

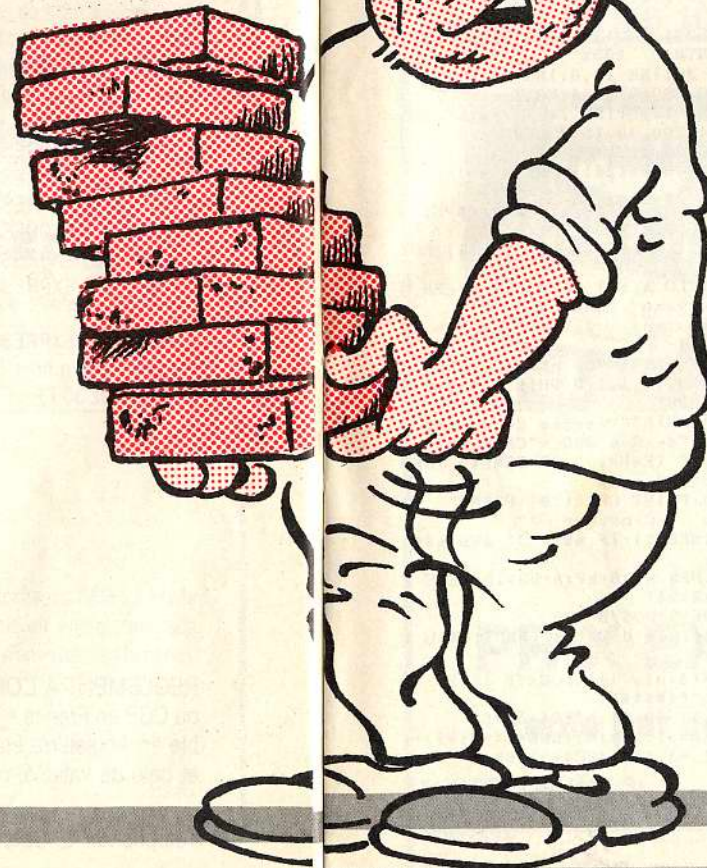
LAISSEZ-MOI LA BLEUE!

Il y avait beaucoup de monde qui grouillait de stand en stand. C'était en novembre 90 dans les allées d'Amstrad Expo. Un certain Anthony m'accosta et sans me laisser le temps de dire un mot, me refila un paquet et disparut avec la même rapidité qu'il était apparu.

Avant de vous raconter la suite de l'histoire, je tiens à ouvrir une parenthèse (Vous étiez nombreux à constater avec stupeur qu'il manquait un



Y'IA DES FOIS J'VOUS JURE ...



petit quelque chose le mois dernier. Voici les quatre lignes qu'il faudra exécuter telles quelles pour rassembler les deux programmes binaires générés par les DATA. Pour tout vous dire j'ai été largement puni en donnant une bonne centaine de fois au téléphone le petit prog manquant). Je ferme la parenthèse et suite de notre histoire.

```
10 MEMORY &4FFF
20 LOAD "MOL.1.BIN",&5000
30 LOAD "MOL.2.BIN",&6A48
40 SAVE "MOLEULE".B,&5000,
&348F,&S45D
```

Il s'agissait d'une disquette remplie de petits prog, mais surtout d'un jeu nommé Xalk. C'est une création à la limite de Klax avec une dose de Tétris.

## XALK, LE MODE D'EMPLOI

Pour profiter de l'offre de ce mois, encodez les cinq programmes que voici. Le premier générera un fichier nommé XALK.BIN enfermant les codes binaires pour l'affichage des sprites. Le second vous sauvera sur votre disquette un programme nommé XALK.SPR. Ce sont les sprites du jeu. Le troisième générera une image de 17 Ko pour le jeu. Le quatrième programme ouvrira sur la disquette le fichier des meilleurs scores. Enfin, le dernier programme est le jeu proprement dit et sera lancé une fois les 4 autres générés sur la disquette. Vous verrez enfin comment avec une dose d'assembleur on peut faire valser les commandes de notre Basic.

```
10 lig=1000
20 FOR ad=&9000 TO &9000+&6C STEP 8
30 chk=lig
40 FOR i=0 TO 7
50 READ vs:v=VAL("&"+vs)
60 POKE ad+i,v
70 chk=chk+(i+1)*v
80 NEXT
90 chk=chk MOD 256+INT(chk/256)
100 IF chk>255 THEN chk=chk-256
110 READ bon$:bon=VAL("&"+bon$)
120 IF chk<>bon THEN 190
130 PRINT "Ligne":lig:"Ok...":CHR$(13)
140 lig=lig+10
150 NEXT
160 PRINT "All is Ok en ligne":Lig
170 SAVE "xalk.bin",b.&9000,&6C
180 END
190 PRINT "Erreur en ligne":lig
200 END
1000 DATA DD,5E,00,DD,56,01,DD,6E,38
1010 DATA 02,DD,66,04,D5,7D,44,CB,51
1020 DATA 3F,CB,19,CB,3F,CB,19,CB,62
1030 DATA 3F,CB,19,CB,39,CB,39,CB,2F
1040 DATA 39,CB,39,CB,39,21,00,C0,AA
1050 DATA 11,50,00,C5,CD,5C,90,C1,18
1060 DATA 79,11,00,08,C5,CD,5C,90,9E
1070 DATA C1,58,16,00,19,D1,DD,46,92
1080 DATA 06,E5,C5,EB,01,05,00,ED,A1
1090 DATA B0,EB,C1,E1,D5,11,00,08,6F
1100 DATA 19,D2,58,90,11,50,C0,19,9F
1110 DATA D1,10,E6,C9,06,08,CB,0F,81
1120 DATA D2,64,90,19,CB,23,CB,12,06
1130 DATA 10,F4,C9,00,00,00,00,00,C5
```

```
100 'copyright AMSTRAD 100 %
110 MEMORY &A000: CLEAR: MEMORY &4FFF
120 ADR=&5000:L=1000
130 FOR i=0 TO 59:GOSUB 180:NEXT i
140 ADR=&5300:L=1600
150 FOR i=0 TO 147:GOSUB 180:NEXT i
160 SAVE "XALK.SPR".B,&5000,2000
170 END
180 FOR J=1 TO 8:READ A$
190 A=VAL("&"+A$)
200 B=(B+i+A*J) AND 255
210 POKE ADR+i*8+J-1,A:NEXT J
220 READ BS:IF B=VAL("&"+BS) THEN 240
230 PRINT "ERREUR EN ":L+i*10:STOP
240 RETURN
1000 DATA 44,CC,CC,CC,88,DD,FF,FF,37
1010 DATA FF,EE,DD,3F,3F,3F,EE,DD,CC
1020 DATA 9D,3F,6E,EE,DD,CC,3F,CC,2B
1030 DATA EE,DD,CC,CC,CC,EE,DD,CC,7A
1040 DATA 0C,CC,EE,DD,8C,0C,4C,EE,04
1050 DATA DD,0C,0C,0C,EE,DD,FF,FF,3A
1060 DATA FF,EE,44,CC,CC,CC,88,00,BD
1070 DATA 00,00,00,00,04,0C,0C,0C,05
1080 DATA 08,5D,FF,FF,FF,AE,5D,CC,FA
1090 DATA CC,CC,AE,5D,4C,CC,8C,AE,AC
1100 DATA 5D,0C,CC,0C,AE,5D,0C,0C,4D
1110 DATA 0C,AE,5D,0C,FF,0C,AE,5D,41
1120 DATA 5D,FF,AE,AE,5D,FF,FF,FF,7A
1130 DATA AE,5D,FF,FF,FF,AE,04,0C,CE
1140 DATA 0C,0C,08,00,00,00,00,00,7A
1150 DATA 15,3F,3F,3F,2A,7F,FF,FF,FB
1160 DATA FF,BF,7F,C0,C0,C0,BF,7F,E6
1170 DATA 6A,C0,95,BF,7F,3F,03,40
1180 DATA BF,7F,3F,3F,3F,BF,7F,3F,6C
1190 DATA CC,3F,BF,7F,6E,CC,9D,B8
1200 DATA 7F,CC,CC,CC,BF,7F,FF,FF,A9
1210 DATA FF,BF,15,3F,3F,3F,2A,00,E4
1220 DATA 00,00,00,00,50,F0,F0,D4
1230 DATA A0,E1,C3,C3,C3,D2,E1,30,a5
1240 DATA 30,30,D2,E1,B0,30,70,D2,1f
1250 DATA E1,F0,30,F0,D2,E1,F0,F0,68
1260 DATA F0,D2,E1,F0,3C,F0,D2,E1,c1
1270 DATA B4,3C,78,D2,E1,3C,3C,3C,c6
1280 DATA D2,E1,C3,C3,C3,D2,50,F0,fa
1290 DATA F0,F0,A0,00,00,00,00,00,92
1300 DATA 10,30,30,30,20,61,C3,C3,95
1310 DATA C3,92,61,C0,C0,C0,92,61,dd
1320 DATA 60,C0,90,92,61,30,C0,30,7a
1330 DATA 92,61,30,30,92,61,30,a9
1340 DATA F0,30,92,61,70,F0,B0,92,73
1350 DATA 61,F0,F0,F0,92,61,C3,C3,e9
1360 DATA C3,92,10,30,30,30,20,00,d0
1370 DATA 00,00,00,00,14,3C,3C,3C,48
1380 DATA 28,69,C3,C3,C3,96,69,F0,79
1390 DATA F0,F0,96,69,78,F0,B4,96,7b
1400 DATA 69,3C,F0,3C,96,69,3C,3C,44
1410 DATA 3C,96,69,3C,C3,3C,96,69,b8
1420 DATA 69,C3,96,69,69,C3,C3,1d
1430 DATA 96,69,C3,C3,C3,96,14,3C,f1
1440 DATA 3C,3C,28,00,00,00,00,00,7d
1450 DATA 01,03,03,03,02,56,FC,FC,d3
1460 DATA FC,A9,56,C0,C0,C0,A9,56,22
1470 DATA 42,C0,81,A9,56,03,00,03,9b
1480 DATA A9,56,03,03,03,A9,56,03,fc
1490 DATA FC,03,A9,56,56,FC,A9,56,56
1500 DATA 56,FC,FC,FC,A9,56,FC,2d
1510 DATA FC,A9,01,03,03,03,02,00,51
1520 DATA 00,00,00,00,11,33,33,33,75
1530 DATA 22,67,CF,CF,CF,9B,67,F3,cc
1540 DATA F3,F3,9B,67,73,F3,B3,9B,70
1550 DATA 67,33,F3,33,9B,67,33,33,08
1560 DATA 33,9B,67,33,CF,33,9B,67,e4
1570 DATA 67,CF,9B,9B,67,CF,CF,ec
1580 DATA 9B,67,CF,CF,CF,9B,11,33,8a
1590 DATA 33,33,22,00,00,00,00,00,61
1600 DATA CF,CF,CF,CF,CF,CF,CF,7d
1610 DATA CF,CF,CF,CF,CF,CF,CF,al
1620 DATA CF,CF,CF,CF,CF,CF,CF,cd
1630 DATA CF,6D,9E,DA,DA,65,30,65,e8
1640 DATA 21,CF,DE,CF,0C,CF,CF,96,0c
```



```
1650 DATA CF,E5,CF,C5,9A,C5,CF,E5,bf
1660 DATA 9E,9E,9E,FO,DA,CF,9A,CF,2d
1670 DATA 65,47,47,ED,4D,EF,CF,9E,95
1680 DATA DA,9A,CA,9A,DA,65,9A,CF,f7
1690 DATA 9E,9E,9E,FO,DA,65,9A,CF,01
1700 DATA 65,47,47,ED,4D,EF,CF,DA,61
1710 DATA 9A,CA,9A,DA,CF,CF,C5,CF,dd
1720 DATA 9E,CF,9E,DA,CF,65,9A,CF,ac
1730 DATA 21,3C,5E,ED,4D,EF,CF,9A,1d
1740 DATA CA,9A,DA,9E,CF,9A,CF,CF,59
1750 DATA 9E,9E,9E,DA,CF,65,9A,CF,04
1760 DATA 65,47,47,ED,4D,EF,CF,CA,14
1770 DATA 9A,DA,9E,CB,CF,E5,9A,C5,b7
1780 DATA 9E,9E,9E,DA,DA,65,9A,CF,b1
1790 DATA 65,47,47,ED,0C,CF,CF,9A,54
1800 DATA CF,6D,CF,C7,9E,CF,DA,65,34
1810 DATA CF,CF,CF,CF,CF,CF,CF,CF,f8
1820 DATA CF,CF,CF,CF,CF,CF,CF,CF,c4
1830 DATA CF,CF,CF,CF,CF,CF,CF,CF,98
1840 DATA FC,FC,FC,FC,FC,FC,FC,FC,c8
1850 DATA FC,FC,FC,FC,FC,FC,FC,FC,00
1860 DATA FC,FC,FC,FC,FC,FC,FC,FC,40
1870 DATA 03,3C,3C,3C,3C,3C,3C,3C,4f
1880 DATA 3C,3C,3C,3C,29,57,03,57,2b
1890 DATA 03,4B,57,03,57,03,AB,03,07
1900 DATA 34,79,FF,FB,FF,FB,FF,FB,b4
1910 DATA FF,FB,FF,FB,3C,30,18,18,3e
1920 DATA 24,24,24,30,24,24,30,32
1930 DATA 40,40,48,0C,48,0C,48,48,82
1940 DATA 48,48,48,0C,0C,0C,0C,02e
1950 DATA C4,CC,C4,C0,C4,C8,C0,C0,b2
1960 DATA 34,F3,OC,59,59,F3,59,59,1a
1970 DATA 0C,F3,OC,F3,B6,30,18,18,3a
1980 DATA 24,24,24,30,24,24,30,32e
1990 DATA 16,33,33,BB,BB,33,BB,BB,65
2000 DATA BB,BB,BB,33,36,57,03,57,b8
2010 DATA 57,57,57,FF,57,03,AB,03,90
2020 DATA BC,39,FF,BB,FF,BB,FF,BB,25
2030 DATA BB,BB,FF,BB,3C,FC,FC,FC,6f
2040 DATA FC,FC,FC,FC,FC,FC,FC,FC,3f
2050 DATA 00,3C,3C,3C,3C,3C,3C,3C,db
2060 DATA 3C,3C,3C,3C,28,00,00,00,6b
2070 DATA 00,00,00,00,00,00,00,00,e3
2080 DATA FF,FF,FF,FF,FF,FF,FF,FF,3f
2090 DATA FF,FF,FF,FF,FF,FF,FF,FF,a3
2100 DATA FF,FF,FF,FF,FF,FF,FF,FF,of
2110 DATA 5C,5C,OC,AC,5C,OC,AC,AC,b7
2120 DATA 5C,FC,FC,FC,FC,FC,FC,FC,2f
2130 DATA FC,FC,FC,FC,FC,FC,FC,FC,oc,bf
2140 DATA 89,89,89,89,89,CC,46,46,58
2150 DATA FC,61,61,61,61,C3,61,C3,ab
2160 DATA 61,C3,61,C3,61,C3,74,EC,84
2170 DATA 3F,35,3A,30,3A,3F,30,3F,47
2180 DATA B8,34,34,34,34,30,34,30,b3
2190 DATA 34,34,34,34,34,30,30,BD,ef
2200 DATA 89,89,89,89,89,CC,46,46,b8
2210 DATA B8,70,FO,70,FO,70,FO,30,28
2220 DATA 70,70,FO,BO,70,BO,30,EC,78
2230 DATA 5C,5C,5C,5C,5C,AC,AC,AC,fo
2240 DATA B8,34,34,34,30,34,34,30,90
2250 DATA 34,34,34,34,34,30,30,AC,74
2260 DATA FF,FF,FF,FF,FF,FF,FF,FF,60
2270 DATA FC,61,61,61,61,C3,61,C3,13
2280 DATA 61,C3,61,61,61,C3,30,FD,70
2290 DATA 00,00,00,00,00,00,00,00,98
2300 DATA 54,FC,FC,FC,FC,FC,FC,FC,90
2310 DATA FC,FC,FC,FC,FC,FC,FC,FC,58
2320 DATA 90,30,30,30,30,30,30,30,b8
2330 DATA 30,30,30,30,30,30,30,30,c0
2340 DATA 30,30,30,30,30,30,30,70,d0
2350 DATA 70,B5,5D,B5,58,B5,5D,B5,16
2360 DATA 5D,B5,58,FO,FC,FO,FO,FC,01
2370 DATA FO,FO,FC,FO,FO,FO,FC,B4,b5
2380 DATA FO,A4,FO,A4,F5,A4,FO,A4,8e
2390 DATA FO,A4,F5,FO,FC,F8,FO,A9,71
2400 DATA F8,FO,B8,F8,FO,E8,F8,B4,59
2410 DATA FO,F5,FF,F5,F5,F5,FA,F5,11
2420 DATA FA,F5,F5,FO,FC,FC,FO,A9,d0
2430 DATA 56,FO,B8,74,FO,E8,D4,B4,22
2440 DATA FO,FO,F5,F5,FA,F5,FO,F5,9d
2450 DATA FO,F5,F5,FO,FC,FC,FO,A9,6a
2460 DATA 56,FO,B8,74,FO,E8,D4,B4,d4
2470 DATA FO,F5,A4,F5,FO,F5,FO,F5,4c
2480 DATA FO,F5,A4,FO,FC,F8,FO,A9,26
2490 DATA F8,FO,B8,F8,FO,E8,F8,B4,56
2500 DATA FO,F5,1D,A4,FO,F5,1D,F5,48
2510 DATA 1D,F5,58,FO,FC,FO,FO,FC,eb
2520 DATA FO,FO,FC,FO,FO,FC,FO,B4,17
2530 DATA B4,3C,3C,3C,3C,3C,3C,3C,e7
2540 DATA 3C,3C,3C,3C,3C,3C,3C,3C,47
2550 DATA 3C,3C,3C,3C,3C,3C,3C,69,17
2560 DATA 90,30,30,30,30,30,30,30,37
2570 DATA 30,30,30,30,30,30,30,30,ff
2580 DATA 30,30,30,30,30,30,30,E1,5f
```

```
2590 DATA 70,FO,FO,FO,FO,FO,33,CF,7c
2600 DATA 9B,33,F3,3F,F2,FO,FO,50,dc
2610 DATA AO,AO,FO,FO,FO,FO,50,69,1c
2620 DATA 70,FO,FO,FO,FO,B1,67,CF,4b
2630 DATA 33,73,B7,7B,B3,FO,FO,00,5c
2640 DATA AO,AO,FO,FO,FO,FO,69,b4
2650 DATA 70,FO,FO,7A,FO,B1,8A,00,a0
2660 DATA 00,00,00,00,33,FO,FO,00,1f
2670 DATA AO,AO,50,AO,FO,AO,FO,69,of
2680 DATA 70,FO,B5,F2,FO,33,8A,00,4e
2690 DATA 00,00,00,00,67,FO,FO,50,69
2700 DATA 00,AO,FO,AO,FO,AO,FO,69,b1
2710 DATA 70,FO,6A,F2,FO,67,8A,00,5f
2720 DATA 00,00,00,00,67,FO,FO,50,92
2730 DATA 00,AO,FO,50,50,AO,FO,69,92
2740 DATA 70,B5,D1,B3,33,67,8A,00,6a
2750 DATA 00,00,00,00,CF,FO,FO,50,bd
2760 DATA AO,AO,50,50,50,AO,FO,69,95
2770 DATA 70,6A,F3,F3,F3,CF,8A,00,85
2780 DATA 00,00,00,00,CF,FO,3C,3C,64
2790 DATA 3C,3C,3C,3C,3C,3C,69,f4
2800 DATA 35,33,33,33,33,CF,8A,00,53
2810 DATA 00,00,00,00,CF,B4,C3,C3,cb
2820 DATA C3,C3,C3,C3,C3,C3,86,1f
2830 DATA 70,E7,9B,CF,CF,CF,8A,00,ed
2840 DATA 00,00,00,00,CF,69,OC,OC,02
2850 DATA OC,OC,OC,OC,OC,OC,OC,9a
2860 DATA 70,F1,CF,33,67,CF,22,00,e0
2870 DATA 00,00,00,00,CF,69,OC,OC,0d
2880 DATA 4C,8C,OC,OC,4C,OC,OC,0c,3d
2890 DATA 70,FO,67,72,FO,CF,22,00,0a
2900 DATA 00,00,00,00,CF,69,OC,OC,4f
2910 DATA 8C,4C,4C,OC,4C,8C,OC,OC,17
2920 DATA 70,FO,B1,72,FO,9B,22,00,a2
2930 DATA 00,00,00,00,CF,69,OC,OC,ff
2940 DATA 8C,4C,4C,OC,4C,OC,OC,9b
2950 DATA 70,FO,FO,DA,FO,B1,22,00,1f
2960 DATA 00,00,00,00,CF,69,OC,OC,94
2970 DATA CC,CC,48,48,48,OC,OC,9c
2980 DATA 70,FO,FO,FO,FO,B1,33,73,9f
2990 DATA B7,3F,F3,33,CF,69,OC,OC,06
3000 DATA 8C,4C,48,48,48,OC,OC,0e
3010 DATA 70,FO,FO,FO,FO,33,F3,7b
3020 DATA 3F,7B,B3,67,9E,69,OC,OC,15
3030 DATA 8C,48,48,48,48,OC,OC,d5
3040 DATA B4,3C,3C,3C,3C,3C,3C,3d
3050 DATA 3C,3C,3C,3C,3C,86,OC,OC,21
3060 DATA OC,OC,OC,OC,OC,OC,OC,61
3070 DATA 00,CO,00,31,03,00,00,00,4c
10 MODE 0:BORDER 0:FOR a=0 TO 15:READ c:
INK a,c:NEXT
15 DATA 0.26,2.14,24.15,6.9,18.3,5.4,8.1
7,23,1
20 DEFINT a-z
30 FOR a=0 TO 150 STEP 2:MOVE 0,a,5:DRAW
R 640,0:NEXT:FOR a=150 TO 400 STEP 2
40 MOVE 0,a:DRAW 640,0,2:NEXT:FOR a=1 T
O 80 STEP 2:FOR b=0 TO 320 STEP a
50 PLOT b-5,146+a*4,3:PLOT 638-b+5,146+a
*4:PLOT b,152-a*4,4
60 PLOT 638-b,152-a*4:NEXT:NEXT
70 A=148:b=1:WHILE a>0:b=b+1:MOVE 0,a:DR
AWR 640,0,6:a=a-b:WEND
80 a=0:b=1:WHILE a<320:b=b+2:MOVE a,148:
DRAW a*2-320,0:MOVE 638-a,148
90 DRAW 638-a*2+320,0:a=a+b:WEND
100 A=148:b=1:WHILE a>0:b=b+1:MOVE 0,a:D
RAWR 640,0,6:a=a-b:WEND:a=0:b=1
110 WHILE a<320:b=b+2:MOVE a,148:DRAW a*
2-320,0:MOVE 638-a,148
120 DRAW 638-a*2+320,0:a=a+b:WEND
130 RESTORE 140:b=0:FOR a=198 TO 214 STE
P 4:READ c:PLOT a,340,c
140 DRAW 0,-340+b:PLOT 630-a,340:DRAW 0,
-340+b:b=b+3:NEXT:DATA 7,8,4,5,6
150 RESTORE 160:b=0:FOR a=10 TO 0 STEP -
2:READ c:PLOT 216-b,a,c
160 DRAW 198+b*2,0:b=b+4:NEXT:DATA 15,2
,3,3,14,14
170 FOR a=12 TO 340 STEP 2:MOVE 216,a,0:
DRAW 198,0:NEXT
180 TAG:GRAPHICS PEN 1,1:b=10:FOR a=120
TO 200 STEP 2:b=b+1:GOSUB 330:NEXT
190 b=10:FOR a=194 TO 100 STEP -2:b=b+1:
GOSUB 330:NEXT
200 b=70:FOR a=90 TO 220 STEP 2:b=b+0.8:
GOSUB 330:NEXT
210 a=140:FOR b=100 TO 150 STEP 5:a=a-1:
GOSUB 330:NEXT:b=136
220 FOR a=220 TO 74 STEP-4:b=b+1:GOSUB 3
30:NEXT:b=464:FOR a=220 TO 80 STEP -2
230 GOSUB 330:NEXT:FOR a=80 TO 90 STEP 1
:b=b+6:GOSUB 330:NEXT
240 b=550:FOR a=100 TO 200 STEP 2:GOSUB
330:NEXT:FOR a=150 TO 190 STEP 2
250 b=b+2:GOSUB 330:NEXT:b=552:FOR a=150
```

```
TO 110 STEP-2:b=b+2:GOSUB 330:NEXT
260 FOR a=188 TO 444 STEP 4:MOVE a,400.5
,0:DRAW 0,-58:NEXT
270 PLOT 444,400,6:DRAW 0,-58:DRAW -68
,0:DRAW 0,22,4:DRAW -38,0,6
280 DRAW 0,22,4:DRAW -44,0,1:DRAW 0,-
22,4:DRAW -40,0,6:MOVER 0,-2
290 DRAW 0,-20,4:MOVER -4,0:DRAW -60,0
,6:DRAW 0,26
300 TAG:MOVE 216,362,1,1:PRINT CHR$(222)
:MOVE 386,362,1,1:PRINT CHR$(223):
310 MOVE 256,382,1,1:PRINT CHR$(222):MO
VE 344,382,1,1:PRINT CHR$(223):
320 SAVE"walk.ecr".b,&C000,&4000,&C000:E
ND
330 MOVE b,a,0,0:PRINT CHR$(46):RETURN
10 'createur du fichier de sauvegardes
XALK.SVG'
20 DIM hisc(20),nom$(20)
30 FOR A=1 TO 20:hisc(a)=21000-a*1000:NO
MS(21-A)=CHR$(15)+""+LEFT$(MID$(XALK
"/AMSTRAD 100% & Antman / XAL
K",21-A,A/2+31),30):NEXT
40 MODE 1:FOR a=1 TO 20:LOCATE 2,a+1:PE
N 3:PRINT USING"##":a:LOCATE 5,a+1:PE
N 1:PRINT DEC$(hisc(a)),"#####":LOCATE 11,a+
1:PRINT LEFT$(nom$(A),30):NEXT
50 OPENOUT"walk.svg":FOR a=1 TO 20:PRINT
#9,hisc(a):PRINT#9,nom$(a):NEXT:CLOSEOUT
1 ENV 1,50,-2,5:ENT 1,55-20,30:ENV 2,1,-
20,5:ENT 2,4,-15,20:ENT 5,20,32,16
2 ENV 5,20,-10,16:BORDER 0:MODE 0:PAPER
0:PEN 1:DEFINT a-z:DEFREAL h:DEFREAL s
3 MEMORY &4FFF:LOAD"walk.spr",&5000:LOAD
"walk.bin",&9000
4 RANDOMIZE TIME:RESTORE 5:FOR a=0 TO 15
:READ c:INK a,c:NEXT:POKE &9045,5
5 DATA 0,26,2,14,24,15,6,9,18,3,5,0,0,0,
23,1
6 AS="2000200200200200200200200200200200200
20020020010100202000101003030001010"
7 AS=a$+"0003000330030030030000055055050
0550000404005050500404000606004040400"
8 AS=a$+"6060060006060606060606060606060606060
606060000000000000000000"
9 FOR y=0 TO 24 STEP 12:FOR b=0 TO 75 ST
EP 5:CALL &9000,11,b,y,&51A4:NEXT:NEXT
10 X=0:Y=12:FOR A=1 TO LEN(AS):c=VAL(MID
$(AS,A,1)):IF c=0 THEN 13
11 FOR B=120 TO Y STEP -12:CALL &9000,11
,X*4,7,B,&4FC4+c*60
12 CALL &9000,11,X*4,7,B+12,&521C:NEXT:S
OUND 4,50,7,13,1,1,1
13 X=X+1
14 IF X>17 THEN X=0:Y=Y+12:FOR b=0 TO 75
STEP 5:CALL &9000,11,b,Y,&51A0:NEXT
15 NEXT:POKE &9045,24:FOR a=0 TO 56 STEP
28:FOR b=152 TO 168 STEP 16
16 CALL &9000,8,a,b,&5300:NEXT:NEXT:POKE
&9045,5
17 POKE &B72F,231:LOCATE 1,24:PRINT CHR$(
164)"1990 ANTMAN 100%"
18 FOR A=1 TO 30:INK 11,0:INK 12,0:INK 1
3,0:FOR T=0 TO 800 STEP A:NEXT
19 INK 11,4:INK 12,8:INK 13,17:CALL &BD1
9:NEXT:SOUND 4,200,30,15,2,2,20
20 SOUND 4,100,30,15,1,5,10
21 DIM hisc(25),nom$(25):OPENIN"walk.svg"
22 FOR a=1 TO 20:INPUT#9,hisc(a):INPUT#9
,nom$(a):NEXT:CLOSEIN
23 SPEED KEY 8,1:MODE 1:hisc=hisc(1):GOS
UB 118
24 CLEAR INPUT:LOCATE 1,24:PAPER 2:PEN 1
:PRINT"Votre niveau :":CHR$(18)
25 PRINT"1:Debutant 2:Moyen 3:Bon 4
:Expert":CHR$(18):
26 a$=INKEY$:IF a$<"1" OR a$>"4" THEN 26
27 SOUND 1,50,20,14,1,1,0:PRINT a$:niv=V
AL(a$):CLEAR INPUT
28 LOCATE 1,24:PRINT"Vitesse de depart :
":CHR$(18):PRINT"de 0 a 200 =":CHR$(18):
29 INPUT temp:IF TEMP>200 OR TEMP<0 THEN
28 ELSE tempo=200-temp
30 LOCATE 1,24:PRINT CHR$(18):PRINT"
-J-oystick ou -C-urseur ?"
31 a$=UPPER$(INKEY$):IF a$<"J" AND a$<
"C" THEN 31
32 IF a$="J" THEN k1=8:k2=9:k3=10 ELSE k
1=242:k2=243:k3=241
33 PAPER 0:MODE 0:GOSUB 128
34 DIM t(6,25):lmax=0:CLEAR INPUT:CALL &
BB06:score=0
35 ol=INT(RND*(4+niv))+1:LOCATE 15,24:PR
INT DEC$(hisc,"#####")
36 CALL &9000,11,63,11,&4FC4+ol*60
37 CALL &BB06:ol=ol:ol=INT(RND*(4+niv))+1
:CALL &9000,11,63,11,&4FC4+ol*60
```

```
38 x=3:y=17:GOSUB 52:FOR t=1 TO 300:NEXT
39 GOSUB 51:y=y-1:GOSUB 52
40 FOR t=1 TO tempo:NEXT
41 AS=INKEYS
42 IF AS=CHR$(k1) THEN IF x>1 AND t(x-1,
y)=0 THEN GOSUB 51:x=x-1:GOSUB 52
43 IF AS=CHR$(k2) THEN IF x<5 AND t(x+1,
y)=0 THEN GOSUB 51:x=x+1:GOSUB 52
44 IF AS=CHR$(k3) THEN GOSUB 48
45 IF t(x,y-1)<0 OR y=1 THEN GOSUB 47:G
OSUB 60:GOTO 37
46 GOTO 39
47 SOUND 1,60,20,15,1,1,1:score=score+n1
v:GOSUB 102:GOSUB 55:t(x,y)=0:RETURN
48 GOSUB 51:FOR yy=y TO 1 STEP -1
49 IF (t(x,yy)=0 AND t(x,y-1)<0) OR yy
=1 THEN y=yy:yy=1
50 NEXT:GOSUB 52:RETURN
51 CALL &9000,11,x*5+22,194-11*y,&521C:R
ETURN
52 CALL &9000,11,x*5+22,194-11*y,&4FC4+o
*60:RETURN
53 CALL &9000,11,k*5+22,194-11*1,&4FC4+o
*60:RETURN
54 CALL &9000,11,k*5+22,194-11*1,&521C:R
ETURN
55 IF y>lmax THEN lmax=y
56 IF y=16 THEN GOSUB 58:FOR tm=1 TO 800
0:NEXT:GOSUB 104:ERASE t:GOTO 23
57 RETURN
58 SOUND 4,100,320,15,5,5,0:SOUND 1,550,
320,15,5,0,0:SOUND 2,140,320,13,0,5,0
59 RETURN
60 GOSUB 62:IF y>2 THEN GOSUB 72 ELSE IF
ef=1 THEN ef=0:GOSUB 77
61 RETURN
62 ef=0:FOR a=1 TO 5:m(a)=0:NEXT
63 FOR n=0 TO 2
64 FOR xx=1 TO 3
65 oo=t(xx,y):IF ef=1 AND xx=mxx THEN oo
=m(xx)
66 IF oo<>0 THEN IF t(xx+1,y)=oo AND t(x
x+2,y)=oo THEN ef=1:GOSUB 68
67 NEXT xx:NEXT n:RETURN
68 GOSUB 99:x1=xx:x2=xx+2:y1=y:t(xx,y1)
=0:k=xx+1:l=y:GOSUB 54:m(xx+1)=oo
69 t(xx,y)=0:k=xx:GOSUB 54:m(xx)=oo:m(xx
)=oo:t(xx+2,y)=0:mxx=x2:k=x2
70 GOSUB 54:m(x2)=oo
71 RETURN
```

```
72 oo=t(x,y):IF oo=0 THEN IF m(x)<>0 THE
N oo=m(x) ELSE 74
73 IF t(x,y-1)=oo AND t(x,y-2)=oo THEN G
OSUB 76
74 IF ef=1 THEN ef=0:GOSUB 77
75 RETURN
76 GOSUB 99:FOR yy=y TO y-2 STEP -1:t(x,
yy)=0:k=xx:l=yy:GOSUB 54:NEXT:RETURN
77 FOR xx=1 TO 5:oo=t(xx,y1):IF oo=0 AND
t(xx,y1+1)<>0 THEN GOSUB 83
78 NEXT:FOR xx=1 TO 3
79 oo=t(xx,y):IF oo<>0 THEN IF t(xx+1,y)
=oo AND t(xx+2,y)=oo THEN ef=1:GOSUB 68
80 NEXT:IF ef=1 THEN ef=0:GOTO 77
81 IF y>1 AND y>1 THEN GOSUB 88:IF ef=1
THEN ef=0:xx=mxx:GOSUB 96
82 RETURN
83 FOR yy=y1 TO lmax:oo=t(xx,yy+1):IF oo
<>0 THEN GOSUB 86
84 NEXT:t(xx,y2)=0
85 RETURN
86 t(xx,yy)=oo:y2=yy+1:SOUND 1,20,10,1,1
,5:k=xx:l=yy:GOSUB 53:l=yy+1
87 GOSUB 54:RETURN
88 my=0:FOR xx=1 TO 5:oo=t(xx,y1):IF oo=
0 THEN 91
89 IF t(xx,y1+1)=oo AND t(xx,y1)=oo TH
EN ef=1:GOSUB 99:GOSUB 92:GOTO 91
90 IF y>1 THEN IF t(xx,y1)=oo AND t(x
x,y1-2)=oo THEN ef=1:GOSUB 94
91 NEXT:RETURN
92 FOR yy=y1-1 TO y1+1:t(xx,yy1)=0:k=xx
:l=yy1:GOSUB 54:NEXT:mxx=xx:my=-1
93 RETURN
94 GOSUB 99:FOR yy=y1-2 TO y1:t(xx,yy1)
=0:k=xx:l=yy1:GOSUB 54:NEXT:mxx=xx
95 my=-2:RETURN
96 FOR yy=y1+my TO lmax-3:oo=t(xx,yy+3):
t(xx,yy)=oo:t(xx,yy+3)=0:k=xx:l=yy+3
97 GOSUB 54:IF t(xx,yy)<0 THEN SOUND 1,
20,15,1,5,1,5:l=yy:GOSUB 53
98 NEXT:RETURN
99 SOUND 1,100,20,15,1,2:score=score+(5*
niv):tempo=tempo-1
100 IF tempo<=0 THEN tempo=0
101 GOSUB 102:RETURN
102 LOCATE 1,24:PRINT DEC$(score,"#####
"):LOCATE 1,1
103 PRINT DEC$(200-tempo,"#####"):RETUR
N
```

```
104 SPEED KEY 30,2:CLEAR INPUT:IF SCORE<
=0 THEN RETURN
105 MODE 1
106 FOR a=1 TO 20:IF score>hisc(a) THEN
GOSUB 109
107 NEXT:PEN 1:IF score=0 THEN GOSUB 112
108 RETURN
109 FOR C=20 TO a STEP-1:hisc(c+1)=hisc(
c):nom$(c+1)=nom$(c):NEXT
110 hisc(a)=score:nom$(a)=STRING$(29,"
"):score=0:b=a:GOSUB 118
111 RETURN
112 PEN 3:LOCATE 1,24:PRINT"Entrez votre
nom":WINDOW 11,39,b+1,b+1:INPUT" ",nom$
113 IF LEN(nom$)>29 THEN 112 ELSE WINDOW
1,40,1,25:nom$(b)=nom$:GOSUB 125:PEN 1
114 LOCATE 1,25:PRINT"Deprotegez la disq
uette ...":CALL &BB06:CALL &BB06
115 IERA,"walk.svg":OPENOUT"walk.svg":FO
R a=1 TO 20:PRINT#9,hisc(a)
116 PRINT#9,nom$(a):NEXT:CLOSEOUT
117 RETURN
118 MODE 1:PEN 1:PRINT CHR$(150)CHR$(154
)CHR$(154)CHR$(158)STRING$(5,154):
119 PRINT CHR$(158)STRING$(29,154)CHR$(1
56)
120 LOCATE 15,23:PRINT"SCORE: ";DEC$(sco
re,"#####")
121 PEN 1:FOR a=2 TO 21:LOCATE 1,a:PRINT
CHR$(149):LOCATE 4,a:PRINT CHR$(149)
122 LOCATE 10,a:PRINT CHR$(149):LOCATE 4
0,a:PRINT CHR$(149):NEXT
123 LOCATE 1,22:PRINT CHR$(147)STRING$(2
,154)CHR$(155)STRING$(5,154)CHR$(155):
124 PRINT STRING$(29,154)CHR$(153)
125 FOR a=1 TO 20:LOCATE 2,a+1:PEN 3:PRI
NT USING"##":a:LOCATE 5,a+1:PEN 1
126 PRINT DEC$(hisc(a)),"#####":LOCATE 1
1,a+1:PRINT LEFT$(nom$(A),30):NEXT
127 RETURN
128 LOAD"walk.ecr"
129 POKE &9045,24:CALL &9000,8,56,0,&530
0:CALL &9000,8,0,192,&53C0
130 CALL &9000,8,0,0,&5300:CALL &9000,8,
56,192,&5480
131 CALL &9000,8,0,8,&5540:CALL &9000,17
,56,8,&5600
132 POKE &9045,5:FOR a=2 TO 4:CALL &9000
,11,a*5+22,18,&521C:NEXT
133 CALL &9000,11,37,7,&521C:RETURN
```