

24. Durezas

Hasta ahora no hemos tenido en cuenta la interacción física con el escenario. Se trata de algo fundamental en un videojuego el hecho de poder responder a paredes, suelos y techos.

Para ello haremos uso de alguna función nueva y también tendremos que añadir unos gráficos nuevos a nuestro FPG.

24.1 La función `map_get_pixel ()`

Esta función obtiene la información de color del píxel de un gráfico determinado, su sintaxis es la siguiente:

```
int map_get_pixel ( int , int , int , int )
```

Parámetro 1	Identificador del fichero FPG en el que se encuentra el gráfico sobre el cual queremos obtener la información de color de uno de sus píxeles.
Parámetro 2	Número de gráfico dentro del FPG.
Parámetro 3	Coordenada x del píxel que queremos leer.
Parámetro 4	Coordenada y del píxel que queremos leer.
Valor de retorno	Retorna un valor numérico que representa el color leído. No podremos apreciar de qué color se trata a partir de este valor numérico, pero sí que podremos utilizarlo para pasárselo a otras funciones, como veremos a continuación.

24.2 Mapas de durezas

Un mapa de durezas es un "clon" de un background que utilizamos como escenario, pero tiene coloreadas las zonas que corresponden a suelos, techos, paredes, etc. En distintos tonos de color uniformes.

Necesitaremos crear un mapa de durezas asociado a cada uno de nuestros escenarios, y para ello haremos una copia de los mismos y sobre ellos colorearemos con Paint.NET u otro programa similar las partes correspondientes de la siguiente manera:

Rojo	Techos	Superficies horizontales en las que queremos que nuestro protagonista golpee y caiga hacia abajo al saltar. Serán infranqueables y no tienen por qué comportarse como suelos, a no ser que añadamos encima de ellas el color de dureza de suelos.
Amarillo	Suelos	Superficies horizontales en las que queremos que nuestro protagonista se apoye y deje de caer hacia abajo. Serán infranqueables y no tienen por qué comportarse como techos, a no ser que añadamos debajo de ellas el color de dureza de techos. Tal y como aplicaremos la gravedad, la máxima pendiente de suelo que reconocerá nuestro personaje dependerá de su velocidad de avance. Normalmente superficies de hasta 30 grados de inclinación funcionarán correctamente.
Azul	Paredes	Superficies verticales que queremos que resulten infranqueables para nuestro protagonista durante su desplazamiento horizontal. Por el momento se comportarán como paredes cuando nos desplacemos a izquierda y cuando nos desplacemos a derecha, aunque más adelante podremos cambiar ésto.

En todos los casos anteriores, el grosor necesario para el dibujado de las durezas dependerá mucho de la velocidad a la que desplacemos nuestro protagonista. Ten en cuenta que nuestro protagonista avanza varios píxeles en cada FRAME, y si avanza más píxeles en un FRAME que el grosor que le damos a la dureza cabe la posibilidad de que la pueda ignorar...

En este tutorial no entraremos en detalle sobre cómo evitar que sucedan situaciones como la anterior, aunque con todo lo que hemos visto es posible solucionarlo si usamos un poco nuestra imaginación.

De momento sólo nos preocupa conseguir que nuestro protagonista sea capaz de detener su movimiento al encontrarse con una pared.

24.3 Implementar la reacción frente a una pared

Para conseguir que nuestro personaje se detenga al encontrarse con una pared, tendremos que añadir unas condiciones que deberán cumplirse antes de realizar su desplazamiento en el eje x.

Nuestro mapa de durezas tiene la resolución exacta de nuestro escenario, y dada una posición x, y de nuestro protagonista en el escenario es posible, gracias a la función `map_get_pixel ()` obtener la información de color de la dureza asociada a ese punto si comprobamos esa coordenada en el mapa de durezas.

La idea es sencilla:

Comprobaremos la dureza en el punto al que vamos a desplazarnos, y sólo nos desplazaremos en el caso de que ese punto no corresponda a una dureza de pared.

Si nuestro gráfico de escenario está numerado como 401, el fondo del mismo como 402 y la dureza asociada al escenario como 403 un código que podría solucionar la detención frente a paredes sería el siguiente:

```
IF ( map_get_pixel ( fpg_escenario , 403 , x + 10 , y ) != <código de color de la dureza de pared> )  
    x = x + 10;  
END
```

Observa nuevamente que antes de desplazarnos 10 píxeles hacia la derecha comprobamos el color de la dureza en ese punto.

El código anterior funcionaría perfectamente, sólo nos falta un detalle. ¿Cuál es el código de color de la dureza de pared?

Para ello haremos uso nuevamente de la función `map_get_pixel ()`. Tendremos un pequeño gráfico en nuestro FPG que guardará los distintos colores de cada dureza en cada uno de sus píxeles. Tal y como los hemos expuesto en la tabla anterior podemos tener en cada píxel la información siguiente:

x	y	Color
0	0	Rojo que utilizamos para las durezas de techo
1	0	Amarillo que utilizamos para las durezas de suelo
2	0	Azul que utilizamos para las durezas de pared
-	-	Resto de colores que podamos asociar a durezas, pueden ser puertas, transportadores, zonas de fin de escenario, pinchos, etc.

Con el mapa anterior, que podemos numerar con el código 999 dentro del FPG de escenarios, podemos cargar dentro de algunas variables los códigos de color asociados a cada tipo de dureza. Es lo que hacemos con la función inicializar_durezas () en el videojuego de ejemplo asociado a este tema.

Si completamos el código encargado de la detección de paredes así:

```
IF ( map_get_pixel ( fpg_escenario , 403 , x + 10 , y ) != durezas [ _PARED ] )  
    x = x + 10;  
END
```

Por fin conseguiremos que nuestro personaje interactue correctamente con las paredes.

Ten en cuenta que sólo interpretamos las paredes de forma horizontal. Observa que si queremos tener un obstáculo en nuestro escenario que se comporte como una columna sólida, además de las durezas de pared tendremos que añadir en su parte más alta las correspondientes durezas de suelo, tal y como veremos en el próximo tema.