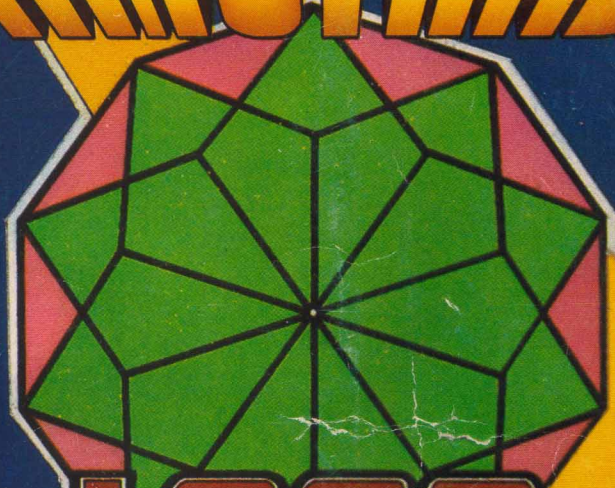


AMSTRAD



LOGO

микро БТЕ

LOGO

PARA EL ORDENADOR AMSTRAD CPC-464

© 1984 S. J. Wainwright

RESERVADOS TODOS LOS DERECHOS

Ninguna parte de este manual o del programa puede ser reproducido por ningún medio sin el permiso por escrito del autor o del editor.

Este programa se suministra con la creencia de que trabaja como se especifica, pero MICROBYTE S.A. (la compañía) no se considerará responsable por alguna pérdida o daño directa o indirecta de la propiedad incurrida o sufrida por el cliente o por cualquier otra persona como resultado de algún fallo o defecto, en bienes o servicios suministrados por la compañía, y en ninguna circunstancia la compañía será responsable por el daño o pérdida consecuente de beneficios (de todos modos la posibilidad de los mismos fue separadamente considerada o razonablemente prevista), originada del uso o ejecución de tales medios o servicios.

Publicado por MICROBYTE, S.A.
Paseo de la Castellana, 179
28046 Madrid

LOGO: INTÉRPRETE DE GRÁFICOS DE TORTUGA

por S. J. Wainwright. 1984

CONTENIDO

Introducción	5
La Tortuga	5
Implementación del LOGO-GRÁFICOS	6
Comenzando	6
Procedimientos	6
Comandos	6
Parámetros	7
Modo inmediato	7
Procesos primarios disponibles:	
hideturtle	8
showturtle	8
penup	8
pendown	8
forward	9
reverse	9
draw	9
back	9
erase	9
jump	9
left	9
right	9
circle	9
fill	10
colour	10
clear	10
home	10
pause	10

size	10
angle	11
reset	12
repeat	12
continue	12
moveto	13
La definición de procesos definidos por el usuario utilizando 'to'	13
end	13
edit	15
text	16
viewtext	16
cleartext	16
showtext	16
Programando una secuencia repetitiva de procesos definidos por el usuario utilizando el proceso 'prog'	16
execute	17
Usando 'size' y 'angle' dentro de definiciones de procesos	17
Grabando o cargando procesos definidos por el usuario y mensajes de texto en cinta utilizando el proceso 'sys'	19
Los procesos de cálculo	19
add	19
subtract	19
multiply	19
divide	20
remainder	20
quotient	20
raise	20
procs	20
menu	20
stop	20
Ejemplos	20

© S. J. Wainwright 1984

LOGO

INTÉRPRETE DE GRÁFICOS DE TORTUGA

INTRODUCCIÓN

Logo Gráficos es un instrumento para la geometría exploratoria práctica. El control de la tortuga en la pantalla da experiencia en longitudes, direcciones, ángulos, radios y colores. El usuario puede descubrir la estructura de formas geométricas, puede experimentar con figuras y ganar valiosa experiencia en la estimación de cantidades tales como distancia y ángulos. Al actuar con el ordenador, el usuario gana confianza en el uso del ordenador como instrumento y puede utilizarlo, por ejemplo, para descubrir soluciones alternas a problemas específicos.

El uso de Turtle Graphics estimula el pensamiento matemático y la discusión entre los usuarios y es particularmente útil para trabajo en equipo. La idea de programación puede ser introducida en el contexto de “enseñar” nuevos procedimientos que, al ser “aprendidos”, pueden utilizarse para realizar una serie de acciones complicadas. Programar con Logo-Gráficos es una actividad creadora que pueden llevar a cabo grupos de cualquier edad y grupos de edades mixtas. Nuevos procedimientos pueden ser grabados en cinta para uso futuro. Facilidades de cálculo están incluidas para liberar al usuario del uso de otros medios de cálculo.

LA TORTUGA

La Tortuga se ve en la pantalla como una figura redonda con un puntero de dirección. Con el uso de varios comandos puede hacerse que la Tortuga mire en cualquier dirección, se mueva por toda

la pantalla dejando una huella según avanza, borrando un trazo previamente dibujado, o saltando de un lugar a otro sin dejar rastro. La Tortuga puede también dibujar círculos alrededor de sí misma, o llenar discos de varios colores y dibujar líneas de varios colores. La Tortuga puede ser visible o invisible. Si es visible, el proceso de dibujar líneas, etc., es más lento que si es invisible.

Mensajes de texto pueden mostrarse en la parte superior de la pantalla y son llamados desde procesos definidos por el usuario. Por ejemplo, si la Tortuga dibuja una casa, el mensaje “Casa” puede mostrarse en lo alto de la pantalla si se requiere.

IMPLEMENTACIÓN DEL LOGO-GRÁFICOS

Comenzando

Cargue el intérprete colocando la cinta en el cassette y escribiendo chain“logo” . El intérprete, después de cargado, se ejecutará automáticamente. Se le preguntará si quiere resolución m o b. Es mejor responder con m a menos que desee usar varios colores. Si responde con b los dibujos serán de una resolución más baja, pero tendrá disponible 16 colores.

Procedimientos

Esta implementación de Logo-Gráficos tiene un diccionario conteniendo 44 procedimientos primarios (suministrados con el sistema). Cada procedimiento o comando tiene un nombre, por ejemplo, ‘forward’ (adelante) o ‘left’ (izquierda).

Comandos

Se da un comando cuando el nombre de un procedimiento es escrito (junto con un parámetro si se requiere). Cuando se da un comando, el procedimiento se ejecuta.

Parámetros

Muchos de los procedimientos requieren parámetros, por ejemplo, números que han de ser escritos después del nombre del procedimiento. Por ejemplo, el procedimiento 'draw' (dibujar) requiere un parámetro que especifica la longitud de la línea que ha de ser dibujada.

Modo inmediato

Los comandos se escriben en respuesta a preguntas en la pantalla de "Nombre de proceso?". El usuario a continuación escribe el nombre del proceso requerido. Si éste requiere un parámetro, la pregunta en pantalla cambiará pidiendo un parámetro. Por ejemplo, cuando la pregunta es "Nombre de proceso?", si escribimos 'draw', nos preguntará "Distancia?". Si escribimos un número, digamos, 100, la Tortuga se moverá hacia adelante (en la dirección en la que está apuntando) una distancia de 100 unidades de pantalla, y dibujará una línea según avanza. La pantalla tiene 640 unidades de izquierda a derecha y 400 unidades de arriba a abajo.

Algunos procesos, como 'clear' no requieren parámetros. En estos casos, cuando el nombre del proceso se escribe en modo inmediato se ejecuta inmediatamente.

En definiciones de procesos, sin embargo, el nombre de cada uno de ellos debe ir seguido de una coma y después un valor, por ejemplo: "draw, 100". Esto se aplica aunque el proceso no requiera un parámetro. En estos casos se ignora el parámetro y es a menudo conveniente usar un cero como parámetro ficticio, por ejemplo: "clear, 0". Esto se analizará en definiciones de procesos.

Procesos primarios disponibles

Muchos de los procesos primarios tienen abreviaturas que pueden ser intercambiadas libremente por el nombre completo. Estas abreviaturas se muestran debajo entre paréntesis, después del nombre completo del proceso. Si un proceso está disponible sólo

en modo inmediato se indica por (IMO). Si por otra parte, un proceso está disponible sólomente dentro de un proceso definido por el usuario se indica por (UDPO).

hideturtle (ht) Hace que la Tortuga se vuelva invisible. Todos los movimientos y dibujos de líneas son mucho más rápidos cuando la Tortuga es invisible.

showturtle (st) Hace que la Tortuga se vuelva visible. Esto decelera el proceso de movimiento alrededor de la pantalla y de dibujar líneas. Es mejor tener una tortuga visible cuando se trabaja en modo inmediato porque se puede ver cada movimiento que hace. Es normal eleborar los detalles de movimiento que habrán de ser incorporados en procesos definidos por el usuario usando una Tortuga visible. (Es buena práctica tomar notas mientras se experimenta de esta manera con una Tortuga visible). Sin embargo, al ejecutarse procesos definidos por el usuario debe esconderse la Tortuga para obtener una ejecución más rápida. Puede hacerse bien inmediatamente antes de llamar un proceso, o usar el comando “ht,0” como el primero de los definidos por el usuario.

penup (pu) Levanta una pluma imaginaria de la pantalla. Si se mueve la Tortuga hacia adelante con el proceso “forward” lo hará sin dibujar una línea.

pendown (pd) Coloca una pluma imaginaria en la pantalla. Si se mueve la Tortuga con el proceso “forward”, lo hará dibujando una línea a su paso.

forward n (adelante n)	(fd)	Mueve la Tortuga hacia adelante (en la dirección a la que está apuntando) una distancia de n unidades. Dependiendo de la situación “pendown” o “penup” en ese momento, dibujará una línea o no.
reverse n (atrás n)	(rvs)	Mueve la Tortuga hacia atrás una distancia de n unidades, dibujando una línea a su paso. La Tortuga permanece mirando en la misma dirección en que estaba antes de darse el comando “reverse”.
draw n (dibujar n)	(dr)	Mueve la Tortuga hacia delante una distancia de n unidades dibujando una línea a su paso.
back n (atrás n)	(bk)	Mueve la Tortuga hacia atrás una distancia de n unidades, borrando lo que encuentra a su paso. La Tortuga queda con la misma orientación que tenía antes.
erase n (borrar n)	(er)	Mueve la Tortuga adelante una distancia de n unidades borrando a su paso cualquier línea anteriormente dibujada.
jump n (saltar n)	(jp)	Hace saltar a la Tortuga hacia adelante una distancia de n unidades, sin dibujar ni borrar nada a su paso.
left n (izqda. n)	(lt)	Mueve la Tortuga a la izquierda (con respecto a la dirección a la que está apuntando en ese momento) n grados.
right n (dcha. n)	(rt)	Mueve la Tortuga a la derecha (con respecto a la dirección a la que está apuntando en ese momento) n grados.
circle (círculo n)	(cr)	Dibuja un círculo de radio n unidades con la Tortuga en su centro.

fill n (llenar n)	(fl)	Dibuja un disco sólido de radio n unidades con la Tortuga en el centro.
colour n (color n)	(col)	Las subsiguientes líneas, circunferencias o círculos serán del color indicado por n, siendo éste del código de colores normales de Amstrad. Los colores disponibles dependerán de la resolución elegida al cargar el Intérprete.
clear (borrar)	(cl)	Borra la pantalla y vuelve a centrar la Tortuga mirando a la derecha.
home (origen)	(h)	Deja la pantalla intacta, pero vuelve a centrar la Tortuga mirando a la derecha.
pause n (pausa n)	(ps)	Pausa de n segundos.
size n (tamaño)	(si)	Añade n a una variable que controla el tamaño. El valor de esta variable se suma automáticamente al parámetro de los procesos 'draw' y 'forward' si existe la situación 'pen-down' antes de ejecutarse aquellos dos. La variable se iguala a cero al cargar el Intérprete y haciendo uso de 'reset'. El proceso 'size' puede ser usado para asignar parámetros a los procesos 'draw' y 'forward' aun cuando éstos estuviesen dentro de un proceso definido por el usuario. Esto significa que los procesos repetitivos pueden ser definidos con sus procesos 'draw' y 'forward' en cero, y el tamaño de la figura que habrá de dibujarse puede ser determinado por el uso de 'size' antes de llamar al proceso.

A manera de ejemplo:

```
size
100
draw
0
```

dibujará una línea de 100 unidades de longitud.

Debe recordar volver a dar el valor cero a la variable de 'size' utilizando el comando 'reset', después de haber usado el primero, de lo contrario la Tortuga no dibujará las líneas de la longitud deseada.

angle
n
(ángulo n)

(an) Este proceso es similar a 'size' pero suma n a una variable que controla los ángulos. El valor de esta variable se suma automáticamente a los parámetros de 'left' o 'right' antes de ejecutarlos. La variable se iguala a cero al cargar el Intérprete y usando el comando 'reset'. El proceso 'angle' puede asignar parámetros a los procesos 'left' y 'right' aun dentro de un proceso definido por el usuario.

A manera de ejemplo:

```
angle
45
right
0
```

hará que la Tortuga gire a la derecha 45°.

Debe recordar volver a dar el valor cero a la variable de 'angle' después de usar este proceso.

reset (re) Iguala las variables de control de 'size' y (iniciar) 'angle' a cero.

repeat (rpt) El proceso 'repeat' se usa en conjunción n con 'continue' para repetir una secuencia de **procesos primarios** n veces. Los bucles de 'repeat'-'continue' pueden hacerse a pares. En modo inmediato cuando se escribe 'repeat' la pregunta en pantalla cambia a "número de veces?". A continuación se escribe n, la cantidad de veces. A partir de este momento todos los comandos subsiguientes deben ser escritos como nombres de procesos seguidos de una coma y un valor de parámetro (real o ficticio).

Por ejemplo:

```
repeat
  2
repeat,3
draw,60
left,120
continue,0
jump,80
continue,0
```

O lo mismo usando abreviaturas:

```
rpt
2
rpt,3
dr,60
lt,120
cont,0
jp,80
cont,0
```

continue (cont) Usado en conjunción con 'repeat' como se (continuar) mostró antes.

moveto x,y (mover a x,y)	(mt) El proceso 'moveto' toma dos parámetros, x, y, que son las coordenadas en la pantalla; se escriben separadas por una coma. Al ejecutarse la Tortuga se mueve a la posición especificada por las coordenadas.
--------------------------------	---

La definición de procesos definidos por el usuario utilizando 'to'

Cuando se escribe el comando 'to' el Intérprete pasa al modo de compilación y pide el nombre del proceso que ha de definirse. Después los comandos que componen la definición del nuevo proceso se escriben en secuencia. Cada comando se escribe como un nombre de proceso, una coma y un valor. Si un proceso en la definición no requiere parámetro, se le asigna un parámetro **ficticio**. Cero es un parámetro ficticio apropiado. La definición termina escribiendo 'end,0'

Nuevos procesos se compilan en el diccionario y se ejecutan cada vez que sus nombres se escriben.

Un proceso definido por el usuario es un proceso de **primera generación** si solamente procesos **primarios** son usados en la definición.

Debajo aparecen dos ejemplos de procesos de primera generación llamados 'case' y 'face'. Estos dos procesos de primera generación serán incorporados en la definición de un proceso de segunda generación llamado 'clock'.

to	to
case	face
ht,0	ht,0
jp,100	rpt,12
rt,90	jp,80
jp,100	rvs,20
rpt,4	bk,60
rt,90	lt,30
dr,200	cont,0
cont,0	cr,90

(continuación)

```
h,0          rpt,48
end,0        dr,60
             bk,60
             rt,30
             cont,0
             dr,60
             h,0
             end,0
```

Debe notar en los ejemplos anteriores, que, aunque ‘ht’, ‘h’, ‘cont’ y ‘end’ no requieren parámetros, tuvieron que ser escritos con parámetros ficticios (0) en la definición.

Un proceso definido por el usuario es un proceso de **segunda generación** si contiene uno o más procesos de primera generación en su definición. Por ejemplo:

```
to
clock
case,0
lt,90
jp,110
fl,10
h,0
face,0
end,0
```

El proceso ‘clock’ es un proceso de segunda generación porque contiene los procesos de primera generación ‘case’ y ‘face’ en su definición. Debe notar que procesos primarios y de primera generación pueden mezclarse en la definición de procesos de segunda generación. Los procesos de segunda generación no pueden ser usados para definir procesos ulteriores. Fíjese que los procesos de primera generación ‘case’ y ‘face’ se escribieron con parámetros ficticios ‘0’, porque los procesos definidos por el usuario no usan parámetros directamente. El usuario debe definir los procesos

'case' y 'face' y después 'clock' como se mostró, y escribir 'clock'. Un reloj será dibujado en la pantalla y una manecilla dará vueltas.

edit (ed) (IMO) Se llama al editor escribiendo 'edit'. El editor es un medio que permite cambios simples en las definiciones de procesos después de haber sido compilados en el diccionario. Al llamarse al editor escribiendo 'edit', se le pide al usuario que escriba el nombre del proceso que se va a editar. Al escribirle, aparece en la pantalla la definición del proceso, con un número precediendo cada comando. El usuario tiene la opción de renunciar o de seguir adelante con la edición. Si el usuario elige editar el proceso, se le pide el número del comando a ser reemplazado y el comando que le reemplazará; que consiste en un nombre de proceso, coma, y un parámetro (real o ficticio). Después el usuario tiene la opción de renunciar a la edición o de editar otra línea.

Puesto que el editor no permite la inserción de líneas, es mejor incluir uno o dos comandos ficticios antes de 'end,0' al definir un proceso. Un ejemplo de esto sería:

'ficticio,0'

El intérprete ignora comandos no reconocidos, por lo que los ficticios no afectarán la ejecución de procesos definidos.

Sin embargo, reservan espacio para cualquier inserción que pueda ser necesaria más adelante. Los comandos ficticios pueden usarse como comentarios mientras tengan el formato correcto.

text (texto)	(IMO)	Permite escribir un mensaje de texto en el fichero de textos. Este puede guardar hasta 15 mensajes. Pueden ser llamados por procesos definidos por el usuario que hacen que el mensaje de textos se muestre en el borde superior de la pantalla. Cuando se escribe el comando 'text' en modo inmediato, el ordenador pide el número de mensaje. Puede ser entre 1 y 15. Al escribirle, el ordenador pide el mensaje. Este puede tener hasta 2 líneas de longitud.
viewtext (ver texto)	(vt) (IMO)	Muestra todos los mensajes de texto en la pantalla.
cleartext (borrar texto)	(clt) (UDPO)	Limpia la pantalla de textos de mensajes.
showtext n (mostrar texto)	(sh) (UDPO)	Hace que el mensaje número n aparezca en el borde superior de la pantalla. Este proceso es disponible sólo desde procesos definidos por el usuario solamente. Por ejemplo, supongamos que el mensaje 1 fue definido como 'Un reloj funcionando' por medio del proceso 'text'. Sería posible entonces que apareciera este mensaje durante la ejecución de 'clock', definido por el usuario anteriormente, haciendo el primer comando de 'clock' que fuera 'sh,1'.

Programando una secuencia repetitiva de procesos definidos por el usuario usando el proceso 'prog'

El proceso 'prog' se utiliza para programar una secuencia repetitiva de procesos definidos por el usuario que han sido compilados

en el diccionario. 'Prog' se comporta como 'repeat', pero utiliza solamente procesos definidos por el usuario y ningún parámetro.

prog (IMO) El proceso 'prog' se utiliza en conjunción con 'execute' para repetir una secuencia de procesos definidos por el usuario n veces (donde n puede ser igual a 1). Por ejemplo, supongamos que hemos definido previamente los procesos 'triangle', 'ball' y 'leap' como sigue:

to	to	to
triangle	ball	leap
rpt,3	fl,20	jp,90
dr,60	end,0	end,0
lt,120		
cont,0		
end,0		

Podríamos usar 'prog' para dibujar un triángulo, una bola, otro triángulo, otra bola, como sigue:

```
prog
2
triangle
leap
ball
leap
execute
```

execute (ex) (IMO) Utilizado en conjunción con 'prog'.

Usando 'size' y 'angle' dentro de definiciones de procesos.

Los procesos 'size' y 'angle' pueden utilizarse para cambiar progresivamente los parámetros de los procesos 'draw', 'forward' cuando 'pendown' es cierto y 'left' & 'right', respectivamente, como sigue:

Tres ejemplos

```
to
spiral
rpt,100
dr,20
lt,15
size,-0.2
cont,0
reset,0
end,0
```

Dibujo resultante



```
to
grow
rpt,50
dr,10
rt,90
size,4
cont,0
reset,0
end,0
```

Dibujo resultante



```
to
snail
rpt,80
dr,20
lt,15
size,-0.2
angle,0.25
cont,0
reset,0
end,0
```

Dibujo resultante



Grabando o cargando procesos definidos por el usuario y mensajes de texto en cinta utilizando el proceso 'sys'

sys (IMO) Al escribir el comando 'sys' el usuario tiene la opción de cargar procesos y mensajes previamente grabados en una cinta o de grabar nuevos procesos definidos y mensajes de texto en cinta.

Una vez dados los comandos apropiados el ordenador tarda algunos segundos en organizar los ficheros y el usuario debe esperar las preguntas del ordenador.

Los procesos de cálculo

add n1, n2
(sumar $n1 + n2$) Suma $n1$ a $n2$
Con los procesos de cálculo, después de escribir el nombre de la operación, por ejemplo 'add', el ordenador imprime éste en la parte inferior de la pantalla. El usuario después escribe los dos parámetros, separados por una coma. Después el ordenador muestra la respuesta en la parte superior de la pantalla. Se le da, entonces, al usuario la opción de dejar la respuesta en pantalla o de borrar la respuesta.

subtract (sub) Resta $n1$ de $n2$
n1, n2
(restar $n1 - n2$)

multiply (mult) Multiplica $n1$ por $n2$
n1, n2
(multipl. $n1 \times n2$)

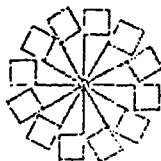
divide n1,n2 (divide n1/n2)	(div)	Divide n1 entre n2
remainder n1,n2 (resto de n1/n2)	(rem)	Muestra el resto de n1/n2
quotient n1,n2 (cociente de n1/n2)	(quo)	Muestra el cociente de n1/n2
raise n1,n2 (elevar n1^n2)		<p>Eleva n1 a la potencia n2.</p> <p>Debe fijarse que el proceso 'raise' puede usarse por ejemplo, para obtener, raíces cuadradas; por ejemplo,</p> <p style="padding-left: 40px;">raise 16,0.5</p> <p>dará una respuesta de 4.</p>
procs (procesos)	(IMO)	Lista los nombres de los procesos definidos por el usuario.
menu (menú)	(IMO)	Muestra los nombres y abreviaturas de los 44 procesos primarios.
stop (parar)	(IMO)	Detiene el Intérprete y regresa el ordenador a BASIC.

Ejemplos

Los siguientes son cuatro procesos definidos que ilustran algunos de los resultados que pueden lograrse con el INTÉRPRETE DE GRÁFICOS DE TORTUGA LOGO:

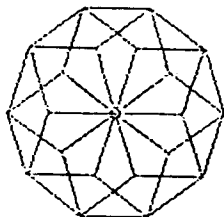
```
to
wheel
repeat,12
draw,60
repeat,4
draw,30
right,90
continue,0
reverse,60
right,30
continue,0
end,0
```

Dibujo resultante



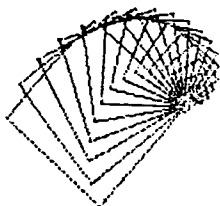
```
to
diamond
repeat,10
right,36
repeat,5
draw,80
right,72
continue,0
continue,0
end,0
```

Dibujo resultante



```
to
fan
repeat, 15
repeat,4
draw,10
left,90
continue,0
left,10
size,10
contine,0
reset,0
home,0
end,0
```

Dibujo resultante



```
to
star
repeat,12
repeat,4
draw,80
left,90
continue,0
left,30
continue,0
end,0
```

Dibujo resultante

