CÓMO SE HACE UN JUEGO. OGEROX (I)

No cabe duda de que una de las mayores inquietudes de los aficionados al Spectrum pasa por la creación de su propio juego. En esta serie de artículos que hoy iniciamos, os trataremos de orientar en esta tarea e iremos analizando uno a uno los aspectos fundamentales que intervienen en la programación de un juego.

Hert no se podía imaginar con lo que se iba a encontrar cuando se presentó como voluntario a la arriesgada misión de encender el gran fuego. Según la leyenda, el campamento vivió épocas de gran esplendor cuando el fuego con su calor retenía a los males...

Pero nadie reparó en aquel día en que unos vientos huracanados se acercaron al valle del campamento y apagaron el gran fuego, quedando los males libres. El desastre se adueñó de aquel inofensivo campamento durante mucho tiempo. Un día, un joven proviniente de la gran ciudad se acercó al campamento y se ofreció voluntario para encender de nuevo el gran fuego. En él se pusieron todas las esperanzas... Ahora, ese joven llamado Hert eres tú y tienes que encender el gran fuego, porque si no, bien podrías ser pasto de las llamas del mal.

Ésta es una pequeña introducción al argumento de un juego que vamos a desarrollar a lo largo de esta serie que hemos dividido en cinco partes. Cada uno de los artículos está dedicado a un apartado específico del juego que se puede ejecutar independientemente para ir viendo cómo avanza su construcción. Hay algunos detalles que cada uno puede particularizar a su gusto y una pantalla final que se ha dejado libre para que cada cual introduzca allí la suya propia. Ni qué decir tiene que con las cinco partes completas tendremos el juego acabado y listo para funcionar. Os podemos asegurar que, si bien el juego no es de la calidad equivalente a los juegos que salen últimamente, sí es lo bastante bueno como para una línea de software barato.

Cada una de las partes se compone de diversos bloques en Código Máquina, de un cargador y de un programa de demostración. Los bloques en Código Máquina se encuentran en forma de listados hexadecimales que se deben introducir con el cargador universal. Los cargadores son listados Basic cortos que se encargan de mostrar rutinas o partes del juego y que necesitan del Código Máquina del artículo en que aparecen y de todos los anteriores. Los cargadores de cada parte se deben ir mezclando, de forma que el último se componga de él mismo y de todos los anteriores.

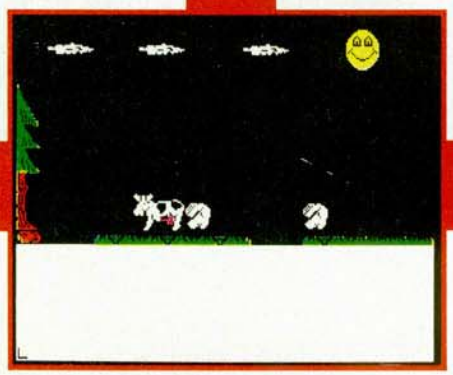
LA RUTINA MAPEADORA

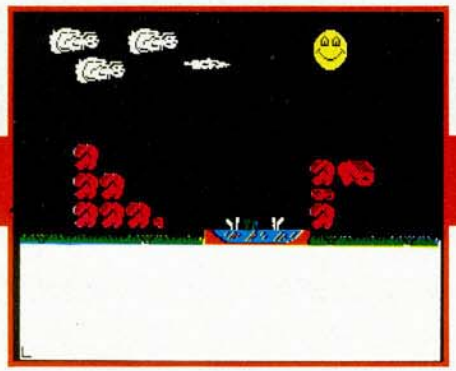
En esta primera parte se incluye la rutina mapeadora (el trozo de Código Máquina que imprime las pantallas del juego) y las seis primeras pantallas.

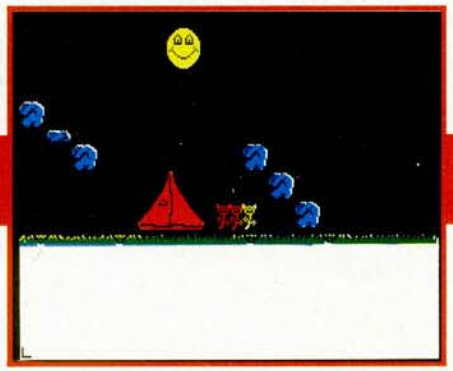
Para todos aquellos interesados en saber cómo funciona la rutina mapeadora y para los que quieran cambiar gráficos o pantallas, vamos a explicar de forma global el funcionamiento de esta primera rutina, que resulta imprescindible en todo juego.

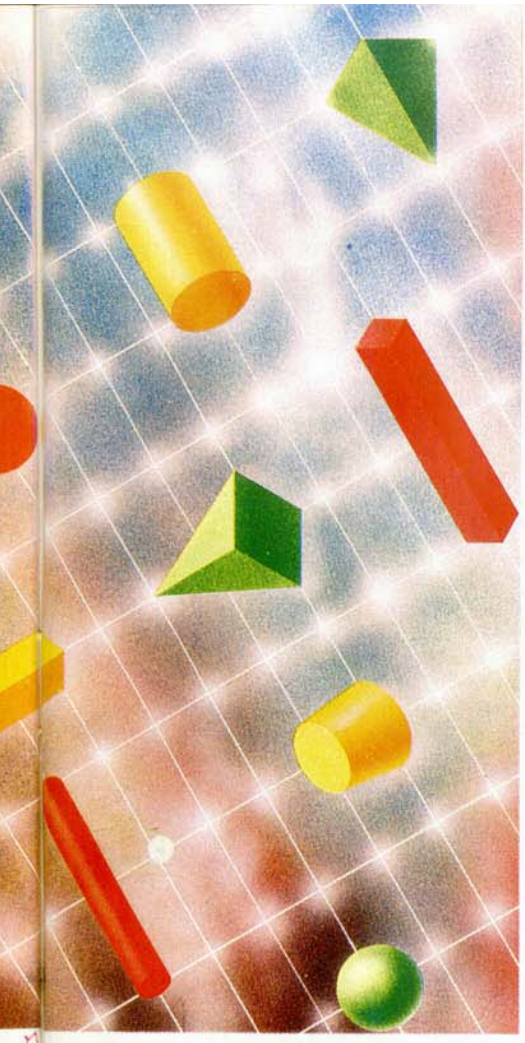
Para empezar, la definición de las pantallas del juego comienza a partir de la dirección definida por la variable de dos bytes que está a partir de la dirección de memoria 62476. Todas las variables se organizan de la misma manera: primero el byte bajo y luego

Estas son las pantallas que podréis contemplar tras haber tecleado todos los listados y ejecutar la Demo.









al alto (low-high). Para todos aquéllos que no se hayan familiarizado todavía con el manejo de variables de esta manera explicaremos brevemente cómo averiguar o cambiar su valor.

Si suponemos que una de estas variables está almacenada a partir de la dirección *dirección:*

Para conocer su contenido desde Basic utilizaremos:

PRINT PEEK *dirección* + 256*PEEK (*dirección* + 1)

Y si queremos cambiar su valor a *valor:*

POKE *dirección*, *valor*—INT (*valor*/256)*256

POKE *dirección* + 1,INT (*valor*/256)

Así, para saber a partir de qué dirección de memoria se encuentran definidas las pantallas utilizaremos:

PRINT PEEK 62476 + 256*PEEK 62477

Y si quisiéramos que su definición comenzara en la dirección 31000 —suponiendo que desde la dirección 31000 hubiera o fuéramos a poner una definición válida— la cambiaríamos mediante:

POKE 62476,31000—INT (31000/256)*256

POKE 62477,INT (31000/256)

Toda pantalla está formada por gráficos cuyo tamaño puede ser el de un carácter o un múltiplo de éstos, tanto a lo ancho como a lo alto. Estos gráficos pueden tener un solo color (atributo) para todo el gráfico, o bien un atributo por cada carácter del gráfico. Las direcciones de todos los gráficos de las pantallas se encuentran en una tabla cuya dirección se define mediante la variable que se encuentra en la dirección 62566 (y ya sabemos cómo averiguar o cambiar su valor). Esta tabla contiene las direcciones de definición de cada gráfico, que se almacenan en el mismo formato que las variables anteriores. De esta manera, para conocer la dirección donde está definido el primer gráfico utilizaríamos las dos primeras direcciones de la tabla, para el segundo las dos siguientes, etc.

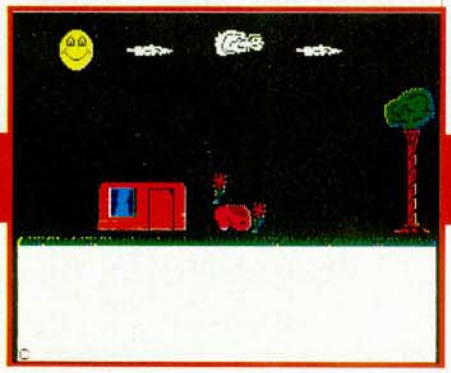
La definición de un gráfico es ligeramente distinta si el gráfico tiene tantos atributos como caracteres o si no los tiene. En ambos casos, el primer byte de la definición contiene el número de caracteres de alto del gráfico (formato vertical) mientras que el segundo contiene el número de caracteres de ancho (formato horizontal). A continuación viene la definición del gráfico por caracteres. Los ocho primeros bytes definen el carácter que está más arriba y a la izquierda; los ocho siguientes el carácter inmediatamente a la derecha; así hasta completar la primera fila de caracteres. Después viene la segunda fila, de la misma forma que la primera; después la tercera, etc. Si el gráfico fuera de una sola fila de alto o de una sola columna de ancho, sólo tendría las definiciones de los caracteres correspondientes, pero siempre de izquierda a derecha y de arriba a abajo. En el caso de que el gráfico no fuera de los *n* primeros que tienen un único atributo, a continuación vendrían los atributos ordenados de la misma forma que las definiciones de los caracteres.

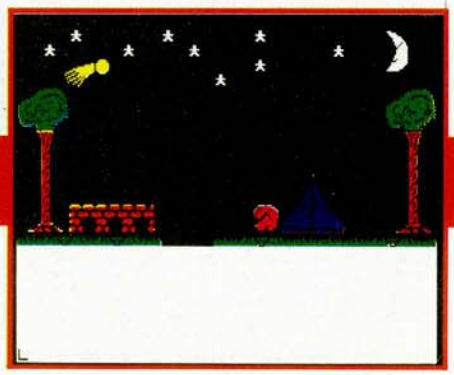
En el juego, son los diez primeros gráficos los que tienen un único atributo, mientras que los siguientes tienen uno por carácter. Podemos cambiar el número de gráficos con un solo atributo a los *n* primeros (que se enumeran desde 0 hasta *n*—1). Para hacerlo sólo tenemos que variar el contenido de la dirección 62550 mediante: POKE 62550,*n*

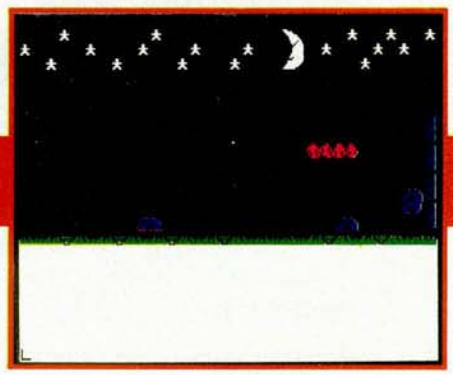
La definición de cada pantalla se hace poniendo el número de gráfico que tenga que haber en cada posición en una dirección de memoria, teniendo en cuenta que el primero es el cero. Si el gráfico es uno de los *n* primeros, le sigue el atributo que le debe corresponder en la siguiente dirección de memoria. En la primera dirección, se encuentra el número del primer gráfico, cuya posición en pantalla corresponderá con la esquina superior izquierda. Si es de los diez primeros (numerados del 0 al 9) en la siguiente dirección estará el atributo que le corresponda. En la siguiente dirección está el número del gráfico que se colocará a continuación del primero, en la misma línea de pantalla, pero *x* columnas más a la derecha (siendo *x* el ancho en caracteres del primer gráfico). A continuación, vendrá el número del siguiente gráfico (puede que antes esté el atributo del anterior) y luego el siguiente hasta completar la pantalla.

El fin de la pantalla se indica poniendo el número de gráfico como 255, máximo númer almacenable en una dirección. De esta forma se termina la definición de esta pantalla y puede comenzar la de la siguiente, que en caso de existir lo hace justo a continuación.

En una pantalla los gráficos se suceden los unos a los otros y al alcanzar el final de una línea se pasa a la siguiente. Pero en el caso de que los gráficos no se sucedan exactamente o de que haya que dejar espacios, se utiliza el código 254 como número de gráfico y en la siguiente dirección se pone el número de espacios a dejar, entre 0 y 255. Para saltar a la línea siguiente y seguir en la misma columna se deben dejar 32 espacios —que son los espacios que caben en una línea— de forma que si el número de espacios sobrepasa a los que caben en una línea se pasa a la siguiente automática-







mente. No se puede dejar un número negativo de espacios, por lo que un gráfico siempre tiene que estar más a la izquierda y en la misma línea o más arriba que el que le sigue. En el caso de que hubiera que dejar más de 255 espacios, hay que volver a repetir el código 254 e indicar a continuación los espacios que faltan (si hicieran falta aún más, se haría la misma operación dejanto antes otros 255 espacios).

Cada pantalla se define por tanto de izquierda a derecha y de arriba a abajo, dejando espacios con 254 y terminando con 255. Sin embargo, la rutina que se encarga de imprimir las pantallas no comprueba si un gráfico se encuentra entre dos líneas (está en una línea) o si se sale de la pantalla. Es responsabilidad de cada uno el encargarse de que los gráficos estén todos en los límites de la pantalla, ya que si están entre dos líneas se verán deformados y si se salen por debajo pueden provocar que el ordenador «se cuelgue». Todas las pantallas del juego tienen 32 caracteres de ancho y 16 caracteres de alto, pero nada impide que el alto no llegue hasta 24 si utilizamos la rutina para imprimir pantallas hechas por nosotros. Sin embargo, si estas pantallas van a formar parte del juego, no deben sobrepasar las 16 líneas de altura porque darían problemas al borrarse y con los marcadores.

Para poder ver la demostración hay que copiar todos los bloques en hexadecimal e irlos salvando al cassette unos a continuación de otros, teniendo cuidado de hacerlo en el mismo orden en que aparecen en la revista y con los nombres que se indican. Hecho esto podemos teclear el programa cargador y salvarlo mediante:

SAVE "cargador 1" LINE 10

Ahora ya podemos teclear el programa de demostración, que debemos salvar después de los bloques de Código Máquina mediante:

GO TO 9999

Cada vez que queramos ver la demostración, rebobinaremos la cinta hasta donde esté el programa cargador y lo cargaremos (mediante LOAD ""). El programa se ejecutará y cargará a su vez los bloques en Código Máquina y el programa de demostración que también se ejecutará. Es conveniente dejar un espacio en la cinta entre el cargador y los bloques de Código Máquina. Para sacar otra copia del cargador y los bloques de Código Máquina podemos ejecutar el programa cargador, pero esta vez desde la línea 9010 mediante:

GO TO 9010

El programa de demostración, una vez ejecutado, nos pedirá un número —que puede variar desde 0 hasta 5— y nos mostrará la pantalla correspondiente.

Con esto termina esta primera parte con la que se pueden ver ya las seis primeras pantallas del juego. En el próximo número analizaremos la rutina de sprites, que se encarga del movimiento de los gráficos del personaje principal y de los enemigos, y donde podremos ver otras cuantas cosas más. Ánimo y hasta la próxima.

Alberto Elices
Roberto Oliva
Javier Elices

CARGADOR 1

```
  10 CLEAR 31299: LOAD "TABLA.BI
N"CODE 65310,110: LOAD "GRA_OGER
"CODE 50300,3514: LOAD "PANTS"CO
DE 42100,391: LOAD "ART_1-IP"COD
E 62369,859: LOAD "CRE_TABS"CODE
 56320,83
  20 LOAD "DEMO1"
9000 STOP
9010 SAVE "CARGADOR1" LINE 10: S
AVE "TABLA.BIN"CODE 65310,110: S
AVE "GRA_OGER"CODE 50300,3514: S
AVE "PANTS"CODE 42100,390: SAVE
"ART_1-IP"CODE 62369,859: SAVE "
CRE_TABS"CODE 56320,83
```

TABLA.BIN

```
  1  7CC41EC530C53AC55EC5  1338
  2  70C592C5C4C5DEC5F0C5  1901
  3  FAC5F8C642C78CC7B2C7  1874
  4  05C83DC848C880C8A6C8  1432
  5  CCC804C92AC93EC952C9  1398
  6  8AC99EC9B2C920CA46CA  1583
  7  6CCA80CA8BCA41CBABCC  1624
  8  19CD3FCDADCDD3CDDECD  1719
  9  E9CD7BCE0DCF33CF55D0  1538
  0  7BD036D292D248D385D4  1579
  1  F3D461D544D608D8FC00  1523
```

DUMP: 40.000
N.° BYTES: 110

GRA_OGER

```
  1  040500000000000000000  9
  2  000000000000000001818  48
  3  182C2C2C6E6E00000000  376
  4  0000010103070F0EEFEF  519
  5  EFE7F7F7F7F700008080  1714
  6  C0E0F0F0000000000000  896
  7  000000000000000010307  11
  8  1D3B707FFFFFFFF77B7  1649
  9  37F7F7F7F7E7F8FCFEFE  2282
 10  FFFFFFFF000000000080  1148
 11  C0E00F1F3F7FFF807F7F  1289
 12  FFFFFFFFFF3F80FFEFEF  2199
 13  EFEFEFD007FFFFFFFFFF  2207
 14  FF0FE0FFF0F8FCFEFFFF  2253
 15  02FE0201000081DBA55A  862
 16  423C0848A8521D123060  647
 17  0101001C6E567E760E00  484
 18  0102003F536D775F1F00  503
 19  00A8D4EEF6C63C000102  1125
 20  003F536D775F1F0000A8  668
```

```
 23  D4EEF6C63C0002010010E  987
 24  265E5E32760E3E7A5A5E  776
 25  66767E00020200 1E2B57  510
 26  AFDEBC7900E0F8DCBE7E  1714
 27  BEBE13071F3F7E3F1C00  717
 28  DEFCFCFCBC7870000203  1403
 29  00011F3F7F7F7F7F00AA  773
 30  D5EAF5FAFDFF00A070B8  1906
 31  5CAE54AA7F3F1F1F0F0F  802
 32  07007FBFDFE7DF8F0300  1148
 33  FEFE9E0EFEFCF8000103  1438
 34  187CFEBBBA5D5B3E187C  1169
 35  BABABA7D3D0E383C7EFE  1254
 36  BABBBA7C020130780C7C  1166
 37  BA9A7C38787C5EDD5D5A  1262
 38  7A26010130780C7CBA9A  982
 39  7C380407FFFFFFFFFFF0  1706
 40  E0E0FFFFFFFFFF000000  1723
 41  FFFFFFFFFF000000FFFF  1785
 42  FFFFFF000000FFFFFFFF  1785
 43  FF000000FFFFFFFFFF00  1530
 44  0000FFFFFFFFFF070301  1286
 45  C0C0C0C0C0C0C0C0FF83  1922
 46  838783878787FFF9F1F9  1796
 47  F1F1E1E1000000000000  932
 48  0000FF8080808080808080  1151
 49  FF0703010101010101010  272
 50  0101010101010C0C0C0C0  774
 51  C0C0C0C08387878F8F8F  1598
 52  8FFFF1F1F1F1F1F1F1FF  2340
 53  0000000000000000808080  256
 54  8080808080808001010101  772
 55  0101010101010101010101  10
 56  0101C0C0C0C0808080808080FF  1345
 57  0000000000000000FF0000  255
 58  0000000000FF000000000  255
 59  000000FF8080808080808080  1023
 60  80FF0101010101010101FF  645
 61  0101010101010101FF1010  294
 62  10101010101000D0D1010  154
 63  1010100D0D1010101010  154
 64  10101010101002040000  102
 66  00073FCF000000067EFE  663
 67  FEFE003C7EFFFFFFFF7E  1840
 68  0304010200010000 1367  133
 69  8D32C5192204FCF06080  1167
 70  000000003C0000000000  60
 71  0000060606060606060606  48
 72  0204FFFFECE0E18182C2  1654
 73  FF0300F70001021CFFFF  1046
 74  1F8C1D837681FFFFFF07  1350
 75  FF0367018484C4C5C2C1  1406
 76  F0FF20210100F806FFFF  1325
 77  0000EA0B0BFFFFFF0DE3  1261
 78  0387FFFFFFFF38383838  1382
 79  383838380104FFFFFFE2  1220
 80  E0FFFFFFFFFF08000300  1510
 81  04FFFF9F000EC3044EFF  1219
 82  FFFFFF0F0117FFFF3838  1426
 83  38380303FFFFFEFCF8F1  1623
 84  E2E2FF81000081422 4A5  1232
 85  FFFF7F3F1F8F4747C2C3  1405
 86  C0C4C8D4C2C1A5E70000  1583
 87  0000000043C303231328  362
 88  4383E0E0F0F8FCFEFFFF  2150
 89  C33C0000000081FF0707  653
 90  0F1F3F7FFFFF30303030  938
 91  3030303030302FF8FC1  836
 92  E0F0F8F8FCFFFFFFFF3F  2295
 93  1F0F27FCFEFFFEFEFEFC  1860
 94  FC4783030303030303F9  721
 95  FEF0F0E0C080FF030303  1542
 96  070F1F7FFF3838383838  715
```

```
 97  380101FFE7E781E7C399  1483
 98  FF38060105060C301010  421
 99  286CE84C0C1814366762  767
100  20606C5E5C40C0C0C1E6  1293
101  F90AE9C681C060301C16  1205
102  3676230603030370F1F  275
103  7FFF040404040404040401  411
104  43324A6E4A43470C0C0C  533
105  0C143676230370E1C1C  319
106  2C6CEC4C0C1C3A776260  875
107  78FC040404040104000E  407
108  1F3068DC99100726728B1  908
109  417E9C80387CC7C1060C  1065
110  1C08081CF860201F0F00  494
111  0404040401060306 3678  206
112  CB1EF0C00038E7919C22  1287
113  2A55000E0CC5667C0000  576
114  00306DA63371180004CE  721
115  C60201C27C00041E1870  689
116  C8265320404040404040  377
117  020203071CF888E7880A  851
118  80E07018080C8C861500  803
119  603C1173FE0C4C1E3C60  816
120  DC222A55040404040102  400
121  040C1C09073CE300040E  365
122  1CC8701C878004040201  642
123  0202267E64040CCADA53  787
124  33371210303E04040203  253
125  0000033FF207CAC0001F  740
126  F3E02622901C0000C0F0  1143
127  1C3E4666606031311818  600
128  1F0F1881C99C88E1FF9E  1330
129  E6E4CC0C38F080000404  1106
130  040404040201484B2CF8  458
131  1F34D212084A3C481020  573
132  60E0020401022C1A2E0C  457
133  8D72CC3334587460ED12  1117
134  E31C07070304FFFFFFFF  1296
135  BF0989C9FFFFFFFFDF0F  1796
136  1F3FFFFFFFFFFFFFFFFF  2134
137  FFFFFFFFFFFFFF1FF0E2  2282
138  C0C084030393CE010343  946
139  8100001001F1F1F0F0E0  1332
140  C20BE7FBFDF961030323  1327
141  FFFFFFFFFFFFFEFC9090  2324
142  8C0F1F1F3F7F367C00F8  835
143  FAFFFFFFC747478F8F87  1777
144  0F1F38383838383838  494
145  3838183804011F3F7F7F  545
146  F7F7F7F7E7EFCFCFEF9F  2270
147  DB83E7F1FDF3E78FB1F7  2116
148  F7FBF3E36D7D3B1F0202  1296
149  0202010400000021 65ED  380
150  FFFF0000002D128CEDE1  1175
151  0000014346EFFFF0000  692
152  0080C46DFFFF04040404  959
153  0102000000025 2DBFFFF  816
154  0000000024ADFFFF0404  727
155  0101000000012DBFFFF  749
156  040A02808080C0C0E0E0  1232
157  F000000000000000000F0  480
158  F8FCFCFEDFD7D7000000  1659
159  0000000080DBEBEAEAEA  1284
160  FFF0F8C0C0E0A0B8E838  1983
161  00F8FCFEFFFFF7F6FA00  2007
162  0000000080C0EBEBFF  1173
163  F8F8FCFEFE7068FC0000  1724
164  000000EEEFEDEDF6F6F6  1689
165  FF000080C0C0E0D8DCB4  1607
166  B8B8B8B8A8A8ACF80000  1492
167  0000000000B28AD454A8  780
168  A8A8A8000000000000  504
169  00A8A8585898A8A4B400  1176
```

```
170  00000000000000D45454  380
171  DAA9AAD6FF0000000000  1026
172  80E0F804040404040404  628
173  0404040404020400202   32
174  0202020A04FFFFFFFFFF  1295
175  FFFFFCFFFFFDF0F0C081  2326
176  02FFA400003FD04000FF  1011
177  7F1F0F87C72313F0E083  1156
178  84848482820000E0E613  825
179  1491610C10000400F000  534
180  004B23212331DBA50381  743
181  C0C6C3E0F0FFFF60002C  1699
182  D70070F0FF0007380000  885
183  0701FF090F1F7F7FFFFF  1082
184  FFFFFFFFFFFFFFFFFFF2  2537
185  F9FDFCFEFEFEFE036B4B  1955
186  979797A7AFFFFFFFFFFF  2070
187  FFFFFFFFFFFFFFFFFFFF  2550
188  FFFEFEFEFEFEFEFEFEAF  2462
189  2F4F1F7F4F8FAFFFFFFF  1446
190  FFFFFFFFFFFFFFFFFFFF  2550
191  FFFFFFFEFEFEFEFEFEFE  2543
192  FEAF2F4F1F7F4F8FAFFF  1365
193  FFFFFFFFFFFFFFFFFFFF  2550
194  FFFFFFFFFFFEFEFEFEFE  2545
195  FEFEFEAF2F4F1F7F4F8F  1443
196  AFFFFFFFFFFFFFFFFFFF  2470
197  FFFFFFFFFFFFFFFEFEFE  2547
198  FEFEFEFEFEAF2F4F1F7F  1729
199  4F8FAFFFFFFFFFFFFFFF  2182
200  FFFFFFFFFFFFFFFFFFFE  2549
201  FEFEFEFEFEFEFEAF2F4F  2079
202  1F7F4F8FAFFFFFFFFFFF  1830
203  FFFFFFFFFFFFFFFFFFFF  2550
204  FEFEFEFEFEF9FFE508AF  2186
205  2FEF03A7FE8FC7FFFFFF  1817
206  FFFFF3F8F2020202020   1131
207  2020202020202030101O  304
208  3030101030301010303O  352
209  1010303010103030101O  288
210  3030101010020GFFF7EB  889
211  57AF5DAE5DFFFFFFFAF5  1882
212  EAF4ECFFFFFFFF7EB97A  2167
213  B9FFFFBF5DBA5DDA51FF  1812
214  F7EBF76AB76BE5FFFFFF  2119
215  7FBF5FBD56B65D950D14  1145
216  0C1408E46CE840C040A0  1088
217  4073B1D1717030100OD0  1062
218  41C041C0C10000AB65E3  1206
219  858280000003556335332  714
220  2202000707070707070707  85
221  0707070707020 2FFFFFF  804
222  FFAED7EFD6FFFFFFF7EB  2344
223  75BB6D9A868A060A060A  871
224  04AB65A7450706040007  536
225  0707070206FFFFFFFFFF  1304
226  FFFEFDFFFFFFFFFF7FBF  2355
227  5FFFFFFFFFFFDFAFDFF  2381
228  FFFFFFFFFFF7FFFFFFFF  2422
229  FFFFFFFF7FFFFFFFFFFF  2422
230  FFFEFFFAE1C080000000  1544
231  00BF5FAF5E2E5C2C18E2  987
232  810000000000000FE79E0  728
233  4000000000BF5FAF5E2E  665
234  1C0C00D72B170B0F0606  359
235  0002020202020202020202  18
236  0202020202FFFFFFFFFF  1285
237  FFFEFDFFFFFFFFFFFFFF  2547
238  FFFAE5C2850210000FF   1319
239  FFF7FBF7CF860020202   1302
240  020101007CFE5C0063F7  820
241  E3020101002070600OE0  695
242  F0500202000000000000  348
243  0000000000006070301O0  17
244  0000181818909090900O  648
245  00A053A5454444000000  613
246  0000000000063676E2C  356
247  6828280000000000000  184
248  00000000000000000F8  248
249  FCFEFFFFFFFFFF000204  1787
250  00C1E0F8FF0020S4A840  1268
251  0000FF0102050A142800  333
252  FF000810040800OFF00  546
253  010204091200FF000142  356
254  8440031FFF0F1F3F7FFF  976
255  FFFFFF07070704070707  811
256  062A2A2A2A2A2A2A2A02  344
257  08000002060E0A0A1B00  77
258  103031D1D9D9BF00004C  1023
259  CCCA8A4B790000736355  1039
260  52DC88010102OB999D95  912
261  E700E0407912222I3E00  787
262  00008C8EEA2A33000000  609
263  0060E05884C1E0E0F0F8  1669
264  FCFEFF0C040000000000  777
265  FF3000000000000FF00  558
266  000000000000FF430000  322
267  00000000FF1E08000000  293
268  0000FF00000000000000  255
269  FF03030707070F0FFFB2  745
270  B2B2B2B2B2B2B2323232  1396
271  3232323232010400O039  312
272  2D452D25C240C0402625  785
273  251C0800182424129363  433
274  2200609050509O9F19B2  940
275  B2B2B2040880C0F0F8FE  1608
276  FEFDFD0000081C3CF8F0  1344
277  E000000000000000000O  224
278  000000000O2070787C3E  450
279  1F01070F1F3F7FFF7FFE  911
280  FFFFFEFDFDFEFE000C1E  1820
281  FEFCF8F0600000000000  1090
282  000000000000000F0F07  37
283  387CFE7F07BFBFBF7FFF  1523
284  FF7FBFFFFFFEFDFDFBFD  2347
285  FE0E3CF8F0E0C0887808  1544
286  1C3C7C7CF8F8F8010303  1087
287  032170F8F8008081C1E3  1321
288  F3FB7B010303A3B1B81C  1176
289  0C10BC9ECFE7F7FB78BF  1621
290  BF7FFF7F7F7FFFFFFFFF  1974
291  FFFFFFFFFFB6CFFFFFFF  2429
292  FFFFFFF8778FFFFFFFFF  2295
293  FFFB778FFFFFFFFFFB5  2224
294  CEFFFFFFFFFFFF75FBFF  2359
295  FFFFFFFFFF97EFFFFFFF  2430
```

```
300  FFFFFFFFFFFFFFFFFFFF  2550
301  FF0307060506060703O3  302
302  1706060606170303171?  122
303  1717171703030303030303  110
304  0303030202006081C1E3  690
305  F3FB7B010303A3B1B81C  1176
306  0CB5CEFFFFFFFFFFFF75  2045
307  FBFFFFFFFFFFF171703  1830
308  030707FCF0E2E7F2F0F0  1688
309  F81F0723732787070FFF  1015
310  FFFFFFFFFFFFFFFCF0E2  2503
311  E7F2F0F0F81F07237327  1428
312  87870FFFFFFFFFFFFFFF  2070
313  FFFFFEFCF8F0E0C080F2  2290
314  F2FAFFDFC7E0FFA7A7AF  2157
315  FFFDFBE31FFFFFFFFFFF  2292
316  FFFFFFF2F2FAFFDFC7E0  2400
317  FFA7A7AFFFFDFBE31FFF  2036
318  FEFCF8F0E0C0800000000  1538
319  0000000000FFFFFFFFFF  1275
320  FFFFFFFCF0E2E7F2F0F0  2436
321  F81F0723732787070FFF  1015
322  FFFFFFFFFFFFFFFFFEFC  2546
323  F8F0E0C0800000000000  1032
325  00FFFFFFFFFFFFFFFFF2  2282
326  F2FAFFDFC7E0FFA7A7AF  2157
327  FFFDFBE31FFFFFFCF8F0  2256
328  E0C0800000000000000O  544
330  000000000000000FFFFF  765
331  FFFFFFFFFFFFFFFFFFFF  2550
332  FFFFFFFFFEFCF8F0E0C0  2430
333  8000000000000000000O  128
336  000000FFFFFFFFFFFFFF  1785
337  FFFFFEFCF8F0E0C08000  2048
341  0000000000000000FF   255
342  FEFCF8F0E0C0800000000  1538
347  0000000000O3A3A3A3A3A  290
348  3A023A3A3A3A3A02063A  416
349  3A3A3A0206063A3A3A02  364
350  0606063A3A0206060606  160
351  3A02060606060206060606  104
352  0606060600000000000O  24
```

DUMP: 50.300
N.° BYTES: 3.514

PANTS

```
1   FE390EFE060DFE030DFE  1122
2   040DFE4B20FEE71BFE20  1176
3   0507FE070507FE280302  584
4   FE041D1D1DFE041D1E1F  693
5   1E1FFFFE220CFE020CFE  1138
6   0A0EFE2B0CFE040DFEB4  1038
7   0503FE3005030603FE09  590
8   05030503FE2E0303FE0C  588
9   0503050305O3FE0C0503  298
10  FE120203FE0328FE0A1D  867
11  1E1E1D1EFE081D1E1DFF  724
12  FE2B0EFE920505FE4003  1042
13  05FE200505FE0B0505FE  630
14  360002FE0505050FE3A01  638
15  0301020106FE0305050FE  534
16  291D1E1D1E1E1E1D1E1D  307
17  1E1E1EFFFE230EFE090C  923
18  FE150DFE090DFE6321FE  1204
19  A60AFE0219FE3F060319  808
20  FE2D1D1D1E1E1E1E1E1D  536
21  1D1D1EFFFE2410FE0610  925
22  FE0610FE090FFE0410FE  1082
23  0510FE0410FE0A10FE0A  839
24  08FE0B10FE1C10FE1021  893
25  FE1821FEB40001FE2B26  1081
26  262626262627FE070503  498
27  FE1026FE0126FE0126FE  1148
28  0127FE1S1E1D1D1FFE04  692
29  1E1D1E1E1E1D1FFFFE23  753
30  10FE0610FE090FFE0310  843
31  FE0210FE0210FE0410FE  834
32  FE0310FE0310FE0310FE  1073
33  0510FE0310FE0110FE04  823
34  10FE0410FE0410FE0310  837
35  FE0910FE840401FE3607  985
36  030903FE050401FE3F04  600
37  01FE1D0501FE200401FE  835
38  090501FE0D0501FE061E  578
39  1D1D1D1D1E1E1D1D1EFF  519
40  0000000000000000000O  0
```

DUMP: 42.100
N.° BYTES: 391

ART_1-IP

```
1   ED730CF831005A210606  796
2   0610E5E5E5E5E5E5E5E5  1854
3   E5E5E5E5E5E5E5E510EE  2086
4   31FF4F2100000680E5E5  1008
5   E5E5E5E5E5E5E5E5E5E5  2290
6   E5E5E5E510EEED7B0CF8  1790
7   324FF857CB3FCB3FCB3F  1262
8   210AFF4F0600097AE607  879
9   0680FE002806A7CB183D  889
10  20FA7EA0200A7EB0773A  1089
11  8FFF3C328FFF2174A47A  1341
12  FE00280CF53EFF01E803  1104
13  EDB1F13D20F4AF320CF8  1477
14  320DF87E23FEFFCA2EF5  1474
15  E5FEFE2023E17E23473A  1319
16  0DF88057CB3FCB3FCB3F  1274
17  CB3FCB3F473A0CF88032  1099
18  0CF87AE61F320DF818D1  1187
19  FE0A573007E17E3244F8  1123
20  23E526006A29111EFF19  776
21  5E23561A320EF8131A32  648
22  0FF813CD7DF4E1C326F4  1558
23  3A0EF8F5D526003A0CF8  1134
24  6F291113F81956235E26  714
25  003A0DF86F19D13A0FF8  985
26  F50608E51A771324110FA  954
27  E123F13D20F03A0CF83C  1212
28  320CF8F13D20CA3A0EF8  1166
29  473A0CF890320CF8D526  1094
30  006F2929292929110058  421
31  1916003A0DF85F19D13A  753
32  44F8FE00201C3A0FF847  1022
33  3A0EF8F5C5E51A772313  1190
34  10FAE1012000009C1F13D  1028
35  20ED10194F3A0FF8473A  847
36  0EF8F5C5E5712310FCE1  1574
37  01200009C1F13D20EF3A  866
38  0FF8473A0DF880FE2020  1099
39  083A0CF83C320CF8AF32  921
40  0DF8AF3244F8C9C9609F  1459
41  3A4FF8FE00280CF501E8  1169
42  033EFFEDB1F13D18F006  1306
43  031159FAC57E23E5D511  1176
44  BEFE26006F29195E2356  874
45  EBD1010600ED80211100  914
46  19EBE1C110E03E03DD21  1237
47  59FAF5DD7E04DD770FD1  1511
48  7E0CF67FDD7714DD7E05  1200
49  E6804FDD7E02DD7704DD  1351
50  7E03DD7705DDE506047E  1060
51  DD770723DD2310F7DDE1  1347
52  DD7E07DD7703DD7E09DD  1274
53  77027EDD7710237EDD77  1104
54  11237ECBF7DD7706237E  1135
55  DD771623E5DD7E04DD77  1317
56  0BDD7E05DD770C79FE00  1090
57  200EDD7E0BDD770DDD7E  1104
58  0CDD770E1828DD7E0087  912
59  87874FAFDD46018110FD  1214
60  16005F210000DD460F19  481
61  10FDEBDD6605DD6E0419  1192
62  DD750DDD740EDD361201  996
63  DD361301DD361501E111  834
64  1700DD19F13DC26FF57E  1247
65  FEFFCA57F6223AFACB7F  1716
66  CA1DE8CB77C2A4E7E63F  1667
67  5F237E320CF8237E320D  790
68  F87BCBFFCBDF3244F811  1638
69  3CFA3E02320EF83D320F  812
70  F8CD7DF43A43F8DD2152  1531
71  FAC947E63FFE17280678  1258
72  CB7F2001C9D5E5060AC5  1219
73  CD42F0C110F9E1D1C93A  1662
74  92FFE607FE07DD7E03C0  1441
75  3A8FFF57FE63DD7E03C0  1438
76  7A3C328FFFED7B90FF21  1422
77  6084110040010O1BEDB0  750
78  0664FB7610FCAFDBFEE6  1621
79  1FFE1F28F7C300DCDBDD  1458
80  E6022014DBDDE604CAE0  1384
81  F0DBDDE608CAE9F0CDA9  1967
82  EDC3CCECDBDDE6012016  1597
83  3EFBDBDDE604CA20F1DB  1681
84  DDE608CA29F1CD57EDC3  1667
85  CCECDBDDE604CAADF0DB  1948
86  DDE608CAB3F0C3CCEC00  1715
```

DUMP: 50.000
N.° BYTES: 859

CRE_TABS

```
1   3E02CD0116ED7390FF21  1076
2   AC7E11D2F9015000ED80  1268
3   21AC7E010800EDB0DD21  1007
4   13F8AFF50E00CDB022DD  1337
5   7400DD7501DD23DD23F1  1208
6   C608FEC038E9DD2150F8  1523
7   AFF50E00CDB022DD7400  1186
8   DD7501DD23DD23F13CFE  1406
9   C020EA0000000000000O  458
```

DUMP: 50.000
N.° BYTES: 83

DEMO 1

```
  10 POKE 62766,201: POKE 56403,
201: RANDOMIZE USR 56320
  20 FOR N=30000 TO 30004: READ
A: POKE N,A: NEXT N: DATA 62,1,1
95,161,243
  30 INPUT A: IF A<6 THEN POKE 3
0001,A: RANDOMIZE USR 30000: GO
TO 30
  40 GO TO 30
9999 SAVE "DEMO1" LINE 10
```

CÓMO SE HACE UN JUEGO

OGEROX (II)

Continuando con el desarrollo del programa, en este capítulo vamos a ver cómo se utiliza la rutina de sprites y cómo se almacenan éstos, de tal manera que cada uno podréis modificar a vuestro gusto las características de los sprites de una pantalla o crear los vuestros propios.

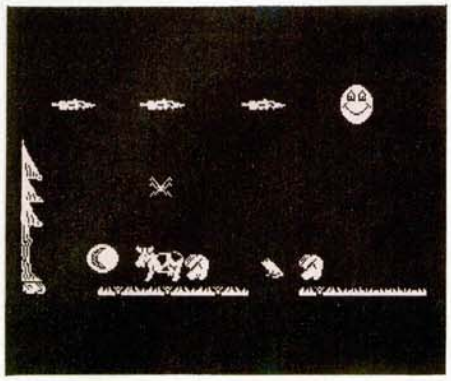
Pero, ¿qué es un sprite? Un sprite se puede definir —de una forma sencilla— como un gráfico que posee movimiento y animación. El movimiento puede realizarse atendiendo a varias causas: una secuencia fija, de acuerdo con unos límites prefijados, según una rutina de comportamiento, obedeciendo fielmente las órdenes que se dan por teclado, etc.

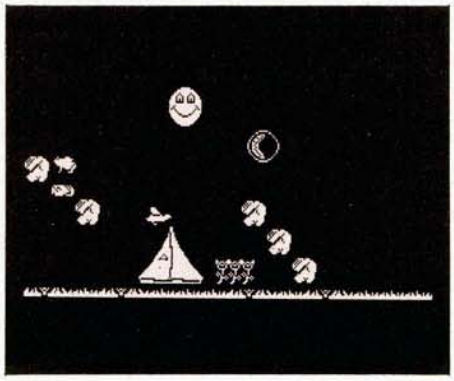
La animación es la sucesión de imágenes que hace que parezca que un sprite realmente se mueve. Si el sprite es una persona, la animación consistiría en una serie de imágenes que, vistas en un determinado orden, dan la apariencia de movimiento: la persona anda, salta... El movimiento y la animación bien combinados pueden conseguir estos efectos con un gran realismo. Por supuesto, no tienen por qué darse ambos a la vez; un sprite puede tener movimiento sin animación o viceversa.

En el caso del juego, los sprites pueden tener cualquier tamaño, limitado a caracteres tanto a lo ancho como a lo alto. El número de pasos de animación también puede ser cualquiera, así como el número de veces que el gráfico debe moverse hasta pasar a la siguiente fase de animación. En el caso de que el gráfico se mueva horizontalmente, tiene la posibilidad de cambiar la animación cuando varía el sentido del movimiento.

Para definir un sprite en una pantalla, primero hay que definir las características generales del mismo y después, en cada pantalla donde aparezca, se pueden cambiar. En la dirección 62799 se encuentra la variable (low-high) que apunta hacia una tabla de punteros a las direcciones de cada definición de sprite. Esta tabla de punteros está constituida por tantas variables de dos bytes como sprites haya. Cada variable apunta a la dirección donde está la tabla de características generales. Si obtenemos la dirección de la tabla de punteros a partir del contenido de la variable en 62799 y la llamamos *tab_punt*, tendremos la dirección de la tabla del primer sprite codificada en *tab_punt* y *tab_punt* + 1, la del segundo en *tab_punt* + 2 y *tab_punt* + 3, y así sucesivamente. Cada tabla es de la forma:

Byte: 0	Formato vertical: es el alto del gráfico de cada una de las fases de animación en caracteres.
Byte: 1	Formato horizontal: es el ancho del gráfico en caracteres.
Bytes: 2 y 3	Dirección del gráfico: es la dirección a partir de la cual están las definiciones del gráfico en cada una de sus fases de animación. (También low-high, como siempre.)





Byte: 4 Número de fases de animación: en el caso de ser uno no hay animación.

Byte: 5 Número de pixels que se tiene que mover el sprite para cambiar de animación. Si el bit séptimo es uno, la figura cambia de animación al volverse horizontalmente; en caso contrario la animación es la misma tanto si se mueve a la derecha como si se mueve a la izquierda.

Los gráficos se organizan por scans (un scan es una línea horizontal de un gráfico, ventana o pantalla). Esto quiere decir que el primer byte define los ochos primeros puntos, de izquierda a derecha; el segundo, los ocho siguientes situados a la derecha de los anteriores; y así sucesivamente hasta completar la primera línea y con todas las siguientes, desde la primera (superior) a la última (inferior).

En el caso de que haya varias fases de animación, tendrá que haber tantas definiciones de gráficos como fases haya. El primer gráfico es siempre la primera fase; cuando se completan las fases de animación, se vuelve a comenzar por la primera. El número de pixels que debe moverse el gráfico para cambiar de animación puede ser uno o más, dependiendo de lo rápido que se mueva. A veces es conveniente repetir varias veces la misma fase de animación en movimientos sucesivos, para evitar que ésta sea demasiado rápida. Si el bit siete de este byte es uno (lo que equivale a sumar 128 al valor normal del byte), el sprite cambia su animación dependiendo de que se mueva de izquierda a derecha o viceversa. El número de fases de animación es el mismo en cada sentido y se almacenan primero las fases de movimiento de derecha a izquierda y después —y justo a continuación— las de izquierda a derecha.

Además de las definiciones de las pantallas, por cada pantalla existe una tabla donde se encuentran todas las características de los sprites que hay en ella, así como de los objetos y puertas. En la dirección 62767 se encuentra la variable que apunta al comienzo de esta tabla. Las tablas de las pantallas están unas a continuación de otras y separadas por un byte con 255. Cada tabla contiene siempre:

Bytes 0-2: Número de los tres sprites que hay en esta pantalla. En cada una hay siempre tres y sólo tres sprites.

A continuación vienen los datos del primer sprite:

Byte 3: Coordenada Y mínima.
Byte 4: Coordenada Y máxima.
Byte 5: Coordenada X mínima.
Byte 6: Coordenada X máxima.
Byte 7: Incremento horizontal: número de pixels que se desplaza el gráfico en horizontal cada vez que se mueve (siempre uno).
Byte 8: Incremento vertical: número de pixels que se mueve en vertical (también uno).
Byte 9: Atributo: valor de la tinta y el papel. No debe haber ni brillo ni parpadeo.
Byte 10: Velocidad del sprite: de 1 en adelante. 1 es la más rápida.

Cada sprite tiene una zona rectangular por la que se mueve y que no puede traspasar. Esta zona se define dando las coordenadas máximas y mínimas tanto de X como de Y. X es la coordenada vertical e Y la horizontal, ambas en alta resolución (X de 0 a 191 e Y de 0 a 255).

A continuación vienen los mismos datos, pero para los sprites segundo y tercero:

Bytes 11-18: Datos del segundo sprite.
Bytes 19-16: Datos del tercer sprite.

Si en la pantalla correspondiente no hay ni puertas ni objetos, la definición se termina aquí con un byte conteniendo 255 y comienza la definición de los sprites de la siguiente pantalla.

Byte 27: Bites 0-5: número de objeto.
Bit 6: si está a 0 indica que si hay objeto no ha sido cogido y si hay puerta que no ha sido abierta; si está a 1 indica que el objeto ha sido cogido la puerta abierta.
Bit 7: si está a 0 indica que en esta pantalla hay un objeto; si está a 1 indica que hay una puerta. El estado del objeto o puerta depende del bit 6.
Byte 28: Coord. X del objeto.
Byte 29: Coord. Y del objeto.

En el caso de que haya una puerta, las coordenadas anteriores son las de la cerradura que abre la puerta. En cualquier caso las coordenadas se dan por caracteres, siendo X la coordenada vertical e Y la horizontal. Si hay un objeto en la pantalla, la definición se acaba aquí. Si es una puerta, hay además:

Byte 30: Coordenada X de la esquina superior izquierda de la ventana que contiene a la puerta.
Byte 31: Coordenada Y.
Byte 32: Formato vertical: alto de la ventana en caracteres.
Byte 33: Formato horizontal: ancho de la ventana en caracteres.

Todos estos datos son necesarios en el caso de que haya que abrir la puerta. En este caso el byte 34 es siempre 255 para indicar el fin de la definición en esta pantalla.

Cada pantalla tiene por tanto su definición gráfica por un lado y la definición de sus sprites y objetos o puertas por otro. Ambas deficiones terminan siempre con un 255 (FF en hexadecimal).

Programa de demostración

Para ver el programa de demostración hay que seguir los siguientes pasos (que son comunes en todas las demostraciones).

1. Teclear el listado Basic del programa cargador, lo cual debe realizarse con el programa cargador del artículo anterior en memoria. De esta forma se ahorra el teclear las partes comunes, que son muchas. Una vez tecleado el programa cargador sobre el anterior, hay que salvarlo antes de los bloques de Código Máquina de artículos anteriores mediante:
SAVE "cargador*n*" LINE 10.

Siendo *n* el número de artículo en el que aparece, en este caso '2'.

2. Teclear los listados hexadecimales con el cargador universal, salvándolos con sus nombres correspondientes y en el mismo orden en que aparecen en la revista a continuación de los bloques de Código Máquina de artículos anteriores.

3. Teclear el programa de demostración, salvándolo a continuación de los bloques de Código Máquina mediante:

SAVE "demo*n*" LINE 10

Siendo *n* el número de artículo en el que aparece (también '2' en este caso).

Para ver la demostración, basta con rebobinar la cinta hasta donde está el programa cargador (con el ordenador previamente inicializado por si acaso) y cargarlo mediante:
LOAD ""

El programa cargador se encargará de cargar todo lo demás y de ejecutar el programa de demostración.

Para todos aquéllos que quieran cambiar cosas o ver cómo están organizadas las pantallas, lo más fácil es parar la demostración pulsando CAPS SHIFT y SPACE. De esta forma todos los datos estarán en memoria. Para poder guardar modificaciones (si es que alguien quiere modificar cosas del juego), lo más fácil es hacerlas mediante POKE y añadir estos pokes en la línea del programa cargador anterior a la que carga el programa de demostración (o bien en el mismo programa de demostración, teniendo en cuenta que la siguiente demostración no incluirá las modificaciones). Si son muchas, puede volverse a salvar el bloque de Código Máquina donde estén encima del antiguo, teniendo cuidado de no borrar parte del siguiente. Como precacución puede dejarse un espacio mayor en la cinta justo a continuación de los bloques susceptibles de modificación.

Con esto queda completo todo lo que a sprites se refiere. En el próximo número veremos cómo se puede utilizar la rutina que imprime los sprites en pantalla desde Basic, con un ejemplo sencillo de un sprite que rebota por la pantalla. Hasta entonces paciencia y a teclear se ha dicho.

Alberto Elices
Roberto Oliva
Javier Elices

CARGADOR 2

```
  20 LOAD "TAB_SP"CODE 65214,96:
 LOAD "SPRITES"CODE 61807,562: L
OAD "IMP"CODE 59763,311: LOAD "S
PR_OGER"CODE 47200,3100: LOAD "S
P_OG.BIN"CODE 40800,1300
  50 LOAD "DEMO2"
9000 STOP
9030 SAVE "CARGADOR2" LINE 10: S
AVE "TAB_SP"CODE 65214,96: SAVE
"SPRITES"CODE 61807,562: SAVE "I
MP"CODE 59763,311: SAVE "SPR_OGE
R"CODE 47200,3100: SAVE "SP_OG.B
IN"CODE 40800,1300
```

TAB_SP

```
 1  D6FEDCFEE2FEE8FEEEFE 2400
 2  F4FEFAFE00FF06FF0CFF 1785
 3  12FF18FF0103E0B90485 1102
 4  0202A0BA04050202208B 582
 5  02850204A00B03850203 629
 6  20BD03850102408E0305 622
 7  1A031A03DB021A03DB02 529
 8  4D00395A1A031A03DB02 503
 9  4D001A03DB024D001A03 433
10  F30DCE0BE25000000000 779
```

DUMP: 40.000
N.° BYTES: 96

SPRITES

```
 1  CD7CF3DD2152FACDAAEA 1767
 2  DD2159FACDA8F111A6FB 1641
 3  CD73E9DD2187FACDA8F1 1806
 4  11B6FDCD73E9DD2170FA 1621
 5  CDA8F111AEFCCD73E9DD 1831
 6  2152FACD96F2C9DD3515 1458
 7  C0DD7E16DD7715DD7E10 1285
 8  CB7FCCC9F1C4EFF1DD7E 1999
 9  11CB7FCC22F2C40AF2C9 1476
10  DD460380F547DD7E0187 1221
11  878780DDBE08380BF1DD 1346
12  7E10ED44DD77101809F1 1077
13  DD7703CD45F2AFC9DD46 1526
14  0380DDBE07300ADD7E10 970
15  ED44DD771018C7DD7703 1227
16  CD4DF2AFC9DD460280DD 1542
17  BE09300ADD7E11ED44DD 1147
18  7711180SDD7702AFC9DD 1104
19  460280F547DD7E008787 1133
20  8780DDBE0A380BF1DD7E 1339
21  11ED44DD771118CBF1DD 1368
22  7702AFC9DD660CDD6E0B 1174
23  1806DD660EDD6E0DDD7E 1058
24  123DDD7712FE00280IC9 933
25  DD7E14DD7712DD7E13DD 1312
26  BE0F2001AFF5DD7E0087 1140
27  87874FAFDD46018110FD 1214
28  16005FF1F5FE00280419 926
29  3D18F8F13CDD7713DD75 1331
30  04DD7405C9DD7E03F5CD 1347
31  FEEEF1C610CDFEEEC957 1932
32  3A4FF8D608FE2838037A 1082
33  3CC9573E673243F87AC3 1195
34  A1F3AF3243F83A4FF8C6 1527
35  08C3A1F3AF3243F83A4F 1284
36  F83CFE28CA7AF6C3A1F3 1771
37  3EEF3243F83A4FF83DC3 1307
38  A1F33A45F8CB7F28183A 1231
39  D1F93D32D1F9200F3E08 1144
40  32D1F93A45F8E603CBF7 1566
41  3245F80604C504047887 837
42  87874FC647050503218 707
43  F33A45F8CB47282E7887 1233
44  212CFA16005F19E5357E 877
45  FEFF2018360723357EFE 1094
46  FF200F360979C6BE3240 988
47  F32145F8CBC6CBFEE1C1 1869
48  10BD1801C10604C50578 755
49  8716005F212CFA197E23 765
50  577E87878782212D2F916 1262
51  005F19E5160058218E50 714
52  19EBE106087E12231410 714
53  FAC110D1C93AD0F93D32 1495
54  D0F9C03E0532D0F9CDE3 1655
55  F2212CFA06087EFE00C0 1155
56  2310F93A46F8CBEF3246 1238
57  F8C900000000000000000 449
```

DUMP: 40.000
N.° BYTES: 562

INP

```
 1  DD7E03D5E607201ADD4E 1157
 2  00DD4601AFB110FD8787 1135
 3  874F0600DD5E04DD6605 883
 4  EDB01840F5DD7E005787 1363
 5  87DD6E04DD6605050600F5 1049
 6  DD4E01EDB00AF1213F13D 1227
 7  20F3D1E1E5DD4E00DD46 1528
 8  0104AF8110FD87878742 1049
 9  E5F5CB1E233D20FAF1E1 1551
10  10F4DD4601041803DD46 874
11  01E1E5545D24DD7E0087 1150
12  8787772370230D7E03CB 1124
13  3FCB3FCB3F7723232372 933
14  2373230D7E0647DD7E02 958
15  E6072802CBF878772B2B 1055
16  2B2BEB2600DD6E022901 734
17  50F809EB722373E111AE 1252
18  FCED52C0DD219EFB0604 1436
19  C5DD6603DD6E047CB528 1203
20  69DD5605DD5E06DD4600 1029
21  E5C5ED530CF856235E23 1256
22  E52600DD6E0219ED5B0C 965
23  F8EB0600DD4E01EDB0EB 1437
24  E1C110DFE156235EEB16 1354
25  00DD5E02197C0F0F0FE6 741
26  03F65867CD07E00CB3FCB 1256
27  3FCB3F57DD7E07E68028 1168
28  011442DD4E01DD5607CB 904
29  BAC5E572230D20FBE101 1283
30  200009C110F1110801DD 738
31  19C105C227EADD2152FA 1276
32  C900000000000000000 201
```

DUMP: 40.000
N.° BYTES: 311

SPR_OGER

```
 1  0000007800F801280288 547
 2  0710008C00720092012A 466
 3  022A024A019202220 3F4 550
 4  029802900 4C805400924 625
 5  3A1E7C0E381C000000000 310
 6  0078800F8012802880712 572
 7  008C00720092012A022A 487
 8  022A024A0555202D400D8 637
 9  00A80088009000900090 736
10  01F003F000000000000F0 724
11  01F00250051 40E26011C 429
12  00E20152024A04250C4A 513
13  14920A6203F402980110 692
14  02280264065E0F8E1F06 438
15  000C000000000007800F8 380
16  012802880710008C0072 456
17  0092012A022A022A024A 353
18  015200D400D800A80088 815
19  009000900090001F003F0 916
20  00000001E001F001480 209
21  114003E031004E004900 513
22  548054405240498044 0 839
23  2FC019400940132012A0 630
24  2490785C703E381C0000 650
25  0000078007C005200450 455
26  12380C4013801240152 0 432
27  151015101490128 50AD0 642
28  06C0054000440024002 40 467
29  024003E003F000000000 536
30  078007C00520145032 38 577
31  1C4023802540292032 10 495
32  291824942328 17E00CA0 743
33  04400A2013203D3038F8 574
34  307C1800000000000780 331
35  07C00520045002380C40 454
36  13801240152015101510 356
37  149012A00AC006C00540 811
38  044002400240024003E0 493
39  03F000000000C00001A 281
40  00601C4090202108401 2 487
41  04800C0300000000000 147
42  000000000000000181CC 334
43  66663A18180C00000000 322
44  000000000180180240 2 191
45  40420420010810C00901A 373
46  00600C0000000000000 108
47  000000000000181CC6666 538
48  3A18180C000000000000 118
49  300000580000380600 04 202
50  0902021084012048 00C0 458
51  300000000000000000 48
52  00000033818 05C666630 652
53  181800000000000000 48
54  00018018024024042042 357
55  381080580900306 0000 351
57  003381805C666630181 8 700
58  000000000000000020 04 36
59  500A0BD015A82BD44422 855
60  0810100820044002 2004 186
61  100800000000000000000 24
62  000000002004500A0BD0 345
63  15A82BD4812108600 6006 660
64  200400000000000000000 36
65  000000000002004500A 126
66  08102424524A0BD005A0 636
67  03C0142828144002 0000 381
68  000000000000000020 04 36
69  500A0BD015A82BD44812 843
70  100850062004000000 162
72  000000300068 01FE03FC 662
73  07F80FF01FC03EC06740 1154
74  78300F000000000000 183
75  000000000000000 0708 15
76  1FF40FFE3FFC5FF86FE0 1537
77  372010201C1000000000 179
78  000000000000C001600 34
79  7F803FC01FE00FF003F8 1271
80  037C02E50C1E00F00000 641
82  000010E02FF87FF03FFC 1217
83  1FFA07F604EC04080838 850
85  000006000000090001E0 240
86  1080021C774004038A20 534
87  0500044002080080 0594 364
88  290004A8120002BC2A00 463
89  0293FA0004902A0007F0 836
90  3E000000000000000180 191
91  000002400000043000 00 118
92  09D0000014A00000E320 656
93  00F700400388008004 00 582
94  0060090C10100B020FD0 385
95  04F139200093C4800134 858
96  030001C800000000000 204
98  000000001800000240 195
99  03F004200 40E1DD00A01 545
100 E2A004000140029 01080 745
101 02D0090004CB0900052C 484
102 EC800A145240041CE1C0 989
104 000006000000090000 0 240
105 01080780028EE38400451 589
106 C020022000A0010010 40 499
107 00942990000481520 0054 558
108 3D40005FC940005409 20 610
109 007C0FE000000000180 492
110 000002400000C200000 110
111 0B900000052800000 4C7 403
112 00000200EF00010011C0 451
113 060000200808309 00BF0 497
114 40D0049C8F200123C900 844
115 00C02C800000138 00000 511
117 000000001800000240 195
```

```
118   000004200FC00BB87020  582
119   0547805002800020010B  455
120   094000900B4000900D320  679
121   013734A0024A28500387  602
122   382000000000000000000  88
124   00000000000000001FDC0  446
125   02FFF00DFBC005F70001  1206
126   E000070C000000000000  423
129   000000000001F9C003FF  700
130   F00FFFC007FF00010000  965
131   00C0000000000000000  192
133   000000000000001C000  193
134   01E00001F00000F80001  715
135   04E003FFF00FFFC007FF  1450
136   0000800000600000000  224
139   0003BF800FFF4003DFB0  1058
140   00EFA0000780000E000  761
143   000000000000000000003  3
144   9F800FFFC003FFF000FF  1502
145   E000008000030000000C  355
148   0000038000078000F80  409
149   001F000720800FFFC003  663
150   FFF000FFE00001000006  981
151   000000000000000E70  126
152   1188218442420000000  450
153   000000007E0399C4242  576
155   6006199807E002400000  576
157   000000008000000C1E00  50
158   0663800784E0068BA006  907
159   49000C25A00810E0000F  545
162   00000000000000000E  14
163   000033000CC480068940  594
164   0784200244C006210004  476
165   1E0000000000000000  30
167   0000100078300 1C66007  486
168   21E005D1600092600584  978
169   3007081000F000000000  319
172   00C3000007000000CC00  316
173   0123300291600421E003  591
174   2240008456000782000000  478
175   0000000001800240042O  231
176   0240018003C007E00910  646
177   0A1015881708138811O8  394
178   0B10081007E001800240  477
179   0420024001800 3C007E0  657
180   085008D0194811A810E8  834
181   13C8099008100 7E000000  627
182   00007E0001810100200040  450
183   0478200D0101 1A00813  592
184   4008120080254004268O  561
185   0425800426800425800 4  512
186   26C004134008 12C00811  560
187   600808B0100478200200  462
188   4001818000 7E000000000  448
190   7E0001818002384004E8  742
191   2009501013A008154008  417
192   2680042580042680042 5  546
193   800416C0081360080980  662
194   1004D020023840018180  640
195   007E0000000000000000  126
197   0000000007E0001818002  386
198   063C6008D01013A00815  602
199   4008268004258004268O  577
200   0425800416E008135008  534
201   08E810063C6001818000  676
202   7E0000000000000000000  126
204   00000007E0001818002  386
205   384004E8200950101 3A0  672
206   0815400826800425800 4  440
207   2680042580041 6C00813  580
208   600809B01004D0200238  607
209   4001818000 7E000000000  448
211   7E0001FF8003FFC00787  1102
212   E00F2FF01E5FF81CBFF8  1366
213   1D3FF83ABFFC397FFC3A  1335
214   7FFC397FFC3A7FFC393F  1372
215   FC1CBFF81D3FF81E9FF8  1496
216   0F4FF00787E003FFC001  1151
217   FF80007E000000000000  509
218   0000000000000007E00  126
219   01FF8003C7C00717E00E  1046
220   AFF01C5FF81ABFF8397F  1435
221   FC3A7FFC397FFC3A7FFC  1562
222   193FF81C9FF80E4FF007  1111
223   2FE003C7C001FF80007E  1175
226   0000007E0001FF8007C3  712
227   E00F2FF01C5FF81ABFF8  1362
228   397FFC3A7FFC397FFC3A  1367
229   7FFC191FF81CAFF80F17  1172
230   F007C3E001FF80007E00  1176
233   00007E0001FF8003C7C0  904
234   0717E00EAFF01C5FF81A  1080
235   BFF8397FFC3A7FFC397F  1496
236   FC3A7FFC193FF81C9FF8  1460
237   0E4FF0072FE003C7C001  1006
238   FF80007E000000000000  509
239   0000000000000009D00  157
240   07FFE007FFF01FFFF83C  1582
241   7E3C387E1C383C1C7818  684
242   1E7F007C7F80FC7F99FE  1322
243   7F99FE3F00FE3E007E78  1159
244   181E383C1C387E1C387E  590
245   3C1FFFF00FFFF007FFE0  1582
```

```
246   00B900000000000000000  185
247   F70003FFC007FFF01FF9  1479
248   F83FF8783FF07839F0FC  1651
249   78E1FE7001FE7001FE7C  1457
250   18FC3F183E7F800E7F80  949
251   0E7F871E3F0F9C1E0FFC  837
252   1E1FFC0F9FF80FFFF007  1252
253   FF8000DF000000000000  606
254   0000D90007FFE00FFFF0  1213
255   0F81F83F81FC3FC3FC3F  1409
256   C3FC7FC3FE33C3CE3000  1523
257   0C70180E70180E70000C  436
258   33C3CC7FC3FE3FC3FC3F  1599
259   C3FC3F81F81F81F80FFF  1565
260   E007FFE0009D00000000  867
261   00000000F70001FFE00F  742
262   FFF01F9FF81E1FF83E0F  1319
263   F83F0F9C7F871E7F800E  1043
264   7F800E7F183C3C18FE70  930
265   01FE7001FE78E1FE39F0  1518
266   FC1FF07C1FF8781FF9F8  1574
267   0FFFF007FF8000DF0000  1123
268   000000000000001C00230  243
269   04B80444048005809C0  726
270   09C009E008E010F010F0  1178
271   10F011F011E011E009C0  1196
272   0780078003300FE800000  568
273   000001C00260053004 58  436
274   04800780BC00BC00BC0  876
275   09E011E011E011E013E0  1199
276   13C013800B400EC42EC8  889
277   1C7018600C800000000  400
278   000001C00230048B0444  503
279   04800580090090C009E0  900
280   08E010F010F010F011F0  1257
281   11E011E009C007800780  953
282   03300FE800000000001C0  491
283   0320067006500680580  506
284   09C008C008C008E01060  945
285   106010610E010C018C0  888
286   0CC00DC42EC81C701860  919
287   0C800000000000000380  271
288   0C401D202220012001A0  397
289   0390039007900 7100F08  491
290   0F080F080F8807880788  483
291   039001E001E00CC017F0  1064
292   000000003800640 0CA0  373
293   1A20012001E003D003D0  738
294   03D007900788078807 88  791
295   07C803C801C802D02370  968
296   13740E3806180130000 0  284
297   000000003800C401D20  260
298   2220012001A003900390  554
299   079007100F080F080F08  243
300   0F8807880788039001E0  809
301   01E00CC017F000000000  692
302   038004C00E600A600160  640
303   01A0039003100 3100710  369
304   0600606806807060308  68
305   031803023B013740E38  494
306   0618013000052410944  303
310   00000000000000000000  0
```

DUMP: 50.000
N.° BYTES: 3.100

SP_OG

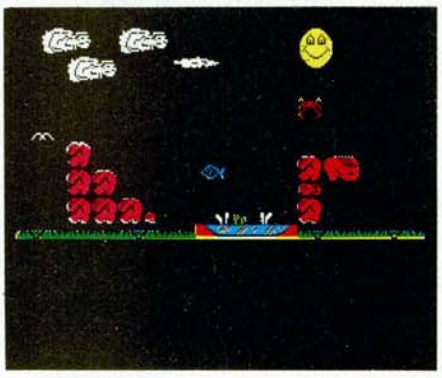
```
1     010211465F00010006  201
2     025000285D0001050279  344
3     AD6700010000401FF0506  548
4     0109003876000107016 9  298
5     AE5000010000501B00028  477
6     4E000102020200600FF02  348
7     040811001 03E00010401  113
8     3186480001000101890 0  395
9     1046000103 02FF010503  356
10    012E307600107017 82E  389
11    2856000101029ADE6729  650
12    01000701FF0704092A53  409
13    5700010006019 1002 85E  374
14    00010201C8DE5F000000  521
15    0502820B04051C0A04FF  454
16    02030A1046670001 0003  208
17    0159BE6700010002 01B4  567
18    0021460001060 1FF0001  367
19    0719966F0001000501 71  413
20    0028760001060 2B9F667  701
21    2801000701840B010512  216
22    0A05FF03050429D63700  592
23    01000101D8F628760101  625
24    020140D65F0001000301  381
25    FF0200054000093600 01  390
26    070262C5480001000102  380
27    5000597600010503040A  310
28    01FF0501077100 0A3600  446
29    0105023AAD4175010101  424
30    01B2F5370001000702 06  495
31    0901FF0502091F55416D  571
32    00010704 7F5509360001  288
33    0501D2550A7500010301  433
34    0504010807225509090 1  163
35    0101015555090901010 1  194
36    0155550909010101 01FF  448
37    0A0807225500A7501010 7  280
38    03629E3F0001000503CA  533
39    F5390001000602850D01  458
40    091E0402FF0103063000  358
41    3257000105021 0A55700  413
42    01000601 8F00092E0001  207
43    0203FF050A03092E4100  398
44    0100060268000 9460001  193
45    01029AF6670001000701  515
46    FF0901044800093D0001  412
47    0203AD0009650001010 2  292
48    90F5170001000201FF03  674
49    040520D71F0001000201  291
50    A0F6400001000701 2100  512
51    41760001 0302FF07010B  463
52    092E380001000602 5D00  213
```

```
53    12480001010270 9F2800  405
54    01000503FF06040B1700  308
55    1159000104017 9D54F00  525
56    0100010289B660000100  420
57    0604830701 0A000402FF  420
58    05040B2F001246000107  163
59    0250C41700010003019A  460
60    E5600001000102FF0B01  596
61    07286D6000010002026 0  353
62    97113E01010502D1F667  797
63    0001000602860309 0A00  165
64    0505FF0501081800127 5  438
65    0001050150004776 0001  290
66    0402D0001135E000101 01  330
67    FF0B0102092E6000010 0  421
68    0102680042660000107 02  285
69    82CF270001000401FF0B  648
70    05090A3660000010003 02  180
71    7800497600010501CC00  522
72    1146000107010305 0BFF  370
73    050B010AC61235010101  299
74    0289F6600001000602 38  578
75    0051760001040 2FF0701  469
76    030A47270001000601C8  331
77    0013360001070218BE57  400
78    0001000201070D01FF03  283
79    0A0958EF270001000601  393
80    0900124600010401 38C7  358
81    5F0001000501FF010607  371
82    1800114E000105022A00  169
83    126D000101 0170B76700  528
84    01000701FF0701080A2E  336
85    270001000605426E4700  298
86    01000102C1F52F000100  490
87    0402040406FF09010538  346
88    762F0001000402780012  310
89    4600010702880 0386E00  382
90    010601FF060507400011  362
91    4E000105017800113600  276
92    010403B9F65700010007  534
93    02FF0701000A36370001  385
94    0006046800517600010S  319
95    0260AF2700010004028F  574
96    0504094700426600010 7  265
97    026A952F0001000402A9  481
98    E6272F0100060302031E  361
99    FF0701000926270001 00  350
100   0602800039560001050 2  287
101   A9F5690001000402820D  669
102   140A1E0402FF01000687  463
103   0030660001010 27CB638  515
104   0001000402A800093500  237
105   010604FF07080181D617  696
106   0001000704380028560 0  194
107   0106015 1E6397601010 5  501
108   01050D01FF0100091800  309
109   4976000107021 14E2000  328
110   01000103A2D62F000100  429
```

DUMP: 50.000
N.° BYTES: 1.300

DEMO 2

```
  10 POKE 56403,201: RANDOMIZE U
SR 56320
  20 CLEAR 30999: FOR N=31000 TO
 31004: READ A: POKE N,A: NEXT N
: DATA 62,1,195,161,243
  30 REM PRUEBA DE PANTALLAS CON
 SPRITES
  40 FOR N=1 TO 3: READ A: FOR J
=A TO A+2: POKE J,0: NEXT J: NEX
T n
  50 DATA 61807,61814,61860
  60 POKE 63008,201: POKE 56404,
201: RANDOMIZE USR 56320
  70 PAPER 0: INK 6: BORDER 0: C
LS
  80 FOR n=0 TO 5
  83 IF INKEY$<>"" THEN GO TO 83
  85 POKE 31001,N: RANDOMIZE USR
 31000: GO TO 100
  90 GO TO 80
 100 RANDOMIZE USR 61807: IF INK
EY$<>"" THEN NEXT N: IF N=6 THEN
  GO TO 80
 110 GO TO 100
9999 SAVE "DEMO2" LINE 10
```



CÓMO SE HACE UN JUEGO OGEROX (III)

En el número anterior estudiamos el funcionamiento de la rutina de sprites. En esta ocasión explicaremos la forma de utilizar la rutina de impresión pixel por pixel que se utiliza para la animación de todos los objetos móviles del juego.

Una rutina de impresión pixel por pixel es un programa en Código Máquina que es capaz de imprimir en la pantalla del ordenador un gráfico. La posición en que se puede imprimir este gráfico no está limitada a las posiciones que coinciden con caracteres —como ocurre con PRINT en Basic— sino que la posición puede ser cualquiera. Como la pantalla del Spectrum posee una resolución de 256 puntos o pixels en horizontal por 192 en vertical, una rutina de impresión pixel por pixel será capaz de imprimir un gráfico en cualquiera de estas posiciones. De esta manera, las coordenadas de un gráfico en pantalla ya no estarán comprendidas entre 0 y 31 (horizontal) y 0 y 23 (vertical) en el mejor de los casos; los límites ahora estarán comprendidos entre 0 y 255 (horizontal) y 0 y 191 (vertical). Sólo mediante una de estas rutinas se puede conseguir que el movimiento de un gráfico o sprite por la pantalla sea lo bastante suave como para dar la impresión de auténtico movimiento.

La desventaja fundamental de una de estas rutinas reside en que como el gráfico puede estar en cualquier posición, lo más probable es que ocupe partes de bytes no completos; esto obliga a efectuar rotaciones. Estas rotaciones hacen que la impresión sea mucho más lenta, por lo que se pierde velocidad. En el caso de un programa hecho en Código Máquina y que no utilice máscaras (la utilización de máscaras es una técnica que se emplea para conseguir que los gráficos se mezclen con el fondo), la pérdida de velocidad no es grande, por lo que se pueden utilizar rutinas de impresión pixel por pixel directamente. (En otros casos hay que recurrir a trucos especiales como tener rotaciones ya hechas.)

La rutina de impresión del juego se encuentra tan metida dentro de él que no se puede utilizar directamente. Es necesario hacer algunas cosas, que realiza el programa de demostración, antes de poder utilizarla. En primer lugar se necesita un programa parcheador (un «programilla» en Código Máquina que hace de interface entre el Basic y la rutina). Este programa se sitúa a partir de la dirección 32.000, aunque variable *inicio* en el programa de demostración. Además, hacen falta unos datos, que la misma rutina se encarga de manejar, que se colocan a partir de la dirección 31.000. Son 5 bytes que deben dejarse a la rutina y que también se pueden cambiar de lugar cambiando el valor de la variable *datos* (también en el programa de demostración). Por último, es necesario un buffer o porción de memoria donde se realizan las rotaciones de 256 bytes. Este buffer se ha situado a partir de la dirección 33.000 y puede cambiarse de lugar modificando el valor de la variable *buffer*. Justo a continuación de este buffer, esto es, en la dirección 33256, están los datos correspondientes al gráfico a imprimir. Son los siguientes:

Buffer + 256: Formato vertical del gráfico. Es el número de scans o líneas de pixels que tiene el gráfico de altura. Puede tomar cualquier valor entero positivo desde el uno en adelante.

Buffer + 257: Formato horizontal del gráfico. Es el número de caracteres que tiene el gráfico de ancho. También puede tomar cualquier valor desde uno en adelante.

Buffer + 258: Coordenada vertical del gráfico. Puede variar entre 0 y 191.

Buffer + 259: Coordenada horizontal del gráfico. Puede variar entre 0 y 255.

Buffer + 260: Dirección del gráfico. En esta dirección y la siguiente se almacena la dirección en que se en-

cuentra definido el gráfico. Como siempre, en el formato low-high.

Buffer + 262: Atributo de todo el gráfico. Aquí se almacena el color de la tinta (ink), el papel (paper), el brillo (bright) y el parpadeo (flash) de la forma habitual.

El gráfico en esta ocasión se define por scans. Esto quiere decir —recordando del número anterior— que el primer byte define los ocho primeros puntos, de izquierda a derecha; el segundo, los ocho siguientes situados a la derecha de los anteriores; y así sucesivamente hasta completar la primera línea y con todas las siguientes, de arriba abajo.

Para imprimir el gráfico con los datos ya definidos, basta ejecutar la rutina de impresión mediante:

RANDOMIZE USR *inicio*.

Una vez más, la rutina de impresión no comprueba si el gráfico a imprimir se encuentra dentro de los límites de la pantalla, por lo que es responsabilidad de cada uno que no los exceda. En caso de salirse lateralmente, aparecería por el otro lado, pero un scan más alto o bajo dependiendo del lado. Si lo hiciera por abajo (por arriba no puede) lo más probable es que pasarán «cosas raras» con los atributos e incluso que el ordenador se «colgara».

Para los valientes dispuestos a cambiar cosas, la dirección de los datos para imprimir, que coincide con la dirección del *buffer* más 256, se almacena en la variable a partir de la dirección 59.939. El resto de las direcciones están en el programa de demostración, que ya se encarga de manejarlas.

Para ver la demostración hay que hacer lo mismo que en anteriores ocasiones. Primero teclear los bloques de Código Máquina con el cargador universal, salvándolos en su orden adecuado y con sus nombres correspondientes justo después de los bloques de artículos anteriores. A continuación copiar el programa cargador, que se debe salvar antes de todos los bloques de Código Máquina, previamente mezclado con los cargadores anteriores y de forma que los sobreimprima. Hecho esto ya se puede copiar el programa de demostración, que hay que salvar a continuación de los bloques de Código Máquina. Rebobinando ahora la cinta y cargando el programa cargador, sólo queda esperar a que se cargue todo lo demás para ver la demostración.

Y esto es todo de momento. En el próximo capítulo veremos cómo utilizar la rutina de impresión por caracteres y otras menores como la de impresión del contador de tiempo. Que ustedes lo tecleen bien.

Alberto Elices
Roberto Oliva
Javier Elices

CARGADOR 3

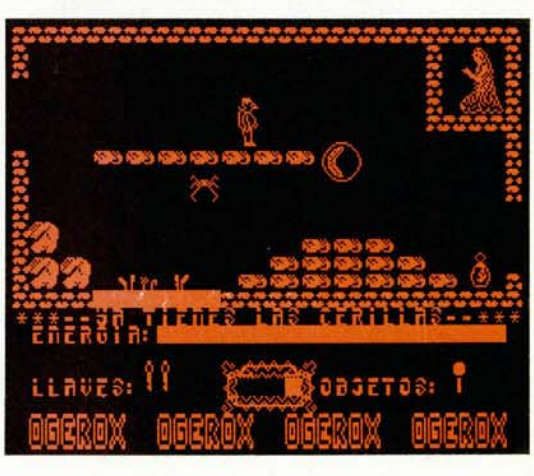
```
  25 LOAD "RESTO_PA"CODE 42490,4
710: LOAD "DATOS_HE"CODE 63500,5
89: LOAD "MENSAJES"CODE 65427,96
: LOAD "LEYENDA"CODE 31300,985
  60 LOAD "DEMO3"
9040 SAVE "CARGADOR3" LINE 10: S
AVE "RESTO_PA"CODE 42490,4710: S
AVE "DATOS_HE"CODE 63500,589: SA
VE "MENSAJES"CODE 65427,96: SAVE
 "LEYENDA"CODE 31300,985
```

RESTO_PA

```
  1  FE2010FE0210FE0310FE  1101
  2  0210FE010FFE0208FE02  811
  3  10FE0410FE0510FE0310  838
  4  FE0310FE0B10FE0210FE  1080
  5  0110FE0210FE0510FE04  822
  6  10FE0A10FE0310FE0610  845
  7  FE3421FE090401FE3F04  928
  8  01FE200701070107010 7  318
  9  01FE1304010901090107  306
 10  0107010701090 1FE3304  336
 11  01FE3E04010201FE1F1E  640
 12  1E1D1D1E1E1E1D1E1E1D  296
 13  1EFFFE220FFE0910FE08  1129
 14  10FE1110FE0210FE0310  848
 15  FE0410FE0510FE1610FE  1095
 16  0610FE0410FE410401FE  874
 17  3F0401FE25050106010 5  377
 18  0106010501060105010 6  33
 19  0105010E040401FE2304  563
 20  010301FE190401FE2005  580
 21  0105010E1B0401FE2007  585
 22  0109010E0407010701 07  292
 23  0107010701070107010 7  40
 24  01FF16FE0413161315FE  871
 25  0415161316 1FE0E11FE  644
 26  0811FE1728FF49 18FE2C  994
 27  161316FE071111FE0315  536
 28  13FE1C13FE1416FE0A15  901
 29  FE2D15FE3C15FE061217  956
 30  FE1416FE1C13FE091416  902
 31  1416141416 16FF141614  443
 32  14141613FE141212FE09  654
 33  17FE3F11FE3415FE0A11  965
 34  FE2015131413FE121616  681
 35  FE3315FE3C1516FE0412  959
 36  FE1916FE1A14FE0317FE  1135
 37  161613FE0413141613 14  421
 38  161313FF1413161413 16  437
 39  131311FE1E17FE811413  784
 40  1616FE070702FE071713  617
 41  141417FE0515FE2611FE  906
 42  1712FE0017FE1C15FE21  908
 43  17FE1C16FE0117FE0D15  893
 44  141314FE0817FE0316FE  877
 45  1013FE0517FE01141413  631
 46  141313FF1414131413 14  431
 47  17FE0A11FE1311FE7D15  994
 48  FE2113FE0712FE0815FE  1125
 49  061713FE1C11FE1318FE  898
 50  081CFE3115FE3C141914  739
 51  FE0317FE0F1316FE0116  867
 52  13FE04141413141314FF  650
 53  1112111416FE04141416  414
 54  1611FE03FE0011FE0515  847
 55  FE1311FE2A1715FE1512  923
 56  FE241AFE0416FE1217FE  1145
 57  011717FE0816FE0318FE  856
 58  0811FE0E1617FE0814FE  877
 59  0113FE0A16FE091316FE  864
 60  1816172FFE1816FE1C19  723
```

```
 61  13FE0517FE161616FE06  881
 62  1414141414161313131 3  198
 63  1313131314141614141 4  198
 64  FF1114141414141 7FE17  672
 65  15FE3E1217FE3F17FE07  979
 66  11FE0511FE1811FE1716  887
 67  FE1E13FE0C15FE0415FE  1123
 68  0816 16FE2C16FE0413FE  903
 69  0F12FE05121712FE0415  630
 70  16FE0E12FE0F13FE1417  893
 71  FE0514FE0D16FE0B1313  871
 72  131316FF1714141314 13  436
 73  1311FE91151 5FE0B11FE  1013
 74  1E12FE1116FE0E1413FE  902
 75  0616FE0614FE021314FE  857
 76  0616FE0513 16FE131517  645
 77  FE2C12FE1013FE142C2C  967
 78  2C2C1613FE1A14FE0817  714
 79  FE0114141414131 6FF13  650
 80  FE0414141413161 1FE1E  660
 81  11FE81151414151314FE  775
 82  0517FE0213131 3FE0213  616
 83  1613FE04111615FE1C16  663
 84  FE95191319FE021717FE  1028
 85  1416FE011616FE0116FE  872
 86  0114131413131416FF04  399
 87  0722222222220407FE20  474
 88  0407FE1E0407FE200407  603
 89  FE1E0407FE200407FE1E  876
 90  0407FE0623050704 07FE  583
 91  150407FE0D0307030705  324
 92  07FE0B0407FE080507FE  811
 93  160407FE110707FE1305  596
 94  07FE0A0607FE090407FE  812
 95  2705072C2C2CFE050207  451
 96  0307FE080407FE140307  567
 97  0307FE060407FE010307  546
 98  FE0403070307030703 07  298
 99  0307030703070307030 7  50
100  0307030703 07FF222222  387
101  2322232323FE200407FE  725
102  3F0407FE3F0407FE1E04  690
103  07FE0923FE150407FE1C  873
104  0507040 7FE0723FE0223  610
105  FE130802FE1A05070507  587
106  0407FE0523FE0623FE11  671
107  0802FE1805070507 0507  324
108  0407FE030507FE0A0507  556
109  FE0F03070207FE140307  572
110  0307030703070407020 7  50
111  03070307FE0A02070307  303
112  0307030703070307030 7  50
113  0307020 7FF2223232322  447
114  2223030703070307FE19  378
115  0407300407FE390407FE  646
116  050407FE390407FE0504  601
117  070407FE2023FE052323  668
118  FE0D0407FE0504070407  559
119  FE2B23FE0B0307030703  620
120  0702070407FE20040728  367
121  0407FE0323FE100407FE  838
122  200407FE080407FE0603  579
123  07FE0D0407FE1C030704  581
124  07FE0103070307030703  295
125  0703070307030703070 3  50
126  0703070307030703070 3  50
127  070307FF0407220507 22  363
128  222223230407FE200407  446
129  FE060507FE160207FE20  843
130  0407FE0623FE3734FE3A  979
131  0407FE0C0307030702 07  306
132  0307FE1B0407FE0423FE  849
133  0A0407FE2E0407FE0723  628
134  FE070407FE2E0407FE09  846
135  0507FE050407FE250307  583
136  0307030703070307FE09  303
137  0307030703070307FF02  297
138  0722230407040722222 2  200
139  2304070407FE280507FE  617
140  140407FE290507FE1404  616
```

```
141  07FE290507FE0A050703  593
142  070207FE050407040703  300
143  07FE2604070407FE0A05  590
144  07FE060507040 7407FE  555
145  060507FE0903070207FE  554
146  100307FE020307FE0506  557
147  07FE0C04070407FE1403  572
148  0703070207FE070207FE  550
149  0C0507FE050507FE0102  552
150  070307FE0404070407 05  302
151  07FE320307FE08040703  597
152  0703070307030703 03  50
153  0703070307030703070 3  50
154  070307FE050307FF23FE  830
155  0404070407FE06040704  301
156  0722222323FE200407FE  696
157  0504072B0407FE0F0407  350
158  FE200407FE050407FE08  829
159  0407FE300407FE052323  653
160  232323FE202222323FE05  754
161  0407FE000407FE3E0407  603
162  0407FE1C0505FE200407  600
163  0407FE1C0305FE070505  572
164  0505050505050505050 5  50
165  0505050505050505050 5  50
166  0505040504070407FE20  327
167  0307030703070307030 7  50
168  0307030703070307030 7  50
169  0307030703070307030 7  50
170  0207FF2222222222230 4  473
171  07FE080507FE140407FE  820
172  290507FE3E0507FE3504  692
173  07FE08232223FE040505  641
174  FE0403070407FE200407  576
175  FE1505050405FE060407  565
176  FE200407FE160305FE06  841
177  0407FE1303050305030 5  308
178  FE070407FE060505FE08  804
179  0505FE0405052C2C2C04  414
180  07FE2003070307030703  326
181  0703070307030703070 3  50
182  0703070307030703070 3  50
183  0703070307FF22222322  419
184  2322232304 07FE1E0407  445
185  FE3F0407FE3F0407FE20  942
186  0407030503050305030 5  43
187  0305030503050305030 5  40
188  0305030503 05FE060407  295
```



```
189  FE01222323222223FE07 723
190  0407FE1E0407FE200407 603
191  FE1E0407FE200407FE1E 876
192  0407FE2003070207FE04 574
193  0307030703070307030703 50
194  0307030703070307030703 50
195  0307030702070FF040225 327
196  FE042424242404020FE20 694
197  0403FE1E0403FE200402 590
198  FE1E0402FE200403FE1E 867
199  0403FE0103030303030303 280
200  0303030303030303030303 30
201  0303030303030303030303 30
202  FE05040224242424225FE 700
203  040402FE200403FE1E04 591
204  03FE200402FE1E0402FE 839
205  200403FE1E0403FE0103 588
206  0203020302030203020303 25
207  0203020302030203020303 25
208  020202FE04030202020FF 528
209  2424242424250403FE1E 508
210  0403FE200403FE3F0403 624
211  FE3F0403FE050303FE03 846
212  2424242504030FE200403 445
213  0502FE1C0403FE200403 589
214  05020502FE1A0403FE20 587
215  0403050205020502FE18 306
216  0403FE2003030303030303 311
217  0303030303030303030303 30
218  0303030303030303020303 29
219  FE0402030303FF242425 633
220  2424240403FE1E0403FE 660
221  300001FE020AFE010403 577
222  FE2D0104FE110403FE20 868
223  0403FE0C252424252504 460
224  03FE2004030502FE1C04 589
225  03FE030302FE1B0403FE 807
226  1E0403FE200403FE1E04 618
227  03FE0228FE160303FE08 843
228  0303030303030303030303 30
229  03030203FE0402030303 280
230  0303FF2424242424242504 482
231  03FE1E0403FE07040331 611
232  0503FE130403FE1E0403 579
233  FE070403FE0304030203 537
234  FE130403FE1E0403FE06 831
235  0403050304030203FE15 302
236  0403030303FE130503FE07 555
237  0403FE060503FE070202 540
238  FE100403FE0D04020402 556
239  0402FE0303030403FE04 534
240  0503FE0[illegible]03FE050303 536
241  FE190403FE0B05023205 613
242  02FE030403FE0804030FE 789
243  010203FE1E04030503FE 559
244  0705020502FE03050205 390
245  02FE0104030FE0804030FE 787
246  20030303030303FE040303C3 311
247  0303030303030303030303 30
248  030203FE0403030303302 280
249  03FF2425242525030303 450
250  030303FE0505030403040 287
251  03FE3C0403050304030FE 593
252  1A0303FE1A0503040305 332
253  03FE0404030FE140503FE 804
254  290403FE120503FE2905 628
255  030403FE080403FE0503 541
256  03FE190403FE13050304 574
257  0328FE000403FE0D0503 582
258  FE110503FE1204030403 565
259  0403FE080403FE090503 548
260  05030403FE100303FE06 552
261  0203FE0B03030302030 286
262  03020302030203020302 25
263  03020302030203020302 25
264  03020302030203020302FF25 312
265  24242424241F1F040304 476
266  03FEAE05020502FE0C04 715
267  030403FE3005020502FE 580
268  0A0403040305020502 43
269  0205020402FE29050205 322
270  02FE0804030403FE3405 589
271  020502FE0604030403FE 537
272  0728FE1103030303030303 336
273  0203FE0803030303030303 285
274  0303030303030303030303 30
275  FF2424242424250403FE 733
276  060403FE170403FE2704 594
277  03FE170403FE270403FE 841
278  170403FE0C0403050303 314
279  0302030303050 3FE0A04 290
280  03FE060403FE170403FE 808
281  0C0603FE0505030503FE 550
282  080403FE0404030503FE 542
283  2C0403FE0204030503FE 576
284  060403FE200403FE1105 582
```

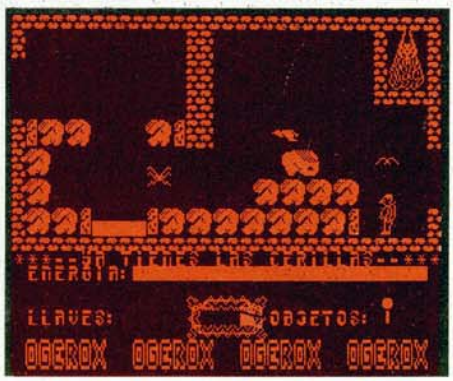
```
285  03FE0205030503040 3FE 536
286  040403FE1E0403FE1703 582
287  0303030303FE03030203 280
288  0203020302030203020303 25
289  0203020302030203020303 25
290  0203020302030203020302FF 277
291  040325FE0424242424404 450
292  03FE200403FE1205 03FE 830
293  0A0403FE200403FE1205 587
294  03FE0A0403FE01030203 537
295  0203020302030203020 2FE 276
296  1304032424FE0605 03FE 620
297  0803020403FE200203FE 565
298  1224FE060403FE3F0403 645
299  FE1D0502FE0C0302FE12 833
300  04030403FE0805020502 290
301  0502FE0D0402050204 03 294
302  FE2003030303030303030303 310
303  0303030303030303030303 30
304  0303030303030303030303 30
305  03030303FF2626262626 457
306  2626262626262626262626 380
307  26262626262626262626FE 596
308  0426262626262FE1E2626 554
309  FE1E2626FE1E2626FE1E 1004
310  2626FE0426262626262626 562
311  2626262626262626262626 380
312  2626262626262626262626 380
313  2626FE0426262626262626 562
314  2626262626262626262626 380
315  2626262626262626262626 380
316  26FE00226FE1E2626FE1E 974
317  2626FE1E2626FE1E1C26 786
318  FE0626262626262626FE03 745
319  2626FE0E26FE06262626 756
320  262626FE032626FE0E26 753
321  FE0426262626FE092626 749
322  FE0C26FE0426262626FE 968
323  01292626FE00FE0B2626 713
324  2626262626262626262626FE 596
325  0826262626262626262626 350
326  2626262626FF26262626 597
327  2626262626262626262626 380
328  2626262626262626262626 380
329  26FE0426262626FE00262D 747
330  FE172626FE1E2626FE1A 993
331  0504FE022626FE1E2626 701
332  FE180504FE042626FE1E 905
333  2626FE160504FE062626 697
334  FE1E2626FE140504FE08 905
335  26FE1F26FE130504FE0A 907
336  26FE1F26FE110504FE0C 907
337  2626FE1E2626262626 588
338  2626262626262626262626 380
339  2626262626262626262626 380
340  2626262626262626FF2626 597
341  2626262626262626262626 380
342  2626262626262626262626 380
343  26FE0426262626262626 2D 569
344  FE132626FE022626FE1A 961
345  2626FE022626FE1A2626 764
346  FE022626FE0F23FE0326 936
347  26FE022626FE0C262626 750
348  FE0826262626262FE0226 746
349  26FE0C2626262626262626 570
350  2626262626262626262626FE 596
351  022626FE1AFE0426262FE 946
352  1E2626FE1F26FE1F26FE 1006
353  0505040504FE1626FE1F 622
354  26FE0305040504050404FE 576
355  130504262 6FE1E262626 502
356  2626262626262626262626 380
357  2626262626262626262626 380
358  2626262626262626262626 380
359  FF2626262626FE042626 779
360  2626262626262626262626 380
361  2626FE0426262626262626FE 778
362  00260404FE010404FE05 569
363  263526FE0B2626FE0926 771
364  FE0826FE0B262604 04FE 903
365  010304FE0526FE0826FE 859
366  0B1C26FE0926FE0826FE 932
367  0C260404FE060304 26FE 617
368  0826FE0C26FE06030402 619
369  0426FE0826FE0C260404 654
370  FE0826FE0826FE070304 868
371  03042626FE0103040204 351
372  FE0526FE0826FE070304 865
373  030426FE0A26FE0826FE 901
374  0503040304030426FE08 326
375  030426FE0826FE050304 611
376  03040304 26FE07030402 322
377  0426FE0826FE0B26FE07 906
378  02040304 26FE08260304 358
379  FE0926 26FE04030402 04 610
380  030426FE0826030403 04 359
381  19191919 19FE022626FE 711
382  0402040304030426FE08 324
383  2603040304FE07262626 427
384  2626262626262626262626 380
```

```
385  2626262626262626262626 380
386  2626262626262626262626 380
387  FF2626262626262626262626 597
388  2626262626262626262626 380
389  2626262626262626262626 380
390  26262626FE1926FE0131 773
391  FE002626FE1926FE0426 943
392  FE1A26FE0426FE1A26FE 1186
393  0426FE1A2626262626 550
394  FE1F2626FE0403040304 633
395  0304030403040304 0304 35
396  FE0C2626FE1E2626FE1E 986
397  2626FE1F260504FE1D26 729
398  FE1103040304030 4FE08 554
399  2605040504FE0B030403 331
400  0403040304030 4FE0626 323
401  FE0428FE0103040304 03 570
402  0403040304030403040 4FE 286
403  0326262626262626FE0826 531
404  2626262626262626262626 380
405  2626262626262626FF26 597
406  2626262626262626262626 380
407  2626262626262626262626 380
408  2626262626262626262626 380
409  26262DFE172626FE1E26 796
410  26FE1E1C26FE1F26FE0A 975
411  0504050405040404FE0E 303
412  26FE1F26FE0805040404 640
413  0504FE0D0304030426 25 366
414  FE1A0502050226 26FE06 630
415  0504040405 04FE1326FE 591
416  1903020302030226FE05 337
417  0504040405 04FE040304 291
418  FE0705020402332 6FE0F 632
419  0304FE0E26FE03050405 584
420  0404040504FE0B050204 297
421  020502FE052626FE0929 648
422  FE0D2626262626262626 571
423  262626FE0826262626 26 566
424  2626262FE0526FF262626 760
425  2626262626262626262626 380
426  2626262626262626262626 380
427  2626262626262626262626 380
428  FE0C2626FE0C2626 3226 772
429  FE0C2626FE0C26FE0326 941
430  FE0D2626FE0C26FE0326 942
431  FE0D2626FE0C26FE0326 942
432  FE0D2626FE0C2626 2626 761
433  26FE0D2626FE1026FE00 943
434  2604040504050 4FE0405 327
435  0404042626FE102626FE 688
436  0C2626FE1026260504FE 697
437  110604FE0926FE1F2605 656
438  04FE0F0504050405 0405 305
439  04FE0626FE1F26050405 639
440  040404FE04040405 0405 292
441  0405040504050405 0405 45
442  040404FE052626FE052A 648
443  FE1526262626262626 26 579
444  2626262626262626262626 380
445  2626262626262626262626 380
446  2626262626FF26262626 597
447  2626262626262626262626 380
448  2626262626262626262626 380
449  2626262626262626262D 387
450  FE012626262626030603 457
451  0603060306030603060306 45
452  0603060206262 6FE0926 400
453  2626260206030603060306 143
454  0603060306030603060306 45
455  0626 26FE0A26262603 06 469
456  0306030603060306030603 06 45
457  0306030602062626FE0B 367
458  2626262626262626262626 380
459  2626262626262626262626 380
460  26FE1D2E26FE1F26FE1F 1013
461  26FE4D2626262626262626 635
462  2626262626262626262626 380
463  26FE0D2626262633333 3 610
464  FE0C2626262626FE0F26 763
465  FE0A2626262626FE0F 761
466  26FE0926262626262626 567
467  FE0F2626262626262626 573
468  2626262626262626262626FF 597
471  00000000000000000000 0
```

DUMP: 50.000
N.° BYTES: 4.710

DATOS_HE

```
1   00000000E7888040040 521
2   20484048104048048048048 880
3   C040E04800482048404 8 864
4   6048804 8A048C048E050 1168
5   0050205040506050805 0 720
6   A050C050E00000400002 802
7   020101000000000004000 68
8   41004200430044004500 335
9   46004700402041204220 432
10  43204420452046204720 505
11  40404140424043404404 40 650
12  45404640474040604160 723
13  42604360446045604660 820
14  47604080418042804380 941
15  44804580468047804 0A0 1014
16  41A042A043A044A045A0 1135
17  46A047A040C041C042C0 1232
18  43C044C045C046C047C0 1305
19  40E041E042E043E044E0 1450
20  45E046E047E048004900 1027
21  4A004B004C004D004E00 380
22  4F0048204 9204A204B20 501
23  4C204D204E204F204840 574
```



```
24  49404A404B404C404D40  695
25  4E404F4048604 9604A60  792
26  4B604C604D604E604F60  865
27  488049804A804B804C80  1010
28  4D804E804F8048A049A0  1083
29  4AA04BA04CA04DA04EA0  1180
30  4FA048C049C04AC04BC0  1301
31  4CC04DC04EC04FC048E0  1374
32  49E04AE04BE04CE04DE0  1495
33  4EE04FE050005100 5200  848
34  5300540055005600 5700  425
35  5020512052205320 5420  570
36  5520562057205040 5140  643
37  5240534054405540 5640  740
38  5740506051605260 5360  861
39  5450556056605760 5080  934
40  5180528053805480 5580  1055
41  5680578050A051A052A0  1152
42  53A054A055A056A057A0  1225
43  50C051C052C053C054C0  1370
44  55C056C057C058E051E0  1443
45  52E053E054E055E056E0  1540
46  57E005987ECB41C8CD34  1175
47  7E18F80600E53E2B3291  933
48  84FD66F62E70E31100000  1135
49  CDAA7EFE282020E3CBD4  1501
50  2CCB98E3CDB07E3814E3  1436
51  FD74F6CB6020B9CB4820  1438
52  BCFD73F4FD72F518B4ED  1853
53  53B784CB69282BFE2220  1109
54  10CD347E7ECB41C8CD34  1250
55  7E1800000001000000003  154
56  0009201020011001000  107
57  1816242424181818181818  276
58  0818180800000030402 05  78
59  0600030232326080 0600  397
```

DUMP: 40.000
N.° BYTES: 589

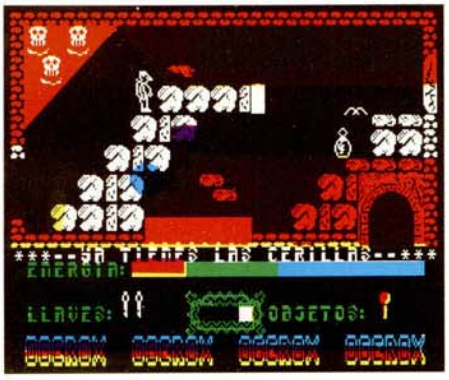
MENSAJES

```
1   2A2A2A2D2D5941205449  559
2   454E4553204C41532043  654
3   4552494C4C41532D2D2A  656
4   2A2A2A2A2A2A2D2D2D2D  432
5   5941205449454E455320  674
6   4C41204C4524412D2D2D  554
7   2A2A2A2A41434141241 53  579
8   2044452045 4E43454E44  630
9   4552204C41532043455 2  657
10  494C4C41532E00000000  419
```

DUMP: 40.000
N.° BYTES: 96

LEYENDA

```
1   B7BAADABDFB1B0DFACBA  1870
2   DFAFB0BBB6BEDFB6B2BE  1908
3   B8B6B1BEADDFBCB0B1DF  1893
4   B3B0DFAEAABADFACBADF  1912
5   B6BDBEDFBEDFBAB1BCB0  1924
6   B1ABADBEADDFBCAABEB1  1832
7   BBB0DFACBADFAFADBAAC  1873
8   BAB1ABB0DFBCB0B2B0DF  1874
9   A9B0B3AAB1ABBEADB6B0  1763
10  DFBEDFB3BEDFBEADAD B6  1946
11  BAACB8BEBBBEDFB2B6AC  1864
12  B6B0B1DFBBBADFBAB1BC  1905
13  BAB1BBBAADDFBAB3DFB8  1904
14  ADBEB1DFB9AABAB8B0D1  1873
15  ACBAB8AAB1DFB3BEDFB3  1883
16  BAA6BAB1BBBED3DFBAB3  1891
17  DFBCBEB2AFBEB2BAB1AB  1856
18  B0DFA9B6A9B6B0DFBAAF  1661
19  B0BCBEACDFBBBADFB8AD  1902
20  BEB1DFBAACAFB3BAB1BB  1852
21  B0ADDFBCAABEB1BBB0DF  1883
22  BAB3DFB9AABAB8B0DFBC  1900
23  B0B1DFACAADFBCBEB3B0  1874
24  ADD3DFADBAABBAB1B6BE  1872
25  DFBEDFB3B0ACDFB2BEB3  1933
26  BAACD1AFBAADB0DFB1BE  1867
27  BBB6BADFADBAAFBEADBE  1865
28  ADBEDFBAB1DFBEAEAABA  1892
29  B3DFBBB6BEDFBAB1DFBA  1956
30  B3DFAEAABADFAAB1B0AC  1850
31  DFA9B6BAB1ABB0ACDFB7  1862
32  AAADBEBCBEB1BEBBB0AC  1813
33  DFACBADFBEBCBAADBCBE  1919
34  ADB0B1DFBEB3DFA9BEB3  1879
35  B3BADFBBBAB3DFBCBEB2  1919
36  AFBEB2BAB1ABB0DFA6DF  1865
37  BEAFBEB8BEADB0B1DFBA  1864
38  B3DFB8ADBEB1DFB9AABA  1890
39  B8B0D3DFAEAABABBEB1  1878
40  BBB0DFB3B0ACDFB2BEB3  1873
41  BAACDFB3B6BDADBAACD1  1871
42  BAB1ABB0B1BCBAACDFBA  1842
43  B3DFBBBAACBEACABADBA  1839
44  DFACBADFBEBBAABADBB0  1932
```

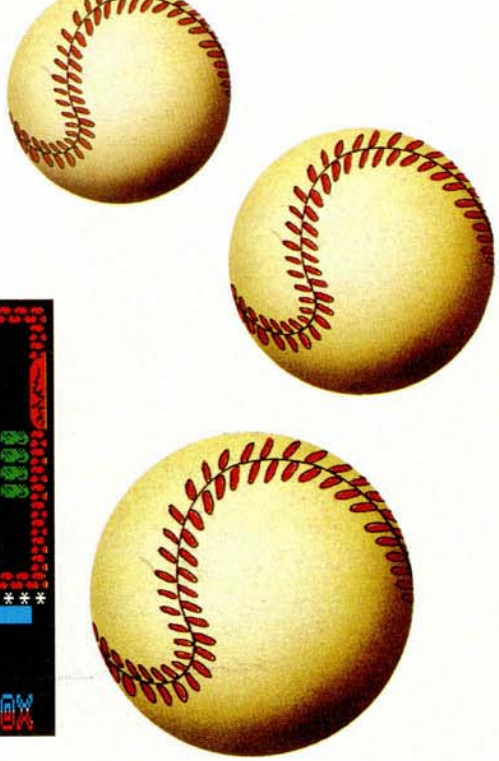



```
45  DFBBBADFBEAEAABAB3DF  1941
46  B6B1B0B9BAB1ACB6A9B0  1782
47  DFBCBEB2AFBEB2BAB1AB  1856
48  B0DFB7BEACABBEDFAEAA  1872
49  BADFAAB1DFBBB6BED3DF  1972
50  AAB1DFB560A96AB1DFAF  1857
51  ADB0A9B6B1B6BAB1ABBA  1779
52  DFBBBADFB3BEDFB8ADBE  1958
53  B1DFBCB6AABBBEBBDFAC  1899
54  BADFBEBCBAADBCB0DFBE  1923
55  B3DFBCBEB2AFBEB2BAB1  1864
56  AB80D1BAB1DFBAB3DFAF  1905
57  AAACB6BAADB0B1DFABB0  1806
58  BBBEACDFACAAACDFBAAC  1867
59  AFBAADBEB1A5BEACDFD1  1860
60  D1D1DFDFBEB7B0ADBEDF  1999
61  B7BAADABDFBAADBAACDF  1876
62  ABAADFA6DFABB6BAB1BA  1855
63  ACDFAEAABADFB8AAB6BE  1874
64  ADB3BADFAFBEADBEDFBA  1898
65  B1BCBAB1BBBAADDFBAB3  1862
66  DFB8ADBEB1DFB9AABAB8  1895
67  B0D3DFAFB0ADAEAABADF  1887
68  ACB6DFB1B0D3DFBDB6BA  1921
69  B1DFAFB05BADB6BEACDF  1875
70  ACBAADDFAFBEACAB80DF  1861
71  BBBADFB3BEACDFB3B3BE  1908
72  B2BEACDFBBBAB3DFB2BE  1906
73  B3DFD1D1D1DFB1BABCBA  1989
74  ACB6ABBEADBEACDFB3BA  1838
75  DBBEDFA6DFBCBAADB6B3  1929
76  B3BEACD3DFAFBAADB0DF  1908
77  ACB0BDADBADFABB0BBB0  1829
78  DFB1B0DFABBADFB0B3A9  1903
79  B6BBBAACDFBBBADFBAB1  1909
80  BCBAB1BBBAADDFB3BEAC  1861
81  DFBCBAADB6B3B3BEACDF  1895
82  D1D1D1DFD7ADBABCAABA  1968
83  BBBEC5DFBAB1BCBAB1BB  1898
84  BAADDFB3BEACDFBCBAAD  1893
85  B6B3B3BEACD6DFDFDFDF  2008
86  BAACABBADFAFADB0B8AD  1819
87  BEB2BEDFB7BEDFACB6BB  1918
88  B0DFADBABEB3B6A5BEBB  1851
89  B0DFAFB0ADDFBED1BAB3  1910
90  B6BCBAACDFBCB0B1DFB3  1894
91  B0ACDFB8ADBEB9B6BCB0  1849
92  ACDFBBBADFADD1B0B3B6  1910
93  A9BEDFA6DFB3BEDFBCB0  1927
94  B3BEBDB0ADBEBCB6B0B1  1820
95  DFBBBADFB5D1BAB3B6BC  1944
96  BAACD1DFDFDFBAB1DFB3  2001
97  B0ACDFBBB0B1B6B1B6B0  1828
98  ACDFBBBAB3DFB9AABAB8  1895
99  B0DFDFDFFF0000000000  1100
```

DUMP: 40.000
N.° BYTES: 985

DEMO 3

```
 10 POKE 56403,201
 20 RANDOMIZE USR 56320
 30 REM PRUEBA DE IMPRESION
       PIXEL POR PIXEL
 40 POKE 59936,0: POKE 59942,1:
LET BUFFER=33000: REM INICIO DE
L BUFFER
 50 LET DATOS=31000: REM INICIO
 DE LOS DATOS
 60 LET INICIO=32000: REM INICI
O DEL PROGRAMA EN C.M. PARA EL U
SO DESDE BASIC DE LA RUTINA
 70 RESTORE 80: FOR N=INICIO TO
 INICIO+9: READ A: POKE N,A: NEX
T N
 80 DATA 221,33,0,0,17,0,0,195,
115,233
 90 LET H=INT (BUFFER/256): LET
 L=BUFFER-256*H: POKE INICIO+6,H
: POKE INICIO+5,L: POKE 59939,L:
 POKE 59940,H+1
 100 LET H1=INT (DATOS/256): LET
 L1=DATOS-256*H1: POKE INICIO+3,
H1: POKE INICIO+2,L1
 110 REM CREACION DE DATOS
 120 RESTORE 130: FOR N=DATOS TO
 DATOS+6: READ A: POKE N,A: NEXT
 N
 130 DATA 3,2,0,0,96,184,6
 140 LET IV=1: LET IH=1: LET X=1
: LET Y=1
 150 POKE DATOS+2,X: POKE DATOS+
3,Y: RANDOMIZE USR INICIO
 160 IF Y<1 OR Y>200 THEN LET IH
=-IH
 170 IF X<1 OR X>160 THEN LET IV
=-IV
 180 LET Y=Y+IH: LET X=X+IV
 190 GO TO 150
9999 SAVE "DEMO3" LINE 10
```



CÓMO SE HACE UN JUEGO

OGEROX (IV)

En este penúltimo capítulo de la serie, veremos cómo se puede utilizar la rutina de impresión carácter por carácter desde Basic, así como una pequeña demostración del contador de tiempo.

La rutina de impresión carácter por carácter es la que utiliza la rutina mapeadora, por lo que todo lo referente a cómo se almacenan los gráficos en memoria, atributos, etc., quedó explicado en el primer artículo de la serie. Los interesados en experimentar deben antes de nada repasar aquellas explicaciones, ya que aquí sólo vamos a tratar el manejo desde Basic.

La rutina de impresión carácter por carácter permite imprimir gráficos de cualquier tamaño en cualquier posición de pantalla, con atributos definidos o con un solo atributo. Las coordenadas de pantalla y los tamaños de los gráficos deben, eso sí, coincidir con caracteres. Para imprimir un gráfico cualquiera, antes debemos inicializar una tabla situada a partir de la dirección 63.500, organizada de la siguiente forma:

Byte 0: Coordenada horizontal del gráfico. Puede variar entre 0 y 31.

Byte 1: Coordenada vertical del gráfico. Puede variar entre 0 y 23.

Byte 2: Formato vertical: es el alto del gráfico en caracteres.

Byte 3: Formato horizontal: es el ancho del gráfico en caracteres.

Si quisiéramos, por ejemplo, que un gráfico de 3 caracteres de alto por 6 de ancho fuera impreso en las coordenadas (5,7) —5 vertical, 7 horizontal— modificaríamos la tabla de la siguiente manera:

```
POKE 63500,5
POKE 63501,7
POKE 63502,3
POKE 63503,6
```

Para poder llamar a la rutina desde Basic se necesita un corta rutina-interface que en el programa de demostración se ha colocado a partir de la dirección 31.000. Esta rutina ocupa 6 bytes y contiene la dirección a partir de la cual está situada la definición del gráfico que se va a imprimir. Se puede asumir que la dirección de definición del gráfico está almacenada en la

CARGADOR 4

```
  30 LOAD "GRA_OG.RES"CODE 53814
,2500: LOAD "GRA_PRES"CODE 33088
,800: LOAD "UDGs"CODE 32300,768
  60 LOAD "DEMO4"
9050 SAVE "CARGADOR4" LINE 10: S
AVE "GRA_OG.RES"CODE 53814,2500:
 SAVE "GRA_PRES"CODE 33088,800:
SAVE "UDGs"CODE 32300,768
```


AZUL

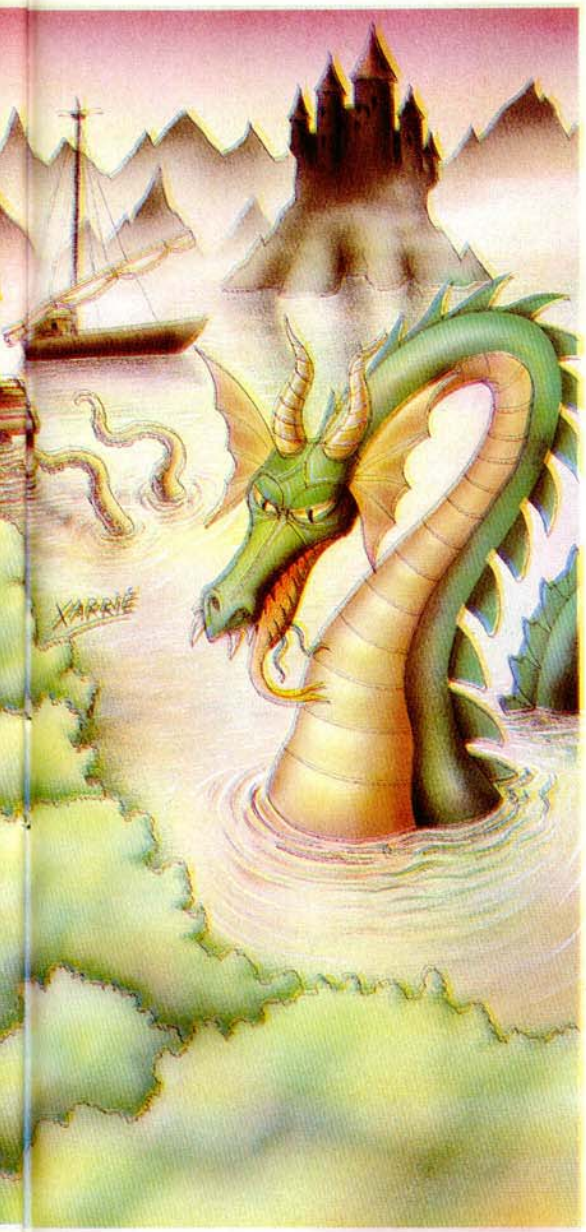
variable situada en la dirección 31.001. La rutina-interface es reubicable y puede cambiarse a cualquier dirección de memoria con tal que esté libre. En caso de cambio, la dirección del gráfico estaría definida mediante la variable (utilizamos la palabra variable, no demasiado precisa, por analogía con los capítulos anteriores) en la dirección *rutina-interface + 1;* siendo *rutina-interface* la dirección a partir de la cual pongamos la rutina-interface.

En el programa Basic de demostración, basta con cambiar el valor de la variable *DIR*, para cambiar la dirección del gráfico (línea 90). Para imprimir el gráfico en pantalla sólo hay que llamar a la rutina-interface mediante: RANDOMIZE USR 31.000

En cuanto al contador de tiempo, no se trata de una rutina muy fácil de utilizar ni de adaptar a nuestras necesidades, por lo que el programa de demostración sólo muestra cómo funcio-

GRA . OG. RES

```
 1  050200000003C7F7F713B 493
 2  00000000FFFFBF1F1F00 763
 3  0F000F0F010EFF00FF00 570
 4  FFFFFFFF0E0E0E0E0E0E 1104
 5  0E0EFFFFFFFFFFFFFFFF 2068
 6  0E0E0E0E0E0E0E0EFFFF 622
 7  FFFFFFFFFFFF0E0E0E0E 1586
 8  0E0E00FFFFFFFFFFFFFF 1813
 9  FFFF0404040404040404 542
10  0404040500000102030 25
11  050700008040F9FFFFFF 1218
12  003C6EDFBF3FFFFF0000 1157
13  01029FFFFFFF000080C0 1247
14  C0C0E0E00050705070502 863
15  1D2EBFFFBFDFEFF77B8F 1735
16  3C99DBFFC3813CFFFDFB 1830
```




```
17  FDFBF7EFDFFFE0E0E0E0 2364
18  E0C0B85C576F361D150A 1004
19  05025BAD3BF5FFEB75BA 1368
20  FFFFFFFFFFFFFFFFDEBD 2451
21  DCAFDFF5AA55BE5E3CB8 1646
22  B870E0C0010000000103 717
23  050E7DBCDEDFBF7FEBD4 1542
24  7EFF000000000000AB37 607
25  6EDFBFFFD62B80000000 1164
26  80C0A070040404040404 616
27  043C04040404040404 96
28  0404040407050000000 28
29  0000000000000000406 10
30  0301000000000000C3 199
31  0000000020600C0800000 448
34  00004427E7E5A663C2418 630
37  0000000000010700F1F3B 113
38  3F77FFFFFFFFE7A5E700 1829
39  0080E0F0F8DCFCFC00000 1564
40  000000000000007153 295
41  0B0F0E07E3C181000000 596
42  0103FFC3BDBDFF81FFFF 1726
43  DC9C9C0C0E0687CF0000 906
45  0000000934B32B24505 524
46  0505FFFFFFF7E3C1C1C1 1828
47  C7A6A6A0D0D0D0D00000 1523
49  000000000507000F0F0F 57
50  0F0FC180008080808080 991
51  D0F000F0F0F0F0F00000 1648
53  0000030000F1F7FFFFF00 686
54  FF0000000101010FFF00 513
55  F0F8FCFCFC00FF000000 1499
56  0000000C0000707070707 220
57  0707078507070703030303 56
58  0702020302070702020202 36
59  0707020202070606060606 51
60  0604030000000000000100 14
61  00000970A8F4BAEAF500 1189
63  083C1E07037D322E5E6F 550
64  5DBAF5008080804040080 1164
65  4000000000000000102FB 318
66  759B76AA7DAA6A804040 1217
67  C0A0A060A00D15354A60 1025
68  75241B54549629A41173 835
69  8C90D0A828540432CC06 1048
70  0607060606060606060606 61
71  0604030000000000000000 13
72  0124246642423C3C3BD00 886
73  0000000000000080010103 133
74  030305040BBDDB7E66BD 851
75  99DB3C8080C0C0C08040 1456
76  400A0A0151925364A2A55 422
77  5D55A2949455296060050 1034
78  90B0C8A8282D25554A48 1041
79  34110E4D4D72884992E6 936
80  19042CD424485060800006 703
81  0606060606060606060606 60
82  060505325DBA5FAF7FB7 925
83  5AE8D7EB57FFFFDCB3A8 1936
84  5EFDFEFEFF017E63FFFF 1846
85  7B7FF437F7F0FFFF2A9E 1746
86  DF9FB9EF5EAF5A6D5B56 1547
87  75CDBE57EC70A04080A5 1464
88  BD810000000000B56EDB 828
89  370E0503012AAD66B7F6 824
90  D26B7AEAEFD9D6DEC0BC 1945
91  AC8000000000000000000 300
92  0000000000000010000 1
93  000000000D5DABD6F7F 858
94  073F2FB4B4BC80BCBCB4 1349
95  AC0000000000000000000 172
97  00000000003373F2F3F03 231
98  3F2F37BC80BCACACB4AC 1365
99  BC0000000000000000000 188
101  0000000003F3F033F2F 239
102  3F373F0202020202020202 195
103  0202020202070202020207 30
104  0707020207070702 0A05 56
105  556E5B5B333161610000 671
106  0000008080800000000000 384
109  6161616173B3D3EE87B3 1445
110  BCBF8FBF7FFFC030E830 1663
111  88D0D7D77FFFFFFC7C7C 1911
112  FCFCF0FEFF0000000000 1253
113  FFFFFFFFFFFFFFFFFFFF 2550
114  FFFFFFFFFFFFFD7D7D7D7 2390
115  C03EFFFFFFCFCF8F00000 1756
117  FFFFFFFFFFFFFFFFF0F0 2520
118  F8FCFEFFFFFFF1F070301 1561
119  03FFFEFFF8080808000000 1280
120  800000000000000000000 128
121  FFFFFFFFFFFFFFFFF8F7 2535
122  EFDFDFDFEFF70FF3FDFE 2159
123  9F0381FF0080C0E0F0F8 1578
124  FCF8000000000000000000 500
125  FFFFFFFFFFFFFFFFFFFF 2550
126  FFFFFF8CDFE7E0E0E0E4 2307
127  E4E606CE0000000000000 670
129  FFFFFCFEFFFFFFFFCF0F 2258
130  0F0F1F9FBFBFECECE9EC 1327
131  BC3C7C7C0000000000000 496
133  FFFEFEFF9F78F7FF07EFD 2148
134  FBF7EFDF3FFFB8D0E0F0 2134
135  F0F0F0F0000000000000 960
137  0FFFFFFFFFFFFFFFFFFF 2310
138  FFFFFFFFFFFEF0F0E0E0 2457
139  D0B078F8000000000000 752
141  00FFFF00FF10FF0001FB 1288
```

```
142  FD00FF14FF00FCFEFE7F 1670
143  9F00FF0000FFFF00FF14 1199
144  FF0000F0F000FE02FE00 1245
145  0202060606020202070 42
146  0202020706023232020 129
147  0202020206020202060 32
148  0202020606020202060 36
149  0202020606020202020 28
150  0E0800001F0703000101 65
151  000000E0FC7F0FC30000 813
152  000000C0FCFE00000000 698
154  0000000000000033F7F 193
155  000000073FFEF0C30000 759
156  F8E0C0008080000000 920
157  0000000E0783C3878D8 796
158  F8D8FF3F3F3E7E7CFCF8 1657
160  0000000000000FFFCFC7C 883
161  7E3E3F1F071E3C1C1E1B 464
162  1F1B0000000000000000 58
163  0303060706D0E1DB858 353
164  B858858AC5CF8FCFFFF 1818
165  7F1F07010007FF7FBFB 937
166  FBF700E0F0EFDFDFDFEF 2109
167  1F3FFFFFFEF8E0801D1A 1513
168  1D1A1D1A353AC0C060E0 925
169  60807 0B81A1D1A1D1A35 757
170  3A35AE56AB55AA55AA55 1137
171  0000008 0E0708B85F691D 877
172  1F0F0F0703036B8F8F0 896
173  F0E0C0C0000000010 70E 870
174  1DFA756AD5AA55AA55AA 1395
175  58B858B858AC5CAC3A35 1179
176  3A353A353A35AA55AA55 843
177  AA55AA55AF55AA55AA55 1283
178  AE5DF6FDBAF5EA55AA55 1771
179  6FBF5DAF57AA55AA55AA 1337
180  55AAD5EAF5FA55AA55AA 1707
181  55AA55AA5CAC5CAC5CAC 1302
182  5CAC3A353A353A353A35 708
183  AA55AA55AB57AF5FBA75 1341
184  EAD5EAD5AA55AF7FAF5F 1721
185  AF5FAF5FF5FEF5FAF5FA 2029
186  F5FA5DAE57AB57AB55AA 1533
187  55AA55AAD5EAF5FA5CAC 1716
188  5CAC5CAC5CAC3A353A35 1014
189  3A353A35AE5DBA5DBA75 1071
190  BA75AA55AA55AF5FBC78 1391
191  AF5FBF5FBFDFFF7FF5FA 1847
192  FDFAFDFBFFFE55AA55AA 2026
193  F5FA3D1E75BA5DBA5DAE 1435
194  5DAE5CAC5CAC5CAC5CAC 1323
195  3A353A353A353A35EA75 795
196  EAD5EBD5EBD5B060E0C0 2031
197  C08080803F1F1F1F0F0F 762
198  0F0FFCF8F8F8F0F0F0F0 1986
199  0D060703030101057AE 296
200  57ABD7ABD7AB5CAC5CAC 1558
201  5CAC5CAC1A1D1A1D1A1D 693
202  1A1DAB5FBE58B070E060 1207
203  0000000000000001F1F 62
204  1F3F3F3F7F7FF8F8F8FC 1470
205  FCFCFEFE0000000000000 1012
206  0000D5FA7D1A0D0E0706 654
207  58B858B858B858B81A1D 1143
208  1A1D0E0D0E0DE060C0C0 813
209  C0C0C0C000000010101 771
210  0103FFFFFEFEFEFCFCFC 2032
211  FFFF7F7F7F3F3F3F0000 1080
212  0080808080C007060303 723
213  0303030358B858B858B8 628
214  58B80F0D0F0D07070707 356
215  0000000000000000303 6
216  0707070303030FCFCF8F8 1030
217  F8F8F8FC3F3F1F1F1F1F 1246
218  1F3FE0E0C0C0C0808080 1502
219  0000000000000000F0B0 416
220  F0B0E0E0E0E0E007070303 1332
221  0301010100000000000 6
222  0000010100000000000 2
223  FCFCFE7E7E3E3E1E3F3F 1290
224  7F7E7E7C7C7800000000 747
226  0000E0E0C0C0C0C0808080 1408
229  000000001E1E1E1E1E1E 180
230  1E3E787878787878787C 1056
235  0000000001010303 3E7C 194
236  7CFCF8F8B8B887C3E3E3F 1551
237  1F1F1D1D0000000000000 376
238  C0C0000000000000000000 384
239  0000000000000000000606 12
240  0602020606060702020607 50
241  0706020202020202020206 37
242  0202020202020202020202 20
243  0202020202020202020202 20
244  0202020202020202020202 20
245  0202020202020202020202 20
246  0202020207020207020202 30
247  0202020202020202020207 25
248  0202020202070202070202 30
249  0202070202070702020207 50
250  0707070702020202070 7 50
```

DUMP: 50.000
N.° BYTES: 2.500

na. Todo lo más que podemos hacer es cambiar el valor inicial que toma el contador, que en un principio es 9999. Esto puede hacerse sabiendo que los dígitos se almacenan a partir de la dirección 64.0044, en direcciones alternas. El dígito más a la izquierda se almacena en el byte de la dirección 64.0044, el siguiente en la 64.0046,... y el último en la 64.0050. Cada dígito puede tomar cualquier valor entre cero y nueve.

Para ver la demostración de la rutina de impresión carácter por carácter hay que ejecutar el programa de demostración desde su principio mediante:

RUN

Para la demostración del contador de tiempo:

RUN 210

El que quiera cambiar el valor inicial sin complicarse la vida, simplemente tiene que modificar la línea 240:

240 DATA *d1,d2,d3,d4*

Siendo *d1, d2, d3* y *d4* los dígitos de izquierda a derecha.

Ni que decir tiene, que antes de poder ejecutar el programa de demostración hay que copiar el programa cargador, mezclarlo con los anteriores y salvarlo antes de los bloques de Código Máquina; teclear los bloques de Código Máquina de este número con el cargador universal y salvarlos a continuación de los anteriores y por último teclear el programa de demostración. Rebobinando la cinta y cargando desde el programa cargador se ejecuta la demostración de la rutina de impresión. Pulsando STOP podemos interrumpir el programa y ver ahora de demostración del contador de tiempo con RUN 210.

El procedimiento completo es exactamente igual que en los artículos anteriores y se haya descrito paso a paso al final del capítulo segundo.

Con esto termina la penúltima parte del juego. En el próximo capítulo veremos cómo utilizar la rutina de impresión de un mensaje, a doble alto y con scroll pixel por pixel y una de scroll vertical también pixel por pixel de ventanas. Por supuesto, con esa última parte el juego completo estará ya tecleado y listo para funcionar. Veremos qué hay que hacer exactamente, las teclas de movimiento y algunos trucos y pokes. Pero todo esto será en el próximo número. Hasta entonces sed buenos y terminad de teclear esta parte...

Alberto Elices
Roberto Oliva
Javier Elices

GRA - PRES

```
 1  000307070707030000C0  226
 2  E0E0E0E0C00001010101  1092
 3  0101010080808080808080  771
 4  8000020202020142B2A  241
 5  261A2B2600684854D428  657
 6  B454160A0A1515150E05  388
 7  54BC485454A8B0500202  940
 8  0202000103030602010020  20
 9  80C0E0602020C0000101  898
10  0101010101080808080  517
11  8080800006060202 0F30  463
12  474890A0A0A0FF00FF00  1277
13  000000FFFF00FF000800  773
14  00FFFF00FF00110000FF  1037
15  FF00FF009C0000FFFF00  1176
16  FF00E70000FFFF00FF00  1251
17  E7E700FFF00CE21289E5  1579
18  6535A1A1A1A1A1A1A1A1  442
19  FF030905F501457DFF03  970
20  0905F1054597BFF831905  872
21  11115555FF810D015F41  762
22  25BDFF1155555501457D  948
23  FFC18D011F01457DB5B5  1178
24  85B585B5B5B5B5A1A1A1A1  1634
25  A1A1A1A17D4501015555  1010
26  55117B4501015555551 1  568
27  5555551101051983BD25  660
28  41411F010D817D450105  504
29  8B8BC7C77D450101F505  1122
30  1B0785B585B5B585B5B5  1338
31  A0A0A0904847300FFF00  1085
32  000000FF00FFFF000000  773
33  00FF00FFFF00001100FF  1037
34  00FFFF00009C00FF00FF  1176
35  FF0000E700FF00FFFF00  1251
36  E7E700FF00FF3565E589  1492
37  12E20CF0040404040404  520
38  040404060606060604  52
39  0405050505050504040 4  46
40  0404040404040 1F204F  166
41  5050505000F804F20A0A  834
42  0A0A001F204F50505050  482
43  00F804F20A0E0000001F  549
44  204F50504F4000F804F2  908
45  0A0E8040007F404F5050  646
46  4F4000F008F40A0AF408  907
47  001F204F5050505000F8  710
48  04F20A0A0A0A00205028  438
49  140A05040004 0A142850  193
50  A020505050504F201F00  654
51  0A0A0A0AF204F8005050  694
52  50504F201F007E427A0A  626
53  F204F8004F5050504F20  924
54  1F0000000E0AF204F800  549
55  5F5050505050 7000D028  855
56  2814140A0E0050505050  424
57  4F201F000A0A0A0AF204  428
58  F80004050A1428502000  439
59  20A0502814 0A04000202  350
60  0202020202020202020202  20
61  0606060606060606060606  60
62  060620548A44214E3B1E  534
63  02051F38B4E3FF550814  869
64  FFA2410855A22050FF8A  1242
65  04205 58A40A0F81C2DC7  1003
66  FFAA042A51228472DC78  1172
67  191519325432191500 00  301
71  98A8984C2A4C98A81F1D  1046
72  251349A44830A21449A2  830
73  FF1408008A51248AFF51  1012
74  2000284592 28FF458200  781
75  A21449A2FF140800F8B8  1132
76  A4C89225120C04040404  593
77  0404040606063804040 4  98
78  0404040400000000000 0  16
80  00000000000000000000  0
```

DUMP: 40.000
N.° BYTES: 800

UGDS

```
 1  0000000000000000183C  84
 2  3C1818001818363612 24  318
 4  00003C0E32383232323232  380
 5  6092640818256490000000  483
 6  0000000000000606020 4  18
 7  00000000060C1C1C1C1C  130
 8  0C06603038383830606  530
 9  000014083E8814000018  142
10  187E7E1818000000000000  324
11  3030102000000000 3C3C  264
12  0000000000000606000  192
13  0002060C183060001824  248
14  0R4A5256241 80C1C003C  412
15  0C0C0C1C1C26000C1830  214
16  707E1C26001C0606261C  410
17  1E2600467E0606063E3E  406
18  0038040646 3C1C360040  342
19  7C4624183E26000C1830  438
20  3030182404182466663C  484
21  1C26201E0606261C0000  206
22  6060606060000606000  576
23  6060204000000C183018  396
24  0C00000003C003C3C00  192
25  000030180C1830001826  218
26  00061C0018181818003 8  186
27  6000641 83C3E023E3232  506
28  32323C3E023E32323232C  496
29  1C3E003030303 21C383C  428
30  0232323234383E3E003C  444
31  3030303E3E3E003C3030  486
32  30301C3E00363632321C  422
33  3232003E323232321818  410
34  0018181818181C3E0006  216
35  06263E1C323400303834  392
36  3232303000303030303E  450
37  7C7E006A6A6A62623C3E  886
38  0032323232321C3E0232  392
39  32323E1C3C3E023E3030  472
40  30301C3E023232343A1A  424
41  3C3E023E34343232 1C26  456
42  0236043234187E7E0018  464
43  1818181832320032323 2  346
44  321C3232003232321400  356
45  626200626A6A6A3C3232  772
46  0032140834322626003E  318
47  06263E1C3E3E000C1830  342
48  3E3E000000000000000 0  124
51  000000000080C7E7E0C  284
52  0800000000000000000 0  8
56  00007C82BE88B8BE827C  1208
57  0000000000000007C82  254
58  BEA2BABA827C00000000  978
64  00007C82BAAAAABA827C  1220
66  000000000000F884B2A4  722
67  88A4B2EE0000000000 00  716
71  00000000C3A55A242 45A  612
72  A5C3000000000000000 0  360
77  3C4299A1A199423C0000  880
```

DUMP: 40.000
N.° BYTES: 768

DEMO 4

```
  10 REM PRUEBA DE IMPRESION
      CARACTER POR CARACTER
  20 POKE 56403,201: RANDOMIZE U
SR 56320
  30 POKE 63500,0: REM COORDENAD
A "X"
  40 POKE 63501,0: REM COORDENAD
A "Y"
  50 POKE 63502,14: REM FORMATO
VERTICAL
  60 POKE 63503,8: REM FORMATO
HORIZONTAL
  70 FOR N=31000 TO 31005: READ
A: POKE N,A: NEXT N
  80 DATA 17,0,0,195,125,244
  90 LET DIR=55306: REM DIRECCIO
N DEL GRAFICO
 100 LET H=INT (DIR/256): LET L=
DIR-H*256: POKE 31001,L: POKE 31
002,H
 110 PRINT AT 0,0;"SI QUIERES CO
LOCAR EL GRAFICO   JUSTO A CONTI
NUACION DEL OTRO,  PULSA 99"
 120 INPUT "C_X ";X
 130 IF X=99 THEN GO TO 170
 140 INPUT "C_Y ";Y
 150 POKE 63500,X
 160 POKE 63501,Y
 170 INPUT "QUIERES DEFINIR COLO
R ";O$
 180 IF O$="S" OR O$="s" THEN IN
PUT "QUE COLOR ";C: POKE 63556,C
*(C<256): RANDOMIZE USR 31000: G
O TO 120
 190 POKE 63556,0: RANDOMIZE USR
 31000
 200 GO TO 120
 210 REM PRUEBA DE IMPRESION DEL
 CONTADOR
 220 RESTORE 230: FOR N=64044 TO
 64044+7 STEP 2: READ A: POKE N,
0: POKE N+1,A: NEXT N
 230 REM DIGITOS DEL CONTADOR
 240 DATA 9,9,9,9
 250 RANDOMIZE USR 62179: PAUSE
5: GO TO 250
9999 SAVE "DEMO4" LINE 10
```

CÓMO SE HACE UN JUEGO OGEROX (y V)

Hemos llegado ya al final de la serie. Hoy explicaremos la utilización de un par de rutinas, para pasar inmediatamente a explicar en qué consiste el juego, cómo jugar, algunos pokes...

El par de rutinas que quedan por ver, casi no necesitan explicación ya que el programa de demostración lo hace todo. La primera de ellas hace aparecer pixel por pixel y a doble tamaño, un mensaje por las dos líneas inferiores de la pantalla. Para cambiar el mensaje basta con modificar en la línea 60 el valor de la variable *a$*. El mensaje puede ser tan largo como se quiera —mientras la memoria lo permita— y debe escribirse con mayúsculas (para que aparezcan los caracteres empleados en el juego) y terminar con un *CHR$ (255)*. El ejemplo está en el programa de demostración.

La otra rutina hace scroll de abajo a arriba de una ventana de pantalla, definida por caracteres, encargándose de borrar las líneas que van apareciendo. En el juego se utiliza para abrir las puertas. El que quiera averiguar en qué direcciones se almacenan los formatos y coordenadas de las ventanas, así como la forma de llamar a la rutina, puede hacerlo examinando el listado Basic del programa de demostración, desde la línea 100 en adelante. Una explicación detallada no tiene demasiado interés. ¡Lo importante es el juego completo!

Antes de nada, el juego tiene una pantalla final (la pantalla que aparece cuando hemos conseguido llegar al final del juego), que no se ha incluido como listado hexadecimal debido a su excesiva longitud incluso compilada. Además, es mucho más interesante el que cada uno se haga su propia pantalla final y personalice un poco el juego. Para realizar la pantalla final se puede utilizar un programa en Basic que la dibuje y luego salvarla o bien uno de los muchos programas comerciales de dibujo que existen. Lo único importante es que la pantalla se debe salvar *sin compilar*.

Hay muchas otras cosas que cada uno puede personalizar. Hemos explicado cómo se almacenan las pantallas en memoria; nada impide modificar todas aquellas pantallas que no tienen trascendencia en el juego. Más exactamente, no se pueden modificar las pantallas número (la primera es la cero): 4, 6, 8, 9, 10, 11, 17, 19, 21, 22, 24, 27, 30, 31, 32, 34, 35, 37 y 39.

El mapa forma una matriz de ocho pantallas de ancho por cinco de alto, organizado de izquierda a derecha y de arriba a abajo. La cero es la que ocupa la esquina superior izquierda, la uno la que está a su derecha, etc.

En el caso de los sprites, pueden cambiarse todos y cada uno de los sprites de cualquiera de las pantallas. Se puede cambiar su forma (número de gráfico) teniendo cuidado de que su tamaño no le haga borrar trozos de pantalla; y se puede cambiar su recorrido: límites entre los que se mueve e incrementos horizontal y vertical. No se puede cambiar ningún objeto de pantalla, pero sí de posición dentro de su pantalla.

Hemos explicado cómo se organizan los gráficos, tanto de sprites como de mapeado. Si alguien tiene la paciencia y los conocimientos suficientes, puede si lo desea cambiar gráficos.

Dado que hemos ido explicando todas estas cosas, cualquiera puede modificar a su antojo lo que se le ocurra. Una advertencia: cuidado con lo que se cambia; antes de cambiar algo hay que estar seguro de que se ha entendido qué es ese algo y cómo está organizado.

Por último, el juego utiliza unos caracteres distintos a los del sistema. Los caracteres que están definidos y que se pueden cambiar son todos los ASCII, desde el código 32 hasta el 127, ambos inclusive. Los nuevos caracteres se encuentran almacenados a partir de la dirección 32.300 y ocupan $96 \times 8 = 768$ bytes. En las direcciones 32.300 a 32.307 está definido el carácter 32, el primer ASCII, que es el espacio. Para cambiar el juego de caracteres lo más práctico es poner el nuevo juego encima, aunque también se puede cambiar la variable del sistema correspondiente, que es CHARS (direcciones 23.606 y 23.607).

Para obtener una copia del juego completo, que ya se utiliza para el quinto programa de demostración, basta con contestar afirmativamente a la pregunta del programa de demostración de *"Quieres salvar el juego completo (S/N)"* y seguir las subsiguientes instrucciones (entre las que se encuentra cargar la pantalla final). En caso de haber hecho modificaciones, éstas deben estar salvadas dentro de los bloques de Código Máquina o, en el caso de ser unos pocos pokes, hacerse al principio del programa *cargador 5*, justo después de cargar el último bloque de Código Máquina. El programa salvará al cassette un solo bloque que contiene todo el Código Máquina, gráficos, pantallas, etc. Para ejecutar el programa basta con hacer:

```
POKE 23606,44
POKE 23607,125
RANDOMIZE USR 56320
```

o bien incluir antes del bloque de *bytes* que contiene todo el juego, el programa *cargador 6*. Este programa se encarga de cambiar el juego de caracteres (los dos pokes) y de ejecutar el juego, sacando un mensaje para la carga.

En cuanto al juego propiamente dicho, en primer lugar se pueden utilizar para jugar tanto el teclado como un joystick tipo Kemston. Para escoger uno u otro sólo hay que pulsar la tecla correspondiente como en cualquier juego. En el caso del teclado las teclas son las siguientes:

O: izquierda
P: derecha
Q: salto largo
A: salto corto
1: parar
0: continuar

En cualquier caso, con 'SPACE' se termina la partida (muy útil si caemos en alguna trampa).

La historia del juego se encuentra al principio del capítulo primero de esta serie y dentro del propio juego. El juego tiene además algunas pistas; sólo hay que leer el mensaje que aparece cuando cesa la música (se puede hacer cesar la música pulsando cualquier tecla).

La misión del juego

El objetivo a grandes rasgos es el de encender el gran fuego, para lo cual hay que hacer algunas cosas primero... Hay varias llaves que abren las correspondientes puertas, cuya cerradura es la misma llave que la abre parpadeante. En cada pantalla hay tres enemigos que nos restan energía al tocarnos y siguen trayectorias definidad. No siempre es fácil evitar que nos rocen, pero tenemos bastante energía.

No vamos a explicar cómo se termina el juego, eso es cosa vuestra; lo que sí vamos a hacer, para que no os volváis locos es daros algunos pokes que facilitarán en gran medida vuestro trabajo:

Energía infinita:	POKE 61506,201
Número de sprites:	POKE 59942,n

siendo n un número entre 1 y 4. El número de sprites incluye al personaje principal, por lo que 4 es lo normal y 1 es sin ningún enemigo.

Tiempo infinito:	POKE 62332,201
Sin sonido:	POKE 61792,201

Hay muchos más, pero también queremos dejar algo a los buscadores y destripadores.

Sobre cómo jugar, algunos consejos son los siguientes:

1. En el juego hay un gran número de trampas de las que no se puede salir. Muchas de ellas no tienen apariencia de trampas. Ve con cuidado, calcula bien los saltos y recuerda que siempre puedes pulsar 'SPACE' para volver a empezar (la única solución).

2. Todas las puertas tienen una cerradura en su misma pantalla. No es fácil adivi-

nar «qué es lo que se va a abrir» pero siempre te dará acceso a nuevas pantallas. Las cerraduras son exactamente iguales a las llaves que las abren, salvo que parpadean. Para abrir una puerta basta con tocar la cerradura llevando la llave correspondiente. Asegúrate, eso sí, de que llevas la llave, no te equivoques de color; tocar una cerradura sin tener la llave quita mucha energía.

3. Los pocos objetos que tiene el juego tienen todos alguna finalidad.

4. El personaje principal —Hert— tiene dos saltos diferentes, uno corto y otro largo. Trata de utilizar el salto necesario en cada momento.

5. Procura no dejar objetos atrás. Muchos caminos son sólo de ida.

6. No te preocupes si no consigues superar todos los obstáculos sin perder energía.

En algunos hay que perder energía y están calculados así. Con la energía que tienes al comenzar el juego hay más que suficiente.

Y esto es todo. Esperamos que el juego sea de vuestro agrado. El estilo es ya clásico, pero tiene bastante calidad. Que lo disfrutéis.

Alberto Elices
Roberto Oliva
Javier Elices

CARGADOR 5

```
  40 LOAD "OG_2-1"CODE 59300,463
: LOAD "OG_2_2"CODE 60074,336: L
OAD "OG_1"CODE 60500,1307: LOAD
"PR_OGER"CODE 56320,2980: LOAD "
OG_MENS"CODE 59070,32
  60 LOAD "DEMOS"
  70 REM PARA CAMBIAR EL JUEGO DE
 CARACTERES, EJECUTAD LA LINEA S
IGUIENTE
  80 POKE 23606,44: POKE 23607,1
25
9060 SAVE "CARGADOR5": SAVE "OG_
2-1"CODE 59300,463: SAVE "OG_2_2
"CODE 60074,336: SAVE "OG_1"CODE
 60500,1307: SAVE "PR_OGER"CODE
56320,2980: SAVE "OG_MENS"CODE 5
9070,32
```

OG_2-1

```
  1  23232323237E320EF823 648
  2  7E320FF82B2B2BCDBBE7 1191
  3  C343E87EE5234E26006F 1111
  4  291113F8193A0FF8473A 800
  5  0EF8F556235E23E52600 1024
  6  69193E08F5AFC5E57723 1200
  7  10FCE124C1F13D20F1E1 1522
  8  F13D20E0E17E234E2600 1050
  9  6F2929292929211005819 445
 10  160059193A0FF8473A0E 600
 11  F80E06F5C5E5712310FC 1355
 12  E101200009C1F13D20EF 1033
 13  C9CB772022E63F5F237E 1138
 14  320CF8237E320DF87BF6 1151
 15  083244F8113CFA3E0232 815
 16  0EF83D320FF8CD7DF43A 1268
 17  43F8DD2152FAC92A3AFA 1452
 18  7ECB7F202308214CFAAF 1065
 19  0E00BE2804230C18F979 689
 20  FE05C8FE0028085A5D2B 981
 21  0600EDB823087C330E9 1065
 22  214CFAE67F0E00BE2809 969
 23  57AFBEC87A0C2318F454 1173
 24  5D233E05914F0600EDB0 838
 25  2A3AFA2323237E57E8787 1080
 26  87234623577E878787782 1023
 27  320CF8234E583A0CF892 975
 28  477A3CC5D9014820857CD 1072
 29  60F10188130B78B120FB 1084
 30  7AD94326006F2D291150 738
 31  F81956235E2600681922 689
 32  2AFAF5F5260006F291150 1069
 33  F81956235E26006819ED 892
 34  5B2AFA222AFAC50600ED 1149
 35  B0C1F13C573A0CF88A7A 1383
 36  20D92A2AFA3600ED5B2A 1007
 37  FA13C506000DEDB0C1F1 1332
 38  58C1109BE123237E320E 937
 39  F8237E320FF82B2B2B11 868
 40  42E9D5C3EFE721E80301 1446
 41  E8037ED3FE230B78B120 1201
 42  F7AFD3FE2A3AFACBF623 1721
 43  3E02320EF83D320FF8CD 955
 44  BBE7214CFA11685A0105 994
 45  00EDB0214CFA11885A01 1016
 46  0500EDB03A8FFFC60332 1125
 47  8FFFC900000000000000 599
```

DUMP: 45.000
N.° BYTES: 463

OG_2_2

```
  1  3EF7DBFEE601200A3EEF 1356
  2  DBFEE601280218F63A46 1144
  3  F8E6402807ED7B90FFC3 1543
  4  98DE3A46F8E6202807ED 1296
  5  7B90FFC38DDE3A4FF8FE 1719
  6  2520383A92FFCB472831 947
  7  CB57202DDD7E02FE4020 1066
  8  26DD7E03FE70201F1188 970
  9  81011B13CDC7EB3A92FF 1274
 10  CBD73292FF3A8FFFC603 1526
 11  328FFF21D3FFC3DEEB3A 1657
 12  4FF8FE1520543A92FFE6 1407
 13  01204D11408101100SCD 547
 14  C7EBDD7E02FE18203DDD 1375
 15  7E03FE803836FE913032 1118
 16  3E02320EF8320FF8210C 734
 17  F8360523361028CDBBE7 1078
 18  114081011B13CDC7EB21 929
 19  93FF3A92FFCBC73292FF 1714
 20  3A8FFF3C328FFFC3DEEB 1616
 21  3A4FF8FE1EC03A92FFE6 1550
 22  02C0116481010F08CDC7 868
 23  EBDD7E02FE50C0DD7E03 1460
 24  FE78D8FE88D03E02320E 1316
 25  F8320FF8210CF8360823 951
 26  360F2BCDBBE711648101 982
 27  1D13CDC7EB2183FF3A92 1358
 28  FFCBCF3292FF3A8FFF3C 1632
 29  328FFF18173244F87832 1031
 30  0CF87932 0DF83E02320E 820
 31  F8320FF8CD7DF4C93E16 1420
 32  D73E10D7AFD73E11D73E 1254
 33  00D73E10D73E07D70620 830
 34  7ED72310FBC900000000 844
```

DUMP: 45.000
N.° BYTES: 336

OG1

```
  1  F3ED56DD2152FA3A45F8 1527
  2  CB873245F83A46F8E6C0 1503
  3  3246F83E7FDBFEE601CA 1463
  4  00DC3A92FFCB5FC2B5F6 1598
  5  3EDFDBFEE60120173EFB 1357
  6  DBFEE601CAE0F03EFDDB 1904
  7  FEE601CAE9F0CDA9ED18 1795
  8  313EDFDBFEE60220173E 1156
  9  FBDBFEE601CA20F13EFD 1745
 10  DBFEE601CA29F1CD57ED 1717
 11  18123EFBDBFEE601CAAD 1434
 12  F03EFDDBFEE601CAB3F0 1880
 13  3A46F8CBDF3246F8CD2B 1418
 14  EE3A46F8CB4728465DD35 1272
 15  02CB572828CB4F2812CD 917
 16  2BEE3A46F8CB472831CD 1225
 17  57EDCD6FF118EECD2BEE 1629
 18  3A46F8CB47281FCDA9ED 1332
 19  CD6FF118EE3A46F8CB9F 1557
 20  3246F8CD2BEECD6FF13A 1469
 21  46F8CB47280218F1CD6F 1215
 22  F13A45F8CB47C257ECCB 1610
 23  4F20122190B8DD7504DD 1053
 24  7405119EFACD73E9C357 1361
 25  EC2150B9DD7504DD7405 1218
 26  119EFACD73E9C357ECDD 1717
 27  7E03FE00CCD7F23D3243 1222
 28  F8CDFEEE3A43F828013C 1419
 29  2A4BF8DD7504DD7405DD 1270
 30  7703CD0EF0F53A46F8CB 1405
 31  5F280BDD46020E05CD60 759
 32  F1F1180AF12007066 40E 916
 33  05CD60F1119EFACD73E9 1525
 34  3A45F8CBC7CB8F3245F8 1490
 35  C9DD7E03FEFFCC7F2C6 1887
 36  103243F8CDFEEE3A43F8 1451
 37  280139DD60F2A4DF8DD75 1036
 38  04DD7405DD7703CDDAEF 1351
 39  F53A46F8CB5F280BDD46 1261
 40  020E05CD60F1F1180AF1 1079
 41  200706640E05CD60F111 723
 42  9EFACD73E93A45F8CBC7 1738
 43  CBCF3245F8C9DD7E02FE 1581
 44  00CCA4F23D3243F8CD67 1344
 45  EF3A43F828013CDD7702 1055
 46  DD46020E05CD60F13A46 982
 47  F8CB5FC0119EFACD73E9 1716
 48  C93A46F8CBC73245F8DD 1568
 49  7E02FE68CCBBF2C61832 1391
 50  43F8CD67EF3A43F8280B 1286
 51  3DF53A46F8CB873246F8 1388
 52  F1D617DD77023A43F8FE 1447
 53  1828063A46F8CB5FC011 953
 54  9EFACD73E9DD46020E05 1273
 55  CD60F1C957CB3FCB3FCB 1565
 56  3F4F7AE6072802 1801AF 743
 57  26006B291150F819473E 689
 58  0790878787C64632A5EE 1277
 59  0608C556235E23E52600 728
 60  6919CB462006E1C110EE 1113
 61  A7C9E1C137C9F5DD7E03 1637
 62  CB3FCB3FCB3F4FDD7E03 1227
 63  E607473E089047F16F26 983
 64  00291150F81956235E26 664
 65  00691978FE082816AF37 804
 66  1710FC5746A0C0233EFF 1152
 67  46A0C0237A2F46A0C0C9 1249
 68  3EFF46A0C0233EFF46A0 1321
 69  C0C94FDD7E02CB3FCB3F 1353
 70  CB3FCDC2EF51DD7E024F 1413
 71  E607F54779905F05033A 980
 72  46F8CBA732466F8F12801 1338
 73  04C57ECB772800D7AE5D5 1266
 74  CD74EEDC42F0D1E11804 1547
 75  FE0620110120000097BC6 672
 76  085FC110DE3A46F8CB67 1216
 77  C9F5CD5FF6F1E638FE08 1781
 78  E5D5CC4BE8D1E13A46F8 1763
 79  CBE73246F818D547CB3F 1376
 80  CB3FCB3FDD4E03CDC2EF 1472
 81  503A46F8CBA73246F8DD 1415
 82  7E03E607060228010 4C5 616
 83  7ECB77280D7AE5D5CDB2 1448
 84  EEC442F0D1E11804FE06 1462
 85  200A23C110E53A46F8CB 1094
 86  67C9F5CD5FF6F1E638FE 1876
 87  08E5D5CC4BE8D1E13A46 1523
 88  F8CBE73246F818DC2600 1332
 89  6F2929292929211005819 446
 90  79CB3FCB3FCB3F16005F 1036
 91  19C93A4AF83D324AF8C0 1231
 92  3E05324AF82120B93A47 818
 93  F8FE0520023E01113000 669
 94  F53D28031918FAF13C32 999
 95  47F8DD7504DD7405224D 1114
 96  F8FE05C93A49F83D3249 1271
 97  F8C03E053249F82160B8 1191
 98  3A48F8FE0520023E0111 751
 99  3000F53D28031918FAF1 937
100  3C3248F8DD7504DD7405 1114
101  224BF8FE05C93A12F83D 1202
102  3212F8C03E143212F83A 964
103  10F83D3210F857FE4820 1084
104  083A46F8CBF73246F87A 1324
105  CB3FCB3FCB3F4F7AE607 1236
106  200300D1807473E089087 499
107  8787C6863296F03A11F8 1365
108  26006F2911150F8190608 574
109  5623SE23E5260006919CB 850
110  86E110F23E07D3FE3E04 1217
111  D3FE3E00D3FE01FF10CD 1469
112  60F1C92134FA7E180421 1060
113  35FA7E473A45F8E60220 1139
114  0B2190B8DD7504DD7405 1056
115  18092150B9DD7504DD74 1010
116  05C5CDFEEDCD6FF1C110 1664
117  F6C3CCEC2136FA7E4723 1450
118  7E18072138FA7E47237E 854
119  2150B9DD7504DD740557 1069
120  3A46F8CBDFCBD7CB8F32 1616
121  46F87AC5F5CDFEEDF1F5 2064
122  F5CDA9EDCD6FF1F13D20 1747
123  F5F1C110EAC3CCEC2136 1651
124  FA7E47237E18072138FA 978
125  7E47237E2190B8DD7504 1061
126  DD7405573A46F8CBDFCB 1434
127  D7CBCF3246F87AC5F5CD 1762
128  FEEDF1F5F5CDS7EDCD6F 2067
129  F1F13D20F5F1C110EAC3 1699
130  CCEC3E10D3FEC510FEC1 1643
131  AFD3FE0D20F2C9000000 1128
```

DUMP: 40.000
N.° BYTES: 1.307

PR_OGER

```
  1  3E02CD0116ED7390FF21 1076
  2  AC7E11D2F9015000EDB0 1268
  3  21AC7E010800EDB0DD21 1007
  4  13F8AFF50E00CDB022DD 1337
  5  7400DD7501DD23DD23F1 1208
  6  C608FEC038E9DD2150F8 1523
  7  AFF50E00CDB022DD7400 1186
  8  DD7501DD23DD23F13CFE 1406
  9  C020EA2188FF5E2356EB 1332
 10  23232277DFE62283E6AFD3 1240
 11  FE32485C3292FFCDC6DE 1544
 12  2191E6CDE8DE3E8532ED 1549
 13  5921C15A11C25A360501 766
 14  1F00EDB02313011F0036 584
 15  02EDB0AF32FF5A32DF5A 1348
 16  2179E60604C57E320EF8 1029
 17  237E320FF8237E320CF8 945
 18  237E320DF8235E235623 757
 19  AF3244F8E5CD7DF4E1C1 1762
 20  10DBDD21F0EDDD5600DD 1479
 21  5E01DD23DD23DD6600DD 1151
 22  6E01DD23DD23DD7E00FE 1224
 23  FF2812DDE5CDB503DDE1 1598
 24  AFDBFEE61FFE1F200218 1252
 25  D30608C5FB7606102180 1022
 26  F9C5S6235E23EB011F00 963
 27  09061FCB16F52BF110F9 1065
 28  EBC110E9C110DE2A92E7 1527
 29  7E2F23FE002011214474 734
 30  2292E73A92FFE610C2D2 1520
 31  DDC3C0DC2292E726006F 1388
 32  292929112C7D1911B0F9 776
 33  EB0608C50602C546234E 834
 34  23E5211F00091A77E1C1 900
 35  10F013C110E93A92FFE6 1406
 36  1020903EF7DBFEE60428 1248
 37  383EF7DBFEE6012123A 1177
 38  92FFCB9F3292FF3E8532 1459
 39  ED593E04322D5A3EF7DB 1105
 40  FEE602C2F1DC3A92FFCB 1803
 41  DF3292FF3E85322D5A3E 1116
 42  0432ED59C3F1DC21BEE6 1489
 43  2292E73A92FFCBE73292 1500
 44  FFCDC6DE21C15A11C25A 1497
 45  3605011F00EDB0231301 559
 46  1F003602EDB0AF32FF5A 1070
 47  32DF5AC3F1DC21DFE6CD 1710
```

```
 48  E8DE21609F0627C5111B 1028
 49  00197EFEFF280CCBB777 1217
 50  3EFF010000EDB1180123 792
 51  C110E63E803245F8AF32 1221
 52  46F8328FFF3EFA3210F8 1392
 53  218AFF114CFA0605AF77 1074
 54  12231310F9212CFA0604 674
 55  36002336092310F82154 566
 56  FA3E5077233E88773E03 928
 57  320EF887320FF83E1332 891
 58  0CF83E0D320DF811A483 958
 59  AF3244F8CD7DF43E1332 1246
 60  0CF83E08320DF83E0232 755
 61  0EF83D320FF8DD21685A 1084
 62  0605C5113CFA3E013244 716
 63  F8CD7DF4DD360000DD36 1372
 64  2000DD23C110E721A1FB 1173
 65  36002336003E03CDA1F3 817
 66  C354ECCDC6DE2122E7CD 1643
 67  E8DE1809CDC6DE2156E7 1462
 68  CDE8DE3A8FFF4F0600CD 1405
 69  2B2DCDE32D217DE7CDE8 1391
 70  DE0664FB7610FC3E00DB 1246
 71  FEE61FFE1F28F6C362DC 1599
 72  ED730CF831005B210000 785
 73  06D8E5E5E5E5E5E5E5E5 2054
 74  E5E5E5E5E5E5E510EFED 2095
 75  7B0CF8C97EFEFFC8D723 1569
 76  18F8001409 1EE00300 3C4 578
 77  00E002F7005102310 03C 501
 78  02F7003003C4005102 31 628
 79  003003C4003C02F70051 637
 80  0231003C02F7003003C4 607
 81  0014091E003003C4003C 366
 82  02F7006001D3003C02F7 866
 83  003003C4006001D30030 603
 84  03C4003C02F7006001D3 816
 85  003C02F7003003C40014 576
 86  091E003603570040 02CA 451
 87  00510231004002CA0036 454
 88  0357005102310036 0357 366
 89  004002CA005102310040 464
 90  02CA003603570014091E 407
 91  0036035700 4002CA004C 488
 92  0254004002CA00360357 498
 93  004C0254003603570040 370
 94  02CA004C0254004002CA 634
 95  0036035700 14091E0030 251
 96  03C4003C02F700510231 640
 97  003C02F7003003C40051 637
 98  0231003003C4003C02F7 607
 99  0051023100 3C02F70030 489
100  03C40014091E00280480 430
101  003003C4003C02F70030 604
102  03C4002804800 03C02F7 680
103  00280480003003C4003C 479
104  02F7003003C4002804 80 668
105  000A125B002D03FF0036 476
106  0357003C02F7003603 57 543
107  002D03FF003C02F7002D 657
108  03FF003603570 03C02F7 711
109  0036035700 2D03FF0014 457
110  091E002804800 03003C4 458
111  003C02F7003003C40028 596
112  0480003C02F7002804 80 613
113  003003C4003C02F70030 604
114  03C4002804800 00F0C36 452
115  002604C6002D03FF003C 603
116  02F7002D03FF002604C6 792
117  003C02F7002604C6002D 581
118  03FF003C02F7002D03FF 870
119  002604C6000F0C360028 361
120  0480003003C4004002CA 647
121  003003C4002804800040 483
122  02CA002804800 03003C4 623
123  004002CA003003C40028 555
124  0480000F0C36002604C6 453
125  002D03FF003C02F7002D 657
126  03FF002604C6003C02F7 807
127  002604C6002D03FF003C 603
128  02F7002D03FF002604C6 792
129  000F0C3600220056000 28 256
130  0480003903250028 0480 401
131  0022056000 1C066A0022 309
132  0560002804800039032 5 370
133  002804800022056000 0F 322
134  0C36002604C6002D03FF 609
135  003C02F7002D03FF0026 650
136  04C6003C02F7002604C6 751
137  002D03FF003C02F7002D 657
138  03FF002604C6000F0C36 579
139  0028048000 3003C40040 483
140  02CA003003C4002804 80 623
141  004002CA002804800030 488
142  03C4004002CA003003C4 714
143  0028048000 0F0C360026 291
144  04C6002D03FF003C02F7 814
145  002D03FF002604C6003C 603
146  02F7002604C6002D03FF 792
147  003C02F7002D03FF0026 650
148  04C6000F0C3600220560 418
149  0028048000 3903250028 309
150  0480002205600 01C066A 407
151  0022056000 2804800039 364
152  0325005280480002205560 347
153  000F0C3600 2604C6002D 366
154  03FF003C02F7002D03FF 870
155  002604C6003C02F70026 587
156  04C6002D03FF003C02F7 814
157  002D03FF002604C6000F 558
158  0C3600280480003003C4 485
159  004002CA003003C40028 555
160  0480004002CA00280480 572
161  003003C4004002CA0030 563
162  03C4002804800 00F0C36 452
163  002B043D0033038C0044 370
164  02A0003303 8C002B043D 464
165  00 4402A0002B043D0033 389
166  038C004402A00033038C 567
167  002B043D000F0C36002D 234
168  03FF003603570 04C0254 564
169  0036035700 2D03FF004C 523
170  0254002D03FF00360357 533
171  004C025400360357002D 351
172  03FF000A125B00280480 549
173  003003C4005102310030 427
174  03C4002804800 0510231 503
175  0028048000 3003C40051 500
176  0231003003C4002804 80 470
177  0014091E003003C4003C 366
178  02F7006001D3003C02F7 866
179  003003C4006001D30030 603
180  03C4003C02F7006001D3 816
```


```
181  003C02F7003003C40014 576
182  091E003603570040 02CA 451
183  0051023100 4002CA0036 454
184  0357005102310036 0357 366
185  004002CA005102310040 464
186  02CA003603570014091E 407
187  0036035700 4002CA004C 488
188  0254004002CA00360357 498
189  004C0254003603570040 370
190  02CA004C0254004002CA 634
191  0036035700 14091E0030 251
192  03C4003C02F700510231 640
193  003C02F7003003C40051 637
194  0231003003C4003C02F7 607
195  0051023100 3C02F70030 489
196  03C4000D0DB800280480 581
197  002D03FF004002CA002D 616
198  03FF00280480004002CA 698
199  0028048000 2D03FF0040 539
200  02CA002D03FF00280480 679
201  000F0C3600280480003 0 301
202  03C4003C02F7003003C4 755
203  0028048000 3C02F70028 521
204  0480003003C4003C02F7 688
205  003003C400280480000F 434
206  0C3600280480002D03FF 541
207  003C02F7002D03FF0028 652
208  0480003C02F7002604C6 681
209  002D03FF003C02F7002D 657
210  03FF002604C6000A125B 617
211  0028048000 3003C40051 500
212  0231003003C4002804 80 470
213  0051023100 2804800030 352
214  03C4005102310003003C4 578
215  0028048000 14091E0030 279
216  03C4003C02F7006001D3 816
217  003C02F7003003C40060 652
218  01D3003003C4003C02F7 768
219  006001D3003C02F70030 665
220  03C40014091E00360357 402
221  004002CA005102310040 464
222  02CA003603570051023 1 480
223  0036035700 4002CA0051 493
224  0231004002CA00360357 463
225  0014091E003603570040 267
226  02CA004C0254004002CA 634
227  0036035700 4C02540036 360
228  0357004002CA004C0254 520
229  004002CA003603570014 432
230  091E003003C4003C02F7 595
231  0051023100 3C02F70030 489
232  03C4005102310003003C4 578
233  003C02F7005102310 03C 501
234  02F7003003C4000D0DB8 706
235  0028048000 2D03FF0040 539
236  02CA002D03FF00280480 679
237  004002CA002804800 02D 485
238  03FF004002CA002D03FF 829
239  0028048000 0F0C360028 293
240  0480003003C4003C02F7 688
241  003003C4002804800 03C 479
242  02F7002804800 03003C4 668
243  003C02F7003003C40028 596
244  0480000F0C3600280480 385
245  002D03FF003C02F7002D 657
246  03FF002804800 03C02F7 739
247  002604C6002D03FF003C 603
248  02F7002D03FF002604C6 792
249  000A125B002804800030 339
250  03C4003C02F7003003C4 755
251  0028048000 5102310028 344
252  0480003003C4003C02F7 688
253  003003C4002804800 00A 429
254  1255002804800 03003C4 528
255  003C02F7003003C40028 596
256  0480003C02F7002804 80 613
257  003003C4003C02F70030 604
258  03C4002804800 00A125B 490
259  001E060D002804800030 269
260  03C4002804800 01E060D 420
261  003003C4001E060D0028 336
262  0480003003C4002804 80 551
263  003003C4000A125B0018 390
264  07A6001E060D00280480 394
265  001E060D001807A60028 286
266  0480001807A6001E060D 378
267  0028048000 1E060D0018 245
268  07A6005C125B005C125B 575
269  FF0E08060188FF0E0806 703
270  1788FF020C020ACC8204 778
271  08080CAC811100100416 388
272  060E504F52160F0C315F 454
273  5445434C41444F16110C 559
274  325F4B454D53544F4E16 712
275  130C335F4A55474152FF 809
276  DF90989A8D9087DFDF90 1683
277  989A8D9087DFDF90989A 1622
278  8D9087DFDF90989A8D90 1601
279  87FFFF130110041161101 725
280  454E4552474941 3A1002 583
281  8F8F8F8F10048F8F8F8F 1164
282  8F8F8F10058F8F8F8F8F 1165
283  8F8F8F8F8F8F10041614 920
284  014C4C415645533A1614 556
285  134F424A4S544F533AFF 866
286  1004110013011 60A0353 175
```

```
287  45205445204841204143 587
288  414241444F20454C2054 636
289  49454D504F160D044841 554
290  532048454348 4F20454C 651
291  20FF1004110013011 60A 376
292  0954452048414E204D41 583
293  5441444F160D04484153 555
294  2048454348 4F20454C20 600
295  FF2520444520455354 41 794
296  160F0B4156454E545552 597
297  41FF447A0000000000000 510
298  0000000000000000000 0
```

DUMP: 50.000
N.º BYTES: 2.980

OG_MENS

```
1  DF90989A8D9087DFDF90 1683
2  989A8D9087DFDF90989A 1622
3  8D9087DFDF90989A8D90 1601
4  87FF0000000000000000 390
```

DUMP: 40.000
N.º BYTES: 32

DEMO5

```
  10 CLS : INPUT "Quieres salvar
 el juego completo (S/N)?";o$: I
F o$="s" OR o$="S" THEN PRINT "I
ntroduce la cinta donde tengas l
a pantalla final": LOAD ""CODE 3
3888: PRINT "introduce la cinta
donde hayas salvado el cargador
": SAVE "ogerox"CODE 31300,34235
: INPUT "Quieres jugar ahora (S/
N)? ";o$: IF o$="s" OR o$="S" TH
EN POKE 23606,44: POKE 23607,125
: RANDOMIZE USR 56320
  20 POKE 56403,201: RANDOMIZE U
SR 56320
  30 FOR n=1 TO 12: READ a,b: PO
KE a,b: NEXT n
  40 DATA 56601,0,56604,255,5661
3,195,56614,241,56615,220,56674,
127,56678,1,56679,200,56672,0,56
680,195,56681,241,56682,220
  50 REM PONED EL MENSAJE EN MAY
USCULAS
  60 LET a$="© A.ELICES ... PROG
RAMA DEMOSTRACION DE LA RUTINA D
E SCROLL DE UN MENSAJE POR LA PA
NTALLA
     "+CHR$ 255
  70 FOR n=1 TO LEN a$: POKE 312
99+n,CODE a$(n): NEXT n
  80 PRINT AT 0,5;"pulsa BREAK p
ara parar"
  90 RANDOMIZE USR 56561
 100 REM PRUEBA DE APERTURA DE
VENTANAS
 110 LET DIR=31000: REM LUGAR DO
NDE SE VAN A ALMACENAR LOS DATOS
 DE LA VENTANA
 120 LET H=INT (DIR/256): LET L=
DIR-H*256: POKE 64058,L: POKE 64
059,H: POKE 59579,0: POKE 59580,
0: POKE 59581,0: POKE 59675,201
 130 INPUT "FORMATO VERTICAL ";F
V: IF FV>21 OR FV<2 THEN GO TO 1
30
 140 INPUT "FORMATO HORIZONTAL "
;FH: IF FH>31 OR FV<2 THEN GO TO
 140
 150 INPUT "COORDENADA X ";X: IF
 X+FV>21 THEN GO TO 150
 160 INPUT "COORDENADA Y ";Y: IF
 Y+FH>31 THEN GO TO 160
 170 RESTORE 180: FOR N=DIR+3 TO
 DIR+6: READ A: POKE N,A: NEXT N
 180 DATA X,Y,FV,FH
 190 FOR N=0 TO 21: PRINT AT N,0
;"©©©©©©©©©©©©©©©©©©©©©©©©©©©©©©
©©": NEXT N
 200 RANDOMIZE USR 59540
 210 GO TO 130
9999 SAVE "DEMO5" LINE 10
```

CARGADOR 6

```
  10 CLEAR 30000: BORDER 0: INK
7: PAPER 0: OVER 0: INVERSE 0: F
LASH 0: BRIGHT 1: CLS
  20 PRINT AT 10,4;"OGEROX SE ES
TA CARGANDO.";AT 12,8; FLASH 1;
INK 2;"ESPERA POR FAVOR"
  30 INK 0: PRINT AT 0,0;: LOAD
"OGEROX"CODE 31300: POKE 23606,4
4: POKE 23607,125: RANDOMIZE USR
 56320
9999 SAVE "cargador6" LINE 10
```