# Ejercicios de motores de videojuegos

## Índice

En esta sesión vamos a implementar diferentes componentes de un videojuego con Cocos2D. Tenemos la plantilla `JuegoCocos2D` con la estructura necesaria. La clase donde está implementada la pantalla principal del juego es `GameLayer`. Trabajaremos sobre esta clase.

## 1. Creación de sprites

*a)* En primer lugar crearemos un primer *sprite* para mostrar una roca en una posición fija de pantalla. El *sprite* mostrará la imagen `roca.png`, y lo situaremos en *(240, 250)*. Esto lo haremos en el método `init` de nuestra capa principal.

*b)* Ahora vamos a crear un *sprite* a partir de una hoja de *sprites* (*sprite sheet*). Para ello primero deberemos crear dicha hoja de *sprites* mediante la herramienta TexturePacker (empaquetaremos todas las imágenes que encontremos en el proyecto). Guardaremos el resultado en los ficheros `sheet.plist` y `sheet.png`, y los añadiremos al proyecto. Dentro del proyecto, añadiremos el contenido de esta hoja de *sprites* a la caché de fotogramas, y crearemos a partir de ella el *sprite* del personaje (el nombre del fotograma a utilizar será `pers01.png`), y lo añadiremos a la posición *(240, 37)* de la pantalla.

## 2. Actualización de la escena

*c)* Vamos a hacer ahora que el personaje se mueva al pulsar sobre la parte izquierda o derecha de la pantalla. Para ello vamos a programar que el método `update:` se ejecute en cada iteración del ciclo del juego (esto se hará en `init`). Posteriormente, en `update:` modificaremos la posición del *sprite* a partir de la entrada de usuario (podremos obtener la entrada de la propiedad `self.direction`, que puede indicar que se esté pulsando izquierda, derecha, o ninguno de ellos). Haremos que el *sprite* se mueva a 100 píxeles por segundo en la dirección indicada por la entrada.

## 3. Acciones

*d)* Debemos también conseguir que la piedra se mueva. Haremos que esté continuamente cayendo, y que cuando alcance la parte inferior de la pantalla, vuelva a aparecer arriba. Esto lo haremos mediante acciones. Definiremos en `init` las acciones que hagan que este comporamiento se repita indefinidamente, y lo ejecutaremos sobre el *sprite* de la roca.

## 4. Animación del personaje (*)

*e)* Ahora haremos que el personaje al moverse reproduzca una animación por fotogramas en la que se le vea caminar. Para ello en primer lugar debemos definir las animaciones en `init`. La animación de caminar a la izquierda estará formada por los fotogramas

`pers02.png` y `pers03.png`, mientras que la de la derecha estará formada por `pers04.png` y `pers05.png`. En ambos casos el retardo será de 0.25 segundos. Añadiremos las animaciones a la caché de animaciones. Una vez hecho esto, deberemos reproducir las animaciones cuando andemos hacia la derecha o hacia la izquierda. Podemos hacer esto mediante una acción de tipo `CCAnimate`. Ejecutaremos estas animaciones en los métodos `moverPersonajeIzquierda` y `moverPersonajeDerecha`. En `detenerPersonaje` deberemos parar cualquier animación que esté activa y mostrar el fotograma `pers01.png`.

## 5. Detección de colisiones (*)

*f)* Por último, vamos a detectar colisiones entre el personaje y la roca. En caso de que exista contacto, haremos que la roca desaparezca. Esto deberemos detectarlo en el método `update:`. Obtendremos los *bounding boxes* de ambos *sprites*, comprobaremos si intersectan, y de ser así pararemos todas las acciones de la roca y haremos que desaparezca con una acción de tipo *fade out*.