

club NACIONAL DE USUARIOS DE LOS ZX

BOLETÍN
INTERNO
DEL
CLUB

ENTER

La privatización de programas
en el ZX-81 (pág. 1)

ENTER

Concurso de Programas para
el ZX Spectrum (pág. 11)

ENTER

Apuntes de programación
ZX-81 (I) (pág. 4)

ENTER

Soft-Bank (pág. 12)

ENTER

Dos rutinas para conservar
gráficos en memoria (pág. 5)

ENTER

Spectruc (pág. 14)

ENTER

“Quiniela ahorro” (ZX-81) y
“Reloj” (ZX Spectrum) (pág. 9)

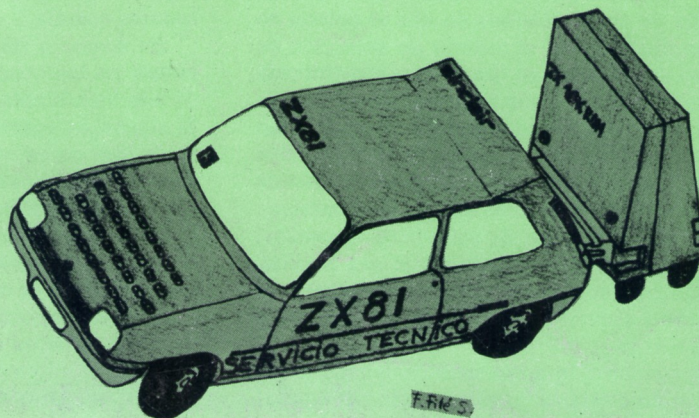
ENTER

El “Quantum Leap”
(pág. 17)

ENTER

“Seguridad”
(ZX Spectrum)
“Ordenación”
(pág. 11)

sinclair
ZX Spectrum



N.º 10 - 1984

Editorial

Como de costumbre, con retraso, llega a vuestras manos este último boletín del año 1983. Espero que su contenido sea suficiente motivo para apaciguar vuestros ánimos y haceros comprender la cantidad de trabajo que supone su confección. Éste es el «rollo» de siempre, pero como es la verdad, no queda otro remedio que volver a soltarlo. Como sabéis, Josep Oriol tiene un trabajo nada descansado: Ventamatic. Por esta razón, el que se os dirige en esta ocasión, como habéis podido comprobar, es otra persona. Con vuestro permiso me presento:

Soy Vicente Mendoza Lama, tengo 21 años y llevo desde los 16 años metido hasta el cuello en el mundo Informático.

A partir de este número, seré el encargado de elaborar los boletines de este Club, y espero llegar a ser vuestro amigo.

No sólo notaréis este cambio sino muchos más. El más importante, a mi entender, es la gran entrada en el boletín del ZX-Spectrum. El motivo es muy sencillo, casi todos vosotros lo poseéis ya o estáis en vías urgentes de adquirirlo. Además, iguala ya el nivel de ventas del ZX-81 en menos de la mitad de tiempo de aparición en el mercado. Esto no quiere decir que olvidemos al ZX-81 ¡ni mucho menos! Primero porque los programas en BASIC son fácilmente adaptables y segundo porque hay que respetar las «canas».

Los Bancos de Pruebas

También observaréis secciones que pretenden ser Bancos de Pruebas, tanto de software (Softbank) como de hardware, donde en esta ocasión he realizado un artículo sobre joysticks, que espero que aclare las ideas de muchos sobre este tema; y si lo creéis conveniente, revisaremos libros y revistas. Ya pienso que sería muy interesante.

Las colaboraciones

Como segundo paso os comentaré mi idea del asunto de las colaboraciones. Yo pienso que un Club debe ser el punto de encuentro de conocimientos, experiencias y realizaciones de sus socios. La mejor manera de conseguirlo es el encuentro personal, pero en España no existe todavía un parque informático suficientemente grande como para que esto se produzca fácilmente. Sin embargo, los usuarios disponemos de estos boletines que pueden servirnos de pizarra donde queden reflejadas nuestras ideas. Todo esto está muy bien, pero para que se convierta en realidad, primero hemos de concienciarnos de que el boletín no es una revista de gran tirada donde el criterio de selección es muy estricto sino que es un magazine de tirada corta y restringida entre un grupo de amigos, que es lo que tenemos que ser todos. Por estas razones el mundo informático español se encuentra en manos de unos pocos atrevidos que, a fuerza de errores, han aprendido lo que saben. El resto se «corta» porque la supuesta «élite» cierra sus puertas. Bien, esto no es así ni debe continuar.

Me gustaría recibir colaboraciones de auténticos «novatos». Con ellas crearemos una sección (que quiero que sea amplia), donde cada colaboración será comentada por nosotros y por vosotros. Así es como hemos aprendido todos: equivocándonos y corrigiendo errores.

En cuanto a los programas que juzguemos completos serán publicados en la sección de colaboraciones y podrán ser «incentivados» según los baremos que rigen en la actualidad, siempre que lo indiquéis claramente.

En resumen, todas las colaboraciones serán publicadas (siempre que el tema no sea repetitivo), en una u otra sección. Y recordad que no se trata de ninguna competición.

Un ruego que quiero haceros en este sentido: ya que habéis de enviar la renovación de la suscripción (así lo espero), podéis acompañar a ésta un comentario acerca del nuevo estilo del boletín y, si es posible, una colaboración.

Sigo metiéndome con los que se autocalifican como «novatos». Si observáis la sección Softbank, encontraréis que dar una explicación y una opinión de un programa, es muy fácil y ocupa poco tiempo. Bien, si no os atrevéis a mandar un programa, coged un juego de los que tengáis y me enviáis una revisión de él. El caso es colaborar y cada uno puede aportar algo.

Todo esto que os comento lo hago de cara a que entre todos hagamos el Club, opinando y comentando sin temores porque así conseguiremos muchas más ventajas.

Entre estas ventajas puede estar el aumento del número de páginas del boletín, así como su periodicidad. Yo estoy intentando convencer a Josep Oriol para que se publique el boletín mensualmente y está de acuerdo si vosotros apoyáis la idea. Espero que lo hagáis.

Sorteo de programas para quienes envíen colaboraciones

De entre las colaboraciones no incentivadas que se reciban, se sortearán en cada boletín (sea mensual o bi-mensual) 2 programas para el Spectrum o el ZX-81 a elegir, independientemente de si se publican o no. Además, espero que al recibir cada boletín me mandéis vuestra opinión de él y vuestras votaciones para los rankings. Cada mes la colaboración más votada pasará a una lista, de la cual al final del año se escogerá un ganador, el cual recibirá una gran sorpresa y si su colaboración es de categoría, la comercialización.

Votaciones sobre programas comercializados en España o en Inglaterra

Otro ranking que hemos de establecer, de cara a aclarar ideas a quien le interese, es el de programas que se comercializan en el mercado español o inglés. Esta sección la hemos abierto realizando una encuesta entre los componentes de Ventamatic, pero esperamos urgentemente vuestras votaciones. Podéis mandar cuantas queráis.

En cuanto a las colaboraciones del presente boletín (que no entran en las condiciones establecidas anteriormente), he procurado incluir todas las que constaban en el archivo del Club. Si alguno de vosotros considera que me he olvidado de él, por favor que no se moleste, pero sucede que muchos temas son repetidos. Sobre este asunto, aconsejo a los «matemáticos» que con ligeras modificaciones conviertan ese programa que sólo entienden unos pocos, en un programa educativo, cosa que no será demasiado complicada y le dará una valoración infinitamente superior sobre la misma base. No es mala idea. ¿Eh?

Y no os doy más la paliza, sólo rogaros nuevamente que perdonéis la tardanza del boletín y animaros a que entre todos hagamos que la informática se desarrolle en nuestro país como debe ser. Es una tarea de todos y tu pequeña colaboración puede ser de lo más importante.

Un saludo y hasta muy pronto.

UNA NUEVA ETAPA

Con esta edición se inicia una nueva etapa del Boletín del Club Nacional de Usuarios del ZX. Crece en dimensiones, cambia la presentación, cambia la composición pero se acentúa la trayectoria. Y se acentúa porque ha sido la que ha permitido iniciar esta nueva etapa. Es una trayectoria fundamentada en la colaboración de cuantos la han impulsado.

El cambio de presentación puede hacer más atractivo el contenido. Sin embargo, este contenido ha de responder igualmente al afán de mejora que nos ha llevado a iniciar esta nueva etapa. La colaboración de todos ha de contribuir a alcanzar cuanto antes los objetivos que, desde ahora, quedan proclamados: convertir en más provechoso para todos el intercambio de experiencias entre cuantos las aportan al Club Nacional de Usuarios del ZX.

La privatización de programas en el ZX-81

por Miguel A. Lerma

En este artículo hablaré sobre ciertos trucos que permiten ocultar un programa o parte de él, con el fin de evitar que el usuario tenga acceso al listado del mismo.

Para comprender bien los trucos de que voy a hablar, es conveniente estudiar primero los capítulos finales del Manual (del Cap. 24 en adelante, sobre todo el 27), pues haré referencia a la forma en que se almacenan en la memoria las sentencias de un programa BASIC.

A fin de posibilitar el estudio del contenido de la memoria, debe introducirse el siguiente programa antes de iniciar los experimentos que propongo:

```
9900 REM "LISTADO DE CM"
9910 INPUT X
9920 PRINT X,
9930 LET A=PEEK X
9940 LET B=INT (A/16)
9950 LET C=A-16*B
9960 PRINT CHR$(B+28); CHR$(C+28)
9970 LET X=X+1
9980 GOTO 9920
```

Para hacerlo funcionar, pulsar: RUN 9900 (NL) e introducir a continuación la

dirección (en decimal) a partir de la que se desea un listado. Por ejemplo, introduciendo la dirección 16509 (principio del programa BASIC), se obtiene:

	comentarios:
16509 26	} n.º línea: 26ACh=9900 (10)
16510 AC	
16511 11	} Long. línea: 11h=17 (10 bytes)
16512 00	
16513 EA	REM
16514 0B	"
16515 31	L
	(etc.)

El listado del contenido de la memoria se da a dos columnas. En la de la izquierda aparece la dirección (decimal) de cada byte, y a la derecha su contenido (en hexadecimal).

Los dos primeros bytes contienen el número de la primera línea en hexadecimal y en forma HIGH-LOW (primero el byte más significativo y luego el menos significativo): 26ACh=9900(10).

Los dos bytes que siguen contienen la longitud (hexadecimal, en forma LOW-HIGH) de la línea de programa: 0011h=17(10). Esta longitud se cuenta desde el siguiente byte (EAh) hasta el NL (código 76h) del final inclusive.

A continuación viene el código de la sentencia REM (EAh), y luego el texto codificado de la misma.

Debe destacarse que:

— Toda línea de programa acaba con un símbolo de NL (76h=118(10)).

— Al final del programa hay dos símbolos de NL consecutivos, uno correspondiente al final de la última línea y otro al principio del archivo de pantalla («display file»).

— Los números (pero no las «cadenas» numéricas) aparecen siempre seguidos del símbolo 7Eh (código de «number») y cinco bytes con dicho número codificado de una forma especial (en código "ZX", que no debe confundirse con «código máquina»). Por ejemplo, el número 1 en código ZX es: 81 00 00 00 00. Así, este número aparecerá en un programa BASIC de la siguiente manera:

```
1D → código máquina para el n.º 1
7E → código de «number»
81 }
00 } → código ZX para el n.º 1
00 }
00 }
```

EXPERIMENTO N.º 1

Introducir (sin borrar el programa de listado de CM):

```
2 REM
10 PRINT 1
20 PRINT 2
30 PRINT 3
40 STOP
```

Este programa se limita a imprimir en la pantalla:

```
1
2
3
```

Y se detiene en la línea 40.

El listado de este programa se entorpece cambiando en la línea 2 el código de REM (EAh) por un NL (76h=118(10)).

Antes de hacerlo, listar en CM:

```
RUN 9900 (NL)
16509 (NL)
```

Se obtiene:

```
16509 00 } línea n.º 2
16510 02 }
16511 02 } longitud: 2 bytes
16512 00 }
16513 EA REM
16514 76 NL
..... ..
```

Ahora pulsar: POKE 16513,118 (NL)

Volver a listar CM como antes y comprobar que EAh en la dirección 16513 se ha cambiado por 76h.

La presencia de dos NL seguidos son interpretados por la máquina como el final del programa al hacer el listado. En efecto, al pulsar LIST se obtiene:

```
2
```

y nada más. El resto del programa queda oculto.

(pasa a la pág. siguiente)

La privatización de programas en el ZX-81

(viene de la pág. anterior)

Este método para ocultar el programa tiene el inconveniente de que no impide el listado a partir de la línea n pulsando LIST n. Hay que comprobarlo pulsando LIST 10.

EXPERIMENTO N° 2

Pretendemos ahora ocultar las líneas 10 y 20 del programa que nos ocupa. De momento estas líneas pueden listarse pulsando LIST 10 ó LIST 20. Moviendo el cursor ($\uparrow \downarrow$) no se pueden listar, pero se pueden editar (EDIT).

El truco que impide el listado y edición de dichas líneas consiste en incluirlas como parte de la línea 2, modificando convenientemente la longitud de ésta. En principio la línea 2 mide 2 bytes (uno para REM, ahora cambiado en NL, y otro para el NL final). Si le añadimos la longitud completa de las líneas 10 y 20 obtendremos un total de 28 bytes. Comprobar esto viendo que la línea 30 (la siguiente a la 20) comienza en la dirección 16514, de modo que:

$$16541 - 16512 = 28 \text{ bytes}$$

La longitud de la línea 2 está almacenada en los bytes 16511-2, en forma LOW-HIGH. Así pues, pulsar:

POKE 16511,28 (NL)

Listar (en CM) y comprobar que el contenido del byte 16511 es ahora 1Ch (=28₍₁₆₎).

Compruébese que en estas condiciones LIST 10 y LIST 20 producen un listado a partir de la línea 30. Mover el cursor ($\uparrow \downarrow$) y comprobar que tampoco es posible editar estas líneas. Para ver mejor esto, conviene hacer desaparecer nuevamente el programa pulsando: POKE 16513,234. Esta operación reintroduce REM en el lugar donde habíamos metido el NL extra, y hace visible otra vez todo el programa. Ahora listar (LIST) y mover el cursor hacia abajo y hacia arriba. Hay que comprobar que el cursor salta hacia abajo de la línea 2 a la 30, sin detenerse en la 10 ó la 20.

Otras sugerencias:

— Poner el cursor en la línea 2 y editar (EDIT). Sólo obtenemos: 2 REM.

— Pulsar: 10 (NL).

Esta operación no borra la línea 10, aunque coloca el cursor en ella. Editar (EDIT). ¿Qué ocurre? (desde luego, no se edita la línea 10).

Si se intenta borrar la línea 2, con ella desaparecerán la 10 y la 20, pero aconsejo hacer esto sólo si la curiosidad es superior a la pereza que dará luego volver a dejar las cosas como

estaban para continuar con el experimento siguiente.

Aunque el efecto obtenido al modificar la longitud de la línea 2 permite ocultar parte del programa, aún queda un inconveniente por superar. Para verlo, pulsar RUN. ¿Qué ocurre?

Las líneas 10 y 20 han quedado incorporadas a la línea 2 no sólo a efectos de listado, sino también a efectos de ejecución del programa, de modo que dichas líneas no pueden editarse, pero tampoco serán ejecutadas. Parece que el remedio ha sido peor que la enfermedad. Ni siquiera funciona correctamente el programa pulsando RUN 10.

EXPERIMENTO N° 3

Aún no está todo perdido. Existe un nuevo truco que permitirá ejecutar las líneas 10 y 20 sin cambiar los efectos sobre el listado que hemos obtenido anteriormente. Para ello introducimos la siguiente línea:

1 POKE 16425,158

Púlsese RUN y compruébese que ahora sí son ejecutadas las líneas 10 y 20. Si se desea impedir el listado, introducir el NL extra POKE 16540,118 (la dirección ha variado al introducir la nueva línea).

El efecto de la línea 1 es modificar el contenido de la variable del sistema NEXTLIN. Esta variable, situada en los bytes 16425-6 (en forma LOW-HIGH) señala a la siguiente línea del programa que debe ser ejecutada. Mientras el ordenador interpreta la línea 1, el contenido de NEXTLIN es 409Ah=16538₍₁₆₎, dirección del primer byte de la línea 2. Después de ejecutar la línea 1, el contenido del byte bajo de NEXTLIN pasa a ser 158₍₁₆₎=9Eh, de modo que esta variable del sistema señalará ahora a la dirección 409Eh=16542₍₁₆₎, donde se encuentra el principio de la línea 10. El efecto de la línea 1, por tanto, es realizar un salto a la línea 10.

Debe tenerse en cuenta que el mismo procedimiento habrá de usarse para cualquier salto dentro de la zona protegida, pues las sentencias GOTO no funcionarán ahí. Si fuera necesario cambiar los dos bytes de NEXTLIN, el salto deberá realizarse en dos etapas, una por cada byte.

Desde luego, no sólo las sentencias GOTO son inútiles dentro de la zona protegida, sino también las GOSUB y las FOR-NEXT, lo que constituye una limitación muy seria. La razón de que tales sentencias no marchen bien dentro de la zona protegida está en que el aparato busca el destino de un salto comprobando uno por uno y desde el principio los números de las líneas del pro-

grama hasta hallar el que corresponde al destino del salto. Pero las operaciones de protección ocultan de tal manera las líneas de la zona protegida que la máquina no puede encontrarlas cuando las busca de esta manera.

PROTECCIÓN CONTRA GRABACIONES

Hemos visto un procedimiento que permite ocultar un programa o parte de él a efectos de listado aunque con serias limitaciones (concretamente en la realización de saltos).

La privatización real de un programa exige no sólo protegerlo contra listados, sino también impedir que pueda ser grabado.

Existe un método definitivo para obtener un programa totalmente protegido contra listados y grabaciones. Consiste en diseñar el programa en código máquina de modo que su ejecución comience automáticamente al cargar (el Cap. 16 del Manual explica cómo hacer para que un programa sea ejecutado automáticamente). Si el programa ha sido diseñado sin retorno a BASIC, el control del teclado puede quedar tan limitado como se quiera, de modo que sean imposibles las operaciones de listado, edición y grabación. Aunque, claro está, nada puede impedir la reproducción de un programa de cinta a cinta.

Si el programa está en BASIC, puede introducirse una pequeña rutina en CM que haga algo esencial al principio o a lo largo de la ejecución del programa principal. La protección contra grabaciones se puede conseguir entonces de la siguiente manera (no voy a dar los detalles de cómo hacer esto): la rutina en CM puede grabarse perfectamente la primera vez si está contenida en una sentencia REM o en la zona de variables pero, tras la carga, se puede ordenar una llamada automática a una parte de la rutina que la copie detrás del RAMTOP (previamente desplazado) o en alguna zona desocupada más allá de la zona de variables, y luego la borre de su lugar original. Las llamadas a la rutina en CM desde el programa BASIC deberán hacerse al lugar donde haya sido trasladada automáticamente. La protección contra grabaciones está asegurada porque el contenido de la memoria por encima de la zona de variables no se graba. La parte en BASIC sí se puede grabar, pero será inoperante al faltarle la rutina en CM.

Seguramente se pueden idear otros trucos semejantes a los aquí indicados, pero dudo que alguno sea tan contundente como diseñar el programa en CM y sin retorno a BASIC.

Un típico juego de laberinto y perseguidores

(ZX-81 - 16 K)

Este es un típico juego de laberintos y perseguidores, pero aquí no hay que comer a los monstruos. Simplemente hay que escapar de ellos. El juego consiste en evitar ser comido por uno de los cinco «ogros» perseguidores durante el máximo tiempo posible aunque a la máxima velocidad no va ser mucho.

El programa está dividido en dos partes: una en código máquina que es la que dibuja el laberinto, mueve las piezas y demás, y otra parte en BASIC que te hace las preguntas de velocidad, etc., coloca al principio los monstruos y te controla el tiempo.

La parte en código máquina es una sentencia REM de 339 bytes de longitud.

Las teclas de control del movimiento son: para arriba (1234567890); para abajo (ZXCVBNM.); para la derecha (YUIOPHJKL N/L); para la izquierda (QWERTASDFG).

CLUB NACIONAL DE USUARIOS DE LOS ZX

En la página 7 se incluye un formulario de suscripción para recibir este Boletín y obtener las ventajas ofrecidas.

PROGRAMA EN BASIC

```

10 LET X=USR 16719
20 PRINT AT 13,16;"+"
30 LET L=PEEK 16396+256*PEEK 16397
40 LET X=16523
50 FOR N=1 TO 5
60 LET A=INT (RND*30)+1
70 LET B=INT (RND 20)+1
80 PRINT AT B,A;
90 IF PEEK (PEEK 16398+256*PEEK 16399)<>0
    THEN GOTO 60
100 PRINT "O"
110 LET B=(B 33)+A+1+L
120 LET A=B/256
130 POKE X,(A-INT A) 256
140 POKE X 1,INT A
150 LET X=X 2
160 NEXT N
170 PRINT AT 21,0;"SELECCIONAR VELOCIDAD
    (1 A 20)"
180 INPUT A
190 POKE 16557, INT(50/A)
200 PRINT AT 21,0;"PULSAR N/L PARA COMEN
    ZAR!!!!!!!!!"
210 INPUT I$
220 PRINT AT 21,0;"!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    !!!"
230 PAUSE 0
240 RAND USR 16543
250 LET T=INT ((65535-PEEK 16436-PEEK 16437*256)/50)
260 PRINT AT 21,0;"HAS DURADO!";T;"!SEGUNDOS"
270 FOR N=1 TO 100
280 NEXT N
290 PRINT AT 21,0;"QUIERES JUGAR OTRA VEZ
    (S/N)
300 INPUT I$
310 IF I$="N" THEN STOP
320 CLS
330 RUN
    
```

PROGRAMA EN CÓDIGO MÁQUINA

```

CC 01 00 21 00 FF FF DF FF 00 00 00 00 00 00 00 00
00 00 00 00 FF FF FF FF FF FF FF FF FF FF 2A
0C 40 11 BE 01 19 22 7B 40 36 15 01 00 04 0B
78 B1 20 FB 3E 8B 32 05 41 32 2F 41 3E 95 32
09 41 32 2A 41 CD BB 02 7D 2F 6F E6 81 28 05
11 21 00 18 1C 7D E6 18 28 05 11 DF FF 18 12
7D E6 00 28 05 11 01 00 18 08 7D E6 06 28 17
11 FF FF 2A 7B 40 19 7E FE 80 28 0B 2A 7B 40
36 00 19 22 7B 40 86 15 06 05 2A 8B 40 ED 5B
95 40 36 00 19 7E E6 81 28 1B FE 01 C8 3A 82
40 CB 07 32 82 40 E6 06 06 83 6F 26 40 5E 23
56 ED 53 95 40 18 D6 22 8B 40 36 3A 05 41
C6 02 32 05 41 32 2F 41 3A 09 41 C6 02 32 09
41 32 2A 41 10 B9 C3 AB 40 80 21 76 41 7E FE
FF C8 0F 0F 0F CD 64 41 7E CD 64 41 23 18
EE E6 0F A7 47 3A 4E 41 28 03 D7 10 FD D6 80
32 4E 41 C9 F0 F0 35 1F 02 16 21 31 11 F1 11
41 2F 0F 21 31 11 71 71 11 41 25 17 11 17 16
21 B1 11 11 C1 21 1D 1C 11 21 11 51 B1 61 11
21 11 5D 61 11 21 11 A1 11 B1 11 2E 1F 41 21
71 11 81 21 46 1F 01 16 21 11 21 11 E1 11 21
11 21 16 1F 04 11 21 61 11 A1 11 61 2F 06 18
21 F0 D1 2F 0F F0 F0 30 FF
    
```

Joan Domingo

Apuntes de programación ZX-81

(I)

por Patricio Sánchez Valls

● FUNCIÓN INKEY\$

Esta función permite reconocer la tecla que es presionada en el teclado en el momento en que la función se auto-ejecuta.

Hasta este momento, para entrar un número, una letra o un grupo de letras, hemos tenido que parar el programa, teclearlo y luego presionar NEWLINE para que continúe su ejecución.

Una nueva función — INKEY\$ — nos facilita efectuar esto más rápida y uniformemente, pero tiene como limitación el que para que pueda emplearse, el programa necesita ejecutarse en el modo de funcionamiento SLOW.

Cuando el ZX-81 se encuentra con INKEY\$, instantáneamente verifica cada tecla del teclado. Si una tecla es presionada, con o sin el empleo de la tecla SHIFT, el carácter correspondiente es puesto dentro de una variable literal simple denominada INKEY\$. Los caracteres de control tales como EDIT, GRAPHICS, RUBOUT, etc., se imprimirán como un interrogante.

Pruebe el siguiente programa:

```
10 IF INKEY$ <> "" THEN GOTO 10
20 IF INKEY$ = "" THEN GOTO 20
30 PRINT INKEY$;
40 GOTO 10
```

Laborioso método de escribir, ¿no le parece? Además, cuando desee escribir un espacio el programa se parará.

En todo caso sólo pretendíamos demostrar que INKEY\$ no le espera a Ud., por lo que no precisa de NEWLINE para que se efectúe cualquier cosa.

Es probable que Ud. se haya formado la idea de que INKEY\$ es una cosa efímera o futil. (Siempre que Ud. mencione INKEY\$ en un programa se produce algo nuevo, por lo tanto tendremos que haber puesto a punto algunas tretas para hacer uso de INKEY\$) pero esperamos convencerle de que no es así.

● RAMIFICACIÓN DE UN PROGRAMA

Vea a continuación un programa de funcionamiento uniforme y rápido de elección de ramificación:

```
10 PRINT "PARO O MARCHA?"
20 PRINT,, "PRESIONE P O M"
30 IF INKEY$ = "M" THEN GOTO 200
40 IF INKEY$ = "P" THEN GOTO 100
50 GOTO 30
100 PRINT,, "TE HAS PARADO"
110 STOP
200 PRINT,, "ESTAS EN FUNCIONA-
MIENTO"
```

A casi a toda prueba (presione cualquier tecla que desee, excepto BREAK,) y el ZX-81 continuará dando vueltas en torno a las líneas 30, 40 y 50 hasta que la letra P o M sea tecleada.

● REGISTRO PERMANENTE DE INKEY\$

En el programa anterior el empleo de INKEY\$ es ocasional, ya que luego la tecla introducida se pierde pero, a menudo, necesitamos realizar un registro permanente del dato introducido por el teclado, en cuyo caso el programa puede ser como sigue:

```
10 PRINT "PRESIONA CUALQUIER
TECLA"
100 IF INKEY$ <> "" THEN GOTO 100
110 IF INKEY$ = "" THEN GOTO 110
120 LET A$ = INKEY$
130 PRINT "HAS TECLEADO "; A$
```

Una pequeña explicación:

La línea 100 mantiene el programa en posición de espera para darle la oportunidad de introducir la siguiente tecla por la instrucción de la línea 110 y registrar en forma permanente la tecla tocada para ser introducida en la variable A\$ mediante la instrucción de la línea 120.

Pruebe este programa eliminando la línea 100 primero, luego vuelva a escribir dicha línea y elimine, por el contrario, la línea 110 y observe que en un caso no escribe siempre la tecla tocada y que en el otro escribe dicha tecla más de una vez.

Así pues empezará a comprender nuestra intención cuando antes hemos hablado de trucos.

Pero volviendo al programa inicial, ejecútelo nuevamente y cuando se haya parado el programa, teclee en forma directa:

```
PRINT INKEY$ (ha desaparecido)
y PRINT A$ (sigue ahí, en la memoria del ZX-81).
```

Con unas pocas modificaciones, podemos usar INKEY\$ para introducir hileras de letras de cualquier longitud específica. Véase sino el siguiente programa:

```
10 PRINT "TECLEE UNA PALABRA DE
TRES LETRAS"
20 LET A$ = ""
100 FOR J= 1 TO 3
110 IF INKEY$ <> "" THEN GOTO 110
120 IF INKEY$ = "" THEN GOTO 120
130 LET A$= A$ + INKEY$
140 NEXT J
150 PRINT "SU PALABRA FUE "; A$
```

A pesar de que Ud. deba hacer trampas con la instrucción INKEY\$ para introducir letras de longitud no específica, se mantiene una pequeña ventaja con respecto al uso de la instrucción INPUT.

● ¿QUÉ HAY RESPECTO A LOS NÚMEROS

Si utiliza de nuevo el anterior programa y teclea 123, el resultado parece igual a un número pero es realmente la hilera de caracteres "123". Por lo tanto Ud. no puede realizar ninguna operación matemática con ellos. Afortunadamente, el BASIC del ZX-81 provee una función que transforma tales caracteres en números. Cambie y amplíe su programa introduciendo:

```
150 PRINT,, "HILERA(STR$)",
"VALOR(VAL)A$"
160 PRINT A$, VAL A$
170 GOTO 10
```

y trate de introducir toda clase de hileras, incluyendo algunas tales como:

ABC (el programa se parará, ya que ahora sólo acepta caracteres que tengan un valor)

9.5 (el valor es 9.5)
9+8 (el valor es 17)
9/4 (el valor es 2.25)
9**8 (el valor es 43046721)

Acabamos de descubrir la mayor parte de las reglas de la función VAL que son:

- 1.º Si una hilera está contenida enteramente por caracteres que puedan ser usados en una expresión u operación aritmética, la función VAL de dicha hilera manejará tal expresión y dará la respuesta. Los caracteres apropiados son:
 - números
 - nombres de variables previamente definidas (p. ej., A, si A ha sido previamente definida por LET A=)
 - Operandos (+ - * /)
 - El punto (.)
 - Funciones (**, SIN, COS, etc.)
 - Paréntesis (para establecer prioridad operandos).
- 2.º Cualesquiera otros caracteres pararán el programa con un código de error C/n ó 2/n.
- 3.º Usted puede mantener un registro permanente de la hilera VAL colocándola dentro de una variable numérica tal como:

LET A = VAL A\$

El ZX-81 provee asimismo otra función que es exactamente la inversa de VAL y que se denomina STR\$

STR\$ 809 = "809"
STR\$ 123 = "123"

Aunque parezca un momento lógico para hacer mención de la función STR\$, lo haremos en un capítulo aparte.

A continuación facilitamos el bien conocido programa de adivinar un número «pensado» por el ordenador para seguir ilustrando las instrucciones INKEY\$ y VAL; advirtiéndole que es necesario más de 1 K de memoria RAM.

```

10 PRINT "ADIVINA EL NUMERO QUE
    ENTRE 0 Y "; "99 HE PENSADO.
    TIENES 8 OPCIONES"
100 LET C = INT (RND*90+10)
130 FOR J = 1 TO 8
135 PRINT
140 PRINT "PRUEBA NO ";J; " ? "
150 LET P$ = ""
200 FOR K = 1 TO 2
210 IF INKEY$ <> "" THEN GOTO 210
220 IF INKEY$ = "" THEN GOTO 220
230 PRINT INKEY$;
240 LET P$=P$ + INKEY$
250 NEXT K
310 LET P=VAL P$
320 IF P=C THEN GOTO 500
330 IF P>C THEN GOTO 370
340 PRINT "ES DEMASIADO BAJO"
350 GOTO 400
370 PRINT "ES DEMASIADO ALTO"
400 NEXT J
450 PRINT "PENSE EL NO "; C
460 STOP
500 PRINT "CORRECTO"
510 PRINT "ACERTADO EN ";J;
    " VECES"
  
```

(pasa a la pág. 6)

DOS RUTINAS PARA CONSERVAR GRÁFICOS EN MEMORIA

con 16 K se pueden almacenar más de 42 pantallas

Las dos rutinas que os envío, aunque puedan utilizarse independientemente una de la otra, una es complementaria de la otra.

La utilidad de estas rutinas es poder conservar en memoria: dibujos, gráficos, laberintos..., etc., pero con la diferencia de que la cantidad de las pantallas a conservar son más del doble que con el sistema convencional. Por ejemplo, con 16 K se pueden almacenar más de 42 pantallas, la única condición que deben tener es que han de estar formados a base de «pixels» mediante PLOTs o bien con cualquiera de los 16 caracteres gráficos del ZX81 que no sean los «grises». Aunque parezca muy restringido, va muy bien para cualquier representación gráfica y, sobre todo, para dibujos animados, ya que son muchos los dibujos conservados.

El primer programa reduce el contenido de la pantalla a 352 bytes de memoria y tiene una longitud de 59 bytes.

```

11 ØØ ØØ 2A ØC 4Ø 3E 16 Ø6 1Ø 32 21 4Ø 23 D5
7E 16 ØØ CB 7F 28 Ø2 16 Ø8 E6 Ø7 82 Ø7 Ø7 Ø7
Ø7 57 23 7E 1E ØØ CB 7F 28 Ø2 1E Ø8 E6 Ø7 83
82 D1 12 13 1Ø DA 23 3A 21 4Ø 3D 2Ø CE C9
  
```

Antes de ejecutar el programa habrá que poner la dirección en donde se quiere guardar la imagen:

```

INPUT X
POKE 16515,X-256 INT ( X/256)
POKE 16516,INT ( X/256)
  
```

Un buen sitio para almacenarlo es buscando la dirección de la variable RAMTOT y a ella restarle los 352 bytes que necesitamos y colocar RAMTOP a la nueva dirección. Si se quiere almacenar más de una imagen habrá que hacer lo mismo para cada una.

La última sentencia del programa que traza lo que queramos conservar será: RAND USR 16514.

El segundo programa pone en pantalla lo que tengamos en memoria:

```

11 ØØ ØØ 2A ØC 4Ø 3E 16 Ø6 1Ø 32 21 4Ø 23 1A
F5 E6 FØ 1F 1F 1F 1F CB 5F 28 Ø2 EE 88 77 23
F1 E6 ØF CB 5F 28 Ø2 EE 88 77 13 1Ø E2 23 3A
21 4Ø 3D 2Ø D6 C9
  
```

En esta rutina habrá que poner la dirección en donde está almacenado el dibujo. Si está sola valdrá la misma dirección que en la anterior, pero si la ponéis a continuación serán: 16574 y 16575.

Para hacer dibujos animados habrá que poner un bucle FORINEXT para hacer de retardador y cambiar la dirección de la próxima imagen.

Joan Domingo

Apuntes de programación ZX-81

(viene de la pág. anterior)

Someramente el programa se explica de la siguiente forma:

línea 100 - determina el número «pensado» aleatoriamente.

línea 130 - lazo que permite entrar hasta ocho pronósticos y que imprime en la línea 450 el número «pensado», caso de no haberlo acertado previamente.

línea 150 - variable «vacía» sobre la que se cargará el valor de los números que se inserten como pronóstico y que además actúa como puesta a cero de pronósticos previos.

línea 200 - lazo que permite introducir los dos números y posteriormente cargarlos en la variable P\$.

líneas 210, 220 - «Truco» que permite introducir los dos números mediante la instrucción INKEY\$.

línea 240 - suma de los números entrados a la variable P\$.

línea 310 - asignación del valor de dichos números (P\$) a la variable P.

líneas 320, 330 - ramificación del programa según la relación existente entre el número «pensado» y el introducido como pronóstico.

líneas 340, 370, 500 y 510 - mensajes de resultados y de pronósticos efectuados.

Ahora haga la modificación siguiente:

```
210 PAUSE 40000
220 POKE 16437,255
```

y verá cómo el programa funciona igual, aunque con la desventaja del parpadeo de la pantalla debido a la instrucción PAUSE.

Así pues llegamos de nuevo a la conclusión de que la función INKEY\$ permite aceptar un dato externo (como en INPUT) pero sin parar el programa, requiriendo de un cierto freno en los casos expuestos, lo cual debe efectuarse mediante una de las dos formas descritas, excepto cuando INKEY\$ se utiliza en bucle cerrado de exploración de teclado para la detección de una determinada tecla (p. ej., IF INKEY\$ = "S").

● PROGRAMA DE EJEMPLIFICACIÓN DE INKEY\$ Y CHR\$

Ahora otro programa de ejemplificación de las instrucciones INKEY\$ y

CHR\$ que volveremos a comentar después del mismo:

```
10 LET T=50
30 LET A$=CHR$ INT (RND*10 + CODE "0")
40 PRINT A$
45 PAUSE T
50 POKE 16437,255
60 LET B$=INKEY$
70 IF B$="Q" THEN GOTO 200
80 IF A$=B$ THEN GOTO 150
90 PRINT "ERROR"
100 LET T=T*1.1
110 GOTO 20
150 PRINT "OKEY"
160 LET T=T*0.9
170 GOTO 20
200 SCROLL
210 PRINT "PUNTUACION ALCANZADA = ";INT(500/T)
```

Comentarios:

línea 10 - asignación del valor 50 al PAUSE de la línea 45, es decir 1 segundo de pausa que es aumentado en un 10 % por cada fallo en la línea 100 y disminuido en un 10 % caso de acertarlo por ramificación a la línea 160 vía la línea 80 del programa.

línea 30 - asignación de un número aleatorio comprendido entre 28 y 37 (código 0=28, código 1=29, 30=2, etc., e introducción directa dentro de la variable A\$ del carácter cuyo código corresponde al establecido aleatoriamente y cuya impresión se lleva a cabo por la línea 40.

línea 45 - pausa que «acondiciona» la entrada de datos por el teclado y que en este caso se utiliza además como «cambio de velocidad».

línea 60 - introducción de un número por medio de INKEY\$.

línea 70 - ramificación que permite puntuar a través de la línea 210, previa subida de una línea de la pantalla, el ratio aciertos/tiempo empleado.

línea 80 - ramificación en el caso de coincidencia entre el número «pensado» y el introducido.

línea 90 - ramificación con mensaje en el caso de no coincidencia entre el número «pensado» y el introducido.

Observe la gran similitud existente entre estos dos últimos programas, y para que se divierta otro poco, estudie este sencillo programa:

```
10 IF INKEY$ = "" THEN GOTO 10
15 LET A = INT (RND*24+5)
20 PRINT AT 11,A; "HOLA"
30 IF INKEY$ <> "" THEN GOTO 30
40 PRINT AT 11,A; " "
50 GOTO 10
```

El empleo de INKEY\$ es común en programas de juegos, en los que es necesario provocar el desplazamiento del cursor mediante las teclas ← ↑ → (5, 6, 7 y 8 con Shift presionado)

Si no vea los dos ejemplos siguientes:

```
10 REM "TEST DE TECLA TOCADA"
20 LET X = INT (RND*64)
30 LET Y = INT (RND*64)
40 LET X1=X
50 LET Y1=Y
60 PLOT X,Y
70 IF INKEY$ = "S" THEN STOP
80 LET TECLA = CODE INKEY$ - 32
90 LET X = ( TECLA = 1 ) * -1 + ( TECLA = 4 ) + X
100 LET Y = ( TECLA = 2 ) * -1 + ( TECLA = 3 ) + Y
110 IF X > 63 OR Y > 43 THEN GOTO 140
120 UNPLOT X1,Y1
130 GOTO 40
140 LET X = X1
150 LET Y = Y1
160 GOTO 70
170 STOP
```

que sirve para desplazar el punto según se teclee ← ↑ → o este otro ejemplo de programa:

```
10 REM DIBUJO
20 LET X = 0
30 LET Y = X
40 PLOT X,Y
50 LET T = CODE INKEY$ - 32
60 IF T = 27 THEN CLS
70 LET X1 = ( (T=1) * -1 ) + (T=4)
80 LET Y1 = ( (T=2) * -1 ) + (T=3)
90 IF (Y1+Y) > 43 OR (X1+X) > 63 THEN GOTO 50
100 LET X = X + X1
110 LET Y = Y + Y1
120 GOTO 40
```

que le permitirá realizar un dibujo, a la vez que borrar la pantalla siempre que lo desee presionando la tecla V cuyo código de carácter es el 59 al que restado 32 da el valor 27 por el cual en la línea 60 se ordena CLS.

Empieza a no ser tan complicado, ¿verdad?

Básicamente consiste en escribir una línea de programa tal como:

```
10 LET C$ = INKEY$
```

que le dice al ordenador que vea qué tecla está siendo presionada y asigne

a la variable literal C\$ el carácter correspondiente para, tal como hemos dicho antes, usar convenientemente algunos trucos y realizar toda clase de cosas. Por ejemplo, vamos a dirigir un tanque indistintamente a la izquierda o a la derecha, usando la tecla 5 para la izquierda y la 8 para ir hacia la derecha (es útil recordarle que esto lo haremos sin presionar la tecla SHIFT como es normalmente obligado). ¿Cómo lo vamos a hacer? Vamos a decirle al ordenador que lea INKEY\$ y que ajuste la posición de imprimir según sea un 5 ó un ocho lo que se le introduzca. Veamos:

```

1 LET B= 15
2 LET A= 12
3 LET C$= INKEY$
6 IF C$ = "5" THEN LET B=B-1
7 IF C$ = "8" THEN LET B=B+1
10 PRINT AT A-2,B+2; "space-space
inverso-space inverso-space inverso-
gráfico 7-gráfico 7-gráfico 7-gráfico
7-space"
20 PRINT AT A-1,B; "space-gráfico
G-gráfico F-gráfico G-gráfico F-gráfico
G-gráfico F-gráfico G-space"
30 PRINT AT A,B; "space-gráfico 8-O
inversa-O inversa-O inversa-O inver-
sa-O inversa-gráfico 8-space".
40 GOTO 3

```

esperamos que habrá comprendido que debe utilizar los gráficos en las líneas PRINT que se señalan, puesto que tales caracteres no existen en la máquina de escribir que utilizo en este momento. Observe el invisible final en ambos extremos de las líneas PRINT (space). El problema con ello consiste en que si Ud. presiona las teclas 5 u 8 demasiadas veces, el dibujo se saldrá de los extremos de la pantalla. Veamos si puede modificar el programa para prevenir esto, mediante la adición de líneas de

programas tales como IF a < 0 THEN... o similares.

Si Ud. mantiene una tecla presionada, este programa sigue ejecutándose y el tanque moviéndose.

A menudo esto es un inconveniente y frecuentemente lo que Ud. realmente quiere es un programa que responda sólo a los cambios en la entrada de INKEY\$, o como mínimo que se renueve cuando se presione el teclado.

En tales casos, modifique el programa anterior como sigue:

```

3 IF INKEY$ <> "" THEN GOTO 3
4 IF INKEY$ = "" THEN GOTO 4
5 LET C$ = INKEY$

```

¿que este truco ya lo había visto? Es cierto, pero trataremos de explicarlo nuevamente:

La línea 3 tiene como efecto el mantenerlo suspendido todo en la línea 3 si Ud. mantiene presionada la tecla. Cuando Ud. levanta el dedo el programa se transfiere a la línea 4 y se mantiene allí. Cuando Ud. presiona una nueva tecla o represiona la anterior, se consigue vía libre...

Nuevamente le recomendamos cuidado con lo que Ud. le diga al ordenador mediante INKEY\$. Ud. sólo puede introducir números y luego utilizarlos como tal si simultáneamente emplea la función VAL que convierte tales números (que hasta dicho momento son simples caracteres) en verdaderos números que podrán ser manejados en operaciones aritméticas. Asimismo para iniciar la ejecución de un programa se requiere presionar NEWLINE y alguna vez (según como se haya estructurado el programa) el ZX-81 leerá esto como una entrada de INKEY\$ tratando de convertirlo en un número, y como ello no es posible el programa se romperá o, peor aún, si no se presionara ninguna tecla dará como valor la cadena vacía (""). Esto

puede parecer un «puzzle» hasta que Ud. no lo lleve a cabo y sortee tales dificultades. De nuevo, puede protegerse contra estos problemas mediante el adecuado uso de IF... THEN..., pero esto requerirá una cantidad de memoria que no siempre está disponible, cuanto menos utilizando sólo 1 K de RAM.

● UN MINITRATAMIENTO DE TEXTOS

Vamos a ver seguidamente una variante interesante de un programa anterior:

```

5 IF INKEY$ <> "" THEN GOTO 5
10 IF INKEY$ = "" THEN GOTO 10
15 LET A$= INKEY$
20 IF CODE A$= 113 THEN GOTO 50
25 IF CODE A$= 114 THEN CLS
30 IF CODE A$= 115 THEN PRINT
35 IF CODE A$= 118 THEN PRINT "";
40 IF CODE A$< 112 OR CODE A$>
118 THEN PRINT A$;
45 GOTO 5
50 COPY

```

y que podríamos titular como un minitratamiento de textos. Observe que como la tecla espacio es asimismo la función BREAK que pararía el programa, hemos tenido que encargar a la tecla NEWLINE la realización del espacio. Compruebe asimismo que la tecla → (código de carácter 115) se encarga de imprimir en el principio de una nueva línea, la tecla ← (código de carácter 114) se encarga de borrar la pantalla y la tecla ↑ (código de carácter 113) ordena la copia de la pantalla mediante la impresora (que naturalmente debe previamente haber sido conectada).

BOLETÍN DE SUSCRIPCIÓN - club nacional de usuarios de los ZX

NOMBRE

APELLIDOS

CALLE N.º

POBLACIÓN D.P.

PROVINCIA

FORMA DE PAGO:

- ADJUNTO TALÓN
- ENVÍO GIRO POSTAL N.º

(Enviar a la dirección del dorso)

- Deseo suscribirme al Boletín, de los números 11 al 16, ambos inclusive, por el **precio de 2.500 ptas.**
- Deseo suscribirme al Boletín, de los números 11 al 16, ambos inclusive, y recibir además tres cassettes con al menos seis programas para Spectrum en cada uno. **Precio conjunto de Boletín + los tres cassettes: 4.000 ptas.**

4 en raya ZX-81 (1 K)

Es el conocido juego del 4 en raya y se tiene como oponente el ZX81. El programa consta de dos partes. La que se podría decir que es el juego en sí está realizada en código máquina y la parte que dibuja el tablero y hace las preguntas está en BASIC.

El juego es muy sencillo. Se basa en entrar el número de la columna en la cual pondremos la ficha «0», la cual irá todo lo abajo posible, y en intentar poner cuatro seguidas en cualquier dirección. Pero no penséis que es fácil ganar al ZX81.

Habría que desconectar las ampliaciones de memoria y luego poner una línea REM de 230 bytes de longitud.

Por falta de espacio, sólo tiene 1 K. Cuando alguien gana, sale la línea en negativo pero no se para; o sea que, cuando acabe la partida, al nuevo INPUT habrá que contestar con un STOP para empezar una nueva partida.

Joan Domingo

PROGRAMA EN CÓDIGO MÁQUINA

```

0C 01 0A 0B 2A 0C 40 23 3E 00 85 6F 7E A7 C0
CD E9 40 36 34 CD F4 40 3A 7B 40 FE 10 3E B4
28 3C 2A 0C 40 23 3E 01 32 7C 40 3D 32 21 40
06 08 23 E5 C5 CD E9 40 CD F4 40 3A 7B 40 FE
0F C1 28 19 E1 3A 7C 40 3C 32 7C 40 10 E5 2A
0C 40 23 3A 7A 40 85 6F CD E9 40 36 3D C9 C1
3E BD 06 04 4F 79 77 1A 85 6F 10 F9 C9 11 0B
00 19 7E A7 28 FB ED 52 C9 11 82 40 0E 04 1A
ED 44 47 7D 80 80 80 6F E5 C5 ED 4B 0C 40 A7
ED 42 38 4B 09 C1 06 04 3E 00 32 7B 40 C5 7E
FE 08 28 3C E6 07 47 3A 7B 40 80 32 7B 40 C1
1A 85 6F 10 EA 3A 7B 40 FE 10 28 33 FE 0F 28
2F FE 0C 28 0A FE 0A 28 06 FE 08 28 02 E6 07
47 3A 21 40 B8 30 0D 78 32 21 40 3A 7C 40 32
7A 40 18 01 C1 E1 1A 85 0D 20 A2 13 1A FE 2A
20 92 C9 E1 C9

```

PROGRAMA EN BASIC

```

10 FOR N= 0 TO 5
20 PRINT "a!!!!!!!"
30 NEXT N
40 PRINT "(10 a)",, "!12345678",,
"EMPIEZAS TU?(S/N)"
50 INPUT I$
60 IF I$="S" THEN GOTO 90
70 POKE 16505,INT (RND*8)+SGN PI
80 RAND USR 16588
90 INPUT A
100 IF A<1 OR A>8 THEN GOTO 90
110 POKE 16523,A
120 RAND USR 16518
130 GOTO 90

```



QUINIELA AHORRO (ZX)

por Julián Benítez Fernández

Se trata de un programa que reduce una quiniela con 7 dobles (128 apuestas) a una combinación en reducción al 13 que sólo tiene 16 apuestas. El programa incluye instrucciones. Sin fallos en la inicial, el trece es seguro y existen posibilidades de sacar un catorce. Si alguien se hace millonario que avise.

```

5 PRINT "          <<<QUINIEL
A AHORRO>>>"
10 PRINT "          INTRODUCZA
LOS PRONOSTI-","          COS U
NO POR UNO, Y ENT. ","          N
OTRA:"
15 PRINT "          LOS PRONO
STICOS SENCI- ","          LLOS
HAY QUE METERLOS ","          D
OS VECES."
20 DIM A$(14)
30 DIM B$(14)
40 FOR N=1 TO 14
50 INPUT A$(N)
55 PRINT AT N,0;"<"VAL "N";">
":AT N,5;A$(N)
60 INPUT B$(N)
65 PRINT AT N,7;B$(N)
80 NEXT N
90 PRINT AT 14,12;"<PULSA UNA
TECLA>"
100 IF INKEY$="" THEN GO TO 10
0
110 CLS
115 PRINT "          <<<16 APUESTA
S>>>"
120 LET N=1
130 GO SUB 8000
140 FOR X=0 TO 29 STEP 4
150 PRINT AT N,X;A$(N);" ";B$(N
);" "
160 NEXT X
170 LET N=N+1
180 GO SUB 8000
190 FOR X=0 TO 29 STEP 8
200 PRINT AT N,X;B$(N);" ";B$(N
);" ";A$(N);" ";A$(N);" "
210 NEXT X
220 LET N=N+1
230 GO SUB 8000
240 FOR X=0 TO 29 STEP 16
250 PRINT AT N,X;B$(N);" ";B$(N
);" ";B$(N);" ";B$(N);" ";A$(N);
" ";A$(N);" ";A$(N);" ";A$(N);"
"
260 NEXT X
270 LET N=N+1
280 GO SUB 8000
290 FOR X=0 TO 15 STEP 16
300 FOR Y=16 TO 29 STEP 16
310 PRINT AT N,X;A$(N);" ";A$(N
);" ";A$(N);" ";A$(N);" ";A$(N);
" ";A$(N);" ";A$(N);" ";A$(N);"
"
320 PRINT AT N,Y;B$(N);" ";B$(N
);" ";B$(N);" ";B$(N);" ";B$(N);
" ";B$(N);" ";B$(N);" ";B$(N);"
"
330 NEXT X
340 NEXT Y
350 LET N=N+1
360 GO SUB 8000
370 FOR X=0 TO 15 STEP 16
380 FOR Y=16 TO 29 STEP 16
390 PRINT AT N,X;A$(N);" ";A$(N
);" ";B$(N);" ";B$(N);" ";B$(N);
" ";B$(N);" ";A$(N);" ";A$(N);"
"

```

```

400 PRINT AT N,Y;B$(N);" ";B$(N
);" ";A$(N);" ";A$(N);" ";A$(N);
" ";A$(N);" ";B$(N);" ";B$(N);"
"
410 NEXT X
420 NEXT Y
430 LET N=N+1
440 GO SUB 8000
450 FOR X=0 TO 15 STEP 16
460 FOR Y=16 TO 29 STEP 16
470 PRINT AT N,X;A$(N);" ";B$(N
);" ";A$(N);" ";B$(N);" ";B$(N);
" ";A$(N);" ";B$(N);" ";A$(N);"
"
480 PRINT AT N,Y;B$(N);" ";A$(N
);" ";B$(N);" ";A$(N);" ";A$(N);
" ";B$(N);" ";A$(N);" ";B$(N);"
"
490 NEXT X
500 NEXT Y
510 LET N=N+1
520 GO SUB 8000

```

```

530 FOR X=0 TO 15 STEP 16
540 FOR Y=16 TO 29 STEP 16
550 PRINT AT N,X;A$(N);" ";B$(N
);" ";B$(N);" ";A$(N);" ";A$(N);
" ";B$(N);" ";B$(N);" ";A$(N);"
"
560 PRINT AT N,Y;B$(N);" ";A$(N
);" ";A$(N);" ";B$(N);" ";B$(N);
" ";A$(N);" ";A$(N);" ";B$(N);"
"
570 NEXT X
580 NEXT Y
590 LET N=N+1
8000 IF N>14 THEN PRINT AT 19,1
5;"FIN": STOP
8005 IF A$(N)<>B$(N) THEN RETUR
N
8010 FOR X=0 TO 29 STEP 2
8020 PRINT AT N,X;A$(N);" ";B$(N
)
8030 NEXT X
8040 LET N=N+1
8050 GO TO 8000
8060 SAVE "QUINIELA" LINE 8070
8070 PRINT AT 2,0;"          <<<QUINI
ELA AHORRO>>>","PARTIMOS DE
UNA QUINIELA CON 7 ","DOBLES<1
28 APUESTAS> Y";
8080 PRINT "OBTENEMOS","UNA DE
16 APUESTAS. CASO DE TE- ","NE
R PLENO EN LA PRIMERA, TENDRIA",
"13 ACIERTOS SEGUROS Y POSI
BI","LIDAD DE TENER 14 ";
8090 PRINT "EN LA SEGUNDA. ",",,,"
"PULSA ENTER."
8100 IF INKEY$="" THEN GO TO 81
00
8105 CLS
8110 GO TO 5

```

RELOJ (ZX SPECTRUM)

por Albert Canet y José Jover

Como su nombre indica, el programa funciona como si de un reloj digital se tratara. Permite entrar el horario actual o dejar como tiempo el que el ordenador lleva enchufado. Está almacenado a partir de la línea 9000 por si se quiere usar como subrutina.

```

09000 REM  ** RELOJ **
9010 REM POR ALBERT CANET Y JOS
E JOVER.
9500 PRINT "RELOJ": PRINT "====
": PRINT : PRINT : PRINT "A =TIE
MPO DESDE CONEXION RED": PRINT :
PRINT "B =PUESTA EN HORA"
9501 PAUSE 0
9502 IF INKEY$="a" THEN CLS : G
O TO 9505
9503 IF INKEY$="b" THEN GO TO 9
660
9504 GO TO 9501
9505 LET dividendo=INT ((65536*P
EEK 23674+256*PEEK 23673+PEEK 23
672)/50)
9510 LET divisor=60
9520 LET minutos=INT (dividendo/
divisor)
9530 LET segundos=dividendo-(min
utos*divisor)
9540 LET a$=STR$ segundos
9550 LET horas=INT (minutos/60)
9560 LET rminutos=minutos-(60*ho
ras)
9570 LET b$=STR$ rminutos
9580 LET c$=STR$ horas
9590 IF horas=24 THEN LET horas
=0: GO SUB 9640: CLS
9600 LET c$=STR$ horas
9610 PRINT AT 0,2-LEN c$;c$;" ";
TAB 5-LEN b$;b$;" ";TAB 8-LEN a$
;a$
9620 GO TO 9505
9630 STOP
9640 POKE 23672,0: POKE 23673,0:
POKE 23674,0
9650 RETURN
9660 REM Puesta en hora
9670 CLS : PRINT "PUESTA EN HORA
DE RELOJ": PRINT "=====
"
9680 INPUT "Horas ? : ";rhora
9690 PRINT "Horas = ";rhora
9700 INPUT "Minutos ? : ";rminu
to
9710 PRINT "Minutos = ";rminuto
9720 INPUT "Segundos ? : ";rsegun
dos
9730 PRINT "Segundos = ";rsegund
os
9740 LET x=(rhora*60*60*50)+(rmi
nuto*60*50)+(rsegundos*50)
9760 LET y=INT (x/65536)
9780 LET z=x-(65536*y)
9800 LET u=INT (z/256)
9820 LET w=z-(256*u)
9840 PRINT : PRINT : PRINT "INDI
QUE EL MOMENTO ": PRINT "PULSE C
UALQUIER TECLA"
9850 PAUSE 0: BEEP .25,25
9860 POKE 23674,y: POKE 23673,u:
POKE 23672,w
9870 CLS : GO TO 9505

```


COLABORACIONES

**SEGURIDAD
ZX Spectrum**

por **Carlos Berche Codina**

El programa, como su nombre indica, es una protección anti-pirata, pues debe teclearse el código, y si no se hace así o se pulsa otra tecla, el programa se destruye. En la línea 4, el código "4359" puede cambiarse por cualquier otro, incluso con distinta extensión.

```

1 CLEAR : PAPER 1: INK 4: BRI
GHT 1
2 OUT 254,2: POKE 23624,18: P
RINT AT 5,7:"C.B.C. @ Software";
FLASH 1:AT 11,1;"-INTRODUZCA CO
DIGO SEGURIDAD-"
3 INPUT a$
4 IF a$="4359" THEN GO TO 20
5 RANDOMIZE USR 0
20 CLS : POKE 23624,56
21 REM PROGRAMA PRINCIPAL
22 REM Autor: Carlos Berche
23 REM
9999 SAVE "Nombre del Programa."
LINE 1

```

ORDENACIÓN

por **Ernesto Navarro Aponte**

Este programa (realizado en BASIC «estándar»), utiliza un algoritmo descubierto por el autor para ordenar una lista de elementos. Consiste en tomar un

```

10 DIM A (9)
20 DIM B (9) : REM Reserva
30 FOR N=1 TO 9
40 INPUT A (N)
50 PRINT A (N); " ";
60 NEXT N : REM Entrada de números
desordenados

70 FOR N=1 TO 9
75 LET T=0
80 FOR M=1 TO 9
90 IF A (N) > A (M) THEN LET T = T+1
100 NEXT M
110 LET B (T) = A (N)
120 NEXT N : REM Comparación de cada
número con los demás e
incremento del contador
si es igual o mayor

130 FOR N=1 TO 9
140 PRINT B (N); " ";
150 NEXT N : REM Imprimir la lista
ordenada

```

SIGUE ABIERTO

**CONCURSO DE PROGRAMAS
PARA EL ZX SPECTRUM**

Una vez más os recordamos que sigue abierto el concurso de programas para el ZX-Spectrum, con el plazo de admisión abierto hasta el 30 de abril de 1984. Os detallo las condiciones de nuevo por si os interesa:

Sólo entrarán en el concurso los programas que se consideren inicialmente comercializables por VENTAMATIC, siendo descartados automáticamente los que no cumplan esta condición.

Las Bases son las siguientes:

1) Los programas pueden ser realizados en BASIC, en código máquina o en mezcla de ambos, aunque tendrán más posibilidades los que estén realizados en código máquina.

2) La selección del programa premiado dentro de cada categoría se realizará sólo cuando se hayan presentado por lo menos 10 programas en la misma que hayan superado la fase inicial de selección. El plazo de admisión de originales se irá prolongando un mes cada vez que expire hasta completar la selección de 10.

3) El importe de los premios será entregado a cuenta de los royalties a percibir por el autor, que otorgará licencia de producción en exclusiva en todo el mundo a VENTAMATIC.

4) Los programas que hayan superado la fase inicial de selección serán inmediatamente comercializados por VENTAMATIC, en las condiciones establecidas habitualmente, aun cuando no se haya decidido el fallo del concurso, por no haberse cumplido el párrafo núm. 2, bien por cualquier otra causa.

5) Los originales pueden entregarse personalmente o enviados por correo a VENTAMATIC. En cualquier caso se acusará recibo, y deben ser acompañados de las señas completas del remitente en todas las partes del envío (hojas, cinta, etc.) y de un sobre suficientemente grande para devolverlo en caso de que no supere la fase inicial de selección.

6) Si no superase esta fase inicial, el programa podrá publicarse en el boletín del Club o en cualquier otra forma, con el consentimiento escrito del autor. En este caso percibirá la remuneración prevista para las colaboraciones.

7) Los programas deberán ir acompañados de explicaciones claras y detalladas sobre los mismos: uso, realización, características especiales, así como de un comentario publicitario resumido.

8) Cualquier duda sobre estas bases puede ser consultada telefónicamente o personalmente con VENTAMATIC.

Los premios establecidos son los siguientes:

— Único premio al mejor video-juego: 100.000 ptas. + 50.000 ptas. en accesorios y programas a elegir.

— Único premio al mejor programa educativo: 60.000 ptas. + 30.000 pesetas en accesorios y programas a elegir.

— Único premio al mejor programa de gestión: 60.000 ptas. + 30.000 pesetas en accesorios y programas a elegir.

— Único premio al mejor programa de juegos inteligentes: 60.000 ptas. + 30.000 ptas. en accesorios y programas a elegir.

— Único premio al mejor programa de utilidades: 60.000 ptas. + 30.000 ptas. en accesorios y programas a elegir.

elemento del listado, compararlo con todos los demás e ir contando cuántas veces es mayor o igual que los demás. El número de veces coincide con el puesto de orden que tomará en el listado ordenado.

Me explico: supongamos un listado de nueve números colocados al azar: 3, 72, 67, 19, 50, 83, 48, 2, 1. Comparemos el primero con todos los demás y contemos de cuánto es mayor o igual: el tres será mayor que el 1, que el 2, e igual al tres o sea tres, luego ocupará el lugar tercero definitivamente. Otro ejemplo, el 67 es mayor que el tres, que el dos, que el uno, que el 19, que el cincuenta, que el cuarenta y ocho, e igual al 67, luego hemos contado siete, su lugar será el séptimo.

Para ordenar éstos y otros números, haríamos el programa adjunto.

Soft-Bank

MANIC-MINER - (Bug-Byte) Spectrum 48K

Tal vez el mejor video-juego para ordenador del mercado, aunque suene exagerado. El juego consiste en llevar al «minero Willy» a través de una profunda mina, hacia la superficie. La mina tiene 20 pisos, o sea, que tendrás que pasar ¡20 pantallas! En cada una de estas pantallas se encuentran dispersadas una serie de llaves, que han de recogerse para poder salir de ese piso hacia el superior. Para dificultar la tarea (faltaría más) en cada pantalla van surgiendo una serie de enemigos que no tienen nada que ver con los clásicos marcianitos, sino que son cosas y animales de la vida real (pingüinos, focas, canguros, teléfonos, skay-labs, etc.).

Con una calidad de gráficos que igualan pocos programas y una dificultad bien distribuida, por lo que ocupa casi todos los 48K de memoria disponibles, se puede decir que con este juego hay para rato.

No se comercializa en España, pero Ventamatic ofrece la posibilidad de importarlo.

JET-PAC - (Ultimate) - Spectrum 16K

Otro de los grandes. Sencillo y muy entretenido. Consiste en recoger los trozos de tu nave, ensamblarlos y luego rellenarla de fuel para poder salir de la superficie lunar, todo esto evitando chocar con todo tipo de criaturas. Así de sencillo y emocionante, pues a esta sencillez de diseño, añade rapidez en los movimientos y una gama de gráficos que tumban de espaldas hasta al más veterano. Es decir, cada vez que se consigue despegar del Planeta, se pasa a otra situación similar donde los marcianitos son diferentes y cada diez pantallas cambia la forma de la propia nave.

De su presentación sorprende gratamente el dibujo que se nos muestra en pantalla durante la carga, y que nos avisa de lo que viene a continuación.

Otra de las grandes sorpresas es comprobar que para hacer un buen programa no hace falta llegar a llenar las 48K de memoria sino que con 16K hay suficiente.

No se comercializa en España, pero está prevista su importación en breve por Ventamatic.

PSSST - (Ultimate)

Otro divertido juego de acción que sigue la tradición de esta compañía in-

Novedades

Ventamatic presenta junto con este boletín una serie de nuevos productos que eran esperados por muchos de vosotros e interesarán a los que no los conozcan.

Entre estas novedades destacan dos libros imprescindibles y de gran utilidad. Estos libros son:

GUÍA PRÁCTICA DE BASIC DEL ZX-81 Y ZX-SPECTRUM

Autor: Ramón Rovira.

Con un estilo fácil, dinámico y muy entretenido, este libro introduce al neófito en las particularidades del BASIC, llevándolo hasta lo más profundo de su estructura de una forma tan amena que, sin apenas darse cuenta, el lector aprende un lenguaje de programación como si leyera una novela.

Se analizan en este libro todas las peculiaridades del BASIC, detallando sus características con relación específica al BASIC de Sinclair, que es el que usan tanto el ZX-81 como el ZX-Spectrum.

El precio de este libro es 1.200 ptas.

GUÍA PRÁCTICA DE CÓDIGO MÁQUINA DEL ZX-81 Y ZX-SPECTRUM

Autor: Joan Sales Roig.

Por fin, tras larga espera, sale a la luz el prometido libro de código máquina de nuestro artista en esta materia.

Con el estilo clásico de Joan Sales, que todos vosotros conocéis, esta guía instruye al no iniciado sobre programación en código máquina. Las explicaciones son sencillas y claras y se ilustran con una buena cantidad de ejemplos, lo que hace de este libro un manual completo de código máquina.

El precio de este libro es 1.200 ptas.

Hay un aspecto a destacar de ambos libros: son enteramente españoles y no traducciones de libros ingleses como el resto de libros que circulan por el mercado. Esto, aunque no lo parezca, es una ventaja, puesto que en nuestro país vemos la informática con diferentes peculiaridades con respecto a los ingleses.

El resto de novedades que presenta Ventamatic, son programas para el ZX-Spectrum. Estos programas son:

ANDROID 2 - (Vortex) - 48K

Si os gustó el Android 1, seguro que os gustará este otro. Basado en una idea parecida, este programa le añade al anterior el estar realizado en tres dimensiones. Esto lo convierte en un juego extraordinario con gran dinamismo y muy aditivo. Los gráficos son excelentes y el movimiento está muy bien conseguido.

El juego consiste en guiar a un androide a través de innumerables laberintos hasta llegar a destruir la base de los aliens, sus enemigos, que se encargarán de dificultarle la tarea taponando las salidas hacia otros laberintos e intentando chocar con el androide.

glesa, Ultimate Play the Game, con unos gráficos sencillamente perfectos, con un excelente sonido y una de las mejores presentaciones que he visto. Y todo esto en 16K, por supuesto en código máquina.

El juego consiste en maniobrar a nuestro robot (Robbie), defendiendo tu planta de los gusanos, avispas y otros insectos que quieren devorarla, usando el correspondiente spray para cada uno de ellos. Existen tres tipos diferentes de spray y cada uno eliminará sólo al insecto adecuado obligándote a estar siempre cambiando de spray. Además, también van saliendo aleatoriamente otros aditivos para cuidar tu planta, como regaderas,

abono o raquetas anti-mosquitos que puntúan mucho.

Si consigues aguantar el ataque de los insectos, la planta irá creciendo hasta florecer, y tras una melodía se comienza de nuevo pero con mayor número de insectos para molestarte.

Considero que es muy buen juego y nada complicado, pero muy entretenido.

Importado por Ventamatic.

E.T. - (Investrónica)

Este juego simula las desventuras de E.T. para conseguir llegar a su casa a

VENTAMATIC

Es un juego realmente bueno, que no defraudará y resulta muy entretenido y divertido.

CONTEXT - (Ventamatic) - 48K

Aunque este gran procesador de textos no es una novedad, sí lo es que añade a su gran calidad la posibilidad de usarlo con cualquier impresora. Así es, esta última versión sale ya preparada y lista para su uso con cualquiera de las impresoras del mercado.

MASTERFILE - (Campbell) - 48K

El conocido programa de archivo-base de datos está ya disponible en España y, por supuesto, en castellano. Esta estupenda base de datos para el Spectrum, deja libres al usuario 32K por archivo. La configuración de estos archivos es libre y permite crear hasta 26 campos por registro y cada campo puede tener hasta 128 caracteres. Es la mejor base de datos que existe para el Spectrum y es un gran logro que Ventamatic la haya traducido y la comercialice en España.

CONTABILIDAD PERSONAL - (Ventamatic) - 48K

Otro gran programa de Oscar Domingo, que está derramando todo su buen saber en programas de gestión para el Spectrum, para demostrarnos que no es sólo un juguete, y el resultado que está consiguiendo es sorprendente, llegando a superar los programas similares de Inglaterra con amplio margen a su favor. En este programa el usuario puede llevar una contabilidad que sobrepasa los límites de la contabilidad personal, posibilitando incluso balances. A mi entender, este programa es más que suficiente para que pequeños comerciantes y profesionales puedan llevar una contabilidad clara y muy organizada. Ni que decir tiene que para la contabilidad de casa es ideal.

CONTROL DE STOCKS - (Ventamatic) - 48K

Este último programa (por el momento), completa un «paquete» imprescindible para cualquier utilización seria del Spectrum. Con él se puede llevar un riguroso control de stocks totalmente profesional pudiendo definir límites de existencias, control de pedidos, facturación y cartera de clientes. Utiliza el Spectrum con 64 caracteres por línea como todos los demás programas de gestión de Oscar.

De todos estos programas realizaremos una amplia revisión en el próximo número de nuestro boletín, detallando mucho más todas las posibilidades y características de ellos.

NOTA: *Se me olvidaba destacar que los programas de gestión mencionados funcionan todos con cualquier tipo de impresora e interfase, al igual que el CONTEXT.*

través de cinco pantallas y como juego de acción. Ocupa 48K de memoria y se ha de pasar primero el cassette para leer las instrucciones y luego cargar el juego. La presentación es muy sencilla y el juego está hecho en BASIC (a pesar de las protecciones es fácil descubrirlo), por lo que su calidad deja mucho que desear y encima tener que esperar a que se carguen las 48K...

Las primeras fases transcurren por el bosque, debiendo nuestro E.T. recoger trozos de teléfono o de su nave (se representan en la pantalla como PUNTOS), para al final pasar a través de una tormenta de meteoros.

Poco más que comentar. Me imagino

que el resto lo imaginaréis. Por si acaso no podéis imaginar lo malo que llega a ser pasar por alguna tienda informática y pedir una demostración y observar sus gráficos (¿cuáles?), su sonido muy sencillo y los fallos que tiene a ser entretenido para niños pequeños, pero ni eso.

Yo, personalmente no lo recomiendo, aunque sobre gustos no hay nada escrito.

Lo comercializa Inestrónica y su precio es 1.800 ptas.

Próximamente...

Ventamatic lanzará próximamente al mercado un programa que seguramente estaréis esperando muchos de vosotros. Se trata de un Emisor-Receptor-Editor de MORSE, o sea, un tratamiento completo del Morse. Para que os hagáis una idea de lo que será os adelanto algunos detalles:

Permite: Preparar textos para emitir (5.000 caracteres).

Disponer de un archivo aparte (3.000 caracteres), en forma de mensajes fijos.

Emisión directa de lo entrado por teclado o de lo archivado.

Posibilidad de cambiar la velocidad de emisión.

Posibilidad de cambiar la velocidad de recepción.

Caracteres españoles y signos especiales de emisión.

Recibir con posibilidad de almacenar lo recibido.

Impresión de los textos por medio de la impresora.

Dar de alta, baja y modificar los textos archivados.

Emisión directa.

La parte más importante del programa está realizada en código máquina, por lo que el programa es muy impresionante (para los entendidos en la materia).

Oscar está dando los últimos retoques a un nuevo programa, que será el primer juego realmente dedicado a los niños pequeños. El programa se llama ARQUITECTO, está preparado para el Spectrum de 48K y está realizado íntegramente en código máquina.

Se trata de llevar el antiguo juego de las construcciones con piezas de madera que usábamos alguno de nosotros en nuestra juventud, al interior del Spectrum. Permite elegir el tipo de pieza a colocar y transportarla por toda la pantalla hasta colocarla donde se desee. Se pueden cambiar los colores de las piezas y hacerlas girar en cualquier sentido. También se dispone de una goma de borrar para eliminar piezas enteras o trozos de piezas.

Se dispone además de la opción para salvar el dibujo, cargar uno ya preparado y seguir construyendo sobre él y se pueden manejar las piezas con el Joystick.

Os daré más detalles en el próximo boletín.

GANADORES DEL SORTEO

● En el sorteo celebrado han resultado ganadores:

— Marín-Dámaso Cruz Caballero - Sevilla - Socio n.º 211

— Francisco Becerra Ruiz - Bolaños (Ciudad Real) - Socio n.º 849

Spectruc

En esta sección de nuestra revista, trataremos de conocer algunas de las maravillas no muy conocidas de nuestro ordenador, como pueden ser algunos POKES que no quedan muy claros en el manual de Introducción.

Cualquier comportamiento extraño que hayas descubierto con tu Spectrum, no dudes en comunicárnoslo, dirigiéndote a esta sección.

PRINT PEEK 23731. — En esta posición de memoria se almacena el número 255 si el Spectrum es de 48K, ó 127 si es de 16K.

PRINT 65561-USR 7962. — Esta instrucción devolverá el número exacto de bytes (octetos) que quedan libres en el ordenador. Estos bytes corresponden sólo al BASIC, y no tiene en cuenta el espacio que ocupan las variables. Puede ser usada en cualquier Spectrum (16K-48K).

SCROLL. — Existen varias formas de conseguir el Scroll con el Spectrum:

RANDOMIZE USR 3280. — Simula exactamente el comportamiento del ZX-81 (que posee la instrucción SCROLL), aunque su comportamiento puede ser, a veces, algo extraño. Como en el ZX-81, esta orden posiciona el cursor en la línea 21.

POKE 23692,X. — Donde X puede ser cualquier número mayor que uno y menor que 256. Esta dirección de memoria corresponde al contador de Scrolls. Su valor, normalmente es 1 (un desplazamiento). Este método debe ir siempre acompañado de una instrucción "PRINT AT 21,0;", para situar el cursor en la última línea de la pantalla.

RANDOMIZE USR 3582. — Esta instrucción es semejante al RANDOMIZE USR 3280, pero con la desventaja de que no posiciona el cursor en la línea 21, por lo que necesita después el comando "PRINT AT 21,0;".

Cualquiera de estas instrucciones, en el momento de ser usada, debe ser repetida cada vez que se desee ejecutar un Scroll.

* SCROLL es el desplazamiento hacia arriba de la pantalla que se produce, por ejemplo, cuando listamos un programa.

POKE 23609,X. — Donde X puede ser un número de 0 a 255, sin sobrepasar estos límites. Este POKE afecta a la variable del sistema llamada PIP, la cual se encarga de controlar la duración del chasquido que se produce al pulsar una

tecla. El valor de X puede ir desde 0 a 255, aumentando el chasquido según aumentemos X.

POKE 35136,0. — Este POKE es especial. No tiene ninguna utilidad usado en un programa normal, pero... para los fanáticos de los video-juegos que posean el Manic-Miner (Bug-Byte), resultará una sorpresa, puesto que introduciéndolo no perderán vidas durante el juego, lo que les permitirá llegar a pasar las 20 pantallas de que consta el juego. La manera de introducirlo es la siguiente:

En primer lugar, se ha de cargar la primera parte del programa con MERGE "", y parar el cassette. Luego se lista el programa y se introduce este POKE justo antes de la última instrucción (RANDOMIZE USR...), con los debidos dos puntos para separarlas. Seguidamente se tecléa RUN y Enter y pulsamos PLAY para cargar el juego. A partir de aquí el programa funcionará como siem-

pre, pero sin descontar vidas. Para comenzar un nuevo juego se tendrá que pulsar BREAK.

POKE 23659,0. — Esta posición de memoria corresponde a las líneas de la mitad inferior de la pantalla (zona de mensajes). Según el manual, no debe ser alterada, puesto que quebrantaríamos el sistema, pero éste es un tipo de protección de programas bastante usado, entre otros por Psion, puesto que no tiene ningún efecto si el ordenador no ha de mostrar ningún tipo de mensaje en esta zona de la pantalla, pero en el momento que ha de hacerlo (por haber pulsado BREAK, por error de carga, etc.), el sistema descubre que no tiene líneas donde imprimir ese mensaje y se queda bloqueado, dejando la pantalla negra. Esta dirección de memoria puede dar mucho más de sí, puesto que por medio de ella podemos tener acceso a esa «segunda» pantalla que esconde el Spectrum. (Este tema será ampliado en breve, en nuestro boletín.)

POKE 23658,8. — Esta dirección de memoria se utiliza, según el manual, como indicador, pero no especifica ningún dato más. Realizando este POKE dentro de un programa, conseguiremos poner en ON las mayúsculas (Caps Locks), sin necesidad de utilizar una instrucción INPUT. Su oponente es:

POKE 23658,0. — El cual retorna a la posición de minúsculas. Ambas instrucciones pueden ser usadas alternativamente, evitando al usuario de nuestro programa tener que estar pendiente de cómo escribe.

AYUDA PARA SUBSANAR UN ERROR

Es el observado en las instrucciones del programa "Caracteres gigantes" de la Enciclopedia Práctica de la Informática

Voy a intentar ayudar a Juan Gómez Martín acerca del programa de «Caracteres gigantes» de la Enciclopedia Práctica de la Informática, y a cualquiera que quiera meterlo en un Spectrum de 48K, porque hay un error en las instrucciones.

No sé si te refieres a que no consigues cargar la rutina desde la cinta de demostración al ordenador, o a que no puedes hacerla funcionar. En cuanto a cargarla, yo tampoco pude hacerlo tal como indicaba la revista porque ni siquiera sé qué signo era el que estaba entre las comillas. La rutina es la que está inmediatamente después del programa «Hardware». Hay dos, pero la primera es la rutina y la segunda una pantalla. Ninguna de las dos tiene aparentemente nombre, porque en la pantalla sólo aparece: Bytes:

Ésa es la rutina que hay que cargar con LOAD ""CODE. Después yo no tuve

ningún problema para hacerla funcionar con mi Spectrum 48K sin hacer ninguna modificación, simplemente con RAND USR 32256.

Pero si se quiere poner en el lugar de la memoria que dice la revista para los de 48K, hay que hacer el conjunto de POKE's que dicen las instrucciones pero sumándole a todas las direcciones el número 32613

Por supuesto es útil no poner en la línea 20 ...; POKE I + 4,8; ... sino POKE I + 4,CS; e incluir la variable CS en una línea anterior dentro de un INPUT, lo mismo que los demás parámetros: XX, XY, etc.

Por cierto, el programa GUS que viene un par de fascículos más adelante es muy simpático. Os aconsejo que lo entéis.

Antonio Tovar Castellanos
(Sevilla)

¿Y por qué no cuatro?

por R. Giné (113)

La cosa va de tres triángulos pitagóricos con la misma área. Ya había preparado otros dos escritos, que acabaron en la papelera, a medida que iba encontrando soluciones más elegantes. El acicate final fue la «demostración» de que no existe solución, publicada en el último boletín. El planteamiento era correcto, pero no lo eran ni el ejemplo, pues 25 es una hipotenusa posible, al ser 5×5 (7, 24, 25), ni la conclusión, pues existen soluciones, y sospecho que en número infinito.

Para empezar, es obvio que no se trata de un problema de programación sino de algoritmo, y que conviene usar algo de papel y lápiz, antes de enchufar el ZX-81.

Como considero que sería una mala pasada para los amantes de pensar, no voy a indicar el programa, y sólo daré ideas muy generales.

En los primeros intentos vi, imagino que como todos, que era preciso mucho tiempo para encontrar números pitagóricos por tanteo, aunque se pudieran eliminar algunos «a priori», que para la máquina no es nada seguro que SQR 25 sea 5, pues a lo mejor sale «casi», o «un poquitín» más de 5, y que se obtenían muchas soluciones repetidas (3, 4, 5 es lo mismo que 6, 8, 10).

Después de usar papel y lápiz, considerando las diferencias entre hipotenusa y catetos, encontré un sistema de generar números pitagóricos sin tanteo, y ¡sin efectuar raíces cuadradas! Mejorándolo un poco, con un tanteo previo muy rápido (ver si dos números son primos entre sí; aconsejo el algoritmo de Euclides), pude eliminar las soluciones «repetidas», quedándose con las básicas, aunque luego las soluciones reales sean múltiplo de las básicas.

Llegado a este punto, en pocos minutos generé 370 triángulos pitagóricos distintos. En una hora más o menos, la máquina nos ordenó según áreas crecientes (puede hacerse mucho más deprisa, pero esto lo comentaré otro día), y en otra hora encontré seis tríos distintos de soluciones, y como para muestra vale un botón, pero dos mejor que uno:

80	798	802	5040	780	5100
336	190	386	7280	540	7300
608	105	617	y 1344	2925	3219

Aquí me pregunté: ¿Y POR QUÉ NO CUATRO?, pero para ahorrar tiempo y memoria, impuse otra restricción, buscando que el área fuera múltiplo de 105 (o sea de 3, 5 y 7 simultáneamente), como es el caso de la solución dada cuan-

do se planteó el problema, y ¡hete aquí cuatro soluciones en dos minutos y sólo con 1K! He de reconocer que tuve una suerte loca, pues si hubiera empezado por el otro lado, hubieran sido 20 min., y además estuve en un tris de que se me escapara la última solución, por limitar demasiado el campo de exploración. Aquí sólo un botón, pero hay más:

5280	5544	7656
3168	9240	9768
1980	14784	14916
1792	16335	16433

Dos cosas para acabar:

1) Lo más difícil de la parte «papel y lápiz» es resolver una ecuación de 2.º grado, o sea que ánimo y al toro, digo al triángulo.

2) Otro problema: hallar triángulos pitagóricos, tales que la diferencia entre sus catetos sea uno.

Con 1K, pues solamente he usado seis líneas BASIC, y seis minutos en FAST, he encontrado 10 soluciones. La primera me da vergüenza indicarla, pero la décima, si no se me ha escapado ninguna, es:

27.304.196, 27.304.197, 38.613.965

La siguiente precisa unos siete minutos extras, pero ya tiene 9 cifras significativas, y el pobre ZX solamente puede darnos ocho en pantalla, aunque, como en sus tripas guarda un poco de reserva, él «sabe» la solución, y hace falta un poco de ingenio para obligarle a que nos la diga. Ánimo también, y hasta la próxima.

Joysticks

Este artículo pretende acercar (de verdad) el tan manipulado asunto de los joysticks, a los usuarios.

Para ello bueno será empezar recordando que lo que normalmente entendemos por joystick está compuesto de dos partes (¡qué listo!): el interfase y la palanca.

Cada una tiene una importancia con respecto al conjunto muy clara y definida. Es por esta razón por lo que, a pesar de tratarlos a la vez, los trataremos por separado.

PALANCAS.— Puesto que la salida de los interfases suelen ser de tipo estándar (se le llama tipo Atari), la palanca es totalmente optativa y existe una gran diversidad de ellas. Trataremos las más usadas en el Spectrum.

COMPETITION-PRO.— Fabricada por la compañía inglesa Kempston (que también fabrica un Interfase), es la más popular. Con una base suficientemente grande y que ofrece gran estabilidad, posee dos grandes pulsadores de fuego muy sensibles y cómodos. El bastón acaba en forma de bola y es muy manejable. Posee muelles de retorno y ofrece respuestas al interfase en 8 direcciones. En resumen, es una buena palanca, especial para «nerviosos» por su resistencia. Como único defecto puede decirse que el recorrido no es muy largo, pero esto también puede ser una ventaja.

QUICK-SHOT.— Una gran palanca, de reciente aparición en el mercado y que promete mucho. Consta de una gran base con cuatro ventosas para su buen asentamiento, donde está insertado un botón de fuego que es «optativo». El bastón podríamos definirlo como ergonómico, puesto que se adapta perfectamente al contorno de la mano. Además,

posee en la parte superior de éste el botón de fuego principal, que resulta muy manejable. Los que gustan de sacar defectos, le achacan una supuesta lentitud (yo creo que no es así), pero su respuesta es tan buena como cualquier otra. No se puede discutir que es la más cómoda.

ATARI.— De todos conocida, no es la mejor palanca de juegos, puesto que sólo incorpora un botón de fuego algo pequeño y el bastón es muy fino y no se adapta fácilmente a la mano. En funcionamiento, sus resultados son tan buenos como los de cualquier otra palanca. Tiene un recorrido bastante bueno.

Otras.— El resto de palancas compatibles, pertenecen a la gama de ordenadores del tipo del Spectrum (¡pero no de su calidad!), como pueden ser las palancas de Commodore, Dragón, etc., las cuales suelen tener características muy parecidas a las Atari.

También se han de mencionar las nuevas palancas que han aparecido en el mercado inglés y que no tienen base, es decir que sólo constan del bastón y del pulsador de disparo.

No cabe duda que la elección de una u otra palanca dependerá de los gustos de cada cual, de su vistosidad y de su facilidad de manejo, pero tratándose de elegir el Interfase, estos criterios cambian y el factor principal no es su diseño ni su vistosidad sino el software. Efectivamente, puesto que de nada sirve un Interfase de gran categoría si no existe software adecuado para ser manejado por medio de él. Siendo así, en el análisis que vamos a realizar en el próximo Boletín sobre Interfases, daremos mayor importancia al software con el que funciona que al resto de características.

CORRESPONDENCIA

— «Te envío un programa que incluye las cuatro reglas (N. R. aclaro que es educativo), +, —, X, : a nivel EGB. Creo que se podría pulir, pero a mi hijo le sirve de mucho. Me costó bastante, pues soy principiante».

Pere García Sanahuja - C/ Misericordia, 35 - 3 1 - Reus - TARRAGONA.

R. Primero te ruego que me vuelvas a enviar el programa, puesto que en la cinta que enviaste está mal grabado. (Seguramente tienes el cabezal del cassette descentrado.) Lo he intentado cargar de muchas maneras y es imposible.

Por otro lado, felicitarte y esperar que esta carta sea un ejemplo para los demás, ya que es muy meritorio mandar un programa siendo novato (¿no es eso lo que muchos piensan?). Otro punto a tu favor es que el programa es educativo, tema en el cual en nuestro país circula mucha teoría pero casi ningún programa. Y para los novatos... ¡Adelante!, que novato es sólo el que se siente idem.

— «Me decido a escribiros para preguntaros sobre los siguientes temas:

1. ¿Conocéis algún programa sobre la declaración de la renta que sepáis que funciona bien? No encuentro ninguno, ni siquiera el que anuncia Investrónica.

2. ¿Sabéis algo de los fallos del interfaz RS-232 de Sinclair al recibir datos?

3. ¿Conocéis algo del nuevo ordenador Sinclair que creo que se llama QL? Creo que es un 32 bits, y tiene 128K».

José María Villanueva - C/ Guzmán el Bueno, 100 - MADRID-3.

R. Otra carta que espero que sirva de ejemplo en cuanto a las consultas

que deseáis realizar (sin retórica y en orden). Ahora paso a contestarte de igual modo:

1. Muy fácil. Ventamatic está preparando uno que espero que te satisfaga.

2. Es verdad, el interfaz (o interfase) que lleva incorporado el Interface 1 (para micro-drives), funciona correctamente en cuanto a emitir se refiere, pero a la hora de recibir falla. Por lo visto se debe a la lentitud con que procesa las entradas, debido a que se analizan por Software (la EPROM que lleva el Interface 1 se encarga de ello). De todos modos, te repito que para el uso normal que se le da (impresora) funciona correctamente.

BOLSA DEL CLUB

— Vendo ZX-81 + 32K: 32.000 ptas. Impresora Seik. GP80: 45.000 ptas. I/F Centronics + cable: 12.000 ptas. Todo con embalaje de origen + instrucciones.

Albert Canet Aymerich - C/ Santander, 4, ático 2.º - GERONA - Teléfono 972-23 67 65.

— Vendo ZX-81 + teclado prof. Investrónica + Impresora ZX + interface para impresora + memotec alta resolución + varios programas. Todo: 25.000 pesetas.

Miquel Solé Roig - Plaza Cataluña, 31, 3.º 2.º - Sant Boi de Llobregat - BARCELONA - Telf. 93-652 36 61.

— Vendo ZX-Printer con alimentador, manual y caja con rollos de papel, con muy poco uso, por 10.000 ptas.

Pilar Cabrerizo - C/ José M.º Durán, 51 - LAS PALMAS-10.

— Vendo caja 10 diskettes (5.1/4) Verbatim, con muy poco uso, por 3.000 pesetas. Boletines Club Commodore (1-11), por 3.000 ptas. Escribir al Club. Ref. BC-1.

— Intercambio programas para el ordenador ZX-Spectrum, todos ellos en C/M 16K y 48K, más de 60 títulos en stock, grabaciones originales. Intercambio información, rutinas, accesorios, etc.

Fco. Javier Delicado Molina - Apartado 365 - MURCIA.

En esta sección se publican todos los anuncios que no son comerciales, de forma gratuita. A partir del presente número os ofrezco la posibilidad de realizar el contacto por medio del Club, para evitaros cualquier tipo de molestia o anónimos de mal gusto.

Por otro lado, algunos anuncios de los que mandáis tienen toda la pinta de ser comerciales, por lo que si os interesa su publicación os remito a las tarifas que rigen para estos casos.

3. En cuanto al QL, en el presente boletín encontraréis un artículo que espero te amplíe lo que ya sabes.

Se han recibido también algunas cartas protestando por el retraso de los boletines. Como os indico en la editorial, os aseguro que este problema estará solucionado dentro de muy poco. A pesar de ello, vuelvo a pedir os disculpas en nombre de J. O.

Especial mención, en este apartado de perdones, a:

Luis Amado Rego.

Ramón Acín García.

Alberto Garrido.

Miguel A. Lerma Usero.

Así como a todos vosotros, que espero sigáis formando parte de esta gran familia en esta nueva etapa que comienza con este boletín.

En otro orden de cosas, comprenderéis ahora por qué indico en alguna sección de este boletín que tenéis la posibilidad de permanecer en el anonimato (sólo la posibilidad, aunque personalmente no lo considero necesario).

Nuestro amigo José Oriol y nuestro colaborador Miguel A. Lerma han recibido sendos anónimos (divertidos, por cierto), que a pesar de que no les han molestado, puede que a otros socios no les agrade que les suceda algo parecido.

El anónimo (que no lo es tanto) es el mismo en ambos casos y por lo visto tiene afán de protagonismo. Espero que se digne mandarme a mí uno o me sentiré menospreciado (si puedes con tu nombre y franqueado. ¿Vale?)

• Toda la correspondencia debe ser dirigida a:
VENTAMATIC - Calle Córcega n.º 89, entlo. - BARCELONA-29

• Hay que indicar claramente las señas del remitente en todas las hojas de la correspondencia y en todo tipo de envíos

• No se devolverán los originales no publicados si no lo solicita el autor, enviando además S.A.F.

Boletines atrasados: 300 ptas. c/u.

• Se permite la reproducción total o parcial de los artículos de este Boletín citando la procedencia.

el Quantum Leap

El «tío» Clive ha vuelto a dar la campanada, como ya se esperaba. Se esperaba la campanada pero no de tal magnitud.

A primeros de febrero se presentó en Inglaterra (ingleses, privilegiados mortales), el nuevo ordenador de Sinclair: el Quantum Leap, o QL (en cristiano, salto cuántico). Este nombre explica todo sobre este ordenador puesto que realmente se trata de un gran paso dentro de este mundo informático, debido a la gran potencia del QL y su costo que es ridículo si lo comparamos con ordenadores con sus mismas prestaciones.

Pasemos a reseñar sus características técnicas:

- Microprocesador: Motorola 68008
- Estructura: 32 bits
- Teclado profesional
- Memoria: 128K
- Ampliable: hasta 640K
- Sistema operativo: QDOS (propio)
- Lenguaje: SuperBASIC
- Almacenamiento: micro-drives
- Interfaces: cartuchos, RS-232, joysticks y monitor
- Precio: 399 libras (unas 100.000 ptas.)

Una vez alucinados por estas características, pasemos a detallarlas y explicarlas:

El microprocesador 68008 de Motorola es un 32 bits con capacidad para direccionar hasta 1 Megabyte de memoria instantáneamente. Posee además un procesador 8049 para controlar el teclado, el sonido y el interfase RS-232.

El teclado es totalmente diferente a cualquier Sinclair anterior, pues se puede decir que es completo y del tipo máquina de escribir, o sea el clásico QWERTY. Se compone de 65 teclas, incluyendo ESC (escape), TABULATE (tabulador), 2 de SHIFT, CTRL (control), ALT (¿?), ENTER y CAPS LOCK; además de cinco teclas de función definibles por el usuario y de teclas de cursor independientes. Por supuesto, tanto el SPACE, como ENTER, etc., tienen un tamaño «decente». Una membrana protege el interior de cualquier intromisión.

En cuanto a la memoria, el aparato sale de fábrica con 128K, de los cuales están libres para programas 90K. En cuanto a la ampliación consiste en un cartucho que se inserta por el lateral izquierdo y que le añade 500K. Es extraño que no se haya anunciado la ampliación hasta el total de memoria que puede direccionar el microprocesador (¿la usará para otros procesos?).

El sistema operativo que utiliza el QL es el QDOS, o sea su propio sistema operativo. Según la publicidad del QL, este sistema operativo no pretende competir con los existentes sino que es especial para la familia de procesadores 68000. Es un sistema mono-usuario pero multi-tarea (hasta 20 tareas), pudiendo presentar resultados simultáneamente en diferentes lugares de la pantalla.

El lenguaje que usa es una nueva versión del BASIC creada especialmente para él. Sinclair la ha bautizado con el nombre SuperBASIC. Es una versión del BASIC ampliada para explotar al máximo al QL. Posee facilidad de procedimiento para programar en bloques definidos de forma que quede muy claro. También se anuncia la ampliación por software del conjunto de comandos, que funcionarán a la misma velocidad que los residentes en la ROM.

El almacenamiento de datos se realiza por medio de los dos micro-drives incorporados (existe la posibilidad de ampliar el número de unidades hasta 8, como en el Spectrum). Estos micro-drives, a pesar de ser iguales a los del Spectrum, permiten almacenar hasta 100K (en el Spectrum sólo 85K). La velocidad de acceso de estas unidades (en el QL) es de 3.5 segundos, y carga unos 15K por segundo. Los micro-drives van insertados en la misma carcasa que contiene el teclado y el ordenador. Son compatibles los cartuchos de los ZX-microdrives, previa formatación.

En cuanto a ports (conectores) dispone de muchos y lleva incluidos los interfaces. Estos ports son:

- ROM. — Slot para ampliación mediante cartuchos.
- RAM. — Slot para conexión de la ampliación de 500K.

Joysticks. — Posee dos conectores para palancas de juegos.

RS-232. — También posee dos conectores de este tipo.

TV. — Como es lógico, salida para televisión en UHF.

RGB. — Salida para monitor en tres modos de resolución 25 líneas de 85 caracteres; 512×256 pixels, con 4 colores y 256×256 pixels con 8 colores.

POWER. — Entrada de alimentación.

NETWORK. — Al igual que el Interface 1, permite crear un área local de comunicaciones.

RESET. — Conexión/desconexión del aparato.

MICRO-DRIVES. — Aberturas para introducir los cartuchos.

Como es lógico, todo este «cacharro» sin software no sirve para nada, por lo que Sinclair ha lanzado junto con el ordenador 4 programas. Estos programas son de gestión y orientan bastante hacia qué tipo de usuarios está dirigido el ordenador. Estos programas son:

QL Quill. — Es un procesador de textos, que según la publicidad, es fácil de usar por principiantes e imprime realmente lo que se ve en pantalla. Me permito aclarar esto: muchos procesadores de textos presentan por pantalla junto al texto, caracteres de control que luego serán enviados a la impresora (márgenes, letras con diferentes tamaños, etc.).

QL Easel. — Es un programa que permite realizar todo tipo de gráficas de negocios (de barras, histogramas, curvas de rendimiento, etc.). Aseguran que es fácil de usar y permite cambiar datos y texto automáticamente. Los gráficos no deben ser diseñados sino que con unas pocas instrucciones son creados por el programa.

QL Abacus. — Se trata de una completa hoja de cálculo. Permite hacer todo tipo de combinaciones para estudiar qué pasaría si... Para ello pueden asignarse teclas de control a un cambio y pulsándolas el programa calcula todo tipo de posibilidades. Permite tener archivadas las «ventanas» con nombre (otros usan las coordenadas). También incluye análisis de cash-flow.

QL Archive. — Es una base de datos. Permite etiquetado de fichas y formato libre de éstas, así como todas las posibilidades clásicas de una base de datos.

Todos los programas están escritos por PSION.

El Qclub

Juntamente con el ordenador, Sinclair ha creado su propio Club: el Qclub. Éste ofrece por 35 libras (unas 8.000 ptas.) uno de los programas antes citados, gratis, y un boletín bi-mensual, además de asistencia en cuanto a dudas sobre el software comercializado.

Comentario

¿Qué comentario puede hacerse de todo esto? A la espera de verlo de cerca y de probarlo, se pueden aventurar algunos comentarios que creo serán unánimes. Primero, lo que de verdad ha sido una bomba no es el ordenador por sus características, sino por su precio. Si nos ponemos a sumar todo lo que nos costaría llegar a tener, más o menos, lo mismo que el QL con el Spectrum, la sorpresa es que con menos memoria y capacidad, resulta un mismo precio. Esto da mucho que pensar. ¿Bajarán los precios del Spectrum? Sinclair ha contestado que no (¡lógico!), pero también sería lógico que sucediera. Por el contrario, alguien puede pensar que estos ordenadores están dirigidos a un usuario muy diferente. Ésta parece ser la política de Sinclair: el Spectrum para juegos y utilidades domésticas y el QL para negocios. En fin, comentadme algo si lo creéis conveniente.

Tratando al QL por sí solo representa un nuevo paso en este mundo informático que corre como un rayo. Esto es en teoría, y debe demostrarse con el paso del tiempo si los pequeños empresarios (creo que éste será su mercado), reaccionan por fin e introducen la informática en sus negocios. Me consta que muchos desean hacerlo pero los precios no son soportables para ellos. En cuanto al uso personal (doméstico), no le acabo de ver la «gracia», pensando en el precio que debe salir en España (normalmente, el doble del inglés).

Por características técnicas debería competir con el mercado del IBM PC y similares pero este mercado está muy bien cerrado a «intrusos». Además habrá que esperar al software que se haga más adelante para poder hablar con mayor conocimiento de causa (no olvidemos que el factor más importante para la compra de un ordenador es éste).

Como ya os he comentado antes, me gustaría que me explicárais vuestro parecer.

