게임프로그래밍

DirectSound

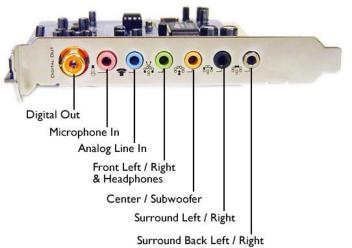
박종승

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

목차

- DirectSound 개요
- DirectSound 장치 생성
- 협력레벨 지정
- 주버퍼 포맷 지정
- 이차버퍼의 생성
- 이차버퍼에 데이터 채우기
- 이차버퍼의 사운드 재생과 정지





DirectSound 개요

- 단순한 사운드 재생
 - 윈도우 멀티미디어 라이버러리(windows multimedia library)
 - 헤더파일: windows.h (반드시 먼저), mmsystem.h
 - 라이버러리 파일: winmm.lib
 - 함수: PlaySound(), sndPlaySound()
 - sndPlaySound 함수는 PlaySound 함수의 단순화된 형태
 - 마지막 인자
 - 흔히 SND ASYNC로 지정함: 재생을 시작시키고 즉시 리턴하도록 함
 - SND_LOOP는 계속 반복해서 재생되도록 지정함
 - 현재 재생되는 사운드를 중지시키려면 첫 번째 인자를 NULL로 하여 함수를 다시 호출하면 됨
 - 사용 예:

```
sndPlaySound("mysound.wav", SND_ASYNC);
PlaySound("mysound.wav", NULL, SND_ASYNC);
```

sndPlaySound("mysound.wav", SND_ASYNC| SND_LOOP); //비동기&반복 sndPlaySound(NULL, SND_ASYNC); // 플래이를 중지

DirectSound 개요

- DirectSound API
 - DirectSound vs. Win32 API
 - Win32에 비해 속도가 빠름, 풍부한 사운드 플레이 기능 제공
 - 사운드 카드의 제어 기능
 - 윈도우 상에서 작동되는 Sound 카드 지원
 - PCM 형태의 디지털 사운드 출력 (WAV 파일)
 - 여러 개의 사운드를 동시에 재생

DirectSound API

- DirectSound API
 - 헤더파일
 - dsound.h
 - _ 링크파일
 - dsound.lib, dxguid.lib
 - DLL: dsound.dll

DirectSound 인터페이스

• 코어

- IDirectSound8 사운드카드의 기능을 결정하고 재생을 위한 버퍼 생성
- IDirectSoundBuffer8 재생 가능한 사운드 데이터를 담기 위한 데이터 버퍼

• 3D

- IDirectSound3DBuffer8 3D 사운드를 담기 위한 버퍼
- IDirectSound3DListener8 3D Listener를 표현하는 사용되는 객체

캡처

- IDirectSoundCapture8 capture buffer 생성에 사용되는 인터페이스
- IDirectSoundCaptureBuffer8 마이크와 같은 장치에서 녹음한 데이터
 를 저장하기 위한 버퍼

기타

- IDirectSoundNotufy8 사운드 재생 종료를 알리는 객체
- IKsPropertySet8 사운드카드 제조업체에서 새로운 기능을 넣기 위해 사용하는 인터페이스

사운드 재생 절차

- 사운드 재생 절차
 - 단계 1: DirectSound 장치에 대한 인터페이스 객체를 생성
 - 한 응용프로그램에 대해서 하나의 객체만 생성
 - 단계 2: 각 사운드에 대해서 하나의 이차버퍼(secondary buffer)를 생성
 - 단계 3: WAV 파일로부터 사운드 데이터(PCM 데이터)를 읽어서 이차버 퍼에 넣음
 - 단계 4: 사운드를 재생하거나 재생 종료하는 기능을 수행

A sec

Primary buffer (SRam)

Software or hardware miker

Output

Speaker

장치 객체 생성

- DirectSound 장치 객체 생성
 - DirectSound 장치 객체: IDirectSound8
 - 장치 객체 생성: DirectSoundCreate8()

- 하드웨어 사양의 점검
 - IDirectSound8::GetCaps()

if (hr!= DS OK) { /* error */ }

협력 수준 설정

- 협력 수준 설정
 - 여러 응용프로그램이 동시에 한 장치를 공유하는 경우
 - 충돌 문제 발생 가능
 - 반드시 지정해야 함
 - IDirectSound8::SetCooperativeLevel(hwnd, dwLevel)
 - return HRESULT
 - HWND hwnd /*기본 윈도우 핸들*/
 - DWORD dwLevel /*협력 수준 flag*/
 - 협력 수준 flag
 - DSSCL_EXCLUSIVE : 사운드장치의 독점, 백그라운드에서는 소리가 출력되지 않음 (더 이상 지원하지 않음)
 - DSSCL NORMAL : 다른 응용 프로그램과 최적으로 협력
 - DSSCL_PRIORITY: 1차 포맷이 변경 가능, DSSCL_NORMAL과 유사
 - DSSCL_WRITEPRIMARY : 가장 높은 우선 순위, 2차 버퍼는 사용 불가능
 - 협력 수준 설정 예제

```
HRESULT hr = IpDirectSound->SetCooperativeLevel(hWnd,DSSCL_PRIORITY);
if (hr != DS_OK) { /* error */ }
```

주버퍼 포맷 지정

- 주버퍼 포맷 지정
 - 선택 사항임
 - 주버퍼의 포인터 얻기
 - IDirectSound8::CreateSoundBuffer()
 - 인자 구조체 DSBUFFERDESC의 필드 지정
 - » dwFlags = DSBCAPS_PRIMARYBUFFER, lpwfxFormat = NULL
 - 주버퍼의 포맷 지정
 - IDirectSoundBuffer8::SetFormat()

버퍼의 생성

• 버퍼 생성

```
HRESULT IDirectSound8::CreateSoundBuffer(
LPCDSBUFFERDESC pcDSBufferDesc, //DSBUFFERDESC 구조체의 포인터
LPDIRECTSOUNDBUFFER * ppDSBuffer, //DirectSoundBuffer의 포인터
LPUNKNOWN pUnkOuter //사용하지 않음; NULL로 지정
);
```

CreateSoundBuffer : 예제

```
LPDIRECTSOUND8 lpDSound;
HRESULT hr;
// 1차 사운드 장치를 이용해 IDirectSound 객체 생성
...
LPDIRECTSOUNDBUFFER pDSBPrimary = NULL;
ZeroMemory(&dsbd, sizeof(DSBUFFERDESC));
dsbd.dwFlags = DSBCAPS_PRIMARYBUFFER
dsbd.dwBufferByters = 0;
dsbd.lpwfxformat = NULL;
hr = lpDSound->CreateSoundBuffer(&dsbd, &pDSBPrimary, NULL);
if (hr != DS_OK) { /* error */ }
```

DirectSound 버퍼

- 1차 버퍼와 여러 개의 2차 버퍼를 혼합하여 재생
 - 1차 버퍼: 현재 카드에서 재생되고 있는 사운드
 - 2차 버퍼: 사운드 효과음
- Sound 버퍼의 구조체 : DSBUFFERDESC

```
typedef struct {
 DWORD dwSize; //구조체의 크기: sizeof(DSBUFFERDESC)
 DWORD dwFlags;
 DWORD dwBufferBytes; //버퍼의 크기; 1차버퍼를 생성하는 경우 0으로 설정
 DWORD dwReserved; //0으로 설정
 LPWAVEFORMATEX lpwfxFormat; //1차 버퍼에 대해서는 0으로 설정
 GUID guid3DAlgorithm; //2-way 스피커에서의 가상 3D사운드 출력에 관련
} DSBUFFERDESC, *LPDSBUFFERDESC;
dwFlags
DSBCAPS_CTRL3D: 3D 제어기능, 1차 또는 2차 버퍼일 수 있음
DSBCAPS_CTRLFREQUENCY: 주파수 제어 기능
DSBCAPS_CTRLFX: 효과 처리 지원 기능
```

판별, 스트리밍 버퍼에 사용 DSBCAPS PRIMARYBUFFER: 1차버퍼 생성시 사용, DBSCAPS CTRLFX와 같이 사용불가

SetFormat

- 주버퍼의 포맷 지정
 - IDirectSoundBuffer8::SetFormat 함수를 호출
- 예제

```
WAVEFORMATEX wfx;
ZeroMemory(&wfx, sizeof(WAVEFORMATEX));
wfx.wFormatTag = (WORD)WAVE_FORMAT_PCM;
wfx.nChannels = (WORD)2;
wfx.nSamplesPerSec = (DWORD)22050;
wfx.wBitsPerSample = (WORD)16;
wfx.nBlockAlign = (WORD)(wfx.wBitsPerSample / 8 * wfx.nChannels);
wfx.nAvgBytesPerSec = (DWORD)(wfx.nSamplesPerSec * wfx.nBlockAlign);
pDSBPrimary->SetFormat(&wfx);
```

WAVEFORMATEX

WAVEFORMATEX

- 파형 오디오 데이터(waveform-audio data)의 포멧을 명세함

```
typedef struct {
 WORD wFormatTag; //wave형식의 오디오 형식;
        // 비압축 데이터(1-,2-채널 PCM 데이터)의 경우는 WAVE FORMAT PCM
 WORD nChannels; //사운드 데이터를 위한 오디오 채널의 수
        // mono는 1, stereo는 2
 DWORD nSamplesPerSec; //초당 샘플링율(Hz)
        // 주로 8.0 kHz, 11.025 kHz, 22.05 kHz, and 44.1 kHz 이 사용됨.
        // 22.05kHz이면 22050을 명시.
 DWORD nAvgBytesPerSec; //데이터 처리 속도
        //PCM의 경우, nSamplesPerSec*nBlockAlign 로 지정
 WORD nBlockAlign; //블록 정렬 최소 단위 (byte 단위). 데이터 처리 및 전송의 단위가 됨.
        //PCM의 경우, nChannels*wBitsPerSample/8 로 지정.
 WORD wBitsPerSample; //wFormatTag 형식을 위한 샘플 당 비트 수.
        //PCM의 경우, 8 또는 16임.
 WORD cbSize; //이 구조체에 뒤에 붙여지는 추가 포맷정보의 크기(byte단위)
        //PCM의 경우, 이 값은 사용되지 않음.
} WAVEFORMATEX;
```

이차버퍼의 생성

- 이차버퍼의 기능
 - 주버퍼: 믹싱 엔진에 해당하며 주버퍼의 데이터가 재생 하드웨어로 전 달됨
 - 이차버퍼: 재생할 사운드 데이터를 가지고 있는 소스임
 - 언제든지 생성할 수 있고 소멸시킬 수 있음
 - 구현 방법: 정적 버퍼 방식, 스트리밍 버퍼 방식
 - 여러 이차버퍼를 동시에 재생하면 자동으로 믹스됨
- 이차버퍼의 생성
 - PCM 데이터 포맷만 완전하게 재생할 수 있음 (WAV 파일)
 - 이차버퍼 객체 IDirectSoundBuffer8의 생성
 - CWaveFile 클래스
 - WAV 파일로부터 데이터를 읽는 기능을 제공함
 - IDirectSound8::CreateSoundBuffer()
- 이차버퍼에 데이터 채우기
 - IDirectSoundBuffer8::Lock()
 - CSound 클래스 내에서 구현
 - 파일로부터 데이터를 읽어서 이차버퍼에 넣음

데이터를 이차버퍼에 저장

Lock 함수

• Unlock 함수

데이터를 이차버퍼에 저장'

• 예제

```
LPDIRECTSOUNDBUFFER8 lpBufer;
int bufferSize; // Lock의 크기
void* pbData = NULL; // lock의 시작주소
void* pbData2 = NULL; // 순환 후 시작주소
ulong dwLength; // lock된 바이트수
ulong dwLength2; // 순환 후 바이트수
HRESULT hr = lpBufer->Lock(0, bufferSize, &pbData, &dwLength, &pbData2, &dwLength2, DSBLOCK_ENTIREBUFFER);
if (hr != DS_OK) { /* error */ }
...
// 데이터를 읽어 저장한다.
...
// lock을 풀어준다
HRESULT hr = lpBufer->Unlock(pbData, bufferSize, NULL, 0);
if (hr != DS_OK) { /* error */ }
```

이차버퍼의 사운드 재생과 정지

- 재생과 정지 관련 함수
 - IDirectSoundBuffer8::Play()

```
HRESULT IDirectSoundBuffer8::Play(
DWORD dwReserved1, //예약값; 0으로 설정
DWORD dwPriority, //사운드 우선 순위 //DBSCAPS_LOCDEFER로 설정되지 않았으면 0
DWORD dwFlags //play 방법 플래그; DSBPLAY_LOOPING : 계속 재생
);
```

IDirectSoundBuffer8::Stop()
 HRESULT IDirectSoundBuffer8::Stop();

- IDirectSoundBuffer8::SetCurrentPosition()
- 예제

```
HRESULT hr = lpBufer->Play(0, 0, DSBPLAY_LOOPING); // 계속 재생 if (hr != DS_OK) { /* error */ } ....
HRESULT hr = lpBufer->Stop(); // 재생 중지 if (hr != DS_OK) { /* error */ }
```

예제

- 네 사운드 클립을 재생
 - 각 클립을 한번만 재생
 - 각 클립을 반복해서 재생
 - 각 클립의 재생 중지

01.DirectSoundPlayWav