

2023

# RETROTRIVIAL

## DOCUMENTACIÓN DEL PROGRAMA



Agustín Morales

RETROTRIVIAL

Retrotrivial es un juego de tablero, realizado íntegramente en el Basic del ZX Spectrum.

En este documento trato de explicar lo que hace cada rutina y para qué se utilizan las variables más importantes, con objeto de que sirva como material didáctico a los que quieran iniciarse o aprender un poco más de este lenguaje.

Como segundo objetivo está el de facilitar la labor aquel que quiera realizar modificaciones sobre el programa para conseguir nuevas versiones o mejoras sobre el mismo. En este sentido he procurado que el programa esté lo más estructurado posible, y que el nombre de las variables, aun a costa de consumir un poco más de memoria, recuerden la función para las que se están utilizando.

Todos los comentarios están en azul, en tanto que las líneas del programa están siempre en color negro. Los comentarios se realizan justamente antes de la rutina (nunca después)

### Comienzo del programa

```
20 REM PROGRAMA RETROTRIVIAL
30 REM BY AGUSTIN MORALES CANO

200 PRINT AT 12,5; "CARGANDO RETROTRIVIAL ..."
```

Aquí se definen las variables globales, es decir aquellas que se declaran solo una vez, y ya no se vuelven a definir, aunque se jueguen varias partidas. Para una descripción de las mismas se puede consultar el [Anexo I](#)

```
300 REM VAR. GLOBALES
310 LET FILAS=5: LET COLUMNAS=12: LET MAXJUG=4
340 LET NUMCUAD=FILAS*COLUMNAS: LET NUMJUG=2: LET NUMOPC=4: LET
ANOACTUAL=2023
342 LET NEGROSINI=20: DIM N$(MAXJUG,15)
345 REM % TIPO PREG.
355 LET P1=40: LET P2=25: LET P3=25: LET P4=5: LET P5=5
```

Las siguientes variables definen el inicio y la longitud de los DATA que se usarán para las preguntas. Esto permite que se puedan cambiar fácilmente los juegos de preguntas.

Si por ejemplo quieres renovar algún tipo de pregunta, solo tienes que hacer dos cosas:

- Modificar en las siguientes líneas las variables correspondientes para que indiquen donde comienzan las DATA y la longitud de las mismas
- Modificar las DATA correspondiente con los nuevos datos.

```
360 REM INICIO Y LONGITUD DE LOS DATA
365 LET INIJUEGOS=8000: LET MAXJUEGOS=85
367 LET INIEMPRESAS=9000: LET MAXEMPRESAS=33
```

```
370 LET INIAUTOR=9500:LET MAXAUTOR=28
380 LET INIBASIC=9700:LET MAXBASIC=12
383 LET INIASSEMBLER=9800:LET MAXASSEMBLER=12
385 LET INIPERSONAJES=9900:LET MAXPERSONAJES=12
```

Aquí se definen los UDGs:

- La eñe
- el muñequito que hace de ficha de los jugadores
- el muro que se usa cuando el jugador no acierta (es un solo UDG que se repite 4 veces)
- la hormiga para el evento Ant Attack ataque de hormigas
- Y el túnel para el evento del minero Willy

```
395 LET R$="-----"
400 FOR K=1 TO NUMJUG: LET N$(K)="JUGADOR "+STR$ K: NEXT K
402 POKE USR "N",255
```

```
405 RESTORE 450
412 FOR L=1 TO 13
415 READ K$
420 FOR K=0 TO 7: READ DATO:POKE USR K$+K,DATO:NEXT K
425 NEXT L
```

```
440 REM FICHA
450 DATA "I",3,7,4,7,15,28,21,20
451 DATA "J",128,192,64,192,224,112,208,80
452 DATA "K",23,36,7,2,6,14,28,0
453 DATA "L",80,72,192,128,192,224,112,0
455 REM MURO
460 DATA "M",255,16,247,247,255,1,127,127
465 REM HORMIGA
470 DATA "P",16,17,11,7,33,31,3,31
473 DATA "Q",8,136,208,226,132,248,192,248
475 DATA "R",35,15,19,39,39,3,1,0
477 DATA "S",196,242,200,228,228,192,128,0
480 REM TUNEL
482 DATA "A",255,145,145,159,145,241,146,148
484 DATA "B",255,17,35,69,141,29,53,117
486 DATA "C",152,145,163,198,140,255,136,255
488 DATA "D",213,151,149,245,149,253,5,255
```

Esta es la rutina que muestra el menú principal:

```
600 REM MENU PRINCIPAL
615 PAPER 0: INK 6: BORDER 0: CLS : FOR I=-6 TO 3 STEP 2
630 PLOT 16,8+I: DRAW 224,0: DRAW 8-I,8-I: DRAW 0,145
640 DRAW I-8,8-I: DRAW -224,0: DRAW I-8,I-8: DRAW 0,-145: DRAW 8-I,I-8: NEXT I
```

```

660 PRINT AT 3,8;"RETROTRIVIAL ZX"
670 PRINT AT 8,3;"1. COMENZAR A JUGAR"
680 PRINT AT 11,3;"2. CONFIGURAR JUEGO"
690 PRINT AT 14,3;"3. REGLAS DEL JUEGO"

710 PLOT 49,138: DRAW 146,0: DRAW 0,18: DRAW -146,0: DRAW 0,-18
715 PRINT AT 19,9;"ELIJE OPCION!"

725 LET A=1
730 PRINT AT 3,8; INK A;"RETROTRIVIAL ZX"
740 LET K$=INKEY$
780 IF K$<>" " THEN IF K$>="1" AND K$<="3" THEN GO TO 830
790 LET A=A+1:IF A=7 THEN LET A=1
795 PAUSE 6
800 GO TO 730

```

Según se pulse se enviará a las distintas subrutinas:

```

830 REM ENVIO A SUBRUTINAS
835 BEEP .1,20
840 IF K$ ="1" THEN GO SUB 1500
845 IF K$ ="2" THEN GO SUB 1000
850 IF K$ ="3" THEN GO SUB 5400
855 GO TO 600
875 LET NADAES=SIN(L OR NA)

```

Este bloque es el que permite configurar el juego. Las opciones configurables son:

- número de jugadores
- nombre de los jugadores
- número de opciones

Está configurado de tal forma que pulsando ENTER en cualquier opción, tomará el valor por defecto.

(El POKE de la línea 1190 es un viejo truco para activar las mayúsculas manipulando una variable del sistema)

```

1000 REM CONFIGURAR JUEGO
1010 PAPER 5: BORDER 5: INK 0: CLS
1020 PLOT 0,0: DRAW 255,0: DRAW 0,175: DRAW -255,0: DRAW 0,-175
1030 PRINT INVERSE 1; AT 1,7;"CONFIGURAR JUEGO"

1120 PRINT AT 6,1;"NUMERO DE JUGADORES (2 A 4)?"
1130 LET K$=INKEY$
1140 IF K$>"1" AND K$<="4" THEN LET NUMJUG=VAL K$: GO TO 1170
1150 IF K$=CHR$ 13 THEN GO TO 1170
1160 GO TO 1130
1170 PRINT AT 7,1;"PARTICIPAN "+ STR$ NUMJUG+" JUGADORES"
1175 FOR I=1 TO 50: NEXT I
1180 PRINT AT 9,1;"NOMBRE DE LOS JUGADORES"
1190 POKE 23658,8: REM ACTIVA MAYUSCULAS
1195 FOR I=1 TO NUMJUG

```

```

1200 PRINT AT 10+I,3;"JUGADOR"; STR$ I;":"
1220 INPUT "NOMBRE: ";K$: LET N$(I)=K$
1225 IF K$="" THEN LET N$(I)= "JUGADOR "+STR$ I
1230 PRINT AT 10+I,15;N$(I)
1240 NEXT I
1250 PRINT AT 12+NUMJUG,1;"NUMERO DE OPCIONES (2 A 5)"
1255 PAUSE 10
1260 LET K$=INKEY$
1270 IF K$>="2" AND K$<="5" THEN LET NUMOPC=VAL K$: GO TO 1300
1280 IF K$=CHR$ 13 THEN GO TO 1300
1290 GO TO 1260
1300 PRINT AT 13+NUMJUG,1;STR$ NUMOPC;" OPCIONES"
1310 PRINT AT 15+NUMJUG,1;"TODO CORRECTO? (S/N)"
1320 LET K$=INKEY$
1330 IF K$="S" OR K$="s" THEN RETURN
1340 IF K$="N" OR K$="n" THEN GO TO 1000
1350 GO TO 1320

```

Este trozo de programa se usa para el premio final que se le da al ganador o ganadores. El premio, se compone de dos partes, una es una beta de un programa (con fama de ser de lo peorcito, como pej Uchi Mata) seguido de una chuchería de la época (como pej un Bollycao). Para la lista de premios hay dos bloques de DATA, que comienzan en las líneas 9940 (programas malos) y 9960 (chucherías) respectivamente:

```

1400 REM PREMIO
1405 LET M$="UNA BETA DE "
1410 LET AUX= INT(RND*5)+1:RESTORE 9940+AUX:READ K$:LET M$=M$+K$+" Y "
1415 LET AUX= INT(RND*10)+1:RESTORE 9960+AUX:READ K$:LET M$=M$+K$

1425 LET OFF=4-(NGAN=1):LET BORRA=0:LET JU=7:GO SUB 3300
1430 RETURN

```

Esta es la rutina principal desde la cual se controla el flujo del programa. Hemos usado una programación lo más modular posible para facilitar la comprensión y posterior modificación del programa.

El final del juego se controla en la línea 1630 finaliza cuando C(7)=0 es decir cuando ya no queda ningún cuadro en blanco que rellenar.

```

1500 REM COMENZAR A JUGAR
1510 GO SUB 1700: REM VAR LOCALES
1520 GO SUB 1800: REM CREA TABLERO DE JUEGO
1530 GO SUB 2300: REM BIENVENIDA JUGADORES
1560 GO SUB 2700: REM GENERA PREGUNTA Y OPCIONES
1580 GO SUB 3250: REM IMPRIME OPCIONES
1590 GO SUB 3400: REM RESPUESTA
1600 GO SUB 3500: REM RESULTADO
1610 IF CORRECTO THEN GO SUB 3700
1620 IF NOT CORRECTO THEN GO SUB 4000
1625 LET JU=JU+1:IF JU>NUMJUG THEN LET JU=1: REM CAMBIO DE JUGADOR

```

```
1630 IF C(7)<>0 THEN GO TO 1560: REM REPITE MIENTRAS HAYA CUADROS EN BLANCO
1640 GO SUB 5200: REM FINAL
1650 RETURN
```

Aquí se declaran las variables locales, es decir aquellas que deben cambiar en cada partida. Hay una descripción de las mismas en el [Anexo I](#)

```
1700 REM VAR. LOCALES
1710 DIM T(COLUMNAS,FILAS): DIM C(7): LET C(7)=NUMCUAD: LET JU=1: DIM P(5,40): DIM
Q(5,40)
1720 DIM Z(NUMJUG): FOR K=1 TO NUMJUG:LET Z(K)=(12/NUMJUG):NEXT K
1725 LET C$="CADA":IF NUMJUG=2 THEN LET C$="TU"
1730 LET HORM=0:LET SAB=0:LET BLANCOS=0:LET MIT=0:LET CONTBL=0:LET OFF=0:LET
BORRA=1
1740 RETURN
```

En este bloque se dibuja el tablero de juego y se colocan las fichas iniciales de los jugadores. Estas coordenadas son:

- para el jugador 1: 6,5 (o lo que es lo mismo F5)
- para el jugador 2: 6,1 (o F1)
- para el jugador 3: 2,3 (o B3)
- para el jugador 4: 10,3 (o J3)

Son las que hemos considerado más equitativas, pero se pueden cambiar fácilmente, sin más que modificarlas en este trozo de código

```
1800 REM CREA TABLERO DE JUEGO
1815 BORDER 0: PAPER 0: INK 6: CLS: FOR K=1 TO 7 STEP 2
1830 PLOT K,K: DRAW 255-2*K,0: DRAW 0,175-2*K
1840 DRAW -255+2*K,0: DRAW 0,-175+2*K: NEXT K
1860 FOR K=8 TO 208 STEP 16: PLOT K,8: DRAW 0,80: NEXT K
1890 FOR K=8 TO 88 STEP 16: PLOT 8,K: DRAW 192,0: NEXT K
1920 FOR K=88 TO 92: PLOT 8,K: DRAW 239,0: NEXT K
1943 FOR K=201 TO 205: PLOT K,8: DRAW 0,80: NEXT K
1960 FOR K=1 TO 12: PRINT PAPER 7; INK 0; AT 21,K*2-1;CHR$(64+K);" ": NEXT K
1975 PRINT PAPER 7; AT 21,0;" "
1985 FOR K=1 TO 5: PRINT PAPER 7; INK 0; AT 22-2*K-1,0;" ": PRINT PAPER 7; INK 0; AT 22-
2*K,0;K: NEXT K
2020 FOR K=1 TO NUMJUG:PRINT PAPER K; INK 0; AT 15-INT (NUMJUG/2)+K,26; "J";STR$
K;"=00":NEXT K
2023 REM CASILLAS INICIALES
2025 LET AROUND=0:LET FLSH=0
2026 LET X=6:LET Y=5:LET COD=1:GO SUB 4700
2027 LET X=6:LET Y=1:LET COD=2:GO SUB 4700
2030 IF NUMJUG=3 THEN GO TO 2045
2035 IF NUMJUG=4 THEN GO TO 2055
2040 RETURN
2045 LET X=2:LET Y=3:LET COD=3:GO SUB 4700
2050 RETURN
2055 LET X=2:LET Y=3:LET COD=3:GO SUB 4700
2060 LET X=10:LET Y=3:LET COD=4:GO SUB 4700
```

## 2065 RETURN

Esta rutina se utiliza para pintar cuadros negros aleatorios (los que tienen la apariencia de un muro). Como entrada la rutina necesita que la variable NUM esté inicializada con el número de cuadros negros a colocar.

Antes de llamar a esta rutina hay que asegurarse de que  $NUM \leq C(7)$  es decir de que haya suficientes casillas vacías para colocar esos cuadros negros.

En 2106 inicializamos las variables G\$ y F\$ con el UDG de muro

La rutina que empieza en 2250 es la que pinta realmente el cuadro y llama a la rutina para actualizar el marcador (que veremos más adelante).

Esta rutina se utiliza para:

- Pintar los cuadros negros iniciales
- Pintar un cuadro negro cuando el jugador no acierta

Cuando se pintan los cuadros negros iniciales solo se actualiza el contador al final (cuando  $CONT=NUM$ ), para optimizar velocidad.

Cuando queremos colocar cuadros negros, pero no de forma aleatoria, sino a partir de unas coordenadas, no utilizamos esta rutina sino la que empieza en 4600

```
2100 REM PINTA NUM CUADROS NEGROS
2103 LET CONT=0:RANDOMIZE
2106 LET G$="{M}"+"{M}":LET H$="{M}"+"{M}"
2107 IF BLANCOS THEN GO SUB 4550:LET XALEA=X:LET YALEA=Y:GO TO 2125
2110 LET XALEA=INT(RND*COLUMNAS)+1
2115 LET YALEA=INT(RND*FILAS)+1
2120 IF T(XALEA,YALEA)<>0 THEN GO TO 2110
2125 LET T(XALEA,YALEA)=6
2130 LET CONT=CONT+1:LET C(6)=C(6)+1:LET C(7)=C(7)-1
2135 GO SUB 2250
2150 IF CONT=NUM THEN RETURN
2155 GO TO 2107
```

```
2250 REM PINTA CUADRO Y ACTUALIZA MARCADOR SI CONT=NUM
2255 PRINT PAPER 0; INK 6;AT 21-YALEA*2,2*XALEA-1;G$
2260 PRINT PAPER 0; INK 6;AT 22-YALEA*2,2*XALEA-1;H$
2263 IF BIP THEN BEEP .05,20
2265 IF CONT=NUM THEN GO SUB 4795
2270 RETURN
```

Esta rutina da la bienvenida a los jugadores al comienzo del juego y rellena los cuadros negros aleatorios al comienzo de la partida. El número de cuadros negros aleatorios viene dado por la variable NEGROSINI

```
2300 REM BIENVENIDA JUGADORES
2320 FOR K=1 TO NUMJUG: PRINT AT 5+K-NUMJUG,2;INK K;N$(K): NEXT K
2360 PRINT AT 6+K-NUMJUG,2;INK 7; "BIENVENIDOS A RETROTRIVIAL"
```

```
2363 LET AROUND=1
2365 LET NUM=NEGROSINI:LET BIP=0:GO SUB 2100
```

```
2370 PRINT AT 9,2;FLASH 1; "PULSA UNA TECLA PARA COMENZAR"
2385 LET K$=INKEY$
2490 IF K$="" THEN GO TO 2385
2495 RANDOMIZE
2500 RETURN
```

En este trozo de programa se decide el tipo de pregunta que se le hará al concursante. Las preguntas pueden ser de cinco tipos:

TIPO 1: Pregunta de qué compañía es un determinado juego

TIPO 2: Pregunta en que año se hizo un juego

TIPO 3: Pregunta quién es el programador de un juego

TIPO 4: Hace una pregunta de Basic

TIPO 5: Hace una pregunta de Assembler

Las variables P1, P2, P3, P4, P5 contienen el porcentaje en tanto por ciento de veces que se realizará este tipo de pregunta. Se pueden cambiar fácilmente en las variables globales (línea 355)

La rutina 3190 se encarga de gestionar las preguntas 4 y 5, ya que la estructura de ambas (preguntas BASIC y preguntas ASSEMBLER) es muy similar.

```
2700 REM GENERA PREGUNTA
2702 RANDOMIZE
2705 LET AUX=INT (RND*100)+1
2710 IF AUX<=P1 THEN LET TIPOPREG=1:GO SUB 2800:RETURN
2715 IF AUX<=(P1+P2) THEN LET TIPOPREG=2:GO SUB 2950:RETURN
2720 IF AUX<=(P1+P2+P3) THEN LET TIPOPREG=3:GO SUB 3100:RETURN
2725 IF AUX<=(P1+P2+P3+P4) THEN LET TIPOPREG=4:GO SUB 3190:RETURN
2730 IF AUX<=(P1+P2+P3+P4+P5) THEN LET TIPOPREG=5:GO SUB 3190:RETURN
2750 RETURN
```

Gestiona la pregunta tipo 1: Pregunta de qué compañía es un determinado juego

- Escribe la pregunta en pantalla
- Lee la opción correcta y almacena el número esa opción en la variable OK
- Genera las posibles opciones de respuesta y las almacena en la matriz O\$()

Un punto interesante desde el punto de vista de trucos de programación se puede ver en la línea 2820, concretamente en el GOSUB 4100. Tras colocar la pregunta en pantalla, lo lógico sería colocar un PAUSE, pero nosotros colocamos este GOSUB 4100 tras cada pregunta que se presenta en pantalla, ya que vamos a utilizar ese tiempo para realizar una serie de cálculos. (ver más adelante la rutina 4100)

```
2800 REM PREG. TIPO 1
```



2815 GO SUB 4200: REM BORRA PAN.

2820 PRINT AT 4,2;INK JU;N\$(JU);AT 6,2; "DE QUE COMPA{N}IA";AT 7,2;"ES EL SIGUIENTE  
JUEGO?":GO SUB 4100

2825 REM LEE OPCION CORRECTA

2830 LET K=INT (RND\*MAXJUEGOS)+1: RESTORE INIJUEGOS+K: READ M\$,A\$

2840 IF A\$(1)<"0" OR A\$(1)>"9" THEN GO TO 2830

2845 LET K=VAL A\$: RESTORE INIEMPRESAS+K: READ A\$

2850 REM GENERA OPCIONES

2855 DIM O\$(NUMOPC,23)

2860 LET AL=INT (RND\*MAXEMPRESAS)+1

2865 RESTORE INIEMPRESAS+AL: READ K\$

2870 IF K\$=A\$ THEN GO TO 2860

2875 LET O\$(1)=K\$

2880 FOR K=2 TO NUMOPC

2885 LET AL=INT (RND\*MAXEMPRESAS)+1

2890 RESTORE INIEMPRESAS+AL: READ K\$

2895 IF K\$=A\$ THEN GO TO 2885

2900 LET O\$(K)=K\$

2905 FOR L=1 TO K-1

2910 IF O\$(L)=O\$(K) THEN GO TO 2885

2915 NEXT L

2920 NEXT K

2925 LET OK=INT (RND\*NUMOPC)+1

2930 LET O\$(OK)=A\$

2935 RETURN

Gestiona la pregunta tipo 2: Pregunta en que año se hizo un juego

- Escribe la pregunta en pantalla
- Lee la opción correcta y almacena el número esa opción en la variable OK
- Genera las posibles opciones de respuesta y las almacena en la matriz O\$()
- Ordena las fechas para que sea más fácil elegir al jugador. La ordenación se hace por el método de la burbuja.

Para generar los años aleatorios esta rutina genera fechas 7 años antes y 7 años después de la realización del juego, pero limitándolo inferiormente y superiormente. Inferiormente se limita por la variable LIMINF que está por defecto en 1981 y superiormente se limita por la variable LIMSUP cuyo limite viene dado por la variable global ANOACTUAL. Esto se ha hecho así para dar cabida a los juegos homebrew en una posible actualización.

2950 REM PREG. TIPO 2 PIDE ANNO

2970 GO SUB 4200: REM BORRA PAN.

```
2975 PRINT AT 4,3;INK JU;N$(JU);AT 6,4; "EN QUE A{N}O";AT 7,5;"SE LANZO ESTE JUEGO?":GO
SUB 4100
```

```
2980 REM LEE OPCION CORRECTA (PREG. TIPO 2)
```

```
2985 LET K=INT (RND*MAXJUEGOS)+1: RESTORE INIJUEGOS+K: READ M$,A$,F$
```

```
2990 IF A$(1)<"0" OR A$(1)>"9" THEN GO TO 2985
```

```
3000 REM GENERA OPCIONES (PREG. TIPO 2)
```

```
3005 DIM O$(NUMOPC,4):DIM O(NUMOPC)
```

```
3007 LET LIMINF=VAL(F$)-7
```

```
3010 IF LIMINF<=1981 THEN LET LIMINF=1981
```

```
3015 LET LIMSUP=VAL(F$)+7
```

```
3020 IF LIMSUP>=ANOACTUAL THEN LET LIMSUP=ANOACTUAL
```

```
3025 LET RANGO=LIMSUP-LIMINF
```

```
3027 LET O(1)=VAL(F$):REM ALMACENA LA FECHA CORRECTA
```

```
3030 FOR K=2 TO NUMOPC
```

```
3035 LET AL=INT(RND*RANGO)+LIMINF
```

```
3036 REM VERIFICA QUE NO ESTA REPETIDO
```

```
3037 FOR L=1 TO K-1
```

```
3038 IF AL=O(L)THEN GO TO 3035
```

```
3039 NEXT L
```

```
3040 LET O(K)=AL
```

```
3050 NEXT K
```

```
3060 REM ORDENA POR BURBUJA
```

```
3063 FOR K=1 TO NUMOPC
```

```
3065 FOR L=1 TO NUMOPC-K
```

```
3070 LET AUX=O(L)
```

```
3073 IF O(L)>O(L+1) THEN LET O(L)=O(L+1):LET O(L+1)=AUX
```

```
3075 NEXT L
```

```
3080 NEXT K
```

```
3084 REM PASA LOS O() A O$() Y ASIGNA LA VARIABLE OK CON LA OPCION CORRECTA
```

```
3085 FOR K=1 TO NUMOPC
```

```
3087 LET O$(K)=STR$(O(K))
```

```
3090 IF O$(K)=F$ THEN LET OK=K
```

```
3095 NEXT K
```

```
3097 RETURN
```

Gestiona la pregunta tipo 3: Pregunta quién es el programador de un juego

- Escribe la pregunta en pantalla
- Lee la opción correcta y almacena el número esa opción en la variable OK
- Genera las posibles opciones de respuesta y las almacena en la matriz O\$()

```
3100 REM PREG. TIPO 3 PIDE AUTOR
```

```
3105 GO SUB 4200: REM BORRA PAN.
```

```
3110 PRINT AT 4,3;INK JU;N$(JU);AT 6,4; "QUIEN FUE PROGRAMADOR";AT 7,7;"DEL SIGUIENTE
JUEGO?":GO SUB 4100
```

```
3115 REM LEE OPCION CORRECTA (PREG. TIPO 3)
```

```
3117 LET K=INT (RND*MAXJUEGOS)+1: RESTORE INIJUEGOS+K: READ M$,D$,D$,T$
```

```
3120 IF T$="0" THEN GO TO 3117
```

3123 LET K=VAL T\$: RESTORE INIAUTOR+K: READ T\$

3130 REM GENERA OPCIONES (PREG. TIPO 3)

3133 DIM O\$(NUMOPC,23)

3135 LET AL=INT (RND\*MAXAUTOR)+1

3140 RESTORE INIAUTOR+AL: READ K\$

3143 IF K\$=T\$ THEN GO TO 3135

3145 LET O\$(1)=K\$

3150 FOR K=2 TO NUMOPC

3153 LET AL=INT (RND\*MAXAUTOR)+1

3155 RESTORE INIAUTOR+AL: READ K\$

3157 IF K\$=T\$ THEN GO TO 3153

3160 LET O\$(K)=K\$

3165 FOR L=1 TO K-1

3167 IF O\$(L)=O\$(K) THEN GO TO 3153

3170 NEXT L

3173 NEXT K

3175 LET OK=INT (RND\*NUMOPC)+1

3177 LET O\$(OK)=T\$

3180 RETURN

[Gestiona las preguntas tipo 4 y 5](#)

[Como tienen una estructura similar y para ahorrar memoria se gestionan los dos tipos de pregunta en la misma rutina:](#)

3190 REM PREG. TIPO 4 Y 5

3193 GO SUB 4200: REM BORRA PAN.

3195 IF TIPOPREG=4 THEN LET AUX1=INIBASIC:LET AUX2=MAXBASIC

3197 IF TIPOPREG=5 THEN LET AUX1=INIASSEMBLER:LET AUX2=MAXASSEMBLER

3200 LET K=INT(RND\*AUX2)+1: RESTORE AUX1+K: READ M\$

3203 IF TIPOPREG=4 THEN PRINT AT 4,3;INK JU;N\$(JU);AT 6,3;"COMANDO BASIC PARA...":GO SUB 4100

3205 IF TIPOPREG=5 THEN PRINT AT 4,3;INK JU;N\$(JU);AT 6,3;"COMANDO ASSEMBLER PARA...":GO SUB 4100

3207 REM GENERA OPCIONES

3210 DIM O\$(6,23)

3213 LET OK=INT (RND\*NUMOPC)+1

3215 READ D\$:LET O\$(OK)=D\$

3220 FOR I=1 TO 5

3225 IF I=OK THEN GO TO 3230

3227 READ D\$: LET O\$(I)=D\$

3230 NEXT I

3235 LET ALEA1=INT(RND\*5)+1:LET ALEA2=INT(RND\*5)+1

3237 IF ALEA1=OK OR ALEA2=OK THEN GO TO 3235

3240 LET O\$(6)=O\$(ALEA1):LET O\$(ALEA1)=O\$(ALEA2):LET O\$(ALEA2)=O\$(6)

3245 RETURN

Esta rutina es compartida por todas las rutinas de generación de preguntas, y su función es mostrar las opciones en pantalla para que el jugador elija. Necesita como entrada la pregunta almacenada en M\$ y las distintas opciones almacenadas en la matriz O\$( )

```
3250 REM MUESTRA OPCIONES EN PANTALLA
3260 GO SUB 4200: REM BORRA PAN.
3265 GO SUB 3300:REM ESCRIBE MENSAJE M$
3280 FOR K=1 TO NUMOPC
3285 PRINT AT 4+K,3;STR$ K;".- ";O$(K): NEXT K
3290 RETURN
```

Esta pequeña subrutina se utiliza desde muchas partes distintas del programa. Sirve para escribir en la parte superior de la pantalla. Tiene como entrada la variable M\$, la variable OFF, y la variable booleana BORRA.

Lo que haya en M\$ lo escribe en pantallita superior de forma centrada, y si es necesario hasta en dos líneas (no está prevista que ocupe 3)

La variable OFF se utiliza para que el texto se escriba más abajo de su ubicación por defecto. El flag BORRA si es 1 borra la pantalla y si es 0 no la borra.

```
3300 REM ESCRIBE MENSAJE M$
3305 IF BORRA THEN GO SUB 4200: REM BORRA PAN.
3307 IF LEN M$<=28 THEN LET L$=M$: GO TO 3335
3310 IF M$(29)=" " THEN LET K=28: GO TO 3325
3315 FOR K=28 TO 1 STEP -1
3318 IF M$(K)=" " THEN GO TO 3325
3320 NEXT K
3325 LET K$=M$( TO K): LET L$=M$(K+1 TO )
3330 PRINT AT 2+OFF,16-LEN K$/2;INK JU;K$
3335 PRINT AT 3+OFF,16-LEN L$/2;INK JU;L$
3337 LET OFF=0:LET BORRA=1
3340 RETURN
```

Esta subrutina se usa por parte de todos los tipos de pregunta para chequear la pulsación de las opciones y determinar si fue o no correcta:

```
3400 REM CHEQUEA RESPUESTA
3410 LET K$=INKEY$
3420 IF K$<"1" OR K$>STR$ (NUMOPC) THEN GO TO 3410
3430 IF OK=VAL K$ THEN LET CORRECTO=1:BEEP .01,1
3440 IF OK<>VAL K$ THEN LET CORRECTO=0:BEEP .01,1
3450 RETURN
```

Esta rutina muestra en pantalla si la respuesta elegida fue o no correcta:

```
3500 REM IMPRIME RESULTADO
3510 GO SUB 4300: REM BORRA DESDE 4
3520 IF CORRECTO=1 THEN GO TO 3600
3530 PRINT AT 3,3;INK JU;"INCORRECTO,"; N$(JU);AT 6,3;"LA RESPUESTA ERA: ";AT 7,3; PAPER
0;INK 7; O$(OK): BEEP .5,-15
3570 RETURN
3600 PRINT AT 3,3;INK JU; "CORRECTO,";N$(JU);AT 6,3;"ES ";PAPER 0;INK7; O$(OK)
```

## 3620 RETURN

Esta rutina es una de las más complejas, ya que a su vez llama a múltiples subrutinas. Determina todo lo que sucede cuando se responde correctamente una opción.

- Pide una coordenada o bien un comodín. Si se pulsó la letra Z (comodín) resta ese comodín del jugador y deriva hacia la rutina que gestiona los comodines (6100).
- Si se introdujo una coordenada, se comprueba en primer lugar que no sea de una casilla ocupada (3790)
- Si no es de una casilla ocupada se comprueba entonces si la coordenada elegida tiene una ficha de su mismo color colindante. Esto se hace mediante una llamada a la subrutina 4400. Caso de que tenga una casilla colindante la jugada es legal y se puede continuar con el flujo del programa
- Si la casilla elegida no está ocupada ni tiene una ficha colindante de su mismo color, aun puede ser una jugada correcta en el caso de que no haya en todo el tablero ninguna casilla de ese jugador que tenga una casilla colindante libre. Esta es la función de las líneas 3830 a 3960. Comprobar esto en un tiempo aceptable es un verdadero reto y ha habido que hacer bastantes esfuerzos para conseguir tiempo medianamente aceptables.

¿Cómo lo hemos hecho?

- En las líneas 3940 a 3960 se comprueba cada una de las fichas del jugador para ver si alguna de ellas tiene un cuadro vacío. Si esto ocurriera la casilla elegida sería ilegal, ya que según hemos visto no colindaba con ninguna propia.
- El problema de este método es que si el jugador tiene muchas fichas en el tablero el tiempo de comprobación se hace demasiado elevado. Para evitar esto tenemos la línea 3830.
- La línea 3830 dice que si el número de fichas del jugador es mayor que el número de cuadros vacíos (más dos, por razones de eficiencia) entonces ejecuta otra rutina que lo que hace es ver por cada cuadro vacío si hay una casilla del color del jugador. Es otra manera de ver si la jugada es legal ya que si se encontrara que al lado de un cuadro vacío hay una ficha del jugador querría decir también que la coordenada elegida es ilegal.
- El problema con esta última rutina era que en principio no llevábamos un conteo de los cuadros vacíos ya que esto ralentizaría mucho el programa. Para solucionar esto lo que hemos hecho es utilizar los tiempos muertos en los que el programa muestra una pregunta en pantalla para hacer una recopilación de estos cuadros vacíos. Para más detalles de cómo se hizo esto ver los comentarios de la rutina 4100

```
3700 REM CORRECTO
```

```
3705 LET YACOMPROBADO=0
```

```
3710 BEEP.1,10
```

```
3715 PRINT AT 8,3;INK JU;"INTRODUCE COORDENADAS: "
```

```
3717 IF Z(JU)>0 THEN PRINT AT 9,3;INK JU;"(O Z PARA COMODIN. TIENES ";Z(JU);")"
```

```
3720 POKE 23658,8:REM ACTIVA MAYUSCULAS
```

```
3725 LET K$=INKEY$
```

```
3727 IF K$="Z" AND Z(JU)>0 THEN LET Z(JU)=Z(JU)-1:BEEP .01,1:GO SUB 6100:RETURN: REM  
EVENTOS
```

```
3730 IF K$<"A" OR K$>"L" THEN GO TO 3725
```

```
3740 BEEP .01,1:PRINT AT 8,25;INK 7;FLASH 1;K$: LET X=CODE(K$)-64
```

```
3760 LET K$=INKEY$
```

```
3765 IF K$="0" THEN GO TO 3710
3770 IF K$<"1" OR K$>"5" THEN GO TO 3760
3780 BEEP .01,1:PRINT AT 8,26;INK 7;FLASH 1;K$: LET Y=VAL K$
3790 IF T(X,Y)<>0 THEN LET Z$="CASILLA OCUPADA":GO TO 3980: REM CASILLA OCUPADA
3800 IF C(JU)=0 OR C(7)=1 THEN GO TO 3965: REM SI PRIMERA O ULTIMA JUGADA NO
COMPRUEBES
```

```
3805 REM TIENE UN PROXIMO PROPIO?
3810 LET PR=JU: LET X1=X: LET Y1 =Y: GO SUB 4400: REM DETECTA PROXIMO PROPIO
3820 IF PROXIMO=1 THEN GO TO 3965
```

```
3826 IF YACOMPROBADO=1 THEN LET Z$="NO COLINDANTE":GO TO 3980: REM YA SE
COMPROBO ANTERIORMENTE
```

```
3830 IF C(JU)<C(7)+2 OR NOT BLANCOS THEN GO TO 3940
```

```
3835 LET PR=JU
3840 FOR K=1 TO CONTBL
3845 LET X1=P(5,K):LET Y1=Q(5,K)
3850 IF T(X1,Y1)=0 THEN GO SUB 4400
3855 IF PROXIMO=1 THEN GO TO 3955
3865 NEXT K
3870 GO TO 3965
```

```
3940 LET PR=0
3945 FOR K=1 TO C(JU)
3950 LET X1=P(JU,K):LET Y1=Q(JU,K):GO SUB 4400: REM DETECTA PROXIMO PROPIO
3955 IF PROXIMO=1 THEN LET YACOMPROBADO=1: LET Z$="NO COLINDANTE":GO TO 3980
3960 NEXT K
```

```
3965 LET COD=JU:LET AROUND=1: GO SUB 4600: REM COLOCA FICHA
3970 RETURN
```

```
3980 REM JUGADA ILEGAL
3982 BEEP .5,-3
3985 PRINT AT 2,2;FLASH1;PAPER 0;INK7;Z$;" :REPITE JUGADA"
3990 PRINT AT 8,25;" "
3993 IF INKEY$="" THEN GO TO 3993
3995 PRINT AT 2,1;" "
3997 GO TO 3727
```

Si la jugada no es correcta simplemente coloca un cuadro negro (cuadro de muro) aleatorio, mediante una llamada a la rutina ya comentada

```
4000 REM NO CORRECTO
4020 PRINT AT 9,2;INK JU;" EL ORDENADOR COLOCA FICHA"
4030 LET NUM=1:LET BIP=1: GO SUB 2100: REM COLOCA CUADRO NEGRO
4040 RETURN
```

Esta es una rutina que se ejecuta aprovechando los tiempos muertos en los que se muestran las preguntas en pantalla. Funciona de la siguiente forma:

- Si el número de cuadros vacíos es mayor o igual que 17, se genera una pausa con el comando PAUSE, en caso contrario ocurre lo que se explica a continuación.
- La primera vez que es llamada esta rutina (MIT=0) escanea la mitad izquierda de la pantalla. Cuando se encuentra el primer cuadro vacío, introduce su coordenada X en P(5,1) y la coordenada Y en Q(5,1); cuando se encuentra el segundo cuadro vacío, almacena la coordenada X en P(5,2) y la Y en Q(5,2), y así sucesivamente.
- La segunda vez que se ejecuta la rutina (MIT=1) hace lo mismo con la mitad derecha de la pantalla. El hecho de dividirlo en dos partes se hace para que el tiempo de espera en el cual se muestra la pregunta en pantalla sea razonable.
- Obsérvese que para los dos escaneos utilizamos un mismo bucle, activando la variable booleana MIT para conmutar entre una y otra mitad.
- A partir de la tercera vez que se ejecuta la rutina (BLANCOS=1) lo que se hace es ir comprobando todas las coordenadas que hay en la matriz P(5,X) y Q(5,Y) para eliminar aquellos cuadros que ya no son vacíos. La forma de hacerlo es copiando las coordenadas de los cuadros vacíos en P(5,20+X) y Q(5,20+Y) y luego pasarlos de nuevo a P(5,X) y Q(5,Y). Esto se hace en las líneas 4150-4175
- ¿qué se consigue con todo esto? Se consigue tener las coordenadas de los cuadros vacíos en una matriz, lo cual se empleará en ciertas rutinas para hacerlas más eficientes. Estas rutinas solo se ejecutarán si la variable booleana BLANCOS es igual a cero, es decir si ya se escanearon las dos partes de la pantalla.

```
4100 REM RECOPILA BLANCOS
4103 IF C(7)>=17 THEN PAUSE 200:RETURN
4107 IF BLANCOS THEN GO TO 4150
4110 FOR K=(MIT=0)+7*(MIT=1) TO 6*(MIT=0)+12*(MIT=1)
4115 FOR L=1 TO 5
4120 IF T(K,L)=0 THEN LET CONTBL=CONTBL+1:LET P(5,CONTBL)=K:LET Q(5,CONTBL)=L
4125 NEXT L
4130 NEXT K
4135 IF MIT=0 THEN LET MIT=1:RETURN
4140 IF MIT=1 THEN LET BLANCOS=1:RETURN

4150 LET CONT=0
4152 FOR K=1 TO CONTBL
4155 IF T(P(5,K),Q(5,K))=0 THEN LET CONT=CONT+1:LET P(5,20+CONT)=P(5,K):LET
Q(5,20+CONT)=Q(5,K)
4160 NEXT K
4165 FOR K=1 TO C(7):LET P(5,K)=P(5,20+K):LET Q(5,K)=Q(5,20+K):NEXT K
4170 LET CONTBL=C(7)
4175 RETURN
```

La siguientes son dos pequeñas rutinas para borrar la parte superior de la pantalla

```
4200 REM BORRA PANTALLITA
4210 DIM B$(1,30): FOR K=1 TO 9
```

```
4230 PRINT AT K,1;B$(1): NEXT K
4250 RETURN
```

```
4300 REM BORRA DESDE 4
4310 DIM B$(1,30): FOR K=1 TO 9
4330 PRINT AT K,1;B$(1): NEXT K
4350 RETURN
```

Esta es una rutina que podía haberse hecho de forma más elegante, pero que hubiera sido menos efectiva en cuanto a tiempo de ejecución.

La entrada para esta rutina es:

- X1,Y1 son las coordenadas de la casilla a comprobar
- PR es el color que se va a comprobar. Es decir se va a comprobar si alrededor de la casilla X1,Y1 hay una ficha de color PR. Si PR fuera 0 lo que se comprobaría entonces es si hay alguna casilla vacía.

La salida del programa es la variable booleana PROXIMO. Devuelve PROXIMO=1 si hay una ficha de color PR alrededor y PROXIMO=0 si no la hay

```
4400 REM DETECTA PROXIMO PROPIO
4410 LET PROXIMO=0
```

```
4420 IF X1<=12 AND Y1+1<=5 AND X1>0 AND Y1+1>0 THEN IF T(X1,Y1+1)=PR THEN LET
PROXIMO=1: RETURN
4421 IF X1+1<=12 AND Y1+1<=5 AND X1+1>0 AND Y1+1>0 THEN IF T(X1+1,Y1+1)=PR THEN LET
PROXIMO=1: RETURN
4422 IF X1+1<=12 AND Y1<=5 AND X1+1>0 AND Y1>0 THEN IF T(X1+1,Y1)=PR THEN LET
PROXIMO=1: RETURN
4423 IF X1+1<=12 AND Y1-1<=5 AND X1+1>0 AND Y1-1>0 THEN IF T(X1+1,Y1-1)=PR THEN LET
PROXIMO=1: RETURN
4424 IF X1<=12 AND Y1-1<=5 AND X1>0 AND Y1-1>0 THEN IF T(X1,Y1-1)=PR THEN LET
PROXIMO=1: RETURN
4425 IF X1-1<=12 AND Y1-1<=5 AND X1-1>0 AND Y1-1>0 THEN IF T(X1-1,Y1-1)=PR THEN LET
PROXIMO=1: RETURN
4426 IF X1-1<=12 AND Y1<=5 AND X1-1>0 AND Y1>0 THEN IF T(X1-1,Y1)=PR THEN LET
PROXIMO=1: RETURN
4427 IF X1-1<=12 AND Y1+1<=5 AND X1-1>0 AND Y1+1>0 THEN IF T(X1-1,Y1+1)=PR THEN LET
PROXIMO=1: RETURN
```

```
4490 RETURN
```

```
4500 DATA 0,1,1,1,1,0,1,-1,0,-1,-1,-1,-1,0,-1,1
```

Esta rutina busca un cuadro blanco aleatorio que esté libre y devolviendo las coordenadas donde se encuentra en las variables X e Y. Solo se puede llamar a esta rutina si la variable booleana BLANCOS es igual a 1 (ver rutina 4100) Es utilizada cuando:

- Se va a colocar una o más vidas extra
- Se va a colocar un cuadro negro aleatorio después de una respuesta incorrecta



```

4550 REM BUSCA BLANCO
4555 LET ALEA= INT(RND*CONTBL)+1
4560 LET X=P(5,ALEA):LET Y=Q(5,ALEA)
4565 IF X=0 OR T(X,Y)<>0 THEN GO TO 4555
4570 RETURN

```

Esta rutina se usa para colocar una ficha de la cual se conocen las coordenadas, ya sea de jugador o negra. Las variables de entrada son:

- X1,Y1 coordenada donde se va a colocar
- COD1 color que se va a colocar

Y estas tres variables booleanas:

- AROUND (1 si se comerá las fichas de alrededor, 0 lo contrario)
- BIP (1 si queremos que emita un pitido al colocar la ficha, 0 lo contrario)
- FLSH (1 si queremos que la ficha parpadee al colocarse, 0 lo contrario)

Cuando se usa esta rutina para colocar una vida extra, lo hacemos con AROUND=0 es decir, la vida extra no se come las fichas del contrario en el caso de que colinde con él. Para que se las comiera solo tendríamos que llamar a la rutina con AROUND=1, sin embargo estimamos que eso podría desequilibrar un poco el juego.

```

4600 REM COLOCA FICHA NEGRA O COLOR
4610 IF COD=6 THEN GO SUB 4650: RETURN
4620 GO SUB 4700
4630 RETURN

```

```

4650 REM COLOCA FICHA NEGRA
4656 IF HORM THEN LET G$="{P}"+"{Q}":LET H$="{R}"+"{S}": GO TO 4670
4660 LET G$="{M}"+"{M}":LET H$="{M}"+"{M}"
4670 LET COD1=COD:LET X1=X: LET Y1=Y: :GO SUB 4900
4680 RETURN

```

```

4700 REM COLOCA FICHA COLOR
4710 REM FICHA EN SU COORDENADA
4715 LET G$="{I}"+"{J}":LET H$="{K}"+"{L}"
4720 LET X1=X: LET Y1=Y: LET COD1=COD: LET FLSH=1: GO SUB 4900
4723 IF NOT AROUND THEN GO TO 4792
4725 REM FICHAS ALREDEDOR
4730 RESTORE 4500
4740 FOR K=1 TO 8: READ X1,Y1: LET X1=X1+X: LET Y1=Y1+Y
4780 IF X1<=12 AND Y1<=5 AND X1>0 AND Y1>0 THEN IF T(X1,Y1)<>0 AND T(X1,Y1)<>JU AND T(X1,Y1)<>6 THEN LET FLSH=0: GO SUB 4900
4790 NEXT K
4792 LET X1=X: LET Y1=Y: LET COD1=COD: LET FLSH=0: LET BIP=0: GO SUB 5100: REM QUITA EL FLASH

```

```
4795 REM ACTUALIZA MARCADOR
4800 FOR Z=1 TO NUMJUG
4810 LET K$=STR$ C(Z)
4820 IF LEN STR$ C(Z)=1 THEN LET K$="0"+STR$ C(Z)
4830 PRINT AT 15-INT(NUMJUG/2)+Z,29;K$: NEXT Z
```

```
4832 LET K$=STR$ C(6):LET K$=STR$ C(6)
4833 IF LEN STR$ C(6)=1 THEN LET K$="0"+STR$ C(6)
4835 PRINT AT 19,26;"N= ";K$
```

```
4840 LET K$=STR$ C(7):LET K$=STR$ C(7)
4843 IF LEN STR$ C(7)=1 THEN LET K$="0"+STR$ C(7)
4845 PRINT AT 20,26;"B= ";K$
4890 RETURN
```

```
4900 REM ACTUALIZA CONTADORES, PINTA Y ACTUALIZA MARCADORES
4905 REM X1,Y1,COD1 (COD1=6 NEGRO; COD1=7 VACIO)
4915 LET BIP=1
4930 LET COLOR=T(X1,Y1)
4932 IF COLOR=0 THEN LET COLOR=7
4933 IF COLOR<=4 THEN GO SUB 5020
4935 LET C(COLOR)=C(COLOR)-1
4940 GO SUB 5100
4950 LET T(X1,Y1)=COD1
4955 IF COD1=0 THEN LET COD1=7
4960 LET C(COD1)=C(COD1)+1
4962 IF COD1<=4 THEN LET P(COD1,C(COD1))=X1:LET Q(COD1,C(COD1))=Y1
4967 GO SUB 4795
5000 RETURN
```

```
5020 REM ELIMINA COORDENADAS P() Y Q() DEL JUGADOR COMIDO
5030 IF C(COLOR)=1 THEN LET P(COLOR,1)=0:LET Q(COLOR,1)=0:RETURN
5035 REM BUSCA LA COORDENADA A ELIMINAR Y EN SU LUGAR COLOCA LA ULTIMA
5040 FOR I=1 TO C(COLOR)
5050 IF P(COLOR,I)=X1 AND Q(COLOR,I)=Y1 THEN LET P(COLOR,I)=P(COLOR,C(COLOR)):LET
Q(COLOR,I)=Q(COLOR,C(COLOR)):RETURN
5055 NEXT I
5070 RETURN
```

```
5100 REM PINTA CUADRITO
5125 LET TINTA=COD1:LET PAP=6
5127 IF COD1=6 THEN LET TINTA=6:LET PAP=0
5128 IF HORM THEN LET TINTA=0:LET PAP=6
5129 IF BIP THEN BEEP .05,20
5130 PRINT PAPER PAP; INK TINTA;FLASH FLSH; AT 21-Y1*2,2*X1-1;G$
5140 PRINT PAPER PAP; INK TINTA;FLASH FLSH; AT 22-Y1*2,2*X1-1;H$
5150 RETURN
```

Esta es la rutina que se ejecuta cuando todos los cuadros blancos están rellenos, es decir cuando termina la partida. Su función es dar a conocer al ganador o ganadores y asignarle un "premio" por ganar. Se consideran ganadores todos los que estén empatados en la puntuación máxima. Para determinar al ganador llama a la rutina 5340 y para asignar el premio a la 1400 La línea 5275 tiene un sistema algo "sui generis" para determinar a la altura a la que se imprimen en pantalla los ganadores, lo hice con variables booleanas para optimizar memoria.

```
5200 REM FINAL
5210 GO SUB 4200: REM BORRA PAN.
5220 GO SUB 5340:REM GANADOR
5250 IF NGAN>1 THEN PRINT AT 1,1;INVERSE 1;"PREMIO PARA LOS GANADORES ..."
5260 IF NGAN=1 THEN PRINT AT 1,1;INVERSE 1;"PREMIO PARA EL GANADOR ..."
5265 LET CONT=0
5270 FOR K=1 TO NUMJUG
5275 IF C(K)=MAX THEN PRINT AT
3*(NGAN=1)+3*(NGAN=2)+2*(NGAN=3)+(NGAN=4)+CONT,4;INK K;N$(K): LET CONT=CONT+1
5280 NEXT K
5285 GO SUB 1400

5290 FOR K=1 TO 2
5295 FOR L=68 TO 0 STEP -1: BEEP .01,L:NEXT L
5300 NEXT K
5305 PRINT AT 9,6;INVERSE 1;FLASH1; "PULSA TECLA PARA MENU"
5310 LET K$=INKEY$
5315 IF K$<>"" THEN RETURN
5320 GO TO 5310
```

Esta rutina da como salida las siguientes variables

MAX= puntuación máxima

MIN=puntuación mínima

NGAN= número de ganadores (es decir número de jugadores con la puntuación máxima)

NPER= número de perdedores (es decir número de jugadores con la puntuación mínima)

```
5340 REM GANADOR-PERDEDOR
5350 LET MAX=0: LET NGAN=0:LET MIN=99:LET NPER=0
5355 FOR K=1 TO NUMJUG
5360 IF C(K)>MAX THEN LET MAX=C(K)
5363 IF C(K)<MIN THEN LET MIN=C(K):LET AUXMIN=K
5365 NEXT K
5370 FOR K=1 TO NUMJUG
5375 IF C(K)=MAX THEN LET NGAN=NGAN+1
5377 IF C(K)=MIN THEN LET NPER=NPER+1
5380 NEXT K
5385 RETURN
```

El bloque 5400-5497 simplemente muestra las reglas de juego.

```
5400 REM REGLAS DEL JUEGO
5410 PAPER 5: INK 9: BORDER 5: CLS
```

5440 PRINT INVERSE 1; AT 0,9;"REGLAS DE JUEGO ";

5443 PRINT AT 3,0;"RETROTRIVIAL ZX ES UN JUEGO DE PREGUNTAS Y RESPUESTAS HECHO PARA EL DISFRUTE DE LA COMUNIDADRETRO."

5445 PRINT AT 8,0;"EL JUEGO SE DESARROLLA SOBRE UN TABLERO DE 60 CUADROS Y PUEDEN JUGAR DE 2 A 4 JUGADORES."

5447 PRINT AT 12,0;"GANA AQUEL QUE TENGA MAS FICHAS EN EL TABLERO UNA VEZ TERMINADA LA PARTIDA."

5450 PRINT AT 16,0;"CUANDO ACIERTES UNA PREGUNTA TENDRAS LA OPCION DE COLOCAR UNAFICHA. SI NO ACIERTAS EL ORDENA-DOR COLOCARA UN CUADRO NEGRO AL AZAR."

5455 PRINT #0;" PULSE UNA TECLA PARA SEGUIR.": PAUSE 0: CLS

5475 PRINT AT 3,0; "SIEMPRE QUE SEA POSIBLE DEBES COLOCAR TU FICHA JUNTO A OTRA TUYA YA SEA EN UN LADO O EN DIA-GONAL"

5480 PRINT AT 8,0;"AL COLOCAR UNA FICHA LAS COLIN-DANTES DE LOS ENEMIGOS SE TRANSFORMAN EN PROPIAS."

5485 PRINT AT 13,0;"CUENTAS CON UN NUMERO LIMITADO DE COMODINES DE HEROE QUE PUEDENAYUDARTE. USALOS SABIAMENTE PEROTEN EN CUENTA QUE LOS HEROES VANA AYUDAR MAS A AQUEL QUE MAS LO NECESITE. USALOS ESPECIALMENTE CUANDO VAYAS PERDIENDO. "

5497 PRINT #0;" PULSE UNA TECLA PARA MENU": PAUSE 0:PRINT #0;" THX P11":PAUSE 20: RETURN

Entre las líneas 5500 a 5965 hay varias subrutinas que serán empleadas por los distintos bloques de héroe para realizar las acciones que tienen que realizar. Aparte de estas también usarán la 7900 (pausa) 7930 (gestión de vidas extra) y 7940 (gestión de vidas que se quitan). Las distintas líneas REM explican lo que hace cada rutina. Son en total 12 rutinas y cada una está explicada por su línea REM inicial

5500 REM VIDA EXTRA CONTR. ALEATORIO

5505 IF BLANCOS THEN GO SUB 4550:GO TO 5517

5510 LET X=INT(RND\*12)+1:LET Y=INT(RND\*5)+1

5515 IF T(X,Y)<>0 THEN GO TO 5510

5517 LET ALEA=INT(RND\*NUMJUG)+1: IF ALEA=JU THEN GO TO 5517

5520 LET AROUND=0:LET COD=ALEA: GO SUB 4600

5525 RETURN

5540 REM QUITA FICHA JUG.

5545 IF C(JU)>0 THEN LET X=P(JU,1):LET Y=Q(JU,1):LET COD=6:LET AROUND=0:GO SUB 4600:RETURN

5555 RETURN

5560 REM QUITA FICHA AL GANADOR

5575 FOR Z=1 TO NUMJUG

5580 IF C(Z)=MAX THEN GO TO 5590

5587 NEXT Z

```
5590 LET X=P(Z,1):LET Y=Q(Z,1):LET COD=6:LET AROUND=0: GO SUB 4600
5595 RETURN
```

```
5600 REM GALERIA
5602 LET G$="{A}"+"{B}":LET H$="{C}"+"{D}"
5603 PRINT AT 9,1;INK JU;"ELIJE FILA (1-5)"
5605 LET K$=INKEY$
5607 IF K$<"1" OR K$>"5" THEN GO TO 5605
5610 LET Y1=VAL K$
5612 FOR W=1 TO 12
5615 LET X1=W:LET COD1=6:LET AROUND=0
5620 GO SUB 4900
5625 NEXT W
5627 GO SUB 4795
5630 RETURN
```

```
5650 REM ATAQUE DE HORMIGAS
5655 LET HORM=1
5660 FOR W=1 TO 5
5665 LET X=INT(RND*12)+1:LET Y=INT(RND*5)+1
5670 IF T(X,Y)=6 OR T(X,Y)=JU THEN GO TO 5665
5675 LET COD=6:LET AROUND=0:GO SUB 4600
5677 IF C(7)=0 THEN RETURN
5680 NEXT W
5683 LET HORM=0
5685 RETURN
```

```
5700 REM COMPRUEBA SI FICHA QUE QUITAR
5705 LET REFUSE=0:LET AUX=0:FOR K=1 TO NUMJUG:LET AUX=AUX+C(K):NEXT K:LET
AUX=AUX-C(JU)
5707 IF AUX=0 THEN LET REFUSE=1:RETURN
5709 RETURN
```

```
5710 REM QUITA UNA FICHA A CADA CONTR.
5715 FOR W=1 TO NUMJUG
5717 IF W=JU OR C(W)=0 THEN GO TO 5725
5720 LET X=P(W,1):LET Y=Q(W,1):LET COD=6:LET AROUND=0:GO SUB 4600
5725 NEXT W
5730 RETURN
```

```
5750 REM VIDA EXTRA JUG.
5753 IF BLANCOS THEN GO SUB 4550:GO TO 5763
5755 LET X=INT(RND*12)+1:LET Y=INT(RND*5)+1
5760 IF T(X,Y)<>0 THEN GO TO 5755
5763 LET AROUND=0:LET COD=JU: GO SUB 4600
5765 RETURN
```

```
5770 REM VIDA EXTRA PARA EL PERDEDOR UNICO
5775 LET COD=AUXMIN
5780 IF BLANCOS THEN GO SUB 4550:GO TO 5800
5795 LET X=INT(RND*12)+1:LET Y=INT(RND*5)+1
5797 IF T(X,Y)<>0 THEN GO TO 5795
5800 LET AROUND=0:GO SUB 4600
5805 RETURN
```

```
5820 REM QUITA FICHA DE UN JUG. ALEATORIO
5825 LET JUGALEA=INT(RND*NUMJUG)+1:IF C(JUGALEA)=0 THEN GO TO 5825
5830 LET X=P(JUGALEA,1):LET Y=Q(JUGALEA,1):LET COD=6:LET AROUND=0:GO SUB 4600
5835 RETURN
```

```
5850 REM KIRKAFLEEG
5855 FOR K=20 TO 0 STEP -1:BEEP .08,3:BEEP .07,K:BEEP .07,K-1:BEEP .08,0: BORDER RND*6:
NEXT K
5860 BORDER 0
5870 RETURN
```

```
5900 REM INTERCAMBIO DE FICHAS
5907 IF NUMJUG=2 AND JU=1 THEN LET AUXJ=2:GO TO 5930
5910 IF NUMJUG=2 AND JU=2 THEN LET AUXJ=1:GO TO 5930
5915 PRINT AT 9,1;INK JU;"POR QUIEN TE CAMBIAS?: (1-";NUMJUG;")"
5917 LET K$=INKEY$
5920 IF K$=STR$(JU) THEN GO TO 5917
5925 IF K$<"1" OR K$>STR$(NUMJUG) THEN GO TO 5917
5927 LET AUXJ=VAL K$
```

```
5930 GO SUB 7900
5933 LET C(5)=C(AUXJ):FOR W=1 TO C(AUXJ):LET P(5,W)=P(AUXJ,W):LET
Q(5,W)=Q(AUXJ,W):NEXT W
5935 FOR W=1 TO C(JU)
5940 LET X=P(JU,W):LET Y=Q(JU,W):LET COD=AUXJ:LET AROUND=0:GO SUB 4600
5945 NEXT W
```

```
5950 FOR W=1 TO C(5)
5955 LET X=P(5,W):LET Y=Q(5,W):LET COD=JU:LET AROUND=0:GO SUB 4600
5960 NEXT W
5963 LET NORMARC=0
5965 RETURN
```

Esta es la rutina que se activa una vez un jugador pulsa Z para activar su comodín de héroe.  
Hace 3 cosas:

- Calcula la ayuda que recibirá el jugador en función de lo mal o bien que le vaya en la partida (ver el siguiente comentario para ver cómo se calcula)
- Elige tres héroes aleatoriamente de los 12 disponibles y los muestra en pantalla.
- En función de la tecla pulsada lo deriva a la rutina del héroe que corresponda. Para ahorrar memoria, la rutina presupone que comienzan en 6400 y están a una distancia de 100 líneas cada uno, por lo que se puede usar la fórmula:  $6400+100*S(\text{VAL } K\$)$

```
6100 REM EVENTO COMODIN
6105 GO SUB 6350:REM CALCULA AYUDA
6110 LET M$=" ATENCION "+N$(JU):GO SUB 3300
```

```
6200 REM GENERA PERSONAJES
6203 RANDOMIZE
6205 DIM O$(3,23):DIM S(3)
6210 FOR K=1 TO 3:LET AL=INT(RND*MAXPERSONAJES)+1:LET S(K)=AL:NEXT K
6215 IF S(1)=S(2) OR S(3)=S(1) OR S(3)=S(2) THEN GO TO 6210
6245 FOR K=1 TO 3
6250 RESTORE INIPERSONAJES+S(K):READ K$
6255 LET O$(K)=K$
6260 PRINT AT 4+K,3;STR$ K;"- ";O$(K)
6265 NEXT K
6270 PRINT AT 9,3;"ELIGE A TU HEROE (1-3)"
6275 LET K$=INKEY$
6280 IF K$<"1" OR K$>"3" THEN GO TO 6275
6290 BEEP .01,1:GO SUB 6400+100*S(VAL K$)
6300 RETURN
```

Esta rutina calcula la ayuda que recibirá el jugador cuando elija un comodín de héroe. El resultado se almacena en la variable AY y será un número entre -0.8 a 0.8  
Para hallarlo se calcula la media de fichas que tienen sus contrincantes, se resta el número de fichas que tiene él, y se divide entre 10. Las cantidades se limitan para que siempre estén en el rango (-0,8,0,8).

Un número negativo indicara que tiene más fichas que la media, y por tanto recibirá menos ayuda, un número positivo indicará lo contrario.

Más adelante esta variable AY se usará en todos los eventos de personajes para calibrar la ayuda que recibe el jugador. Este detalle es fundamental para mantener el juego más emocionante.

```
6350 REM CALCULA AYUDA
6360 LET SUM=0:LET MAXC=0
6365 FOR K=1 TO NUMJUG
6370 IF K<>JU THEN LET SUM=SUM+C(K):IF C(K)>MAXC THEN LET MAXC=C(K)
6375 NEXT K
6380 LET MEDIA=SUM/(NUMJUG-1)
6385 LET AY=INT(MEDIA-C(JU))
6390 IF AY>8 THEN LET AY=8
6395 IF AY<-8 THEN LET AY=-8
```

```
6400 LET AY=AY/10
6410 RETURN
```

A partir de aquí comienzan las rutinas de comodín o de héroe. Son 12 bloques, uno por personaje, que van de la línea 6500 a la 7660. Algunos personajes como por ejemplo Phantomas o Dizzy, se quedaron fuera, pero había que ahorrar memoria 😞

```
6500 REM HORACIO
6505 LET ALEA=INT(RND*100)+1
6507 IF ALEA<70*(1+AY) THEN GO TO 6515
6510 IF ALEA<100 THEN GO TO 6530
6513 GO TO 6550
6515 GO SUB 5700:IF REFUSE=1 THEN GO TO 6550
6520 LET M$="HORACIO HAMBRIENTO SE JAMA UNA FICHA DE "+C$+" CONTRINCANTE":GO
SUB 3300:GO SUB 7900: GO SUB 5710:RETURN
6530 IF C(JU)=0 THEN GO TO 6550
6535 LET M$=" AY! A HORACIO LE PICA UNA ARA{N}A":GO SUB 7940:RETURN
6550 LET M$="HORACIO SE VA A ESQUIAR":GO SUB 3300:GO SUB 7900:RETURN
```

```
6600 REM JHONY JONES
6605 LET ALEA=INT(RND*100)+1
6607 IF ALEA<40 THEN GO TO 6615
6610 IF ALEA<80*(1+AY) THEN GO TO 6630
6613 GO TO 6650
6615 LET M$="JOHNY JONES ENCUENTRA EL ELIXIR DE LA VIDA (TM).":GO SUB 7930:RETURN
6630 LET M$="ENCUENTRAS TODAS LAS LETRAS: BABALIBA!":GO SUB 7930:RETURN
6650 IF C(JU)=0 THEN GO TO 6615
6655 LET M$="TE CAE (OTRA VEZ!) LA GOTA DE ACIDO DE ABU SIMBEL":GO SUB 7940:RETURN
```

```
6700 REM WILLY
6705 LET ALEA=INT(RND*100)+1
6707 IF ALEA<70*(1+AY) THEN GO TO 6715
6710 GO TO 6730
6715 LET M$="MANIC MINER EXCAVA UNA GALERIA":GO SUB 3300:GO SUB 5600:RETURN
6730 LET M$="TENEMOS QUE HACER UN QUIRKAFLEEG":GO SUB 3300:GO SUB 7900:GO SUB
5850:RETURN
```

```
6800 REM SIR FRED
6805 LET ALEA=INT(RND*100)+1
6807 IF ALEA<70 THEN GO TO 6815
6810 GO TO 6840
6815 GO SUB 5340
6817 IF NGAN>1 OR NPER>1 THEN GO TO 6840
6830 LET M$="SIR FRED LE BIRLA UNA FICHA AL GANADOR...":GO SUB 3300:LET OFF=3:LET
BORRA=0:LET M$="Y SE LA DA AL PERDEDOR":GO SUB 3300:GO SUB 7900:GO SUB 5560:GO
SUB 5770:RETURN
6840 LET M$="SIR FRED RESCATA A LA PRINCESA":GO SUB 7930:RETURN
```



6900 REM ANT ATTACK  
6905 LET ALEA=INT(RND\*100)+1  
6907 IF ALEA<70 THEN GO TO 6915  
6910 IF ALEA<75\*(1+AY) THEN GO TO 6930  
6912 GO TO 6940  
6917 LET M\$="ATAQUE DE HORMIGAS...":GO SUB 3300:LET OFF=3:LET BORRA=0:LET M\$=C\$+"  
CONTRINCANTE ESTA EN PELIGRO!":GO SUB 3300:GO SUB 7900:GO SUB 5650:RETURN  
6930 IF C(7)=1 THEN GO TO 6940  
6935 LET M\$="MY HERO! SALVAS A TU COMPA{N}ERO":GO SUB 3300:LET OFF=3:LET  
BORRA=0:LET M\$="VIDA EXTRA PARA TI Y PARA OTRO":GO SUB 3300:GO SUB 7900:GO SUB  
5750:GO SUB 5500:RETURN  
6940 IF C(JU)=0 THEN GO TO 6950  
6945 LET M\$="BITTEN! LAS MALDITAS HORMIGAS TE DEJAN KAO":GO SUB 7940:RETURN  
6950 LET M\$="LIFE HACK -> RAID(TM): LAS MATA BIEN MUERTAS!":GO SUB 7930:RETURN

7000 REM CAPITAN SEVILLA  
7003 LET AUX=0  
7005 FOR W=1 TO NUMJUG:LET AUX=AUX+C(W):NEXT W  
7010 IF AUX=0 THEN GO TO 7020  
7015 LET M\$="LANZA UNA MORCILLA RADIATIVA (IBERICA)":GO SUB 3300:LET OFF=3:LET  
BORRA=0:LET M\$="A UN JUGADOR ALEATORIO":GO SUB 3300:GO SUB 7900:GO SUB 5820  
:RETURN  
7020 LET M\$="CAPITAN SEVILLA":GO SUB 3300:LET OFF=3:LET BORRA=0:LET M\$="SE PIRA  
VOLANDO Y PASA DE LARGO":GO SUB 3300:GO SUB 7900:RETURN

7100 REM PHILIP MARLOWE  
7105 LET ALEA=INT(RND\*100)+1  
7107 IF ALEA<70\*(1+AY) THEN GO TO 7115  
7110 GO TO 7130  
7115 LET M\$="LA BELLA TANYA TE LLEVA A TU OBJETIVO":GO SUB 7930:RETURN  
7130 IF C(JU)=0 THEN GO TO 7115  
7135 LET M\$="LA PERVERSA VANYA TE JOROBA LA MISION":GO SUB 7940:RETURN

7200 REM SABOTEUR  
7205 LET ALEA=INT(RND\*100)+1  
7207 IF ALEA<70\*(1+AY) THEN GO TO 7215  
7210 GO TO 7230  
7215 LET SAB=1  
7220 LET M\$="PILLAS EL DISQUETTE (5 1/4). A JUIR EN HELICOPTERO!":GO SUB 7930:RETURN  
7230 IF C(JU)=0 THEN GO TO 7215  
7235 LET M\$="UN PERRO TE MUERDE Y TE DEJA TIESO. MISION ABORTADA":GO SUB  
7940:RETURN

7300 REM MONTY MOLE  
7305 LET ALEA=INT(RND\*100)+1  
7307 IF ALEA<70\*(1+AY) THEN GO TO 7315  
7310 GO TO 7330  
7315 GO SUB 5340  
7317 IF NPER>1 THEN GO TO 7330  
7320 LET M\$="MONTY ES INOCENTE":GO SUB 3300:LET OFF=3:LET BORRA=0:LET M\$="VIDA  
EXTRA PARA EL PERDEDOR":GO SUB 3300:GO SUB 7900:GO SUB 5770:RETURN

7330 LET M\$="AUF WIEDERSEHEN MONTY O SEA HASTA LUEGO LUCA!":GO SUB 3300:LET  
OFF=3:LET BORRA=0:LET M\$="NO PASA NADA":GO SUB 3300:GO SUB 7900:RETURN

7400 REM WALLY

7405 LET ALEA=INT(RND\*100)+1

7407 IF ALEA<25 THEN GO TO 7440

7410 IF ALEA<65\*(1-AY) THEN GO TO 7430

7412 GO TO 7440

7415 LET M\$="WALLY ENCUENTRA LA LLAVE DEL DESPERTADOR":GO SUB 7930:RETURN

7430 GO SUB 5700:IF REFUSE=1 THEN GO TO 7415

7435 LET M\$="LOS CANIBALES SE MERIENDAN A LA FAMILIA WEEK":GO SUB 3300:LET  
OFF=3:LET BORRA=0:LET M\$=C\$+" CONTRINCANTE PIERDE UNA VIDA":GO SUB 3300:GO SUB  
7900: GO SUB 5710:RETURN

7440 IF C(JU)=0 OR C(JU)+MAXC>8 OR MAXC-C(JU)>4 THEN GO TO 7415

7445 LET M\$="CAMBIAS A WALLY POR WILMA":GO SUB 3300:LET OFF=3:LET BORRA=0:LET  
M\$="INTERCAMBIAS TUS FICHAS":GO SUB 3300:GO SUB 5900:RETURN

7500 REM SABREMAN

7505 LET ALEA=INT(RND\*100)+1

7507 IF ALEA<35 THEN GO TO 7515

7510 IF ALEA<75\*(1+AY) THEN GO TO 7530

7512 GO TO 7540

7515 LET M\$="COMPLETAS EL AMULETO!(EN LA ULTIMA JODIDA CUEVA)":GO SUB  
7930:RETURN

7530 LET M\$="CONSIGUES LOS 14 INGREDIENTES DEL CALDERO!":GO SUB 7930:RETURN

7540 IF C(JU)=0 THEN GO TO 7515

7545 LET M\$="AUUUU EL LOBO WULF SALTA SOBRE TI":GO SUB 7940:RETURN

7600 REM ERIC

7605 LET ALEA=INT(RND\*100)+1

7607 IF ALEA<30\*(1-AY) THEN GO TO 7615

7610 IF ALEA<80\*(1-AY) THEN GO TO 7630

7612 GO TO 7660

7615 IF C(JU)=0 THEN GO TO 7660

7620 LET M\$="800 LINEAS ERIC!":GO SUB 7940:RETURN

7630 LET AUX=0

7635 FOR W=1 TO NUMJUG:LET AUX=AUX+C(W):NEXT W

7640 IF AUX=0 THEN GO TO 7660

7650 LET M\$="QUE PESTE!.ERIC LANZA UNA BOMBA FETIDA":GO SUB 3300:LET OFF=3:LET  
BORRA=0:LET M\$="A UN JUGADOR ALEATORIO":GO SUB 3300:GO SUB 7900:GO SUB 5820  
:RETURN

7660 LET M\$="ERIC (POR FIN) BESA A HAYLEY":GO SUB 7930:RETURN

Rutina usada por todos los eventos de héroe para realizar la pausa que hay antes de que se ejecute el evento de personaje. La línea 7905 es para hacer la pequeña broma de simular el sonido de un helicóptero para el evento Saboteur:

7900 REM PAUSA

```

7910 PRINT AT 9,1;INK JU;"PULSA UNA TECLA PARA CONTINUAR"
7915 IF INKEY$="" THEN GO TO 7915
7920 PRINT AT 9,1;INK JU;"PROCESANDO ...      "
7923 IF SAB=1 THEN FOR K=25 TO 1 STEP -1:BEEP .001,K: PAUSE 3:NEXT K:LET SAB=0
7925 RETURN

```

Esta subrutina es usada por muchos eventos de héroe para gestionar las vidas extra que se concederán, pueden ser una dos o tres, la variable AY influye en ello, favoreciendo generalmente al que va perdiendo:

```

7930 GO TO 7930+INT(RND*2)+1
7931 IF C(7)>2 AND AY>=0.3 THEN GO SUB 3300:LET OFF=3:LET BORRA=0:LET M$="TRES VIDAS
EXTRA PARA TI!":GO SUB 3300:GO SUB 7900:GO SUB 5750:GO SUB 5750:GO SUB
5750:RETURN
7932 IF C(7)>1 AND AY>0 THEN GO SUB 3300:LET OFF=3:LET BORRA=0:LET M$="DOS VIDAS
EXTRA PARA TI!":GO SUB 3300:GO SUB 7900:GO SUB 5750:GO SUB 5750:RETURN
7933 GO SUB 3300:LET OFF=3:LET BORRA=0:LET M$="VIDA EXTRA PARA TI!":GO SUB 3300:GO
SUB 7900:GO SUB 5750:RETURN

```

Esta subrutina es usada por muchos eventos de héroe para gestionar las vidas que perderá el jugador. Puede ser una o dos, la variable AY influye en ello, y desfavorece al que va ganando:

```

7940 IF AY<=-0.4 THEN GO SUB 3300:LET OFF=3:LET BORRA=0:LET M$="PIERDES DOS
VIDAS":GO SUB 3300:GO SUB 7900:GO SUB 5540:GO SUB 5540:RETURN
7945 GO SUB 3300:LET OFF=3:LET BORRA=0:LET M$="PIERDES UNA VIDA":GO SUB 3300:GO
SUB 7900:GO SUB 5540:RETURN

```

A partir de aquí comienzan las DATA. Todos los campos de los DATA son alfanuméricos, es decir, deben ir entrecomillados.

La estructura de datos de las DATA de JUEGOS es:

- Nombre del juego
- Número que indica la empresa que realizó el juego (es el número de orden en la DATA EMPRESAS)
- Fecha en la que se realizó el juego.
- Es un número que indica el programador que hizo ese juego (número de orden en la DATA PROGRAMADORES). Un cero indica que no se contempla al programador.

```

8000 REM JUEGOS
8001 DATA "FAIRLIGHT","21","1985","1"
8002 DATA "THE GREAT ESCAPE","1","1986","10"
8003 DATA "R-TYPE","16","1988","0"
8004 DATA "TURBO ESPRIT","15","1986","18"
8005 DATA "MOVIE","6","1986","0"
8006 DATA "THE TRAP DOOR","22","1986","5"
8007 DATA "BOMB JACK","10","1986","21"

```

8008 DATA "ANT ATTACK","13","1983","23"  
8009 DATA "AVALON","7","1984","25"  
8010 DATA "HEAD OVER HEELS","1","1987","12"  
8011 DATA "NEBULUS","7","1987","11"  
8012 DATA "SIR FRED","32","1986","20"  
8013 DATA "WHERE TIME STOOD STILL","1","1988","10"  
8014 DATA "DUN DARACH","25","1985","7"  
8015 DATA "RANARAMA","7","1987","25"  
8016 DATA "THREE WEEKS IN PARADISE","19","1986","4"  
8017 DATA "DRAGONTORC","7","1985","25"  
8018 DATA "THE SENTINEL","5","1987","16"  
8019 DATA "ALIEN 8","14","1985","0"  
8020 DATA "SABOTEUR II","15","1987","2"  
8021 DATA "JACK THE NIPPER II","4","1987","0"  
8022 DATA "ATIC ATAC","14","1983","0"  
8023 DATA "PHANTIS","3","1987","0"  
8024 DATA "SABOTEUR!","15","1985","2"  
8025 DATA "THANATOS","15","1986","18"  
8026 DATA "CHASE H.Q.","1","1989","0"  
8027 DATA "GHOST 'N GOBLINS","10","1986","14"  
8028 DATA "HIGHWAY ENCOUNTER","24","1985","3"  
8029 DATA "TARGET: RENEGADE","6","1988","24"  
8030 DATA "TRANZ AM","14","1983","0"  
8031 DATA "AUF WIEDERSEHEN MONTY","4","1987","0"  
8032 DATA "DYNAMITE DAN","17","1985","0"  
8033 DATA "MIDNIGHT RESISTANCE","1","1990","0"  
8034 DATA "BRUCE LEE","2","1984","0"  
8035 DATA "KOKOTONI WILF","10","1984","0"  
8036 DATA "COOKIE","14","1983","0"  
8037 DATA "RENEGADE","6","1987","17"  
8038 DATA "VIAJE AL CENTRO DE LA TIERRA","8","1989","0"  
8039 DATA "LA PULGA (BUGABOO)","13","1983","0"  
8040 DATA "EQUINOX","19","1986","22"  
8041 DATA "JACK THE NIPPER","4","1986","0"  
8042 DATA "SKOOL DAZE","27","1984","0"  
8043 DATA "BACK TO SKOOL","27","1985","0"  
8044 DATA "COBRA","1","1986","13"  
8045 DATA "BATMAN","1","1986","12"  
8046 DATA "URIDIUM","7","1986","0"  
8047 DATA "CYBERUN","14","1986","0"  
8048 DATA "COMMANDO","10","1985","0"  
8049 DATA "MATCH DAY","1","1984","12"  
8050 DATA "DYNAMITE DAN II","17","1986","0"  
8051 DATA "ARKANOID","6","1987","17"  
8052 DATA "OPERATION WOLF","1","1988","0"  
8053 DATA "THE WAY OF THE EXPLODING FIST","9","1985","27"  
8054 DATA "LA ABADIA DEL CRIMEN","11","1988","20"  
8055 DATA "ABU SIMBEL PROFANATION","3","1985","8"  
8056 DATA "CAULDRON II","23","1986","0"  
8057 DATA "FREDDY HARDEST","3","1987","0"  
8058 DATA "MANIC MINER","30","1983","15"  
8059 DATA "GREMLINS","29","1985","0"

8060 DATA "JET SET WILLY","26","1984","15"  
8061 DATA "MATCH DAY II","1","1987","12"  
8062 DATA "MATCH POINT","20","1984","0"  
8063 DATA "GUNFRIGHT","14","1986","0"  
8064 DATA "PHANTOMAS","3","1986","0"  
8065 DATA "TIR NA NOG","25","1984","7"  
8066 DATA "LIVINGSTONE SUPONGO","11","1987","6"  
8067 DATA "ALIEN HIGHWAY","24","1986","0"  
8068 DATA "FAIRLIGHT II","21","1986","1"  
8069 DATA "EL MISTERIO DEL NILO","11","1987","0"  
8070 DATA "EVERYONE'S A WALLY","19","1985","0"  
8071 DATA "GAUNTLET","2","1987","0"  
8072 DATA "GAUNTLET II","2","1988","0"  
8073 DATA "JETPAC","14","1983","0"  
8074 DATA "PHEENIX","33","1983","0"  
8075 DATA "GREEN BERET","6","1986","13"  
8076 DATA "YIE AR KUNG-FU","6","1985","0"  
8077 DATA "SHADOWFIRE","28","1985","0"  
8078 DATA "BATMAN: THE MOVIE","1","1989","17"  
8079 DATA "NODES OF YESOD","31","1985","26"  
8080 DATA "ANDROID TWO","24","1983","3"  
8081 DATA "T.L.L.","24","1984","3"  
8082 DATA "KNIGHT LORE","14","1984","0"  
8083 DATA "FRANKIE GOES TO HOLLYWOOD","1","1985","0"  
8084 DATA "SCUBA DIVE","15","1983","18"  
8085 DATA "CAMELOT WARRIORS","3","1986","8"

9000 REM EMPRESAS

9001 DATA "OCEAN SOFTWARE"  
9002 DATA "US GOLD LTD"  
9003 DATA "DINAMIC SOFTWARE"  
9004 DATA "GREMLIN GRAPHICS SOFTWARE"  
9005 DATA "FIREBIRD SOFTWARE"  
9006 DATA "IMAGINE SOFTWARE"  
9007 DATA "HEWSON CONSULTANTS"  
9008 DATA "TOPO SOFT"  
9009 DATA "MELBOURNE HOUSE"  
9010 DATA "ELITE SYSTEMS LTD"  
9011 DATA "OPERA SOFT S.A."  
9012 DATA "ZIGURAT SOFTWARE"  
9013 DATA "QUICKSILVA LTD"  
9014 DATA "ULTIMATE PLAY THE GAME"  
9015 DATA "DURELL SOFTWARE LTD"  
9016 DATA "ELECTRIC DREAMS SOFTWARE"  
9017 DATA "MIRRORSOFT LTD"  
9018 DATA "MARTECH GAMES LTD"  
9019 DATA "MIKRO-GEN LTD"  
9020 DATA "SINCLAIR RESEARCH"

9021 DATA "THE EDGE"  
9022 DATA "PIRANHA"  
9023 DATA "PALACE SOFTWARE"  
9024 DATA "VORTEX SOFTWARE"  
9025 DATA "GARGOYLE GAMES"  
9026 DATA "SOFTWARE PROJECTS"  
9027 DATA "MICROSPHERE"  
9028 DATA "BEYOND SOFTWARE"  
9029 DATA "ADVENTURE INTERNATIONAL"  
9030 DATA "BUG-BYTE SOFTWARE"  
9031 DATA "ODIN COMPUTER GRAPHICS"  
9032 DATA "MADE IN SPAIN"  
9033 DATA "MEGADODO SOFTWARE"

[Lista de programadores. El segundo campo no se utiliza.](#)

9500 REM PROGRAMADORES  
9501 DATA "BO JANGEBORG" ,"1"  
9502 DATA "CLIVE TOWNSEND" ,"2"  
9503 DATA "COSTA PANAYI" ,"3"  
9504 DATA "DAVID PERRY" ,"4"  
9505 DATA "DON PRIESTLEY" ,"5"  
9506 DATA "GONZO SUAREZ" ,"6"  
9507 DATA "GREG FOLLIS" ,"7"  
9508 DATA "HERMANOS RUIZ" ,"8"  
9509 DATA "IAN WEATHERBURN" ,"9"  
9510 DATA "JOHN HEAP" ,"10"  
9511 DATA "JOHN PHILLIPS" ,"11"  
9512 DATA "JON RITMAN" ,"12"  
9513 DATA "JONATHAN SMITH" ,"13"  
9514 DATA "KEITH BURKHILL" ,"14"  
9515 DATA "MATTHEW SMITH" ,"15"  
9516 DATA "MIKE FOLLIN" ,"16"  
9517 DATA "MIKE LAMB" ,"17"  
9518 DATA "MIKE RICHARDSON" ,"18"  
9519 DATA "PABLO ARIZA" ,"19"  
9520 DATA "PACO MENENDEZ" ,"20"  
9521 DATA "PAUL HOLMES" ,"21"  
9522 DATA "RAFFAELLE CECCO" ,"22"  
9523 DATA "SANDY WHITE" ,"23"  
9524 DATA "SIMON BUTLER" ,"24"  
9525 DATA "STEVE TURNER" ,"25"  
9526 DATA "STEVE WETHERILL" ,"26"  
9527 DATA "THE OLIVER TWINS" ,"27"  
9528 DATA "WILLIAM TANG" ,"28"

9700 REM PREG. BASIC  
9701 DATA "HACER UN COMENTARIO" ,"REM" ,"COMMENT" ,"EXP" ,"EXPLAIN" ,"TITLE"  
9702 DATA "REALIZAR UNA PAUSA" ,"PAUSE" ,"STOP" ,"DELAY" ,"STEP" ,"NOP"  
9703 DATA "EJECUTAR UN PROGRAMA" ,"RUN" ,"EXE" ,"START" ,"COM" ,"INI"

9704 DATA "AVERIGUAR LA LONGITUD DE UNA  
CADENA","LEN","LONG","LON","CAD","EXTEND"  
9705 DATA "CONVERTIR UNA CADENA EN UN VALOR  
NUMERICO","VAL","NUM","VALUE","CONV","CONVERT"  
9706 DATA "CONVERTIR UN NUMERO EN UNA  
CADENA","STR\$","CAD","CONV","CONVERT","STRING"  
9707 DATA "GENERAR UN NUMERO ALEATORIO","RND","RAND","RANDOMIZE","ALEA","JACTA"  
9708 DATA "DECLARAR UNA MATRIZ","DIM","MAT","MATRIX","DECLARE","DOM"  
9709 DATA "BORRAR LA PANTALLA","CLS","CLEAR","RESET","NOSCREEN","CLEARSCR"  
9710 DATA "OBTENER LA PARTE ENTERA DE UN  
NUMERO","INT","ENT","TRUN2C","INTEGER","NODEC"  
9711 DATA "LEER UNA PULSACION DE TECLADO","INKEY\$","READ","SCAN","KEYB","KEY"  
9712 DATA "PRODUCIR UN SONIDO","BEEP","SOUND","THX","BIP","MUSIC"

9800 REM PREG. ASSEMBLER

9801 DATA "CARGAR VALORES EN UN REGISTRO","LD","LOAD","VALUE","REGISTER","LOADING"  
9802 DATA "REALIZAR UN SALTO RELATIVO A OTRA  
INSTRUCCION","JR","JP","JUMPR","JRELATIVE","BRINK"  
9803 DATA "REALIZAR UN SALTO ABSOLUTO A OTRA  
INSTRUCCION","JP","JR","JUMP","ABS","SALT"  
9804 DATA "SUMAR DOS REGISTROS","ADD","SUM","CHECKSUM","MERGE","COPULE"  
9805 DATA "RESTAR DOS REGISTROS","SUB","RES","SUBSTRACT","SUBS","REST"  
9806 DATA "ROTAR BITS A LA IZQUIERDA","RLC","ROT","YIRA","RLEFT","ROTATEL"  
9807 DATA "DESPLAZA BITS A LA DERECHA","SRL","SLIDER","RIGHT","BITR","DESP"  
9808 DATA "PONER UN BIT A CERO","RES","ZERO","NONE","BLIND","NOBIT"  
9809 DATA "PONER UN BIT A UNO","SET","SEM","CAM","JAFET","SETI"  
9810 DATA "SUMAR UNO A UN REGISTRO","INC","SUM1","PLUS","PLUSONE","INC++"  
9811 DATA "RESTAR UNO A UN REGISTRO","DEC","REST","RES1","MINUS","DOWN"  
9812 DATA "VOLVER DE UNA SUBROUTINA","RET","RETURN","EXIT","GU-ERBE","FINISH"

9900 REM PERSONAJES

9901 DATA "HORACIO"  
9902 DATA "JHONY JONES"  
9903 DATA "WILLY"  
9904 DATA "SIR FRED"  
9905 DATA "ANT ATTACK"  
9906 DATA "CAPITAN SEVILLA"  
9907 DATA "PHILIP MARLOWE"  
9908 DATA "SABOTEUR"  
9909 DATA "MONTY MOLE"  
9910 DATA "WALLY"  
9911 DATA "SABREMAN"  
9912 DATA "ERIC"

9940 REM PREMIO1

9941 DATA "UCHI MATA"  
9942 DATA "ONE ON ONE"  
9943 DATA "SABRINA"  
9944 DATA "HIGHLANDER"

9945 DATA "KUNG FU MASTER"  
  
9960 REM PREMIO2  
9961 DATA "UN BOLLICAO"  
9962 DATA "UN FRIGOPIE"  
9963 DATA "UN PITAGOL"  
9964 DATA "UNA PELOTA DE NIVEA"  
9965 DATA "UN CHICLE DE FRESA ACIDA"  
9966 DATA "UN CHIMO SABOR... MORADO"  
9967 DATA "UN PALODU"  
9968 DATA "UN CIGARRO DE CHOCOLATE"  
9969 DATA "UN CARAMELO DE CUBA LIBRE"  
9970 DATA "UN PETA ZETA"

## ANEXO I : VARIABLES

En este apartado describimos las principales variables del programa

### Variables Globales

**FILAS:** número de filas que tiene el tablero de juego  
**COLUMNAS:** número de columnas que tiene el tablero de juego  
**NUMCUAD:** número de cuadros que tiene el tablero de juego  
**NUMJUG:** Número de jugadores, por defecto 2  
**MAXJUG:** Número máximo de jugadores que se admiten en el juego; por defecto 4.  
**NUMOPC:** Número de opciones de respuesta que se mostrarán. Por defecto 4  
**ANOACTUAL:** Año actual. Se usa en la rutina de generación de fechas.  
**NEGROSINI:** Número de cuadros negros iniciales que se colocarán en el tablero. Por defecto 20.  
**P1, P2, P3, P4, P5:** Probabilidad en tanto por ciento de aparición de cada tipo de pregunta.

Las siguientes variables van por pares. Señalan la posición inicial de las DATA, y el número de ocurrencias de cada DATA. Por ejemplo, INIJUEGOS señala el comienzo de las DATAS donde están los nombres de los juegos (línea 8000) y MAXJUEGOS contiene el número de juegos (110). De esta forma se puede cambiar fácilmente las DATA del programa:

**INIJUEGOS**  
**MAXJUEGOS**  
**INIEMPRESAS**  
**MAXEMPRESAS**  
**INIAUTOR**  
**MAXAUTOR**  
**INIBASIC**  
**MAXBASIC**  
**INIASSEMBLER**  
**MAXASSEMBLER**  
**INIPERSONAJES**  
**MAXPERSONAJES**



## Variables locales

**T(X,Y):** es una matriz, que almacena lo que hay en cada casilla del tablero. Puede contener:  
Número del jugador dueño de la pieza que coincide con su código de color (1-4)  
Un 0 , que indica un cuadro vacío (inicialmente todo se inicializa a cero)

**C():** es el número de fichas que tiene cada jugador en el tablero. En C(1),C(2),C(3) y C(4) se almacena el número de fichas que tiene cada jugador. En C(7) se almacena el número de cuadros vacíos. No se contabiliza el número de cuadros “negros” (de muro)

**JU:** Contiene el número del jugador activo en cada momento

**P() y Q()** variables donde se almacenan las coordenadas de las fichas de los jugadores(primer subíndice 1-4). Pej P(3,1), Q(3,1) se corresponden con las coordenadas de una ficha colocada en el tablero del jugador 3. P(4,2),Q(4,2) contendría la segunda ficha del jugador 4. El índice 5 se reserva para almacenar las coordenadas de los cuadros blancos en la subrutina 4100 (ver comentarios de esta rutina). Estas variables aceleran el proceso de buscar dichas coordenadas, solo usando la variable T() sería inviable.

**C\$:** variable auxiliar que se usa para colocar uno u otro texto en función de que sea dos o que sean más jugadores.

**HORM:** Flag para saber si se está ejecutando la rutina de “ataque de hormigas”

**SAB:** Flag para saber si se está ejecutando la rutina de “helicóptero saboteur”

**BLANCOS:** Flag para saber si se terminó de recopilar todos los blancos en la rutina 4100

**MIT:** Flag para saber si se terminó de recopilar la primera parte de la rutina 4100

**CONTBL:** contador de cuadros blancos (vacíos)

**OFF:** offset para imprimir más abajo en la rutina de impresión en pantalla

**BORRA:** flag para saber si la rutina de impresión en pantalla debe borrar o no

## Otras variables

**JU** número de jugador actual y color del jugador actual (coincide)

**PR** contiene la casilla colindante que se va a explorar (ver comentarios en la rutina correspondiente)

**X e Y** son las coordenadas en formato numérico, pej A3=1,3; C8= (3,8)

**X1 e Y1** se usan como copias de X e Y antes de lanzarlas a la subrutina “detecta próximo propio” (4400)

**PROXIMO**= si es 1 es que hay un próximo propio y si no es que no lo hay

**COD** = almacena el jugador actual (1-4), y además 6=ficha negra; 0= cuadro vacío).

**O\$( )** Variable donde se almacenarán las posibles opciones de respuesta. (2855 DIM  
O\$(NUMOPC,23))

**OK:** variable que contiene el número de la opción correcta