Sonido en videojuegos

FMOD Core API

En primer lugar crearemos un proyecto básico de FMOD en C++. Debemos localizar la capeta de instalación de FMOD en el equipo (en particular el Core API), enlazar la librería estática de FMOD (lib), poner la librería dinámica (dll) accesible al ejecutable de nuestro proyecto y incluir las cabeceras *.hpp* en el programa (ver diapositivas de clase). Para comprobar que todo funciona correctamente, cargar un sonido, asociarlo a un canal y reproducirlo. A continuación realizar los siguientes ejercicios.

1. El archivo `Battle.wav` contiene un sonido ambiental de combate, y *Gun1.wav* y *Gun2.wav* dos sonidos de disparo. Reproducir la primera muestra en loop y las otras dos en instantes y posiciones aleatorias en el plano para conseguir un efecto realista de batalla.

2. En este ejercicio utilizaremos el archivo *motor.wav* e implementaremos un sencillo bucle que permita interactuar con el usuario para simular el sonido de un motor a distintas revoluciones. Puede conseguirse este efecto subiendo y bajando dinámicamente el pitch de dicha muestra utilizando el canal asociado. ¿Cómo se escucha a medida que el pitch se aleja más de su valor natural (1.0)?

3. El archivo *footstep.wav* contiene el sonido de una pisada. Editarlo con Audacity para obtener un loop de pasos. Después cargarlo en FMOD y realizar un programa que permita mover la fuente sonora en el espacio con un control sencillo de teclado (wasd).

4. El archivo *rifle.wav* contiene el sonido de varios disparos de rifle. En primer lugar convertir la pista a mono, a continuación realzar los graves (ecualizado y/o añadiendo más sonidos) para .$^{\text{en}}$grosar.$^{\text{el}}$ sonido y después separar cada disparo aislado en una pista distinta. Utilizar estos sonidos para crear en FMOD un efecto de disparo que elija aleatoriamente una de estas muestras. Para darle aun más variedad, introducir una ligera variación aleatoria de pitch cada vez que se reproduzca una de las muestras.

5. En este ejercicio haremos un pequeño piano virtual con FMOD. Utilizaremos el "do" de piano de la muestra *piano.ogg* y el propio teclado para simular las teclas del piano. Comenzaremos con una octava utilizando las teclas "zxcvbnm,", de modo que `'z'` corresponde al propio do, `'x'` a re, hasta llegar a `','` que es do, una octava por arriba. Estas notas se pueden obtener reproduciendo el sample dado, pero alterando su pitch. Para la escala de do a do que nos interesa (diatónica, teclas blancas) los pitchs se obtienen como $2^{i/12}$ con $i \in \{0, 2, 4, 5, 7, 9, 11, 12\}\}$.

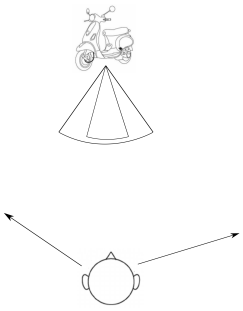
   El programa debe implementar un bucle que detecte la pulsación de teclado y haga sonar la nota correspondiente en un nuevo canal de FMOD, de modo que tengamos polifonía para hacer acordes (más de un canal sonando a la vez).

   A continuación extender el programa para incluir

   - Las notas restantes de la escala cromática (teclas negras) con las teclas "sdghj",
   - Una segunda octava con las teclas "qwe...",
   - La posibilidad de transportar la octava hacia arriba o hacia abajo.

6. En este ejercicio vamos experimentar el efecto combinado de las posiciones y orientaciones de los objetos de FMOD (*listener* y *emisores*). Para ello utilizaremos un sonido de motor (*scooter.wav*) y un listener que podremos mover por la escena. Simularemos un renderizado simple en pantalla para modificar los distintos parámetros:

   - *listener*: posición y orientación (vector *up-at*)
   - *emisor* (canal): posición, orientación y apertura de los conos interior y exterior, ganancia fuera de los conos



El renderizado puede tener el siguiente aspecto:

```
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . S . . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . L . . . . . .
. . . . . .
. . . . . .

Cone Out 10.00   In 5.00   Outer gain 0.00     Music vol: 0.22
```

A continuación incluir también los parámetros *min-max distance* para el *emisor* y comprobar su funcionamiento. Añadir una pista de música estéreo y reproducirla (en la posición del listener). Incorporar un control de volumen para esta pista.