CREANDO APPS

aprende a programar aplicaciones móviles

PARTE 5

SENSORES

En este módulo vamos a explorar los sensores, es decir, unos componentes electrónicos de nuestro dispositivo móvil que miden distintas cosas. Entre los sensores del celular, vamos a centrarnos en el acelerómetro que nos permite conocer la inclinación del dispositivo. Y nada mejor para explorar el acelerómetro que desarrollar una app de un juego.

5.1 Sensores /

REPRESENTAR DATOS DE ACELERACIÓN

Para empezar nuestra app vamos a aprender a acceder a los datos de aceleración proporcionados por el acelerómetro. Partimos de una plantilla HTML vacía donde ponemos 3 cajas (`div`) para representar las lecturas del acelerómetro. En el CSS colocamos las cajas centradas usando un margen auto a izquierda y derecha.

En el JavaScript accedemos al plugin de Cordova que nos permite acceder al acelerómetro, llamado *device motion*[1]. Para ello, primero escuchamos el evento `deviceready` que nos avisa de cuándo podemos acceder con garantías al acelerómetro. Sobre el objeto `navigator.accelerometer` que nos provee el plugin de Cordova ejecutamos la función `watchAcceleration` para leer el valor de la aceleración cada cierto tiempo. Si hay éxito en la lectura se invocará una función de éxito y si no, una de error, que son los dos primeros parámetros. El tercero será la frecuencia de refresco en milisegundos.

A la función de éxito (`onSuccess`) Cordova le pasa como parámetro los datos de aceleración. Simplemente representamos en el `div` correspondiente del HTML el valor recogido de la aceleración en los ejes x, y, z. Antes de representarlo, redondeamos el número a 2 decimales[2].

Al poner en marcha la aplicación en el dispositivo, comprobamos que el valor de x tiene que ver con la inclinación a izquierda (positivo) y derecha (negativo). Respecto a la y, el valor tiene que ver con la inclinación del terminal hacia delante (negativo) o hacia abajo (positivo). El eje z es espacial y empieza en 9 que es el valor de la fuerza de la gravedad y las variaciones de este parámetro son pequeñas al mover el terminal arriba y abajo en el mismo eje.

Finalmente, podemos observar que si agitamos el dispositivo los valores de aceleración pasan de 10. Desde nuestro JavaScript podemos detectar la agitación con ese umbral y, por ejemplo, pintar el fondo de la app de un color diferente. En el próximo paso usaremos estos valores de aceleración como entrada de datos para el juego.

(1) Cordova plugin device motion `https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-device-motion`

(2) Función de redondeo Math.round `https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Math/round`

CREANDO UN JUEGO CON PHASER

Para crear nuestro juego con el acelerómetro usaremos Phaser[3], una librería para crear juegos en JavaScript. Nos proporciona muchas funcionalidades, pero de momento usaremos el ciclo del juego, el motor de física para que una bolita se mueva con el acelerómetro y la técnica de detección de límites para saber cuando se está saliendo de la pantalla.

Para usar Phaser en nuestra app, enlazamos la librería JavaScript desde el HTML y mejor si usamos la versión minificada (como hemos visto con otras librerías). Necesitamos también una imagen para la bola (en un juego se suele llamar *sprite* a esta imagen que movemos por la pantalla) que ponemos en una carpeta `assets`.

En nuestro fichero de JavaScript vamos a indicar el tamaño de la imagen de la bolita para calcular dónde la hacemos aparecer.

(3) Phaser `https://phaser.io`

También medimos el alto y ancho de la pantalla para hacer un juego que ocupe todo el ancho de la pantalla. En el inicio también llamaremos a la función que vigila la agitación de los sensores que al comienzo tiene una frecuencia de muestreo de los datos del acelerómetro de un segundo pero que bajaremos más adelante para mejorar la jugabilidad.

Por otro lado, veamos cómo controlamos con Phaser el ciclo del juego. De momento vamos a usar 2 estados:

- Preload: para preparar datos antes de juego. Nosotros vamos a pintar el color de fondo del juego y a cargar la imagen de la bolita.

- Create: prepara el inicio del juego. Nosotros pintamos la bolita en una posición aleatoria de la pantalla calculando con su tamaño que no se pinte fuera.

Con toda esta información (tamaño de pantalla, id del elemento HTML donde se pinta el juego y estados) ya podemos arrancar nuestra primera versión de juego que sólo pintará la bolita en

una posición aleatoria de la pantalla.

Continuamos con el juego, esta vez agregando nuevas funcionalidades al ciclo de juego:

- Preload: además de lo anterior, también arrancamos el motor de física de Phaser, en modo ARCADE.
- Create: añadimos que el motor de física actúe sobre la bola.
- Update: se actualiza cada cierto tiempo para mover cosas del juego. En este caso, tomamos los datos leídos del acelerómetro y se los aplicamos al *sprite* de la bola. En el caso de la posición x e y usamos un factor multiplicador sobre la velocidad de 300 de forma empírica para que funcione con fluidez. Además, para la x incluimos un factor negativo porque es contrario al sentido de la aceleración que leemos del acelerómetro.

Los datos del acelerómetro los vamos actualizando en variables x e y cuando tiene éxito la lectura del acelerómetro. La frecuencia de lectura de datos del acelerómetro la hemos bajado a 10 milisegundos para manejar la bolita con fluidez.

Ahora vamos a añadir una puntuación al juego, que depende de si tocamos con la bolita en los bordes de la pantalla. Para ésto hemos hecho modificaciones en el ciclo del juego:

- Create: creamos un elemento `scoreText` que posicionamos en la pantalla para mostrar la puntuación, y depende de una variable puntuación. Sobre el sprite de la bolita indicamos a Phaser que detecte colisiones con los bordes de la pantalla (`collideWorldBounds`). Y cuando suceda este evento de colisión con el borde (`onWorldBounds`) ejecutamos una función manejadora para decrementar la puntuación en el `scoreText` .

Al probar el juego en el terminal, vemos que la bolita se mueve y al chocar contralos bordes de la pantalla ya no se sale pero sí se decrementa la puntuación. Además, la bolita se dibuja encima del texto (puntuación del juego) ya que en Phaser aparece encima lo que se pinta más tarde. De momento tenemos un juego simplón con escasa jugabilidad, pero vamos a mejorarlo en la siguiente parte.

5.3 Sensores /

MEJORANDO LA JUGABILIDAD

En primer lugar, hemos implementado una forma de reiniciar el juego cuando se agita el terminal. Cuando accedemos a acelerómetro y detectamos agitación, se ejecuta una función de recomienzo que recarga la página y por tanto reinicia el juego. No lo hacemos inmediatamente sino después de un tiempo, usando la función `setTimeout` de Javascript que nos permite ejecutar una función pasado un tiempo en milisegundos.

Para mejorar la jugabilidad, vamos a usar un nuevo sprite para dibujar un objetivo al que alcanzar con la bolita para subir nuestra puntuación. Para conseguirlo hemos modificado el ciclo del juego:

- Preload: añadimos la precarga de una nueva imagen, que previamente hemos almacenado en la carpeta `assets`.

- Create: creamos un nuevo *sprite* para el objetivo, y lo colocamos en una posición aleatoria de la pantalla. Además, sobre este nuevo *sprite* también aplicamos las leyes de la física ARCADE.

- Update: con el motor de físicas de Phaser detectamos la colisión de la bolita con el objetivo, y cuando sucede se llama a la función `incrementaPuntuacion`. Esta función modifica la puntuación en el `scoreText` de la pantalla.

Al probar el juego en el terminal comprobamos que es muy poco jugable ya que puedo posicionarme sobre el objetivo de forma constante. Necesitamos mejorar la jugabilidad para completar nuestro juego. Para esto añadimos un factor de dificultad, que en el update del ciclo de juego vamos a usar para aplicarlo sobre la velocidad de la bola en el eje `x` e `y`. En incrementa puntuación, además de mostrar la nueva en pantalla, volvemos a repintar el objetivo en una nueva posición de la pantalla y aumentamos el nivel de dificultad (sólo si estamos en puntuación positiva).