COMPILATION

Il y a longtemps, c'était au temps de la préhistoire, le gibier et les disquettes 3 pouces étaient très rares. Les précieuses petites plaquettes noires s'échangeaient alors contre 70 parfois 80 silex ; le prix d'un kilo de mammouth 1er choix. Il est bien évident qu'à ce taux, chaque programme sauvegardé devait en valoir la peine. C'est sûrement à cette époque qu'est née l'idée de la compilation.

Valable pour
- CPC 464
- CPC 664
- CPC 6128

La compilation est un système qui permet de réduire très sensiblement la taille d'un fichier binaire. Il est surtout bien adapté aux pages écran qui, sur une disquette, prennent 17 K. Une fois compilée, la taille de la même page écran ne fera que 5 ou 6 K.
Les programmes que je vous propose permettent également la compilation de fenêtres. Ceux d'entre vous qui écrivent des jeux d'aventure multicadres sauront en tirer le meilleur parti.

Comment compile-t-on un programme ? Nous savons qu'une page écran occupe &4000 octets de l'adresse &C000 à l'adresse &FFFF. Sur ces &4000 octets beaucoup ont la même valeur. Tous les octets représentant le fond du décor ont la valeur "0".
Le principe de la compilation consiste à compter le nombre d'octets qui se suivent et dont la valeur est identique.

Premier cas. L'octet n° 1 est différent de l'octet n° 2. Il va être recopié tel quel dans le fichier compilé et nous allons comparer le n° 2 avec les suivants.
Deuxième cas. L'octet n° 2 et les dix suivants ont la même valeur. Le programme de compilation va charger dans le fichier compilé un premier octet avec une valeur témoin pour indiquer qu'il y a compilation. Un deuxième octet contiendra le nombre de répétitions. Ici, nous aurons 11 (octet n° 2 plus les dix suivants).

Enfin un troisième octet contiendra la valeur de ces 11 octets.
Comme vous le voyez avec seulement 3 octets, nous en avons compilé 11. Dès que 4 octets se suivent en ayant la même valeur, la compilation nous fait gagner de la place.

Deux cas nous en feront perdre.
Si nous n'avons que deux octets identiques qui se suivent ou si un octet a la valeur du témoin. Là aussi, il nous faudra 3 octets pour en compiler un seul. D'où l'intérêt d'avoir une valeur témoin ayant peu de chance de se trouver dans votre dessin. Pour ma part, dans mes programmes, j'ai utilisé la valeur 103. C'est une configuration de pixels assez rare. Pour vous en convaincre, faites dans les 3 modes :

```
FOR H=&C100 TO &C200:POKE H,103:NEXT
```

```
10 REM ::::::::::::::::::::::::::::::                    >EN
20 REM :                            :                    >EF
30 REM :     COMPILATION BASIC      :                    >XU
40 REM :                            :                    >EH
50 REM ::::::::::::::::::::::::::::::                    >ET
60 MEMORY &2FFF                                          >GA
70 INK 0,0:INK 1,24:MODE 2                               >TE
80 DIM ENCRE (16):WINDOW #1,1,78,25,25                   >EX
90 CLS #1:INPUT #1,"MODE ";MO                            >XV
100 POKE &3000,MO                                        >LY
110 IF MO=0 THEN NBENC=15:GOTO 140                       >BZ
120 IF MO=1 THEN NBENC=3:GOTO 140                        >AU
130 IF MO=2 THEN NBENC=1 ELSE 90                         >YV
140 FOR H=0 TO NBENC                                     >NH
150 CLS #1:PRINT #1,"NUMERO DE L'ENCRE";:PRINT #1,H;:IN  >KY
PUT #1,X
160 ENCRE(H)=X:POKE &3001+H,X:NEXT                       >DJ
170 CLS #1:INPUT #1,"NOM DU DESSIN A COMPILER ";F$       >VE
180 MODE MO:FOR H=0 TO NBENC                             >WE
190 INK H,ENCRE(H):NEXT                                  >TK
200 LOAD "!"+F$,&C000                                    >MD
210 ecran=&C000                                          >BE
220 pack=&3020                                           >QC
230 temoin=103                                           >ZK
240 finecran=&FFA0                                       >NF
250 a=PEEK(ecran):x=0:compteur=0                         >CN
260 IF a=103 THEN 340                                    >NN
270 x=x+1:b=PEEK(ecran+x)                                >VJ
280 IF a=b THEN compteur=compteur+1:GOTO 270             >PD
290 IF compteur=0 THEN 330                               >UP
300 POKE pack,temoin:POKE pack+1,compteur:POKE pack+2,a  >DG

310 pack=pack+3:ecran=ecran+compteur                     >HA
320 IF ecran=finecran OR ecran>finecran THEN 360 ELSE 2  >BE
50
330 POKE pack,a:pack=pack+1:ecran=ecran+1:GOTO 320       >XC
340 POKE pack,temoin:POKE pack+1,1:POKE pack+2,103       >WH
350 pack=pack+3:ecran=ecran+1:GOTO 320                   >HP
360 longpac=pack-&3000                                   >RY
370 lg$=HEX$(longpac):POKE &3012,VAL("&"+RIGHT$(lg$,2))  >AX

380 POKE &3013,VAL("&"+LEFT$(lg$,2))                     >CW
390 MODE 2:CLS:INK 0,0:INK 1,24                          >YA
400 LOCATE 1,1:PRINT "COMPILATION TERMINEE...."          >TB
410 LOCATE 1,3:PRINT "DEBUT DE COMPILATION EN &3000"     >YA
420 LOCATE 1,5:PRINT "LONGUEUR DU FICHIER &";:PRINT HEX  >RL
$(LONGPAC)
```

BASIC/BINAIRE

Claude le MOULLEC

Un autre problème peut se présenter. Le deuxième octet contient le nombre d'octets identiques. Ce nombre ne peut être supérieur à 255. Un dessin avec de grandes surfaces de la même couleur peut avoir cette configuration. Mais cela est très rare. Si vous pensez que votre dessin est dans ce cas, faites :

PLOT 640,1,1:DRAW 640,400

Grâce à ce trait, vous n'aurez jamais plus de 80 octets identiques à se suivre.
Voici pour ce qui est de la compilation. Qui dit compilation dit aussi décompilation. C'est ce que nous allons étudier maintenant.

Le programme de décompilation va lire le fichier créé par la compilation. Si l'octet est différent de 103, celui-ci va être affiché tel quel à l'écran. Si sa valeur est 103, le programme va afficher à l'écran autant d'octets que la valeur contenue après 103. Ces octets auront tous la valeur contenue dans le deuxième octet après 103.

Voici succinctement expliqué le principe de la compilation/décompilation.
Je vous ai écrit deux programmes de démonstration, l'un en BASIC, l'autre utilisant une routine en assembleur.

LE PROGRAMME BASIC

Ce programme n'est là que pour bien vous faire comprendre le principe expliqué plus haut. Les variables que j'ai utilisées sont très parlantes. Néanmoins, ce programme ne compile que des pages écran entières. De plus, malgré la rapidité du BASIC Amstrad, la compilation prend plus de trois minutes et la décompilation à peu près autant.

LE PROGRAMME BINAIRE

Ici, la compilation et la décompilation ne prendront pas plus de deux secondes et il vous sera possible de travailler sur des fenêtres. Le fichier compilé débute à l'adresse &3000 mais le premier octet compilé seulement à l'adresse &3020. Ces 32 octets contiennent tous les paramètres nécessaires au programme de décompilation (mode, couleurs, données de la fenêtre, etc.).
La différence entre un écran complet et une fenêtre réside dans le fait que pour l'écran tous les octets se suivent de façon linéaire, tandis qu'avec une fenêtre, le premier travail du logiciel sera de retrouver cette forme linéaire en transférant tous les octets de la fenêtre à partir de l'adresse &6000. La suite sera identique à la compilation d'un écran sauf qu'elle débutera en &6000 et non plus en &C000. Pour le retour, la décompilation de la fenêtre se fera, elle aussi, en &6000 puis un sous-programme binaire fera le transfert à l'écran.
Vous voici en possession d'un outil performant, essayez d'en faire le meilleur usage.
Bon courage et à vos claviers...

```
430 LOCATE 1,10:INPUT "NOM DE SAUVEGARDE DE CETTE COMPI >WA
LATION ";NOM$
440 SAVE "!"+NOM$,B,&3000,LONGPAC                       >AV
450 END                                                 >TC
```

```
10 REM ::::::::::::::::::::::::::::                     >CA
20 REM :                          :                     >ED
30 REM :   DECOMPILATION BASIC    :                     >ZA
40 REM :                          :                     >EF
50 REM ::::::::::::::::::::::::::::                     >CE
60 MEMORY &2FFF                                         >GA
70 MODE 2:INK 0,0:INK 1,24                              >TH
80 LOCATE 1,1:INPUT "NOM DU DESSIN A DECOMPILER ";NOM$  >BH
90 LOAD "!"+NOM$,&3000                                  >PD
100 MO=PEEK(&3000):MODE MO                              >UH
110 FOR H=0 TO 15:X=PEEK(&3001+H):INK H,X:NEXT          >PZ
120 ecran=&C000                                         >BE
130 pack=&3020                                          >QC
140 temoin=103                                          >ZK
150 finpack=&3000+(PEEK(&3012)+256*PEEK(&3013))         >PU
160 a=PEEK(pack)                                        >LB
170 IF a=103 THEN 210 ELSE 180                          >VA
180 POKE ecran,a                                        >LD
190 ecran=ecran+1:pack=pack+1                           >ZZ
200 IF pack=finpack OR pack>finpack THEN 280 ELSE 160   >WE
210 compteur=PEEK(pack+1)                               >WF
220 valeur=PEEK(pack+2)                                 >TG
230 x=0:FOR h=1 TO compteur                             >WH
240 POKE ecran+x,valeur                                 >UR
250 x=x+1:NEXT                                          >DJ
260 pack=pack+3:ecran=ecran+compteur                    >HE
270 GOTO 200                                            >YH
280 CALL &BB18:END                                      >MY
```

```
10 REM ::::::::::::::::::::::::::::::::                 >FJ
20 REM :                              :                 >EG
30 REM :     COMPILATION BINAIRE      :                 >ZL
40 REM :                              :                 >EJ
50 REM ::::::::::::::::::::::::::::::::                 >FN
60 MEMORY &2FFF                                         >GA
70 FOR h=&A000 TO &A100:READ A$:POKE H,VAL("&"+A$):NEXT >YM
```

```
80 INK 0,0:INK 1,24:MODE 2                                          >TF
90 DIM ENCRE (16):WINDOW #1,1,78,24,25                              >EX
100 CLS #1:INPUT #1,"MODE ";MO                                      >XT
110 POKE &3000,MO                                                   >LZ
120 IF MO=0 THEN NBENC=15:GOTO 150                                  >BB
130 IF MO=1 THEN NBENC=3:GOTO 150                                   >AW
140 IF MO=2 THEN NBENC=1 ELSE 100                                   >ZR
150 FOR H=0 TO NBENC                                                >NJ
160 CLS #1:PRINT #1,"NUMERO DE L'ENCRE";:PRINT #1,H;:IN             >KZ
PUT #1,X
170 ENCRE(H)=X:POKE &3001+H,X:NEXT                                  >DK
180 CLS #1:PRINT #1,"ECRAN COMPLET OU FENETRE  ( E / F              >EQ
) ?"
190 A$=INKEY$:IF A$="" THEN 190                                     >XC
200 A$=UPPER$(A$):IF A$="E" THEN 230                                >BJ
210 IF A$="F" THEN 370 ELSE 190                                     >VH
220 REM :::: COMPILATION D'ECRAN ::::                               >VJ
230 CLS #1:INPUT #1,"NOM DU DESSIN A COMPILER ";F$                  >VB
240 MODE MO:FOR H=0 TO NBENC                                        >WB
250 INK H,ENCRE(H):NEXT                                             >TG
260 LOAD "!"+F$,&C000                                               >NB
270 CALL &A000                                                      >JG
280 CLS:MODE 2:LOCATE 1,1:PRINT "COMPILATION DE ";F$;"              >UW
EFFECTUEE..."
290 LOCATE 1,3:PRINT"DEBUT DE COMPILATION EN &3000"                 >YY
300 LOCATE 1,5:PRINT"LONGUEUR DU FICHIER &";                        >QJ
310 LGFICH=(PEEK(&A00E)+256*PEEK(&A00F))-&3000                      >PB
320 PRINT HEX$(LGFICH)                                              >RQ
330 LOCATE 1,10:INPUT "DONNEZ UN NOM A CETTE COMPILATIO             >MN
N POUR SAUVEGARDE ";FC$
340 SAVE "!"+FC$,B,&3000,LGFICH                                     >YQ
350 END                                                             >TB
360 REM ::: COMPILATION DE FENETRE :::                              >FD
370 INPUT #1,"PARAMETRES COMME POUR UNE WINDOW :(EX 10,             >ZE
20,4,8)";A,B,C,D
380 IF MO=0 THEN X=4                                                >NQ
390 IF MO=1 THEN X=2                                                >NQ
400 IF MO=2 THEN X=1                                                >NG
410 DEPTRANS=&C000+(C-1)*80+(A-1)*X:A$=HEX$(DEPTRANS)               >ZQ
420 NBLIGNE=((D+1)-C)*8                                             >RD
430 OCTRANS=((B+1)-A)*X                                             >RP
440 FINTRANS=(NBLIGNE*OCTRANS)+&6000:B$=HEX$(FINTRANS)              >DR
450 POKE &3016,VAL("&"+RIGHT$(B$,2))                                >CZ
460 POKE &3017,VAL("&"+LEFT$(B$,2))                                 >BF
470 POKE &A007,NBLIGNE:POKE &3019,NBLIGNE                           >KA
480 POKE &A008,OCTRANS:POKE &3018,OCTRANS                           >LB
490 POKE &A003,VAL("&"+RIGHT$(A$,2)):POKE &3014,VAL("&"             >QP
+RIGHT$(A$,2))
500 POKE &A004,VAL("&"+LEFT$(A$,2)):POKE &3015,VAL("&"+             >MF
LEFT$(A$,2))
510 INPUT #1,"NOM DU DESSIN A COMPILER ";F$                         >NN
520 MODE MO:FOR H=0 TO NBENC                                        >WC
530 INK H,ENCRE(H):NEXT                                             >TH
540 LOAD "!"+F$,&C000                                               >NC
550 CALL &A0D0:REM TRANSFERT DE LA FENETRE EN &6000                 >TL
560 CALL &A000:REM COMPILATION                                      >ZP
570 GOTO 280                                                        >ZJ
```

```
580 REM ::::: DATA DE LA ROUTINE ::::                               >CD
590 DATA C3,12,A0,00,00,00,60,00,00,00,C0,A0,FF,00,20,3             >UC
0,01,00,3A,11,30,FE,01,20,0E,21,00,60,22,09,A0,2A,16,30
,22,0B,A0,00,00,2A,09,A0,7E,FE,67,C2,3A,A0,23,22,09,A0,
CD
600 DATA A0,A0,C3,66,A0,23,46,B8,28,0E,22,09,A0,2A,0E,A             >WK
0,77,23,22,0E,A0,C3,66,A0,32,11,A0,F5,3A,10,A0,3C,32,10
,A0,F1,00,23,46,B8,CA,50,A0,22,09,A0,CD,B6,A0,CD,83,A0,
3A
610 DATA 0D,A0,FE,01,CA,79,A0,3E,01,32,10,A0,C3,27,A0,2             >XQ
A,0E,A0,22,12,30,C9,00,00,00,2A,09,A0,ED,5B,0B,A0,7C,BA
,CA,90,A0,C9,7B,BD,F2,96,A0,C9,21,0D,A0,3E,01,77,C9,00,
00
620 DATA 00,2A,0E,A0,3E,67,77,23,3E,01,77,23,3E,67,77,2             >UU
3,22,0E,A0,C9,00,00,00,2A,0E,A0,3E,67,77,23,00,3A,10,A0
,77,23,00,3A,11,A0,77,23,22,0E,A0,C9,00,00,00,2A,03,A0,
ED
630 DATA 5B,05,A0,3A,07,A0,47,C5,E5,3A,08,A0,47,7E,12,1             >PZ
3,23,10,FA,E1,CD,26,BC,C1,10,ED,3E,01,32,11,30,C9,00,00
,00,00,00,00,00,00,00,00,00,00,00●
```

```
10 REM ::::::::::::::::::::::::::::::                               >EN
20 REM :                            :                               >EF
30 REM :   DECOMPILATION BINAIRE    :                               >BU
40 REM :                            :                               >EH
50 REM ::::::::::::::::::::::::::::::                               >ET
60 MEMORY &2FFF                                                     >GA
70 FOR h=&A100 TO &A1B0:READ a$:POKE h,VAL("&"+a$):NEXT             >YH

80 MODE 2:INK 0,0:INK 1,24                                          >TJ
90 LOCATE 1,1:INPUT "NOM DU FICHIER A DECOMPILER ";NOM$             >DH

100 LOAD "!"+NOM$,&3000                                             >PV
110 MO=PEEK(&3000):MODE mo                                          >VA
120 FOR H=0 TO 15:X=PEEK(&3001+H):INK H,X:NEXT                      >PA
130 CALL &A100:REM DECOMPILATION                                    >BH
140 CALL &BB18:END                                                  >MT
150 DATA C3,0D,A1,00,C0,0A,FF,20,30,00,00,00,00,3A,11,3             >VA
0,FE,01,C2,24,A1,21,00,60,22,03,A1,2A,16,30,22,05,A1,00
,00,00,2A,07,A1,ED,5B,03,A1,7E,FE,67,CA,41,A1,12,13,ED,
53
160 DATA 03,A1,23,22,07,A1,C3,4F,A1,00,00,00,23,46,23,7             >WL
E,12,13,10,FC,C3,33,A1,00,00,00,CD,6C,A1,3A,09,A1,FE,01
,CA,5D,A1,C3,24,A1,3A,11,30,FE,01,C2,68,A1,CD,89,A1,C9,
00
170 DATA 00,00,2A,03,A1,ED,5B,05,A1,7C,BA,CA,79,A1,C9,7             >WA
B,BD,F2,7F,A1,C9,21,09,A1,3E,01,77,C9,00,00,00,2A,14,30
,11,00,60,3A,19,30,47,C5,E5,3A,18,30,47,1A,77,13,23,10,
FA
180 DATA E1,CD,26,BC,C1,10,ED,C9,00,00,00,00,00,00,00,0             >EB
0,00,00●
```