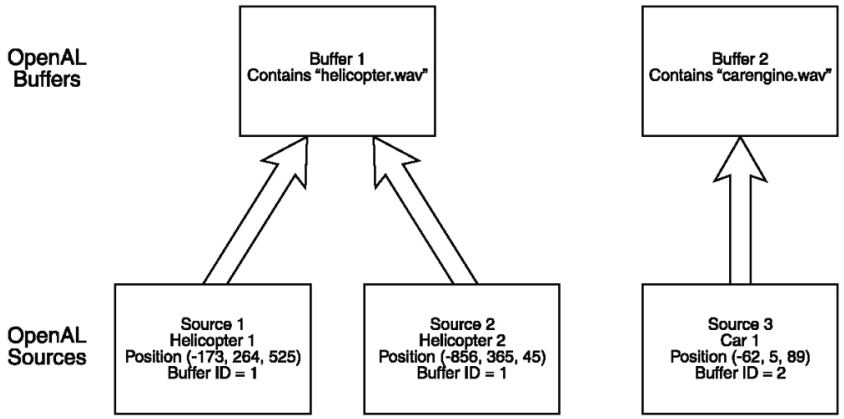
Sound Programming

제3장 Wave Audio Playback in OpenAL

OpenAL에서 Wav 파일 재생하기

- 3D 사운드는 아니지만 OpenAL에서도 단순히 Wav 파일을 재생할 수 있다.
- 단순 Wav 파일 재생을 위해서도 Buffer, Source는 생성해야 함.
  - Listener 속성은 건드릴 필요 없음.



Wav 파일 재생 순서

샘플 예제 PlayStatic의 main()

- 샘플 **PlayStatic: WAV** 파일을 재생하는 예

```
#include "Framework.h"

#define TEST_WAVE_FILE "Footsteps.wav"

int main()
{
    ALuint    uiBuffer;
    ALuint    uiSource;
    Alint     iState;

    // Initialize Framework
    ALFWInit();

    ALFWprintf("PlayStatic Test Application\n");

    if (!ALFWInitOpenAL())
    {
        ALFWprintf("Failed to initialize OpenAL\n");
        ALFWShutdown();
        return 0;
    }

    // Generate an OpenAL Buffer
    alGenBuffers( 1, &uiBuffer );

    // Load Wave file into OpenAL Buffer
    if (!ALFWLoadWaveToBuffer((char*)ALFWaddMediaPath(TEST_WAVE_FILE), uiBuffer)) {
        ALFWprintf("Failed to load %s\n", ALFWaddMediaPath(TEST_WAVE_FILE));
    }
    ……
```

OpenAL 초기화 부분

Buffer 생성 부분

Buffer로 Wav 파일 내용을 적재하는 부분

샘플 예제 PlayStatic의 main()(계속)

```
// Generate a Source to playback the Buffer
alGenSources( 1, &uiSource );

// Attach Source to Buffer
alSourcei( uiSource, AL_BUFFER, uiBuffer );

// Play Source
alSourcePlay( uiSource );
ALFWprintf("Playing Source ");

do {
   Sleep(100);
   ALFWprintf(".");
   // Get Source State
   alGetSourcei( uiSource, AL_SOURCE_STATE, &iState);
} while (iState == AL_PLAYING);

ALFWprintf("\n");

// Clean up by deleting Source(s) and Buffer(s)
alSourceStop(uiSource);
alDeleteSources(1, &uiSource);
alDeleteBuffers(1, &uiBuffer);

ALFWShutdownOpenAL();

ALFWShutdown();

return 0;
}
```

Source 생성 부분

Source와 버퍼의 연결 부분

Source 재생

종료 시점을 알기 위해 재생 중인 Source의 상태를 검사하는 부분.

재생 중지 및 Cleanup

OpenAL 셧다운(context와 device 제거)

Step 1 - OpenAL 초기화

- **framework 라이브러리에서 OpenAL 초기화 및 cleanup wrapping 함수 지원**
  - SDK 설치 시 samples/framework 폴더에 소스가 제공됨
  - OpenAL 응용에서 자주 사용하는 기능들을 라이브러리화 한 것
  - 샘플 소스 프로젝트에 이미 포함되어 있음.
- **C 기본 수형 사용 방법**
  - ALint, ALuint 등 수형 앞에 AL을 붙여 사용할 것을 권장
  - 헤더파일 al.h에 typedef로 정의되어 있음.
- **초기화 과정**
  - void ALFWinit(): Wave 파일 로드를 위한 클래스인 CWaves 인스턴스를 생성함.
  - ALboolean ALFWInitOpenAL(): 사운드 장치를 선택한 후, 선택된 device 오픈하고 OpenAL context를 오픈한다.

Step 2 – buffer 생성

- **OpenAL에서의 buffer**
  - Wav 음원을 저장하는 용도
  - alGenBuffers() 함수로 생성

```
void alGenBuffers(
    ALsizei n,          // 생성할 버퍼 개수
    ALuint *buffers  // 새로 생성된 버퍼의 ID를 넘겨 받을 ALuint 배열의 시작 주소
);
```

- **PlayStatic 예제에서는 buffer 1개만 생성하고 있음**

```
ALuint     uiBuffer;
...

alGenBuffers( 1, &uiBuffer );     // 만들 버퍼 1개뿐 이므로 배열을 사용할 필요 없음.
```


Step 3 – Wav 파일을 buffer로 적재

- **OpenAL 자체가 Wav 로딩을 지원하지는 않음**
- **framework 라이브러리에서 별도로 지원하고 있음**
  - 우리는 framework 라이브러리를 이용하기만 하면 됨.

```
ALboolean ALFWLoadWaveToBuffer(
    const char *szWaveFile,              // 적재할 Wav 파일 이름(경로명)
    ALuint uiBufferID,                   // Wav 파일이 적재될 buffer ID
    ALenum eXRAMBufferMode = 0           // X-Fi 사운드카드인 경우에만 사용됨
);
```

- **우리 예제에서는 samples/media/Footstep.wav 파일을 적재하고 있음**

```
#define  TEST_WAVE_FILE  "Footsteps.wav"
.....
if (!ALFWLoadWaveToBuffer((char*)ALFWaddMediaPath(TEST_WAVE_FILE), uiBuffer)) {
    ALFWprintf("Failed to load %s\n", ALFWaddMediaPath(TEST_WAVE_FILE));
}
```

Step 4 - source 생성

- **OpenAL에서의 source**
  - 버퍼에 적용될 다양한 3D 음향효과 정보를 담는 용도
  - 여러 buffer가 스트림 형태도 하나의 source에 연결될 수도 있음
  - 반대로 한 buffer에 여러 source가 연결되어, 같은 음원에 서로 다른 음향효과를 주고 재생할 수도 있음.
  - alGenSources() 함수로 생성

```
void alGenSources(
    ALsizei n,          // 생성할 source 개수
    ALuint *sources     // 새로 생성된 source의 ID를 넘겨 받을 ALuint 배열의 시작 주소
);
```

- **PlayStatic 예제에서는 버퍼 1개만 생성하고 있음**

```
ALuint    uiSource;
...

alGenSources( 1, &uiSource );     // 만들 소스가 1개뿐 이므로 배열을 사용할 필요 없음.
```


Step 5 - source를 buffer에 연결하기

- 생성된 source의 속성을 변경하는 함수 alSourcei()를 이용하여 source와 buffer를 연결한다.

```
// sets an integer property of a source
void alSourcei(
    ALuint source,    // 속성을 변경하고자 하는 source ID
    ALenum param,     // 어떤 속성을 변경할지를 나타내는 source 속성 ID
    ALint value       // 새로 설정할 속성 값
);
```

  - param에 AL_BUFFER를 주면 source 속성 중 buffer ID를 변경하겠다는 의미.
  - 이 경우 value에는 연결을 원하는 buffer ID를 주면 된다.

```
alSourcei( uiSource, AL_BUFFER, uiBuffer );
```

Step 6 - 재생

- **source** 함수 **alSourcePlay()**를 이용하여 재생
  - source에 연결된 버퍼의 내용이 재생된다.

```
// play a source
void alSourcePlay(
     ALuint source      // 재생할 source ID
);
```

- `alSourcePlay( uiSource );`
  `ALFWprintf("Playing Source ");` → 재생 시작 → 제어는 바로 다음 라인으로 넘어감
- **alSourcePlay() 후 source**의 상태 변화
  - AL_PLAYING으로 바뀜.
  - 재생 중에 alSourcePlay()가 중복 호출되면 재생은 처음부터 다시 시작함
  - 연결되어 있는 모든 buffer들의 재생이 끝나면 AL_STOPPED 상태로 바뀜.

Step 7 - 재생 종료 검사

- **alSourcePlay()**를 호출하면 재생은 시작되고, 제어는 다음 줄로 넘어감 → 재생 종료 검사가 필수적임.
- **source** 상태 검사 함수 **alGetSourcei()**를 이용

```
void alGetSourcei(
     ALuint source,     // 속성을 알아볼 source ID
     ALenum pname,      // 속성 ID
     ALint *value       // 속성을 넘겨 받을 포인터
);
```

  - 이 예제에서는 pname에 AL_SOURCE_STATE를 주어 source의 재생 상태를 알아본다.

```
Alint       iState;
.....
do {
        Sleep(100);
        ALFWprintf(".");
        // Get Source State
        alGetSourcei( uiSource, AL_SOURCE_STATE, &iState);
    } while (iState == AL_PLAYING);
```

Step 8 – Cleanup & OpenAL shutdown

- 재생이 종료되면 다음과 같은 순서로 **cleanup** 함
  - 재생을 중지시킴 → source 삭제 → buffer 삭제 → context 삭제 → device 닫음 → CWaves 클래스 인스턴스 삭제

```
// Clean up by deleting Source(s) and Buffer(s)
alSourceStop(uiSource);          // 재생 중지
alDeleteSources(1, &uiSource);   // source 삭제
alDeleteBuffers(1, &uiBuffer);   // buffer 삭제

ALFWShutdownOpenAL();            // context 삭제, device close

ALFWShutdown();                  // CWaves 클래스 인스턴스 삭제
```


Reports

- **PlayStatic** 프로그램을 컴파일하고, **Wav** 파일을 다른 파일로 변경한 후 실행시키시오.(제출 내용 없음)
  - mp3를 Wav로 변환하는 프로그램을 사용하여 자신이 좋아하는 음악을 재생해 볼 것!

- **PlayStatic** 프로그램을 확장하여, 임의의 **Wav** 파일 두 개를 재생하는 프로그램을 작성하시오.
  - 종료 검사를 안 하고 두 Wav를 재생하면 어떤 결과가 나오는가?
  - 제출내용: 프로젝트 폴더 전체 압축 파일, 프로그램 설명서(워드 또는 한글 또는 PDF)

Q & A