

2ª EDICION  
REVISADA Y AMPLIADA

COMPATIBLE CON UNIDAD  
DE DISCO Y MSX-2  
INCLUYE APENDICE SOBRE  
GRAFICOS AVANZADOS  
Y EFECTOS DE SONIDO



Andrew Lacey

■ El libro gigante de los

# Juegos

para MSX

MSX

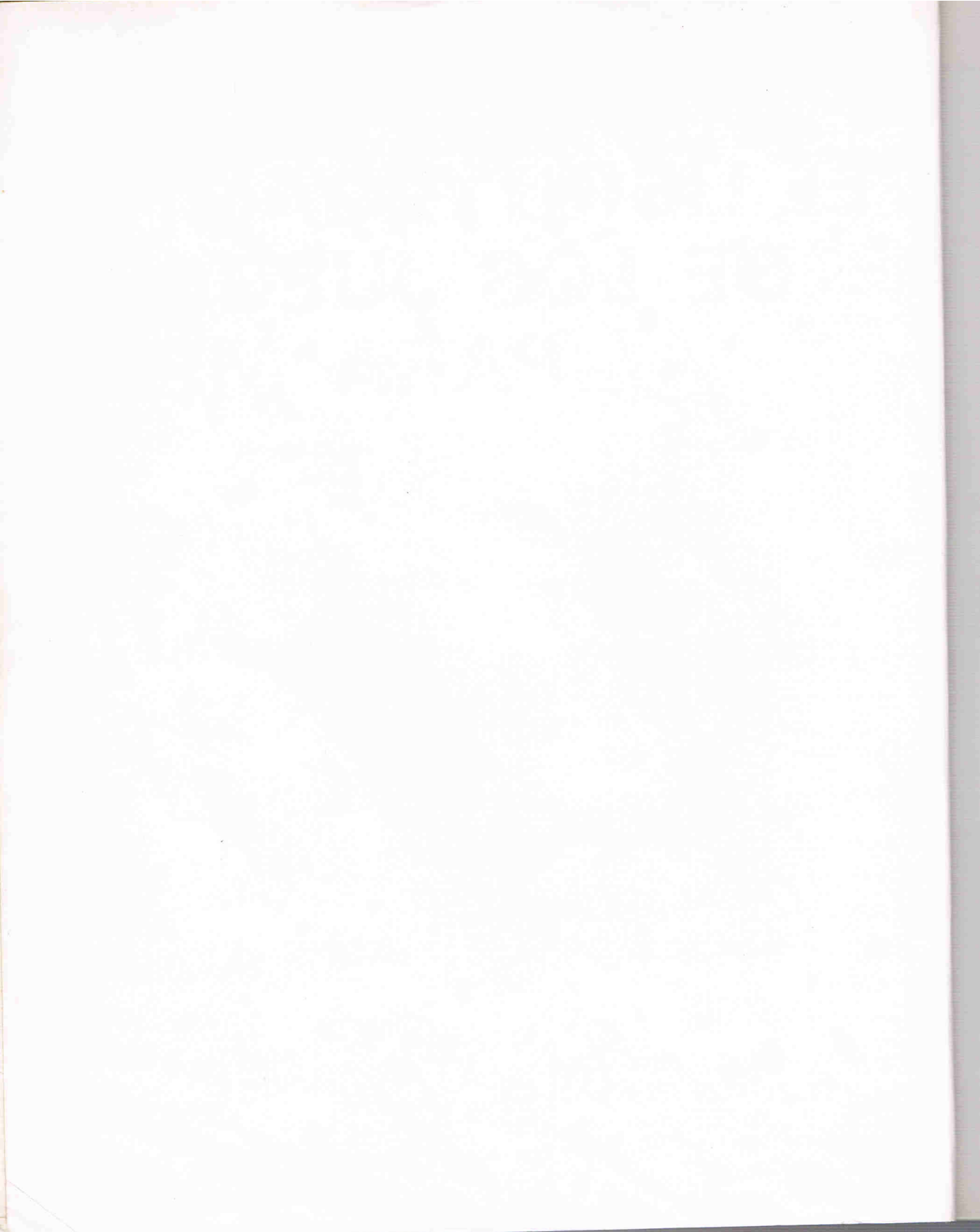


THE UNIVERSITY OF CHICAGO



# **EL LIBRO GIGANTE DE LOS JUEGOS PARA MSX**







# **EL LIBRO GIGANTE DE LOS JUEGOS PARA MSX**

**Andrew Lacey**

**ANAYA**  
MULTIMEDIA



## MICROINFORMATICA

Título de la obra original:  
MSX GAMES BOOK

Traducción de: Celso Losada, Juan Carlos Contreras, Juan José Rodríguez  
Diseño de colección: Antonio Lax  
Diseño de cubierta: Narcís Fernández

Primera edición, 1985  
Segunda edición, 1987

Reservados todos los derechos. Ni la totalidad ni parte de este libro puede reproducirse o transmitirse por ningún procedimiento electrónico o mecánico, incluyendo fotocopia, grabación magnética o cualquier almacenamiento de información y sistema de recuperación, sin permiso escrito de Ediciones Anaya Multimedia, S. A.

© 1984 Andrew Lacey

Edición publicada por acuerdo con  
Melbourne House (Publishers) Ltd.,  
Londres

© EDICIONES ANAYA MULTIMEDIA, S. A., 1987  
Villafranca, 22. 28028 Madrid  
Depósito legal: M. 23.647-1987  
ISBN: 84-7614-028-2  
Printed in Spain  
Imprime: Anzos, S. A. - Fuenlabrada (Madrid)



# Indice

**Introducción** ..... 7

## **Juegos educativos**

El gorila matemático ..... 141  
Minotauro Mastermind ..... 277  
Minotauro Mastermind Jr. .... 249

## **Gráficos interactivos**

Diseños ..... 17

## **Juegos con tiempo límite**

Cocodrilo ..... 121  
Convoy ..... 131

## **Juego de carreras**

Dragster ..... 57

## Juegos de disparo

Batalla laser .....	225
Invasión de robots.....	111
O.V.N.I.....	39

## Juegos de evasión

Ataque astral .....	23
Las cucarachas cósmicas.....	65

## Juegos memorísticos

Concentración.....	171
Test de inteligencia .....	73

## Juegos de mesa

En busca del tesoro .....	237
Treinta y una.....	263

## Juegos de simulación

Acróbata aéreo .....	181
Cross .....	201
El saltarín del río .....	81
Emboscada de tanques .....	161
Rescate aéreo .....	151

## Juegos de tiro al blanco

Caseta de tiro .....	213
Globos.....	47
Harrier.....	99
Naves secretas de investigación.....	191
Planetoide.....	31
Prueba de fútbol .....	89

Apéndice D.....	303
-----------------	-----



# Introducción

Los ordenadores con el sistema MSX han aparecido para revolucionar el mercado del ordenador personal. Primero ofrecen un BASIC perfeccionado, con posibilidades gráficas y de sonido potentes, haciéndolo un modelo ideal para la programación educativa y de entretenimiento. Segundo, dan a las casas de *software* la ventaja de crear programas ejecutables en una amplia gama de ordenadores; esto implica que el usuario del sistema MSX tiene una amplia gama de programas de primera línea para su ordenador.

Con este libro, Anaya Multimedia continúa con su tradicional edición de libros de alta calidad, tanto literaria como de programas, en beneficio del ordenador personal.

## Organización de los programas

Los programas de juegos están ordenados en este libro, en orden gradual, de más fáciles a más complejos. Sin embargo, y teniendo en cuenta que hay gente que prefiere determinados tipos de juegos, hay un índice que agrupa los programas según el tipo de juego. Si deseas juegos de estrategia, encontrarás los programas consultando el índice.

## Fácil de usar

Para acelerar la depuración de los programas, el programa verificador facilita la detección y corrección en la transcripción de alguna sentencia.

Pero —y esto es importante— primero lee el capítulo «Fiabilidad en los programas», si quieres obtener un máximo beneficio a la hora de transcribir exactamente los programas.

## A la hora de jugar

Para ayudarte a manejar más rápidamente los juegos, se han prefijado las órdenes de juego. Exceptuando los casos que se especifican en la introducción a cada juego, las teclas de espaciado y movimiento de cursor se emplean del siguiente modo:

TECLA	FUNCION
Movimiento del cursor hacia arriba (↑)	Mueve la mira u objeto arriba
Movimiento del cursor hacia abajo (↓)	Mueve la mira u objeto hacia abajo
Movimiento del cursor a la izquierda (←)	Mueve la mira u objeto a la izquierda
Movimiento del cursor a la derecha (→)	Mueve la mira u objeto a la derecha
Barra espaciadora	Botón de disparo

## Objetivos de esta estructuración

Espero que según vayas leyendo y probando los programas de este libro aprendas las bases de cómo programar tu ordenador con el sistema MSX. Para ello hay una lista con las variables, y cada subrutina va encabezada con un título dentro de los listados de cada programa; esto te hará comprender su funcionamiento.

Te desafío a que hagas las mejoras que se te ocurran en los programas de los juegos.

## Sugerencias

Anaya Multimedia estará siempre interesada en recibir correspondencia de sus lectores, ya sean elogios o quejas.

¡Diviértete con los juegos del ordenador!

## FIABILIDAD EN LOS PROGRAMAS

Los programas invierten mucho tiempo en su transcripción y depuración. Para facilitarte la lectura y depuración de éstos, tus listados han sido especialmente estructurados; incluimos el útil programa verificador para facilitarte la ejecución de los programas de juegos. Continúa la lectura de este capítulo e introduce el programa verificador antes que cualquier otro programa de juego.

# REGLAS DE TRANSCRIPCION

En los listados de los programas los espacios se han empleado sólo para facilitar su lectura; por ejemplo, la línea.:

```
10      PRINT "_CARACTERES_"
```

has de introducirla en tu ordenador como:

```
10      PRINT " CARACTERES "
```

En las cadenas de caracteres se introducen a menudo espacios, pero sólo has de pulsar un espacio cuando veas el símbolo (↔). Este carácter no lo verás en tu teclado; lo incluimos únicamente para que sepas cuándo has de teclear un espacio.

## Caracteres confusos

Existen caracteres muy parecidos que quizá te induzcan a introducirlos incorrectamente y producir un error en tu programa. Estos son: la letra I y el número 1, la letra O y el número 0. En los listados el número cero lo escribimos con una barra atravesándole (∅). Estos caracteres confusos se reconocen fácilmente en los listados:

CARACTER	LISTADO
Letra I	I
Número 1	1
Letra O	O
Número 0	∅

Si optas por modificar programas e introducirlos a partir de tus propias notas, te ocurrirá que hay caracteres muy parecidos escritos a mano. Nosotros te aconsejamos:

CARACTER CONFUSO	ESCRITURA RECOMENDADA
Letra Z y número 2	Z, 2
Letra O y número 0	O, ∅
Letra S y número 5	S, 5
Letra I y números 1 y 7	I, 1, 7

Los símbolos de puntuación son también una trampa:

- Las comas (,) y los puntos (.) no son intercambiables.
- Los dos puntos (:) y el punto y coma (;) tampoco lo son.
- Lo mismo sucede con los apóstrofos (') y las comillas ("
- Usa los paréntesis —()— y los corchetes —[]— correctamente. Asegúrate de que están bien emparejados: una forma rápida de comprobarlo es contar los paréntesis



- o corchetes izquierdos y derechos, y ver que hay la misma cantidad dentro de la expresión matemática.
- La tecla SHIFT has de pulsarla cuando introduzcas algunos caracteres como !, “, %, etcétera.

## Graba (CSAVE o SAVE) los programas antes de ejecutarlos (RUN)

No te olvides del viejo refrán que dice:

«Los programas transcritos y grabados hoy sobreviven hasta mañana.»

Asegúrate de grabar en *diskette* o cinta los programas según los introduzcas. De vez en vez el teclado se bloqueará, especialmente si tratas de ejecutar un programa con un error. Nosotros te recomendamos, y no lo olvides, que grabes siempre una copia del programa primero y luego elimines todos los errores de transcripción.

**Nota:** Algunos programas tienen subrutinas en lenguaje máquina en zonas altas de la memoria y alterarán los punteros del sistema. Si se interrumpe la ejecución de un programa de este tipo, el programa en lenguaje máquina y los punteros quedarán intactos en memoria, incluso si introducimos el comando NEW. Cuando depures los programas, grábalos antes de ejecutarlos e inicializa (RESET) tu ordenador después de un error de ejecución, para evitar continuos errores en la ejecución de los programas transcritos o cargados en memoria.

## EL PROGRAMA VERIFICADOR

El mayor problema que hay a la hora de transcribir programas de un libro son los errores cometidos al teclearlos. La mayoría se detectan cuando ejecutas el programa con la orden RUN y te aparece un mensaje de error (*Syntax error*). Sólo tienes que listar la línea en cuestión y compararla con la del libro. Desafortunadamente, se cometen errores más sutiles que no son detectados en la ejecución del programa. El programa se ejecuta, pero incorrectamente, y el ordenador no es capaz de detectarlo.

El programa verificador es un programa especial, que genera una clave para cada línea y una clave total del programa. Después del listado de cada programa viene una tabla con los valores de las claves de cada sentencia. Sólo necesitas comparar los números de la tabla de claves de cada programa con los de la tabla que has obtenido. Si dos números son diferentes, comprueba esa línea.

Teclea el programa verificador y sálvalo. Para salvarlo en cinta, utiliza la orden SAVE “CAS:VERIFI”; si lo vas a salvar en disco, utiliza SAVE “VERIFI”,A. Resulta necesario utilizar este tipo de instrucciones porque para realizar un MERGE —y habrá que hacerlo varias veces con el verificador— es necesario que los programas estén grabados en formato ASCII.

```

1 GOTO 62000
62000 REM PROGRAMA VERIFICADOR
62010 REM
62020 REM Lea las observaciones
62030 REM     en el libro
62040 REM
62050 REM
62055 INF=2: SUP=62000!
62060 CLS:PRINT TAB(10) "VERIFICADOR"
:PRINT:PRINT:PRINT "Salida por impres
ora"TAB(24)"<I>":PRINT:PRINT:PRINT"Sa
lida por pantalla" TAB(24)"<P>":PRINT
:PRINT:PRINT:PRINT:PRINT
62070 PRINT"Pulsa 'P' o 'I'"
62080 X$=INKEY$:IF X$=""THEN 62080
62090 IF X$>"Z"THEN X$=CHR$(ASC(X$)-3
2)
62100 IF NOT(X$="P" OR X$="I") THEN 6
2080
62110 CLS:PRINTTAB(10);"VERIFICADOR":
PRINT:PRINT:PRINT
62120 INPUT "Linea de comienzo:";ST
62130 IF ST<INF THEN ST=INF
62140 CLS
62150 PRINT "VERIFICANDO"
62160 PRINT:PRINT:PRINT
62170 I=PEEK(64584!)+256*PEEK(64585!)
+1:REM comienzo del basic
62180 REM
62190 REM
62200 REM BUCLE PRINCIPAL
62210 REM
62220 CT=0
62230 IS=PEEK(I)+256*PEEK(I+1)
62240 IF IS=0 THEN GOTO 62410
62250 LI=PEEK(I+2)+256*PEEK(I+3)
62260 IF LI=SUP THEN GOTO 62410
62270 IF LI <ST THEN 62390
62280 CS=0
62290 CD=0

```

```

62300 FOR J= I+4 TO IS-2
62310 CR=PEEK(J)
62320 IF CD=0 AND CR=34 THEN CD=1:GOT
D 62340
62330 IF CD=1 AND CR=34 THEN CD=0
62340 IF NOT (CD=0 AND CR=32) THEN CS
=CS+CR
62350 NEXT J
62360 PRINT LI TAB(7); "="; CS
62370 IF X$="I" THEN LPRINT LI TAB(7)
; "="; CS
62380 CT=CT+CS
62390 I=IS
62400 GOTO 62230
62410 PRINT:PRINT:PRINT"TOTAL" TAB(7)
" =";CT
62420 IF X$="I" THEN LPRINT:LPRINT:LP
RINT"TOTAL" TAB(7) " =";CT

```

El programa verificador, tal como está, no se puede verificar a sí mismo. Está pensado para verificar las líneas mayores que 1 y menores que 62000. Por tanto, si quieres verificar el verificador, deberás cambiar algunas líneas:

```
62055 INF=1: SUP=65000!
```

Antes de ejecutar el programa es importante que leas el epígrafe siguiente: "Limitaciones del verificador".

Para realizar la autoverificación, simplemente ejecuta el programa (IMPORTANTE: antes, sálvalo) y sigue sus instrucciones.

## Tabla de claves

1	= 288	62060	= 8834	62140	= 159
62000	= 1558	62070	= 1246	62150	= 1023
62010	= 143	62080	= 1591	62160	= 551
62020	= 2132	62090	= 2592	62170	= 4666
62030	= 1099	62100	= 2308	62180	= 143
62040	= 143	62110	= 2197	62190	= 143
62050	= 143	62120	= 2107	62200	= 1180
62055	= 1222	62130	= 1612	62210	= 143



62220 = 407	62300 = 1412	62380 = 932
62230 = 2287	62310 = 949	62390 = 468
62240 = 1365	62320 = 2299	62400 = 335
62250 = 2542	62330 = 1909	62410 = 1662
62260 = 1589	62340 = 2642	62420 = 2559
62270 = 1352	62350 = 205	
62280 = 406	62360 = 975	
62290 = 402	62370 = 1848	TOTAL = 65768

Cuando hayas acabado de comprobar el verificador y funcione perfectamente, deshaz los cambios anteriores, déjalo como la primera versión y sálvalo como te indicamos antes.

62055 INF=2: SUP=62000!

## Limitaciones del verificador

Es MUY IMPORTANTE saber que el programa verificador sólo es una ayuda para la corrección de los programas, pero que no resulta infalible en el 100 por 100 de los casos. El funcionamiento del verificador está basado en la suma de todos los códigos que componen una línea de programa, por tanto, el resultado, o clave, que se genera, depende estrechamente del contenido de la línea. Cualquier pequeña modificación, aunque no sea un error, como por ejemplo un espacio de más en un PRINT o algo similar, puede hacer que el resultado del verificador sea distinto. Resumiendo: SOLO se obtendrán claves iguales a las del libro si el texto de los programas es idéntico. Cualquier modificación en los programas, aunque no sean errores, harán que las claves varíen.

Otro dato a tener en cuenta es el siguiente: las claves del verificador varían al ejecutarse el programa verificado. Esto es debido a que los GOTO y los GOSUB se almacenan internamente en el programa de forma distinta, y, por tanto, con claves distintas, si han sido ejecutados o no. Resumiendo: los resultados, o claves, del verificador, SOLO podrán coincidir con las del libro si el programa no se ha ejecutado nunca.

Por otra parte, también es posible, aunque no probable, que las claves sean correctas pero la línea esté mal escrita; debido a que se cometan dos errores que contrarresten sus efectos sobre las claves del verificador.

Hay que decir, por tanto, que el programa puede estar bien y funcionar correctamente aunque las claves del verificador sean distintas. Vuelvo a repetir que el verificador no es más que una herramienta de ayuda y no resulta absolutamente imprescindible.

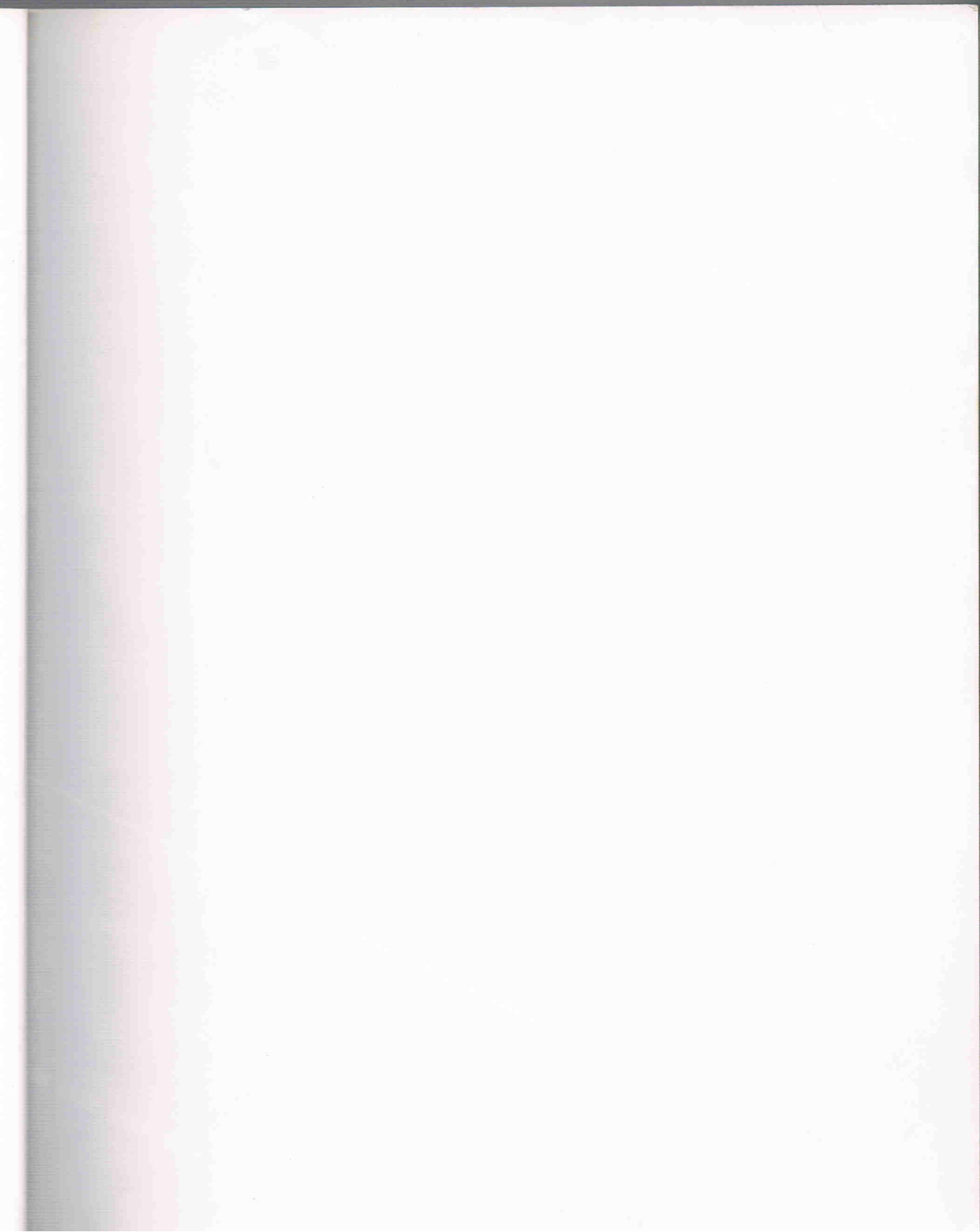
## Instrucciones para el uso del verificador

- Si no has leído el epígrafe anterior, "Limitaciones del verificador", léelo ahora. Es importante.
- Teclea o carga el programa que quieras verificar.
- Comprueba que tu programa no contiene la línea 1 o líneas superiores a la 62000 (no es el caso de los programas de este libro).

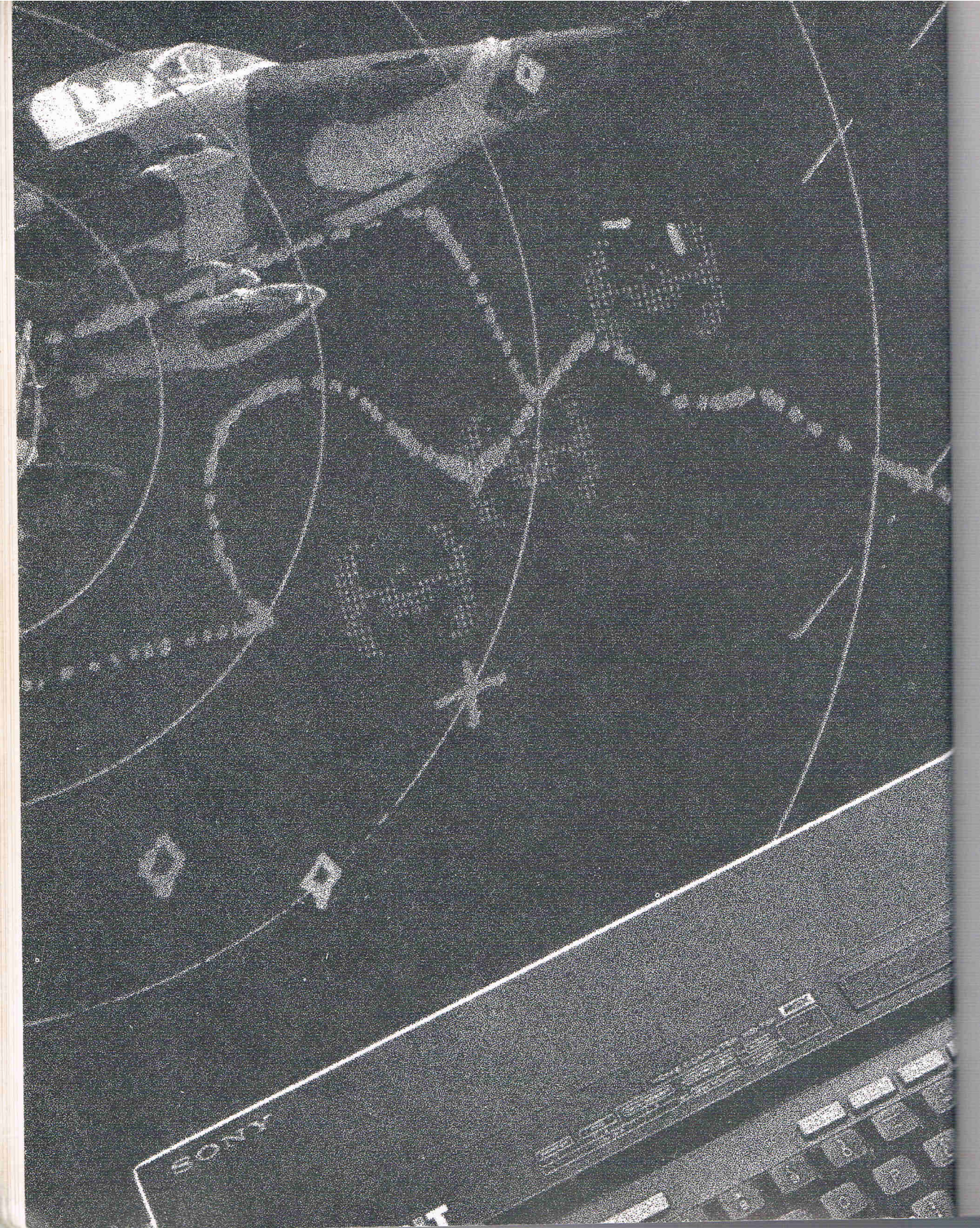
- Carga con MERGE el verificador (MERGE "CAS:VERIFI" para cinta o MERGE "VERIFI" para disco).
- Ejecuta el programa.
- Cuando todo esté correcto, borra la línea 1 y de la 62000 a la 65000.
- Salva el programa, ejecútalo y diviértete.

**NOTA MUY IMPORTANTE:**

Para que algunos programas funcionen, resulta imprescindible ejecutar antes el programa SOPORTE, contenido en el apéndice C, página 301.



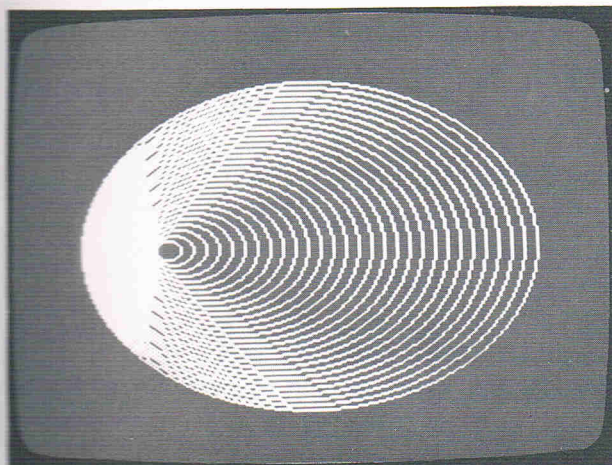






# Diseños

# 1



**Tipo de juego:**

**GRAFICOS INTERACTIVOS**

¡Crea estupendos diseños con círculos y rectángulos! Este programa te demostrará el superpoder gráfico del sistema MSX. Responde a las preguntas y fijate en la forma de las figuras.

Los diseños de círculos se forman moviendo los círculos horizontalmente, y cambiando sus formas y medidas.

Los rectángulos se generan a partir de una línea que une dos puntos, P1 y P2.

Pulsa cualquier tecla cuando hayas acabado tu diseño.

# SUGERENCIAS AL PROGRAMA

Aquí tienes algunas combinaciones para que empieces:

CIRCULOS            50,    3,    -0.05,    0  
RECTANGULOS        60,    -2,    -2,    -2,    -2

Con simples sumas dibujarás líneas en vez de rectángulos o incluso puntos aislados. Puedes hacer cambios: modifica los números de la línea 380 o los de la línea 310.

## Variables del programa

NC            Número de círculos.  
IN            Incremento en la dirección del eje X.  
VR            Variación del radio.  
CR            Cambiar el radio.  
CX, CY        Coordenadas del centro del círculo.  
R            Radio.  
NR            Número de rectángulos.  
IX, IY        Incrementos en las coordenadas X e Y del punto P1.  
SX, SY        Incrementos en las coordenadas X e Y del punto P2 (han de ser múltiplos de IX e IY).

Inicialización

```
100 K=3
110 C1=15
120 SCREEN 1:KEY OFF:PRINT TAB(20);"D
ISEXDS":PRINT TAB(20);";;;;;;;;":PRINT
:PRINT:PRINT
130 PRINT:PRINT TAB(4);"CIRCULOS":PRI
NT:PRINT:PRINT TAB(4);"RECTANGULOS":P
RINT:PRINT:PRINT TAB(4);"ELIGE_COLOR"
140 FOR I=1 TO 5:PRINT:NEXT:PRINT "Ut
iliza_las_teclas_del_cursorpara_mover
_la_flecha":PRINT:PRINT "Pulsa_<RETUR
N>_para_____seleccionar"
150 K1=60:GOSUB 210
160 X%=INKEY$:IF X%=CHR$(30) THEN IF
K=0 THEN 160 ELSE K1=32:GOSUB 210:K1
=60:K=K-1:GOSUB 210:GOTO 160
170 IF X%=CHR$(31) THEN IF K=2 THEN 1
```

```

60 ELSE K1=32:GOSUB 210:K1=60:K=K+1:G
OSUB 210:GOTO 160
180 IF X$<>CHR$(13) THEN 160 ELSE IF
K=0 THEN 220 ELSE IF K=1 THEN 260 EL
E CLS:INPUT "Tinta (1-15)";X$:C1=VAL(
X$):IF C1<1 OR C1>15 THEN RUN
190 INPUT "Papel (1-15)";X$:K2=VAL(X$
):IF((K2<1) OR (K2>15)) THEN 190 ELSE
INPUT "Borde (1-15)";X$
200 K3=VAL(X$):IF K3<1 OR K3 >15 THEN
PRINT "Fuera_de_rango":GOTO 190 ELSE
COLOR C1,K2,K3:GOTO 120
210 VPOKE6356+96*K,K1:VPOKE 6357+96*K
,K1:VPOKE 6358+96*K,K1:RETURN
220 CLS:INPUT "Numero de circulos";NC
230 PRINT:INPUT "Incremento_en_la_dir
eccion___del_eje_X";X$:IN=VAL(X$)
240 PRINT:INPUT "Variacion_del_radio"
;X$:IF VAL(X$)<-1 OR VAL(X$)>1 THEN 2
40 ELSE VR=VAL(X$)
250 PRINT:INPUT "Cambiar_el_radio";CR
:GOTO 310
260 CLS:INPUT "Numero de rectangulos"
;NR
270 PRINT:INPUT "Incremento_en_la_dir
eccion___del_eje_X";IX:PRINT:INPUT "I
ncremento_en_la_direccion___del_eje_Y
";IY
280 PRINT : INPUT "Incremento_o_decr
emento_P2___coordenada-X";SX
290 PRINT : INPUT "incremento_o_decr
emento_P2___coordenada-Y";SY
300 GOTO 380

```

Círculos
----------

```

310 SCREEN 2 :CX = 90 : CY = 90 : IF
CR<0 THEN R=100 ELSE IF CR>0 THEN R=5
ELSE R=50
320 AR = 1
330 FOR I=1 TO NC

```



```

340 IF CX<0 OR CX>255 OR CY<0 OR CY>1
92 OR R<=0 OR AR<=0 THEN 430 ELSE CIR
CLE(CX,CY),R,C1,0,6.28,AR
350 CX=CX+IN : AR=AR+VR : R=R+CR
360 NEXT
370 GOTO 430

```

Rectángulos

```

380 SCREEN 2 : X1=100 :Y1=100 :X2=150
: Y2=90
390 FOR I=1 TO NR : X1=X1+IX : Y1=Y1+
IY :X2=X2+SX*IX : Y2=Y2+SY*IY
400 LINE (X1,Y1)-(X2,Y2),C1,B
410 NEXT
420 GOTO 430

```

Espera

```

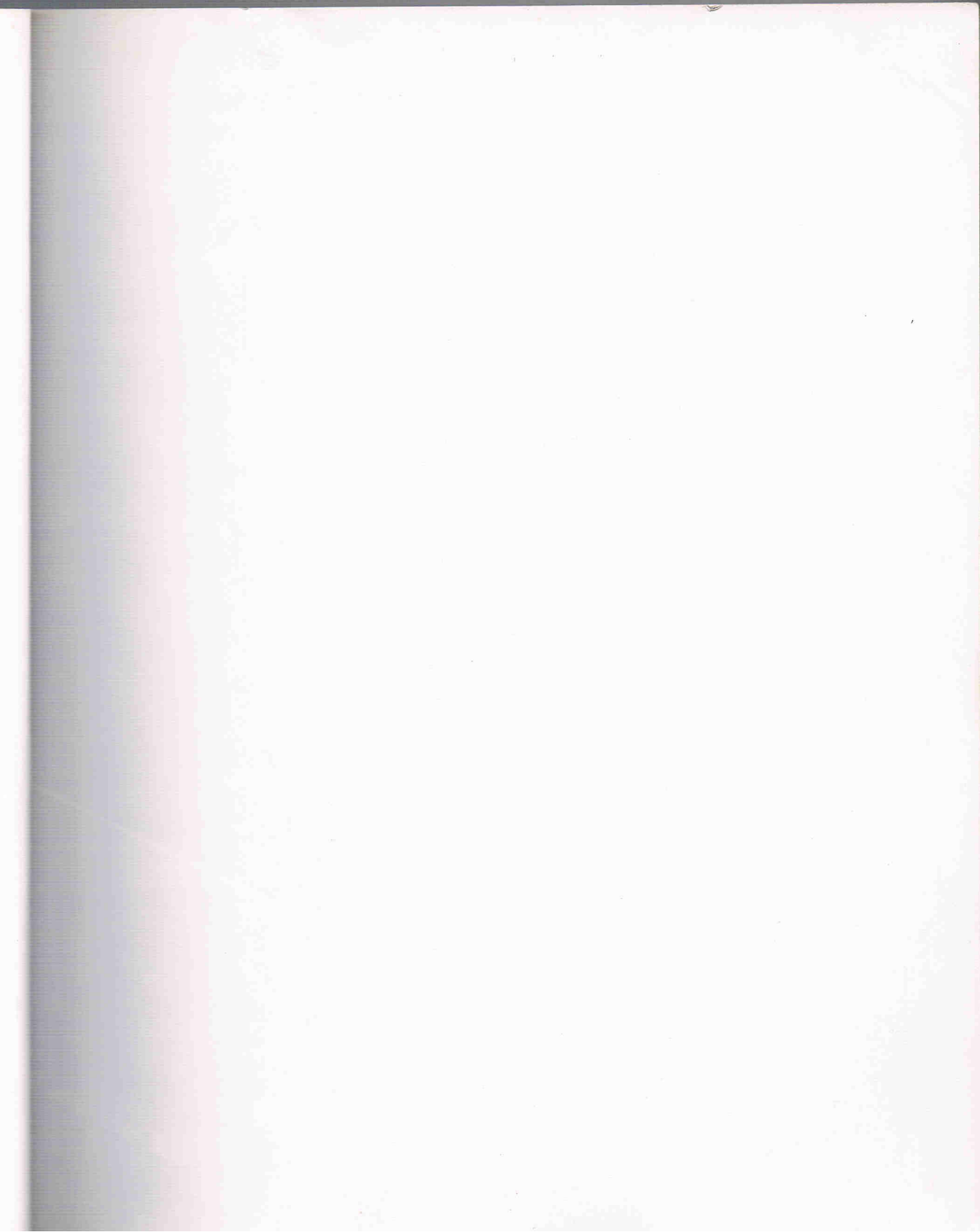
430 IF INKEY$="" THEN 430
440 GOTO 120

```

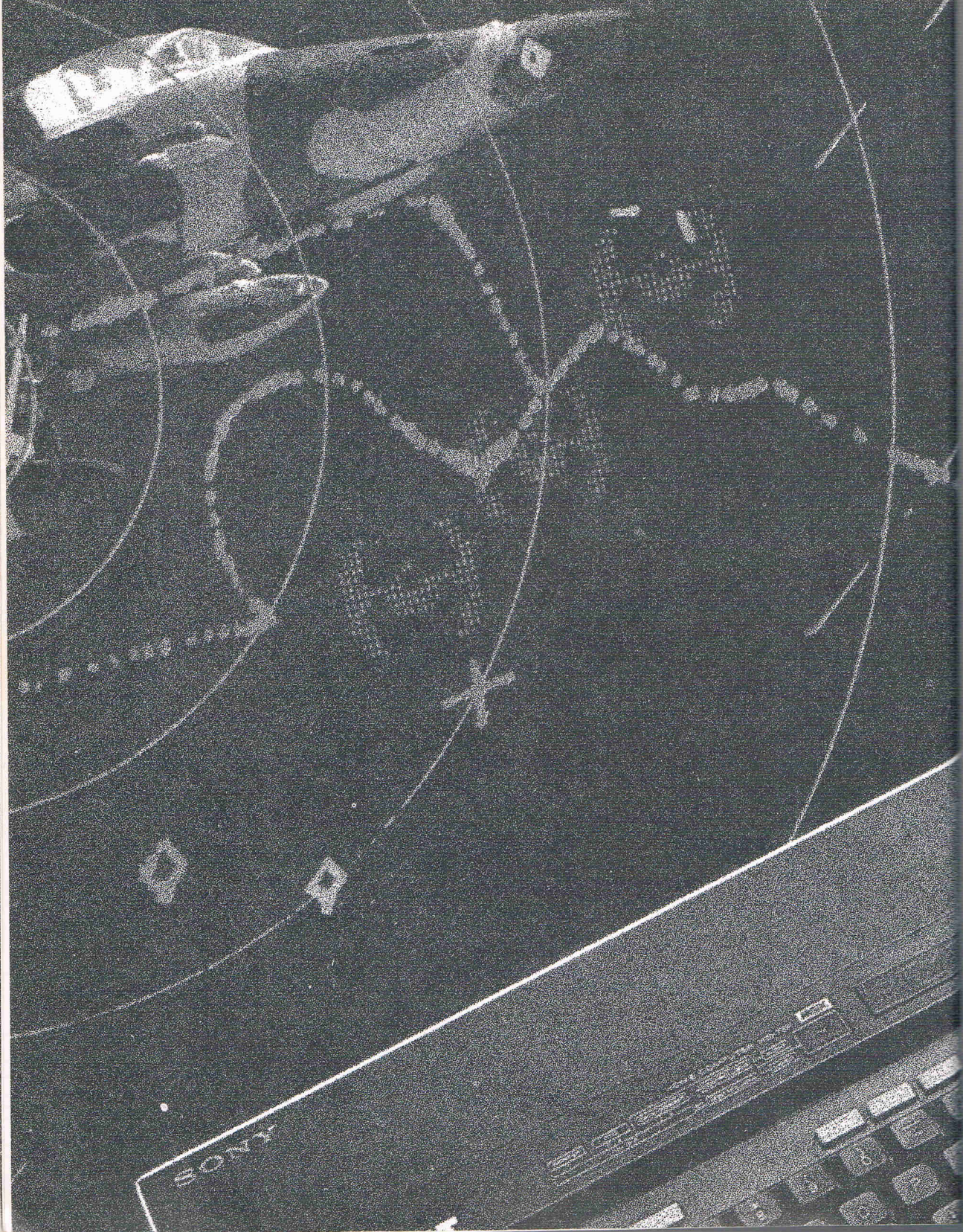
## Tabla de claves

100	=	334	220	=	2385	340	=	6836
110	=	385	230	=	5008	350	=	2792
120	=	4500	240	=	6533	360	=	131
130	=	5014	250	=	2363	370	=	326
140	=	9605	260	=	2723	380	=	2456
150	=	861	270	=	8038	390	=	5568
160	=	5785	280	=	4346	400	=	1489
170	=	5130	290	=	4380	410	=	131
180	=	7935	300	=	276	420	=	326
190	=	5510	310	=	4516	430	=	1089
200	=	5516	320	=	404	440	=	271
210	=	4219	330	=	822	TOTAL	=	118003









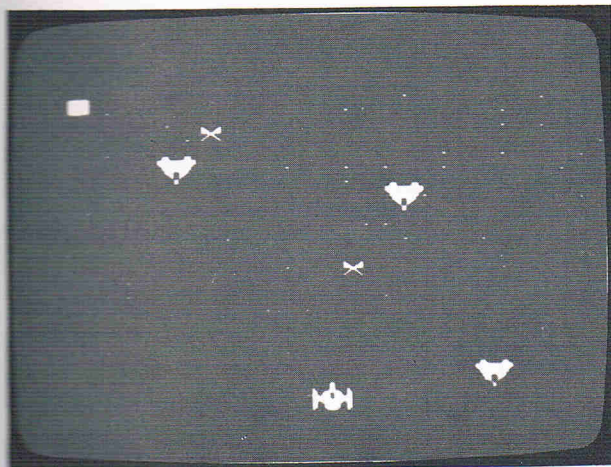
SONY

SONY  
MP3  
1 2 3 4 5 6 7 8 9 0  
+ - \* /



# Ataque astral

# 2



## ¡ATENCIÓN!

Este programa utiliza la subrutina SOPORTE en lenguaje máquina. Véase apéndice C.

**Tipo de juego:**

**JUEGO DE EVASION**

Has conseguido llevar a tu enemigo hasta un vórtice, donde no funciona ningún tipo de arma. Choca contra las naves pequeñas, pero ten cuidado con las grandes.

La batalla termina cuando se te escapa una de las naves pequeñas.

Utiliza las teclas de movimiento de cursor para moverte a la izquierda y a la derecha.





```

140     DEFUSR=A+10: DEFUSR1=A+128!:P
OKE V4VVV,10
150     FOR I = 1 TO 5: VPOKE 6914+4*
I,8:NEXT
160     NE=1:PRINT:PRINT:PRINT:INPUT
"NIVEL_DE_DIFICULTAD_(1-4)";DI$:DN=2*
VAL(DI$)+2:IF DN>10 THEN 160
170     FOR I = 1 TO 8: PRINT : NEXT
: PRINT "PULSA_UNA_TECLA_PARA_EMPEZAR
"
180     D=RND(1): IF INKEY$="" THEN 1
80
190     SPRITE DN:DN SPRITE GOSUB 350
200     FOR I = 1089 TO 1095: VPOKE I
,0: NEXT: VPOKE 1088,1
210     CLS: FOR I = 1 TO 60 : VPOKE
6144+INT(RND(1)*700),136:NEXT
220     TIME=0:PUT SPRITE 0,(120,160)
,11

```

Control

```

230     POKE V1VVV,7:D=USR(D)
240     GOSUB 270
250     GOSUB 320
260     GOTO 230

```

Nave nueva

```

270     IF TIME<105-25*DI THEN RETURN
280     K=US+1:IF K= 6 THEN K=1
290     IF SA AND 2^K THEN RETURN ELSE
US=K:TIME=0:IF RND(1)<.6 THEN K1=8: K
2=7 ELSE K1=4:K2=8
300     VPOKE 6914+4*US,K1:PUT SPRITE
US,(INT(RND(1)*(150-20*DI))+40+20*DI),

```

```
0),K2:SA=(SA) OR 2^US  
310 RETURN
```

Movimiento de naves

```
320 POKE V1VVV, DN: POKE V2VVV, 2: FO  
R I=1 TO 5: IF SA AND 2^I THEN POKE V3  
VVV, I: D=USR1(D): IF VPEEK(6912+4*I)>19  
0 AND(SA AND 2^I) THEN SA=SA AND(255-  
2^I): PUT SPRITE I, (10+30*I, 200): IF VP  
EEK(6914+4*I)=8 THEN FJ=1: GOTO 390 EL  
SE NE=NE+1: IF NE=6 THEN NE=1  
330 NEXT  
340 RETURN
```

Choque

```
350 SPRITE OFF: IF VPEEK (6914+4*N  
E)=4 THEN 390  
360 PUT SPRITE NE, (10+30*NE, 200): S  
A=SA AND (255-2^NE): ND=ND+1: PLAY "15  
4m1700s10n54n32"  
370 NE=NE+1: IF NE=6 THEN NE=1  
380 SPRITE ON: RETURN
```

Fin de juego

```
390 SCREEN 1: PRINT "_____FIN_  
DE_JUEGO": PRINT : PRINT: PRINT "_____P  
ARA_OTRA_PARTIDA": PRINT "_____INSE  
RTA_MONEDA"  
400 FOR I=1 TO 8: PRINT: NEXT : PRIN  
T "NAVES_DERRIBADAS"; ND  
410 PRINT: PRINT: PRINT: PRINT: IF
```

```

FJ=1 THEN PRINT "NAVE_ENEMIGA_ESCAPAD
A!!!" ELSE PRINT "DESTRUIDO_POR_UNA_N
AVE_DE": PRINT "TRANSPORTE!"
420 END

```

Datos de sprites

```

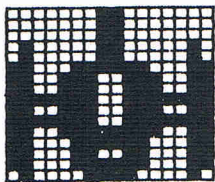
430 DATA 1,1,1,129,129,199,206,206,2
54,206,255,207,199,198,199,131,128,12
8,128,130,130,227,115,115,127,115,255
,243,227,99,227,194
440 DATA 56,56,255,255,255,127,63,63
,31,30,14,6,2,1,1,1,28,28,255,255,255
,254,252,252,248,120,112,96,64,128,12
8,128
450 DATA 129,195,231,255,24,36,66,12
9

```

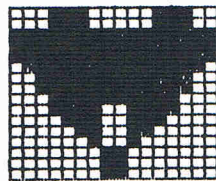
## Tabla de claves

10	= 904	170	= 3295	330	= 131
20	= 143	180	= 1949	340	= 142
30	= 1637	190	= 1004	350	= 2455
40	= 1541	200	= 1852	360	= 5780
50	= 1712	210	= 3322	370	= 2020
60	= 143	220	= 1461	380	= 548
70	= 143	230	= 1348	390	= 5219
80	= 2571	240	= 170	400	= 2704
90	= 2998	250	= 220	410	= 6553
100	= 2607	260	= 381	420	= 129
110	= 3659	270	= 1728	430	= 6089
120	= 3660	280	= 1825	440	= 5369
130	= 3205	290	= 5668	450	= 1532
140	= 2852	300	= 6473		
150	= 1848	310	= 142		
160	= 5703	320	= 15776	TOTAL	= 120611

# Diseño de los sprites



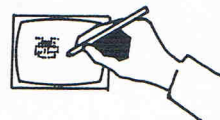
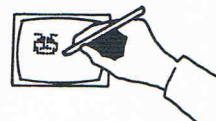
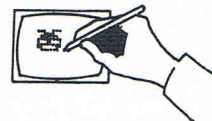
NAVE PROPIA



TRANSPORTE



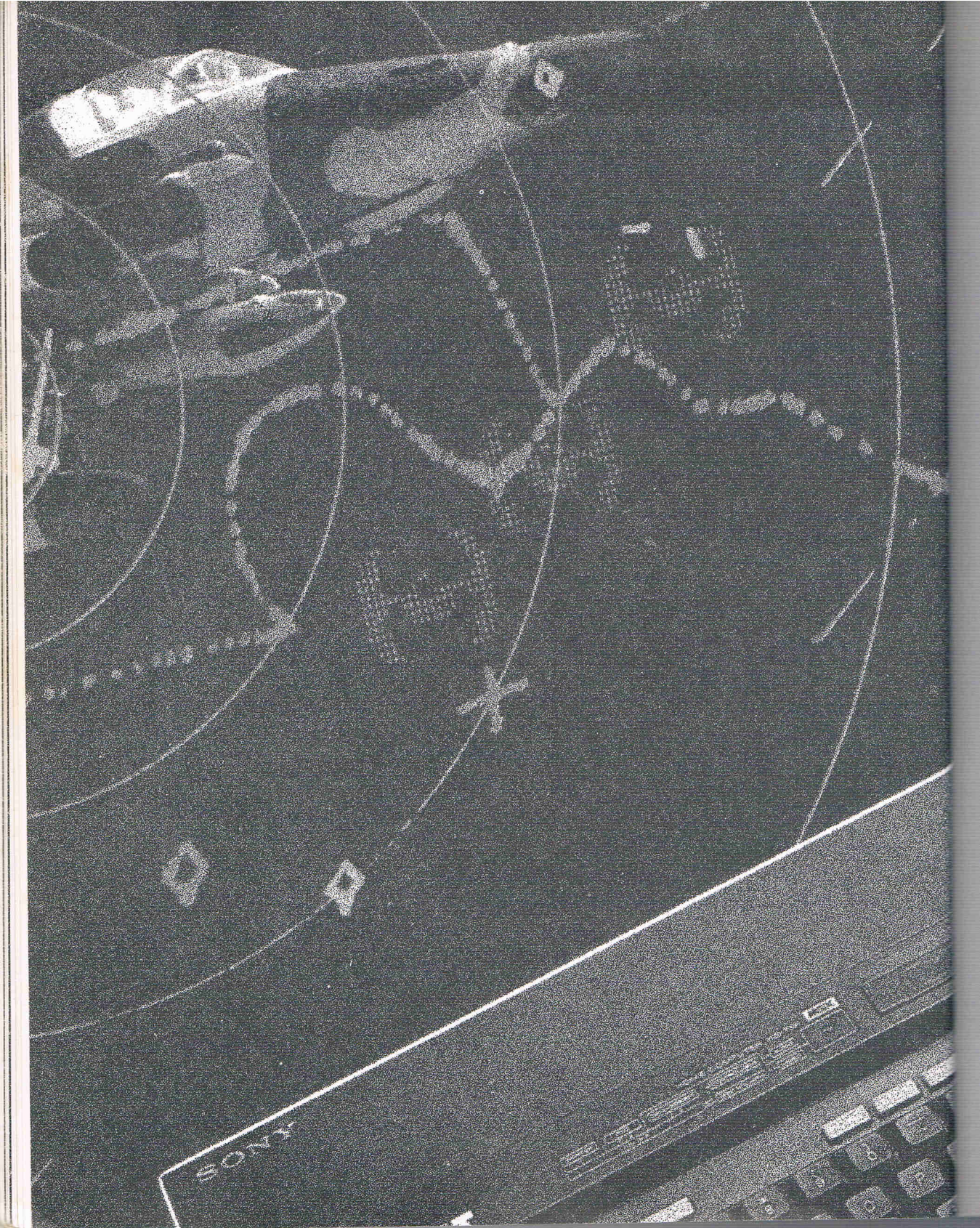
CAZA







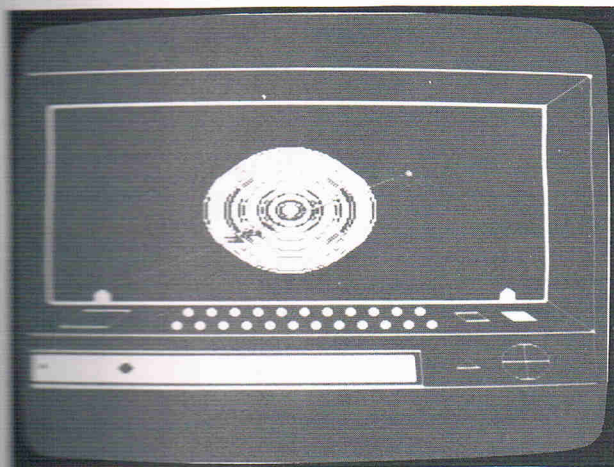






# Planetoide

# 3



## ¡ATENCIÓN!

Este programa utiliza la subrutina  
SOPORTE en lenguaje máquina.  
Véase apéndice C.

**Tipo de juego:**

**JUEGO DE TIRO AL BLANCO**

Atrévete a atravesar un corredor espacial lleno de planetas destructores. Tus lasers deben incidir en un pequeño punto escondido en el planeta, a no ser que quieras convertirte en polvo cósmico.

Observa el ambiente que crean los espectaculares gráficos en esta rara zona del espacio.



# SUGERENCIAS AL PROGRAMA

La proximidad del planeta está controlada por las líneas del programa 430 a 470; sería muy interesante que hicieras alguna modificación.

Haz que R se incremente más rápidamente, o cambia el valor de VR para obtener un aspecto diferente del planeta.

También puedes cambiar esta zona del programa para complementar diversos niveles de dificultad.

## Variables del programa

PL	¿Está el planeta en pantalla?
R	Radio del planeta.
PX, PY	Coordenadas del punto a destruir.
CX, CY	Centro del planeta.
VR	Variación del radio del planeta.
NP	Número de planetas destruidos.

Inicialización

```
10      REM    ** ATENCION **
20      REM
30      REM ESTE PROGRAMA NECESITA
40      REM LA SUBROUTINA SOPORTE
50      REM Vease el Apendice C
60      REM
70      REM
80      A=PEEK(64586!)+256*PEEK(6458
7!)
90      V1=A+9:V2=A+8:V3=A+7:V4=A+6
100     CLS:PRINT "_____PULSA_
UNA_TECLA"
110     IF INKEY$="" THEN D=RND(1):
GOTO 110
120     COLOR 15,1,1:SCREEN 2,2
130     FOR I=1 TO 32: READ Q: A$=A$
+CHR$(Q): NEXT: SPRITE$(0) = A$
140     DEFUSR=A+10:POKE V4VVV,15:PO
KE V1VVV,8
```

```
150 A$="":FOR I=1 TO 8:READ Q: A$
=A$+CHR$(Q): NEXT: SPRITE$(1)=A$
```

Preparación de pantalla

```
160 LINE (20,140)-(235,20),15,B:LINE(0,160)-(255,0),15,B: LINE (0,192)
-(255,160),15,B: LINE (10,190)-(180,170),15,B
170 LINE (13,187)-(177,173),15,B:
PAINT (50,186)
180 LINE(0,160)-(20,140):LINE(255,160)-(235,140):LINE(0,0)-(20,20):LINE
E (255,0)-(235,20)
190 FOR I=80 TO 180 STEP 10: CIRCLE(I,146),2: PAINT (I,146): NEXT: FOR
I= 75 TO 185 STEP 10: CIRCLE(I,154),2: PAINT (I,154):NEXT
200 DRAW "bm35,145r20g10120e10bm195,147r10f5110h5bm215,147r10f5110h5":
PAINT (220,150)
210 DRAW "bm200,175g5r10h5":CIRCLE(225,177),10:LINE(225,167)-(225,187)
:LINE(215,177)-(235,177)
220 LINE(21,141)-(234,21),15,B:DRAW "bm40,139u4e3f3d4bm215,139u4e3f3d4":PAINT (42,138):PAINT (217,138)
230 IF NP=0 THEN 240 ELSE DRAW "bm35,8":FOR I=1 TO NP: DRAW "g3f3e3h3br9": NEXT
240 PRESET(17,178): PRESET (17,179): PRESET (17,180): PRESET (18,179)
250 IF NP <> 0 THEN 290
260 FOR J= 1 TO 3: X1=120:X2=141:Y1=80:Y2=90
270 FOR I=1 TO 10:B$="n"+STR$(I+20): PLAY "m299s1119xb$":LINE(X1,Y1)-(X2,Y2),15,B:X1=X1-I:X2=X2+I:Y1=Y1-I:Y2=Y2+I: NEXT: LINE (45,130)-(211,24),15,B
```

```

280   X1=120:X2=141:Y1=80:Y2=90:FOR
      I=1 TO 10:B$="n"+STR$(30-I):PLAY"m29
951119xb$;":LINE(X1,Y1)-(X2,Y2),1,B:X
1=X1-I:X2=X2+I:Y1=Y1-I:Y2=Y2+I:NEXT:L
INE(45,130)-(211,24),1,B:NEXT
290   PUT SPRITE 0,(120,70),9
300   PX=0:PY=0:STRIG(0)ON:ON STRIG
      GOSUB 520

```

Control

```

310   A=PEEK(V4VVV):X=VPEEK(6913):Y
      =VPEEK(6912):IF X <30 THEN POKEV4VVV,
PEEK(V4VVV) AND 13 ELSE POKE V4VVV,FE
EK(V4VVV) OR 2
320   IF X>209 THEN POKEV4VVV,PEEK(
V4VVV) AND 7 ELSE POKEV4VVV,PEEK(V4VV
V) OR 8
330   IF Y<30 THEN POKEV4VVV,PEEK(V
4VVV) AND 14 ELSE POKE V4VVV,PEEK(V4V
VV) OR 1
340   IF Y>100 THEN POKEV4VVV,PEEK(
V4VVV) AND 11 ELSE POKEV4VVV,PEEK(V4V
VV) OR 4
350   D=USR(D)
360   IF PL=0 AND RND(1)<.1 THEN GO
SUB 390
370   IF PL=1 THEN GOSUB 430
380   GOTO 310

```

Aparición planetoide

```

390   CX=INT(RND(1)*90+80):CY=INT(R
ND(1)*30+60):R=3:CIRCLE(CX,CY),R:PAIN
T(CX,CY)
400   PL=1:PUT SPRITE 1,(170,176),1
410   PX=CX+INT(RND(1)*80-40):PY=CY
+INT(RND(1)*70-35)
420   RETURN

```



Agrandamiento de planetaoide

```
430     IF RND(1)<.2 THEN R=R+2
440     VR=RND(1)/3+.8:CIRCLE(CX,CY),
R,15,0,6.28,VR
450     VPOKE6917,160-3*R
460     IF VPEEK(6917)<15 THEN 580
470     RETURN
```

Planetoide destruido

```
480     STRIG(0) OFF:FOR I=1 TO 50 :P
LAY"140m380s8n24":X1=INT(RND(1)*90-45
+CX):Y1=INT(RND(1)*90-45+CY):X2=INT(R
ND(1)*200+25):Y2=INT(RND(1)*110+25)
490     LINE(X1,Y1)-(X2,Y2),1:CIRCLE(
X1,Y1),1:CIRCLE(X2,Y2),2:PAINT(X2,Y2)
:NEXT
500     PUT SPRITE 1,(100,200):PL=0
510     NP=NP+1:CLS:GOTO 160
```

Disparo del laser

```
520     IX=VPEEK(6913)+8:IY=VPEEK(691
2)+8:LINE(44,132)-(IX,IY),11:LINE(217
,132)-(IX,IY),11
530     PLAY"110m1000s14n33"
540     LINE(44,132)-(IX,IY),1:LINE(2
17,132)-(IX,IY),1
550     IF (PX-CX)^2+(PY-CY)^2>=R*R T
HEN RETURN
560     IF ABS(IX-PX)<15 AND ABS(IY-P
Y)<15 THEN 480
570     RETURN
```

Fin de juego

```
580 FOR I=1 TO 100 : PLAY "164m200
s14n23":X1=INT(RND(1)*255):Y1=INT(RND
(1)*192):X2=INT(RND(1)*255):Y2=INT(RN
D(1)*192):LINE(X1,Y1)-(X2,Y2):NEXT
590 SCREEN 1: PRINT "-COLISION_CON
_EL_PLANETOIDE!":PRINT:PRINT:PRINT:PR
INT"PLANETOIDES_DESTRUIDOS: ";NP
600 IF INKEY$="" THEN END ELSE 600
```

Datos de sprites

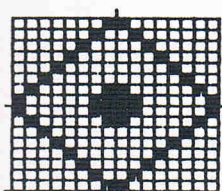
```
610 DATA 1,2,4,8,16,32,65,131,131,6
5,32,16,8,4,2,1,128,64,32,16,8,4,130,
193,193,130,4,8,16,32,64,128
620 DATA 24,60,126,126,60,24,0,0
```

## Tabla de claves

10	= 904	170	= 2101	330	= 4440
20	= 143	180	= 4750	340	= 4508
30	= 1637	190	= 5392	350	= 677
40	= 1541	200	= 4938	360	= 2144
50	= 1712	210	= 5386	370	= 1100
60	= 143	220	= 5307	380	= 206
70	= 143	230	= 4194	390	= 5945
80	= 2571	240	= 2359	400	= 1474
90	= 2998	250	= 1059	410	= 4839
100	= 1792	260	= 2929	420	= 142
110	= 2016	270	= 10214	430	= 1843
120	= 680	280	= 12621	440	= 3142
130	= 3193	290	= 854	450	= 1064
140	= 2294	300	= 2420	460	= 1259
150	= 3670	310	= 7542	470	= 142
160	= 5811	320	= 4618	480	= 11782

490	=	3540	540	=	2747	590	=	5291
500	=	1365	550	=	3446	600	=	1352
510	=	1400	560	=	3466	610	=	4586
520	=	5299	570	=	142	620	=	1250
530	=	1196	580	=	10975	TOTAL	=	198694

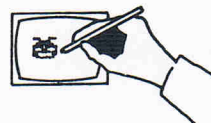
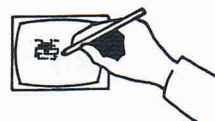
## Diseño de los sprites



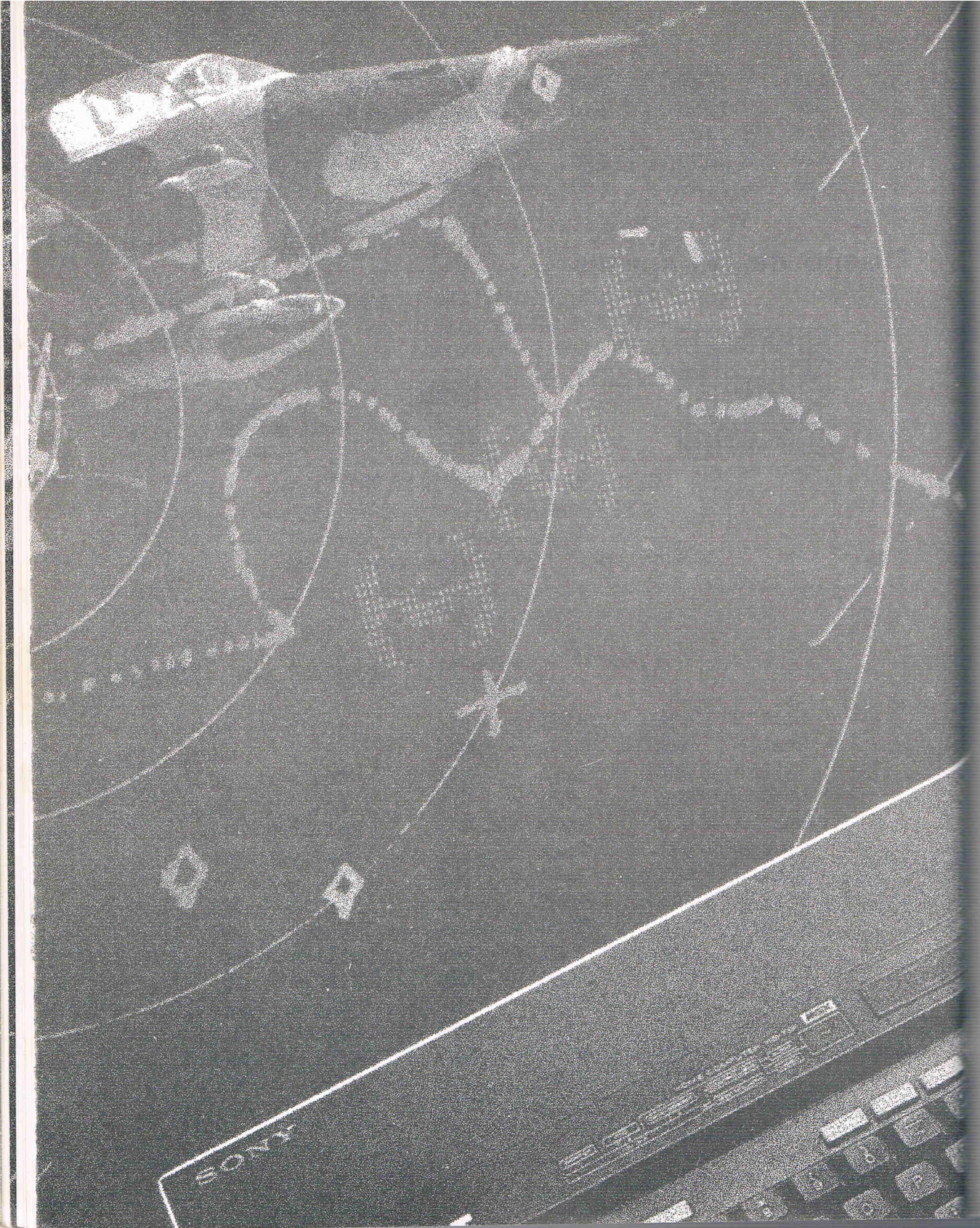
PUNTO DE MIRA



PLANETOIDE  
EN LA PANTALLA DE RADAR







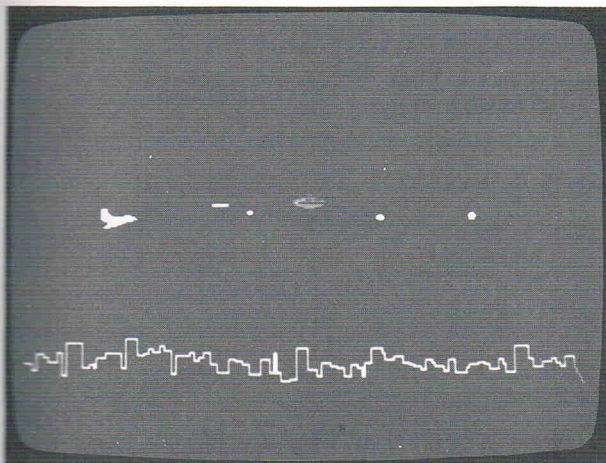
SONY

Calculator keypad with keys labeled 1 through 9, 0, and other function keys.



# O.V.N.I.

# 4



## ¡ATENCIÓN!

Este programa utiliza la subrutina SOPORTE en lenguaje máquina. Véase apéndice C.

**Tipo de juego:**

**JUEGO DE DISPARO**

Da caza a los O.V.N.I.s en tu nave espacial equipada con misiles. Lograrás despegar de la Tierra si no chocas con el terreno o con un O.V.N.I. Date prisa, tu tiempo se va agotando.

Utiliza las teclas de movimiento de cursor para subir, bajar y variar tu velocidad; para disparar, tecléa la barra espaciadora. Para que aparezca tu nave al comienzo, pulsa el cursor hacia abajo.

# SUGERENCIAS AL PROGRAMA

Fija el tiempo de duración de cada pantalla; T1 y T2 se ajustan en la línea 200.

Si te sientes con ganas, añade otra nueva pantalla a continuación de la segunda; pero la tienes que diseñar tú mismo.

## Variables del programa

DM Disparo de misil.  
T1, T2 Límites de tiempo de las pantallas 1 y 2.  
NO Número de O.V.N.I.s destruidos.  
FJ Cómo finalizó el juego.

Inicialización

```
10      REM      ** ATENCION **
20      REM
30      REM ESTE PROGRAMA NECESITA
40      REM LA SUBROUTINA SOPORTE
50      REM Vease el Apendice C
60      REM
70      REM
80      A=PEEK(64586!)+256*PEEK(6458
7!)
90      V1=A+9:V2=A+8:V3=A+7:V4=A+6
100     COLOR 15,1,1: SCREEN 2,2
110     FOR I= 1 TO 32 : READ Q : A$
=A$ + CHR$ (Q) : NEXT : SPRITE$ (0) =
A$ : A$=""
120     FOR I= 1 TO 32 : READ Q : A$
= A$ + CHR$ (Q) : NEXT : SPRITE$ (1)
= A$ : SPRITE$ (2)= CHR$ (255) + CHR
$ (255) : FOR I = 14432 TO 14464: VPO
KE I,2 ^ (INT(RND(1)*8))
130     NEXT
140     DRAW "BM10,30D134R1D4R2D3R3D
2R4D1R55U1R4U2R3U3R2U4R1U134L21D110L1
```



```

D4L1D3L1D2L1D1L25U1L1U2L1U3L1U4L1U110
L20"
150      DRAW "BM99,30D145R20U50R30U2
0L30U45R55U20L75": PAINT (100,100)
160      DRAW"BM195,30R40D1R4D2R3D3R2
D4R1D126L1D4L2D3L3D2L4D1L40U1L4U2L3U3
L2U4L1U126R1U4R2U3R3U2R4U1"
170      DRAW "BM208,51R14D1RD1R2D2R1
D98L1D2L2D1L14U1L2U2L1U98R1U2R2U1": P
AINT (200,60)
180      PUT SPRITE 1, (0,5),8:DEFUSR
= A+10: DEFUSR1 =A+128: POKE V4VVV,1
3 : POKE V3VVV,1 : POKE V2VVV,3 : POK
E V1VVV,1
190      IF INKEY$="" THEN D=USR1(D)
: GOTO 190
200      T1=60*60: T2= 69*30 : TIME=0

```

Pantalla 1

```

210      CLS : COLOR 15,1,1 : DRAW "bm
0,170r5d12r2u3r6d4r2u10r3d2r2d3r4u6r2
d14r2u20r7d15r3u2r2d2r1u6r4u3r7d9r2u1
8r5d10r3u1r2u3r2d2r3u4r2d3r3d14r2u11r
6u2r2d5r2u2r2u3r3d7"
220      DRAW "r3d5r5u8r3d1r3d2r3d6r5u
9r4d8r2u12r1d12r2d5r4u1r3u19r5d13r4d4
r2u8r4d2r3d1r3d5r3u7r3d2r2d5r1u8r3u9r
5d3r1d4r4u2r4d4r3d1r2d1r3d2r2u6r3d2r1
d2r4d3r2u7r3"
230      DRAW "d8r5u3r1u2r4u1r3d1r2d3r
4u6r2d3r1d5r4u16r6d8r2d5r4u3r4u2r2d3r
2d2r2u6r4d6m255,192"
240      PUT SPRITE 1, (120,70),8: PUT
SPRITE 2, (0,80),11
250      STRIG(0) ON: ON STRIG GOSUB 4
10
260      SPRITE ON : ON SPRITE GOSUB 4
40

```

Control 1

```
270     IF VPEEK(6912)>175 THEN POKE
V4VVV,12 ELSE POKE V4VVV,13
280     D=USR(D) : IF VPEEK(6912)>1
60 AND VPEEK(6912)<175 THEN CJ=2: GO
TO 580
290     POKE V2VVV,3 : POKE V1VVV,4 :
POKE V3VVV,0 : D =USR1(D)
300     GOSUB 340
310     IF DM =1 THEN GOSUB 380
320     IF TIME>T1 THEN GOTO 470
330     GOTO 270
```

Movimiento O.V.N.I.

```
340     POKE V3VVV,1: POKE V1VVV,7 :
IF DM=0 THEN POKE V2VVV,3 : GOTO 360
350     IF RND(1)<.6 THEN POKE V2VVV
,3 ELSE IF RND(1) <.52 THEN POKE V2VV
V,0 ELSE POKE V2VVV,2
360     D=USR1(D)
370     RETURN
```

Movimiento del avión

```
380     IF VPEEK(6921)>238 THEN DM=0:
PUT SPRITE 2,(101,200): RETURN
390     VPOKE 6921,VPEEK(6921)+16
400     RETURN
```

Disparo del avión

```
410     IF DM=1 OR VPEEK(6913)>243 TH
EN RETURN
```

```
420     DM=1 : PUT SPRITE 2,(VPEEK(69
13)+16,VPEEK(6912))
430     RETURN
```

Choque

```
440     SPRITE OFF : IF DM =0 AND ABS
(VPEEK(6913)-VPEEK(6917))< 17 THEN CJ
= 1 : VPOKE 6914,12 : FOR I1 =1TO 200
: NEXT : VPOKE 6918,12: FORI1 = 1 TO
800 : NEXT : GOTO 580
450     IF ABS(VPEEK(6916)-VPEEK(6920
))<8 THEN NO = NO +1: VPOKE 6918,12 :
PUT SPRITE 2,(101,200) : PLAY "14m20
00s14n15" : FOR I1 = 1 TO 500 : NEXT
: VPOKE 6918,4 : DM =0 : PUT SPRITE 1
,(0,70+INT(RND(1)*20)) : SPRITE ON :
RETURN
460     DM = 0 : PUT SPRITE 2,(101,22
0): SPRITE ON : RETURN
```

Pantalla 2

```
470     COLOR 15,1,1: CLS
480     CIRCLE(220,5),5,15,0,6,.5
490     FOR I=1 TO 9: CIRCLE(220,5+3*
I),10+2*I,15,0,6.28,.5: NEXT : FOR I=
1 TO 9 : CIRCLE(220,33+3*I),28-2*I,15
,0,6.28,.5 : NEXT
500     FOR I=1 TO 255 STEP 6 : FOR J
=0 TO 190 STEP 6 : IF RND(1)<.1 THEN
PSET(I,J),15
510     NEXT : NEXT :TIME =0: SPRITE
ON
```

Control 2

```
520     D=USR(D)
530     POKE V2VVV,3 : POKE V1VVV,5 :
```



```
POKE V3VVV,0: D=USR1(D)
540 GOSUB 340
550 IF DM=1 THEN GOSUB 380
560 IF TIME>T2 THEN CJ=3: GOTO 58
0
570 GOTO 520
```

Fin de juego

```
580 SCREEN 1 : PRINT "_______FI
N_DE_JUEGO": PRINT : PRINT
590 PRINT "HAS_DESTRUIDO_";NO;" 0
.V.N.I.s INVASORES": PRINT : PRINT
600 IF CJ=1 THEN PRINT "COLISION_
CON_UN_D.V.N.I."
610 IF CJ=2 THEN PRINT "CHOCASTE_
CON_LA_SUPERFICIE!"
620 IF CJ=3 THEN PRINT "TIEMPO_AC
ABADO:LOS_INVASORES":PRINT "REGRESAN_
A_SU_PLANETA_DE_____ORIGEN"
630 IF INKEY$="" THEN END ELSE 63
0
```

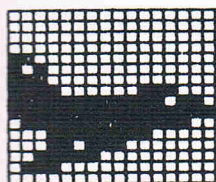
Datos de sprites

```
640 DATA 0,0,0,224,224,224,240,248,
255,255,127,31,31,62,112,0,0,0,0,0,0,
0,0,56,254,255,252,224,128,0,0,0
650 DATA 0,0,0,3,5,15,126,255,240,1
27,63,7,0,0,0,0,0,0,0,192,64,240,102,
255,0,254,252,224,0,0,0,0
```

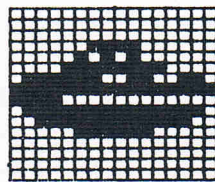
## Tabla de variables

10	= 904	240	= 1660	470	= 560
20	= 143	250	= 1600	480	= 1055
30	= 1637	260	= 1094	490	= 7019
40	= 1541	270	= 2786	500	= 4167
50	= 1712	280	= 3945	510	= 1243
60	= 143	290	= 2697	520	= 677
70	= 143	300	= 240	530	= 2698
80	= 2571	310	= 1039	540	= 240
90	= 2998	320	= 1297	550	= 1039
100	= 680	330	= 166	560	= 1611
110	= 3659	340	= 3020	570	= 161
120	= 8633	350	= 4802	580	= 1980
130	= 131	360	= 695	590	= 3346
140	= 6351	370	= 142	600	= 2559
150	= 3135	380	= 2956	610	= 2800
160	= 5624	390	= 1130	620	= 5462
170	= 4339	400	= 142	630	= 1382
180	= 5487	410	= 2188	640	= 4728
190	= 1994	420	= 2385	650	= 4370
200	= 2085	430	= 142		
210	= 11201	440	= 7213		
220	= 11927	450	= 10693	TOTAL	= 184600
230	= 6451	460	= 1982		

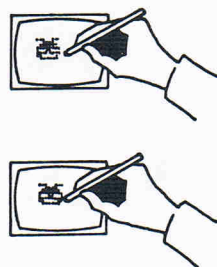
## Diseño de los sprites



CAZA



O.V.N.I.



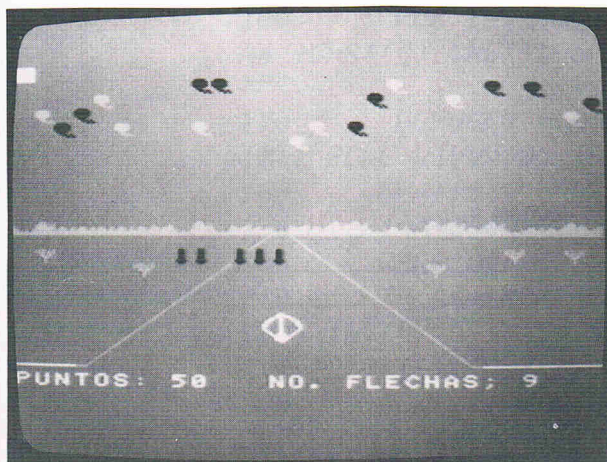






# Globos

# 5



## ¡ATENCIÓN!

Este programa utiliza la subrutina SOPORTE en lenguaje máquina. Véase apéndice C.

**Tipo de juego:**

**JUEGO DE TIRO AL BLANCO**

Prueba tu puntería lanzando flechas entre unos globos, intentando no estallarlos. Un buen disparo marca 10 puntos; pero ten en cuenta que sólo tienes 15 disparos.

El juego emplea una subrutina en código máquina para mover la pantalla de derecha a izquierda.

# SUGERENCIAS AL PROGRAMA

El número de lanzamientos permitidos es arbitrario; puedes cambiarlo dándole otro valor a FL en la línea 320.

Puedes modificar este juego cambiando K-3 por K-5 en la línea 440, y el 15 por un 8 en la línea 350.

## Variables del programa

PT	Puntuación.
FL	Número de flechas lanzadas.
L	Lanzamientos restantes.
CL	Control de lanzamientos.
CF, FF	Columna de flechas; fila de flechas.

Inicialización

```
10      REM    ** ATENCION **
20      REM
30      REM ESTE PROGRAMA NECESITA
40      REM LA SUBROUTINA SOPORTE
50      REM Vease el Apendice C
60      REM
70      REM
80 COLOR 15,7,7:SCREEN 1,2:KEY OFF
90      A=PEEK(64586!)+256*PEEK(6458
7!)
100     V1=A+9:V2=A+8:V3=A+7:V4=A+6
110     W1=A+249:W2=A+248:W3=A+247:W
4=A+246
120     FOR I=1072 TO 1079:READ Q:VP
OKE I,Q:VPOKE I+48,Q:VPOKE 80+I,Q:NEX
T:FOR I=1088 TO 1119:READ Q:VPOKE I,Q
:NEXT:FOR I=1128 TO 1135:READ Q:VPOKE
I,Q:NEXT:FOR I=1280 TO 1311:READ Q:V
POKE I,Q:NEXT:FOR I=1216 TO 1231:READ
Q:VPOKE I,Q:NEXT
130     VPOKE 8208,135 : VPOKE 8209,
231 : VPOKE 8210,167 : VPOKE 8211,23
```



```

: VPOKE 8212,231
140   A$="":FOR I=1 TO 32 : READ Q
: A$=A$ + CHR$(Q) : NEXT : SPRITE$(0
) = A$ : A$ =" "
150   FOR I =1 TO 32: READ Q : A$=
A$ + CHR$(Q) :NEXT: SPRITE$(1)=A$: A$
=" "
160   FOR I=1 TO 32 : READ Q : A$=
A$ +CHR$(Q) :NEXT: SPRITE$(2)=A$: A$=
" "
170   DEFUSR0 = A+10 : POKE V4VVV,
10 : POKE V1VVV,8
180   B=A+10
190   I=A+250
200   READ A$
210   IF A$="R" THEN READ C:C=C+B:
C1 = INT(C/256):C2=256*(C/256-C1):POK
E I,C2:I=I+1:POKE I,C1 :GOTO 230
220   POKE I, VAL(A$)
230   IF I<A+250+27THEN I=I+1:GOTO
200
240   DEFUSR2=A+250! : POKE W4WWW,3
3: POKE W3WWW,24 : POKE W2WWW,255 : P
OKE W1WWW,24
250   FOR I=6441 TO 6454:VPOKE I,1
34:NEXT:FOR I=6569 TO 6582:VPOKE I,13
4:NEXT:VPOKE 6473,134:VPOKE 6486,134:
VPOKE 6537,134:VPOKE 6550,134:FOR I=1
TO 11:PRINT:NEXT:PRINT TAB(7) CHR$(1
34) "___GLOBOS___" CHR$(134)
260   IF INKEY$="" THEN D=RND(1) :
GOTO 260

```

Preparación pantalla

```

270   CLS
280   FOR I=6816 TO 6820 : VPOKE I,
23 : NEXT : FOR I =6842 TO 6847 : VPO
KE I,23 : NEXT : FOR I = 6542 TO 6790
STEP 31 : VPOKE I,162 : NEXT

```

```

290   FOR I=6544 TO 6808 STEP 33:VP
DKE I,163:NEXT:VPOKE 6821,160:VPOKE 6
841,161
300   FOR I=6496 TO 6527 : VPOKE I,
INT(RND(1)*4+136): NEXT
310   VPOKE 6592,141: VPOKE 6590,14
1 :VPOKE 6600,141: VPOKE 6615,141 : V
POKE 6563,141 : VPOKE 6587,141
320   FL=15: L=FL
330   PUT SPRITE 0,(117,140) : PUT
SPRITE 2,(118,139)
340   GOSUB 630
350   INTERVAL ON: ON INTERVAL= 15
GOSUB 510
360   STRIG(0) ON :ON STRIG GOSUB 5
20
370   FOR I=6240 TO 6303 : IF RND(1
)<.1THEN VPOKE I,134
380   NEXT

```

Control

```

390   D=USR(D) : IF CL = 0 THEN VPO
KE 6921, VPEEK(6913)
400   K=VPEEK(6913) : IF K<= 69 THE
N POKE V4VVV,8 ELSE IF K>=165 THEN PO
KE V4VVV,2 ELSE POKE V4VVV,10
410   IF RND(1)<.03+PT/2000THEN GOS
UB 480
420   IF CL=1 THEN GOSUB 440
430   GOTO 390

```

Movimiento de flecha

```

440   K=VPEEK(6920) : VPOKE 6920,K-
3: IF K>55 THEN RETURN

```



Comprobación de choque

```
450     IF K < 7 THEN PLAY "s1m200019
n55n53": VPOKE 6570+FL-L,152: PT = P
T+10 : L=L-1 : GOSUB 630 : CL=0 : VP
OKE 6920,139 : VPOKE 6914,0 :RETURN
460     FF = INT(K/8) : J=6144+FF*32+
CF:K1 = VPEEK(J):IF K1 <> 134 AND K1
<>140 THEN RETURN
470     INTERVAL OFF : VPOKE J,153 :
CL = 0 : PLAY "110s10m90n20" : VPOKE
6920,139 : VPOKE 6921,VPEEK(6913)+1 :
VPOKE 6914,0: L=L-1 : GOSUB 630 : VP
OKE J,32 : INTERVAL ON : RETURN
```

Añadir globos

```
480     IF RND(1) <.5 THEN K1=134 ELS
E K1=140
490     VPOKE 6207+INT(RND(1)*6)*32,K
1
500     RETURN
```

Movimiento de pantalla

```
510     D=USR2(D) : RETURN
```

Disparar flecha

```
520     IF CL = 1 THEN RETURN
530     IF L=0 THEN 570
540     CL=1 : VPOKE 6914,4 : VPOKE 6
920,130
550     CF = INT(VPEEK(6921)/8)+1
560     RETURN
```

Fin de juego

```
570     ON INTERVAL= 7 GOSUB 510 : IN
TERVAL ON :POKE W2WWW,223 : POKE W1WW
W,26
580     PUT SPRITE 0,(100,200) : PUT
SPRITE 2,(100,200) : CLS : FOR I = 1
TO 30 : VPOKE 6177+ INT(RND(1)*700),1
34 : PLAY "164s8m500n50": NEXT
590     PRINT "_____TU_PUNTUACION_ES:
";PT
600     IF INKEY$="" THEN 610 ELSE 600
610     IF INKEY$="" THEN 610
620     SCREEN 1: END
```

Puntuación

```
630     FOR I = 1 TO 22 : PRINT :NEXT
: PRINT "PUNTOS:";PT ; TAB(13) ; "ND.
FLECHAS;" ; L ; CHR$(11);
640     RETURN
```

Datos de sprites pequeños

```
650     DATA 120,252,252,252,120,48,26,
5
660     DATA 0,0,0,49,123,255,255,255
670     DATA 0,0,131,207,255,255,255,25
5
680     DATA 48,121,251,255,255,255,255
,255
690     DATA 0,0,0,0,129,201,255,255
700     DATA 81,213,85,54,28,8,8,8
710     DATA 1,2,4,248,0,0,0,0
720     DATA 128,64,32,31,0,0,0,0
730     DATA 1,2,4,8,16,32,64,128
```

740 DATA 128,64,32,16,8,4,2,1  
 750 DATA 16,16,16,16,16,16,56,16  
 760 DATA 146,4,32,9,128,16,69,8

Control de sprites

770 DATA 1,6,24,48,96,64,192,128,12  
 8,64,32,16,8,4,2,1,128,96,24,12,6,2,3  
 ,1,1,2,4,8,16,32,64,128  
 780 DATA 1,6,24,48,96,192,192,255,0  
 ,0,0,0,0,0,0,0,128,96,24,12,6,3,3,255  
 ,0,0,0,0,0,0,0,0  
 790 DATA 1,3,5,1,1,1,1,1,1,1,1,3,  
 5,3,5,0,128,64,0,0,0,0,0,0,0,0,128,  
 64,128,64

Datos de código máquina

800 DATA 42,R,236,43,35,205,R,187,4  
 3,205,R,204,35,237,91,R,238,123,189,1  
 94,R,244,122,188,194,R,244,201

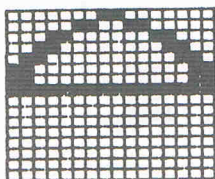
## Tabla de claves

10	= 904	130	= 2534	250	= 9534
20	= 143	140	= 4125	260	= 1911
30	= 1637	150	= 3660	270	= 159
40	= 1541	160	= 3661	280	= 5598
50	= 1712	170	= 2306	290	= 3479
60	= 143	180	= 636	300	= 3165
70	= 143	190	= 883	310	= 4143
80	= 1188	200	= 236	320	= 934
90	= 2571	210	= 6502	330	= 1762
100	= 2998	220	= 854	340	= 275
110	= 3954	230	= 2577	350	= 2919
120	= 10889	240	= 4075	360	= 1455

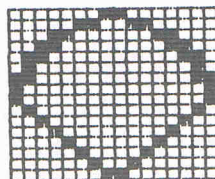


370	= 2736	530	= 763	690	= 1247
380	= 131	540	= 1286	700	= 1183
390	= 2341	550	= 1925	710	= 941
400	= 5283	560	= 142	720	= 1094
410	= 2499	570	= 4495	730	= 1112
420	= 1097	580	= 6402	740	= 1112
430	= 286	590	= 1813	750	= 1268
440	= 2505	600	= 1337	760	= 1230
450	= 6895	610	= 1014	770	= 4352
460	= 5926	620	= 402	780	= 3935
470	= 8172	630	= 4182	790	= 3561
480	= 2509	640	= 142	800	= 4907
490	= 2205	650	= 1458		
500	= 142	660	= 1311		
510	= 896	670	= 1462	TOTAL	= 195432
520	= 899	680	= 1628		

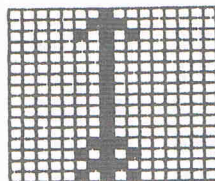
## Diseño de los sprites



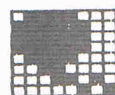
ARCO



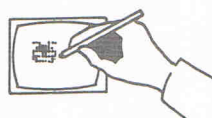
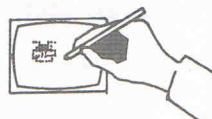
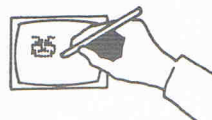
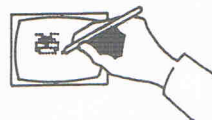
ARCO TENSO



FLECHA

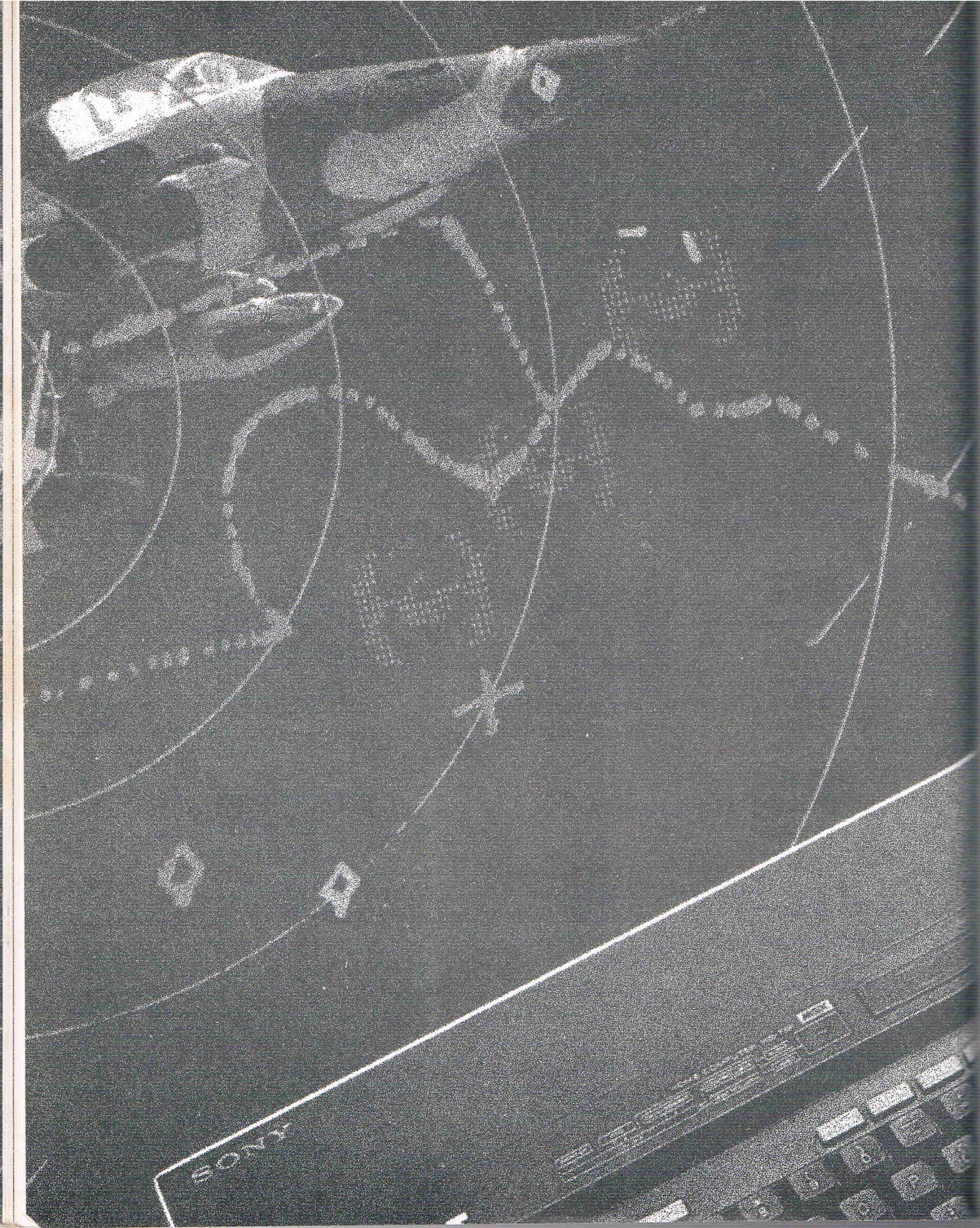


GLOBO  
(CARACTER)









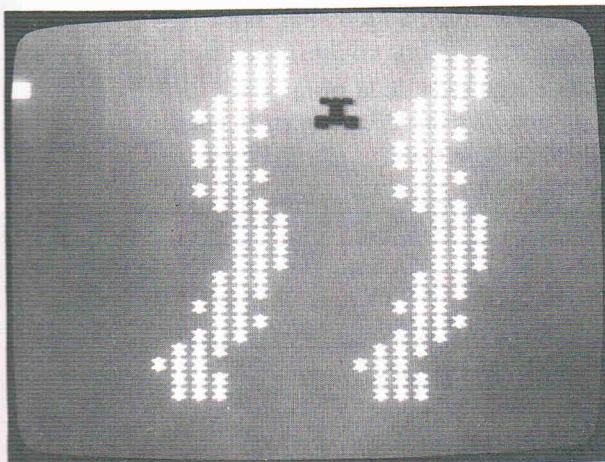
SONY

Calculator keypad with numbers 1-9, 0, and function keys.



# Dragster

# 6



## ¡ATENCIÓN!

Este programa utiliza la subrutina  
SOPORTE en lenguaje máquina.  
Véase apéndice C.

**Tipo de juego:**

**JUEGO DE CARRERAS**

Conduce todo lo rápido que puedas, pero cuídate de no estrellarte con los límites de la pista o pegar un patinazo sobre el aceite.

# SUGERENCIAS AL PROGRAMA

La velocidad está controlada por el valor 9 de la línea 360. Si deseas mayor facilidad, puedes ensanchar la pista en la subrutina "Línea nueva".

## Variables del programa

D1	Distancia.
LN\$(I)	Línea a imprimir en pantalla.
I(I), D(I)	Coordenadas de la derecha e izquierda de la pista.
CA	Control de la existencia de una mancha de aceite en la pista.
C	Columna de la parte izquierda de la pista.

```

Inicialización

10      REM      ** ATENCION **
20      REM
30      REM ESTE PROGRAMA NECESITA
40      REM LA SUBROUTINA SOPORTE
50      REM Vease el Apendice C
60      REM
70      REM
80      A=PEEK(64586!)+256*PEEK(6458
90      7!)
90      V1=A+9:V2=A+8:V3=A+7:V4=A+6
100     W1=A+249:W2=A+248:W3=A+247:W
110     4=A+246
110     SCREEN 1,2:KEY OFF : COLOR 1
120     5,4,4
120     FOR I=1 TO 32:READ Q:A$ = A$
130     +CHR$(Q):NEXT:SPRITE$(0)=A$:PUT SPRIT
140     E 0,(123,0),1
130     A$="":FOR I=1 TO 8:READ Q:A$
140     =A$+CHR$(Q):NEXT:SPRITE$(1)=A$
140     DEFUSR=A+10:DEFUSR1=A+128!:P
150     OKE V4VVV,10:POKE V1VVV,3:POKE V3VVV,
160     0:POKE V2VVV,0
150     B=A+10
160     I=A+250
```



```

170      READ A$
180      IF A$="R" THEN READ C:C=C+B:
        C1 = INT(C/256):C2=256*(C/256-C1):PO
KE I,C2:I=I+1:POKE I,C1 :GOTO 200
190      POKE I, VAL(A$)
200      IF I<A+250+31 THEN I=I+1:GO
TO 170
210      DEFUSR2=A+250:POKE W4WWW,223
:POKE W3WWW,26:POKE W2WWW,0:POKE W1WW
W,24
220      PRINT "_____***__***":
PRINT "___DA_A___***__***":PRINT "___
_____***__***":PRINT "CUALQUIER_
**_D_***":PRINT "___TECLA___**_R_***"
:PRINT"_____***_A_***"
230      PRINT "___PARA___**_G_***":
PRINT "___EMPEZAR_***_S_***":PRINT "___
_____***_T_***":PRINT "_____
**_E_***":PRINT "_____***_R_***"
:PRINT "_____***__***"
240      PRINT "_____***__***":
PRINT "_____***__***": PRINT "
_____***__***": PRINT "
_____***__***": PRINT "_____
***": PRINT "_____***__***": PR
INT "_____***__***"
250      FOR I =1 TO 5:PRINT "_____
___***__***": NEXT: PRINT "_____
_***__***";
260      IF INKEY$("<") THEN 280
270      D=RND(1):D=USR1(D):GOTO 260

```

Preparación pantalla

```

280      PRINT :PUT SPRITE 0,(123,176
),1:FOR I=1 TO 23:PRINT"_____***
___***": NEXT
290      PRINT CHR$(11):FOR I=1 TO 6:
PRINT : NEXT:PRINT "_____+-+": FOR I=

```

```

1 TO 7: PRINT ".....!_!": NEXT:PRINT"
.....+--+": FOR I=1 TO 7 : PRINT ".....
_!":NEXT
300      VPOKE 6439,133:VPOKE 8208,14
8
310      FOR I=1 TO 800: NEXT :VPOKE
6503,133:VPOKE 6208,164:VPOKE 6439,32
:FOR I=1 TO 800: NEXT: VPOKE 6503,32:
VPOKE 6567,133:VPOKE 8208,36
320      IF INKEY$="" THEN 320 ELSE PO
KE V1VVV,2:PLAY"12s8m40000n2"
330      D=USR1(D):IF VPEEK (6912)>25
THEN 330
340      PRINT CHR$(11);: FOR I=1 TO
4:LN$(I)=".....**.....**.....
.....": PRINT LN$(I);: PRINT ".....
.....";:I(I)=110:D(I)=1
36: NEXT
350      POKE V1VVV,7:C=11:ON SPRITE G
OSUB 540: SPRITE ON
360      ON INTERVAL=9 GOSUB 450:INTE
RVAL ON
370      FOR I=152 TO 175 STEP 2:VPOK
EI,170:VPOKE I+1,85: NEXT:VPOKE 8194,
228

```

No hacer nada

```

380      GOTO 380

```

Nueva línea

```

390      FOR I=4 TO 2 STEP-1:LN$(I)=LN
$(I-1):D(I)=D(I-1):I(I)=I(I-1):NEXT
400      I=INT(RND(1)*3-1):C=C+I: IF C
<2 THEN C=2 ELSE IF C>12 THEN C=12
410      LN$(1)="
":MID$(LN$(1),C)="***

```



```
***":I(1)=8*C+28:D(1)=I(1)+48
420 IF RND(1)<.05+DI/1000 AND CA=
0 THEN PUT SPRITE 1,(I(1)+INT(RND(1)*
35+5),0),1:CA=1
430 IF VPEEK(6916)>190 THEN PUT S
PRITE 1,(200,200):CA=0
440 RETURN
```

Control

```
450 D=USR2(D)
460 GOSUB 390:PRINT CHR$(11);LN$(
1):PLAY"110m59000s10n4"
470 IF CA=1 THEN VPOKE 6916,VPEEK
(6916)+8
480 DI=DI+1
490 GOSUB 520
500 D=USR(D)
510 RETURN
```

Comprobación de choque

```
520 IF VPEEK(6913)<I(4) OR VPEEK(
6913)>D(4) THEN 540
530 RETURN
```

Fin de juego

```
540 INTERVAL OFF:PLAY"11m60000s8n
20":FOR I=6144 TO 6303:VPOKE I,32:NEX
T:PRINT CHR$(11);"E_S_T_R_E_L_L_A_D_D
_L!_!_!":PRINT:PRINT"DISTANCIA:";DI;"K
M"
550 IF INKEY$="" THEN END ELSE 55
0
```

Datos de sprites

560 DATA 57,63,59,3,3,3,3,3,3,3,3,247  
 ,247,255,243,240,240,156,252,220,192,  
 192,192,192,192,192,192,239,239,255,2  
 07,15,15  
 570 DATA 112,236,88,248,204,0,0,0

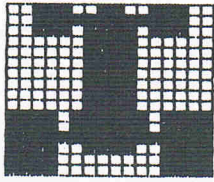
Rutina de movimiento

580 DATA 42,R,236,35,43,205,R,187,1  
 7,32,0,25,205,R,204,237,82,237,91,R,2  
 38,123,189,194,R,244,122,188,194,R,24  
 4,201

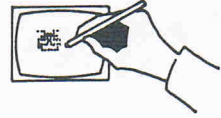
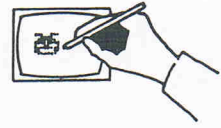
## Tabla de claves

10	= 904	210	= 4146	410	= 5292
20	= 143	220	= 6393	420	= 5801
30	= 1637	230	= 6277	430	= 2788
40	= 1541	240	= 6515	440	= 142
50	= 1712	250	= 2825	450	= 696
60	= 143	260	= 1178	460	= 2614
70	= 143	270	= 1764	470	= 1860
80	= 2571	280	= 3059	480	= 780
90	= 2998	290	= 5852	490	= 165
100	= 3954	300	= 989	500	= 677
110	= 1182	310	= 4992	510	= 142
120	= 4032	320	= 2967	520	= 2559
130	= 3670	330	= 2020	530	= 142
140	= 4850	340	= 6521	540	= 7687
150	= 636	350	= 2000	550	= 1302
160	= 883	360	= 2855	560	= 5477
170	= 236	370	= 3307	570	= 1307
180	= 6472	380	= 276	580	= 5494
190	= 854	390	= 4625		
200	= 2551	400	= 4846	TOTAL	= 159444

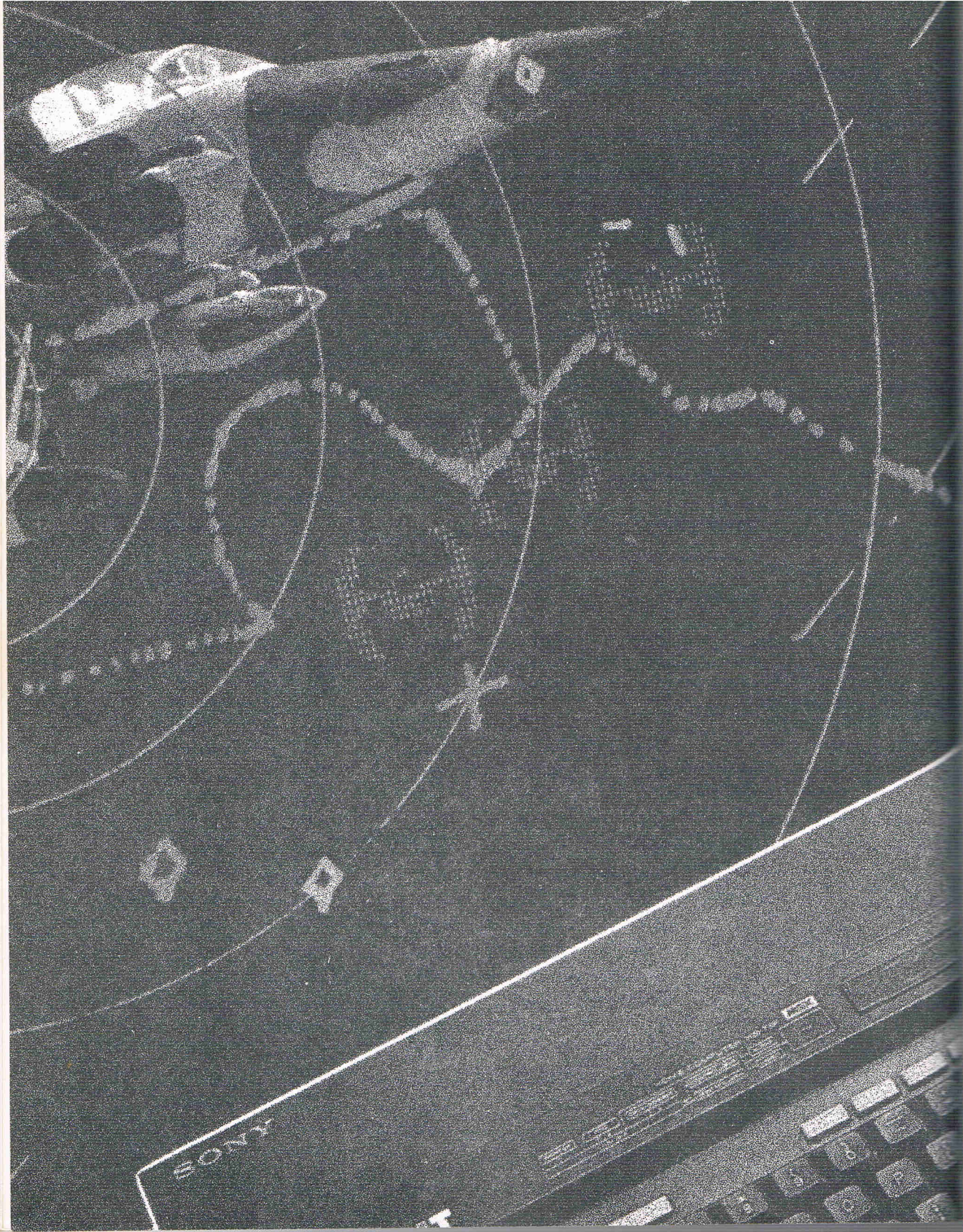
# Diseño de los sprites



COCHE



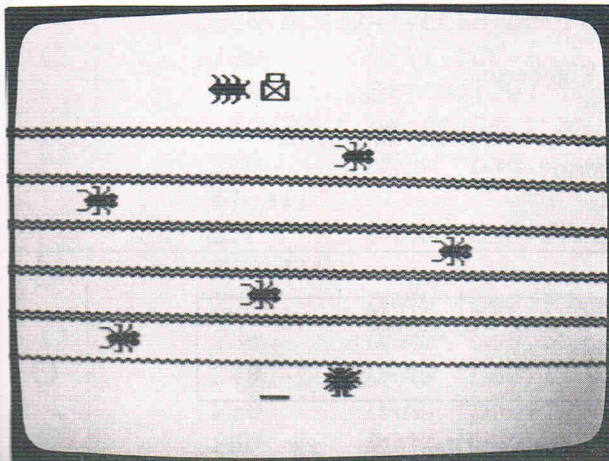






# Las cucarachas cósmicas

# 7



**¡ATENCIÓN!**

Este programa utiliza la subrutina  
SOPORTE en lenguaje máquina.  
Véase apéndice C.

**Tipo de juego:**

**JUEGO DE EVASION**

Rigoberto, "el Hambriento", peludo hombre de las montañas, ha llegado al dominio de las cucarachas cósmicas en busca de alimento. Conduce a Rigoberto a través de los túneles para que llegue hasta la cucaracha reina, y le quite parte de la comida que guarda. Cuando lo logre, todavía ha de volver al campamento.

Haz todos los viajes que puedas hasta que las cucarachas cósmicas te atrapen.

# SUGERENCIAS AL PROGRAMA

En la línea 400 se fija la frecuencia con que se acercan las cucarachas cósmicas a Rigoberto; haz los cambios que creas convenientes para aumentar o disminuir la dificultad. Si la rapidez de las cucarachas no te emociona, cámbiala en la línea 420.

## Variables del programa

- CC Cucaracha más cercana a Rigoberto.
- DC(I) Dirección de las cucarachas.
- NV Número de viajes completos que ha hecho Rigoberto.
- CO ¿Ha cogido comida Rigoberto?

### Inicialización

```
10      REM      ** ATENCION **
20      REM
30      REM ESTE PROGRAMA NECESITA
40      REM LA SUBROUTINA SOPORTE
50      REM Vease el Apendice C
60      REM
70      REM
80      CLEAR 700
90      A=PEEK(64586!)+256*PEEK(6458
7!)
100     V1=A+9:V2=A+8:V3=A+7:V4=A+6
110     COLOR 1,14,14 : SCREEN 2,2
120     FOR I=1 TO 32 : READ Q : A$
=A$ + CHR$(Q) : NEXT : SPRITE$(0)=A$:
A$=""
130     FOR I=1 TO 32: READ Q : A$=
A$+ CHR$(Q) : NEXT : FOR I=1 TO 5 : S
PRITE$(I)=A$: NEXT:A$=""
140     FOR I=1 TO 32 :READ Q : A$=A$+CHR
$(Q) : NEXT : SPRITE$(6) =A$
150     DEFUSR=A+10 : DEFUSR1=A+128
: POKE V4VVV,15
```



```

160      PUT SPRITE 0,(120,173),1 : P
UT SPRITE 1,(255,46),12 : PUT SPRITE
2,(255,72),12 : PUT SPRITE 3,(255,98)
,12 : PUT SPRITE 4,(255,124),12 : PUT
  SPRITE 5,(255,150),12 : PUT SPRITE 6
,(255,10),1
170      FOR I=1 TO 6: IF RND(1)<.5 T
HEN DC(I)=1 ELSE DC(I)=3
180      NEXT :CC=5
190      B$="e1f2e1":A$="": FOR I = 1
  TO 32 : A$=A$+B$ : NEXT
200      DRAW "bm0,41xa$;xa$;bm0,44xa
$;xa$;"
210      DRAW "bm0,67xa$;xa$;bm0,70xa
$;xa$;"
220      DRAW "bm0,93xa$;xa$;"
230      DRAW "bm0,96xa$;xa$;"
240      DRAW "bm0,119xa$;xa$;"
250      DRAW "bm0,122xa$;xa$;"
260      DRAW "bm0,145xa$;xa$;"
270      DRAW "bm0,148xa$;xa$;"
280      DRAW "bm0,171xa$;xa$;"
290      GOSUB 320
300      DRAW "bm115,192r10u2110"
310      GOTO 340
320      DRAW "bm115,14r10d10110u10f1
0b110e10b18u4r6d4":RETURN
330      DRAW"bm115,178r10d10110u10f1
0b110e10b18u4r6d4":RETURN
340      SPRITE ON : ON SPRITE GOSUB
530

```

Control

```

350      POKE V1VVV,10 : D=USR(D) : GO
SUB 440
360      GOSUB 380
370      GOTO 350

```

Ajuste dirección de cucarachas

```
380 GOSUB 420
390 I=INT(RND(1)*6+1) : IF DC(I)=
1 THEN DC(I)=3 ELSE DC(I)=1
400 IF RND(1)<.9-NV/10 THEN 410 E
LSE IF VPEEK(6913+4*CC)<VPEEK(6913) T
HEN DC(CC)=3 ELSE DC(CC)=1
410 RETURN
```

Movimiento de cucarachas

```
420 POKE V1VVV,9 : FOR I =1 TO 6
: POKE V2VVV,DC(I) : POKE V3VVV,I : D
=USR1(D) : NEXT
430 RETURN
```

Comprobación de movimiento

```
440 K=VPEEK(6912) : CC=INT((K-20)
/26)
450 IF K<72 THEN CC=6
460 IF K>180 THEN K1=11 ELSE IF K
<10 THEN K1=14 ELSE K1 =15
470 POKE V4VVV,K1
480 IF K>25 AND K<165 THEN RETURN
490 IF VPEEK(6913)<115 OR VPEEK(6
913)>125 THEN RETURN
500 IF K<=25 THEN IF CC=1 THEN RET
URN ELSE PLAY "124m160s8n60n60":CC=1
: DRAW "c14" : GOSUB 320 :GOSUB 330 :
RETURN
510 IF CC=1 THEN PLAY"s8m200140n2
0" : NV= NV+1 : GOSUB 570 : CC=0: DR
AW "c1" : GOSUB 330: GOSUB 320 : RET
URN
520 RETURN
```



Fin de juego

```
530     FOR I=1 TO 7 : PLAY "150m1300
s10n64r10n74": NEXT
540     SCREEN 1: PRINT "___LAS_CUCAR
ACHAS_COSMICAS_____ATACAN_DE_NUE
VO!"
550     PRINT:PRINT:PRINT:IF NV>1 THE
N PRINT "PERO_PEPE_EL_ALPINISTA_": P
RINT"HA_HECHO_";NV;"VIAJES A FOR
COMIDA"ELSE IF NV=1 THEN PRINT "Y_P0
BRE_PEPE_SOLO_HIZO_":PRINT "UN_VIAJE"
ELSE PRINT "Y_-PEPE_NO_TIENE_COMIDA!
"
560     IF INKEY$="" THEN END ELSE 56
0
570     A$="bm"+STR$(NV*4)+" ,1":DRAW "
xa$;c1d4": RETURN
```

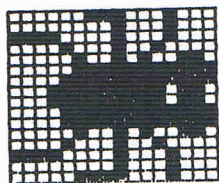
Datos de sprites

```
580     DATA 18,10,5,82,47,21,37,87,14,
23,39,75,18,2,2,14,144,161,74,148,104
,178,164,234,113,232,228,208,64,64,64
,112
590     DATA 3,0,240,8,4,7,15,31,31,15,
15,4,8,240,0,3,136,136,136,144,166,25
5,255,246,246,255,255,166,144,136,136
,136
600     DATA 68,34,17,17,17,63,127,255,
255,127,63,17,17,17,34,68,64,32,16,17
,17,250,252,252,252,252,250,17,17,16,
32,64
```

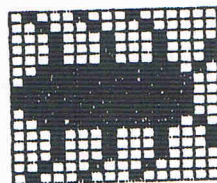
## Tabla de claves

10	= 904	220	= 1289	430	= 142
20	= 143	230	= 1292	440	= 2475
30	= 1637	240	= 1336	450	= 1155
40	= 1541	250	= 1330	460	= 3174
50	= 1712	260	= 1335	470	= 716
60	= 143	270	= 1338	480	= 1593
70	= 143	280	= 1334	490	= 2582
80	= 364	290	= 220	500	= 5141
90	= 2571	300	= 1399	510	= 4710
100	= 2998	310	= 236	520	= 142
110	= 690	320	= 3239	530	= 2579
120	= 3659	330	= 3298	540	= 3557
130	= 4661	340	= 929	550	= 11374
140	= 3199	350	= 1747	560	= 1312
150	= 2758	360	= 280	570	= 3297
160	= 7329	370	= 246	580	= 5266
170	= 3331	380	= 320	590	= 5251
180	= 584	390	= 4070	600	= 5326
190	= 3108	400	= 6146		
200	= 2309	410	= 142		
210	= 2316	420	= 3974	TOTAL	= 141392

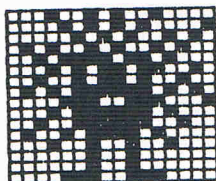
## Diseño de sprites



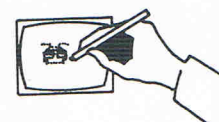
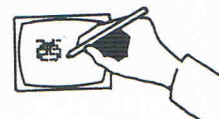
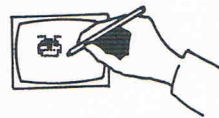
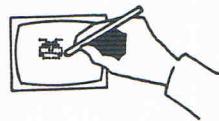
CUCARACHA



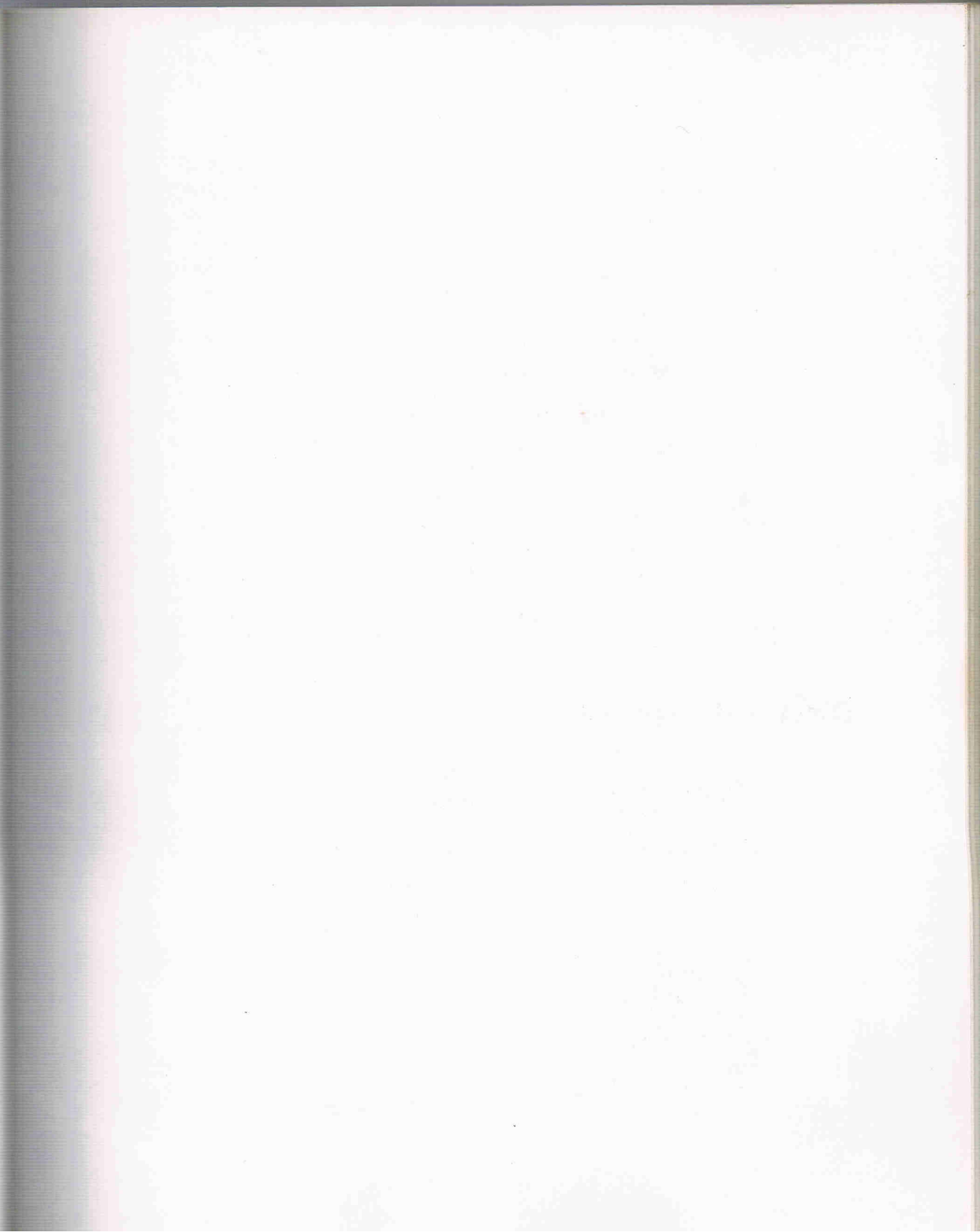
CUCARACHA REINA



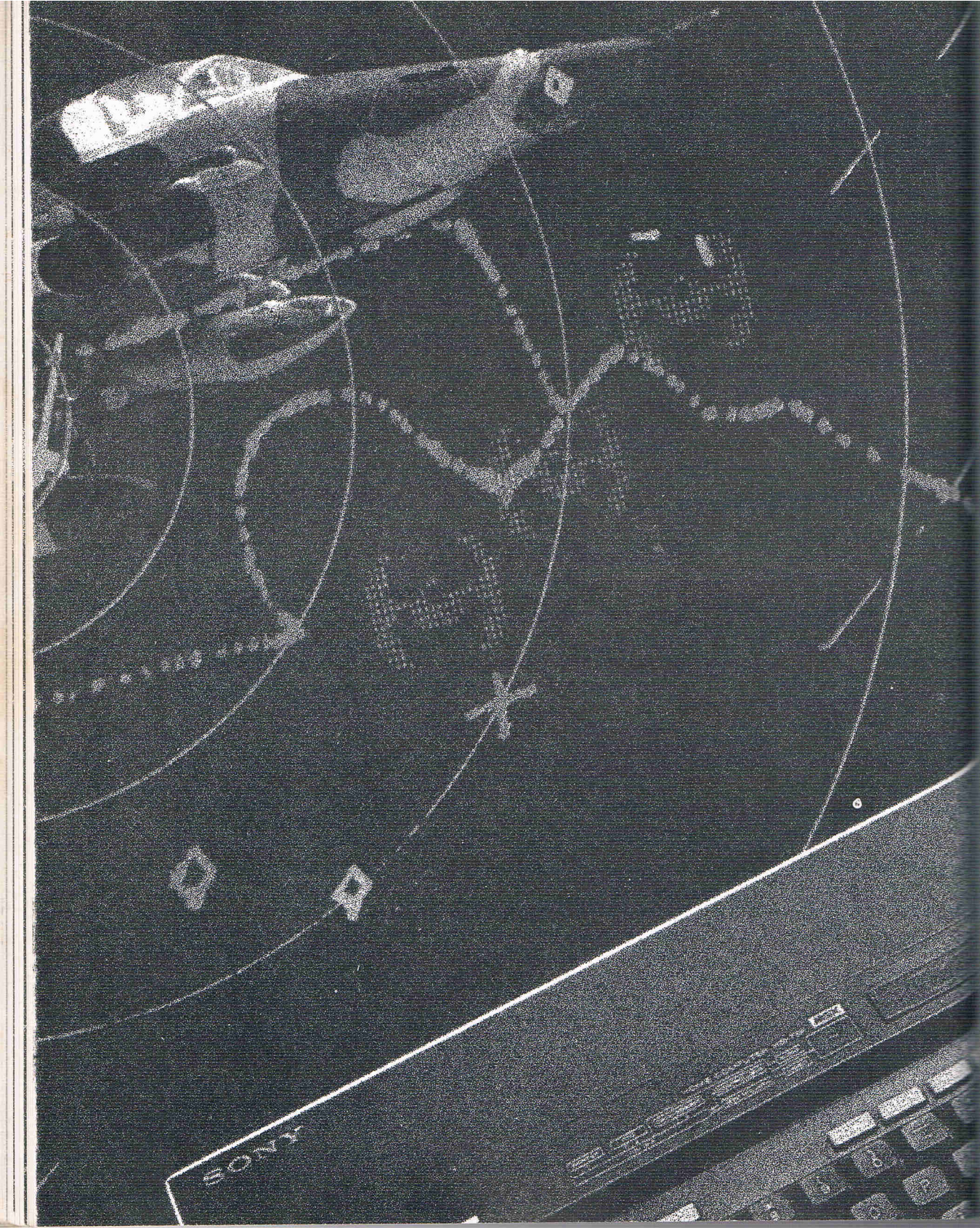
RIGOBERTO,  
«EL HAMBRIENTO»









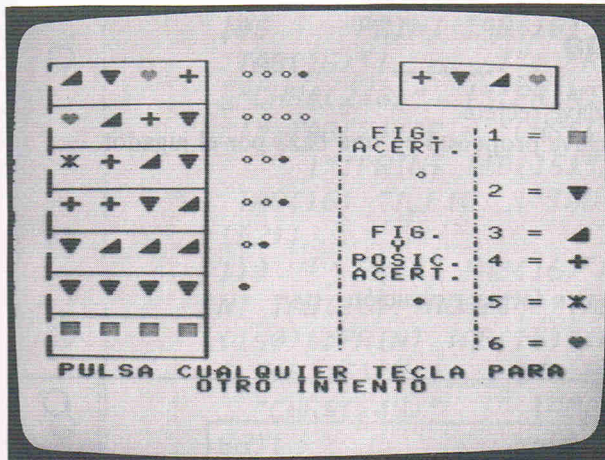


SONY



# Test de inteligencia

# 8



Tipo de juego:

**JUEGO MEMORISTICO**

El código secreto está formado por la combinación de cuatro figuras, elegidas de seis tipos diferentes. Adivina el código en menos de ocho intentos.

Un círculo significa que tienes una figura correcta, pero en una posición equivocada; un círculo coloreado te indica que tienes una figura en la posición correcta.

Las instrucciones van incluidas en el programa.

# SUGERENCIAS AL PROGRAMA

Si encuentras algo difícil el juego puedes tener más oportunidades, cambiando la rutina "Juego perdido" por otra que dé al jugador más facilidades o se rinda.

Por otra parte, puedes añadir otra figura, con el objetivo de aumentar el nivel de dificultad.

## Variables del programa

- CL(I) Código de caracteres de la clave secreta.
- CJ(I) Código de caracteres de la clave propuesta en una baza por el jugador.
- N1 Nivel (0-6).
- C Columna (0-3).
- M\$ Mensaje.
- FC Número correcto de figuras.
- FA Número de figuras acertadas en forma y posición.

### Inicialización

```
10      VPOKE 8209,79: VPOKE 8210,207
      : VPOKE 8212,111 : VPOKE 8214,159 :
FOR I= 136 TO 184 STEP 8 : FOR J=0 TO
  7 : READ Q : VPOKE 8*I+J,Q : NEXT :
NEXT
15      FOR I=1 TO 5 : PRINT : NEXT :
PRINT " _UTILIZA LAS TECLAS DE LA
'1' A LA '6' PARA PONER EN LA PANTA
LLA LAS 6 FIGURAS POSIBLES." :PRIN
T " LA TECLA DEL ' PUEDE USARSE PARA
CAMBIAR LA ULTIMA FIGURA"
20      PRINT: PRINT " _"CHR$(136)"
      "CHR$(144)" "CHR$(152)" "CHR$(16
0)" "CHR$(168)" "CHR$(176)
30      PRINT : PRINT : PRINT "PULSA
UNA TECLA PARA EMPEZAR"
35      IF INKEY$="" THEN D=RND(1) :G
OTO 35
```



Preparación pantalla

```
100      CLS : FOR I=1 TO 6 :PRINT : N
EXT : PRINT TAB(16) "!_FIG._!"TAB(24)
"1 = " CHR$(136);
105      PRINT TAB(16)"!ACERT.!" :PRINT
TAB(16)"!_!_" : PRINT TAB(16)"!_
_CHR$(248)" !":PRINT TAB(16)"!_!_"
_!" TAB(24)"2 = "CHR$(144) TAB(16)"!
!":PRINT TAB(16)"!_!_" :PRINT
TAB(16) "!_FIG._!"TAB(24)"3 = "CHR$(
152);
110      PRINT TAB(16) "!_!_Y_!" : PRI
NT TAB(16)"!POSIC.!"TAB(24)"4 = "CHR$(
160);:PRINT TAB(16)"!ACERT.!" : PRIN
T TAB(16)"!_!_" : PRINT TAB(16)"!_
_CHR$(249)" !" TAB(24)"5 = "CHR$(1
68);
115      PRINT TAB(16)"!_!_" : PRIN
T TAB(16) "!_!_" : PRINT TAB(16)"!
_!_"!TAB(24) "6 = "CHR$(176)CHR$(1
1);
140      VPOKE 6209,84:VPOKE 6241,69:V
POKE 6273,83:VPOKE 6305,84:VPOKE 6369
,68:VPOKE 6401,69:VPOKE 6465,73:VPOKE
6497,78:VPOKE 6529,84
141      VPOKE 6561,69:VPOKE 6593,76:V
POKE 6625,73:VPOKE 6657,71:VPOKE 6689
,69:VPOKE 6721,78:VPOKE 6753,67:VPOKE
6785,73:VPOKE 6817,65
145      FOR I=6179TO 6755 STEP 96 :
FOR J=I+97 TO I+103: VPOKE J,23: NEXT
: VPOKE I,20: VPOKE I+8,19: VPOKE I=
32,22: VPOKE I+64,22: VPOKE I+40,22:
VPOKE I+72,22: NEXT
150      VPOKE 6179,24: VPOKE 6187,25:
VPOKE 6851,26: VPOKE 6859,27: FOR I=
6180 TO 6186: VPOKE I,23: NEXT
155      FOR I= 6197 TO 6261 STEP32 :
```

```
FOR J= 0 TO 6 : VPOKE I+J,184: NEXT:
NEXT: FOR I=1 TO 4
160     K=INT(RND(1)*6): IF T(K)<>1
THEN CL(I)=K*8+136: T(K)=1 ELSE 160
165     NEXT: M$="PREPARADO. CUANDO Q
UIERAS": GOSUB 1000
```

Editor

```
200     X$=INKEY$: IF X$<>CHR$(127)A
ND(X$<"1" OR X$>"6") THEN 200
205     IF X$=CHR$(127) THEN IF C=0 T
HEN 200 ELSE C=C-1: VPOKE 6788+2*C-96
*L,32:GOTO 200
210     CJ(C+1)=VAL(X$)*8+128: VPOKE
6788+2*C-96*L,CJ(C+1):C=C+1
215     IF C=4 THEN M$="COMPROBANDO T
U RESPUESTA": GOSUB 1000: FOR I=1 TO
1500: NEXT: GOTO 300
290     GOTO 200
```

Comprobación jugada

```
300     FC=0:FA=0:FOR I=1 TO 4
305     IF CJ(I)=CL(I) THENFA=FA+1
310     NEXT: FOR I=1 TO 4
315     IF CL(I)=CJ(1) OR CL(I)=CJ(2
) OR CL(I)=CJ(3) OR CL(I)=CJ(4) THEN
FC=FC+1
320     NEXT: FC=FC-FA
```

Listar resultados

```
400     IF FC=0 THEN 415
410     FOR I=1 TO FC: VPOKE 6796-96*
L+I,248: NEXT
```



```

415   IF FA=0 THEN 425
420   FOR I=1 TO FA: VPOKE 6796+FC-
96*L+I,249: NEXT
425   C=0:L=L+1:M$="
      I": GOSUB 1000: IF FA=4 THEN

900
430   IF L=7 THEN 500
435   GOTO 200

```

Juego perdido

```

500   FOR I=6197 TO 6261 STEP 32: F
OR J=0 TO 6: VPOKE I+J,32: NEXT: NEXT
: FOR J=6198 TO 6204: VPOKE J,23: NEX
T: VPOKE 6197,24: VPOKE 6205,25: VPOKE
6229,22: VPOKE 6261,22: VPOKE 6237,2
2: VPOKE 6269,22: VPOKE 6293,26: VPOK
E 6301,27: FOR I=6294 TO 6300
505   VPOKE I,23:NEXT
510   FOR I=1 TO 4: PLAY"n20s8m160n
30": VPOKE 6228+2*I,CL(I):NEXT
515   FOR I=1 TO 2500: NEXT: M$="PU
LSA CUALQUIER TECLA PARA OTRO INTEN
TO": GOSUB 1000
520   IF INKEY$="" THEN 520 ELSE RU
N

```

Juego ganado

```

900   M$="----SUPER!!!!": GOSUB 100
0: PLAY"s14m2000110n30n32n34n36n38n40
n42n44m1000n46n48n49n50n51m700n52n53n
54n55m500m500n56n58n60"
910   FOR I=1 TO 3000: NEXT: M$="-L
O HAS LOGRADO EN"+STR$(L)+"INTENTOS!"
: GOSUB 1000: BEEP
920   FOR I=1 TO 3000: NEXT:M$="PUL

```

```

SA UNA TECLA PARA JUGAR OTRA VEZ": 60
SUB 1000 : BEEP
930 IF INKEY$="" THEN 930 ELSE RUN

```

Listas mensaje

```

1000 FOR I=1 TO 23: PRINT:NEXT:PRI
NT M$;CHR$(11);
1010 RETURN

```

Datos de sprites pequeños

```

10000 DATA 255,255,255,255,255,255,
255,255
10002 DATA 255,255,126,126,60,60,24
,24
10004 DATA 1,3,7,15,31,63,127,255
10006 DATA 24,24,24,255,255,24,24,2
4
10008 DATA 146,84,56,16,56,84,146,0
10010 DATA 108,254,254,254,124,56,1
6,0
10012 DATA 255,255,255,255,255,255,
255,255

```

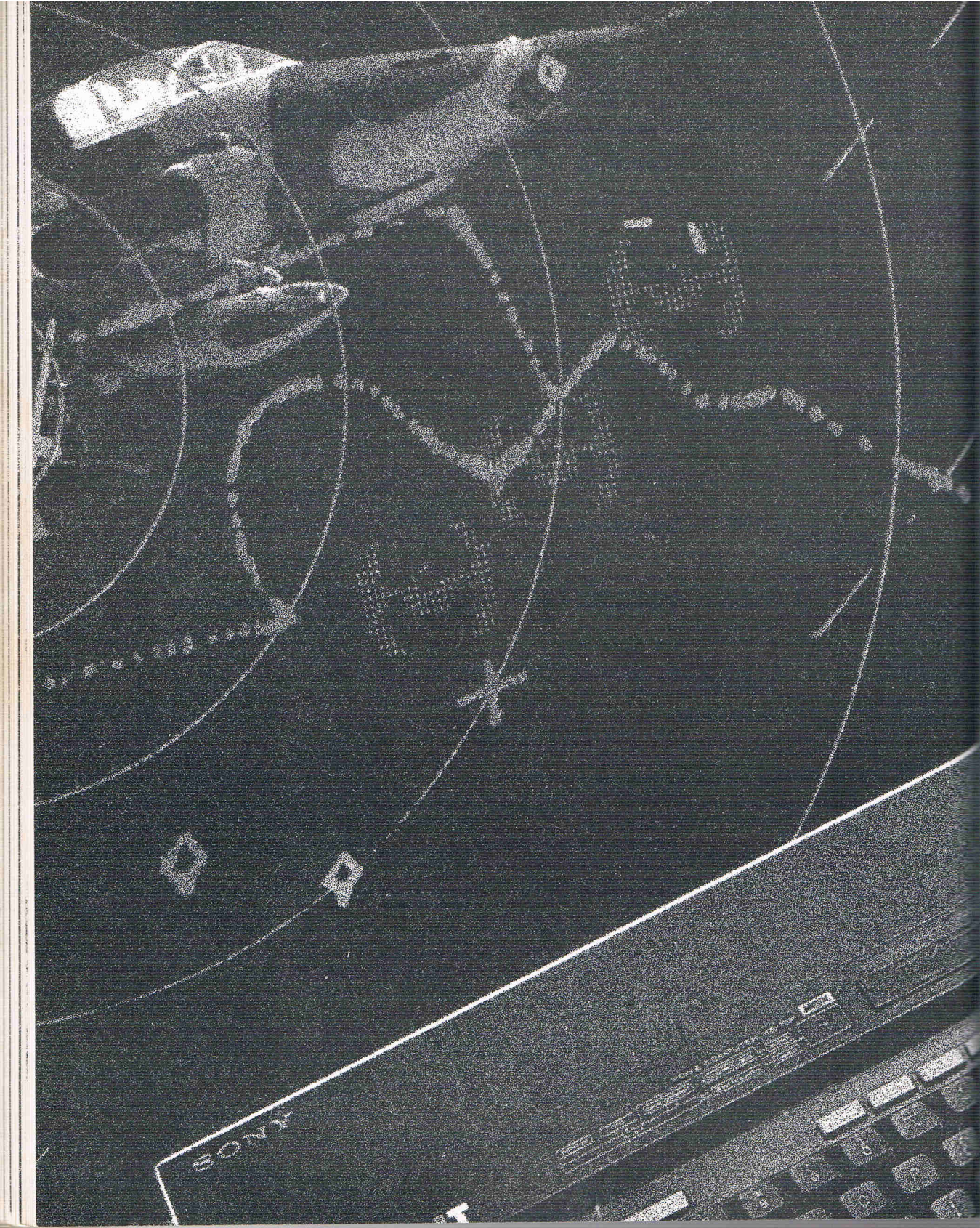
## Tabla de claves

10	= 5707	140	= 4910	210	= 5013
15	= 11226	141	= 4966	215	= 4736
20	= 5274	145	= 7515	290	= 351
30	= 2549	150	= 3491	300	= 1598
35	= 1941	155	= 3947	305	= 1956
100	= 3498	160	= 4909	310	= 887
105	= 9059	165	= 2817	315	= 4980
110	= 7879	200	= 3997	320	= 1079
115	= 4196	205	= 5246	400	= 924



410	=	2746	515	=	4843	10002	=	1466
415	=	932	520	=	1281	10004	=	1212
420	=	3123	900	=	8495	10006	=	1364
425	=	3684	910	=	5346	10008	=	1331
430	=	955	920	=	4659	10010	=	1467
435	=	351	930	=	1436	10012	=	1688
500	=	9637	1000	=	2053			
505	=	542	1010	=	142			
510	=	3363	10000	=	1688	TOTAL	=	178455

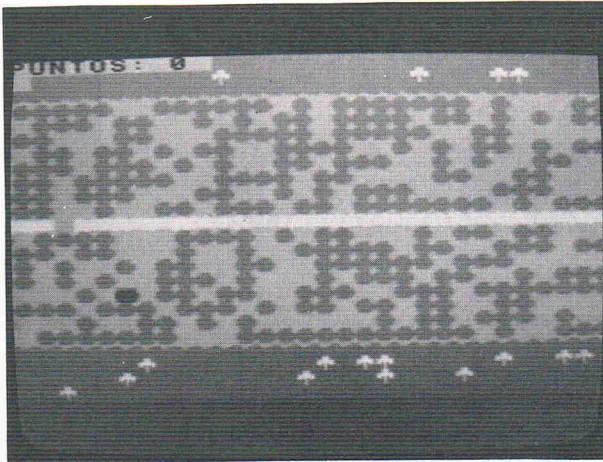






# El saltarín del río

# 9



## ¡ATENCIÓN!

Este programa utiliza la subrutina SOPORTE en lenguaje máquina. Véase apéndice C.

**Tipo de juego:**

**JUEGO DE SIMULACION**

La vida es dura para la pequeña rana que quiere cruzar el río. Cruza el río todas las veces que puedas; la puntuación depende del número total de movimientos que emplees para cruzarlo cada vez.

Se emplea una subrutina en código máquina para mover la pantalla.

# SUGERENCIAS AL PROGRAMA

La corriente del río se controla por el valor 20 en la línea 330; para acelerarla, cambia el 20 por 10. La densidad de los nenúfares está controlada por los valores 0.65 y 0.60 en las líneas 280 y 300, respectivamente.

¿Por qué no tener otros objetos flotantes diferentes en una de las mitades del río? Puedes poner hasta un barquito (como un sprite).

## Variables del programa

D	Direcciones de la rana (0 ó 2).
PR	Posición de la rana.
NM	Número de movimiento.
PT	Puntuación.
FJ, CJ	Fila y columna del jugador.
RS	Objeto sobre el que está la rana
LT	Controlador de llegada al tablón.

Inicialización

```
10      REM      ** ATENCION **
20      REM
30      REM ESTE PROGRAMA NECESITA
40      REM LA SUBROUTINA SOPORTE
50      REM Vease el Apendice C
60      REM
70      REM
80      A=PEEK(64586!)+256*PEEK(6458
7!)
90      V1=A+9:V2=A+8:V3=A+7:V4=A+6
100     W1=A+249:W2=A+248:W3=A+247:
W4=A+246
110     COLOR 4,7,1: SCREEN 1,2: KEY
OFF
120     B=A+10
130     I=A+250
140     READ A$
150     IF A$="R" THEN READ C:C=C+B:
C1 = INT(C/256):C2=256*(C/256-C1):PO
```



```

KE I,C2:I=I+1:POKE I,C1 :GOTO 170
160     POKE I, VAL(A$)
170     IF I<A+250+51 THEN I=I+1:GO
TO 140
180     FOR I=1072 TO 1079: READ Q:
VPOKE I,Q: NEXT: FOR I=1104 TO 1135:
READ Q: VPOKE I,Q: NEXT: FOR I=1152 T
O 1175: READ Q: VPOKE I,Q: NEXT: FOR
I=1216 TO 1223: READ Q: VPOKE I,Q: NE
XT
190     FOR I=1 TO 11: PRINT: NEXT:
PRINT "      _PULSA_UNA_TECLA"
200     PLAY"124m160s8n67n67":PRINT
CHR$(11)TAB(6)"SALTARIN-DEL-RIO": FOR
I=1 TO 300: NEXT
210     VPOKE 6208+INT(RND(1)*60),13
4: IF INKEY$="" THEN FOR I=6148 TO 61
70: VPOKEI,32: NEXT: FOR I=1 TO 300:
NEXT: GOTO 200
220     DEFUSR=A+250!: SA=6240: FA=6
783: POKE W4WWW,SA MOD 256: POKE W3WW
W,INT(SA/256): POKE W2WWW,FA MOD 256:
POKE W1WWW,INT(FA/256)

```

Preparación pantalla

```

230     CLS: FOR I=6155 TO 6207: VPOK
E I,141: NEXT: VPOKE 6144,141: VPOKE
6145,141
240     FOR I=6208 TO 6239: VPOKE I,1
40: NEXT: FOR I=6784 TO 6815: VPOKE I
,139: NEXT
250     FOR I=6816 TO 6911: VPOKE I,1
41: NEXT
260     FOR I=6496 TO 6525: J =INT(RN
D(1)*3+144): VPOKE I,J: NEXT
270     GOSUB 600
280     FOR I=6240 TO 6464 STEP 32: F
OR J=0 TO 31: IF RND(1)<.65 THEN VPOK
E I+J,138

```

```
290     NEXT: NEXT
300     FOR I=6528 TO 6752 STEP 32: F
OR J=0 TO 31: IF RND(1)<.6 THEN VPOKE
    I+J,138
310 NEXT: NEXT
320     RS=139: PR=6799: VPOKEPR,134:
CJ=15
330     ON INTERVAL=20 GOSUB 550: INT
ERVAL ON
340     FOR I=1 TO 5: VPOKE 6176+INT(
RND(1)*31),152: NEXT: FOR I=1 TO 12:
VPOKE 6816+INT(RND(1)*95),152: NEXT
350     VPOKE 8208,19: VPOKE 8209,55:
VPOKE 8210,231: VPOKE 8211,179
```

Control

```
360     X$=INKEY$: IF X$<>" " THEN GOS
UB 390
370     IF RND(1)<.1 THEN GOSUB 460
380     GOTO 360
```

Leer del teclado

```
390     FJ=INT((PR-6144)/32): CJ=PR-3
2*FJ-6144
400     K=ASC(X$)-27: IF K>0 AND K<5
THEN PLAY"124m160s8n67":ON K GOTO 410
,420,430,440 ELSE RETURN
410     IF CJ=31 THEN RETURN ELSE INT
ERVAL OFF: VPOKE PR,RS:PR=PR+1: GOTO
450
420     IF CJ=0 THEN RETURN ELSE INTE
RVAL OFF:VPOKE PR,RS: PR=PR-1: GOTO 4
50
430     IF FJ<=2 THEN RETURN ELSE INT
ERVAL OFF:VPOKE PR,RS: PR=PR-32: GOTO
450
```



```
440 IF FJ>=20 THEN RETURN ELSE IN
TERVAL OFF:VPOKE PR,RS: PR=PR+32
450 NM=NM+1: RS=VPEEK(PR): VPOKE
PR,134:INTERVAL ON: IF RS=32 THEN 570
ELSE 490
```

Cambiar pantalla

```
460 IF RND(1)<.5 THEN I=6240 ELSE
I=6528
470 VPOKE I+32*(INT(RND(1)*8)),13
8: VPOKE I+32*(INT(RND(1)*8)),32
480 RETURN
```

Comprobación de posición

```
490 IF D=2 THEN 530
500 IF FJ=2 THEN PT=PT+200-NM: D=
2: LT=0: GOSUB 600: NM=0: RETURN
510 IF FJ=11 AND LT=0 THEN LT=1:
PT=PT+50: GOSUB 600
520 RETURN
530 IF FJ=20 THEN D=0: PT=PT+200-
NM: LT=0: GOSUB 600: NM=0: RETURN
540 GOTO 510
```

Movimiento de pantalla

```
550 D=USR(D): IF PR<=6783 AND PR>
=6240 THEN PR=PR-1 ELSE RETURN
560 IF PR/32=INT(PR/32) THEN 570
ELSE RETURN
```

Fin de juego

```
570 CLS: FOR I=1 TO 50: PLAY"164s
14m50n50": VPOKE 6176+INT(RND(1)*700)
```

```
,134: NEXT
580 PRINT CHR$(11);"PUNTOS:";PT
590 END
```

Puntuación

```
600 PLAY"12s10m700n44": PRINT CHR$(
(11);"PUNTOS:";PT
610 RETURN
```

Datos del código máquina

```
620 DATA 6,31,42,R,236,43,35,205,R,18
7,120,254,31,202,R,265,4,43,205,R,204
,35,195,R,277,6,0,17,31,0,25,205,R,20
4,183,237,82,237,91,R,238,123,189,194
,R,246,122,188,194,R,246,201
```

Datos de sprites pequeños

```
630 DATA 0,36,90,255,255,189,66,60
640 DATA 60,124,254,255,255,255,126
,124
650 DATA 0,131,207,255,255,255,255,
255
660 DATA 255,255,255,255,255,247,97
,0
670 DATA 255,255,255,255,255,255,25
5,255
680 DATA 32,255,255,255,255,255,255
,11
690 DATA 8,255,255,255,255,255,255,
1
700 DATA 97,251,255,255,255,255,255
,204
710 DATA 56,56,254,254,214,16,16,16
```



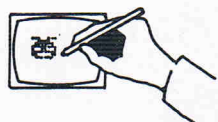
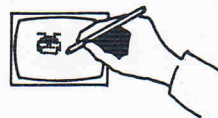
## Tabla de claves

10	= 904	260	= 3615	510	= 3079
20	= 143	270	= 245	520	= 142
30	= 1637	280	= 4062	530	= 4023
40	= 1541	290	= 320	540	= 406
50	= 1712	300	= 4123	550	= 4131
60	= 143	310	= 320	560	= 2342
70	= 143	320	= 2293	570	= 4480
80	= 2571	330	= 2709	580	= 1554
90	= 2998	340	= 6081	590	= 129
100	= 3954	350	= 1868	600	= 2717
110	= 1173	360	= 1974	610	= 142
120	= 636	370	= 1556	620	= 8621
130	= 883	380	= 256	630	= 1382
140	= 236	390	= 3186	640	= 1620
150	= 6442	400	= 5304	650	= 1570
160	= 854	410	= 4232	660	= 1537
170	= 2541	420	= 4204	670	= 1688
180	= 7698	430	= 4446	680	= 1575
190	= 2600	440	= 4055	690	= 1481
200	= 4546	450	= 5056	700	= 1634
210	= 5759	460	= 2432	710	= 1424
220	= 7476	470	= 4470		
230	= 2789	480	= 142		
240	= 3467	490	= 717	TOTAL	= 182098
250	= 1900	500	= 4009		

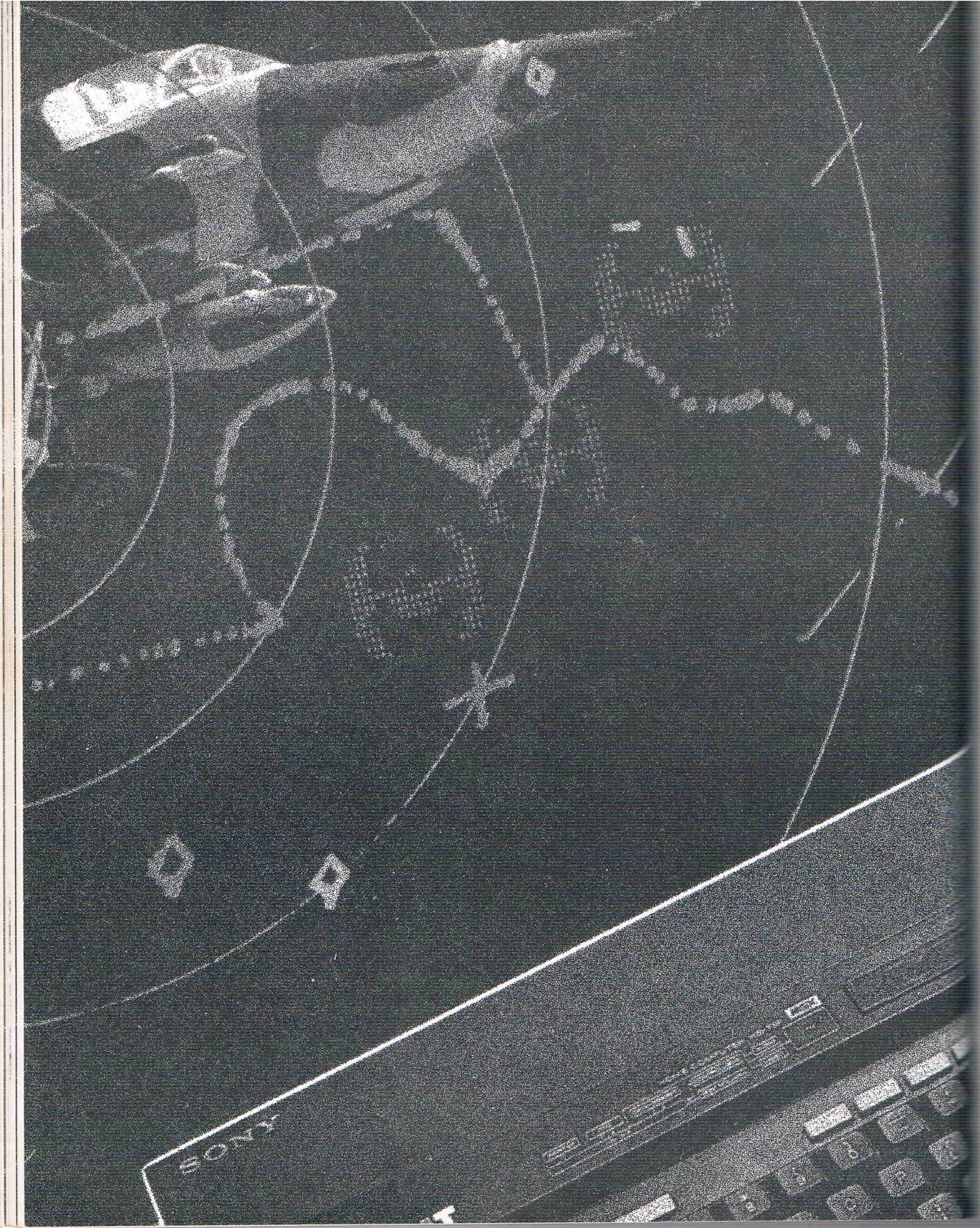
## Diseño de los sprites



RANA  
(CHARACTER)



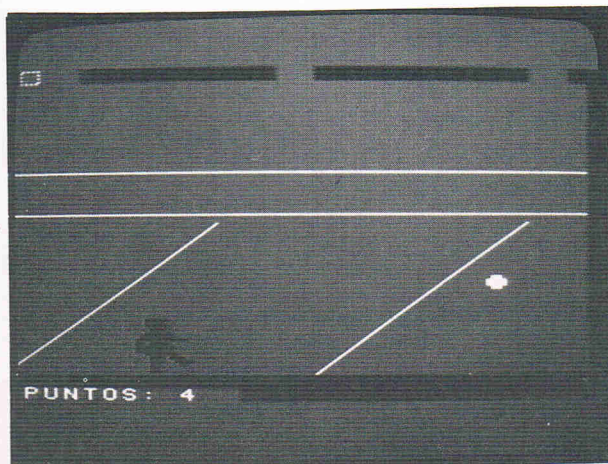






# Prueba de fútbol

# 10



## ¡ATENCIÓN!

Este programa utiliza la subrutina  
SOPORTE en lenguaje máquina.  
Véase apéndice C.

**Tipo de juego:**

**JUEGO DE TIRO AL BLANCO**

¿Cuánto tiempo puedes mantener el balón en el aire?

Intenta que el balón dé en los agujeros del techo; son puntos extra. Puedes dar cabezazos o patadas; para las patadas usa la barra espaciadora.

# SUGERENCIAS AL PROGRAMA

Es fácil cambiar la velocidad del balón —mira VB en la línea 240—. Te sugiero que pongas algunos obstáculos en la pared para que cuando el balón choque con ellos rebote con un ángulo aleatorio.

Revisa las líneas 510 a 560 para poder golpear el balón con la rodilla o poder dar otros ángulos al golpearla.

## Variables del programa

- DR Dirección del balón (0-7).
- PT Puntuación.
- X, Y Coordenadas del balón.
- JX Coordenada X del jugador.
- VB Velocidad del balón.
- GA Golpe en el agujero del techo (1-3).
- GJ Control de golpe del balón sobre el jugador.

### Inicialización

```
10      REM   ** ATENCION **
20      REM
30      REM ESTE PROGRAMA NECESITA
40      REM LA SUBROUTINA SOPORTE
50      REM Vease el Apendice C
60      REM
70      REM
80      A=PEEK(64586!)+256*PEEK(6458
7!)
90      V1=A+9:V2=A+8:V3=A+7:V4=A+6
100     SCREEN 1,3:KEY OFF:COLOR 15,
3,3:PRINT "          FUTBOL"
110     FOR I=1 TO 32:READ Q:A#=A#+C
HR$(Q):NEXT:SPRITE$(0)=A#:A#=""
120     FOR I=1 TO 32:READ Q:A#=A#+C
HR$(Q):NEXT:SPRITE$(1)=A#:A#=""
130     FOR I=1 TO 32:READ Q:A#=A#+C
HR$(Q):NEXT:SPRITE$(2)=A#:A#=""
140     FOR I=1 TO 8:READ Q:A#=A#+CH
```



```

R$(Q):NEXT:SPRITE$(3)=A$
150     DEFUSR=A+10:DEFUSR1=A+128:PO
KE V4VVV,10
160     FOR I=1 TO 20:PRINT:NEXT:PRI
NT"_____PULSA_UNA_TECLA":POKE V3VV
V,3:POKE V2VVV,3:POKE V1VVV,1:PUT SPR
ITE 0,(120,100),1:PUT SPRITE 3,(0,20)
170     FOR I=1088 TO 1103:READ Q:PO
KE I,Q:NEXT
180     D=USR1(D):IF VPEEK(6925)=255
THEN POKE V2VVV,1 ELSE IF VPEEK(6925
)=1 THEN POKE V2VVV,3
190     IF INKEY$="" THEN 180

```

Preparación pantalla

```

200     CLS:K=134:VPOKE 1072,170:VPD
KE 1073,127:VPOKE 1074,254:VPOKE 1075
,127:VPOKE 1076,254:VPOKE 1077,127:VP
OKE 1078,254:VPOKE 1079,85:FOR I=6144
TO 6175:VPOKE I,K:NEXT:VPOKE 6147,32
:VPOKE 6148,32:VPOKE 6159,32:VPOKE 61
60,32:VPOKE 6172,32:VPOKE 6173,32
210     FOR I=6176 TO 6880 STEP 32:V
POKE I,K:VPOKE I+31,K:NEXT:FOR I=6849
TO 6879:VPOKE I,K:NEXT
220     FOR I=6893 TO 6911:VPOKE I,K
:NEXT
230     VPOKE 8208,195:GOSUB 620:PUT
SPRITE 0,(120,142):PUT SPRITE 3,(11,
11)
240     DR=6:VB=7:STRIG(0) ON:ON STR
IG GOSUB 570
250     K=6817:FOR I=1 TO 11:VPOKE K
,29:VPOKE K+16,29:K=K-31:NEXT:FOR I=6
465 TO 6494:VPOKE I,23:VPOKE I-96,23:
NEXT
260     SPRITE ON :ON SPRITE GOSUB 5
10

```

Control

```
270     POKE V1VVV,6:D=USR(D)
280     IF PEEK(V3VVV)=0 THEN POKE 69
14,0
290     GOSUB 320
300     GOSUB 350
310     GOTO 270
```

Movimiento del balón

```
320     IF DR>3 THEN SPRITE OFF:GOTO
330:ELSE POKE V1VVV,VB:POKE V3VVV,3:P
OKE V2VVV,DR:D=USR1(D):GOTO 340
330     POKE V3VVV,3:POKE V1VVV,VB-1:
K1=DR-4:K2=(DR-3) MOD 4:POKE V2VVV,K1
:SPRITE ON :D=USR1(D):POKE V2VVV,K2:D
=USR1(D)
340     X=VPEEK(6925): Y=VPEEK(6924):
RETURN
```

Comprobación de posición

```
350     IF Y<8 OR Y>192 THEN GJ=0:PLA
Y "19s14m1900n30":GOTO 400
360     IF X<8 OR X>248 THEN PLAY "s8
m20019n20":GOTO 450
370     IF X>237 THEN GJ=0:PLAY "m300
0s819n35":GOTO 480
380     IF Y>168 THEN 590
390     RETURN
```

Choque con el techo

```
400     IF X>19 AND X<35 THEN GA=1:GO
TO 430 ELSE IF X>115 AND X<131 THEN G
```



```

A=2:GOTO 430 ELSE IF X>219 AND X<235
THEN GA=3:GOTO 430
410     IF DR=7 THEN DR=6:K=2 ELSE IF
DR=0 THEN DR=2:K=-2 ELSE IF DR=4 THE
N DR=5:K=-2 ELSE RETURN
420     VPOKE 6925,VPEEK(6925)+K:RETU
RN
430     PLAY "m1000814n33":FOR I=1 TO
1000:NEXT:IF GA=1 OR GA=3 THEN PT=PT
+50 ELSE PT=PT+100
440     GOSUB 620:PUT SPRITE 3,(9,11)
:DR=6:RETURN

```

Choque con la pared izquierda

```

450     IF DR=4 THEN DR=7:K=-2 ELSE I
F DR=1 THEN DR=3:K=-2 ELSE DR=6:K=-2
460     VPOKE 6924,VPEEK(6924)+K
470     RETURN

```

Choque con la pared derecha

```

480     IF DR=7 THEN DR=4:K=-2 ELSE I
F DR=6 THEN DR=5:K=2 ELSE DR=1:K=-2
490     VPOKE 6924,VPEEK(6924)+K
500     RETURN

```

Cabezazo/patada a la pelota

```

510     IF GJ=1 THEN RETURN ELSE SPRI
TE OFF:PT=PT+1:GJ=1:GOSUB 620:JX=VPEE
K(6913):IF Y>160 THEN 540
520     IF DR=2 THEN IF RND(1)<.5 THE
N DR=4 ELSE DR=7 ELSE IF DR=6 THEN DR
=0 ELSE IF DR=5 THEN DR=0 ELSE IF DR=
1 THEN DR=7 ELSE IF DR=3 THEN DR=4

```

```
530     GOTO 560
540     IF VPEEK(6914)=8 THEN IF X<17
0 THEN DR=4:GOTO 560 ELSE DR=1:GOTO 5
60
550     IF VPEEK(6914)=4 THEN IF X>16
2 THEN DR=7:GOTO 560 ELSE DR=3:GOTO 5
60
560     SPRITE ON:RETURN
```

Intento de patada

```
570     JX=VPEEK(6913):IF JX+4<X THEN
VPOKE 6914,4 ELSE VPOKE 6914,8
580     RETURN
```

Fin de juego

```
590     STRIG(0) OFF:SPRITE OFF:CLS:P
RINT "_____FIN_DE_JUEGO":PRINT:PR
INT:PRINT
600     PRINT "TUS_PUNTOS_:";PT
610     IF INKEY$<>" " THEN END ELSE 6
10
```

Puntuación

```
620     FOR I=1 TO 23:PRINT:NEXT:PRINT
"PUNTOS:";PT;CHR$(11);
630     RETURN
```

Datos de sprites

```
640     DATA 3,5,7,3,15,31,27,19,19,23,
6,6,6,6,6,14,128,64,192,128,224,240,1
```



76, 144, 144, 208, 192, 192, 192, 192, 192, 224

650 DATA 3, 2, 3, 1, 3, 7, 13, 9, 11, 11, 3, 1, 1, 1, 1, 192, 128, 192, 128, 224, 240, 220, 192, 224, 224, 240, 152, 142, 4, 0, 128

660 DATA 1, 0, 1, 0, 7, 63, 3, 1, 1, 3, 7, 12, 56, 16, 0, 0, 224, 160, 224, 192, 240, 216, 204, 216, 192, 224, 224, 224, 192, 64, 64, 192

670 DATA 48, 120, 120, 48, 0, 0, 0, 0

Datos de sprites pequeños

680 DATA 1, 2, 4, 8, 16, 32, 64, 128

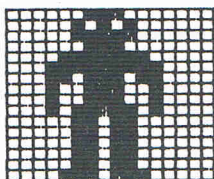
690 DATA 0, 0, 0, 0, 0, 0, 0, 255

## Tabla de claves

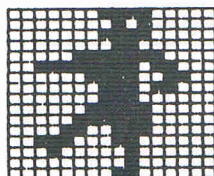
10	= 904	200	= 9052	390	= 142
20	= 143	210	= 4474	400	= 7387
30	= 1637	220	= 1896	410	= 5987
40	= 1541	230	= 2410	420	= 1382
50	= 1712	240	= 2448	430	= 5643
60	= 143	250	= 5670	440	= 1612
70	= 143	260	= 1164	450	= 5102
80	= 2571	270	= 1347	460	= 1180
90	= 2998	280	= 1765	470	= 142
100	= 2264	290	= 220	480	= 4862
110	= 3659	300	= 250	490	= 1180
120	= 3660	310	= 166	500	= 142
130	= 3661	320	= 5015	510	= 5185
140	= 3206	330	= 7148	520	= 8645
150	= 2753	340	= 2024	530	= 201
160	= 6286	350	= 3467	540	= 3590
170	= 1737	360	= 3010	550	= 3581
180	= 4782	370	= 2943	560	= 548
190	= 1094	380	= 961	570	= 2975

580	=	142	630	=	142	680	=	1112
590	=	3428	640	=	5121	690	=	932
600	=	1299	650	=	4692			
610	=	1601	660	=	4863			
620	=	2778	670	=	1142	TOTAL	=	187062

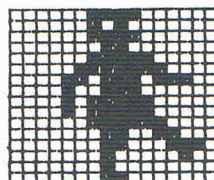
## Diseño de sprites



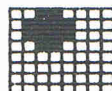
HOMBRE: DE PIE



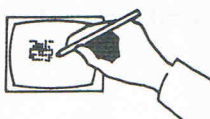
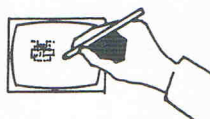
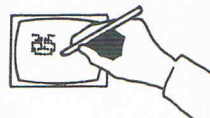
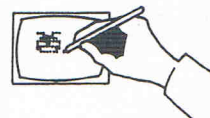
HOMBRE: PATADA  
A LA IZQUIERDA



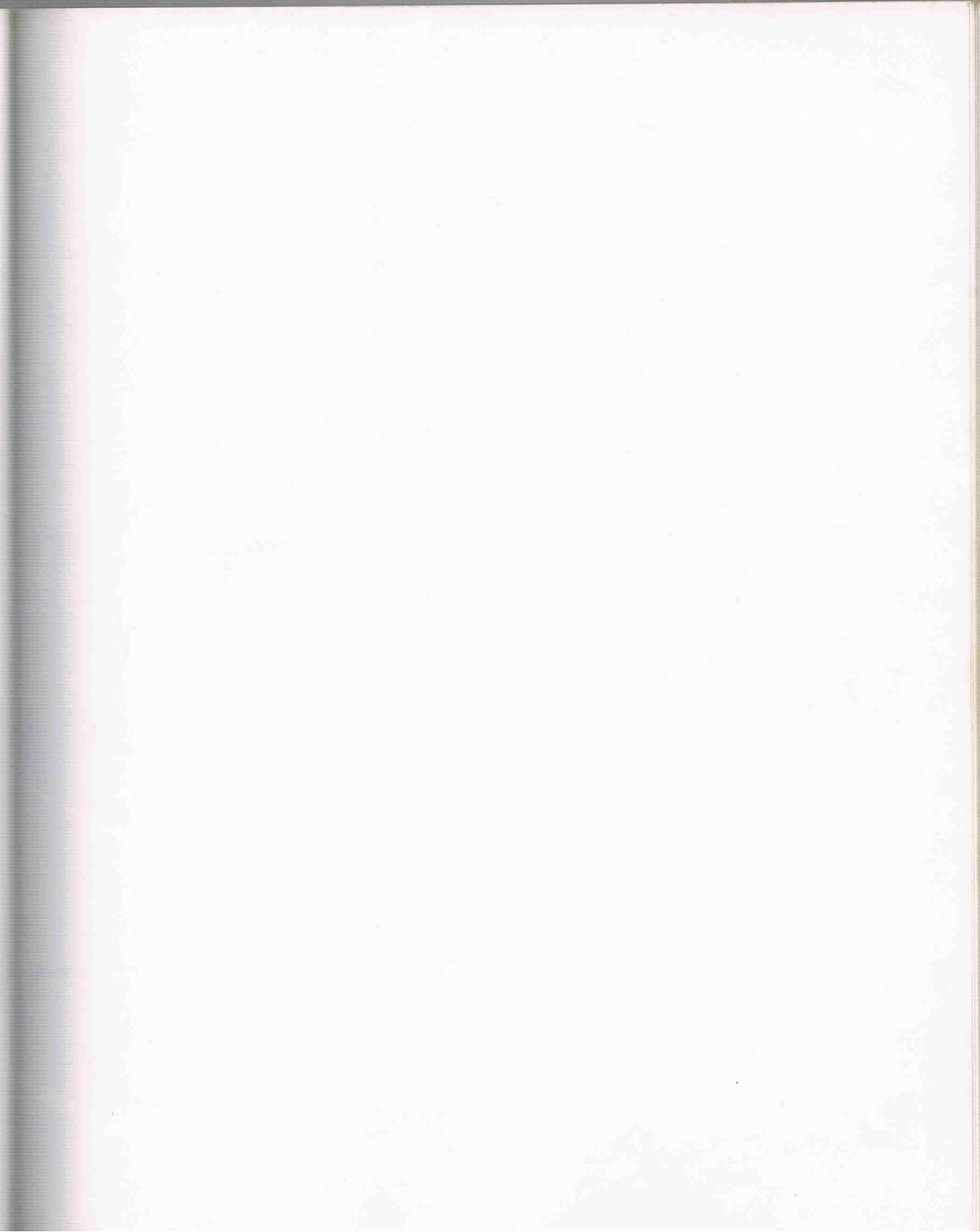
HOMBRE: PATADA  
A LA DERECHA



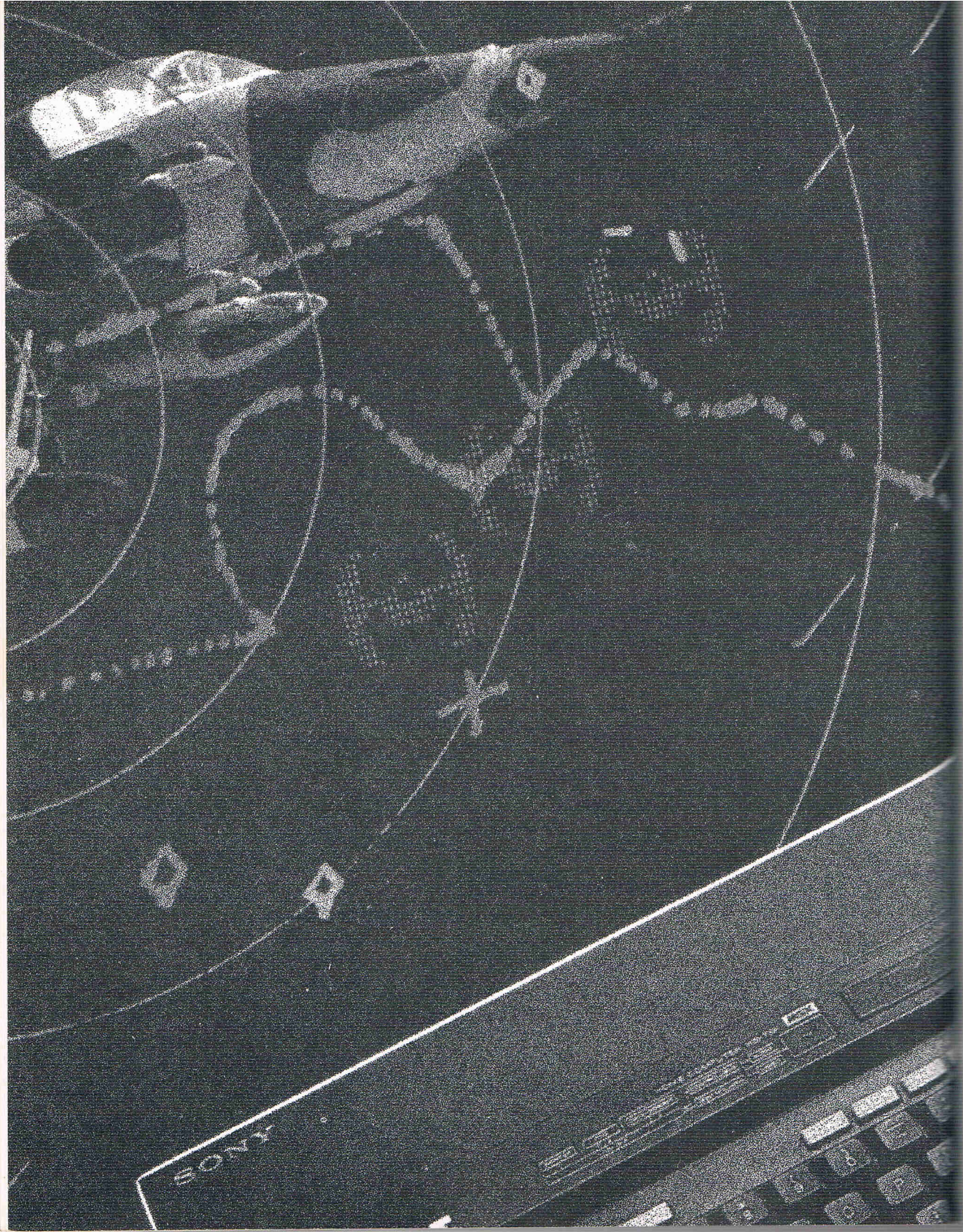
BALON







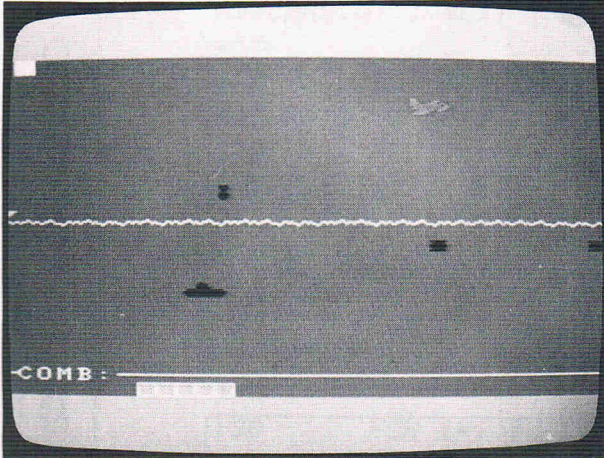






# Harrier

# 11



## ¡ATENCIÓN!

Este programa utiliza la subrutina SOPORTE en lenguaje máquina. Véase apéndice C.

**Tipo de juego:**

**JUEGO DE TIRO AL BLANCO**

**Misión:** Lanzar cargas de profundidad desde tu avión hacia el submarino sin que explote ninguna mina. Aterriza en el portaaviones para reponer combustible. El juego termina cuando se acabe el combustible del portaaviones.

**Control:**

Cursor ARRIBA	Sube o despegas.
Cursor ABAJO	Baja.
Cursor IZQUIERDA	Aterriza.
Cursor DERECHA	Acelera.
Barra espaciadora	Lanza cargas de profundidad.

# SUGERENCIAS AL PROGRAMA

Si encuentras difícil aterrizar, en la línea 510 tienes los requisitos para un buen aterrizaje.

Añadiendo más minas se hará más difícil lanzar las cargas de profundidad sin destruirte a ti mismo.

## Variables del programa

- CP Control de lanzamiento de cargas de profundidad.
- CR Combustible que resta en tu avión.
- TC Combustible total que queda.
- AT Control de aterrizaje.
- PA Control de parada.
- RE Control de recarga de combustible.
- CH Número de sprites que han chocado con el portaaviones.
- SD Número de submarinos destruidos.
- EM Explosión de una mina.

### Inicialización

```
5 BEEP
10 REM ** ATENCION **
20 REM
30 REM ESTE PROGRAMA NECESITA
40 REM LA SUBROUTINA SOPORTE
50 REM Vease el Apendice C
60 REM
70 REM
80 A=PEEK(64586!)+256*PEEK(6458
7!)
90 V1=A+9:V2=A+8:V3=A+7:V4=A+6
100 COLOR 15,4,7 : SCREEN 1,2 :
KEY OFF
110 FOR I=1 TO 32 : READ Q : A$=
A$+CHR$(Q) : NEXT : SPRITE$(0)=A$ : P
UT SPRITE 0,(120,35) : A$=""
120 FOR I=1 TO 32 : READ Q : A$=
A$+CHR$(Q) : NEXT : SPRITE$(1)=A$ : A
$=""
```



```

130     FOR I=1 TO 8 : READ Q : A#=A
      $+CHR$(Q) : NEXT : SPRITE$(2)=A$ : SP
RITE$(3)=A$ : A$=""
140     FOR I=1 TO 32 : READ Q : A#=
A#+CHR$(Q) : NEXT : SPRITE$(4)=A$ : A
$=""
150     FOR I=1 TO 32 : READ Q : A#=
A#+CHR$(Q) : NEXT : SPRITE$(5)=A$ : A$
=""
160     FOR I=1 TO 8 : READ Q : A#=A
$+CHR$(Q) : NEXT : SPRITE$(6)=A$
170     FOR I=1088 TO 1095 : READ Q
: VPOKE I,Q : NEXT : FOR I=1152 TO 11
75 : READ Q : VPOKE I,Q : NEXT : DEFU
SR0=A+10 : DEFUSR1=A+128 : POKE V4VVV
,15
180     FOR I=1 TO 10 : PRINT : NEXT
: PRINT"_Da_a_una_teclea_para_empezar
"
190     FOR I=1 TO 11 : D=RND(1) : I
F INKEY$<>" THEN 210 ELSE READ A$ :
VPOKE 6153+I,ASC(A$) : READ N : B$="n
"+STR$(N) : PLAY"s1m255115xB$;" : FOR
T=1 TO 500 : NEXT : NEXT
200     FOR I=1 TO 500 : NEXT : FOR
I=6154 TO 6164 : VPOKE I,32 : NEXT :
RESTORE 870 : GOTO 190

```

Preparación pantalla

```

210     CLS : FOR I=6496 TO 6527 : V
POKE I,INT(RND(1)*3+144) : NEXT : FOR
I=6848 TO 6879 : VPOKE I,23 : NEXT :
PUT SPRITE 0,(0,16)
220     FOR I=6848 TO 6879 : VPOKE I
,23 : NEXT : PUT SPRITE 0,(0,16),8
230     FOR I=1 TO 22 : PRINT : NEXT
: PRINT"COMB:";CHR$(11);

```

```

240      PUT SPRITE 4,(0,75),14 : PUT
        SPRITE 1,(0,140),1
250      K1=INT(RND(1)*150+40) : PUT
        SPRITE 2,(K1,100),1 : K2=INT(RND(1)*5
        0+30) : PUT SPRITE 3,(K1+K2,100),1
260      STRIG(0) ON : ON STRIG GOSUB
        600 : SPRITE ON : ON SPRITE GOSUB 47
        0
270      ON INTERVAL=180 GOSUB 630 :
        INTERVAL ON : TC=30 : CR=16 : GOSUB 6
        50 : CH=100

```

Control

```

280      D=USR(D) : IF PEEK(V3VVV)=0 T
        HEN IF PEEK(V2VVV)=1 THEN AT=1
290      K1=VPEEK(6912) : IF K1>75 AND
        K1<190 THEN 580
300      GOSUB 350
310      GOSUB 400
320      IF CP=1 THEN GOSUB 440
330      IF AT=1 THEN POKE V4VVV,13 EL
        SE POKE V4VVV,15
340      GOTO 280

```

Movimiento del Harrier

```

350      IF PA=1 AND VPEEK(6912)>65 TH
        EN RETURN
360      IF RE=1 AND VPEEK(6912)<65 TH
        EN RE=0 : PA=0 : AT=0 : SPRITE ON
370      IF AT=1 AND RE=0 THEN K1=2 :
        K2=1 : GOTO 390
380      K1=3 : K2=4
390      POKE V3VVV,0 : POKE V2VVV,K1
        : POKE V1VVV,K2 : D=USR1(D) : RETURN

```



Movimiento del submarino

```
400 K1=RND(1) : IF K1<.5 THEN K2=
3 ELSE IF K1<.75 THEN K2=0 ELSE K2=2
410 IF VPEEK(6916)<110 THEN K2=2
ELSE IF VPEEK(6916)>160 THEN K2=0
420 POKE V3VVV,1 : POKE V2VVV,K2
: POKE V1VVV,2 : D=USR1(D)
430 RETURN
```

Movimiento de las cargas de profundidad

```
440 POKE V3VVV,6 : POKE V2VVV,2 :
POKE V1VVV,3 : D=USR1(D)
450 IF VPEEK(6936)>175 THEN CP=0
: PUT SPRITE 6,(100,200)
460 RETURN
```

Choque

```
470 SPRITE OFF : IF CP=1 AND (VPE
EK(6912)<65 OR VPEEK(6912)>190 OR VPE
EK(6913)>16) THEN IF VPEEK(6936)<90 T
HEN CH=6 : GOTO 530 ELSE IF VPEEK(693
6)<110 THEN 560 ELSE 540
480 IF AT=0 AND CH=0 THEN CH=0 :
GOTO 530
490 IF AT=1 THEN PA=1 : AT=0 ELSE
IF PA=1 THEN RETURN
500 AT=0 : RE=1 : TC=TC-16+CR : I
F TL<0 THEN 580 ELSE IF TC>0 THEN CR=
16 : GOSUB 650
510 IF VPEEK(6913)<>5 AND VPEEK(6
913)<>6 THEN CH=0 : GOTO 530
520 SPRITE ON : RETURN
```

Portaaviones hundido

```
530      VPOKE 6914+CH*4,20 : GOSUB 67
0 : FOR K1=1 TO 200 : NEXT : VPOKE 69
30,20 : GOSUB 670 : FOR K1=1 TO 1000
: NEXT : GOTO 690
```

Submarino bombardeado

```
540      SD=SD+1 : VPOKE 6919,15 : VPO
KE 6918,20: GOSUB 670 : FOR I4=1 TO 1
000 : NEXT : PUT SPRITE 6,(100,200) :
PUT SPRITE 1,(0,140),1 : VPOKE 6918,
4
550      SPRITE ON:RETURN
```

Explosión de mina

```
560      VPOKE 6938,20 : VPOKE 6939,15
: GOSUB 670 : FOR K1=1 TO 1000 : NEX
T : EM=1
570      GOTO 690
```

Harrier tocado

```
580      VPOKE 6912,70 : VPOKE6914,20
: GOSUB 670 : FOR K1=1 TO 1000 : NEXT
590      GOTO 690
```

Disparo de carga de profundidad

```
600      IF CP=1 THEN RETURN
610      CP=1 : I1=VPEEK(6912) : I2=VP
```



```
EEK(6913) : IF I2<8 THEN CP=0 : RETUR  
NELSE PUT SPRITE 6,(I2-8,I1),1  
620 RETURN
```

Depósito de combustible

```
630 CR=CR-1 : VPOKE 6888+CR,32  
640 IF CR<0 THEN 580 ELSE RETURN
```

Recarga de combustible

```
650 FOR I3=6888 TO 6903 : VPOKE I3  
,136 : NEXT  
660 RETURN
```

Explosión

```
670 PLAY"13m200s8n27"  
680 RETURN
```

Fin de juego

```
690 SCREEN 1 :PRINT"_____FIN_D  
E_JUEGO"  
700 PRINT : PRINT  
710 PRINT"Numero_De_Subs_Destruido  
s:";SD : PRINT : PRINT  
720 IF CH=0 THEN PRINT"FUE_UN_MAL_  
ATERRIZAJE_-" : PRINT"HAS_SIDO_DESTR  
UIDO!!!" : GOTO 760  
730 IF CH=6 THEN PRINT"HAS_TIRADO_  
UNA_CARGA_DE_____PROFUNDIDAD" : PRIN  
T"SOBRE_TU_PROPIO_BARCO" : GOTO 760  
740 IF EM=1 THEN PRINT "EXPLOSION_  
DE_MINA_NUCLEAR_!!!" : GOTO 760
```

```
750 IF TC<0 OR CR<0 THEN PRINT"___  
SIN_COMBUSTIBLE_!!!" : GOTO 760  
760 IF INKEY$<>" " THEN END ELSE 76  
0
```

Datos de sprites

```
770 DATA 0,0,0,0,0,192,224,241,255,  
127,127,31,63,124,0,0,0,0,0,0,0,112  
,236,242,255,238,216,0,0,0,0  
780 DATA 0,0,0,0,0,2,7,6,7,127,255,  
255,127,0,0,0,0,0,0,0,128,128,192  
,254,255,255,254,0,0,0  
790 DATA 0,90,60,126,126,60,90,0  
800 DATA 0,0,0,0,192,192,240,240,24  
0,255,255,255,255,255,255,0,0,0,0  
,0,0,0,0,0,255,254,252,248,240,224,19  
2  
810 DATA 0,4,66,136,2,64,18,42,10,1  
65,43,144,40,1,64,18,128,16,4,32,1,72  
,33,146,204,22,200,92,64,9,0,32  
820 DATA 40,56,16,16,56,56,56,16
```

Datos de sprites pequeños

```
830 DATA 255,171,213,171,213,171,21  
3,255  
840 DATA 0,0,4,154,97,0,0,0  
850 DATA 0,0,0,13,210,32,0,0  
860 DATA 0,0,8,181,66,0,0,0
```

Datos de pantalla inicial

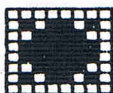
```
870 DATA S,30,E,32,A,35," ",34,H,35  
,A,34,R,33,R,32,I,33,E,34,R,35
```



## Tabla de claves

5	= 192	300	= 250	600	= 903
10	= 904	310	= 300	610	= 5099
20	= 143	320	= 1101	620	= 142
30	= 1637	330	= 2224	630	= 1788
40	= 1541	340	= 176	640	= 1208
50	= 1712	350	= 2008	650	= 2061
60	= 143	360	= 3609	660	= 142
70	= 143	370	= 2582	670	= 1061
80	= 2571	380	= 826	680	= 142
90	= 2998	390	= 3105	690	= 1574
100	= 1185	400	= 4322	700	= 348
110	= 4466	410	= 3569	710	= 3257
120	= 3660	420	= 2801	720	= 4635
130	= 4405	430	= 142	730	= 5699
140	= 3663	440	= 2701	740	= 3316
150	= 3664	450	= 2713	750	= 3408
160	= 3209	460	= 142	760	= 1751
170	= 6585	470	= 8470	770	= 4512
180	= 3840	480	= 2089	780	= 4234
190	= 8981	490	= 2748	790	= 1256
200	= 3283	500	= 4803	800	= 5054
210	= 5899	510	= 3316	810	= 4727
220	= 2521	520	= 548	820	= 1277
230	= 2354	530	= 4949	830	= 1661
240	= 1605	540	= 5397	840	= 998
250	= 6218	550	= 548	850	= 1028
260	= 2717	560	= 2842	860	= 998
270	= 4823	570	= 331	870	= 3024
280	= 4131	580	= 2385		
290	= 2672	590	= 331	TOTAL	= 228496

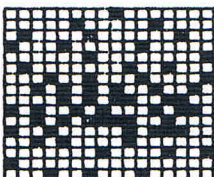
# Diseño de los sprites



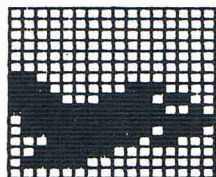
MINA



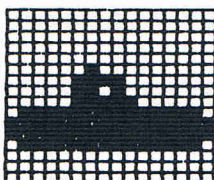
CARGA DE PROFUNDIDAD



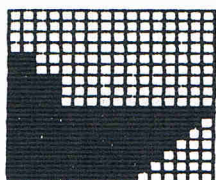
EXPLOSION



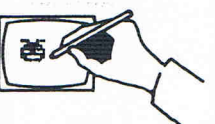
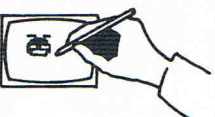
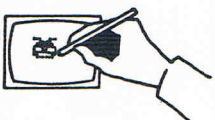
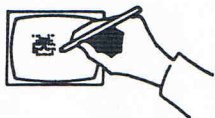
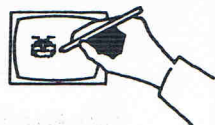
HARRIER



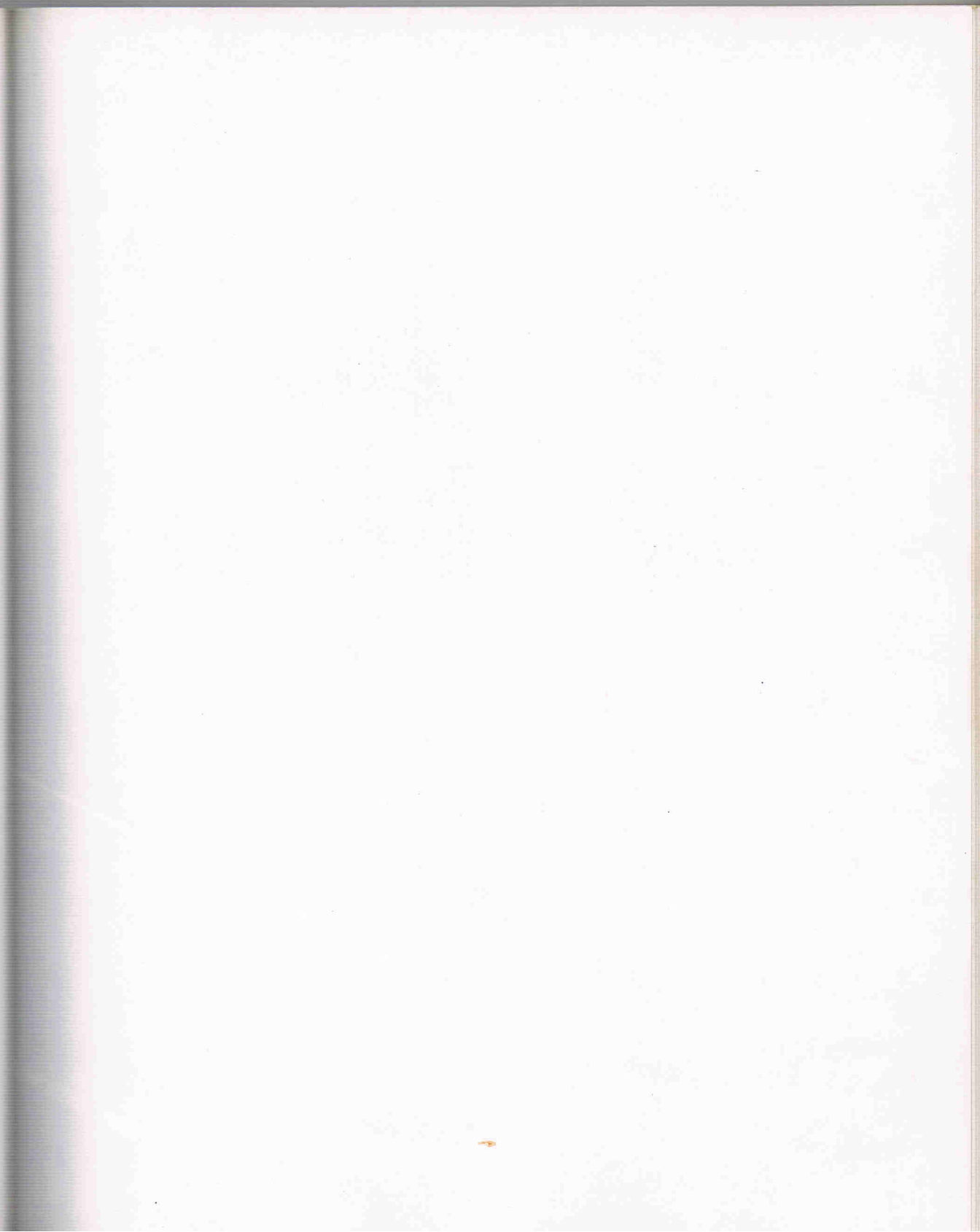
SUBMARINO



PORTAAVIONES











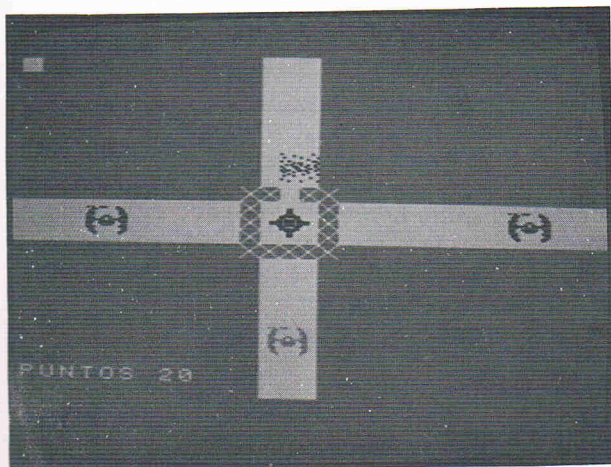


# Invasión de robots

# 12

## ¡ATENCIÓN!

Este programa utiliza la subrutina  
SOPORTE en lenguaje máquina.  
Véase apéndice C.



Tipo de juego:

**JUEGO DE DISPARO**

Flotas de naves-robot suicidas están atacando tu nave de caza. Debes usar tu cañón giratorio para destruir a los robots antes de que lleguen a tu nave. Algunos robots desaparecen de repente; ten cuidado, son peores, pero valen más puntos.

Usa la tecla I para girar el cañón y la barra espaciadora para disparar.

## SUGERENCIAS AL PROGRAMA

Te darás cuenta de que la velocidad de las bombas y de los robots aumenta según juegas. Este efecto lo produce la línea 270, que fija la distancia en que se mueve cada sprite. Intenta cambiar la expresión que se "pokea" en la dirección 59999, para dar más rapidez al juego.

## Variables del programa

NG, PT Número de giros; puntos.  
D(I) Dirección del camino I.  
C Número de camino (1-4).  
DI Control de disparo.  
MS, S(I) Movimiento de sprite; control del sprite I.  
TD Todos los robots destruidos.

Inicialización

```
5 BEEP
10 REM ** ATENCION **
20 REM
30 REM ESTE PROGRAMA NECESITA
40 REM LA SUBROUTINA SOPORTE
50 REM Vease el Apendice C
60 REM
70 REM
80 A=PEEK(64586!)+256*PEEK(6458
7!)
90 V1=A+9:V2=A+8:V3=A+7:V4=A+6
100 TIME=0
110 COLOR 15,4,7:SCREEN 1,2:CLS:
KEY OFF:PRINT"__INVASION__DE__ROBO
TS__":PRINT:PRINT:PRINT
120 FOR I=1 TO 8:READ Q:A#=A#+CH
R$(Q):NEXT:SPRITE$(0)=A#:A#=""
130 FOR I=1 TO 32:READ Q:A#=A#+C
HR$(Q):NEXT:SPRITE$(1)=A#:SPRITE$(2)=
A#:SPRITE$(3)=A#:SPRITE$(4)=A#:A#=""
140 FOR I=1 TO 32:READ Q:A#=A#+C
```



```

HR$(Q):NEXT:SPRITE$(5)=A$:A$=""
150     FOR I=1 TO 32:READ Q:A#=A#+C
HR$(Q):NEXT:SPRITE$(6)=A$
160     DEFUSR=A+128
170     D(1)=2:D(2)=3:D(3)=0:D(4)=1
180     INPUT " QUIERES EMPEZAR A J
UGAR";X$
190     C=1

```

Preparación pantalla

```

200     CLS:COLOR 11,6,6
210     FOR I=6158 TO 6894 STEP 32: F
OR J=0 TO 2: VPOKE I+J,255:NEXT:NEXT
220     FOR I=6464 TO 6559:VPOKE I,25
5:NEXT
230     FOR I=6445 TO 6449:VPOKE I,28
:NEXT:FOR I=6464 TO 6528 STEP 32 : VP
OKE I+13,28:VPOKE I+17,28:NEXT
240     FOR I=6573 TO 6577 :VPOKE I,2
8:NEXT
250     PUT SPRITE 5,(116,83),1:GOSUB
820

```

Nueva jugada

```

260     TD=0:NG=NG+1:PUT SPRITE 1,(11
6,0),8:PUT SPRITE 2,(0,83),1:PUT SPRI
TE 3,(116,192),8:PUT SPRITE 4,(255,83
),1
270     POKE V1VVV,INT(NG/3)+1
280     GOSUB 510:STRIG(0) ON:ON STRI
G GOSUB 440
290     SPRITE ON:ON SPRITE GOSUB 730
300     FOR I=1 TO 4:S(I)=1:NEXT

```

Control

```
310 IF DI=1 THEN 330
320 GOSUB 410
330 IF DI=1 THEN GOSUB 570
340 SPRITE OFF: GOSUB 640:SPRITE
DN
350 ST$=STR$(TIME):DG=VAL(RIGHT$(
ST$,1))
360 IF DG>1 OR RND(1)<.95 THEN 31
0
370 IF RND(1)<.5 THEN PUT SPRITE
2,(116,200):S(2)=0:GOTO 390
380 TD=1:FOR I=1 TO 4: IF S(I)=1
THEN TD=0
390 TD=1:FOR I=1 TO 4:IF S(I)=1 T
HEN TD=0
400 NEXT:GOTO 310
```

Leer del teclado

```
410 KP$=INKEY$:IF KP$<>"i" AND KP
$<>"I" THEN RETURN
420 C=C+1: IF C=5 THEN C=1
430 GOSUB 510:RETURN
440 IF DI=1 THEN RETURN
450 PLAY"119m380s10n37"
460 DI=1:ON C GOTO 470,480,490,50
0
470 PUT SPRITE 0,(120,70),1:RETUR
N
480 PUT SPRITE 0,(103,87),1:RETUR
N
490 PUT SPRITE 0,(120,103),1:RETU
RN
500 PUT SPRITE 0,(137,87),1:RETUR
N
```



Visualizar el búnker

```
510      VPOKE 6447,28:VPOKE 6509,28:V
POKE 6575,28:VPOKE 6513,28
520      ON C GOTO 530,540,550,560
530      VPOKE 6447,255:RETURN
540      VPOKE 6509,255:RETURN
550      VPOKE 6575,255:RETURN
560      VPOKE 6513,255:RETURN
```

Disparar bomba

```
570      TP=PEEK(V1VVV):POKE V3VVV,0:P
DKE V1VVV,2
580      POKE V2VVV,C-1:D=USR(D):POKE
V1VVV,TP
590      ON C GOTO 600,610,620,630
600      IF VPEEK(6912)<5 THEN DI=0:PU
T SPRITE 0,(200,200):RETURN ELSE RETU
RN
610      IF VPEEK(6913)<5 THEN DI=0: P
UT SPRITE 0,(200,200):RETURN ELSE RET
URN
620      IF VPEEK(6912)>187 THEN DI=0:
PUT SPRITE 0,(200,200):RETURN ELSE R
ETURN
630      IF VPEEK(6913)>250 THEN DI=0:
PUT SPRITE 0,(200,200):RETURN ELSE R
ETURN
```

Movimientos del enemigo

```
640      MS=MS+1:IF MS=5 THEN MS=1
650      IF TD=1 THEN 260
660      IF S(MS)=0 THEN 640
670      POKE V3VVV,MS:POKE V2VVV,D(MS
):D=USR(D)
```

```
680     DN MS GOTO 690,700,710,720
690     IF VPEEK(6916)>62 THEN 790 EL
SE RETURN
700     IF VPEEK(6921)>95 THEN 790 EL
SE RETURN
710     IF VPEEK(6924)<111 THEN 790 E
LSE RETURN
720     IF VPEEK(6929)<145 THEN 790 E
LSE RETURN
```

Nave bombardeada

```
730     STRIG(0) OFF:SPRITE OFF : DI=
0: IF C=1 OR C=3 THEN PT=PT+20:ELSE P
T=PT+100
740     GOSUB 820:S(C)=0:PUT SPRITE C
,(10,200):VPOKE 6914,24
750     PLAY "t255164r64m70s10n26n2"
760     FOR T=1 TO 150:NEXT: VPOKE 69
14,0: PUT SPRITE 0,(200,200)
770     TD=1: FOR I=1 TO 4: IF S(I)=1
THEN TD =0
780     NEXT:STRIG(0) ON: SPRITE ON:
RETURN
```

Fin de juego

```
790     STRIG(0) OFF
800     CLS:SCREEN 1: PRINT "_____"
FIN_DE_JUEGO":PRINT:PRINT:PRINT:PRINT
:PRINT"____TUS_PUNTOS_SON:";PT
810     END
```

Puntuación

```
820     FOR I=1 TO 22:PRINT: NEXT: PRI
NT "PUNTOS";PT;CHR$(11);:RETURN
```



Datos de sprites

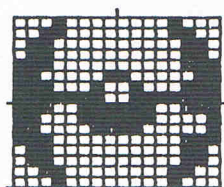
830 DATA 24, 24, 60, 255, 255, 60, 24, 24  
 840 DATA 16, 48, 96, 96, 224, 193, 226, 25  
 4, 231, 199, 195, 224, 96, 96, 48, 16, 8, 12, 6,  
 6, 7, 131, 71, 127, 231, 227, 195, 7, 3, 3, 6, 4  
 850 DATA 1, 1, 1, 3, 15, 12, 27, 251, 251, 2  
 7, 12, 15, 3, 1, 1, 1, 128, 128, 128, 192, 240, 4  
 8, 216, 223, 223, 216, 48, 240, 192, 128, 128,  
 128  
 860 DATA 128, 4, 64, 0, 144, 5, 40, 134, 67  
 , 177, 68, 16, 68, 0, 128, 16, 4, 0, 129, 16, 6, 1  
 29, 146, 34, 25, 162, 16, 65, 0, 8, 0, 129

## Tabla de claves

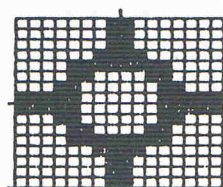
5	= 192	210	= 3342	420	= 1699
10	= 904	220	= 1820	430	= 610
20	= 143	230	= 4277	440	= 897
30	= 1637	240	= 1720	450	= 1167
40	= 1541	250	= 1128	460	= 1917
50	= 1712	260	= 4991	470	= 1046
60	= 143	270	= 1730	480	= 1046
70	= 143	280	= 2098	490	= 1079
80	= 2571	290	= 1129	500	= 1080
90	= 2998	300	= 1439	510	= 1970
100	= 459	310	= 844	520	= 681
110	= 4062	320	= 310	530	= 812
120	= 3669	330	= 970	540	= 874
130	= 5862	340	= 1183	550	= 940
140	= 3664	350	= 2779	560	= 878
150	= 3199	360	= 2040	570	= 2615
160	= 1060	370	= 3013	580	= 2463
170	= 1878	380	= 2482	590	= 961
180	= 2060	390	= 2482	600	= 3172
190	= 324	400	= 395	610	= 3173
200	= 566	410	= 3121	620	= 3350

630	=	3414	720	=	1717	810	=	129
640	=	2071	730	=	5053	820	=	2919
650	=	785	740	=	2063	830	=	1364
660	=	1081	750	=	1765	840	=	5056
670	=	2443	760	=	2468	850	=	5174
680	=	1414	770	=	2482	860	=	4813
690	=	1619	780	=	1460			
700	=	1657	790	=	751			
710	=	1678	800	=	4360	TOTAL	=	172246

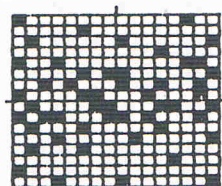
## Diseño de los sprites



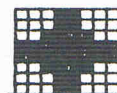
NAVE ENEMIGA



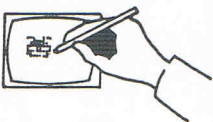
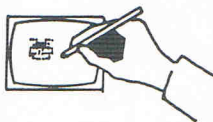
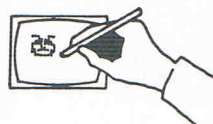
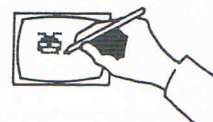
CAZA PROPIO



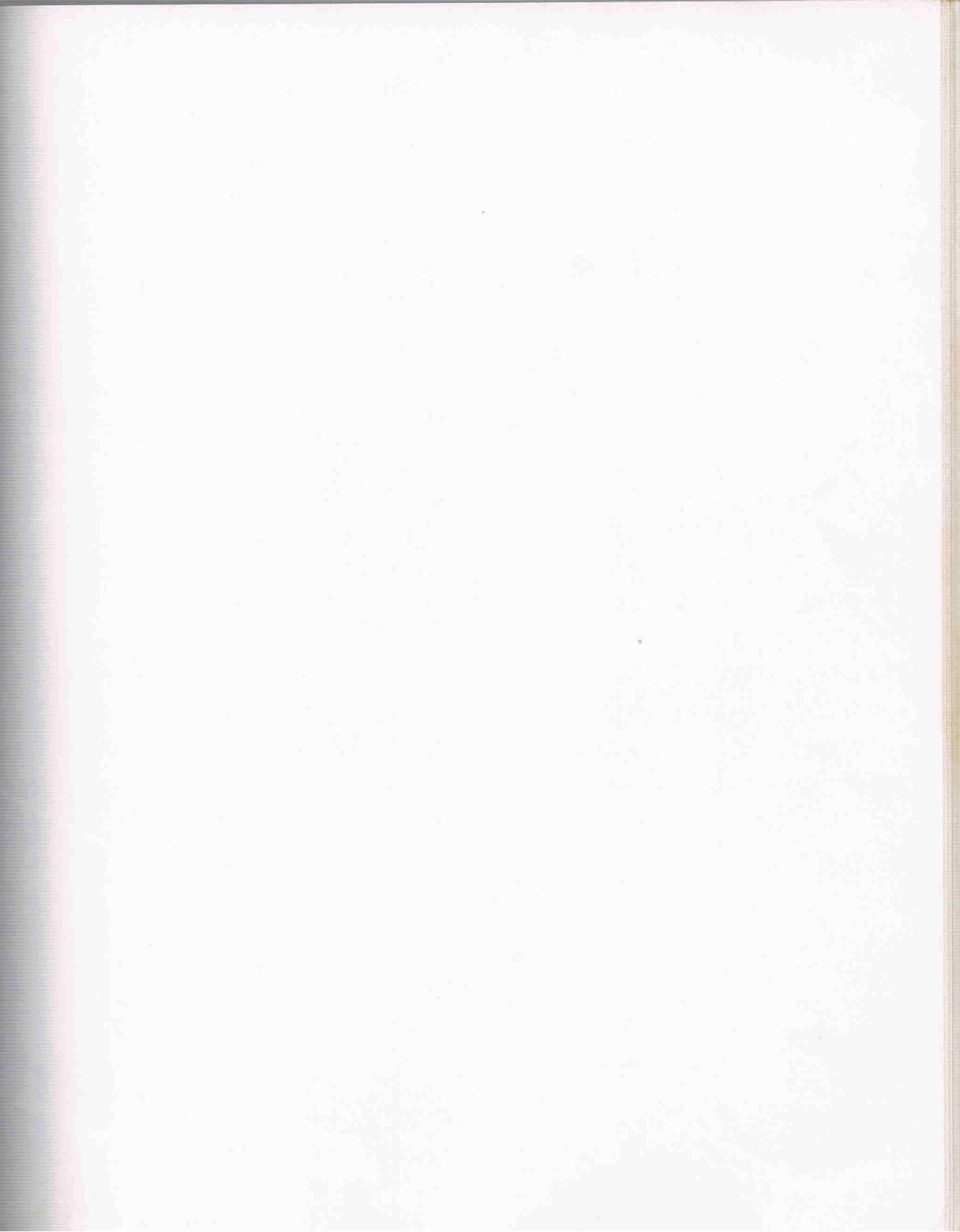
EXPLOSION



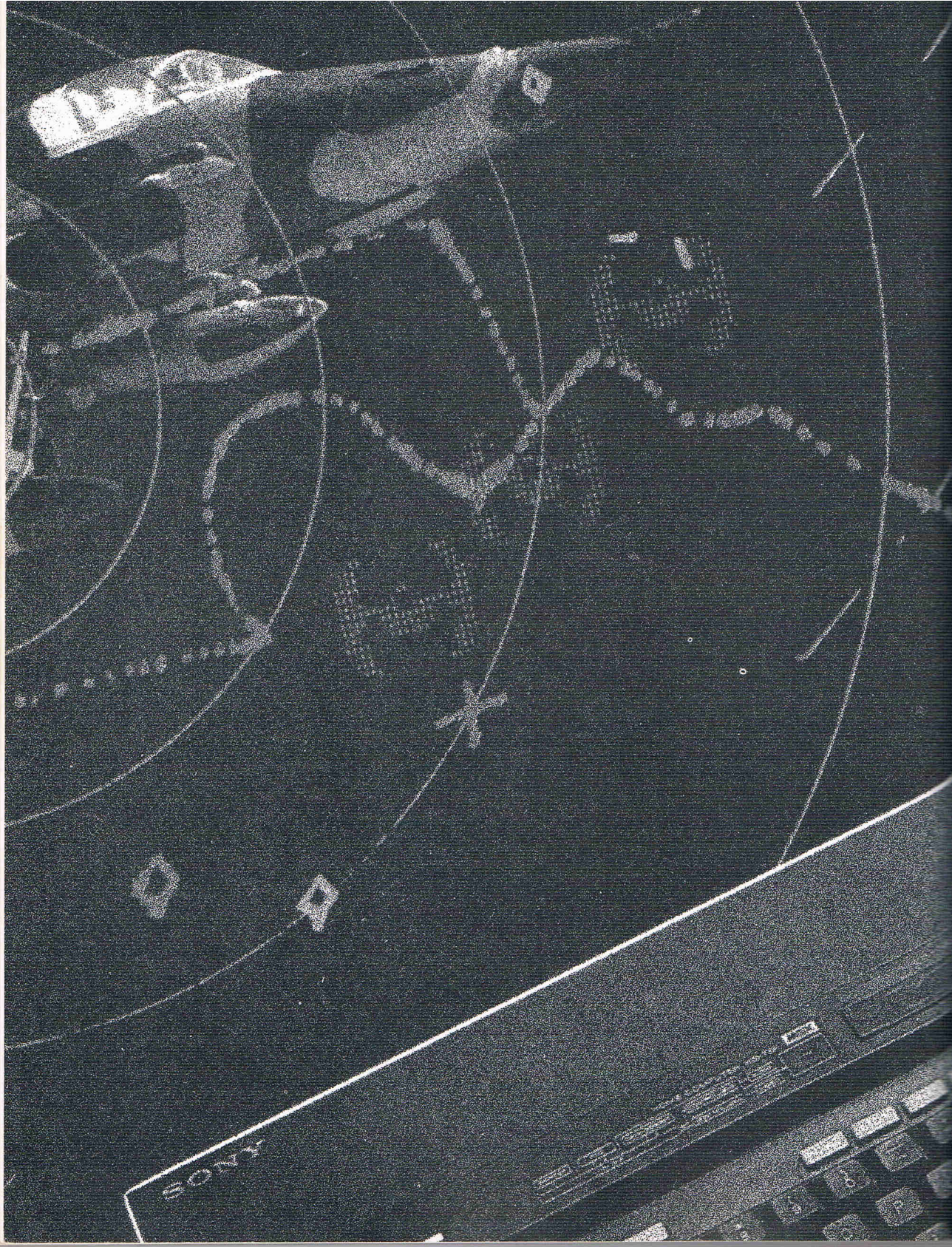
BOMBA







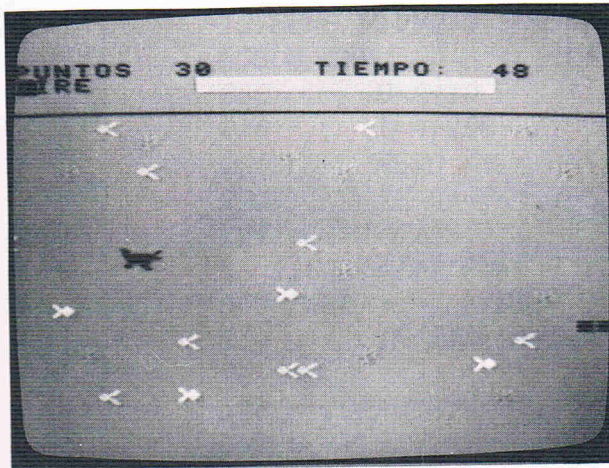






# Cocodrilo

# 13



## ¡ATENCIÓN!

Este programa utiliza la subrutina  
SOPORTE en lenguaje máquina.  
Véase apéndice C.

**Tipo de juego:**

**JUEGO CON TIEMPO LIMITE**

Ayuda al cocodrilo hambriento a comer, guiándolo en dirección a las plantas, los peces exóticos y los deliciosos peces amarillos para que coma.

Se tiene que comer todos los peces amarillos y situarse en la parte de arriba de la pantalla dentro del tiempo límite.

No te olvides de que los cocodrilos no pueden estar toda la vida bajo el agua; se te puede ahogar el pobrecito. Estáte atento a la parte superior de la pantalla, donde se te indica la medida de aire que te queda.

# SUGERENCIAS AL PROGRAMA

En la línea 250 se almacena un número en la dirección 59999 para controlar la velocidad del cocodrilo; cámbialo a 2 para aumentar el nivel de dificultad, y a 4 para disminuirlo.

¿Qué pasaría si el cocodrilo tuviera más grandes los pulmones? Si aumentas el número 20 de la línea 460, el cocodrilo podrá permanecer más tiempo debajo del agua.

## Variables del programa

AI	Aire inicial.
TC	Tiempo total consumido.
AR	Aire que queda.
PN	Modelo de números del sprite 0.
R, PT	Número de inmersiones; puntuación.
RA	Control de retoma de aire.
NC	Número de cocodrilos.
TR	Tiempo que resta para llegar al otro lado.

```

Inicialización

100      REM   ** ATENCION **
110      REM
120      W1=A+249:W2=A+248:W3=A+247:
W4=A+246
130      REM ESTE PRDGRAMA NECESITA
140      REM LA SUBRUTINA SOPORTE
150      REM Vease el Apendice C
160      REM
170      REM
180      A=PEEK(64586!)+256*PEEK(645
87!)
190      V1=A+9:V2=A+8:V3=A+7:V4=A+6
200      W1=A+249:W2=A+248:W3=A+247:
W4=A+246
210      KEY OFF: COLOR 1,4,4:SCREEN
1,2: PRINT "_____COCODRILOS!":PRI
NT:PRINT:PRINT:PRINT
220      FOR I=1 TO 32:READ Q:A$=A$+C
```



```

HR$(Q):NEXT:SPRITE$(0)=A$:A$=""
230     FOR I=1 TO 8:READ Q:A#=A#+CH
R$(Q):NEXT:FOR I=1 TO 4:SPRITE$(I)=A
$:NEXT:A$=""
240     FOR I=1 TO 32:READ Q:A#=A#+C
HR$(Q):NEXT:SPRITE$(5)=A$
250     NC=3:DEFUSR=A+10:POKE V4VVV,
13:POKE V1VVV,3:SPRITE ON :ON SPRITE
GOSUB 700
260     B=A+10
270     I=A+250
280     READ A$
290     IF A$="R" THEN READ C:C=C+B:
C1 = INT(C/256):C2=256*(C/256-C1):PO
KE I,C2:I=I+1:POKE I,C1 :GOTO 310
300 POKE I, VAL(A$)
310     IF I<A+250+51 THEN I=I+1:GO
TO 280
320     :DEFUSR1=A+250!:POKE W4WWW,96:P
OKE W3WWW,24:POKE W2WWW,255:POKE W1WW
W,26
330     FOR I=1088 TO 1280 STEP 64:F
OR J=0 TO 7:READ Q:VPOKE I+J,Q:NEXT:N
EXT
340     PRINT"_PULSA_UNA_TECLA_PARA_
EMPEZAR"
350     IF INKEY$="" THEN 350

```

Nueva inmersión

```

360     CLS:IF NC=0 THEN 840
370     FOR I=6240 TO 6271:VPOKE I,23
: NEXT
380     FOR I=6274 TO 6882 STEP 32: F
OR J=0 TO 29:K=10*(TIME/10-INT(TIME/1
0)):IF RND(1)<.95 THEN 420
390     IF K<1 THEN VPOKE I+J,144:GOT
O 420
400     IF K<6 THEN VPOKE I+J,136:GOT
O 420

```

```

410     VPOKE I+J,152
420     NEXT:NEXT
430     K=K*3+80:PUT SPRITE 1,(30,K+3
0),11:PUT SPRITE 2,(80,K+77),11:PUT S
PRITE 3,(120,K+20),11:PUT SPRITE 4,(2
00,K+60),11
440     R=R+1
450     PUT SPRITE 0,(0,80),1
460     TIME=0:AI=20-R:TG=0:GOSUB 920
:PRINT CHR$(11):PRINT "AIRE":GOSUB 95
0
470     ON INTERVAL=15 GOSUB 870:INTE
RVAL ON
480     VPOKE 8209,36:VPOKE 8210,20:V
POKE 8211,148:VPOKE 8212,164

```

Control

```

490     D=USR(D):SPRITE DN:IF PN=0 TH
EN PN=20 ELSE PN=0
500     GOSUB 920:VPOKE 6914,PN:IF AR
=AI THEN 880
510     IF TR<1 THEN 840
520     IF VPEEK(6912)<20 THEN GOSUB
640:POKE V4VVV,12:GOTO 540 ELSE IF VP
EEK(6912)<185 THEN POKE 5996!,13 ELSE
POKE V4VVV,9
530     RA=0
540     IF VPEEK(6913)<253 THEN 570
550     IF VPEEK(6916)=200 AND VPEEK(
6920)=200 AND VPEEK(6924)=200 AND VPE
EK(6928)=200 THEN 360 ELSE CLS:INTERV
AL OFF : FOR I=0 TO 5: PUT SPRITE I,(
30*I,200):NEXT:PRINT"Has_perdido_vari
os_peces!!":PRINT:PRINT"Tienes";NC-1;
"cocodrilos vivos"
560     FOR T=1 TO 2500:NEXT:NC=NC-1:
GOTO 360
570     K1=INT((VPEEK(6913)+14)/8)+32
*(INT((VPEEK(6912)+4)/8))+6144:IF VPE

```



```
EK(K1)=32 THEN 490
580   IF VPEEK(K1)=152 THEN GOSUB67
    0
590   IF VPEEK(K1)=144 THEN 800
600   IF VPEEK(K1)=136 THEN GOSUB 6
    20
610   GOTO 490
```

Comer algas

```
620   INTERVAL OFF:PT=PT+10:VPOKE K
    1,32:GOSUB 920
630   INTERVAL ON:RETURN
```

Toma de aire del cocodrilo

```
640   IF RA=1 THEN 660
650   TG=TG+INT(TIME/60):TIME=0:GOS
    UB 360:RA=1
660   RETURN
```

Comer peces exóticos

```
670   INTERVAL OFF:PT=PT+20:VPOKE K
    1,32:GOSUB 920
680   PLAY"154m1200s10n54"
690   INTERVAL ON:RETURN
```

Comer peces amarillos

```
700   SPRITE OFF:I1=VPEEK(6913):IF
    I1<39 THEN 710 ELSE IF I1<89 THEN 730
    ELSE IF I1<129 THEN 750 ELSE GOTO 77
    0
710   IF I1>22 THEN RETURN
```

```
720 PUT SPRITE 1,(30,200):GOTO 79
0
730 IF I1>72 THEN RETURN
740 PUT SPRITE 2,(80,200):GOTO 79
0
750 IF I1>112 THEN RETURN
760 PUT SPRITE 3,(120,200):GOTO 7
90
770 IF I1>192 THEN RETURN
780 PUT SPRITE 4,(200,200)
790 PT=PT+150:PLAY"124m160s8n27n2
7":GOSUB 920:RETURN
```

Choque con una mina

```
800 INTERVAL OFF:FOR I=1 TO 4:COLOR
R 1,15,15:PLAY"m8000s112n30":FOR J=1
TO 150: NEXT: COLOR 1,4,4: FOR J=1 TO
150: NEXT:NEXT
810 CLS:FOR I=0 TO 5:PUT SPRITE I
,(30*I,200):NEXT:PRINT"El_cocodrilo_c
hoco_con_minas":PRINT"Ha_sufrido_inde
gestion!!!!"
820 PRINT:PRINT:PRINT:PRINT"_Tien
es"NC-1"COCODRILOS VIVOS":FOR J=1 TO
2500:NEXT
830 NC=NC-1:GOTO 360
```

Fin de juego

```
840 SCREEN1:PRINT"_____FIN_D
E_JUEGO":PRINT:PRINT:PRINT:PRINT
850 PRINT"____Tus_puntos_son_";PT
860 IF INKEY$="" THEN END ELSE 86
0
```

Animación del juego

```
870 D=USR1(D): RETURN
```



Cocodrilo ahogado

```
880     CLS:INTERVAL OFF:FOR I=0 TO 5
      :PUT SPRITE I,(30*I,200):NEXT:PRINT"_
      _COCODRILO_ahogado_"
890     PRINT:PRINT:PRINT:PRINT"_Tien
      es_";NC-1;"COCODRILOS VIVOS"
900     PLAY"11m59000s8n2":FOR T=1 TO
      2500:NEXT
910     NC=NC-1:GOTO 360
```

Tiempo/puntos

```
920     AR=INT(TIME/60):VPOKE 6183+AR,
      32
930     TR=50-TG-AR:PRINT CHR$(11);"PU
      NTOS_";PT;TAB(15);"TIEMPO: ";TR
940     RETURN
```

Recarga de aire

```
950     FOR I=6183 TO 6183+AI:VPOKE I,
      160:NEXT:RETURN
```

Datos de sprites

```
960     DATA 0,0,0,0,192,96,96,119,63,6
      3,31,15,24,48,48,0,0,0,0,0,0,1,2,180,
      248,255,240,224,96,192,96,48
970     DATA 128,196,110,63,63,110,196,
      128
980     DATA 0,0,0,128,192,192,96,119,6
      3,63,31,15,24,48,97,97,0,0,0,0,0,16
      ,184,255,255,240,224,96,192,128,128
```

Datos de animación

990 DATA 6,31,42,R,236,43,35,205,R,18  
7,120,254,31,202,R,265,4,43,205,R,204  
,35,195,R,277,6,0,17,31,0,25,205,R,20  
4,183,237,82,237,91,R,238,123,189,194  
,R,246,122,188,194,R,246,201

Datos de sprites pequeños

1000 DATA 146,84,185,82,181,24,16,4  
0  
1010 DATA 0,90,60,126,126,60,90,0  
1020 DATA 3,6,108,176,240,108,6,3  
1030 DATA 255,255,255,255,255,255,2  
55,255

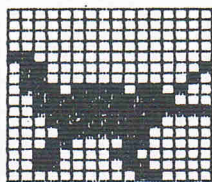
## Tabla de claves

100	=	904	260	=	636	420	=	320
110	=	143	270	=	883	430	=	5767
120	=	3954	280	=	236	440	=	662
130	=	1637	290	=	6327	450	=	738
140	=	1541	300	=	854	460	=	3763
150	=	1712	310	=	2426	470	=	2769
160	=	143	320	=	4197	480	=	1756
170	=	143	330	=	3267	490	=	2977
180	=	2571	340	=	2175	500	=	1893
190	=	2998	350	=	1009	510	=	870
200	=	3954	360	=	1064	520	=	5332
210	=	3327	370	=	1586	530	=	403
220	=	3659	380	=	6077	540	=	1483
230	=	4670	390	=	1853	550	=	15055
240	=	3198	400	=	1850	560	=	2271
250	=	3906	410	=	797	570	=	6023



580	=	1690	740	=	1110	900	=	2280
590	=	1384	750	=	986	910	=	1103
600	=	1624	760	=	1151	920	=	2171
610	=	386	770	=	1066	930	=	4095
620	=	2875	780	=	998	940	=	142
630	=	1291	790	=	2931	950	=	2157
640	=	925	800	=	6269	960	=	4593
650	=	3078	810	=	7849	970	=	1572
660	=	142	820	=	4293	980	=	4958
670	=	2885	830	=	1103	990	=	8621
680	=	1205	840	=	2418	1000	=	1426
690	=	1291	850	=	1993	1010	=	1256
700	=	5429	860	=	1357	1020	=	1264
710	=	896	870	=	895	1030	=	1688
720	=	1059	880	=	5545			
730	=	946	890	=	3285	TOTAL	=	233460

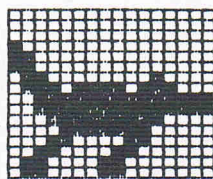
## Diseño de los sprites



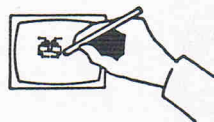
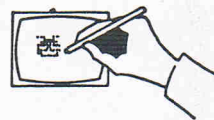
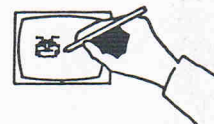
COCODRILO:  
BOCA ABIERTA



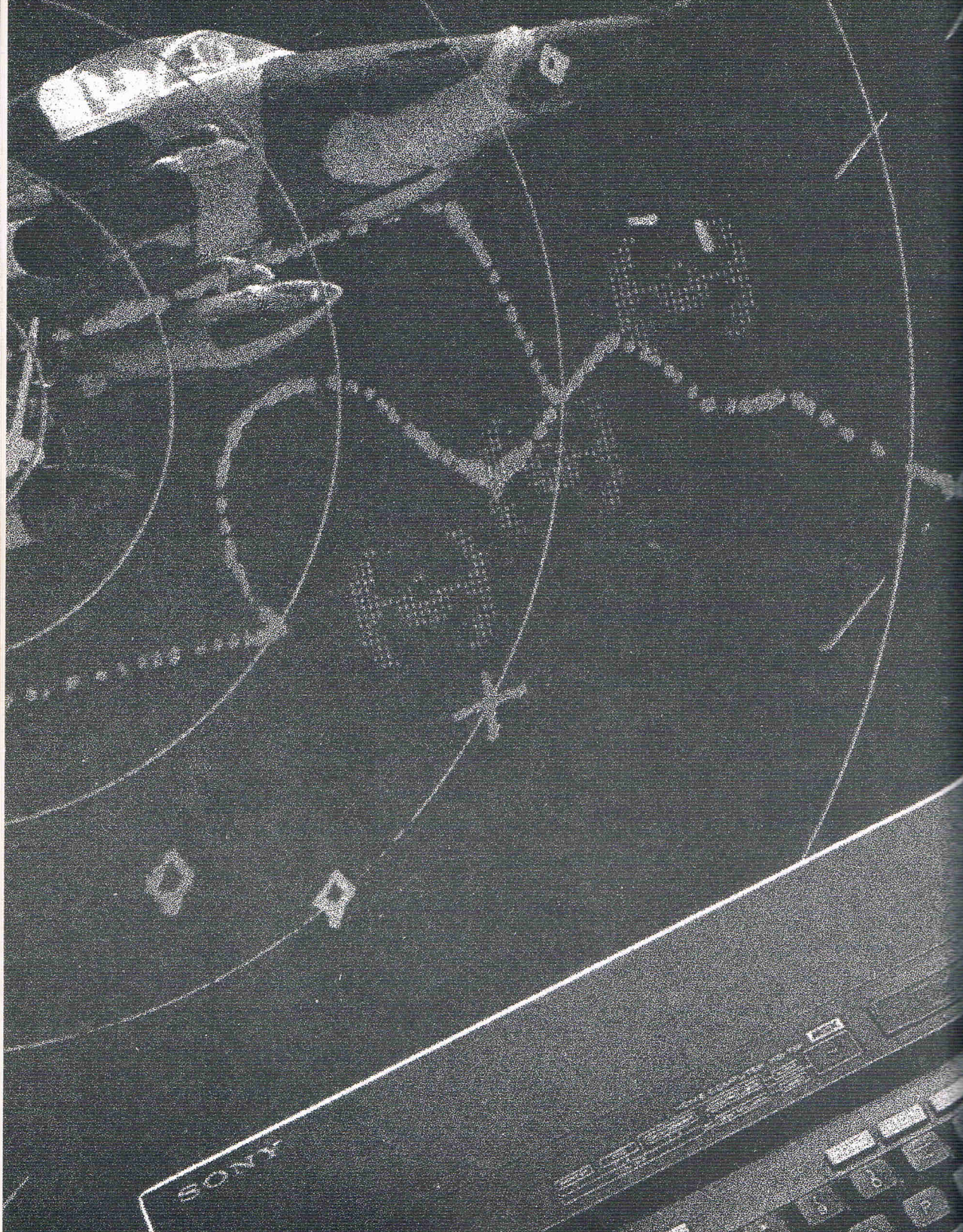
PEZ AMARILLO



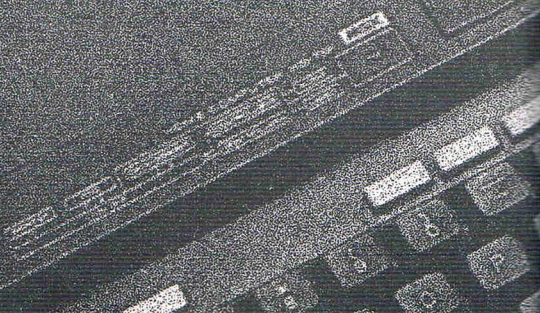
COCODRILO:  
BOCA CERRADA







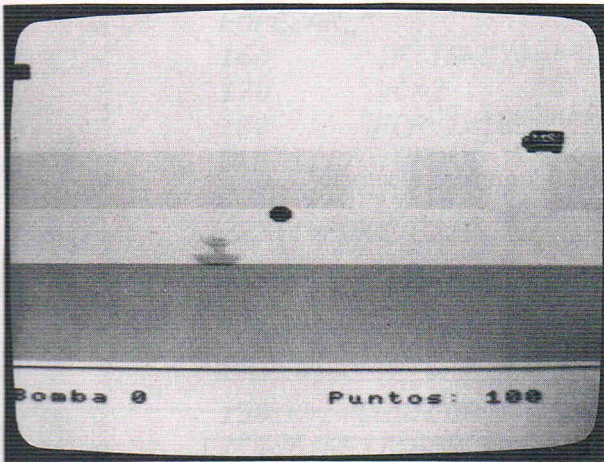
SONY





# Convoy

# 14



## ¡ATENCIÓN!

Este programa utiliza la subrutina SOPORTE en lenguaje máquina. Véase apéndice C.

**Tipo de juego:**

**JUEGO CON TIEMPO LIMITE**

Eres un saboteador y tu misión es detener un convoy de camiones enemigos. Se trata de colocar unas bombas de relojería bajo la carretera, moviendo tu barca bajo un río subterráneo.

El problema es que tus camiones (pintados de rojo) deben poder pasar, y no puedes controlar la mecha de tus bombas.

Para poner una bomba, o arreglar la carretera destruida, pulsa la barra espaciadora, pero ten cuidado con los camiones que caen.

# SUGERENCIAS AL PROGRAMA

Los camiones que caen parecerían más grandes si salpicaran mucho e hicieran bastante ruido al golpear el agua. Estos efectos se podrían incluir en la rutina "Caída de camión" en las líneas 670-720.

## Variables del programa

UC	Ultimo camión.
P	Puntuación.
DM	Distancia máxima entre camiones.
C(I)	Colores de los sprites.
NA, NF	Número aleatorio; semilla del número aleatorio.
LM, CM	Longitud mecha; control mecha.
VC	Velocidad camiones.
CJ	Columna jugador.
PB	Posición bomba.
CP	Control posición bomba.
UM	Ultimo camión movil.

Inicialización

```
5 BEEP
10 REM ** ATENCION **
20 REM
30 REM ESTE PROGRAMA NECESITA
40 REM LA SUBROUTINA SOPORTE
50 REM Vease el Apendice C
60 REM
70 REM
80 A=PEEK(64586!)+256*PEEK(6458
7!)
90 V1=A+9:V2=A+8:V3=A+7:V4=A+6
100 COLOR 1,7,7:SCREEN 1,2:KEY 0
FF:PRINT" _CONVOY_":FOR I=1
TO 9:PRINT:NEXT
110 FOR I=1 TO 32:READ Q:A#=A#+C
HR$(Q):NEXT:SPRITE$(0)=A#:A#=""
120 FOR I=1 TO 32:READ Q:A#=A#+C
HR$(Q):NEXT:FOR I= 1 TO 4:SPRITE$(I)=
```



```

A$:NEXT
130     PUT SPRITE 1,(20,50),1:PUT S
PRITE 2,(80,50),8:PUT SPRITE 3,(160,5
0),1:PUT SPRITE 4,(220,50),8
140     DEFUSR0=A+10:DEFUSR1=A+128:P
DKE V4VVV,10
150     PRINT"_PULSA_UNA_TECLA_PARA_
EMPEZAR_"
160     IF INKEY$="" THEN 160
170     VC=2
180     FOR I=1088 TO 1095 STEP 2:VP
DKE I,170:VPOKE I+1,85:NEXT:FOR I=115
2 TO 1159:VPOKE I,255:NEXT:VPOKE 8209
,231:VPOKE 8210,71

```

Preparación pantalla

```

190     CLS:FOR I=6336 TO 6463:VPOKE
I,136:NEXT:FOR I=6592 TO 6815:VPOKE I
,144:NEXT
200     FOR I=6816 TO 6847:VPOKE I,23
:NEXT
210     FOR I=1 TO 23: PRINT:NEXT:PRI
NT"Bomba";CHR$(11);:GOSUB 850
220     PUT SPRITE 1,(10,200):PUT SPR
ITE 2,(40,200):PUT SPRITE 3,(70,200):
PUT SPRITE 4,(100,200):UC=1:DM=25:GDS
UB 490
230     PUT SPRITE 0,(120,95),11
240     STRIG(0) ON:ON STRIG GOSUB 81
0
250     NF=10001*(TIME/60-INT(TIME/60
))
260     ON SPRITE GOSUB 750

```

Control

```

270     POKE V1VVV,5:D=USR0(D)
280     GOSUB 780:IF CM=0 AND NA<.2 T

```

```

HEN GOSUB 780:LM=INT(NA*5+1):CM=1:TIM
E=0
290     GOSUB 870
300     LT=LT+1:IF LT=5 THEN LT=1
310     IF VPEEK(6912+4*LT)=200 THEN
300 ELSE POKE V3VVV,LT:POKE V2VVV,1:P
OKE V1VVV,VC:D=USR1(D):IF(VPEEK(6913+
4*LT)<2+VC)AND C(LT)=1 THEN 760
320     IF VPEEK(6913+4*LT)<2+VC THEN
PUT SPRITE LT,(30*LT-20,200)
330     C=INT((VPEEK(6913+4*LT)+8)/8)
:IF VPEEK(6368+C)=32 AND VPEEK(6367+C
)=32AND VPEEK(6369+C)= 32 THEN GOSUB
670
340     IF CM=1 AND CP=0 AND INT(TIME
/60)>LM THEN 730
350     IF CP=1 AND INT(TIME/60)>LM T
HEN GOSUB 560
360     GOSUB 780: IF NA<.05 THEN GDS
UB 490
370     GOTO 270

```

Reparar carretera

```

380     PLAY"19m1000s14n53":IF CJ<3 T
HEN 430
390     IF CJ>28 THEN 460
400     FOR I1=6336 TO 6432 STEP 32:F
OR I2=CJ-2 TO CJ+2
410     IF VPEEK(I1+I2)=32 THEN VPOKE
I1+I2,136
420     NEXT:NEXT:RETURN
430     FOR I1=6336 TO 6432 STEP 32:F
OR I2=0 TO CJ
440     IF VPEEK(I1+I2)=32 THEN VPOKE
I1+I2,136
450     NEXT:NEXT:RETURN
460     FOR I1=6336 TO 6432 STEP 32:
FOR I2=29 TO 31

```



```
470 IF VPEEK(I1+I2)=32 THEN VPOKE
    I1+I2,136
480 NEXT:NEXT:RETURN
```

Nuevo camión

```
490 J=0:FOR I=1 TO 4:IF VPEEK(691
2+4*I)=200 THEN J=I
500 NEXT
510 IF J=0 THEN RETURN
520 XP=VPEEK(6913+4*UC):IF 255-XP
<DM THEN RETURN
530 UC=J:GOSUB 780:IF NA<.4 THEN
C(UC)=1 ELSE C(UC)=8
540 PUT SPRITE UC,(255,34),C(UC)
550 RETURN
```

Explosión de bomba

```
560 PLAY"s8m200164n30n28n26n24":C
M=0:VPOKE 6464+PB,32:CP=0:IF PB=31 TH
EN 600
570 IF PB=0 THEN 620
580 FOR J=6336 TO 6432 STEP 32:FO
R K=PB-1 TO PB+1
590 VPOKE J+K,32:NEXT:NEXT:RETURN
600 FOR J=6336 TO 6432 STEP 32:FO
R K= 29 TO 31
610 VPOKE J+K,32:NEXT:NEXT:RETURN
620 FOR J=6336 TO 6432 STEP 32:FO
R K=0 TO 2
630 VPOKE J+K,32:NEXT:NEXT:RETURN
```

Colocar bomba

```
640 CP=1:PB=CJ:VPOKE 6464+CJ,2
650 PLAY"154m1200s10n35"
660 RETURN
```

Caída de camión

```
670     SPRITE ON:POKE V3VVV,LT:POKE
V2VVV,2:POKE V1VVV,3
680     D=USR1(D):IF VPEEK(6912+4*LT)
<100 THEN 680
690     IF C(LT)=8 THEN 760
700     PT=PT+50:VPOKE 6912+4*LT,200:
VPOKE 6913+4*LT,LT*30-20
710     IF PT<300 THEN VC=3 ELSE IF P
T<500 THEN VC=4 ELSE IF PT<700 THEN V
C=5 ELSE IF PT<900 THEN VC=6 ELSE IF
PT<1400 THEN VC=7 ELSE VC=8
720     GOSUB 490:SPRITE OFF:GOSUB 85
0:RETURN
```

Jugador destruido

```
730     FOR I=1 TO 40:GOSUB 780:A$="n
"+STR$(INT(10*NA+1)):PLAY "164m350s8x
a$;":VPOKE 14336+INT(60*NA),NA*255:NE
XT
740     GOTO 760
```

Hombre aplastado

```
750     CLS:PRINT"ACABAS_DE_SER_APLAS
TADO":FOR I=1 TO 2500:NEXT
```

Fin de juego

```
760     STRIG(0) OFF:SCREEN 1:PRINT"
_____FIN_DE_JUEGO":FOR I=1 TO 9:PRI
NT:NEXT:PRINT"_____TUS_PUNTOS_SON:";P
T
```



```
770 IF INKEY$="" THEN END ELSE 77
0
```

Número aleatorio

```
780 NA=(9999*NA+NF)MOD 2997!:NA=N
A/2997!
790 IF NA<.5 THEN NF=NA*10000+1
800 RETURN
```

Espacio pulsado

```
810 CJ=INT((VPEEK(6913)+8)/8)
820 IF VPEEK(6432+CJ)<>32 AND CP=
0 THEN GOSUB 640:RETURN
830 IF VPEEK(6432+CJ)=32 THEN GOS
UB 380:RETURN
840 RETURN
```

Puntuación

```
850 FOR I=1 TO 22:PRINT:NEXT:PRINT
TAB(16) CHR$(10)"Puntos:";PT;CHR$(11
);
860 RETURN
```

Tiempo restante

```
870 I=LM-INT(TIME/60):IF I<0 THEN
I=0
880 IF CM=0 THEN VPOKE 6888,32:RET
URN ELSE VPOKE 6888,48+I:RETURN
```

Datos de sprites

890 DATA 1,19,17,15,7,3,1,3,2,6,255  
 ,255,255,127,63,31,128,200,136,240,22  
 4,192,128,192,64,96,255,255,255,254,2  
 52,248

900 DATA 0,15,23,18,114,149,181,255  
 ,248,231,95,63,56,0,0,0,0,255,69,21,7  
 5,101,21,255,0,255,254,254,14,0,0,0

## Tabla de claves

5	= 192	250	= 2550	500	= 131
10	= 904	260	= 743	510	= 829
20	= 143	270	= 1363	520	= 3230
30	= 1637	280	= 4439	530	= 2988
40	= 1541	290	= 260	540	= 1362
50	= 1712	300	= 2071	550	= 142
60	= 143	310	= 8700	560	= 4440
70	= 143	320	= 3866	570	= 883
80	= 2571	330	= 7800	580	= 2722
90	= 2998	340	= 3269	590	= 1225
100	= 3494	350	= 2594	600	= 2001
110	= 3659	360	= 1530	610	= 1225
120	= 4194	370	= 166	620	= 1947
130	= 3502	380	= 2163	630	= 1225
140	= 2770	390	= 998	640	= 1806
150	= 2207	400	= 2810	650	= 1204
160	= 1074	410	= 2464	660	= 142
170	= 411	420	= 520	670	= 2491
180	= 5475	430	= 2165	680	= 2857
190	= 3845	440	= 2464	690	= 1193
200	= 1718	450	= 520	700	= 3854
210	= 2787	460	= 2097	710	= 8267
220	= 4977	470	= 2464	720	= 1380
230	= 879	480	= 520	730	= 7076
240	= 1490	490	= 3462	740	= 401

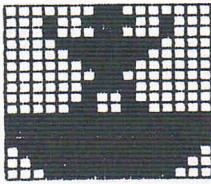


750 = 3130  
760 = 5261  
770 = 1267  
780 = 2976  
790 = 2028  
800 = 142

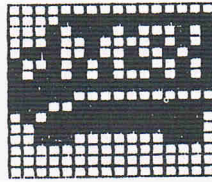
810 = 2009  
820 = 2907  
830 = 2014  
840 = 142  
850 = 3739  
860 = 142

870 = 2772  
880 = 2824  
890 = 5369  
900 = 4926  
TOTAL = 213133

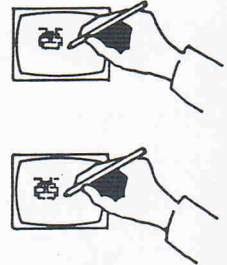
## Diseño de los sprites



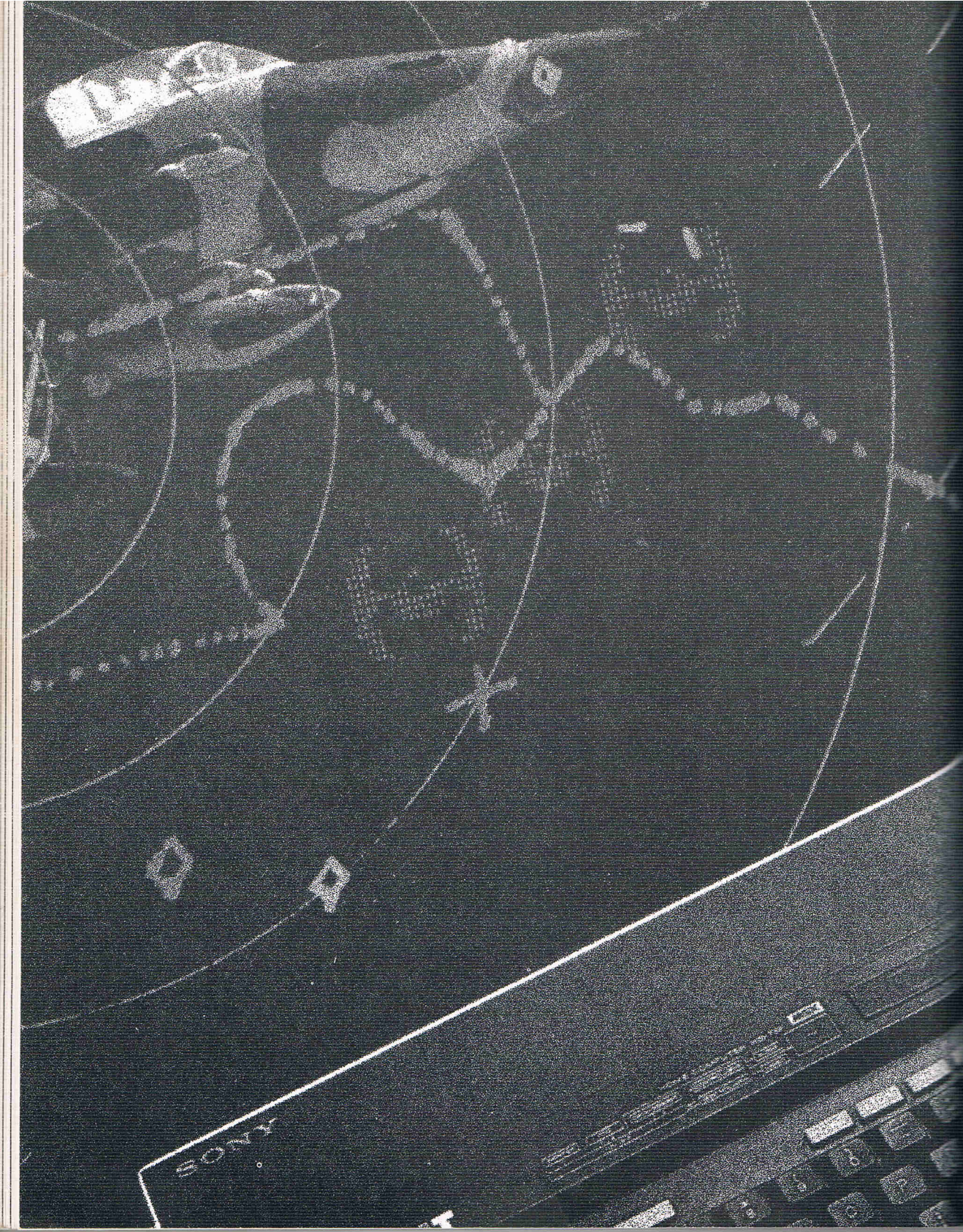
HOMBRE EN LA BARCA



CAMION



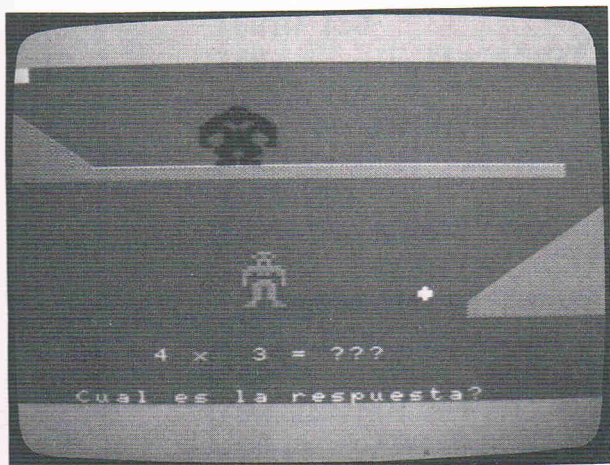






# El gorila matemático

# 15



Tipo de juego:

EDUCATIVO

Godofredo está ansioso de subir la colina y encontrar el gorila matemático. Cuando respondas una pregunta correctamente, la roca que cae no golpeará a Godofredo, llevándolo al principio, y éste se acercará al gorila.

Hay dos niveles de dificultad y las preguntas son sobre suma y multiplicación. Cuando tengas la respuesta pulsa <RETURN>.

# SUGERENCIAS AL PROGRAMA

Quizá puedas animar al gorila mientras cae la roca, o que estreche la mano a Godofredo cuando se encuentren, o un grito agudo cuando la roca golpea a Godofredo...

## Variables del programa

C	Centinela.
ND	Nivel de dificultad.
P	Pregunta (1-2).
MR	Modelos de rocas (8 ó 12).
N1, N2	Números a sumar/multiplicar.
NC	Nuevo centinela.
RJ	Respuesta del jugador.
VR	Velocidad de la roca.
M\$	Mensaje.

### Inicialización

```
10      COLOR 15,4,7:SCREEN 1,3:KEY  
OFF:ON INTERVAL=1 GOSUB 30:INTERVAL 0  
N  
20      GOTO 40  
30      D=RND(1):RETURN  
40      FOR I=6145 TO 6148:VPOKE I,2  
3:NEXT:VPOKE 6144,24:VPOKE 6175,25:VP  
OKE 6880,26:VPOKE 6911,27:FOR I=6176  
TO 6848 STEP 32:VPOKE I,22:VPOKE I+31  
,22:NEXT:FOR I=6881 TO 6910:VPOKE I,2  
3:NEXT  
50      PRINT ".....G_D_R_I_L_L_A"  
:PRINT:PRINT ".....M_A_T_E_M_A_T_I_C  
_D":FOR I=1 TO 9:PRINT:NEXT  
60      FOR I=1 TO 32:READ Q:A$=A$+CH  
R$(Q):NEXT:SPRITE$(0)=A$:A$=""  
70      FOR I=1 TO 32:READ Q:A$=A$+CH  
R$(Q):NEXT:SPRITE$(1)=A$:A$=""  
80      FOR I=1 TO 8:READ Q:A$=A$+CHR  
$(Q):NEXT:SPRITE$(2)=A$:A$=""  
90      FOR I=1 TO 8:READ Q:A$=A$+CHR
```



```

$(Q):NEXT:SPRITE$(3)=A$
100   FOR I=1088 TO 1119:READ Q:VP
OKE I,Q:NEXT
110   PUT SPRITE 1,(116,30),1
120   INPUT "Quieres sumar(s) o
multiplicar(m) numeros";X$
130   IF X$="m" OR X$="M" THEN P=2
:GOTO 150
140   IF X$="s" OR X$="S" THEN P=1
ELSE 120
150   PRINT:INPUT "NIVEL_DE_DIFICU
LTAD_(1_O_2)";X$:IF X$<"1" OR X$>"2"
THEN 150
160   ND=VAL(X$)
170   CLS:FOR I=6374 TO 6397:VPOKE
I,138:NEXT:FOR I=6688 TO 6712:VPOKE
I,138:NEXT
180   VPOKE 8209,228:PUT SPRITE 1,
(92,23):PUT SPRITE 0,(0,104),9
190   FOR I=0 TO 5:VPOKE 6176+33*I
,136:NEXT:FOR I= 0 TO 6:VPOKE 6495+31
*I,137:NEXT
200   FOR I=6208 TO 6368 STEP 32:F
OR J=0 TO K:VPOKE I+J,139:NEXT:K=K+1:
NEXT
210   K=0:FOR I=6527 TO 6719 STEP
32:FOR J=-K TO 0:VPOKE I+J,139:NEXT:K
=K+1:NEXT
220   INTERVAL OFF:SPRITE ON:ON SP
RITE GOSUB 520

```

Imprimir pregunta

```

230   CD=0:NC=0:MR=8:RJ=0:PUT SPRIT
E 2,(0,1)
240   M$="Cual es la respuesta?":GO
SUB 590
250   IF P<>1 THEN 280
260   IF ND=1 THEN N1=INT(RND(1)*10

```

```

+1):N2=INT(RND(1)*10+1) ELSE N1=INT(R
ND(1)*20+1):N2=INT(RND(1)*20+1)
270     K=43:GOTO 300
280     IF ND=1 THEN N1=INT(RND(1)*5+
1):N2=INT(RND(1)*5+1) ELSE N1=INT(RND
(1)*10+1):N2=INT(RND(1)*10+1)
290     K=120
300     FOR I=1 TO 20:PRINT:NEXT:PRIN
T TAB(6);:PRINT USING"##";N1;:PRINT T
AB(9);CHR$(K);TAB(11);:PRINT USING"##
";N2;:PRINT TAB(14);"=_???" ;CHR$(11);
310     C=0

```

Control

```

320     GOSUB 350:IF NC=1 THEN 230
330     GOTO 420
340     GOTO 320

```

Movimiento de roca

```

350     IF ND=1 THEN VR=2 ELSE VR=3
360     K1=VPEEK(6921):K2=VPEEK(6920)
:IF K1/4<>INT(K1/4) THEN IF MR=8 THEN
MR=12 ELSE MR=8
370     VPOKE 6922,MR:IF K1<49 AND K2
<46 THEN VPOKE 6921,K1+VR:VPOKE 6920,
K2+VR:RETURN
380     IF K2<50 AND K1<240 THEN VPOK
E 6921,K1+VR:IF K1>62 AND K1<125 THEN
VPOKE 6916,0:RETURN ELSE VPOKE 6916,
23:RETURN
390     IF K2<80 THEN VPOKE 6920,K2+V
R:RETURN
400     IF K2<125 THEN VPOKE 6920,K2+
VR:VPOKE 6921,K1-VR:RETURN
410     IF K1<5 THEN NC=1:RETURN ELSE
VPOKE 6921,K1-VR:RETURN

```



Leer

```
420 X$=INKEY$:IF X$="" THEN 320
430 IF(X$<"0" OR X$>"9") AND X$<>
CHR$(13) THEN 320
440 IF X$=CHR$(13) THEN 460 ELSE
IF CD=0 THEN VPOKE 6802,32:VPOKE 6803
,32:VPOKE 6804,32
450 CD=CD+1:IF CD>3 THEN 460 ELSE
RJ=RJ*10+VAL(X$):VPOKE 6801+CD,48+VA
L(X$):GOTO 320
```

Comprobar respuesta

```
460 IF CD=0 THEN 320 ELSE M$="
":GOSUB 590:IFP<>1 T
HEN 490
470 IF N1+N2<>RJ THEN 320
480 GOTO 500
490 IF N1*N2<>RJ THEN 320
500 IF NC=1 THEN 320 ELSE C=1:SPR
ITE OFF:K1=VPEEK(6920):K2=VPEEK(6921)
:IF K1<50 THEN VPOKE 6912,23:IF K2<20
0 THEN VPOKE 6913,K2 ELSE VPOKE 6913,
200:PUT SPRITE 2,(100,200):NC=1 ELSE
IF K2>195 THEN VPOKE 6913,165 ELSE VP
OKE 6913,K2
510 SPRITE ON:GOTO 320
```

Choque de sprites

```
520 SPRITE OFF:IF C=1 THEN GOSUB
610:RETURN
530 IF VPEEK(6912)<70AND VPEEK(69
13)<115 THEN 560
540 FOR I2=1 TO 10:VPOKE6912,VPEE
```

```
K(6912)-10:PLAY"m380s8164n56":FOR I1=
1 TO 100:NEXT:VPOKE 6912,VPEEK(6912)+
10:PLAY "m700164s8n50":NEXT
550 C=0:PUT SPRITE 0,(0,104):PUT
SPRITE 2,(0,0):NC=1:SPRITE ON:RETURN
```

Fin de juego

```
560 SCREEN 1:FOR I=1 TO 10:PRINT:
NEXT:PRINT "VIVA!!!!_ _has_encontrado
_lal":PRINT "Gorila_matematico"
570 PLAY "m255s118o5edco4bo5co4b5
agao5co4bo5dl1c"
580 END
```

Imprimir mensaje

```
590 FOR I=1 TO 23:PRINT:NEXT:PRINT
"___";M$;CHR$(11);
600 RETURN
```

Salto del jugador

```
610 PLAY "164n60":NC=1:C=0:K3=VPEE
K(6913):K4=VPEEK(6912):IF K4>50 THEN
IF K3<165 THEN VPOKE 6913,K3+20:SPRIT
E ON:RETURN ELSE VPOKE 6912,23:SPRITE
ON:RETURN
620 VPOKE 6913,K3-30:SPRITE ON:RET
URN
```

Datos de sprites

```
630 DATA 1,2,1,1,7,15,13,8,9,11,3,3
,3,3,6,14,192,160,192,64,240,248,216,
```



8,200,232,96,96,96,96,48,56  
 640 DATA 3,5,31,59,113,248,237,199,  
 135,135,207,30,28,12,6,30,128,64,240,  
 184,28,62,110,198,194,194,230,240,112  
 ,96,192,240  
 650 DATA 96,240,96,0,0,0,0,0  
 660 DATA 64,224,224,64,0,0,0,0

Datos de sprites pequeños

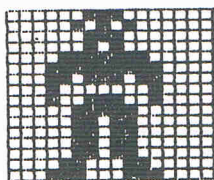
670 DATA 128,64,160,80,168,84,170,8  
 5  
 680 DATA 1,2,5,10,21,42,85,170  
 690 DATA 255,170,85,170,85,170,85,1  
 70  
 700 DATA 85,170,85,170,85,170,85,17  
 0

## Tabla de claves

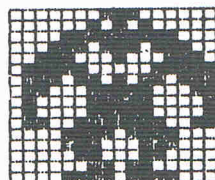
10	= 3926	180	= 2126	350	= 1832
20	= 191	190	= 4144	360	= 5875
30	= 997	200	= 4022	370	= 3917
40	= 7902	210	= 4558	380	= 5371
50	= 3986	220	= 2154	390	= 1856
60	= 3659	230	= 2460	400	= 2799
70	= 3660	240	= 2670	410	= 2604
80	= 3671	250	= 972	420	= 1524
90	= 3206	260	= 8525	430	= 3093
100	= 1799	270	= 626	440	= 3910
110	= 803	280	= 8499	450	= 5743
120	= 4582	290	= 449	460	= 3639
130	= 2349	300	= 6322	470	= 1566
140	= 2354	310	= 323	480	= 396
150	= 4041	320	= 1311	490	= 1568
160	= 993	330	= 316	500	= 10831
170	= 3774	340	= 216	510	= 622

520	=	1628	590	=	2276	660	=	1148
530	=	2449	600	=	142	670	=	1484
540	=	6718	610	=	7676	680	=	1151
550	=	2805	620	=	1317	690	=	1531
560	=	5812	630	=	4562	700	=	1484
570	=	3497	640	=	5618			
580	=	129	650	=	1052	TOTAL	=	211211

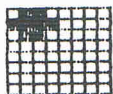
## Diseño de los sprites



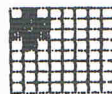
HOMBRE



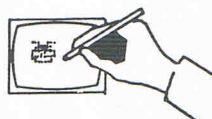
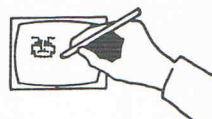
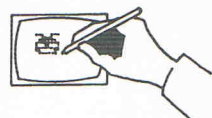
GORILA



ROCA:  
POSICION 1



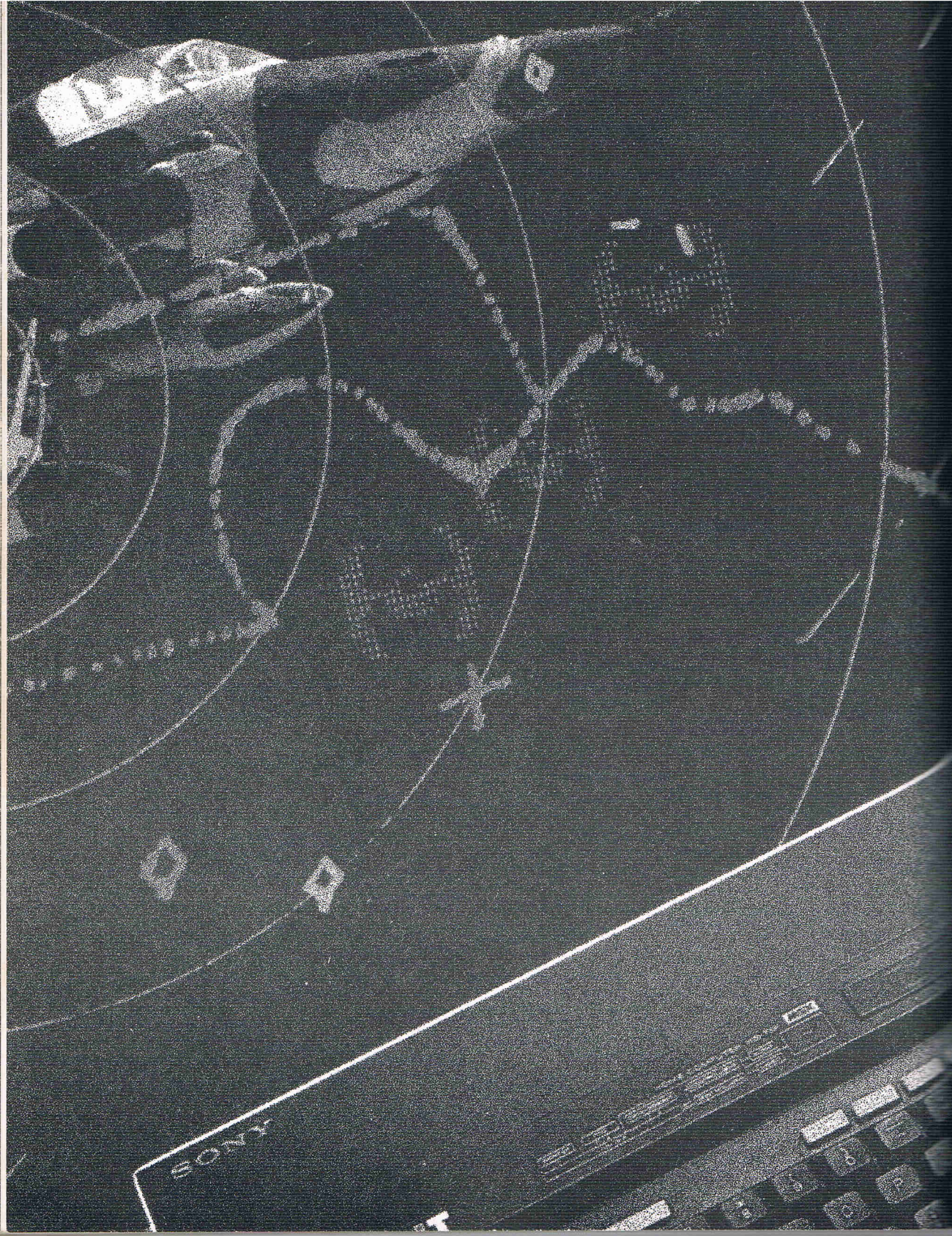
ROCA:  
POSICION 2









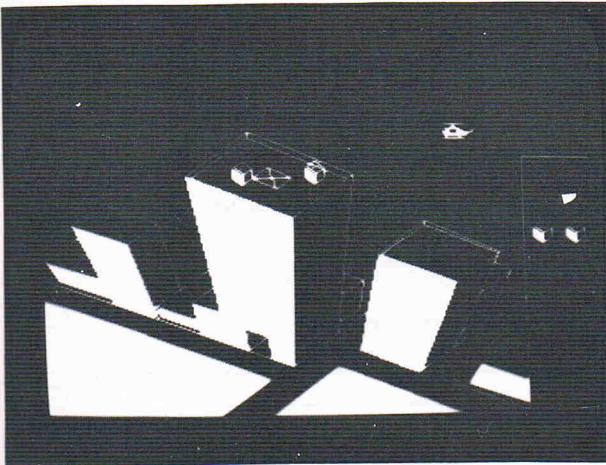




# Rescate aéreo 16

## ¡ATENCIÓN!

Este programa utiliza la subrutina SOPORTE en lenguaje máquina. Véase apéndice C.



**Tipo de juego:**

**JUEGO DE SIMULACION**

Un rascacielos necesita, desesperadamente, unos paquetes especiales, los cuales debes recoger con tu helicóptero de los edificios más bajos y llevarlos al rascacielos. En éste no puedes aterrizar, ya que no hay helipuerto, y podría producir un accidente.

Debes recoger cuatro paquetes. Tus puntos son:

Objetivo alcanzado	100.
Edificio alcanzado	50
Recogida satisfactoria	10

Cuanto más rápido trabajes, mejor será tu puntuación. Utiliza la barra espaciadora para soltar el paquete.





```

HR$(Q):NEXT:SPRITE$(0)=A$:PUT SPRITE
0,(240,0),10:A$=""
140     FOR I=1 TO 32:READ Q:A#=A#+C
HR$(Q):NEXT:FOR I=1 TO 4:SPRITE$(I)=A
$:NEXT:A$="":PUT SPRITE 1,(226,70),14
:PUT SPRITE 2,(241,70),14:PUT SPRITE
3,(226,100),14:PUT SPRITE 4,(241,100)
,14
150     FOR I=1 TO 32:READ Q:A#=A#+C
HR$(Q):NEXT:SPRITE$(5)=A$:A$=""
160     FOR I=1 TO 32:READ Q:A#=A#+C
HR$(Q):NEXT:SPRITE$(6)=A$:A$=""
170     FOR I=1 TO 32:READ Q:A#=A#+C
HR$(Q):NEXT:SPRITE$(7)=A$
180     DEFUSR=A+10:DEFUSR1=A+128:PO
KE V4VVV,15:POKE V3VVV,0:PD=4:DR=2:VH
=1:TIME=0:C=1
190     STRIG(0) ON:ON STRIG GOSUB 4
50

```

Control

```

200     POKE V1VVV,VH:D=USR(D)
210     GOSUB 250
220     GOSUB 300
230     C=1
240     GOTO 200

```

Movimiento del helicóptero

```

250     IF RND(1)<.7 THEN K=DR:J=1:GO
TO 280
260     J=4:IF RND(1)<.5 THEN K=1:GOT
O 280
270     K=3
280     POKE V1VVV,J:POKE V2VVV,K:D=U
SR1 ( D )
290     RETURN

```

```
300      K=VPEEK(6912):IF K<4 OR K>191
        THEN VPOKE 6912,4:RETURN
```

Comprobar posición

```
310      IF K>160 THEN 410
320      IF K<30 THEN RETURN
330      X=VPEEK(6913):IF X>209 THEN V
        POKE 6913,209:IF K>160 THEN 410 ELSE
        RETURN
340      IF X<150 THEN 350 ELSE IF K<8
        0 THEN RETURN ELSE IF X>200 THEN 410
        ELSE IF X<165 THEN RETURN ELSE IF C=1
        THEN 410 ELSE M2=M2+1:GOTO 380
350      IF K>140 THEN 410 ELSE IF X<6
        0 THEN 360 ELSE IF K>50 THEN RETURN E
        LSE 410
360      IF K<60 THEN RETURN ELSE IF K
        >100 THEN 410 ELSE IF K>80 THEN RETUR
        N ELSE IF C=1 THEN 410 ELSE M1=M1+1:G
        OTO 380
370      RETURN
```

Recoger paquete

```
380      C=1:PUT SPRITE PD,(235,130):P
        UT SPRITE 6,(233,160),15:DR=0:PT=PT+1
        0:VPOKE 6912,VPEEK(6912)-2
390      IF M1>2 OR M2>2 THEN 430
400      RETURN
```

Choque

```
410      PUT SPRITE 6,(100,200):PUT SP
        RITE 7,(233,160),15:VPOKE 6914,20:FDR
        I=1 TO 30:PLAY"L60M1000S14N50":NEXT:
```



```
FOR I=1 TO 2500:NEXT:PUT SPRITE 7,(10
0,200)
420 GOTO 520
```

Edificio incorrecto

```
430 SCREEN 1:PRINT "DEBES COGER D
OS PAQUETES":PRINT "DE CADA EDIFICIO"
:FOR I=1 TO 3000:NEXT
440 PRINT "Intentalo otra vez":IF
INKEY$="" THEN END ELSE 440
```

Dejar paquete

```
450 IF C=0 THEN END
460 PUT SPRITE6,(100,200):PLAY"18
m1000s14n40":C=0:DR=2:I1=VPEEK(6913):
IF I1<150 THEN 470 ELSE IF I1<170 THEN
PUT SPRITEPD,(I1,160):PD=PD-1:GOTO500
ELSE IF I1<200 THEN PUT SPRITEPD,(I
1,90):PD=PD-1:GOTO500 ELSE PUT SPRITE
PD,(I1,160):PD=PD-1:GOTO500
470 IF I1>140 THEN PUT SPRITE PD,
(I1,90):PD=PD-1:GOTO 500 ELSE IF I1<6
0 THEN 490 ELSE IF I1<72 THEN PUT SPRI
TE PD,(I1,145):PD=PD-1:GOTO500 ELSE I
F I1<100 OR I1>106 THEN PUT SPRITE PD,
(I1,32):PT=PT+50:PD=PD-1:GOTO 500
480 PUT SPRITE PD,(I1,32):PT=PT+1
00:PLAY "m1100s813n50":PD=PD-1:GOTO 5
00
490 IF I1<22 OR I1>55 THEN PUT SP
RITE PD,(I1,110):PD=PD-1:GOTO 500 ELS
E PUT SPRITEPD,(I1,70):PD=PD-1
500 IF PD=0 THEN FOR I1=1 TO 2000
:NEXT:GOTO 520
510 RETURN
```

Fin de juego

```
520     SCREEN 1:PRINT"_____FIN_
DE_JUEGO":PRINT:PRINT:PRINT
530     PRINT "___PUNTOS:___";PT
540     IF PD<>0 THEN 560 ELSE K=150-
TIME/60:IF K<0 THEN K=0
550     PRINT:PRINT:PRINT"Bonos_por_1
os_paquetes:";INT(8*K):PRINT:PRINT "
___TOTAL_:___";PT+INT(8*K)
560     IF INKEY$="" THEN END ELSE 56
0
```

Datos del dibujo

```
570     DATA 100,14,100,18,100,14,148,4
1,100,14,72,41,100,18,145,44,100,18,7
5,42,120,68,148,41,120,68,72,41,120,6
8,120,160,72,41,91,145,148,41,140,140
580     DATA 0,100,120,160,0,112,110,16
8,88,190,110,168,110,190,140,160,140,
160,195,190,120,160,153,127,149,150,1
76,165,186,169,225,190,186,169,225,13
1,176,165,225,118
590     DATA 98,136,109,142,98,136,100,
150,100,150,108,142,109,142,110,155,1
02,43,120,43,102,43,113,50,102,43,109
,36,113,50,120,43,113,50,109,36,109,3
6,120,43
600     DATA 180,70,159,91,180,70,214,9
0,180,73,161,91,180,73,213,91,180,70,
180,73,159,91,194,109,159,91,150,150,
194,109,177,164,214,90,192,149,194,10
9,215,90
610     DATA 79,139,75,120,75,120,87,12
6,75,120,84,111,78,137,61,128,61,128,
59,122,59,122,76,130,59,122,82,100,60
,130,46,83,46,83,21,70,46,83,59,70
```



620 DATA 59,70,35,57,35,57,21,70,21  
 ,70,40,119,40,117,16,105,16,105,10,93  
 ,10,93,35,105,10,93,24,78,0,190,88,19  
 0,110,190,195,190,59,70,70,112  
 630 DATA 143,103,152,107,152,107,15  
 0,125,150,125,140,134,142,116,152,107  
 ,205,116,219,102,219,102,212,97,219,1  
 02,209,128,209,128,195,141,198,172,22  
 5,186,198,172,204,160,204,160,225,171  
 ,204,160,217,146,217,146,225,150  
 640 DATA 225,190,225,30,225,30,255,  
 30

Datos de sprites

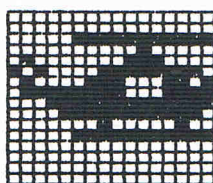
650 DATA 0,0,255,4,14,127,177,241,2  
 55,127,32,255,0,0,0,0,0,224,0,2,133  
 ,194,254,248,240,128,224,0,0,0,0  
 660 DATA 1,2,4,8,21,26,29,30,31,31,  
 31,15,7,3,1,0,128,64,32,16,8,24,40,20  
 0,136,136,136,144,160,192,128,0  
 670 DATA 0,4,66,136,2,64,18,42,10,1  
 65,43,144,40,1,64,18,128,16,4,32,1,72  
 ,33,146,204,22,200,92,64,9,0,32  
 680 DATA 255,255,255,227,221,190,19  
 0,190,190,190,221,227,255,255,255,255  
 ,255,255,255,187,183,175,159,175,183,  
 187,187,187,255,255,255,255  
 690 DATA 192,160,198,170,170,198,0,  
 0,0,0,0,255,0,0,0,0,0,0,198,170,170,1  
 66,2,2,10,4,0,255,0,0,0,0

## Tabla de claves

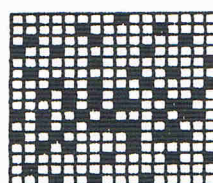
5	= 192	40	= 1541	80	= 2571
10	= 904	50	= 1712	90	= 2998
20	= 143	60	= 143	100	= 676
30	= 1637	70	= 143	110	= 3180

120	=	5658	320	=	859	520	=	2215
130	=	4622	330	=	3791	530	=	1239
140	=	8844	340	=	6973	540	=	3604
150	=	3664	350	=	3576	550	=	6152
160	=	3665	360	=	5560	560	=	1312
170	=	3200	370	=	142	570	=	6823
180	=	5728	380	=	5111	580	=	7719
190	=	1640	390	=	1560	590	=	7214
200	=	1482	400	=	142	600	=	7335
210	=	405	410	=	6633	610	=	6772
220	=	200	420	=	161	620	=	6530
230	=	324	430	=	4607	630	=	10272
240	=	351	440	=	3503	640	=	1506
250	=	2379	450	=	809	650	=	4741
260	=	2218	460	=	14414	660	=	4711
270	=	334	470	=	12523	670	=	4727
280	=	2139	480	=	4283	680	=	6483
290	=	142	490	=	5656	690	=	4383
300	=	2892	500	=	2138			
310	=	1014	510	=	142	TOTAL	=	243062

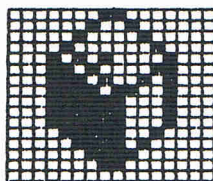
## Diseño de los sprites



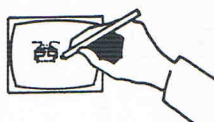
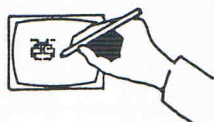
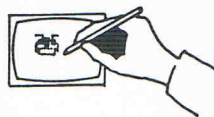
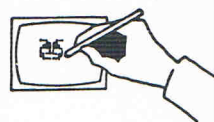
HELICOPTERO



EXPLOSION



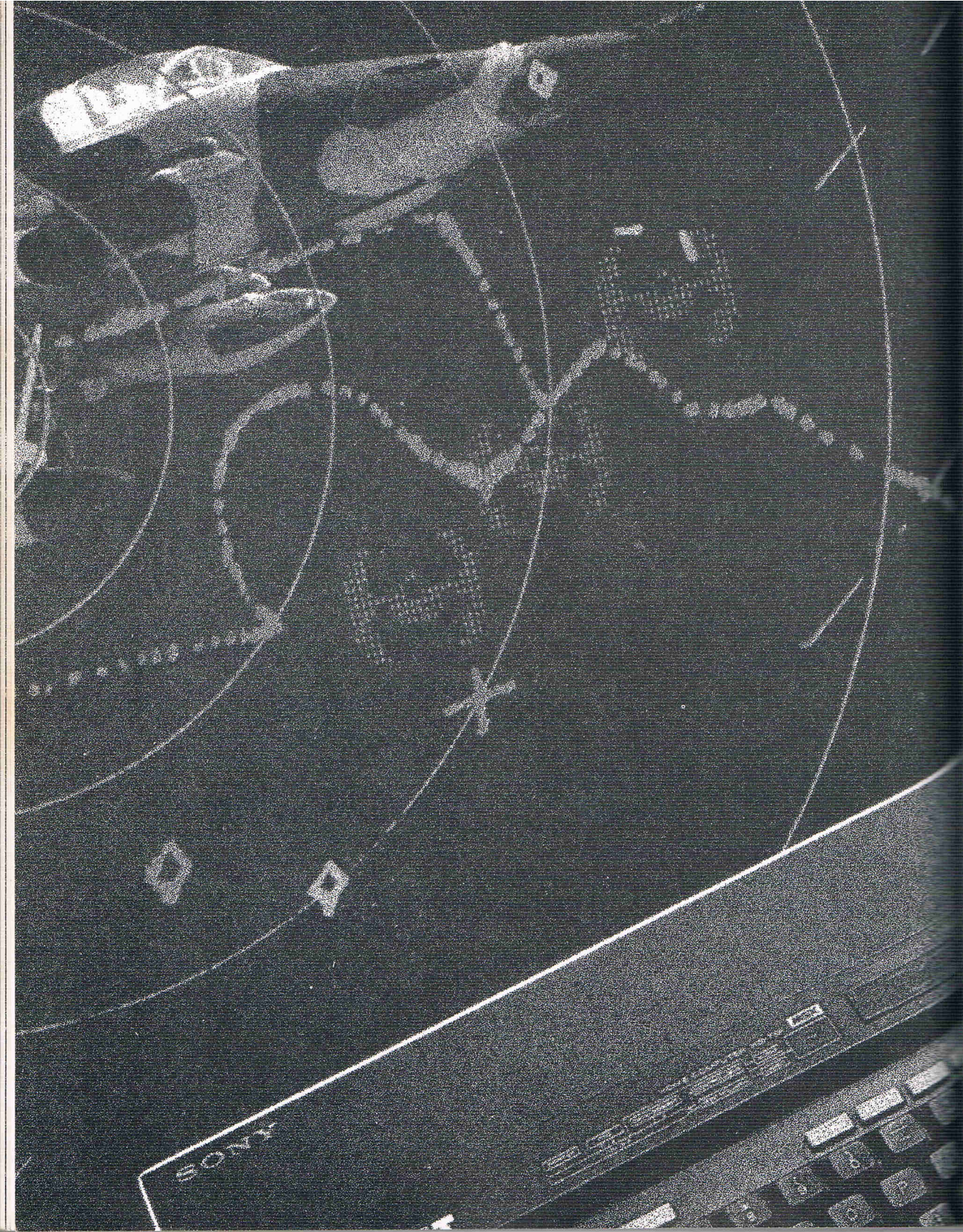
PAQUETE











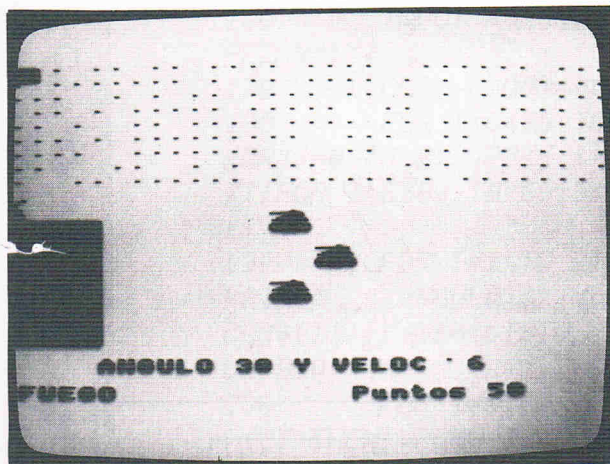


# Emboscada de tanques

# 17

## ¡ATENCIÓN!

Este programa utiliza la subrutina SOPORTE en lenguaje máquina. Véase apéndice C.



Tipo de juego:

**JUEGO DE SIMULACION**

Se ha preparado una emboscada a un escuadrón de tanques enemigos. Tu tanque está situado en un risco, desde donde disparas al enemigo que viene por la derecha. Con las teclas de 0 a 9 eliges el ángulo y velocidad de disparo; después pulsa la barra espaciadora para disparar. También puedes lanzar una bengala para poder ver los tanques enemigos pulsando sólo la barra espaciadora.

Cada tanque abatido te suma 50 puntos en el marcador. El juego termina cuando un tanque alcanza la base del risco.

## SUGERENCIAS AL PROGRAMA

Creando otros sprites y modificando la rutina "movimiento de tanques enemigos" se puede añadir más tanques.

La dificultad del juego la determina, por otra parte, la distancia entre los tanques, así que si se modifican las líneas 540 a 580 se puede obtener efectos interesantes.

## Variables del programa

NP, PT	Número de pantalla; puntuación.
M\$	Mensaje.
OP	¿Otra pantalla?
TM	Ultimo tanque movido.
F	¿Fuego?
A, V	Angulo; velocidad.
T	Tiempo.
XI, YI	Coordenadas iniciales de la bomba.
X, Y	Coordenadas calculadas.

Inicialización

5 BEEP

```
10      REM      ** ATENCION **
20      REM
30      REM ESTE PROGRAMA NECESITA
40      REM LA SUBROUTINA SOPORTE
50      REM Vease el Apendice C
60      REM
70      REM
80      A=PEEK(64586!)+256*PEEK(6458
7!)
90      V1=A+9:V2=A+8:V3=A+7:V4=A+6
100     SCREEN 1,2: KEY OFF: PRINT T
AB(5)"EMBOSCADA_DE_TANQUES": PRINT: P
RINT: PRINT: PRINT
110     FOR I=1 TO 32: READ Q: A#=A$
+CHR$(Q): NEXT: SPRITE$(0)=A$: A$=""
120     FOR I=1 TO 32: READ Q: A#=A$
+CHR$(Q): NEXT: SPRITE$(1)=A$: SPRITE
```



```

$(2)=A$: SPRITE $(3)=A$: SPRITE$(4)=A
$: A$=""
130   FOR I=1 TO 8: READ Q: A$=A$+
CHR$(Q): NEXT: SPRITE$(5)=A$: A$=""
140   FOR I=1 TO 8: READ Q: A$=A$+
CHR$(Q): NEXT: SPRITE$(6)=A$
150   DEFUSR=A+128: POKE V2VVV,1
160   INPUT"PULSA 'RETURN' PARA EM
PEZAR":X$
170   CLS: COLOR 14,1,1
180   FOR I=6496 TO 6752 STEP 32:
FOR J=0 TO 5: VPOKE I+J,255: NEXT: NE
XT:FOR I=6784 TO 6815: VPOKE I,23: NE
XT
190   FOR I=1 TO 21: PRINT: NEXT:
PRINT"____ANGULO____Y_VELOC.": PRINT
CHR$(11);: M$="LISTO,_____": GDS
UB 810
200   FOR I=6144 TO 6431: IF RND(1
)<.7 THEN VPOKE I,46
210   NEXT: FOR I=36 TO 375: VPOKE
I,0: NEXT:VPOKE 369,1

```

Nuevo juego

```

220   PUT SPRITE 0,(2,74),10: TM=1
: F=0: K1=35*(TIME/10-INT(TIME/10))+6
0: PUT SPRITE 1,(235,K1),1
230   PUT SPRITE 2,(255,K1+20),1:
PUT SPRITE 3,(235,K1+40),1: PUT SPRIT
E 4,(255,K1+55),1
240   DP=0: SPRITE ON: ON SPRITE G
DSUB 650

```

Control

```

250   IF F=0 THEN 270
260   GDSUB 410

```

```

270      GOSUB 510: IF OP=1 THEN NP=N
P+1: GOTO 220
280      X$=INKEY$: IF X$="" OR F<>0 T
HEN 250
290      IF X$=" " THEN M$="BENGALA
": GOSUB 810: F=2: A=.628: V=9:
GOSUB 410: GOTO 250
300      IF X$<"0" OR X$>"9" THEN 280
310      K1=VAL(X$): VPOKE 6829,K1+48
320      X$=INKEY$: IF X$="" THEN GOSU
B 510: GOTO 320
330      IF X$<"0" OR X$>"9" THEN 320
340      A=10*K1+VAL(X$): VPOKE 6830,V
AL(X$)+48: A=A*.01745
350      X$=INKEY$: IF X$="" THEN GOSU
B 510: GOTO 350
360      IF X$<"0" OR X$>"9" THEN 350
370      V=VAL(X$): VPOKE 6841,V+48
380      F=0: X$=INKEY$: IF X$="" THEN
GOSUB 510: GOTO 380
390      IF X$=" " THEN M$="FUEGO
": GOSUB 810: F=1: GOSUB 410: G
OTO 250
400      GOTO 380

```

Movimiento de bomba

```

410      T=T+3: XI=V*CDOS(A)/2.1: YI=V*
SIN(A)/2.1
420      IF T<>3 AND T<>0 THEN 440
430      IF F=0 THEN RETURN ELSE PUT S
PRITE 5,(20,55),15: PLAY"m80s1164n30n
20"
440      X=INT(20+T*XI)
450      Y=INT(55-T*YI+.049*T*T)
460      IF X<8 OR X>247 OR Y>154 THEN
GOSUB 730: RETURN
470      IF Y<8 THEN PUT SPRITE 5,(100
,210) ELSE PUT SPRITE 5,(X,Y),15

```



```
480 IF F<>2 THEN 500
490 IF T>=30 THEN 620
500 RETURN
```

Movimiento de tanques enemigos

```
510 SPRITE OFF: TM=TM+1: IF TM=5
THEN TM=1
520 IF VPEEK(6912+TM*4)>200 THEN
RETURN
530 POKE V3VVV, TM: ON NF GOTO 540
, 550, 550, 560, 560, 570, 580
540 K=1: GOTO 590
550 K=2: GOTO 590
560 K=3: GOTO 590
570 K=4: GOTO 590
580 K=5
590 POKE V1VVV, K: D=USR(D)
600 IF VPEEK(6913+TM*4)<41 THEN 7
70
610 SPRITE ON: RETURN
```

Lanzallamas

```
620 PUT SPRITE 5, (100, 210): FOR I
=1 TO 5: COLOR 1, 15, 15: FOR J=1 TO 70
: NEXT: COLOR 14, 1, 1: FOR J=1 TO 70:
NEXT: NEXT
630 M$="LISTO " : GOSUB
810: F=0: T=0: VPOKE 6829, 32: VPOKE 6
830, 32: VPOKE 6841, 32
640 RETURN
```

Explosión tanque enemigo

```
650 IF F<>1 THEN RETURN ELSE VPOK
E 6934, 24: COLOR 1, 15, 15: PLAY"m13000
```

```

s118n40n35n40n35": FOR I1=1 TO 200: N
EXT: COLOR 14,1,1
660     I2=0: FOR I1=1 TO 4: IXD=VPEE
K(6913+I1*4)-VPEEK(6933): IYD=VPEEK(6
912+I1*4)-VPEEK(6932): IF IXD<0 THEN
IXD=-IXD
670     IF IYD<0 THEN IYD=-IYD
680     IF VPEEK(6933)+6>=VPEEK(6913+
I1*4) AND IXD<=16 AND VPEEK(6932)+ 6>
= VPEEK(6912+I1*4) AND IYD<=14 THEN I
2=I1
690     NEXT: VPOKE 6914+4*I2,24: FOR
I1=1 TO 200: NEXT
700     P=P+50: VPOKE 6914+4*I2,4*I2:
VPOKE 6934,20: PUT SPRITE 5,(100,210
): PUT SPRITE 12,(I2*25-25,210): F=0:
T=0: M$="LISTO           ": GOSUB 81
0
710     IF VPEEK(6916)>200 AND VPEEK(
6920)>200 AND VPEEK(6924)>200 AND VPE
EK(6928)>200 THEN OP=1
720     VPOKE 6829,32: VPOKE 6830,32:
VPOKE 6841,32: RETURN

```

Explosión de bomba

```

730     VPOKE 6934,24: T=0:F=0: M$="L
ISTO           ": GOSUB 810
740     PLAY"m80164s8n10n12": VPOKE 6
829,32: VPOKE 6830,32: VPOKE 6841,32:
FOR I=1 TO 400: NEXT
750     VPOKE 6934,20: PUT SPRITE 5,(
100,210)
760     RETURN

```

Fin de juego

```

770     SCREEN 1: PRINT TAB(8)"FIN_DE
L_JUEGO": PRINT: PRINT: PRINT: PRINT

```



```

780 PRINT "Tus_puntos_son:"; P; PRINT
NT
790 PRINT "Has_superado"; NP; PRINT
"oleadas"; PRINT "de_tanques"
800 END

```

Programa/mensaje

```

810 FOR I=1 TO 23: PRINT: NEXT: PRINT M$; TAB(17); "Puntos"; P; CHR$(11); RETURN

```

Datos de sprites

```

820 DATA 0,0,63,63,63,63,127,0,127,
200,135,200,127,0,,,1,7,156,240,192,2
48,254,0,254,19,225,19,254,0,0,0
830 DATA 0,0,255,1,7,31,127,0,127,1
99,146,199,127,0,0,0,0,64,248,248,248
,252,254,0,254,227,73,227,254,0,0,0
840 DATA 0,0,24,60,60,24,0,0
850 DATA 128,4,64,0,144,5,40,134,67
,177,68,16,68,0,128,16,4,0,129,16,6,1
29,146,34,25,162,16,65,0,8,0,129

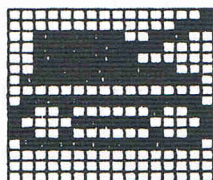
```

## Tabla de claves

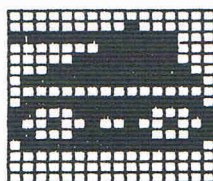
5	= 192	80	= 2571	160	= 2251
10	= 904	90	= 2998	170	= 559
20	= 143	100	= 3525	180	= 4998
30	= 1637	110	= 3659	190	= 4887
40	= 1541	120	= 5862	200	= 2521
50	= 1712	130	= 3674	210	= 2086
60	= 143	140	= 3209	220	= 5426
70	= 143	150	= 1726	230	= 4048

240	= 1522	460	= 2863	680	= 8006
250	= 712	470	= 2659	690	= 2343
260	= 310	480	= 1183	700	= 7362
270	= 2484	490	= 1087	710	= 5535
280	= 2521	500	= 142	720	= 1781
290	= 4579	510	= 2567	730	= 2591
300	= 1610	520	= 2161	740	= 4011
310	= 1926	530	= 1956	750	= 1321
320	= 2129	540	= 621	760	= 142
330	= 1650	550	= 622	770	= 2459
340	= 3839	560	= 623	780	= 2010
350	= 2159	570	= 624	790	= 3945
360	= 1680	580	= 336	800	= 129
370	= 1862	590	= 1399	810	= 3519
380	= 2573	600	= 1882	820	= 4761
390	= 3436	610	= 548	830	= 4941
400	= 276	620	= 4688	840	= 1040
410	= 4064	630	= 3846	850	= 4813
420	= 1963	640	= 142		
430	= 3119	650	= 5491		
440	= 1560	660	= 8029	TOTAL	= 215010
450	= 2658	670	= 1785		

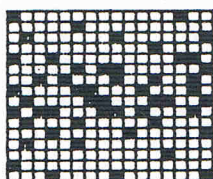
## Diseño de los sprites



TANQUE PROPIO



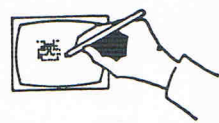
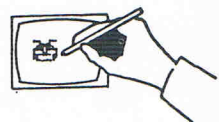
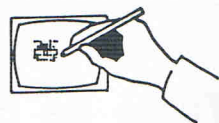
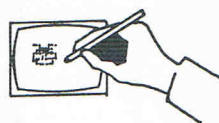
TANQUE ENEMIGO



EXPLOSION



BOMBA





1875

1876

1877

1878

1879

1880

1881

1882

1883

1884

1885

1886

1887

1888

1889

1890

1891

1892

1893

1894

1895

1896

1897

1898

1899

1900

1901

1902

1903

1904

1905

1906

1907

1908

1909

1910

1911

1912

1913

1914

1915

1916

1917

1918

1919

1920

1921

1922

1923

1924

1925

1926

1927

1928

1929

1930

1931

1932

1933

1934

1935

1936

1937

1938

1939

1940

1941

1942

1943

1944

1945

1946

1947

1948

1949

1950

1951

1952

1953

1954

1955

1956

1957

1958

1959

1960

1961

1962

1963

1964

1965

1966

1967

1968

1969

1970

1971

1972

1973

1974

1975

1976

1977

1978

1979

1980

1981

1982

1983

1984

1985

1986

1987

1988

1989

1990

1991

1992

1993

1994

1995

1996

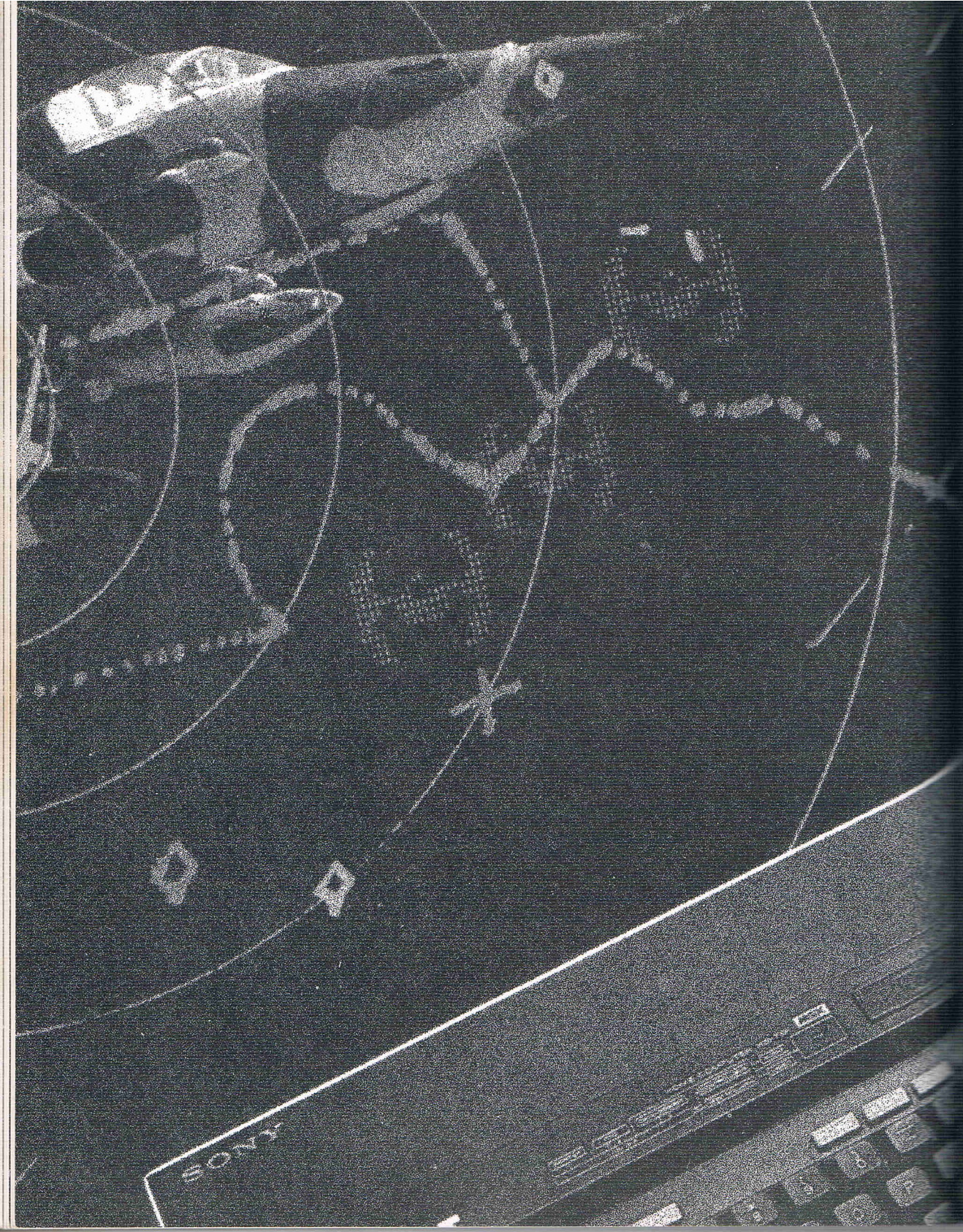
1997

1998

1999

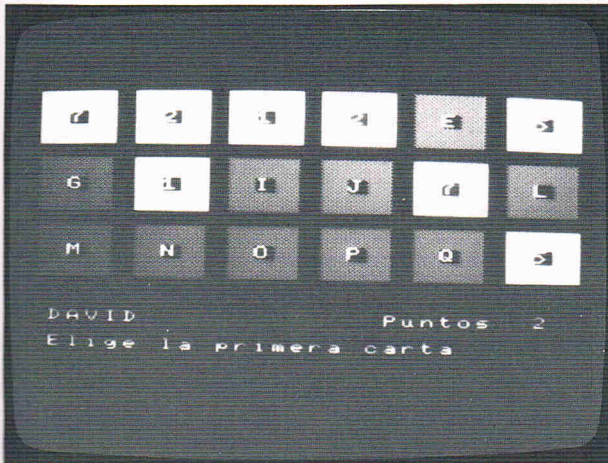
2000







# Concentración 18



Tipo de juego:

**JUEGO MEMORISTICO**

Desafia a tu oponente a una batalla mental. Elige dos cartas entre dieciocho; si ambas tienen el mismo número, en el reverso, tienes un punto; si no, se vuelven otra vez boca abajo y se le da una oportunidad a tu contrario.

¡Que gane el que mejor memoria tenga!

# SUGERENCIAS AL PROGRAMA

Para facilitar el juego puedes usar un rango de números menor. Si, por el contrario, quieres hacerlo más difícil, en vez de tener dos grupos de números, del 1 al 9, pon tres grupos de números del 1 al 6 y haz que cada jugador elija tres cartas.

La presentación en pantalla quedaría mejor poniendo más colores a las cartas.

## Variables del programa

J\$(1), J\$(2) Nombres de los jugadores.  
P(1), P(2) Puntuaciones.  
N(I, J) Números de las cartas.  
JC Jugador en acción.  
CL Carta a levantar.  
UC Ultima carta.

Inicialización

```
10 SCREEN 1: COLOR 15,4,4: KEY 0
FF: PRINT TAB(8) "CONCENTRACION"
20 DIM N(18,2)
30 FOR I= 1072 TO 1143: READ Q:
VPOKE I,Q: NEXT: FOR I= 1152 TO 1223:
READ Q: VPOKE I,Q: NEXT: VPOKE 8208,
164: VPOKE 8209,164: VPOKE 8210,228:
VPOKE 8211,228
40 FOR I=1 TO 5: PRINT : NEXT: P
RINT "Nombre_del_jugador_1?"
50 X%=INKEY$: IF X%="" THEN D=RN
D(1): GOTO 50 ELSE IF X%=CHR$(13) THE
N J$(1)="RICARDO":GOTO 80 ELSE J$(1)=
X%
60 L=LEN(J$(1)): BEEP: VPOKE 640
3+L,ASC(X%): IF L > 15 THEN 80
70 X%=INKEY$: IF X%="" THEN D=RN
D(1): GOTO 70 ELSE IF X%=CHR$(13) THE
N 80 ELSE J$(1)=J$(1)+X%: GOTO 60
80 FOR I=1 TO 5: PRINT : NEXT: P
RINT "Nombre_del_jugador_2?"
```



```

90      X$=INKEY$: IF X$="" THEN 90 E
LSE IF X$=CHR$(13) THEN J$(2)="JULIO"
:GOTO 120 ELSE J$(2)=X$
100     L=LEN(J$(2)):BEEP: VPOKE 659
5+L,ASC(X$): IF L>15 THEN 120
110     X$=INKEY$: IF X$="" THEN 110
ELSE IF X$=CHR$(13) THEN 120 ELSE J$
(2)=J$(2)+X$: GOTO 100
120     FOR I=1 TO 5: PRINT : NEXT:
PRINT "Vale_";J$(1);" -": PRINT: PRIN
T "Tu_primero._PULSA_UNA_TECLA"
130     IF INKEY$="" THEN 130

```

Preparación pantalla

```

140     CLS: FOR CL=1 TO 18: GOSUB 40
0: NEXT
150     JC=1: GOSUB 560: I=0: J=0
160     I=I+1: IF I= 10 THEN I=1: J=J
+1: IF J=2 THEN 190
170     K=INT(RND(1)*18+1): IF N(K,1)
=0 THEN N(K,1)=I ELSE 170
180     GOTO 160

```

Editor

```

190     PRINT CHR$(11);: FOR I=1 TO 2
0: PRINT : NEXT : PRINT"Elige_la_prim
era_carta"
200     X$=INKEY$: IF X$="" OR ((X$<"
A" OR X$>"R") AND (X$<"a" OR X$>"r"))
THEN 200
210     IF X$>"R" THEN CL= ASC(X$)-96
ELSE CL=ASC(X$)-64
220     IF N(CL,2)=1 THEN 200 ELSE GO
SUB 320: UC=CL
230     PRINT "Ahora_la_segunda_"
240     X$=INKEY$: IF X$="" OR ((X$<"

```

```

A" OR X$>"R") AND (X$<"a" OR X$>"r"))
  THEN 240
250   IF X$>"R" THEN I=ASC(X$)-96
  ELSE I=ASC(X$)-64
260   IF I=CL THEN 240 ELSE CL=I: I
  F N(CL,2)=1 THEN 240 ELSE GOSUB 320
270   IF N(CL,1)=N(UC,1) THEN GOSUB
  480: GOTO 290
280   PLAY"14m61680s10n2": PRINT"No
  _latinaste_": FOR T=1 TO 2000: NEXT: G
  OSUB 400: CL=UC : GOSUB 400
290   IF JC=1 THEN JC=2 ELSE JC=1
300   PRINT CHR$(11);: FOR I=1 TO 1
  9: PRINT : NEXT : FOR I=1 TO 3: PRINT
  "_____":NEXT:
  PRINT "_____";
310   GOSUB 560: GOTO 190

```

Levantar carta

```

320   PLAY"s8m1000120n20r30n35": K=
  INT((CL-1)/6)+1: DN K GOTO 330,340,35
  0
330   AD=6172+5*CL: GOTO 360
340   AD=6332+5*(CL-6): GOTO 360
350   AD=6492+5*(CL-12)
360   VPOKE AD,144: VPOKE AD+1,148:
  VPOKE AD+2,148: VPOKE AD+3,148: VPOK
  E AD+4,146: VPOKE AD+32,150: VPOKE AD
  +64,150: VPOKE AD+96,150: VPOKE AD+12
  8,145: VPOKE AD+129,149
370   VPOKE AD+130,149: VPOKE AD+13
  1,149: VPOKE AD+132,147: VPOKE AD+100
  ,151: VPOKE AD+68,151: VPOKE AD+36,15
  1
380   FOR I= AD+33 TO AD+97 STEP 32
  : FOR J=0 TO 2: VPOKE I+J,152: NEXT:
  NEXT: VPOKE AD+66,N(CL,1)+48
390   RETURN

```



Volver cartas

```
400     PLAY"S8L50M1000N35N20":K=INT(
(CL-1)/6)+1: DN K GOTO 410,420,430
410     AD=6172+5*CL: GOTO 440
420     AD=6332+5*(CL-6): GOTO 440
430     AD=6492+5*(CL-12)
440     VPOKE AD,134: VPOKE AD+1,138:
      VPOKE AD+2,138: VPOKE AD+3,138: VPOK
E AD+4,136: VPOKE AD+32,140: VPOKE AD
+64,140: VPOKE AD+96,140: VPOKE AD+12
8,135: VPOKE AD+129,139
450     VPOKE AD+130,139: VPOKE AD+13
1,139: VPOKE AD+132,137: VPOKE AD+100
,141: VPOKE AD+68,141: VPOKE AD+36,14
1
460     FOR I= AD+33 TO AD+97 STEP 32
: FOR J=0 TO 2: VPOKE I+J,142: NEXT:
NEXT: VPOKE AD+66,64+CL
470     RETURN
```

Acertaste

```
480     P(JC)=P(JC)+1
490     N(UC,2)=1: N(CL,2)=1
500     K1=204: PLAY"M2000L10S8N35N40
": GOSUB 540
510     K1=193: PLAY"110S8M1000N30N45
": GOSUB 540: K1=32: GOSUB 540
520     IF P(1)+P(2)=9 THEN 580
530     RETURN
540     FOR I=6144 TO 6175: VPOKE I,K
1: NEXT: FOR I=6176 TO 6656 STEP 32:
VPOKE I,K1: FOR J=1 TO 50: NEXT: PLAY
"S8L64M500N25": VPOKE I+31,K1: NEXT:
FOR I= 6656 TO 6687: VPOKE I,K1: NEX
T
550     RETURN
```

Nombre/puntuación

```
560 PRINT CHR$(11);: FOR I=1 TO 1
8: PRINT : NEXT : PRINT J$(JC)":_" TA
B(18) "Puntos:";P(JC)
570 RETURN
```

Fin de juego

```
580 CLS: PRINT "El_ganador_es_":
IF P(1)>P(2) THEN PRINT J$(1): PRINT
"con_una_puntuacion_de_";P(1) ELSE P
RINT J$(2): PRINT "con_una_puntuacion
_de_";P(2)
590 PLAY"v10o414a18gel8egl2f11a",
"o314a18egl8gel2f11f","o514c18gel8gel
2f11c"
600 FOR I=1 TO 3000: NEXT: PRINT:
PRINT : PRINT" Mala_suerte,_";: IF P(
1)<P(2) THEN PRINT J$(1): PRINT "tus_
puntos_son_";P(1) ELSE PRINT J$(2):
PRINT "tus_puntos_son_";P(2)
610 PLAY "v15140s8m60000n2r40n1r3
0n4r20n1r30n2r10n5r10n4r10n3r2013n1"
620 END
```

Datos de sprites pequeños

```
630 DATA 0,0,0,0,10,5,10,5
640 DATA 10,5,10,5,0,0,0,0
650 DATA 0,0,0,0,160,64,160,64
660 DATA 160,64,160,64,0,0,0,0
670 DATA 0,0,0,0,170,85,170,85
680 DATA 170,85,170,85,0,0,0,0
690 DATA 10,5,10,5,10,5,10,5
```



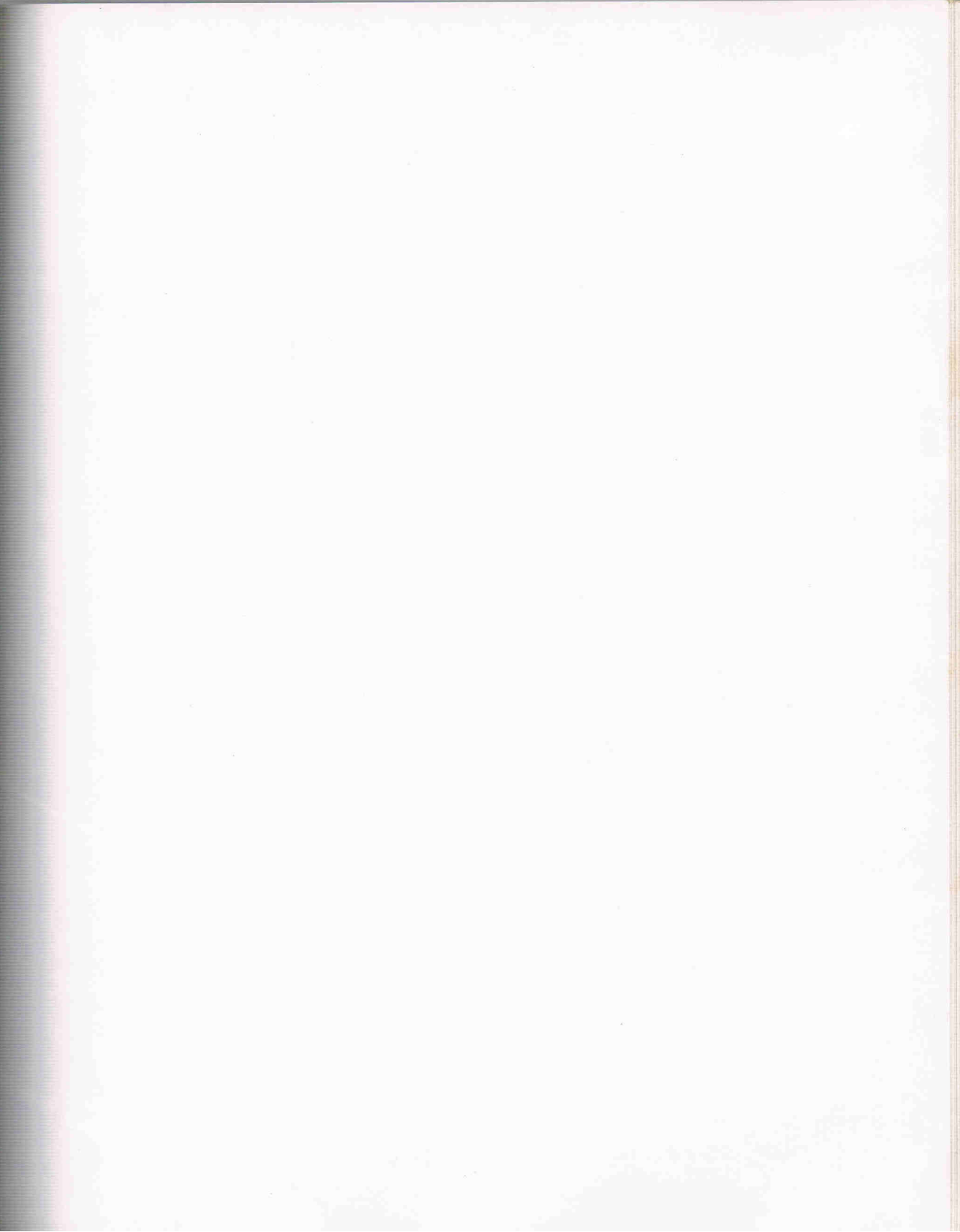
700	DATA 160,64,160,64,160,64,160,64
710	DATA 170,85,170,85,170,85,170,85
720	DATA 0,0,0,0,15,15,15,15
730	DATA 15,15,15,15,0,0,0,0
740	DATA 0,0,0,0,240,240,240,240
750	DATA 240,240,240,240,0,0,0,0
760	DATA 0,0,0,0,255,255,255,255
770	DATA 255,255,255,255,0,0,0,0
780	DATA 15,15,15,15,15,15,15,15
790	DATA 240,240,240,240,240,240,240,240
800	DATA 255,255,255,255,255,255,255,255

## Tabla de claves

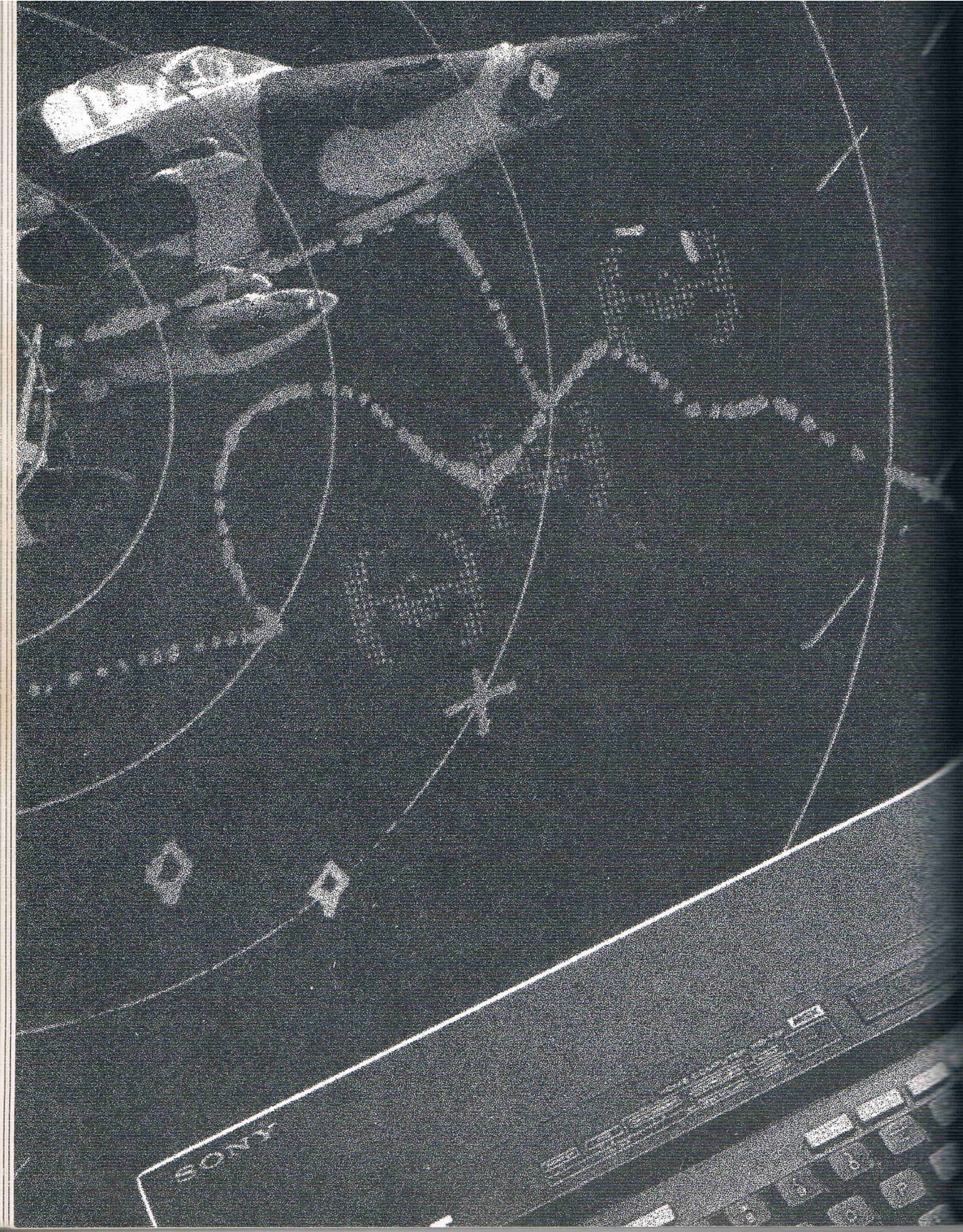
10	= 2641	210	= 3744	410	= 1495
20	= 389	220	= 2224	420	= 2001
30	= 6058	230	= 1780	430	= 1516
40	= 3238	240	= 5011	440	= 8539
50	= 6066	250	= 3604	450	= 5601
60	= 3391	260	= 3470	460	= 4997
70	= 5578	270	= 1957	470	= 142
80	= 3239	280	= 5030	480	= 1102
90	= 5027	290	= 1771	490	= 1311
100	= 3624	300	= 4702	500	= 2087
110	= 4708	310	= 604	510	= 2786
120	= 4851	320	= 4230	520	= 1306
130	= 1044	330	= 1415	530	= 142
140	= 1544	340	= 1921	540	= 7587
150	= 1436	350	= 1516	550	= 142
160	= 3381	360	= 8639	560	= 3998
170	= 3786	370	= 5661	570	= 142
180	= 311	380	= 5212	580	= 8866
190	= 4211	390	= 142	590	= 5719
200	= 4971	400	= 4100	600	= 8923

610	=	4516	690	=	1040	770	=	1256
620	=	129	700	=	1468	780	=	1256
630	=	932	710	=	1484	790	=	1640
640	=	932	720	=	1040	800	=	1688
650	=	1146	730	=	1040			
660	=	1146	740	=	1232			
670	=	1154	750	=	1232	TOTAL	=	231370
680	=	1154	760	=	1256			









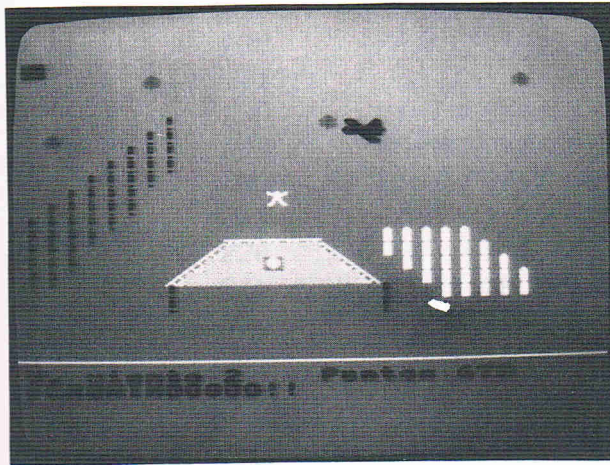
SONY

Calculator keypad with various function keys and numbers.



# Acróbata aéreo

# 19



## ¡ATENCIÓN!

Este programa utiliza la subrutina SOPORTE en lenguaje máquina. Véase apéndice C.

**Tipo de juego:**

**JUEGO DE SIMULACION**

Un grupo de acróbatas tratan de saltar de una avioneta a una cama elástica. Para saltar, pulsa la barra espaciadora, pero, ¡controla la velocidad del viento y la altura!, ya que pierdes puntos si caes al suelo o a la cerca de pinchos, y tu hombre muere si cae en los cables de alta tensión.

# SUGERENCIAS AL PROGRAMA

Puedes modificar la velocidad y el efecto del viento en la trayectoria del hombre cambiando los valores de las variables VV y T en la línea 470.

## Variables del programa

ND	Nivel de dificultad.
VV	Velocidad del viento.
T	Tiempo.
V	¿Vuelo?
NF, NH	Número de fallos; número de hombres.
M\$	Mensaje.
X, Y	Coordenadas del hombre.
NP, P	Número de pantalla; puntuación.
XI, YI	Coordenadas iniciales.

Inicialización

```
5 BEEP
10     REM     ** ATENCION **
20     REM
30     REM ESTE PROGRAMA NECESITA
40     REM LA SUBROUTINA SOPORTE
50     REM Vease el Apendice C
60     REM
70     REM
80     A=PEEK(64586!)+256*PEEK(6458
7!)
90     V1=A+9:V2=A+8:V3=A+7:V4=A+6
100    SCREEN 1,2: KEY OFF: CLS: PR
INT TAB(7)"ACROBATA_AERED": PRINT: PR
INT: PRINT
110    FOR I=1 TO 8: READ Q: A#=A#+
CHR$(Q): NEXT: SPRITE$(0)=A$: A#=""
120    FOR I=1 TO 32: READ Q: A#=A$
+CHR$(Q): NEXT: SPRITE$(1)=A$
130    FOR I=1088 TO 1095: READ Q:
VPOKE I,Q: NEXT: FOR I=1152 TO 1191:
```



```

READ Q: VPOKE I,Q: NEXT
140     FOR I=1216 TO 1223: READ Q:
VPOKE I,Q: NEXT: FOR I=1280 TO 1287:
READ Q: VPOKE I,Q : NEXT
150     DEFUSR=A+128: POKE V3VVV,1:
POKE V2VVV,1
160     ON STRIG GOSUB 770
170     NH=3: P=100
180     PRINT" Nivel_de_dificultad_(1
-4)?" ;
190     K$=INKEY$: IF K$="" THEN 190
200     ND=VAL(K$): IF ND<1 OR ND>4
THEN 190 ELSE PRINT ND
210     POKE V1VVV,ND
220     FOR I=1 TO 5: PRINT: NEXT: P
RINT"PULSA_UNA_TECLA_PARA_EMPEZAR"
230     IF INKEY$="" THEN 230

```

Nuevo juego

```

240     STRIG(0) ON: CLS: COLOR 1,5,5
250     IF NH=0 THEN 700
260     FOR I=6816 TO 6847: VPOKE I,2
3: NEXT
270     PRINT: PRINT"_____"CHR$(160)
"
"CHR$(160): PRINT:
PRINT: PRINT"_____!"CHR$(160
):PRINT"_"CHR$(160)"  !!": PRINT"__
___!!!": PRINT"_____!!!!"
280     PRINT"____!!!!": PRINT"__!!!!"
: PRINT"__!!!!": PRINT"!!!!": PRINT"
!!!!_____CHR$(146)CHR$(146)CHR$(146
)CHR$(146)CHR$(146)" "CHR$(148)CHR$(
148)CHR$(148)CHR$(148)CHR$(148)
290     PRINT"!!!_____CHR$(144)CHR$(
136)CHR$(136)CHR$(136)CHR$(136)CHR$(
136)CHR$(145);
300     PRINT TAB(18)CHR$(148)CHR$(14
8)CHR$(148)CHR$(148)CHR$(148)CHR$(148
):PRINT"!!!_____CHR$(144)CHR$(136)CH

```

```

R$(136)CHR$(136)CHR$(152)CHR$(136)CHR
$(136)CHR$(136)CHR$(145);
310 PRINT " _ "CHR$(148)CHR$(148)C
HR$(148)CHR$(148)CHR$(148)CHR$(148);P
RINT"!! _ _ _ _ "CHR$(144);: FOR I=1 TO 9
: PRINT CHR$(136);: NEXT: PRINT CHR$(
145)" _ "CHR$(148)CHR$(148)CHR$(148)CH
R$(148)CHR$(148)CHR$(148)
320 PRINT" _ _ _ _ _ ! _ _ _ _ _ ! _ _ ";
: FOR I=1 TO 5: PRINT CHR$(148);: NEX
T: PRINT : PRINT" _ _ _ _ _ ! _ _ _ _ _ !"
:PRINT: PRINT: PRINT CHR$(11);
330 VV=5: V=0: T=0: NF=0: GOSUB 8
00
340 PUT SPRITE 1, (255,40),1
350 VPOKE 8194,229: VPOKE 8209,16
5: VPOKE 8210,229: VPOKE 8211,149: VP
OKE 8212,197

```

Control

```

360 D=USR(D)
370 IF VPEEK(6917)>(255-ND) THEN
VPOKE 6916,INT(RND(1)*85): P=P-10: GO
SUB 800
380 IF V=0 THEN 360
390 GOSUB 460
400 IF X<5 OR X>250 THEN 670
410 IF Y<105 THEN 360
420 IF X>85 AND X<140 THEN GOSUB
530: GOTO 360
430 IF X>160 AND X<210 THEN GOSUB
590: GOTO 360
440 IF X<30 THEN 570
450 GOSUB 630: GOTO 360

```

Movimiento del acróbata

```

460 XV=3+ND/3: M=50
470 T=T+1: X=VV*T*T/M-XV*T+XI: Y=

```



```

4.9*T*T/M+YI
480     X=INT(X): Y=INT(Y): A$="n"+ST
R$(N): IF T/3=INT(T/3) THEN PLAY"M650
86L5S10XA$;":N=N-1
490     IF X>255 THEN X=255
500     IF X<0 THEN X=0
510     PUT SPRITE 0,(X,Y),10
520     RETURN

```

Acróbata en el centro de la red

```

530     IF X>105 AND X<117 THEN P=P+2
00: M$="Estupendo!!           ": PLAY"L
8N50L8N45L2N50": GOTO 550
540     F=P+100: M$="Buen salto!!
"
550     VV=INT(RND(1)*10): GOSUB 800:
GOSUB 850: V=0: T=0
560     RETURN

```

Acróbata en los cables de alta tensión

```

570     GOSUB 830: CLS: PRINT"Acrobat
a_electrocutado": NH=NH-1: PUT SPRITE
0,(100,200): PUT SPRITE 1,(250,200)
580     FOR TM=1 TO 3000: NEXT: P=P-1
00: GOTO 240

```

Acróbata en la acera

```

590     M$="Caiste en la cerca   ":
P=P-50: PLAY"L54M1200S10N74R20N74"
600     VV=INT(RND(1)*10): GOSUB 800:
V=0: T=0: NF=NF+1: GOSUB 850
610     IF NF>2 THEN 740
620     RETURN

```

Acróbata en el suelo

```
630 M$="AY!-Intentalo otra vez
": P=P-30: NF=NF+1: PLAY"L2N7"
640 VV=INT(RND(1)*10): GOSUB 800:
GOSUB 850: V=0: T=0
650 IF NF>2 THEN 740
660 RETURN
```

Acróbata fuera de límites

```
670 GOSUB 830: CLS: PRINT"El_acro
bata_cayo_en": PRINT"un_pantano_lejos
_del_blanco"
680 PUT SPRITE 0,(200,200): PUT S
PRITE 1,(200,200): FOR TM=1 TO 3000:
NEXT
690 NH=NH-1: GOTO 240
```

Fin de juego

```
700 CLS: PRINT TAB(6) "_FIN_DEL_J
UEGO": PRINT: PRINT: PRINT
710 PRINT"Tus_puntos_son";P
720 PUT SPRITE 1,(100,200): PUT S
PRITE 0,(100,200)
730 END
```

Acróbata dado de baja

```
740 V=1: CLS: PRINT"Demasiados_ac
cidentes": PRINT"Acrobata_dado_de_baj
a"
750 PUT SPRITE 0,(200,200): PUT S
PRITE 1,(200,200):NH=NH-1: FOR TM=1 T
O 3000: NEXT
760 GOTO 240
```



Espacio pulsado

```
770     IF V=1 THEN RETURN
780     K1=VPEEK(6916): K2=VPEEK(6917
): PUT SPRITE 0,(K2,K1),10: V=1: T=0:
M$="JERONIM00000!!"     ": GOSUB
850
790     YI=K1: XI=K2: N=50: RETURN
```

Puntuación

```
800     FOR I=1 TO 22: PRINT: NEXT: PR
INT"V. _viento";VV; TAB(15);"Puntos";P
; CHR$(11);:F=1
810     FOR J=6895 TO 6910: VPOKE J,32
: NEXT
820     RETURN
```

Acróbata muerto

```
830     CLS: PLAY"19m1000s14n33n33n33n
33n33n33n33n33n33n33n33": FOR I=15 TO 1
STEP -1: FOR T=1 TO 100: NEXT: COLOR
15,I,I: NEXT
840     RETURN
```

Mensaje

```
850     PRINT CHR$(11);: FOR I=1 TO 22
: PRINT: NEXT: FOR I=1 TO 13: PRINT C
HR$(28);: NEXT
860     PRINT:PRINT M$: CHR$(11);: RET
URN
```

Datos de sprites

870 DATA 153,255,60,24,60,36,102,19  
5

880 DATA 0,0,0,112,120,60,30,127,25  
5,255,127,1,0,0,0,0,0,0,4,14,30,62,  
254,255,255,254,254,240,120,60,28

Datos de sprites pequeños

890 DATA 170,85,170,85,170,85,170,8  
5

900 DATA 1,2,4,10,17,34,85,138

910 DATA 128,64,32,80,136,68,170,81

920 DATA 0,0,0,0,0,0,255,36

930 DATA 56,56,56,56,56,56,56,56

940 DATA 16,56,56,56,56,56,56,56

950 DATA 24,60,126,126,126,60,24,0

960 DATA 16,56,84,186,84,186,84,16

## Tabla de claves

5	= 192	140	= 3738	280	= 9051
10	= 904	150	= 2393	290	= 5039
20	= 143	160	= 727	300	= 10687
30	= 1637	170	= 901	310	= 12096
40	= 1541	180	= 2418	320	= 5067
50	= 1712	190	= 1623	330	= 1909
60	= 143	200	= 3165	340	= 952
70	= 143	210	= 735	350	= 2702
80	= 2571	220	= 3292	360	= 677
90	= 2998	230	= 1144	370	= 4425
100	= 3125	240	= 1279	380	= 818
110	= 3669	250	= 967	390	= 360
120	= 3194	260	= 1718	400	= 1719
130	= 3768	270	= 6680	410	= 925

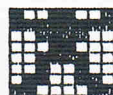


420	=	2001	610	=	1006	800	=	4712
430	=	2206	620	=	142	810	=	1839
440	=	804	630	=	4739	820	=	142
450	=	589	640	=	2950	830	=	6276
460	=	1523	650	=	1006	840	=	142
470	=	5274	660	=	142	850	=	3623
480	=	6898	670	=	5301	860	=	1291
490	=	1550	680	=	3282	870	=	1466
500	=	1046	690	=	1248	880	=	4779
510	=	810	700	=	2230	890	=	1484
520	=	142	710	=	1749	900	=	1160
530	=	5646	720	=	1847	910	=	1427
540	=	2432	730	=	129	920	=	989
550	=	2950	740	=	5095	930	=	1296
560	=	142	750	=	4139	940	=	1292
570	=	5839	760	=	391	950	=	1355
580	=	2440	770	=	842	960	=	1395
590	=	4469	780	=	5676			
600	=	3802	790	=	1748	TOTAL	=	241840

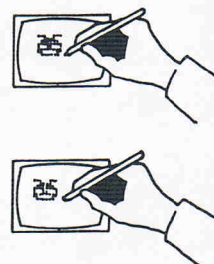
## Diseño de los sprites



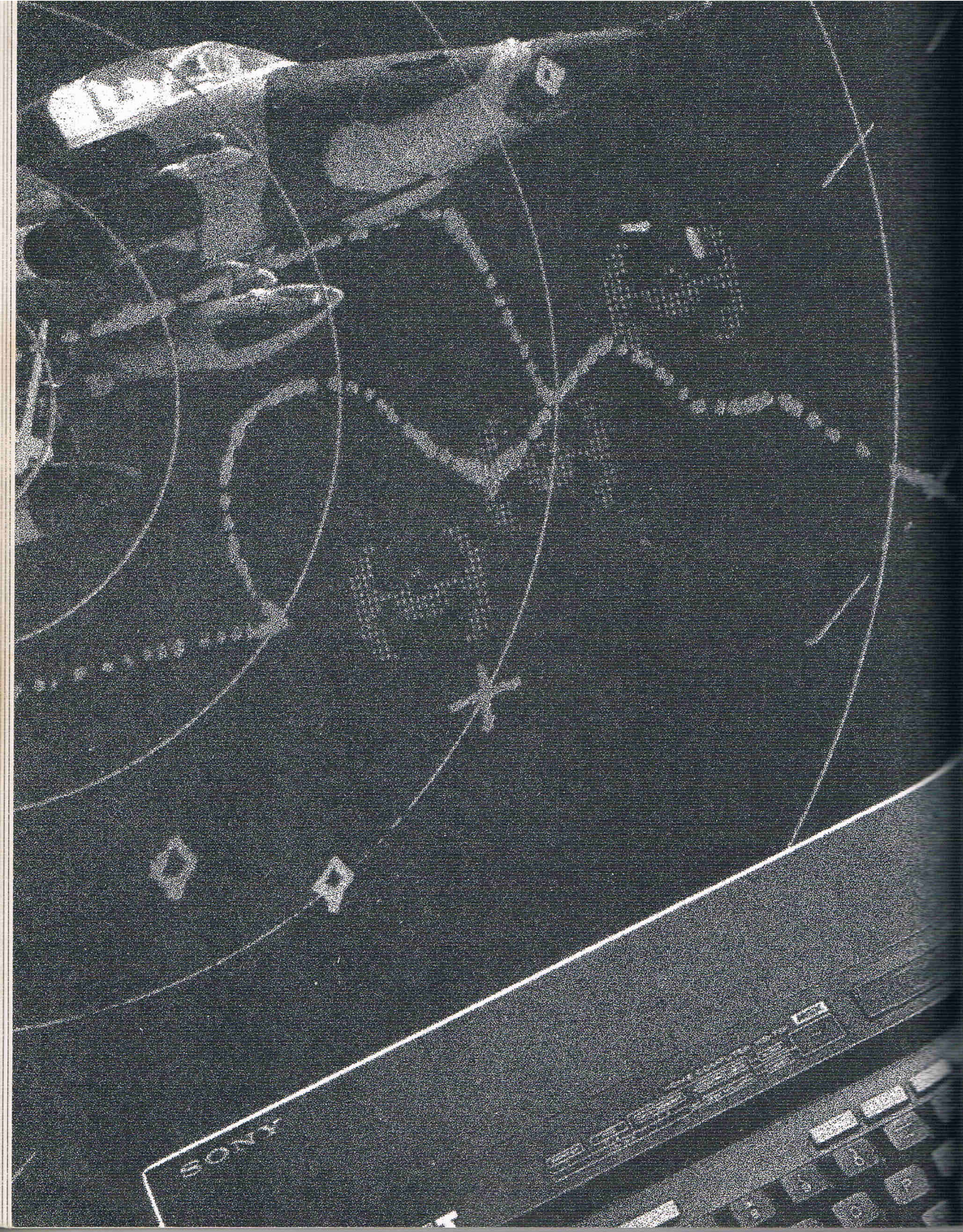
AVION



ACROBATA

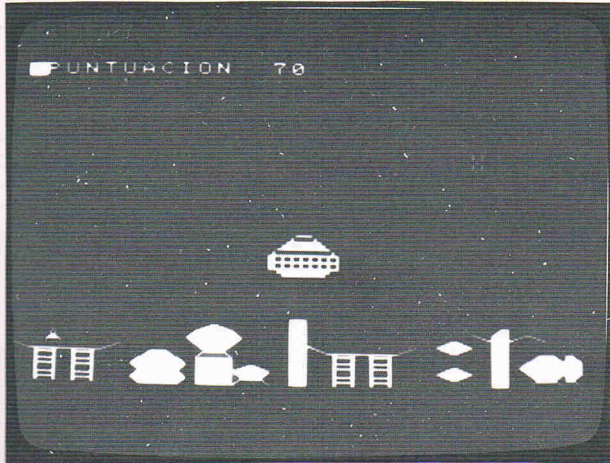








# Naves secretas 20 de investigación



## ¡ATENCIÓN!

Este programa utiliza la subrutina SOPORTE en lenguaje máquina. Véase apéndice C.

Tipo de juego:

JUEGO DE TIRO AL BLANCO

Los espías del planeta Circon han robado la fórmula de un componente valioso. El juego consiste en poner agentes de contraespionaje en pequeñas naves que aterricen en lugares especiales del planeta para que la fórmula pueda ser recuperada.

Debes depositar tantas naves como puedas sin rebasar el límite del tiempo, y evitar que la nave nodriza choque con las enemigas.

Se utiliza una subrutina en código máquina para animar la pantalla.





```

220     FOR I=1 TO 32: READ Q: A#=A$
+ CHR$(Q) : NEXT: SPRITE$(0)=A$:A$=""
230     FOR I=1 TO 8 : READ Q: A#=A$
+ CHR$(Q) : NEXT :SPRITE$(1)=A$
240     FOR I=1072 TO 1231 : READ Q
: VPOKE I,Q: NEXT
250     B=A+10
260     I=A+250
270     READ A$
280     IF A$="R" THEN READ C:C=C+B:
C1 = INT(C/256):C2=256*(C/256-C1):PO
KE I,C2:I=I+1:POKE I,C1 :GOTO 300
290     POKE I, VAL(A$)
300     IF I<A+250+51 THEN I=I+1:GO
TO 270
310     POKE W4WWW,224: POKE W3WWW,2
4 : POKE W2WWW,255 : POKE W1WWW,26 :
DEFUSR2 = A+250!
320     POKE V1VVV,4 : POKE V4VVV,15
:DEFUSR = A+10
330     PUT SPRITE 0,(112,55),11 : P
RINT "PULSA_UNA_TECLA_PARA_EMPEZAR"
340 IF INKEY$="" THEN D=USR2(D):FOR I
=1 TO 50: NEXT :D=RND(1): GOTO 340

```

Preparación pantalla

```

350     COLOR 15,1,1:CLS:VPOKE 8208,1
45:VPOKE 8209,145:VPOKE 8210,145: PUT
SPRITE 0,(114,80),11
360     VPOKE 6753,137: VPOKE 6768,14
4: VPOKE 6769,145: VPOKE 6773,134
370     VPOKE 6784,136: VPOKE 6785,13
4: VPOKE 6786,138: VPOKE 6787,136: VP
OKE 6799,139: VPOKE 6800,134: VPOKE 6
801,134: VPOKE 6802,141: VPOKE 6805,1
34: VPOKE 6813,144: VPOKE 6814,145
380     VPOKE 6817,134: VPOKE 6819,13
7: VPOKE 6820,140: VPOKE 6822,136: VP
OKE 6823,148: VPOKE 6824,149: VPOKE 6

```

```

825,148: VPOKE 6826,149
390     VPOKE 6827,138: VPOKE 6829,13
7: VPOKE 6830,140: VPOKE 6832,142: VP
OKE 6833,143: VPOKE 6837,134: VPOKE 6
838,136
400     VPOKE 6839,148: VPOKE 6840,14
9: VPOKE 6841,148: VPOKE 6842,149: VP
OKE 6843,138: VPOKE 6845,146: VPOKE 6
846,147
410     VPOKE 6849,134: VPOKE 6850,13
9: VPOKE 6851,134: VPOKE 6852,134: VP
OKE 6853,141: VPOKE 6855,150: VPOKE 6
856,151: VPOKE 6857,150
420     VPOKE 6858,151: VPOKE 6860,13
9: VPOKE 6861,134: VPOKE 6862,134: VP
OKE 6863,141: VPOKE 6864,134: VPOKE 6
865,134
430     VPOKE 6866,144: VPOKE 6867,14
5: VPOKE 6869,134: VPOKE 6871,150: VP
OKE 6872,151: VPOKE 6873,150: VPOKE 6
874,151: VPOKE 6877,144: VPOKE 6878,1
45
440     VPOKE 6881,134: VPOKE 6883,14
2: VPOKE 6884,143: VPOKE 6887,150: VP
OKE 6888,151: VPOKE 6889,150: VPOKE 6
890,151: VPOKE 6892,139
450     VPOKE 6893,134: VPOKE 6894,13
4: VPOKE 6895,141: VPOKE 6896,134: VP
OKE 6897,134
460     VPOKE 6898,142: VPOKE 6899,14
3: VPOKE 6901,134: VPOKE 6903,150: VP
OKE 6904,151: VPOKE 6905,150: VPOKE 6
906,151: VPOKE 6909,146: VPOKE 6910,1
47
470     FOR I= 6176 TO 6367: IF RND(1
)<.2 THEN VPOKE I,153
480     NEXT
490     GOSUB 800: PUT SPRITE 1,(255,
100),13
500     TIME = 0: ON INTERVAL= 10 GOS
UB 660: INTERVAL ON: TL=100

```



```
510   STRIG(0) ON:ON STRIG GOSUB 71
0
520   SPRITE ON: ON SPRITE GOSUB 69
0
```

Control

```
530   D=USR(D): IF VPEEK(6912)>136
      THEN FJ=1: GOTO 740
540   X=VPEEK(6913): IF X<57 THEN P
      OKE V4VVV,13 ELSE IF X>184 THEN POKE
      V4VVV,7 ELSE POKE V4VVV,15
550   GOSUB 580
560   IF LN<>0 THEN GOSUB 610
570   GOTO 530
```

Movimiento nave enemiga

```
580   SPRITE OFF: IF VPEEK(6917)<9
      THEN VPOKE 6916,INT(RND(1)*128+24): V
      POKE 6917,255
590   SPRITE ON : VPOKE 6917,VPEEK(
      6917)-7
600   RETURN
```

Movimiento nave investigación

```
610   INTERVAL OFF: STRIG(0) OFF: K
      =VPEEK(LN+32): INTERVAL ON
620   IF K=148 OR K=149 THEN VPOKE
      8208,241: VPOKE 8209,241: VPOKE 8210,
      241: P=P+10: PLAY "S1M2000L14N50N45":
      GOSUB 800: VPOKE LN,32: LN=0: VPOKE 8
      208,145: TH=1
630   IF TH=1 THEN VPOKE 8209,145:
      VPOKE 8210,145: STRIG(0) ON:TH=0: RET
      URN
```

```
640     IF K= 32 THEN LN=LN+32: VPOKE
LN,152: VPOKE LN-32,32: STRIG(0) ON:
RETURN
```

Estampido

```
650     VPOKE LN,135: FOR I=1 TO 100:
NEXT: P=P-10: GOSUB 800: VPOKE LN,32
: LN=0: STRIG(0) ON: RETURN
```

Movimiento de pantalla

```
660     D=USR2(D): IF TIME/60 >TL THE
N 740
670     IF LN<>0 THEN LN=LN-1
680     RETURN
```

Choque

```
690     INTERVAL OFF: STRIG(0) OFF: F
OR I= 1 TO 20: COLOR 15,1,1:PLAY"164s
8m20000n21n12": COLOR 1,15,15: NEXT
700     FJ=2: GOTO 740
```

Dejar nave investigación

```
710     IF LN<>0 THEN RETURN
720     I3=INT((VPEEK(6913)+20)/8): L
N=6144+32*INT(((VPEEK(6912)+20)/8))+I
3: IF LN<6368 THEN LN=0
730     RETURN
```



Fin de juego

```
740    COLOR 15,4,7: SCREEN 1: PRINT  
      "TUS_PUNTOS_SON";P  
750    PRINT: PRINT: PRINT: ON (FJ+1  
      ) GOTO 760,770,780  
760    PRINT "TU_TIEMPO_SE_ACABO": G  
      OTO 790  
770    PRINT "COLOSIONO_LA_NAVE_NODR  
      IZA": GOTO 790  
780    PRINT "LA_NAVE_NODRIZA_CHOCO_  
      CON_____UNA_NAVE_ENEMIGA"  
790    IF INKEY$="" THEN END ELSE 79  
      0
```

Puntuación

```
800    PRINT "PUNTUACION:";P;CHR$(11)  
      ;  
810    RETURN
```

Datos de sprites

```
820    DATA 3,4,15,31,63,64,255,213,25  
      5,213,255,63,31,0,0,0,192,32,240,248,  
      252,2,255,85,255,85,255,252,248,0,0,0  
830    DATA 160,160,64,160,160,0,0,0
```

Datos de sprites pequeños

```
840    DATA 255,255,255,255,255,255,25  
      5,255  
850    DATA 68,16,130,40,84,130,16,68  
860    DATA 192,48,12,3,0,0,0,0
```

870 DATA 0,0,0,0,15,31,63,127  
 880 DATA 3,12,48,192,0,0,0,0  
 890 DATA 1,3,7,15,15,7,3,1  
 900 DATA 0,0,0,0,240,248,252,254  
 910 DATA 128,192,224,240,240,224,192,128  
 920 DATA 255,127,63,31,31,32,64,128  
 930 DATA 255,254,252,248,248,4,2,1  
 940 DATA 0,0,0,0,3,15,63,255  
 950 DATA 0,0,0,0,192,240,252,255  
 960 DATA 255,63,15,3,0,0,0,0  
 970 DATA 255,252,240,192,0,0,0,0  
 980 DATA 0,0,0,0,255,24,31,24  
 990 DATA 0,0,0,0,255,24,248,24  
 1000 DATA 24,24,31,24,24,31,24,24  
 1010 DATA 24,24,248,24,24,248,24,24  
 1020 DATA 16,16,40,40,56,124,254,130  
 1030 DATA 0,0,0,32,0,0,0,0

Datos de código máquina

1040 DATA 6,31,42,R,236,43,35,205,R,1  
 87,120,254,31,202,R,265,4,43,205,R,20  
 4,35,195,R,277,6,0,17,31,0,25,205,R,2  
 04,183,237,82,237,91,R,238,123,189,19  
 4,R,246,122,188,194,R,246,201

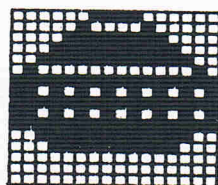
## Tabla de claves

100	= 904	170	= 143	240	= 1895
110	= 143	180	= 2571	250	= 636
120	= 3954	190	= 2998	260	= 883
130	= 1637	200	= 3954	270	= 236
140	= 1541	210	= 5201	280	= 6317
150	= 1712	220	= 3659	290	= 854
160	= 143	230	= 3204	300	= 2416

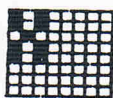


310	= 4268	570	= 171	830	= 1294
320	= 2290	580	= 4219	840	= 1688
330	= 3032	590	= 1522	850	= 1370
340	= 3734	600	= 142	860	= 1046
350	= 2991	610	= 4405	870	= 1093
360	= 2417	620	= 7416	880	= 1046
370	= 7076	630	= 3116	890	= 954
380	= 5363	640	= 3759	900	= 1248
390	= 4729	650	= 4385	910	= 1666
400	= 4854	660	= 2275	920	= 1417
410	= 5602	670	= 1813	930	= 1371
420	= 4933	680	= 142	940	= 1046
430	= 6520	690	= 5232	950	= 1247
440	= 5900	700	= 841	960	= 1046
450	= 3659	710	= 1148	970	= 1247
460	= 6808	720	= 6583	980	= 1092
470	= 2707	730	= 142	990	= 1150
480	= 131	740	= 2080	1000	= 1252
490	= 1238	750	= 1779	1010	= 1368
500	= 3898	760	= 1668	1020	= 1407
510	= 1645	770	= 2210	1030	= 877
520	= 1089	780	= 3123	1040	= 8621
530	= 2864	790	= 1287		
540	= 4864	800	= 1814		
550	= 225	810	= 142		
560	= 1261	820	= 5038		
				TOTAL	= 240097

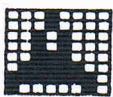
## Diseño de los sprites



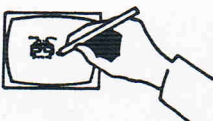
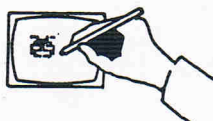
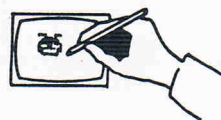
NAVE SECRETA



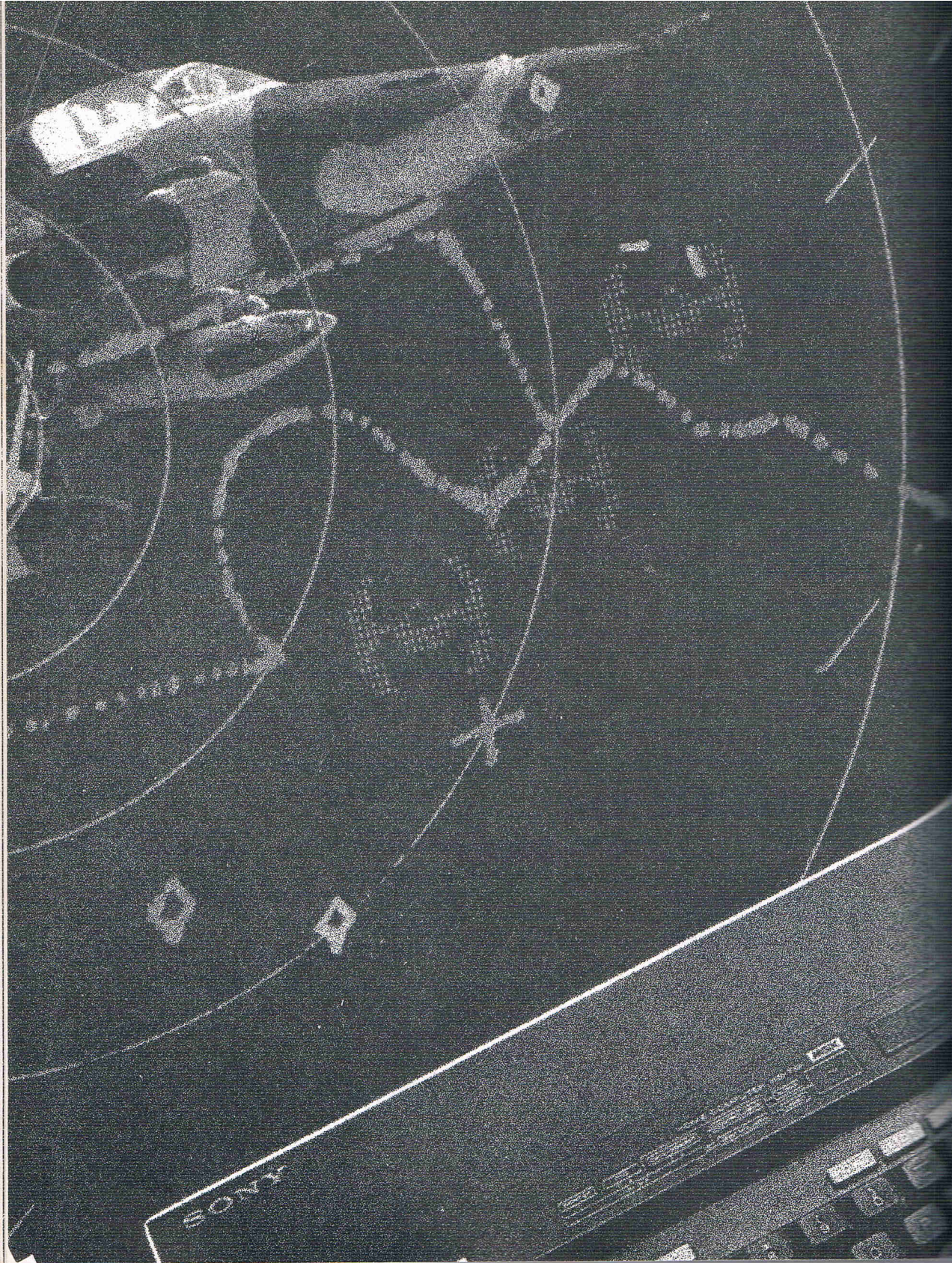
NAVE ENEMIGA



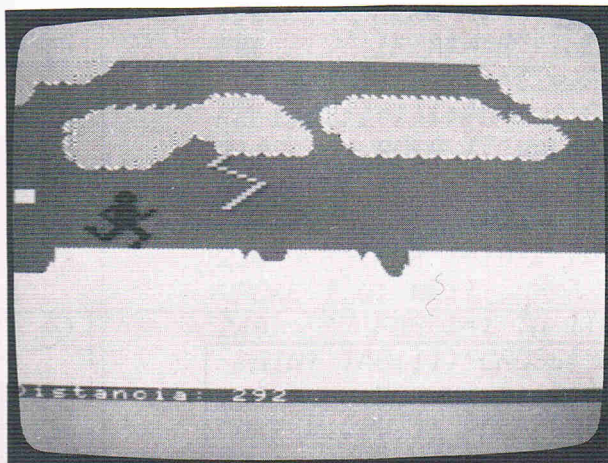
CAPSULA  
(CARACTER)











## ¡ATENCIÓN!

Este programa utiliza la subrutina  
SOPORTE en lenguaje máquina.  
Véase apéndice C.

**Tipo de juego:**

**JUEGO DE SIMULACION**

Este es un juego para corredores entusiastas.

Consigue tu mejor marca personal en los mil metros obstáculos. Para saltar los hoyos pulsa la barra espaciadora, y las flechas para controlar tu velocidad; pero ten cuidado de que no te dé el rayo.

Se usa una subrutina en código máquina para desplazar la pantalla.

# SUGERENCIAS AL PROGRAMA

Si no estás satisfecho con el suelo o con las nubes, modifica la rutina que da movimiento a la pantalla en las líneas 320 a 470.

También se puede cambiar la velocidad usual en la línea 310 (para mover al hombre más lejos) y en la línea 450 (para acelerar el movimiento de pantalla).

## Variables del programa

S Salto.  
DR Distancia recorrida.  
CJ Columna del jugador.  
FJ Final del juego.  
DL Destellos de luz.  
ND Nivel de dificultad.

Inicialización

```
10      REM      ** ATENCION **
20      REM
30      W1=A+249:W2=A+248:W3=A+247:W
4=A+246
40      REM ESTE PROGRAMA NECESITA
50      REM LA SUBROUTINA SOPORTE
60      REM Vease el Apendice C
70      REM
80      REM
90      A=PEEK(64586!)+256*PEEK(6458
7!)
100     V1=A+9:V2=A+8:V3=A+7:V4=A+6
110     W1=A+249:W2=A+248:W3=A+247:
W4=A+246
120     KEY OFF: COLOR 15,4,7: SCREE
N 1,3: FOR I=1 TO 32: READ Q: A#=A#+C
HR$(Q): NEXT: SPRITE$(0)=A$: A#=""
130     FOR I=1 TO 32: READ Q: A#=A$
+CHR$(Q): NEXT: SPRITE$(1)=A$: A#=""
140     FOR I=1 TO 32: READ Q: A#=A$
+CHR$(Q): NEXT: SPRITE$(2)=A$: A#=""
```



```

150     FOR I=1 TO 32: READ Q: A#=A#
+CHR$(Q): NEXT: SPRITE$(3)=A$
160     DEFUSR2=A+250!: POKE W4WWW,0
: POKE W3WWW,24: POKE W2WWW,223: POKE
W1WWW,25
170     B=A+10
180     I=A+250
190     READ A$
200     IF A$="R" THEN READ C:C=C+B:
C1 = INT(C/256):C2=256*(C/256-C1):PO
KE I,C2:I=I+1:POKE I,C1 :GOTO 220
210     POKE I, VAL(A$)
220     IF I<A+250+51 THEN I=I+1:GO
TO 190
230     FOR I=1072 TO 1215: READ Q:
VPOKE I,Q: NEXT
240     FOR I=1 TO 11: PRINT : NEXT:
PRINT TAB(11)"CROSS"
250     DEFUSR=A+10: DEFUSR1=A+128:
POKE V1VVV,1: POKE V4VVV,10: POKE V3V
VV,0: POKE V2VVV,3
260     PUT SPRITE 0,(0,55),1
270     IF INKEY$="" THEN IF VPEEK(6
914)=0 THEN VPOKE 6914,4 ELSE VPOKE 6
914,0 ELSE GOTO 290
280     D=USR1(D): GOTO 270
290     PRINT: PRINT: INPUT"Nivel_de
_dificultad_(1-3)";ND
300     IF ND<1 OR ND>3 THEN 290
310     POKE V1VVV,ND+2

```

Preparación pantalla

```

320     PUT SPRITE 0,(100,200): CLS:
FOR I=6609 TO 6879: VPOKE I,134: NEXT
330     VPOKE 6592,134: VPOKE 6593,13
4: VPOKE 6594,134: VPOKE 6595,139: VP
OKE 6596,138: FOR I=6597 TO 6606: VPO
KE I,134: NEXT: VPOKE 6607,139: VPOKE
6609,138

```

340 VPOKE 6560,135: VPOKE 6561,13  
5: VPOKE 6562,136: VPOKE 6563,137: FO  
R I=6565 TO 6574: VPOKE I,135: NEXT:  
VPOKE 6575,137:FOR I=6578 TO 6586: VP  
OKE I,135: NEXT: VPOKE 6587,136: VPOK  
E 6588,137: VPOKE 6589,138  
350 VPOKE 6590,135: VPOKE 6591,13  
5  
360 VPOKE 6152,142: FOR I=6153 TO  
6163: VPOKE I,140: NEXT: VPOKE 6164,  
146: VPOKE 6185,142: FOR I=6186 TO 61  
92: VPOKE I,140: NEXT: VPOKE 6193,144  
: VPOKE 6194,144: VPOKE 6195,145  
370 VPOKE 6209,150: FOR I=6210 TO  
6215: VPOKE I,149: NEXT: VPOKE 6216,  
147: VPOKE 6218,143: FOR I=6219 TO 62  
23: VPOKE I,140: NEXT: VPOKE 6224,146  
: VPOKE 6233,150: VPOKE 6234,149: VPO  
KE 6235,149: VPOKE 6236,147  
380 VPOKE 6240,141: FOR I=6241 TO  
6248: VPOKE I,140: NEXT: VPOKE 6249,  
148: VPOKE 6250,147: VPOKE 6251,143:  
VPOKE 6252,144: VPOKE 6253,144: VPOKE  
6254,144: VPOKE 6255,145: VPOKE 6259  
,150: FOR I=6260 TO 6263: VPOKE I,151  
: NEXT  
390 VPOKE 6264,151: FOR I=6265 TO  
6268: VPOKE I,140: NEXT: VPOKE 6269,  
148  
400 VPOKE 6272,142: FOR I=6273 TO  
6283: VPOKE I,140: NEXT: VPOKE 6284,  
148: VPOKE 6290,141: FOR I=6291 TO 63  
01: VPOKE I,140: NEXT: VPOKE 6302,148  
410 VPOKE 6305,142: FOR I=6306 TO  
6315: VPOKE I,140: NEXT: VPOKE 6316,  
146: VPOKE 6322,141: FOR I=6323 TO 63  
27: VPOKE I,140: NEXT: VPOKE 6328,146  
: VPOKE 6329,143: FOR I=6330 TO 6334:  
VPOKE I,140: NEXT  
420 VPOKE 6338,143: FOR I=6339 TO  
6346: VPOKE I,144: NEXT: VPOKE 6347,



```
145: VPOKE 6354,142: FOR I=6355 TO 63
58: VPOKE I,140: NEXT: VPOKE 6359,146
: VPOKE 6362,143: VPOKE 6363,144: VPO
KE 6364,144: VPOKE 6365,144: VPOKE 63
66,145
430     VPOKE 6387,143: VPOKE 6388,14
4: VPOKE 6389,144: VPOKE 6390,145
440     PUT SPRITE 0,(154,72)
450     INTERVAL ON: ON INTERVAL=15-2
*ND GOSUB 600
460     SPRITE ON: ON SPRITE GOSUB 70
0
470     STRIG(0) ON: ON STRIG GOSUB 6
80: TIME=0
```

Control

```
480     D=USR(D): K=INT((VPEEK(6913)+
8)/8): IF K-CJ>0 THEN DR=DR+8*(K-CJ)
490     CJ=K: IF VPEEK(6560+CJ)<>135
AND S=0 AND CJ<>32 THEN GOSUB 650
500     IF S<>0 THEN S=S+1: IF S=10-2
*ND THEN S=0: VPOKE 6914,0
510     GOSUB 750
520     IF RND(1)<.07 THEN GOSUB 550
```

Rayos

```
530     IF DL<>0 THEN GOSUB 570
540     GOTO 480
550     IF DL<>0 THEN RETURN
560     PUT SPRITE 3,(INT(RND(1)*255)
,50),11: DL=1: PLAY"12m2000s10n45": R
ETURN
570     DL=DL+1
580     IF DL=5 THEN PUT SPRITE 3,(10
0,200),11: DL=0: RETURN
590     RETURN
```

Movimiento de pantalla

```
600      D=USR2(D)
610      DR=DR + 1
620      IF S<>0 THEN RETURN
630      STRIG(0) OFF: I1=VPEEK(6914):
        IF I1=4 THEN I1=0 ELSE I1=4
640      VPOKE 6914,I1: STRIG(0) ON: R
        ETURN
```

Corredor en un bache

```
650      SPRITE OFF: INTERVAL OFF: STR
        IG(0) OFF: VPOKE 6914,8: VPOKE 6912,94
        : PLAY"12m40000s8n4": FOR I2=1 TO 150
        0: NEXT: VPOKE 6912,72: VPOKE 6914,0
660      VPOKE 6913,(VPEEK(6913)+1) MO
        D 255: CJ=INT((VPEEK(6913)+8)/8): IF
        VPEEK(6560+CJ)<>135 THEN 660
670      SPRITE ON: INTERVAL ON: STRIG
        (0) ON: RETURN
```

Salto

```
680      IF S<>0 THEN RETURN
690      S=1: VPOKE 6914,8: RETURN
```

Corredor fulminado por un rayo

```
700      INTERVAL OFF: FOR I=1 TO 20:
        PLAY"164m800s14n50": FOR T=1 TO 50: N
        EXT: COLOR 15,1,1: FOR T=1 TO 50: NEX
        T: COLOR 15,4,7: NEXT
710      FJ=1
```



Fin de juego

```
720   CLS: IF FJ=1 THEN PRINT"Te_ha
s_electrocutado.": PRINT: PRINT"Dista
ncia:";DR;"metros en": PRINT USING"##
###.##";TIME/60;; PRINT"_segundos.":
GOTO 740
730   PRINT"Se_acabo_la_carrera.":
PRINT: PRINT"Tu_tiempo_fue";: PRINT U
SING"#####.##"; TIME/60 : PRINT"_segu
ndos."
740   IF INKEY$("<>") THEN END ELSE 7
40
```

Distancia recorrida

```
750   FOR I=1 TO 23: PRINT: NEXT: PR
INT"Distancia:";DR;CHR$(11);
760   IF DR>1000 THEN 720
770   RETURN
```

Datos de sprites

```
780   DATA 1,3,3,1,7,11,19,11,7,3,195
,70,44,24,0,0,128,192,192,128,192,196
,232,208,192,192,96,48,24,16,32,96
790   DATA 1,3,3,1,3,3,3,3,3,1,1,1,
2,4,2,128,192,192,128,192,192,240,192
,192,128,128,128,128,128,128,192
800   DATA 33,19,11,5,3,3,3,3,131,2
55,0,0,0,0,0,132,200,208,160,192,192,
192,192,192,193,255,0,0,0,0,0
810   DATA 6,12,24,48,12,3,0,0,0,0,0,
0,0,1,3,6,0,0,0,0,0,192,48,12,24,48
,96,192,128,0,0
```

Datos de código máquina

820 DATA 6,31,42,R,236,43,35,205,R,18  
7,120,254,31,202,R,265,4,43,205,R,204  
,35,195,R,277,6,0,17,31,0,25,205,R,20  
4,183,237,82,237,91,R,238,123,189,194  
,R,246,122,188,194,R,246,201

Datos de sprites pequeños

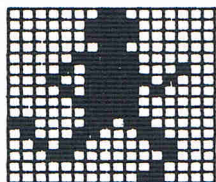
830 DATA 255,255,255,255,255,255,25  
5,255  
840 DATA 0,170,255,255,255,255,255,  
255  
850 DATA 0,192,225,227,243,247,247,  
255  
860 DATA 0,0,128,192,192,224,224,24  
0  
870 DATA 0,2,3,3,7,7,7,15  
880 DATA 240,240,248,248,252,252,25  
4,255  
890 DATA 170,85,170,85,170,85,170,8  
5  
900 DATA 1,5,10,13,21,26,13,21  
910 DATA 10,21,10,13,6,10,5,2  
920 DATA 170,85,106,53,10,0,0,0  
930 DATA 170,85,170,205,134,0,0,0  
940 DATA 85,170,84,172,80,0,0,0  
950 DATA 85,170,86,168,84,168,80,16  
0  
960 DATA 0,0,0,160,80,176,84,170  
970 DATA 128,80,160,64,168,80,170,8  
4  
980 DATA 0,0,140,198,85,170,85,170  
990 DATA 0,0,0,2,1,5,10,13  
1000 DATA 1,5,10,85,170,85,170,85



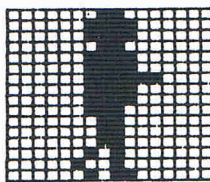
## Tabla de claves

10	= 904	360	= 6420	710	= 401
20	= 143	370	= 8088	720	= 8968
30	= 3954	380	= 9041	730	= 6346
40	= 1637	390	= 3003	740	= 1731
50	= 1541	400	= 6209	750	= 3187
60	= 1712	410	= 9060	760	= 1232
70	= 143	420	= 10365	770	= 142
80	= 143	430	= 2964	780	= 4914
90	= 2571	440	= 820	790	= 4783
100	= 2998	450	= 3404	800	= 4589
110	= 3954	460	= 1099	810	= 3878
120	= 4903	470	= 2132	820	= 8621
130	= 3660	480	= 5393	830	= 1688
140	= 3661	490	= 4341	840	= 1576
150	= 3196	500	= 3724	850	= 1575
160	= 4013	510	= 395	860	= 1457
170	= 636	520	= 1486	870	= 907
180	= 883	530	= 1211	880	= 1673
190	= 236	540	= 376	890	= 1484
200	= 6492	550	= 1138	900	= 1141
210	= 854	560	= 4040	910	= 1087
220	= 2591	570	= 786	920	= 1197
230	= 1879	580	= 2387	930	= 1300
240	= 2046	590	= 142	940	= 1211
250	= 4752	600	= 696	950	= 1492
260	= 713	610	= 798	960	= 1257
270	= 3318	620	= 1077	970	= 1479
280	= 919	630	= 3491	980	= 1369
290	= 2894	640	= 1344	990	= 933
300	= 1461	650	= 6530	1000	= 1270
310	= 995	660	= 6060		
320	= 3078	670	= 2420		
330	= 6797	680	= 1077	TOTAL	= 281636
340	= 9135	690	= 922		
350	= 1329	700	= 6168		

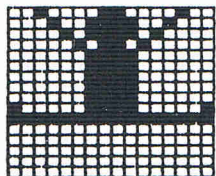
# Diseño de los sprites



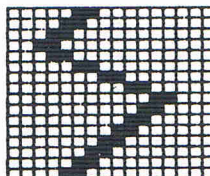
CORREDOR: POSICION 1



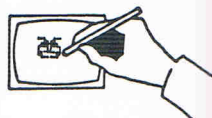
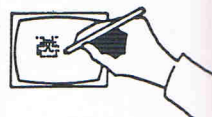
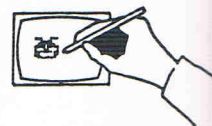
CORREDOR: POSICION 2



CORREDOR: CAIDA



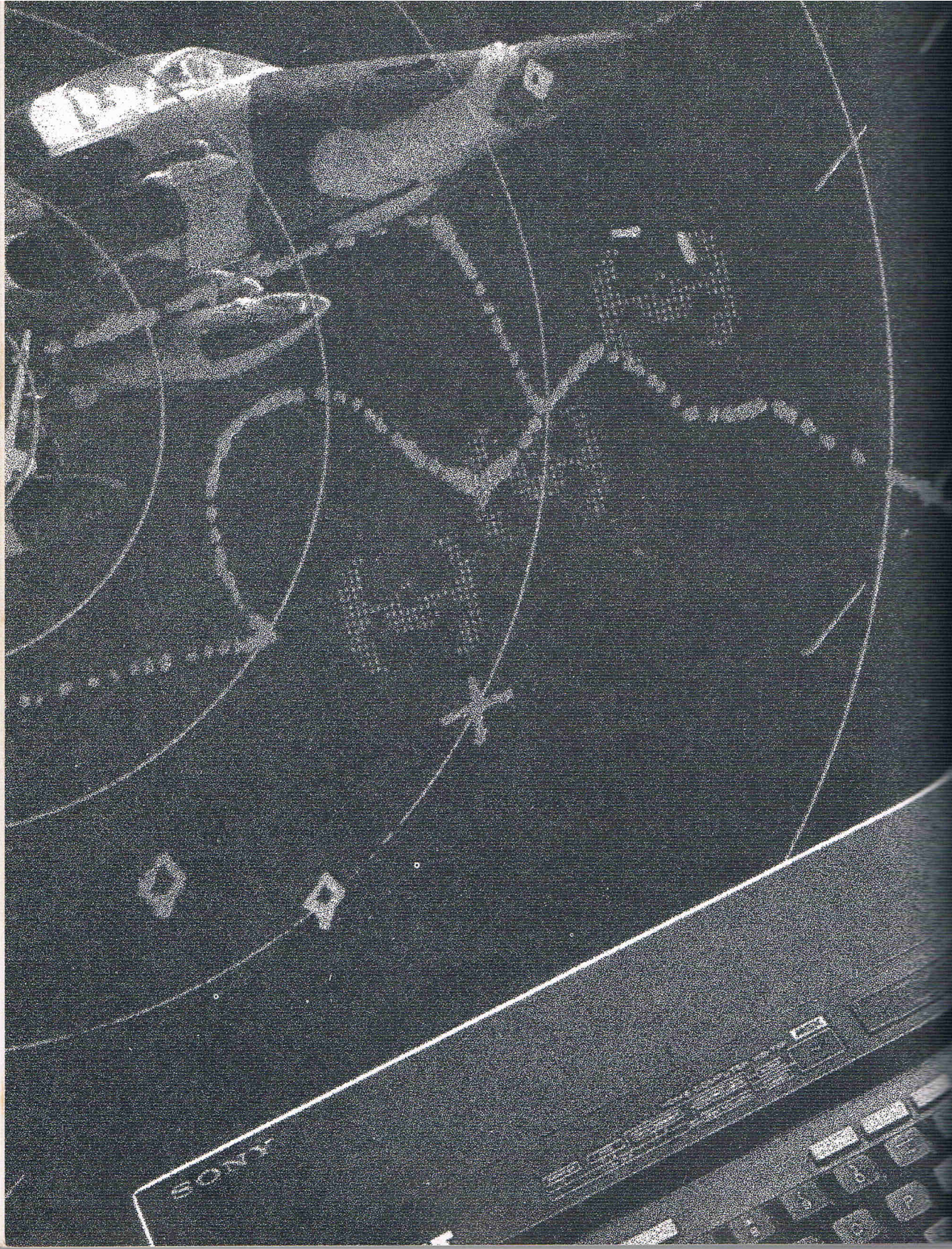
RAYO





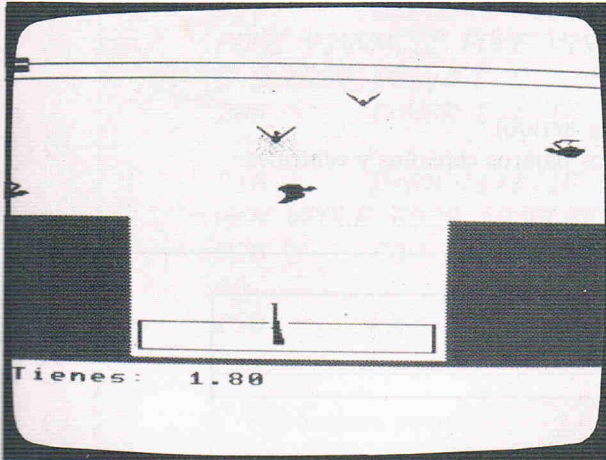








# Caseta de tiro 22



**¡ATENCIÓN!**

Este programa utiliza la subrutina  
SOPORTE en lenguaje máquina.  
Véase apéndice C.

**Tipo de juego:**

**TIRO AL BLANCO**

Prueba tu puntería. ¡Cinco tiros por sólo cien pesetas!

Puntuación:

Patos cercanos	20 pesetas.
Patos centrales	50 pesetas.
Pájaro loco	100 pesetas.

# SUGERENCIAS AL PROGRAMA

Se puede hacer más difícil alcanzar a los pájaros modificando las líneas 710-750 y su velocidad se controla en la línea 400 (pájaros centrales) y la línea 440 (cercanos).

Intenta añadir pájaros con distintos valores para dar más variedad.

## Variables del programa

DR Dinero restante.  
NT Número de tiros (0-5).  
PH Pájaro herido (-1 si no está herido).  
C1, C2 Espacio de tiempo entre los pájaros cercanos y centrales.  
D ¿Te queda dinero?

### Inicialización

5 BEEP

```
10 REM ** ATENCION **
20 REM
30 REM ESTE PROGRAMA NECESITA
40 REM LA SUBROUTINA SOPORTE
50 REM Vease el Apendice C
60 REM
70 REM
80 A=PEEK(64586!)+256*PEEK(6458
7!)
90 V1=A+9:V2=A+8:V3=A+7:V4=A+6
100 SCREEN 1,2: COLOR 1,15,15: K
EY OFF: PRINT TAB(7)"CASETA_DE_TIRO"
110 FOR I=1 TO 32: READ Q: A#=A$
+CHR$(Q): NEXT: SPRITE$(0)=A$: A$=""
120 FOR I=1 TO 32: READ Q: A#=A$
+CHR$(Q): NEXT: SPRITE$(1)=A$: A$=""
130 FOR I=1 TO 32: READ Q: A#=A$
+CHR$(Q): NEXT: SPRITE$(2)=A$: A$=""
140 FOR I=1 TO 32: READ Q: A#=A$
+CHR$(Q): NEXT: SPRITE$(3)=A$: A$=""
150 FOR I=1 TO 32: READ Q: A#=A$
```



```

+CHR$(Q): NEXT: SPRITE$ (4)=A$: A$=""
160   FOR I=1 TO 32: READ Q: A$=A$
+CHR$(Q): NEXT: SPRITE$ (5)=A$: A$=""
170   FOR I=1 TO 32: READ Q: A$=A$
+CHR$(Q): NEXT: SPRITE$ (6)=A$: A$=""
180   FOR I=1072 TO 1119: READ Q:
VPOKE I,Q: NEXT
190   DEFUSR=A+128: POKE V3VVV,2:
POKE V2VVV,1: POKE V1VVV,5: PUT SPRIT
E 2,(255,80),4
200   D=USR(D): IF INKEY$("<") THEN
230
210   D=RND(1): IF VPEEK(6922)=8 T
HEN VPOKE 6922,12 ELSE VPOKE 6922,8
220   FOR I=1 TO 120: NEXT: GOTO 2
00
230   PH=-1

```

Preparación pantalla

```

240   CLS: VPOKE 6922,8: PUT SPRITE
2,(255,36),12: PUT SPRITE 3,(140,36)
,12: PUT SPRITE 0,(100,16),8: PUT SPR
ITE 4,(0,64),4: PUT SPRITE 5,(115,64)
,4
250   FOR I=6145 TO 6206: VPOKE I,2
3: NEXT: FOR I=6729 TO 6742: VPOKE I,
23: NEXT: FOR I=6793 TO 6806: VPOKE I
,23: NEXT
260   FOR I=6816 TO 6847: VPOKE I,1
39: NEXT: FOR I=6496 TO 6784 STEP 32:
FOR J=0 TO 7: VPOKE I+J,134: VPOKE I
+J+24,134: NEXT: NEXT
270   VPOKE 6144,24: VPOKE 6728,24:
VPOKE 6176,26: VPOKE 6792,26: VPOKE
6175,25: VPOKE 6743,25: VPOKE 6207,27
: VPOKE 6807,27
280   VPOKE 6760,22: VPOKE 6775,22:
VPOKE 6703,135: VPOKE 6735,136: VPOK
E 6767,137

```

```
290 STRIG(0) DN: DN STRIG GOSUB 7
00
300 GOSUB 770
```

Control

```
310 GOSUB 350: GOSUB 400: GOSUB 4
30
320 IF PH>=0 THEN GOSUB 460
330 IF MD=1 THEN GOSUB 770: MD=0
340 GOTO 310
```

Movimiento del pájaro loco

```
350 POKE V1VVV,7: POKE V3VVV,0: I
F RND(1)<.5 THEN POKE V2VVV,1 ELSE P
KE V2VVV,3
360 IF VPEEK(6913)>194 THEN POKE
V2VVV,1 ELSE IF VPEEK(6913)<65 THEN P
OKE V2VVV,3
370 D=USR(D)
380 IF RND(1)<.5 THEN VPOKE 6914,
4 ELSE VPOKE 6914,0
390 RETURN
```

Movimiento de pájaros de perfil

```
400 POKE V1VVV,13: POKE V3VVV,2:
K1=VPEEK(6922): VPOKE 6922,VPEEK(6926
): VPOKE 6926,K1
410 POKE V2VVV,1: D=USR(D): POKE
V3VVV,3: D=USR(D)
420 RETURN
```



Movimiento de pájaros de frente

```
430      C2=(C2+1) MOD 2: POKE V3VVV,4
: IF C2=0 THEN K1=VPEEK(6930): VPOKE
6930,VPEEK(6934): VPOKE 6934,K1
440      POKE V1VVV,4: POKE V2VVV,3: D
=USR(D): POKE V3VVV,5: D=USR(D)
450      RETURN
```

Más pájaros

```
460      DN (PH+1) GOTO 470,470,480,50
0,520,540
470      PUT SPRITE 0,(100,16): GOTO 5
50
480      VPOKE 6922,8: VPOKE 6926,12
490      GOTO 550
500      VPOKE 6922,8: VPOKE 6926,12
510      GOTO 550
520      VPOKE 6930,16: VPOKE 6934,20
530      GOTO 550
540      VPOKE 6930,16: VPOKE 6934,20
550      PH=-1: RETURN
```

Pájaro derribado

```
560      PLAY"L50M1200S10N74R30N74R30N
74R30N74R30N74R30N74R30N74": DN (PH+1
) GOTO 570,580,590,610,630,650
570      DR=DR+1: VPOKE 6914,24: FOR I
2=1 TO 500: NEXT
580      VPOKE 6914,0: GOTO 670
590      DR=DR+.5: VPOKE 6922,24: FOR
I2=1 TO 500: NEXT
600      VPOKE 6922,8: GOTO 670
```

```
610 DR=DR+.5: VPOKE 6926,24: FOR
I2=1 TO 500: NEXT
620 VPOKE 6926,12: GOTO 670
630 DR=DR+.2: VPOKE 6930,24: FOR
I2=1 TO 500: NEXT
640 VPOKE 6930,16: GOTO 670
650 VPOKE 6934,24: DR=DR+.2: FOR
I2=1 TO 500: NEXT
660 VPOKE 6934,20
670 MD=1: STRIG(0) ON: RETURN
```

Pájaro fallado

```
680 STRIG(0) ON: VPOKE 6191,138:
FOR I1=1 TO 100: NEXT: VPOKE 6191,23
690 MD=1: RETURN
```

Disparo

```
700 STRIG(0) OFF: PLAY"L64M60S8N2
7": NT=NT+1
710 IF ABS(VPEEK(6929)-114)<3 THE
N PH=4: GOTO 560
720 IF ABS(VPEEK(6933)-114)<3 THE
N PH=5: GOTO 560
730 IF ABS(VPEEK(6921)-114)<3 THE
N PH=2: GOTO 560
740 IF ABS(VPEEK(6925)-114)<3 THE
N PH=3: GOTO 560
750 IF ABS(VPEEK(6913)-114)<3 THE
N PH=0: GOTO 560
760 GOTO 680
```

Dinero ganado/perdido

```
770 IF DR>=0 THEN A$="Tienes:" EL
SE A$="Debes:"
```



```
780     IF DR<0 THEN I3=INT(-DR): I4=-
-(DR+I3) ELSE I3=INT(DR): I4=DR-I3
790     FOR I=1 TO 22: PRINT: NEXT: P
RINT A$;: PRINT USING"###";I3;: PRINT
USING".##";I4: PRINT CHR$(11);
800     IF NT>4 THEN 810 ELSE RETURN
```

¿Más disparos?

```
810     PLAY"L10S14M2000N30N40N20N27"
: STRIG(0) OFF: FOR I=1 TO 23: PRINT
: NEXT: PRINT "5_tiros_mas?(S/N)";CHR
$(11);
820     X$=INKEY$: IF X$="" THEN 820
ELSE IF X$<>"S" AND X$<>"s" AND X$<>"
N" AND X$<>"n" THEN 820
830     IF X$="N" OR X$="n" THEN 860
840     DR=DR-1: NT=0: GOSUB 770
850     FOR I=6882 TO 6910: VPOKE I,3
2: NEXT: STRIG(0) ON: RETURN
```

Fin de juego

```
860     SCREEN 1: IF DR<0 THEN PRINT"
Debes: ";: PRINT USING"###.##";-DR: GO
TO 880
870     PRINT "Has_ganado: ";: PRINT U
SING"###.##";DR: GOTO 880
880     IF INKEY$<>" " THEN END ELSE 8
80
```

Datos de sprites

```
890     DATA 128,64,96,56,28,13,7,3,1,2
,0,0,0,0,0,0,1,2,6,28,56,176,224,192,
128,64,0,0,0,0,0,0
```

900 DATA 0,0,0,0,0,1,7,15,25,50,32,  
32,32,0,0,0,0,0,0,0,0,128,224,240,152  
,76,4,4,4,0,0,0

910 DATA 0,0,0,0,3,14,24,12,6,115,2  
55,63,7,0,0,0,0,0,0,0,252,32,64,32,24  
8,255,255,254,252,248,0,0

920 DATA 0,0,0,0,1,115,255,63,3,7,7  
,3,0,0,0,0,0,0,0,0,248,254,255,255,25  
2,248,248,248,252,62,15,0

930 DATA 0,0,0,0,63,4,2,4,63,255,25  
5,127,63,31,0,0,0,0,0,0,192,112,24,48  
,224,206,255,252,224,0,0,0

940 DATA 0,0,0,0,31,127,255,255,63,  
31,31,31,63,124,240,0,0,0,0,128,206  
,255,252,192,224,224,192,0,0,0,0

950 DATA 130,200,96,49,25,45,39,75,  
19,5,66,4,65,144,4,34,1,35,6,140,216,  
177,224,201,196,144,66,32,4,18,128,1

Datos de sprites pequeños

960 DATA 170,85,170,85,170,85,170,8  
5

970 DATA 96,32,32,32,32,32,32,32

980 DATA 32,32,48,48,48,48,112,56

990 DATA 48,56,56,56,60,60,60,60

1000 DATA 0,0,24,36,82,133,64,18

1010 DATA 170,85,170,85,0,0,0,0

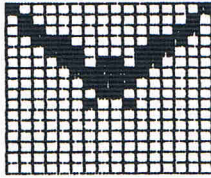
## Tabla de claves

5	= 192	60	= 143	120	= 3660
10	= 904	70	= 143	130	= 3661
20	= 143	80	= 2571	140	= 3662
30	= 1637	90	= 2998	150	= 3663
40	= 1541	100	= 2695	160	= 3664
50	= 1712	110	= 3659	170	= 3665

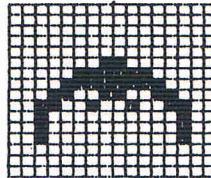


180	= 1783	470	= 959	760	= 321
190	= 4117	480	= 728	770	= 3251
200	= 2118	490	= 191	780	= 5387
210	= 2914	500	= 728	790	= 3888
220	= 1410	510	= 191	800	= 1198
230	= 651	520	= 758	810	= 5829
240	= 4992	530	= 191	820	= 5963
250	= 4722	540	= 758	830	= 1763
260	= 6208	550	= 851	840	= 1493
270	= 3630	560	= 4981	850	= 2747
280	= 2699	570	= 2441	860	= 3345
290	= 1635	580	= 685	870	= 2517
300	= 160	590	= 2604	880	= 1616
310	= 996	600	= 701	890	= 4055
320	= 1363	610	= 2608	900	= 3851
330	= 1378	620	= 707	910	= 4391
340	= 206	630	= 2532	920	= 4404
350	= 4034	640	= 715	930	= 4434
360	= 4006	650	= 2536	940	= 4729
370	= 677	660	= 354	950	= 5018
380	= 2115	670	= 1325	960	= 1484
390	= 142	680	= 2742	970	= 1258
400	= 3674	690	= 602	980	= 1329
410	= 2747	700	= 2633	990	= 1277
420	= 142	710	= 2689	1000	= 1211
430	= 4904	720	= 2694	1010	= 1154
440	= 3419	730	= 2679		
450	= 142	740	= 2684		
460	= 2022	750	= 2669	TOTAL	= 234168

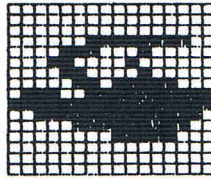
# Diseño de los sprites



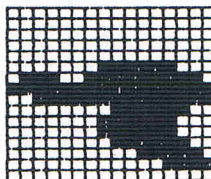
PAJARO LOCO: POSICION 1



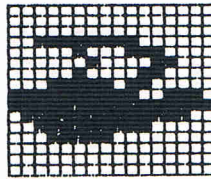
PAJARO LOCO: POSICION 2



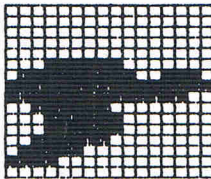
PATO MEDIANO: POSICION 1



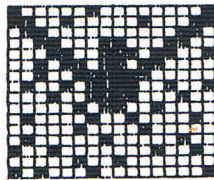
PATO MEDIANO: POSICION 2



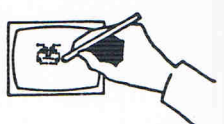
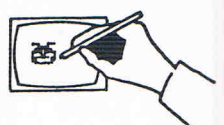
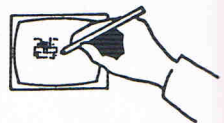
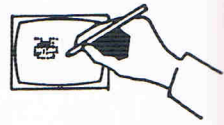
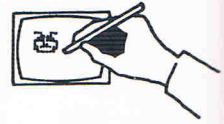
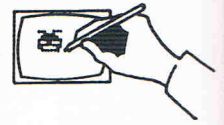
PAJARO DE FRENTE: POSICION 1



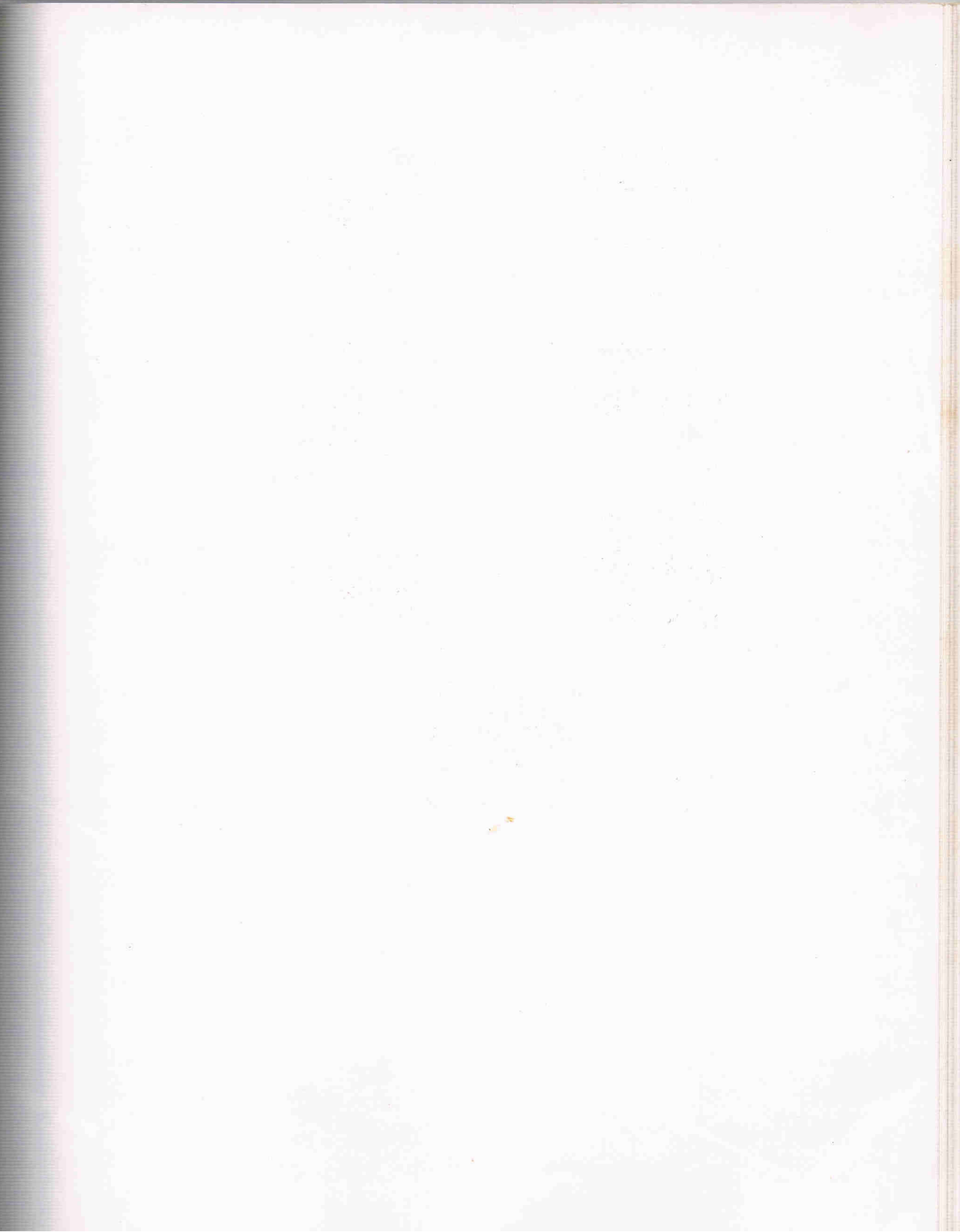
PAJARO DE FRENTE: POSICION 2



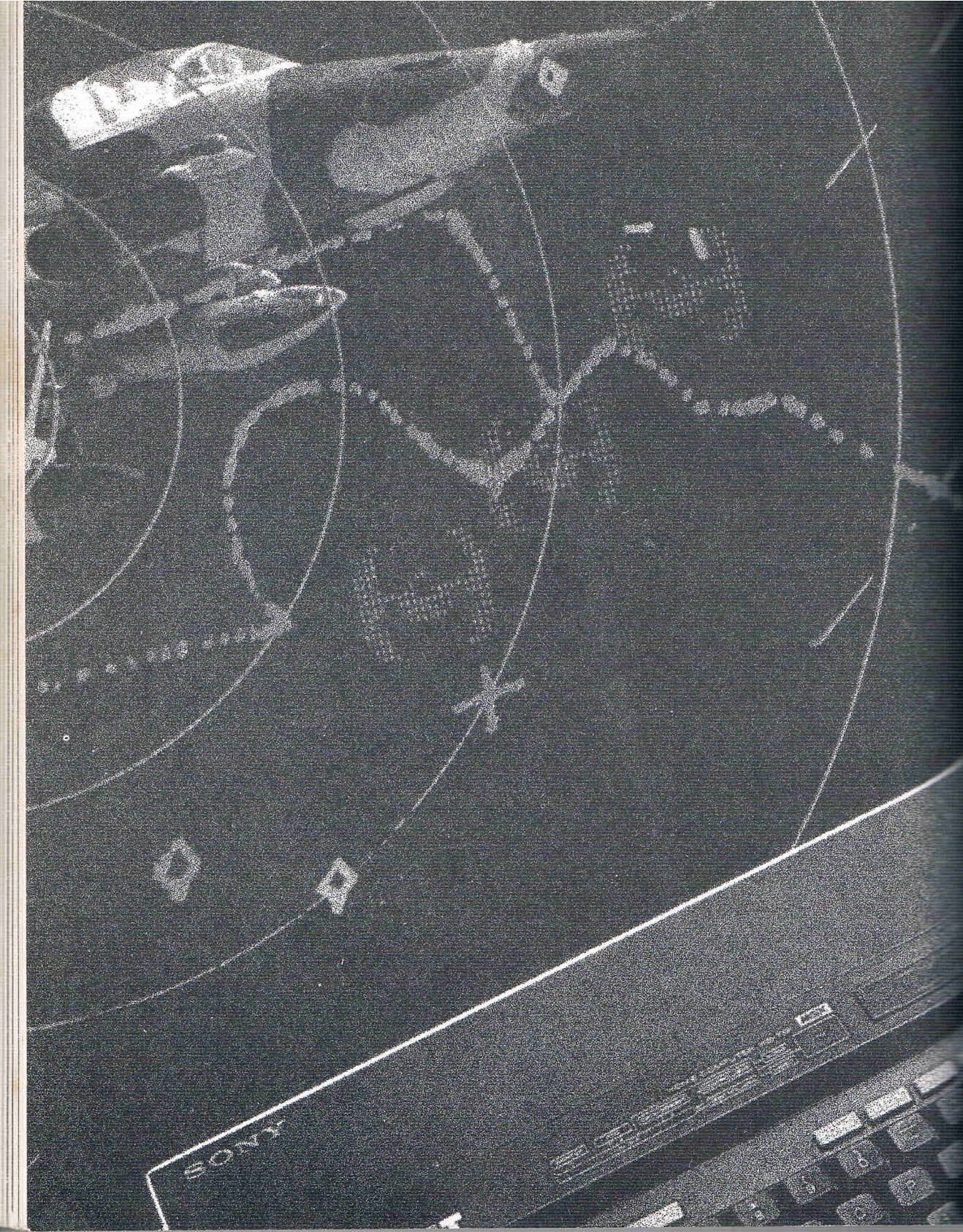
PAJARO DERRIBADO







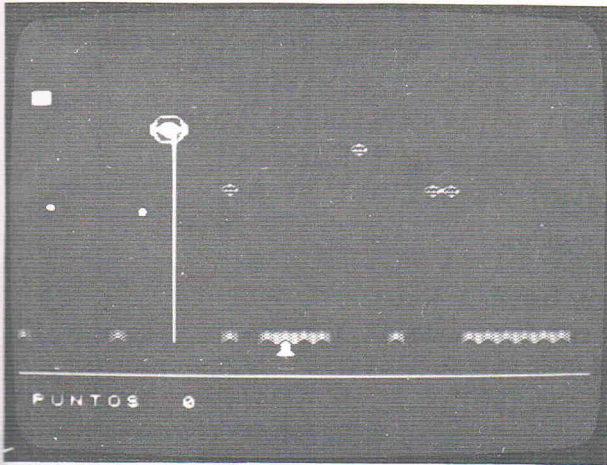






# Batalla laser

# 23



## ¡ATENCIÓN!

Este programa utiliza la subrutina SOPORTE en lenguaje máquina. Véase apéndice C.

**Tipo de juego:**

**JUEGO DE DISPARO**

Este es uno de los juegos más rápidos y que con deslumbrantes efectos laser y super-sonidos conmoverá a los más entusiastas.

Naves guardianas tratan de arrasar tu planeta mediante la destrucción de uno de los tres acumuladores de energía de tu cañón laser.

La nave nodriza provee de energía a las demás, y, si la destruyes, te suma más puntos.

Para crear los efectos laser se usa una subrutina en código máquina.

Emplea la barra espaciadora para disparar y las teclas de cursor para mover el cañón a derecha e izquierda.

# SUGERENCIAS AL PROGRAMA

Para dar velocidad a la nave nodriza, modifica el POKE de la línea 770 a 59999.

Se pueden añadir más o menos naves guardianas, para lo cual debes corregir las rutinas "Posición de las naves guardianas", "Fuego de las naves" y "Vuelo de las naves".

## Variables del programa

FA	Fase (02)
AN(N, I)	Atributos de las naves: N = Número de nave. I = 1 (Columna). I = 2 (Fila). I = 3 (Señal de parada).
CC	Columna del cañón del jugador.
SN	Señal nave herida.
PP	Punto de partida del láser.
NN	Número de la nave que está disparando.
NC	Número de cañones que quedan.

### Inicialización

```
10      REM      ** ATENCION **
20      REM
30      REM ESTE PROGRAMA NECESITA
40      REM LA SUBROUTINA SOPORTE
50      REM Vease el Apendice C
60      REM
70      REM
80      A=PEEK(64586!)+256*PEEK(6458
7!)
90      V1=A+9:V2=A+8:V3=A+7:V4=A+6
100     V5=A+5:V6=A+4:V7=A+3:V8=A+2
:V9=A+1
110     KEY OFF :SCREEN 1,2 : COLOR
15,1,1
120     FOR I=1 TO 8 : READ Q : A$=A
$+CHR$(Q) : NEXT : SPRITE$(0) = A$
: A$=""
```



```

130     FOR I=1 TO 32 : READ Q : A$=
A$+ CHR$(Q) : NEXT : SPRITE$(1) =A$:N
N$=A$
140     DEFUSR0=A+10:DEFUSR1=A+128!:
POKE V4VVV,10: POKE V3VVV,1 :POKE V2V
VV,3: POKE V1VVV,4:DEFUSR2=A+250
150 B=A+10
160 I=A+250
170 READ A$
180 IF A$="R" THEN READ C:C=C+B: C1 =
INT(C/256):C2=256*(C/256-C1):POKE I,
C2:I=I+1:POKE I,C1 :GOTO 200
190 POKE I, VAL(A$)
200 IF I<A+250+28 THEN I=I+1:GOTO 17
0
210     FOR I= 1088 TO 1344 STEP 64:
FOR J=0 TO 7 : READ Q : VPOKE I+J,Q
: NEXT : NEXT
220     PUT SPRITE 1,(100,45),7:PUT
SPRITE 0,(127,100),11:NC=3: ON STRIG
GOSUB 380
230     CLS : PRINT "_____BATALLA
_LASER": FOR I =1 TO 18 : PRINT : NEX
T : PRINT "PULSA_UNA_TECLA_PARA_EMPEZ
AR"
240     IF INKEY$="" THEN D=USR1(D):
FOR I=1 TO 100 : NEXT :PLAY "L9M1000S
14N33": GOTO 230

```

Preparación muro protector

```

250     CLS : PUT SPRITE 1,(100,200)
260     VPOKE 8209,225: VPOKE 8210,4
9: VPOKE 8211,177:VPOKE 8212,97 : VPO
KE 8213,241
270     FOR I=6720 TO 6751 :VPOKE I,
136 : NEXT : FOR I= 6816 TO 6847 : VP
OKE I,23 : NEXT :VPOKE 6787,160: VPOK
E 6799,160 : VPOKE 6812,160

```

```
280      CC=15 :GOSUB 1200 : PUT SPRI  
TE 0,(120,150),15: STRIG(0) ON
```

Control

```
290      POKE V1VVV,8 : D=USR0(D):CC=  
INT(VPEEK(6913)/8)  
300      IF FA=2 THEN GOSUB 1000 ELSE  
      IF FA=0 THEN GOSUB 610 : GOTO 320  
310      GOSUB 750  
320      IF RND(1)<.01 AND FA=1 THEN  
FA=2  
330      IF FA=0 THEN 360  
340      IF RND(1)<.8 OR FA=2 THEN 36  
0  
350      GOSUB 500  
360      IF VPEEK(6912)=200 THEN 1060  
370      GOTO 290
```

Disparos del jugador

```
380      D=RND(1) : STRIG(0) OFF :PLAY  
"19m1000s14n33"  
390      SN=0:I2=0:FOR I1=1 TO 4 : IF  
AN(I1,1)=CC THEN I2=I1  
400      NEXT : POKE V9VVV,152  
410      IF I2<>0 THEN PP=6144+32*AN(I  
2,2)+CC :SN=2 : GOTO 450  
420      IM=VPEEK (6917) :IP=CC*8  
430      IF IP<IM+2 OR IP-IM>6 OR VP  
K (6916)=200 THEN PP=6144+CC: GOTO 45  
0  
440      SN=1:PP=6208+CC  
450      POKE V8VVV,PP MOD 256 : POKE  
V7VVV,INT(PP/256): I3=6720 + CC: POKE  
V6VVV,I3 MOD 256: POKE V5VVV,INT(I3/  
256):D=USR2(D)  
460      POKE V9VVV,32:FOR I3=1 TO 20
```



```
: NEXT : D=USR2(D)
470 IF SN=1 THEN GOSUB 920
480 IF SN=2 THEN GOSUB 970
490 STRIG(0) ON : RETURN
```

Disparo nave guardiana

```
500 IF AN(1,1)=0 AND AN(2,1)=0 AND AN(3,1)=0 AND AN(4,1)=0 THEN RETURN
ELSE STRIG(0) OFF
510 NN=NN+1: IF NN=5 THEN NN=1
520 IF AN(NN,1)=0 THEN 510
530 FL=0 : K1=0 : POKE V9VVV,152 :
I=6176+32*AN(NN,2)+AN(NN,1) : POKE V
8VVV,I MOD 256 : POKE V7VVV,INT(I/256
)
540 IF VPEEK(6720+AN(NN,1))=136 T
HEN J=6752+AN(NN,1) : GOTO 570
550 J=6816 + AN(NN,1) : IF AN(NN,
1)=CC THEN K1=1
560 K=AN(NN,1) : IF K=3 OR K=15 O
R K=28 THEN FL=1
570 POKE V6VVV,J MOD 256: POKE V5
VVV,INT(J/256): D=USR2(D) : IF FL=1 T
HEN FOR J=1 TO 10 : COLOR 1,15,15 : F
OR T=1 TO 30 :NEXT : PLAY "139m59000s
8n2":COLOR 15,1,1 : FOR T= 1 TO 30 :N
EXT : NEXT : GOTO 1060
580 IF K1=1 THEN PUT SPRITE 0,(10
0,200) : FOR K=1 TO 7 : PLAY "164m100
0s14n20n21n20n21": NEXT
590 PLAY "119m380s10n50":POKE V9V
VV,32:D=USR2(D)
600 STRIG(0) ON : RETURN
```

Posición de la nave guardiana

```
610 IF AN(1,1)<>0 THEN 630
620 I=INT(RND(1)*31+1): IF I=AN(2
```

```

,1) OR I=AN(3,1) OR I=AN(4,1) THEN 62
0 ELSE AN(1,1)=I : AN(1,2)=0 : VPOKE
6144+I,144: RETURN
630     IF AN(2,1)<>0 THEN 650
640     I=INT(RND(1)*31+1) : IF I=AN(
1,1) OR I=AN(3,1) OR I=AN(4,1) THEN 6
40 ELSE AN(2,1)=I : AN(2,2)=0 : VPOKE
6144+I,144 : RETURN
650     IF AN(3,1)<>0 THEN 670
660     I=INT(RND(1)*31+1) : IF I=AN(
1,1) OR I=AN(2,1) OR I=AN(4,1) THEN 6
60 ELSE AN(3,1)=I : AN(3,2)=0 : VPOKE
6144+I,144: RETURN
670     IF AN(4,1)<>0 THEN 690
680     I=INT(RND(1)*31+1) : IF I=AN(
1,1) OR I=AN(2,1) OR I=AN(3,1) THEN 6
80 ELSE AN(4,1)=I : AN(4,2)=0 : VPOKE
6144+I,144: RETURN
690     IF AN(1,3)=1 AND AN(2,3)=1 AN
D AN(3,3)=1 AND AN(4,3)=1 THEN FA=1:
RETURN
700     J=INT(RND(1)*4+1) : IF AN(J,3
) =1 THEN 700
710     VPOKE AN(J,1)+32*AN(J,2)+6144
,32 : AN(J,2)=AN(J,2)+1
720     IF AN(J,1)<> 0 THEN VPOKE AN(
J,1)+32*AN(J,2)+6144,144
730     IF RND(1)<.1 OR AN(J,2)>=7 TH
EN AN(J,3)=1
740     RETURN

```

Movimiento nave nodriza

```

750     IF VPEEK(6916)<>200 THEN 770
760     MC=2 : PUT SPRITE 1,(0,15): R
ETURN
770     POKE V3VVV,1 : POKE V2VVV,3:P
OKE V1VVV,6: D=USR1(D):IF VPEEK (6917
)>251 THEN PUT SPRITE 1,(200,200):RET
URN

```



```

780   MC=INT((VPEEK(6917)+8)/8): IF
      RND(1)<.85 THEN RETURN
790   IF MC<>AN(1,1) AND MC<>AN(2,1
      ) AND MC<>AN(3,1) AND MC<>AN(4,1) TH
      EN GOSUB 810
800   RETURN

```

Disparo nave nodriza

```

810   PLAY "124m160s8n67"
820   STRIG(0) OFF : FL=0 : K1 = 0
      : POKE V9VVV,152 : I=6240+MC : POKE V
      8VVV, I MOD 256 : POKE V7VVV,INT(I/25
      6)
830   IF VPEEK (MC +6720) =136 THEN
      J=6752+MC : GOTO 860
840   J=6816 +MC : IF MC = 3 OR MC
      = 15 OR MC = 28 THEN FL = 1
850   IF MC = CC THEN K1= 1
860   POKE V6VVV, J MOD 256 : POKE
      V5VVV,INT(J/256): D = USR2 (D)
870   IF FL=1 THEN FOR J=1 TO 10 :
      COLOR 1,15,15 : FOR T =1 TO 30 : NEXT
      : PLAY "139m59000s8n2": COLOR 15,1,1
      : FOR T =1 TO 30 : NEXT : NEXT : GOTO
      1060
880   PLAY "124m160s8n67"
890   IF K1=1 THEN PUT SPRITE 0,(10
      0,200) : FOR J= 1 TO 7 : PLAY "164m1
      000s14n20n21n20n21": NEXT
900   POKE V9VVV,32 : D=USR2(D)
910   STRIG(0) ON : RETURN

```

Nave nodriza derribada

```

920   SC = SC+50 : GOSUB 1200
930   PLAY "164m60000s8n20n21n24n28
      n40n45n43n29n20n1513n10"

```

```
940   FOR I4=1 TO 60 : VPOKE 14368
      + INT(RND(1)*30), INT(RND(1)*255) : N
      EXT:SPRITE$(1)=NN$
950   PUT SPRITE 1, (200,200)
960   RETURN
```

Nave guardianas derribada

```
970   VPOKE 6144+AN(I2,1)+32* AN(I2
      ,2),168: SC = SC + 10 : GOSUB 1200 :
      FOR T = 1 TO 30 : NEXT
980   VPOKE 6144+AN(I2,1)+32*AN(I2,
      2),32 : AN(I2,1)=0
990   RETURN
```

Posición de naves guardianas (fase anterior)

```
1000  IF AN(1,1)<> 0 OR AN(2,1)<>0
      OR AN(3,1)<> 0 OR AN(4,1)<>0 THEN ST
      RIG(0) OFF : GOTO 1020
1010  FOR J= 1 TO 4 : AN(J,3)=0 :
      NEXT : FA =0 : RETURN
1020  J= INT(RND(1)*4+1) : IF AN (
      J,1)=0 THEN 1020 ELSE STRIG(0) ON
1030  VPOKE 6144 +AN(J,1)+32 * AN
      (J,2),32 : IF AN(J,2) > 0 THEN AN(J,2
      )= AN(J,2)-1
1040  VPOKE 6144 + AN(J,1) + 32 *
      AN(J,2),144 : IF AN(J,2) = 0 THEN VPO
      KE 6144 + AN(J,1) , 32: AN( J,1)= 0
1050  RETURN
```

Nuevo juego

```
1060  IF VPEEK(6912)<>200 THEN 114
      0
1070  STRIG(0) OFF : FOR T=1 TO 15
      00 : NEXT
```



```

1080     CLS : PUT SPRITE 0,(100,200)
        : PUT SPRITE 1,(100,200): PRINT"____
____BATALLA_LASER": PRINT : PRINT : P
RINT
1090     NC=NC-1 : IF NC=0 THEN 1170
1100     PRINT "____HOMBRES_VIVOS:___
"; NC
1110     FOR J=1 TO 2500: NEXT
1120     FOR J=1 TO 4 : AN(J,1) = 0 :
NEXT : FA = 0
1130     GOTO 250
1140     STRIG(0) OFF: FOR T=1 TO 150
0: NEXT
1150     CLS : PUT SPRITE 0,(100,200)
        : PUT SPRITE 1,(100,200) : PRINT "____
____BATALLA_LASER": PRINT : PRINT :
PRINT
1160     PRINT "____!!!PLANTA_DESTRUID
A!!!":PRINT:PRINT:GOTO 1090

```

Fin de juego

```

1170     PRINT : PRINT : PRINT
1180     PRINT "____TUS_PUNTOS_SON:"
;SC
1190     IF INKEY$="" THEN END ELSE 1
190

```

Puntuación

```

1200     FOR I=1 TO 22: PRINT : NEXT :
PRINT : PRINT "PUNTOS:";SC;CHR$(11);
1210     RETURN

```

Datos de sprites

```

1220     DATA 24,60,24,60,24,52,122,255
1230     DATA 15,16,32,32,99,198,203,25

```

4, 255, 207, 199, 99, 33, 48, 16, 15, 240, 8, 4,  
4, 198, 99, 211, 127, 255, 243, 227, 198, 132,  
12, 8, 240

Datos de código máquina

1240 DATA 42, R, -8, 58, R, -9, 79, 205, R,  
204, 17, 32, 0, 25, 237, 91, R, -6, 124, 186, 19  
4, R, 247, 125, 187, 194, R, 247, 201

Datos de sprites pequeños

1250 DATA 90, 165, 90, 165, 90, 165, 66, 1  
29

1260 DATA 24, 126, 213, 171, 255, 66, 60,  
24

1270 DATA 24, 24, 24, 24, 24, 24, 24, 24

1280 DATA 24, 36, 90, 165, 219, 165, 219,  
255

1290 DATA 2, 144, 4, 17, 64, 4, 161, 8

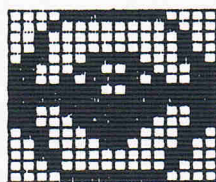
## Tabla de claves

10	= 904	130	= 3784	250	= 1112
20	= 143	140	= 6131	260	= 2541
30	= 1637	150	= 636	270	= 5432
40	= 1541	160	= 883	280	= 2515
50	= 1712	170	= 236	290	= 3079
60	= 143	180	= 6472	300	= 2636
70	= 143	190	= 854	310	= 395
80	= 2571	200	= 2548	320	= 2226
90	= 2998	210	= 3331	330	= 867
100	= 3762	220	= 3124	340	= 2067
110	= 1176	230	= 4951	350	= 400
120	= 3669	240	= 4159	360	= 1408

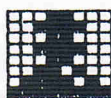


370	=	186	690	=	3945	1010	=	2213
380	=	2820	700	=	3033	1020	=	3979
390	=	3339	710	=	3048	1030	=	4022
400	=	953	720	=	3126	1040	=	4530
410	=	3591	730	=	2924	1050	=	142
420	=	1789	740	=	142	1060	=	1727
430	=	4691	750	=	1616	1070	=	1939
440	=	1366	760	=	1272	1080	=	4104
450	=	6832	770	=	5362	1090	=	1769
460	=	2376	780	=	3529	1100	=	1666
470	=	1085	790	=	5005	1110	=	1100
480	=	1136	800	=	142	1120	=	2011
490	=	865	810	=	1121	1130	=	401
500	=	4453	820	=	5849	1140	=	1939
510	=	2055	830	=	2890	1150	=	4104
520	=	1324	840	=	3466	1160	=	2349
530	=	6184	850	=	1255	1170	=	551
540	=	3451	860	=	3169	1180	=	1685
550	=	2821	870	=	5904	1190	=	1432
560	=	3103	880	=	1121	1200	=	2966
570	=	9131	890	=	4466	1210	=	142
580	=	4467	900	=	1366	1220	=	1358
590	=	2586	910	=	865	1230	=	5453
600	=	865	920	=	1238	1240	=	4799
610	=	1290	930	=	3487	1250	=	1487
620	=	7185	940	=	5160	1260	=	1466
630	=	1311	950	=	995	1270	=	1256
640	=	7206	960	=	142	1280	=	1532
650	=	1332	970	=	4272	1290	=	1165
660	=	7227	980	=	2591			
670	=	1353	990	=	142			
680	=	7248	1000	=	5515	TOTAL	=	339252

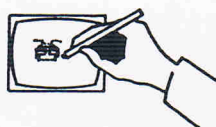
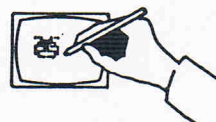
## Diseño de los sprites



NAVE NODRIZA



LASER









# En busca del tesoro

# 24



Tipo de juego:

JUEGO DE MESA

Es una carrera para encontrar los diamantes antes que tu oponente. Hallarás rubíes, esmeraldas y ópalos a lo largo del camino. Puede ser mejor ir primero a por los rubíes. Como táctica, es una buena forma de ganar el juego.

Puntuación:

Diamantes	1.000
Rubíes	750
Esmeraldas	500
Opalos	250
Serpiente	-100

# SUGERENCIAS AL PROGRAMA

Hay dos tácticas principales en este juego: ir derecho por los diamantes o pasar primero por una esquina a recoger algunos rubíes. Puedes aumentar las opciones técnicas reorganizando la pantalla y añadiendo algunas gamas diferentes. No es un trabajo difícil, ya que la disposición de la pantalla se controla únicamente en la rutina que va de la línea 160 a la 380.

## Variables del programa

PJ(1) Posición jugador 1.  
P(1) Puntuación jugador 1.  
P\$(1) Nombre del jugador.  
M\$(1) Mensaje a los jugadores.  
JA Jugador en acción.  
CJ Caracteres bajo el jugador.

### Inicialización

```
10      COLOR 15,4,7: SCREEN 1,2: KE
YOFF
20      VPOKE 8210,180: VPOKE 8211,3
6: VPOKE 8212,132: VPOKE 8213,20
30      FOR I=1 TO 8: READ Q: A#=A#+C
HR$(Q): NEXT: SPRITE$(0)=A$:SPRITE$(1
)=A$
40      FOR I=1072 TO 1079: READ Q: V
POKE I,Q: NEXT: FOR I=1152 TO 1344 ST
EP 64: FOR J=0 TO 7: READ Q: VPOKE I+
J,Q: NEXT: NEXT: FOR I=1112 TO 1143:
READ Q : VPOKE I,Q: NEXT
50      PRINT"____EN_BUSCA_DEL_TESOR
0"
60      PRINT"__Gema____Valor_(mill
ones)";: PRINT"____-____-____-____-
____-";
70      PRINT: PRINT: PRINT"Opalos" T
AB(11)CHR$(144)TAB(19)"250"
80      PRINT: PRINT"Esmeraldas" TAB(
```



```

11)CHR$(152)TAB(19)"500"
90 PRINT: PRINT"Rubies" TAB(11)C
HR$(160)TAB(19)"750"
100 PRINT: PRINT"Diamantes"TAB(1
1)CHR$(139)CHR$(140)TAB(18)"1000":PRI
NT TAB(11)CHR$(141)CHR$(142)
110 PRINT: PRINT"Serpiente" TAB(
11)CHR$(168)TAB(18)"-100"
120 PRINT: PRINT: P$(1)="RICARDO
_":INPUT"Jugador 1";P$(1):PRINT: P$(2
)="JULIO_": INPUT"Jugador 2"; P$(2)
130 IF LEN(P$(2))>9 OR LEN(P$(1)
)>9 THEN PRINT"MUY_LARGO": GOTO 120 E
LSE PRINT : PRINT"VALE:_ " P$(1) " Y "
P$(2)
140 FOR I=1 TO 20: PRINT: NEXT:
PRINT"_____PULSA_UNA_TECLA"
150 IF INKEY$="" THEN D=RND(1):
GOTO 150

```

Preparación pantalla

```

160 CLS: VPOKE 6511,139: VPOKE 65
12,140: VPOKE 6543,141: VPOKE 6544,14
2
170 FOR I=6241 TO 6244: VPOKE I,
134:NEXT: FOR I=6267 TO 6270: VPOKE I
,134: NEXT: VPOKE 6273,134: FOR I=627
6 TO 6287 : VPOKE I,134: NEXT: FOR I=
6289 TO 6299: VPOKE I,134: NEXT: VPOK
E 6302,134
180 VPOKE 6305,134: VPOKE 6308,13
4: VPOKE 6317,134: VPOKE 6319,134: VP
OKE 6321,134: VPOKE 6323,134: VPOKE 6
331,134:VPOKE 6334,134
190 VPOKE 6337,134: VPOKE 6340,13
4: VPOKE 6341,134: VPOKE 6342,134: VP
OKE 6347,134: VPOKE 6348,134: VPOKE 6
349,134:VPOKE 6351,134:VPOKE 6352,134

```

:VPOKE 6353,134: VPOKE 6355,134 :VPOK  
E 6361,134  
200 VPOKE 6362,134: VPOKE 6363,13  
4: VPOKE 6366,134  
210 VPOKE 6369,134:VPOKE 6374,134  
:VPOKE 6379,134:VPOKE 6384,134: VPOKE  
6387,134: VPOKE 6388,134: VPOKE 6389  
,134: VPOKE 6393,134: VPOKE 6398,134  
220 VPOKE 6401,134: FOR I=6404 TO  
6411: VPOKE I,134: NEXT: VPOKE 6414,1  
34: VPOKE 6415,134: VPOKE 6416,134: V  
POKE 6421,134: VPOKE 6425,134: VPOKE  
6430,134  
230 FOR I=6433 TO 6436: VPOKE I,1  
34: NEXT: VPOKE 6440,134: VPOKE 6446,  
134: FOR I=6453 TO 6459: VPOKE I,134:  
NEXT: VPOKE 6462,134  
240 VPOKE 6466,134: FOR I=6468 TO  
6474: VPOKE I,134: NEXT: FOR I=6478 T  
O 6483: VPOKE I,134:NEXT: VPOKE 6487,  
134: FOR I=6491 TO 6494: VPOKE I,134:  
NEXT  
250 VPOKE 6498,134:VPOKE 6501,134  
: VPOKE 6506,134: VPOKE 6507,134:VPOK  
E 6508,134:VPOKE 6510,134:VPOKE 6513,  
134: VPOKE 6515,134: FOR I=6517 TO 65  
23:VPOKE I,134: NEXT: VPOKE 6525,134  
260 VPOKE 6530,134: FOR I=6532 TO  
6538: VPOKE I,134: NEXT: VPOKE 6540,1  
34: VPOKE 6542,134: VPOKE 6545,134: V  
POKE 6547,134: VPOKE 6548,134: VPOKE  
6549,134: VPOKE 6554,134: VPOKE 6557,  
134  
270 VPOKE 6562,134:VPOKE 6563,134  
: VPOKE 6564,134: VPOKE 6568,134: FOR  
I=6572 TO 6577: VPOKE I,134: NEXT: F  
OR I=6581 TO 6587: VPOKE I,134: NEXT:  
VPOKE 6589,134  
280 VPOKE 6594,134: FOR I=6596 TO  
6602: VPOKE I,134: NEXT: VPOKE 6609,1  
34: VPOKE 6615,134: VPOKE 6619,134: V



```

POKE 6620,134: VPOKE 6621,134
290   VPOKE 6626,134:VPOKE 6628,134
: VPOKE 6631,134: VPOKE 6634,134:VPOK
E 6635,134:VPOKE 6636,134: VPOKE 6639
,134: VPOKE 6640,134: VPOKE 6641,134:
FOR I=6644 TO 6651: VPOKE I,134:NEXT
: VPOKE 6653,134
300   VPOKE 6658,134:VPOKE 6660,134
: VPOKE 6663,134: VPOKE 6668,134:VPOK
E 6671,134:VPOKE 6676,134: VPOKE 6680
,134: VPOKE 6683,134: VPOKE 6685,134
310   FOR I=6689 TO 6692: VPOKE I,1
34:NEXT:VPOKE 6695,134:VPOKE 6696,134
:VPOKE 6697,134:VPOKE 6700,134:VPOKE
6702,134:VPOKE 6703,134:VPOKE 6704,13
4:VPOKE 6706,134:VPOKE 6707,134:VPOKE
6708,134:VPOKE 6711,134:VPOKE 6712,13
4
320   FOR I=6715 TO 6718: VPOKE I,
134: NEXT
330   VPOKE 6721,134:VPOKE 6724,134
: VPOKE 6729,134: VPOKE 6732,134:VPOK
E 6734,134:VPOKE 6736,134: VPOKE 6738
,134: VPOKE 6743,134: VPOKE 6747,134:
VPOKE 6750,134
340   FOR I=6753 TO 6766: VPOKE I,1
34: NEXT: FOR I=6768 TO 6782: VPOKE I
,134: NEXT
350   VPOKE 6792,134:VPOKE 6797,134
: VPOKE 6801,134: VPOKE 6806,134: FOR
I=6829 TO 6833: VPOKE I,134: NEXT
360   VPOKE 6241,160:VPOKE 6270,160
: FOR I=6242 TO 6908: IF VPEEK(I)=134
THEN K=RND(1): IF K<.05 THEN VPOKE I
,168 ELSE IF K<.1 THEN VPOKE I,152 EL
SE IF K<.15 THEN VPOKE I,144
370   NEXT: VPOKE 6530,160: VPOKE 6
589,160: VPOKE 6278,160: VPOKE 6297,1
60
380   M$(1)="Tu turno   ": M$(2)="
Espera   ": GOSUB 840: JA=1: Q=1:

```

```
PUT SPRITE 0,(64,160),1:PUT SPRITE 1,  
(176,160),12:CJ(1)=134: CJ(2)=134: PJ  
(1)=6792: PJ(2)=6806
```

Editor

```
390 X$=INKEY$: C=C+1  
400 IF C=6 THEN Q=1-Q: C=0  
410 IF Q<>0 THEN 430  
420 Q=1: K=VPEEK(PJ(JA)): IF K=32  
THEN VPOKE PJ(JA),CJ(JA) ELSE VPOKE P  
J(JA),32  
430 IF X$="" THEN 390  
440 K=ASC(X$)-27: IF K<1 OR K>4 T  
HEN 390 ELSE ON K GOTO 450,480,510,54  
0
```

Test de movimientos permitidos

```
450 IF VPEEK(PJ(JA)+1)=32 THEN 39  
0  
460 VPOKE PJ(JA),CJ(JA): CJ(JA)=V  
PEEK(PJ(JA)+1): PJ(JA)=PJ(JA)+1:VPOKE  
6909+4*JA,VPEEK(6909+4*JA)+8  
470 GOTO 560  
480 IF VPEEK(PJ(JA)-1)=32 THEN 39  
0  
490 VPOKE PJ(JA),CJ(JA): CJ(JA)=V  
PEEK(PJ(JA)-1): PJ(JA)=PJ(JA)-1:VPOKE  
6909+4*JA,VPEEK(6909+4*JA)-8  
500 GOTO 560  
510 IF VPEEK(PJ(JA)-32)=32 THEN 3  
90  
520 VPOKE PJ(JA),CJ(JA): CJ(JA)=V  
PEEK(PJ(JA)-32):PJ(JA)=PJ(JA)-32: VPO  
KE 6908 + 4*JA,VPEEK(6908+4*JA)-8  
530 GOTO 560
```



```
540 IF VPEEK(PJ(JA)+32)=32 THEN 3
90
550 VPOKE PJ(JA),CJ(JA): CJ(JA)=V
PEEK(PJ(JA)+32):PJ(JA)=PJ(JA)+32: VPO
KE 6908 + 4*JA,VPEEK(6908+4*JA)+8
```

Comprobación de movimiento

```
560 K=CJ(JA): IF K=134 THEN GOSUB
740 ELSE IF K=144 THEN GOSUB 650 ELSE
IF K=152 THEN GOSUB 680 ELSE IF K=16
0 THEN GOSUB 620 ELSE IF K=168 THEN G
OSUB 710 ELSE GOSUB 780
570 K=CJ(JA) : ON K GOSUB 740,650
,680,620,710,780,780,780,780
580 IF JA=1 THEN JA=2 ELSE JA=1
590 M$(JA)="Tu turno ": IFJA=
1 THEN M$(2)="Espera " ELSE M$(
1)="Espera "
600 GOSUB 840
610 GOTO 390
```

Rubies

```
620 M$(JA)="RUBIES+750":P(JA)=P(J
A)+750:GOSUB 840: CJ(JA)=134: VPOKE P
J(JA),134
630 PLAY"14s8m30000n4518n4014n45"
,"14s8m20000n3618n3114n36": FOR T=1 T
O 1000: NEXT
640 RETURN
```

Opalos

```
650 M$(JA)="OPALOS+250":P(JA)=P(J
A)+250:GOSUB 840: CJ(JA)=134: VPOKE P
```

```
J(JA),134
660   PLAY"14s14m3000n45": FOR T=1
      TO 1000: NEXT
670   RETURN
```

Esmeraldas

```
680   M$(JA)="ESMER.+500":P(JA)=P(J
      A)+500:GOSUB 840: CJ(JA)=134: VPOKE P
      J(JA),134
690   PLAY"14s14m3000n4012n45": FOR
      T=1 TO 1000: NEXT
700   RETURN
```

Serpiente

```
710   M$(JA)="SERP.-100 ":P(JA)=P(J
      A)-100:GOSUB 840: CJ(JA)=134: VPOKE P
      J(JA),134
720   PLAY"164s8m59000n45n46n47n48n
      49n50n51n51n50n49n48n47n46n45n44n45n4
      6n47n48n49n50n51n50n49n48n47n46n45n44
      13n43": FOR T =1 TO 2000: NEXT
730   RETURN
```

Jugador en casilla normal

```
740   IF RND(1)<.7 THEN RETURN
750   IF RND(1)<.3 THEN M$(JA)="PLA
      TA + 50   ": P(JA)=P(JA)+50: GOSUB
      840: PLAY"16s1m777n34n50n23", "12s5m20
      00n40": FOR T=1 TO 1000: NEXT: RETURN
760   IF RND(1)<.5 THEN M$(JA)="AMA
      TISTA + 20  ": P(JA)=P(JA)+20: GOSUB
      840: PLAY"14s10m500n35n30", "12s1m1000
      n20": FOR T=1 TO 1000: NEXT: RETURN
```



```
770 M$(JA)="ES ORD -50": P(JA)=P(
JA)-50: GOSUB 840: PLAY"11s8m20000n2"
: FOR T=1 TO 1000: NEXT: RETURN
```

Fin de juego

```
780 SCREEN 1: P(JA)=P(JA)+1000
790 PLAY"18s2m30000n45n4614n4512n
4611n37", "s2m3000018n37r814n37r211n33
"
800 IF P(1)>P(2) THEN PRINT P$(1)
" _HA_GANADO" ELSE IF P(2)>P(1) THEN P
RINT P$(2)" _HA_GANADO" ELSE PRINT"EMP
ATADOS"
810 FOR I=1 TO 3: PRINT: NEXT: PR
INT"Las_puntuaciones_son:": PRINT: PR
INT P$(1),P(1): PRINT: PRINT P$(2),P(
2)
820 PRINT: PRINT: PRINT
830 END
```

Puntuación y mensaje

```
840 PRINT P$(1)":": PRINT"PUNTOS:"
;P(1);TAB(12)M$(1): FOR I=1 TO 20: PR
INT: NEXT: PRINT P$(2)":": PRINT "PUN
TOS: ";P(2);TAB(11)M$(2);CHR$(11);
850 RETURN
```

Datos de sprites

```
860 DATA 24,60,126,126,60,36,66,129
```

Datos de sprites pequeños

```
870 DATA 255,213,171,213,171,213,17
1,255
```

880	DATA 66, 231, 66, 0, 0, 66, 231, 66
890	DATA 32, 112, 248, 116, 46, 31, 14, 4
900	DATA 60, 126, 255, 255, 255, 255, 126, 60
910	DATA 2, 7, 12, 76, 140, 134, 195, 126
920	DATA 1, 3, 7, 15, 31, 63, 127, 255
930	DATA 128, 192, 224, 240, 248, 252, 254, 255
940	DATA 255, 127, 63, 31, 15, 7, 3, 1
950	DATA 255, 254, 252, 248, 240, 224, 192, 120

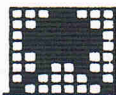
## Tabla de claves

10	= 1185	260	= 7576	510	= 1879
20	= 1760	270	= 7083	520	= 7011
30	= 3935	280	= 6164	530	= 201
40	= 7101	290	= 9354	540	= 1878
50	= 1671	300	= 4609	550	= 7008
60	= 3920	310	= 8152	560	= 7811
70	= 2685	320	= 1599	570	= 2486
80	= 2891	330	= 5766	580	= 1765
90	= 2499	340	= 3459	590	= 5334
100	= 5229	350	= 4412	600	= 230
110	= 2865	360	= 8921	610	= 286
120	= 5373	370	= 2847	620	= 4621
130	= 5398	380	= 8830	630	= 4864
140	= 2609	390	= 1289	640	= 142
150	= 2056	400	= 1728	650	= 4617
160	= 2703	410	= 1122	660	= 2354
170	= 8380	420	= 4143	670	= 142
180	= 5356	430	= 937	680	= 4587
190	= 8402	440	= 4158	690	= 2722
200	= 2104	450	= 1881	700	= 142
210	= 6616	460	= 7016	710	= 4394
220	= 5127	470	= 201	720	= 8807
230	= 4845	480	= 1882	730	= 142
240	= 6189	490	= 7019	740	= 1434
250	= 7217	500	= 201	750	= 8000

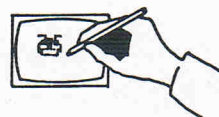


760	=	7940	840	=	6052	920	=	1212
770	=	5290	850	=	142	930	=	1675
780	=	1616	860	=	1421	940	=	1212
790	=	4583	870	=	1661	950	=	1667
800	=	5340	880	=	1268			
810	=	5106	890	=	1358			
820	=	551	900	=	1574	TOTAL	=	352485
830	=	129	910	=	1366			

## Diseño de los sprites



JUGADOR







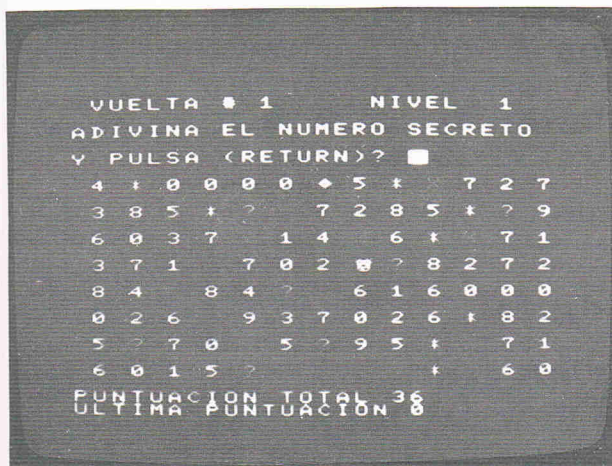
SONY

BIT



# Minotauro Mastermind Jr.

# 25



Tipo de juego:

JUEGO EDUCATIVO

Este es un juego educativo, versión del "Minotauro Mastermind", en el que se han simplificado las pistas para ayudar a los jóvenes jugadores a que encuentren el número secreto entre 10 y 99.

Tu puntuación es la suma de los números sobre los que te posas; por eso son mejores las puntuaciones bajas. Cuando alcanzas la salida en lo alto del bloque tienes una oportunidad de averiguar el número secreto. Los símbolos no numéricos tienen distintos efectos cuando pasas por ellos:

- X Te da una pista.
- ? Tiene un valor aleatorio.
- \* Altera la puntuación aleatoriamente.

La situación se indica mediante un punto intermitente que controlas con:

- |                       |                      |
|-----------------------|----------------------|
| <P> Arriba.           | <:> Abajo/derecha.   |
| <0> Arriba/izquierda. | <a> Derecha.         |
| <;> Abajo.            | <L> Abajo/izquierda. |
| <O> Izquierda.        | <T> Atrapado.        |
| <-> Arriba/derecha.   |                      |

## SUGERENCIAS AL PROGRAMA

Puedes modificar la complejidad del juego añadiéndole varios niveles de dificultad. Una posibilidad es simplificar el cuadro que hay en pantalla (líneas 140-220), de forma que haya menos números y símbolos.

Igual que en el juego original, puedes añadir más pistas (líneas 1060-1300) o cambiar las existentes.

## Variables del programa

ND	Nivel de dificultad.
V	Número de vuelta.
NS	Número secreto.
D1, D2	Dígitos del número secreto.
PJ	Posición jugador.
M	Posición minotauro.
UP, TP	Última puntuación; total puntos.
NA	Número aleatorio.
CB, J	Carácter debajo del minotauro; jugador.
FJ, CJ	Fila y columna del jugador.
NP	Número de pistas.
T(I)	¿Pista utilizada?
G1, G2	Dígitos supuestos por el jugador.
MC, MF	Columna y fila del minotauro.

Inicialización

```
10      GOSUB 1660
20      TM=1: INTERVAL ON: ON INTERV
AL=10 GOSUB 40
30      GOTO 50
40      TM=TM*2: RETURN
50      KEY OFF: SCREEN 1: CLS: COLOR
15,1,1: PRINT"___MINOTAURO_MASTER_M
IND"
60      PRINT: PRINT: PRINT: PRINT "_
NIVEL_DE_DIFICULTAD_(1-9)";
70      ND=VAL(INKEY#): IF ND<1 OR ND
>9 THEN 70 ELSE PRINT ND
80      FOR I=1088 TO 1344 STEP 64: F
```



```

OR J=0 TO 7: READ Q: VPOKE I+J,Q: NEX
T: NEXT
90     VPOKE 8209,49: VPOKE 8210,97:
VPOKE 8211,177: VPOKE 8212,145: VPOK
E 8213,129
100    FOR I=1 TO 8: PRINT: NEXT: P
RINT "PULSA_UNA_TECLA_PARA_EMPEZAR"
110    IF INKEY$="" THEN 110
120    INTERVAL OFF: IF TM>1000 THE
N TM=TM/3: GOTO 120
130    TP=0: NP=0: FOR I=1 TO 6: T(
I)=0: NEXT: V=1: NS=INT(89*TM/1000+10
): N=NS*100: D1=INT(NS/10): D2=INT(NS
-D1*10)

```

Nueva jugada

```

140    CLS: PJ=6799: M=6607
150    FOR I=6336 TO 6784 STEP 64: F
OR J=3 TO 27 STEP 2: GOSUB 1450: K1=I
NT(NA*13): IF K1<10 THEN VPOKE I+J,K1
+48: GOTO 190CONT
160    IF K1=10 THEN VPOKE I+J,144:
GOTO 190
170    IF K1=11 THEN VPOKE I+J,152:
GOTO 190
180    VPOKE I+J,160
190    NEXT: NEXT
200    VPOKE PJ,48: UP=0: CM=VPEEK(M
): VPOKE M,136: VPOKE 6351,168: CJ=48
210    PRINT "_VUELTA_#";V;" NIVE
L ";ND; CHR$(11);
220    GOSUB 1500

```

Editor

```

230    C=0:Q=1
240    X$=INKEY$: C=C+1

```

```

250   IF C=7 THEN Q=1-Q: C=0
260   IF Q<>0 THEN 290
270   IF VPEEK(PJ)=255 THEN VPOKE P
J,J: Q=1: GOTO 290
280   VPOKE PJ,255: Q=1
290   IF X$="" THEN 240
300   GOSUB 1480
310   IF VPEEK(PJ)=255 THEN VPOKE P
J,J
320   IF X$="F" OR X$="p" THEN 420
330   IF X$="O" OR X$="o" THEN 460
340   IF X$="A" OR X$="a" THEN 500
350   IF X$=";" THEN 540
360   IF X$="-" THEN 580
370   IF X$="0" THEN 620
380   IF X$="L" OR X$="1" THEN 660
390   IF X$=":" THEN 700
400   IF X$="T" OR X$="t" THEN 740
410   GOTO 240

```

Test de movimientos permitidos

```

420   IF VPEEK(PJ-64)=32 THEN 240
430   UP=VPEEK(PJ)-48
440   VPOKE PJ,32: PJ=PJ-64
450   GOTO 760
460   IF VPEEK(PJ-2)=32 THEN 240
470   UP=VPEEK(PJ)-48
480   VPOKE PJ,32: PJ=PJ-2
490   GOTO 760
500   IF VPEEK(PJ+2)=32 THEN 240
510   UP=VPEEK(PJ)-48
520   VPOKE PJ,32: PJ=PJ+2
530   GOTO 760
540   IF VPEEK(PJ+64)=32 THEN 240
550   UP=VPEEK(PJ)-48
560   VPOKE PJ,32: PJ=PJ+64
570   GOTO 760
580   IF VPEEK(PJ-62)=32 THEN 240
590   UP=VPEEK(PJ)-48

```



```

600 VPOKE PJ,32: PJ=PJ-62
610 GOTO 760
620 IF VPEEK(PJ-66)=32 THEN 240
630 UP=VPEEK(PJ)-48
640 VPOKE PJ,32: PJ=PJ-66
650 GOTO 760
660 IF VPEEK(PJ+62)=32 THEN 240
670 UP=VPEEK(PJ)-48
680 VPOKE PJ,32: PJ=PJ+62
690 GOTO 760
700 IF VPEEK(PJ+66)=32 THEN 240
710 UP=VPEEK(PJ)-48
720 VPOKE PJ,32: PJ=PJ+66
730 GOTO 760
740 FJ=INT(PJ/32): CJ=PJ-FJ*32: G
OSUB 870: IF PJ<>M THEN 740
750 GOTO 1360

```

Comprobación de movimiento

```

760 J=VPEEK(PJ): FJ=INT(PJ/32): C
J=PJ-FJ*32
770 IF (UP>9) OR (UP<0) THEN UP=0
780 K1=VPEEK(PJ): IF K1=152 THEN
GOSUB 1040
790 IF K1=144 THEN GOSUB 1060
800 IF K1=160 THEN GOSUB 1310
810 IF K1=168 THEN 1520
820 K1=VPEEK(PJ)-48: IF K1<0 OR K
1>9 THEN K1=0
830 IF U+K1>=10-ND THEN GOSUB 870
840 TP=TP+K1: GOSUB 1500
850 IF PJ=M THEN 1360
860 GOTO 230

```

Movimiento del minotauro

```

870 VPOKE 8198,31: MF=INT(M/32):
MC=M-MF*32

```

```

880   IF MC<>J THEN 910
890   IF M>FJ THEN MF=MF-2: GOTO 98
0
900   MF=MF+2: GOTO 980
910   IF MF<>FJ THEN 940
920   IF MC>J THEN MC=MC-2: GOTO 98
0
930   MC=MC+2: GOTO 980
940   IF M>PJ THEN 970
950   IF MC<PJ THEN MC=MC+2: GOTO 9
80
960   MC=MC-2: GOTO 980
970   MF=MF-2
980   K=32*MF+MC: VPOKE 8198,241:IF
    VPEEK(K)=136 THEN RETURN
990   VPOKE M,CB
1000  M=K: CB=VPEEK(M)
1010  VPOKE M,136
1020  PLAY "n2"
1030  RETURN

```

Jugador sobre (?)

```

1040  GOSUB 1450:I=INT(NA*10): VPO
KE PJ,I+48
1050  J=I+48: RETURN

```

Jugador sobre (X)

```

1060  VPOKE PJ,48
1070  IF NP>=6 THEN BEEP: GOTO 117
0
1080  GOSUB 1450: K2=INT(NA*6+1):
IF T(K2)=1 THEN 1080
1090  T(K2)=1: NP=NP+1: BEEP
1100  ON K2 GOTO 1110,1130,1170,12
10,1250,1290
1110  PRINT: PRINT: PRINT"La_suma_

```



```

de_los_digitos_del":PRINT"numero_secr
eto_es_:";
1120 PRINT D1+D2: PRINT CHR$(11);
: RETURN
1130 PRINT: PRINT: PRINT"El_numer
o_secreto_es_:";
1140 IF NS>50 THEN PRINT"_Mayor_q
ue_50": GOTO 1160
1150 PRINT"_Menor_o_igual_que_50"
1160 PRINT CHR$(11);: RETURN
1170 GOSUB 1450:K=INT(NA*15)+NS:J
=K-15: IF J<0 THEN J=0
1180 IF K>99 THEN K=99
1190 PRINT: PRINT: PRINT"El_numer
o_esta_entre";J: PRINT"y";K
1200 PRINT CHR$(11);: RETURN
1210 PRINT: PRINT: PRINT"El_prime
r_digito_es"
1220 IF D1<5 THEN PRINT"menor_que
_5":GOTO 1240
1230 PRINT"mayor_o_igual_que_5"
1240 PRINT CHR$(11);: RETURN
1250 PRINT: PRINT: PRINT"El_segun
do_digito_es"
1260 IF D2>5 THEN PRINT "menor_qu
e_5": GOTO 1280
1270 PRINT"mayor_o_igual_que_5"
1280 PRINT CHR$(11);: RETURN
1290 PRINT: PRINT: PRINT"El_produ
cto_de_los_dos": PRINT"digitos_es";
1300 PRINTD1*D2; CHR$(11);: RETUR
N

```

Jugador sobre (\*)

```

1310 VPOKE PJ,48
1320 IF T<11 THEN RETURN
1330 GOSUB 1450: I=INT(NA*100): I
F I>50 THEN T=T-10: PLAY"L64N10N3N6N4
": RETURN

```

```
1340     IF I<10 THEN T=T+50: PLAY"L3
2N10N8N6N4N10N8N6N4": RETURN
1350     T=T+10: PLAY"L64N10N8N6N4":
RETURN
```

Fin de juego

```
1360     CLS: IF M=PJ THEN 1400
1370     FOR I=1 TO 8: PRINTTAB(2*I)"
ENHORABUENA": PRINT: NEXT: PRINT
1380     PRINT"HAS_VENCIDO_AL_MINOTAU
RO": PRINT"TU_PUNTUACION_ES_";TP
1390     PRINT"EN";V;" VUELTAS": GOTO
1420
1400     FOR I=1 TO 5: PRINT TAB(2*I)
"VENCIO_EL_MINOTAURO": PRINT: NEXT: P
RINT
1410     PRINT"El_numero_secreto_era_
";NS
1420     PRINT: PRINT"PULSA_UNA_TECLA
_PARA": PRINT"JUGAR_OTRA_VEZ"
1430     X$=INKEY$: IF X$="" THEN 143
0
1440     RUN 20
```

Número aleatorio

```
1450     NA=(9999*NA+N) MOD 5997!: NA
=NA/5997!
1460     IF NA<.2 THEN N=NA*10000+1
1470     RETURN
```

Mensaje en blanco

```
1480     FOR I=6208 TO 6271: VPOKE I,
32: NEXT
1490     RETURN
```



Puntuación

```
1500   FOR I=1 TO 22: PRINT: NEXT: P
PRINT"PUNTUACION_TOTAL";TP: PRINT"ULTI
MA_PUNTUACION";UP;CHR$(11);
1510   RETURN
```

Adivina el número secreto

```
1520   PRINT: PRINT: PRINT"ADIVINA_E
L_NUMERO_SECRETO": PRINT
1530   INPUT"Y PULSA (RETURN)";K$
1540   I=VAL(K$): IF I=NS THEN 1360
1550   G1=INT(I/10): G2=I-G1*10
1560   CLS: PRINT: IF D1=G1 THEN PRI
NT"El_primer_digito_es_correcto":PRIN
T
1570   IF D2=G2 THEN PRINT"El_segund
o_digito_es_correcto":PRINT
1580   IF D1<>G1 AND D2<>G2 THEN PRI
NT"Ningun_digito_es_correcto"
1590   FOR Z=1 TO 2000: NEXT
1600   V=V+1: GOTO 140
```

Datos de los sprites pequeños

```
1610   DATA 165,126,219,126,102,126,8
2,60
1620   DATA 136,136,80,32,80,136,136,
0
1630   DATA 112,136,8,16,32,0,32,0
1640   DATA 32,168,112,32,112,168,32,
0
1650   DATA 16,56,124,254,124,56,16,0
```

Instrucciones

```

1660  SCREEN 1:CLS:INPUT"QUIERES INS
TRUCCIONES (S/N)?"X$: IF X$="N" OR X
$="n" THEN 1730
1670  CLS: PRINT"PARA_MOVERTE_EN_EL_
LABERINTO_PULSA_:"
1680  PRINT:PRINT"__P-p__ARRIBA": P
RINT"__D-o__IZQUIERDA": PRINT"__@-2_
__DERECHA": PRINT"__:_ABAJO": PRI
NT"__-__ARRIBADERECHA:
1690  PRINT"__@__ARIBA-IZQUIERDA"
: PRINT"__L-l__ABAJO-IZQUIERDA": PRI
NT"__:_ABAJO-DERECHA": PRINT"__T-
t__ATRAPADO"
1700  PRINT:PRINT:PRINT"SI_PISAS_:"
PRINT: PRINT"__X__TENDRAS_UNA_PISTA
": PRINT"__?__SUMARAS_PUNTUACION__
__ALEATORIA": PRINT"__*__ALTERA
RA_TU_PUNTUACION__ALEATORIAMENTE
"
1710  PRINT: PRINT"PULSA_UNA_TECLA_P
ARA_SEGUIR"
1720  IF INKEY$="" THEN 1720
1730  RETURN

```

## Tabla de claves

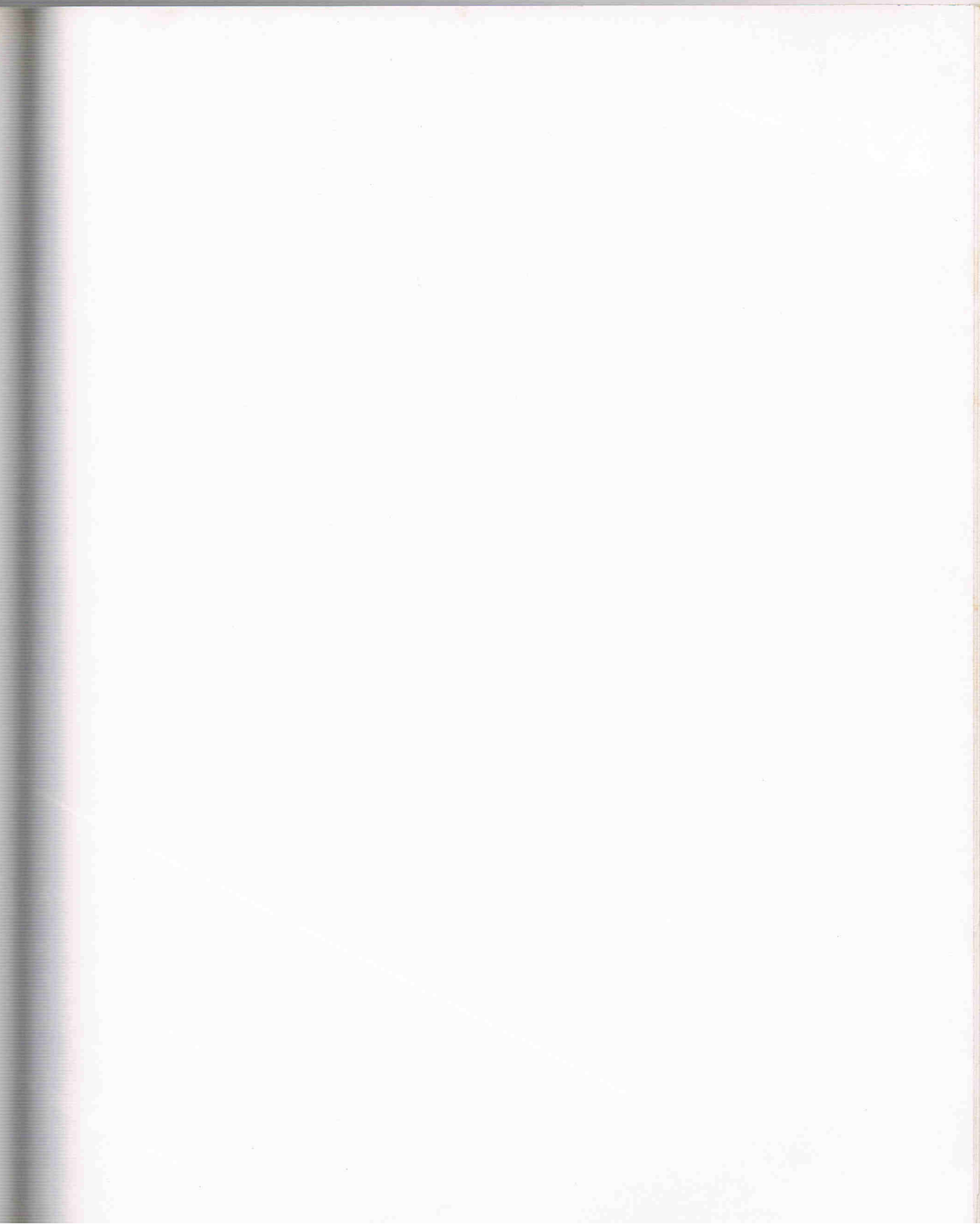
10	= 285	100	= 3295	190	= 320
20	= 3175	110	= 1024	200	= 3715
30	= 201	120	= 3408	210	= 2426
40	= 1023	130	= 8879	220	= 380
50	= 3251	140	= 1441	230	= 719
60	= 2489	150	= 6509	240	= 1289
70	= 3175	160	= 1933	250	= 1729
80	= 3331	170	= 1942	260	= 982
90	= 2349	180	= 805	270	= 2618



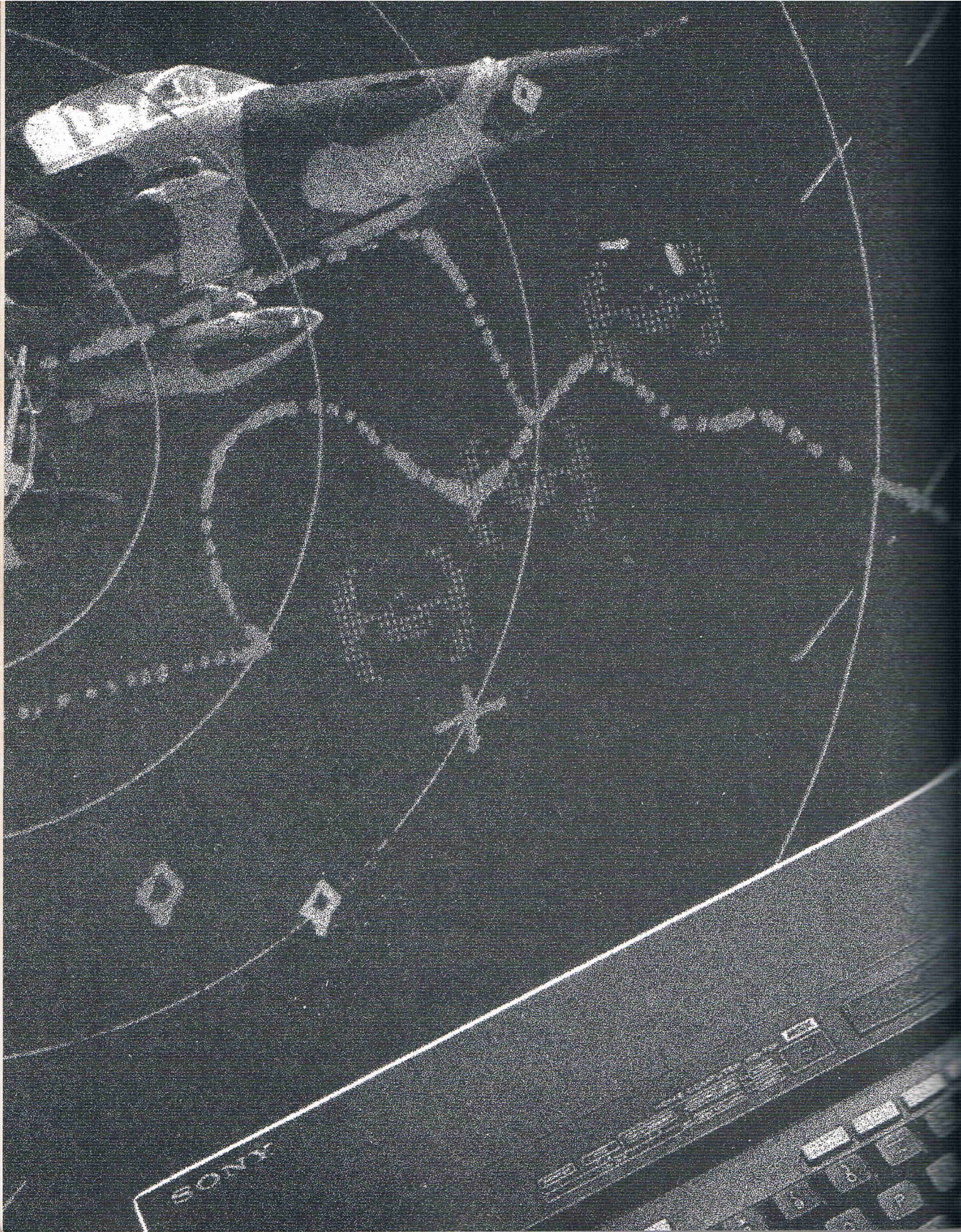
280	= 1062	690	= 401	1100	= 1500
290	= 1042	700	= 1829	1110	= 5108
300	= 360	710	= 1351	1120	= 1595
310	= 1978	720	= 1338	1130	= 2676
320	= 1837	730	= 401	1140	= 2431
330	= 1875	740	= 4187	1150	= 1959
340	= 1887	750	= 236	1160	= 916
350	= 891	760	= 3514	1170	= 3783
360	= 917	770	= 2038	1180	= 1212
370	= 960	780	= 2125	1190	= 3184
380	= 1814	790	= 1074	1200	= 916
390	= 1050	800	= 1085	1210	= 2403
400	= 1910	810	= 1162	1220	= 2371
410	= 391	820	= 3121	1230	= 1918
420	= 1828	830	= 1957	1240	= 916
430	= 1351	840	= 1370	1250	= 2505
440	= 1337	850	= 926	1260	= 2155
450	= 401	860	= 381	1270	= 1918
460	= 1768	870	= 2736	1280	= 916
470	= 1351	880	= 1212	1290	= 3998
480	= 1277	890	= 2034	1300	= 1453
490	= 401	900	= 1217	1310	= 459
500	= 1767	910	= 1315	1320	= 849
510	= 1351	920	= 2025	1330	= 4407
520	= 1276	930	= 1211	1340	= 3246
530	= 401	940	= 1045	1350	= 1945
540	= 1827	950	= 2106	1360	= 1183
550	= 1351	960	= 1212	1370	= 2971
560	= 1336	970	= 794	1380	= 3577
570	= 401	980	= 3204	1390	= 1566
580	= 1826	990	= 452	1400	= 3523
590	= 1351	1000	= 1386	1410	= 2469
600	= 1335	1010	= 470	1420	= 3049
610	= 401	1020	= 421	1430	= 1614
620	= 1830	1030	= 142	1440	= 172
630	= 1351	1040	= 2411	1450	= 3002
640	= 1339	1050	= 890	1460	= 1878
650	= 401	1060	= 459	1470	= 142
660	= 1825	1070	= 1566	1480	= 1531
670	= 1351	1080	= 2925	1490	= 142
680	= 1334	1090	= 1669	1500	= 5185

1510 = 142	1600 = 1019	1690 = 6010
1520 = 2574	1610 = 1566	1700 = 9386
1530 = 1474	1620 = 1413	1710 = 2277
1540 = 1894	1630 = 1199	1720 = 1104
1550 = 2291	1640 = 1405	1730 = 142
1560 = 4350	1650 = 1365	
1570 = 4034	1660 = 4759	
1580 = 4688	1670 = 2871	TOTAL = 327737
1590 = 1126	1680 = 6072	



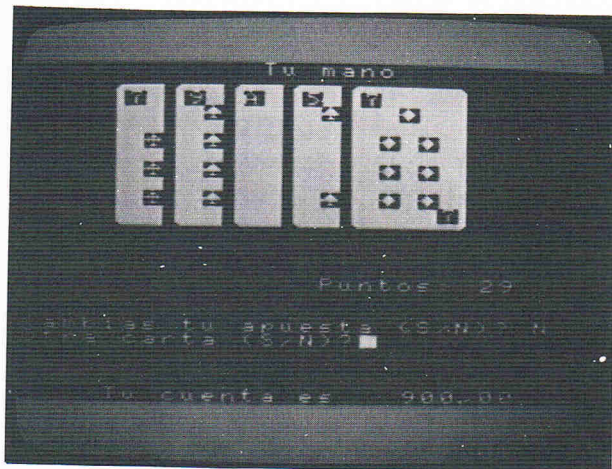








# Treinta y una 26



Tipo de juego:

JUEGO DE MESA

En este juego, el MSX, te desafía a una mano de cartas; se suman los puntos de cada carta y gana el que más cerca esté de 31, pero sin sobrepasarlas, ya que, si lo haces, pierdes (o el MSX, si es éste quien las sobrepasa).

Los puntos de las cartas son:

Rey	13
Dama	12
Jota	11
As	1

Resto de cartas, su número.

# SUGERENCIAS AL PROGRAMA

El MSX actúa de forma sencilla, se planta cuando alcanza 25; pero si, aun así, tú le superas, cambia de táctica memorizando las cartas utilizadas y calculando las probabilidades de la carta siguiente. Puedes incluso hacer un poco de trampa, mirando un instante la siguiente carta (con las líneas 580-620).

## Variables del programa

M1\$, M2\$	Mensajes.
MC(I, J)	Matriz de cartas.
BC(I)	Baraja de cartas.
SC	Siguiente carta de la baraja.
AB	Apuestas de esta baza.
BC	Balance de la cuenta.
PJ	Puntos jugador.
PO	Puntos ordenados.
LC	Localización de la carta
D	Descarte.
CO	Número de cartas del ordenador.
CJ	Número de cartas del jugador.

Inicialización

```
10      KEY OFF: SCREEN 1,2: COLOR 15
      ,4,7
20      FOR I=1 TO 32: READ Q: A$=A$+
CHR$(Q): NEXT: SPRITE$(0)=A$: A$=""
30      FOR I=1 TO 32: READ Q: A$=A$+
CHR$(Q): NEXT: SPRITE$(1)=A$: A$=""
40      FOR I=1 TO 32: READ Q: A$=A$+
CHR$(Q): NEXT: SPRITE$(2)=A$: A$=""
50      FOR I=1072 TO 1143: READ Q: V
POKE I,Q: NEXT
60      PRINT: PRINT: PRINT"_____T
REINTA_Y_UNA": VPOKE 6152,134:FOR I=6
153 TO 6167: VPOKE I,135: NEXT:VPOKE 6
168,136: VPOKE 6200,137: VPOKE 6232,1
37: VPOKE 6184,141: VPOKE 6216,141
```



```

70     FOR I=6281 TO 6295:VPOKE I,13
9: NEXT
80     VPOKE 6264,137: VPOKE 6296,13
8: VPOKE 6248,141: VPOKE 6280,140
90     FOR I=384 TO 463: K=VPEEK(I):
      VPOKE I,INT(K/4): NEXT
100    FOR I=1 TO 6: PRINT: NEXT: C
=100: PRINT"_Empiezas_con_100_pesetas
": PRINT: PRINT"_____Pulsa_una_tecla"
110    IF INKEY$="" THEN D=RND(1):
GOTO 110
120    CLS: DIM MC(52,2),BC(52)
130    FOR I=1 TO 4: FOR J=1 TO 13:
      K=(I-1)*13+J: MC(K,1)=J: MC(K,2)=I:
NEXT: NEXT: FOR I=1 TO 52: BC(I)=I: N
EXT
140    C=100: SC=1
150    PRINT TAB(12)"Tu_mano"
160    PRINT CHR$(11);:M1$="Por_fav
or_espera_que_baraje": M2$="las carta
s.": GOSUB 840: GOSUB
860: GOSUB 880: IF SC=1 THEN GOSUB 6
90
170    GOSUB 580: GOSUB 310:GOSUB 3
10

```

Control
---------

```

180    IF ((ST) AND 1)=1 THEN 190 EL
SE GOSUB 630
190    IF VPEEK(6156)=77 THEN 270 EL
SE IF ((ST) AND 1)<>1 THEN 240
200    FOR I=1 TO 5: BEEP: PRINT CHR
$(11) TAB(28): FOR I2=1 TO 300: NEXT:
      PRINT CHR$(11) TAB(10)"Mano_del_MSX_
_____" CHR$(11);: NEXT: FOR I=6176 TO 6
625: VPOKE I,32: NEXT
210    FOR I=0 TO 2:PUT SPRITE I,(10
0,200):NEXT
220    FOR I1=1 TO CO: IF I1<7 THEN

```

```

LC=6179+3*I1 ELSE LC=6278+3*(I1-7)
230     D=CH(I1): GOSUB 360: NEXT: FO
R I=1 TO 15: PRINT TAB(15)"Puntos:";P
0;CHR$(11);
240     IF ((ST) AND 1)=1 THEN 270
250     M1$="Tu siguiente carta
":M2$="
":
GOSUB 840
260     GOSUB 310: IF PJ>31 THEN 740
270     IF((ST) AND 2)=2 THEN 290 ELS
E M1$="Siguiente carta del MSX ": M
2$="
": GOSU
B 840: FOR T=1 TO 1500: NEXT: GOSUB 5
80
280     IF(PO>24 AND PO<32) THEN ST=(
(ST) OR 2) ELSE IF PO>31 THEN 780
290     IF ST=3 THEN 720
300     GOTO 180

```

Mano del jugador

```

310     D=BC(SC): SC=SC+1
320     PJ=PJ+MC(D,1): GOSUB 860: CJ=
CJ+1
330     IF CJ<7 THEN LC=6179+3*CJ ELS
E LC=6278+3*(CJ-7)
340     GOSUB 360
350     RETURN

```

Dibujar carta

```

360     VPOKE LC,134: FOR I=LC+1 TO L
C+5: VPOKE I,135: NEXT: VPOKE LC+6,13
6: FOR I=LC+33 TO LC+289 STEP 32: FOR
J=0 TO 4: VPOKE I-1,141: VPOKE I+5,1
37: VPOKE I+J,142: NEXT: NEXT
370     VPOKE LC+320,140: FOR I=LC+32
1 TO LC+325: VPOKE I,139: NEXT: VPOKE

```



```

LC+326,138
380      K1=MC(D,2): IF K1=1 THEN K1=3
        ELSE IF K1=2 THEN K1=4 ELSE IF K1=3
        THEN K1=5 ELSE K1=6
390      K2=MC(D,1): ON K2 GOTO 400,41
0,420,430,440,450,460,470,480,490,510
,530,550
400      VPOKE LC+33,65: VPOKE LC+163,
K1: VPOKE LC+293,65: RETURN
410      VPOKE LC+33,50: VPOKE LC+67,K
1: VPOKE LC+259,K1: VPOKE LC+293,50:
RETURN
420      VPOKE LC+33,51: VPOKE LC+67,K
1: VPOKE LC+163,K1: VPOKE LC+259,K1:
VPOKE LC+293,51: RETURN
430      VPOKE LC+33,52: VPOKE LC+293,
52: VPOKE LC+66,K1: VPOKE LC+68,K1: V
POKE LC+258,K1: VPOKE LC+260,K1: RETU
RN
440      VPOKE LC+33,53: VPOKE LC+293,
53: VPOKE LC+66,K1: VPOKE LC+68,K1: V
POKE LC+258,K1: VPOKE LC+260,K1: VPOK
E LC+163,K1: RETURN
450      VPOKE LC+33,54: VPOKE LC+293,
54: VPOKE LC+67,K1: VPOKE LC+130,K1:
VPOKE LC+132,K1: VPOKE LC+194,K1: VPO
KE LC+196,K1: VPOKE LC+259,K1: RETURN
460      VPOKE LC+33,55: VPOKE LC+293,
55: VPOKE LC+67,K1: VPOKE LC+130,K1:
VPOKE LC+132,K1: VPOKE LC+194,K1: VPO
KE LC+196,K1: VPOKE LC+258,K1: VPOKE
LC+260,K1: RETURN
470      VPOKE LC+33,56: VPOKE LC+293,
56: VPOKE LC+66,K1: VPOKE LC+68,K1: V
POKE LC+130,K1: VPOKE LC+132,K1: VPOK
E LC+194,K1: VPOKE LC+196,K1: VPOKE L
C+258,K1: VPOKE LC+260,K1: RETURN
480      VPOKE LC+33,57: VPOKE LC+293,
57: VPOKE LC+163,K1: VPOKE LC+66,K1:
VPOKE LC+68,K1: VPOKE LC+130,K1: VPOK
E LC+132,K1: VPOKE LC+194,K1: VPOKE L

```

```

C+196,K1: VPOKE LC+258,K1: VPOKE LC+2
60,K1: RETURN
490     VPOKE LC+33,49: VPOKE LC+293,
48: VPOKE LC+292,49: VPOKE LC+34,48:
VPOKE LC+99,K1: VPOKE LC+227,K1: VPOK
E LC+66,K1: VPOKE LC+68,K1: VPOKE LC+
130,K1: VPOKE LC+132,K1
500     VPOKE LC+194,K1: VPOKE LC+196
,K1: VPOKE LC+258,K1: VPOKE LC+260,K1
: RETURN
510     I=8*INT((LC-6014)/32): J=8*(L
C-6014-32*I): PUT SPRITE 0,(J+4,I+4):
FOR I=LC+130 TO LC+194 STEP 32: FOR
J=0 TO 2: VPOKE I+J,32: NEXT: NEXT: V
POKE LC+33,74: VPOKE LC+293,74: VPOKE
LC+37,K1
520     VPOKE LC+289,K1: RETURN
530     I=8*INT((LC-6014)/32): J=8*(L
C-6014-32*I): PUT SPRITE 1,(J+4,I+4):
FOR I=LC+130 TO LC+194 STEP 32: FOR
J=0 TO 2: VPOKE I+J,32: NEXT: NEXT: V
POKE LC+33,81: VPOKE LC+293,81: VPOKE
LC+37,K1
540     VPOKE LC+289,K1 : RETURN
550     I=8*INT((LC-6014)/32): J=8*(L
C-6014-32*I): PUT SPRITE 2,(J+4,I+4):
FOR I=LC+130 TO LC+194 STEP 32: FOR
J=0 TO 2: VPOKE I+J,32: NEXT: NEXT: V
POKE LC+33,75: VPOKE LC+293,75: VPOKE
LC+37,K1
560     VPOKE LC+289,K1: RETURN
570     RETURN

```

Mano del MSX

```

580     CO=CO+1
590     D=BC(SC): SC=SC+1: PO=PO+MC(D
,1): CH(CO)=D
600     IF((ST) AND 1)<>1 THEN 620 EL
SE IF CO<7 THEN LC=6179+3*CO ELSE LC=

```



```
6278+3*(CO-7)
610     GOSUB 360: FOR I=1 TO 15: PRIN
T: NEXT: PRINT TAB(15)"Puntos:";PO; C
HR$(11);
620     RETURN
```

Apuesta

```
630     GOSUB 900: FOR I=1 TO 18: PRI
NT: NEXT: IF AB<>0 THEN 640 ELSE 660
640     INPUT"Cambias tu apuesta (S/N
)";X$: IF X$="n" OR X$="N" THEN 670 E
LSE INPUT"Cuanto mas";X$: IF VAL(X$)>
1000 THEN PRINT CHR$(11);: GOTO 630 E
LSE IF VAL(X$)<0 THEN PRINT CHR$(11);
: GOTO 630 ELSE EL=1
650     IF EL=1 THEN AB=AB+VAL(X$): C
=C-VAL(X$): GOTO 670
660     INPUT"Apuesta inicial";X$: IF
VAL(X$)>1000 THEN PRINT CHR$(11);: G
OTO 630 ELSE AB=VAL(X$): C=C-VAL(X$):
GOTO 670
670     IF((ST) AND 1)=0 THEN INPUT"D
tra carta (S/N)";X$: IF X$<>"S" AND X
$<>"s" THEN ST=(ST)OR 1
680     PRINT CHR$(11);: GOSUB 900: G
OSUB 880: RETURN
```

Barajar cartas

```
690     FOR I=1 TO 100: J=INT(RND(1)*
52+1): K=INT(RND(1)*52+1): SWAP BC(J
),BC(K)
700     NEXT
710     RETURN
```

Descubrir las cartas

```
720      GOSUB 900: FOR I=1 TO 18: PRI
NT: NEXT: PRINT"El_MSX_tiene";PO;"pun
tos": PRINT: K4=31-PO: K5=31-PJ: IF K
4=K5 THEN PRINT"-Estais_empatados!":
PRINT"-Otra_mano!": C=C+AB: GOSUB 690
: CLS: SC=1: GOTO 770
730      FOR T=1 TO 2000: NEXT: IF K4>
K5 THEN 780 ELSE 740
```

Pierde el jugador

```
740      FLAY"m59000s816n2r6n3r6n2": G
OSUB 900: PRINT CHR$(11);: FOR I=1 TO
17: PRINT: NEXT: PRINT"Has_perdido_e
sta_mano": PRINT" Tus_puntos_son";PJ;
", los del MSX:":PO: INPUT"Otra mano
(S/N)";X$
750      IF X$="S" OR X$="s" THEN IF S
C<25 THEN PRINT"Bien_la_misma_baraja"
: FOR T=1 TO 2000: NEXT: CLS: GOTO 77
0 ELSE PRINT"Bien_otra_baraja";: FOR
T=1 TO 2000: NEXT: CLS: SC=1: GOTO 77
0
760      GOTO 820
770      AB=0: PJ=0: ST=0: PO=0: CO=0:
CJ=0: FOR I=0 TO 2: PUT SPRITE I,(10
0,200): NEXT: GOTO 150
```

Gana el jugador

```
780      FLAY"11s14m3000n44","11s14m20
00n27": GOSUB 900: PRINT CHR$(11);: F
OR I=1 TO 17: PRINT: NEXT: PRINT"Has_
ganado_esta_mano": PRINT" Tus_puntos_s
```



```
on";PJ; ", los del MSX:";PO: INPUT"O
tra mano (S/N)";X$
790 IF X$="S" OR X$="s" THEN IF S
C<25 THEN PRINT"Bien_la_misma_baraja"
: FOR T=1 TO 2000: NEXT: CLS: GOTO 81
0 ELSE PRINT"Bien_otra_baraja";: FOR
T=1 TO 2000: NEXT: CLS: SC=1: GOTO 81
0
800 C=C+2*AB: GOTO 820
810 C=C+2*AB:AB=0: PJ=0: ST=0: PO
=0: CD=0: CJ=0: FOR I=0 TO 2: PUT SPR
ITE I,(100,200): NEXT: GOTO 150
```

Fin de juego

```
820 SCREEN 1: IF C<=0 THEN PRINT"
Debes_";:PRINT USING"#####.##";-C;:PR
INT"_Pesetas": PRINT: PRINT"Los_cobra
dores_te_visitaran": PRINT"dentro_de_
unos_dias.": END
830 PRINT"Has_ganado_";: PRINT US
ING "#####.##";C;:PRINT"_Pesetas": PR
INT: PRINT"--Apuesto_a_que_no_lo_hace
s": PRINT"dos_veces!!": END
```

Imprimir mensaje

```
840 FOR I=1 TO 18: PRINT: NEXT: PR
INT M1$: PRINT M2$;CHR$(11);
850 RETURN
```

Puntuación

```
860 FOR I=1 TO 15: PRINT: NEXT: PR
INT TAB(15)"Puntos:";PJ;CHR$(11);
870 RETURN
```

Imprimir cuenta

```
880   FOR I=1 TO 23: PRINT: NEXT: PR  
INT TAB(4)"Tu_cuenta_es";: PRINT USIN  
G "#####.##";C;: PRINT CHR$(11);  
890   RETURN
```

Area de mensaje en blanco

```
900   FOR I=6722 TO 6881: VPOKE I,32  
: NEXT  
910   RETURN
```

Datos de sprites

```
920   DATA 255,127,42,31,19,29,9,16,3  
2,56,16,28,16,16,17,14,255,254,172,24  
8,248,248,188,136,116,92,84,108,106,2  
18,151,7  
930   DATA 0,5,11,15,31,62,60,121,120  
,120,120,124,124,124,62,31,128,80,232  
,248,28,12,14,39,7,7,199,15,207,15,30  
,252  
940   DATA 4,10,15,5,7,5,10,10,10,10,  
21,21,26,54,73,48,68,170,254,84,252,7  
2,136,152,132,130,28,8,24,8,8,240
```

Datos de sprites pequeños

```
950   DATA 0,0,0,0,3,7,15,15  
960   DATA 0,0,0,0,255,255,255,255  
970   DATA 0,0,0,0,192,224,240,240
```



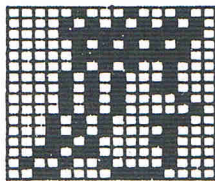
980	DATA	240, 240, 240, 240, 240, 240, 240, 240
990	DATA	240, 240, 224, 192, 0, 0, 0, 0
1000	DATA	255, 255, 255, 255, 0, 0, 0, 0
1010	DATA	15, 15, 7, 3, 0, 0, 0, 0
1020	DATA	15, 15, 15, 15, 15, 15, 15, 15
1030	DATA	255, 255, 255, 255, 255, 255, 255, 255

## Tabla de claves

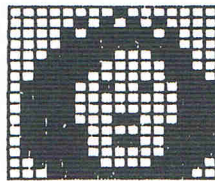
10	=	1185	290	=	1007	570	=	142
20	=	3659	300	=	331	580	=	790
30	=	3660	310	=	1527	590	=	3469
40	=	3661	320	=	2289	600	=	5459
50	=	1807	330	=	3655	610	=	3533
60	=	6672	340	=	260	620	=	142
70	=	1767	350	=	142	630	=	3018
80	=	2478	360	=	9015	640	=	11581
90	=	3356	370	=	4203	650	=	3759
100	=	6082	380	=	5189	660	=	7284
110	=	2016	390	=	4113	670	=	5990
120	=	1031	400	=	2770	680	=	1592
130	=	6326	410	=	3484	690	=	5434
140	=	886	420	=	4472	700	=	131
150	=	1160	430	=	5218	710	=	142
160	=	7959	440	=	6206	720	=	12079
170	=	761	450	=	7436	730	=	2520
180	=	1905	460	=	8278	740	=	11280
190	=	3188	470	=	9170	750	=	10158
200	=	7610	480	=	10158	760	=	206
210	=	1892	490	=	9204	770	=	5095
220	=	4528	500	=	3858	780	=	11536
230	=	4037	510	=	11528	790	=	10238
240	=	1236	520	=	1012	800	=	1271
250	=	4134	530	=	11543	810	=	6160
260	=	1307	540	=	1012	820	=	9674
270	=	6638	550	=	11532	830	=	7758
280	=	3583	560	=	1012	840	=	2463

850	= 142	920	= 5495	990	= 1240
860	= 3210	930	= 5213	1000	= 1256
870	= 142	940	= 4830	1010	= 942
880	= 4623	950	= 942	1020	= 1256
890	= 142	960	= 1256	1030	= 1688
900	= 1635	970	= 1240		
910	= 142	980	= 1640	TOTAL	= 414086

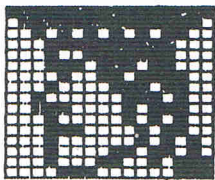
## Diseño de los sprites



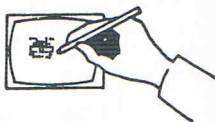
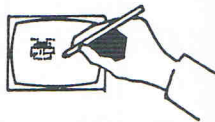
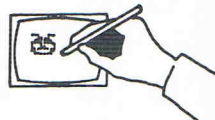
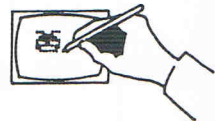
REY



REINA.



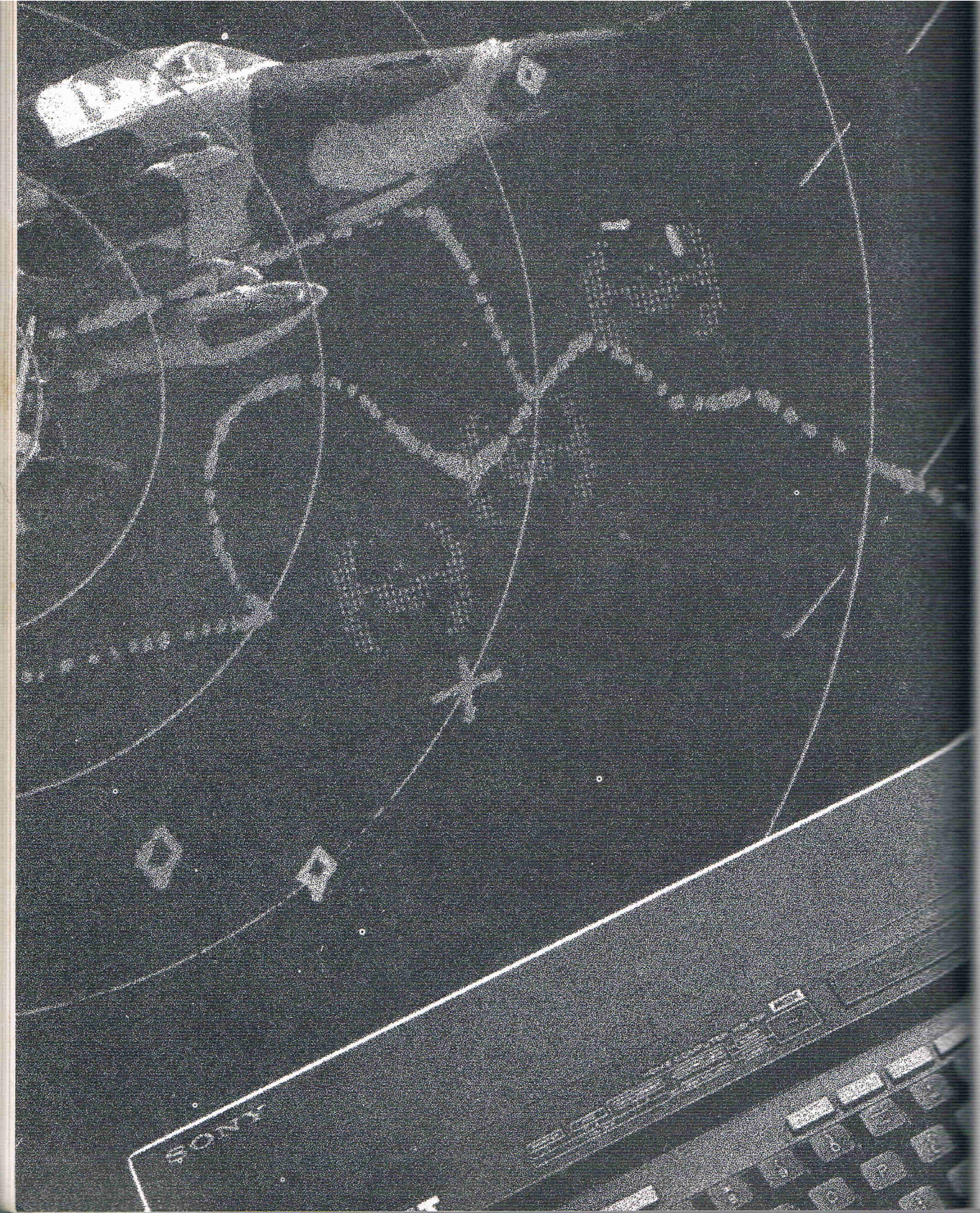
JOTA







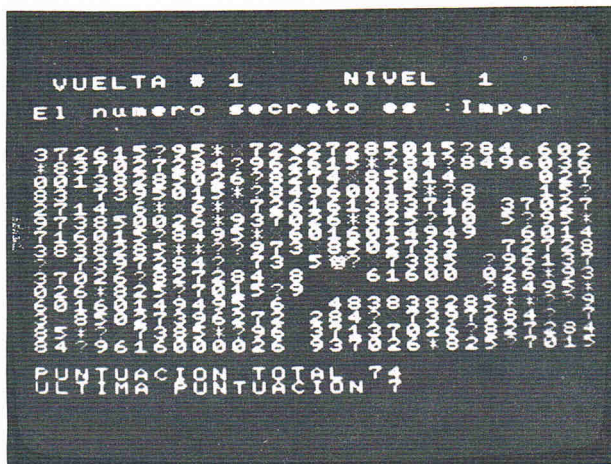






# Minotauro Mastermind

# 27



Tipo de juego:

EDUCATIVO

Este es un juego estratégico que prueba la capacidad mental, incluso de los más experimentados jugadores. El reto es encontrar un número entre 0 y 1000, abriéndote paso a través de un muro de número y símbolos sin ser atrapado por el minotauro.

Tu puntuación es la suma de los dígitos sobre los que te poses; por lo tanto, cuanto más bajos sean los puntos, mejor. Cuando consigas salir por la parte superior del bloque, tienes una oportunidad para averiguar el número.

Los efectos de los símbolos no numéricos son:

- X Te da una pista.
- ? Tiene un valor aleatorio.
- \* Altera tu puntuación aleatoriamente.





```

NIVEL_DE_DIFICULTAD_(1-9)";
70      ND=VAL(INKEY$): IF ND<1 OR ND
>9 THEN 70 ELSE PRINT ND
80      FOR I=1088 TO 1344 STEP 64: F
OR J=0 TO 7: READ Q: VPOKE I+J,Q: NEX
T: NEXT
90      VPOKE 8209,49: VPOKE 8210,97:
VPOKE 8211,177: VPOKE 8212,145: VPOK
E 8213,129
100     FOR I=1 TO 8: PRINT: NEXT: P
RINT "PULSA_UNA_TECLA_PARA_EMPEZAR"
110     IF INKEY$="" THEN 110
120     INTERVAL OFF: IF TP>1000 THE
N TP=TP/3: GOTO 120
130     T=0: NP=0: FOR I=1 TO 10: T(
I)=0: NEXT: V=1: NS=INT(TP): N=NS: D1
=INT(NS/100): D2=INT((NS-D1*100)/10):
D3=INT(NS-D1*100-D2*10)

```

Jugada nueva

```

140     CLS: PJ=6767: PM=6575
150     FOR I=6304 TO 6752 STEP 32: F
OR J=1 TO 30: GOSUB 1620: K1=INT(NA*1
3): IF K1<10 THEN VPOKE I+J,K1+48: GO
TO 190CONT
160     IF K1=10 THEN VPOKE I+J,144:
GOTO 190
170     IF K1=11 THEN VPOKE I+J,152:
GOTO 190
180     VPOKE I+J,160
190     NEXT: NEXT
200     VPOKE 6767,48: UP=0: CM=VPEEK
(PM): VPOKE PM,136: VPOKE 6319,168: C
J=48
210     PRINT "_VUELTA_#";V;" NIVE
L ";ND; CHR$(11);
220     GOSUB 1670

```

Editor

```
230 C=0:Q=1
240 X$=INKEY$: C=C+1
250 IF C=7 THEN Q=1-Q: C=0
260 IF Q<>0 THEN 290
270 IF VPEEK(PJ)=255 THEN VPOKE P
J,CJ: Q=1: GOTO 290
280 VPOKE PJ,255: Q=1
290 IF X$="" THEN 240
300 GOSUB 1650
310 IF VPEEK(PJ)=255 THEN VPOKE P
J,CJ
320 IF X$="P" OR X$="p" THEN 420
330 IF X$="O" OR X$="o" THEN 460
340 IF X$="@" OR X$="2" THEN 500
350 IF X$=";" THEN 540
360 IF X$="-" THEN 580
370 IF X$="0" THEN 620
380 IF X$="L" OR X$="1" THEN 660
390 IF X$=":" THEN 700
400 IF X$="T" OR X$="t" THEN 740
410 GOTO 240
```

Comprobar movimientos permitidos

```
420 IF VPEEK(PJ-32)=32 THEN 240
430 UP=VPEEK(PJ)-48
440 VPOKE PJ,32: PJ=PJ-32
450 GOTO 760
460 IF VPEEK(PJ-1)=32 THEN 240
470 UP=VPEEK(PJ)-48
480 VPOKE PJ,32: PJ=PJ-1
490 GOTO 760
500 IF VPEEK(PJ+1)=32 THEN 240
510 UP=VPEEK(PJ)-48
520 VPOKE PJ,32: PJ=PJ+1
530 GOTO 760
```



```

540 IF VPEEK(PJ+32)=32 THEN 240
550 UP=VPEEK(PJ)-48
560 VPOKE PJ,32: PJ=PJ+32
570 GOTO 760
580 IF VPEEK(PJ-31)=32 THEN 240
590 UP=VPEEK(PJ)-48
600 VPOKE PJ,32: PJ=PJ-31
610 GOTO 760
620 IF VPEEK(PJ-33)=32 THEN 240
630 UP=VPEEK(PJ)-48
640 VPOKE PJ,32: PJ=PJ-33
650 GOTO 760
660 IF VPEEK(PJ+31)=32 THEN 240
670 UP=VPEEK(PJ)-48
680 VPOKE PJ,32: PJ=PJ+31
690 GOTO 760
700 IF VPEEK(PJ+33)=32 THEN 240
710 UP=VPEEK(PJ)-48
720 VPOKE PJ,32: PJ=PJ+33
730 GOTO 760
740 JF=INT(PJ/32): JC=PJ-JF*32: G
DSUB 870: IF PJ<>PM THEN 740
750 GOTO 1530

```

Comprobar movimiento

```

760 CJ=VPEEK(PJ): JF=INT(PJ/32):
JC=PJ-JF*32
770 IF (UP>9) OR (UP<0) THEN UP=0
780 K1=VPEEK(PJ): IF K1=152 THEN
GOSUB 1040
790 IF K1=144 THEN GOSUB 1060
800 IF K1=160 THEN GOSUB 1480
810 IF K1=168 THEN 1690
820 K1=VPEEK(PJ)-48: IF K1<0 OR K
1>9 THEN K1=0
830 IF U+K1>=10-ND THEN GOSUB 870
840 T=T+K1: GOSUB 1670
850 IF PJ=PM THEN 1530
860 GOTO 230

```

Movimiento del minotauro

```
870      VPOKE 8198,31: MF=INT(PM/32):  
        MC=PM-MF*32  
880      IF MC<>CJ THEN 910  
890      IF PM>FJ THEN MF=MF-1: GOTO 9  
80  
900      MF=MF+1: GOTO 980  
910      IF MF<>FJ THEN 940  
920      IF MC>CJ THEN MC=MC-1: GOTO 9  
80  
930      MC=MC+1: GOTO 980  
940      IF PM>PJ THEN 970  
950      IF MC<PJ THEN MC=MC+1: GOTO 9  
80  
960      MC=MC-1: GOTO 980  
970      MF=MF-1  
980      K=32*MF+MC: VPOKE 8198,241: IF  
        VPEEK(K)=136 THEN RETURN  
990      VPOKE PM,CB  
1000     PM=K: CB=VPEEK(PM)  
1010     VPOKE PM,136  
1020     PLAY "n2"  
1030     RETURN
```

Jugador sobre (?)

```
1040     GOSUB 1620: I=INT(NA*10): VPO  
KE PJ, I+48  
1050     CJ=I+48: RETURN
```

Jugador sobre (X)

```
1060     VPOKE PJ,48  
1070     IF NP>=9 THEN BEEP: GOTO 117  
0  
1080     GOSUB 1620: K2=INT(NA*9+1):
```



```

IF T(K2)=1 THEN 1080
1090     T(K2)=1: NP=NP+1: BEEP
1100     ON K2 GOTO 1110,1130,1170,12
10,1250,1280,1330,1350,1390
1110     PRINT: PRINT: PRINT"Suma_de_
los_digitos_del":PRINT"numero_secreto
_es_";
1120     PRINT D1+D2+D3: PRINT CHR$(1
1);: RETURN
1130     PRINT: PRINT: PRINT"E1_numer
o_secreto_es_";
1140     IF NS/2=INT(NS/2) THEN PRINT
"Par": GOTO 1160
1150     PRINT"Impar"
1160     PRINT CHR$(11);: RETURN
1170     GOSUB 1620:K=INT(NA*200)+NS:
J=K-200: IF J<0 THEN J=0
1180     IF K>1000 THEN K=1000
1190     PRINT: PRINT: PRINT"E1_numer
o_esta_entre";J: PRINT"y";K
1200     PRINT CHR$(11);: RETURN
1210     PRINT: PRINT: PRINT"E1_numer
o_";
1220     IF NS/5=INT(NS/5) THEN PRINT
"es_divisible":GOTO 1240
1230     PRINT"no_es_divisible"
1240     PRINT"por_5": PRINT CHR$(11)
;: RETURN
1250     L=D1: IF D2>L THEN L=D2
1260     IF D3>L THEN D3=L
1270     PRINT: PRINT: PRINT"E1_mayor
_digito_del_numero": PRINT"secreto_es
_";L: CHR$(11);: RETURN
1280     PRINT: PRINT: PRINT"E1_prime
r_digito_del_numero": PRINT"secreto_e
s";
1290     IF D1>5 THEN PRINT"Mayor_que
_5": GOTO 1320
1300     IF D1<5 THEN PRINT"Menor_que
_5": GOTO 1320
1310     PRINT"Igual_a_5"

```

```

1320 PRINT CHR$(11);: RETURN
1330 PRINT: PRINT: PRINT"La_suma_
de_los_digitos": PRINT"primero_y_terc
ero_es";
1340 PRINT D1+D3; CHR$(11);: RETU
RN
1350 PRINT: PRINT: PRINT"Uno_de_l
os_digitos_del": PRINT"numero_secreto
_s_es";
1360 GOSUB 1620: K=INT(NA*3): IF
K=0 AND NS>99 THEN PRINT D2: GOTO 138
0
1370 PRINT D3
1380 PRINT CHR$(11);: RETURN
1390 PRINT: PRINT: PRINT"El_produ
cto_de_los_digitos": PRINT"que_no_son
_cero_es";
1400 IF D1=0 AND D2=0 THEN PRINT
D3: GOTO 1470
1410 IF D1=0 AND D3=0 THEN PRINT
D2: GOTO 1470
1420 IF D1=0 THEN PRINT D2*D3: GO
TO 1470
1430 IF D2=0 AND D3=0 THEN PRINT
D1: GOTO 1470
1440 IF D2=0 THEN PRINT D1*D3: GO
TO 1470
1450 IF D3=0 THEN PRINT D1*D2: GO
TO 1470
1460 PRINT D1*D2*D3
1470 PRINT CHR$(11);: RETURN

```

Jugador sobre (\*)

```

1480 VPOKE PJ,48
1490 IF T<11 THEN RETURN
1500 GOSUB 1620: I=INT(NA*100): I
F I>50 THEN T=T-10: PLAY"L64N10N8N6N4
": RETURN

```



```
1510 IF I<10 THEN T=T+50: PLAY"L3
2N10N8N6N4N10N8N6N4": RETURN
1520 T=T+10: PLAY"L64N10N8N6N4":
RETURN
```

Fin de juego

```
1530 CLS: IF PM=PJ THEN 1570
1540 FOR I=1 TO 8: PRINTTAB(2*I)"
ENHORABUENA": PRINT: NEXT: PRINT
1550 PRINT"HAS_VENCIDO_AL_MINOTAU
RO": PRINT"TU_PUNTUACION_ES_";T
1560 PRINT"EN";V;" VUELTAS": GOTO
1590
1570 FOR I=1 TO 5: PRINT TAB(2*I)
"VENCIO_EL_MINOTAURO": PRINT: NEXT: P
RINT
1580 PRINT"El_numero_secreto_era_
";NS
1590 PRINT: PRINT"PULSA_UNA_TECLA
_PARA": PRINT"JUGAR_OTRA_VEZ"
1600 X$=INKEY$: IF X$="" THEN 160
0
1610 RUN 20
```

Número aleatorio

```
1620 NA=(9999*NA+N) MOD 5997!: NA
=NA/5997!
1630 IF NA<.2 THEN N=NA*10000+1
1640 RETURN
```

Mensaje en blanco

```
1650 FOR I=6208 TO 6271: VPOKE I,
32: NEXT
1660 RETURN
```

Puntuación

```
1670   FOR I=1 TO 21: PRINT: NEXT: P
RINT"PUNTUACION_TOTAL";T: PRINT"ULTIM
A_PUNTUACION";UP;CHR$(11);
1680   RETURN
```

Adivinar número

```
1690   PRINT: PRINT: PRINT"ADIVINA_E
L_NUMERO_SECRETO"
1700   INPUT"Y PULSA (RETURN)";K$
1710   I=VAL(K$): IF I=NS THEN 1530
1720   G1=INT(I/100): G2=INT((I-G1*1
00)/10): G3=I-G1*100-G2*10
1730   CLS: PRINT: IF D1=G1 THEN PRI
NT"El_primer_digito_es_correcto":PRIN
T
1740   IF D2=G2 THEN PRINT"El_segund
o_digito_es_correcto":PRINT
1750   IF D3=G3 THEN PRINT"El_tercer
_digito_es_correcto"
1760   IF D1<>G1 AND D2<>G2 AND D3<>
G3 THEN PRINT"Ningun_digito_es correc
to"
1770   FOR Z=1 TO 2000: NEXT
1780   V=V+1: GOTO 140
```

Datos de sprites pequeños

```
1790   DATA 165,126,219,126,102,126,8
2,60
1800   DATA 136,136,80,32,80,136,136,
0
1810   DATA 112,136,8,16,32,0,32,0
```



```

1820 DATA 32,168,112,32,112,168,32,
0
1830 DATA 16,56,124,254,124,56,16,0

```

Instrucciones

```

1840 CLS:INPUT"QUIERES INSTRUCCIONE
S (S/N)?";X$: IF X$="N" OR X$="n" THE
N 1910
1850 CLS: PRINT"PARA_MOVERTE_EN_EL_
LABERINTO_PULSA_:"
1860 PRINT:PRINT"__F-p__ARRIBA": P
RINT"__D-o__IZQUIERDA": PRINT"__@-2_
__DERECHA": PRINT"__;____ABAJO": PRI
NT"__-____ARRIBA-DERECHA:
1870 PRINT"__@____ARIBA-IZQUIERDA"
: PRINT"__L-1__ABAJO-IZQUIERDA": PRI
NT"__:_____ABAJO-DERECHA": PRINT"__T-
t__ATRAPADO"
1880 PRINT:PRINT:PRINT"SI_PISAS_:"
PRINT: PRINT"__X____TENDRAS_UNA_PISTA
": PRINT"__?____SUMARAS_PUNTUACION____
____ALEATORIA": PRINT"__*____ALTERA
RA_TU_PUNTUACION____ALEATORIAMENTE
"
1890 PRINT: PRINT"PULSA_UNA_TECLA_P
ARA_SEGUIR"
1900 IF INKEY$="" THEN 1900
1910 RETURN

```

## Tabla de claves

10	= 210	70	= 3175	130	= 10247
20	= 3178	80	= 3331	140	= 1457
30	= 201	90	= 2349	150	= 6058
40	= 1029	100	= 3295	160	= 1933
50	= 3251	110	= 1024	170	= 1942
60	= 2489	120	= 3417	180	= 805

190	= 320	600	= 1304	1010	= 550
200	= 3854	610	= 401	1020	= 421
210	= 2426	620	= 1797	1030	= 142
220	= 295	630	= 1351	1040	= 2326
230	= 719	640	= 1306	1050	= 957
240	= 1289	650	= 401	1060	= 459
250	= 1729	660	= 1794	1070	= 1569
260	= 982	670	= 1351	1080	= 2843
270	= 2685	680	= 1303	1090	= 1669
280	= 1062	690	= 401	1100	= 1909
290	= 1042	700	= 1796	1110	= 4871
300	= 275	710	= 1351	1120	= 1955
310	= 2045	720	= 1305	1130	= 2676
320	= 1837	730	= 401	1140	= 2766
330	= 1875	740	= 4267	1150	= 718
340	= 1839	750	= 406	1160	= 916
350	= 891	760	= 3581	1170	= 4068
360	= 917	770	= 2038	1180	= 1510
370	= 960	780	= 2125	1190	= 3184
380	= 1814	790	= 1074	1200	= 916
390	= 1050	800	= 1255	1210	= 1581
400	= 1910	810	= 1077	1220	= 3764
410	= 391	820	= 3121	1230	= 1669
420	= 1764	830	= 1957	1240	= 1609
430	= 1351	840	= 1125	1250	= 1712
440	= 1273	850	= 1176	1260	= 1224
450	= 401	860	= 381	1270	= 5360
460	= 1767	870	= 2896	1280	= 4525
470	= 1351	880	= 1279	1290	= 2169
480	= 1276	890	= 2113	1300	= 2164
490	= 401	900	= 1216	1310	= 925
500	= 1766	910	= 1315	1320	= 916
510	= 1351	920	= 2091	1330	= 4933
520	= 1275	930	= 1210	1340	= 1452
530	= 401	940	= 1125	1350	= 4796
540	= 1763	950	= 2105	1360	= 3574
550	= 1351	960	= 1211	1370	= 264
560	= 1272	970	= 793	1380	= 916
570	= 401	980	= 3204	1390	= 5081
580	= 1795	990	= 532	1400	= 2018
590	= 1351	1000	= 1546	1410	= 2018



1420	=	1759	1600	=	1529	1780	=	1019
1430	=	2018	1610	=	172	1790	=	1566
1440	=	1759	1620	=	3002	1800	=	1413
1450	=	1759	1630	=	1878	1810	=	1199
1460	=	985	1640	=	142	1820	=	1405
1470	=	916	1650	=	1531	1830	=	1365
1480	=	459	1660	=	142	1840	=	4411
1490	=	849	1670	=	5104	1850	=	2871
1500	=	4322	1680	=	142	1860	=	6117
1510	=	3246	1690	=	2371	1870	=	6010
1520	=	1945	1700	=	1474	1880	=	9386
1530	=	1178	1710	=	2064	1890	=	2277
1540	=	2971	1720	=	5133	1900	=	1029
1550	=	3497	1730	=	4350	1910	=	142
1560	=	1481	1740	=	4034			
1570	=	3523	1750	=	3721			
1580	=	2469	1760	=	5653	TOTAL	=	377447
1590	=	3049	1770	=	1126			

1870	1871	1872	1873	1874	1875
1876	1877	1878	1879	1880	1881
1882	1883	1884	1885	1886	1887
1888	1889	1890	1891	1892	1893
1894	1895	1896	1897	1898	1899
1900	1901	1902	1903	1904	1905
1906	1907	1908	1909	1910	1911
1912	1913	1914	1915	1916	1917
1918	1919	1920	1921	1922	1923
1924	1925	1926	1927	1928	1929
1930	1931	1932	1933	1934	1935
1936	1937	1938	1939	1940	1941
1942	1943	1944	1945	1946	1947
1948	1949	1950	1951	1952	1953
1954	1955	1956	1957	1958	1959
1960	1961	1962	1963	1964	1965
1966	1967	1968	1969	1970	1971
1972	1973	1974	1975	1976	1977
1978	1979	1980	1981	1982	1983
1984	1985	1986	1987	1988	1989
1990	1991	1992	1993	1994	1995
1996	1997	1998	1999	2000	2001
2002	2003	2004	2005	2006	2007
2008	2009	2010	2011	2012	2013
2014	2015	2016	2017	2018	2019
2020	2021	2022	2023	2024	2025



# Apéndice A

## Uso del *Joystick*

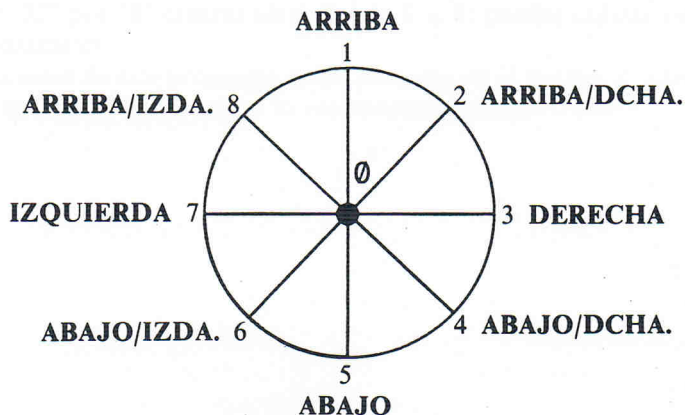
Para usar el "joystick" en juegos que empleen el teclado necesitarás reemplazar las llamadas de `USR` a la posición 60000 (que lee el teclado) por una subrutina en `BASIC` como la siguiente. (Utiliza `STICK(1)` para leer del mando 1 o `STICK(2)` para el mando 2, o `STICK(0)` para leer del teclado.)

```
A = STICK(1)
IF (A = 0) THEN RETURN (al programa principal)
IF (A = 1) OR (A = 2) OR (A = 8) THEN (mover arriba)
IF (A = 8) OR (A = 7) OR (A = 6) THEN (mover a la izquierda)
IF (A = 6) OR (A = 5) OR (A = 4) THEN (mover abajo)
IF (A = 2) OR (A = 3) OR (A = 4) THEN (mover a la derecha)
```

Estas sentencias de "mover" podrían ser llamadas `USR` a la posición 60118, acompañadas de los `POKEs` adecuados (ver apéndice C, "Programa soporte").

Para alterar el fuego, modifica, simplemente, cualquier referencia a `STRIG(0)` por `STRIG(1)` para el mando 1, o `STRIG(2)` para el mando 2.

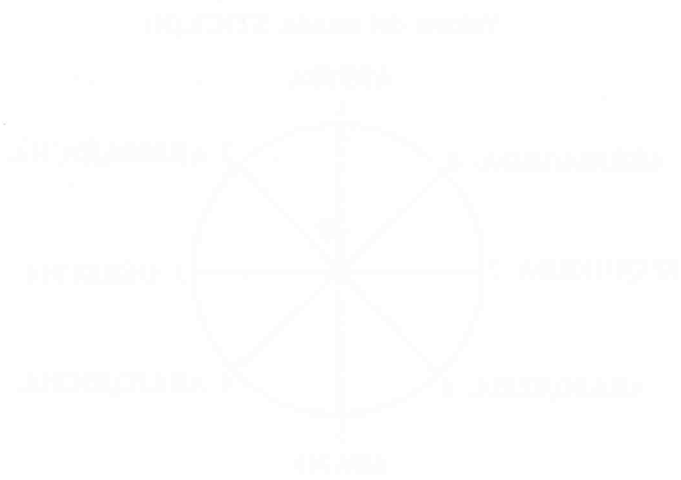
Valores del mando, `STICK(N)`



# Uso del Joystick

Este es el "botón" de juego que se utiliza al jugar con el joystick. El botón se utiliza para mover el cursor del mouse y para hacer clic. El botón se utiliza para mover el cursor del mouse y para hacer clic. El botón se utiliza para mover el cursor del mouse y para hacer clic.

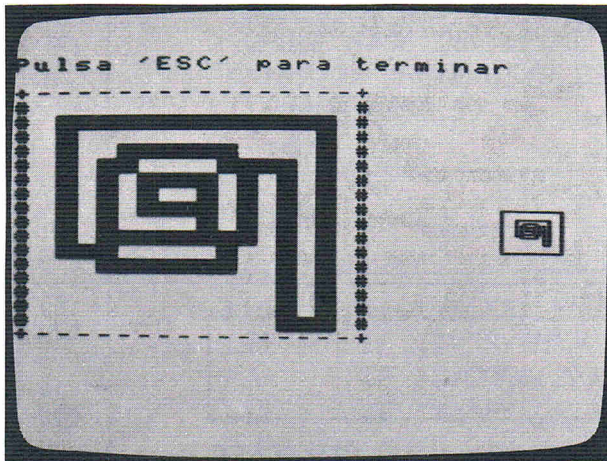
El joystick se utiliza para mover el cursor del mouse y para hacer clic. El joystick se utiliza para mover el cursor del mouse y para hacer clic. El joystick se utiliza para mover el cursor del mouse y para hacer clic. El joystick se utiliza para mover el cursor del mouse y para hacer clic. El joystick se utiliza para mover el cursor del mouse y para hacer clic.





## Apéndice B

# Constructor de sprites



Este programa facilita la creación de sprites moviendo el cursor y definiendo puntos en los lugares deseados. Cuando has terminado, el programa te facilita los números que necesitas para crear el sprite en tu programa. Si vas a poner estos números en una sentencia DATA, con las siguientes instrucciones crearás un sprite de  $16 \times 16$ :

```
10 A$=" ": FOR I=1 TO 32: READ Q: A$=A$+CHR$(Q): NEXT I:  
   SPRITE$(0)=A$
```

Si cambias "32" por "8" crearás un sprite de  $8 \times 8$ ; puedes utilizar este tipo de sprites para redefinir caracteres.

Las instrucciones de este programa están incluidas en él mismo, y, además, a la derecha de la matriz te aparece el sprite según lo vas construyendo.

# SUGERENCIAS AL PROGRAMA

Puedes establecer el color de sprite añadiendo el número de color al final de la sentencia "PUT SPRITE" en las líneas 90 y 380.

Si quieres ver el sprite en modo ampliado, puedes hacer que el programa dé a elegir entre los modos, y modificar las cajas de alrededor de los sprites adecuadamente.

## Variables del programa

MA(I, J) Matriz del sprite (1 a 0).  
PC Posición del cursor.  
CB Carácter bajo el cursor.  
FI Fila.  
CO Columna.

### Inicialización

```
10 KEY OFF: COLOR 1,15,15: SCREE
N 1,2: PRINT"__CONSTRUCTOR_DE_SPRITES
"
20 PRINT: PRINT: PRINT: INPUT"EL
IJE_8*8_('8')_D_____16*16_(
'16')";X$
30 IF X$="16" THEN 340 ELSE IF X
$="8" THEN 40 ELSE PRINT"REPITE": GOT
O 20
```

### Preparar pantalla (sprite 8 x 8)

```
40 CLS:PRINT: PRINT: PRINT"__Pulsa
_'ESC'_para_finalizar."
50 LOCATE 0,21:PRINT"_'P'_pinta,_'B'_b
orra":LOCATE 0,2
60 PRINT: PRINT: PRINT"__+-----
--+":FOR I= 1 TO 8 : PRINT"_____
__": NEXT: PRINT"__+-----+""
70 PRINT: PRINT: PRINT"__Utiliza
```



```

las_teclas_del_cursor_para_mover"
: PRINT: PRINT" _Pulsa 'P' para fijar
un punto.": PRINT : PRINT" _Pul
sa 'B' para borrar. "
80     PC=6473-64: CB=32: FI=4: CO=4
90     A$="": SPRITE$(0)=A$: PUT SPRI
TE 0, (167,86)
100    VPOKE 6484,24: VPOKE 6485,23:
      VPOKE 6486,25: VPOKE 6516,22: VPOKE
6518,22: VPOKE 6549,23: VPOKE 6548,26
: VPOKE 6550,27

```

Control (sprite 8 x 8)

```

110    X$=INKEY$: VPOKE PC,255
120    IF X$="" THEN 110
130    IF X$=CHR$(27) THEN 290
140    IF X$="P" OR X$="p" THEN VPOK
E PC,CB: GOTO 210
150    IF X$="B" OR X$="b" THEN VPOK
E PC,CB: GOTO 220
160    IF X$=CHR$(30) THEN 230
170    IF X$=CHR$(29) THEN 240
180    IF X$=CHR$(31) THEN 250
190    IF X$=CHR$(28) THEN 260
200    GOTO 110

```

Mover/dibujar/borrar

```

210    IF VPEEK(PC)=255 THEN 110 ELS
E CB=255: MA(FI,CO)=1: GOTO 270
220    IF VPEEK(PC)=32 THEN 110 ELSE
CB=32: MA(FI,CO)=0: GOTO 270
230    IF FI<2 THEN 110 ELSE FI=FI-1
: VPOKE PC,CB: CB=VPEEK(PC-32): PC=P
C-32: GOTO 110
240    IF CO<2 THEN 110 ELSE CO=CO-1
: VPOKE PC,CB: CB=VPEEK(PC-1): PC=P
C

```

```
-1: GOTO 110
250     IF FI>7 THEN 110 ELSE FI=FI+1
: VPOKE PC,CB: CB=VPEEK(PC+32): PC=PC+32: GOTO 110
260     IF CO>7 THEN 110 ELSE CO=CO+1
: VPOKE PC,CB: CB=VPEEK(PC+1): PC=PC+1: GOTO 110
```

Visualizar sprite 8 x 8

```
270     K2=14335+FI:K=VPEEK(K2): K1=2
^(8-CO): IF MA(FI,CO)=1 THEN K=K+K1:
VPOKE K2,K: GOTO 110
280     K=K-K1: VPOKE K2,K: GOTO 110
```

Fin (sprite 8 x 8)

```
290     CLS: PRINT"Numeros_para_las_s
entencias,_DATA_": PRINT: PRINT: PRIN
T
300     FOR I=14336 TO 14342: PRINT U
SING"###"; VPEEK(I);: PRINT"_": NEXT
310     PRINT: PRINT: PRINT: PRINT US
ING"###"; VPEEK(14343)
320     FOR I=1 TO 4: PRINT: NEXT: PR
INT"Pulsa_una_tecla_para_empezar_de_n
uevo": PUT SPRITE 0,(127,150)
330     IF INKEY$="" THEN 330 ELSE RU
N
```

Preparar pantalla (sprite 16 x 16)

```
340     CLS: PRINT"Pulsa_'ESC'_para_t
erminar": PRINT
350 LOCATE 0,21:PRINT"'P'_pinta,'B'_
borra":LOCATE 0,2
360     PRINT"+-----+": FO
```



```

R I=1 TO 16: PRINT"#.....#
":NEXT:PRINT"+-----+"
370 DIM MA(16,16): PC=6474: CB=32
: FI=8: CO=8
380 A$="": SPRITE$(0)=A$: PUT SPR
ITE 0,(216,86)
390 VPOKE 6490,24: VPOKE6491,23:
VPOKE 6492,23
400 VPOKE 6493,25: VPOKE 6522,22:
VPOKE 6525,22
410 VPOKE 6554,22: VPOKE 6557,22:
VPOKE 6587,23
420 VPOKE 6588,23: VPOKE 6586,26:
VPOKE 6589,27

```

Control (sprite 16 x 16)

```

430 X$=INKEY$: VPOKE PC,255
440 IF X$="" THEN 430
450 IF X$=CHR$(27) THEN 630
460 IF X$="P" OR X$="p" THEN VPOK
E PC,CB: GOTO 530
470 IF X$="B" OR X$="b" THEN VPOK
E PC,CB: GOTO 540
480 IF X$=CHR$(30) THEN 550
490 IF X$=CHR$(29) THEN 560
500 IF X$=CHR$(31) THEN 570
510 IF X$=CHR$(28) THEN 580
520 GOTO 430

```

Mover/dibujar/borrar

```

530 IF VPEEK(PC)=255 THEN 430 ELS
E CB= 255: MA(FI,CO)=1 : GOTO 590
540 IF VPEEK(PC)=32 THEN 430 ELSE
CB=32: MA(FI,CO)=0: GOTO 590
550 IF FI<2 THEN 430 ELSE FI=FI-1
: VPOKE PC,CB: CB=VPEEK(PC-32): PC=PC
-32: GOTO 430

```

```

560     IF CD<2 THEN 430 ELSE CD=CD-1
      : VPOKE PC,CB: CB=VPEEK(PC-1): PC=PC-
      1: GOTO 430
570     IF FI>15 THEN 430 ELSE FI=FI+
      1: VPOKE PC,CB: CB=VPEEK(PC+32): PC=P
      C+32: GOTO 430
580     IF CD>15 THEN 430 ELSE CD=CD+
      1: VPOKE PC,CB: CB=VPEEK(PC+1): PC=PC
      +1: GOTO 430

```

Visualizar sprite 16 x 16

```

590     IF CD<9 THEN K1=0 ELSE K1=16
600     K2=14335+FI+K1: K=VPEEK(K2):
      IF CD<9 THEN K1=2^(8-CD) ELSE K1=2^(1
      6-CD)
610     IF MA(FI,CD)=1 THEN K=K+K1: V
      POKE K2,K: GOTO 430
620     K=K-K1:VPOKE K2,K: GOTO 430

```

Fin (sprite 16 x 16)

```

630     CLS: PRINT"Numeros_para_las_s
      entencias,_DATA_": PRINT"_Leer_de_izq
      uierda_a_derecha": PRINT
640     FOR I=14335 TO 14356 STEP 7:
      PRINT: PRINT: FOR J=1 TO 7 : PRINT US
      ING"###": VPEEK(I+J): PRINT "_": NE
      XT: NEXT
650     PRINT: PRINT: FOR I=14364 TO
      14367: PRINT USING"###": VPEEK(I): P
      RINT"_": NEXT
660     FOR I=1 TO 3: PRINT: NEXT: PR
      INT"_Pulsa_una_tecla_para_empezar_de_
      nuevo": PUT SPRITE 0,(120,165)
670     IF INKEY$="" THEN 670ELSE RUN

```



## Tabla de claves

10	= 3148	240	= 4971	470	= 2391
20	= 2920	250	= 4956	480	= 1305
30	= 3225	260	= 4971	490	= 1314
40	= 3204	270	= 6051	500	= 1326
50	= 2454	280	= 1574	510	= 1333
60	= 3794	290	= 4067	520	= 326
70	= 10374	300	= 2667	530	= 3711
80	= 2203	310	= 1793	540	= 3200
90	= 2044	320	= 5778	550	= 5086
100	= 4014	330	= 1346	560	= 5101
110	= 1316	340	= 2829	570	= 5092
120	= 912	350	= 2454	580	= 5107
130	= 1297	360	= 3904	590	= 1762
140	= 2599	370	= 2412	600	= 5730
150	= 2584	380	= 2093	610	= 2808
160	= 1495	390	= 1389	620	= 1639
170	= 1504	400	= 1455	630	= 6472
180	= 1516	410	= 1611	640	= 4849
190	= 1268	420	= 1685	650	= 3126
200	= 261	430	= 1316	660	= 5817
210	= 3581	440	= 977	670	= 1431
220	= 3070	450	= 1382		
230	= 4956	460	= 2409	TOTAL	= 196755





# Apéndice C

## Programa soporte

Este programa (se necesita en la mayoría de los programas de este libro) debe ser cargado y ejecutado antes de que cargues tu programa. Resulta imprescindible para la animación de los juegos.

Pasos a seguir:

1. Mete este programa, utiliza el verificador para depurarlo y después grábalo como "SOPORTE".
2. Cárgalo y ejecútalo cada vez que enciendas la máquina o pulses RESET.
3. A continuación carga y ejecuta cualquier programa del libro; él llamará y usará a éste cuando sea necesario.
4. Si cambias de juego, no es necesario volver a ejecutar el programa soporte.
5. Los datos del programa soporte permanecen en una zona de memoria reservada y no se destruyen, aunque se carguen otros programas, mientras no se apague la máquina o se pulse RESET.

```
10 CLS
20 REM PROGRAMA SOPORTE
30 PRINT "PROGRAMA_SOPORTE"
40 PRINT:PRINT:PRINT
50 PRINT"ZONA_RESERVADA"
60 PRINT"COMIENZO",
70 A=PEEK(64586!)+256*PEEK(64587!)
80 PRINT A
90 CLEAR 255,A-500
100 A=PEEK(64586!)+256*PEEK(64587!):B
=A+10
110 I=B
120 READ A$
130 IF A$="R" THEN READ C:C=C+B: C1 =
INT(C/256):C2=256*(C/256-C1):POKE I,
C2:I=I+1:POKE I,C1 :GOTO 150
140 POKE I, VAL(A$)
150 IF I=B+217THEN PRINT"FINAL...", A
:END ELSE I=I+1:GOTO 120
```

160 DATA 219,170,230,240,33,R,-4,94,2  
 46,8,0,0,0,0,0,0,0,0,0,0,211,170,21  
 9,169,71,0,0,0,0,0,0,0  
 170 DATA 0,0,0,230,128,194,R,55,203,9  
 1,202,R,55,62,3,50,R,-2,195,R,113,120  
 ,230,64,194,R,74,203,83,202,R,74,62,2  
 180 DATA 50,R,-2,195,R,113,120,230,32  
 ,194,R,93,203,67,202,R,93,62,0,50,R,-  
 2,195,R,113,120,230,16,194,R,112,203,  
 75,202,R,112,62,1,50,R,-2,195,R,113,2  
 01,62,0,50,R,-3,58  
 190 DATA R,-3,203,39,203,39,79,6,0,33  
 ,0,27,9,58,R,-1,71,58,R,-2,87,254,0,1  
 94,R,154,205,R,187,121,152,79,195,R,2  
 04,254  
 200 DATA 1,194,R,166,35,205,R,187,195  
 ,R,148,254,2,194,R,180,205,R,187,121,  
 128,79,195,R,204,35,205,R,187,195,R,1  
 74,125,243,211  
 210 DATA 153,124,211,153,0,0,0,0,219,  
 152,79,183,251,201,125,243,211,153,12  
 4,246,64,211,153,121,211,152,251,201

## Tabla de claves

10	= 159	100	= 3265	190	= 5624
20	= 1300	110	= 378	200	= 6113
30	= 1402	120	= 236	210	= 5059
40	= 551	130	= 6422	220	= 0
50	= 1226	140	= 854	230	= 0
60	= 869	150	= 3367		
70	= 2571	160	= 4425		
80	= 210	170	= 5288	TOTAL	= 58450
90	= 1040	180	= 8091		



## Apéndice D

# Gráficos avanzados (I)

### Los *sprites*

El procesador de video del MSX tiene la capacidad de generar *sprites*. Los *sprites* son caracteres o figuras definidas por el usuario, que pueden visualizarse en pantalla sin afectar a la pantalla de fondo, ya que se sitúan en diferentes planos de pantalla. Puedes hacer que se escondan uno detrás de otro y también moverlos. Te brindan la posibilidad de hacer juegos de un modo sencillo sin preocuparte por la pantalla de fondo.

### Tamaño de *sprites*

Puedes emplear cuatro tamaños de *sprites*, pero en un instante determinado sólo podrás utilizar un único tamaño. El tamaño se determina mediante la instrucción SCREEN.

	<u>TAMAÑO</u>	
SCREEN,0	8 × 8	punto normal
SCREEN,1	8 × 8	punto ampliado a 16 × 16
SCREEN,2	16 × 16	punto normal
SCREEN,3	16 × 16	punto ampliado a 32 × 32

La ampliación te da una resolución de 2 × 2 puntos.

## Definición de *sprites*: SPRITES

Puedes definir tus *sprites* empleando la instrucción SPRITES. Puedes tener hasta 256 modelos de *sprites* cuando trabajes con los tamaños de *sprite* 0 ó 1 (8 × 8 puntos), o 64 con el tamaño de *sprite* 2 ó 3 (16 × 16 puntos).

### Sintaxis

SPRITES(<entero>) = <cadena de caracteres>.  
<entero> = número de modelos *sprite*.  
<entero> = de 0 a 255 cuando el tamaño del *sprite* es 0 ó 1.  
<entero> = de 0 a 63 cuando el tamaño del *sprite* es 2 ó 3.

SPRITES es una cadena de caracteres. La longitud de la cadena del *sprite* es de 32 bytes (o caracteres), pero para *sprites* pequeños de 8 × 8 sólo necesitas definir los ocho primeros bytes.

Cada byte de la cadena del *sprite* representa una fila de 8 puntos y se suma a la cadena del *sprite* el carácter con código ASCII igual al del byte. Puedes hacerlo de un modo sencillo mediante la función CHR\$(). Por ejemplo, para formar un *sprite* de 8 × 8:

$$\text{SPRITES}(\langle \text{entero} \rangle) = \text{CHR}\langle 1.^\text{a} \text{ línea} \rangle + \text{CHR}\langle 2.^\text{a} \text{ línea} \rangle + \\ \text{CHR}\langle 3.^\text{a} \text{ línea} \rangle + \text{CHR}\langle 4.^\text{a} \text{ línea} \rangle + \text{CHR}\langle 5.^\text{a} \text{ línea} \rangle + \\ \text{CHR}\langle 6.^\text{a} \text{ línea} \rangle + \text{CHR}\langle 7.^\text{a} \text{ línea} \rangle + \text{CHR}\langle 8.^\text{a} \text{ línea} \rangle.$$

La función CHR\$() se puede reemplazar por el carácter, si sabes cuál es. Para definir un *sprite* de 16 × 16 tienes que definir los 33 bytes que lo componen.

Aquí tienes un método para definir *sprites* de 8 × 8.

Primero dibuja tu *sprite* en un papel.

									<u>BINARIO</u>	<u>DECIMAL</u>
128	64	32	16	8	4	2	1		= &B00010000	= 16
.	.	.	1	.	.	.	.		= &B00110000	= 48
.	.	1	1	.	.	.	.		= &B01110000	= 112
1	1	1	1	1	1	1	1		= &B11111111	= 255
1	1	1	1	1	1	1	1		= &B11111111	= 255
.	1	1	1	.	.	.	.		= &B01110000	= 112
.	.	1	1	.	.	.	.		= &B00110000	= 48
.	.	.	1	.	.	.	.		= &B00010000	= 16

Forma una expresión de caracteres como la siguiente:

$$\text{SPRITES}(0) = \text{CHR}\langle 16 \rangle + \text{CHR}\langle 48 \rangle + \text{CHR}\langle 112 \rangle + \text{CHR}\langle 255 \rangle + \\ \text{CHR}\langle 255 \rangle + \text{CHR}\langle 112 \rangle + \text{CHR}\langle 48 \rangle + \text{CHR}\langle 16 \rangle.$$



Ejemplo:

La manera más sencilla de definir *sprites* es mediante los números binarios; te resultará más fácil.

```
○ 10 SCREEN 2,0
○ 20 FOR I=1 TO 8
○ 30 READ A$
○ 40 S$=S$+CHR$(VAL("&B"+A$))
○ 50 NEXT I
○ 60 SPRITE$(0)=S$
○ 70 PUT SPRITE 0,(100,100),15,0
○ 80 GOTO 80
○ 90 REM Datos en binario
○ 100 DATA 00010000
○ 110 DATA 00110000
○ 120 DATA 01110000
○ 130 DATA 11111111
○ 140 DATA 11111111
○ 150 DATA 01110000
○ 160 DATA 00110000
○ 170 DATA 00010000
```

LINEA 10 PANTALLA DE ALTA RESOLUCION, TAMAÑO NORMAL DE SPRITE.

LINEA 30 LEE EL NUMERO BINARIO COMO CADENA DE CARACTERES.

LINEA 40 S\$ ES UNA CADENA DE CARACTERES TEMPORAL.

LINEA 60 DEFINE EL SPRITE 0.

LINEA 70 PONE EL SPRITE 0 EN X = 100, Y = 100, COLOR BLANCO, Y PLANO DE SPRITE 0.

Resultado: verás una flecha en mitad de la pantalla.

Más adelante veremos con más detalle la instrucción PUT SPRITE.

Si sabes calcular los valores decimales de cada byte, entonces se puede escribir el programa en una versión más corta. El programa de arriba se puede reducir a las siguientes líneas:

```
○ 10 SCREEN 2,0
○ 20 SPRITE$(0)=CHR$(16)+CHR$(48)+CHR$(112)+CHR$(255
○ )+CHR$(255)+CHR$(112)+CHR$(48)+CHR$(16)
○ 70 PUT SPRITE 0,(100,100),15,0
○ 80 GOTO 80
```

LINEA 10 PANTALLA DE ALTA RESOLUCION.

LINEA 20 DEFINICION DEL SPRITE.

LINEA 70 PONE EL SPRITE 0 EN X = 100, Y = 100, COLOR BLANCO, PLANO DE SPRITE 0.

## Cómo definir un *sprite* de 16 × 16

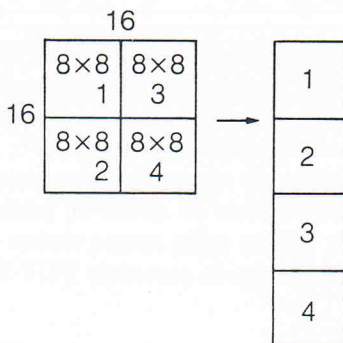
128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1
.	.	.	1	1	1	1	1	1	1	1	1	1	1	.	.
.	.	1	1	1	1	1	1	1	1	1	1	1	1	1	.
.	1	1	.	.	.	.	.	.	.	.	.	.	1	1	.
1	1	.	.	.	1	1	1	1	1	1	1	.	.	1	1
1	1	.	.	1	1	1	1	1	1	1	1	.	.	1	1
1	1	.	.	1	.	.	.	.	.	.	1	.	.	1	1
.	1	1	.	1	.	.	.	.	.	.	1	.	1	1	.
.	1	1	.	.	1	.	.	.	.	1	.	.	1	1	.
.	.	1	1	.	1	1	1	1	.	.	1	1	1	.	.
.	.	1	1	.	1	.	.	.	.	1	1	1	.	.	.
.	.	.	1	1	.	1	1	1	.	1	1	.	.	.	.
.	.	.	.	1	1	.	.	.	.	1	1	.	.	.	.
.	.	.	.	1	1	.	.	.	.	1	1	.	.	.	.
.	.	.	.	.	1	1	.	.	.	1	1	.	.	.	.
.	.	.	.	.	.	1	1	1	1	.	.	.	.	.	.
.	.	.	.	.	.	1	1	1	1	.	.	.	.	.	.
.	.	.	.	.	.	.	1	1	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	1	1	.	.	.	.	.	.

El diagrama arriba indicado, expresado en binario y decimal:

00011111	= 31	11111000	= 248
00111111	= 63	11111100	= 252
01100000	= 96	00000110	= 6
11000111	= 199	11100011	= 227
11001000	= 200	00010011	= 19
01101000	= 104	00010110	= 22
01100100	= 100	00100110	= 38
00110011	= 51	11001100	= 204
00110100	= 52	00101100	= 44
00011011	= 27	11011000	= 216
00011000	= 24	00011000	= 24
00001100	= 12	00110000	= 48
00001100	= 12	00110000	= 48
00000110	= 6	01100000	= 96
00000011	= 3	11000000	= 192
00000001	= 1	10000000	= 128



Los modelos de los *sprites* de  $16 \times 16$  se almacenan en la RAM de video de la siguiente forma:



Un `SPRITE$` de  $16 \times 16$  = “*sprite* del cuadrante 1” + “*sprite* del cuadrante 2” + “*sprite* del cuadrante 3” + “*sprite* del cuadrante 4”.

Aquí tienes un programa que define y visualiza el *sprite* de  $16 \times 16$  que hemos diseñado anteriormente. Los datos se almacenan en las instrucciones `DATA` y suman uno a uno a `SPRITE$(0)` dentro del bucle `FOR/TO/NEXT`.

```

○ 10 SCREEN 2,2
20 FOR I=1 TO 32
30 READ B%
40 S%=S%+CHR$(B%)
50 NEXT I
60 SPRITE$(0)=S%
70 PUT SPRITE 0, (100,100), 15,0
80 GOTO 80
90 DATA 31,63,96,199,200,104,100,51
100 DATA 52,27,24,12,12,6,3,1
110 DATA 248,252,6,227,19,22,38,204
120 DATA 44,216,24,48,48,96,192,128
○
```

- LINEA 10 SELECCIONA LA PANTALLA EN MODO 2 CON UN TAMAÑO PARA EL SPRITE DE  $16 \times 16$ .
- LINEA 20 BUCLE PARA LEER 32 BYTES.
- LINEA 30 LEE.
- LINEA 40 AÑADE DATOS A `S$`.
- LINEA 60 DEFINE LA CADENA DEL SPRITE 0.
- LINEA 70 PONE EL SPRITE 0 EN EL PLANO 0 DANDOLE COLOR BLANCO.
- LINEA 90 DATOS PARA EL CUADRANTE 1.
- LINEA 100 DATOS PARA EL CUADRANTE 2.
- LINEA 105 DATOS PARA EL CUADRANTE 3.
- LINEA 110 DATOS PARA EL CUADRANTE 4.

# Cómo situar un *sprite* en pantalla:

## PUT SPRITE

Delante de los planos de texto/gráficos hay 32 planos de *sprites*, siendo el plano *sprite* número 0 el más exterior de todos. O sea, cuanto menor sea el número de plano, mayor es su prioridad. Si hay dos *sprites* superpuestos, el que tenga el número de plano de *sprite* menor se verá por delante, quedando el otro escondido detrás de él.

Puedes definir hasta 256 modelos de *sprite* diferentes, pero sólo puedes tener un *sprite* en cada plano. Esto limita el número máximo de *sprites* en pantalla a 32.

En una línea horizontal sólo puedes tener cuatro *sprites* como máximo.

Para situar un *sprite* en pantalla usa la sentencia PUT SPRITE. Esta tiene la siguiente sintaxis:

```
PUT SPRITE <número de plano sprite> [, <especificación de coordenadas> ],  
[ <color> ] [, <número de modelo> ]  
<número del plano sprite> tiene un rango de 0 a 31  
<especificación de coordenadas> indica el punto donde se situará la esquina  
superior izquierda del sprite en pantalla.
```

Hay dos formatos para <especificación de coordenadas>:

1. (<X-coordenada>, <Y-coordenada>)  
(especifica un punto absoluto).
2. STEP (<X-incremento>, <Y-incremento>)  
(de las coordenadas del punto relativas al último punto que se hizo referencia).

<X-incremento>, <Y-incremento>, <X-coordenada>, <Y-coordenada> pueden ser variables o expresiones, así como simples constantes numéricas. Lo que significa que los *sprites* se pueden controlar mediante variables, permitiendo el movimiento por pantalla.

Si <especificación de coordenada> se omite, la instrucción PUT SPRITE situará el *sprite* en el último punto al que se hizo referencia.

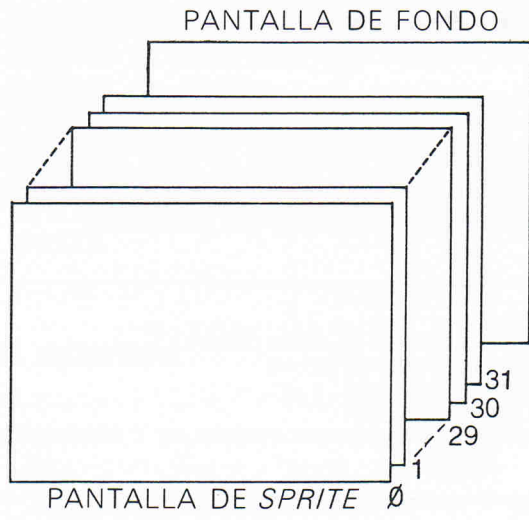
*Ejemplos:*

```
PUT SPRITE 0,(50,50),1,1  
PUT SPRITE 1,STEP)10,10),12,2
```

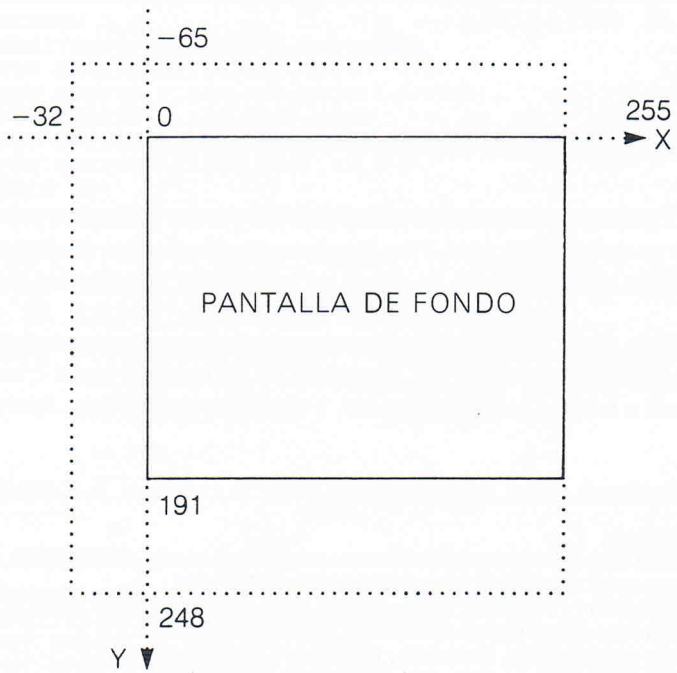
La pantalla de *sprites* es ligeramente más grande que la principal. La coordenada X está entre -32 y 255, y la coordenada Y entre -65 y 248. La pantalla principal tiene un rango menor, por lo que si el *sprite* se sitúa fuera de las coordenadas de esta pantalla, éste quedará escondido total o parcialmente.

Los *sprites* tienen la propiedad de tener la pantalla unida por sus extremos. Por lo que si un *sprite*, en su movimiento, se sale fuera de la pantalla, aparece de nuevo por el extremo opuesto.





La pantalla de *sprite*:



## Movimiento de *sprites*

Variando el valor del especificador de coordenadas de la instrucción PUT SPRITE, puedes mover los *sprites* por toda la pantalla. Cuando se ejecuta la instrucción PUT SPRITE se borra automáticamente el *sprite* que se hallaba en dicha pantalla de *sprites*.

Este programa te muestra un cuadrado que se mueve de derecha a izquierda, desapareciendo por la izquierda y reapareciendo por la derecha:

```
○ 10 SCREEN 2,0 ○
○ 20 SPRITE$(0)=STRING$(8,CHR$(255)) ○
○ 30 FOR X=200 TO -200 STEP -1 ○
○ 40 PUT SPRITE 0,(X,100),1,0 ○
○ 50 FOR D=1 TO 50: NEXT ○
○ 60 NEXT ○
○ 70 END ○
```

LINEA 10 SELECCIONA PANTALLA, TAMAÑO NORMAL.  
LINEA 20 DEFINE UN SPRITE CUADRADO.  
LINEA 40 PONE EL SPRITE EN X,100.  
LINEA 50 RETARDO.

## El color de los *sprites*

Puedes usar los dieciséis colores. Sin embargo, sólo puedes utilizar un color por cada *sprite*; es imposible tener un *sprite* multicolor. Para los *sprites* multicolor has de situar dos o más *sprites* de diferentes colores, uno encima de otro, en diferentes planos de *sprites* y moverlos conjuntamente.

El color de fondo de un *sprite* es siempre transparente y, por tanto, puedes ver a través de él.

Este ejemplo define dos *sprites* 0 y 1. El *sprite* 0, en blanco, y el *sprite* 1, en amarillo oscuro. En la pantalla verás a los dos *sprites* separados, y también superpuestos aparentando un *sprite* multicolor.

```
○ 10 SCREEN 2,0 ○
○ 20 SPRITE$(0)=CHR$(16)+CHR$(48)+CHR$(112)+CHR$(255) ○
○ )+CHR$(255)+CHR$(112)+CHR$(48)+CHR$(16) ○
○ 30 SPRITE$(1)=CHR$(224)+CHR$(192)+CHR$(128)+CHR$(0) ○
○ )+CHR$(0)+CHR$(128)+CHR$(192)+CHR$(224) ○
○ 40 PUT SPRITE 0,(20,20),15,0 ○
○ 50 PUT SPRITE 1,(40,40),10,1 ○
○ 60 PUT SPRITE 2,(60,60),15,0 ○
○ 70 PUT SPRITE 3,(60,60),10,1 ○
○ 80 GOTO 80 ○
```



LINEA 10 SELECCIONA LA PANTALLA, TAMAÑO NORMAL.  
 LINEA 20 DEFINE EL SPRITE 0.  
 LINEA 30 DEFINE EL SPRITE 1.  
 LINEA 40 PONE EL SPRITE 0 CON COLOR BLANCO.  
 LINEA 50 PONE EL SPRITE 1 CON COLOR AMARILLO OSCURO.  
 LINEA 60 PONE LOS SPRITES 2 y 3 EN LAS MISMAS COORDENADAS  
 UTILIZANDO DOS COLORES, EN DIFERENTES PLANOS DE  
 SPRITE.

## Escondiendo *sprites*

Asignando a la coordenada Y un número especial, puedes esconder tus *sprites*:

Y = 208

Si a la coordenada Y la asignas el valor 208 (&HD0), desaparecerán todos los planos de *sprite* con mayor número hasta que cambies el valor de Y.

○	10 SCREEN 2,0	○
	20 SPRITE\$(0)=STRING\$(8,CHR\$(255))	
	30 PUT SPRITE 0,(20,20),15,0	
○	40 PUT SPRITE 1,(40,40),15,0	○
	50 PUT SPRITE 2,(60,208),15,0	
	60 PUT SPRITE 3,(80,80),15,0	
○	70 PUT SPRITE 4,(100,100),15,0	○
	80 GOTO 80	

LINEA 10 SELECCIONA LA PANTALLA, TAMAÑO NORMAL.  
 LINEA 20 DEFINE UN SPRITE CUADRADO.  
 LINEA 50 Y = 208. INUTILIZA LOS PLANOS DE SPRITE CON UN  
 NUMERO MAYOR DE 2: LOS SPRITES DE LAS LINEAS 60 Y 70  
 NO APARECEN.

Si a la coordenada Y le damos el valor 209 (&HD1) el *sprite* desaparece de pantalla.

○	10 SCREEN 2,0	○
	20 SPRITE\$(0)=STRING\$(8,CHR\$(255))	
○	30 PUT SPRITE 0,(20,20),15,0	○
	40 PUT SPRITE 1,(40,40),15,0	
○	50 PUT SPRITE 2,(60,209),15,0	○
	60 PUT SPRITE 3,(80,80),15,0	
	70 PUT SPRITE 4,(100,100),15,0	
○	80 GOTO 80	○

LINEA 10 SELECCIONA PANTALLA, TAMAÑO NORMAL.  
 LINEA 20 DEFINE UN SPRITE CUADRADO.  
 LINEA 30 SPRITE CUADRADO EN PLANO 0.  
 LINEA 40 SPRITE CUADRADO EN PLANO 1.  
 LINEA 50 Y = 209. DESAPARECE EL SPRITE.  
 LINEA 60 SPRITE CUADRADO EN PLANO 3.  
 LINEA 70 SPRITE CUADRADO EN PLANO 4.

## La "quinta" regla de los *sprites*

El número máximo de *sprites* que puede haber en una línea horizontal es cuatro. Si violas esta regla sólo se verán los cuatro últimos *sprites* referidos; el resto no se mostrará en dicha línea. En el registro de estado del VDP tienes el número del "quinto" *sprite* que rompió la regla; lo puedes obtener mediante la función VDP. (Véase la parte correspondiente a la función VDP.)

Ejemplo:

○	10 SCREEN 2,0	○
	20 SPRITE\$(0)=STRING\$(8,CHR\$(255))	
○	30 PUT SPRITE 0,(20,100),15,0	○
	40 PUT SPRITE 1,(40,100),15,0	
○	50 PUT SPRITE 2,(60,100),15,0	○
	60 PUT SPRITE 3,(80,100),15,0	
○	70 PUT SPRITE 4,(100,104),15,0	○
○	80 GOTO 80	○

LINEA 10 SELECCIONA PANTALLA, TAMAÑO NORMAL.  
 LINEA 20 DEFINE UN SPRITE CUADRADO.  
 LINEA 30 SPRITE CUADRADO EN LINEA 100.  
 LINEA 40 SPRITE CUADRADO EN LINEA 100.  
 LINEA 50 SPRITE CUADRADO EN LINEA 100.  
 LINEA 60 SPRITE CUADRADO EN LINEA 100.  
 LINEA 70 SPRITE CUADRADO EN LINEA 104.

En el ejemplo de arriba verás cuatro *sprites* cuadrados, en la misma línea horizontal (100). Del quinto *sprite*, que está en la línea 104, se ve la mitad; la otra mitad está escondida, a consecuencia de la regla del "quinto" *sprite*. Si estuviera en la línea 108 se vería por entero.

## Animación de *sprites*

Puedes programar con tu MSX una simple animación de *sprites*. Todo lo que tienes que hacer es intercambiar dos *sprites* rápidamente, con lo que parecerá que están vivos.



El siguiente programa te muestra un invasor del espacio al que se le mueven las piernas. El SPRITE\$(0) es el invasor con las piernas cerradas, y el SPRITE\$(1) es el invasor con las piernas abiertas.

```

○ 10 SCREEN 2,1
○ 20 SPRITE$(0)=CHR$(60)+CHR$(126)+CHR$(129)+CHR$(21
○ 9)+CHR$(126)+CHR$(36)+CHR$(36)+CHR$(36)
○ 30 SPRITE$(1)=CHR$(60)+CHR$(126)+CHR$(129)+CHR$(21
○ 9)+CHR$(126)+CHR$(36)+CHR$(60)+CHR$(129)
○ 40 PUT SPRITE 0,(100,100),11,0
○ 50 FOR I=1 TO 500: NEXT
○ 60 PUT SPRITE 0,(100,100),11,1
○ 70 FOR I=1 TO 500: NEXT
○ 80 GOTO 40

```

LINEA 20 INVASOR CON PIERNAS CERRADAS.  
 LINEA 30 INVASOR CON PIERNAS ABIERTAS.  
 LINEA 40 SPRITE 0 (PIERNAS CERRADAS) AMARILLO.  
 LINEA 50 RETARDO.  
 LINEA 60 SPRITE 1 (PIERNAS ABIERTAS) AMARILLO.  
 LINEA 70 RETARDO.

### Programa de demostración

Con este ejemplo verás dos planetas girando alrededor de una estrella. El sistema aparecerá por el lado superior izquierdo de la pantalla.

```

○ 10 SCREEN 2,0
○ 20 COLOR 15,1,1
○ 30 CLS
○ 40 SPRITE$(0)=CHR$(126)+STRING$(6,CHR$(255))+CHR$(
○ 126)
○ 50 SPRITE$(1)=STRING$(3,CHR$(0))+CHR$(24)+CHR$(24
○ )+STRING$(3,CHR$(0))
○ 60 FOR I=0 TO 6.28 STEP .2
○ 70 X=X+1.5
○ 80 Y=Y+1
○ 90 X1=30*COS(I)
○ 100 Y1=30*SIN(I)
○ 110 X2=15*COS(I)
○ 120 Y2=15*SIN(I)
○ 130 PUT SPRITE 0,(X,Y),11,0
○ 140 PUT SPRITE 1,(X1+X,Y1+Y),9,1
○ 150 PUT SPRITE 2,(X2+X,Y2+Y),15,1
○ 160 NEXT
○ 170 GOTO 60

```

LINEA 10 SELECCIONA LA PANTALLA, TAMAÑO NORMAL, FONDO Y BORDE NEGROS, TEXTO BLANCO.  
 LINEA 30 LIMPIA LA PANTALLA.

```

LINEA 40  EL SPRITE 0 ES LA ESTRELLA.
LINEA 50  EL SPRITE 1 ES EL DISEÑO DEL PLANETA.
LINEA 60  BUCLE.
LINEA 70  MOVER LA ESTRELLA CON X DE 1.5 EN 1.5.
LINEA 80  MOVER LA ESTRELLA CON Y DE 1 EN 1.
LINEA 90  X1 = X-COORDENADAS PLANETA 1.
LINEA 100 Y1 = Y-COORDENADAS PLANETA 1.
LINEA 110 X2 = X-COORDENADAS PLANETA 2.
LINEA 120 Y2 = Y-COORDENADAS PLANETA 2.
LINEA 130 MUEVE LA ESTRELLA A UNA NUEVA POSICION.
LINEA 140 MUEVE EL PLANETA 1 A UNA NUEVA POSICION.
LINEA 150 MUEVE EL PLANETA 2 A UNA NUEVA POSICION.
LINEA 160 SIGUIENTE BUCLE.
LINEA 170 COMIENZO DEL BUCLE DE NUEVO.

```

## Choque de *sprites*

El TMS 9918/9929 VDP puede detectar colisiones de *sprites*. Es muy útil para los juegos con choques de *sprites*. Hay dos instrucciones BASIC que permiten la detección de los choques: ON SPRITE GOSUB y SPRITE ON/OFF/STOP.

La instrucción ON SPRITE GOSUB fija la subrutina de interrupción por choque de *sprites*. Indica al ordenador en qué línea comienza la subrutina de tratamiento de dichas interrupciones.

La instrucción SPRITE ON activa la detección de los choques haciendo saltar el programa a la subrutina especificada por la instrucción ON SPRITE GOSUB, cuando choquen dos *sprites*. Has de ejecutar la orden SPRITE ON para poder detectar las colisiones.

Si un punto de dos *sprites* se superpone, se considerará como una colisión. Incluso si los dos puntos superpuestos son transparentes.

Aunque el ordenador puede detectar los choques, no puede decirte qué *sprites* han chocado, ni tampoco dónde se produjo la colisión.

Cuando haya una interrupción, o sea, dos *sprites* colisionan, se ejecuta automáticamente SPRITE STOP. Así se evita que se llame de nuevo a la subrutina de tratamiento, mientras se esté tratando la colisión actual y hubiese otro nuevo choque. Pero recuerda que si hubo otro choque durante la ejecución de la subrutina, saltará inmediatamente a dicha subrutina una vez haya ejecutado por completo la actual, a menos que dentro de la subrutina inhibas la detección de colisiones mediante SPRITE OFF.

Después de abandonar la subrutina de tratamiento el ordenador ejecutará automáticamente SPRITE ON, permitiendo otra interrupción, excepto si en la subrutina incluiste un SPRITE OFF.

La instrucción SPRITE OFF desactiva la detección de colisiones.

*Ejemplo:*

En este ejemplo verás dos *sprites* cuadrados: uno amarillo y otro blanco, acercándose el uno al otro desde lados opuestos de la pantalla. Cuando choquen, la instrucción ON SPRITE GOSUB se activará y hará que la ejecución continúe en la subrutina de tratamiento. La

instrucción `SPRITE OFF`, dentro de la rutina de interrupción de *sprites*, evita la detección de más choques. Incluimos esta orden ya que los dos *sprites* continuarán chocando después de la ejecución de la subrutina.

Si no se ejecutase dicha instrucción, el ordenador ejecutaría de forma infinita la subrutina de tratamiento (prueba esta rutina sin `SPRITE OFF` y observa lo que sucede).

○	10 ON SPRITE GOSUB 110	○
	20 SCREEN 2,0	
○	30 SPRITE\$(0)=STRING\$(8,CHR\$(255))	○
	40 SPRITE\$(1)=STRING\$(8,CHR\$(255))	
	50 SPRITE ON	
○	60 FOR I=10 TO 240	○
	70 PUT SPRITE 0,(I,100),11,0	
	80 PUT SPRITE 1,(250-I,100),15,1	
○	90 NEXT I	○
	100 GOTO 50	
○	105 REM Rutina de choque de sprites	○
	110 SPRITE OFF	
	120 BEEP	
○	130 RETURN	○

LINEA 10 FIJA LA SUBROUTINA EN LA LINEA 110.  
LINEA 20 PANTALLA GRAFICA.  
LINEA 30 EL SPRITE 0 ES UN CUADRADO.  
LINEA 40 EL SPRITE 1 ES UN CUADRADO.  
LINEA 50 ACTIVA LA DETECCION DE COLISION DE SPRITES.  
LINEA 60 BUCLE.  
LINEA 70 SPRITE (AMARILLO) SE MUEVE DESDE LA IZQUIERDA.  
LINEA 80 SPRITE (BLANCO) SE MUEVE DESDE LA DERECHA.  
LINEA 90 SIGUIENTE BUCLE.  
LINEA 100 COMIENZA EL NUEVO BUCLE.  
LINEA 110 INHIBE LA DETECCION (CON UNA SOLA VEZ BASTA).  
LINEA 120 PITIDO DE AVISO.  
LINEA 130 VUELVE AL BUCLE.

## Notas sobre colisión de *sprites*

Cada vez que se ejecute una instrucción, el ordenador analiza si se ha producido una colisión de *sprites*. No se puede decir que se busque un choque de *sprites* en un determinado instante.

La función `VDP` te puede dar el estado del choque de *sprites* del registro 9 del procesador de video. (Véase la parte correspondiente al `VDP`.)



TABLE I

Summary of the results of the experiments

Experiment	No. of subjects	Mean score		S.E.M.
		Pretest	Posttest	
1	10	1.5	2.5	0.5
2	10	1.5	2.5	0.5
3	10	1.5	2.5	0.5
4	10	1.5	2.5	0.5
5	10	1.5	2.5	0.5
6	10	1.5	2.5	0.5
7	10	1.5	2.5	0.5
8	10	1.5	2.5	0.5
9	10	1.5	2.5	0.5
10	10	1.5	2.5	0.5
11	10	1.5	2.5	0.5
12	10	1.5	2.5	0.5
13	10	1.5	2.5	0.5
14	10	1.5	2.5	0.5
15	10	1.5	2.5	0.5
16	10	1.5	2.5	0.5
17	10	1.5	2.5	0.5
18	10	1.5	2.5	0.5
19	10	1.5	2.5	0.5
20	10	1.5	2.5	0.5

The results of the experiments are summarized in Table I. The mean scores for the pretest and posttest are shown for each experiment. The standard error of the mean (S.E.M.) is also indicated. The results show that the mean scores for the posttest are generally higher than the mean scores for the pretest, indicating that the subjects improved their performance after the intervention. The S.E.M. values are relatively small, suggesting that the results are consistent across the subjects in each experiment.

The data in Table I shows that for all 20 experiments, the mean score increased from the pretest to the posttest. The increase in mean score was consistent across all experiments, with the posttest mean score being approximately 1.0 unit higher than the pretest mean score. The S.E.M. for each experiment was 0.5, which is a small value relative to the mean scores, indicating that the results are reliable and consistent across the subjects in each experiment.

The overall results of the experiments suggest that the intervention had a positive effect on the subjects' performance. The consistent increase in mean scores across all experiments indicates that the subjects improved their performance after the intervention. The small S.E.M. values further support the reliability of the results, suggesting that the improvement was not due to chance or individual differences among the subjects.

# Gráficos avanzados (II)

## Acceso al procesador de video

### Función VDP

El VDP TMS 9929 A tiene ocho registros de sólo escritura, y un registro de sólo lectura. Puedes acceder a estos registros desde el BASIC mediante la función VDP.

A los registros de sólo escritura del VDP(0) al VDP(7) únicamente les puedes asignar valores; por ejemplo, VDP(1)=2. No hay modo de leer estos registros directamente desde el VDP.

Sin embargo, el ordenador guarda una copia de estos registros en el área de trabajo del sistema, en las posiciones RG0SAV a RG7SAV. (Véase la parte de la RAM del sistema, tema de las BIOS.)

VDP(8) es el registro de sólo lectura. Para leer el contenido del registro escribe:

```
PRINT VDP(8)
```

Recuerda que si asignas un número erróneo a estos registros la pantalla se podría bloquear. Si te sucede esto debes pulsar RESET o apagar el ordenador para inicializar el VDP. Antes de utilizar la función VDP asegúrate de lo que haces.

# Registros del VDP

## Registros de sólo escritura

### Registro 0 ... VDP(0)

El registro 0 tiene dos bits de control, BIT 1(M3) y BIT 0(EV). Del BIT 2 al BIT 7 deben ser 0.

BIT 0 EV: VDP externo. Posibilita la entrada de datos desde otro VDP externo.  
EV = 0 en casi todos los MSX.

0 = inactivo

1 = activo

BIT 1 M3: Bit de modo 3. El M3 se utiliza para seleccionar el modo de pantalla conjuntamente con el registro 1 (véase registro 1).

RO

B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	0	0	M3	EV=0

### Registro 1 ... VDP(1)

El registro 1 tiene 7 bits de control del VDP. El BIT 2 está reservado, y debe estar siempre a 0.

BIT 0 MAG: Ampliación de *sprites*.

0 = Sin ampliar resolución de  $1 \times 1$  puntos.

1 = Ampliados, resolución de  $2 \times 2$  puntos.

BIT 1 SIZE: Selector del tamaño *sprites*

0 = *Sprites* de  $8 \times 8$  puntos.

1 = *Sprites* de  $16 \times 16$  puntos.

BIT 2 Reservado = 0

BIT 3 M2: Bit de modo 2. El M2 se emplea para seleccionar el modo de pantalla (véase bit 4).

BIT 4 M1: Bit de modo 1. El M1 se utiliza para seleccionar el modo de pantalla. La combinación de M1, M2 y M3 proporciona dicho modo.

M1 M2 M3

0 0 0 Modo 1 de texto

0 0 1 Modo 2 de alta resolución gráfica

0 1 0 Modo 3 de gráficos multicolor.

0 1 1 Modo 0 de texto

BIT 5 IE: Activa las interrupciones.

0 = Desactiva las interrupciones del VDP.

1 = Activa las interrupciones del VDP.

BIT 6 BL: Activa/desactiva la pantalla. Te permite desconectar la pantalla.



0 = Desactiva la pantalla poniéndola del color del borde.

1 = Activa la pantalla.

**BIT 7 VRAM:** Selecciona el tipo de RAM de video a utilizar (VRAM = 1 es lo normal en el MSX).

0 = 4K RAM

1 = 16K RAM

R1

B7	B6	B5	B4	B3	B2	B1	B0
VRAM	BL	IE	M1	M2	0	SIZE	MAG

### Registro 2... VDP(2)

El registro 2 define la dirección base de la tabla de nombres. Su rango se encuentra entre 0 y 15. Este registro tiene los 4 bits más significativos de los 14 de la dirección de la tabla de nombres. Por tanto, debes dividir por &H400 la dirección.

R2

B7	B6	B5	B4	B3-	B2	B1	B0
0	0	0	0	DIRECCION DE LA TABLA DE NOMBRES			

*Nota:*  $R2 * \&H400 = \text{DIRECCION DE LA TABLA DE NOMBRES}$

### Registro 3... VDP(3)

El registro 3 define la dirección base de la tabla de color. Su rango se encuentra entre 0 y 255. Este registro representa a los 8 bits más significativos de los 14 de la dirección de la tabla de color. Por tanto, deberás escribir la dirección dividida por &H40.

R3

B7	B6	B5	B4	B3	B2	B1	B0
DIRECCION DE LA TABLA DE COLOR							

*Nota:*  $R3 * \&H40 = \text{DIRECCION DE LA TABLA DE COLOR}$

### Registro 4... VDP(4)

El registro 4 define la dirección base de la tabla generadora de modelos, texto o multicolor, dependiendo del modo de pantalla.

Su rango se encuentra entre 0 y 7. Este registro representa los 3 bits más significativos de los 14 de la dirección de la tabla generadora de modelos/texto/multicolor. Por tanto, has de escribir la dirección dividida por &H800.

R4	B7	B6	B5	B4	B3	B2	B1	B0
	0	0	0	0	0	DIRECCION TABLA D./T./M.		

*Nota:* R4\* & H800 = DIRECCIONES DE LA TABLA GENERADORA DE MODELOS/  
TEXTO/MULTICOLOR

### Registro 5 ... VDP(5)

El registro 5 define la dirección base de la tabla de atributos de *sprites*. Su rango se encuentra entre 0 y 127. Este registro representa los 7 bits más significativos de los 14 de la dirección de la tabla de atributos de *sprites*. Por tanto, debes escribir la dirección dividida por &H80.

R5	B7	B6	B5	B4	B3	B2	B1	B0
	0	DIRECCION TABLA DE ATRIBUTOS DE <i>SPRITES</i>						

*Nota:* R5\* & H80 = DIRECCION DE LA TABLA DE ATRIBUTOS DE *SPRITES*

### Registro 6 ... VDP(6)

El registro 6 define la dirección base de la tabla generadora de modelos de *sprite*. Su rango se encuentra entre 0 y 127. Este registro representa los 3 bits más significativos de los 14 de la dirección de la tabla generadora de diseños de *sprite*. Por tanto, debes escribir la dirección dividida por &H800.

R6	B7	B6	B5	B4	B3	B2	B1	B0
	0	0	0	0	0	DIRECCION DE LA TABLA GEN. MO. SP.		

*Nota:* R6\* & H800 = DIRECCION DE LA TABLA GENERADORA DE DISEÑOS  
DE *SPRITE*

### Registro 7 ... VDP(7)

El registro 7 está dividido en dos partes. Los 4 bits más significativos contienen el color de la tinta en modo texto, mientras que los 4 menos significativos tienen el color del fondo en los modos texto y gráfico.

El rango para el color del texto está entre 0 y 15, al igual que para el color de fondo.

$$R7 = \langle \text{color del texto} \rangle * \&H10 + \langle \text{color del fondo} \rangle$$

R7	B7	B6	B5	B4	B3	B2	B1	B0
	COLOR DE LA TINTA				COLOR DEL FONDO			

## Registro de sólo lectura

### El registro de estado ... VDP(8)

Este registro tiene el *flag* pendiente de interrupción, el *flag* de colisión de *sprites* y el *flag* del quinto *sprite*.

#### **Flag de interrupción F: BIT 7**

El *flag* de interrupción se activa (1) cuando ha terminado el análisis de pantalla. Se pone a 0 después de leer el registro de estado, o cuando se inicializa el VDP desde fuera.

#### **Flag de colisión de *sprite* C: BIT 5**

El VDP analiza si dos o más *sprites* se superponen en un punto. Este análisis se hace cada 1/50 segundo.

- 1 = colisión de *sprites*
- 0 = no hay superposición

#### **Flag del quinto *sprite* 5S: BIT 6**

El 5S se activa cuando hay más de cuatro *sprites* en una línea horizontal (líneas 0 a 192). El VDP no puede tener más de cuatro *sprites* en una sola línea. Si hay más de cuatro *sprites* no se visualizarán a partir del quinto.

#### **Número del quinto *sprite***

Si hay más de cuatro *sprites* en una línea horizontal y 5S se activa, entonces el número de *sprite* que ha roto la regla se guarda en los 5 bits menos significativos de este registro.

R2	B7	B6	B5	B4	B3	B2	B1	B0
	F	5S	C	NUMERO DEL QUINTO <i>SPRITE</i>				



The following table shows the results of the experiment. The first column shows the number of trials, the second column shows the number of correct responses, and the third column shows the percentage of correct responses.

The results show that the number of correct responses increases with the number of trials. This suggests that the subjects are learning the task as they go along.

The percentage of correct responses also increases with the number of trials. This suggests that the subjects are becoming more accurate as they practice.

The overall results of the experiment are positive. The subjects showed improvement in both the number of correct responses and the percentage of correct responses over time.

The following table shows the results of the experiment. The first column shows the number of trials, the second column shows the number of correct responses, and the third column shows the percentage of correct responses.

The results show that the number of correct responses increases with the number of trials. This suggests that the subjects are learning the task as they go along.

The percentage of correct responses also increases with the number of trials. This suggests that the subjects are becoming more accurate as they practice.

The overall results of the experiment are positive. The subjects showed improvement in both the number of correct responses and the percentage of correct responses over time.

The following table shows the results of the experiment. The first column shows the number of trials, the second column shows the number of correct responses, and the third column shows the percentage of correct responses.

# Gráficos avanzados (III)

## La RAM de video

El MSX posee una RAM de video de 16K separada de la memoria principal. El procesador de video la trata y refresca; es independiente del procesador central Z-80 (la RAM dinámica utilizada por el MSX ha de ser refrescada para no perder los datos almacenados).

Esto tiene las siguientes ventajas:

1. El procesador Z-80 no necesita refrescar a la RAM de pantalla, por lo que ahorra tiempo.
2. La pantalla y los gráficos no utilizan la memoria principal utilizada por el Z-80, por lo que el procesador central tiene más memoria.

La RAM de video está dividida en secciones; cada una desempeña una única función. Es posible acceder a la RAM de video desde el BASIC mediante las siguientes instrucciones y funciones:

BASE	Proporciona la dirección base de varias tablas de la RAM de video.
VPEEK	Lee el contenido de una posición de la RAM de video.
VPOKE	Escribe en una posición de la RAM de video.

La función BASE da la dirección base actual de todas las tablas de pantalla de la RAM de video (para más detalles, véase BASE).

Nombre de tabla y dirección en hex.	Modo de texto 0	Modo de texto 1	Modo gráfico 2	Modo gráfico 3
Tabla de modelos	&H0000	&H1800	&H1800	&H0800
Tabla de color		&H2000	&H2000	
Tabla generadora de modelos	&H0800	&H0000	&H0000	&H0000
Tabla de atributos de <i>sprites</i>		&H1B00	&H1B00	&H1B00
Tabla generadora de <i>sprites</i>		&H3800	&H3800	&H3800



# Efectos de sonido con el PSG

El generador de sonido del MSX, AY-3-8910, lleva en el mercado bastante tiempo. Fabricado por General Instruments a finales de los 70, es todavía uno de los *chips* de sonido más populares. Se utiliza mucho en los sistemas de ordenadores de 8 bits y en las máquinas de juegos. La razón de su éxito es su flexibilidad y facilidad para generar tres voces, con sonido periódico, y un canal de sonido puro, simultáneamente. También porque tiene un microprocesador que no interrumpe al procesador central mientras ejecuta la música. Aunque actualmente está algo desfasado, puedes sacarle un gran partido.

Esta parte trata sobre el funcionamiento del generador de sonido AY-3-8910 y de la construcción SOUND.

A continuación, como ejemplo, prueba el siguiente programa que genera un fantástico ruido de reactor:

<input type="radio"/>	10 SOUND 0,87	<input type="radio"/>
<input type="radio"/>	20 SOUND 7,62	<input type="radio"/>
<input type="radio"/>	30 SOUND 8,16	<input type="radio"/>
<input type="radio"/>	40 SOUND 11,179	<input type="radio"/>
<input type="radio"/>	50 SOUND 12,45	<input type="radio"/>
<input type="radio"/>	60 SOUND 13,14	<input type="radio"/>

# Formas de onda generadas por el PSG

Básicamente el PSG sólo puede generar dos tipos de formas de onda: una onda periódica cuadrada y un sonido uniforme (llano).

Ejemplo de onda periódica		Ejemplo de sonido uniforme	
○	10 SOUND 7,62	10 SOUND 7,55	○
	20 SOUND 8,15	20 SOUND 8,15	
○	30 SOUND 1,1		○
	40 SOUND 0,28		

Estas ondas se pueden combinar y modificar para formar ruidos complejos, intentando sonidos similares a los reales. Para ello debes controlar la frecuencia, el tono y la forma de onda de cada canal de sonido mediante la instrucción SOUND, con la que se accede al PSG.

## Registros de sonido PSG

El AY-3-8910 PSG tiene catorce registros en los que puedes escribir utilizando la instrucción SOUND. La sentencia SOUND posee la siguiente sintaxis:

SOUND <número de registro>, <valor asignado>

Asignando valores adecuados a estos registros puedes seleccionar los canales, ondas de sonido, frecuencia de tono, sonido y volumen.

Hay catorce registros, con las siguientes funciones:

REGISTRO	DESCRIPCION
0	Ajuste fino de frecuencia del canal A.
1	Ajuste básico de frecuencia del canal A.
2	Ajuste fino de frecuencia del canal B.
3	Ajuste básico de frecuencia del canal B.
4	Ajuste fino de frecuencia del canal C.
5	Ajuste básico de frecuencia del canal C.
6	Frecuencia del generador de ruido.
7	Mezclador de los canales A, B y C.
8	Control del volumen en el canal A.
9	Control del volumen en el canal B.
10	Control del volumen en el canal C.
11	Período de envolvente (bajo).
12	Período de envolvente (alto).
13	Modelo de onda.

## Elección del sonido de cada canal: registro 7, el mezclador

Este es el registro que activa o desactiva el sonido de los tres canales. Puede seleccionar la fuente de la onda de sonido desde el generador de sonido, o desde el generador de tono individual de cada canal.

Los bits 5, 4 y 3 del registro 7 activan el generador de sonido de los canales C, B y A, respectivamente. Con 1 está desconectado y con 0 está conectado. Los bits 2, 1 y 0 del registro 7 ponen en funcionamiento el generador de tono de los canales C, B y A, respectivamente. Por ejemplo, si quieres que el canal C emita un sonido, el bit 2 debe estar a 0.

BIT	B7	B6	B5	B4	B3	B2	B1	B0
	/	/	RUIDO			TONO		
	/	/	C	B	A	C	B	A
	0	0	1	1	1	0	1	1
			OFF	OFF	OFF	ON	OFF	OFF

El formato de la sentencia **SOUND**, según se encuentra reflejado arriba, sería:

**SOUND 7,&00111011;** o en decimal: **SOUND 7,59**

El tono del canal A depende de los registros 0 y 1; lo veremos más adelante.

Si deseas que el canal B emita ruido blanco, y C y A emitan según el formato anterior, tendrás que cambiar el registro 7 de la siguiente manera:

BIT	B7	B6	B5	B4	B3	B2	B1	B0
	/	/	RUIDO			TONO		
	/	/	C	B	A	C	B	A
	0	0	1	0	1	0	1	1
			OFF	ON	OFF	ON	OFF	OFF

El formato de la sentencia **SOUND**, según se encuentra reflejado arriba, sería:

**SOUND 7,&B00101011;** o en decimal: **SOUND 7,43**



Puedes hacer que un canal emita un ruido uniforme y un tono al mismo tiempo activando ambos generadores. Si deseas que todos los canales emitan tono y ruido, obtendrás:

BIT	B7	B6	B5	B4	B3	B2	B1	B0
	/	/	RUIDO			TONO		
	/	/	C	B	A	C	B	A
	0	0	0	0	0	0	0	0
			ON	ON	ON	ON	ON	ON

El formato de la sentencia SOUND, según se encuentra reflejado arriba, sería:

SOUND 7,&B00000000; o en decimal: SOUND 7,0

*Nota:* Los bits B7 y B6 no se utilizan, y por eso se encuentran a 0.

## Generador de tono: registros 0, 1, 2, 3, 4 y 5

Para fijar la frecuencia del generador de tono de cada canal escribe en los registros 0, 1, 2, 3, 4 y 5. Cada canal tiene dos registros que fijan la frecuencia de tono a generar.

Registro 0 y 1 para el canal A.  
Registro 2 y 3 para el canal B.  
Registro 4 y 5 para el canal C.

REGISTRO	RANGO	DESCRIPCION
0	0-255	Ajuste fino de la frecuencia del canal A.
1	0-15	Ajuste básico de la frecuencia del canal A.
2	0-255	Ajuste fino de la frecuencia del canal B.
3	0-15	Ajuste básico de la frecuencia del canal B.
4	0-255	Ajuste fino de la frecuencia del canal C.
5	0-15	Ajuste básico de la frecuencia del canal C.
6	0-31	Frecuencia del generador de sonido.

La frecuencia (f) se calcula como:

$$f = (\text{registro } 0) + (\text{registro } 1) \times 256$$

Esta fórmula no proporciona la frecuencia en Hz, pero cuanto mayor sea la frecuencia menor será el tono.

La frecuencia más alta se genera como SOUND 0,0: SOUND 1,0, que proporciona una  $f = 0$ ; y la frecuencia más baja se dará cuando SOUND 0, 255: SOUND 1,15.

Para obtener la frecuencia generada en Hz utiliza la siguiente fórmula:

$$f = \frac{178900}{(16 \times \text{Hz})}$$

178900 es la frecuencia del reloj del PSG (que será distinta en el modelo británico del MSX).

Si quieres que la frecuencia que genere el PSG sea exacta debes utilizar el cálculo anterior. Pero no lo hagas con una calculadora; utiliza en su lugar el ordenador.

```
○ 10 INPUT "FRECUENCIA EN HERTZIOS";HZ
○ 20 F=INT(1789800#/(16*HZ))
○ 30 PRINT "TONO FINO,REGISTROS 0,2 o 4: ";F MOD 256
○ 40 PRINT "TONO,REGISTROS 1,3, o 5: ";INT(F/256
```

Este programa te hará los cálculos.

*Ejemplo:*

Genera sonido de 2000 Hz por el canal A.

$$f = 1789800/(16*2000) = 55.9$$

```
○ 10 SOUND 0,255
○ 20 SOUND 1,0
○ 30 SOUND 7,62 ;MEZCLADOR
○ 40 SOUND 8,10 ;VOLUMEN
```

## Frecuencia del generador de sonido: registro 6

Para cambiar la frecuencia del generador de sonido uniforme escribe en el registro 6 del PSG.

Su rango se encuentra comprendido entre 0 y 31; cuanto más bajo sea el valor, más alta será la frecuencia.

Ejemplo:

```
○ 10 SOUND 7,55 ;CONECTA EL CANAL A ○
○ 20 SOUND 8,15 ;VOLUMEN DEL CANAL 1 AL 15 ○
○ 30 FOR F=0 TO 31 ○
○ 40 SOUND 6,F ○
○ 50 FOR I=1 TO 200: NEXT ;RETARDO ○
○ 60 NEXT F ○
```

Oirás un sonido uniforme con la frecuencia disminuyendo.

## Volumen: registros 8, 9, 10

El volumen del sonido emitido por los canales A, B y C se controla con los registros 8, 9 y 10, respectivamente.

sin sonido máximo volumen  
0 <-----> 15  
REGISTRO 8 = CANAL A  
REGISTRO 9 = CANAL B  
REGISTRO 10 = CANAL C

Cuando a estos registros se les asigne el valor 16, se activa la onda; o sea, el sonido varía su volumen según la forma de la onda y la frecuencia. La onda se define mediante los registros 11, 12 y 13.

Recuerda que el volumen del altavoz del televisor depende de su control de volumen. Si pones el volumen al mínimo no oirás nada. El volumen de sonido del MSX también depende del amplificador externo.

## Las ondas: registros 11, 12 y 13

Cuando no se emplean ondas el volumen del sonido permanece constante en el nivel fijado por los registros 8, 9 y 10. Esto es bastante aburrido; todo lo que oirás será un ruido de frecuencia constante. Sin embargo, el AY-3-8910 PSG te ofrece una amplia gama de ondas para cambiar periódicamente el volumen, de acuerdo con la forma de la onda elegida.

Cuando pones los registros de control del volumen 8, 9 y 10 a 16, activas las ondas. Al activar las ondas el volumen del sonido cambia según los valores de los registros 11, 12 y 13.

		<u>SIN ONDA</u>	<u>CON ONDA</u>
CANAL A	REGISTRO 8	0-15	16
CANAL B	REGISTRO 9	0-15	16
CANAL C	REGISTRO 10	0-15	16



La forma de la onda se selecciona poniendo en el registro 13 un número comprendido entre 0 y 15. El PSG tiene duplicadas algunas formas de onda, por lo que sólo tienes ocho formas de onda diferentes.

La frecuencia de cambio del volumen, o sea, el período de la onda, lo has de fijar en los registros 11 y 12; en el 12 ajustas el período y en el 11 proporcionas un ajuste más fino.

	<u>RANGO</u>
REGISTRO 11	0-255
REGISTRO 12	0-255

$$\text{Período} = (\text{registro 12}) + 256 * (\text{registro 11})$$

Sólo puedes tener una onda en cada instante, y los tres canales han de usar la misma forma de onda.

(Consultar SOUND, en la *Guía de referencia del BASIC*, para ver las formas de las ondas.)

1. **OBJETIVO**  
 2. **ALCANCE**  
 3. **DEFINICIONES**  
 4. **PROCEDIMIENTO**  
 5. **REVISIÓN**  
 6. **ANEXOS**

Este procedimiento tiene como objetivo establecer los requisitos mínimos para la selección y contratación de personal que desempeñará funciones de apoyo administrativo en el área de Recursos Humanos.

Este procedimiento aplica a todos los cargos de apoyo administrativo en el área de Recursos Humanos, tanto en el nivel de personal de confianza como en el nivel de personal de planta.

Este procedimiento no aplica para el personal que se contrata a través de los procedimientos de selección de personal de planta.

Este procedimiento se aplicará a partir de la fecha de su aprobación, por lo que cualquier proceso de selección en curso se continuará de acuerdo a lo establecido en el procedimiento que se estaba aplicando en ese momento.

### Letras de crédito: números 11, 12 y 13

Estas letras de crédito se emiten a favor de los beneficiarios que han sido seleccionados para ocupar los cargos de apoyo administrativo en el área de Recursos Humanos, de acuerdo a lo establecido en el presente procedimiento.

Estas letras de crédito se emiten a favor de los beneficiarios que han sido seleccionados para ocupar los cargos de apoyo administrativo en el área de Recursos Humanos, de acuerdo a lo establecido en el presente procedimiento.

CARGO	NOMBRE	FECHA	VALOR
CARGO A	NOMBRE A	01/01/2020	10
CARGO B	NOMBRE B	01/01/2020	10
CARGO C	NOMBRE C	01/01/2020	10

## **MSX: Guía del programador y manual de referencia**

---

T. Sato  
P. Mapstone  
I. Muriel

Un libro definitivo para el estándar MSX. Valioso para todos los usuarios del MSX, de principiantes a avanzados. Permite a los usuarios de todos los niveles obtener el máximo rendimiento de su ordenador. Está dividido en cuatro partes. La primera de ellas es un acercamiento a la máquina que desmitifica el aprendizaje de la programación. La segunda parte introduce al lector en técnicas avanzadas de programación, incluyendo información sobre sonido y gráficos. La explicación detallada del lenguaje BASIC y del lenguaje máquina ocupa la tercera parte, en tanto que la última se ocupa de explicarle cómo funciona su MSX dándole una guía completa de su sistema operativo.

## **Lenguaje máquina para MSX. Introducción y conceptos avanzados**

---

Joe Pritchard

El objetivo de este libro es introducir al propietario del ordenador MSX un manual de programación del Z-80 para que el programador pueda escribir rutinas que hagan uso de varios componentes del sistema MSX. Es un libro interesante para todos aquellos que quieran saber cómo aplicar sus ya existentes conocimientos de programación del Z-80 al ordenador MSX en este lenguaje.



## **Descubre tu MSX** **Programación y aplicaciones**

---

Joe Pritchard

Esta guía para sacar el máximo partido del MSX posee un eficaz planteamiento didáctico, con multitud de programas ejemplo, explicación detallada de todos los temas, y tablas que resumen los datos interesantes, como posiciones de memoria o variables importantes del sistema, que permite, desde los primeros pasos en BASIC, llegar al dominio absoluto de este ordenador.

## **Códigos y claves secretas** **Criptografía en BASIC**

---

Gareth Greenwood

Es una introducción a la criptología para usuarios de microordenadores. A lo largo del libro se describen y utilizan las distintas técnicas de cifrado y descifrado de mensajes y se acompañan los programas en BASIC para que el ordenador realice el trabajo repetitivo a partir del ingenio, la intuición y la destreza del usuario. Libro eminentemente ameno y práctico; no sólo introduce al lector en los principios básicos de las técnicas criptográficas, sino que le ofrece la posibilidad de profundizar en la materia y diversificar las aplicaciones y el ocio con su microordenador.

El libro contiene versiones de todos los programas para Spectrum, Amstrad, MSX, Commodore y Apple II.









**CATALONIA**

Casa del Libro, S.A.

Ronda de Sant Pere, 3 - T. 318 91 83 - 08010 Barcelona

¿Está tu MSX preparado para la invasión de las cucarachas cósmicas, de los espías del planeta Circón y de las naves-robot suicidas? ¿Conoces al gorila de Godofredo?

**EL LIBRO GIGANTE DE LOS JUEGOS PARA MSX** es la recopilación más fascinante que puedes encontrar de juegos que desafían a tu microordenador MSX.

Los listados están especialmente estructurados para que su transcripción y depuración sean fáciles y rápidas. Encontrarás juegos que son todo un reto para tus reflejos, tu lógica o tu intuición: simulaciones, juegos de aventura, mata-marcianos, juegos educativos... ¡Y muchos más!

Todos los programas utilizan al máximo la capacidad gráfica (manejo de "sprites") y de sonido de tu MSX, y muchos de ellos usan el código máquina. Con cada programa encontrarás la descripción de las variables utilizadas, el diseño de los "sprites" y sugerencias para que amplíes y modifiques los juegos hasta donde tu imaginación te lleve.

**EL LIBRO GIGANTE DE LOS JUEGOS PARA MSX** abre las puertas a una nueva y fascinante dimensión de los videojuegos.