

INTRODUCCIÓN A LA PROGRAMACIÓN DE VIDEOJUEGOS (IPV)

Programación de Audio

Fundamentos

Hw y Sw

Formatos

Reproducción

- Música

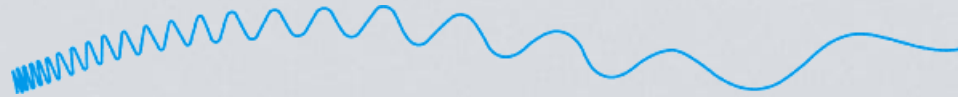
 - Operaciones básicas

- Espacialización

- Otros

APIs

Sonido Analógico. Terminología



Amplitud

Presión/intensidad de la onda al desplazarse en el tiempo

Frecuencia

Nº de veces que una onda se repite por unidad de tiempo

Se mide en Hertzios (Hz)

Pitch

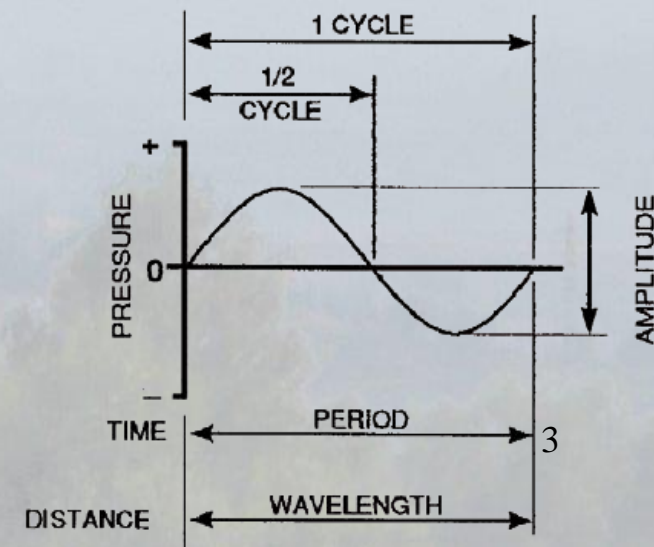
Percepción de la frecuencia en formato musical. Tono.

Afinamiento o tuning

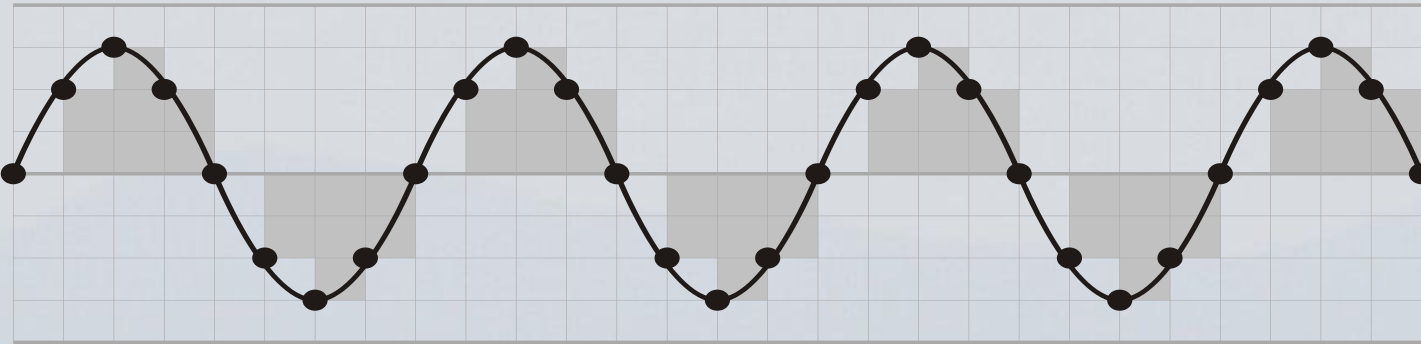
Distribución musical de las frecuencias sobre las notas

Decibelio

Unidad de medida de la amplitud del sonido



Representación digital del Sonido (I)



Muestreo o *sampling*

Medición de la amplitud de una onda en intervalos discretos

Frecuencia de muestreo: Cantidad de veces que se toma una muestra por unidad de tiempo

Resolución de la muestra de 4 a 24 bits. Profundidad de bits

Ej:

COMPACT
disc
DIGITAL AUDIO

HDCD

Música en un CD. Frec. Muestreo 44.1 kHz y 16 bits de profundidad de bit

HD CD. Frec. Muestreo 88.2 kHz y 20 bits de profundidad de bit virtuales sobre 16 reales

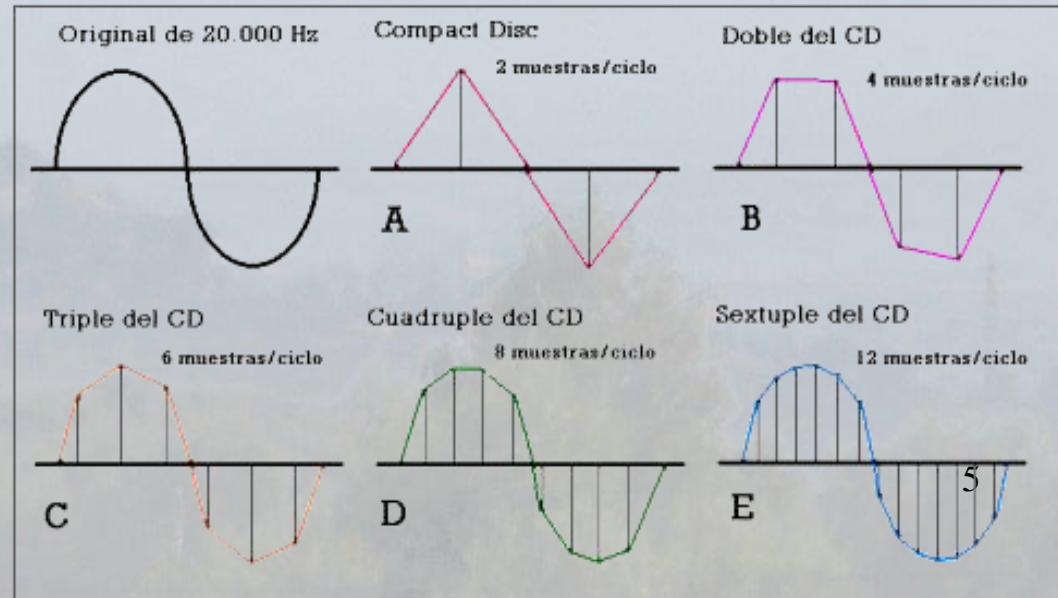
Representación digital del Sonido (II)



Frecuencia de muestreo afecta al rango y calidad de la reproducción de las frecuencias altas

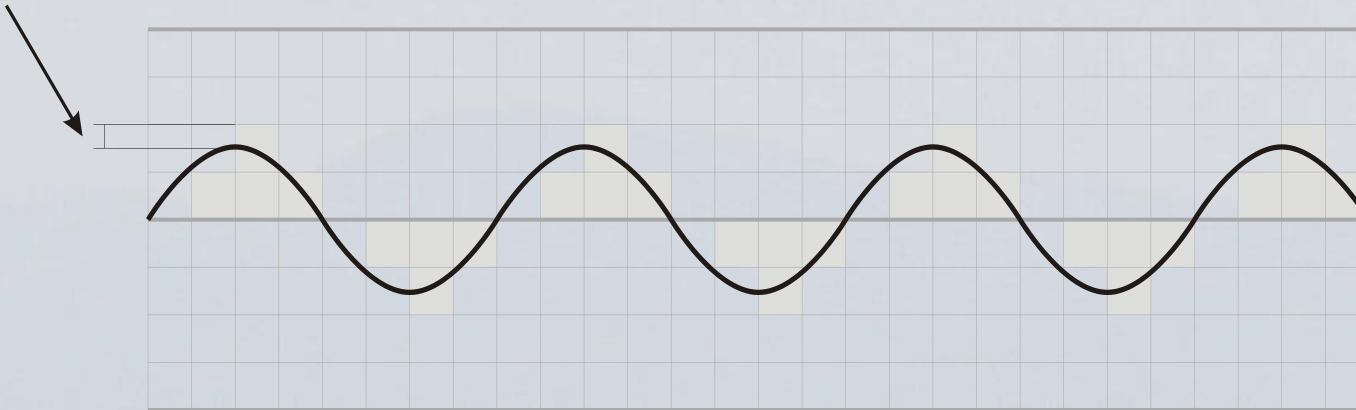
Teorema de Nyquist

Frecuencia máxima reproducible = mitad frecuencia de muestreo



Error de cuantización (I)

Quantization Error



Relación (R) entre dos magnitudes A y B se mide en Belios

$$R = \log_{10}(A/B) \text{ belios}$$

Ej: Si $A = 10B$, $R = 1$ // Si $A = 100B$, $R = 2$ // Si $A = 10^n B$, $R = n$

Relación señal/ruido (SNR) depende de la cantidad de bits disponible (Profundidad de muestreo). Se mide en decibelios. Evita tener que emplear decimales.

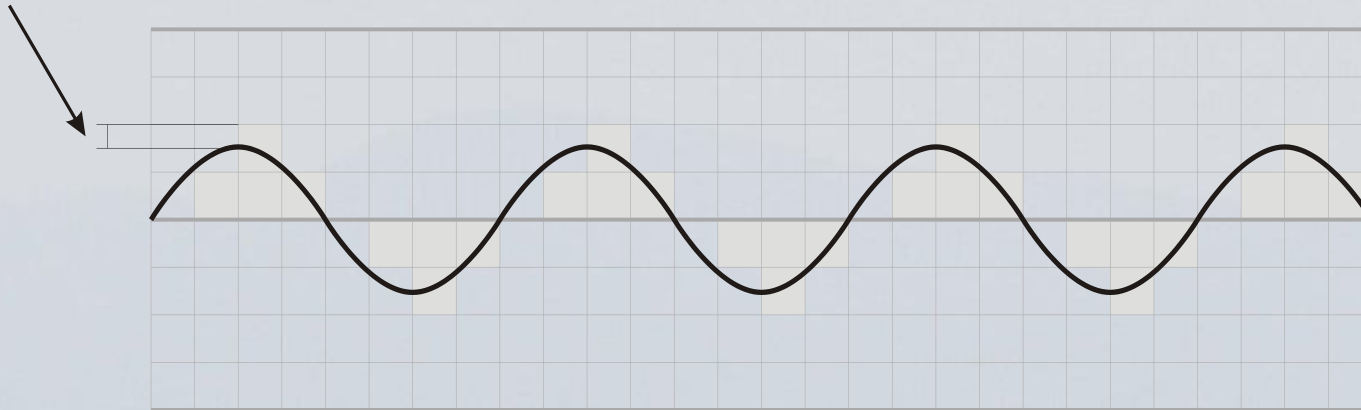
$$R = 10\log_{10}(S/N)$$

S = Max amplitud de la señal

N = El valor mínimo que se puede representar

Error de cuantización (II)

Quantization Error



El error de cuantización se entiende como el valor mínimo de cuantización (1)

En audio de n bits,

$$S = 2^n \text{ y } N = 1$$

$$\text{SNR} = 2R = 2 \cdot 10 \log_{10}(S/N) = 20 \log_{10}(2^n/1) = 20 \log_{10}(2^n) = n \cdot 20 \log_{10}(2) = 6,02 \cdot n \approx 6n$$

Sonido 8 bits tiene una relación 256:1 SNR (~48 dB)

Sonido 16 bits tiene una relación 65,536:1 SNR (~96 dB)

Sonido 24 bits tiene una relación 16E6:1 SNR (~144 dB)

Hardware



Canales o tuberías
≈ tuberías gráficas

Muestras se procesan por caminos independientes

Pueden ser Hw o Sw. Depende de API y potencia del Sw

Procesamiento de volumen, pan, pitch, ecos, reverb,...

Espacialización del sonido y empleo de algoritmos HRTF (Head-Related Transfer Function)

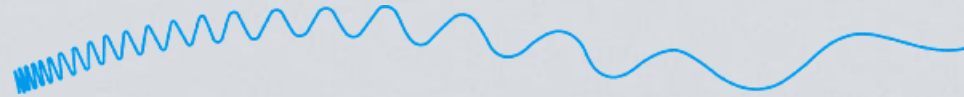
Mezcla final de todos los canales en buffers de salida (render)

Los canales de sonido

- No son las conexiones externas de audio
- Gestionan las diferentes fuentes de sonido a mezclar
- Son independientes del proceso de espacialización



APIs



DirectSound (parte del API DirectX)
Sólo sobre Windows



OpenAL

Sigue filosofía OpenGL. Creative Labs (Sound Blaster)

Disponible sobre múltiples plataformas PC: Mac, linux,
Windows



Propietarias

Típicamente disponibles sobre consolas

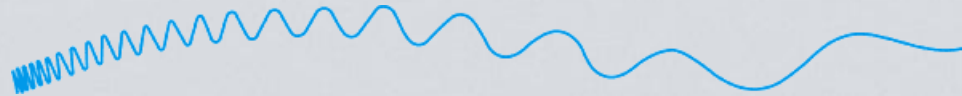


Licenciables

Middleware múltiples plataformas: consolas, PCs,...



Formatos de audio (I)



Codifican sonido de diversas formas

Formatos sin pérdidas (lossless)

No comprimidos

Comprimidos

Formatos con pérdidas (lossy)

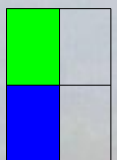
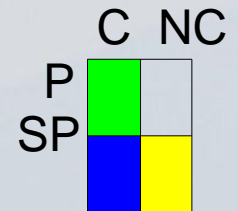
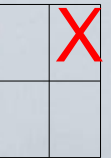
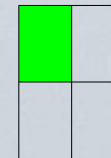
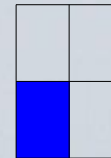
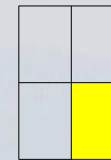
Todos son comprimidos

No existen formatos con pérdidas no comprimidos

Formatos híbridos

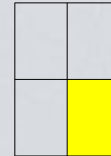
El mismo formato puede admitir pérdidas o no

En ambos casos suelen ser comprimidos



Formatos de audio (II)

Sin pérdidas. No comprimidos



Razón de compresión 1:1

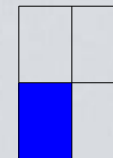
Descodificación genera contenido **idéntico** al original

Ext	Nombre	Open Source	Canales	Frec. Muestreo	Res. bits	Comentarios
.ai{f/ff }	AIFC	Sí				AIFF con y sin compresión. MAC
.au, .snd	Audio / Sound	Sí	1 a 2	Todas	Todos	Sun, unix y Java. PCM / μ -law / a-law
.raw / .wav	RAW	Si		Todas	Todos	Datos en bruto

AIFF = Audio Interchange File Format

Formatos de audio (III)

Sin pérdidas. Comprimidos



Razón de compresión 2:1

Descodificación genera contenido idéntico al original

Ext	Nombre	Open Source	Canales	Frec. Muestreo	Res. bits	Comentarios
.alac	ALAC	No	>2	44,1-48 KHz	16-24	Apple
.ape	Monkey's audio	Sí	2	8-48 KHz	16	www.monkeysaudio.com
.flac	FLAC	Sí	8	1-1048 KHz	4 a 32	xiph.org/flac/ BSD / GNU

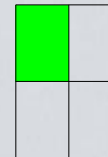
Suelen emplear RLE y predicción lineal con residuos

FLAC. Free Lossless Audio Codec

ALAC. Apple Lossless Audio Codec

Formatos de audio (IV)

Comprimidos con pérdidas



Razón de compresión variable y muy alta

Descodificación genera contenido diferente al original

Tienen en cuenta la psicoacústica

Ext	Nombre	Libre / gratis	Canales	Frec. Muestreo	Res. bits	Comentarios
.aac	Advanced Audio Coding	No	48	8-192 KHz	Todas	mp2 y mp4
.ac3	AC3	Sí	6	32-48 KHz		Lab Dolby
.mpc	Musepack	Sí	2	32-48 KHz	8 a32	MPEGplus, MP+ o MPEG+. LGPL
.mp3	MPEG{2/3}	Sí	2	8a48 Khz	Todas	Fraunhofer Society
.ogg	Vorbis	Sí	255	1-200 KHz	Todas	

Formatos de audio (V)

Comprimidos con y sin pérdidas

Soportan tanto algoritmos de codificación con pérdidas como sin ellas



Ext	Nombre	Open Source	Canales	Frec. Muestreo	Res. bits	Comentarios
.ra {m}	Real Audio	No	6			
.wma	Windows Media Audio	No	8	8-96 KHz	Todas	
.wav	ATRAC	No				ATRAC1/3/3plus. ATRAC advanced lossless. Sony
.wav	Wave	Sí				µSoft. Soporta gsm
.wv	WavPack	Sí	16	1 Hz - 16.77 MHz	Vble	Wav. Coma flotante

Formatos de audio (VI)



Compresión sonido

Permiten almacenar sonido y música de forma más compacta

Codecs basados en la Reducción de bits

Generalmente suelen ser ligeros

Compresión ADPCM suele estar implementado en la mayoría del hardware de las consolas actuales

Psico-acústica suelen tener mejor compresión

Potencia de cálculo realizar codif >> descodif

Coste codif/descodif asimétrico

Formatos de audio (VII)



MP3

Patente finalizada el 23 de abril de 2017

Ogg Vorbis

Similar a MP3

Open source y sin patente (royalty-free)



Cuidado

Tener en cuenta también las patentes y licencias del sw
que maneja estos ficheros

Reproducción de música (I)



Dos tecnologías

MIDI - Musical Instrument Digital Interface

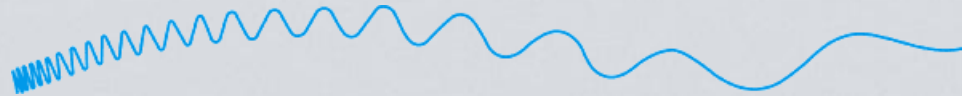
Audio Digital por streaming

Reproducción. CD audio

Descompresión. MP3, etc



Reproducción de música (II)



MIDI

Tamaño ínfimo del fichero

Fácil de controlar, alterar y generar en t⁰ real

Calidad de la música depende de calidad del reproductor,
bancos de sonido, SNR,...

A emplear cuando música sólo emplea un cjto de
instrumentos predefinidos

Almacena la “partitura” de la música, no el sonido

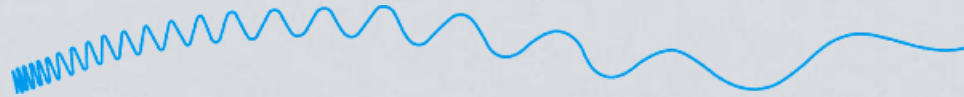


Otros formatos alternativos

DLS. Down Loadable Sound. Empleados por reprod. MIDI

iXMF. Interactive eXtensible Music Format

Reproducción de música (III)



Reproducción streaming. Solución compromiso

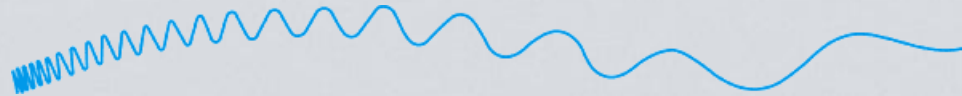
Dividir música en pequeños trozos (segmentos) de 2 a 8 compases

Codificar música a base de temas o secuencias de segmentos: bucles, ramas,...

Reproducir música = reproducir un tema

El mapa de transiciones secuencia el orden de los segmentos a reproducir

Reproducción de audio (I)



2 métodos de reproducción

Desde memoria

Streaming en t^0 real

Consume menos memoria si ficheros de audio son grandes: música fondo, diálogos, etc

Empleo de buffer circular con punteros de lectoescritura o algoritmos de doble buffering

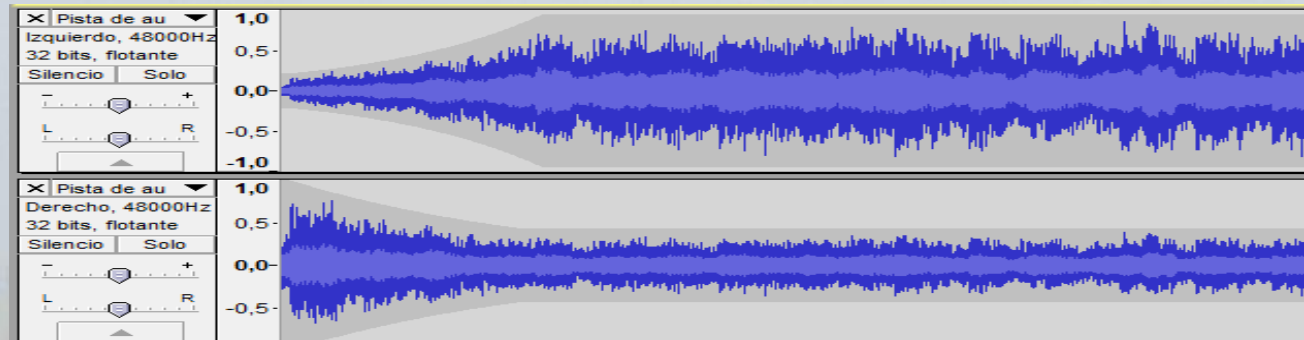
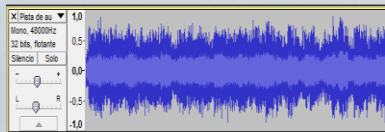
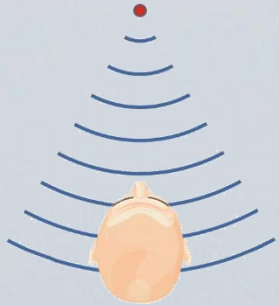
Reproducción de audio (II)



Operaciones básicas de postproducción (I)

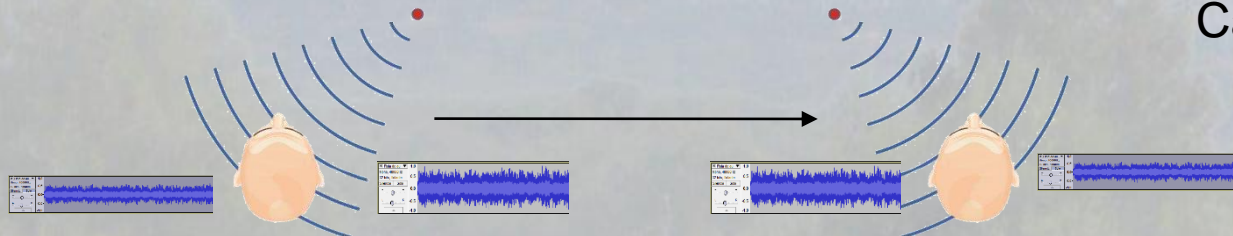
Panning. Atenuación asimétrica de canales estéreo para simular espacialización de sonido

No hay desfase de onda. Sólo se juega con la amplificación

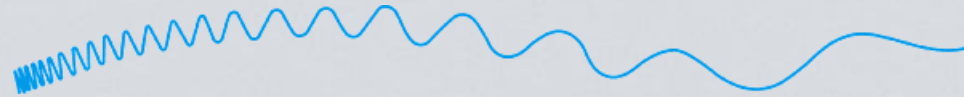


Canal izquierdo

Canal derecho

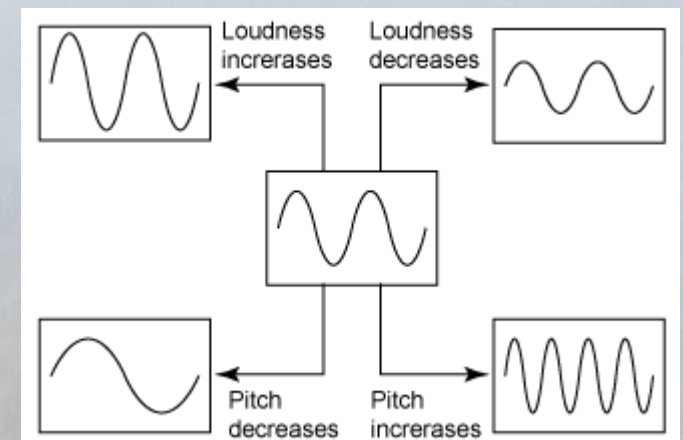
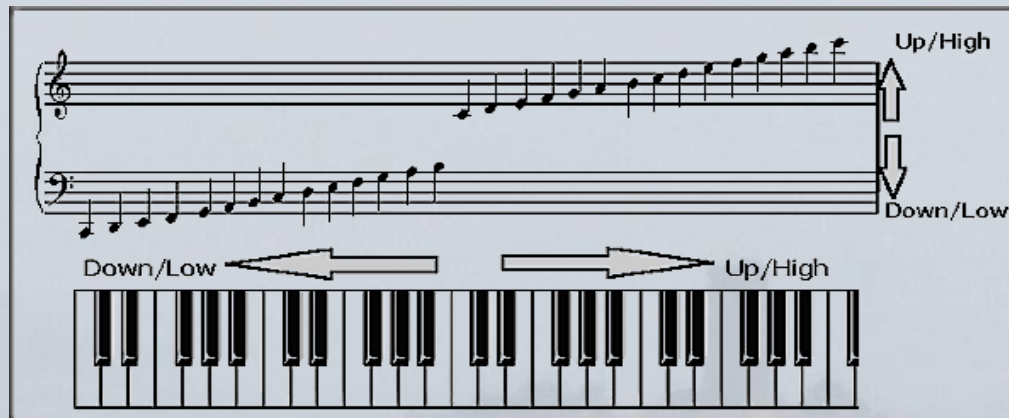


Reproducción de audio (III)



Operaciones básicas de postproducción (II)

Pitch. Ajuste de la frecuencia de reproducción. Hace el mismo sonido más agudo o más grave

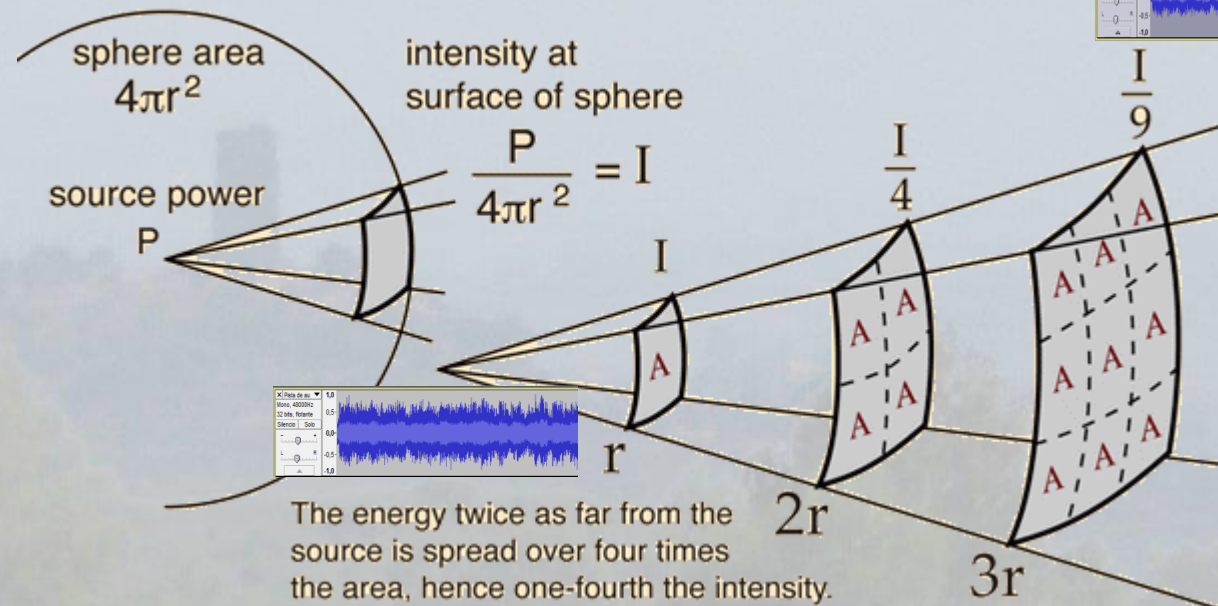
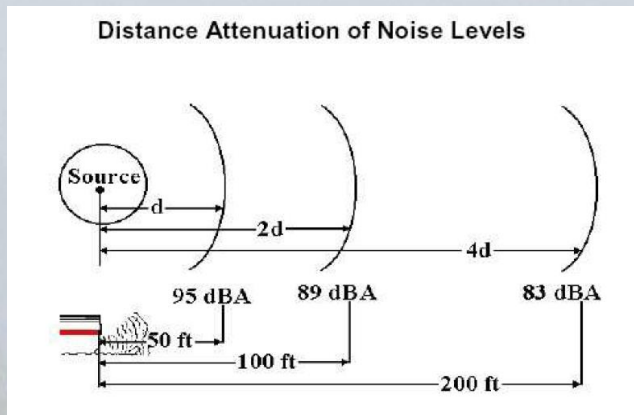


Reproducción de audio (IV)



Operaciones básicas de postproducción (III)

Volumen. Normalmente se presenta como una atenuación de la intensidad del sonido



Espacialización del sonido (I)



Se requiere al menos
Oyente

Posición y orientación

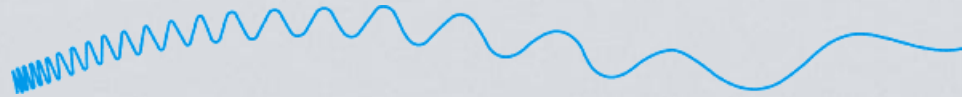
“Micrófonos estéreo” en el mundo virtual

Fuente

Posición, orientación, velocidad, etc de la emisión del
sonido en el mundo virtual

Debido a la potencia de cálculo necesaria, los efectos
ambientales casi siempre se implementan en hardware

Espacialización del sonido (II)

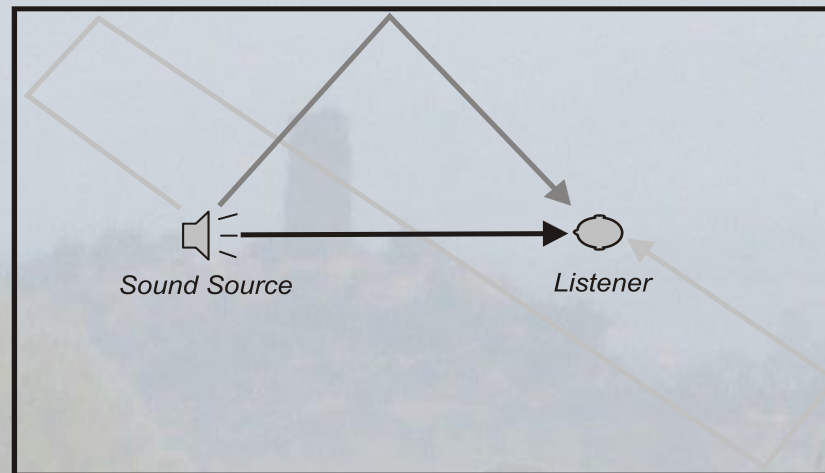


Transmisión del sonido se manifiesta

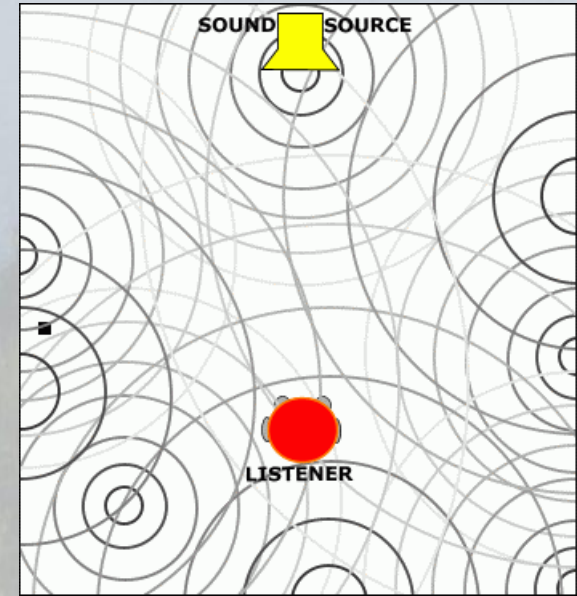
Transmisión directa

Reflexiones tempranas (eco)

Reflexiones tardías (reverberación)



Direct Path →
Echo →
Reverberation



Espacialización del sonido (III)



Efectos ambientales dependen de la forma de la habitación y composición material de las paredes

Cálculo complejo. Necesario APIs que calculen efectos automáticamente en t^0 real (fmod, OAL,...)

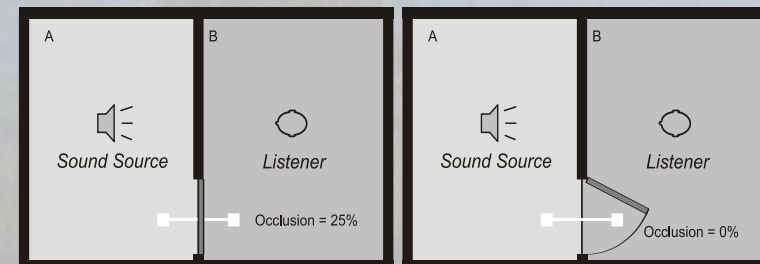
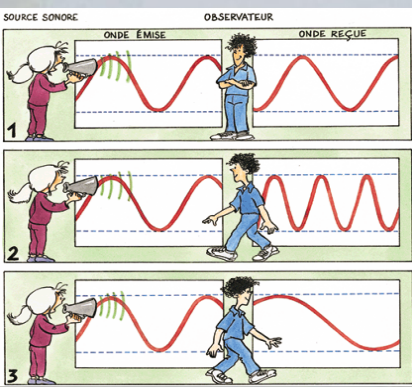
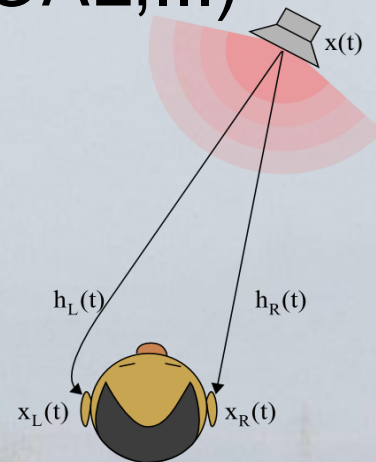
Doppler

HRTF. Head-Related Transfer Function

Atenuación por

Distancia

Oclusiones dinámicas



Programación avanzada de audio (I)

Fundamentos
Hw y Sw
Formatos
Reproducción
Música
Operaciones básicas
Especialización
Otros
APIs



Reproducción de voz

Sincronización de labios – Lip Synch

Control por la voz – Voice Rec



Programación avanzada de audio (II)



Reproducción de voz

Por grabación directa. Ficheros de audio

Empleo de técnicas TTS. Síntesis en tº real

Empleo de énfasis según estado del juego

Los primeros videojuegos no usaban ningún sonido de voz, debido a limitaciones técnicas

En 1970's y 1980's, se usaban sonidos electrónicos simples

En 1990's y 2000's, aparece la sincronización de labios y la actuación de voz



Programación avanzada de audio (III)



Sincronización de labios

Mezcla tecnología de audio y visual

Crear diálogo realista en vactors

Técnicas empleadas

Sincronización TTS y gesto vocal

Amplitud de onda mueve labios

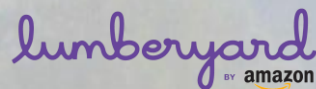
Medal of Honor y Half-Life 2 utilizan simulan en tiempo real los movimientos de la boca para sincronizarlos con los sonidos como si los dijera una persona en vivo

Personajes increíblemente realistas.



Half-Life 2

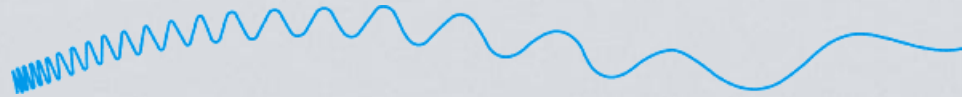
APIs



SPEECH FX™

Reconocimiento de voz
Lip Synch

Programación avanzada de audio (IV)



Reconocimiento de voz

Comandos vocales

Generan nuevo canal comunicación con VJ

Facilitan interacción en determinadas situaciones de
juego: FPS

Uso minoritario de técnicas de reconocimiento en VJ

Programación avanzada de audio (V)



Videojuegos con reconocimiento de voz



Bot Colony. Desarrollado por North Side Inc. 2013
Reconocimiento voz continua, no sólo comandos sueltos.
Ventas insuficientes.

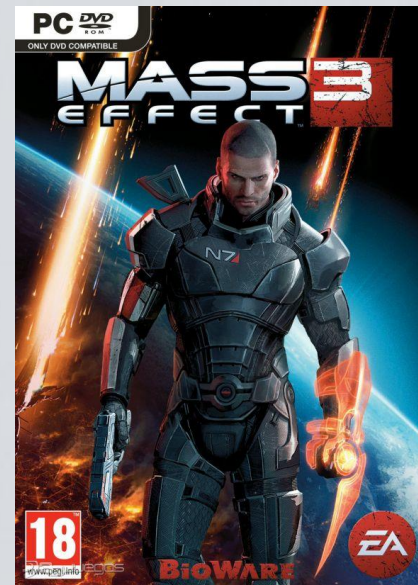


Nevermind. Realizado por Flying Mollusk, es un juego de aventuras y terror de pacientes con traumas psicológicos. Utiliza Intel *Realsense*

Seaman. Mascota virtual para Sega Dreamcast que interacciona por voz y pasa por etapas evolutivas



LifeLine. Realizado por Konami para PlayStation 2. Usa hasta 500 comandos verbales sencillos como "date ³¹prisa", "alto", "esquivar" y "girar a la izquierda"



Mass Effect 3. Videojuego RPG de acción multiplataformas desarrollado por BioWare y publicado por Electronic Arts. Última entrega de la trilogía. Kinect con comandos de voz para cambiar de habilidad, arma o abrir puertas.



APIs (I)

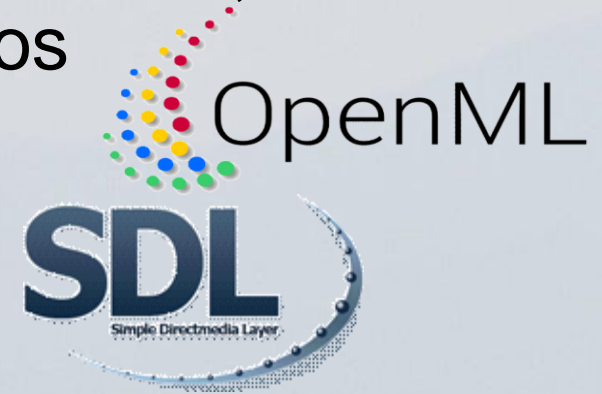
Al igual que los paquetes gráficos 2D-3D, existen paquetes para gestión de sonidos

OpenML

SDL

OpenAL

fmod



Paquetes de desarrollo de videojuegos con capacidades de audio

Allegro

APIs (II)

OpenML (Open Media Library)

Código abierto

Multi-plataforma

Captura, transporte, procesamiento, visualización y
sincronización de diferentes medios

Gráficos 2D/3D

Flujos de audio/video

Sincronización de flujo por muestra

Soporta

OpenGL para procesamiento de video acelerado

Control de visualización profesional

Flujos asíncronos entre aplicaciones hardware



APIs (III)

SDL

API completa de control de sonido de CD

Reproducción de sonido

- 8 bits y 16 bits

- Mono o estéreo

Se ejecuta en hilo separado por call-back

Usa mezcladores software

Prestaciones algo desfasadas

Muy utilizado todavía en 2D e indie

Simulación aproximada de espacialización de audio

GNU, multiplataforma

También soporta I/O, gestión de ventanas, hilos, temporizadores,...



APIs (IV)

OpenAL

OpenAL ("Open Audio Library")

Interfaz software del hardware de audio

Soporta salidas multicanal espacializado simulando espacio 3D

Conceptos 2D como el panning o canales derecho o izquierdo no son directamente soportados

Multiplataforma

Fácil de usar

Funcionamiento análogo al API OpenGL en estilo de codificación y convenciones

Incluye extensiones compatibles con IA-SIG 3D Level 1 y 2

- Direccionalidad de la fuente de sonido

- Atenuación en la distancia

- Efecto Doppler

- Efectos de entorno

 - Reflexión

 - Obstrucción

 - Transmisión

 - Reverberación



APIs (V)

fmod

Desarrollado por Fairlight Tehnologies

DSP interno con múltiples efectos: osciladores senoidales, onda cuadrada, rampa, triángulo y ruido, filtros de resonancia pasa baja y pasa alta, eco, flange, distorsión, normalización, ecualización paramétrica, desplazamiento de pitch en tiempo real, coros, reverberación,...

Sonido 3D completo

Atenuación lineal/no lineal/personalizable

Múltiples oyentes

Oclusión y obstrucción usando polígonos reales

Estéreo o multicanal 5.1, 7.1

Miles de fuentes virtuales de sonido a la vez sobre hardware limitado

Empleado en los principales motores de desarrollo de videojuegos: Unity y Unreal



APIs (VI)

fmod



- 20 formatos de sonido
- Streaming
- Formatos comprimidos: ADPCM, MPEG, XMA
- API OO en C, C++, C#, Delphi y Visual Basic
- Plug-ins: FMOD y VST
- SIMD (SSE, VMX, VFPU, ALTIVEC)
- Múltiples tarjetas simultáneas
- Plataformas
 - Microsoft Windows y Windows 64bit. (AMD64)
 - Linux y Linux 64bit. (AMD64)
 - Macintosh. OS8 / 9 / X and OSX for x86.
 - Sony PlayStation
 - Microsoft Xbox
 - Nintendo

APIs (VII)



Unity Soporta

Ficheros monofonicos, estereo y multicanal (o quadrafonicos de hasta ocho canales)

Ficheros modulares (generados con secuenciadores → trackers)

Dependiendo de la plataforma

- Escritorio: Uncompressed Audio u Ogg Vorbis
- Android / iOS: Uncompressed Audio o MP3 Compressed
- Consolas: PS Vita (HEVAG) XBox One (XMA)



APIs (VIII)

Allegro Midi



MIDI nativo con hasta 64 efectos simultáneos
Control dinámico, note on/off, volumen principal, pan, pitch bend, parches
generales de MIDI (SF2 / GUS)

Wave

Formatos WAV y VOC. Bucle adelante, atrás o bidireccional
Streaming audio
Cambio de volumen, pan, pitch, etc durante la reproducción

Unix	Windows	MacOS X	Dos
OSS ALSA (0.5/0.9/1.0) ESD aRts JACK SGI AL	DirectSound WaveOut MIDI	Core Audio Sound Manager Core Audio MIDI QuickTime MIDI	Adlib, SB, SB Pro, SB16, AWE32 MPU-401 ESS AudioDrive Ensoniq Soundscape Windows Sound System

Bibliografía

Cap 5.5. Introduction to Game Development.

Steve Rabin. Charles River Media

ISBN: 978-1-58450-377-4

Páginas webs del fabricante

Web Audio API. Boris Smus. ISBN: 978-1-44933-268-6

Documentación generada por
Dr. Ramón Mollá Vayá
Grupo de Informática Gráfica
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia

Reconocimiento-NoComercial-CompartirIgual 2.5

Usted es libre de:

copiar, distribuir y comunicar públicamente la obra
hacer obras derivadas bajo las condiciones siguientes:



Reconocimiento. Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador.



No comercial. No puede utilizar esta obra para fines comerciales.



Compartir bajo la misma licencia. Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.

Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor

Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.