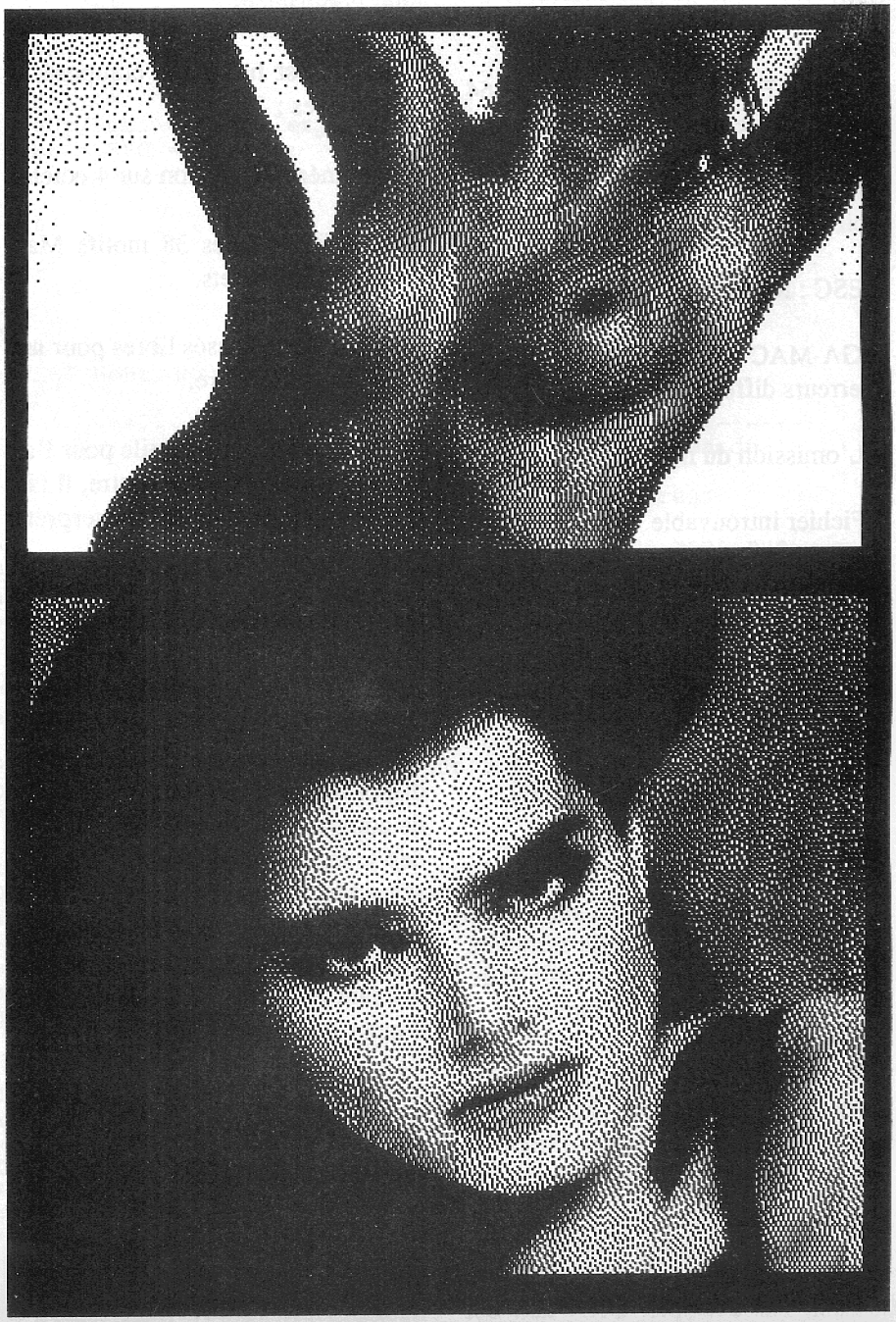
VGA-MAC

Visualiser des images de MAC en VGA

Des programmes tels que FMAC, READMAC, READHERC, ATTMAC ou EGAMAC (disponibles en téléchargement sur le serveur ARCADES) sont géniaux car ils permettent à l'infortuné utilisateur de PC de visualiser des images issues du Macintosh - sans "i" majuscule SVP - et plus précisément du célèbrissime MacPaint du non moins célèbre Bill Atkinson.

Malheureusement, ces images sont chaque fois déformées. En cause : la résolution très élevée et CARREE du Mac. Cependant, depuis l'arrivée de la carte VGA, un PC peut maintenant égaler le Mac sur ce terrain. En effet, le standard VGA dispose (notamment) de la résolution 640x480 qui est comme par hasard (!) celle du Mac. De là, on imagine facilement un programme affichant les images Mac en mode VGA. J'ai donc écrit le programme en assembleur qui accompagne cet article et qui, je l'espère, vous donnera satisfaction.



MISE EN OEUVRE DU PROGRAMME

Pour utiliser VGA-MAC - c'est son nom - il suffit d'entrer en paramètre sur la ligne de commande le nom (et le chemin) de l'image à afficher. Il est capital de laisser UN et UN SEUL espace entre VGA-MAC et le nom de l'image.

Exemple : VGA-MAC C:\IMAGES \ALF.MAC

VGA-MAC charge le fichier et le décompacte afin de l'afficher. Ces étapes sont très rapides, étant donné le langage utilisé : l'assembleur.

Ensuite, l'écran passe en mode 640x480 pixels monochromes.

Un cadre de 576 x 400 pixels est affiché avec au-dessus le nom du fichier et en bas l'aide en ligne. 576 pixels représentent la largeur exacte d'un document MacPaint.

L'aide en ligne reprend les 6 touches disponibles.

• Flèche Haut : Fait défiler le dessin de 8 pixels vers le haut.

• Flèche Bas : Fait défiler le dessin de 8 pixels vers le bas.

• PgUp : Fait défiler le dessin de 64 pixels vers le haut.

• PgDn : Fait défiler le dessin de 64 pixels vers le bas.

• "I" ou "i" - Inverse le dessin. Le noir devient blanc et vice versa.

• ESC : Retourne au DOS.

VGA-MAC est en mesure de détecter 4 erreurs différentes :

• L'omission du nom de fichier.

• Fichier introuvable.

• Carte VGA introuvable.

• Mémoire disponible insuffisante. Il faut au minimum 128 kilo-octets.

AUTRES UTILISATIONS POSSIBLES

Utilisé seul, VGA-MAC permet de visualiser une image, mais associé à d'autres programmes, il peut faire beaucoup mieux. Par exemple :

• Imprimer l'image à l'écran. L'utilitaire GRAPHICS.COM du Dos doit être chargé avant de lancer l'impression en appuyant sur la touche PrtScr.

• Convertir l'image à l'écran en un format utilisable par un logiciel de dessin. Par exemple, DRHALO est livré avec l'utilitaire GRAB.EXE qui permet la capture d'écrans graphiques au format DRHALO. Avec VGA-MAC, il devient possible de récupérer une image Mac sous DRHALO.

UN PEU DE TECHNIQUE

Un document MacPaint est composé de deux parties distinctes :

• Un en-tête de 512 octets.

• Un tableau BitMap de 720 scan-lines compactées.

L'en-tête est lui-même composé de trois parties :

• Le numéro de version sur 4 octets.

• La définition des 38 motifs MacPaint sur 304 octets.

• et 204 octets laissés libres pour une utilisation ultérieure.

En fait, l'en-tête est inutile pour l'affichage sur un PC. Par contre, il faut que le programme puisse interpréter le tableau BitMap.

Un document MacPaint a une résolution de 576 x 720 pixels, soit 414720 pixels. A 8 pixels par octet, il faudrait 51840 octets (50.5 Ko) pour stocker l'image sur disque, si... le Mac ne disposait pas en ROM d'une routine de compactage (et de décompactage) des scan-lines d'un BitMap. Le BitMap MacPaint contenait 720 scan-lines compactées dans un format inconnu de moi. En effet, la routine résidant en ROM, il est inutile d'en donner le format.

Comme il fallait absolument que VGA-MAC puisse décompacter les scan-lines, j'ai dû tricher. J'ai fait ce que l'on appelle du REVERSE ENGINEERING... ou plus vulgairement du désassemblage. J'ai donc désassemblé FMAC avec DEBUG jusqu'à tomber sur une portion de code qui ressemblait à une routine de décompactage. En exclusivité, voici comment ça marche.

Un octet signé précède une série d'autres octets. S'il est négatif, ceux qui suivent sont des octets compactés. Il faut donc d'abord soustraire le premier octet de 0 (NEG en ASSEMBLEUR) pour obtenir un nombre positif. Ce nombre indique le nombre de fois qu'il faudra répéter l'octet suivant. S'il est positif, il indique le nombre d'octets suivant qu'il faudra reproduire fidèlement. On imagine l'intérêt de ce compactage dans des zones de noir, de gris ou de blanc. Le résultat est que la majorité des documents MacPaint occupent environ 20 Ko (au lieu de 50.5 Ko).

CONCLUSION

Voilà ! Je vous souhaite beaucoup d'amusement avec VGA-MAC. Si vous comptez l'employer pour visualiser des images scannées de représentantes du sexe dit faible dans un état que la décence m'interdit de préciser davantage, sachez que vous pourrez toujours compter sur la touche ESC pour rendre à votre écran sa virginité (!), sans confirmation. Merci le programmeur ! On ne sait jamais qui peut surgir à l'improviste... Mon patron... Ma femme... Ma maman !

Si vous trouvez que les détails que je donne sur le Mac sont troublants de précision, vous avez entièrement raison de vous poser des questions... Je suis un traître, un horrible traître, car je possède non seulement un AT 286 mais aussi un Macintosh SE...

Allez, sans rancune !

Denis BAURAIN

```
; ------------------------------------------------------------------------------
; VGA-MAC -  Affichage de documents MacPaint en mode VGA 640x480 - monochrome.
; V1.0 - (c) 1992 - Denis BAURAIN et PC Micro-Informatique
; ASSEMBLEUR MASM MICROSOFT
; ------------------------------------------------------------------------------
CODE        SEGMENT
            ASSUME  CS:CODE, DS:CODE, SS:CODE, ES:CODE
            ORG     100h
; ------------------------------------------------------------------------------
VGAMAC :    JMP     Start
Scrolling   DW      0                       ;└ Gère le déroulement du dessin
InvertFlag  DB      0FFh                    ;└ Gère l'inversion des couleurs
Handle      DW      ?                       ;└ Contient le Handle du fichier
NameLength  DW      ?                       ;└ Contient la longueur du nom
FileName    DB      65 DUP (?)              ;└ Contient le nom du fichier
Msg         DW      ?                       ;└ Pointe sur un des messages
Copyright   DB 13,10,'VGA-MAC - V1.0 - (C) 1991 par Denis BAURAIN'
            DB 13,10,'──────   Blue House Software (Belgique)',13,10,36
NoParmsMsg  DB 13,10,'SYNTAXE = VGA-MAC [Path\]FileName',13,10,36
NoHardMsg   DB 13,10,7,' ERREUR - Une carte VGA est nécessaire...',13,10,36
NoFileMsg   DB 13,10,7,' ERREUR - Fichier introuvable...',13,10,36
NoRAMMsg    DB 13,10,7,' ERREUR - Mémoire disponible insuffisante...',13,10,36
OkMsg       DB 13,10,'- Oooh ! Mmmh... Very Nice.',13,10,'- Merci.',13,10,36
TitleBar    DB 'VGA-MAC - Denis BAURAIN'
TopLine     DB '┌',72 DUP (196),'┐'
SideLines   DB '│',72 DUP (032),'│'
BottomLine  DB '└',72 DUP (196),'┘'
InfoBar     DB 24,'/',25,'/PgUp/PgDn = Dérouler Haut/Bas '
            DB '- I = Inverser - ESC = Quitter'
; ------------------------------------------------------------------------------
Start :     MOV     AH, 4Ah                 ;┐
            MOV     BX, 1000h               ;│
            INT     21h                     ;└ Réduire la RAM réservée à 64 Ko
            MOV     AX, 1A00h               ;┐
            INT     10h                     ;└ Déterminer la carte vidéo
            CMP     AL, 1Ah                 ;┐
            JE      VGACardOk               ;└ VGA présente -> On continue
            MOV     Msg, OFFSET NoHardMsg   ;└ Fixer Msg sur NoHardMsg
            JMP     EndVGAMAC               ;└ Terminer prématurément
; ------------------------------------------------------------------------------
VGACardOk : MOV     SI, 80h                 ;└ Fixer SI sur Ligne de commande
            MOV     CL, [SI]                ;┐
            MOV     CH, 0                   ;└ CX <- Nombre de caractères
            CMP     CX, 0                   ;┐
            JA      ParmsOk                 ;└ Paramètres passés -> On continue
            MOV     Msg, OFFSET NoParmsMsg  ;└ Fixer Msg sur NoParmsMsg
            JMP     EndVGAMAC               ;└ Terminer prématurément
; ------------------------------------------------------------------------------
ParmsOk :   ADD     SI, 2                   ;┐
            DEC     CX                      ;│
            MOV     DI, OFFSET FileName     ;└ Préparer le transfert
            MOV     NameLength, CX          ;└ NameLength <- Longueur du nom
            REP     MOVSB                   ;└ Copier le nom de fichier
            MOV     BYTE PTR [DI], 0        ;└ Ajouter un 0 à la fin du nom
            MOV     AX, 3D00h               ;┐
            MOV     DX, OFFSET FileName     ;│
            INT     21h                     ;└ Ouvrir le fichier
            JNC     FileOk                  ;└ Fichier ouvert -> On continue
            MOV     Msg, OFFSET NoFileMsg   ;└ Fixer Msg sur NoFileMsg
            JMP     EndVGAMAC               ;└ Terminer prématurément
; ------------------------------------------------------------------------------
FileOk :    MOV     Handle, AX              ;└ Sauvegarder le Handle du fichier
            MOV     AH, 48h                 ;┐
            MOV     BX, 1000h               ;│
            INT     21h                     ;└ Réserver 64 nouveaux Ko
            JNC     RAMOk                   ;└ RAM réservée -> On continue
            MOV     Msg, OFFSET NoRAMMsg    ;└ Fixer Msg sur NoRAMMsg
            JMP     EndVGAMAC               ;└ Terminer prématurément
; ------------------------------------------------------------------------------
RAMOk :     MOV     DS, AX                  ;└ Fixer DS sur la RAM réservée
            MOV     AH, 3Fh                 ;┐
            MOV     BX, CS:Handle           ;│
            MOV     CX, 0FFFFh              ;│
            MOV     DX, 0                   ;│
            INT     21h                     ;└ Lire 64 Ko dans le fichier
```

```
                MOV     AH, 3Eh                 ;|
                INT     21h                     ;L Fermer le fichier
                MOV     SI, 512                 ;L Fixer SI sur les données
                MOV     DI, OFFSET LastByte     ;L Fixer DI sur le buffer
                MOV     BX, 0                   ;L BX = Nombre d'octets par ligne
                MOV     CX, 720                 ;L CX = Nombre de lignes (720)
                CLD                             ;L Travailler à l'endroit
;---------------------------------------------------------------------------
UCompLoop :     PUSH    CX                      ;L Sauvegarder CX
                LODSB                           ;|
                CBW                             ;L Charger un octet dans AX
                CMP     AX, 0                   ;|
                JL      Compacted               ;L AX négatif -> Données compactées
                MOV     CX, AX                  ;|
                INC     CX                      ;L Préparer le transfert
                ADD     BX, CX                  ;L BX <- BX + CX
                REP     MOVSB                   ;L Copier CX octets
                JMP     EndLoop                 ;L Sauter à EndLoop
;---------------------------------------------------------------------------
Compacted :     MOV     CX, AX                  ;|
                NEG     CX                      ;|
                INC     CX                      ;L Préparer le décomptage
                ADD     BX, CX                  ;L BX <- BX + CX
                LODSB                           ;|
                REP     STOSB                   ;L Stocker CX octets
;---------------------------------------------------------------------------
EndLoop :       POP     CX                      ;L Récupérer CX
                CMP     BX, 72                  ;|
                JNE     UCompLoop               ;L Pas fin de la ligne -> Boucler
                MOV     BX, 0                   ;L Nouvelle ligne
                LOOP    UCompLoop               ;L Pas fin du document -> Boucler
                MOV     AH, 49h                 ;|
                PUSH    DS                      ;|
                POP     ES                      ;|
                INT     21h                     ;L Libérer RAM réservée au fichier
                MOV     AX, CS                  ;|
                MOV     DS, AX                  ;|
                MOV     ES, AX                  ;L Recopier CS dans DS et ES
                MOV     AX, 0011h               ;|
                INT     10h                     ;L Passer en mode 640x480 mono
                MOV     AX, 1300h               ;|
                MOV     BX, 0001h               ;|
                MOV     CX, NameLength          ;|
                MOV     DX, 0004h               ;|
                MOV     BP, OFFSET FileName     ;|
                INT     10h                     ;L Sortir le nom du fichier
                MOV     CX, 23                  ;|
                MOV     DX, 0035h               ;|
                MOV     BP, OFFSET TitleBar     ;|
                INT     10h                     ;L Sortir la barre de titre
                MOV     CX, 74                  ;|
                MOV     DX, 0103h               ;|
                MOV     BP, OFFSET TopLine      ;|
                INT     10h                     ;L Sortir la ligne supérieure
                MOV     DI, 25                  ;|
                MOV     BP, OFFSET SideLines    ;|
;---------------------------------------------------------------------------
LoopBox :       INC     DH                      ;|
                INT     10h                     ;|
                DEC     DI                      ;|
                JNZ     LoopBox                 ;L Sortir les lignes de côté
                INC     DH                      ;|
                MOV     BP, OFFSET BottomLine   ;|
                INT     10h                     ;L Sortir la ligne inférieure
                MOV     CX, 64                  ;|
                MOV     DX, 1C08h               ;|
                MOV     BP, OFFSET InfoBar      ;|
                INT     10h                     ;L Sortir la barre d'infos
;---------------------------------------------------------------------------
UpDate :        MOV     SI, OFFSET LastByte     ;|
                MOV     AX, 72                  ;|
                MUL     Scrolling               ;L Fixer SI sur la bonne
                ADD     SI, AX                  ;L portion du BitMap
                MOV     AX, 0A000h              ;|
                MOV     ES, AX                  ;L Fixer ES sur la RAM vidéo VGA
```

```
                MOV     DI, 2564                ;|
                MOV     CX, 400                 ;└ Initialiser la boucle YLoop
; ---------------------------------------------------------------------------
YLoop :         PUSH    CX                      ;└ Sauvegarder CX
                MOV     CX, 72                  ;└ Initialiser la boucle XLoop
; ---------------------------------------------------------------------------
XLoop :         LODSB                           ;└ Charger un octet dans AL
                XOR     AL, InvertFlag          ;└ Inverser si c'est nécessaire
                STOSB                           ;└ L'envoyer en RAM vidéo
                LOOP    XLoop                   ;└ Faire de même pour 72 octets
                ADD     DI, 8                   ;└ Passer à la ligne suivante
                POP     CX                      ;└ Récupérer CX
                LOOP    YLoop                   ;└ Faire de même pour 720 lignes
; ---------------------------------------------------------------------------
Use :           MOV     AH, 0                   ;|
                INT     16h                     ;└ Lire une touche
                CMP     AL, 0                   ;|
                JE      ExtCodes                ;└ AL = 0 -> C'est un code étendu
                AND     AL, 11011111b           ;└ Capitaliser la lettre
                CMP     AL, 27                  ;|
                JE      EndUse                  ;└ ESC -> On s'en va
                CMP     AL, 'I'                 ;|
                JNE     Use                     ;└ Pas 'I' -> Lire une autre touche
                XOR     InvertFlag, 0FFh        ;|
                JMP     UpDate                  ;└ Inverser et mettre à jour
; ---------------------------------------------------------------------------
ExtCodes :      CMP     AH, 48h                 ;|
                JNE     NotUp                   ;└ Pas UpArrow -> Essayer encore
                SUB     Scrolling, 8            ;└ Remonter de 8 pixels
                JNL     UpDate                  ;└ Pas trop haut -> Mettre à jour
                MOV     Scrolling, 0            ;└ Ramener au minimum : 0
                JMP     UpDate                  ;└ Mettre à jour
; ---------------------------------------------------------------------------
NotUp :         CMP     AH, 50h                 ;|
                JNE     NotDown                 ;└ Pas DnArrow -> Essayer encore
                ADD     Scrolling, 8            ;└ Descendre de 8 pixels
                CMP     Scrolling, 320          ;|
                JNA     UpDate                  ;└ Pas trop bas -> Mettre à jour
                MOV     Scrolling, 320          ;└ Réduire au maximum : 320
                JMP     UpDate                  ;└ Mettre à jour
; ---------------------------------------------------------------------------
NotDown :       CMP     AH, 49h                 ;|
                JNE     NotPgUp                 ;└ Pas PgUp -> Essayer encore
                SUB     Scrolling, 64           ;└ Remonter de 64 pixels
                JNL     UpDate                  ;└ Pas trop haut -> Mettre à jour
                MOV     Scrolling, 0            ;└ Ramener au minimum : 0
                JMP     UpDate                  ;└ Mettre à jour
; ---------------------------------------------------------------------------
NotPgUp :       CMP     AH, 51h                 ;|
                JNE     Use                     ;└ Pas PgDn -> Lire autre touche
                ADD     Scrolling, 64           ;└ Descendre de 64 pixels
                CMP     Scrolling, 320          ;|
                JA      TooLow                  ;└ Trop bas -> Corriger
                JMP     Update                  ;└ Mettre à jour
; ---------------------------------------------------------------------------
TooLow :        MOV     Scrolling, 320          ;└ Réduire au maximum : 320
                JMP     UpDate                  ;└ Mettre à jour
; ---------------------------------------------------------------------------
EndUse :        MOV     Msg, OFFSET OkMsg       ;└ Fixer Msg sur OkMsg
                MOV     AX, 0003h               ;|
                INT     10h                     ;└ Repasser en mode texte
; ---------------------------------------------------------------------------
EndVGAMAC :     MOV     DX, OFFSET Copyright    ;|
                MOV     AH, 09h                 ;|
                INT     21h                     ;└ Sortir Copyright
                MOV     DX, Msg                 ;|
                INT     21h                     ;└ Sortir le message pointé par Msg
                MOV     AX, 4C00h               ;|
                INT     21h                     ;└ Terminer - Code de sortie 00h
; ---------------------------------------------------------------------------
LastByte :      NOP                             ;└ Premier octet du buffer BitMap
CODE            ENDS
                END     VGAMAC                  ;└ Ouf ! C'est fini !
; ---------------------------------------------------------------------------
```