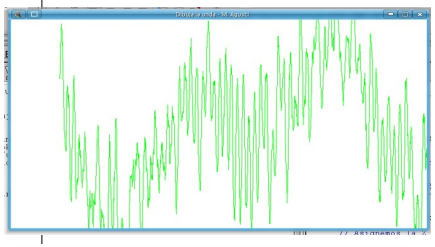
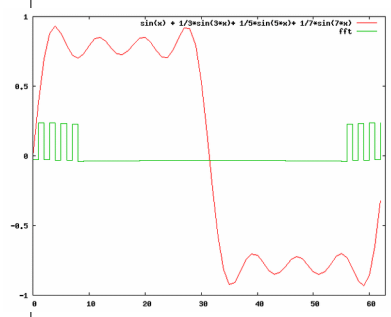
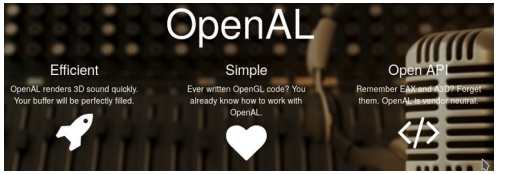
Seminario

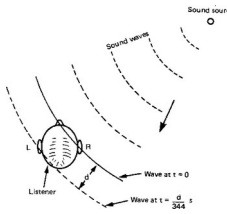
Interacción mediante el uso de audio espacial con OpenAL

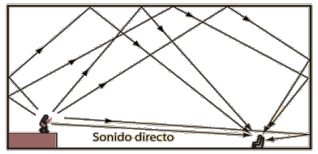
101010
101010











Sistemas Multimedia Interactivos e Inmersivos

Grado de Ingeniero en Informática. Escola Tècnica Superior de Enginyeria Informàtica. Curso 2020/2021

Manuel Agustí

Seminario de OpenAL

- Objetivos
  - Estándard
    - API, documentación. Extensiones
    - Conexión con otras librerías
  - Ejemplos de uso

Objetivos

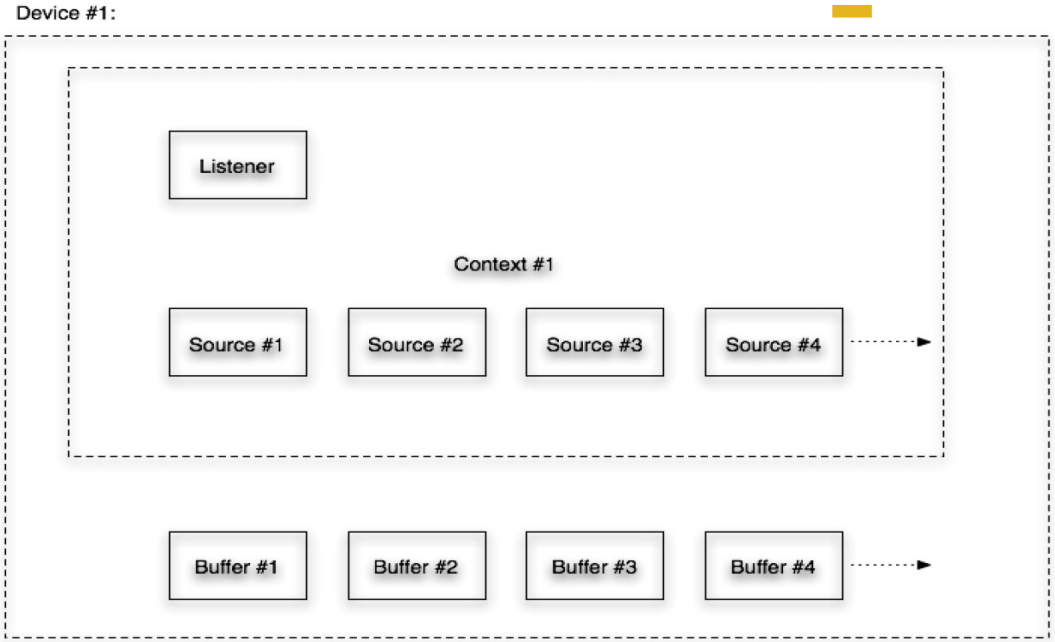
- **Exponer el contenido del estándar de audio espacial OpenAL**
- **Describir la especificación referenciando a la documentación oficial y las extensiones propuestas**
- **Mostrar el API de OpenAL con ejemplos**
- **Describir cómo se puede ampliar la funcionalidad de OpenAL con el uso de otras librerías relacionadas**

Índice

- Estándar
  - API en 3 niveles: AL, ALC y ALUT
  - Documentación y extensiones
  - Conexión con otras librerías
- Ejemplos de uso
  - Hello, World!
  - Generación de sonidos simples
  - Generación de sonidos desde fichero
  - Audio posicional
  - Información
  - Efecto Doppler
  - Extensiones: uso del micrófono
  - Streaming
  - Extensiones: efectos ambientales

Estándar OpenAL

- Estándar *Open Audio Library*
  - Motor de *rendering* de audio en 3D



  - Otros objetos:
    - Contexto (*context*)
      - Oyente + fuente/s
      - Modela la propagación del audio: define una "escena"
    - Dispositivo (*device*)
      - Abstracción: hardware y manejador
- Estándar: API

Estándar OpenAL

- Estándar *Open Audio Library*
  - API: Niveles (1)
    - *Audio Library* (AL)

Relativas a los *buffers*
•alGenBuffers
•alDeleteBuffers
•alIsBuffer
•alBufferf
•alBuffer3f
•alBufferfv
•alBufferi
•alBuffer3i
•alBufferiv
•alGetBufferf
•alGetBuffer3f
•alGetBufferfv
•alGetBufferi
•alGetBuffer3i
•alGetBufferiv

Relativas a las fuentes
•alGenSources
•alDeleteSources
•alIsSource
•alSourcef
•alSource3f
•alSourcefv
•alSourcei
•alSource3i
•alSourceiv
•alGetSourcef
•alGetSource3f
•alGetSourcefv
•alGetSourcei
•alGetSource3i
•alGetSourceiv
•alSourcePlayv
•alSourceStopv
•alSourceRewindv
•alSourcePausev
•alSourcePlay
•alSourceStop
•alSourceRewind
•alSourcePause
•alSourceQueueBuffers
•alSourceUnqueueBuffers

Relativas al oyente
•alListenerf
•alListener3f
•alListenerfv
•alListeneri
•alListener3i
•alListeneriv
•alGetListenerf
•alGetListener3f
•alGetListenerfv
•alGetListeneri
•alGetListener3i
•alGetListeneriv

Relativas a propiedades y estados
•alEnable
•alDisable
•alIsEnabled
•alGetString
•alGetBooleanv
•alGetIntegerv
•alGetFloatv
•alGetDoublev
•alGetBoolean
•alGetInteger
•alGetFloat
•alGetDouble
•alDopplerFactor
•alDopplerVelocity
•alSpeedOfSound
•AlDistanceModel

Relativas a gestión de errores
•AlGetError

Relativas a las extensiones
•alIsExtensionPresent
•alGetProcAddress
•alGetEnumValue

Estándar OpenAL

- Estándar *Open Audio Library*
  - API: Niveles (2)
    - AL Context (ALC)
      - alcCreateContext
      - alcMakeContextCurrent
      - alcProcessContext
      - alcSuspendContext
      - alcDestroyContext
      - alcGetCurrentContext
      - alcGetContextsDevice
      - alcOpenDevice
      - alcCloseDevice
      - alcGetError
      - alcIsExtensionPresent
      - alcGetProcAddress
      - alcGetEnumValue
      - alcGetString
      - alcGetIntegerv
      - alcCaptureOpenDevice
      - alcCaptureCloseDevice
      - alcCaptureStart
      - alcCaptureStop
      - alcCaptureSamples

Estándar OpenAL

- Estándar *Open Audio Library*
  - API: Niveles (y 3)
    - *The Open AL Utility Toolkit* (ALUT)

Error Handling

alutGetError
alutGetErrorStringInitialization / Exit
alutInit
alutInitWithoutContext
alutExit

Sound Sample File Loading

alutCreateBufferFromFile
alutCreateBufferFromFileImage
alutCreateBufferHelloWorld
alutCreateBufferWaveform
alutLoadMemoryFromFile
alutLoadMemoryFromFileImage
alutLoadMemoryHelloWorld
alutLoadMemoryWaveform
alutGetMIMETypes

Version Checking

alutGetMajorVersion
alutGetMinorVersion

Sleeping

alutSleep

Estándar OpenAL

- OpenAL: Estándar vs Implementación
  - V1.1: Creative Labs (~ 2005) vs OpenAL.org (~ 2018)
  - Implementación Hardware: *Creative Labs*.
  - Implementación Software: OpenAL-Soft
    - Web <https://openal-soft.org/>, GitHub <https://github.com/kcat/openal-soft>
    - *This library is a compatible update/replacement to the deprecated OpenAL Sample Implementation (the SI). It is a fork of the old Windows software driver, modified to be cross-platform with multiple output backends: PulseAudio, ALSA, OSS, MMDevAPI, DirectSound, CoreAudio, Solaris, QSA, SoundIO, OpenSL, WinMM, PortAudio, "Null" Output, and a .wav writer are currently implemented.*
    - *OpenAL Soft has been further improved to support mono, stereo, 4-channel, 5.1, 6.1, 7.1, HRTF, and B-Format output. It does not support the Vorbis and MP3 extensions however, these are considered deprecated. It does, though, support many extensions like AL_EXT_FLOAT32 and AL_EXT_MCFORMATS for multi-channel and floating-point formats, as well as ALC_EXT_EFX for environmental audio effects, and others.*
  - Documentación y extensiones
  - Otras librerías

Imágenes obtenidas de http://lokigames.com/ y http://creative.com

Estándar OpenAL

- OpenAL: Estándar vs Implementación
  - Estándar vs Implementación
  - Documentación y extensiones
    - OpenAL 1.1 specification (PDF) y OpenAL Programmers Guide (PDF) https://www.openal.org/documentation/
    - Extensiones
      - *Multi-Platform Extensions*
      - *Linux/Standard Implementation Extensions*
      - *MacOS Extensions*
      - *Windows Extensions*
    - *OpenAL Extension API Database OpenAL Extension Registry* https://icculus.org/alextreg/index.php
  - Otras librerías

Extensions:

- 1.0_specification
- ALC_ENUMERATE_ALL_EXT
- ALC_ENUMERATION_EXT
- ALC_EXT_ASA_DISTORTION
- ALC_EXT_ASA_ROGER_BEEP
- ALC_EXT_BRS_GAME_LICENSE_REQUIRED
- ALC_EXT_capture
- ALC_EXT_DEDICATED
- ALC_EXT_DEFAULT_FILTER_ORDER
- ALC_EXT_disconnect
- ALC_EXT_EFX
- AL_EXT_ALAW
- AL_EXT_BFORMAT
- AL_EXT_double
- AL_EXT_float32
- AL_EXT_FOLDBACK
- AL_EXT_IMA4
- AL_EXT_MCFORMATS
- AL_EXT_mp3
- AL_EXT_MULAW
- AL_EXT_MULAW_BFORMAT
- AL_EXT_MULAW_MCFORMATS
- AL_EXT_SOURCE_RADIUS
- AL_EXT_STEREO_ANGLES
- AL_EXT_vorbis
- EAX-RAM
- EAX2.0
- EAX3.0
- EAX4.0
- EAX5.0

Estándar OpenAL

- OpenAL: Estándar vs Implementación
  - Estándar vs Implementación
  - Documentación y extensiones,
  - Otras librerías:
    - Gestión de formatos de ficheros
      - libsndfile, libsoundio
      - libvorbis, libopus,
      - libflac
      - libmpg123
    - Gestión de hardware y formatos
      - ALUT -→ ¿ALURE, SDL, ...?

Estándar OpenAL

- OpenAL: Estándar vs Implementación
  - Estándar vs Implement., Doc y extensiones,
  - Otras librerías
    - Gestión de hardware y formatos
      - ***The OpenAL Utility Toolkit*** **(ALUT)**
        - http://distro.ibiblio.org/rootlinux/rootlinux-ports/more/freealut/freealut-1.1.0/doc/alut.html
      - *AL Utilities Retooled (ALURE)*
        - <*https://github.com/kcat/alure*>
      - *Simple DirectMedia Layer* (SDL)

        Simple DirectMedia Layer is a cross-platform development library designed to provide low level access to audio, keyboard, mouse, joystick, and graphics hardware via OpenGL and Direct3D. It is used by video playback software, emulators, and popular games including Valve's award winning catalog and many Humble Bundle games.

        SDL officially supports Windows, Mac OS X, Linux, iOS, and Android. Support for other platforms may be found in the source code.

        SDL is written in C, works natively with C++, and there are bindings available for several other languages, including C# and Python.

      - Audio Device Management, Playing and Recording (SDL_audio.h)

Estándar OpenAL

- OpenAL: Estándar vs Implementación
  - *The OpenAL Utility Toolkit* (ALUT)
  - Implementación
    - Freealut
      <https://github.com/vancegroup/freealut>

The OpenAL Utility Toolkit (ALUT)

Contents

- http://distro.ibiblio.org/rootlinux/rootlinux-ports/more/freealut/freealut-1.1.0/doc/alut.html

Estándar OpenAL

- ¿Cómo es OpenAL sin ALUT?
  - Sin "alutInit", ni "alutExit", ...

```
#include <AL/al.h>
#include <AL/alc.h>


int main(  ... ) {
  ALCdevice *device;
  ALCcontext *context;

  device = alcOpenDevice(NULL);
  if (!device)
    // handle errors

  ALboolean enumeration;
   enumeration = alcIsExtensionPresent(NULL,
                          "ALC_ENUMERATION_EXT");
   if (enumeration == AL_FALSE)
       // enumeration not supported
   else
      // enumeration supported
```

```
   error = alGetError();
   if (error != AL_NO_ERROR)
      // something wrong happened

context = alcCreateContext(device, NULL);
if (!alcMakeContextCurrent(context))
// failed to make context current
// test for errors here using alGetError();
}

//
// Bucle principal
//


// cleanup context
   // alDeleteSources(1, &source);
   // alDeleteBuffers(1, &buffer);

 device = alcGetContextsDevice(context);
 alcMakeContextCurrent(NULL);
 alcDestroyContext(context);
 alcCloseDevice(device);
}
```

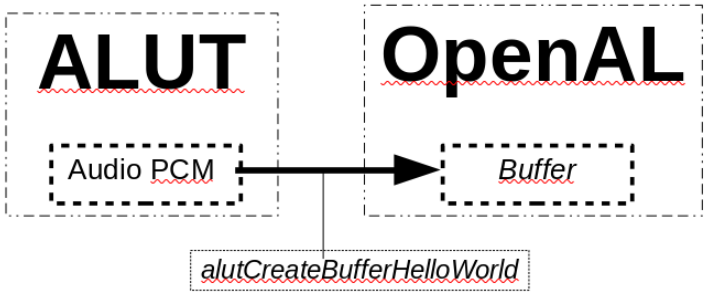
Fuente: OpenAL short example <https://ffainelli.github.io/openal-example/ >.

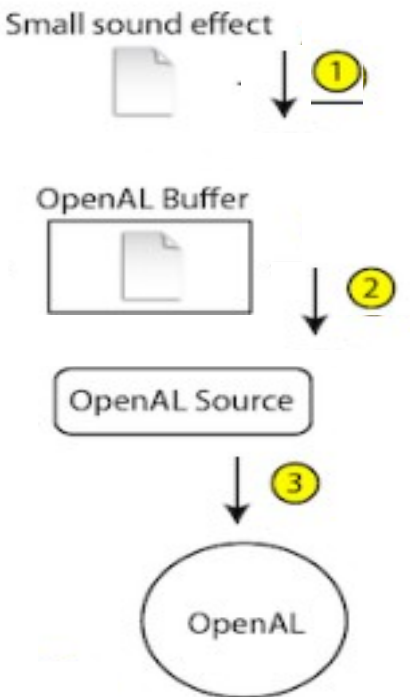
Seminario de OpenAL

- Ejemplos de uso del API
  - Hello, World!
  - Generación de sonidos simples
  - Generación de sonidos desde fichero
  - Audio posicional
  - Información
  - Efecto Doppler
  - Extensiones: uso del micrófono
  - Streaming
  - Extensiones: efectos ambientales

Ejemplos de uso

- Hello, World!
  - Pasos (1), (2) y (3)



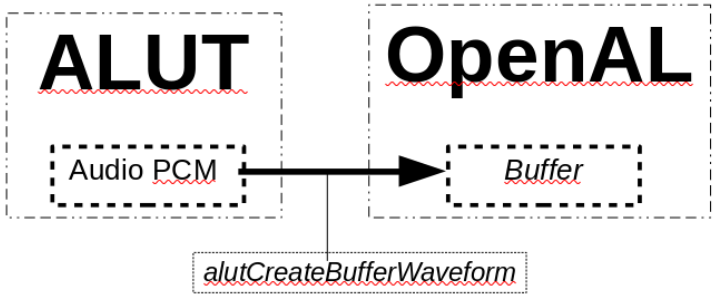


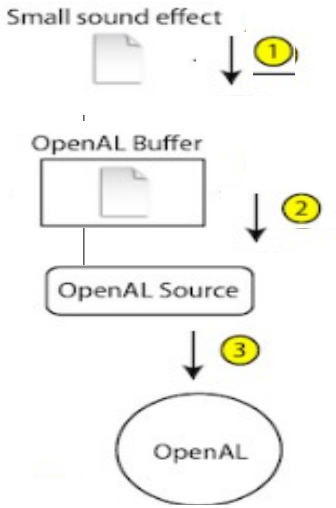

```
#include <AL/alut.h>

int main (int argc, char **argv) {
  ALuint elBuffer, laFuente;

  alutInit (&argc, argv);

  elBuffer = alutCreateBufferHelloWorld ();
  alGenSources (1, &laFuente);
  alSourcei (laFuente, AL_BUFFER, elBuffer);

  alSourcePlay (laFuente);
  alutSleep (1);

  alDeleteSources(1, &elBuffer);
  alDeleteBuffers(1, &laFuente);
  alutExit ();
  return EXIT_SUCCESS;
}
```

Fuente: Imagen variación de Valin. J.M., Vos, K y Terriberry, T. (2012). Definition of the Opus Audio Codec. Disponible en <https://tools.ietf.org/html/rfc6716 >.

Ejemplos de uso

- Generando sonidos simples
  - Pasos (1), (2) y (3)



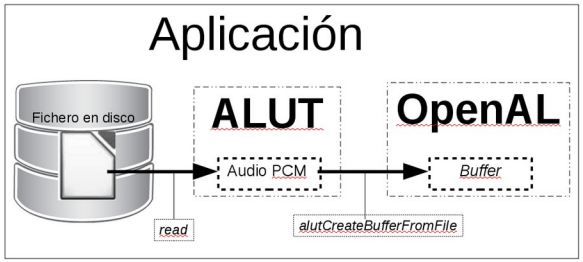


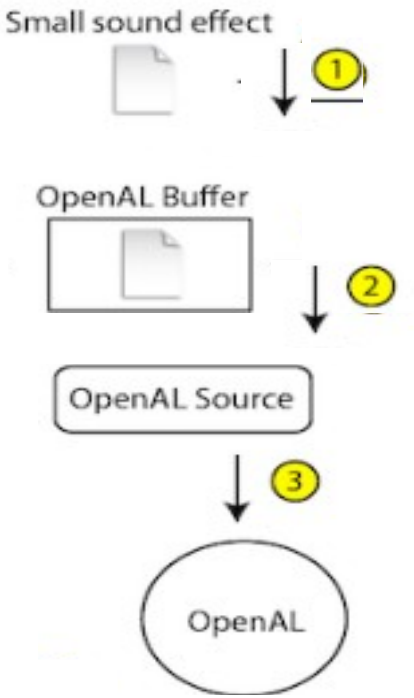

```
#include <AL/alut.h>

int main (int argc, char **argv) {
 ALuint elBuffer, laFuente;

 alutInit (&argc, argv);

 elBuffer = alutCreateBufferWaveform(ALUT_WAVEFORM_SINE, 440.0, 0.0, 1.0);
  // ALUT_WAVEFORM_SQUARE, ALUT_WAVEFORM_SAWTOOTH,
  // ALUT_WAVEFORM_WHITENOISE, ALUT_WAVEFORM_IMPULSE
 alGenSources (1, &laFuente);
 alSourcei (laFuente, AL_BUFFER, elBuffer);
 alSourcePlay (laFuente);

 alutSleep (1);

 alDeleteSources(1, &elBuffer);
 alDeleteBuffers(1, &laFuente);
 alutExit ();
 return( EXIT_SUCCESS );  }
```

Fuente: Imagen variación de Valin. J.M., Vos, K y Terriberry, T. (2012). Definition of the Opus Audio Codec. Disponible en <https://tools.ietf.org/html/rfc6716 >.

Ejemplos de uso

- Generación de sonidos desde fichero
  - Pasos (1), (2) y (3)






```
#include <AL/alut.h>
 int main (int argc, char **argv) {
   ALuint elBuffer, laFuente;
   ALint sourceState;

   alutInit (&argc, argv);

   helloBuffer = alutCreateBufferFromFile( argv[1] );

   alGenSources (1, &laFuente);
   alSourcei(laFuente, AL_BUFFER, elBuffer);
   alSourcePlay (laFuente);

   // Lo vamos a dejar sonar mientras hayan datos.
   alGetSourcei( laFuente, AL_SOURCE_STATE, &sourceState);
   while (sourceState == AL_PLAYING)
     alGetSourcei( laFuente, AL_SOURCE_STATE, &sourceState);

   alutExit ();
   return( EXIT_SUCCESS );
}
```

Fuente: Imagen variación de Valin. J.M., Vos, K y Terriberry, T. (2012). Definition of the Opus Audio Codec. Disponible en <https://tools.ietf.org/html/rfc6716 >.

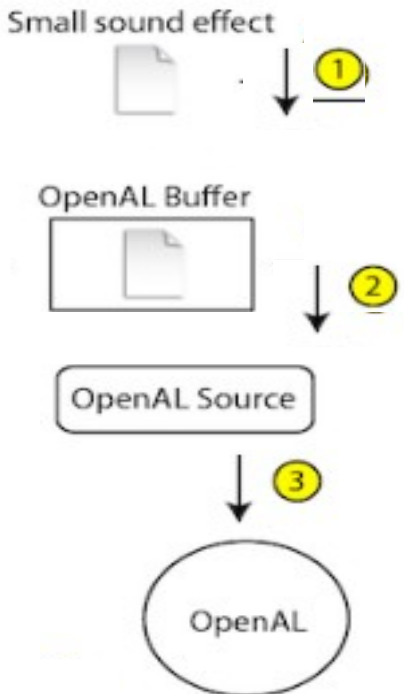
Ejemplos de uso

- Audio espacial (posicional)
  - Oyente

```
ALfloat listenerPos[]={0.0,0.0,4.0};
ALfloat listenerVel[]={0.0,0.0,0.0};
Alfloat  listenerOri[]={0.0,0.0,1.0, 0.0,1.0,0.0};
...
   alListenerfv(AL_POSITION, listenerPos);
   alListenerfv(AL_VELOCITY, listenerVel);
   alListenerfv(AL_ORIENTATION, listenerOri);
```

  - La/s fuente/s de sonido

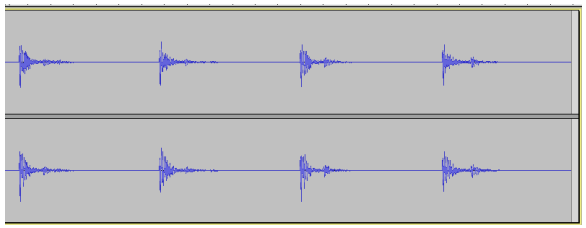
```
#define NUM_BUFFERS 3
Aluint   buffer[NUM_BUFFERS];

ALfloat source0Pos[]={ -2.0, 0.0, 0.0};
ALfloat source0Vel[]={ 0.0, 0.0, 0.0};
...
   alSourcefv(source[0], AL_POSITION, source0Pos);
   alSourcefv(source[0], AL_VELOCITY, source0Vel);
```



Fuente: Imagen variación de Valin. J.M., Vos, K y Terriberry, T. (2012). Definition of the Opus Audio Codec. Disponible en <https://tools.ietf.org/html/rfc6716 >.

Ejemplos de uso

- Ejemplo de espacialización
  - Sonido: footsteps-4.wav
    - 16 bit, 48000 Hz., estéreo
  - Implementación
    - …



```
alSourcei( source, AL_BUFFER, buffer );
…
alSourcePlay( source );
…
float toVel = vel/dist;
float v[3] = {dx*toVel, dy*toVel, dz*toVel};
curr[0]+= v[0]; curr[1]+= v[1]; curr[2]+= v[2];

alSource3f( source, AL_POSITION, curr[0],curr[1],curr[2] );
alSource3f( source, AL_VELOCITY, v[0],v[1],v[2] );
```

Fuente: “ OpenAL Soft -- demonstration of binaural 3D audio “ <http://cranialburnout.blogspot.com.es/2012/08/openal-soft-demonstration-of-binaural.html>

Ejemplos de uso

- Información: de versiones

```
...
  alutInit (&argc, argv);

  printf("Versió OpenAL: %s\n", alGetString( AL_VERSION ) );

  printf("OpenAL Renderer is '%s'\n", alGetString(AL_RENDERER) );
  printf("OpenAL Version is '%s'\n", alGetString(AL_VERSION) );
  printf("OpenAL Vendor is '%s'\n", alGetString(AL_VENDOR) );

  printf("Versió ALUT: %d.%d\n", alutGetMajorVersion(), alutGetMinorVersion() );
...
```

```
$ informacioOpenAL
Versió OpenAL: 1.1 ALSOFT 1.14
OpenAL Renderer is 'OpenAL Soft'
OpenAL Version is '1.1 ALSOFT 1.14'
OpenAL Vendor is 'OpenAL Community'


Versió ALUT: 1.1
```

Ejemplos de uso

- Información: de extensiones

```
 const ALchar *pDeviceList, *llista;
...

printf("OpenAL Extensions supported are :\n%s\n", alGetString(AL_EXTENSIONS) );
 llista = alGetString(AL_EXTENSIONS);
 i=1;
 if (llista)
   {
    printf("\nExtensions disponibles per separat:");
    printf("\n%2d - ", i);
    while (*llista)
      {
       if (*llista != ' ' )
                 printf("%c", *llista);
               else
                 printf("\n%2d - ", ++i);
               llista++;
      }
   }
...
```

Ejemplos de uso

- Información de extensiones

$ informacioOpenAL
OpenAL Extensions supported are :
AL_EXT_ALAW AL_EXT_DOUBLE AL_EXT_EXPONENT_DISTANCE AL_EXT_FLOAT32
AL_EXT_IMA4 AL_EXT_LINEAR_DISTANCE AL_EXT_MCFORMATS AL_EXT_MULAW
AL_EXT_MULAW_MCFORMATS AL_EXT_OFFSET AL_EXT_source_distance_model
AL_LOKI_quadriphonic AL_SOFT_buffer_samples AL_SOFT_buffer_sub_data
AL_SOFTX_deferred_updates AL_SOFT_direct_channels AL_SOFT_loop_points

Extensions disponibles per separat:
1 - AL_EXT_ALAW
2 - AL_EXT_DOUBLE
3 - AL_EXT_EXPONENT_DISTANCE
4 - AL_EXT_FLOAT32
5 - AL_EXT_IMA4
6 - AL_EXT_LINEAR_DISTANCE
7 - AL_EXT_MCFORMATS
8 - AL_EXT_MULAW
9 - AL_EXT_MULAW_MCFORMATS
10 - AL_EXT_OFFSET
11 - AL_EXT_source_distance_model
12 - AL_LOKI_quadriphonic
13 - AL_SOFT_buffer_samples
14 - AL_SOFT_buffer_sub_data
15 - AL_SOFTX_deferred_updates
16 - AL_SOFT_direct_channels
17 - AL_SOFT_loop_points

Ejemplos de uso

- Información: hardware

```
 const ALchar *pDeviceList, *llista;
...
 printf("OpenAL hardware presente:\n %s\n",
        alcGetString(NULL, ALC_DEVICE_SPECIFIER) );

 // Get list of available Capture Devices (Tret de Capture.c)
 pDeviceList = alcGetString(NULL, ALC_DEVICE_SPECIFIER);
 i=1;
 if (pDeviceList)
   {
    printf("\nAvailable Capture Devices are:\n");

    while (*pDeviceList)
      {
       printf("%2d - %s\n", i, pDeviceList);
       pDeviceList += strlen(pDeviceList) + 1;
              i++;
      }
   }

...
```

Ejemplos de uso

- Información: hardware

$ informacioOpenAL

...
OpenAL hardware presente:
 OpenAL Soft

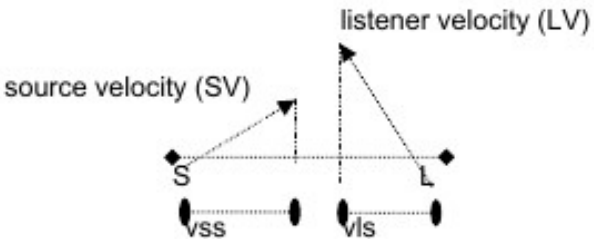
Drivers disponibles d'audio:
 1 - OpenAL Soft
Per defecte:
OpenAL Soft

Available Capture Devices are:
 1 - QuickCam Pro 9000 Mono analógico
 2 - Monitor of Audio Interno Estéreo analógico

Ejemplos de uso

- Efecto *Doppler*
  - Modelo:

$$f' = \frac{freq * (SPEED_{SOUND} - DOPPLER_{FACTOR} * LV)}{SPEED_{SOUND} - DOPPLER_{FACTOR} * SV}$$



    - Velocidad del oyente (***l****istener*) y de la fuente (***s****ource*)
    - Frecuencia (*unaltered*) de la señal (wave) original
    - "shift" (f') es el resultado (*altered frequency*) que es aplicado al flujo de audio

  - OpenAL:

```
ALvoid alDopplerFactor( ALfloat factor );  // >= 0
ALvoid alSpeedOfSound( ALfloat velocity );  // 343,3
```

    - Velocidad fuentes y oyente
      - 'AL_VELOCITY' de 'alListenerfv' y 'alSourcefv'.
      - La 'freq' está en las propiedades del *buffer* asignado a la fuente.

Ejemplos de uso

- Extensiones: uso del micrófono
  - ALC_EXT_capture
    - alcCaptureCloseDevice
    - alcCaptureOpenDevice
    - alcCaptureSamples
    - alcCaptureStart
    - alcCaptureStop

OpenAL Extension Registry

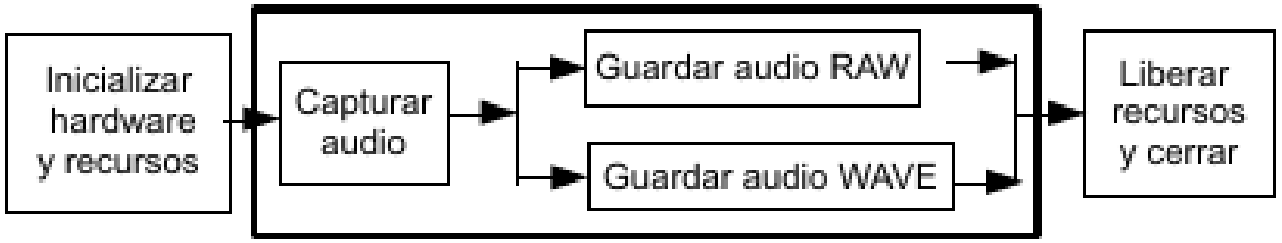
Extensions:

- 1.0_specification
- ALC_ENUMERATE_ALL_EXT
- ALC_ENUMERATION_EXT
- ALC_EXT_ASA_DISTORTION
- ALC_EXT_ASA_ROGER_BEEP
- ALC_EXT_BRS_GAME_LICENSE_REQUIRED
- ALC_EXT_capture
- ALC_EXT_DEDICATED
- ALC_EXT_DEFAULT_FILTER_ORDER
- ALC_EXT_disconnect
- ALC_EXT_EFX
- AL_EXT_ALAW
- AL_EXT_BFORMAT
- AL_EXT_double
- AL_EXT_float32
- AL_EXT_FOLDBACK
- AL_EXT_IMA4
- AL_EXT_MCFORMATS
- AL_EXT_mp3
- AL_EXT_MULAW
- AL_EXT_MULAW_BFORMAT
- AL_EXT_MULAW_MCFORMATS
- AL_EXT_SOURCE_RADIUS
- AL_EXT_STEREO_ANGLES
- AL_EXT_vorbis
- EAX-RAM
- EAX2.0
- EAX3.0
- EAX4.0
- EAX5.0

Total results: 30

Ejemplos de uso

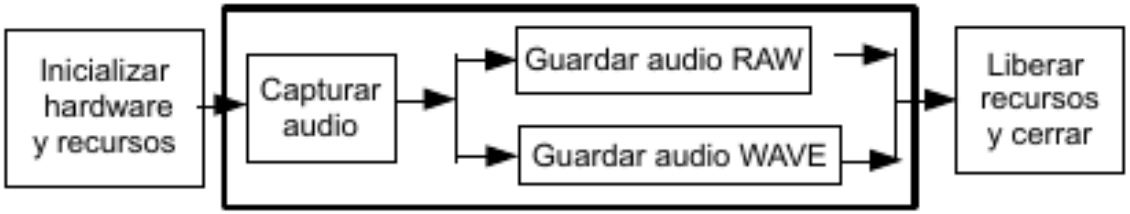
- Extension de captura de audio
  - Inicialización




```
...
// Hardware
ALCcontext *pContext = alcGetCurrentContext();
ALCdevice *pDevice = alcGetContextsDevice(pContext);
if (alcIsExtensionPresent(pDevice, "ALC_EXT_CAPTURE") == AL_FALSE)
{
   printf("Fallo al detectar extensión de captura.\n"); return 0;
}

szDefaultCaptureDevice = alcGetString(NULL, ALC_CAPTURE_DEFAULT_DEVICE_SPECIFIER);
printf("\nDispositivo de captura por defecto es '%s'\n\n", szDefaultCaptureDevice);


ALCdevice *device;
device = alcCaptureOpenDevice(szDefaultCaptureDevice,  freq, AL_FORMAT_MONO16, BUFFERSIZE);
printf("Dipositivo de captura: '%s' esta abierto\n\n",  alcGetString(device,  ALC_CAPTURE_DEVICE_SPECIFIER));

// Ficheros: RAW y WAVE (mediante libsndfile).
...
```

Ejemplos de uso

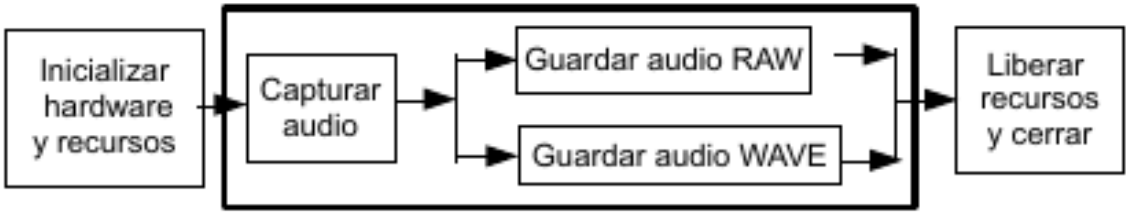
- Extension de captura de audio
  - Inicialización




```
...
// Ficheros:
//  RAW
pFileRAW = fopen(OUTPUT_WAVE_FILE_RAW, "wb");

// y WAVE mediante libsndfile.
SF_INFO info;
info.format = SF_FORMAT_WAV | SF_FORMAT_PCM_16;
info.channels = 1;
info.samplerate = SRATE;
SNDFILE *pFileWAVE = sf_open(OUTPUT_WAVE_FILE, SFM_WRITE, &info);
..
```

Ejemplos de uso

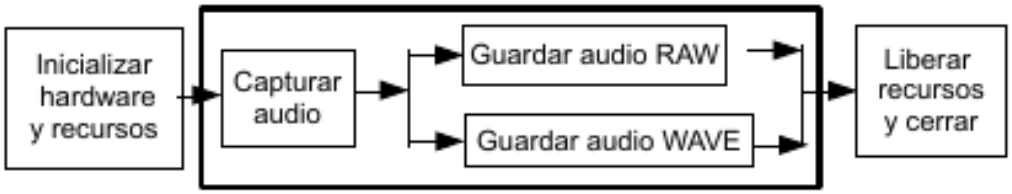
- Extension de captura de audio
  - Captura




```
...
alcCaptureStart(device);
float bloq = info.channels * 16/8; // bytes por bloque
while (!kbhit()) {
   alcGetIntegerv(device, ALC_CAPTURE_SAMPLES, BUFFERSIZE, &nMostres);
   printf("nMostress: %d\n", nMostres);
   if (nMostres > (BUFFERSIZE / bloq))  {
     alcCaptureSamples(device, Buffer, BUFFERSIZE/bloq);
     fwrite(Buffer,BUFFERSIZE , 1, pFileRAW);
     sf_writef_short(pFileWAVE, (short*)Buffer, BUFFERSIZE/bloq);
     }
     else{
     alcCaptureSamples(device, (ALCvoid *)Buffer, nMostres);
     fwrite(Buffer, nMostres, 1, pFileRAW);
     sf_writef_short(pFileWAVE, (short*)Buffer, nMostres/bloq);
  }
}
...
```

Ejemplos de uso

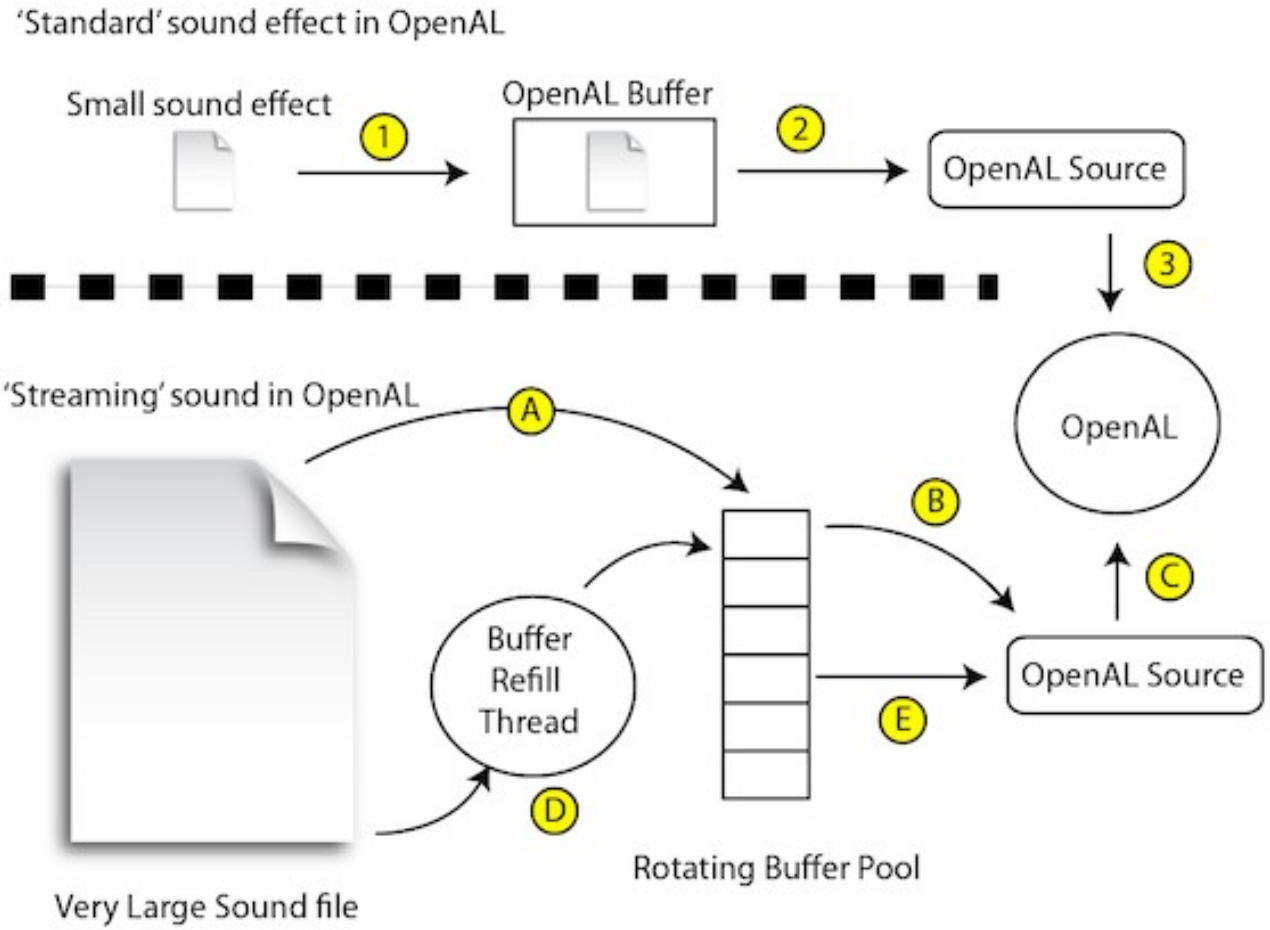
- Extension de captura de audio
  - Liberar recursos




```
...
//cerramos todo: Hard. ...
alcCaptureStop(device);
alcCaptureCloseDevice(device);

// y ficheros: RAW
fclose( pFileRAW );

// ...  y WAVE
sf_write_sync( pFileWAVE );
sf_close( pFileWAVE );
printf("\nSaved captured audio data to '%s'\n", OUTPUT_WAVE_FILE);

alcMakeContextCurrent(NULL);
alcDestroyContext(pContext);
alcCloseDevice(pDevice);

return EXIT_SUCCESS;
}
```

Ejemplos de uso

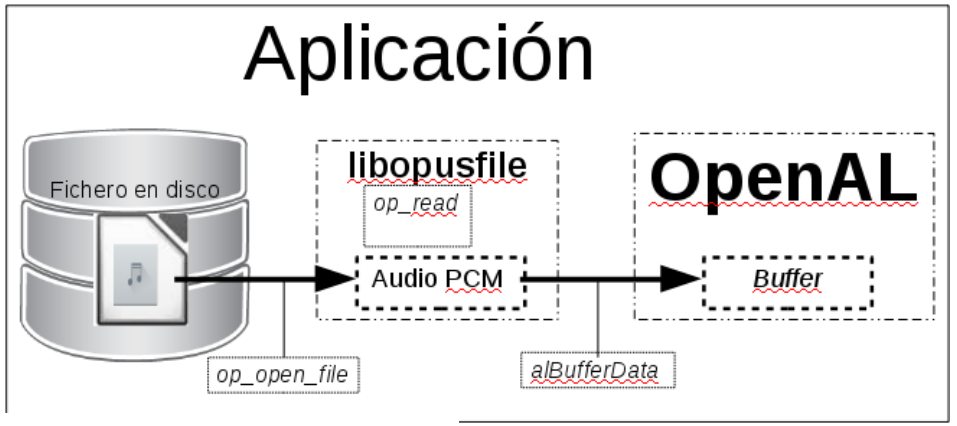
- Precarga vs Streaming
  - Planteamiento



Fuente: [6] Valin. J.M., Vos, K y Terriberry, T. (2012). Definition of the Opus Audio Codec. Disponible en <https://tools.ietf.org/html/rfc6716>.

Ejemplos de uso

- Ejemplo de precarga con ficheros: Opus
  - OpenAL + libopusfile




```
int main(int argc, char **argv) {
 ALuint testBuffer,  testSource;
 ...
 load_opus(testBuffer, argv[1]);
 ...
 alSourcePlay(testSource);
 ...
 int sourceState;
 do {
  alGetSourcei(testSource, AL_SOURCE_STATE, &sourceState);
 } while (sourceState != AL_STOPPED);
 ...
 return 0;
}
```

Fuente: M. Agustí. (2018). Reproducción de ficheros Opus con OpenAL: precarga vs "streaming". URL: <http://hdl.handle.net/10251/109211 >.

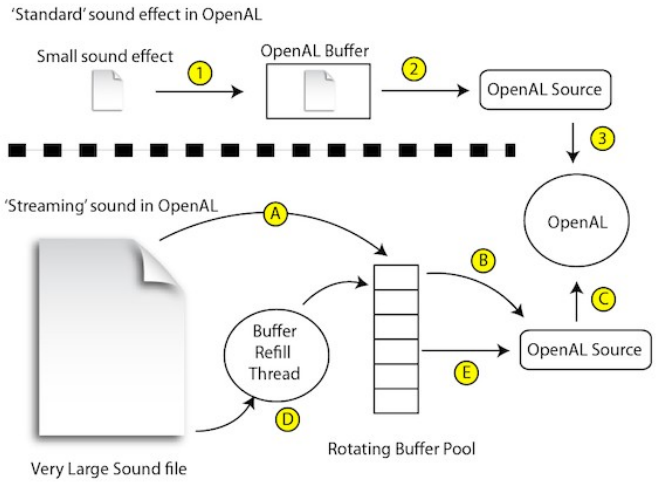
Ejemplos de uso

```
int load_opus(ALuint buffer, const char *filename) {
  int error = 0;
  OggOpusFile *file = op_open_file(filename, &error); // Open the file.
  ...
  // Get the number of channels in the current link.
  int num_channels = op_channel_count(file,-1);
  // Get the number of samples (per channel) in the current link.
  int pcm_size = op_pcm_total(file,-1);
  ...
  // Opus always uses 16-bit integers, unless the _float are used
  ALenum format;
  if (num_channels == 1) { format = AL_FORMAT_MONO16;  }
  else if (num_channels == 2)    { format = AL_FORMAT_STEREO16;   }
  else {...}
  // Allocate a buffer big enough to store the entire uncompressed file.
  int16_t *buf = new int16_t[pcm_size*num_channels];
  ...
  // Keep reading samples until we have them all.
  while (samples_read < pcm_size)           {
     // op_read returns number of samples read (per channel), ....
     int ns = op_read(file, buf + samples_read*num_channels, pcm_size*num_channels, 0);
     …
     samples_read += ns;
      …
  }
  op_free(file);      // Close the opus file.
  // Send it to OpenAL (which takes bytes).
  alBufferData(buffer, format, buf, samples_read*num_channels*2, 48000);
  …
 }
```

Fuente: M. Agustí. (2018). Reproducción de ficheros Opus con OpenAL: precarga vs "streaming". URL: <http://hdl.handle.net/10251/109211 >.

Ejemplos de uso

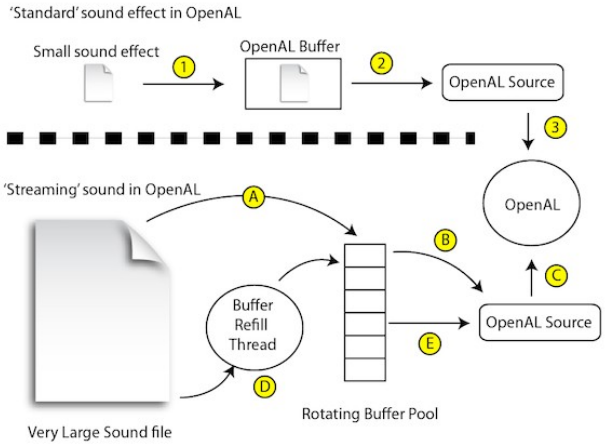
- Precarga vs Streaming
  1. Inicializar OpenAL
  2. Obtener información
     del fichero
  3. Mientras haya datos que
     leer: rellenar los buffers
  4. Descomprimir y decodificar
     Añadir el buffer a la lista
     Asignar a la fuente un primer buffer relleno
     de la lista y reproducir
     Sacar buffer leído de la lista
  5. Liberar recursos utilizados



Fuente: Valin. J.M., Vos, K y Terriberry, T. (2012). Definition of the Opus Audio Codec. Disponible en <https://tools.ietf.org/html/rfc6716>.

Ejemplos de uso

- Ejemplo de *streaming* con fich
  - OpenAL + libopusfile

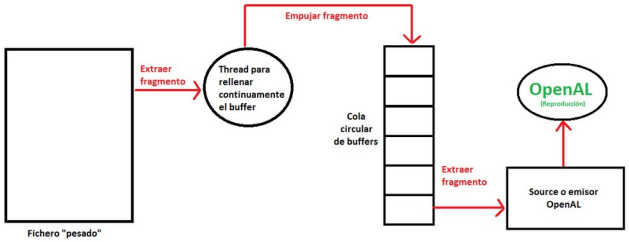


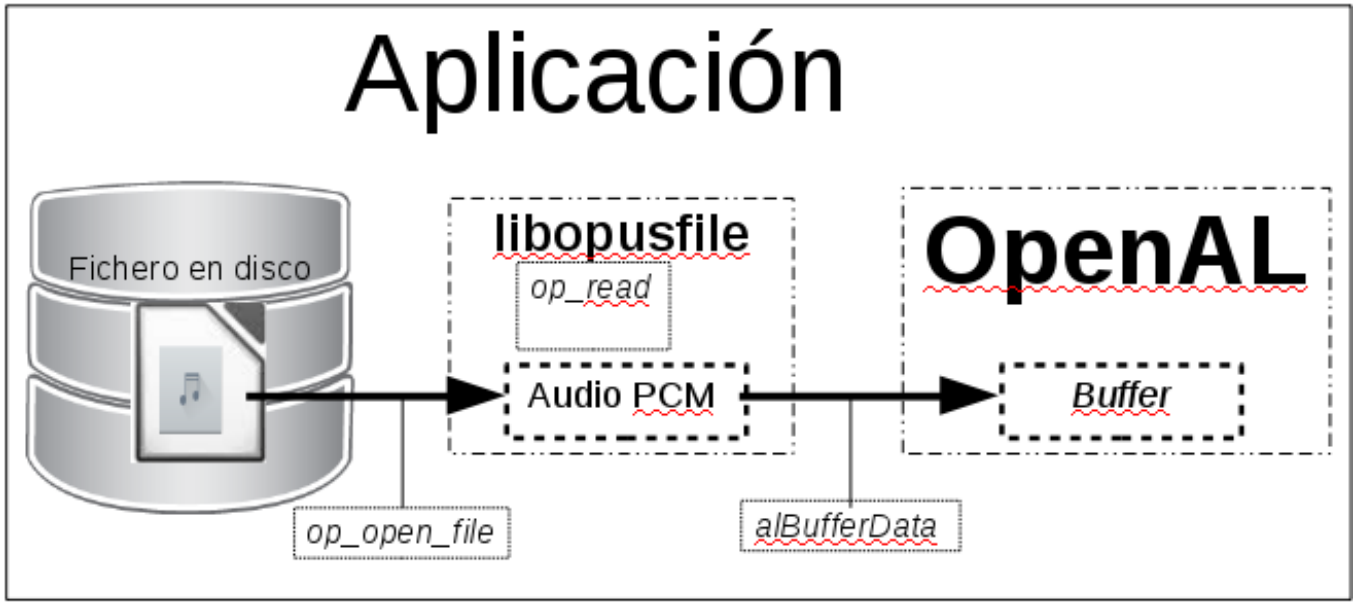

```
int update_stream(ALuint source,
                  OggOpusFile *file) {
  int num_processed_buffers = 0;
  ALuint currentbuffer;

  // How many buffers do we need to fill?
  alGetSourcei(source, AL_BUFFERS_PROCESSED, &num_processed_buffers);

  ALenum source_state;
  alGetSourcei(source, AL_SOURCE_STATE, &source_state);
  if (source_state != AL_PLAYING) {
    printf("Source not playing!\n");
    alSourcePlay(source);
  }

  // Unqueue a finished buffer, fill it with new data, and re-add it
  while (num_processed_buffers--)    {
    alSourceUnqueueBuffers(source, 1 ,&currentbuffer);
    if (fill_buffer(currentbuffer, file) <= 0)return 0;
    alSourceQueueBuffers(source, 1, &currentbuffer);
  }
  return( 1 ); }
```

Fuente: M. Agustí. (2018). Reproducción de ficheros Opus con OpenAL: precarga vs "streaming". URL: <http://hdl.handle.net/10251/109211 >.

Ejemplos de uso

- Ejemplo de *streaming* con ficheros: OGG Vorbis
  - OpenAL + libopusfile

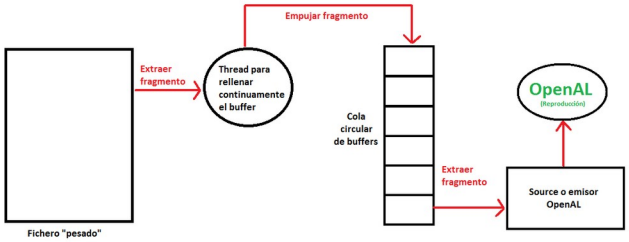


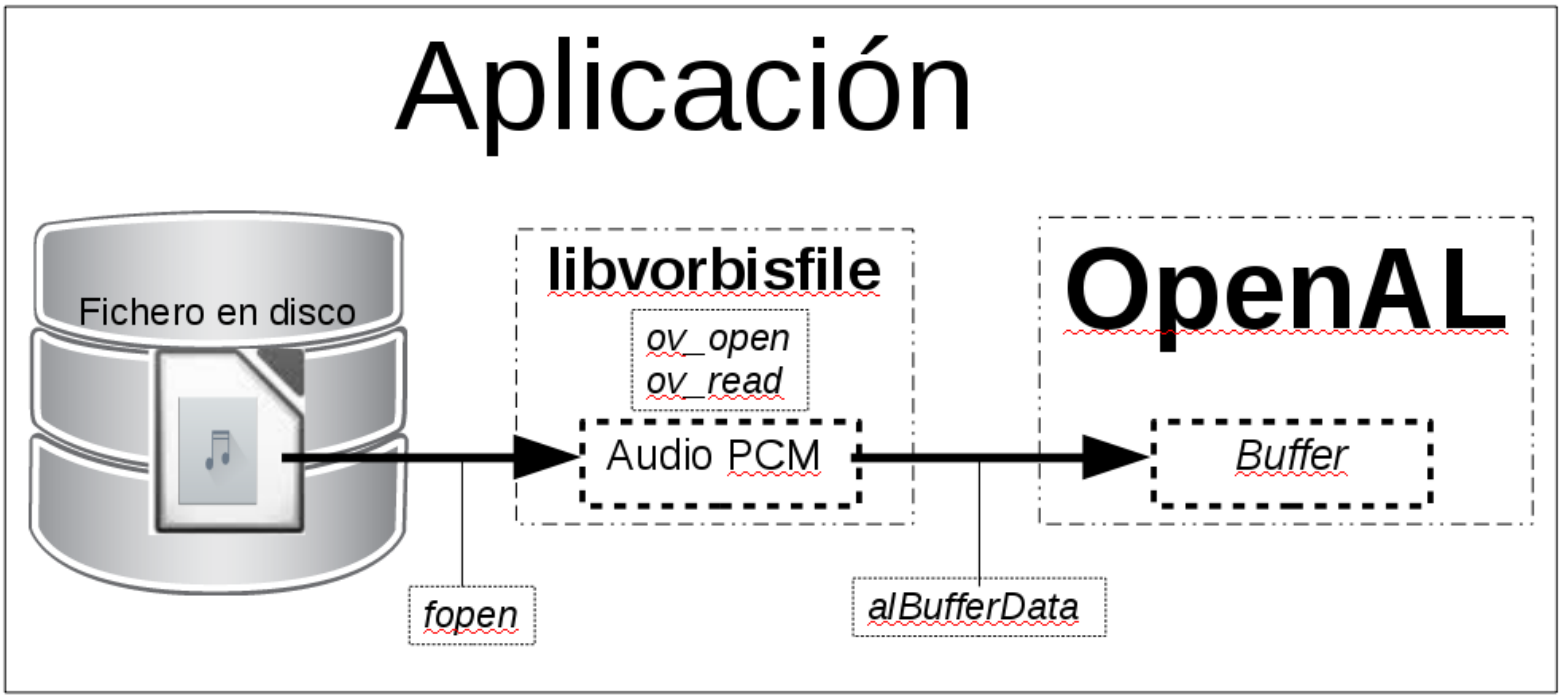


Fuente: M. Agustí. (2018). Reproducción de ficheros Opus con OpenAL: precarga vs "streaming". URL: <http://hdl.handle.net/10251/109211 >.

Ejemplos de uso

- Ejemplo de *streaming* con ficheros: OGG Vorbis
  - OpenAL + libvorbis

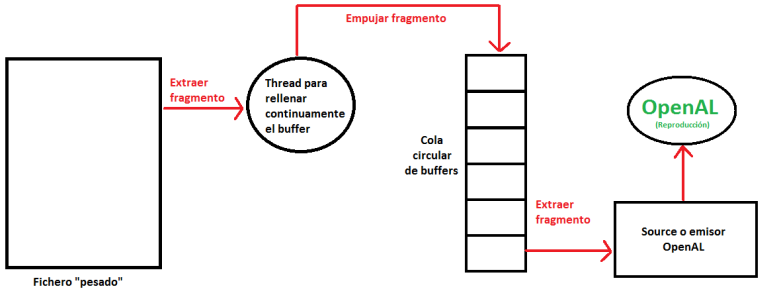


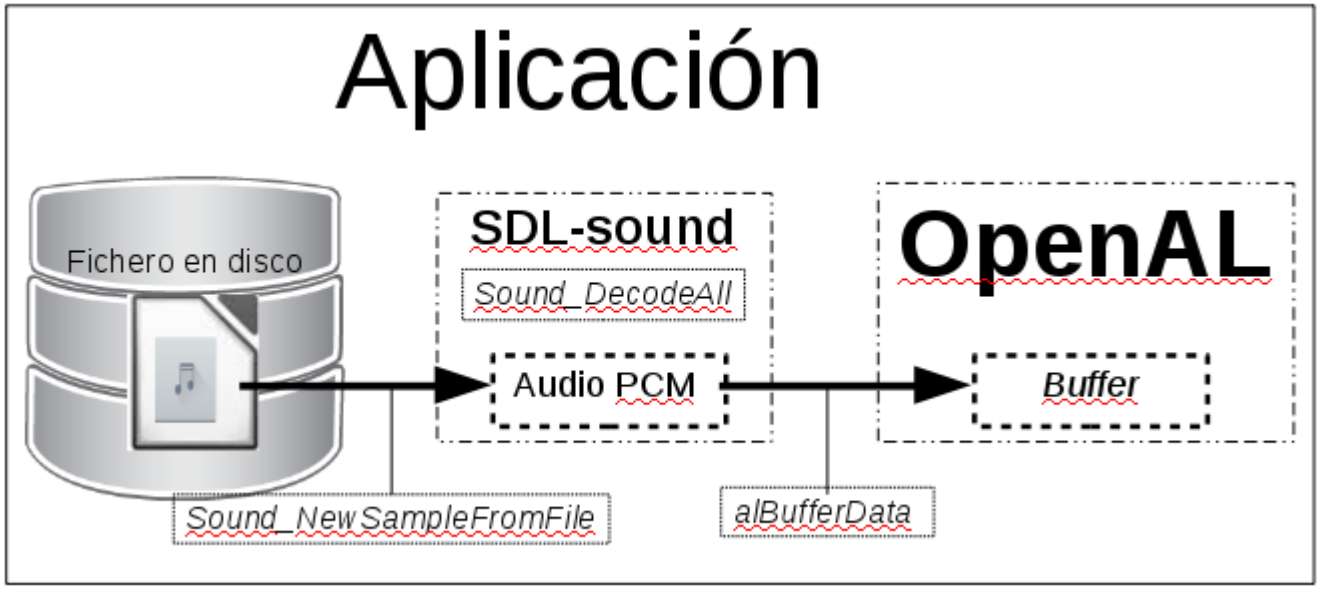


Fuente: M. Agustí. (2018). Extendiendo OpenAL con OGG Vorbis. URL < http://hdl.handle.net/10251/109210 >.

Ejemplos de uso

- Ejemplo de *streaming* con ficheros: MP3
  - OpenAL + SDL





Fuente: M. Agustí. (2018). Extendiendo OpenAL con SDL. Caso de estudio MP3.Disponible en < http://hdl.handle.net/10251/105383>.

Ejemplos de uso

- *Effects Extension (EFX)*
  - ALC_EXT_EFFECTS
  - Amplia la arquitectura de OpenAL
  - Acceso a modelos de **efectos ambientales:**
    - Reverb, eco, coros,
    - Distorsión y
    - Filtros de frecuencia

OpenAL Extension Registry

Extensions:

- 1.0_specification
- ALC_ENUMERATE_ALL_EXT
- ALC_ENUMERATION_EXT
- ALC_EXT_ASA_DISTORTION
- ALC_EXT_ASA_ROGER_BEEP
- ALC_EXT_BRS_GAME_LICENSE_REQUIRED
- ALC_EXT_capture
- ALC_EXT_DEDICATED
- ALC_EXT_DEFAULT_FILTER_ORDER
- ALC_EXT_disconnect
- ALC_EXT_EFX
- AL_EXT_ALAW
- AL_EXT_BFORMAT
- AL_EXT_double
- AL_EXT_float32
- AL_EXT_FOLDBACK
- AL_EXT_IMA4
- AL_EXT_MCFORMATS
- AL_EXT_mp3
- AL_EXT_MULAW
- AL_EXT_MULAW_BFORMAT
- AL_EXT_MULAW_MCFORMATS
- AL_EXT_SOURCE_RADIUS
- AL_EXT_STEREO_ANGLES
- AL_EXT_vorbis
- EAX-RAM
- EAX2.0
- EAX3.0
- EAX4.0
- EAX5.0

Total results: 30

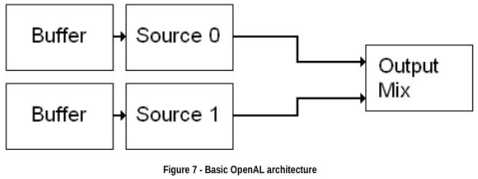


Figure 7 - Basic OpenAL architecture

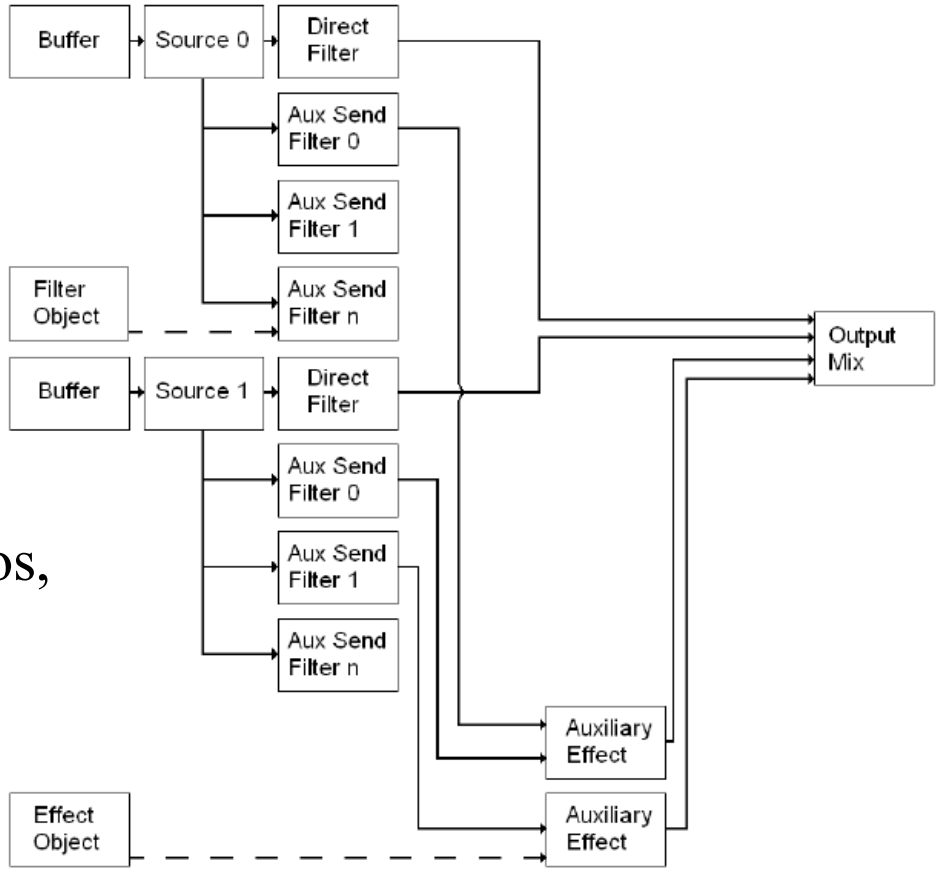
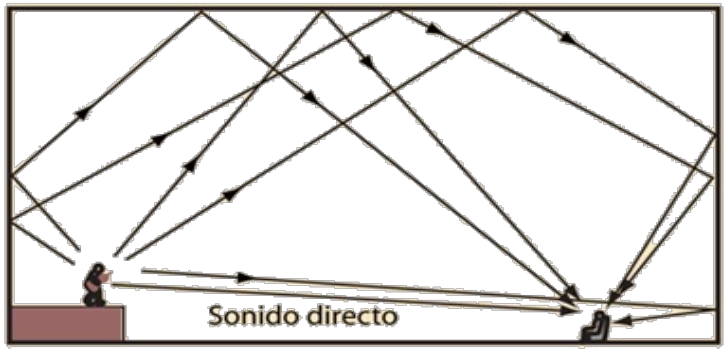


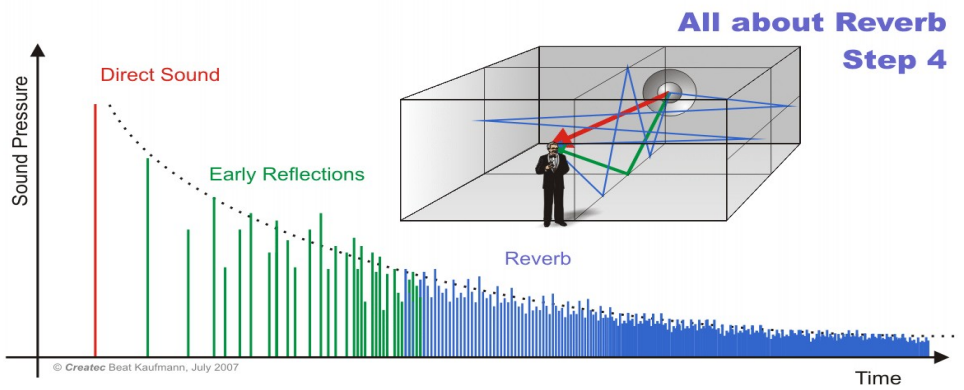
Figure 8 - OpenAL with Effects Extension architecture

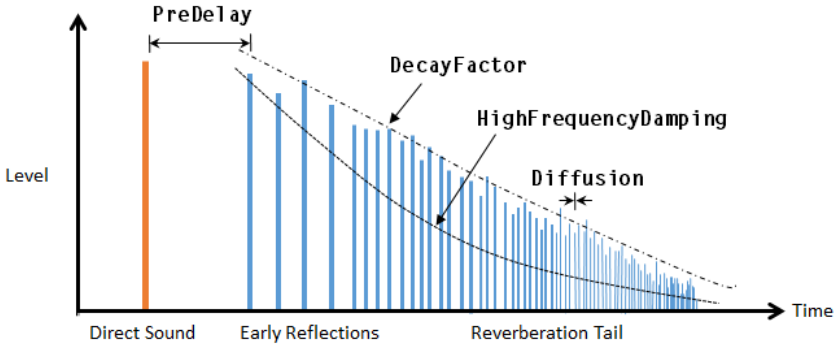
Fuente: M. Agustí. (2018). Extensiones para OpenAL: efectos ambientales. http://hdl.handle.net/10251/105664 y Peacock, D, Harrison, P., D'Orta, A., Carpentier, V., Cooper, E. (2006). Effects Extension Guide v 1.1. <http://kcat.strangesoft.net/misc-downloads/Effects%20Extension%20Guide.pdf >.

Ejemplos de uso

- *Effects Extension (EFX)*
  - Reverberación: sonido directo vs reflexiones
    - Primeros rebotes (10..30 ms.) y últimos (50..100ms.)







AL_EFFECT_REVERB

Version	Parameter Name	Units	Range	Default
1.0	AL_REVERB_DENSITY		[0.0, 1.0]	1.0
1.0	AL_REVERB_DIFFUSION		[0.0, 1.0]	1.0
1.0	AL_REVERB_GAIN		[0.0, 1.0]	0.32
1.0	AL_REVERB_GAINHF		[0.0, 1.0]	0.89
1.0	AL_REVERB_DECAY_TIME	Seconds	[0.1, 20]	1.49
1.0	AL_REVERB_DECAY_HFRATIO		[0.1, 2.0]	0.83
1.0	AL_REVERB_REFLECTIONS_GAIN		[0.0, 3.16]	0.05
1.0	AL_REVERB_REFLECTIONS_DELAY	Seconds	[0.0, 0.3]	0.007
1.0	AL_REVERB_LATE_REVERB_GAIN		[0.0, 10.0]	1.26
1.0	AL_REVERB_LATE_REVERB_DELAY	Seconds	[0.0, 0.1]	0.011
1.0	AL_REVERB_AIR_ABSORPTION_GAINHF		[0.892,1.0]	0.994
1.0	AL_REVERB_ROOM_ROLLOFF_FACTOR		[0.0. 10.0]	0.0
1.0	AL_REVERB_DECAY_HFLIMIT	ON / OFF	[AL_FALSE, AL_TRUE]	AL_TRUE

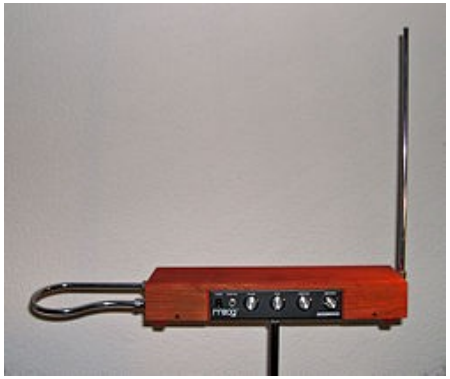
Fuente: Imágenes de <http://hyperphysics.phy-astr.gsu.edu/hbasees/Acoustic/reverb.html > ,
<https://cursodesonido.webnode.com.co/curso/teoria-del-sonido/modulo-05-acustica-ii-reverberacion-eco-/ >.,
<https://es.mathworks.com/help/audio/ref/reverberator-system-object.html > y .<http://kcat.strangesoft.net/misc-downloads/Effects%20Extension%20Guide.pdf >.

Audio e interacción

- *Sonido e instrumentos virtuales*
  - *Lanzar la idea*
    "Interacción natural en instrumentos musicales"
    → Sonido e instrumentos virtuales
  - Encargar la tarea de OpenCV + OpenAL
    - Generación de sonidos básicos a partir del movimiento en la escena
  - Presentar el uso del movimiento para implementar instrumentos musicales virtuales
    - Theremín
    - Arpa láser

Sonido e instrumentos virtuales

- *Interacción natural en instrumentos*
  - *Theremín*
    - *Instrumento musical sin contacto directo*
    - *"the loop antenna on the left controls the volume while the upright antenna controls the pitch".*



An Etherwave-Theremin, assembled from Robert Moog's kit

Lev Theremin, 1927

  - *Arpa láser*

Imágenes de: <http://en.wikipedia.org/wiki/L%C3%A9on_Theremin>, <http://en.wikipedia.org/wiki/Theremi

Arpa láser

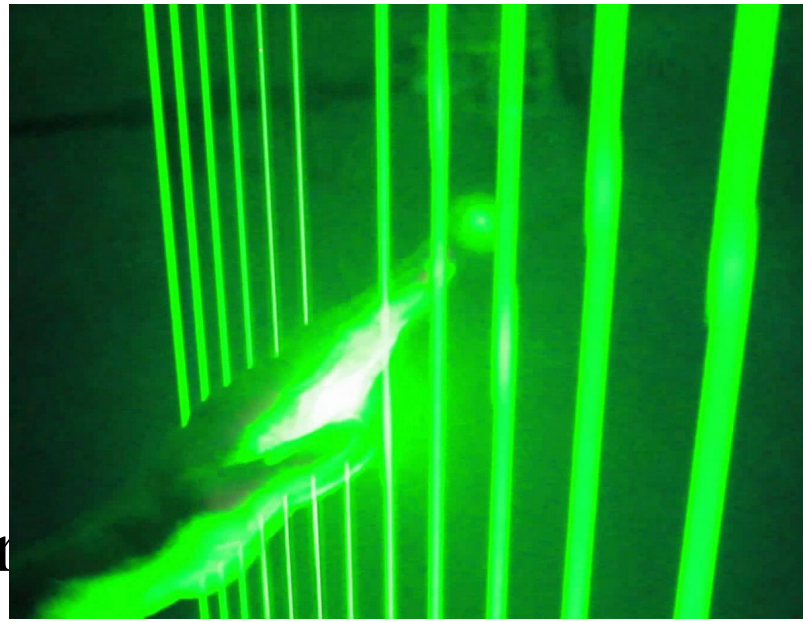
- *Interacción natural en instrumentos*
  - *Theremín*
  - *Arpa láser (Laser Harp)*

- *A light-sensitive musical instrument. It is played by moving hands over laser light sources in order to send MIDI commands when a beam is interrupted. ...*
- *"The first Laser Harp was invented and played by Bernard SZAJNER"*
  - *(<www.harpelaser.com>, 1981)*
- *Jean Michel Jarre*

  - *A partir de los sonidos del Elka Synthethizer Laser Harp Sound*
- *Lasertron: Laser Harp App for iPhone, iPad*

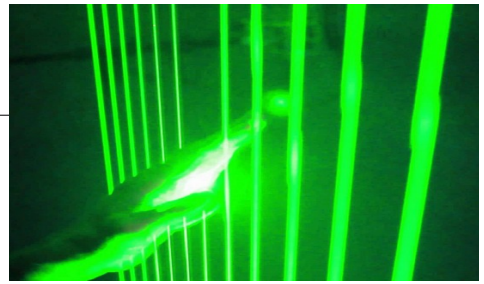
Fuente: <https://www.youtube.com/watch?v=9-f0bz8S9Ug>, <https://www.youtube.com/watch?v=zaYkgY5_toE&feature=youtu.be> y <https://jmjarrefan.wordpress.com/2012/12/19/fun-laser-harp-app-for-iphone-ipad/ >.

Arpa láser



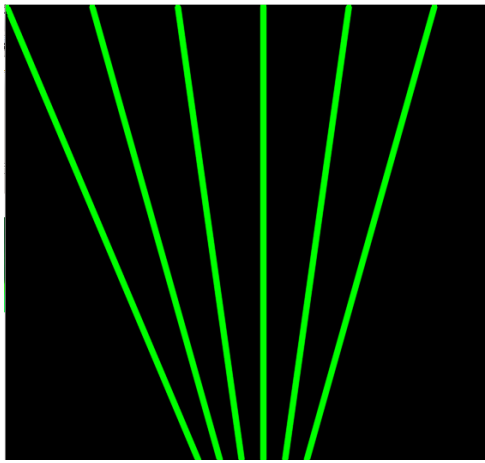
- Idea
  - Crear una aplicación que simule tocar un arpa
  - OpenCV para análisis de imágen de la cámar
  - OpenAL para las fuentes de audio.
- Solución
  - Identificar qué haz de luz es tocada.
    - ¿Interacción por color con OpenCV?
    - ¿Detectar movimiento sobre el “haz de luz”?

Arpa láser



- Eventos ← OpenCV
  - Imagen del “instrumento”

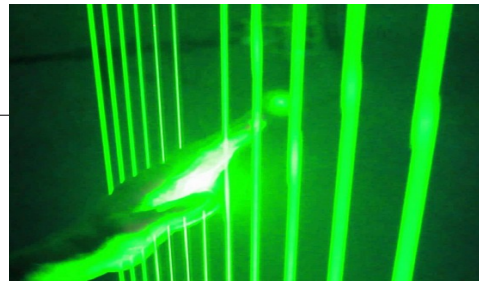
```
for(int i=0;i<6;i++){
    cvLine(imagen, cvPoint(300,800), cvPoint(i*100,0),
        cvScalar(0,255,0, 1),5, line_type, shift );
    cvLine(imagen2, cvPoint(300,800), cvPoint(i*100,0),
        cvScalar(10+i*10,10+i*10,10+i*10, 1),5,
            line_type, shift );
  }
```



  - Imagen intermedia: IDs de las “cuerdas

```
elColor = cvGet2D(imagenIntermedia, y, x);
for(int i=0;i<6;i++){
  if((int)elColor.val[0]==i*10+10){
      antigua=i;
    cvLine(imagen, cvPoint(300,800), cvPoint(i*100,0),
            cvScalar(0,0,0, 1), 5, line_type, shift );
    cvLine(imagen, cvPoint(300,800), cvPoint(x,y),
            cvScalar(0,255,0, 1), 5, line_type, shift );

    //sonido que tiene que reproducir
…
```

Arpa láser



- Sonido ← OpenAL
  - Inicializar sonidos

```
alGenBuffers(6,Buffer);
Buffer[0]= alutCreateBufferFromFile( "arpa.wav" );
…
Buffer[5]= alutCreateBufferFromFile( "arpa3.wav" );
// ¿Polifonía?
alGenSources(6,Source);
```

  - Reproducir uno

```
alSourceStop(Source[antigua]);
alSourcei (Source[i], AL_BUFFER, Buffer[i]);
alSourcePlay(Source[i]);
```

Bibliografía y enlaces

- *The Tapeless Studio: online magazine of audio recording on the PC.*
- *Short MPEG-2 Description (drogo.cselt.stet.it/ufv/leonardo/mpeg/standards/mpeg-2/mpeg-2.htm)*
- *Frecuently Askef Questions about MPEG Audio Layer-3 (www.iis.fhg.de/amm/techinf/layer3faq/index.html)*
- *MPEG Audio FAQ's Version 8 (drogo.cselt.stet.it/ufv/leonardo/mpeg/faq/faq-audio.htm)*
- *MPEG Audio Web Page (www.tnt.uni-hannover.de/project/mpeg/audio)*

Bibliografía y enlaces (2)

- **Introducción a la Codificación de Audio Digital (MPEG y DOLBY AC-3)**
  Francisco García López (www.timagazine.net; Diciembre 1997)
- *Grupos de noticias*
  - *rec.music.makers*
  - *rec.audio.pro*
  - *alt.music.midi*
  - *alt.binaries.sounds.midi*
  - *comp.music.midi*
- *Steve Rabin, 2010, Introduction to Game Development*
- *OpenAL* < *http://connect.creativelabs.com/openal/* >
- *Festival* <*http://www.cstr.ed.ac.uk/projects/festival/*>
- *CMU Sphinx* <*http://cmusphinx.sourceforge.net/*>

Bibliografía y enlaces (y 3)

- *M. Agustí. (2011). Introducción al procesado de audio mediante OpenAL. http://hdl.handle.net/10251/12694*
- *M. Agustí. (2011). Introducción al empleo de técnicas de audio posicional mediante OpenAL. http://hdl.handle.net/10251/12697*
- *M. Agustí. (2011). Efectos de audio básicos mediante OpenAL. http://hdl.handle.net/10251/12696.*
- *M. Agustí. (2012). Uso del micrófono para captura de audio en OpenAL. http://hdl.handle.net/10251/17547*
- *M. Agustí. (2018). Extensiones para OpenAL: efectos ambientales. http://hdl.handle.net/10251/105664*
- *M. Agustí. (2018). Extendiendo OpenAL con OGG Vorbis. http://hdl.handle.net/10251/109210*
- *M. Agustí. (2018). Extendiendo OpenAL con SDL. Caso de estudio MP3. http://hdl.handle.net/10251/105383*
- *M. Agustí. (2018). OpenAL: efecto Doppler. Posicionamiento y velocidad del sonido. http://hdl.handle.net/10251/104052*
- *M. Agustí. (2018). Reproducción de ficheros Opus con OpenAL: precarga vs "streaming". http://hdl.handle.net/10251/109211*
- *M. Agustí. (2018). OpenAL y OpenGL: escuchar y ver el sonido. http://hdl.handle.net/10251/105550*

- *Ivars Badía, A.; M. Agustí. (2011). Interacción con OpenCV: detección de movimiento para realizar un instrumento virtual con OpenCV + OpenAL. http://hdl.handle.net/10251/12684*