

Memoria del Proyecto: [*PineSounds on Pizza*](#) 🎵🍕

Autores

Enrique Juan Gamboa & Manuel Prada Mínguez

Resumen del Proyecto

PineSounds on Pizza es un proyecto desarrollado en Unity que busca recrear una experiencia sonora inmersiva en una isla generada proceduralmente. Inspirado en el videojuego indie *Pineapple on Pizza*, este trabajo destaca por su enfoque en la creación de un paisaje auditivo que complementa un entorno visual tranquilo y exploratorio, el cual nunca se repetirá entre ejecución y ejecución.

El objetivo principal es construir una experiencia sensorial que haga "vivir" la isla a través del sonido, ofreciendo al jugador una inmersión total mediante efectos dinámicos que se adaptan a los distintos biomas y a las interacciones del entorno.

Diseño Sonoro

Efectos Sonoros Básicos

El núcleo del proyecto se centra en los siguientes elementos sonoros:

- **Pasos:** Sonidos dinámicos que varían según la superficie (arena, césped, rocas, etc.).
- **Ambiente:** Ruidos de fondo como viento, oleaje y sonidos ambientales generales que ayudan a establecer la atmósfera del juego.
- **Fauna:** Efectos de sonido de insectos y pequeños animales que refuerzan la sensación de un entorno vivo.

Objetivos Adicionales Conseguidos

- **Zonas de Audio Dinámico:** El paisaje sonoro cambia según el bioma que atraviesa el jugador, creando transiciones entre ambientes (playa, pradera, montañas).
- **Ambientes Especiales:** Añadido de efectos específicos como ecos en cañones y oleaje.

Sistemas Avanzados de Audio

Integración con FMOD

FMOD se utilizó como herramienta principal para gestionar los sonidos, permitiendo el uso de:

- **Reverb:** Simulación de ecos y reverberaciones según el espacio y los obstáculos. Esto es visible en la zona montañosa, con el eco de los propios pasos del jugador y de la fauna que le rodea.
- **Oclusión:** Atenuación de sonidos bloqueados por elementos del entorno, como rocas o árboles o el propio terreno, apreciable por toda la isla en cuanto el jugador se pega a una pared con ruido detrás.
- **Audio Espacializado:** Adaptación dinámica del volumen y posición de los sonidos para simular profundidad y localización precisa. Apreciable tanto en la fauna, las zonas de sonido distantes y efectos de sonido presentes en la actual.
- **Scatterers:** Aleatorización de diferentes sonidos en un mismo evento, incluyendo sus volúmenes, posiciones y *pitch*, dando voz a la fauna y textura al sonido de las áreas.

Biomass y Efectos Dinámicos

Cada tile de la isla está asociado a un bioma, y esto determina:

- Los sonidos ambientales (olas en la playa, cantos de pájaros en el bosque).
- Los efectos de pisadas según el terreno (arena, hierba, roca).
- Los objetos y criaturas que se generan y que producen sus propios sonidos.

Descripción de los Scripts del Proyecto

1. ReverbOcclusionCalc

Este script es responsable de calcular y ajustar la oclusión y reverberación del sonido basado en los obstáculos entre el jugador y un emisor de audio. Su funcionalidad permite simular cómo el sonido interactúa con el entorno para crear una experiencia inmersiva.

Principales funcionalidades:

- **Detección de oclusión:** Usa un Raycast para verificar si hay obstáculos (según capas configurables) entre el jugador y el emisor de audio.
- **Parámetros dinámicos:** Ajusta parámetros de FMOD (Occlusion y Reverb)
- **Reverberación según distancia:** Si no hay oclusión, aplica un nivel básico de reverberación que aumenta la inmersión auditiva.
- **Configuración personalizada:** Incluye variables para ajustar intensidad de oclusión, reverberación y distancia máxima de detección.

Este script se asocia con el componente FMODStudioEventEmitter que controla los eventos de audio y sus parámetros dinámicos.

2. AreaDetector

El AreaDetector se utiliza para identificar en qué bioma o área del mapa está el jugador y aplicar efectos de sonido específicos. También permite detectar áreas cercanas y ajustar sus sonidos en base a la posición y dirección del jugador.

Principales funcionalidades:

- **Identificación del bioma actual:** Asocia el Tile bajo el jugador con un tipo de bioma y ajusta parámetros de sonido específicos usando FMOD.
- **Detección de áreas cercanas:** Usa un OverlapSphere para identificar biomas cercanos y calcula el área más cercana de cada tipo.
- **Sonidos direccionales y por distancia:** Establece parámetros como dirección y distancia relativa de las áreas cercanas para un posicionamiento preciso del sonido.
- **Simulación de oclusión:** Usa rayos para detectar si los biomas cercanos están ocluidos, ajustando los niveles de oclusión y reverberación.
- **Compatibilidad con pasos:** Integra un sistema de reverberación que afecta los sonidos de pasos si el jugador está cerca de áreas ocluidas.

Este script trabaja en conjunto con StepDetector para unificar la interacción entre las áreas detectadas y el sistema de pasos. Puede trabajar sin él, pero entre los dos se complementan para dar un entorno más immersivo.

3. StepDetector

StepDetector gestiona los sonidos de pasos del jugador dependiendo del terreno sobre el que camina y de si se encuentra en áreas reverberantes.

Principales funcionalidades:

- **Detección del terreno:** Usa un Raycast hacia abajo para identificar el Tile actual bajo el jugador y ajusta un parámetro de FMOD (Surface) en función de la etiqueta del terreno detectado (hierba, arena, agua, etc.).
- **Sonido dinámico de pasos:** Activa o detiene el sonido de pasos dependiendo del estado del jugador (movimiento y contacto con el suelo).
- **Reverberación:** Añade o elimina reverberación de los pasos en función de si el jugador está en un área reverberante. AreaDetector es capaz de modificar este parámetro dependiendo del área.
- **Gestión de movimiento:** Usa la velocidad del Rigidbody del jugador para determinar si está en movimiento y activa/desactiva los sonidos de pasos dinámicamente.

El script asegura una experiencia sonora realista y adaptativa al entorno, haciendo que el sonido de los pasos refleje tanto el tipo de terreno como las condiciones ambientales.

Herramientas y Tecnologías Utilizadas

Motor Gráfico y Programación

- **Unity:** Motor principal del proyecto, utilizado para la creación del entorno procedural y la integración de los sistemas de audio.
- **Lenguaje C#:** Para la lógica de generación procedural de tiles, la gestión de biomas y la implementación de los sistemas de sonido.

Audio

- **FMOD:** Herramienta para la creación e integración de efectos de sonido avanzados.
- **Audacity:** Para la edición y creación de sonidos personalizados.
- **Assets:** Todos los assets de audios usados en el proyecto vienen de youtube.
 - **Fauna:**
 - Sonidos de mono
 - Sonidos de pájaro
 - Sonidos de serpiente
 - Sonidos de ratón
 - **Pisadas:**
 - Agua
 - Arena
 - Césped
 - Roca
 - **Ambiente**
 - Olas en la orilla
 - Cigarras
 - Viento sobre hojas
 - Viento en la montaña
 - Escombros
 - Aguila

Modelos

- Animales: <https://assetstore.unity.com/packages/3d/characters/animals/quirky-series-free-animals-pack-178235>
- Objetos: <https://assetstore.unity.com/packages/3d/environments/low-poly-tropical-island-lite-242437>

Otros

- Herramienta de generación de terreno y objetos: <https://venova-games.itch.io/terrain-creator-asset>

Conclusiones

PineSounds on Pizza logra combinar una estética visual serena con un paisaje sonoro profundo y dinámico, ofreciendo al jugador una experiencia inmersiva única. A través de herramientas avanzadas como FMOD y Unity, el proyecto explora el potencial del diseño sonoro en juegos, destacando la importancia del audio como elemento narrativo y atmosférico.

Este trabajo no solo rinde homenaje a *Pineapple on Pizza*, sino que establece una base sólida para proyectos futuros que busquen potenciar la inmersión mediante la combinación de diseño procedural y audio interactivo.