

5. El lenguaje: Incrementos

En el tema anterior establecimos valores numéricos fijos a los diferentes aspectos de un proceso. La asignación de valores fijos no es suficiente para poder crear nuestro videojuego, así que ahora aprenderemos a hacer que esos aspectos se modifiquen automáticamente durante la ejecución gracias a las instrucciones de incremento.

5.1 Definición de incremento

Una incremento establece una variación sobre cualquiera de los aspectos de un proceso a partir de una operación matemática que generalmente es muy sencilla, por ejemplo una suma. Su sintaxis es la siguiente:

Aspecto = Aspecto + Valor; //Una vez más, no olvides el ':' (Punto y coma).

Prueba a añadir las siguientes asignaciones entre las etiquetas LOOP y END del proceso protagonista de tu videojuego y comprueba que el incremento da resultado.

graph=graph+1;	El proceso toma el siguiente número de gráfico del fichero .fpg cada vez que se muestra en pantalla.
x=x+4;	El proceso se desplaza 4 píxeles hacia la derecha cada vez que se muestra en pantalla.
y=y-1;	El proceso sube 1 píxel cada vez que se muestra en pantalla.
size=size+2;	El proceso aumenta en un 2% su tamaño cada vez que se muestra en pantalla.
angle=angle-1000;	El proceso rota 1 grado en sentido antihorario cada vez que se muestra en pantalla.

5.2 Aclaraciones sobre las asignaciones y los incrementos

Tanto las asignaciones como los incrementos son instrucciones, y por tanto siempre deben situarse entre las etiquetas BEGIN y END, que indican el principio y el final de la zona de instrucciones que ejecuta un proceso.

Los procesos tiene un único bloque de instrucciones BEGIN – END, y para facilitar su ubicación, la etiqueta BEGIN se tabula exactamente a la misma altura que la etiqueta END que la cierra.

Dentro del bloque BEGIN – END, justo al final del mismo, encontraremos un bloque LOOP – END. Las instrucciones que vienen antes del bloque LOOP – END se ejecutan una única vez al iniciar el proceso y suelen ser asignaciones, mientras que las instrucciones dentro del LOOP – END se ejecutan todo el tiempo y suelen ser incrementos.

5.3 Ejercicio: Uso de los aspectos básicos de un proceso

Con lo aprendido hasta ahora ya deberías saber hacer que tu proceso protagonista aparezca inicialmente en la esquina superior izquierda de la pantalla, y se desplace hasta la esquina inferior derecha al tiempo que gira y aumenta de tamaño.

Truco: La proporción de pantalla en píxeles es 4:3, por tanto avanzando 4 píxeles en horizontal (Eje X) y 3 píxeles en vertical (Eje Y) debería recorrer exactamente la diagonal de la pantalla.

Ejemplos de resolución de pantalla que cumplen la proporción 4:3 son 640x480, 800x600 y 1024x768. Estas resoluciones pueden establecerse modificando los valores de la instrucción `set_mode(800,600,16,MODE_WINDOW)`; que encontrarás en el primer BEGIN del código de tu videojuego.

5.4 Ejercicio: Uso de los aspectos avanzados de un proceso

El aspecto `flags` de un proceso produce efectos visuales muy interesantes. Aprovechando lo aprendido sobre incrementos puedes hacer que tu proceso incremente en 1 el valor de su variable `flags` con la instrucción: `flags=flags+1`;

De esta forma podremos ver la totalidad de los efectos visuales que podremos aplicar modificando esta variable.

Truco: Los diferentes `flags` son apilables, esto significa que cuando `flags` vale 5 estamos aplicando `flags=1` (Espejo horizontal) y `flags=4` (Transparencia 50%) al mismo tiempo, puesto que $1+4=5$. Otro ejemplo más complejo sería cuando `flags` vale 209, que aplicaría los `flags 128+64+16+1` y cuyo efecto sería bastante extraño... En general no usaremos `flags` más allá del 16.