

5/10.2

Utilitaires de manipulation de dessins

Lors de la création d'un écran graphique, vous pouvez être confronté au problème suivant : *un objet doit être représenté plusieurs fois sur la même image. Selon la complexité de l'objet, sa duplication peut prendre de quelques minutes à quelques heures.*

Pour éviter cette tâche rébarbative, nous vous proposons trois utilitaires de copie d'objets :

- le premier reproduit un bloc rectangulaire que l'on a défini par ses extrémités bas-gauche et haut-droite ;
- les deuxième et troisième permettent d'obtenir un effet de miroir respectivement par rapport à un axe vertical et par rapport à un axe horizontal.

5/10.2.1

Reproduction de blocs graphiques

Après avoir entré les noms d'écran (avant et après duplication) et les couleurs du dessin, un curseur graphique apparaît en bas et à gauche de l'écran. Dirigez-le avec les touches du bloc numérique pour l'amener en bas et à gauche du rectangle délimitant le bloc à recopier. Appuyez ensuite sur la touche « ENTER ». Déplacez à nouveau le curseur pour l'amener en haut et à droite du même rectangle. Appuyez sur la touche « ENTER ». Un rectangle apparaît. Si l'encadrement est correct (Ques-

tion « OK ? », Réponse « O » (Oui)) déplacez le curseur pour l'amener en bas et à gauche de l'endroit où le bloc doit être dupliqué. Appuyez sur « ENTER ». La copie s'effectue instantanément. A ce niveau, vous pouvez appuyer sur :

- « R » pour revenir à l'écran précédent,
- « S » pour obtenir une sauvegarde magnétique,
- une autre touche pour poursuivre la duplication.

Le programme BASIC de duplication est le suivant :

```

1000 REM Recopie d'un objet sur l'ecran
1010 REM =====
1020 REM 1) Determiner le coin INF GAUCHE de l'objet a deplacer
1030 REM 2) Determiner le coin SUP DROIT de l'objet a deplacer
1040 REM 3) L'objet est encadre
1050 REM 4) Repondre "O" si l'encadrement est correct,"N" sinon
1060 REM 5) Determiner le coin superieur gauche de la nouvelle position
1070 REM 6) L'objet apparait a sa nouvelle position
1080 REM =====
1090 INK 0,0:BORDER 0:INK 1,10:INK 3,10,6:PEN 1:MODE 1:C=1 'Initialisation
1100 '
1110 'Prog. ASM Sauvegarde et affichage ecran
1120 '
1130 FOR I=0 TO &17:READ A:POKE &9000+I,A:NEXT
1140 DATA &21,0,&CD,&11,0,&40,1,&FF,&3F,&ED,&BD,&C9
1150 DATA &21,0,&40,&11,0,&CD,1,&FF,&3F,&ED,&BD,&C9
1160 '
1170 'Prog ASM Interface entre MBG et ABG
1180 '
1190 FOR I=0 TO 28:READ A:POKE &9018+I,A:NEXT
1200 DATA 1,&64,0,&11,&64,0,&26,&A,&2E,&A,&3E,0,&CD,&35,&90,&C9
1210 DATA 1,&64,0,&11,&64,0,&21,0,&80,&CD,&A2,&90,&C9
1220 '
1230 'Programme MBG (Memorisation de blocs graphiques)
1240 '
1250 FOR I=0 TO 108:READ A:POKE &9035+I,A:NEXT

```

```

1260 DATA &18,&B,&73,&20,&58,&20,&61,&75,&20,&64,&65,&7D,&61,&ED,&43,&37,&30,&ED
,&53,&39,&30,&32,&3D,&30,&7C,&32,&3E,&30,&7D,&32,&3C
,&30,&CD,&3,&B9,&21,&0,&80,&11,&94,&2,&3A,&3D,&30,&47,&B7,&28,&4,&ED,&5A,&10,&FC
,&3A,&3E,&30,&77,&23,&22,&3E,&30,&ED,&5B

1270 DATA &37,&30,&2A,&39,&30,&CD,&1D,&BC,&22,&40,&30,&ED,&5B,&3E,&30,&3A,&3E,&3
0,&47,&7E,&12,&23,&13,&10,&FA,&2A,&40,&30,&CD,&26,&B
C,&22,&40,&30,&3A,&3C,&30,&3D,&32,&3C,&30,&20,&E4,&3E,&CF,&12,&C9

1280 '

1290 'Programme ABG (Affichage de blocs graphiques)

1300 '

1310 FOR I=0 TO 72:READ A:POKE &90A2+I,A:NEXT

1320 DATA&18,&A,&41,&42,&47,&D,&A,&3B,&D,&A,&4F,&52,&ED,&43,&A4,&30,&ED,&53,&A6,
&30,&7E,&32,&AA,&30,&23,&22,&AB,&30,&CD,&3,&B9,&ED,&
5B,&A4,&30,&2A,&A6,&30,&CD,&1D,&BC,& 22,&AC,&30,&ED,&5B,&AB,&30,&3A,&AA,&30,&47,
&1A,&77,&23,&13,&10,&FA,&2A,&AC,&30,&CD,&26

1330 DATA &BC,&22,&AC,&30,&1A,&FE,&CF,&20,&EB,&C9

1340 '

1350 PRINT"          COPIE DE BLOCS GRAPHIQUES":PRINT:PRINT

1360 LOCATE 1,6:INPUT"Affichage monochrome (O/N)";R#:R#=UPPER$(R#)

1370 IF R#<>"O" AND R#<>"N" THEN 1360

1371 IF R#="O" THEN NBCOUL=1 ELSE NBCOUL=3

1380 PRINT

1390 FOR I=0 TO NBCOUL

1400   PRINT"INK";I;" : (O A 26) ";:INPUT A:INK I,A

1410 NEXT I

1420 PRINT:PRINT

1421 INPUT"Nom de l'ecran a charger ";N#

1430 INPUT"Nom de l'ecran a sauver  ";N2#

1440 PRINT:PRINT"Une fois toutes les modifications faites, appuyez sur 'S' pour
obtenir une sau- -vegarde de l'ecran sur K7 ou disqu
ette"

1450 PRINT"Appuyez sur 'R' pour retourner a l'ecran precedent et sur une autre t
ouche pour continuer le deplacement des blocs."

1460 PRINT:PEN 3:PRINT"          Appuyez sur une touche"

1470 a#=INKEY$:IF a#="" THEN 1470

1480 LOAD N#,&C000

1490 CALL &9000 'Sauvegarde ecran charge

```

```
1500 B=TEST(X,Y) 'Memo. du pt ou se trouve le curseur
1510 PLOT X,Y,3
1520 '
1530 'Boucle principale
1540 '
1550 A%=INKEY%:IF A%="" THEN 1550 'Attente action
1560 A=ASC(A%)
1570 IF A=55 THEN PLOT X,Y,B:Y=Y+2:X=X-2:GOTO 1500 'En haut a gauche
1580 IF A=57 THEN PLOT X,Y,B:Y=Y+2:X=X+2:GOTO 1500 'En haut a droite
1590 IF A=49 THEN PLOT X,Y,B:Y=Y-2:X=X-2:GOTO 1500 'En bas a gauche
1600 IF A=51 THEN PLOT X,Y,B:Y=Y-2:X=X+2:GOTO 1500 'En bas a droite
1610 IF A=56 THEN PLOT X,Y,B:Y=Y+2:GOTO 1500 'Vers le haut
1620 IF A=50 THEN PLOT X,Y,B:Y=Y-2:GOTO 1500 'Vers le bas
1630 IF A=52 THEN PLOT X,Y,B:X=X-2:GOTO 1500 'Vers la gauche
1640 IF A=54 THEN PLOT X,Y,B:X=X+2:GOTO 1500 'Vers la droite
1650 IF A=13 THEN 1670 'Appui sur ENTER
1660 GOTO 1510
1670 REM Appui sur ENTER
1680 ON C GOTO 1690,1710,1800
1690 'Memo coin superieur gauche
1700 P1=X:P2=Y:C=2:PLOT X,Y,3:GOTO 1500
1710 'Memo coin inferieur droit et trace rectangle
1720 P3=X:P4=Y
1730 PLOT P1,P2:DRAW X,P2:DRAW X,Y:DRAW P1,Y:DRAW P1,P2
1740 IF X>P1 THEN SX=X ELSE SX=P1 'SX=SUP(X,P1)
1750 IF Y>P2 THEN SY=Y ELSE SY=P2 'SY=SUP(Y,P2)
1760 LOCATE ABS(SX/15+2),ABS(26-SY/15):INPUT "OK (O/N) ";R$:R%=UPPER$(R%)
1770 IF R%<>"O" AND R%<>"N" THEN 1760
1780 IF R%="N" THEN CALL &900C:C=1:X=0:Y=0:GOTO 1550
1790 C=3:GOTO 1550
1800 'Tranfert a la position indiquee
1810 'Memo bloc graphique
1820 CALL &900C 'Restitution ecran initial
```

```

1830 A=P1/2:GOSUB 1970:POKE &9019,B:POKE &901A,C
1840 A=P4/2:GOSUB 1970:POKE &901C,B:POKE &901D,C
1850 POKE &901F,INT(ABS((P3-P1)/8))+1:POKE &9021,ABS((P4-P2)/2)+1
1860 CALL &9018 'MBG
1870 'Transfert
1880 A=X/2:GOSUB 1970:POKE &9029,B:POKE &902A,C
1890 A=Y/2:GOSUB 1970:POKE &902C,B:POKE &902D,C
1900 CALL &9028 'ABG
1910 C=1:X=0:Y=0 'Repositionnement curseur
1920 A#=INKEY#:IF A#="" THEN 1920
1930 A#=UPPER$(A#) 'Conversion majuscule
1940 IF A#="R" THEN CALL &900C:GOTO 1500 'Retour a l'ecran precedent
1950 IF A#="S" THEN SAVE N2#,B,&C000,&3FFF
1960 GOTO 1490 'Nouvelle intervention sur le dessin
1970 'Extraction MSB/LSB d'un nombre 16 bits
1980 C=INT(A/256):B=A-C*256
1990 RETURN

```

Lignes 1110 à 1330 : Chargement des sous-programmes ASSEMBLEUR.

Lignes 1350 à 1470 : Initialisation du programme.

Ligne 1480 : Chargement de l'image.

Ligne 1490 : Sauvegarde de l'image pour permettre l'option « R » par la suite.

Lignes 1550 à 1660 : Gestion du curseur.

Lignes 1670 à 1790 : Mémorisation des positions du curseur suite à l'appui sur la touche « ENTER ».

Lignes 1820 à 1900 : Duplication.

Lignes 1910 à 1960 : Attente d'une action au clavier (R, S ou autre).

Le programme BASIC défini ci-dessus utilise des programmes écrits en **ASSEMBLEUR**.

Les sous-programmes de mémorisation et de restitution d'écrans décrits précédemment sont repris ici. De plus, pour permettre les sauvegarde et restitution de portions d'écrans de tailles plus modestes définies par l'utilisateur, deux sous-programmes supplémentaires sont nécessaires : nous les appellerons **MBG** (Mémorisation de Blocs Graphiques) et **ABG** (Affichage de Blocs Graphiques).

MBG :

Avant de décrire la structure du sous-programme MBG, il est nécessaire de faire un rappel sur la structure de l'écran.

L'écran peut être considéré comme une mémoire RAM implantée entre les adresses #C000 et #FFFF.

Reportez-vous à la partie 5/7 pour avoir le détail du codage de l'écran au niveau pixel.

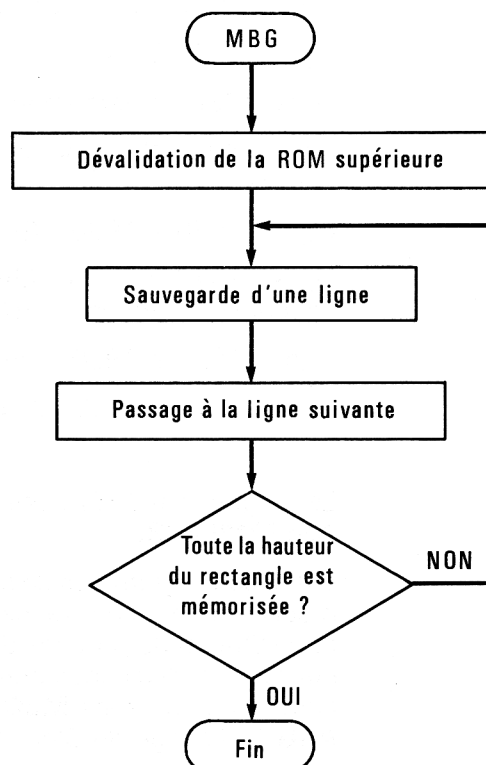
Le problème suivant se pose :

Soit un rectangle de dimensions quelconques dessiné sur l'écran. Comment mémoriser le dessin qui s'y trouve inscrit ?

Vu la complexité structurale de la mémoire d'écran, les constructeurs des CPC ont eu le tact d'agrémenter le FIRMWARE de la routine « DOT-POS » qui, lorsqu'on lui fournit des coordonnées d'écran absolues X et Y, est capable de donner l'adresse de l'octet correspondant en mémoire RAM.

Partant de cette constatation, notre tâche est grandement simplifiée. Connaissant la largeur du rectangle, et vu qu'un octet contient quatre pixels en MODE 1, il sera très simple de sauver une ligne élémentaire. Sachant quelle est la hauteur du rectangle et munis de la routine DOTPOS, le problème devient enfantin. Cependant, précisons un autre petit détail : il ne faut pas oublier de dévalider la ROM supérieure qui occupe également les octets #C000 à #FFFF.

La logique de MBG suit donc l'organigramme suivant :



Voici le listing ASSEMBLEUR correspondant occupant les lignes 1250 à 1270 dans le programme BASIC de recopie d'objets.

```
1          ;
2          ;Memorisateur de blocs graphiques
3          ;
4          ;Entree: BC=Pos X au depart
5          ;          DE=Pos Y au depart
6          ;          H =Largeur en octets
7          ;          L =Hauteur en pixels
8          ;          A =No de bloc a memoriser
9          ;Sortie: Memorisation du bloc
10         ;Pt d'entree: MBG
11         ;
12         ORG 9C6FH
13         LOAD 9C6FH
14         MBG: EQU $          ;Point d'entree
15 9C6F 180B JR MBGDP          ;Debut du programme
16         ;
17         ;Variables locales
18         ;
19         PX: DS 2          ;Pos Abs X au depart
20         PY: DS 2          ;Pos Abs Y au depart
21         LA: DS 1          ;Largeur d'image(octets)
22         HA: DS 1          ;Hauteur d'image(pixels)
23         NB: DS 1          ;No de bloc a memoriser
24         AS: DS 2          ;@ Sauvegarde bloc
25         SAD: DS 2         ;Sauvegarde @ mem ecran
26         APS: EQU 7000H    ;@ 1ere Sauvegarde
27         DOTPOS: EQU OBC1DH
28         NTLINE: EQU OBC26H
29         ;
```

```

30          ;Initialisation
31          ;
32          MBGDP:      EQU  $
33 9C7C ED43719C      LD  (PX),BC          ;Pos X au depart
34 9C80 ED53739C      LD  (PY),DE          ;Pos Y au depart
35 9C84 32779C        LD  (NB),A          ;No de bloc
36 9C87 7C            LD  A,H
37 9C88 32759C        LD  (LA),A          ;Largeur
38 9C8B 7D            LD  A,L
39 9C8C 32769C        LD  (HA),A          ;Hauteur
40 9C8F CD03B9        CALL OB903H          ;UROM DIS
41          ;
42 9C92 210070        LD  HL,APS          ;1ere sauvegarde
43 9C95 119402        LD  DE,294H          ;Espace entre 2 sauveg.
44 9C98 3A779C        LD  A,(NB)
45 9C9B 47            LD  B,A
46 9C9C B7            OR  A
47 9C9D 2B04          JR  Z,MBGDD
48          MBGO:      EQU  $          ;Calcul @ Sauvegarde
49 9C9F ED5A          ADC  HL,DE
50 9CA1 10FC          DJNZ MBGO
51          MBGDD:     EQU  $
52 9CA3 3A759C        LD  A,(LA)
53 9CA6 77            LD  (HL),A
54 9CA7 23            INC  HL          ;Largeur en octets
55 9CAB 227B9C        LD  (AS),HL          ;@ Sauvegarde
56          ;
57 9CAB ED5B719C      LD  DE,(PX)
58 9CAF 2A739C        LD  HL,(PY)
59 9CB2 CD1DEC        CALL DOTPOS          ;OUT HL=@ Mem ecran
60 9CB5 227A9C        LD  (SAD),HL          ;Sauv mem ecran

```



```

61 9CBB ED5E789C      LD  DE, (AS)

62                    ;

63                    MBG1:    EQU  $

64 9CBC 3A759C      LD  A, (LA)

65 9CBF 47          LD  B, A

66                    MBG2:    EQU  $

67 9CCD 7E          LD  A, (HL)      ;Mem ecran

68 9CC1 12          LD  (DE), A      ;Mem sauvegarde

69 9CC2 23          INC  HL

70 9CC3 13          INC  DE

71 9CC4 10FA        DJNZ MBG2

72 9CC6 2A7A9C      LD  HL, (SAD)

73 9CC9 CD26BC      CALL NTLINE

74 9CCC 227A9C      LD  (SAD), HL

75 9CCF 3A769C      LD  A, (HA)      ;Hauteur

76 9CD2 3D          DEC  A

77 9CD3 32769C      LD  (HA), A

78 9CD6 2DE4        JR   NZ, MBG1

79 9CDB 3ECF        LD  A, OCFH

80 9CDA 12          LD  (DE), A      ;Terminateur

81 9CDB C9          RET

82                    END                    ;Zi eind arf arf arf...

```

ABG :

Puisque nous savons sauvegarder en mémoire un objet situé sur l'écran, nous pouvons penser qu'il est simple d'afficher un objet sauvegardé en RAM à une position quelconque sur l'écran. La démarche est effectivement simple : elle consiste à faire l'opposé de la démarche précédente, à savoir :

- 1°) Lecture d'une ligne élémentaire en mémoire,
- 2°) Affichage de la ligne sur l'écran,
- 3°) Passage à la ligne suivante,
- 4°) Si toute la hauteur du rectangle n'est pas parcourue, retour en 1.

D'où le listing du sous-programme ABG dont les données hexadécimales occupent les lignes 1310 à 1330 dans le programme BASIC de copie d'objet.

```

1          ;
2          ; Afficheur de blocs graphiques
3          ;
4          ; Entree: BC=Pos X depart
5          ;          DE=Pos Y depart
6          ;          HL=@ RAM depart
7          ; Sortie: Affichage du bloc
8          ; Pt d'entree: ABG
9          ;
10         ORG 9CDCH
11         LOAD 9CDCH
12         ABG: EQU $           ;Point d'entree
13 9CDC 180A JR ABGDP         ;Debut programme
14         ;
15         ;Variables locales
16         ;
17         PX: DS 2           ;Pos abs X au depart
18         PY: DS 2           ;Pos abs Y au depart
19         MI: DS 2           ;@ de depart RAM image
20         LA: DS 2           ;Largeur image
21         SAD: DS 2          ;Sauveg @ mem ecran
22         DOTPOS: EQU 0BC1DH
23         NTLINE: EQU 0BC26H
24         ;
25         ;Initialisation
26         ;
27         ABGDP: EQU $
28 9CEB ED43DE9C LD (PX),BC   ;Pos X depart
29 9CEC ED53ED9C LD (PY),DE   ;Pos Y depart

```

```

30 9CF0 7E          LD  A,(HL)
31 9CF1 32E49C     LD  (LA),A          ;Largeur image
32 9CF4 23         INC HL
33 9CF5 22E29C     LD  (MI),HL        ;@ Mem image
34 9CF8 CD03B9     CALL DB903H        ;UROM DIS
35                ;
36 9CFB ED5BDE9C   LD  DE,(PX)
37 9CFF 2AE09C     LD  HL,(PY)
38 9D02 CD1DBC     CALL DOTPOS        ;OUT HL=@ Mem ecran
39 9D05 22E69C     LD  (SAD),HL       ;Sauveg @ mem ecran
40 9D08 ED5BE29C   LD  DE,(MI)
41                ABGO: EQU $
42 9D0C 3AE49C     LD  A,(LA)
43 9D0F 47         LD  B,A            ;Largeur image
44                ABG1: EQU $
45 9D10 1A         LD  A,(DE)
46 9D11 77         LD  (HL),A
47 9D12 23         INC HL
48 9D13 13         INC DE
49 9D14 10FA      DJNZ ABG1          ;Affichage 1 ligne elem.
50 9D16 2AE69C     LD  HL,(SAD)
51 9D19 CD26BC     CALL NTLINE        ;Next line
52 9D1C 22E69C     LD  (SAD),HL
53 9D1F 1A         LD  A,(DE)
54 9D20 FECF      CP  DCFH           ;Fin?
55 9D22 20E8      JR  NZ,ABGD        ;Non
56 9D24 C9        RET              ;Zi einde.
57                END

```

Remarque :

Pour faciliter l'interfaçage entre le programme BASIC de recopie d'objets et MBG/ABG, un sous-programme interface a été écrit pour MBG et un autre pour ABG.

Le premier donne les informations nécessaires à MBG dans les registres A, BC, DE et HL :

BC est l'abscisse absolue du rectangle source en haut et à gauche.
 DE est l'ordonnée absolue du rectangle source en haut et à gauche.
 H est la largeur en octets (1 octet = 4 pixels) du rectangle.
 L est la hauteur en pixels du rectangle.
 A donne le numéro de bloc à sauvegarder (0 si un seul bloc est sauvegardé à la fois. C'est le cas ici.)

Le deuxième donne les informations nécessaires à ABG dans les registres BC, DE et HL :

BC est l'abscisse absolue du rectangle à recopier en haut et à gauche.
 DE est l'ordonnée absolue du rectangle à recopier en haut et à gauche.
 HL est l'adresse en RAM où se trouve le pavé à recopier (largeur et hauteur sont contenues dans le bloc de mémoire rempli par MBG : il est donc inutile de les préciser ici).

```

1          ORG 3018H
2          LOAD 3018H
3          MBG: EQU 3035H          ;Memo bloc graph
4          ABG: EQU 30A2H          ;Affich bloc graph
5          ;-----
6          ;Interface avec MBG
7          ;-----
8 3018 016400      LD BC,100          ;X en haut a gauche
9 301B 116400      LD DE,100          ;Y en haut a gauche
10 301E 260A       LD H,10           ;Largeur en octets
11 3020 2E0A       LD L,10           ;Hauteur en pixels
12 3022 3E00       LD A,0            ;Numero de bloc
13 3024 CD3530     CALL MBG
14 3027 C9         RET
15          ;-----
16          ;Interface avec ABG
17          ;-----
18 3028 016400     LD BC,100          ;X en haut a gauche
19 302B 116400     LD DE,100          ;Y en haut a gauche
20 302E 210080     LD HL,0B000H          ;@ RAM bloc
21 3031 CDA230     CALL ABG
22 3034 C9         RET
23          END

```