Hp 49 en couleurs *Hp 49*

Beaucoup de jeux sont créés en niveaux de gris ! Mais comment leurs concepteurs font-ils ? En voici la recette...

La Hp 49 est livrée en stantard avec un écran qui ne diffuse que deux couleurs : le noir et le blanc. Les niveaux de gris sont issus d'une astuce qui joue avec la rémanence de l'écran. Et puis la Rom de la Hp 49 dissimule des routines intéressantes qui permettent facilement de programmer des applications en quatre niveaux de gris sans véritablement en comprendre le fonctionnement, le tout étant d'en connaitre le principe.

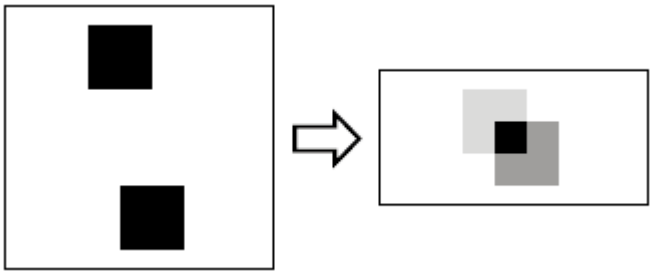
• principe •

Il n'est bien évidemment pas question d'utiliser le langage Rpl pour afficher des niveaux de gris : seul l'assembleur est assez rapide pour en venir à bout. Voyons comment comment afficher une image en quatre niveaux de gris, soient le noir, le gris foncé, le gris clair et le blanc. Le principe est simple, on affiche consécutivement deux grobs de façon très rapide en les superposant et en utilisant la rémanence de l'écran, les pixels n'ont pas le temps de s'allumer ou de s'éteindre complètement. Nous avons donc des pixels entre blanc et noir, c'est à dire du gris. En fait, on utilise le système booléen pour régler les niveaux de gris, comme le présente ce petit tableau :

Tout d'abord, on peut remarquer qu'il faut afficher une fois le premier grob et deux fois le deuxième, ceci continuellement. Si par superposition un pixel se trouve sur le premier grob et ne se trouve pas sur le second, nous aurons du gris clair. Si on applique le même raisonnement pour le gris foncé, alors le pixel se trouvera sur le second grob. Enfin, pour faire du blanc, aucun pixel n'est affiché, alors que pour le noir c'est l'inverse. En général, on ne prend pas deux grobs différents, mais plutôt un grob qui contient les deux. Exemple : pour afficher une image en quatre niveaux de gris de taille 131x64, il faudra utiliser un grob de dimension 131x128, soit le double. Voyons un petit exemple d'image qu'on va préalablement construire avec ce programme en Rpl :

```
« # 20h DUP BLANK NEG DUP
  # 83h # 80h BLANK
  { # 28h # 8h } ROT REPL
  { # 38h # 58h } ROT REPL »
```

Le programme donne à gauche le grob brut, et à droite le résultat tant attendu de la superposition donne effectivement quatre niveaux de gris :



• méthode classique •

Elle consiste à créer soit même son propre sous-programme d'affichage en quatre niveaux de gris. C'est sans doute encore la méthode la plus utilisée et la plus répandue. Notre exemple se basera sur une simple image de taille 131x64 (soit 131x128 en taille réelle, si vous avez bien suivi). Au tout début, on s'efforcera de détecter si les dimensions du grob présent sur le niveau 1 de la pile correspondent à notre critère de taille. Ensuite, il faut régler les marges ! Qu'est-ce que c'est ? Dans notre cas, les marges servent à recaler une image si celle ci est stockée en adresse impaire, mais dans d'autres applications elles permettent de réaliser de superbes scrolls de manière très élégante. Il existe deux types de marges : la marge à gauche et la marge à droite. La marge à gauche est exprimée en bits sur trois bits de poids faibles à l'adresse # 100h : elle sert à déplacer une image au pixel près. La marge à droite est exprimée en quartets sur trois quartets à l'adresse # 125h : elle permet de déplacer une image de quatre pixels à la fois (ou un quartet). En ce qui nous concerne, si le grob est placé en adresse impaire, il faut décaler l'image d'un quartet vers la gauche à l'écran. Passons maintenant à l'affichage des couleurs proprement dit. Comme il l'a déjà été dit, on utilise la rémanence de l'écran pour arriver à nos fins. Pour cela, on va se régler sur le balayage de l'écran (afin d'éviter l'effet de scintillement) et on affichera la première partie du grob initial une fois et la seconde deux fois. C'est à dire qu'on affiche le premier grob, on attend que le balayage de l'écran s'effectue, on affiche le second grob une fois, on effectue un balayage, on réaffiche le second grob encore une fois, on réalise un nouveau balayage, et on boucle... jusqu'à un pression d'une touche. Un balayage d'écran (appelé aussi VSync et placé en # 128h sur 6 bits) s'effectue en 1/64ème de seconde. Et pour finir, il reste juste à placer les adresses respectives de chaque grob après chaque balayage. Plutôt que d'allonger le discours, passons à la source :

```
"!RPL !NO CODE
::
CK1&Dispatch grob
::
TURNMENUOFF
CODE
SAVE INTOFF2 %sauve registres et pointeurs, et interdit les interruptions
A=DAT1.A D1=A D1+5 LC 0110F A=DAT1.A %taille du grob hors prologue
?A#C.A -> FIN %si ce n'est pas la bonne taille, on quitte
AD1EX A+15.A B=A.A %on récupère l'adresse du premier grob
D1=00100 C=DAT1.1 R0=C.P %on sauve la marge à gauche
?ABIT=0.0 -> BOUCLE %si le grob est en adresse impaire,
LC C DAT1=C.1 %on décale la marge à gauche
LC FFF D1=25 DAT1=C.X %ainsi que la marge à droite
*BOUCLE %début de boucle
GOSUB AFF %appel du sous-programme d'affichage
LC 00880 A+C.A %on se décale sur le deuxième grob,
GOSUB AFF GOSUB AFF %et on l'affiche deux fois
A=B.A GOTO BOUCLE %on récupère l'adresse du premier grob et on boucle

*FIN2
```

```
D1=00 C=R0.P DAT1=C.1 %on recharge la marge à gauche,
D0=8068D A=DAT0.8 D1=20 DAT1=A.8 %l'écran courant et la marge à droite
*FIN
INTON2 GOSBVL Flush %réautorise les interruptions, vide le buffer clavier
LOADRPL %récupère registres et pointeurs, et retourne au Rpl

*AFF
D0=00120 DAT0=A.A %on écrit l'adresse du grob à afficher
D1=00129 %on pointe la zone de balayage d'écran
*BAL1
C=DAT1.1 ?CBIT=1.1 -> BAL1 %première partie du balayage
*BAL2
C=DAT1.1 ?CBIT=0.1 -> BAL2 %balayage complet
LC 001 OUT=C=IN %test de la touche [DROP]
?CBIT=1.6 -> FIN2 %si la touche est pressée, on sort
RTN %et on retourne

ENDCODE
;
;
@"
```

La source hexadécimale est à assembler avec H¨ sans espace ni saut :

```
D9D20 00362 F7133 D9D20 430F2 CCD20 FE000 8FB97
608F1 97621 43131 17434 F0110 1438A 65713 3818F
0ED81 F0010 015F0 81A00 88086 06130 C15D0 32FFF
1D521 5537D 40340 8800C A7040 7C30D 468EF 1D008
1A018 15D01 BD860 815A7 1D021 5978F 76762 8F6A7
628D3 41501 B0210 01401 F9210 015F0 808B
```

• méthode par interruptions •

La Hp 49 intègre désormais une routine permettant de réaliser simplement des applications en quatre niveaux de gris, ceci grâce à une routine interne. Que sont les interruptions ? Pour être simpliste, à chaque interruption la machine s'interrompt momentanément pour lancer une autre tâche et revient ensuite où elle en était. Il serait donc intéressant d'utiliser les interruptions pour afficher les niveaux de gris. Dans une application, on peut donc définir une tâche quelconque par interruptions (dans notre cas, l'affichage en niveaux de gris) et de ne plus s'en occuper par la suite, sachant qu'elle se fera automatiquement. La méthode de détournement des interruptions n'est pas chose facile à assimiler et à programmer, c'est pourquoi une routine toute prête s'occupe uniquement de notre cas. Son utilisation est d'ailleurs très simple : il suffit de dire qu'on travaille en niveaux de gris armant un bit et de déclarer l'adresse de chacun des grobs. Les adresses :

• GreyOn? = # 8069Ch sur 1 quartet
• GreyScr1 = # 8069Dh sur 5 quartets
• GreyScr2 = # 806A7h sur 5 quartets
• GreyScr3 = # 806B1h sur 5 quartets

Pour afficher une image en niveaux de gris, il faut placer une valeur différente de zéro dans GreyOn? (les autres routines internes qui utilisent les niveaux de gris y placent la valeur F, chose que nous feront également). Ensuite, on place juste l'adresse du premier grob dans GreyScr1 et l'adresse du second grob dans GreyScr2 et GreyScr3. Remarquez que nous disposons de trois adresses, ceci au cas où un utilisateur utiliserait trois images pour afficher quatre niveaux de gris. Et comme dans la première méthode, n'oublions pas d'enlever la barre des menus et de régler les marges pour les mêmes raisons.

Maintenant, pour revenir à un affichage classique en noir et blanc, il faut remettre la valeur 0 dans GreyOn? mais aussi récupérer la sauvegarde de l'adresse d'écran courant et rétablir les marges qui se trouvent en Ram système en # 8068Dh. N'oublions pas non plus de rétablir les menus, car la routine interne permet également d'afficher des niveaux de gris pour les menus. Voici d'ailleurs les adresses respectives des grobs des menus qui fonctionnent exactement comme pour la partie écran en utilisant GreyOn? :

• GreySoft1 = 806A2 sur 5 quartets
• GreySoft2 = 806AC sur 5 quartets
• GreySoft4 = 806B6 sur 5 quartets

Passons maintenant à la source qui affichera une image de taille 131x64 en quatre niveaux de gris (soit un grob de taille 131x128) :

```
"!RPL !NO CODE
::
CK1&Dispatch grob
::
TURNMENUOFF
CODE
SAVE INTOFF %sauve registres et pointeurs, et interdit les interruptions
A=DAT1.A D1=A D1+5 LC 0110F A=DAT1.A %taille du grob hors prologue
?A#C.A -> FIN %si ce n'est pas la bonne taille, on quitte
GOSUB COULEUR %on appelle la routine d'affichage du grob
*TOUCHE
LC 001 OUT=C=IN ?CBIT=1.6 -> SUITE %test de la touche [DROP]
GOSBVL Flush %on vide le buffer clavier
GOTO TOUCHE %on boucle jusqu'à un appui sur cette touche
*SUITE
D0=8069C C=0.B DAT0=C.1 %on repasse la pile en noir et blanc
D1=00100 C=R0.P DAT1=C.1 %on remet la marge à gauche
D0=8D A=DAT0.8 D1=20 DAT1=A.8 %puis l'écran courant et la marge à droite
D0=95 A=DAT0.A D1=30 DAT1=A.A %et on remet les menus
*FIN
INTON %on autorise les interruptions
GOSBVL Flush %on vide le buffer clavier
LOADRPL %on charge registres et pointeurs, et on retourne au RPL

*COULEUR
D1+15 AD1EX %on avance sur le contenu du grob
D0=00100 C=DAT0.1 R0=C.P %on sauve la marge à gauche
?ABIT=0.0 -> OK %si l'adresse est impaire,
LC C DAT0=C.1 %on se décale sur la marge à gauche
LC FFF D0=25 DAT0=C.X %et sur la marge à droite
```

```
*OK
D0=8069D DAT0=A.A %on place le première écran
LC 00880 A+C.A D0=A7 DAT0=A.A %on place le deuxième écran
D0=B1 DAT0=A.A %on place le dernier écran
D0=9C LC F DAT0=C.1 %et on affiche la couleur
RTN

ENDCODE
;
;
@"
```

Comme pour la méthode précédente, voici la source hexadécimale, toujours à assembler avec la commande `H¨` :

```
D9D20 00362 F7133 D9D20 430F2 CCD20 9F000 8FB97
60808 F1431 31174 34F01 10143 8A616 7D603 21008
FF020 0808B 6D08F 6A762 65EF1 BC960 8AE21 5C01F
00100 81A01 815D0 19D81 5A71D 02159 71959 1421D
03141 80808 F6A76 28D34 15017 E1331 B0010 015E0
81A00 88086 06130 C15C0 32FFF 19521 5431B D9608
14
```

• particularités •

Les deux méthodes ne s'utilisent bien évidemment pas de la même façon et chaque méthodes possède un avantage comme un inconvénient. Avec la première méthode, on est absolument obligé d'appeler la routine qui gère les niveaux de gris à chaque tour de boucle. Ce n'est pas forcément génant lorsque le programme ne contient que peu de boucles. L'avantage de cette méthode est qu'on peut la généraliser pour créer une routine d'affichage de grobs en huit niveaux de gris (ou pourquoi pas plus !). Seul ce morceau de source changerait :

```
[...]
SAVE INTOFF2
A=DAT1.A D1=A D1+5 LC 0198F A=DAT1.A
?A#C.A -> FIN
AD1EX A+15.A B=A.A
D1=00100 C=DAT1.1 R0=C.P
?ABIT=0.0 -> BOUCLE
LC C DAT1=C.1
LC FFF D1=25 DAT1=C.X
*BOUCLE
GOSUB AFF
LC 00880 A+C.A
GOSUB AFF GOSUB AFF
LC 00880 A+C.A
GOSUB AFF GOSUB AFF GOSUB AFF GOSUB AFF
A=B.A GOTO BOUCLE
[...]
```

Pour afficher huit niveaux de gris, le principe est le même que pour quatre niveaux de gris sauf qu'il faut un grob de taille 131x192 en entrée, c'est à dire un grob qui contient trois grobs de 131x64. Le reste de la source est complètement identique.

Pour la deuxième méthode, le problème vient des interruptions : si l'on interdit les interruptions les niveaux de gris ne s'afficheront pas. Par contre, si on les autorise, on les autorise aussi pour le clavier, c'est à dire qu'à chaque tour de boucle il faudra vider le buffer clavier grâce à `GOSBVL Flush`. Par contre, une fois la routine appelée, il n'est plus nécessaire de s'en occuper, l'idéal !
Dans les deux méthodes, nous ne nous sommes pas attardés sur le contraste qui pourtant est primordial lorsqu'on désire un confort d'affichage parfait. Je vous laisse rajouter cet artifice sachant que le contraste se règle sur 5 bits à partir du bit 0 de l'adresse # 101h. Attention toutefois à ne pas dépasser ces fatidiques 5 bits, ce qui pourrait provoquer quelques surprises pas forcément agréables... Les valeurs conseillées sont établies entre # 09h et # 18h, valeurs qui correspondent au mini et au maxi lorsqu'on utilise les touches [ON]-[-] et [ON]-[+] pour régler le contraste.

• the end •

Pour cette fois l'application ne se prête pas aux Hp 48 en ce qui concerne la deuxième méthode, mais la première est totallement compatible (aux adresses de routines internes près).
Une fois acquis le principe de fonctionnement de l'affichage en niveaux de gris, il est tout à fait possible de vous lancer dans des animations ou des scrolls en tout genre. Les jeux qui circulent librement sur l'internet en font des exemples concrêts, comme ces quelques références :

Philippe Pamart
[phpamart@nordnet.fr]