

2006년 2학기 윈도우 게임 프로그래밍

제9강 사운드 처리 (1)



이대현

한국산업기술대학교



P 한국산업기술대학교

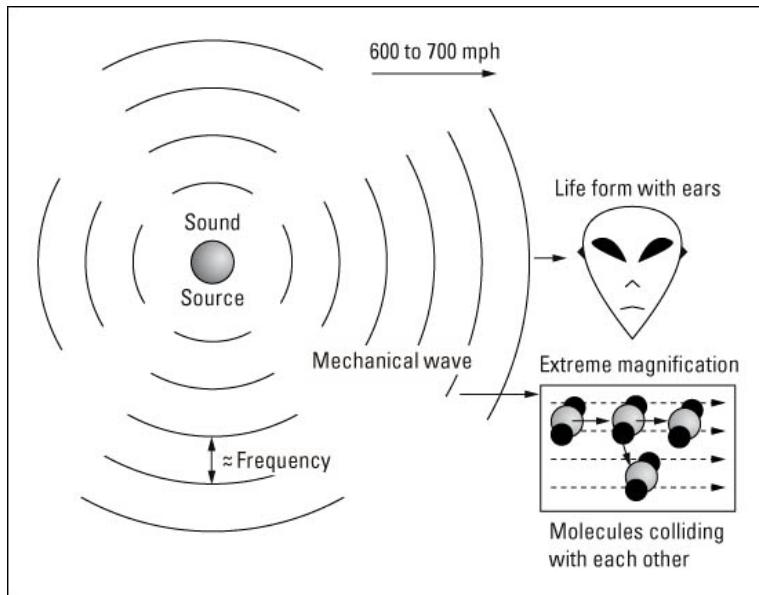
오늘의 학습 내용

- 컴퓨터 사운드의 이해
- SDL을 이용한 사운드 처리
 - 배경 음악 연주
 - 효과음 발생

사운드란 무엇인가?

■ 사운드

- 음원으로부터의 기계적 진동이 공기 등의 매체를 통해 전파되는 것.
- 전파되는데 시간이 걸린다.
 - 공기: 344m/s
 - 바다: 1478 m/s
 - 강철: 5064 m/s



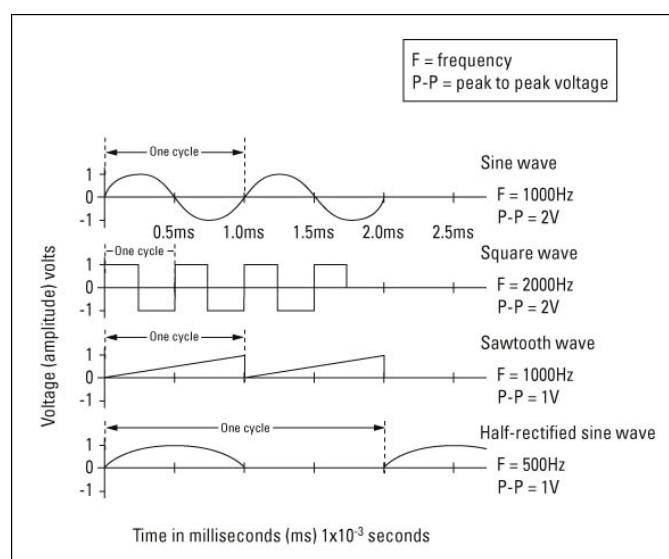
진폭과 주파수

■ 진폭(Amplitude)

- 파형의 크기

■ 주파수(Frequency)

- 초당 특정 파형이 반복되는 횟수. 단위는 Hz
- 가정 주파수: 20 – 20,000 Hz
- 남자: 20 – 20,000 Hz
- 여자: 70 – 30,000 Hz



디지털 사운드와 합성 사운드

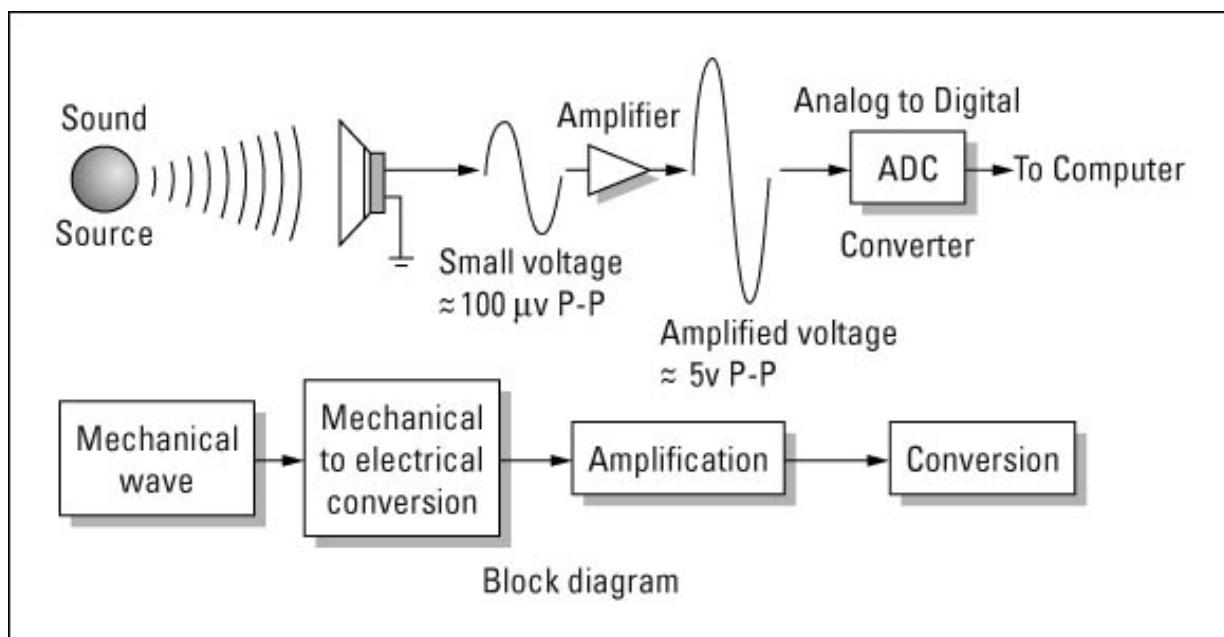
■ 디지털 사운드(Digital Sound)

- 소리의 직접적인 녹음
- 효과음 등에 사용(폭발, ...)

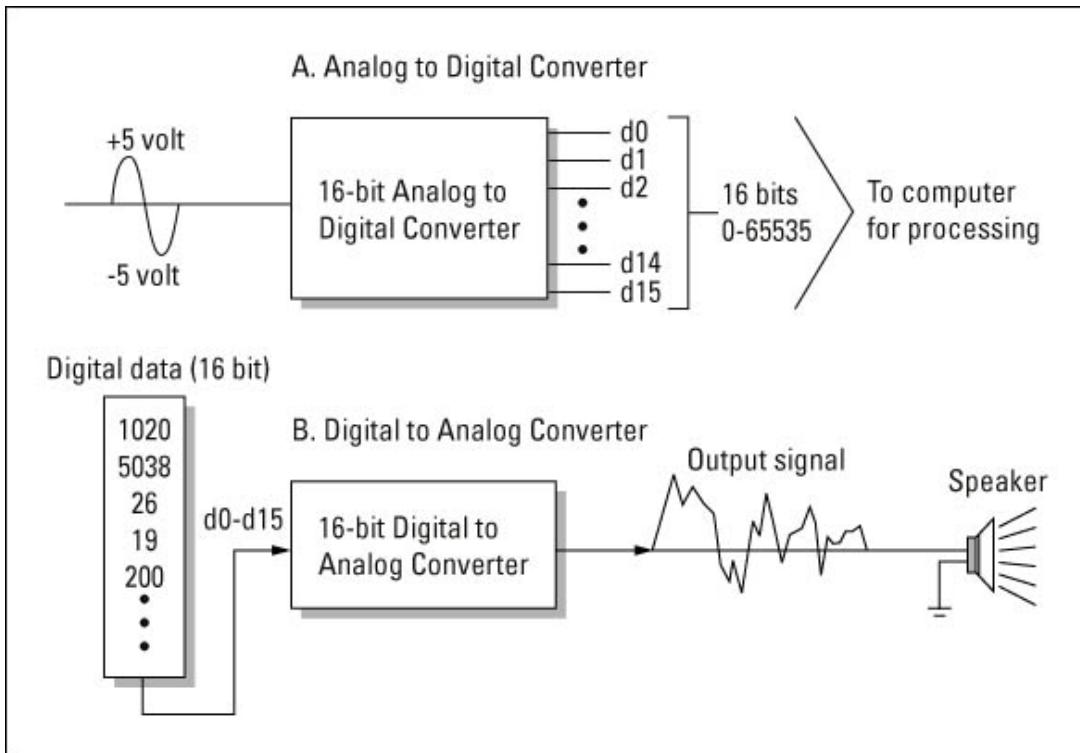
■ 합성 사운드

- 알고리즘과 톤 발생기에 의하여 합성된 소리.
- 주로 음악의 재생에 사용.

디지털 사운드의 기록



디지털 사운드의 재생



디지털 사운드의 샘플링 주파수와 데이터 비트수

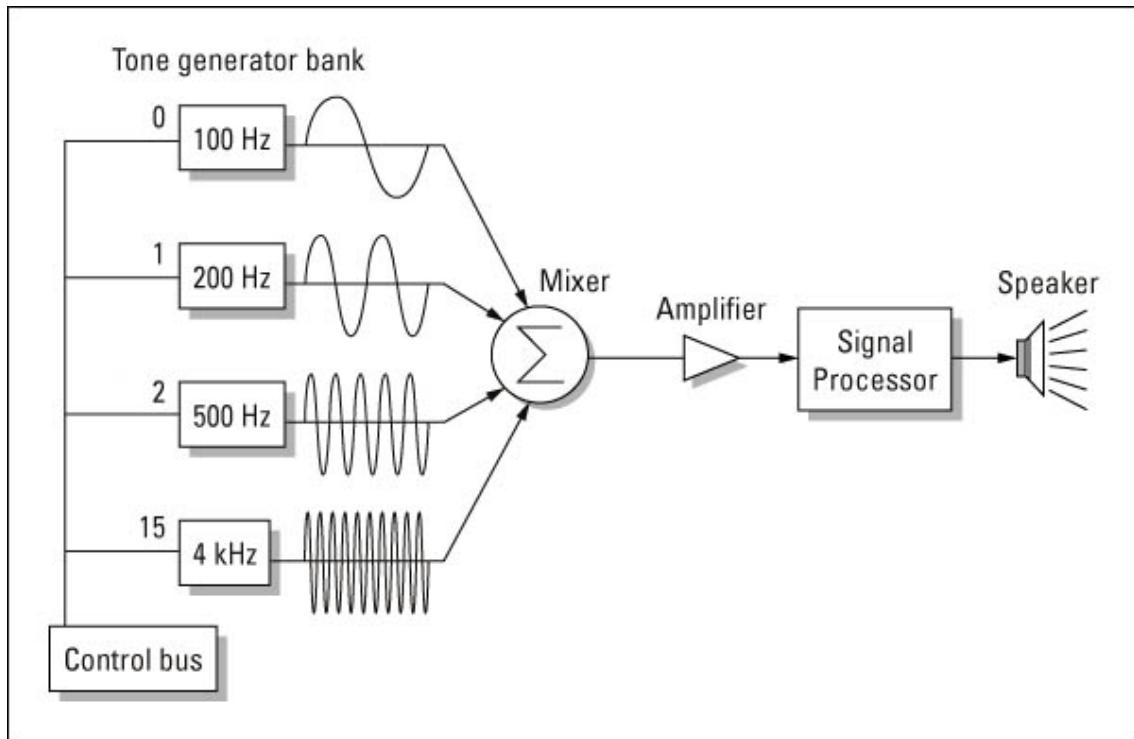
■ 샘플링 속도

- 디지털 사운드를 기록할 때, 초당 몇 번의 샘플링을 하는가?
- 샘플링되는 사운드 주파수보다 2배 이상으로 샘플링을 해야함.

■ 샘플 당 비트수(bits per samples)

- 8 비트 샘플: 256개의 진폭 크기. 게임 효과음 등에는 충분.
- 16 비트 샘플: 65536개의 진폭 크기. 음악 등에 사용.

합성 사운드의 재생



미디(MIDI)

- 미디 (MIDI: Musical Instruments Digital Interface)
 - 음악 재생을 기술하는 일종의 언어.

Turn on Channel 1 with a B flat

Turn on Channel 2 with a C sharp

Turn off Channel 1.

...

Turn all channels off.

첨크(Chunk)와 채널(Channel)

■ 첨크

- 일정한 크기를 가지는 사운드 데이터.

```
typedef struct {
    int allocated;
    Uint8 *abuf;
    Uint32 alen;
    Uint8 volume;
} Mix_Chunk;
```

■ 채널

- 소리를 전달하는 통로.
- 모노: 1채널
- 스테레오: 2채널
- 영화: 5.1 채널, 7.1 채널 등등.

SDL 을 이용한 사운드의 재생

Step0: SDL_mixer 라이브러리를 포함.(SDL_mixer.h, SDL_mixer.lib, SDL_mixer.dll 사용)

Step1: SDL_mixer 라이브러리 초기화(Mix_OpenAudio() 호출)

Step2: 사운드 파일의 로딩(Mix_LoadWav(), Mix_LoadMUS() 호출)

Step3: 사운드의 재생(Mix_PlayChannel(), Mix_PlayMusic() 등을 호출)

Step4: 사운드 파일의 해제(Mix_FreeChunk(), Mix_FreeMusic() 등을 호출)

Step5: 사운드 라이브러리 종료(Mix_CloseAudio() 호출)

참고 사이트: http://jcatki.no-ip.org/SDL_mixer/



Lecture09-01: 음악의 재생

Lecture09-01 프로젝트의 구성

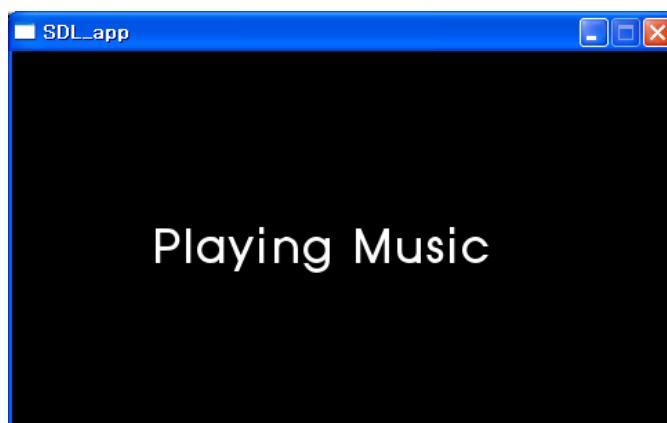
- C++ 소스 파일
 - main.cpp – 실습 시간에 작성.

- 음악 파일
 - Footballbgm.mp3 – 재생할 음악 파일(BGM 폴더 밑에 있느.)

main.cpp

```
Mix_OpenAudio(44100, MIX_DEFAULT_FORMAT, 2, 2048);
Mix_Music *bgm = Mix_LoadMUS("BGM/footballbgm.mp3");
Mix_PlayMusic(bgm, -1);
bool quit = false;
while (!quit) {
    SDL_Event event;
    if (SDL_PollEvent(&event)) {
        if (event.type == SDL_KEYDOWN)
            switch (event.key.keysym.sym) {
                case SDLK_ESCAPE:
                    quit = true;
                    break;
                case SDLK_p:
                    Mix_PauseMusic();
                    break;
                case SDLK_r:
                    Mix_ResumeMusic();
                    break;
            }
        if (event.type == SDL_QUIT)
            quit = true;
    }
}
Mix_HaltMusic();
Mix_FreeMusic(bgm);
Mix_CloseAudio();
```

실행 화면



사운드 라이브러리의 초기화 및 종료

```
int Mix_OpenAudio(int frequency, Uint16 format, int channels, int chunksize);
void Mix_CloseAudio();
```

```
Mix_OpenAudio(44100, MIX_DEFAULT_FORMAT, 2, 2048);
```

- 44.1khz 샘플링 주파수 사용. 일반적으로 게임의 경우는 22.05khz 도 충분함.
- MIX_DEFAULT_FORMAT: 16 비트 샘플링
- 스테레오 채널.
- 오디오 버퍼의 크기는 2048 바이트. 크기가 너무 작으면, 소리가 틀 수 있음. 버퍼를 너무 크게 하면 메모리 소요량이 많아지고, 재생 시작 시점에 밀릴 수 있음. 따라서, 적절하게 조절이 필요함.

음악의 로딩 및 해제

```
Mix_Music *Mix_LoadMUS(const char *file);
void Mix_FreeMusic(Mix_Music *music);
```

```
Mix_Music *bgm = Mix_LoadMUS("BGM/footballbgm.mp3");
```

```
.
.
.
.
.
```

```
Mix_FreeMusic(bgm);
```

- 음악 파일의 로딩. Wav, mp3 등등 로딩 가능.

- 음악 파일의 해제.

음악의 재생

```
int  
Mix_PlayMusic(Mix_Music *music, int loops);
```

```
Mix_PlayMusic( bgm, -1 );
```

- 로드된 *bgm* 을 재생
- *loops*는 음악의 재생 횟수.
- *loops* = -1 이면 무한 반복 재생.
- *loops* = 0 이면 정지.

음악 재생의 정지 및 재개

```
switch (event.key.keysym.sym) {  
    case SDLK_ESCAPE:  
        quit = true;  
        break;  
    case SDLK_p:  
        Mix_PauseMusic();  
        break;  
    case SDLK_r:  
        Mix_ResumeMusic();  
        break;  
}
```



Lecture09-02: 효과음의 재생

Lecture09-02 프로젝트의 구성

- C++ 소스 파일
 - main.cpp – 실습 시간에 작성.

- 음악 파일
 - Walk1-1.wav, walk1-2, walk2-1, walk2-2 – 재생할 음향 파일(Effect 폴더 밑에 있습니다.)

main.cpp (1)

```
if(Mix_OpenAudio(44100, MIX_DEFAULT_FORMAT, 2, 2048)==-1) {
    printf("Mix_OpenAudio: %s\n", Mix_GetError());
    exit(-1);
}

Mix_Chunk *walk1 = Mix_LoadWAV("Effect/walk1-1.wav");
Mix_Chunk *walk2 = Mix_LoadWAV("Effect/walk1-2.wav");
Mix_Chunk *walk3 = Mix_LoadWAV("Effect/walk2-1.wav");
Mix_Chunk *walk4 = Mix_LoadWAV("Effect/walk2-2.wav");
```

main.cpp (2)

```
bool quit = false;
while (!quit) {
    SDL_Event event;
    int no = 0;
    if (SDL_PollEvent(&event)) {
        if (event.type == SDL_KEYDOWN)
            switch (event.key.keysym.sym) {
                case SDLK_ESCAPE: quit = true; break;
                case SDLK_1: Mix_PlayChannel(-1, walk1, 0); no = 1; break;
                case SDLK_2: Mix_PlayChannel(-1, walk2, 0); no = 2; break;
                case SDLK_3: Mix_PlayChannel(-1, walk3, 0); no = 3; break;
                case SDLK_4: Mix_PlayChannel(-1, walk4, 0); no = 4; break;
            }
        if (event.type == SDL_QUIT)
            quit = true;
    }
    if (no) {
        char str[100];
        SDL_FreeSurface(msg);
        sprintf(str, "Playing Effect %d ", no);
        msg = TTF_RenderText_Shaded(font,str,color,bg);
        SDL_BlitSurface(msg, NULL, screen, &rect);
        SDL_Flip(screen);
    }
}
```

main.cpp (3)

```
Mix_FreeChunk(walk1);  
Mix_FreeChunk(walk2);  
Mix_FreeChunk(walk3);  
Mix_FreeChunk(walk4);  
  
Mix_CloseAudio();
```

실행 화면



- 1,2,3,4 키를 누르면 각기 다음 효과음이 재생된다.

효과음의 로딩 및 해제

```
Mix_Chunk *Mix_LoadWAV(char *file);  
void Mix_FreeChunk(Mix_Chunk *chunk);
```

```
Mix_Chunk *walk1 = Mix_LoadWAV("Effect/walk1-1.wav");  
Mix_Chunk *walk2 = Mix_LoadWAV("Effect/walk1-2.wav");  
Mix_Chunk *walk3 = Mix_LoadWAV("Effect/walk2-1.wav");  
Mix_Chunk *walk4 = Mix_LoadWAV("Effect/walk2-2.wav");
```

- 효과음 파일의 로딩

효과음의 재생

```
int Mix_PlayChannel(int channel,  
                    Mix_Chunk *chunk, int loops);
```

```
case SDLK_1: Mix_PlayChannel(-1, walk1, 0); no = 1; break;  
case SDLK_2: Mix_PlayChannel(-1, walk2, 0); no = 2; break;  
case SDLK_3: Mix_PlayChannel(-1, walk3, 0); no = 3; break;  
case SDLK_4: Mix_PlayChannel(-1, walk4, 0); no = 4; break;
```

- channel: 플레이할 채널을 선택. -1이면 할당되지 않는 채널을 임의로 사용.
- loops: 효과음 파일의 반복 횟수를 지정. +1 만큼 반복함. 따라서 여기서는 한번만 재생.

실습 과제 #7



- 배경음악 연주와 함께, 걸을 때 효과음의 연출
 - lecture09-03-dist.zip을 사용.
 - 플레이 화면이 시작되면, 배경 음악이 계속 연주되고, 캐릭터가 걸어다니면서 효과음이 재생됨.
 - 배경음악은 footballbgm.mp3 사용, 효과음은 네개의 효과음 중 한 개를 사용.

