

Propuesta de Trabajo de Fin de Grado

Desarrollo de un API musical avanzado para videojuegos
integrable en Game Maker Studio 2



Universidad
Rey Juan Carlos

**Doble grado en diseño y desarrollo de videojuegos +
Ingeniería de computadores**

Antón Rodríguez Seselle

Introducción y motivación

El audio es fundamental en el ámbito de los videojuegos, no solo en cuanto a ambientación, sino también en la narrativa y la interacción con el usuario final. Sin embargo, muchos motores de desarrollo limitan el control avanzado del audio.

Este TFG pretende elaborar el diseño y desarrollo de un API musical modular pensado para su integración directa con Game Maker Studio 2 y ofrecer facilidades y herramientas a los desarrolladores para elaborar soluciones flexibles y potentes en cuanto a audio se refiere.

Game Maker Studio 2 es un motor accesible y versátil, pero su sistema de audio es limitado comparado con otros motores profesionales. Este proyecto tiene como objetivo cubrir estas carencias con un sistema externo propio que pueda ser utilizado en el motor como plugin.

Funcionalidades del sistema

Se presentan a continuación algunas funcionalidades requeridas y ejemplos de funciones expuestas a los usuarios del API.

Reproducción básica

Inicialización de sistema de audio, reproducir, pausar, continuar, adelantar o retrasar el audio, así como hacer que se repita infinitas veces.

`audio_init()`, `audio_shutdown()`

`audio_play(path)`, `audio_pause(id)`, `audio_resume(id)`, `audio_stop(id)`

`audio_seek(id, seconds)`, `audio_set_volume(id, value)`

`audio_set_loop(id)`

Loop avanzado

Loop de secciones concretas asignadas mediante marcas temporales o funciones avanzadas de temporalización mediante compás y tempo. Útil para secciones de introducción y fases de música en bucle.

`audio_play_loop_section(path, start, end)`

`audio_play_intro_loop`

`audio_play_intro_loopBars`

Música por estados

Para realizar transiciones entre estados de una misma canción de música dinámica.

`audio_define_state(name, path)`

`audio_set_state(name)`, `audio_crossfade(from, to, time)`

Motor melódico

Poder “construir” canciones en el propio motor de juego, seleccionando sonidos como notas y construyendo secuencias musicales (tono, tiempo).

`audio_register_note(name, path)`

`audio_play_melody(sequence, bpm, time_signature)`

Voz artificial

Útil para videojuegos con diálogos estilo “Animal crossing” o “Undertale” en los que se construyen voces secuenciando fonemas.

`audio_register_voice_profile(name, array_fonemas)`

`audio_speak(name)`

Sincronización musical

Funciones que soporten la sincronización audio, imagen.

`audio_get_beat_position(id, bpm)`

Modificación de parámetros del audio

Modificar el audio desde el propio motor de juego sin necesidad de herramientas externas, como modificar la velocidad o la altura de los sonidos.

`audio_set_speed(id, speed)`

`audio_set_pitch(id, pitch)`

`audio_set_inverse(id)`

Modificación en tiempo real de audio

Aplicar efectos al audio en tiempo real, para mejorar la inmersión, como reverberación, delay y demás filtros.

`audio_apply_reverb(id, amount)`, `audio_apply_lowpass(id, cutoff)`

Agrupación de sonidos por tipo

Agrupar distintos tipos de sonidos en grupos para tratarlos de manera conjunta y poder aplicarles filtros a todos ellos o modificar sus parámetros.

`audio_tag_sound(id, tipo)`, `audio_set_group_volume(tipo, v)`, etc

Edición en tiempo real

Hacer efectos de fade, o recortes sin un editor externo.

`audio_trim(id, start, end)`, `audio_fade_in(id, dur)`, etc.

Eventos

Permitir que el sistema reaccione a cambios en el juego o en el audio

`audio_on_finish(id, callback_function)`

`audio_on_beat(id, bpm, callback_function)`

Audio dimensional

Utilizar el panning del audio para dar inmersión en el juego.

`audio_set_position(id, x, y)`

`audio_update_listener(x, y)`

Arquitectura del sistema

API escrita en C++, compilada como DLL. Uso de librería Miniaudio para gestión de audio como backend. Game maker accede a la DLL mediante `external_define()` y `external_call()` como un plugin.

<https://miniaud.io/>