게임프로그래밍

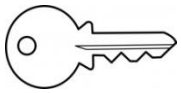
XAudio2

박종승

Dept. of CSE, Incheon Nat. Univ.
jong@inu.ac.kr
http://ecl.inu.ac.kr

목차

- XAudio2 소개
- 사전 지식 준비
- XAudio2 시작하기



XAudio2 소개

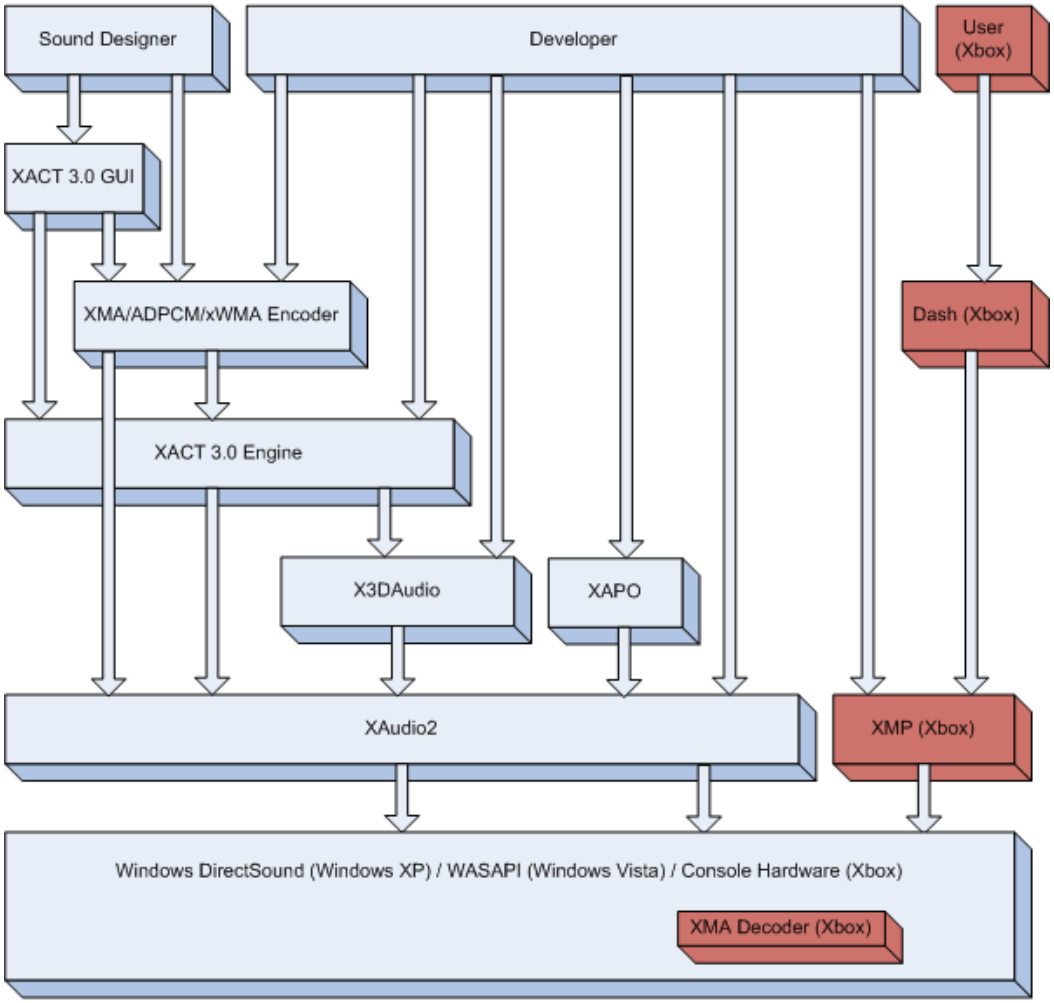
- 가장 기본적인 API
  - DirectSound (Windows XP)
  - WASAPI(Windows Audio Session API) (Windows Vista/7+)
- XAudio2
  - DirectSound를 대체할 다음 단계 API
  - 지원하는 플랫폼: Xbox 360(XAudio API 사용), Windows XP-(DirectSound 사용), Windows Vista/7+(WASAPI를 사용)
- XAudio2를 사용해야 하는 경우
  - 게임을 위한 고성능 오디오 엔진을 개발하는 경우
    - 게임에서 사운드효과 및 배경음악을 처리
  - 게임 개발에서는 XAudio2를 사용하는 것을 권장
    - WASAPI를 직접 사용하는 것보다 : 코드 길이가 매우 짧아짐
    - Media Foundation(DirectShow의 다음 단계 API)을 사용하는 것보다 : 더 저수준이며 지연이 적음
      - 단순한 음악 재생이라면 Media Foundation이 더 적합할 수 있음

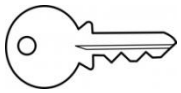
DirectX 오디오 API들

- DirectX 오디오 API들
  - 멀티 플랫폼: Xbox 360과 Windows
  - 여러 API들을 동시에 혼합하여 사용 가능
- API 종류
  - XACT : 컨텐츠 중심의 API
    - 디자인 시에 오디오 제어가 가능하도록 함
    - XACT GUI: XACT API에서 사용하기 위한 오디오 컨텐츠를 패키징하는 도구
  - XAudio2 : 프로그래밍 중심의 API
    - 개발자가 자체 오디오 엔진을 구축하도록 지원
  - XAPO API
    - XAudio2에서 사용되기 위한 오디오 효과를 생성하는 도구임
  - X3DAudio API
    - 3차원 공간에서의 입체 음향

DirectX 오디오 API들'

- DirectX 오디오 API들 관계도





사전 지식 준비: 3D공간의 좌표

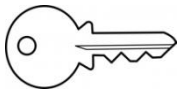
- 3D공간의 좌표
  - 사운드 소스 및 청자의 3D 공간에서의 위치, 속도, 방향 등의 표현 : (x,y,z)
    - X축은 왼쪽에서 오른쪽으로
    - Y축은 아래에서 위로
    - Z축은 가까이서 멀리로
  - 구조체: X3DAUDIO_VECTOR
    - 월드공간에서의 좌표 단위로 표현
    - 속도의 경우 초당 이동 거리

소리 위치의 인식

- 소리 위치의 인식의 단서
  - 소리의 크기(loudness)
    - 음원(sound source)이 청자로부터 멀어지는 경우, 소리 볼륨이 고정비율로 감소함
  - 두 귀 사이의 강도 차(interaural intensity difference)
    - 청자의 오른쪽에서 오는 소리는 오른쪽 귀에서 더 크게 들림
  - 두 귀의 시간 차(interaural time difference)
    - 음원에서 청자의 오른쪽으로 향하는 소리는 오른쪽 귀에 더 빨리 도달함
      - 두 귀의 시간 차이는 대략 1/1000 초임
  - 귀의 생김새 및 머리로 인해서 소리가 약하게 되는 현상
    - 뒤에서 오는 소리는 (앞에서 오는 소리에 비해서) 소리가 약하게(muffling) 됨.
      - 귀가 앞으로 향하도록 생겨서
    - 오른쪽에서 오는 소리는 왼쪽 귀에 들어 올 때에 소리가 약하게 됨.
      - 왼쪽 귀의 방향이 왼쪽으로 치우쳐 있고 또한 머리가 가로막고 있어서
  - 귀의 모양새로 인한 변형
    - 소리의 도착하는 방향에 따라서 귓바퀴가 소리의 음조(pitch)와 타이밍을 약간 변화시킴.

사운드 원뿔

- 사운드: 방향성(orientation)을 가질 수 있음
  - 방향성이 없는 경우
    - 소리의 크기는 음원과의 거리와만 관련되고 방향과는 무관함
  - 방향성이 있는 경우
    - 해당 방향에서 소리의 크기가 가장 큼
      - 사운드 원뿔은 방향성 사운드에 대해서 소리의 크기를 명시하는 방법임
- 사운드 원뿔
  - 내부 원뿔과 외부 원뿔로 정의함
  - 내부 원뿔 내의 모든 각도에 대해서
    - 사운드 볼륨은 내부원뿔 볼륨으로 지정됨
    - 버퍼의 기본 볼륨과 청자와의 거리를 고려하여 최종 볼륨을 결정함
  - 외부 원뿔 외부의 각도에 대해서
    - 응용이 명시한 인자에 의해서, 선형으로 진폭을 스케일링함
      - 0은 묵음, 1은 감쇄 없음, 0.5는 6 dB만큼 감쇄
      - 1 이상도 허용: 2까지만 허용
  - 내부 원뿔과 외부 원뿔 사이의 각도에 대해서
    - 내부 원뿔 볼륨에서 외부 원뿔 볼륨으로 전이하는 구간



XAudio2 시작하기: XAudio2의 핵심 개념

- XAudio2 엔진
  - IXAudio2 인터페이스 : XAudio2 엔진의 코어
    - 지원하는 기능: 가용한 오디오 장치들의 열거, 전역 API 속성값들의 설정, 보이스의 생성, 성능을 감시
  - XAudio2의 생성과 초기화: 도움함수 XAudio2Create 를 호출
    - 단일 프로세스에서 여러 번 생성할 수 있음
    - 각 객체는 독립적으로 운영됨
- 필요한 것
  - 헤더파일: XAudio2.h
  - 라이버러리: XAudio2.lib
    - 현재 버전: lib 파일 없음... dll 파일이 자동 로드됨

XAudio2의 핵심 개념'

- 보이스(voice)
  - XAudio2가 오디오 데이터의 처리나 재생을 위해 다루는 객체임
- 보이스의 3가지 타입
  - 소스 보이스(source voice): 오디오 데이터의 스트림을 표현함. 소스 보이스는 자신의 데이터를 다른 타입의 보이스로 보냄.
  - 부믹스 보이스(submix voice): 수신하는 오디오 데이터에 어떤 처리(예: 샘플율 변환)를 수행함. 처리된 데이터는 다른 부믹스 보이스나 마스터링 보이스로 보내짐.
  - 마스터링 보이스(mastering voice): 소스 보이스 및 부믹스 보이스로부터 데이터를 받아서 이를 오디오 하드웨어로 보냄.

XAudio2의 핵심 개념”

- 오디오 그래프(audio graph)
  - XAudio2 보이스들의 묶음
  - 오디오는 소스 보이스들에서 시작하여, 선택적으로 부믹스 보이스들을 거쳐서, 한 마스터링 보이스에서 끝남
  - 한 오디오 그래프는, 재생되는 각 사운드에 대해서,
    - 1개의 소스 보이스, 0+개의 부믹스 보이스, 1개의 마스터링 보이스를 가짐
- 콜백(callback)
  - 오디오 재생은 비동기방식임
  - 보이스나 엔진 객체에서 어떤 이벤트가 발생하였음을 XAudio2가 클라이언트 코드에 알리는 유일한 방법임
    - 예: 사운드의 재생이 종료된 이벤트를 알림

XAudio2의 버전

- XAudio2
  - Cross-platform API
  - Xbox 360, Windows XP, Windows Vista/7/8
    - Xbox 360에서는 정적 library 형태, Windows에서는 DLL 형태
- 윈도우에서의 XAudio2의 현재 버전
  - 최초 버전: 2008년 3월 XAudio2 2.0 배포
  - 최근 버전: 2010년 6월 XAudio2 2.7 배포
  - Windows 8에 시스템 컴포넌트로 XAudio2 버전 2.8이 배포됨
    - XAUDIO2_8.DLL
    - 개발자들은 Windows SDK로 XAudio2를 사용하면 됨
    - 개발자가 별도로 XAudio2를 배포할 필요 없음

XAudio2의 초기화

- 단계1: XAudio2 엔진의 인스턴스 생성
  - 방법: XAudio2Create 함수를 호출

```
IXAudio2* pXAudio2 = NULL;
XAudio2Create( &pXAudio2, 0, XAUDIO2_DEFAULT_PROCESSOR );
```

- 단계2: 마스터링 보이스를 생성
  - 방법: CreateMasteringVoice 함수를 호출
  - 마스터링 보이스는 오디오 장치를 캡슐화함
    - 오디오 그래프를 통해서 통과해오는 오디오들의 최종 목적지가 됨

```
IXAudio2MasteringVoice* pMasterVoice = NULL;
pXAudio2->CreateMasteringVoice( &pMasterVoice );
```

참고: Resource Interchange File Format (RIFF)

- Resource Interchange File Format (RIFF)
  - ".wav" 파일에서 사용되는 포맷
  - XAudio2가 다루는 전형적인 포맷임
  - 한 RIFF 파일은 "chunk"들의 나열로 구성됨
- FOURCC identifier
  - four-character code (FOURCC) identifier
    - 한 chunk에서의 데이터의 타입을 명시함
    - 예: 'RIFF', 'fmt ', 'data'
    - 한 FOURCC는 한 32-bit unsigned integer로 내부적으로 처리함
      - four ASCII character들을 붙여서 하나의 정수를 만듦
      - 예: FOURCC 'abcd'는 0x64636261임 (little-endian 시스템의 경우)
  - 'RIFF': 표준 RIFF 파일임을 명시함.
    - 하부에 'WAVE' 또는 'XWMA' 포맷의 데이터가 있음.
  - 'fmt ': 포맷 헤더.
    - 이 chunk는 다음 구조체중 하나에 해당함:
      - WAVEFORMATEX, WAVEFORMATEXTENSIBLE, ADPCMWAVEFORMAT.
  - 'data': 오디오 데이터.
    - XAudio2가 버퍼로 읽은 다음에 XAUDIO2_BUFFER 구조체의 pAudioData 요소로 명시하여 소스 보이스로 패스함

오디오 데이터 파일 로드하기

- 오디오 파일 parse를 위한 준비
  - XAudio2가 지원하는 오디오 파일은 RIFF임.
    - 파일에서 RIFF chunk를 찾고 그 하부 chunk들을 로드해야 함
- 찾은 RIFF chunk로부터 XAudio2 구조체를 준비
  - XAudio2가 오디오를 재생하려면 다음 두 구조체가 필요함
    - WAVEFORMATEX 구조체
      - 포맷 정보를 가지는 구조체
      - 확장된 구조체일 수 있음
        - » 예: 다채널을 지원하는 확장 구조체: WAVEFORMATEXTENSIBLE
    - XAUDIO2_BUFFER 구조체
      - 데이터 버퍼를 가지는 구조체
- 자세한 구현관련 내용
  - 샘플 코드: "SDKwavefile.cpp/h"의 "CWaveFile" 클래스 참조

오디오 데이터 파일 로드하기'

- WAV 파일을 읽어들이기
  - "CWaveFile" 클래스를 이용

```
CWaveFile wav;
wav.Open( strFilePath, NULL, WAVEFILE_READ );

WAVEFORMATEX* pwfx = wav.GetFormat(); //WAV파일의 포맷을 얻기
DWORD cbWaveSize = wav.GetSize(); //WAV파일 내의 크기를 계산하기

BYTE* pbWaveData = new BYTE[ cbWaveSize ];
wav.Read( pbWaveData, cbWaveSize, &cbWaveSize ); //데이터를 메모리로 읽어오기
```

사운드 재생하기

- XAudio2 로 사운드 재생 절차
  - 단계 1: XAudio2의 초기화
  - 단계 2: 오디오 데이터의 로드
  - 단계 3: 소스 보이스 생성
    - XAudio2 엔진 객체의 함수 IXAudio2::CreateSourceVoice 호출

```
IXAudio2SourceVoice* pSourceVoice;
pXaudio2->CreateSourceVoice( &pSourceVoice, pwfx )
```

  - 단계 4: XAUDIO2_BUFFER를 소스 보이스로 제출
    - 함수 SubmitSourceBuffer 호출

```
XAUDIO2_BUFFER buffer = {0};
buffer.pAudioData = pbWaveData;
buffer.Flags = XAUDIO2_END_OF_STREAM; //이 버퍼 이후에 추가데이터 없음을 알림
buffer.AudioBytes = cbWaveSize;
pSourceVoice->SubmitSourceBuffer( &buffer );
```

  - 단계 5: 소스 보이스의 재생 시작
    - 함수 Start 호출
      - 디폴트로 모든 XAudio2 보이스들은 출력을 마스터링 보이스로 보냄

```
pSourceVoice->Start( 0 );
```

사운드 재생하기'

- XAudio2 로 사운드 재생 절차
  - 단계 6: 종료하기
    - IXAudio2SourceVoice::DestroyVoice 함수 호출

```
pSourceVoice->DestroyVoice();
```

  - 기타 IXAudio2SourceVoice 함수들 (link)
    - 일시정지와 재개: Start, Stop
    - 이 보이스의 볼륨 조정: SetVolume
    - 이 보이스의 현재 상태: GetState

```
BOOL isRunning = TRUE;
while( isRunning ) {
        XAUDIO2_VOICE_STATE state;
        pSourceVoice->GetState( &state );
        isRunning = ( state.BuffersQueued > 0 ) != 0;
        if ( GetAsyncKeyState( VK_ESCAPE ) ) break; //ESC가 눌러지면 탈출
        Sleep( 10 );
    }
//눌러진 ESC가 원위치될 때까지 기다림
while( GetAsyncKeyState( VK_ESCAPE ) ) Sleep( 10 );

pSourceVoice->DestroyVoice(); //보이스 종료
SAFE_DELETE_ARRAY( pbWaveData ); //자원 반납
```

예제

01.XAudio2BasicSound

- CWaveFile
  - WAV 파일의 데이터 입출력
- 메인
  - XAudio2로 사운드 재생 절차 예시

참고 예제

- 예제1
  - 스트리밍
    - XACT 사용
- 예제2
  - 3D사운드
    - X3DAudio 사용

21.XAudio2BasicStream

22.XAudio2Sound3D

참고: XAudio2 추가 학습

- XAudio2 Voices (link)
- XAudio2 Callbacks (link)
- XAudio2 Audio Graph (link)
- XAudio2 Audio Effects (link)
- XAudio2 Streaming Audio Data (link)
- X3DAudio Overview (link)
- XAudio2 Operation Sets (link)

- 기타 XAudio2 Tutorials (1-8) (link)