

4. El lenguaje: Asignaciones

En el tema anterior explicamos los aspectos básicos de cualquier proceso gráfico en Bennu. Estos eran:

Aspectos básicos	
graph	Su número de gráfico dentro del fichero .fpg que creamos en las primeras sesiones.
x,y	Su posición horizontal y vertical medida en pixels respecto de la esquina superior izquierda de la pantalla.
size	Su tamaño medido en % respecto del tamaño original de su graph (Por defecto 100).
angle	Su ángulo medido en milésimas de grado (Por defecto 0).
Aspectos avanzados	
flags	Efectos visuales (0 = Normal, 1 = Espejo horizontal, 2 = Espejo vertical, 4 = Transparencia 50%, etc.)
ctype	Indica si las coordenadas x,y son relativas a la pantalla o al scroll (Valores 0 ó 1 respectivamente).
z	Indica la profundidad de dibujado, que indica si un proceso se dibuja por encima o por debajo de otro.
file	Especifica el fichero .fpg en el que se buscará el graph (Sólo cuando usemos múltiples ficheros .fpg).

4.1 Definición de asignación

Una asignación establece un valor numérico a cualquiera de los aspectos de un proceso. Su sintaxis es la siguiente:

Aspecto = Valor; //Es importante no olvidar el símbolo ';' (Un punto y coma) que indica el fin de la asignación.

A continuación añadiremos las siguientes asignaciones entre las etiquetas BEGIN y END del proceso principal de nuestro videojuego y comprobaremos que el funcionamiento es exactamente el indicado en la tabla anterior.

graph=1;	El proceso toma el gráfico número 1 de nuestro fichero .fpg.
x=100;	Posiciona el proceso a 100 pixels del lateral izquierdo de la pantalla.
y=200;	Posiciona el proceso a 200 pixels de la parte superior de la pantalla.
size=50;	Cambia el tamaño del proceso a un 50% de su tamaño original.
angle=90000;	El proceso se muestra rotado 90 grados (Recuerda que angle se mide en milésimas de grado).

4.2 Primeras pruebas con nuestro videojuego Bennu

En la carpeta Videojuego que lleva asociada este tema, encontrarás un programa tan sencillo que no puede considerarse un videojuego, pero que nos servirá para realizar todas las pruebas iniciales y será la base para el videojuego que crearemos a medida que avancemos en este tutorial.

El contenido de la carpeta es el siguiente:

DLL	En esta carpeta se encuentra toda la maquinaria (Archivos DLL) que hace que nuestro videojuego pueda ejecutarse. Por el momento no tocaremos nada de esa carpeta.
images	En esta carpeta se guarda el/los ficheros para gráficos (FPG) del videojuego. Por ahora trabajaremos con el FPG que tiene y en el próximo tema lo sustituiremos por el nuestro.
videojuego.prg	Este archivo contiene el código fuente que editaremos para añadir nuevas instrucciones. Podrías editarlo perfectamente con el Bloc de Notas, aunque si instalas la utilidad Bennupack podrás editarlo con el programa Notepad++, que resulta mucho más cómodo de utilizar e incluye algunas otras utilidades interesantes.
Ejecutar en Windows.bat	Comprueba errores en el código (Ésto se llama compilar) y si todo es correcto ejecuta el videojuego.
Ejecutar en Linux.sh	Lo mismo que lo anterior, pero funciona en el sistema operativo Linux.
videojuego.dcb	Este fichero se genera a partir del código cuando compilamos, y es lo que realmente se ejecuta.

Ya estás listo para hacer las pruebas propuestas en este tema y otras que se te puedan ocurrir. No olvides **guardar el archivo .prg** cada vez que modifiques el código, no olvides asegurarte de que el resultado de la compilación **no muestra errores**, y durante tus primeros pasos como programador, no olvides que las instrucciones deben terminarse con un `;`.

Los errores anteriores serán el 99% de tus posibles problemas a la hora de empezar a programar, el resto los iremos viendo más adelante.