QUAND ELMAR FAIT DES SIENNES

Elmar Krieger est l'auteur de plusieurs jeux récents sur CPC. Prehistorik 2 et Super Cauldron. Il est aussi l'auteur de Zap'T'Balls « Édition avancée », qui utilise au mieux toutes les techniques de sprites possibles.

Dans ce numéro d'*Amstrad Cent Pour Cent*, nous vous proposons, en exclusivité, les routines de ce grand programmeur qui, pour vous faire plaisir, a préparé quelques RSX de derrière les fagots. Après cela, si vous n'êtes pas capable de créer des petits jeux sympa, c'est que vous n'êtes pas digne de la confiance que nous avons mise en vous.

MODE D'EMPLOI

Quatre programmes vous sont proposés. Les trois premiers doivent être encodés, puis lancés pour générer trois fichiers. Seuls les sprites et les routines en langage machine seront utilisés (l'image écran de 17 Ko pourra être supprimée par la suite).

Pour bien comprendre le principe de ces RSX, encodez le quatrième et dernier programme, et lancez-le tel quel. Vous comprendrez dès lors ce que veut dire une animation fluide sous Basic. C'est aussi ça la classe de nos lecteurs !

Pour les plus mordus d'entre vous, nous allons laisser la parole à notre ami qui va, sous forme de minicours, vous expliquer le principe de ses routines. Cela en vaut la peine.

LAISSONS-LUI LA PAROLE

Pendant les neuf ans que nous avons passés avec notre cher CPC, d'innombrables articles ont été écrits au sujet des sprites. En 1993, il est donc devenu assez difficile de trouver quelque chose dont on n'a pas encore parlé. Cherchons chez nos ancêtres, chez Ghosts'n'Goblins, par exemple, nous trouvons : « la simulation avec deux bitplanes ».

Une technique intéressante, même un peu géniale, qu'on utilise normalement sur PC et qui peut donner de bons résultats sur nos CPC.

Pour gagner de la vitesse, on va presser l'écran à une largeur de 64 octets en X, comme cela se fait dans 80 % des jeux sur CPC. La raison ? En assembleur, on peut utiliser l'instruction INC L (qui est plus rapide que INC HL) pour augmenter une adresse dans la Ram-vidéo. Le deuxième truc : nous nous servons des tableaux pour faire des calculs complexes. Comme nos routines vont accepter les coordonnées d'un sprite en X (0 - 63) et en Y (0 - 255), on doit les transférer en une adresse vidéo avant. Par exemple, avec cette formule :

ADR = &C000 + (Y\8)*64 + (Y MOD 8)*&800 + X

En assembleur, on utilise un tableau de 512 octets à &A400, contenant une adresse vidéo pour chaque ligne en Y. Si le registre HL contient les coordonnées X et Y, la formule se réduit à :

```
LD A,H
LD H,&A4
ADD A,(HL)
INC H
LD H,(HL)
LD L,A
```

Passons maintenant à la « simulation deux bitplanes » elle-même. Dans un mode à 16 couleurs (comme le MODE 0), quatre bits sont nécessaires pour choisir une des 16 couleurs. Le mot « bitplane » exprime que ces quatre bits ne se trouvent pas dans le même octet (comme sur CPC), mais sont partagés en quatre « bitplanes ». L'idée centrale de notre technique : si on met l'arrière-plan dans les bitplanes 0 et 1 (deux bits d'information, alors quatre couleurs comme le MODE 1) et les sprites dans les bitplanes 2 et 3 (encore quatre couleurs), on a réussi. On peut faire n'importe quoi avec les sprites dans 2 et 3 sans toucher le fond dans 0 et 1. Le seul point faible : on doit sacrifier une des quatre couleurs des sprites pour laisser le fond transparent. En MODE 0, il reste donc seulement sept couleurs. Mais honnêtement, l'aviez-vous remarqué dans Ghouls'n'Ghosts, Wonderboy ou Mission Genocide ?

D'autre part, il y a assez de points forts :
- on ne doit pas s'occuper de sauvegarder l'arrière-plan ;
- il faut seulement 50 % de mémoire ;
- en changeant les couleurs, on change aussi la priorité des sprites.

EN AVANT, MARCHE !

Le programme n° 1 génère cinq sprites de démonstration. Si vous voulez dessiner des sprites vous-même, faites ça en MODE 0 avec les quatre premières couleurs de la palette (pour les sprites ET le fond !). Puis, il faut mettre les sprites en mémoire dans une forme appropriée.

UNE OFFRE EXCEPTIONNELLE

Nous vous signalions qu'Elmar était l'auteur du fameux Zap'T'Balls. Si vous désirez recevoir la démo jouable de ce jeu (10 tableaux !), envoyez une disquette et une enveloppe auto-adressée à :
Olivier Martinerie, 9, avenue des Tilleuls, 92290 Châtenay-Malabry.
Le jeu complet, quant à lui, est au prix de 169 F, ou 189 F avec un T-shirt (large ou extra-large). Règlement par chèque à la même adresse. Notez enfin que le jeu fonctionne sur CPC 6128 ou 464 avec extension 64 Ko. Pour le 6128+, le mode deux joueurs n'est pas disponible. Quoi qu'il en soit, si vous désirez avoir plus d'informations, appelez le correspondant d'Elmar en France (Olivier Martinerie) au (1) 46-30-54-54 ; Fax : (1) 46-32-33-68 ; ou écrivez-lui (voir l'adresse plus haut).

C'est fait par le programme n° 2. Regardez ici pour obtenir toutes les informations. La taille des sprites est mise dans un tableau, on ne doit plus s'en occuper après. Dans les lignes DATA, vous devez mettre, pour chaque sprite qui se trouve sur l'écran que vous avez dessiné, son numéro (0 - 127), sa position dans la Ram-vidéo (&C000-&FFFF), sa largeur (un multiple de quatre) et sa hauteur (en octets). Pour les sprites, j'ai réservé l'espace mémoire de &6000 à &97FF, c'est sans doute assez. Les données sont sauvegardées dans le fichier SPRITES.DAT.

Le programme n° 4 génère toutes les routines en assembleur et les tableaux de calcul pour décompacter et afficher les données.

LES COMMANDES RSX

Voilà une explication des commandes RSX (notez que pour les claviers AZERTY vous devez remplacer le « | » qui précède un RSX par « ù »). Chargez 'SPRMCODE. BIN' à l'adresse &9800, et initialisez avec un CALL &9800. Les instructions sont prêtes.

|INIT : initialise l'écran (32*32 caractères).

|PUTBKG,A,X,Y : imprime sprite n° A à la position X,Y dans les couleurs du fond (PUT BACKGROUND).

|PUTSPR,A,R,X,Y : imprime sprite n° A à la position X,Y en direction R [0 = normal, 1 = transparent] (PUT SPRITE).

|PUTSWF,A,R,X,Y : comme l'instruction |PUTSPR, mais attend un Frame Fly avant (PUT SPRITE AND WAIT FOR FRAME FLY).

|REMSPR,A,X,Y : enlève sprite n° A (REMOVE SPRITE).

Autour de chaque sprite, un cadre d'un octet en X et de quatre octets en Y est effacé. Pour simuler un mouvement, changez les coordonnées et imprimez le sprite encore une fois. Si le mouvement est plus grand que le cadre autour, des parties du premier sprite restent visibles. Maintenant, les couleurs. Il y a quatre couleurs au fond. On les appelle COLFON0 à COLFON3. Quant aux trois couleurs des sprites, on les nomme COLSPR1 à COLSPR3. Chaque variable contient une valeur de 0 à 26. On regarde maintenant le cas où les sprites se trouvent devant (la deuxième possibilité doit être trouvée par vous-même, la ligne 360 de la démo peut vous aider...).

Donc, si les bits 2 et 3 sont à zéro, il n'y a pas de sprite et on peut voir la couleur du fond choisie par les bits 1 et 0. En Basic, il faut écrire :

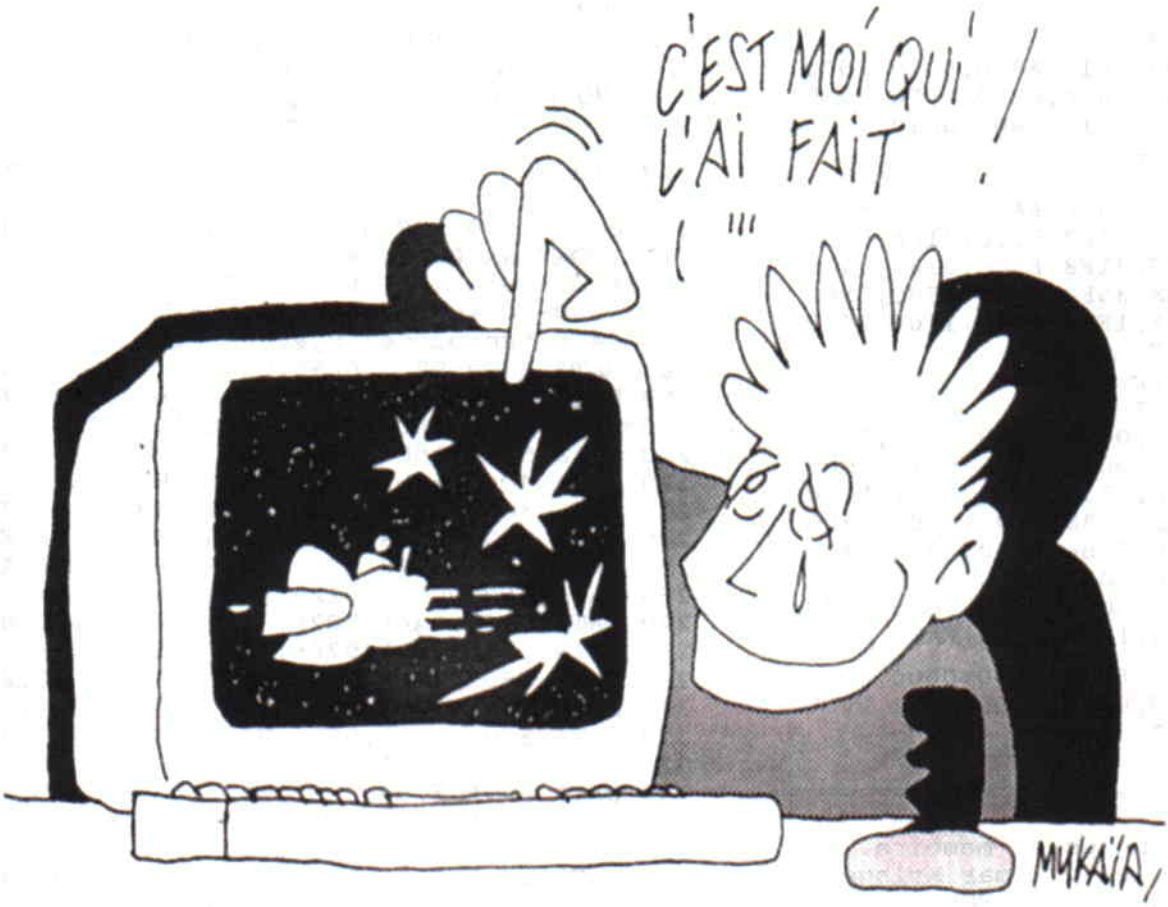
```
INK &X0000,COLFON0
INK &X0001,COLFON1
INK &X0010,COLFON2
INK &X0011,COLFON3
```

Si les bits 2 et 3 ne sont pas vides, la couleur du fond est supprimée :

```
INK &X01YY,COLSPR1
(&X0100,&X0101,&X0110,&X0111)
INK &X10YY,COLSPR2
(&X1000,&X1001,&X1010,&X1011)
INK &X11YY,COLSPR3
(&X1100,&X1101,&X1110,&X1111)
```

Une démonstration de toutes les instructions se trouve dans le programme 4.

Elmar Krieger (Elmsoft)



```
100 'Creation des sprites pour la demo.
110 '(C) AMSTRAD 100% & Elmar Krieger
120 MODE 0:INK 0,0:INK 1,13:INK 2,9:INK
3,3
130 FOR a=0 TO 3:READ adr,br,ho:FOR b=1
TO ho:FOR c=0 TO br-4 STEP 4:READ a$:d=V
AL("&"+LEFT$(a$,2)):POKE adr+c,d AND &CC
:POKE adr+c+1,(d AND &33)*4:d=VAL("&"+RI
GHT$(a$,2)):POKE adr+c+2,d AND &CC:POKE
adr+c+3,(d AND &33)*4:NEXT
140 adr=adr+&800-&C050*(adr>&F7FF):NEXT:
NEXT
150 'Fantome
160 RAD:FOR a=0 TO PI STEP 0.04:b=320+60
*COS(a):MOVE b,296+100*SIN(a):DRAW b,296
,1:DRAW b,182-20*SIN(a*8+PI),2:MOVE b,29
6:DRAW b,196-20*SIN(a*8),1:DRAWR 0,-16,3
:NEXT
170 FOR a=0 TO PI STEP 0.08:b=12*COS(a):
c=320+40*SIN(a):d=SIN(a)*80:MOVE 300+b,c
:DRAWR 0,-d,0:MOVE 336+b,c:DRAWR 0,-d:NE
XT:LOCATE 10,3:PRINT CHR$(22);CHR$(1);:P
EN 1:FOR a=0 TO 5:PRINT CHR$(207);CHR$(2
07);CHR$(8);CHR$(8);CHR$(10);:NEXT
180 SAVE "test.scr",b,&C000,&4000:END
190 'Cloture et terre
200 DATA &C000,4,48
210 DATA 2000,6020,6020,6060,E060,C0F0
220 DATA F0F0,F000,0060,48E0,4868,0C48
230 DATA 850C,8585,0F0F,0F0F,200A,4185
240 DATA 4B96,5043,964A,2C5A,4B5A,5B4F
250 DATA 5F0F,2E2F,2FBD,9FBF,BF2D,4A1B
260 DATA 5F6D,266B,73EE,76D9,73EE,E6D9
270 DATA DDE6,E6DD,D5F7,A2F6,73EA,F6D9
280 DATA 73EE,E6D9,D9E6,E6DD,DDF7,AAFE
290 'Croix
300 DATA &C008,8,9
310 DATA 0010,8000,0010,8800,0010,8000
320 DATA 0010,8800,10F0,F080,10F0,F088
330 DATA 10F0,F080,0010,8800,0010,8000
340 'Pierre tombale
350 DATA &C010,8,16
360 DATA 10F0,F080,30FA,FAC8,31F5,F5CC
370 DATA 31F8,F0C8,31F2,F1CC,31FA,F4C8
380 DATA 31F2,F9CC,31F8,F0C8,31F0,F1C4
390 DATA 30F8,D3C8,35F2,97C4,25BC,B5CE
400 DATA 07B7,BF4E,1EAD,AF8E,1F0F,2D8F
410 DATA 0F0F,0F0F
420 'Chevalier (sur balai?)
430 DATA &C018,8,31
440 DATA 0000,0C00,0001,C200,0012,E000
450 DATA 0012,0E00,0025,C000,0016,E000
460 DATA 0016,4A00,0034,E000,0032,E000
470 DATA 0032,EE00,013D,EF00,01B5,FE00
480 DATA 12F0,FE00,03B4,F400,101E,E100
490 DATA 1087,4B00,10B4,E000,0096,F000
500 DATA 0069,B400,0069,EE00,6625,E000
510 DATA F70F,E108,11EF,5B8C,FAEB,B68C
520 DATA BB1E,8708,6601,4800,0000,C000
530 DATA 0001,C000,0021,E000,0021,F000
540 DATA 0030,0C00
```

=========0=========

```
100 'Mettre les donnees en memoire.
110 '(C) AMSTRAD 100% & Elmar Krieger
120 MEMORY &5FFF:dat=&6000:MODE 0:FOR a=
0 TO 3:READ b:INK a,b:NEXT:FOR a=4 TO 15
:INK a,0,26:NEXT:READ a$:LOAD a$,&C000
130 READ a:IF a=-1 THEN 180
140 READ quell,br,ho
150 IF br MOD 4<>0 THEN br=br+1:GOTO 150
160 br=br-2:POKE &9600+a,dat MOD 256:POK
E &9680+a,dat\256:POKE &9700+a,ho:POKE &
9780+a,br+2
170 FOR b=1 TO ho:FOR c=0 TO br STEP 2:d
=PEEK(quell+c):e=PEEK(quell+c+1):f=d OR
((e\4) AND &33):POKE dat,f:dat=dat+1:NEX
T:POKE quell,255:quell=quell+&800-&C050*
(quell>&F7FF):NEXT:GOTO 130
180 SAVE "sprites.dat",b,&6000,&3800:END
190 'Modifier pour d'autres graphismes..
200 'Couleurs du fond
210 DATA 0,13,9,3
220 'Nom de l'ecran
230 DATA test.scr
240 'Sprite num,adr ecran,format X/Y
250 DATA 0,&C000,4,48
260 DATA 1,&C008,8,9
270 DATA 2,&C010,8,16
280 DATA 3,&C018,8,31
290 DATA 4,&C020,16,120
300 DATA -1
```

=========0=========

```
100 'Programme generant le 'MCODE' etc.
110 '(C) AMSTRAD 100% & Elmar Krieger
120 MEMORY &97FF:MODE 1:PRINT"Attendez 2
6 secondes.."
130 RANDOMIZE 0:FOR a=&9800 TO &9B17 STE
P 12:e=0:FOR b=0 TO 5:READ a$:c=VAL("&"+
LEFT$(a$,2)):d=VAL("&"+RIGHT$(a$,2)):POK
E a+b*2,c:POKE a+b*2+1,d:e=(e+c*RND*50+d
*RND*50) AND &F:NEXT:READ e$:IF e<>VAL("
&"+e$) THEN PRINT "Erreur en ligne:";(a-
&9800)\12+19:EN
140 NEXT
150 FOR a=0 TO 255:POKE &A000+a,(a\4) AN
D &33:POKE &A100+a,a AND &33:POKE &A200+
a,((a AND &40)\2) OR ((a AND &80)\8) OR
((a AND &4)\2) OR ((a AND &8)\8):POKE &A
300+a,((a AND &10)*2) OR ((a AND &20)\2)
 OR ((a AND &1)*2) OR ((a AND &2)\2):NEX
T
160 adr=&C000:FOR a=0 TO 255:ad$=HEX$(ad
r,4):POKE &A400+a,VAL("&"+RIGHT$(ad$,2))
:POKE &A500+a,VAL("&"+LEFT$(ad$,2)):adr=
adr+&800-&C040*(adr>&F7FF):NEXT
170 SAVE "sprmcode",B,&9800,&E00
180 DATA 2109,9801,0D98,C3D1,BC00,0000,5
190 DATA 001E,98C3,3B98,C35C,98C3,5398,7
200 DATA C3C5,9AC3,789A,494E,49D4,5055,6
210 DATA 5453,50D2,5055,5453,57C6,5245,9
220 DATA 4D53,50D2,5055,5442,4BC7,0021,6
230 DATA 4C98,0E07,06BC,ED49,06BE,EDA3,7
240 DATA 0D20,F5C9,2220,0026,8E2A,204F,E
250 DATA 06F5,ED78,1F30,F979,3277,9ADD,A
260 DATA 6E00,DD66,0225,FA6A,9A7D,D604,0
270 DATA DA6A,9A6F,7C26,A486,2466,6FEB,0
280 DATA DD6E,0626,974E,CBFD,46FD,6978,6
290 DATA C602,32C8,9832,7299,32A2,9932,2
300 DATA 4C9A,3E10,904F,8767,8784,8132,0
310 DATA E698,32C0,9926,967E,CBBD,6E67,9
320 DATA ED73,689A,F3F9,DD7E,043D,CA8B,2
330 DATA 993E,FF90,32D1,9832,5C99,327B,3
340 DATA 990E,CCFD,2604,EB06,007E,A177,4
350 DATA 2C10,FA2D,1100,0719,3004,1140,4
360 DATA C019,FD25,20E9,EB0E,CC1A,A112,6
370 DATA 1C18,00E1,4426,A01A,A1B6,1224,5
380 DATA 1C1A,A1B6,1225,1C68,1AA1,B612,4
390 DATA 241C,1AA1,B612,1CE1,4426,A01A,8
400 DATA A1B6,1224,1C1A,A1B6,1225,1C68,6
410 DATA 1AA1,B612,241C,1AA1,B612,1CE1,9
420 DATA 4426,A01A,A1B6,1224,1C1A,A1B6,7
430 DATA 1225,1C68,1AA1,B612,241C,1AA1,8
440 DATA B612,1CE1,4426,A01A,A1B6,1224,2
```

```
450 DATA 1C1A,A1B6,1225,1C68,1AA1,B612,F
460 DATA 241C,1AA1,B612,1C1A,A112,EB01,5
470 DATA 0007,0930,0401,40C0,09EB,FD2D,7
480 DATA C2DF,980E,CCFD,2604,EB06,007E,7
490 DATA A177,2C10,FA2D,1100,0719,3004,8
500 DATA 1140,C019,FD25,20E9,C362,9A3E,1
510 DATA 0180,32AB,9932,369A,3255,9A7B,6
520 DATA 805F,1C0E,CCFD,2604,EB06,007E,5
530 DATA A177,2D10,FA2C,1100,0819,3004,B
540 DATA 1140,C019,FD25,20E9,EB0E,CC1A,0
550 DATA A112,1D18,00E1,4426,A21A,A1B6,B
560 DATA 1224,1D1A,A1B6,1225,1D68,1AA1,0
570 DATA B612,241D,1AA1,B612,1DE1,4426,A
580 DATA A21A,A1B6,1224,1D1A,A1B6,1225,F
590 DATA 1D68,1AA1,B612,241D,1AA1,B612,B
600 DATA 1DE1,4426,A21A,A1B6,1224,1D1A,0
610 DATA A1B6,1225,1D68,1AA1,B612,241D,D
620 DATA 1AA1,B612,1DE1,4426,A21A,A1B6,7
630 DATA 1224,1D1A,A1B6,1225,1D68,1AA1,B
640 DATA B612,241D,1AA1,B612,1D1A,A112,C
650 DATA EB01,0008,0930,0401,40C0,09EB,7
660 DATA FD2D,C2B9,990E,CCFD,2604,EB06,0
670 DATA 007E,A177,2D10,FA2C,1100,0819,0
680 DATA 3004,1140,C019,FD25,20E9,1108,8
690 DATA 00DD,1931,0000,FB3A,779A,D604,4
700 DATA 3277,9AC2,5F98,C900,ED47,CDF2,1
710 DATA 9ACB,3AFD,6A4B,EBDD,6E04,2696,B
720 DATA 7ECB,FD66,6FEB,FD45,7DFD,671A,2
730 DATA E6CC,772C,1A17,17E6,CC77,2C13,F
740 DATA 10F1,FD7C,6F7C,C608,6730,083E,6
750 DATA 4085,6F3E,C08C,670D,20D8,1106,4
760 DATA 00DD,19ED,57D6,03ED,47C2,7A9A,2
770 DATA C9FD,67CD,F29A,425D,0ECC,7EA1,6
780 DATA 772C,10FA,6B7C,C608,6730,0401,B
790 DATA 40C0,09FD,2D20,E711,0600,DD19,B
800 DATA FD7C,D603,FD67,C2C7,9AC9,DD5E,9
810 DATA 00DD,5602,DD6E,0426,977E,CBFD,1
820 DATA 666F,EBFD,6B7C,26A4,8624,666F,E
830 DATA C900,0000,0000,0000,0000,0000,C
```

=0=

```
100 'Demo des commandes RSX
110 '(C) AMSTRAD 100% & Elmar Krieger
120 MEMORY &5FFF:MODE 0:LOAD "sprites.da
t",&6000:LOAD "sprmcode",&9800
130 'Initialiser
140 CALL &9800:|INIT:MODE 0:hgfa(1)=0:hg
fa(2)=13:hgfa(3)=9:hgfa(4)=3:spfa(1)=26:
spfa(2)=2:spfa(3)=15:v=1:GOSUB 310
150 LOCATE 1,1:PRINT"UN CIMETIERE....":F
OR a=0 TO 60 STEP 4:|PUTBKG,0,a,208:NEXT
:b=0:FOR a=6 TO 48 STEP 14:x(b)=a:y(b)=1
99:b=b+1:|PUTBKG,1,a,199:|PUTBKG,2,a,208
:NEXT:h(0)=85:h(1)=40:h(2)=60:h(3)=75:GO
SUB 300
160 LOCATE 1,1:PRINT"MAL S'APPROCHE..":F
OR a=0 TO 3:FOR b=0 TO h(a):CALL &BD19:|
PUTBKG,1,x(a),y(a):y(a)=y(a)-2:NEXT:NEXT
:GOSUB 300
170 LOCATE 1,1:PRINT"LE GRAND SAUVEUR":x
=28:y=177:r=0:FOR a=0 TO y:|PUTSWF,3,r,x
,a:NEXT
180 LOCATE 1,1:PRINT"CURS.,COPY,SPACE"
190 |PUTSWF,3,r,x,y:x2=x
200 x=x-INKEY(8)+INKEY(1)+(x>53)-(x<2):y
=y-4*INKEY(0)+4*INKEY(2)+4*(y>193)-4*(y<
8):IF x2=x THEN 210 ELSE IF x2<x THEN r=
0 ELSE r=1
210 IF NOT INKEY(9) THEN GOSUB 310
220 IF INKEY(47) THEN 190
230 |REMSPR,3,x,y
240 LOCATE 1,1:PRINT"  IL VIENT....  ":|
PUTSPR,3,0,2,191:GOSUB 300
250 v=0:r=0:GOSUB 310:x=24:GOSUB 290:WHI
LE INKEY$="":FOR a=130 TO 20 STEP -4:|PU
TSPR,4,r,x,a:NEXT:FOR a=0 TO 10:|PUTSPR,
4,r,x,20:r=r XOR 1:FOR b=1 TO 200:NEXT:N
EXT:GOSUB 310:FOR a=20 TO 130 STEP 4:|PU
TSPR,4,r,x,a:NEXT:WEND
260 v=0:GOSUB 310:LOCATE 1,1:PRINT "C'ES
T COURAGEUX!":GOSUB 300
270 FOR a=191 TO 6 STEP -1:|PUTSWF,3,0,2
,a:NEXT:FOR a=2 TO 28:|PUTSWF,3,0,a,6:NE
XT:FOR A=6 TO 94 STEP 4:|PUTSWF,3,0,28,a
:NEXT:FOR a=130 TO 250 STEP 4:POKE &9704
,251-a:|PUTSWF,3,0,28,a-36,4,0,x,a:NEXT:
FOR A=214 TO 6 STEP -1:|PUTSWF,3,0,28,a:
NEXT
280 LOCATE 1,1:PRINT" LA VICTOIRE ?  ":G
OSUB 300:LOCATE 1,1:PRINT "AAAARRRRGGHH!
!!!":OUT &BC00,8:OUT &BD00,1:x=4:GOSUB 2
90:x=24:GOSUB 290:x=44:GOSUB 290:OUT &BD
00,0:LOCATE 1,1:PRINT "NO HAPPY END...."
:GOSUB 300:GOTO 370
290 FOR a=250 TO 130 STEP -4:POKE &9704,
251-a:|PUTSWF,4,0,x,a:NEXT:RETURN
300 WHILE INKEY$<>"":WEND:CALL &BB18:RET
URN
310 'Couleurs
320 v=v XOR 1:IF v=1 THEN 350
330 'Sprites devant
340 FOR a=0 TO 3:INK a,hgfa(a+1):NEXT:FO
R a=4 TO 12 STEP 4:FOR b=a TO a+3:INK b,
spfa(a/4):NEXT:NEXT:RETURN
350 'Sprites derriere
360 FOR a=1 TO 3:FOR b=1 TO 3:INK a*4+b,
hgfa(b+1):NEXT:NEXT:FOR a=4 TO 12 STEP 4
:INK a,spfa(a/4):NEXT:RETURN
370 'Fin (reset CRTC)
380 OUT &BC00,1:OUT &BD00,40
390 OUT &BC00,2:OUT &BD00,46
400 OUT &BC00,6:OUT &BD00,25
410 OUT &BC00,7:OUT &BD00,30
420 END
```

Jouez et gagnez plein de jeux sur 3615 ACPC, rubrique MOT FLASH.