

### 3. Introducción al lenguaje de programación Benu

En el tema anterior construimos un fichero FPG con todos los gráficos que vamos a utilizar en nuestro videojuego. Ya estamos listos para comenzar a hacer uso de él y empezar a ver en pantalla nuestros gráficos.

#### 3.1 Concepto de lenguaje de programación

Un lenguaje de programación consiste en una serie de palabras clave que, correctamente asociadas y ordenadas, nos permiten dialogar con el procesador expresando el comportamiento que queremos obtener para nuestro programa o videojuego.

El lenguaje Benu posee una gran cantidad de palabras clave, con distinta utilidad, potencia y complejidad, que iremos introduciendo poco a poco durante este tutorial. Si es la primera vez que te encuentras con un lenguaje de programación debes saber que en general son una mezcla entre la lengua inglesa y las matemáticas. No te dejes intimidar por su aparente complejidad, seguro que vas a disfrutar mucho utilizándolo.

#### 3.2 Concepto de proceso en Benu

Existen muchos tipos de lenguajes de programación. Java es un lenguaje orientado a objetos, C es un lenguaje imperativo, Benu en cuestión es un poco especial, según la teoría de lenguajes de programación puede considerarse un **lenguaje orientado al proceso**.

Para hacernos una idea, todo elemento que haya en pantalla es un proceso: Nuestro protagonista, cada disparo, cada enemigo...

El lenguaje nos permite **definir** el comportamiento de un proceso, por ejemplo el de un disparo, y éste puede ser **invocado** tantas veces como queramos de una manera tan sencilla como es escribir su nombre. Es algo que veremos a fondo en los siguientes temas, por ahora vamos a trabajar a fondo con un único proceso, que definiremos como queramos, y que será invocado tan sólo 1 vez.

### 3.3 Aspectos básicos de un proceso en Benu

Todo proceso en Benu tiene una serie de aspectos que determinan cómo lo visualizaremos en la pantalla, o incluso si estará visible en la pantalla o fuera de ella.

A continuación vamos a enumerar los aspectos básicos de un proceso en Benu. Todos los aspectos tienen un determinado valor numérico. Te darás cuenta de que con ellos y con nuestro FPG tenemos lo suficiente para modelar prácticamente cualquier comportamiento que hayas visto en cualquier videojuego comercial en 2 dimensiones.

Aspecto	Funcionamiento
graph	Es el aspecto fundamental de un proceso. Es el <b>número de gráfico del FPG</b> que nuestro proceso utilizará para mostrarse en pantalla. Por ejemplo, si hacemos que graph valga 201 y ese número corresponde al gráfico de un misil en nuestro FPG, entonces nuestro proceso se mostrará en pantalla con el aspecto de ese misil. Si no le decimos a nuestro proceso cuál es su valor de graph, entonces graph valdrá 0 y el proceso será invisible.
x	Sirve para determinar la posición de un proceso en pantalla. Generalmente la pantalla tiene una resolución de 640x480 píxeles, aunque Benu nos permitirá cambiarla más adelante. <b>El valor de x es el píxel, medido desde el extremo izquierdo de la pantalla, en el que se mostrará el proceso.</b> Por ejemplo, si hacemos que x valga 320, entonces el proceso se mostrará en el centro de la pantalla horizontalmente. Si no le decimos a nuestro proceso cuál es su valor de x, entonces x valdrá 0 y el proceso aparecerá en el extremo izquierdo de la pantalla.
y	De forma similar a x, sirve para determinar la posición de un proceso en pantalla. <b>El valor de y es el píxel, medido desde la parte superior de la pantalla, en el que se mostrará el proceso.</b> Por ejemplo, si hacemos que y valga 240, entonces el proceso se mostrará en el centro de la pantalla verticalmente. Si no le decimos a nuestro proceso cuál es su valor de y, entonces y valdrá 0 y el proceso aparecerá en el extremo superior de la pantalla.
size	Determina cuál es el tamaño con el que se dibujará el proceso en pantalla. Observa que la pantalla tiene una cierta resolución en píxeles y cada gráfico del FPG un cierto tamaño en píxeles. <b>El valor de size indica en tanto por ciento (%) el tamaño con el que veremos el gráfico en pantalla.</b> Por ejemplo, si hacemos que size valga 200 y el graph del proceso tiene un tamaño de 20x20 píxeles, entonces el proceso se dibujará en pantalla con un tamaño de 40x40 píxeles, un 200% más grande que su tamaño original. Si no le decimos a nuestro proceso cuál es el valor de size, entonces size valdrá 100 y el proceso se dibujará con el tamaño original de su gráfico.

<b>angle</b>	Indica cuál es el ángulo de giro con el que se dibujará el proceso en pantalla. Gracias a este aspecto, con un único gráfico de un balón podremos hacer que éste gire. <b>El valor de angle indica en milésimas de grado la inclinación con la que se dibujará el gráfico en pantalla.</b> Por ejemplo si hacemos que angle valga 180000 el proceso se verá girado 180 grados, boca abajo. Si no le decimos a nuestro proceso cuál es su valor de angle, entonces angle vale 0 y el proceso se dibujará normalmente. La razón por la cual angle se mide en milésimas de grado es para permitir giros con mucha precisión, sobre todo en gráficos grandes, como pudiera ser un escenario.
--------------	--

### 3.4 Aspectos avanzados de un proceso en Benu

A continuación una serie de aspectos de un proceso que no utilizaremos por el momento, pero que nos permitirán más adelante programar comportamientos más complejos de una manera mucho más cómoda.

Aspecto	Funcionamiento
<b>flags</b>	Este aspecto avanzado permite que el proceso sea dibujado aplicando distintos efectos gráficos. Su valor numérico no tiene ninguna unidad de medida, simplemente cada número ofrece un efecto distinto y hay que consultar esta tabla para conocer su funcionamiento. <b>Valores posibles de flags son: 0 para dibujado normal, 1 para dibujar el gráfico espejado horizontalmente, 2 para dibujar el gráfico espejado verticalmente, 4 para dibujar el gráfico con una transparencia del 50%, 8 para dibujar el gráfico con una transparencia del 90%</b> , y el resto los probaremos más adelante, en general no se usan los flags a partir del 8. Los valores de flags corresponden a potencias de 2, esto permite que puedan sumarse entre ellos obteniendo la aplicación de todos los valores sumados. Por ejemplo si hacemos que flags valga 7 estaremos aplicando $1+2+4$ , y el proceso se dibujará espejado horizontalmente, espejado verticalmente y además con una transparencia del 50%. Si no indicamos cuál es el valor de flags, entonces flags vale 0 y el proceso se dibuja normalmente.
<b>z</b>	Este aspecto avanzado es útil cuando tenemos una gran cantidad de procesos distintos en pantalla. En esos casos observaremos que unos procesos son dibujados "por encima" o "por debajo" de otros, y no siempre obtendremos el orden de dibujado esperado. <b>El valor de z indica la profundidad de dibujado</b> , de forma que procesos con valores mayores de z se dibujarán por debajo de los procesos con menores valores de z. Por ejemplo si a nuestro proceso protagonista le damos un valor de z que vale 1 y a un proceso aura situado sobre el protagonista le damos un valor de z que vale 2, entonces veremos el aura detrás de nuestro protagonista. Sin embargo si intercambiamos los valores de z veremos el aura por delante de nuestro protagonista, ocultando su gráfico. Si no especificamos el valor de z, entonces z vale 0 y los procesos se dibujan con el orden que decida el procesador.

Aspecto	Funcionamiento
<b>ctype</b>	Este aspecto avanzado es útil cuando tenemos un escenario o scroll cuyo tamaño en píxeles es superior a la resolución en píxeles de la pantalla. En ese caso el scroll "no cabe" en la pantalla y sólo podemos ver una determinada zona del scroll cada vez. Si esto sucede, es probable que nos interese que los aspectos básicos x,y de nuestro proceso no tengan como origen la esquina superior izquierda de la pantalla, sino que tengan como origen la esquina superior izquierda del escenario o scroll. Si no especificamos el valor de ctype, entonces <b>ctype vale 0 y las coordenadas x,y tienen como origen la esquina superior izquierda de la pantalla</b> . En cambio <b>si hacemos que ctype valga 1, entonces las coordenadas x,y tienen como origen la esquina superior izquierda de nuestro scroll</b> .

### 3.5 Ejercicio teórico: Los aspectos de un proceso Benu en un videojuego comercial

Piensa en Galaxian, un sencillo arcade de los años 80, e intenta adivinar el posible valor de los aspectos de proceso en Benu que corresponderían para cada uno de los elementos que muestra en pantalla.

Piensa después en The Legend of Zelda, un RPG más complejo desarrollado para SNES en los años 90, e imagina cómo cambiarían los valores de los aspectos, especialmente los avanzados como ctype, que facilitaría calcular las coordenadas x,y de los diferentes procesos que recorren el escenario o que se mantienen fijos en pantalla.



Imagen 1: Galaxian, un arcade de los años 80



Imagen 2: The Legend of Zelda, un RPG de los años 90