Game Programming

DirectSound, DirectMusic

HyoungSeok Kim

DirectSound

- 윈도우 상에서 작동되는 Sound 카드 지원
- PCM 형태의 디지털 사운드 출력(WAV 파일)
- 2D 사운드와 3D 사운드로 분류
- 3D 사운드 emulating
- 2 스피커, 4 스피커 등의 옵션 지정 가능
- 파일 입출력을 지원하지 않음

- DirectSound로 할 수 있는 일
  - WAV 형식의 파일을 재생
  - 여러 개의 사운드를 동시에 재생
  - 하드웨어 버퍼에 고음질의 사운드 저장
  - 3D환경에서의 사운드 재생
  - 에코우, 코러스 등의 효과
  - 마이크나 다른 입력 장치로 부터 사운드 녹음

- DirectSound vs. Win32 API

DirectSound	Win32 API
풍부한 사운드 플레이 기능	제한된 기능
게임에서 사용	게임에 사용하기에 부적합
믹싱, 시그널 프로세싱 가능	믹싱, 시그널 프로세싱 불가능
속도가 빠르다	속도가 느리다

- 개발환경
  - Header File
    - dsound.h
  - Library File
    - dsound.lib
  - DLL
    - dsound.dll

DirectSound 인터페이스

- IDirectSound8
  - 사운드카드의 기능을 결정하고 재생을 위한 버퍼 생성에 사용
- IDirectSoundBuffer8
  - 재생가능한 사운드 data를 담기위한 data버퍼
- IDirectSound3DBuffer8
  - 3D 사운드를 담기위한 버퍼
- IDirectSound3DListener8
  - 3D Listner를 표현하는 사용되는 객체

DirectSound 인터페이스 (계속)

- IDirectSoundCapture8
  - 캡쳐버퍼 생성에 사용되는 인터페이스
- IDirectSoundCaptureBuffer8
  - 마이크와 같은 장치에서 녹음한 data를 저장하기 위한 버퍼
- IDirectSoundNotufy8
  - 사운드 재생 종료를 알리는 객체
- IKsPropertySet8
  - 사운드카드 제조업체에서 새로운 기능을 넣기 위해 사용하는 인터페이스

DirectSound 버퍼

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

DSBUFFERDES

```
typedef struct {
        DWORD dwSize;
        DWORD dwFlags;
        DWORD dwBufferBytes;
        DWORD dwReserved;
        LPWAVEFORMATEX lpwfxFormat;
        GUID guid3DAlgorithm;
} DSBUFFERDESC, *LPDSBUFFERDESC;
```

DSBUFFERDESC : 변수

- **DWORD** *dwSize*
  - 구조체의 크기
  - sizeof(DSBUFFERDESC)
- **DWORD** *dwFlags*
  - 뒤에 설명
- **DWORD** *dwBufferBytes*
  - 버퍼의 크기
  - 1차버퍼를 생성하는 경우 0으로 설정
- **DWORD** *dwReserved*
  - 0으로 설정
- **LPWAVEFORMATEX** *lpwfxFormat*
  - WAVEFORMATEX 구조체에 대한 포인터
  - 1차 버퍼에 대해서는 0으로 설정
- **GUID** *guid3DAlgorithm*
  - 2-way 스피커에서의 가상 3D사운드 출력에 관련

DSBUFFERDESC : dwFlags

- dwFlags
  - DSBCAPS_CTRL3D : 3D 제어기능, 1차 또는 2차 버퍼일 수 있음
  - DSBCAPS_CTRLFREQUENCY : 주파수 제어 기능
  - DSBCAPS_CTRLFX : 효과 처리 지원 기능
  - DSBCAPS_CTRLPAN : pan 제어 기능
  - DSBCAPS_CTRLPOSITIONNOTIFY : 위치 알림 기능
  - DSBCAPS_CTRLVOLUME : 볼륨 제어 기능
  - DSBCAPS_GETCURRENTPOSITION2 : 현재 위치 판별, 스트리밍 버퍼에 사용
  - DSBCAPS_GLOBALFOCUS : Global 사운드 버퍼
  - DSBCAPS_LOCDEFER : 음성 관리 기능을 사용하려면 설정
  - DSBCAPS_LOCHARDWARE : 버퍼의 믹싱을 하드웨어에서 함
  - DSBCAPS_LOCSOFTWARE : 버퍼의 믹싱을 소프트웨어에서 함
  - DSBCAPS_MUTE3DATMAXDISTANCE : 3D버퍼의 효율을 높임
  - DSBCAPS_PRIMARYBUFFER : 1차버퍼 생성시에 사용, DBSCAPS_CTRLFX와 같이 사용할 수 없음

WAVEFORMATEX

```
typedef struct {
        WORD wFormatTag;
        WORD nChannels;
        DWORD nSamplesPerSec;
        DWORD nAvgBytesPerSec;
        WORD nBlockAlign;
        WORD wBitsPerSample;
        WORD cbSize;
} WAVEFORMATEX;
```

WAVEFORMATEX : 변수

- **WORD** *wFormatTag*
  - wave형식의 오디오 형식
  - 비압축 데이터의 경우는 WAVE_FORMAT_PCM
- **WORD** *nChannels*
  - 사운드 데이터를 위한 오디오 채널의 수
- **DWORD** *nSamplesPerSec*
  - 초당 샘플링 속도
- **DWORD** *nAvgBytesPerSec*
  - 데이터 처리 속도
- **WORD** *nBlockAlign*
  - byte 단위의 블록 정렬
- **WORD** *wBitsPerSample*
  - wFormatTag 형삭을 위한 샘플 당 비트 수
- **WORD** *cbSize*
  - 구조체에 [illegible]

DirectSound 실행과정

1. 1차 버퍼
   - DirectSound 객체 생성
   - 협력 수준 설정
2. 2차 버퍼
   - WAVEFORMATEX, DSBUFFERDESC 등 초기화
   - 2차 버퍼 생성
   - data를 버퍼에 저장
   - 버퍼를 재생 및 중지

1차 버퍼 : DirectSound 객체생성

```
HRESULT WINAPI DirectSoundCreate8(
        LPCGUID lpcGuidDevice,
        LPDIRECTSOUND8 * ppDS8,
        LPUNKNOWN pUnkOuter
);
```

DirectSoundCreate8 : 인자

- **LPCGUID** *lpcGuidDevice*
  - 사운드 장치 GUID의 포인터
  - DirectSoundEnumerate를 이용하여 얻을 수 있음
  - NULL 또는
  - DSDEVID_DefaultPlayback,DSDEVID_DefaultVoicePlayback
- **LPDIRECTSOUND8 *** *ppDS8*
  - IDirectSound8 인터페이스 포인터
- **LPUNKNOWN** *pUnkOuter*
  - 사용하지 않음
  - NULL 지정

DirectSoundCreate8 : 결과

- 성공
  - DS_OK
- 에러
  - DSERR_ALLOCATED
  - DSERR_INVALIDPARAM
  - DSERR_NOAGGREGATION
  - DSERR_NODRIVER
  - DSERR_OUTOFMEMORY

DirectSoundCreate8 : 예제

```
LPDIRECTSOUND8 lpDirectSound = 0;
HRESULT hr;

// 1차 사운드 장치를 이용해 IDirectSound 생성
hr = DirectSoundCreate8(NULL, &lpDirectSound, NULL);
```

HRESULT SetCooperativeLevel(

HWND *hwnd*,

DWORD *dwLevel*

);

SetCooperativeLevel : 인자

- **HWND** *hwnd*
  - 기본 윈도우에 대한 핸들
- **DWORD** *dwLevel*
  - 협력 수준 flag
  - DSSCL_EXCLUSIVE : 사운드장치의 독점, 백그라운드에서는 소리가 출력되지 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

SetCooperativeLevel : 결과

- 성공
  - DS_OK
- 에러
  - DSERR_ALLOCATED
  - DSERR_INVLIDPARAM
  - DSERR_UNINITIALIZED
  - DSERR_UNSUPPORTED

SetCooperativeLevel : 예제


```
LPDIRECTSOUND8 lpDirectSound;
HRESULT hr;

// 1차 사운드 장치를 이용해 IDirectSound 객체 생성
…
// 1차 포맷 변경
hr =  lpDirectSound-
      >SetCooperativeLevel(hWnd,DSSCL_PRIORITY);
```

HRESULT IDirectSound8::CreateSoundBuffer(

LPCDSBUFFERDESC *pcDSBufferDesc*,

LPDIRECTSOUNDBUFFER * *ppDSBuffer*,

LPUNKNOWN *pUnkOuter*

);

CreateSoundBuffer : 인자

- **LPCDSBUFFERDESC** *pcDSBufferDesc*
  - DSBUFFERDESC 구조체의 포인터
- **LPDIRECTSOUNDBUFFER** * *ppDSBuffer*
  - DirectSoundBuffer의 포인터
- **LPUNKNOWN** *pUnkOuter*
  - 사용하지 않음
  - NULL로 지정

CreateSoundBuffer : 결과

- 성공
  - DS_OK
  - DS_NO_VIRTUALIZATION
- 에러
  - DSERR_ALLOCATED
  - DSERR_BADFORMAT
  - DSERR_BUFFERSMALL
  - DSERR_CONTROLUNAVIL
  - DSERR_DS8_REQUIRED
  - DSERR_INVALIDCALL
  - DSERR_INVALIDPARAM
  - DSERR_NOAGGREGATION
  - DSERR_OUTOFMEMORY
  - DSERR_UNINITIALIZED
  - DSERR_UNSUPPORTED

CreateSoundBuffer : 예제


```
LPDIRECTSOUND8 lpDSound;
HRESULT hr;

// 1차 사운드 장치를 이용해 IDirectSound 객체 생성
...

LPDIRECTSOUNDBUFFER pDSBPrimary = NULL;

memset(dsbd, 0, sizeof(DSBUFFERDESC));
dsbd.dwFlags            = DSBCAPS_PRIMARYBUFFER
dsbd.dwBufferByters     = 0;
dsbd.lpwfxformat        = NULL;

hr = lpDSound->CreateSoundBuffer(&dsbd, &pDSBPrimary, NULL);
```

2차 버퍼 : data를 버퍼에 저장


```
HRESULT IDirectSoundBuffer8::Lock(
        DWORD dwOffset,
        DWORD dwBytes,
        LPVOID * ppvAudioPtr1,
        LPDWORD pdwAudioBytes1,
        LPVOID * ppvAudioPtr2,
        LPDWORD pdwAudioBytes2,
        DWORD dwFlags
);
```

Lock : 인자

- **DWORD** *dwOffset*
  - lock이 걸릴 부분을 지정
  - 보통 0
- **DWORD** *dwBytes*
  - lock이 설정될 바이트 수
- **LPVOID** * *ppvAudioPtr1*
  - lock이 시작되는 첫번째 블록에 대한 포인터
- **LPDWORD** *pdwAudioBytes1*
  - 첫번째 블록의 바이트 수를 받는 포인터
- **LPVOID** * *ppvAudioPtr2*
  - lock된 두번째 블록에 대한 포인터
- **LPDWORD** *pdwAudioBytes2*
  - 두번째 블록의 바이트 수를 받는 포인터
- **DWORD** *dwFlags*
  - lock 방법을 정하는 flag
  - DSBLOCK_FROMWRITECURSOR : 현재 쓰기를 하는 부분부터 lock
  - DSBLOCK_ENTIREBUFFER : 전체 버퍼에 대해 lock

Lock : 결과

- 성공
  - DS_OK
- 에러
  - DSERR_BUFFERLOST
  - DSERR_INVALIDCALL
  - DSERR_INVALIDPARAM
  - DSERR_PRIOLEVELNEEDED

Lock : 예제


```
LPDIRECTSOUNDBUFFER8 lpBufer;
HRESULT hr;

int bufferSize;                    // Lock의 크기
void* pbData = NULL;               // lock의 시작주소
void* pbData2 = NULL;      // 순환 후 시작주소
ulong dwLength;            // lock된 바이트수
ulong dwLength2;           // 순환 후 바이트수

// 버퍼 전체에 lock을 건다
hr = lpBufer->Lock(0, bufferSize, &pbData, &dwLength,
     &pbData2, &dwLength2, DSBLOCK_ENTIREBUFFER);
```

2차 버퍼 : data를 버퍼에 저장

HRESULT IDirectSoundBuffer8::**Unlock(**

LPVOID *pvAudioPtr1*,

DWORD *dwAudioBytes1*,

LPVOID *pvAudioPtr2*,

DWORD *dwAudioBytes2*

);

Unlock : 인자

- **LPVOID** *pvAudioPtr1*
  - Lock이 시작되는 주소
- **DWORD** *dwAudioBytes1*
  - ppvAudioPtr1에 있는 바이트수
  - 즉, lock을 풀고자 하는 첫번째 블럭의 크기
- **LPVOID** *pvAudioPtr2*
  - pdwAudioBytes2로 되돌려진 포인터
  - 즉, lock을 풀고자 하는 두번째 블록의 포인터
- **DWORD** *dwAudioBytes2*
  - pdwAudioBytes2에 있는 바이트수
  - 즉, lock을 풀고자 하는 두번째 블록의 크기

Unlock : 결과

- 성공
  - DS_OK
- 에러
  - DSERR_INVALIDCALL
  - DSERR_INVALIDPARAM
  - DSERR_PRIOLEVELNEEDED

Unlock : 예제


```
LPDIRECTSOUNDBUFFER8 lpBufer;
HRESULT hr;

int bufferSize;                 // Lock의 크기
void* pbData = NULL;            // lock의 시작주소

// lock을 건다.
...
// 데이터를 읽어 저장한다.
...

// lock을 풀어준다
hr = lpBufer->Unlock(pbData, bufferSize, NULL, 0);
```

2차 버퍼 : 버퍼를 재생

```
HRESULT IDirectSoundBuffer8::Play(
        DWORD dwReserved1,
        DWORD dwPriority,
        DWORD dwFlags
);
```

Play : 인자

- **DWORD** *dwReserved1*
  - 예약값
  - 0으로 설정
- **DWORD** *dwPriority*
  - 사운드 우선 순위
  - DBSCAPS_LOCDEFER로 설정되지 않았으면 0
- **DWORD** *dwFlags*
  - play 방법
  - DSBPLAY_LOOPING : 계속 재생
  - DSBPLAY_LOCHARDWARE : 하드웨어 버퍼에서만 사운드 재생
  - DSBPLAY_LOCSOFTWARE : 소프트웨어 버퍼에서만 사운드 재생
  - DSBPLAY_TERMINATEBY_TIME : 다른 버퍼에서 자원을 가져오는 것 허용, 시간을 가지고 선택
  - DSBPLAY_TERMINATEBY_DISTANCE : 3D버퍼에만 관련
  - DSBPLAY_TERMINATEBY_PRIORITY : 다른 버퍼에서 자원을

Play : 결과

- 성공
  - DS_OK
- 에러
  - DSERR_BUFFERLOST
  - DSERR_INVALIDCALL
  - DSERR_INVALIDPARAM
  - DSERR_PRIOLEVELNEEDED

HRESULT IDirectSoundBuffer8::Stop();

Stop : 결과

- 성공
  - DS_OK
- 에러
  - DSERR_PRIOLEVELNEEDED

```
LPDIRECTSOUNDBUFFER8 lpBufer;
HRESULT hr;

// 계속 재생
hr = lpBufer->Play(0, 0, DSBPLAY_LOOPING);
….

// 재생 중지
hr = lpBufer->Stop();
```

DirectMusic

- 게임 또는 멀티미디어에서 음악 출력
- Midi와 같은 작은 용량의 파일 사용
- DirectMusic 전용 음악 포맷 사용
- mp3등에 비해 음질이 떨어짐
- 파일 입출력 지원

DirectMusic vs. DirectSound

기능	DirectMusic	DirectSound
WAV 재생	O	O
MIDI 재생	O	X
DirectMusic Producer 재생	O	X
파일 입출력	O	X
실행시 음악 인자 제어	O	X
timeline 제어	O	X
DLS	O	X
각 사운드의 볼륨, pitch 설정	O DirectSound API 이용	O

DirectMusic vs. DirectSound

기능	DIrectMusic	DirectSound
다중사운드의 볼륨설정	O	X
3D 사운드 구현	O DirectSoundAPI 이용	O
효과음	O	O
믹싱 효과를 위한 chain 버퍼	O	X
WAV 사운드 녹음	X	O
full duplex 구현	X	O
MIDI 녹음	O	X
하드웨어 버퍼 제어	X	O

- Header File
  - dmusicc.h
  - dmusici.h
  - dmerr.h
  - dsound.h

- DLL File
  - dmusic.dll

DirectMusic 인터페이스

- IDirectMusicLoader8
  - DirectMusic이 지원하는 파일을 load
- IDirectMusicPerformance8
  - 사운드 재생에 관련된 인터페이스
  - 하나의 채널을 가짐
- IDirectMusicSegment8
  - DirectSound의 버퍼와 같은 개념
  - 훨씬 강력하고 유연
  - 서로 다른 data 형식간의 바인딩이 가능

DirectMusic 실행과정

1. Performance 객체 생성
   - CoCreateInstance로 생성
2. DirectMusic 초기화
3. Loader 객체 생성
   - CoCrateInstance로 생성
4. 음악 파일 load
5. 신디사이저로 이동
6. 재생

Performance 객체 생성


```
IDirectMusicPerformance8** ppPerf;
HRESULT hr = CoCreateInstance(
    CLSID_DirectMusicPerformance,
    NULL,
    CLSCTX_INPROC,
    IID_IDirectMusicPerformance8,
    (void**)ppPerf
);
```

```
HRESULT IDirectMusicPerformance8::InitAudio(
        IDirectMusic** ppDirectMusic,
        IDirectSound** ppDirectSound,
        HWND hWnd,
        DWORD dwDefaultPathType,
        DWORD dwPChannelCount,
        DWORD dwFlags,
        DMUS_AUDIOPARAMS *pParams
);
```

InitAudio : 인자

- **IDirectMusic**** *ppDirectMusic*
  - DirectMusic 객체의 인터페이스 포인터
- **IDirectSound**** *ppDirectSound*
  - DirectSound 객체의 인터페이스 포인터
- **HWND** *hWnd*
  - 기본 윈도우 핸들
- **DWORD** *dwDefaultPathType*
  - 기본 audiopath 형식을 지정
- **DWORD** *dwPChannelCount*
  - 채널의 수
- **DWORD** *dwFlags*
  - 초기화 방법 지정
- **DMUS_AUDIOPARAMS *** *pParams*
  - 신디사이저와 관련된 DMUS_AUDIOPARAMS의 포인터, NULL로 지정

InitAudio : 인자 - dwFlags

- dwFlags
  - DMUS_AUDIOF_3D : 3D 버퍼 사용
  - DMUS_AUDIOF_ALL : 모든 flag 사용
  - DMUS_AUDIOF_BUFFERS : 다중 버퍼
  - DMUS_AUDIOF_STREAMING : 스트리밍 지원

InitAudio : 결과

- 성공
  - S_OK
- 에러
  - DMUS_E_ALREADY_INITED
  - DSERR_BUFFERLOST
  - DSERR_PRIOLEVELNEEDED
  - DSERR_UNINITIALIZED
  - E_NOINTERFACE
  - E_OUTOFMEMORY
  - E_POINTER

InitAudio : 예제

```
IDirectMusicPerformance8* pPerf;
HRESULT hr;

hr = pPerf->InitAudio(NULL, NULL, hWnd,
    DMUS_APATH_SHARED_STEREOPLUSREVERB, 16,
    DMUS_AUDIOF_ALL, NULL);
```

Loader 객체 생성


```
IDirectMusicLoader8* pLoader;
HRESULT hr = CoCreateInstance(
    CLSID_DirectMusicLoader,
    NULL,
    CLSCTX_INPROC,
    IID_IDirectMusicLoader8,
    (void**)&pLoader
);
```

음악 파일 Load

```
HRESULT IDirectMusicLoader8::LoadObjectFromFile(
        REFGUID rguidClassID,
        REFIID iidInterfaceID,
        WCHAR *pwzFilePath,
        void ** ppObject
);
```

LoadObjectFromFile : 인자

- **REFGUID** *rguidClassID*
  - 객체 클래스의 ID
- **REFIID** *iidInterfaceID*
  - 인터페이스 ID
- **WCHAR** ***pwzFilePath**
  - 음악파일 경로
- **void **** *ppObject*
  - 인퍼페이스 포인터

LoadObjectFromFile : 결과

- 성공
  - S_OK
  - DMUS_S_PARTIALLOAD
- 에러
  - DMUS_E_LOADER_FAILEDCREATE
  - DMUS_E_LOADER_FAILEDOPEN
  - DMUS_E_LOADER_FORMATNOTSUPPORTED
  - DMUS_E_LOADER_NOCLASSID
  - E_FAIL
  - E_INVALIDARG
  - E_OUTOFMEMORY
  - REGDB_E_CLASSNOTREG

LoadObjectFromFile : 예제


```
IDirectMusicLoader8* pLoader;
IDirectMusicSegment8 * pSegment;
WCHAR wszName[MAX_PATH] = L”test.mid”;

HRESULT hr = pLoader->LoadObjectFromFile(
    CLSID_DirectMusicSegment,
    IID_IDirectMusicSegment8,
    wszNames,
    (void**) &pSegment);
```

HRESULT IDirectMusicSegment8:: Download(

IUnknown* *pAudioPath*

);

- **IUnknown*** *pAudioPath*
  - data를 받을 IUnknown 인터페이스 포인터

Download : 결과

- 성공
  - S_OK
  - DMUS_S_PARTIALDOWNLOAD
- 에러
  - DMUS_E_NOT_FOUND
  - DMUS_E_TRACK_NOT_FOUND
  - E_POINTER

Download : 예제


```
IDirectMusicPerformance8* pPerf;
IDirectMusicSegment8 * pSegment;

// 음악 data load
…

HRESULT hr = pSegment->Download(pPerf);
```

```
HRESULT PlaySegmentEx(
        IUnknown* pSource,
        WCHAR *pwzSegmentName,
        IUnknown* pTransition,
        DWORD dwFlags,
        __int64 i64StartTime,
        IDirectMusicSegmentState** ppSegmentState,
        IUnknown* pFrom,
        IUnknown* pAudioPath
);
```

PlaySegmentEx : 인자

- **IUnknown*** *pSource*
  - 재생할 곳의 주소
- **WCHAR** ***pwzSegmentName***
  - NULL로 지정
- **IUnknown*** *pTransition*
  - template segment의 IUnknown 인터페이스 포인터
- **DWORD** *dwFlags*
  - 재생 방법 지정
- **__int64** *i64StartTime*
  - segment의 재생을 시작하는 시간
- **IDirectMusicSegmentState**** *ppSegmentState*
  - 재생중의 segement 인터페이스
- **IUnknown*** *pFrom*
  - 새로운 segment의 재생을 시작할 때 정지하는 segment 상태나 audiopath의 인터페이스 포인터

PlaySegmentEx : 결과

- 성공
  - S_OK
- 에러
  - DMUS_E_AUDIOPATH_INACTIVE
  - DMUS_E_AUDIOPATH_NOPORT
  - DMUS_E_NO_MASTER_CLOCK
  - DMUS_E_SEGMENT_INIT_FAILED
  - DMUS_E_TIME_PAST
  - E_OUTOFMEMORY
  - E_POINTER

PlaySegmentEx : 예제


```
IDirectMusicPerformance8* pPerf;
IDirectMusicSegment8 * pSegment;
IDirectMusicSegmentState8* pSegmentStateB;

// 음악 data load
...
// 신디사이저로 download
...

HRESULT hr = pPerf->PlaySegmentEx(
        pSegment,
        NULL, NULL,
        DMUS_SEGF_QUEUE |
```