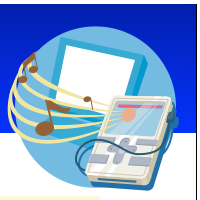
Playing Audio
OpenAL Overview
oalTouch Application

사운드 콘텐츠 응용 12주차

2010-2
멀티미디어과학과 이종우

Index

OpenAL(Open Audio Library)



OpenAL

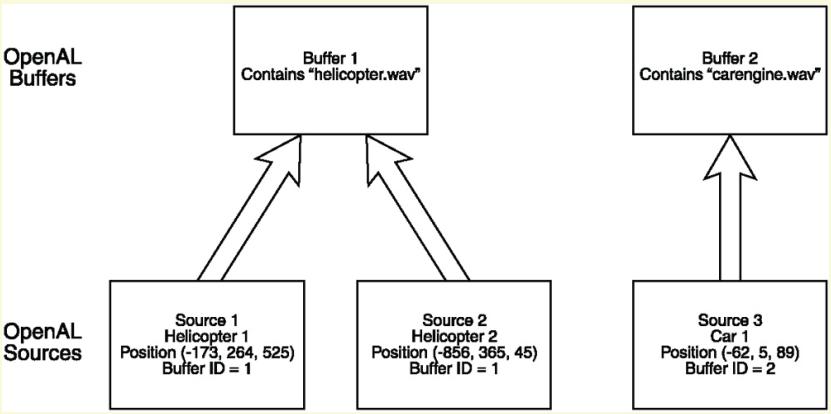
- is cross-platform, positional audio API included in iPhone OS.
- is the recommended technology for adding audio to games and audio features to many other types of applications.
- OpenAL framework in iPhone OS 2.1 implements the OpenAL 1.1 specification.
- is available in the OpenAL framework - OpenAL.framework
- on the iPhone/iPod *vs.* on the desktop
  - Roger Beep, Distortion, Reverb, Obstruction, Occlusion effects are not available in iPhone OS

OpenAL API의 특징

- Tried-and-true method of API
- API 함수 이름이 엄격한 규칙에 의해 정해져 있다.
  - 모든 함수 이름은 "al"로 시작
  - 끝에 f 또는 fv, i 가 붙어 있음: 인자 수형을 의미
    - alListenerf / alListener3f / alListenerfv / alListeneri

OpenAL에서의 객체

- **Source**와 **Buffer**, **Listener**로 구성됨
  - Listener: DirecXAudio와 같은 의미
    - 오직 하나의 listener만 있고, 이는 사용자(플레이어)를 의미
  - Buffer: the PCM sampled data itself
  - Source: 버퍼 인스턴스 의미 + 다른 버퍼가 파생될 수 있는 음원의 의미



다중 소스 버퍼의 장점

- 프로그램 내에서 **3**개의 다른 헬리콥터 소리가 필요할 때
  - 세 개의 Wav 파일이 다 없어도 흉내 낼 수 있다.
  - 버퍼는 2개만 생성하고, 파생된(3D 사운드 효과가 적용된) 소스를 여러 개 만들면 됨
- **OpenAL**의 **Buffer == DirectMusic**의 **Segment,**
  OpenAL의 **Source == audio path of that Segment**

OpenAL Buffer의 특성

- 버퍼 생성 함수: alGenBuffers.
- 생성된 버퍼의 특성 get 함수: alBuffer[f, i]와 alGetBuffer[f, i]
- 버퍼 특성:

Property	Data Type	Description
AL_FREQUENCY	i, iv	frequency of buffer in Hz
AL_BITS	i, iv	bit depth of buffer
AL_CHANNELS	i, iv	number of channels in buffer > 1 is valid, but buffer won't be positioned when played
AL_SIZE	i, iv	size of buffer in bytes
AL_DATA	i, iv	original location where data was copied from generally useless, as was probably freed after buffer creation

- 예:
```
// Retrieve Buffer Frequency
    alBufferi(g_Buffers[0], AL_FREQUENCY, iFreq);
```

OpenAL Source의 특성

- 소스(버퍼의 인스턴스) 생성 함수: alGenSources
- 소스의 특성을 get 함수: alSource[f, 3f, fv, i] and alGetSource[f, 3f, fv, i]

Property	Data Type	Description
AL_PITCH	f, fv	pitch multiplier always positive
AL_GAIN	f, fv	source gain value should be positive
AL_MAX_DISTANCE	f, fv, i, iv	used with the Inverse Clamped Distance Model to set the distance where there will no longer be any attenuation of the source
AL_ROLLOFF_FACTOR	f, fv, i, iv	the rolloff rate for the source default is 1.0
AL_REFERENCE_DISTANCE	f, fv, i, iv	the distance under which the volume for the source would normally drop by half (before being influenced by rolloff factor or AL_MAX_DISTANCE)
AL_MIN_GAIN	f, fv	the minimum gain for this source
AL_MAX_GAIN	f, fv	the maximum gain for this source
AL_CONE_OUTER_GAIN	f, fv	the gain when outside the oriented cone
AL_CONE_INNER_ANGLE	f, fv, i, iv	the gain when inside the oriented cone
AL_CONE_OUTER_ANGLE	f, fv, i, iv	outer angle of the sound cone, in degrees default is 360

OpenAL Source의 특성(계속)

- source properties:

AL_POSITION	fv, 3f	X, Y, Z position
AL_VELOCITY	fv, 3f	velocity vector
AL_DIRECTION	fv, 3f, iv, 3i	direction vector
AL_SOURCE_RELATIVE	i, iv	determines if the positions are relative to the listener default is AL_FALSE
AL_SOURCE_TYPE	i, iv	the soruce type – AL_UNDETERMINED, AL_STATIC, or AL_STREAMING
AL_LOOPING	i, iv	turns looping on (AL_TRUE) or off (AL_FALSE)
AL_BUFFER	i, iv	the ID of the attached buffer
AL_SOURCE_STATE	i, iv	the state of the source (AL_STOPPED, AL_PLAYING, …)
AL_BUFFERS_QUEUED*	i, iv	the number of buffers queued on this source
AL_BUFFERS_PROCESSED*	i, iv	the number of buffers in the queue that have been processed
AL_SEC_OFFSET	f, fv, i, iv	the playback position, expressed in seconds
AL_SAMPLE_OFFSET	f, fv, i, iv	the playback position, expressed in samples
AL_BYTE_OFFSET	f, fv, i, iv	the playback position, expressed in bytes

** Read Only (alGetSourcei)*

OpenAL Source의 특성(계속)

- source 특성 세팅 예:

```
alGetError(); // clear error state
alSourcef(source[0],AL_PITCH,1.0f);
if ((error = alGetError()) != AL_NO_ERROR)
    DisplayALError("alSourcef 0 AL_PITCH : \n", error);
alSourcef(source[0],AL_GAIN,1.0f);
if ((error = alGetError()) != AL_NO_ERROR)
    DisplayALError("alSourcef 0 AL_GAIN : \n", error);
alSourcefv(source[0],AL_POSITION,source0Pos);
if ((error = alGetError()) != AL_NO_ERROR)
    DisplayALError("alSourcefv 0 AL_POSITION : \n", error);
alSourcefv(source[0],AL_VELOCITY,source0Vel);
if ((error = alGetError()) != AL_NO_ERROR)
    DisplayALError("alSourcefv 0 AL_VELOCITY : \n", error);
alSourcei(source[0],AL_LOOPING,AL_FALSE);
if ((error = alGetError()) != AL_NO_ERROR)
    DisplayALError("alSourcei 0 AL_LOOPING true: \n", error);
```

OpenAL Listener의 특성

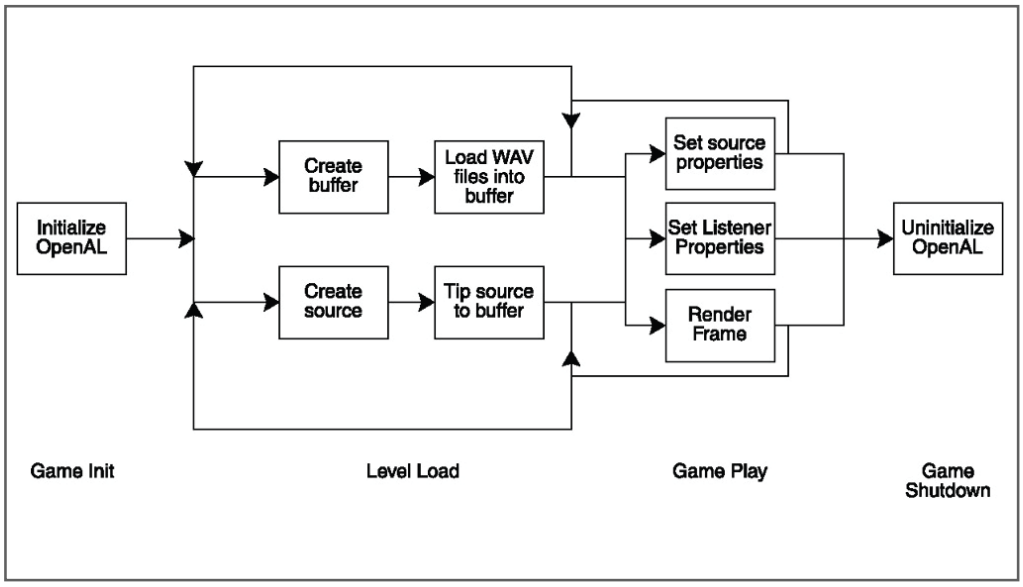
- OpenAL로 프로그래밍하려면 항상 최소 *1*개의 *Listener* 객체는 있어야 함.
- Listener 특성 set/get 함수: *alListener[f, 3f, fv, i]* and *alGetListener[f, 3f, fv, i] :*

Property	Data Type	Description
AL_GAIN	f, fv	"master gain" value should be positive
AL_POSITION	fv, 3f, iv, 3i	X, Y, Z position
AL_VELOCITY	fv, 3f, iv, 3i	velocity vector
AL_ORIENTATION	fv, iv	orientation expressed as "at" and "up" vectors

좌표계 시스템의 차이

- **OpenAL uses a left-handed coordinate system.**
  - Negative z values, in OpenAL, descend into the screen.
- DirectX uses a right-handed coordinate system.
  - Negative z values, in DirectX, go out of the screen.

OpenAL 사용을 위한 기본 흐름



OpenAL 기본 흐름

- **Initialization** (oalTouch내 oalPlayback.m)

  1단계 : opening a device

  2단계 : (1단계 성공시) a context is opened on that device

  이로써 listener, various sources, buffers를 다룰 수 있다.

```
// Create a new OpenAL Device
// Pass NULL to specify the system's default output device
device = alcOpenDevice(NULL);
if (device != NULL)
{
    // Create a new OpenAL Context
    // The new context will render to the OpenAL Device just created
    context = alcCreateContext(device, 0);

      ...
  }
```

OpenAL 기본 흐름

- **Buffer 생성**
  - 음원을 담을 버퍼 생성이 가장 먼저 해야 할 일! 에러 상태를 알아보려면 alGetError를 사용
  - 버퍼 수를 생성하기 위해 alGenBuffers를 호출

```
// Create some OpenAL Buffer Objects
alGenBuffers(1, &buffer);     // (NUM_BUFFERS, g_Buffers)
if((error = alGetError()) != AL_NO_ERROR) {
       NSLog(@"Error Generating Buffers: %x", error);
       exit(1);
}
```

- **Error Handling**
  - 방법 1: 함수의 리턴 값이 AL_FALSE 인지 검사하는 방법
  - 방법 2: 함수 호출 후 alGetError() 함수를 호출하여 에러 코드를 알아내는 방법

Error Code	Description
AL_NO_ERROR	there is not currently an error
AL_INVALID_NAME	a bad name (ID) was passed to an OpenAL function
AL_INVALID_ENUM	an invalid enum value was passed to an OpenAL function
AL_INVALID_VALUE	an invalid value was passed to an OpenAL function
AL_INVALID_OPERATION	the requested operation is not valid
AL_OUT_OF_MEMORY	the requested operation resulted in OpenAL running out of memory

OpenAL 기본 흐름

- 버퍼를 생성했으면 Source를 생성할 수 있다.

```
// Create some OpenAL Source Objects
alGenSources(1, &source);
if(alGetError() != AL_NO_ERROR) {
       NSLog(@"Error generating sources! %x\n", error);
       exit(1);
}
```

- 본 예제소스에서는
  초기화시 생성된 현재 context 좌표를 이용하여 사운드 처리
  (이는 다음 주차에서 설명할 예정)

OpenAL 기본 흐름

- Sources 재생

```
- (void)startSound
{
    ALenum error;

    NSLog(@"Start!\n");

    // Begin playing our source file
    alSourcePlay(source);

    if((error = alGetError()) != AL_NO_ERROR) {
         NSLog(@"error starting source: %x\n", error);
    } else {
         // Mark our state as playing (the view looks at this)
         self.isPlaying = YES;
    }
}
```

- Sources 정지

```
- (void)stopSound
{
    ALenum error;

    NSLog(@"Stop!!\n");

    // Stop playing our source file
    alSourceStop(source);
    if((error = alGetError()) != AL_NO_ERROR) {
         NSLog(@"error stopping source: %x\n", error);
    } else {
         // Mark our state as not playing (the view looks at this)
         self.isPlaying = NO;
    }
}
```

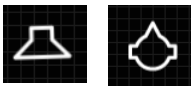
OpenAL 기본 흐름

- Exit (uninitializing OpenAL)

```
- (void)teardownOpenAL
{
    // Delete the Sources
    alDeleteSources(1, &source);
    // Delete the Buffers
    alDeleteBuffers(1, &buffer);

    //Release context
    alcDestroyContext(context);
    //Close device
    alcCloseDevice(device);
}
```

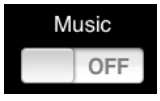
oalTouch Application



를 움직여 입체 음향 효과를 재생

이것이 멀어지거나 가까워지면
음악도 멀리 들리고 가까이 들리는 효과

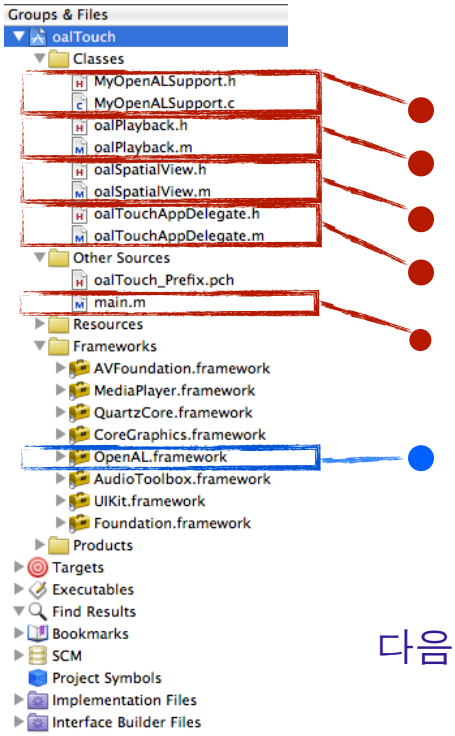
사운드 재생/정지 버튼



배경음 ON/OFF

중력가속센서 ON/OFF
(실제 기기에서 실행시 작용)

oalTouch Classes



MyOpenALSupport 클래스 : OpenAL 관련 task를 위한 클래스

oalPlayback 클래스 : OpenAL환경에서 audio playback 객체를 정의하기 위한 클래스

oalSpatialView 클래스 : 공간 view객체를 정의하기 위한 클래스

oalTouchAppDelegate 클래스 : 어플리케이션 델리게이트

main 클래스

OpenAL 프레임워크

다음 시간에는 전체적인 흐름과 각 클래스 설명!