



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA
DIVISIÓN DE INGENIERÍA ELÉCTRICA
INGENIERÍA EN COMPUTACIÓN
COMPUTACIÓN GRÁFICA e INTERACCIÓN HUMANO
COMPUTADORA



Audio Implementado en el Proyecto Final

NOMBRE COMPLETO:

Alvarez Badillo Rodrigo

Arriaga Vitela Carlos Eduardo

Rivas Arteaga Enrique Alan

Nº de Cuenta:

317282755

315636712

316329695

GRUPO DE LABORATORIO:

04

02

10

GRUPO DE TEORÍA: 04

SEMESTRE 2024-1

FECHA DE ENTREGA LÍMITE: 29 de noviembre de 2023

CALIFICACIÓN: _____

Implementación de Audio en el Proyecto Final.....	3
Introducción.....	3
Opciones de Librerías a Implementar.....	3
Implementación de IrrKlang.....	3
Acoplamiento al Proyecto.....	4

Implementación de Audio en el Proyecto Final

Introducción

El presente documento busca fungir como un documento de evidencia que muestre con detalle la implementación de una librería de audio especializada para el funcionamiento del código optimizado con C++ y opengl.

Opciones de Librerías a Implementar

Para este apartado se investigó tres librerías de audio para cumplir el objetivo de reproducir dos pistas de audio de forma simultánea y audio espacial. Las librerías que encontramos son OpenAL, FMOD e irrKlang, este último fue el que usamos en nuestro proyecto, pero explicaremos un poco de lo que investigamos de las otras dos librerías.

OpenAL: Es una API de audio multiplataforma que puede integrarse con OpenGL y GLFW para lograr reproducción de audio espacial.

FMOD: Ofrece potentes capacidades de audio, incluyendo la reproducción simultánea de varias pistas y efectos espaciales, y es compatible con OpenGL y GLFW.

IrrKlang: Ofrece una API sencilla de usar capaz de integrarse con OpenGL, brindando comodidades al programador.

Implementación de IrrKlang

Es una aplicación enfocada al desarrollo de código de una manera que nos permita realizar una gestión de control de cambios e implementación de código en función de las necesidades definidas por los colaboradores, esta herramienta fue fundamental para que nosotros pudiésemos demostrar nuestro avance a través de esta plataforma ya que con la herramienta del historial es posible ir viendo de una manera muy detallada todos los elementos que se fueron cambiando a lo largo del ciclo de vida de nuestro código de tal manera que también podemos recuperar elementos que pudiesen llegar a requerirse en caso de una falla imprevista.

IrrKlang es una biblioteca de audio multiplataforma de alto nivel para C++, C# y todos los lenguajes .NET. Fue diseñado para ser utilizado en juegos, simulaciones científicas, visualizaciones arquitectónicas y similares. Puede reproducir varios formatos de archivo como WAV, MP3, OGG, FLAC, MOD, XM, IT, S3M y más. Es gratuito para uso no comercial, mientras que la versión comercial se llama 'IrrKlang pro' y tiene esquemas de precios ideales para desarrolladores independientes.

Optamos por usar esta librería especializada ya que al investigar esta era la que más información se encontraba con respecto a la investigación e implementación en proyectos múltiples que utilizaron opengl. Asimismo la facilidad con la que se planteaban las variables era realmente impresionante, es por esa razón que optamos por utilizarla.

Acoplamiento al Proyecto

La configuración de esta librería fuera bastante similar a la que ya habíamos usado en el laboratorio de la asignatura, como por ejemplo para ASSIMP, simplemente requerimos los archivos que se proporcionan en el sitio web oficial de la librería y así ya poder implementarlos con ayuda de las herramientas de configuración de proyecto que nos ofrece visual Studio code. Esto fue una gran ventaja ya que nos ayudó a configurar de mejor manera el proyecto.

Para la configuración del audio, se usaron las siguientes variables:

```
if (!soundEngine) {  
    // Manejo de error si no se pudo crear el motor de sonido  
    printf("NO SE PUDO CREAR MOTOR DE SONIDO");  
    return 1;  
}  
  
const char* FilePathIntro = "Models/Kirby_s-Pinball-Land-Title-Theme.wav";  
const char* FilePathCarga = "Models/FNAF-World-OST-Final-Boss-Music-Extended-_Perfect-Loop_.wav";  
  
const char* FilePathStarWars = "Models/ARC-TROOPER-FORDO.wav";  
  
ISound* soundIntro = soundEngine->play2D(FilePathIntro, true, false, true);  
ISound* soundCarga = soundEngine->play2D(FilePathCarga, true, false, true);  
  
vec3df sound2Position(200.0f, 40.0f, 470.5f);  
ISound* sound2 = soundEngine->play3D(FilePathCarga, sound2Position, true, false, true);  
  
float minDistance = 100.0f;  
float distance=0.0f;
```

Podemos observar que la implementación de las variables fue realmente sencilla de una manera en la que con pocas líneas de código pudimos adaptar nuestro código para que esté ahora tuviera elementos de audio integrados en función del usuario, en el video que se va a anexar se podrá observar cómo existen grandes posibilidades para el uso de esta librería.