

## 6. El lenguaje: Condiciones

En el tema anterior establecimos incrementos en los aspectos del proceso principal de nuestro videojuego logrando que su gráfico, su posición, su tamaño, etc. Variasen durante la ejecución.

El uso de instrucciones de incremento ofrecen dinamismo en el videojuego, pero el comportamiento que obtenemos siempre es el mismo y no hay ningún tipo de interacción. Para lograr interacción y un resultado mucho más sofisticado debemos hacer uso de las condiciones.

### 6.1 Definición de condición

Su nombre exacto es **condicional**. Consiste en establecer una condición, de manera que, sólo cuando ésta se cumpla se ejecutarán una o más instrucciones, que pueden ser asignaciones, incrementos o incluso otras condiciones.

Los condicionales son algo que utilizaremos extensivamente durante toda nuestra vida como programadores, así que es conveniente que te quede totalmente claro su funcionamiento. Su sintaxis es la siguiente:

```
IF (<condición>
    <Asignaciones, incrementos u otras condiciones, cada una en una línea>
END
```

Observa que hemos encerrado entre <> la condición sin explicar su sintaxis exacta. Esto se debe a que puede haber una gran diversidad de ellas. Vamos a ver unas cuantas en este tema, y más adelante veremos una descripción más formal de todas las condiciones que podemos llegar a usar.

key(_right)	Se cumple cuando se pulsa la tecla direccional derecha (_right) del teclado. Podemos sustituir _right por cualquier otra tecla direccional, cualquier letra o número, _enter, _space, etc. El nombre de la tecla siempre va precedido por el carácter '_'.
x<0	Se cumple cuando el aspecto x (La posición horizontal en píxeles) del proceso tiene un valor menor que 0. Esto ocurre cuando el proceso "se sale" por la izquierda de la pantalla.
y>600	Se cumple cuando el aspecto y (La posición vertical en píxeles) del proceso tiene un valor mayor que 600. Esto ocurre cuando el proceso "se sale" por la parte inferior de la pantalla. Siempre y cuando estemos ejecutando con una resolución de pantalla de 800x600 píxeles.
graph==9	Se cumple cuando el aspecto graph (El número de gráfico del FPG) del proceso tiene un valor exactamente igual a 9. Esta condición nos permite por ejemplo detectar cuándo termina una animación. <u>Observa que se usan 2 símbolos '=' para comprobar (leer), y 1 símbolo '=' para asignar (escribir).</u>

## 6.2 Instrucciones a ejecutar cuando se cumple una determinada condición

Vamos a ver unos cuantos ejemplos típicos de uso de las condiciones:

<pre>IF (x&gt;800)   x=0; END</pre>	<p>Cuando el aspecto x del proceso tiene un valor mayor que 800 píxeles (Se sale por la derecha de la pantalla), se le asigna un 0 a su aspecto x para hacer que vuelva a aparecer por la izquierda.</p>
<pre>IF (key(_down))   y=y+4; END</pre>	<p>Cuando se pulsa la tecla direccional abajo, el proceso incrementa su aspecto y en 4, bajando 4 píxeles cada vez. Serviría para desplazar nuestro proceso por la pantalla.</p>
<pre>IF (key(_right))   graph=graph+1;   IF (graph&gt;9)     graph=1;   END END</pre>	<p>Cuando se pulsa la tecla direccional derecha, el proceso incrementa su gráfico en 1 (Recorre 1 a 1 los gráficos del FPG y por tanto se anima).</p> <p>Si además de pulsar la tecla direccional derecha el valor de graph se hace mayor que 9, volvemos a asignar el graph 1, logrando así que el proceso utilice la animación cuyos gráficos van del 1 al 9.</p>

Ten en cuenta que las nuevas instrucciones condicionales que estamos añadiendo pueden entrar en conflicto con incrementos o asignaciones anteriores que pudiera haber en nuestro proceso. Asegúrate de que entiendes lo que ocurrirá cuando añadas nuevas instrucciones, y si notas algún comportamiento extraño, no dudes en borrar todas las instrucciones y volver a empezar.

## 6.3 Ejercicio: Programar los controles de un juego de naves

Con las indicaciones dadas en los ejemplos anteriores ya deberías ser capaz de añadir las condiciones necesarias en el proceso principal de tu videojuego para que se mueva hacia arriba, abajo, izquierda y derecha con las teclas direccionales correspondientes.

Además puedes programarlo de manera que, si el proceso se sale por arriba, por abajo, por la izquierda o por la derecha de la pantalla, vuelva a aparecer por el lado opuesto.

No te compliques demasiado programando la animación, ya que es un tema más complejo de lo que parece a simple vista y lo veremos con detenimiento más adelante.