '
x<<<        par      NOP
x>>>ABND    par      ABND
#addr SYSEVAL par      addr

```

Voici la source System RPL :

```

TITLE STRIP v1.0 23/04/94 (c) 1994 Lu Vy HUYNH

ASSEMBLE
  NIBASC \HPHP48-P\

RPL
::
  OLASTOWDOB! CK&DISPATCH1 EIGHT
  ::
    INHARDROM? ?SEMI
    DUPNULLCOMP? ?SEMI
    INNERCOMP
    OVER' x>>
    EQIT
  ::
    DUP #1+ROLL ROT2DROP #2-
  ;

  (
    ISTOP@ #1- ISTOPSTO INDEX@ #1- INDEXSTO
  )
  1LAMBIND

  DUP#0_DO (DO)
  ROLL
  ::
    ' xSYSEVAL
    EQcase
  ::
    OVER TYPEHSTR? NOT?SEMI DROP
    TWO NTHCOMPDROP 1GETLAM COMPEVAL
  ;

  ::
  ::
    DUP' xIF
    EQ
    caseDROP

    DUP' xUNTIL
    EQ
    caseDROP

```

```

::
DUP' xTHEN
EQ ?SEMI

DUP' xERRTHEN
EQ ?SEMI

' xCASE
EQcasedrop
:: 1GETLAM COMPEVAL ;

' x'
EQcasedrop
:: ' ' ;

2RDROP
;
ISTOP@ #1-DUP ISTOPSTO ROLL
DUP' xELSE
EQIT
::
ISTOP@ SWAP OVER #1+UNROLL
#2- ROLL
;
DROP 2RDROP
;
1GETLAM COMPEVAL RDROP
;

' x<<
EQcasedrop
'NOP

' x>>ABND
EQcasedrop
:: ' ABND ;

' xSILENT'
EQcasedrop
:: ' ' ;

DUPTYPECOL?
IT
ID STRIP
;
ISTOP@
LOOP
::N
;
;

```

STRIP.M

```

OUPUT STRIP
LLIST STRIP.LR
SUPPRESS XREF
SEARCH C:\HP\HPTOOLS\TOOLS\ENTRIES.O
REL STRIP.O
END

```

MKSTRIP.BAT

```

@ECHO OFF
RPLCOMP STRIP.S STRIP.A
SASM STRIP.A
SLOAD -H STRIP.M

```

Lu Vy Huynh (579)

CHASSEZ LES EXTREMES

Dans l'article *Allègements en douceur* (JPC 96), le programme COMPIL48 permet de compiler des Sysevals et aussi de supprimer les délimiteurs '«' et '»'. Peut-être que cette toute dernière possibilité vous intéresse seulement, alors SHRINK est fait pour vous et pour le prix de 30 octets. Son emploi est simple puisqu'il suffit de mettre le programme sur la pile et de lancer SHRINK pour obtenir une copie allégée de 5 octets. Le nouveau venu ne sera pas éditable pour le modifier, donc sauvegardez une copie de l'original.

SHRINK48

30 octets cksum # 6460h

```

::
CK1NoBlame
CK&DISPATCHO
seco
::
INNERCOMP
SWAPDROP
#2-
::N
SWAPDROP
;
;

```

A titre d'information, voici les fichiers STRIP.M et MKSTRIP.BAT permettant de générer ce programme :

Le principe est simple:

- on teste s'il y a un niveau de pile et si sur celle-ci l'argument est un objet composé de type programme. Si oui, il est décomposé en ses éléments avec INNERCOMP qui renvoie aussi le nombre N des éléments. Le dernier élément '»' est supprimé avec SWAPDROP (échange de N et '»' sur la pile puis DROP sur celui-ci). Le nombre N d'éléments est actualisé: N - 2 (pour '«' et '»' supprimés). On recompose le programme pour ce nouveau nombre d'éléments avec ::N puis '«' est éliminé avec SWAPDROP (échange de '«' et du nouveau programme puis DROP sur '«'). Voilà, le tour est joué! Ceux qui n'osent pas encore se lancer dans ce type de programmation peuvent commencer avec des petits programmes de ce genre. Evidemment, ne laissez pas tomber le User Rpl qui est largement suffisant pour beaucoup de choses.

Voir la liste des codes dans *le coin des codes*.

Guy Toublanc (276)

MISE AU POINT POUR MANDEL

Dans l'article sur les fractales (JPC 96) deux erreurs se sont glissées:

- Page 20, pour le mode d'emploi de MZOOM il fallait lire :

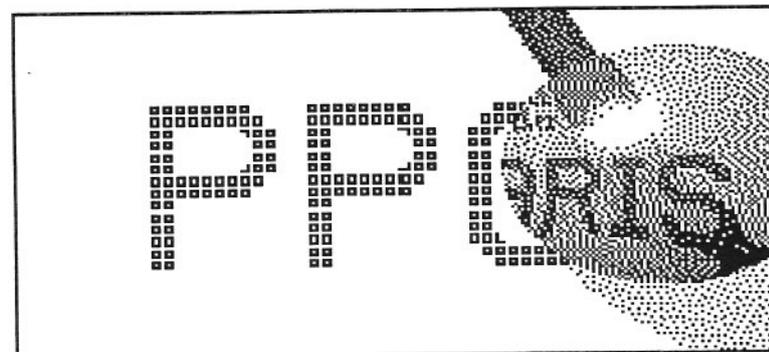
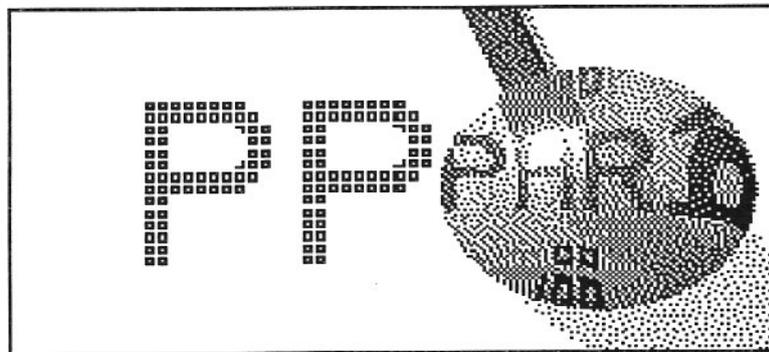
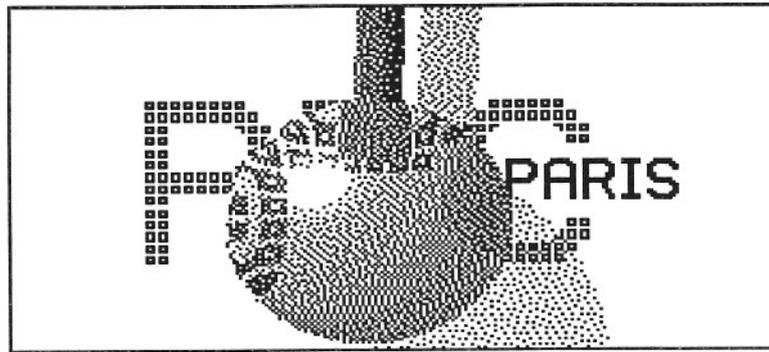
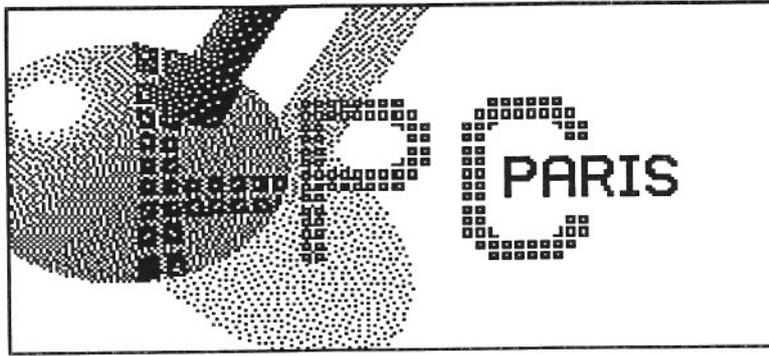
```
niveau 4 : ( Xc , Yc )
niveau 3 : ( Px , Py )
niveau 2 : iter
niveau 1 : le grob
```

- page 20 après : le curseur peut changer de couleur avec [+ -] sur les HP-48G/GX comme sur les séries S.

Mea culpa.

Guy Toublanc (276)





HP95 / HP100

R. Amram	Réflexions et carte Fax EXP	16
J. Belin	Conflit entre 100BUDDY et HELV100	19
J. Belin	Création d'animations (1ere partie)	19

REFLEXIONS ET CARTE FAX

Reflexions...

Note de Jacques Belin : Comme à notre habitude, lorsque nous recevons ce genre d'article, nous nous permettons (avec l'accord de l'auteur) d'y inclure nos propres réponses. Tout ce qui est en italique doit donc m'être attribué.

Je voudrais avant tout remercier Jacques Belin et notamment Guy Toublanc pour l'accueil sympathique réservé à mon premier article sur la HP48 et le petit programme de conversion des caractères accentués. Le programme tellement plus professionnel de Guy que mon propre effort modeste a suscité ne fait qu'accentuer le gouffre en matière de programmation qui sépare les membres comme moi de spécialistes comme Guy, Jacques et quelques autres. J'aimerais donc que l'avis que j'exprimerai ici soit pris comme émanant d'un membre qui n'a rien à envier à qui que ce soit sur le plan enthousiasme pour ces remarquables petites machines à nous, mais qui aimerait voir notre club se développer et pour ce faire ne pas hésiter à explorer toutes les voies possibles.

A mes yeux, c'est simple. De deux choses l'une : ou bien le Bureau modifie sensiblement sa politique éditoriale pour tenter d'attirer de nouvelles adhésions, ou bien il se résigne à demeurer sclérosé, avec tous les risques que cela comporte. J'en ai souvent parlé à Jacques, qui comprend parfaitement le problème, et aujourd'hui je m'adresse aussi à Guy (que je n'ai pas encore eu le plaisir de connaître) (*Remarque amicale : combien de fois vous ais-je encouragé à aller aux réunions ? Si vous aviez pris la peine d'y venir, vous auriez déjà rencontré Guy !*) pour lui poser la question suivante : Trouve-t-il normal de consacrer tant d'heures à écrire des programmes dont seuls les programmeurs chevronnés du Club peuvent suivre le raisonnement qui a présidé à leur élaboration, et que d'autres, j'en suis persuadé, ne savent même pas comment rentrer dans leur machine ?

Réponse : Tout d'abord, je voudrais faire quelques remarques à propos de cette fameuse "Politique éditoriale". En fait, nous devons vous avouer que nous (ainsi que les autres clubs d'ailleurs), n'avons jamais eu de Politique Éditoriale ! Comment pouvez vous en avoir une, lorsque c'est les adhérents eux-mêmes qui décident du sujet de l'article qu'ils vont écrire ? Rappelez vous que nous ne fonctionnons pas comme une rédaction, nous ne faisons que mettre en page ce que nous recevons. Le problème est qu'il y a quelques années, une grande partie des articles étaient écrits par

des personnes n'ayant pas obligatoirement un très grand niveau, et les débutants y trouvaient leur compte. Quant au sujet des articles que nous publions, Guy ou moi, il ne faut pas se leurrer. Nous sommes comme les autres adhérents. Nous ne programmons pas dans le but de faire des articles, mais avant tout pour résoudre des besoins personnels. Une fois ces programmes écrits, nous les publions dans JPC. Le problème est que nos programmes deviennent de plus en plus complexes (donc de plus en plus difficiles à expliquer aux débutants), que cela est de plus en plus long à développer, et que nous n'avons plus forcément le temps de faire des articles qui vous seraient vraiment utiles. Nous sommes les premiers à le regretter, mais les journées ne font que 24 heures... La seule chose que nous pouvons espérer est qu'une partie des adhérents (ne serait-ce que ceux qui sont plus jeunes que vous !) donnent eux même une partie de leur temps pour écrire des articles d'un niveau inférieur au nôtre.

Je sais qu'une revue comme *The HP Palmtop Paper* dispose de moyens sans commune mesure avec les nôtres, mais n'oublions pas qu'elle est avant tout une entreprise commerciale et qu'il faut donc qu'elle s'avère rentable... Ce qui a conduit à opter pour une ligne éditoriale susceptible d'attirer un maximum d'abonnements. Or tous ceux qui connaissent cette revue savent qu'elle met l'accent sur les conseils, les astuces, et sur des applications d'une trivialité qui fait parfois honte quand on pense au potentiel tout à fait exceptionnel de nos machines. Par rapport à notre journal, cette revue américaine se trouve aux antipodes, bien sûr. Mais sans aller jusque là, il me semble que, si nous voulons attirer toute une autre catégorie d'adhérents, nous nous devons de consacrer une partie substantielle de notre journal à des articles à caractère didactique. Une série d'articles expliquant les mystères de tous ces programmes (avec leurs pages entières de codes), serait vraiment le bien-venu...

Réponse : Il est évident que nous ne pouvons pas nous comparer au Palmtop Paper. Le problème n'est pas qu'ils soient une entreprise "commerciale", mais qu'ils ont une audience 200 fois plus importante que la notre. Ce qui implique qu'ils ont un potentiel d'auteurs actifs 200 fois supérieur au notre. (remarquez bien que j'ai parlé de "potentiel", ce qui n'a rien à voir à la réalité...). Nous sommes donc sûrs que si nous voulions "clôner" le Palmtop Paper, nous serions sûrs de perdre, car comme vous l'avez dit, nous ne sommes pas de taille. Par contre, le fait que de nombreuses personnes soient en même temps membres de PPC Paris et abonnés au Palmtop Paper indique que nous assurons une certaine complémentarité. Le fait que nous ne travaillions pas sous la contrainte de "l'audience", nous permet souvent de faire des articles plus originaux. Et si il s'avère que peu d'adhérents soient intéressés par cet article, ce n'est

pas une catastrophe, c'était un risque à prendre... Mais bien sûr, nous ne sommes absolument pas hostiles à la publication de trucs. La preuve est que Guy a créé une rubrique "trucs et astuces" pour HP48, et qu'il peut très bien y en avoir une pour les palmtops. Il ne tient qu'à vous de l'alimenter !.

Il est évident qu'on peut s'acheter plein de bouquins. Oui, mais ce serait autrement plus sympathique de pouvoir lire des articles expliquant, pas à pas, le pourquoi de chaque ligne de programmes-exemples intéressant nos machines plus particulièrement... Des articles écrits par des membres comme moi - dont les connaissances en informatique sont tellement plus modestes - sur des sujets d'ordre général ou concernant des matériels spécifiques (comme le présent article justement) - ne suffisent pas. Cela aide, mais c'est tout. Bien entendu, je continuerai pour ma part d'offrir ma contribution, aussi modeste soit-elle, chaque fois que je le pourrai.

Réponse : Comme je l'ai dit plus haut, nous sommes tout à fait favorables à faire paraître ce genre d'articles, mais en ce qui nous concerne, le temps nous manque un peu... D'autant plus que ce type d'article est beaucoup plus long à rédiger qu'un article "normal" (c'est déjà très long de commenter "normalement" un source...). Mais ce type d'articles peut tout à fait basé autour d'un programme peu compliqué, donc écrit par tout programmeur.

Voilà, c'était pour moi l'occasion de dire ce que je pense... Après tout, notre revue doit pouvoir servir aussi de tremplin pour faire connaître le point de vue de tous les membres, surtout lorsqu'il y a des avis divergeants. The ball's in your camp now, my friends!

... et carte FAX

Ayant dit ce que je pense, passons au véritable but de ce deuxième article, qui est tout autre puisqu'il concerne le HP100 et mes récentes expériences en matière de cartes fax/modem pour cette machine.

Ceux des lecteurs qui ont eu l'occasion de lire mon précédent article se souviendront que je suis traducteur technique et assez souvent en déplacement à l'étranger, mais que je dois néanmoins être en mesure parfois de répondre à des appels urgents émanant de mon bureau. Or j'avais constaté que je ne pouvais pas toujours avoir facilement accès à un télécopieur, ce qui occasionnait parfois des délais gênants. J'ai donc eu l'idée d'acquérir une carte fax/modem pour le HP100.

Impressionné sans doute par une publicité savante, j'ai décidé d'abord d'essayer la carte EGO fabriquée par *Chaplet Systems*, donnée comme étant capable

d'envoyer et de recevoir des fax à 14400 baud sur le HP100. Décision désastreuse ! A part que cette carte acceptait parfois d'envoyer un fax, rien ne marchait. Après de nombreux échanges de fax avec *Chaplet* aux USA, et après avoir même envoyé ma machine chez eux avec leur carte (sur leur propre demande et à leurs frais, bien entendu), toujours sans résultat, *Chaplet* m'a retourné la machine avec une lettre d'excuses et m'a remboursé intégralement.

Frustré mais nullement découragé, j'ai décidé ensuite de tenter ma chance avec *EXP Computers, Inc.*, dont j'avais remarqué entretemps l'annonce que faisait paraître cette société dans *The HP Palmtop Paper*.

Bref, j'ai commandé cette carte (la *ThinFax Basic-LX*) chez *EduCALC* pour la somme de \$219.95 (plus frais d'envoi). Très simple à installer, j'ai eu l'agréable surprise de constater que toutes les fonctions étaient bel et bien présentes et qu'elles marchaient du premier coup.

Avant de donner des précisions sur le fonctionnement de cette carte, je voudrais ajouter que celle-ci présente trois particularités auxquelles la publicité du fabricant ne faisait aucune allusion : premièrement, la carte est "system manager compliant" et toutes ses fonctions sont gérées par menus déroulants; deuxièmement, moyennant une modification minime du CONFIG.SYS, elle fonctionne impeccablement dans n'importe quel PC muni d'un slot PCMCIA au Standard 2.0 et équipé de ses propres drivers et d'un logiciel de communication fax/modem adéquat (puisque le logiciel *MiniFax* résidant dans la carte ne fonctionne qu'avec le HP100); troisièmement, enfin, cette carte de type II présente les caractéristiques physiques d'une carte de type I, son épaisseur étant de 3,3mm au lieu de 5mm.

Le câble adaptateur fourni avec la carte porte un miniconnecteur qui se branche sur la tranche extérieure de la carte, l'autre extrémité du câble étant reliée à un adaptateur téléphonique femelle RJ10. Le brochage des fils à l'intérieur de ce câble étant conforme aux normes américaines, il y a lieu de se procurer (chez Auchan, dans mon cas) une prise téléphonique PTT de type "convertisseur USA/France", dans laquelle viendra se ficher l'une des extrémités d'un câble prolongateur fourni également avec la carte.

Voyons maintenant de plus près le comportement de la carte *Basic-LX* dans le HP100. J'avais tout de même décelé un ou deux petits inconvénients. Par exemple, bien que les télécopies reçues avec la carte pouvaient être imprimées sur une imprimante munie d'un port série, ça ne marchait pas sur imprimante parallèle par l'intermédiaire d'un convertisseur

série/parallèle (dans mon cas le BSE). Il s'est avéré que c'était parceque la carte ne s'occupait pas de savoir quels étaient les paramètres "imprimante" choisis par l'utilisateur et supposait une vitesse de 9600 baud, alors que le BSE ne fonctionne qu'à 4800 baud. L'autre problème était que les caractères accentués n'étaient pas interprétés correctement.

Lorsque j'ai fait un compte-rendu de ces problèmes à EXP Tech Support, la réaction a été immédiate : j'ai été prié d'appeler le numéro correspondant à leur Bulletin Board et de charger un certain fichier de mise à jour qui corrigerait les défauts signalés. J'ai donc transféré le fichier en question (un .EXE) et l'ai ensuite exécuté. Cette dernière opération est totalement transparente et fournit, en temps réel, un compte-rendu de ce qui se passe pendant l'opération "upgrade" ("Erasing chip xxxx... "Reprogramming card..." etc.). Cette première mise à jour a servi à corriger les deux petits défauts signalés, et il devenait désormais possible d'utiliser les caractères accentués en minuscules. Quelques jours plus tard, cependant, j'ai signalé à EXP que j'avais remarqué deux autres inconvénients, plus un bug.

Premièrement, le wordwrap automatique sur les fichiers ASCII générés par les applications internes du HP100 (MEMO, etc.) n'était reconnu par aucun télécopieur recevant un fax du HP100 si le fichier en question avait été sauvegardé dans MEMO, le texte étant simplement rendu "au kilomètre" avec retour de chariot automatique à la 80ème colonne du télécopieur, ce qui rendait le fax inutilisable par le destinataire.

Deuxièmement, si on voulait envoyer une page de couverture avant un fax, la date figurant sur celle-ci était toujours au format américain (ce qui peut prêter à confusion en Europe, ai-je fait remarquer à EXP Tech Support...)

Quant au bug, il n'était toujours pas possible, dans le cas d'un fax reçu, d'imprimer celui-ci à partir du menu LOG, qui pourtant offrait cette option (il n'étant possible de réaliser cette opération qu'à partir du menu principal \UTILS).

Comme d'habitude, la réaction ne s'est pas fait attendre. J'ai reçu d'abord un message me disant que le software engineer n'arrivait pas à reproduire le problème du wordwrap, mais que les deux autres défauts signalés étaient en cours d'étude. Je m'apprétais quelques jours plus tard à leur faxer un exemple du problème en question, lorsque j'ai reçu un message de Tech Support m'informant que le problème avait été en effet identifié et que "We're working on it". A mon retour d'un voyage à l'étranger d'une dizaine de jours, j'ai trouvé un fax me

remerciant et me priant de "download" un deuxième fichier de mise à jour qui reprogrammerait la carte... Ce que j'ai fait, et tout est rentré dans l'ordre depuis !!

On pourra reprocher à la petite notice (en anglais) d'être un peu trop succinte, mais les instructions sont claires. La première chose à faire est de copier le fichier D:\AUTOEXEC.BAT dans le drive C:\ en supprimant le "rem" qui précède la ligne D:\BIN\CIC100/GEN1. Ensuite on insère la carte et on fait un "reboot" (Ctrl-Alt-Del) afin que la carte puisse être reconnue grâce au "driver" qu'on vient d'ajouter à l'AUTOEXEC.BAT. La suite des opérations correspond à celle - classique - que doit exécuter l'utilisateur pour installer une nouvelle application System Manager (t&...], [tF2], etc.), en se conformant aux instructions de la notice. Ce qui aura pour effet de créer l'icône *MiniFax* dans System Manager.

Comme je l'ai dit, tout se passe ensuite avec des menus déroulants. Donc une fois branchée sur la prise murale du téléphone, on sélectionne l'icône *MiniFax* dans SysMgr et on attend la suite des évènements.

J'ai été particulièrement impressionné par la vitesse à laquelle les fichiers (obligatoirement du type ASCII) sont envoyés à 9600 baud. Mais même à 4800 baud, ça ne traîne pas en mode réception non plus. Les télécopies reçues peuvent être visionnées en vraie grandeur ou en 'zoom' et sont parfaitement claires; elles peuvent être déroulées (scroll) latéralement ou verticalement au moyen des flèches (la vitesse de déroulement a été augmentée grâce aux mises à jour successives). Comme j'ai expliqué plus haut, elles peuvent être imprimées sur imprimante série ou parallèle. J'ai omis de mentionner que le logiciel *MiniFax* - très compact - est capable de puiser les noms et numéros de téléphone dans l'application Phonebook lorsqu'il s'agit de remplir les champs "destinataire" correspondants avant l'envoi d'un fax.

En mode "data modem", les transferts se font à 2400 baud - que ce soit avec le HP100 ou avec un PC (dans mon cas, un IBM ThinkPad comportant un slot PCMCIA).

En conclusion, je ne peux que recommander cette carte fax/modem à tous ceux qui, comme moi, sont susceptibles d'avoir besoin de recevoir ou d'envoyer des télécopies ne dépassant pas deux ou trois pages lorsqu'ils se trouvent en déplacement. En dehors de sa fiabilité et de son prix tout à fait raisonnable, le service après-vente m'a paru irréprochable.

Robert Amram (390)

BUDDY ET HELV100 : PAS COPAINS !

En addition à mon article consacré à 100BUDDY, paru dans le dernier numéro, il est à signaler que certaines fonctions du programme peuvent ne plus fonctionner lorsque le programme de changement de fontes d'affichage HELV100 est chargé. Cela est dû au fait qu'afin de fonctionner, Buddy décode le contenu de l'écran afin de récupérer les informations qui l'intéressent (noms des fichiers ou de l'application active...). Or le problème est qu'il ne peut reconnaître les caractères que lorsque ils sont identiques à ceux qui sont inclus dans la Rom. Jeffrey Mattox (l'auteur du programme) espère pouvoir corriger ce problème dans une prochaine version.

Le problème est que lorsque j'ai préparé l'article, je n'avais pas prêté attention à cette information. Il est donc probable (je ne m'en rapelle plus) que j'aie effectué certains tests alors que HELV100 était actif. Ce qui a pu fausser certains résultats. N'ayant pas eu le temps dernièrement de faire une vérification complète, je ne peux pas vous indiquer les fonctions concernées. Cependant, je n'ai pas constaté de différences sur les fonctions principales.

D'autre part, une petite erreur de mise en page s'est glissée dans l'article, page 28. En effet, afin de désactiver et réactiver le programme, il ne vous faudra pas taper les cabalistiques "ςΓυσμ#ς-#" et "ςΓυσμ#ς+#", comme indiqué en page 28, mais tout simplement [Ctrl][-] et [Ctrl][+] ! Petit cryptage involontaire dû à une impression en police "mathématique" au lieu de "line printer"....

Jacques Belin (123)

CA BOUGE SUR LES PALMTOPS !

Bien que nos palmtops possèdent des capacités graphiques intéressantes, très peu de programmeurs se sont intéressées à les montrer.

Le but de ce gros article (dont la deuxième partie paraîtra des le prochain numéro) a donc pour but de vous montrer que les HP95 et HP100 peuvent tout à fait afficher des animations parfaitement fluides et crédibles.

Attention, quand je parle ici d'*animations*, il ne s'agit pas d'images calculées en temps réel (comme pour les "demos" écrites par certains programmeurs), mais de restitution d'images stockées dans un fichier. Cela est peut être moins complexe, mais nécessaire tant que les ordinateurs ne seront pas capables de calculer des images hyper-réalistes (en ray-tracing, par exemple) en temps réel.

De plus, si cet article a pour but de vous montrer comment créer des animations en image de synthèse, rien ne vous empêche de remplacer ce type d'animation par des images vidéo, à la condition que vous ayez la possibilité d'extraire ces images (à partir d'un CD-Rom, par exemple) en une succession de fichiers graphiques indépendants.

Un autre type d'animation est aussi possible, il s'agit du Morphing. Pour cela, il suffit d'avoir deux fichiers graphiques (au format de l'écran du HP95) et d'utiliser un des nombreux utilitaires Freeware ou Shareware permettant de calculer les images intermédiaires.

Cependant, les possibilités de nos palmtops étant ce qu'elles sont, il est nécessaire de se fixer certaines règles, et de penser à certaines contraintes :

- Afin de conserver la compatibilité entre les palmtops, les images seront affichées suivant le mode graphique du HP95 (240x128 pixels).
- Nos palmtops n'étant pas ce qui se fait de plus rapide, il faut essayer d'obtenir la plus grande vitesse d'affichage.
- Enfin, il est très souhaitable que les fichiers d'animations soit les plus petits possibles...

Afin de tenir compte de ces différents points, j'ai créé pour l'occasion un nouveau type de fichier (portant l'extension "A95"), possédant la particularité de stocker les images en suivant différents modes de compression, que le créateur de l'animation pourra sélectionner suivant le style des images et le type d'animation :

- Stockage de chaque image sans compression.
- Compression de chaque image en PCX.
- Compression plus de chaque image, mais plus intense, en utilisant des adaptations (pour le monochrome) des algorithmes GIF ou JPEG.
- Utilisation des algorithmes de création des fichiers de type FLI ou MPEG, ne stockant que les portions changeantes entre deux images consécutives.

Pourquoi utiliser différents modes de compression ?

Tout d'abord, parce que suivant le type de l'image ou de l'animation, les résultats de la compression peuvent être contraire à l'effet désiré. Par exemple, il est très fréquent que des fichiers utilisant le mode de compression RLE (fichiers PCX) soient plus beaucoup plus gros que l'image non compressée ! Cela est dû au fait qu'au delà d'une certaine complexité de l'image, il y a plus d'octets de contrôle que de d'information utiles... De plus, le passage d'une zone de couleur uniforme en monochrome (tramé) crée une zone contenant des points placés de façon aléatoire, rendant la compression plus difficile. En ce qui concerne les animations, les algorithmes de type MPEG sont très efficaces lorsque seulement une petite partie de l'image change, mais sont quasiment inefficaces lorsque, par exemple, l'animation contient un traveling dans un décor complexe.

D'autre part, tous les algorithmes de compression ne sont pas égaux en vitesse. Par exemple, si les algorithmes de type GIF ou JPEG sont relativement puissants, ils sont assez lents. Et si les algorithmes de type MPEG sont très efficaces lorsque une très petite partie de l'écran change, ils le sont moins lorsque toute l'image est en mouvement. Bien sûr, une image non compressée permet théoriquement l'affichage le plus rapide, mais il faut aussi considérer le fait qu'une grande partie du temps est passé à lire le fichier à partir du disque, et qu'il faut donc essayer d'avoir le minimum d'octets à lire.

Il s'agira donc au créateur de l'animation de déterminer, au cas par cas, quel est le meilleur mode de compression à utiliser.

Actuellement, seuls deux modes de compression sont implantés : "sans compression" et PCX. Faute de temps, je n'ai pas encore commencé à travailler sur les autres modes.

En ce qui concerne les performances que l'on peut attendre de ce type de programme, elles sont bien évidemment dépendantes du matériel utilisé (HP95 ou HP100, type de carte, présence ou non de Stack...). A titre indicatif, une animation compressée en PCX peut être affichée à un rythme d'environ 15 images par seconde sur le HP95, et une animation non

compressée peut atteindre les 50 images par seconde sur un HP100 équipé d'une carte SRAM non stackée ! Bien sûr, afin de conserver un fichier de taille raisonnable, il est préférable de ne pas utiliser une vitesse d'affichage trop importante. Mes essais ont montré qu'une quinzaine d'images par seconde est une valeur acceptable. Ce qui permet aussi d'être sûr que la même animation fonctionnera de façon identique sur les deux palmtops.

Afin de servir d'exemple pour cet article, nous allons créer une petite animation en image de synthèse, consistant à afficher un pendule passant devant un logo PPC. Vous pouvez en voir quelques extraits dans ces pages.

Ne soyez pas trop effrayés par la complexité de la méthode utilisée pour créer cette animation, c'est intentionnellement que je l'ai utilisée, afin de décrire en détail toutes les étapes. En fait, en utilisant certains autres programmes annexes (modeleurs, avec pour certains d'entre eux un module d'animation), une grande partie du travail peut être simplifiée. Bien sûr, certains logiciels tels que 3D Studio intègrent toutes ces fonctions, mais un avantage avec ma méthode est que nous pouvons utiliser ici des programmes Freeware ou Shareware...

Les Outils

Les outils qui seront utilisés se répartissent en deux groupes :

- Des outils déjà existants, dont le but final est d'obtenir une suite de fichiers PCX monochromes, représentant l'animation.
- Des outils spécifiquement créés pour l'occasion, permettant, à partir de ces fichiers PCX, de créer les fichiers A95, puis (bien sûr) les afficher sur l'écran des palmtops.

La plupart des programmes de la première catégorie sont disponibles chez *DP Tool Club* (voir les publicités dans les revues informatiques pour plus de détails). Si vous possédez un modem, vous pouvez aussi les trouver sur la plupart des BBS.

La création des images

Dans l'exemple qui nous intéresse ici, notre principal outil sera le logiciel de création d'images de synthèse. De nombreux programmes de ce types existent dans le monde du Freeware (VIVID, POLYRAY...), mais le plus connu est probablement Persistence Of Vision (appelé tout simplement POV par les intimes). Comme tous ses concurrents, il est capable de générer des images hyper réalistes (grâce au *Ray Tracing*), en utilisant un fichier ASCII contenant des *scripts* écrits dans un langage spécifique. De nombreux autres

programmes ont été créés autour de POV, tels que MORAY ou PV3D, qui sont des modeleurs (des interfaces graphiques en 3D) permettant de créer automatiquement les scripts nécessaires.

Il est à noter que depuis quelques mois un logiciel Français connaît un certain succès. Il s'agit de LUXART. En plus des possibilités déjà offertes par POV, il possède quelques nouveautés, telles que la sélection de la profondeur de champ ou un meilleur contrôle de l'illumination autour des objets transparents. Cependant, n'ayant pas eu la possibilité de l'essayer, je ne peux pas vous dire si il est adapté à notre utilisation. En effet, il est utilisé par l'intermédiaire d'une interface graphique (incluant un modeleur), et je ne sais pas si il est capable de générer et lire des scripts, pour une utilisation en mode "ligne de commande".

En ce qui concerne l'animation proprement dite, si vous n'avez pas de logiciel incluant cette fonctionnalité, la meilleure alternative est probablement RTAG (shareware), qui est un langage de programmation (très proche du C) vous permettant de créer un fichier "batch" contenant tous les ordres permettant de créer les images (définition des positions intermédiaires, commandes de lancement du raytracer...).

Comme je l'ai dit plus haut, il doit être possible de créer des animations à partir d'images vidéo. Je ne peux malheureusement pas vous donner de noms de programmes de conversion (toute information est la bienvenue), mais il faudra prendre en compte une étape supplémentaire, consistant à recadrer l'image (en supprimant les parties haute et basse) en raison du format "cineamscope" de l'écran des palmtops. Là aussi, il vous faudra trouver un programme travaillant en mode "batch".

Certains logiciels (tels que POV) ne créant que des fichiers graphiques au format TGA en 16 millions de couleurs, il vous faudra ensuite convertir l'image en PCX monochrome inversée (afin de tenir compte de l'affichage noir sur blanc de l'écran des palmtops). Mais, si certains programmes, tels que *Graphic Workshop*, font cela très bien, ils sont peu utilisables ici, car dès que l'on a un grand nombre d'images à convertir, le travail devient vite exténuant. Il faut donc utiliser des utilitaires travaillant en mode "ligne de commande", afin de pouvoir les inclure dans des fichiers "batch".

Malheureusement, je n'ai pas trouvé de programme pouvant effectuer à lui seul la totalité des conversions nécessaires. Il faudra donc passer par deux étapes :

- Tout d'abord, conversion du TGA 16 millions de couleurs en GIF 256 couleurs (étape inutile si vous utilisez Luxart). Pour cela, vous pouvez utiliser, par exemple, DTA (de David Mason) qui permet en outre de créer des fichiers d'animation au format FLI. Idéal pour contrôler la qualité de l'animation sur son PC...

- Ensuite, il vous faudra convertir ce format GIF couleur en PCX monochrome inversé. Malheureusement, je n'ai trouvé qu'un seul utilitaire permettant d'arriver à ce résultat. Il s'agit de GIFPCX. Si il ne semble pas être distribué par DP Tool, il doit être relativement facile à trouver autre part (BBS, CD-ROM, autres vendeurs de Sharewares... Je n'ai pas vérifié). Dans tous les cas, il est présent sur la disquette de Jan/Fev 1993 du *Palmtop Paper*.

Du PCX à l'animation

Une fois que nous aurons obtenu une série de fichiers PCX, il ne nous restera plus qu'à utiliser les utilitaires créés pour l'occasion :

- MAKEA95, qui fusionne les fichiers PCX en un seul fichier A95 (suivant le mode de compression désiré) et sélectionne divers paramètres (type d'animation, vitesse d'affichage idéale...).

- PLAYA95, qui est le programme d'affichage proprement dit, commun aux HP95 et HP100.

La création des images

Une fois cette (longue !) introduction effectuée, il est temps maintenant de passer aux détails pratiques, en commençant par la création des fichiers PCX.

Les outils utilisés dans cet article seront : POV (version 2.x), RTAG, DTA et GIFPCX.

Utilisation de Persistence of Vision

Il m'est impossible de décrire ici en détail la façon d'utiliser POV. D'une part parce que cela serait beaucoup trop long, et d'autre part parce que je n'ai pas encore une expérience suffisante de ce logiciel pour vous donner des explications détaillées. Je ne vous donnerais donc ici que des informations ayant trait à l'utilisation qui nous intéresse aujourd'hui. A vous de consulter la documentation (ainsi que les nombreux exemples) accompagnant le logiciel.

Tout d'abord, avant de parler de l'utilisation proprement dite, il est peut être nécessaire de donner quelques conseils concernant les images elles mêmes. Il est à noter que la plupart sont aussi valides pour la création d'images fixes pour le HP95 :

- Pensez toujours que l'image sera convertie en noir et blanc tramée. Il vous faudra donc faire très attention aux couleurs des objets, afin qu'ils soient suffisamment contrastés, pour apparaître distinctement en noir et blanc.

- Si vous utilisez Luxart, désactivez les options telles que la profondeur de champ. Ce genre d'effet sera invisible avec la résolution que nous utilisons (240x128).

- Dans le même ordre d'idée, évitez d'utiliser des textures très complexes. Dans le meilleur des cas, elle seront invisibles, dans le pire des cas, ce sera l'objet qui ne sera plus identifiable !

- Evitez d'utiliser des arrière plans très complexes, par exemple un ciel nuageux. C'est peut être plus beau, même converti en noir et blanc tramé, mais ce sera très gourmand en occupation disque après conversion en PCX. Essayez donc d'utiliser des fonds blancs ou noirs.

En résumé, il est inutile d'essayer de faire des images trop "léchées". Vous perdrez votre temps, car tous les détails seront perdus au cours de la conversion en PCX. Cependant, il sera au contraire nécessaire dans certains cas de porter une plus grande attention lors de l'utilisation de certains effets. Par exemple, dans le cas de notre animation, la grande difficulté n'a pas été de modéliser le balancement de la sphère, mais de paramétrer ses propriétés optiques (transparence, réflexion, réfraction...) afin qu'elle conserve un certain "volume" et n'apparaisse pas comme un simple disque. Si ce paramétrage est très simple lorsque nous désirons créer une image en 256 couleurs (ou même en 16 niveaux de gris) cela est beaucoup plus difficile lorsque l'image doit être tramée...

En dehors de ces contraintes, il n'existe qu'une seule différence dans le paramétrage de POV par rapport à une image pour PC. Il s'agit de la prise en compte du format "cinémascope" de l'écran des palmtops.

Cette information est prise en compte dans le module de définition de la caméra, où il est nécessaire de remplacer le premier paramètre de `right` (qui donne le format de l'image) par 1.875 (au lieu de 1.33). Cette nouvelle valeur tient compte du changement du rapport largeur/hauteur de l'écran des palmtops, mais aussi du fait que les pixels ne sont pas carrés, comme sur un écran VGA. Il vous faudra donc paramétrer le module `camera` comme ceci :

```
camera
{
  location <x, y, z>
  up <0.0, 1.0, 0.0>
  right <1.875, 0.0, 0.0>
  look_at <x1, y1, z1>
}
```

Il est à noter que si cette valeur est exacte pour le HP95, elle est légèrement erronée pour le HP100, car son écran est un peu plus large. Mais la différence étant minime, il n'est pas nécessaire d'en tenir compte.

Si POV n'a été conçu à l'origine que pour générer des images fixes, il est toutefois possible de détourner certaines de ses possibilités originelles, afin qu'il puisse être utilisé par un programme extérieur.

La première modification à apporter consiste à remplacer toutes les valeurs concernant un élément mobile (position, orientation...) par une constante définie dans un ordre "declare". Par exemple si nous avons la ligne :

```
rotate <0.0, 0.0, 15>
```

et que nous voulons définir un mouvement suivant l'axe z (celui pour lequel il y a actuellement la valeur "15"), il suffira d'écrire :

```
rotate <0.0, 0.0, angle>
```

après avoir placé, au début du programme, la ligne suivante :

```
#declare angle = 15
```

Mais ce n'est pas suffisant pour créer nos animations, puisque il faudrait réécrire tout le fichier entre chaque image. La bonne méthode est donc d'isoler toutes ces "pseudo-variables" dans un fichier séparé, afin de n'avoir à réécrire que ce qui change. Pour cela, il nous suffit d'utiliser la directive `#include`, en replaçant toutes les directives `#declare` utilisées par l'animation par un seul :

```
#include "pendul.inc"
```

et de les placer dans le fichier "pendul.inc".

C'est ce fichier, que nous allons créer dans le chapitre suivant.

Mais avant cela, je vous invite à examiner le fichier `PENDUL.POV`, qui contient toutes les informations permettant de créer l'image de base.

Sa structure est simple, puis qu'il ne comprend que :

- Un pendule, composé d'un balancier (en bois) soutenant une boule de verre.
- Derrière celui-ci, un "mur" vertical sur lequel est collé une image, ici un logo "PPC Paris". Il est bien sûr possible de remplacer ce logo par tout autre fichier graphique.

- Pour éclairer ces éléments, une source de lumière blanche placée derrière nous (à gauche et légèrement en hauteur).
- Et pour visualiser le tout, une caméra.

Animons ces images !

Si POV est le programme principal pour créer les images, RTAG en est le maître, puisque c'est lui qui définira tous les variables nécessaires à POV pour créer les animations proprement dites.

Son but est de créer un fichier batch contenant toute les commandes à effectuer pour créer chaque image PCX. Cela peut être aussi bien l'initialisation des fichiers INC que les commandes de lancement des utilitaires (POV, conversions de format...) proprement dits.

La syntaxe des fichiers utilisés par RTAG est quasiment identique à celle du langage C (boucles, tests, tableaux, fonctions trigonométriques...). Ont été ajoutées d'autres fonctions, relative à la génération du fichier batch (numérotation des images...) ou à la génération des images intermédiaires entre deux points (par interpolation linéaire ou par des "splines").

La première étape consistera donc à créer une boucle (exécutée autant de fois qu'il y a d'images). Dans celle ci, nous calculerons les paramètres nécessaires au fichier INC (qui sera recréé, entre chaque génération d'image, par le fichier batch, grâce à la commande Dos "ECHO").

Pour notre exemple, nous générerons 18 images, afin d'avoir un cycle total d'environ trois secondes. Ce chiffre devrait être suffisant pour créer une animation suffisamment fluide, sans être trop gourmand en espace disque. Notez aussi qu'elles ne représenteront que le parcours "aller" du balancier (le retour étant géré par le programme d'affichage).

Le balancement de notre pendule sera simulé tout simplement par un mouvement de type sinusoidal, avec une amplitude de +/- 30 degrés. Le résultat du calcul étant transmis à la variable "angle" de notre fichier INC.

La génération de l'image sera effectuée par un appel à un autre batch comprenant le lancement de POV ainsi que les commandes de conversion de l'image en GIF et PCX. Voir le chapitre suivant pour plus de détails.

Suivant les conventions de RTAG, le nom des fichiers générés seront composés du nom de l'animation, suivis du numéro de l'image sur deux ou trois digits (Attention à ne pas dépasser la limite Dos de 8

caractères). Pour notre exemple, cela donnera, pour les fichiers PCX : PENDUL01.PCX, PENDUL02.PCX, PENDUL03.PCX....

Après l'exécution de la boucle principale, nous n'aurons plus qu'à déclarer les commandes permettant de créer les fichiers d'animations proprement dits.

Vous pouvez constater que je crée en tout premier lieu un fichier PENDUL.FLI (à partir des images GIF). Il me permettra de visualiser (avec un des nombreux utilitaires adéquats) la qualité de l'animation, sans avoir à transférer un fichier A95 sur le palmtop. (Il faudra que je me décide à faire une adaptation de PLAYA95 fonctionnant sur le PC...).

Ensuite, je crée autant de fichiers A95 qu'il existe de modes de compression. Cela me permettra de vérifier immédiatement lequel génère le plus petit fichier A95.

En définitive, voici un extrait du fichier batch que nous obtiendrons après lecture du fichier PENDUL.RTA par RTAG (j'ai supprimé une grande partie des ordres intermédiaires et coupé les deux dernière lignes, pour des raisons de mise ne page) :

```

echo #declare angle = -30 > pendul.inc
call 95render pendul pendul01
echo #declare angle = -29.48919299 > pendul.inc
call 95render pendul pendul02
echo #declare angle = -27.97416688 > pendul.inc
call 95render pendul pendul03
.
.
.
echo #declare angle = 29.48919299 > pendul.inc
call 95render pendul pendul17
echo #declare angle = 30 > pendul.inc
call 95render pendul pendul18
DTA \images\pendul*.gif /O\images\pendul95.FLI /S7
makea95 -a2 -c0 \images\pendul_0.a95
\images\pendul*.pcx
makea95 -a2 -c1 \images\pendul_1.a95
\images\pendul*.pcx

```

Du POV au PCX

Afin de simplifier l'écriture des fichiers RTA, j'utilise quelques fichiers batch se chargeant de toutes les opération de création et de conversion des fichiers graphiques.

Le premier (appelé POV240) permet de lancer POV, avec certains paramètres par défauts : image en 240x128, affichage de l'image, attente d'une pression de touche après exécution, pas de génération de

fichier. Certains de ces paramètres seront annulés dans le cadre du batch suivant. Ceci me permet de conserver le même fichier batch pour la génération des animations et pour la simple visualisation des images au cours de leur conception.

Dans ce cas, il peut être appelé comme ceci :

```
pov240 pendul.pov
```

En voici le texte (coupé sur deux lignes pour des raisons de mise en page) :

```
\povray\bin\povray +L\povray\include +i%1 +d -v
-f +p +x +w240 +h128 %2 %3 %4 %5 %6
```

Afin de générer les fichiers graphiques, j'utilise un autre batch (appelé 95RENDER.BAT), dont voici le texte :

```
call pov240 %1.pov -p +ft -oF:\%1.tga
dta F:\%1.tga /FG /O\images\%2.gif
gifpcx \images\%2.gif \images\%2.pcx /v /b+2
```

L'exécution de "95render pendul pendul01" créant les fichiers suivants :

```
F:\PENDUL.TGA
\IMAGES\PENDUL01.GIF
\IMAGES\PENDUL01.PCX
```

Le directory \IMAGES contient tous mes fichiers graphiques intermédiaires et mes animations.

Le disque F est mon "Ramdisk". J'ai placé le fichier TGA sur celui-ci pour deux raisons :

- Etant donné la taille de ce fichier (près de 100 Ko), je ne peux pas me permettre de conserver toutes les images de l'animation sous ce format, surtout si les animations sont longues. Je ne crée donc pas de fichiers PENDUL01.TGA, PENDUL02.TGA....

- Les accès au Ramdisk étant très rapides, cela permet de réduire sensiblement le temps total de création de l'animation.

Avant de terminer ce chapitre, je voudrais noter que si j'utilise GIFPCX, ce programme possède cependant quelques petits défauts :

- Lorsque l'image contient des zones de couleur blanche, quelques points parasites peuvent apparaître dans ces zones. Pour corriger cela, il suffit souvent d'augmenter légèrement le paramètre de luminosité, en ajoutant "/b+2" à la ligne de commande. Fait intéressant, la plupart de ces points restent fixes sur certaines animations (c'est à dire qu'ils sont reproduits à la même place sur plusieurs images

consécutives), même lors de grandes évolutions des images (travelings...).

- Plus gênant, la dernière ligne de l'image peut apparaître totalement noire, alors que toute la zone au dessus-de celle ci est blanche.

- Enfin, une série de points noirs peuvent apparaître sur l'avant dernière colonne, à droite de l'image.

Ceci peut malheureusement vous amener à retravailler manuellement les fichiers PCX à l'aide d'un programme de type PAINTBRUSH.

La création des animations

Une fois que les fichiers PCX sont créés, nous devons quitter à partir de maintenant les utilitaires classiques pour aborder les fichiers et programmes créés pour l'occasion.

Structure des fichiers A95

Le but des fichiers A95 ayant été décrit au début de cet article, je passerais tout de suite aux détails pratiques :

La structure générale d'un fichier A95 est la suivante :

Zone	Taille
Entête	128 octets
longueur image 1	2 octets
données image 1	n_1 octets
longueur image 1	2 octets
longueur image 2	2 octets
données image 2	n_2 octets
longueur image 2	2 octets
.	
.	
.	
longueur image z	2 octets
données image z	n_z octets
longueur image z	2 octets

Comme vous pouvez le constater, la longueur de chaque image est indiquée deux fois. Ceci afin de permettre la lecture du fichier (et donc l'affichage des images) à l'envers. En fait, la version actuelle de mon programme d'affichage n'utilise plus cette deuxième variable, mais je l'ai laissée au cas où elle pourrait être utile à un autre programmeur.

Quant à l'entête, il est défini comme ceci :

Zone	Taille
MAGIC	2 octets
Numéro de version	1 octet
Mode de compression	1 octet
Nombre d'images de l'animation	2 octets
Type d'animation	1 octet
Vitesse d'affichage	1 octet
(réservés pour un usage futur)	120 octets

La zone MAGIC contient une valeur constante (5435h), servant à contrôler que nous avons bien à faire à un fichier A95.

Le numéro de version a pour but d'indiquer un éventuel changement du format du fichier (et non l'ajout d'autres modes de compression). Actuellement, sa valeur est fixée à 1.

Les codes actuellement admis pour les modes de compression sont :

- 0 : Pas de compression.
- 1 : Compression en PCX.

Le type d'animation indique la façon dont sera effectué l'affichage :

- 0 : Un seul affichage de l'animation.
- 1 : Affichage de l'animation en boucle. Lorsque la dernière image est affichée, on reprend à partir de la première. L'arrêt est contrôlé par l'utilisateur.
- 2 : Affichage "Ping-Pong". Lorsque la dernière image est affichée, on reprend l'animation à l'envers, jusqu'à la première image, puis on reprend dans le sens normal. L'arrêt est contrôlé par l'utilisateur. Notez que dans le cas des fichiers d'animations contenant des images indépendantes (non compressées, PCX, GIF...), il n'est pas nécessaire de stocker un aller-retour complet. Le programme gèrera lui même le changement de sens. Par contre, dans les algorithmes de type FLI ou MPEG ne sont pas réversibles. Il faudra donc stocker l'aller-retour complet, et spécifier une animation en boucle.

La vitesse d'affichage indique la valeur par défaut (idéale) de l'intervalle de temps entre deux images, exprimée en unités de 3.4 ms. La valeur par défaut est 24, correspondant à un rythme d'environ 12 images par seconde.

En ce qui concerne les zones de données de chaque image, elles ne contiennent actuellement que des données "utiles" (les octets, compressés ou non, à afficher). Mais rien n'empêche que ces zones contiennent des octets de contrôle supplémentaires, dans le cas de l'utilisation d'autres modes de compression.

Le programme MAKEA95

Ce programme a pour but de créer, à partir d'une suite de fichiers graphiques, un fichier A95.

Sa syntaxe est la suivante :

```
makea95 [-a type ] [-c mode] [-s speed] anim_file
                                             files [files...]
```

Avec :

- type : Type d'animation. 0, 1 ou 2 actuellement.
- mode : Mode de compression. 0, 1.
- speed : Vitesse idéale.
- anim_file : Nom du fichier A95.
- files [files] : Nom des fichiers graphiques en entrée. Les "wildcards" sont bien entendu acceptés.

Actuellement, le programme n'accepte en entrée que des fichiers PCX, mais rien n'interdit que dans le futur d'autres types de fichiers soient acceptés.

En ce qui concerne la structure de la version actuelle de ce programme, je ne m'étendrais pas dessus, car elle est relativement simple (une grande partie de son apparente complexité est due uniquement à la présence de nombreux tests de détection d'erreurs de lectures/ecritures). Le principe général est de créer l'entête, puis, pour lire successivement tous les fichiers PCX, les décompresser si l'on a choisi le mode "0", puis les écrire dans le fichier A95. En fin d'exécution, on affiche un petit rapport indiquant le nombre d'images lues et les différentes options choisies.

Ceci termine l'article de ce mois-ci. Dans le prochain numéro, je vous présenterai le programme d'affichage.

En attendant, n'oubliez pas que si j'ai fait un article aussi détaillé, c'est pour deux raisons :

- D'une part, je suis tout à fait disposé de faire de ces programmes une oeuvre collective. Toutes les personnes souhaitant y participer (en ajoutant de nouveaux modes de compression, par exemple) sont donc les bienvenues. N'ayant pas prévu de continuer à travailler sur ce sujet pendant les prochains mois, vous avez carte blanche. La seule contrainte étant qu'il est préférable que j'attribue moi-même les numéros de modes, afin d'éviter que plusieurs personnes utilisent un même numéro...
- D'autre part, le fait que j'ai créé ces programmes n'implique absolument pas que je possède les qualités artistiques suffisantes pour créer de belles animations... Alors, si vous vous sentez inspirés, n'hésitez pas à nous faire profiter de vos créations !

Jacques Belin (123)


```

/*****
/**
/** PENDUL.RTA
/** Pendule HP95 (sources RTAG)
/**
/** dernière modification : 12/01/94 Auteur : Jacques Belin
/** Création
/**
/*****
/**** Animation de pendule sur HP95 */
bfile "pendul";
// --- DECLARATION DES FICHIERS ---
// Le fichier batch sera appelé ORBIT.BAT
// --- VARIABLES GENERALES ---
var numframes = 18; // Nombre d'images de l'animation
var anglemax = 30; // Angle maximum que prendra le balancier
var timestep; // Intervale de temps
var currangle; // Angle courant
// --- INITIALISATIONS ---
var time = 0;
var timevar = 0;
timestep = 180 / (numframes-1);
for (time=0; time<numframes; time++)
{
nextframe;
currangle = -anglemax * cos(timevar); // selectionne le numéro de l'image suivante
// calcule l'angle courant
// Ajoute dans le fichier batch la commande de création du
// fichier .INC correspondant à cette image.
bwrite("recho #declare angle = 'currangle' > pendul.inc\n");
// Ajoute dans le fichier batch la commande de génération de l'image
bwrite("call 95render pendul'##'\n");
timevar += timestep; // Calcule le nouveau temps écoulé
}
epilog;
bwrite("DATA \\images\\pendul*.gif /O\\images\\pendul95.FLI /P /S7\n");
bwrite("makea95 -a2 -c0 \\images\\pendul_0.a95 \\images\\pendul*.pcx\n");
bwrite("makea95 -a2 -c1 \\images\\pendul_1.a95 \\images\\pendul*.pcx\n");

```

```

/*****
/**
/** MAKEA95.C
/** Construction des fichiers anim HP95
/**
/** Dernière modification : 27/05/94 Auteur : Jacques Belin
/** ajout de l'option "G"
/**
/*****
#include <stdio.h>
#include <dir.h>
#include <dos.h>
#include <fcntl.h>
#include <string.h>
#include "getopt.c"
/* Routine d'analyse des options (voir JPC 78) */
#define MAGIC 0x5435 /* code de reconnaissance des fichiers A95 */
#define NOCOMP 0 /* modes de compression supportés */
#define PCXCOMP 1 /* types d'animation */
#define ONETIME 0 /* Entête des fichiers PCX */
#define LOOP 1 /* numéro fabricant (toujours 0x0A) */
#define PINGPONG 2 /* flag compilation RLE */
/* nombre de bits par pixel */
typedef struct
{
char manuf;
char vers;
char rle;
char bitpx;
unsigned left;
unsigned top;
unsigned right;
unsigned bottom;
unsigned hres;
unsigned vres;
unsigned char rgb[48];
char vmode;
char rplanes;
unsigned bplin;
unsigned palinfo;
unsigned shres;
unsigned svres;
char filter[54];
} PCX_HEADER;
typedef struct
{
unsigned magic;
char vers;
char comp;
unsigned nimages;
char anim;
unsigned char speed;
char filter[120];
} A95_HEADER;
int fd_in;
int fd_out;
char far *ncbuff;
char far *cbuf;

```

```

" speed = ideal display speed\n"
" between 1 and 255\n"
" anim_file = A95 file\n"
" files [files...] = 240*128 mono PCX files\n"
"\n"
"(c) Jacques Belin, 1994\n" );

exit(-1);
}

strcpy(destname, argv[optind]);
optind++;

allocmem(8000/16, &ncseg);
allocmem(8000/16, &cseg);
ncbuff = MK_FP(ncseg, 0);
cbuff = MK_FP(cseg, 0);

memset(&a95hdr, 0, 128);
a95hdr.magic = MAGIC;
a95hdr.vers = 1;
a95hdr.comp = typecomp;
a95hdr.anim = typeanim;
a95hdr.speed = speed;

fd_out = _creat(destname, FA_ARCH); /* création du fichier A95 */
if(fd_out == -1)
{
printf(stdout, "Unable to create file %s\n", destname);
errflag=1;
}
else
{
if(write(fd_out, &a95hdr, 128) != 128) /* écriture du buffer */
{
printf(stdout, "Unable to write file %s\n", argv[optind]);
errflag=1;
}
}

for( optindkargc) && ierrflag; optind++) /* analyse des opérandes */
{
path = filename + strlen(argv[optind])-1; /* analyse du chemin */
strcpy(filename, argv[optind]);
while((path >= filename) && (*path != '\\') && (*path != ':'))
path--;
path++;
findret = findfirst(argv[optind], &dirdesc, 0); /* recherche noms */
if(findret != 0) /* si pas de fichier trouvé */
{
printf(stdout, "%s not found\n", argv[optind]);
errflag++;
}
while(!findret) /* tant que l'on trouve des fichiers */
{
strcpy(path, dirdesc.ff name); /* constitution path+nom */
fd_in=open(filename, O_RDONLY | O_BINARY);
if(fd_in == NULL) /* si erreur d'ouverture */
{
printf(stdout, "unable to open %s\n", filename);
errflag++;
}
else
{
if(read(fd_in, &pcxhdr, 128) != 128)
{

```

```

/* flag d'erreur */
/* structure d'entrée de répertoire */
/* flac de fin de recherche de fichiers */
/* stockage des noms de fichiers */
/* pointeur sur nom de fichier */
/* nombre de fichiers lus */
/* taille d'un fichier graphique */
/* numéros des segments des buffers */
/* resultat des opérations de lecture */

/* valeurs par défaut */

/* selection des options */
/* type d'animation */

/* mode de compression */

/* vitesse */

/* si pas assez de paramètres */
/* sortie en cas d'erreur */

type = animation type :\n"
0 = one time\n"
1 = loop\n"
2 = ping-pong\n"
mode = compression mode :\n"
0 = no compression\n"
1 = PCX\n"

cdecl main(int argc, char *argv[]);
main(int argc, char *argv[])
{
int errflag=0;
struct ffbk dirdesc;
int findret;
char destname[80];
char filename[80];
char *path;
int nbfiles=0;
int nbytes;
unsigned ncseg, cseg;
int ioflag;
int n, i, j, k;
char c;
PCX_HEADER pcxhdr;
A95_HEADER a95hdr;

char typeanim=LOOP;
char typecomp=PCXCOMP;
unsigned char speed=24;

while((c=getopt(argc, argv, "a:c:s:")) != EOF)
{
switch(c)
{
case 'a' : typeanim=atoi(optarg);
if(typeanim > PINGPONG)
errflag++;
break;

case 'c' : typecomp=atoi(optarg);
if(typecomp > PCXCOMP)
errflag++;
break;

case 's' : speed=atoi(optarg);
if((speed < 1) || (speed > 255))
errflag++;
break;

case '?' : errflag++;
break;
}
}

if((argc-optind) < 2)
errflag++;

if(errflag)
{
printf(stderr, "usage : makea95 [-a type ] [-c mode] [-s speed] "
"anim_file files [files...]\n"
"\n"
" type = animation type :\n"
" 0 = one time\n"
" 1 = loop\n"
" 2 = ping-pong\n"
" mode = compression mode :\n"
" 0 = no compression\n"
" 1 = PCX\n"

```

```

fprintf(stdout, "unable to read %s\n", filename);
errflag++;
}
if(!errflag) /* test (sommaire) validité fichier PCX */
{
if((pcxhdr.manuf != 10) || (pcxhdr.rle != 1) ||
(pcxhdr.bitpx != 1) || (pcxhdr.right != 239) ||
(pcxhdr.bottom != 127) )
{
fprintf(stdout, "unable to read %s\n", filename);
errflag++;
}
}
if(!errflag)
{
nbfiles++;
fprintf(stdout, "%s\n", filename); /* affichege nom */
nbytes = read(fd_in, cbuff, 9999);
fprintf(stdout, " %5d bytes read\n", nbytes);
switch(typecomp)
{
case NOCOMP : k = 0; /* décompression fichier PCX */
for(i=0; i<nbytes; i++)
{
c = cbuff[i];
if ((c >> 6) & 3) == 3) /* si rle */
{
n = c & 0x3f;
c=cbuff[i+1];
for(j=1; j<=n; j++)
ncbuff[k++]=c;
}
else
{
ncbuff[k++]=c;
}
}
nbytes = 3840;
if(write(fd_out, &nbytes, 2) != 2)
errflag++;
if(write(fd_out, ncbuff, nbytes)
!= nbytes)
errflag++;
if(write(fd_out, &nbytes, 2) != 2)
errflag++;
break;
case PCXCOMP : if(write(fd_out, &nbytes, 2) != 2)
errflag++;
if(write(fd_out, cbuff, nbytes)
!= nbytes)
errflag++;
if(write(fd_out, &nbytes, 2) != 2)
errflag++;
break;
}
}
if(errflag) /* si erreur d'écriture */
{
fprintf(stdout, "unable to write %s\n", destname);
}

```

```

}
else
{
fprintf(stdout, " %5d bytes written\n", nbytes);
close(fd_in); /* fermeture fichier PCX */
}
}
findret = findnext(&dirdesc);
}
}
if(!errflag) /* si pas d'erreur */
{
a95hdr.nimages = nbfiles; /* écriture nombre d'images dans fichier */
lseek(fd_out, 0L, SEEK_SET);
write(fd_out, &a95hdr, 128);
fprintf(stdout, "\nCopied in %s\n", destname); /* affichage du rapport final */
fprintf(stdout, "Number of pict. = %d\n", a95hdr.nimages);
fprintf(stdout, "Animation type = ");
switch(typeanim)
{
case ONETIME : fprintf(stdout, "One time\n");
break;
case LOOP : fprintf(stdout, "Loop\n");
break;
case PINGPONG : fprintf(stdout, "Ping Pong\n");
break;
default : fprintf(stdout, "Unknown\n");
break;
}
fprintf(stdout, "Compression mode = ");
switch(typecomp)
{
case NOCOMP : fprintf(stdout, "No compression\n");
break;
case PCXCOMP : fprintf(stdout, "PCX\n");
break;
default : fprintf(stdout, "Unknown\n");
break;
}
fprintf(stdout, "Speed = %d\n", speed);
}
close(fd_out); /* fermeture du fichier */
freemem(ncsegs); /* libération des buffers */
freemem(csegs);
}

```

LE COIN DES CODES

La compilation de certains programmes, tels ceux écrits en assembleur, nécessitent souvent un logiciel que ne possèdent pas tous nos lecteurs. Le *Coin des Codes* permet de résoudre ce problème.

Note importante:

Même si la présentation des listings est identique, le traitement de ceux-ci est différent suivant le programme d'entrée et la machine de destination. Chaque listing est prévu pour être entré sur sa machine de destination. Par exemple, ne tentez pas d'entrer un programme HP48 dans un fichier MS-DOS à l'aide du programme MAKEDOS. Vous obtiendriez un fichier au transfert impossible dans la HP48.

Programmes HP48 et HP28S

Par rapport aux méthodes habituelles sur HP48, notre méthode effectuant un calcul local du checksum en fin de chaque ligne permet de faciliter la recherche d'erreurs, par rapport à une même recherche dans une chaîne de plusieurs centaines d'octets.

Par mesure de sécurité sauvegardez vos programmes et fichiers, éventuellement verrouillez vos cartes.

Tapez le programme correspondant à votre machine:
- ASSCOD48 pour HP-48 ou ASSCOD28 pour HP-28S
(attention aux parties distinctes HP-48 ou HP-28S)
cela avec la plus grande attention car un mauvais fonctionnement peut entraîner un désordre fatal pour les objets contenus dans votre machine. La commande SPEED (JPC-85 page 12) peut être incluse au début de ASSCOD28. NEWLINE (+) s'obtient sur HP-48 par [flèche bleue] puis [.]

Pour tout assemblage de chaîne de codes procédez de manière suivante :

- 1- lancez le programme ASSCOD48 ou ASSCOD28
- 2- donnez le nombre d'octets (1/2 octet compris) puis validez avec ENTER.
- 3- tapez chaque ligne de codes après son numéro à 3 chiffres et sans les espaces puis validez.
- 4- tapez la somme de contrôle sm, validez. S'il y a erreur la ligne de codes sera demandée à nouveau. Pour HP-28 l'appui sur EDIT fera apparaître la ligne des codes qui pourra être corrigée, ensuite relancez avec CONT.
- 5- lorsque l'objet apparaîtra sur la pile stockez le dans la variable donnée en tête.

Nota: des versions ASSCODxx plus performantes (System Rpl et Assembleur) existent dans JPC numéro 95.

ASSCOD48 1114.5 octets ASSCOD28 1032.5 octets
cksum # 621Eh cksum # 55CEh

```
@ ***** HP-48 et HP-28S *****
« RCLF « CLLCD 3 DISP 1 DISP ""
DO DUP 4 DISP
DO UNTIL KEY END
@ ***** HP-48 ***** @@ ***** HP-28S *****
IF DUP 51 == THEN DROP 1 @@ IF DUP "ENTER" ==
ELSE IF DUP 55 == THEN DROP @@ THEN DROP 1
1 OVER SIZE 1 - SUB @@ ELSE
ELSE @@ IF DUP "BACK" ==
CASE DUP 17 < THEN 54 + END @@ THEN DROP 1 OVER
DUP 66 < THEN 7 - END @@ SIZE 1 - SUB
DUP 76 < THEN 20 - END @@ ELSE
DUP 86 < THEN 33 - END @@
DROP 48 END CHR @@
@ ***** HP-48 et HP-28S *****
+ DUP SIZE 1 + 5 MOD NOT 1 FC? AND IF THEN " " +
END END 0 END UNTIL END »
HEX 64 STWS "nombre d'octets ?" ""
@ ***** HP-48 ***** @@ ***** HP-28S *****
INPUT @@ 1 SF 3 PICK EVAL
@ ***** HP-48 et HP-28S *****
1 CF STR→ 2 * 16 DUP2 / IP 3 ROLLD MOD
DUP2 0 > + SWAP 1 + # 0h "" 1 5 ROLL
FOR i
DO "ligne " i 1 - R→B # 1000h + →STR 4 6 SUB + DUP
" @ NEWLINE
codes ?" + "---- ---- ---- ----" 7 PICK i <
IF THEN 6 PICK DUP 4 / IP + 1 SWAP OVER - SUB END
1 FS?C IF THEN DROP2 SWAP
@ ***** HP-48 ***** @@ ***** HP-28S *****
OVER " a corriger" + @@
{ -1 } ROT + INPUT @@ CLMF HALT
@ ***** HP-48 et HP-28S *****
ELSE 8 PICK EVAL END DUP
WHILE DUP " " POS DUP
REPEAT DUP2 1 SWAP 1 - SUB 3 ROLLD 1 + 25 SUB +
END DROP 0 OVER SIZE 1 SWAP
FOR j OVER j DUP SUB NUM j * + NEXT
6 PICK + DUP # FFFh AND
" @ NEWLINE
somme de controle ?" 6 ROLL SWAP + "===="
11 PICK EVAL "#" SWAP + STR→ ==
IF THEN 1 ELSE DROP2 1000 1 BEEP 1 SF 0 END
UNTIL END ROT 5 ROLL DROP2 3 ROLLD +
NEXT 5 ROLLD 4 DROPN
@ **** HP-48 **** @@ ***** HP-28S *****
"GROB 8 " OVER @@ "" SWAP 1 OVER SIZE
SIZE 2 / " " + @@ FOR j DUP j DUP 15 + SUB "#"
+ SWAP + STR→ @@ OVER SIZE 1
# 4017h SYSEVAL @@ FOR i OVER i DUP SUB + -1 STEP
# 62B9Ch @@ SWAP DROP STR→ ROT SWAP
SYSEVAL NEWOB @@ # 3B82h SYSEVAL SWAP 16
SWAP STOF » @@ STEP DROP # 20238h SYSEVAL
@@ # 4F3Dh SYSEVAL SWAP STOF CLMF »
```

INPUT28R (HP28)
461Dh 160 octets

0123 4567 89AB CDEF sm
000: 76C2 01E3 C05A 7C07 E08
001: C170 76C2 0A21 109C B1E
002: 2116 A5A0 F704 0376 7D6
003: 50A4 0203 1CB1 3765 48E
004: 09DA E087 7A0B 8A11 2AF
005: FB65 076C 2009 F206 F93
006: B650 A402 014D C1A0 CE7
007: D603 ACD3 76C2 0601 9C6
008: 203B 1704 8120 2C04 5B4
009: 0F7F 60FC 140F 7040 2ED
00A: 4FB6 009F 2076 C20A 0A4
00B: 4020 1BC9 1A0D 6056 E08
00C: C600 9F20 FB65 076C C4A
00D: 2028 B300 9F20 6B65 995
00E: 0297 A060 1207 FAE0 787
00F: 3ACD 38E4 5076 C20A 58C
010: 1120 163C 069C 2032 1D5
011: 0001 321B F510 CD21 F4B
012: 4413 2326 018D D848 CC2
013: 109F 2009 F200 9F20 9A9

AML2 (HP28)
739Ah 92.5 octets

0123 4567 89AB CDEF sm
000: 76C2 0E4A 2034 0000 B59
001: 1BB7 BFC3 A8BA 2010 96F
002: 0200 2A18 0152 1916 4FD
003: 6F13 21B7 800C 1401 105
004: 8914 08D0 0000 0E4A DE0
005: 2032 0003 2346 4247 92B
006: 3448 6023 5953 5546 52B
007: 514C 4A21 10ED 2116 211
008: A5A0 3315 228B 30FC 026
009: FB0F 1450 21D2 0303 BE2
00A: 5146 5A40 20C6 2A00 85B
00B: 22A0 09F2 0 1E4

SHRINK28 (HP28)
57ACh 30 octets

0123 4567 89AB CDEF sm
000: 76C2 01E3 C05A 7C09 E28
001: F170 76C2 014F 30C1 B43
002: 6D3F BF60 BEE3 0C16 A52
003: D309 F200 9F20 AED

TRAINPPC (HP48)
6A17h 505.5 octets

0123 4567 89AB CDEF sm
000: D9D2 024F 50C4 130E D77
001: 1B20 FC00 0800 0006 8B2
002: 000E 7000 0000 0000 2A9
003: 0000 0000 000C 38F0 EEE
004: 0000 0000 0837 C070 A93
005: 0208 1850 8FFF 1FFF B98
006: 7849 219C 6038 995E 98B
007: 8555 1555 5837 207A 633
008: AA0C FFFF 8555 1555 449
009: 5801 211A 221E FFFF 4AA
00A: AFF7 5DFF 5901 C41C 3A1
00B: 222F FFFF FFFF FFFF 859
00C: FFFF FFFF FF18 93C6 91A
00D: 0606 0C08 1030 60C0 505
00E: 8101 1920 2E00 00CA 1ED
00F: 3053 6216 2D25 03D4 EAE
010: 3AF2 500D 470F EF30 D3A
011: B704 0FD6 21AD B461 B7D
012: 1920 3E00 0F85 1111 74D
013: 9200 6000 FBD8 19F3 5B3
014: 455D 5509 B136 4D30 2AD
015: 5AE8 C149 F211 1920 F16
016: 0600 0FEF 3025 0F42 C66
017: A170 E041 67CC 30EE B7E
018: 170D 9D20 E041 6B20 86D
019: 4083 4160 B821 2CFA 6EB
01A: 2339 2099 9000 0000 182
01B: 0005 30A8 8A2C B916 F9F
01C: D9D2 0C41 30C5 4168 C63
01D: F726 98E3 6442 30D9 9B2
01E: D20F BD81 2CFA 2CB9 A04
01F: A25B CF18 8B26 F083 83A
020: 1B21 30B2 130E 0416 450
021: E0E3 0006 167E 316C 1AD
022: B916 D9D2 00CA 300F FC6
023: 516E 0416 A652 6B20 C8E
024: 4052 831E 0416 A852 945
025: 6710 40CA 1305 2831 527
026: 9B13 6E04 16A9 5267 232
027: 1040 CA13 0528 319B F27
028: 136B 2130 2CFA 2339 CBC
029: 2099 9000 0000 0007 71A
02A: 90A8 8A2E 0416 1192 377
02B: 0060 0038 D306 4B30 FD5
02C: CB91 6D9D 2012 040B CAD
02D: D370 3392 0300 0000 6EA
02E: 0000 0002 04B4 94CA 472
02F: 130A 5141 E041 69FF 2EB
030: 306C 426A 1826 0061 EBF
031: 6B20 4083 4160 B821 ACB
032: A514 1433 70B2 1302 637
033: CFA2 3392 0999 0000 1CA
034: 0000 0057 0A88 A2CB 03D

035: 916D 9D20 18A3 00F5 DAF
036: 16E0 416A 6526 8813 A24
037: 0B20 4052 8311 2040 4E8
038: CA13 0528 319B 136E 259
039: 0416 A752 6120 40CA F65
03A: 1305 2831 9B13 6B21 C03
03B: 304B 4949 C7A1 B213 985
03C: 05E1 70C4 130C 5416 60D
03D: 0ED3 0E62 30D4 1316 284
03E: B316 49F2 1794 70B2 F96
03F: 130 0BD

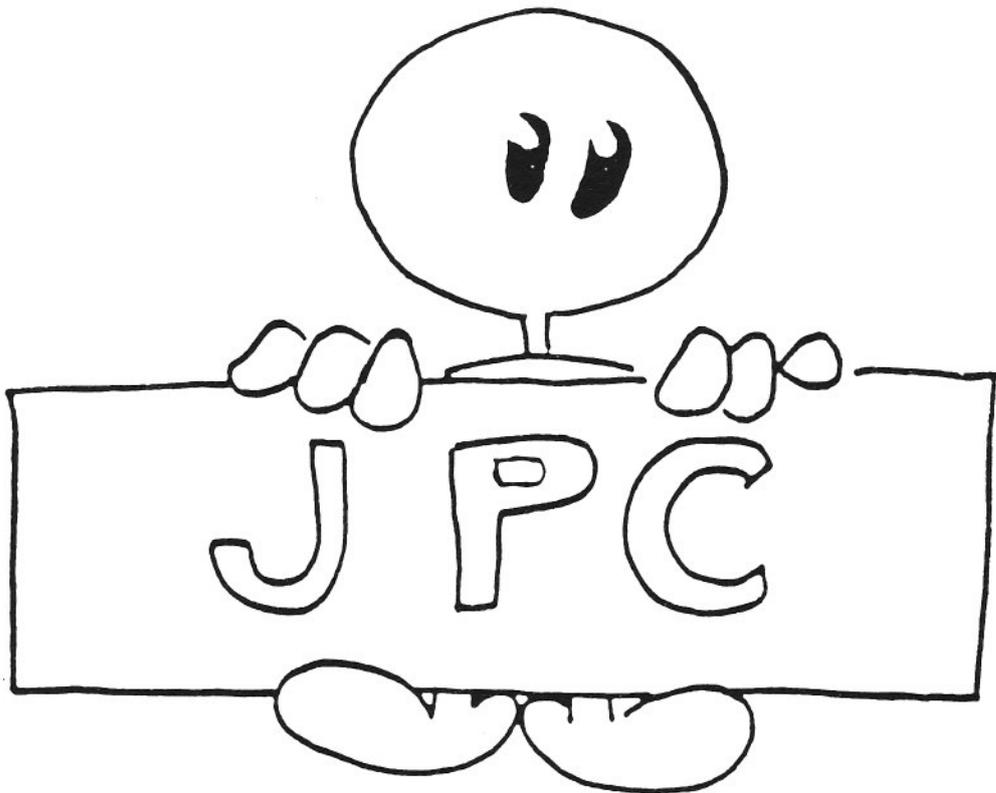
GSCROLL (HP48)
0F57h 129 octets

0123 4567 89AB CDEF sm
000: D9D2 0D29 512B F817 DEA
001: 6040 D9D2 049F 21CF CEA
002: 1462 50F4 2A17 OCCD BE2
003: 2017 0008 FB97 60D0 96E
004: 84F3 2080 7050 808B 604
005: 1243 1047 1408 08B0 209
006: F280 8B1A 2808 B2F1 000
007: 3101 7420 808B 0813 BF9
008: 2FF1 8018 0758 BE4E AD5
009: 4E4E 48DC 7530 8DCE ACA
00A: E10D 9D20 9FF3 0C78 99A
00B: 168C 1313 0040 C781 5CE
00C: 6022 31D0 040C 7816 249
00D: F753 1710 40C7 8164 E97
00E: E431 B213 06AC 308C C70
00F: 1701 7D00 B213 0B21 8AD
010: 30 940

STRIP (HP48)
7649h 326 octets

0123 4567 89AB CDEF sm
000: D9D2 0D48 812B F81F EFB
001: 3040 D9D2 0AD2 A1B3 D87
002: A16D 1236 B3A1 6FA4 C1A
003: 5016 9369 3632 F2E3 9A0
004: 6D9D 2088 1303 F216 62F
005: 6272 6E4E 30B2 1304 2A6
006: 7A20 9427 0E0E 3059 FB0
007: 2701 2270 E0E3 0072 BDC
008: 70B2 130F C436 F743 990
009: 6523 30D9 D207 9E60 726
00A: E25A 1339 16D9 D202 453
00B: C230 4412 6C2A 1644 0E9
00C: 2303 0040 C9B2 66B3 E5C
00D: 16AB E81B 2130 D9D2 C9F
00E: 0D9D 2052 9363 CE22 A1F
00F: E2B3 0B49 1652 936D 797

010: E032 E2B3 0B49 16D9 5B1	01C: 3035 316E 4E30 5233 57A	028: 50B2 130B 2130 96E
011: D205 2936 AFE2 2E2B 4D4	01D: 0B21 3044 230E 4116 150	
012: 30B3 A165 2936 F173 1E0	01E: B213 06B3 16AB E817 F87	
013: 2E2B 30B3 A167 9E60 FAD	01F: BF60 B213 079E 60E1 D1A	SHRINK48 (HP48)
014: D873 2AB8 16D9 D206 DAD	020: 632A B816 DB23 679E BDA	# 6460h 30 octets
015: B316 ABE8 1B21 3079 AF3	021: 60EF 532A B816 D9D2 ACB	
016: E604 5632 AB81 6D9D 9E4	022: 079E 6079 470B 2130 6F6	0123 4567 89AB CDEF sm
017: 2079 E607 9E60 B213 70E	023: 79E6 0C94 32AB 816D 589	
018: 0E41 16B2 1309 4270 2CB	024: 9D20 79E6 079E 60B2 37D	000: D9D2 0D29 51D9 F81F F34
019: A182 6592 7052 3305 E38	025: 1307 E126 CB91 684E 1E5	001: 3040 D9D2 0FA4 50B9 D9F
01A: 2936 5BF2 2F2E 36D9 D53	026: 2050 3545 2594 05B2 E32	002: F06E 4E30 5445 0B9F BE2
01B: D209 4270 3223 02C2 922	027: 1309 4270 4337 0544 9A7	003: 06B2 130B 2130 BB0



Le Journal JPC est le bulletin de liaison entre les membres de l'Association "PPC Paris", régie par la loi de 1901. Le Club est éditeur de JPC, et son siège social est au 56, rue Jean-Jacques Rousseau, 75001 Paris.

PPC Paris est le représentant Français de HEX (Handhelds European Clubs EXchange), la fédération des principaux clubs Européens d'utilisateurs de calculateurs HP.

La maquette de ce numéro a été préparée et réalisée par Jacques Belin et Asdin Aoufi.

Les dessins sont de Jean-Jacques Dhénin et Paul Courbis.

Les informations et programmes parus dans ce journal sont publiées "Tels quels" et ne peuvent en aucun cas engager la responsabilité de Hewlett-Packard ou de PPC Paris. Hewlett-Packard se réserve le droit de ne pas répondre aux questions concernant le sujet de certains articles.

Les programmes publiés peuvent être utilisés librement. Cependant, ils ne peuvent être vendus ou fournis dans un ensemble commercialisé, sous quelque forme que ce soit, sans l'accord écrit de l'auteur ou de PPC Paris.

Directeur de la publication : Jacques Belin
Numéro ISSN : 0762 - 381X

Veuillez adresser toute correspondance à :
PPC Paris, BP 604, 75028 Paris Cedex 01.

Imprimé par Paris Copie, 4 rue Linné, 75005 Paris