APRENDE A PROGRAMAR

GRÁFICOS MÓVILES

LOS INVASORES

ANAYA

PARA MSX Y SPECTRUM

Diseño: Cooper West
Edición: James McCarter
Programas: Steve Rodgers
Programas MSX: Angel García García
Ilustraciones: Gerard Brown
Andrew Farmer

Traducción: Angel García García

El traductor desea expresar su agradecimiento a
Gustavo Flores Maza por su colaboración en este libro

Título original: MOVING GRAPHICS
© 1985, Aladdin Books Ltd.
© 1986, de la edición española, E. G. Anaya.
Villafranca, 22. 28028 Madrid.
I.S.B.N.: 84-7525-362-8
Depósito legal: M. 20985-1986
Impreso por: Edime, S. A. Calle D, esquina a F.
Polígono Industrial Arroyomolinos. Móstoles (Madrid)
Impreso en España - Printed in Spain.

APRENDE A PROGRAMAR

GRÁFICOS MÓVILES

Marcus Milton

ANAYA

Introducción

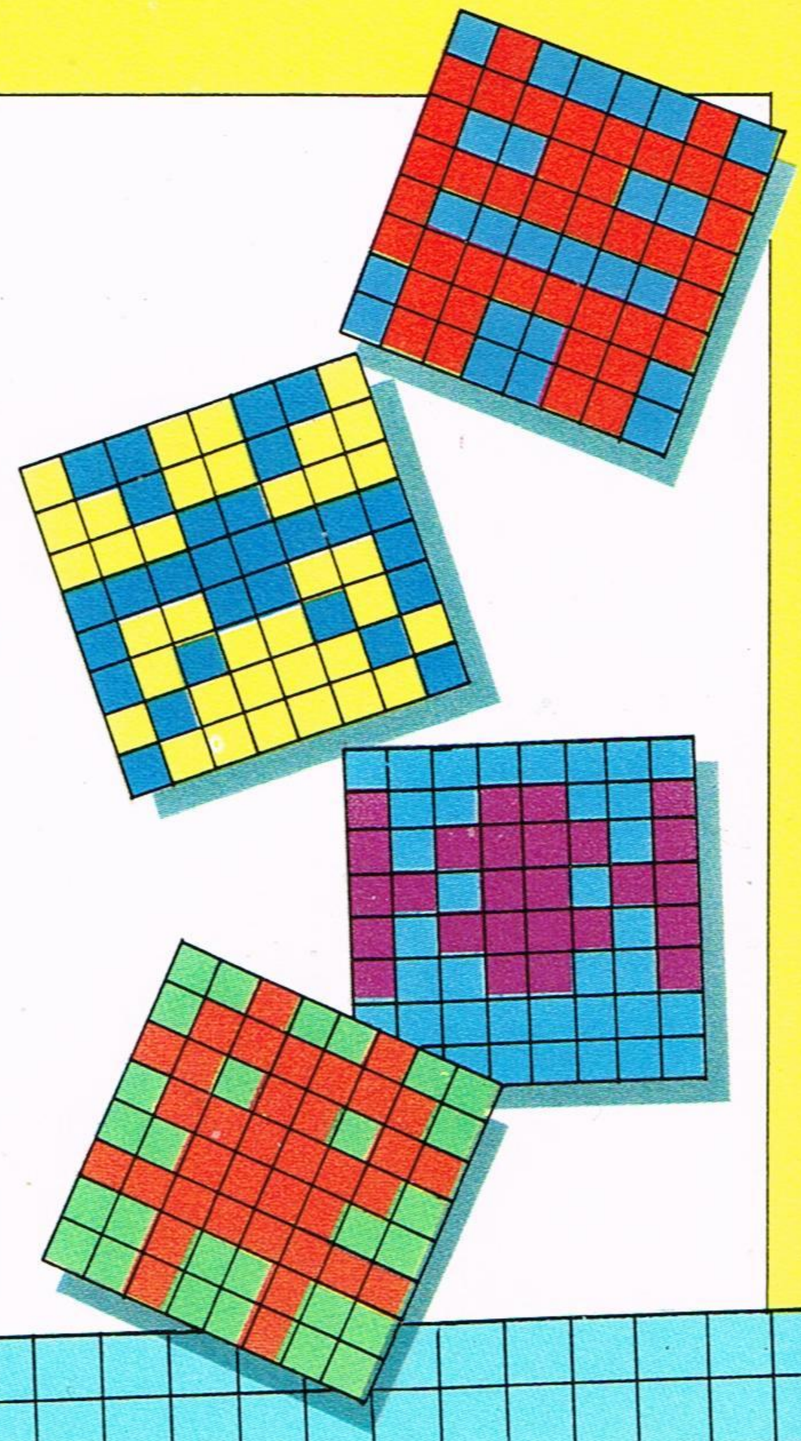
Si tienes un ordenador seguro que alguna vez has jugado a algún tipo de juego de Invasores. En este libro encontrarás un programa que te permitirá crear tu propio juego de "marcianos". El programa se da tanto para el sistema MSX como para el Spectrum.

La efectividad de un juego de invasores se basa en los gráficos móviles. El principio es similar al que se utiliza en los dibujos animados: los caracteres antiguos son borrados de la pantalla antes de imprimirlos en su nueva posición, logrando así la impresión de movimiento. El programa ha sido desglosado en bloques lógicos, y un texto paralelo explica cómo funciona cada sección. Encontrarás, por ejemplo, cómo "sabe" el ordenador cuándo un invasor ha sido alcanzado por uno de tus misiles. El escribir un programa en bloques lógicos no sólo facilita su comprensión tanto al programador como a otras personas; también le hace rodar más rápido y reduce la posibilidad de errores del mismo. Al finalizar alguno de los apartados puedes probar la sección de programa que acabas de teclear (incluso el error más pequeño puede hacer que el programa no funcione). Si algo falla, examina el listado con cuidado. Buscar fallos en un programa es como jugar a un juego de detectives; las pistas están ahí para que las localices.

Contenido


```
 30 GOSUB 1000: REM ENTRADA
 40 GOSUB 2000: REM SELECCION
 50 IF FG<>1 AND FP<>1 THEN GOTO 30
 60 GOSUB 5500: REM FIN
 70 STOP
1000 REM *****ENTRADA*****
1010 VB$="": NO$="": R$=""
1020 INPUT "QUE HAGO AHORA ? ";R$
1030 FOR I=1 TO LEN(R$)
1040 IF MID$(R$,I,1)=" " THEN VB$=LEFT$(R$,3):
     NO$=RIGTH$(R$,I+1): GOSUB 1500: I=LEN(R$)
1050 NEXT I
1060 IF NO$<>"" THEN RETURN
1070 R$=LEFT$(R$,3)
1080 IF R$="NOR" OR R$="SUR"
```

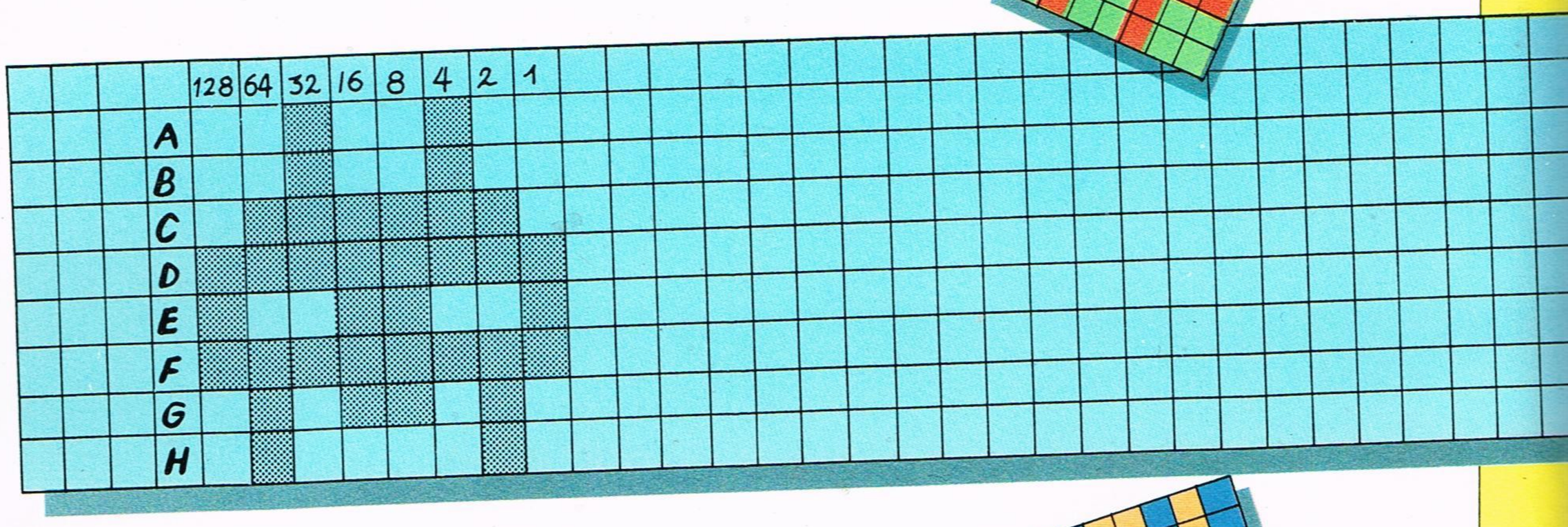
Caracteres definidos por el usuario



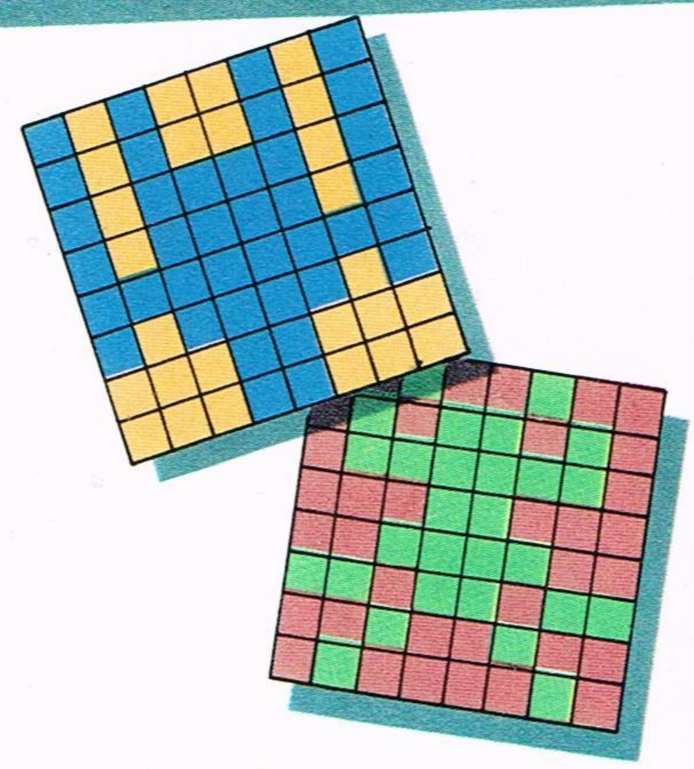
Cada carácter del teclado que pulses se imprime en la pantalla a base de puntos oscuros sobre una matriz de 8 por 8. Los caracteres que forman el teclado están ya definidos en la memoria del ordenador. Sin embargo es posible definir nuestros propios caracteres. ¿Cómo hacerlo?

MSX

En el sistema MSX estos caracteres reciben el nombre de **sprites.** Una de las ventajas de este sistema frente a otros ordenadores es que podemos mover los sprites muy fácilmente por la pantalla, como luego veremos. La base de la definición de sprites es una matriz de 8 por 8 cuadrados sobre la cual sombreamos la figura deseada. El ejemplo que vamos a seguir es el invasor que se empleará en el juego.



Para convertir la matriz en una expresión comprensible por el ordenador toma cada fila, empezando por la A, y suma los números correspondientes a los cuadrados sombreados. Por ejemplo la fila A tiene asociado el número 32+4=36. Para introducir el carácter en el ordenador teclea el programa inferior.

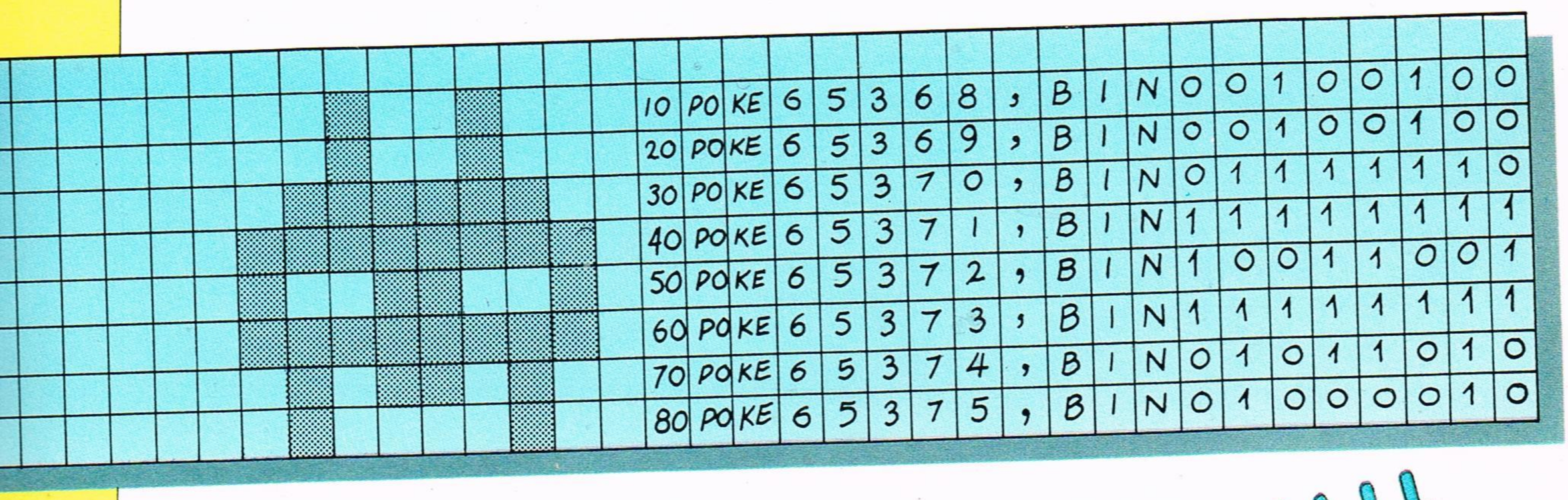


SPRITE (I) es una variable de cadena especial empleada en la definición de sprites. El invasor ha sido asignado al plano de definición 1, según indica el argumento de dicha función. Los datos del programa corresponden a los valores de las filas A a H. En la siguiente sección veremos cómo situar el sprite en la pantalla.

```
10 SCREEN 2,0
20 FOR I=1 TO 8:READ N:A$=A$+CHR$(N):NEXT
30 SPRITE$(1)=A$
40 GOTO 40
80 DATA 36,36,126,255,153,255,90,66
```

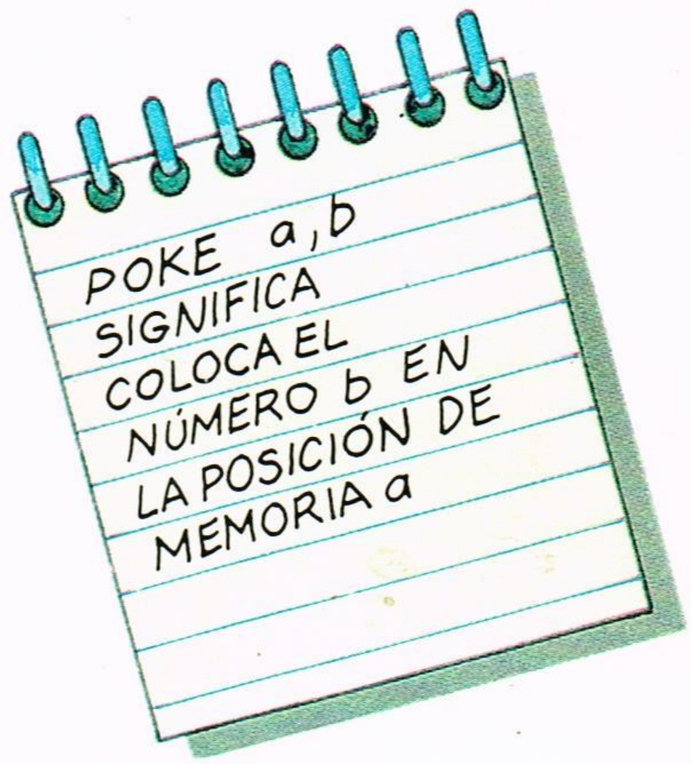
SPECTRUM

En primer lugar diseña sobre el papel el carácter que quieres conseguir mediante una cuadrícula de 8 por 8. Sombrea los cuadros correspondientes para construir una figura; ésta puede tener cualquier forma. El Spectrum te permite describir hasta 21 caracteres definidos por el usuario. El ejemplo que aquí mostramos es el invasor que se empleará en el juego. El ordenador asigna a cada cuadro el valor 0 ó 1. El valor 1 corresponde a un cuadrado sombreado en el gráfico, y 0 corresponde a un cuadrado blanco. Toda la información que el ordenador necesita para construir el carácter debe presentarse de este modo. Cada fila de la cuadrícula que compone el carácter se define mediante una secuencia de ceros y unos o dígitos binarios **(bits)**. Cuando se ruede el programa el ordenador almacenará cada fila como un paquete de ocho bits llamado **byte.** Las sentencias **POKE** alteran la posición de memoria a la que hacen referencia introduciendo el valor indicado. En nuestro caso este valor lo definimos en forma binaria mediante **BIN.**



```
10 POKE 65368,BIN 00100100
20 POKE 65369,BIN 00100100
30 POKE 65370,BIN 01111110
40 POKE 65371,BIN 11111111
50 POKE 65372,BIN 10011001
60 POKE 65373,BIN 11111111
70 POKE 65374,BIN 01011010
80 POKE 65375,BIN 01000010
```

El ordenador interpreta los bytes como números binarios, que son almacenados en una parte especial de la memoria reservada para caracteres definidos por el usuario. Las direcciones correspondientes a estas posiciones de memoria comienzan en la 32600 en el Spectrum de 16 K y en la 65368 en el de 48 K. Para introducir el invasor en el de 16 K modificarás las posiciones 32600 a 32607 ambas inclusive; en el de 48 K e la 65368 a la 65375. Una vez hecho ésto para dibujarlo en el centro de la pantalla teclea:




```
RUN
PRINT AT 10,10;"👾"
```

El 👾 se obtiene poniendo el ordenador en modo gráfico **(CAPS SHIFT y 9)** y pulsando A.

Principios de la animación

MSX

Hemos visto en el apartado anterior cómo definir un **sprite,** ahora veremos cómo situarlo y moverlo por la pantalla. Para localizar sprites emplearemos la siguiente sentencia:

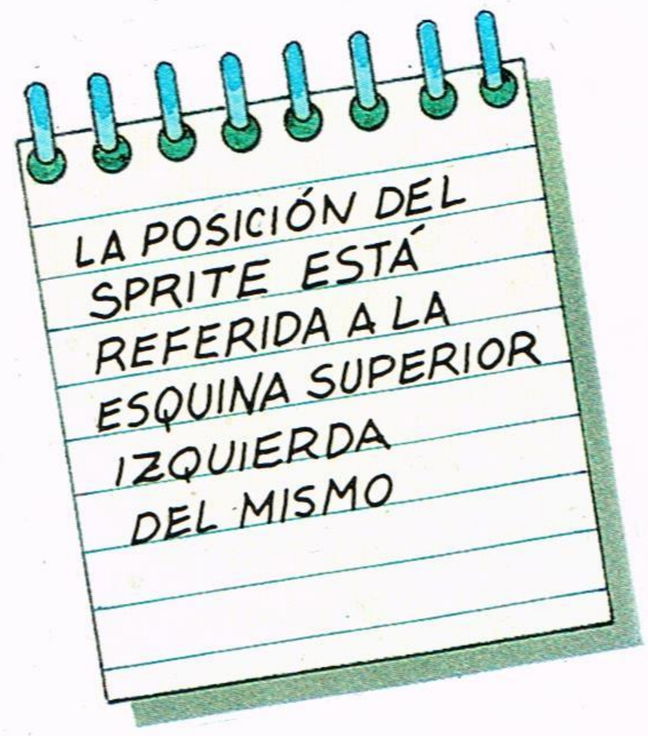
PUT SPRITE nº de plano, (X,Y), color

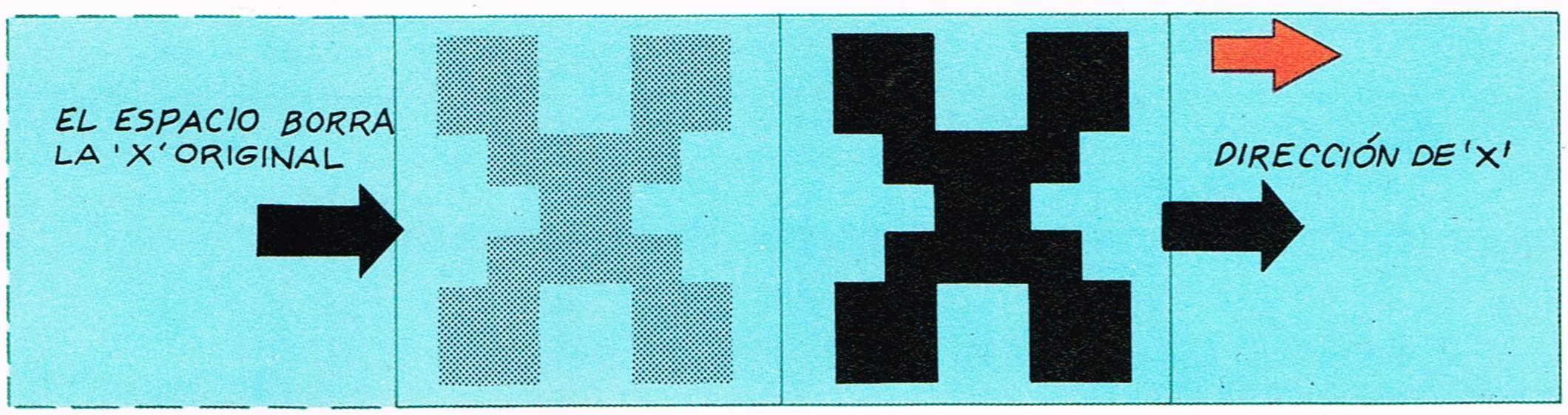
Al asignar un número de plano conseguimos que los sprites de planos inferiores tengan una prioridad mayor que los superiores, de modo que en caso de superponerse las imágenes de los dos planos sólo se verá la del plano de más prioridad. El sprite se localizará en la pantalla donde indiquen las coordenadas (X,Y); estas se refieren a la esquina superior izquierda; sólo hay que destacar que si Y = 209 el sprite correspondiente desaparece de la pantalla. El color debe ser un número del 0 al 15 y selecciona el color del sprite.

Borra la línea 40 y añade al programa anterior las siguientes sentencias:

```
40 FOR I=0 TO 190
50 PUT SPRITE 1,(I,I),1
60 NEXT
70 GOTO 40
```

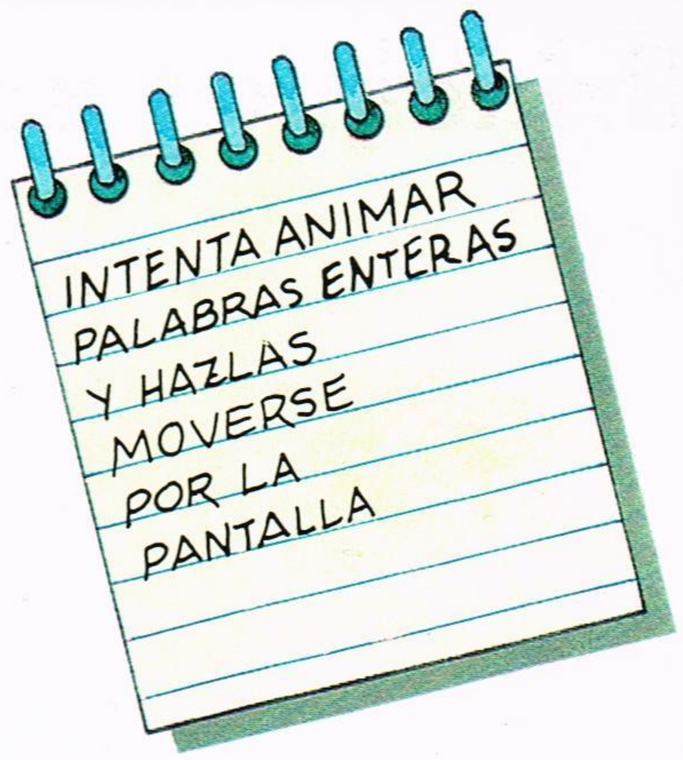
Una de las principales ventajas de los sprites es que se pueden mover por la pantalla muy fácilmente. Para ello sólo hay que modificar las coordenadas X e Y; de esta forma el sprite será eliminado de su posición anterior y colocado en la nueva.
En nuestro ejemplo el bucle entre las líneas 40 y 60 se encarga de alterar estas coordenadas en una unidad, y la línea 50 imprime el sprite en su nueva posición. De esta forma conseguiremos que el sprite se mueva diagonalmente por la pantalla. La línea 70 evita que una vez finalizado el programa regresemos al modo directo.





SPECTRUM

El efecto de movimiento lo conseguiremos imprimiendo una figura en la pantalla, borrándola y volviéndola a imprimir en una posición contigua. Al repetir esta acción la figura parece moverse rápidamente por la pantalla. El programa inferior demuestra este efecto.

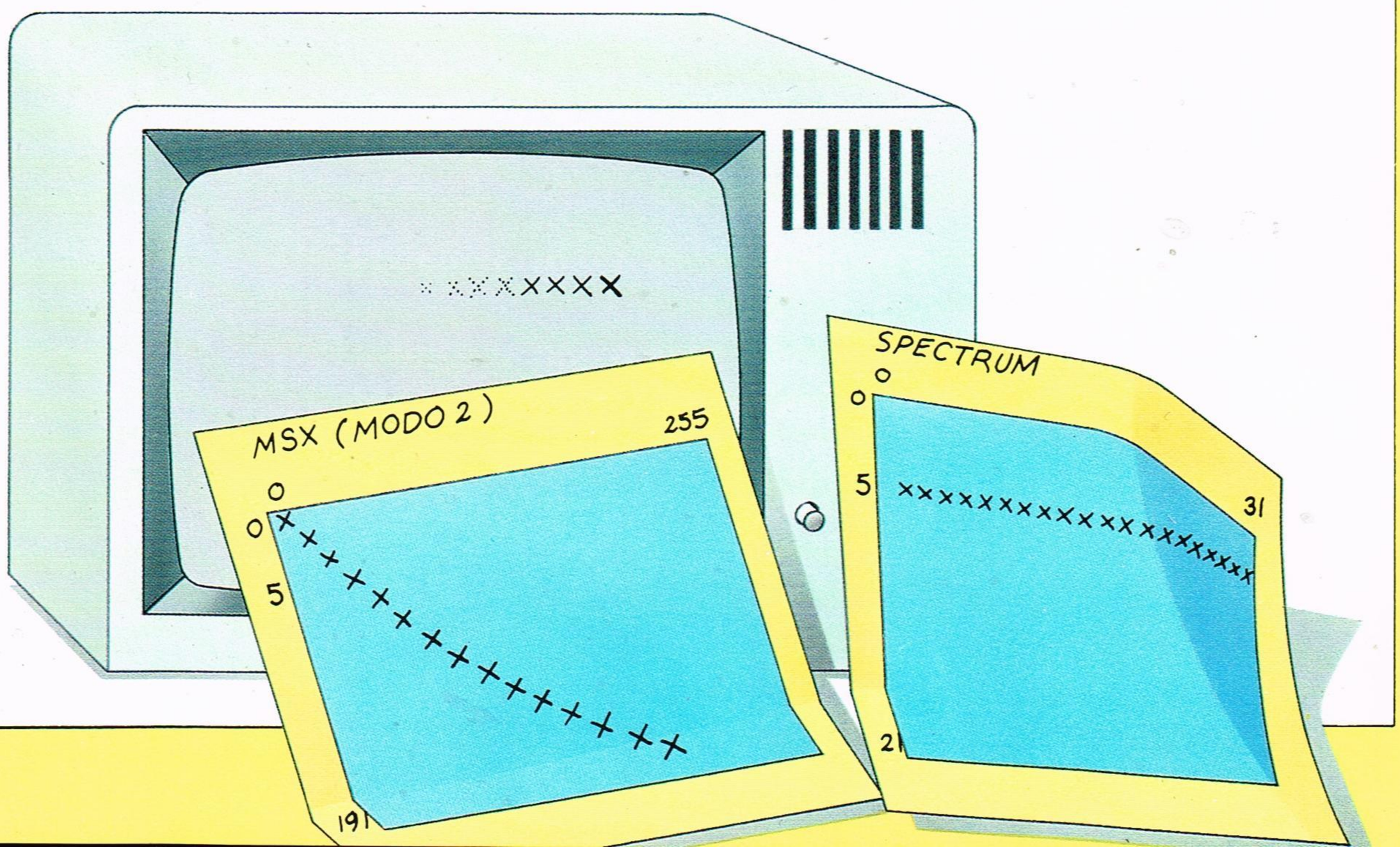


La letra X se mueve por la pantalla mediante el lazo **FOR... NEXT** de variable **I.** Los valores de esta localizan la "X". Cuando la letra se mueve una posición a la derecha los espacios se mueven con ella. El carácter blanco de la izquierda se escribirá sobre la antigua X, borrándola de la pantalla.

```
10 CLS
20 FOR I=0 TO 29
30 PRINT AT 5,I;" X "
40 FOR D=1 TO 50:NEXT D
50 NEXT I
```

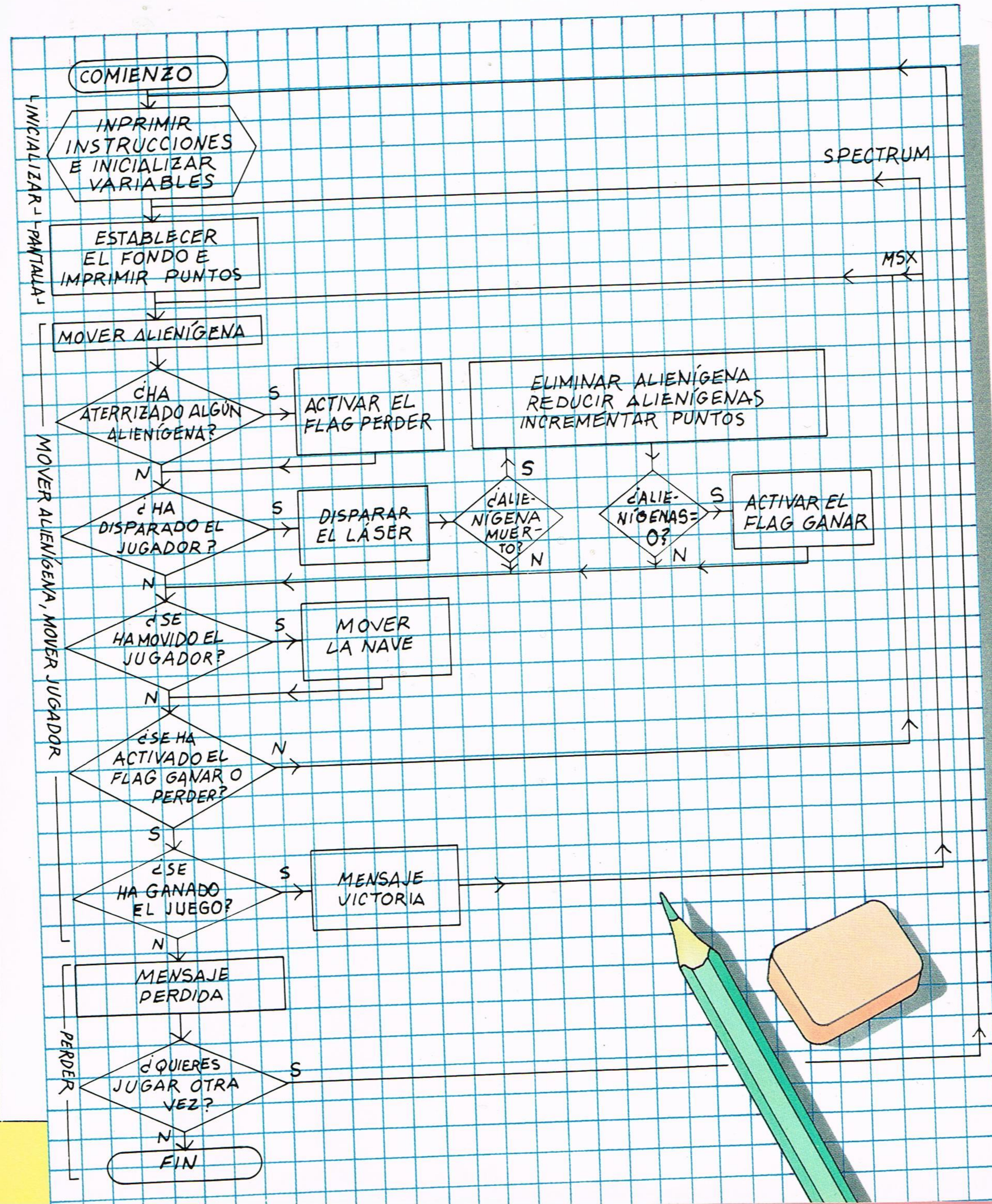
En la línea 40 se introduce un pequeño bucle de retardo; sin él el ordenador ejecutaría el movimiento tan rápido que sería imposible seguirlo.

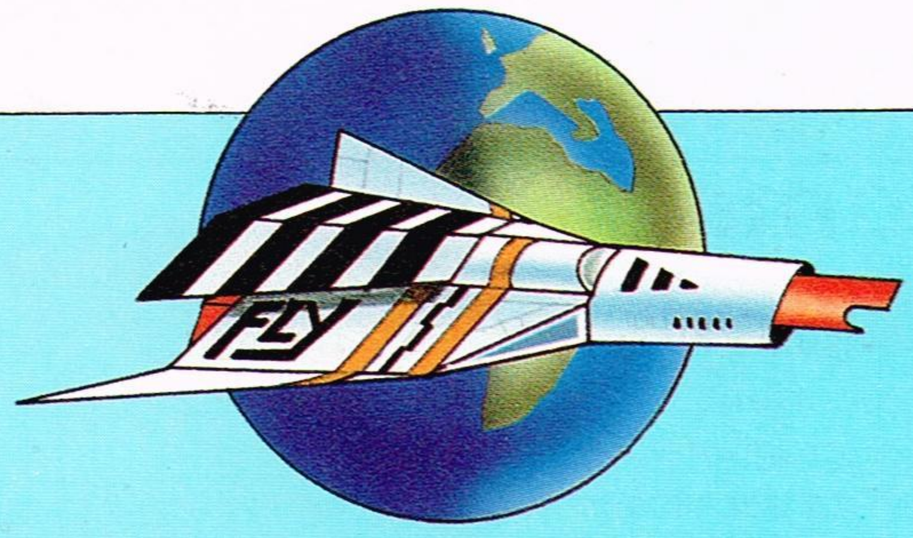
En el juego de los invasores emplearemos los mismos principios de animación que acabas de ver para conseguir que los alienígenas se muevan por la pantalla.



El organigrama

A la hora de planificar el juego es una buena idea hacer un organigrama. Este es el modo oportuno de establecer la secuencia de pasos lógicos que definen tu programa. Consiste en cajas de diferentes formas que representan decisiones, acciones, etcétera y que se conectan entre sí mediante "líneas de flujo". El organigrama se emplea para crear el programa de control. Este es el núcleo del programa, que llama a otras secciones del mismo cuando son necesarias.





EL PROGRAMA DE CONTROL Y LA INICIALIZACION

El programa de control muestra la estructuración del programa, empleando una serie de bloques lógicos llamados subrutinas. Estas están agrupadas en el organigrama para mostrar su secuencia de ejecución. La primera sección del programa se encarga de presentar en pantalla las instrucciones del juego.

Eres un piloto solitario encargado
de proteger el planeta Tierra.
En unos instantes estarás bajo el
ataque de invasores del planeta
Vargón.

Tu misión es evitar que aterricen
las naves vargonianas.

Emplea el cursor para mover tu nave.

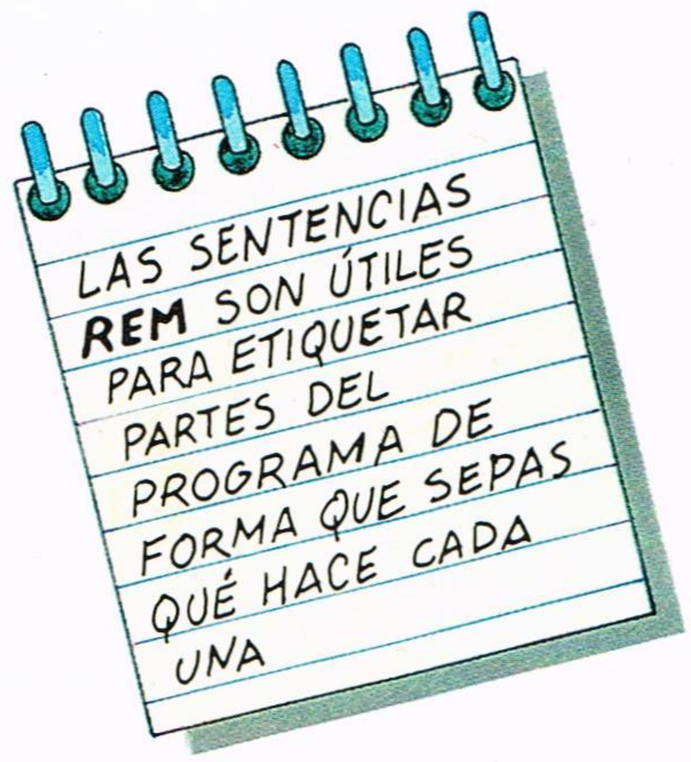
Pulsa la barra espaciadora para
disparar el láser.

Pulsa cualquier tecla para entablar
combate con el enemigo.

```
10   REM Invasores
20   GOSUB 8000:REM Inicializar
30   GOSUB 7000:REM Pantalla
40   GOSUB 1000:REM Alienigena
50   GOSUB 2000:REM Jugador
60   IF (GANAR OR PERDER)=1 THEN GOTO 70 ELSE GOTO 40
70   IF GANAR=1 THEN GOSUB 5000:GOTO 40
80   GOSUB 6000:REM Perder
90   END
```

Generalmente la línea inicial de un programa —10 en nuestro caso— se emplea para identificarle. En la programación de juegos los requisitos iniciales son mostrar en la pantalla las instrucciones e inicializar todas las variables que se emplearán en el juego. Todo ésto lo lleva a cabo la subrutina **Inicializar**, llamada en la línea 20. A continuación ejecutamos la subrutina **Pantalla** que se encarga de dibujar el escenario del fondo compuesto por estrellas y planetas; además, escribe el mensaje que nos indicará el número de alienígenas abatidos.

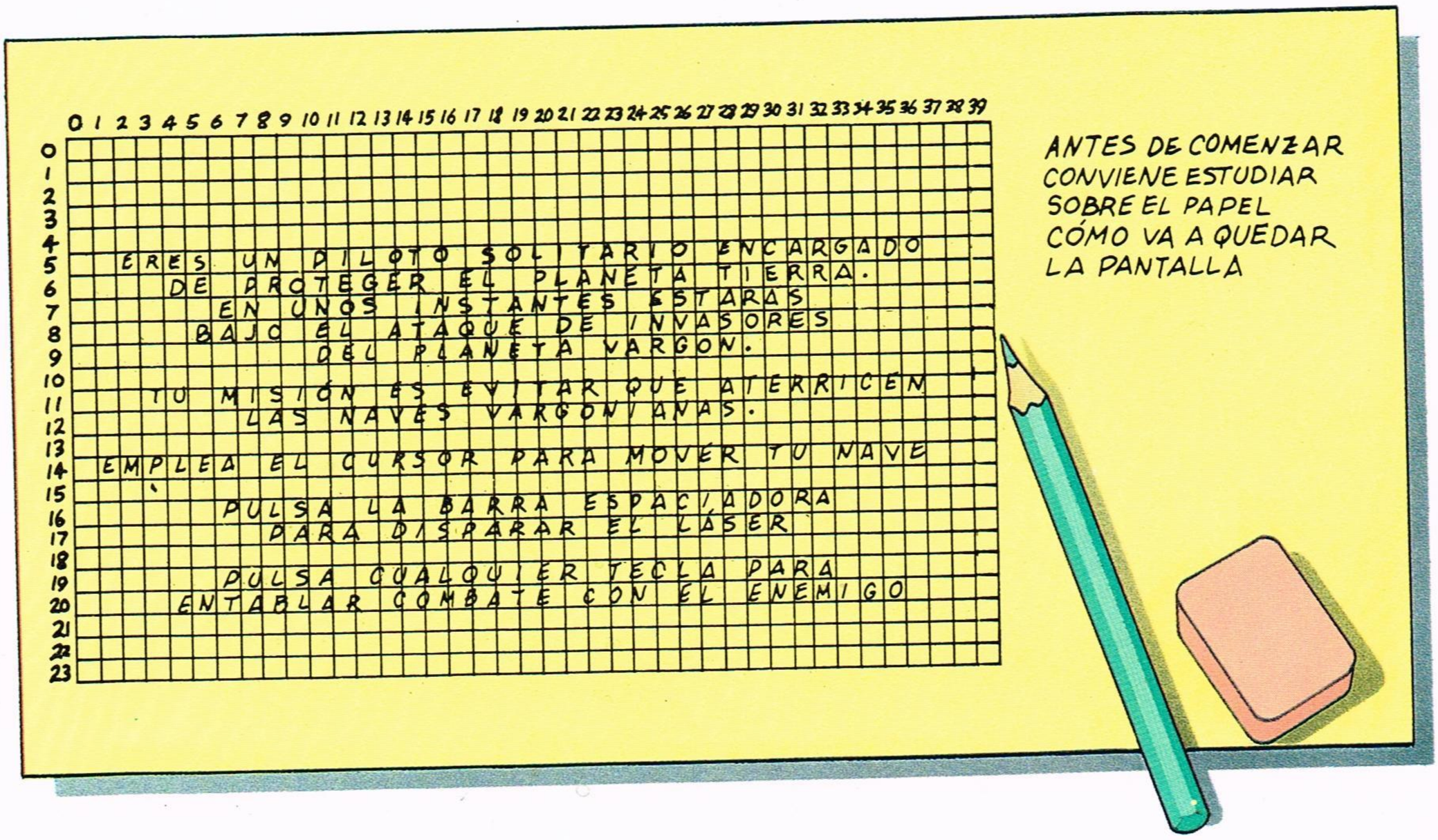
Las líneas 40 a 60 constituyen el lazo principal del programa y contienen las subrutinas más empleadas en el mismo. La subrutina **Alienígena** consigue que cada invasor, elegido aleatoriamente, descienda por la pantalla. La subrutina **Jugador** permite que éste mueva su nave y ataque a los invasores con lásers. La línea 60 estudia si se ha ganado o se ha perdido, comprobando si alguno de los indicadores **GANAR** o **PERDER** están activados, en cuyo caso sale del lazo y salta a la línea 70. Si estos indicadores aún no se han activado volvemos al comienzo del lazo (línea 40).



La línea 70 comprueba si el indicador **GANAR** está a 1, en cuyo caso llama a la subrutina **Ganar,** y tras ejecutarla el programa vuelve a la línea 40 preparado para renovar la pantalla de invasores. Si no se ha ganado el juego se llama a la subrutina **Perder,** que imprime en pantalla un mensaje e invita al jugador a intentarlo de nuevo. El programa finaliza en la línea 90 con la sentencia **END.**

Antes de comenzar conviene estudiar sobre el papel cómo va a quedar la pantalla. En los ordenadores MSX la estructura de la misma depende del modo en que estemos operando. La primera pantalla de presentación opera en modo 0 (modo texto). La pantalla en este modo se compone de 24 líneas de 40 caracteres por línea.

La primera subrutina que llama el programa es **Inicializar** definida a partir de la línea 8000, al final del listado. La razón de ponerla al final es que cada vez que se llama a una subrutina el ordenador estudia el programa desde el principio hasta encontrarla. Como



ANTES DE COMENZAR CONVIENE ESTUDIAR SOBRE EL PAPEL CÓMO VA A QUEDAR LA PANTALLA

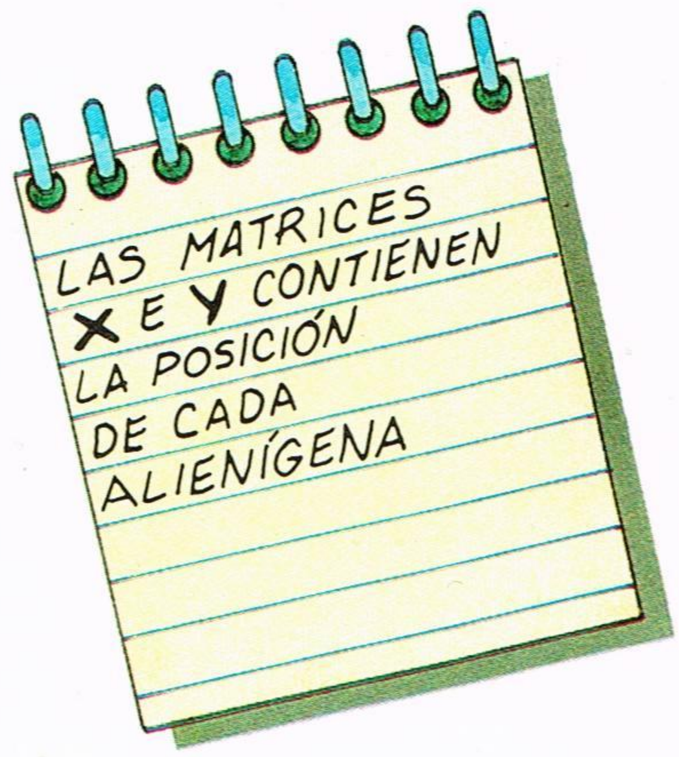
esta subrutina se utiliza una sóla vez conviene ponerla al final, con lo que evitaremos retardos inútiles en la ejecución del programa. Las líneas 8010 a 8130 presentan la primera pantalla; ésta indica al jugador en qué consiste el juego y cómo mover la nave y disparar al enemigo.

La línea 8010 borra la pantalla estableciendo el color del fondo y borde en negro y el texto en blanco. Asimismo borra la línea que indica las teclas de función y desactiva el "click" de las teclas, que sería bastante molesto a la hora de actuar sobre los mandos de la nave.

```
8000 REM Inicializar
8010 SCREEN 0,0,0:COLOR 15,1,1:KEY OFF:CLS
8020 LOCATE 2,5,0:
     PRINT "ERES UN PILOTO SOLITARIO ENCARGADO"
8030 LOCATE 4,6,0:
     PRINT "DE PROTEGER EL PLANETA TIERRA."
8040 LOCATE 6,7,0:PRINT "EN UNOS INSTANTES ESTARAS"
8050 LOCATE 5,8,0:PRINT "BAJO EL ATAQUE DE INVASORES"
8060 LOCATE 9,9,0:PRINT "DEL PLANETA VARGON."
8070 LOCATE 3,11,0:
     PRINT "TU MISION ES EVITAR QUE ATERRICEN"
8080 LOCATE 7,12,0:PRINT "LAS NAVES VARGONIANAS."
8090 LOCATE 1,14,0:
     PRINT "EMPLEA EL CURSOR PARA MOVER TU NAVE"
8100 LOCATE 6,16,0:PRINT "PULSA LA BARRA ESPACIADORA"
8110 LOCATE 8,17,0:PRINT "PARA DISPARAR EL LASER"
8120 LOCATE 6,19,0:PRINT "PULSA CUALQUIER TECLA PARA"
8130 LOCATE 4,20,0:
     PRINT "ENTABLAR COMBATE CON EL ENEMIGO"
```

```
8140 PERDER=0:GANAR=0:MAR=0:XP=120:NIVEL=8:INVAS=10:
     ALIENS=INVAS:COLIS=0:AUX=1
8150 DIM X(INVAS):DIM Y(INVAS)
8160 FOR A=1 TO INVAS
8170 X(A)=24*A:Y(A)=NIVEL
8180 NEXT A
```

Mientras el jugador lee las instrucciones el ordenador puede seguir inicializando las variables empleadas en el juego. Esto se lleva a cabo en las líneas 8140 a 8180. En la primera se establecen además una serie de indicadores **(flags);** son variables que pueden ser activadas o desactivadas para indicar si una condición es verdadera o falsa. Así **PERDER** y **GANAR** se desactivan pues aún no se ha empezado a jugar. **COLIS** y **AUX** nos serán útiles cuando estudiemos si un misil ha colisionado con un alienígena. **MAR** será el marcador del juego e indicará el número de invasores destruidos. **XP** es la posición de la nave sobre el eje **X** de la pantalla y **NIVEL** la posición vertical desde la que comenzarán a descender los alienígenas. El número de ellos al comienzo del juego se establece mediante la variable **INVAS**, que permanece invariante a lo largo del juego. Se necesita una segunda variable **ALIENS** que contenga el número de alienígenas que quedan en cualquier momento del juego.



La línea 8150 establece dos matrices **X** e **Y** en las que guardaremos la posición horizontal y vertical de cada alienígena —de ahí que se dimensionen al valor de **INVAS**. El bucle **FOR... NEXT** establece las posiciones iniciales de cada invasor.

Las instrucciones siguientes definen la forma de los tres sprites empleados en el juego: el invasor, la nave y el misil. El procedimiento es el mismo seguido en la introducción (hemos reproducido también los datos necesarios para cada sprite). La sentencia 8230 es un recurso típico para detener la ejecución de un programa hasta que no se pulse una tecla.

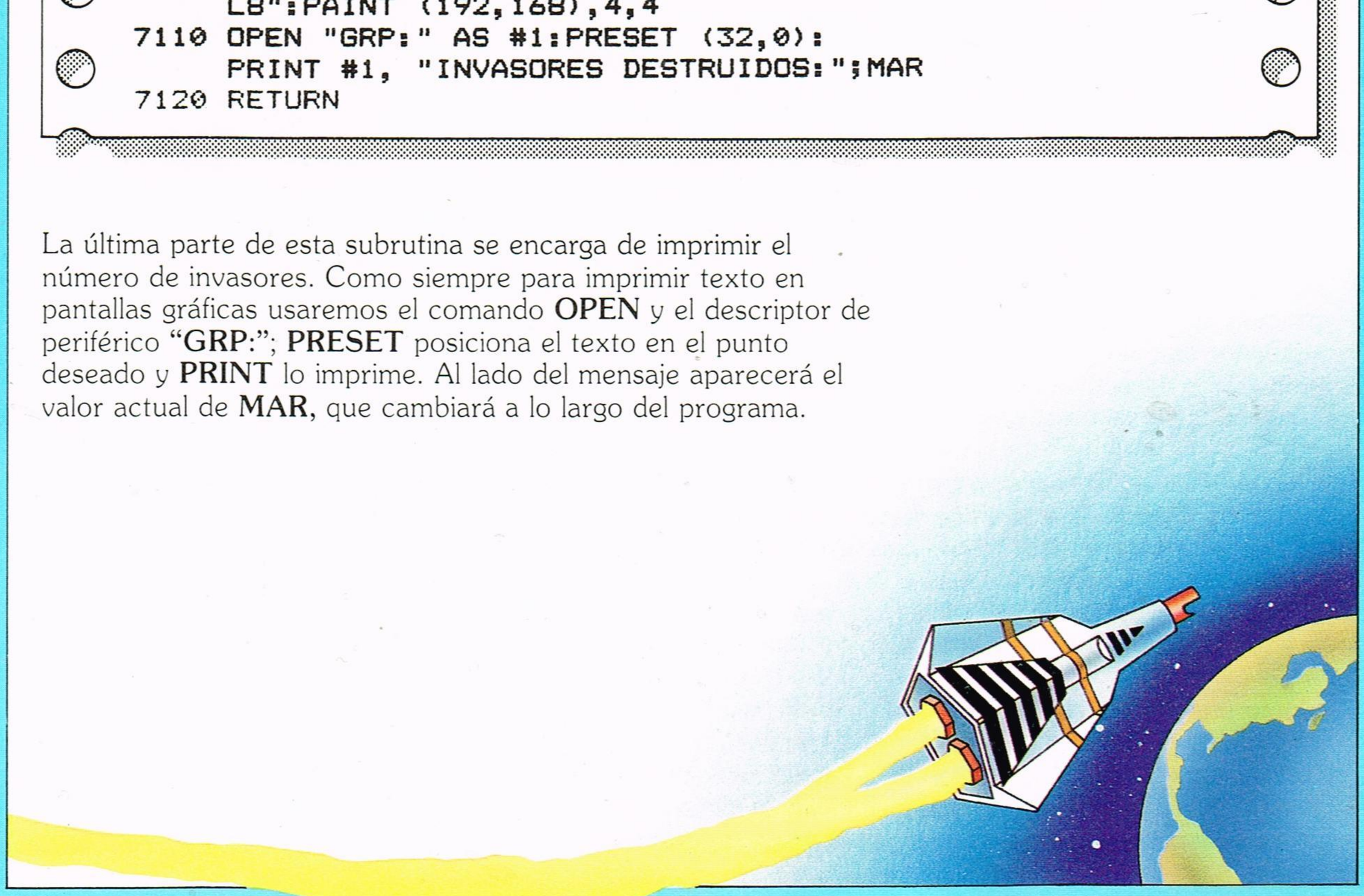
```
8190 FOR I=1 TO 8:READ N:M$=M$+CHR$(N):NEXT:REM Misil
8200 FOR I=1 TO 8:READ N:N$=N$+CHR$(N):NEXT:REM Nave
8210 FOR I=1 TO 8:READ N:A$=A$+CHR$(N):NEXT:REM Invasor
8220 ON SPRITE GOSUB 4000:SPRITE OFF
8230 Q$=INPUT$(1)
8240 RETURN
9000 REM Datos sprites
9010 DATA 16,16,56,16,16,16,56,108:REM Misil
9020 DATA 16,56,16,186,186,186,238,108:REM Nave
9030 DATA 36,36,126,255,153,255,90,66:REM Invasor
```

La siguiente subrutina que llamaremos se encarga de establecer el fondo de la pantalla. En primer lugar pasa el ordenador al modo 2 (modo gráfico) en el que tendremos una resolución de 256 por 192 "pixels" —puntos en la pantalla—, y establece el color del fondo y borde a negro y el de texto a blanco. Tras borrar la pantalla la línea 7020 establece los sprites correspondientes a la nave y el misil en los planos 11 y 12. La línea 7030 asigna a los planos 1 a 10 sprites con la forma del invasor.

El bucle 7040 a 7060 dibuja 60 estrella colocadas en puntos aleatorios de la pantalla y con uno de los colores disponibles —elegido también de forma aleatoria. Las líneas 7070 a 7100 se encargan de dibujar el resto de la pantalla.

```
7000 REM Pantalla
7010 SCREEN 2,0,0:COLOR 15,1,1:CLS
7020 SPRITE$(12)=N$:SPRITE$(11)=M$
7030 FOR I=1 TO 10:SPRITE$(I)=A$:NEXT
7040 FOR S=1 TO 60
7050 X=INT(RND(5)*250):Y=INT(RND(5)*190):
     PSET (X,Y),RND(1)*15
7060 NEXT S
7070 CIRCLE (52,28),16,9:CIRCLE (44,28),16,9,4.95673,
     1.32645:PAINT (44,28),9
7080 CIRCLE (192,42),25,7:CIRCLE (180,42),24,7,4.95673,
     1.32645:PAINT (180,42),7
7090 CIRCLE (0,192),60,10,0,-3.1459/2:
     LINE (0,192)-(60,192),10:PAINT (5,190),10,10
7100 PRESET (200,144),4:DRAW"G32F16E32H16R8F16L8BR8G32
     L8":PAINT (192,168),4,4
7110 OPEN "GRP:" AS #1:PRESET (32,0):
     PRINT #1, "INVASORES DESTRUIDOS:";MAR
7120 RETURN
```

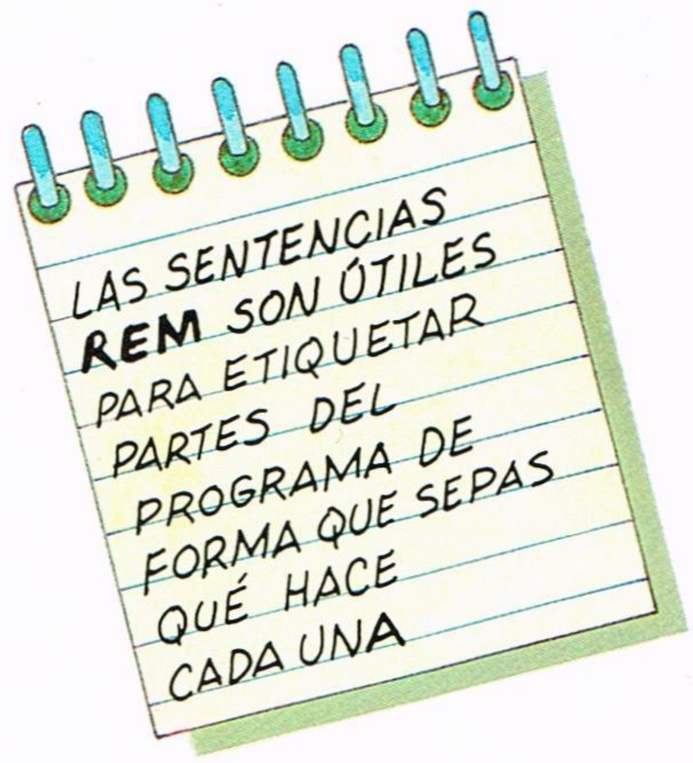
La última parte de esta subrutina se encarga de imprimir el número de invasores. Como siempre para imprimir texto en pantallas gráficas usaremos el comando **OPEN** y el descriptor de periférico "**GRP:**"; **PRESET** posiciona el texto en el punto deseado y **PRINT** lo imprime. Al lado del mensaje aparecerá el valor actual de **MAR**, que cambiará a lo largo del programa.



El programa de control llama a todas las subrutinas empleadas en el juego siguiendo un orden lógico. Cada subrutina es llamada mediante la instrucción **GO SUB**, seguida del número de línea correspondiente. Las declaraciones **REM** se emplean para recordar qué es lo que hace cada subrutina.

```
  5 REM Invasores
 10 GO SUB 8000: REM Inicializar
 20 GO SUB 7000: REM Pantalla
 30 GO SUB 1000: REM Alienigena
 40 GO SUB 2000: REM Jugador
 50 IF (ganar  OR perder)=1 THEN  GO TO 70
 60 GO TO 30
 70 IF ganar=1 THEN  GO SUB 5000: GO TO 20
 80 GO SUB 6000: REM Perder
 90 CLS : PAPER 7: BORDER 7: INK 0: STOP
```

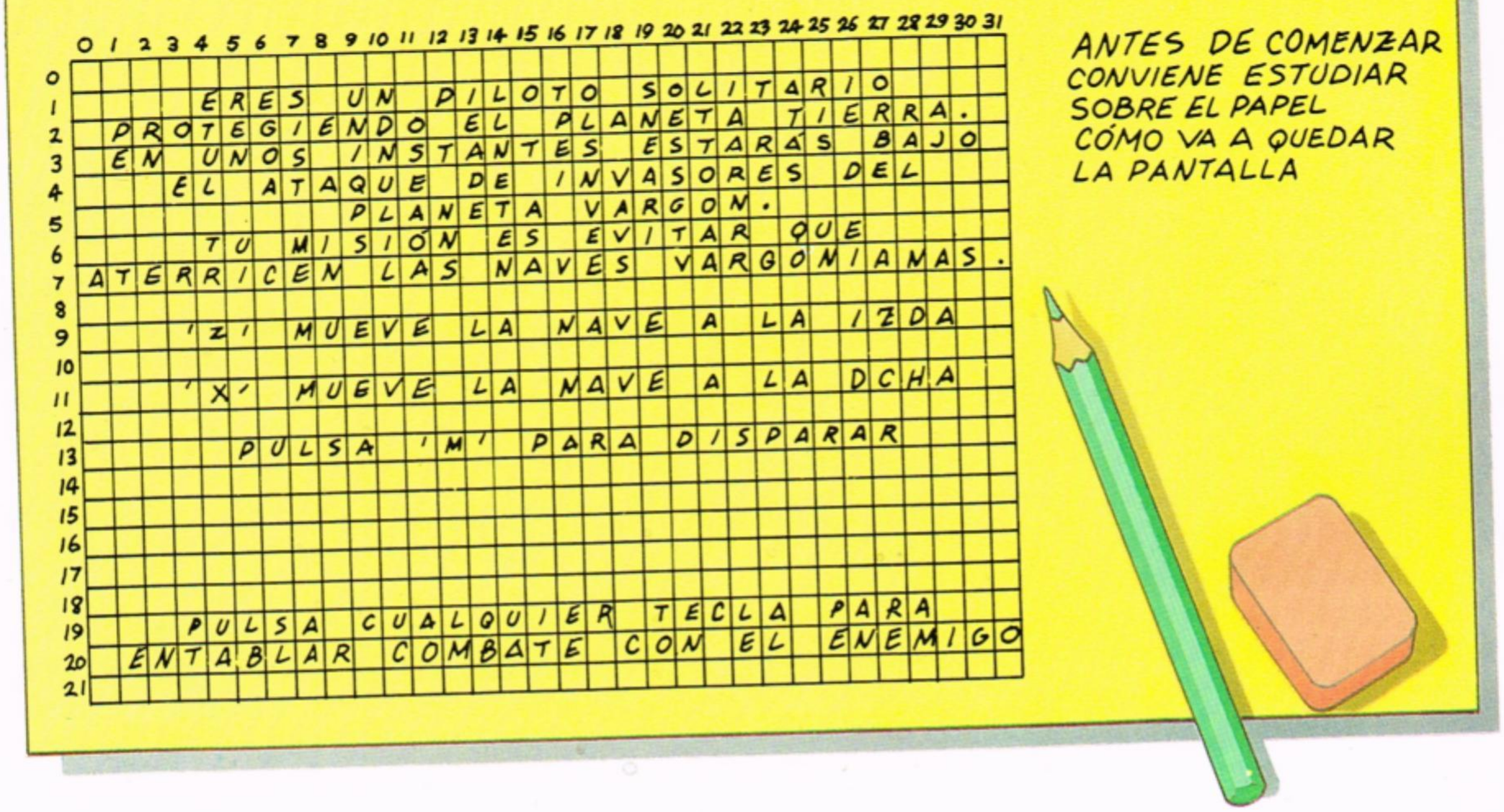
Los requisitos iniciales en la programación de juegos consisten en presentar las instrucciones en la pantalla e inicializar las variables con los valores oportunos. La línea 5 da título al programa y la sentencia **GO SUB** de la línea 10 bifurca el mismo para que ejecute la subrutina de inicialización que comienza en la línea 8000. La subrutina llamada en la línea 20 es la parte del programa que se encarga de dibujar las estrellas y escribir el número de alienígenas destruidos.



Las líneas 30 a 60 constituyen un lazo contínuo y se pueden considerar el corazón del programa. La subrutina llamada en la línea 30 se encarga de avanzar los invasores en la pantalla. La línea 40 llama a una subrutina que permite al jugador moverse y disparar un misil a los invasores. La siguiente línea es una sentencia **IF... THEN** que estudia si se ha ganado o perdido. Si se cumple una de estas dos condiciones el programa sale del lazo y salta a la línea 70. De lo contrario la línea 60 hace que se repita el proceso de movimiento y disparo regresando a la línea 30. **ganar** y **perder** son simplemente indicadores —flags— que pueden activarse o desactivarse —1 ó 0— cuando se verifiquen ciertas condiciones durante el juego.

La línea 70 estudia si el flag **ganar** está a 1. Si lo está llama a la subrutina **Ganar.** Cuando eśta finaliza el ordenador devuelve el control al programa y salta a la línea 20 para renovar la pantalla de invasores. Si en cambio está a 0, el programa llama a la subrutina **Perder** en la línea 80. El programa de control finaliza en la línea 90 con la instrucción **STOP.**

Como puedes ver el programa principal es bastante corto y su efectividad se basa en el empleo de subrutinas encargadas de labores específicas. De esta forma la estructura del programa es mucho más clara y permite comprobar con sencillez cualquier tipo de fallo, así como introducir modificaciones en el mismo.



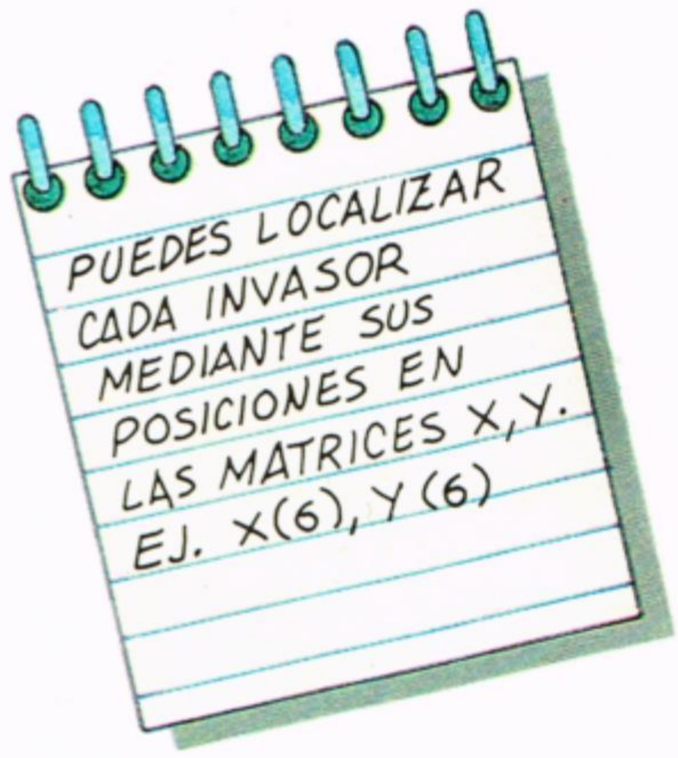
Antes de comenzar conviene plantear la pantalla sobre el papel. Esta se compone en el Spectrum de 22 líneas de 32 caracteres cada una; las columnas están numeradas de 0 a 31 hacia la derecha y las filas de 0 a 21 hacia abajo.

La primera subrutina que llama el programa es **Inicializar** —línea 10—. Cada vez que se llama a una subrutina el ordenador comprueba el listado completo desde el principio hasta encontrar el número de línea requerido. Si una subrutina es llamada una sola vez, como es el caso, conviene definirla cerca del fin del programa, en esta ocasión en la línea 8000. Las líneas 8010 a 8125 dibujan la primera pantalla, encargada de mostrar las instrucciones. Las teclas Z y X permiten al jugador mover la nave, y pulsando M se dispara un misil. Las instrucciones necesarias para conseguirlo se definirán en una sección posterior.

```
8000 REM Inicializar
8010 PAPER 0: INK 7: BORDER 0: CLS
8020 PRINT AT 1,4;"Eres un piloto solitario"
8030 PRINT AT 2,1;"protegiendo el planeta Tierra."
8040 PRINT AT 3,1;"En unos instantes estaras bajo"
8050 PRINT AT 4,3;"el ataque de invasores del"
8060 PRINT AT 5,9;"planeta VARGON."
8070 PRINT AT 6,4;"Tu mision es evitar que"
8080 PRINT AT 7,0;"aterricen las naves vargonianas."
8090 PRINT AT 10,2;"'Z'  MUEVE LA NAVE A LA IZDA"
8100 PRINT AT 12,2;"'X'  MUEVE LA NAVE A LA DCHA"
8110 PRINT AT 14,4;"PULSA 'M' PARA DISPARAR"
8120 PRINT AT 20,2;"Pulsa cualquier tecla para"
8125 PRINT AT 21,0;"entablar combate con el enemigo"
```

```
8130 LET perder=0: LET ganar=0: LET puntos=0:
     LET colalien=0: LET colest=0: LET xp=16:
     LET altura=1: LET invasores=10:
     LET alienigenas=invasores
8140 DIM x(invasores): DIM y(invasores)
8150 FOR a=1 TO invasores
8160 LET x(a)=a*3: LET y(a)=altura
8170 NEXT a
```

Mientras el jugador lee las instrucciones, el ordenador puede continuar inicializando las variables empleadas en el juego. En la línea 8130 se establecen una serie de indicadores —flags—. Un flag puede ser "activado" —tomando el valor 1— o "desactivado" —tomando el valor 0—. Los flags **perder** y **ganar** se desactivan, pues el juego acaba de empezar. **puntos** representa el número de alienígenas abatidos. **colalien** y **colest** son dos flags que detectan si un misil ha colisionado con un alienígena o una estrella. **xp** es la posición de la nave del jugador sobre el eje X de la pantalla. **altura** es la posición vertical inicial de cada invasor respecto a la parte superior de la pantalla. El número de alienígenas se establece mediante la variable **invasores.** La variable **alienígenas** contendrá el número de invasores por derribar.



Cada invasor lleva asociado un par de valores incluidos en sendas matrices unidimensionales. Así la instrucción **DIM x (invasores)** reserva espacio en memoria para la posición horizontal de cada invasor, y **DIM y (invasores)** hace lo propio con la posición vertical. Un bucle **FOR... NEXT** establece la posición horizontal inicial de cada uno de los diez invasores a un valor que va de 3 a 30 incrementándose de 3 en 3 para abarcar toda la pantalla.

El siguiente grupo de instrucciones define la forma del invasor. El programa es el creado en la página 9, en la sección de caracteres definidos por el usuario, y constituye las líneas 8180 a 8250.

```
8180 POKE 65368,BIN 00100100
8190 POKE 65369,BIN 00100100
8200 POKE 65370,BIN 01111110
8210 POKE 65371,BIN 11111111
8220 POKE 65372,BIN 10011001
8230 POKE 65373,BIN 11111111
8240 POKE 65374,BIN 01011010
8250 POKE 65275,BIN 01000010
8260 LET R$=INKEY$: IF R$="" THEN  GO TO 8260
8270 RETURN
```

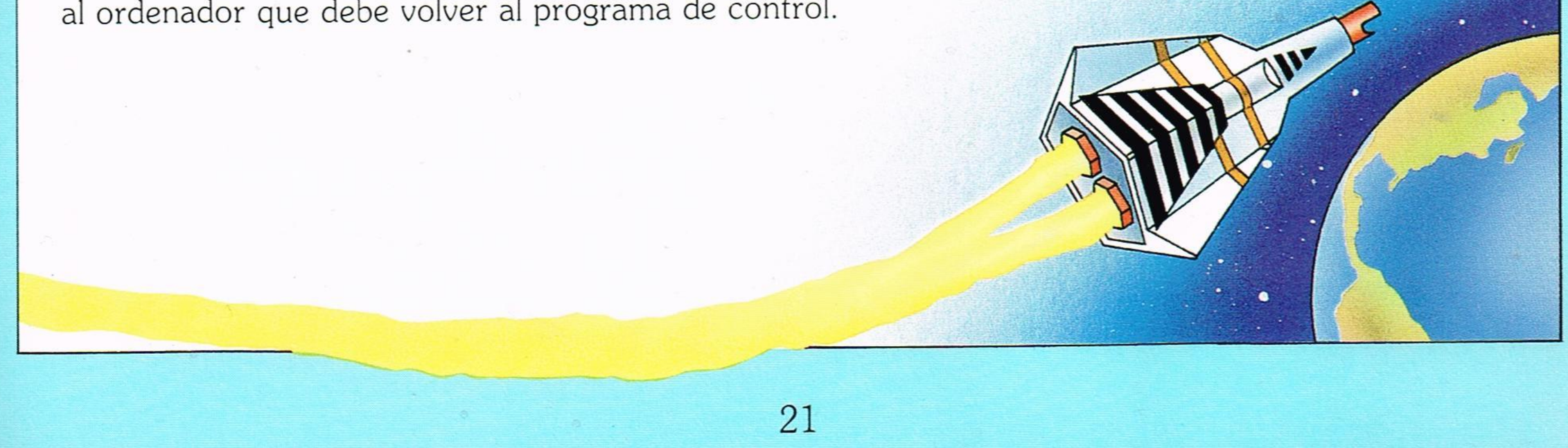
En la línea 8260 **INKEY** explora el teclado para detectar la tecla pulsada, almacenando el resultado en **R$**; si no contiene nada **(R$=" ")** no se ha pulsado ninguna. El ordenador repite el proceso hasta que se da un valor a la variable, en cuyo caso retorna al programa de control. Este sistema permite al jugador comenzar el juego con sólo pulsar una tecla del teclado.

La siguiente subrutina que llamaremos dibuja el fondo de la pantalla. Esta es la segunda subrutina menos usada en el programa, de ahí que se defina la penúltima en el listado, comenzando en la línea 7000. Se encarga de dibujar un fondo de estrellas en puntos aleatorios cada vez que comienza el juego o cuando avanza la siguiente oleada de alienígenas una vez eliminada la anterior.

En primer lugar se borran de la pantalla los textos o gráficos que hayan podido dejar las subrutinas anteriores, y un bucle **FOR... NEXT** dibuja aleatoriamente una serie de cuarenta estrellas. En la línea 7030 la función **RND** selecciona una posición de carácter de forma aleatoria y dibuja en ella una estrella —un único "pixel"— en la misma posición para cada cuadrado. Más tarde, cuando se dispare un misil y éste ocupe la misma posición de carácter que la estrella, la borrará. Así pues es importante conocer la posición de la misma de modo que pueda ser dibujada de nuevo cuando el misil haya pasado. La recuperación de las estrellas perdidas se lleva a cabo en otra sección del programa. Los alienígenas en su avance también borran estrellas, lo que sin embargo no se tiene en cuenta, ya que su efecto no llama tanto la atención, y el proceso de "recuperación de estrellas" retarda bastante el programa.

```
7000 REM Pantalla
7010 CLS
7020 FOR s=1 TO 40
7030 PLOT 8*INT (RND*32),8*INT (RND*21)+10
7040 NEXT s
7050 INK 7: PRINT AT 0,5;"INVASORES DESTRUIDOS=";puntos
7060 RETURN
```

La última parte de esta subrutina imprime la puntuación en la mitad de la línea superior de la pantalla. El valor de **puntos** cambia en el programa según se destruyen alienígenas. Como antes, la subrutina finaliza con una sentencia **RETURN** que indica al ordenador que debe volver al programa de control.



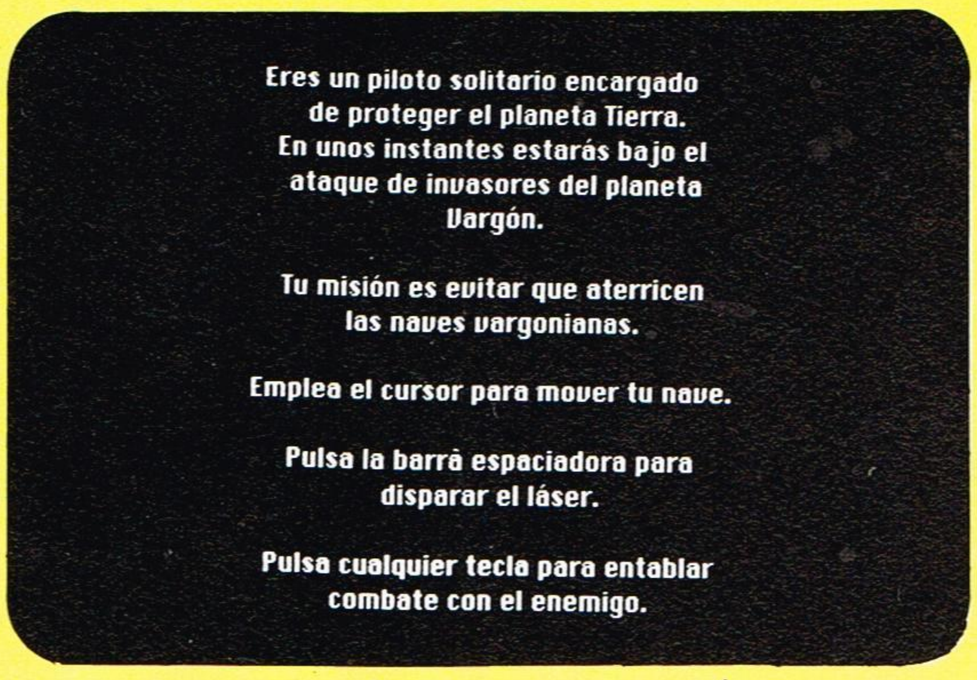
Prueba tu programa

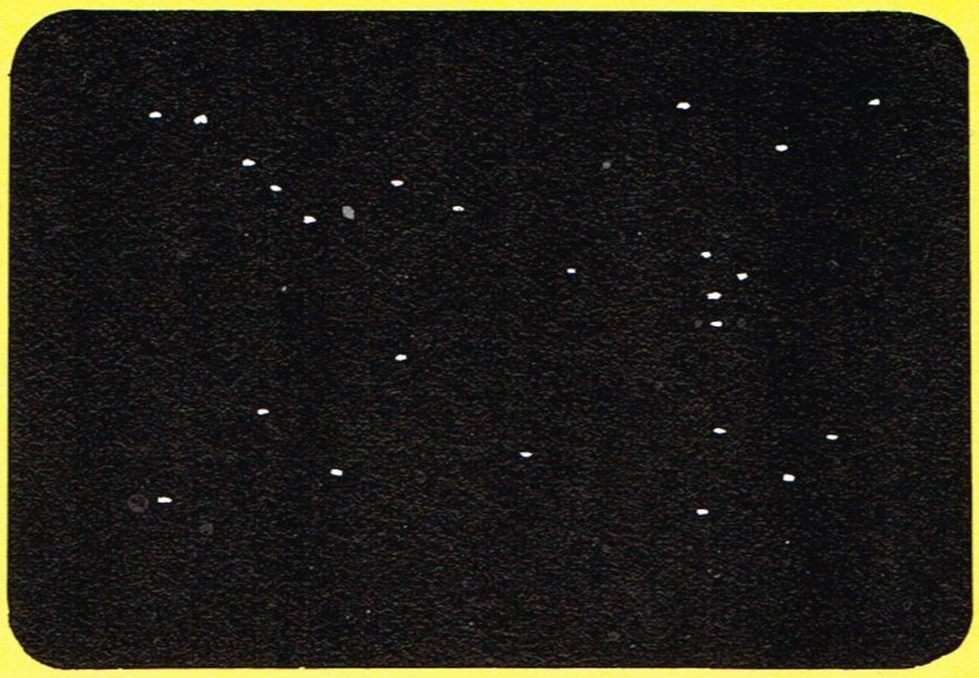
Puedes probar ahora la parte del programa que llevas escrita. Sigue las instrucciones que se indican abajo y obsevarás la pantalla de presentación. Al pulsar cualquier tecla obtendrás la segunda pantalla. Si el programa no funciona, o si ocurre algo raro revisa lo que has hecho con cuidado. Incluso el más pequeño error puede provocar que el programa no funcione.

MSX

1. TECLEA LA LÍNEA PROVISIONAL **35 GOTO 35** Y PULSA [RETURN]
2. TECLEA **RUN** Y PULSA [RETURN]

DEBERÍAS VER AHORA LA PANTALLA DE PRESENTACIÓN QUE SE MUESTRA A LA IZQUIERDA. AL PULSAR CUALQUIER TECLA EL PROGRAMA CONTINUARÁ Y SE ESTABLECERÁ EL FONDO DE ESTRELLAS Y PLANETAS DE LA DERECHA. BORRA LA LÍNEA PROVISIONAL ANTES DE CONTINUAR CON LA SIGUIENTE SECCIÓN DEL TEXTO

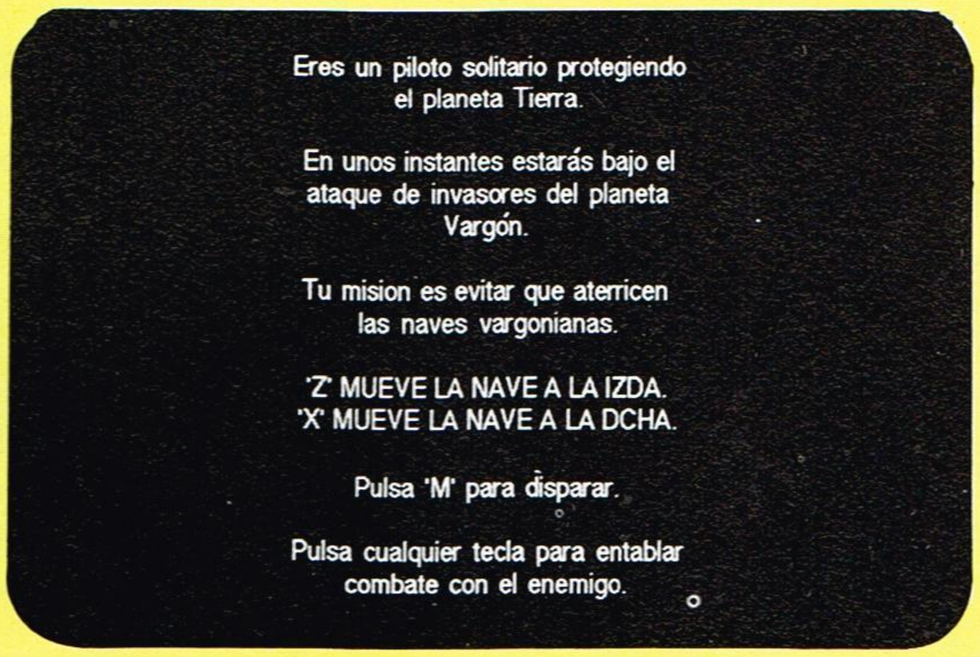


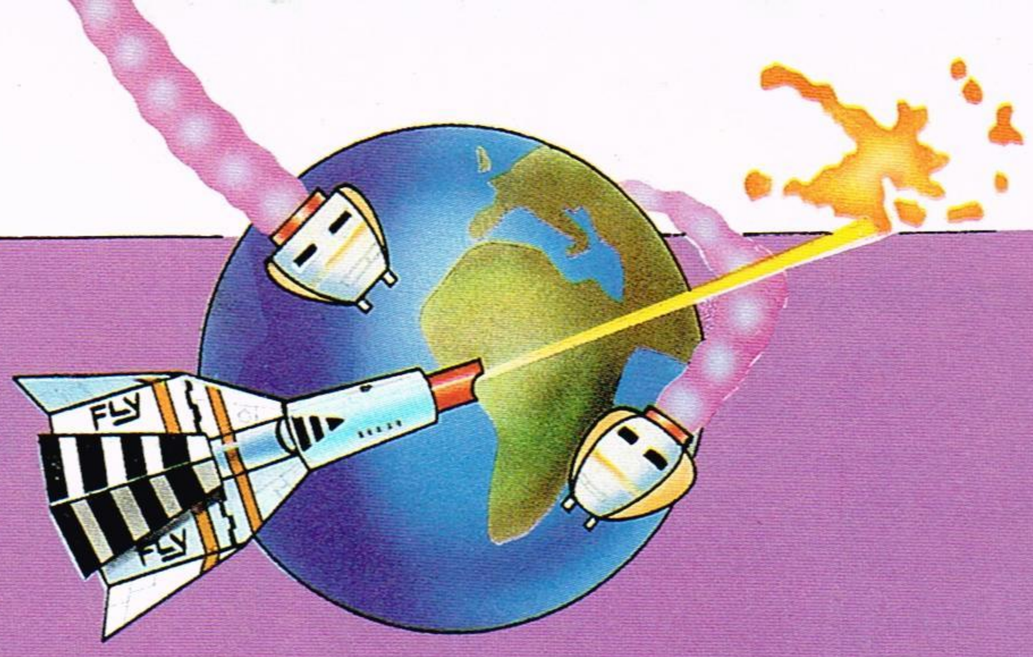


SPECTRUM

1. INTRODUCE LA LÍNEA PROVISIONAL **25 STOP** Y PULSA [ENTER]
2. PULSA [RUN] Y LUEGO [ENTER]

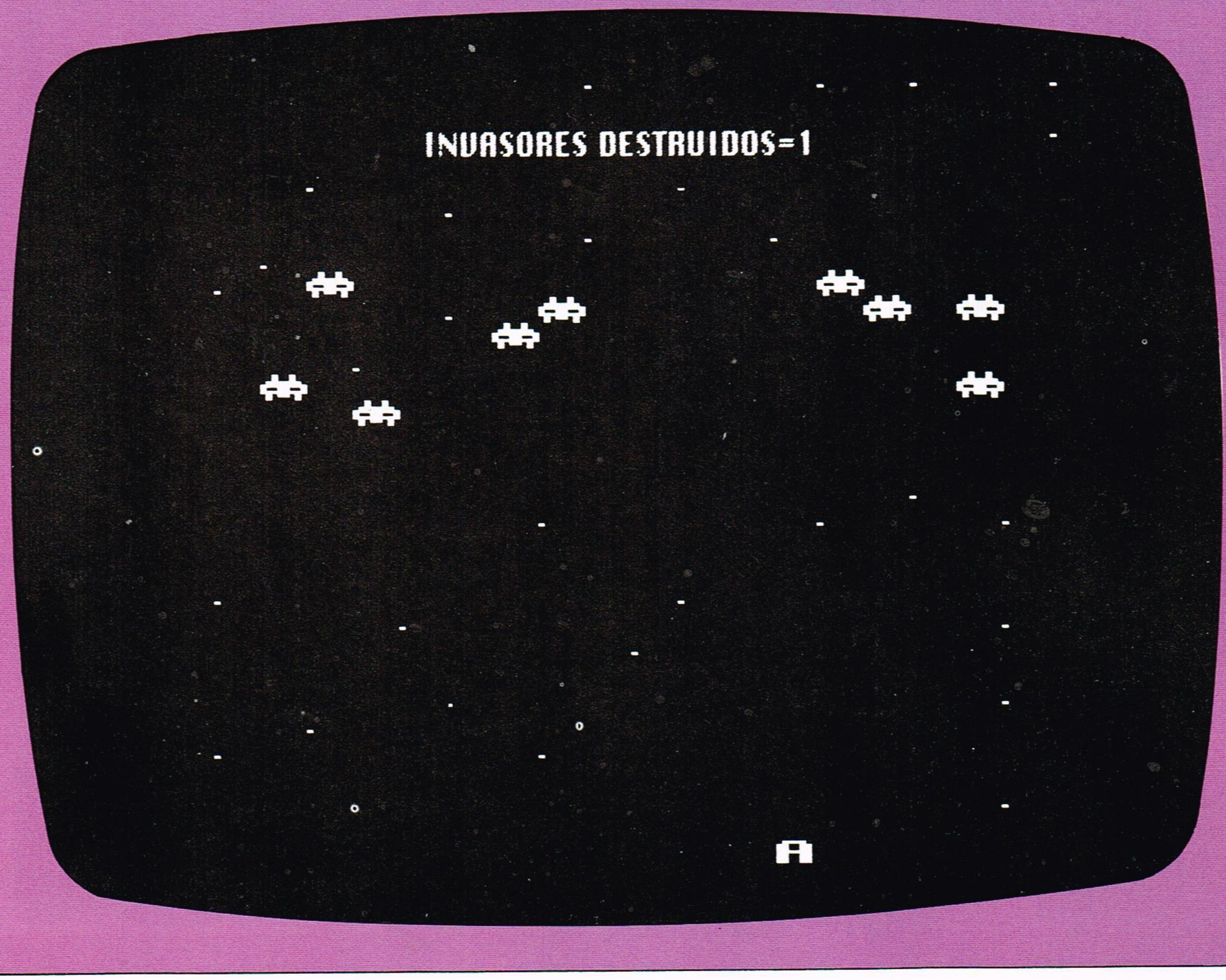
DEBERÍAS VER AHORA LA PANTALLA DE PRESENTACIÓN QUE SE MUESTRA A LA IZQUIERDA. AL PULSAR CUALQUIER TECLA EL PROGRAMA CONTINUARÁ Y SE ESTABLECERÁ EL FONDO DE ESTRELLAS Y PLANETAS DE LA DERECHA. BORRA LA LÍNEA PROVISIONAL ANTES DE CONTINUAR CON LA SIGUIENTE SECCIÓN DEL TEXTO



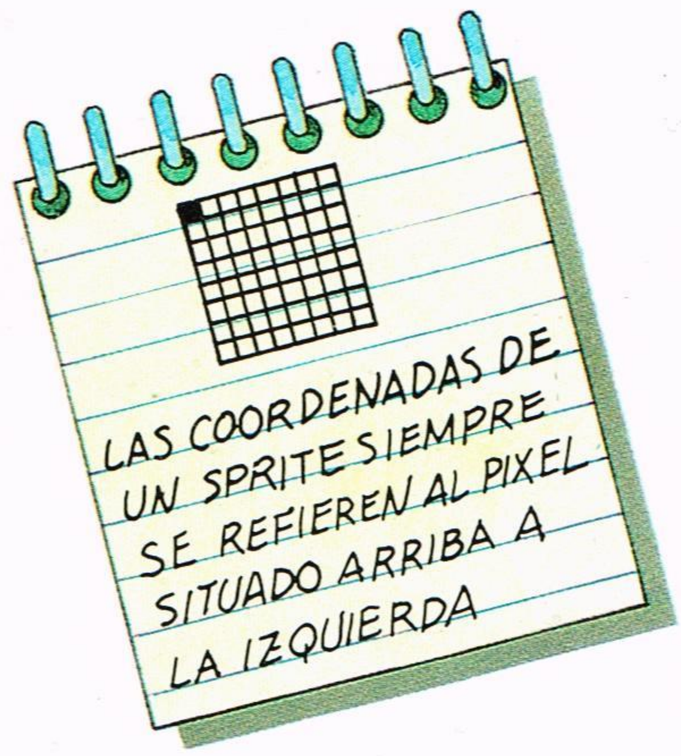


MOVIMIENTO Y DISPARO

En esta sección del programa verás cómo conseguir que los invasores se muevan aleatoriamente hacia abajo por la pantalla. También contiene las instrucciones que permiten al jugador mover su nave y disparar misiles para derribar alienígenas. En la parte superior de la pantalla se da el número de invasores destruidos. ¡Buena suerte!



La siguiente subrutina **Alienígena** se encarga de avanzar los invasores hacia la parte baja de la pantalla. Recuerda que en la subrutina **Inicializar** habíamos establecido las coordenadas iniciales de cada invasor. Como estamos en el segundo modo de pantalla —gráfico— trabajaremos sobre una cuadrícula de 256 por 192 pixels, pudiendo colocar cada sprite en cualquier lugar de la pantalla. No obstante, los movimientos de todos los sprites los haremos avanzando de 8 en 8 pixels; para ello simplemente cambiaremos las coordenadas del mismo y lo colocaremos en las nuevas como habíamos hecho en el capítulo introductorio.



La línea 1010 selecciona un valor de **R** entre 1 y 10, que nos servirá para escoger el invasor. Si se ha escogido un alienígena cuya coordenada vertical tiene el valor 209 (valor escogido para representar un alienígena destruido que esté ya fuera del juego) la subrutina finaliza.

```
1000 REM Alienigena
1010 R=INT(RND(1)*(INVAS))+1:IF Y(R)=209 THEN RETURN
1020 X(R)=X(R)+8*(INT(RND(1)*3)-1)
1030 Y(R)=Y(R)+8
1040 IF X(R)>248 THEN X(R)=248
1050 IF X(R)<16 THEN X(R)=16
1060 IF Y(R)=184 THEN PERDER=1
1070 PUT SPRITE R,(X(R),Y(R)),5
1080 RETURN
```

Una vez escogido un alienígena válido, las líneas 1020 y 1030 calculan sus nuevas coordenadas. Añadiremos un número aleatorio (–8, 0 u 8) al valor actual de la coordenada **X(R)** para variar su posición horizontal, e incrementaremos en 8 unidades el valor de **Y(R).** Con esto conseguiremos que el alienígena descienda y se desplace a su nueva posición 8 pixels a la izquierda, con la misma coordenada horizontal, u 8 pixels a la derecha respectivamente.

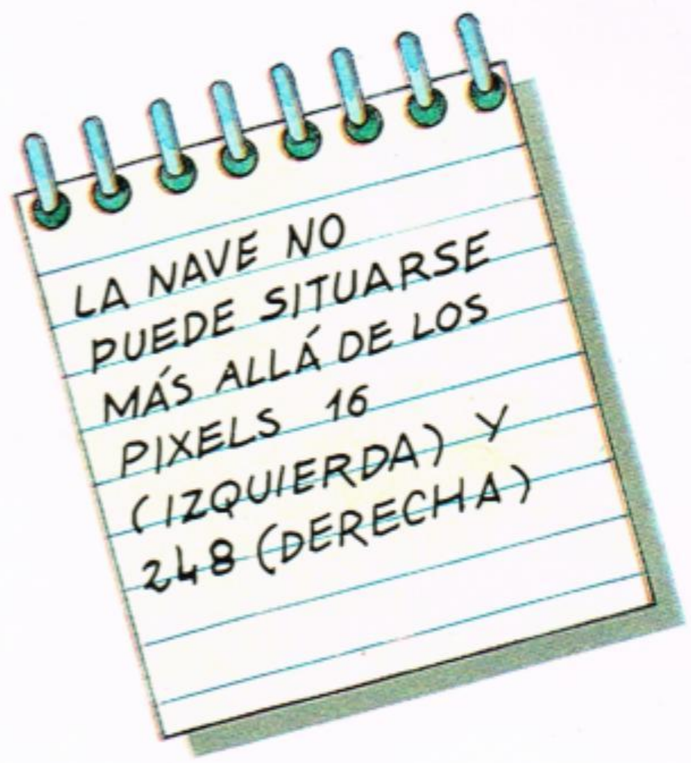
Las líneas 1040 y 1050 aseguran que el alienígena no se salga de los límites que tiene asignados; éstos son las coordenadas horizontales 248 (por la derecha) y 16 (por la izquierda). Si un invasor desciende hasta la altura de la nave, la línea 1060 activa el flag **PERDER,** que indica que el juego ha terminado. Por último la línea 1070 coloca el sprite alienígena en su nueva posición.

La subrutina **Jugador** permite mover la nave horizontalmente por la línea inferior de la pantalla. En la línea 2010 **INKEY** explora el teclado y asigna a **R$** la tecla pulsada; caso de no pulsarse ninguna se le asigna una tecla cualquiera (X). Si la tecla pulsada es la barra espaciadora, la línea 2020 desvía el control a la subrutina 3000 que se encarga de disparar el láser.

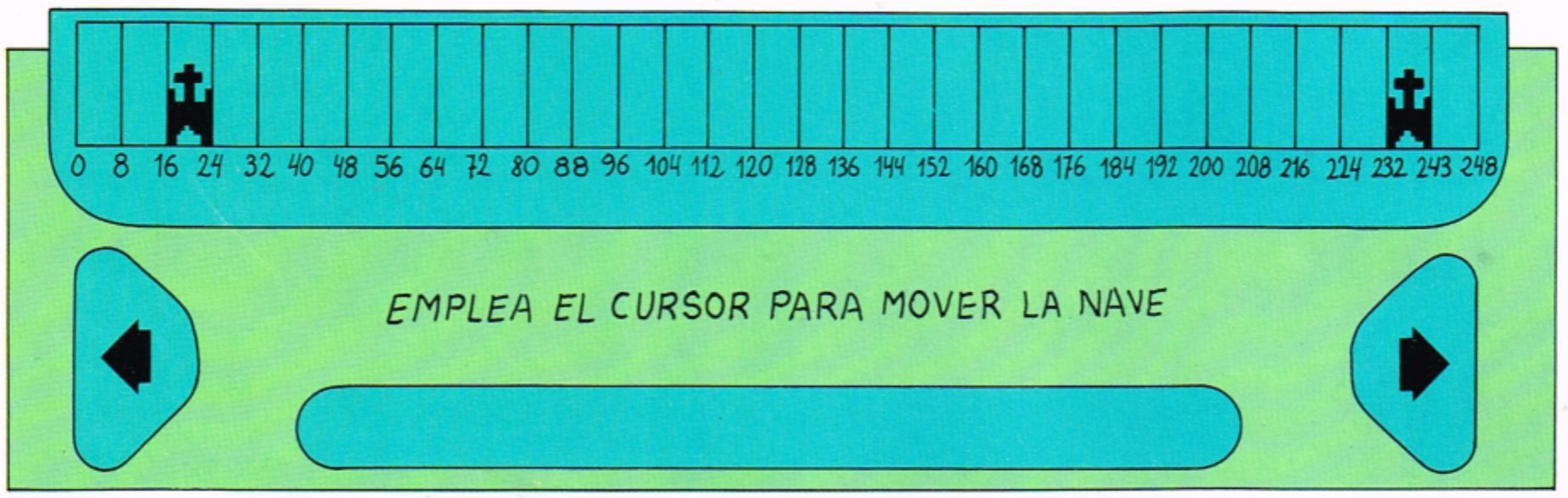
```
2000 REM Jugador
2010 R$=INKEY$:IF R$="" THEN R$="X"
2020 IF R$=" " THEN GOSUB 3000:REM Laser
2030 IF ((ASC(R$)=29) AND XP>0) THEN XP=XP-8:
     IF XP<16 THEN XP=16
2040 IF (ASC(R$)=28) AND XP<256 THEN XP=XP+8:
     IF XP>248 THEN XP=248
2050 PUT SPRITE 12,(XP,184),8
2060 RETURN
```

Las líneas 2030 y 2040 controlan el movimiento de la nave. Antes de cambiar su coordenada horizontal comprobaremos que se encuentra dentro de los límites impuestos (que lógicamente coinciden con los de los invasores).

Cuando la tecla pulsada sea el cursor izquierdo, **R$** tendrá el valor ASCII 29 y si **XP** es mayor que 16 decrementaremos su valor 8 unidades —línea 2030—. Análogamente, si pulsamos el cursor derecho **R$** tendrá el valor ASCII 28, y si está dentro de los límites aumentaremos la posición horizontal en 8 unidades (línea 2040). Por último, la línea 2050 coloca el sprite correspondiente a la nave en su nueva posición.



La subrutina **Láser** comienza en la línea 3000 y es la encargada de dibujar el misil ascendiendo por la pantalla. Esto lo consigue de una forma muy simple. Una vez fijada la posición horizontal del misil —que coincide con la de la nave—, dibujaremos el sprite misil consecutivamente en la misma posición horizontal pero disminuyendo su posición vertical. La línea 3020 decrementa la posición vertical del misil en 8 unidades y la 3030 lo dibuja en la posición correspondiente. Por último, la línea 3040 comprueba si el misil ha llegado a la parte superior de la pantalla, borrándolo en ese caso. De esta forma el bucle 3020-3060 consigue el movimiento ascendente.

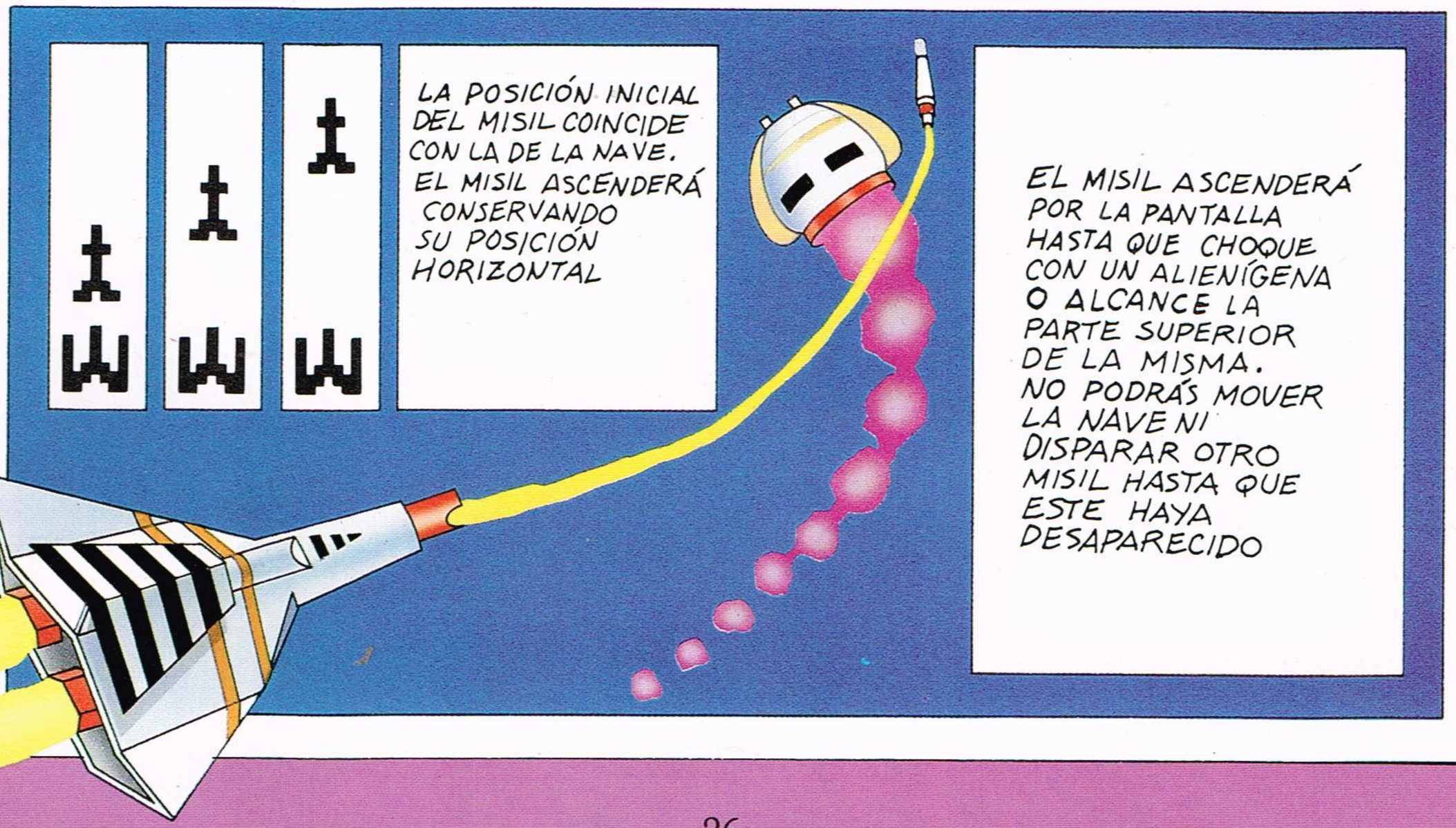


```
3000 REM Laser
3010 YM=184:XM=XP:GOSUB 3500
3020 YM=YM-8
3030 PUT SPRITE 11,(XM,YM),11
3040 IF YM=0 THEN PUT SPRITE 11,(0,209):AUX=1:RETURN
3050 IF AUX=1 THEN SPRITE ON
3060 GOTO 3020
3500 SOUND 0,128:SOUND 1,1:SOUND 2,0:SOUND 3,0:
     SOUND 4,0:SOUND 5,0:SOUND 6,1:SOUND 7,54:
     SOUND 8,16:SOUND 9,0:SOUND 10,0:SOUND 11,251:
     SOUND 12,10:SOUND 13,15
3510 RETURN
```

Antes de continuar conviene explicar la línea 8220 de la subrutina **Inicializar** que hemos reservado a propósito para este apartado. **ON SPRITE** permite detectar cuándo se cruzan dos sprites en la pantalla, y si está activa la instrucción **ON SPRITE** se ejecuta inmediatamente la subrutina de tratamiento de esta interrupción. Pues bien, la línea 8220 indica que en caso de cruzarse dos sprites hay que ejcutar la subrutina que comienza en la línea 4000. **SPRITE OFF** desactiva por el momento este tipo de interrupciones.

En su ascenso por la pantalla, el misil puede chocar con un alienígena, debiendo saltar a la subrutina de coincidencia de sprites. Cuando se ha disparado el misil hay que distinguir la coincidencia de dos sprites invasores —que no debe dar lugar a ningún efecto especial— de la coincidencia de un sprite invasor con un misil. Para ello emplearemos el flag **AUX**, que se pondrá a 0 cuando los sprites que coinciden sean dos invasores.

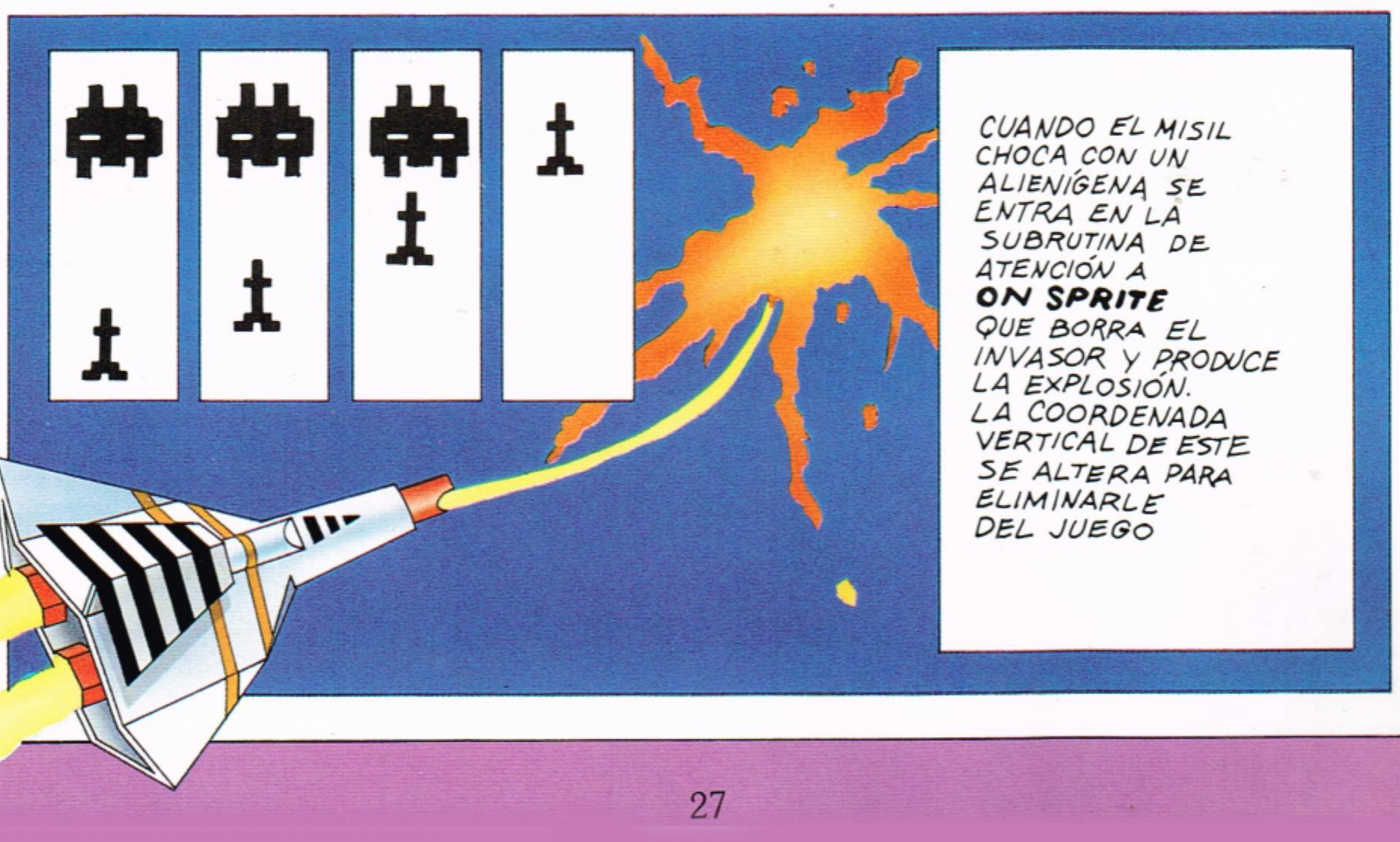
La subrutina de coincidencia de sprites comienza desactivando la interrupción **ON SPRITE** hasta que se ejecute el siguiente **SPRITE ON**. El flag **COLIS** indicará si ha habido o no colisión entre el misil y el invasor, de ahí que se inicialice a 0.



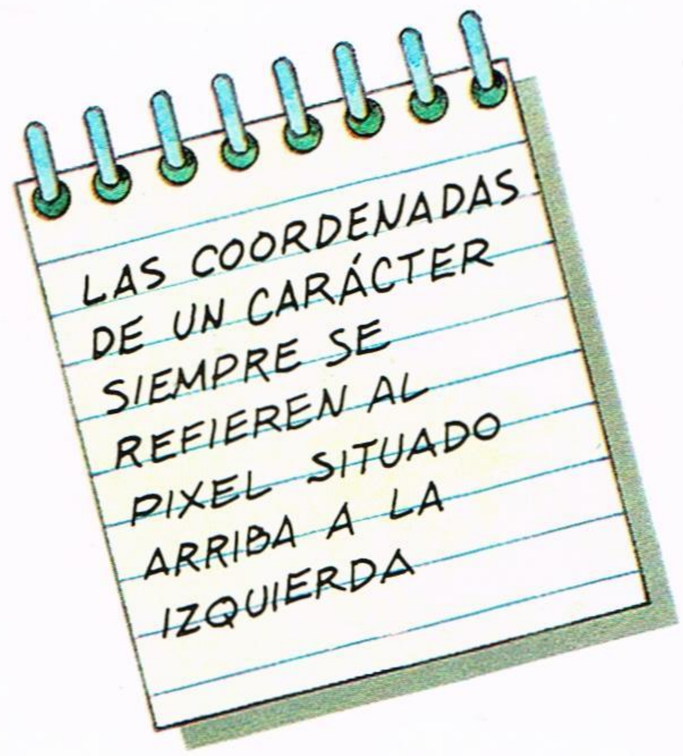
```
4000 REM Coincidencia de sprites
4010 SPRITE OFF:COLIS=0
4020 FOR I=1 TO INVAS
4030 IF (X(I)<>XM OR ABS(YM-Y(I))>8) THEN GOTO 4050
4040 Y(I)=209:PUT SPRITE I,(0,209):I=INVAS:GOSUB 4500:
     COLIS=1
4050 NEXT I
4060 AUX=COLIS
4070 IF COLIS=0 THEN RETURN
4080 ALIENS=ALIENS-1:MAR=MAR+1
4090 IF ALIENS=0 THEN GANAR=1
4100 PRESET (200,0):COLOR 1:PRINT#1, MAR-1:
     PRESET (200,0):COLOR 15:PRINT#1,MAR
4110 RETURN
4500 SOUND 0,0:SOUND 1,0:SOUND 2,0:SOUND 3,0:SOUND 4,0:
     SOUND 5,0:SOUND 6,31:SOUND 7,7:SOUND 8,16:
     SOUND 9,16:SOUND 10,16:SOUND 11,0:SOUND 12,60:
     SOUND 13,0
4510 RETURN
```

A continuación, el bucle 4020-4050 compara las posiciones de todos los invasores con las del misil y determina si ha habido colisión. En ese caso la línea 4040 borra el invasor, le asigna la coordenada 209 —invasor destruido— llama a la subrutina 4500 que consigue el sonido de la explosión y activa el flag **COLIS**. Para salir del bucle establece la variable I a su valor final **INVAS**. La línea 4060 asigna al flag **AUX** el valor de **COLIS** de modo que si no ha habido colisión —se habrá entrado en esta subrutina por coincidencia de dos invasores— **AUX** se encargará en la subrutina **Láser** de que se desactiven las interrupciones **ON SPRITE** que retardarían el ascenso del misil.

Si ha habido colisión, el número de alienígenas se decrementa en uno y el marcador aumenta —línea 4080—. Si ya no quedan invasores se activa el flag **GANAR**. 4100 actualiza el marcador.



La subrutina **Alienígena** elige un invasor de forma aleatoria y lo pinta de negro; de esta forma permanece invisible en el fondo hasta que se calculan las nuevas coordenadas adyacentes y se vuelve a dibujar el alienígena en amarillo. La impresión de movimiento se consigue borrando el invasor y dibujándolo en una posición inferior a la anterior. La línea 1010 selecciona un número entero entre 1 y el valor de **invasores.** Este valor es almacenado en la variable **r.** El valor de **r** hace referencia a un alienígena en concreto. No obstante si se ha elegido un alienígena que resulta tener unas coordenadas verticales fuera de la pantalla —un alienígena "muerto"— la subrutina finaliza con una instrucción **RETURN**. Esto se explicará más detenidamente en la subrutina **Impacto** que describiremos más adelante.



Cuando se escoge un alienígena correcto la línea 1020 lo borra de sus antiguas coordenadas, **x(r), y(r).**

```
1000 REM Alienigena
1010 LET r=INT (RND*invasores)+1: IF y(r)=22  THEN
     RETURN
1020 PAPER 0: PRINT AT y(r),x(r);" "
1030 LET x(r)=x(r)+INT (RND*3)-1
1040 LET y(r)=y(r)+1
1050 IF x(r)>30 THEN  LET x(r)=30
1060 IF x(r)<1 THEN  LET x(r)=1
1070 IF y(r)=21 THEN  LET perder=1
1080 INK 6: PRINT AT y(r),x(r);"👾"
1090 RETURN
```

Las nuevas coordenadas se calculan en la línea 1030 mediante la función **RND.** Añadiremos un número aleatorio —1, 0 ó 1, al valor actual de **x(r),** para variar la posición horizontal, e incrementaremos en una unidad el valor de **y(r)**. De esta forma conseguiremos que el alienígena se desplace a su nueva posición un espacio hacia abajo y a la izquierda, sin variar su posición horizontal, o abajo y a la derecha respectivamente.

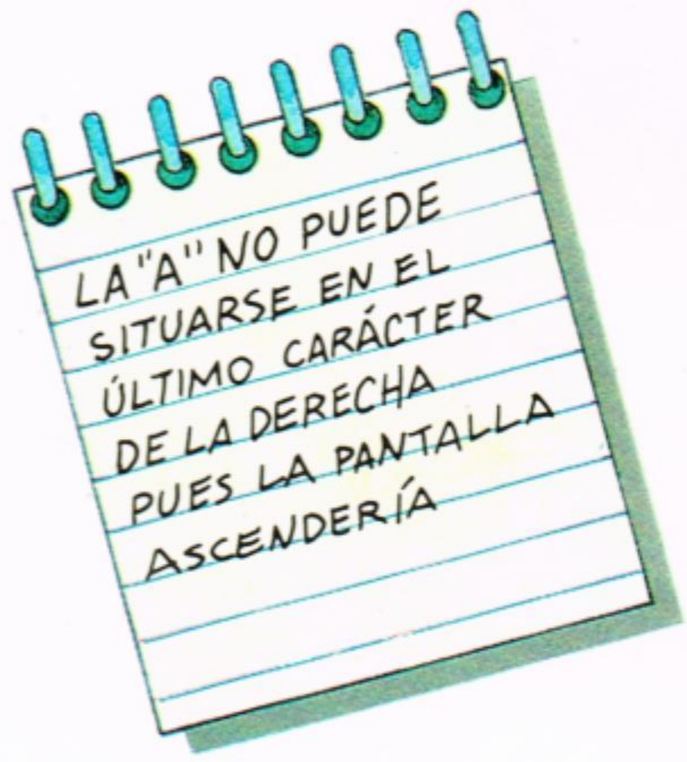
Las líneas 1050 y 1060 establecen los límites del valor de **x(r)** para asegurar que la posición horizontal del alienígena nunca pueda salirse de la pantalla (si no lo hiciéramos el programa finalizaría al intentar imprimir el invasor y el Spectrum daría un mensaje de error). La siguiente línea, 1070, estudia el valor de **y(r).** Si es igual a 21 el invasor ha alcanzado el fondo de la pantalla y se activa el flag **perder.** Se imprime ahora el alienígena en su nueva posición mediante la línea 1080. La figura del invasor es la definida en la página 9; sigue las instrucciones que se dieron allí para conseguir el carácter para tu listado.

El movimiento conseguido en esta subrutina es doblemente aleatorio en cuanto a la elección del invasor y en cuanto a su nueva posición. De esta manera se consigue el efecto de descenso al azar necesario para este tipo de juegos en el que cualquier periodicidad disminuye su atractivo.

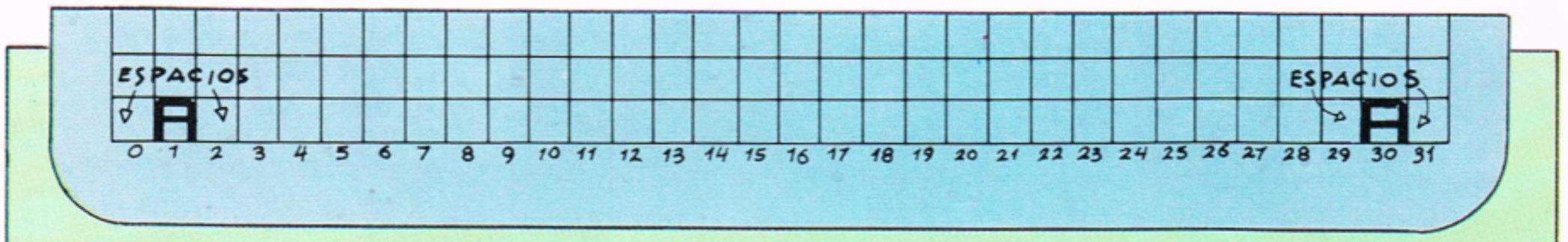
La siguiente subrutina que llama el programa principal permite al jugador mover su nave por la línea inferior de la pantalla. En la línea 2010 **NKEY$** explora el teclado para ver si se ha pulsado una tecla, y almacena el resultado en **l$**. La siguiente línea estudia esta variable, y si **l$** es **"m"** o **"M"** (con lo que aceptamos letras mayúsculas o minúsculas) se ejecuta la instrucción **GOSUB 3000** que es la subrutina que se encarga de disparar el misil y que luego definiremos.

```
2000 REM Jugador
2010 LET l$=INKEY$
2020 IF l$="m" OR l$="M" THEN  GO SUB 3000: REM Laser
2030 IF (l$="z" OR l$="Z") AND xp>0 THEN  LET xp=xp-1
2040 IF (l$="x" OR l$="X") AND xp<29 THEN  LET xp=xp+1
2050 INK 2: PRINT AT 21,xp;" A "
2060 RETURN
```

Las líneas 2030 y 2040 controlan el movimiento de la nave. Si **l$** es **"z"** o **"Z"** y la posición horizontal del jugador es mayor que 0, entonces se decrementa el valor de **xp** con lo que moveremos la nave hacia la izquierda. Si el valor de **l$** es **"x"**o **"X"** entonces la moveremos un espacio a la derecha. Los límites impuestos al valor de **xp** aseguran que esta coordenada nunca puede salirse de la pantalla. La expresión **xp=xp-1** es muy usual en el lenguaje informático; en general significa que la variable es alterada en la cantidad especificada respecto a su valor en el instante de ejecución. En nuestro caso el valor de **xp** —la coordenada horizontal de la nave en la pantalla— se decrementa una unidad.



La nave la representaremos mediante la letra A, que se imprime en la pentalla mediante los caracteres "A", es decir, entre dos espacios en blanco. La misión de estos espacios es borrar la antigua **A** al mover la nave hacia la izquierda o la derecha, como vimos en la introducción.

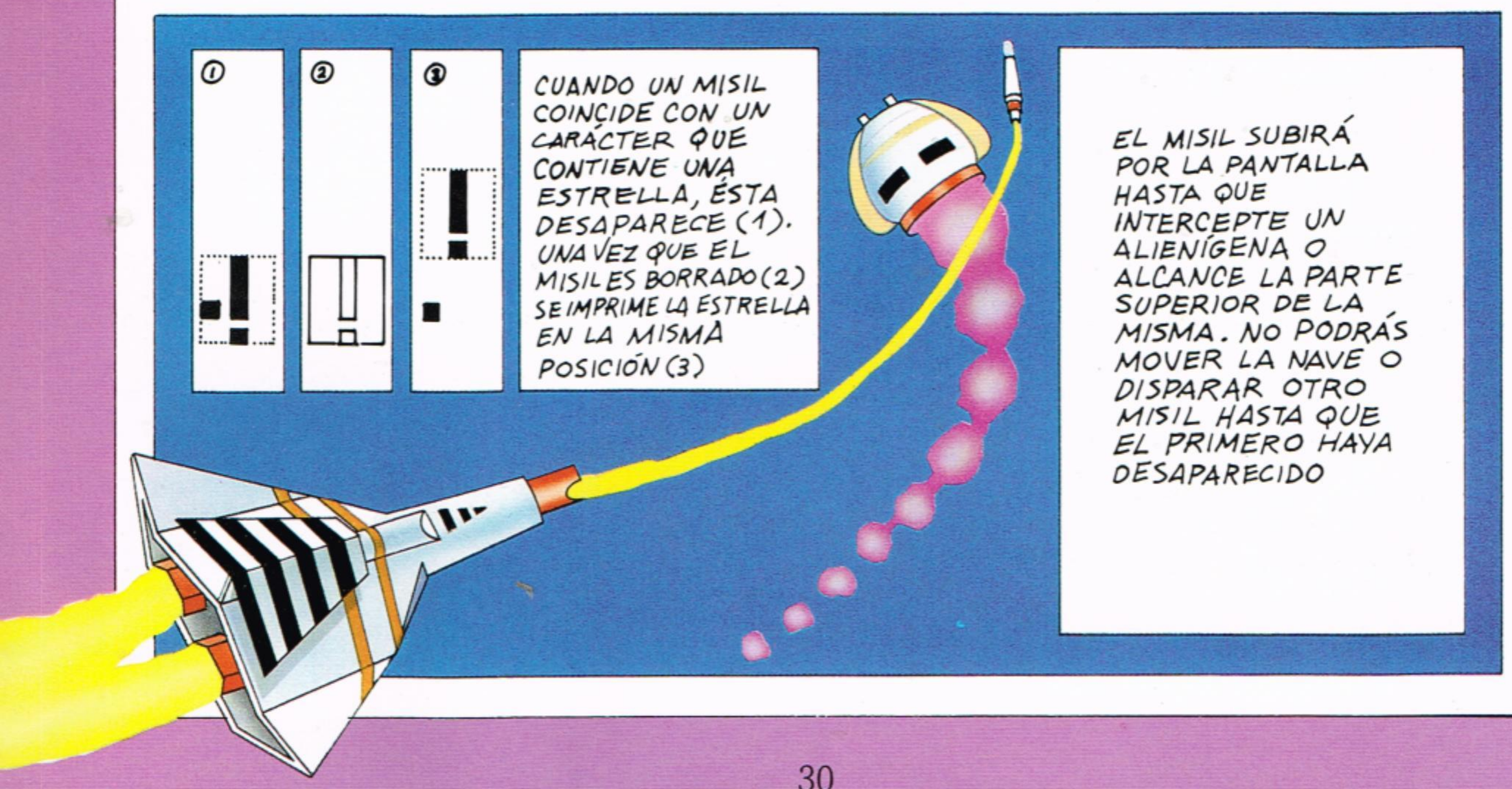


LA TECLA Z MUEVE LA NAVE (A) A LA IZQUIERDA HASTA LA COLUMNA 1

LA TECLA X MUEVE LA NAVE (A) A LA DERECHA HASTA LA COLUMNA 29

```
3000 REM Laser
3010 LET ym=21: LET xm=xp+1
3020 LET ym=ym-1
3030 IF POINT (xm*8,(21-ym)*8+2)=1  THEN  LET colest=1
3040 IF POINT (xm*8,(21-ym)*8+3)=1 THEN  LET colalien=1
3050 INK 7: PRINT AT ym,xm;"!"
3060 PAPER 0: PRINT AT ym,xm;" "
3070 IF colest=1 THEN  PLOT xm*8,(21-ym)*8+2:
     LET colest=0
3080 IF colalien=1 THEN  GO SUB 4000
3090 IF ym=1 THEN  RETURN
3100 GO TO 3020
```

La subrutina **Láser** comienza en la línea 3000. En primer lugar la línea 3010 establece las coordenadas iniciales del misil en las del jugador **(ym, xm)**. Conseguiremos que el misil cruce la pantalla mediante un bucle que altera la posición vertical **(ym)** en una unidad cada vez. Dentro del bucle hay que estudiar si el misil va a borrar una estrella, en cuyo caso ésta debe ser restaurada tras su paso, o si va a interceptar un alienígena, debiendo pasar el control a la subrutina **Impacto.** Esto lo conseguimos mediante la función **POINT**, que estudia un pixel en concreto para ver si corresponde al color del fondo (0) ó a otro (1). La línea 3030 estudia la posible coincidencia con una estrella, en cuyo caso se activa el flag **colest.** La línea 3040 comprueba un pixel que debería estar pintado si la posición correspondiente estuviera ocupada por un invasor, y actúa sobre el flag **colalien** en consecuencia. El misil se consigue mediante una exclamación (!) impresa en color blanco y borrada posteriormente con negro. Si el flag **colest** se ha activado, la estrella se dibuja de nuevo en su antigua posición. Si el flag **colalien** está a **1** el ordenador salta entonces a la subrutina **Impacto** que ahora describiremos. Por último, cuando **ym=1,** es decir, cuando el misil ha llegado a la parte superior de la pantalla, la subrutina finaliza con una sentencia **RETURN.**



En la subrutina **Impacto** se estudia cada invasor uno por uno mediante un bucle **FOR... NEXT** para ver si su posición coincide con la del misil. Si sus coordenadas x ó y no son las mismas, el ordenador salta a la línea 4040 para pasar al siguiente alienígena. Sin embargo, cuando sus posiciones coinciden la línea 4030 cambia la posición vertical del invasor de modo que éste pase a la fila 22 —fuera de la pantalla—. El alienígena se borra mediante el carácter en blanco asociado al misil. El bucle finaliza estableciendo el valor de **a** a su valor máximo (el número de invasores) ya que es una práctica poco recomendable saltar desde el interior de un bucle.

```
4000 REM Impacto
4010 FOR a=1 TO invasores
4020 IF x(a)<>xm OR y(a)<>ym THEN  GO TO 4040
4030 LET y(a)=22: LET a=invasores
4040 NEXT a
4050 LET alienigenas=alienigenas-1: LET puntos=puntos+1
     : LET colalien=0: LET ym=1
4060 IF alienigenas=0 THEN  LET ganar=1
4070 INK 7: PRINT AT 0,26;puntos
4080 RETURN
```

La línea 4050 modifica la puntuación, número de alienígenas eliminados, y desactiva el flag **colalien.** Una vez que se ha ejecutado la subrutina **Impacto** debe finalizar el proceso de disparo, lo que se consigue cambiando el valor de **ym** a 1, con lo que concluirá la subrutina **Láser.** Las dos últimas líneas analizan si el juego ha acabado (cuando se hayan eliminado todos los invasores) e imprime el nuevo tanteo en la parte superior de la pantalla.

Prueba tu programa

Puedes comprobar ahora las secciones del programa correspondientes a movimiento y disparo antes de pasar al capítulo final del libro. Sigue las instrucciones del recuadro amarillo. Deberías ser capaz de mover tu nave y atacar a los invasores en su descenso. Como antes, si recibes un mensaje de error estudia todo lo que hayas tecleado y compáralo con los listados del libro.

MSX

1. TECLEA LA LÍNEA PROVISIONAL 55 **GOTO** 40 Y PULSA RETURN

2. TECLEA **RUN** Y PULSA RETURN

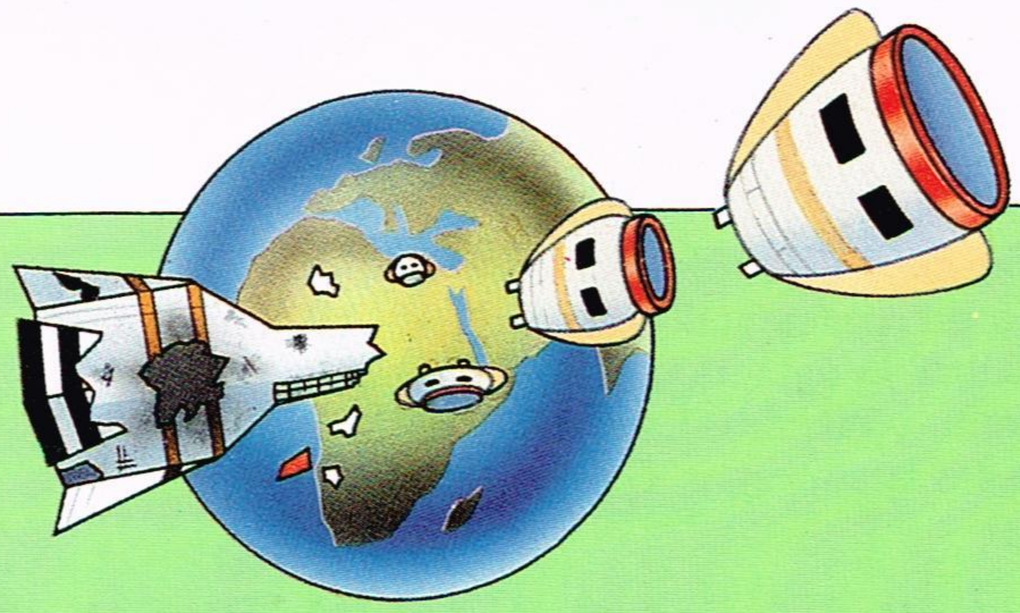
SI TODO ES CORRECTO DEBERÍAS SER CAPAZ DE DISPARAR A LOS ALIENÍGENAS EN SU DESCENSO POR LA PANTALLA. BORRA LA LÍNEA PROVISIONAL ANTES DE CONTINUAR

SPECTRUM

1. INTRODUCE LA LÍNEA PROVISIONAL **45 GOTO 30** Y PULSA ENTER

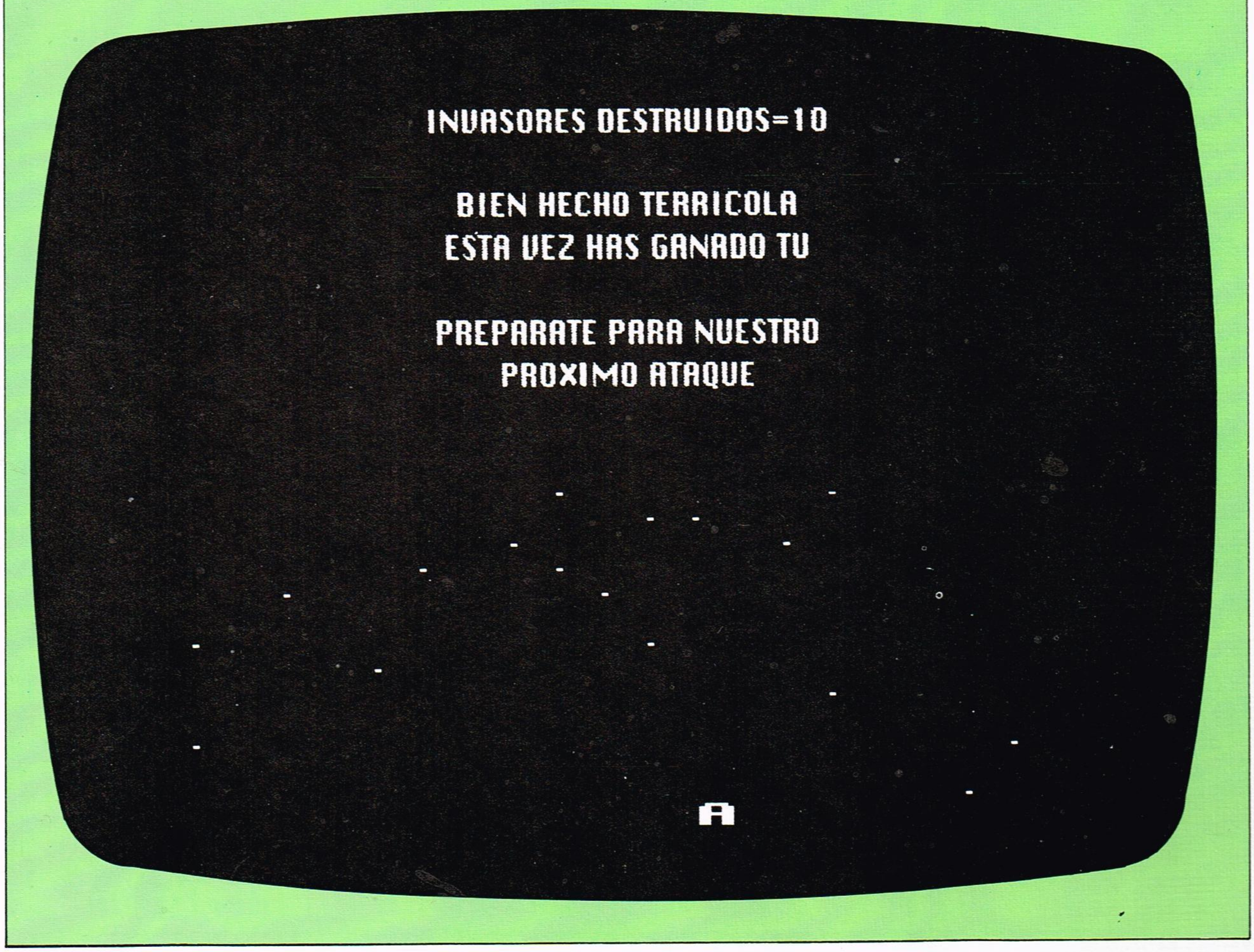
2. PULSA **RUN** Y LUEGO ENTER

SI TODO ES CORRECTO DEBERÍAS SER CAPAZ DE DISPARAR A LOS ALIENÍGENAS EN SU DESCENSO POR LA PANTALLA. EL PROGRAMA FINALIZARÁ SI ALGUNO DE ELLOS ALCANZA LA PARTE INFERIOR DE LA MISMA. BORRA LA LÍNEA PROVISIONAL ANTES DE CONTINUAR



GANAR Y PERDER

Esta es la sección que falta para completar el juego. Si el jugador consigue eliminar los diez alienígenas aparece una nueva oleada que comienza su descenso desde una altura menor y dan menos tiempo al jugador para atacarlos. Al finalizar la sección hay algunos trucos que hacen el programa más atractivo. Por último se da un listado completo.



Cuando el último alienígena ha sido destruido, la subrutina de atención a coincidencia de sprites activa el flag **GANAR**. La línea 60 del programa de control lo detecta y se sale del bucle encargado del movimiento y disparo. La línea 70 del programa principal llama entonces a la subrutina **Ganar.**

```
5000 REM Ganar
5010 PRESET (48,72):COLOR 13:
     PRINT #1, "BIEN HECHO TERRICOLA"
5020 PRESET (40,88):PRINT #1, "ESTA VEZ HAS GANADO TU"
5030 FOR D=1 TO 300:NEXT
5040 PRESET (72,112):PRINT #1, "PREPARATE PARA"
5050 PRESET (40,128):PRINT #1, "NUESTRO PROXIMO ATAQUE"
5060 FOR D=1 TO 500:NEXT
5070 ALIEN=INVAS:NIVEL=NIVEL+16:GANAR=0
5080 FOR A=1 TO INVAS
5090 X(A)=24*A:Y(A)=NIVEL
5100 NEXT A
5105 PRESET (48,72):COLOR 1:PRINT#1, "BIEN HECHO TERRIC
     OLA":PRESET (40,88):PRINT#1,"ESTA VEZ HAS GANADO
     TU":PRESET (72,112):PRINT#1,"PREPARATE PARA":
     PRESET (40,128):PRINT#1,"NUESTRO PROXIMO ATAQUE"
5110 RETURN
```

Aparece en la pantalla un mensaje de enhorabuena; como aún estamos en modo 2 es necesario localizarlo mediante **PRESET** e imprimirlo con **PRINT** . Antes de imprimir el siguiente mensaje se introduce un retardo mediante un bucle **FOR... NEXT.** Este obliga a contar de 1 a 300 antes de imprimir la segunda parte del texto. Tras el segundo mensaje hay otro bucle de retardo para dar tiempo al jugador a leer el texto antes de que finalice la subrutina y se borre la pantalla. La línea 5070 vuelve a establecer el valor de **ALIENS** a **INVAS,** desactiva el flag **GANAR** e incrementa **NIVEL** en 16 unidades de forma que la siguiente oleada de alienígenas comience en una posición más baja en la pantalla.

Un bucle **FOR... NEXT** repone las coordenadas x e y de cada alienígena de la misma forma que hicimos en la subrutina de inicialización. La línea 5105 borra el mensaje imprimiéndolo de nuevo con el color de fondo y **RETURN** devuelve el control al programa principal que entra de nuevo en el bucle activo del juego.

La subrutina **Perder** comienza cerrando el fichero 1 mediante **CLOSE** y estableciendo el modo de pantalla 0 (modo texto). Además cambia el color del texto y borra la pantalla, imprimiendo a continuación un mensaje. Tras un nuevo retardo aparece otro mensaje que indica el número de invasores destruidos.

```
6000 REM Perder
6010 CLOSE #1:SCREEN 0,0,0:COLOR 13,1,1:CLS
6020 LOCATE 16,3,0:PRINT "L O S"
6030 LOCATE 10,5,0:PRINT "I N V A S O R E S"
6040 LOCATE 16,7,0:PRINT "H A N"
6050 LOCATE 9,9,0:PRINT "A T E R R I Z A D O"
6060 FOR I=1 TO 800:NEXT
6070 LOCATE 13,13,0:PRINT "SIN EMBARGO"
6080 LOCATE 12,15,0:PRINT "HAS DESTRUIDO"
6090 LOCATE 17,17,0:PRINT MAR:COLOR 10
6100 LOCATE 10,19,0:PRINT "NAVES ALIENIGENAS"
6110 LOCATE 4,22,0:
     INPUT "QUIERES ENFRENTARTE DE NUEVO";A$
6120 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN RUN
6130 CLS:SCREEN 0,,1:COLOR 1,15,15:RETURN
```

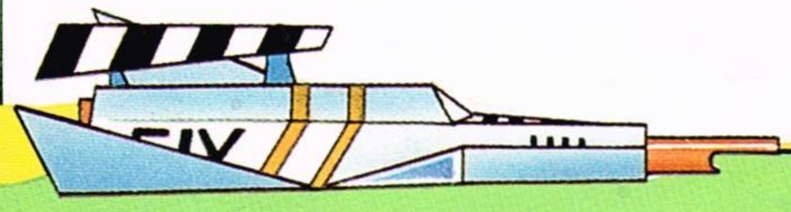
A continuación se ofrece la opción de jugar de nuevo en la línea 6110. La línea 6120 estudia la primera letra de la palabra y si esta comienza por "**s**" o "**S**", el programa se ejecuta de nuevo. De esta forma el jugador puede responder con cualquier palabra que comience por "**s**" para jugar otra vez. Con cualquier otra respuesta el ordenador, tras borrar la pantalla, volverá al programa principal, donde finalizará en la línea 90.

MSX POR FÍN PUEDES RODAR EL PROGRAMA COMPLETO. CUANDO HAYAS ELIMINADO TODOS LOS INVASORES VERÁS EN PANTALLA EL MENSAJE DE LA IZQUIERDA, Y EL JUEGO CONTINUARÁ CON LOS ALIENÍGENAS AVANZANDO DESDE UNA POSICIÓN MÁS BAJA. SI UN INVASOR ALCANZA EL FONDO DE LA PANTALLA HABRÁS PERDIDO, Y APARECERÁ EN ELLA EL MENSAJE DE LA DERECHA

RUN

```
INVASORES DESTRUIDOS=10

BIEN HECHO TERRICOLA
ESTA VEZ HAS GANADO TU

PREPARATE PARA NUESTRO
PROXIMO ATAQUE
```

```
LOS
INVASORES
HAN
ATERRIZADO

SIN EMBARGO
HAS DESTRUIDO
8
NAVES ALIENIGENAS

QUIERES ENFRENTARTE DE NUEVO?
```



Cuando el jugador es lo suficientemente hábil para eliminar el último alienígena de la pantalla, el flag **ganar** se activa en la subrutina **Impacto**, y la línea 70 del programa de control llama a la subrutina **Ganar** que comienza en la línea 5000.

```
5000 REM Ganar
5010 INK 5: PRINT AT 6,5;"BIEN HECHO TERRICOLA"
5020 PRINT AT 8,4;"ESTA VEZ HAS GANADO TU"
5030 FOR D=1 TO 200:NEXT D
5040 PRINT AT 12,8;"PREPARATE PARA"
5050 PRINT AT 14,4;"NUESTRA PROXIMO ATAQUE"
5060 FOR d=1 TO 400: NEXT d
5070 LET alienigenas=invasores: LET altura=altura+2:
     LET ganar=0
5080 FOR a=1 TO invasores
5090 LET x(a)=a*3: LET y(a)=altura
5100 NEXT a
5110 RETURN
```

Las líneas 5010 y 5020 imprimen un mensje de "enhorabuena" en la mitad superior de la pantalla. En la línea 5030 se introduce un retardo a base de un bucle **FOR... NEXT** que obliga a contar de 1 a 200 antes de imprimir la segunda parte del mensaje. Una vez hecho ésto se origina otro retardo del mismo modo para que el jugador tenga tiempo suficiente para leer el mensaje antes de que la subrutina concluya, borrándose la pantalla para dejarla lista para el siguiente ataque alienígena.

El número de alienígenas es inicializado al valor original de **invasores** en la linea 5070, y el flag **ganar** es desactivado poniéndolo a 0. El valor inicial de **altura** se ve incrementado en 2 unidades, lo que provoca que la altura inicial de los alienígenas disminuya de modo que cada oleada va siendo más difícil de destruir. Puedes modificar esta subrutina de forma que los invasores vuelvan a descender desde la altura inicial una vez logrado el grado de dificultad que consideres oportuno.

Un bucle **FOR... NEXT** restaura las coordenadas x e y de cada alienígena de un modo similar al empleado anteriormente en la subrutina **Inicializar.**

Queda por último definir la subrutina **Perder** (línea 6000). En ella se borra la pantalla y se imprime un mensaje intermitente. Tras un pequeño retardo aparece un segundo mensaje que indica al jugador su puntuación. La instrucción **FLASH 1** intercambia alternativamente los colores de la tinta y el papel para lograr el efecto de intermitencia. Este efecto se desactiva mediante la instrucción **FLASH 0.**

La mayoría de los juegos ofrecen al jugador la posibilidad de intentarlo de nuevo; esto lo conseguimos en las líneas 6110 a 6130. Mediante una sentencia **INPUT** introducimos la respuesta desde el teclado, que se almacena en **a$** cuando se pulsa

```
6000 REM Perder
6010 CLS
6020 INK 5: FLASH 1: PRINT AT 1,13;"L O S"
6030 PRINT AT 3,7;"I N V A S O R E S"
6040 PRINT AT 5,13;"H A N"
6050 PRINT AT 7,6;"A T E R R I Z A D O": FLASH 0
6060 FOR D=1 TO 200:NEXT D
6070 INK 6: PRINT AT 10,10;"SIN EMBARGO"
6080 PRINT AT 12,9;"HAS DESTRUIDO"
6090 INK 2: FLASH 1: PRINT AT 14,15;puntos: FLASH 0
6100 INK 6: PRINT AT 16,7;"NAVES ALIENIGENAS"
6110 PRINT AT 20,1;"QUIERES ENFRENTARTE DE NUEVO?"
6120 INPUT a$
6130 IF a$(1)="s"  OR a$(1)="S" THEN  RUN
6140 RETURN
```

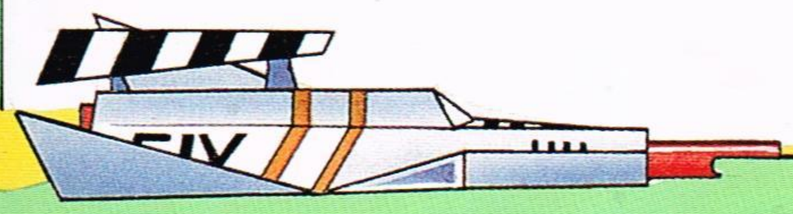
ENTER. La línea 6130 estudia la primera letra de **a$** y si se trata de una "s" (tanto mayúscula como minúscula) el programa comienza de nuevo. Con cualquier otra respuesta el ordenador vuelve al programa de control y finaliza en la línea 90 con la sentencia **STOP**. Puedes probar las subrutinas de ganar y perder siguiendo las instrucciones del siguiente cuadro:

SPECTRUM POR FÍN PUEDES RODAR EL PROGRAMA COMPLETO. CUANDO HAYAS ELIMINADO TODOS LOS INVASORES VERÁS EN PANTALLA EL MENSAJE DE LA IZQUIERDA, Y EL JUEGO CONTINUARÁ CON LOS ALIENÍGENAS AVANZANDO DESDE UNA POSICIÓN MÁS BAJA. SI UN INVASOR ALCANZA EL FONDO DE LA PANTALLA HABRÁS PERDIDO, Y APARECERÁ EN ELLA EL MENSAJE DE LA DERECHA

RUN

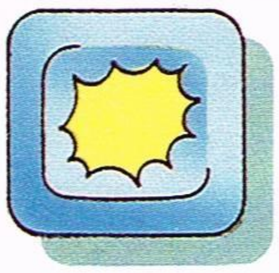
```
INVASORES DESTRUIDOS=10

BIEN HECHO TERRICOLA
ESTA VEZ HAS GANADO TU

PREPARATE PARA NUESTRO
PROXIMO ATAQUE
```

```
LOS
INVASORES
HAN
ATERRIZADO

SIN EMBARGO
HAS DESTRUIDO
8
NAVES ALIENIGENAS

QUIERES ENFRENTARTE DE NUEVO?
```



Mejora tu programa

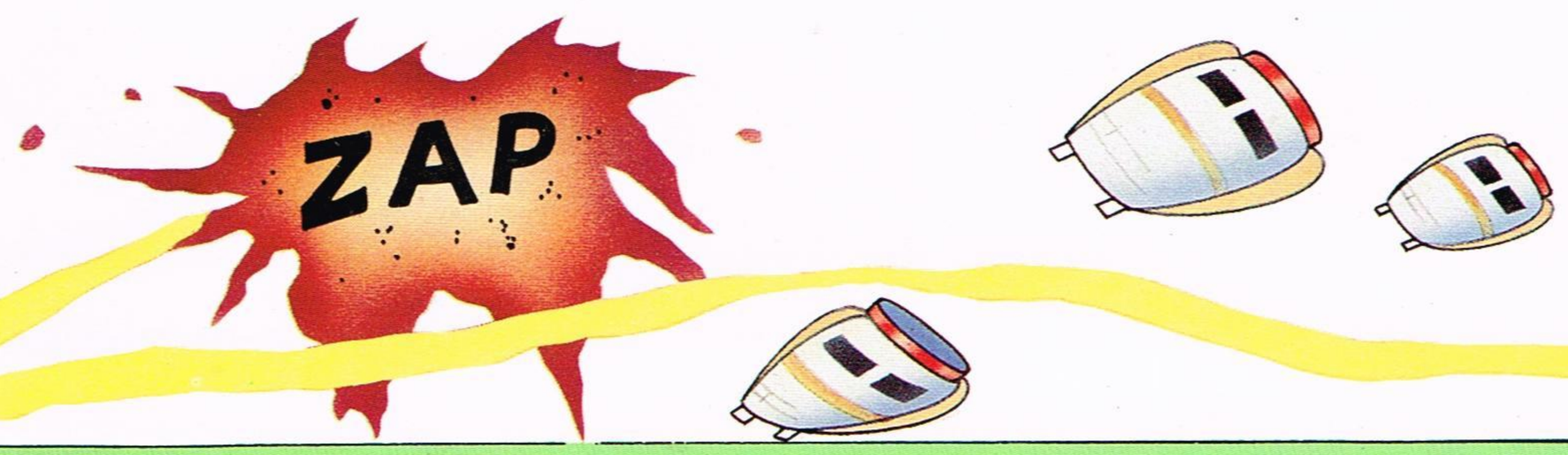
Tal y como está ahora tu juego de invasores resulta divertido y atrayente. Es difícil eliminar todos los alienígenas, que parecen moverse como si tuvieran vida. Sin embargo puedes hacer que el juego se parezca más a uno de tipo arcadia si añades algunos efectos de sonido bastante sencillos. A continuación se dan algunos trucos que puedes añadir al programa para que se parezca más a uno profesional y con el que podrás disfrutar tú y tus amigos.



Añade sonido MSX

Hemos incorporado en el programa los efectos de disparo y explosión, ambos basados en el comando **SOUND** de acceso directo al generador de sonido programable. A continuación te proponemos tres nuevas sentencias basadas en el comando **PLAY** con las que mejorarás la presentación de tu programa. La primera, 8015, ejecuta la melodía de comienzo; las líneas 5030 y 6060 en realidad sustituyen a sendos bucles de retardo e interpretan otras dos melodías.

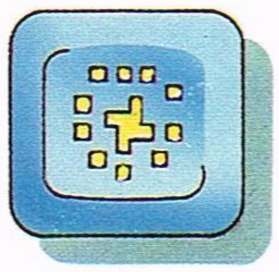
```
5030 FOR D=1 TO 300:NEXT:BEEP:PLAY "T100S1M60000O5L16CE
     GO6L8CO5L16GO6L8C"
6060 BEEP:PLAY "T90S1M500001A4A8.A32A402C8.01B16B8.A16A
     8.A16A2":FOR I=1 TO 2800:NEXT
8015 BEEP:PLAY "T160S8M60000O5L4CGL10FEDL606C7...O5L4GL
     10FEDL606C7...O5L4GL10FEFD4"
```



SPECTRUM

Podemos obtener sonido del Spectrum empleando la instrucción **BEEP** seguida de dos números, uno gobierna la duración del sonido y el otro su tono. No obstante debes tener mucho cuidado con los sonidos que incorpores al programa, pues pueden retrasar la velocidad del juego hasta un nivel inaceptable. La línea 4005 origina un pequeño "beep" cuando se intercepta un alienígena en la subrutina **Impacto.** Las líneas 5030 y 6060 son ligeramente diferentes pues no son una incorporación, sino una sustitución. En este caso las instrucciones **BEEP** ocupan el lugar de los bucles de retardo de las subrutinas **Ganar** y **Perder.** La línea 7055 advierte del comienzo del juego.

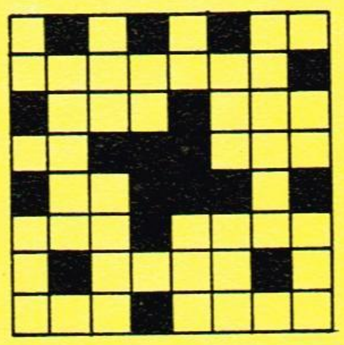
```
 4005 BEEP .05,-10
 5030 BEEP .5,2: BEEP .5,10: BEEP 1,20
 6060 BEEP 2,-10
 7055 FOR b=1 TO 20: BEEP .1,20: NEXT b
```



Gráficos extra MSX

Podemos mejorar aún más los gráficos añadiendo una explosión cuando derribemos un alienígena. Para ello definiremos el sprite explosión con el método usual —líneas 8215 y 7025. El carácter se imprime en la línea 4045 y tras un pequeño retardo se borra en la línea 4046. Por último, la línea 9040 contiene los datos necesarios para su definición.

```
4045 PUT SPRITE 0,(XM,YM+1),6:FOR I=1 TO 150:NEXT
4046 PUT SPRITE 0,(0,209),9
7025 SPRITE$(0)=B$
8215 FOR I=1 TO 8:READ N:B$=B$+CHR$(N):NEXT:REM Explos
9040 DATA 84,1,136,56,157,16,66,16:REM explosion
```



AL DERRIBAR UN INVASOR VEREMOS LA EXPLOSIÓN DE LA FIGURA DURANTE UN BREVE INTERVALO DE TIEMPO

Almacena tus programas

Una vez hayas tecleado el programa por completo, conviene almacenarlo en cinta. Las instrucciones para guardar programas varían muy poco de una máquina a otra, y debes consultar el manual de usuario para las instrucciones concretas. A continuación se da el listado completo del programa "Invasores del Espacio" para ambos tipos de ordenadores. En él se han incorporado las sugerencias de la página anterior.

Listados de los programas

MSX

```
  10 REM Invasores
  20 GOSUB 8000:REM Inicializar
  30 GOSUB 7000:REM Pantalla
  40 GOSUB 1000:REM Alienigena
  50 GOSUB 2000:REM Jugador
  60 IF (GANAR OR PERDER)=1 THEN GOTO 70 ELSE GOTO 40
  70 IF GANAR=1 THEN GOSUB 5000:GOTO 40
  80 GOSUB 6000:REM Perder
  90 END
1000 REM Alienigena
1010 R=INT(RND(1)*(INVAS))+1:IF Y(R)=209 THEN RETURN
1020 X(R)=X(R)+8*(INT(RND(1)*3)-1)
1030 Y(R)=Y(R)+8
1040 IF X(R)>248 THEN X(R)=248
1050 IF X(R)<16 THEN X(R)=16
1060 IF Y(R)=184 THEN PERDER=1
1070 PUT SPRITE R,(X(R),Y(R)),5
1080 RETURN
2000 REM Jugador
2010 R$=INKEY$:IF R$="" THEN R$="X"
2020 IF R$=" " THEN GOSUB 3000:REM Laser
2030 IF ((ASC(R$)=29) AND XP>0) THEN XP=XP-8:
     IF XP<16 THEN XP=16
2040 IF (ASC(R$)=28) AND XP<256 THEN XP=XP+8:
     IF XP>248 THEN XP=248
2050 PUT SPRITE 12,(XP,184),8
2060 RETURN
3000 REM Laser
3010 YM=184:XM=XP:GOSUB 3500
3020 YM=YM-8
3030 PUT SPRITE 11,(XM,YM),11
3040 IF YM=0 THEN PUT SPRITE 11,(0,209):AUX=1:RETURN
3050 IF AUX=1 THEN SPRITE ON
3060 GOTO 3020
3500 SOUND 0,128:SOUND 1,1:SOUND 2,0:SOUND 3,0:
     SOUND 4,0:SOUND 5,0:SOUND 6,1:SOUND 7,54:
     SOUND 8,16:SOUND 9,0:SOUND 10,0:SOUND 11,251:
     SOUND 12,10:SOUND 13,15
3510 RETURN
4000 REM Coincidencia de sprites
4010 SPRITE OFF:COLIS=0
4020 FOR I=1 TO INVAS
4030 IF (X(I)<>XM OR ABS(YM-Y(I))>8) THEN GOTO 4050
```

Continuación MSX

```
4040 Y(I)=209:PUT SPRITE I,(0,209):I=INVAS:GOSUB 4500:
     COLIS=1
4045 PUT SPRITE 0,(XM,YM+1),6:FOR I=1 TO 150:NEXT
4046 PUT SPRITE 0,(0,209),9
4050 NEXT I
4060 AUX=COLIS
4070 IF COLIS=0 THEN RETURN
4080 ALIENS=ALIENS-1:MAR=MAR+1
4090 IF ALIENS=0 THEN GANAR=1
4100 PRESET (200,0):COLOR 1:PRINT#1, MAR-1:
     PRESET (200,0):COLOR 15:PRINT#1,MAR
4110 RETURN
4500 SOUND 0,0:SOUND 1,0:SOUND 2,0:SOUND 3,0:SOUND 4,0:
     SOUND 5,0:SOUND 6,31:SOUND 7,7:SOUND 8,16:
     SOUND 9,16:SOUND 10,16:SOUND 11,0:SOUND 12,60:
     SOUND 13,0
4510 RETURN
5000 REM Ganar
5010 PRESET (48,72):COLOR 13:
     PRINT #1, "BIEN HECHO TERRICOLA"
5020 PRESET (40,88):PRINT #1, "ESTA VEZ HAS GANADO TU"
5030 FOR D=1 TO 300:NEXT:BEEP:PLAY "T100S1M6000005L16CE
     GO6L8CO5L16GO6L8C"
5040 PRESET (72,112):PRINT #1, "PREPARATE PARA"
5050 PRESET (40,128):PRINT #1, "NUESTRO PROXIMO ATAQUE"
5060 FOR D=1 TO 500:NEXT
5070 ALIEN=INVAS:NIVEL=NIVEL+16:GANAR=0
5080 FOR A=1 TO INVAS
5090 X(A)=24*A:Y(A)=NIVEL
5100 NEXT A
5105 PRESET (48,72):COLOR 1:PRINT#1, "BIEN HECHO TERRIC
     OLA":PRESET (40,88):PRINT#1,"ESTA VEZ HAS GANADO
     TU":PRESET (72,112):PRINT#1,"PREPARATE PARA":
     PRESET (40,128):PRINT#1,"NUESTRO PROXIMO ATAQUE"
5110 RETURN
6000 REM Perder
6010 CLOSE #1:SCREEN 0,0,0:COLOR 13,1,1:CLS
6020 LOCATE 16,3,0:PRINT "L O S"
6030 LOCATE 10,5,0:PRINT "I N V A S O R E S"
6040 LOCATE 16,7,0:PRINT "H A N"
6050 LOCATE 9,9,0:PRINT "A T E R R I Z A D O"
6060 BEEP:PLAY "T90S1M500001A4A8.A32A402C8.01B16B8.A16A
     8.A16A2":FOR I=1 TO 2800:NEXT
6070 LOCATE 13,13,0:PRINT "SIN EMBARGO"
6080 LOCATE 12,15,0:PRINT "HAS DESTRUIDO"
6090 LOCATE 17,17,0:PRINT MAR:COLOR 10
6100 LOCATE 10,19,0:PRINT "NAVES ALIENIGENAS"
6110 LOCATE 4,22,0:
     INPUT "QUIERES ENFRENTARTE DE NUEVO";A$
6120 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN RUN
6130 CLS:SCREEN 0,,1:COLOR 1,15,15:RETURN
7000 REM Pantalla
7010 SCREEN 2,0,0:COLOR 15,1,1:CLS
7020 SPRITE$(12)=N$:SPRITE$(11)=M$
7025 SPRITE$(0)=B$
7030 FOR I=1 TO 10:SPRITE$(I)=A$:NEXT
7040 FOR S=1 TO 60
7050 X=INT(RND(5)*250):Y=INT(RND(5)*190):
     PSET (X,Y),RND(1)*15
7060 NEXT S
7070 CIRCLE (52,28),16,9:CIRCLE (44,28),16,9,4.95673,
     1.32645:PAINT (44,28),9
```

Continuación MSX

```
7080 CIRCLE (192,42),25,7:CIRCLE (180,42),24,7,4.95673,
     1.32645:PAINT (180,42),7
7090 CIRCLE (0,192),60,10,0,-3.1459/2:
     LINE (0,192)-(60,192),10:PAINT (5,190),10,10
7100 PRESET (200,144),4:DRAW"G32F16E32H16R8F16L8BR8G32
     L8":PAINT (192,168),4,4
7110 OPEN "GRP:" AS #1:PRESET (32,0):
     PRINT #1, "INVASORES DESTRUIDOS:";MAR
7120 RETURN
8000 REM Inicializar
8010 SCREEN 0,0,0:COLOR 15,1,1:KEY OFF:CLS
8015 BEEP:PLAY "T160S8M6000005L4CGL10FEDL606C7...05L4GL
     10FEDL606C7...05L4GL10FEFD4"
8020 LOCATE 2,5,0:
     PRINT "ERES UN PILOTO SOLITARIO ENCARGADO"
8030 LOCATE 4,6,0:
     PRINT "DE PROTEGER EL PLANETA TIERRA."
8040 LOCATE 6,7,0:PRINT "EN UNOS INSTANTES ESTARAS"
8050 LOCATE 5,8,0:PRINT "BAJO EL ATAQUE DE INVASORES"
8060 LOCATE 9,9,0:PRINT "DEL PLANETA VARGON."
8070 LOCATE 3,11,0:
     PRINT "TU MISION ES EVITAR QUE ATERRICEN"
8080 LOCATE 7,12,0:PRINT "LAS NAVES VARGONIANAS."
8090 LOCATE 1,14,0:
     PRINT "EMPLEA EL CURSOR PARA MOVER TU NAVE"
8100 LOCATE 6,16,0:PRINT "PULSA LA BARRA ESPACIADORA"
8110 LOCATE 8,17,0:PRINT "PARA DISPARAR EL LASER"
8120 LOCATE 6,19,0:PRINT "PULSA CUALQUIER TECLA PARA"
8130 LOCATE 4,20,0:
     PRINT "ENTABLAR COMBATE CON EL ENEMIGO"
8140 PERDER=0:GANAR=0:MAR=0:XP=120:NIVEL=8:INVAS=10:
     ALIENS=INVAS:COLIS=0:AUX=1
8150 DIM X(INVAS):DIM Y(INVAS)
8160 FOR A=1 TO INVAS
8170 X(A)=24*A:Y(A)=NIVEL
8180 NEXT A
8190 FOR I=1 TO 8:READ N:M$=M$+CHR$(N):NEXT:REM Misil
8200 FOR I=1 TO 8:READ N:N$=N$+CHR$(N):NEXT:REM Nave
8210 FOR I=1 TO 8:READ N:A$=A$+CHR$(N):NEXT:REM Invasor
8215 FOR I=1 TO 8:READ N:B$=B$+CHR$(N):NEXT:REM Explos
8220 ON SPRITE GOSUB 4000:SPRITE OFF
8230 Q$=INPUT$(1)
8240 RETURN
9000 REM Datos sprites
9010 DATA 16,16,56,16,16,16,56,108:REM Misil
9020 DATA 16,56,16,186,186,186,238,108:REM Nave
9030 DATA 36,36,126,255,153,255,90,66:REM Invasor
9040 DATA 84,1,136,56,157,16,66,16:REM explosion
```

SPECTRUM

```
 5 REM Invasores
10 GO SUB 8000: REM Inicializar
20 GO SUB 7000: REM Pantalla
30 GO SUB 1000: REM Alienigena
40 GO SUB 2000: REM Jugador
50 IF (ganar  OR perder)=1 THEN  GO TO 70
60 GO TO 30
```

Continuación Spectrum

```
  70 IF ganar=1 THEN  GO SUB 5000: GO TO 20
  80 GO SUB 6000: REM Perder
  90 CLS : PAPER 7: BORDER 7: INK 0: STOP
1000 REM Alienigena
1010 LET r=INT (RND*invasores)+1: IF y(r)=22  THEN
     RETURN
1020 PAPER 0: PRINT AT y(r),x(r);" "
1030 LET x(r)=x(r)+INT (RND*3)-1
1040 LET y(r)=y(r)+1
1050 IF x(r)>30 THEN  LET x(r)=30
1060 IF x(r)<1 THEN  LET x(r)=1
1070 IF y(r)=21 THEN  LET perder=1
1080 INK 6: PRINT AT y(r),x(r);"ж"
1090 RETURN
2000 REM Jugador
2010 LET 1$=INKEY$
2020 IF 1$="m" OR 1$="M" THEN  GO SUB 3000: REM Laser
2030 IF (1$="z" OR 1$="Z") AND xp>0 THEN  LET xp=xp-1
2040 IF (1$="x" OR 1$="X") AND xp<29 THEN  LET xp=xp+1
2050 INK 2: PRINT AT 21,xp;" A "
2060 RETURN
3000 REM Laser
3010 LET ym=21: LET xm=xp+1
3020 LET ym=ym-1
3030 IF POINT (xm*8,(21-ym)*8+2)=1  THEN  LET colest=1
3040 IF POINT (xm*8,(21-ym)*8+3)=1 THEN  LET colalien=1
3050 INK 7: PRINT AT ym,xm;"!"
3060 PAPER 0: PRINT AT ym,xm;" "
3070 IF colest=1 THEN  PLOT xm*8,(21-ym)*8+2:
     LET colest=0
3080 IF colalien=1 THEN  GO SUB 4000
3090 IF ym=1 THEN  RETURN
3100 GO TO 3020
4000 REM Impacto
4005 BEEP .05,-10
4010 FOR a=1 TO invasores
4020 IF x(a)<>xm OR y(a)<>ym THEN  GO TO 4040
4030 LET y(a)=22: LET a=invasores
4040 NEXT a
4050 LET alienigenas=alienigenas-1: LET puntos=puntos+1
     : LET colalien=0: LET ym=1
4060 IF alienigenas=0 THEN  LET ganar=1
4070 INK 7: PRINT AT 0,26;puntos
4080 RETURN
5000 REM Ganar
5010 INK 5: PRINT AT 6,5;"BIEN HECHO TERRICOLA"
5020 PRINT AT 8,4;"ESTA VEZ HAS GANADO TU"
5030 BEEP .5,2: BEEP .5,10: BEEP 1,20
5040 PRINT AT 12,8;"PREPARATE PARA"
5050 PRINT AT 14,4;"NUESTRA PROXIMO ATAQUE"
5060 FOR d=1 TO 400: NEXT d
5070 LET alienigenas=invasores: LET altura=altura+2:
     LET ganar=0
5080 FOR a=1 TO invasores
5090 LET x(a)=a*3: LET y(a)=altura
5100 NEXT a
5110 RETURN
6000 REM Perder
6010 CLS
6020 INK 5: FLASH 1: PRINT AT 1,13;"L O S"
6030 PRINT AT 3,7;"I N V A S O R E S"
6040 PRINT AT 5,13;"H A N"
```

Continuación Spectrum

```
6050 PRINT AT 7,6;"A T E R R I Z A D O": FLASH 0
6060 BEEP 2,-10
6070 INK 6: PRINT AT 10,10;"SIN EMBARGO"
6080 PRINT AT 12,9;"HAS DESTRUIDO"
6090 INK 2: FLASH 1: PRINT AT 14,15;puntos: FLASH 0
6100 INK 6: PRINT AT 16,7;"NAVES ALIENIGENAS"
6110 PRINT AT 20,1;"QUIERES ENFRENTARTE DE NUEVO?"
6120 INPUT a$
6130 IF a$(1)="s"  OR a$(1)="S" THEN  RUN
6140 RETURN
7000 REM Pantalla
7010 CLS
7020 FOR s=1 TO 40
7030 PLOT 8*INT (RND*32),8*INT (RND*21)+10
7040 NEXT s
7050 INK 7: PRINT AT 0,5;"INVASORES DESTRUIDOS=";puntos
7055 FOR b=1 TO 20: BEEP .1,20: NEXT b
7060 RETURN
8000 REM Inicializar
8010 PAPER 0: INK 7: BORDER 0: CLS
8020 PRINT AT 1,4;"Eres un piloto solitario"
8030 PRINT AT 2,1;"protegiendo el planeta Tierra."
8040 PRINT AT 3,1;"En unos instantes estaras bajo"
8050 PRINT AT 4,3;"el ataque de invasores del"
8060 PRINT AT 5,9;"planeta VARGON."
8070 PRINT AT 6,4;"Tu mision es evitar que"
8080 PRINT AT 7,0;"aterricen las naves vargonianas."
8090 PRINT AT 10,2;"'Z'  MUEVE LA NAVE A LA IZDA"
8100 PRINT AT 12,2;"'X'  MUEVE LA NAVE A LA DCHA"
8110 PRINT AT 14,4;"PULSA 'M' PARA DISPARAR"
8120 PRINT AT 20,2;"Pulsa cualquier tecla para"
8125 PRINT AT 21,0;"entablar combate con el enemigo"
8130 LET perder=0: LET ganar=0: LET puntos=0:
     LET colalien=0: LET colest=0: LET xp=16:
     LET altura=1: LET invasores=10:
     LET alienigenas=invasores
8140 DIM x(invasores): DIM y(invasores)
8150 FOR a=1 TO invasores
8160 LET x(a)=a*3: LET y(a)=altura
8170 NEXT a
8180 POKE 65368,BIN 00100100
8190 POKE 65369,BIN 00100100
8200 POKE 65370,BIN 01111110
8210 POKE 65371,BIN 11111111
8220 POKE 65372,BIN 10011001
8230 POKE 65373,BIN 11111111
8240 POKE 65374,BIN 01011010
8250 POKE 65275,BIN 01000010
8260 LET R$=INKEY$: IF R$="" THEN  GO TO 8260
8270 RETURN
```

ESTA NUEVA SERIE CONSTITUYE UNA INTRODUCCIÓN AL ARTE DE PROGRAMAR ORDENADORES. CADA LIBRO ENSEÑA CÓMO ESTRUCTURAR UN PROGRAMA EN RUTINAS, Y AL MISMO TIEMPO EXPLICA Y ANALIZA LO QUE SE LE PIDE AL ORDENADOR QUE HAGA Y POR QUÉ.

¡Y ADEMÁS ES DIVERTIDO! CADA LIBRO CONTIENE COMO MÍNIMO UN JUEGO DE ORDENADOR, APASIONANTE Y ORIGINAL, ESTRUCTURADO DE TAL MODO QUE PUEDE AMPLIARSE O REDUCIRSE, DE ACUERDO CON EL NIVEL O LOS DESEOS DEL USUARIO.

TÍTULOS DE LA SERIE

Gráficos - Iniciación al basic - El banco de datos

Animación - Sonido - Gráficos móviles