

2D 게임 프로그래밍

# 제11강 사운드

이대현  
한국산업기술대학교



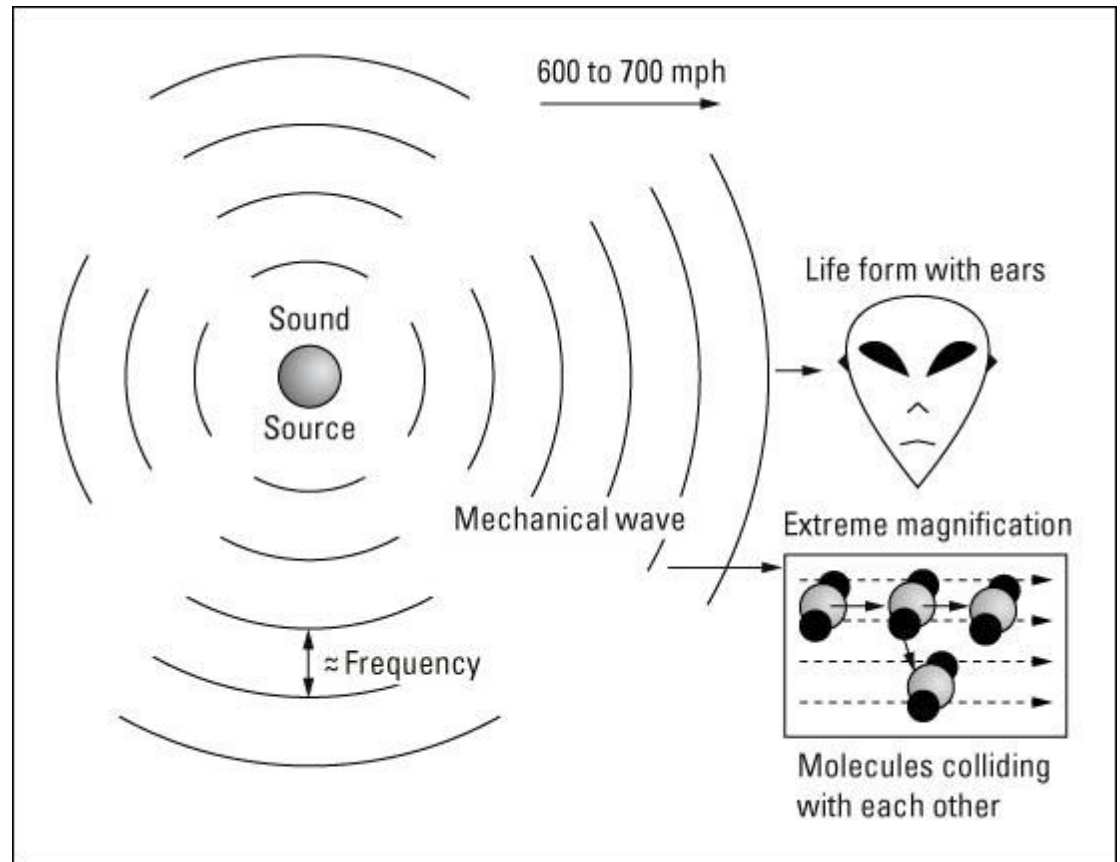
# 학습 내용

- 컴퓨터 사운드의 이해
- Pico2d의 사운드 플레이

# 사운드란 무엇인가?

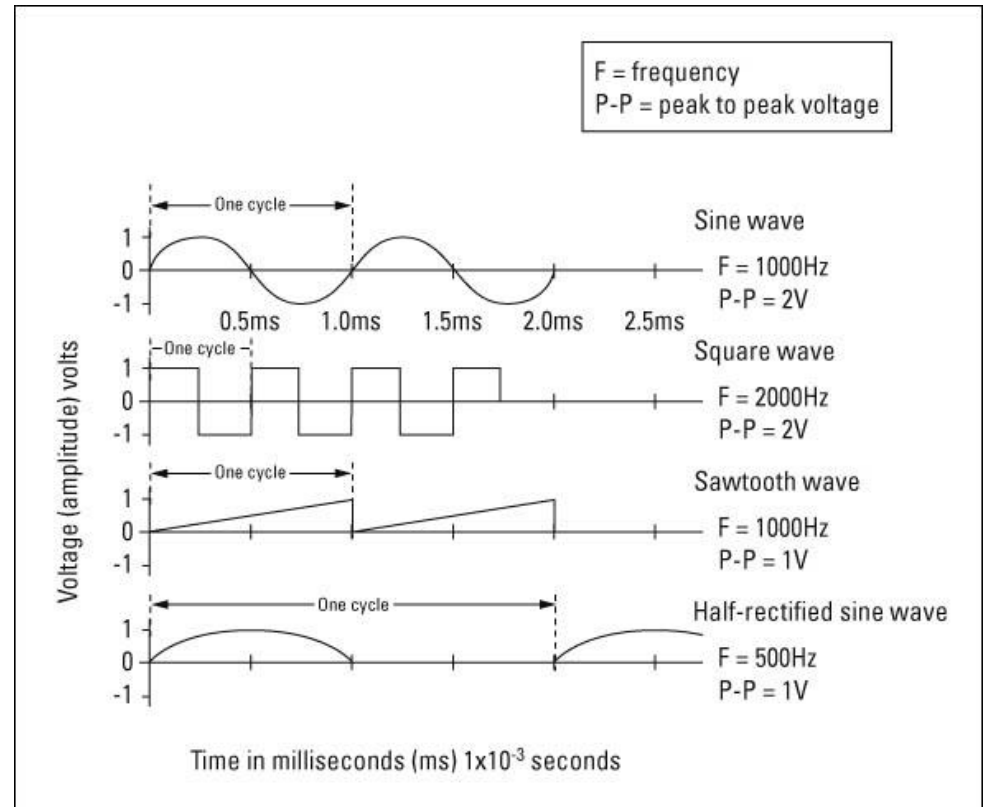
## ■ 사운드

- 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

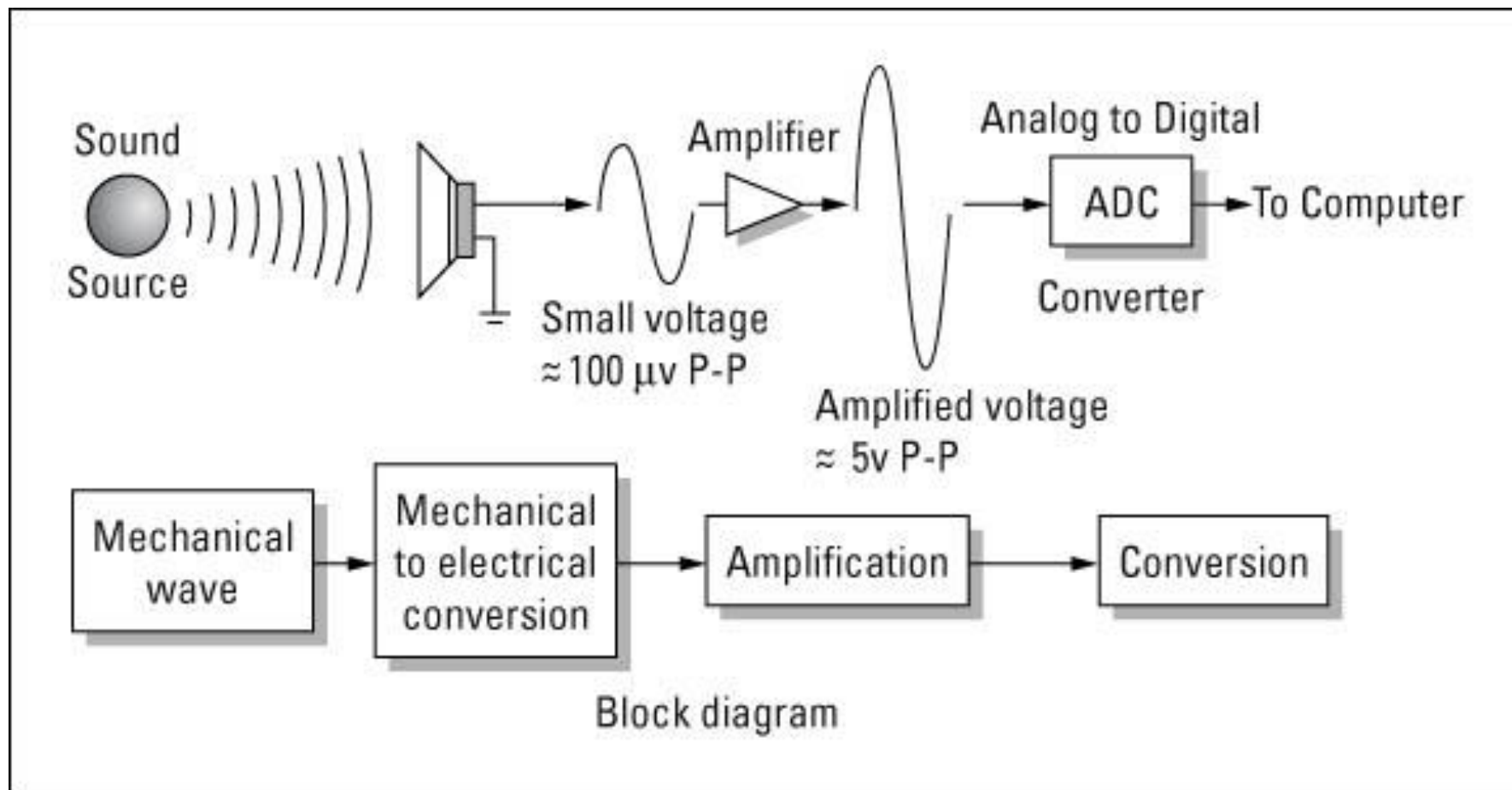
- 진폭(Amplitude)
  - 파형의 크기
- 주파수(Frequency)
  - 초당 특정 파형이 반복되는 횟수. 단위는 Hz
  - 가청 주파수: 20 - 20,000 Hz
  - 남자: 20 - 20,000 Hz
  - 여자: 70 - 30,000 Hz



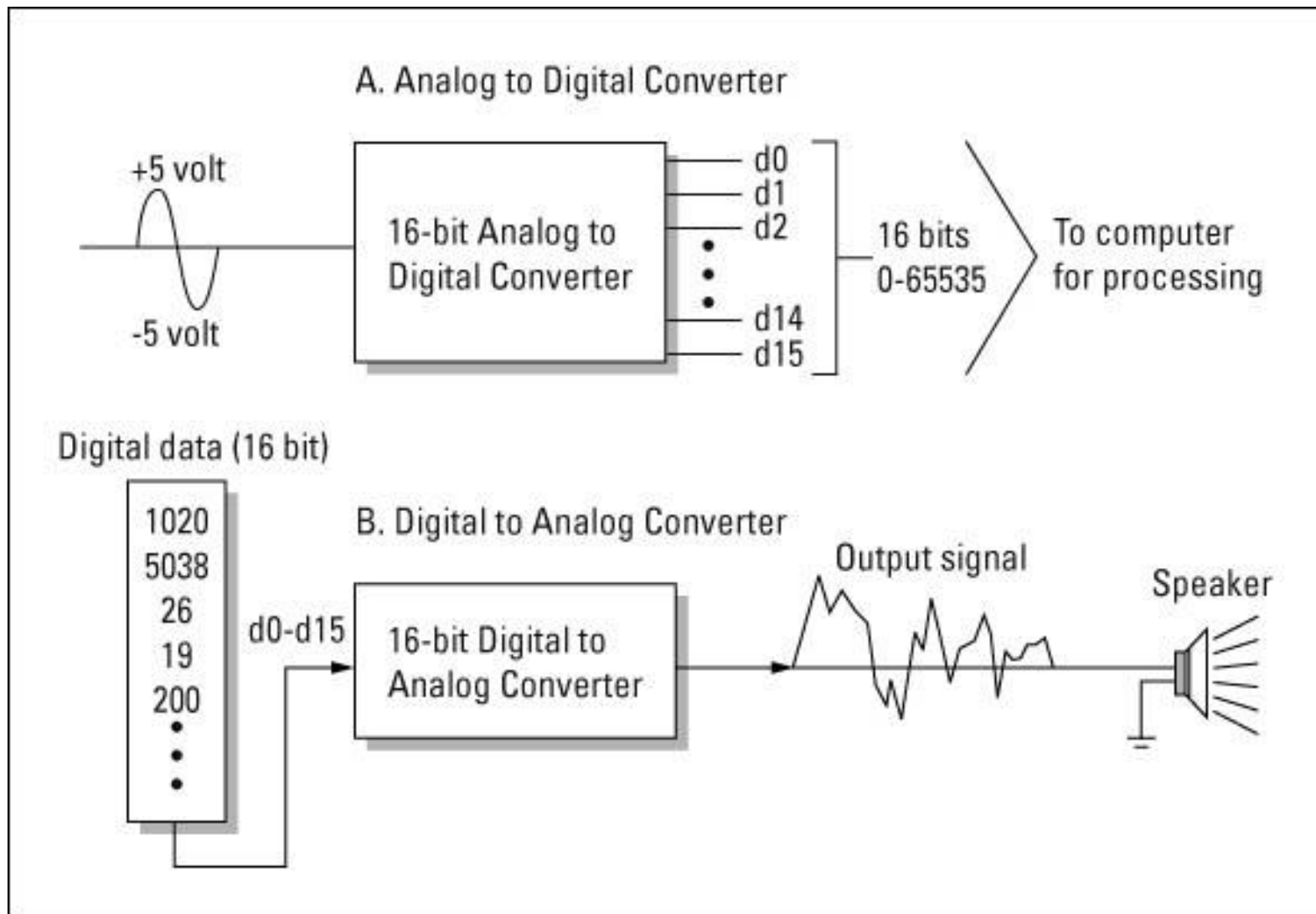
# 디지털 사운드와 합성 사운드

- 디지털 사운드(Digital Sound)
  - 소리의 직접적인 녹음
  - 효과음등에 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

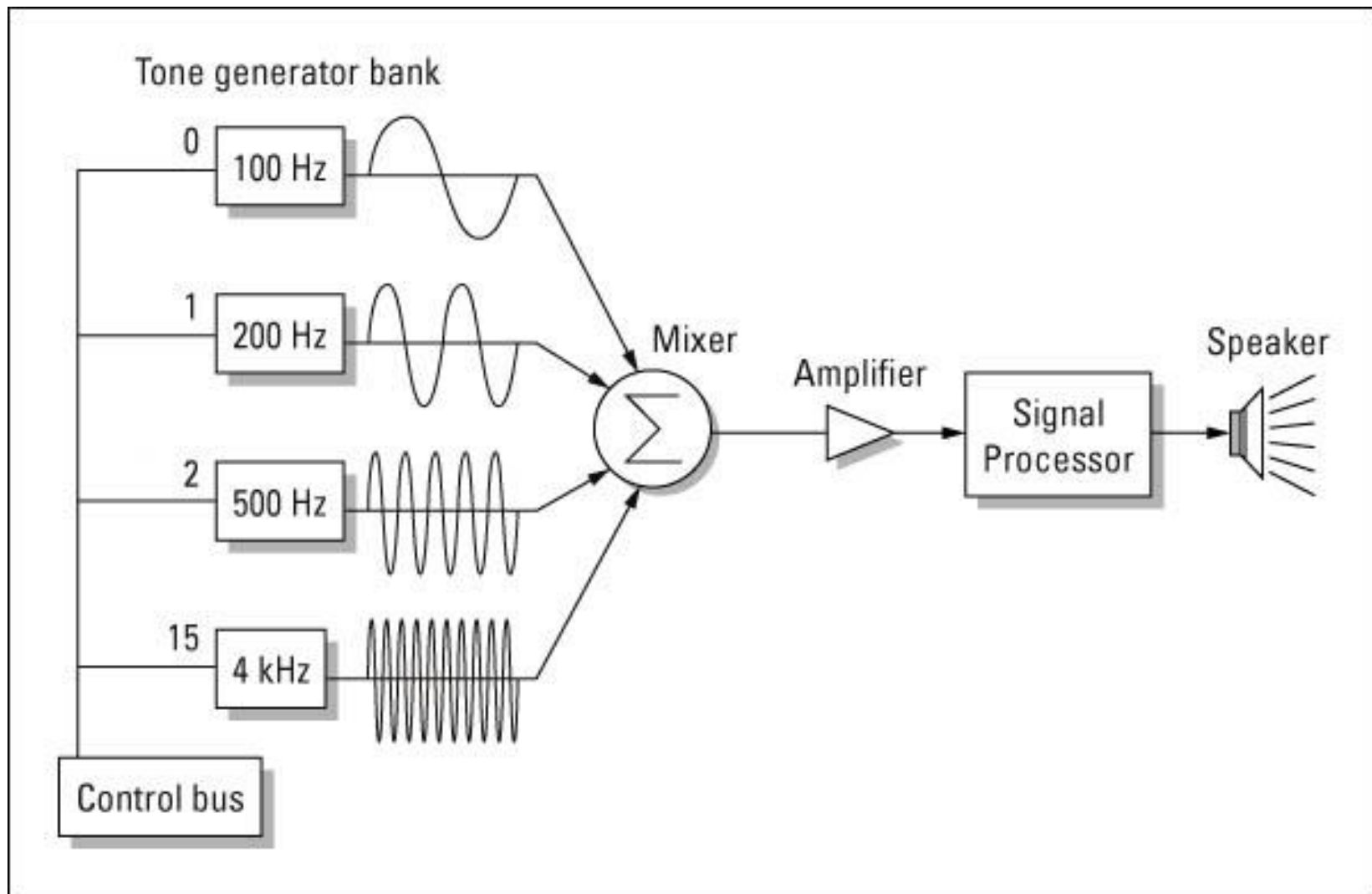
- 디지털 사운드를 기록할 때, 초당 몇번의 샘플링을 하는가?
- 샘플링되는 사운드 주파수보다 2배 이상으로 샘플링을 해야함.

## ■ 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.

# 청크(Chunk)와 채널(Channel)

## ■ 청크

- 일정한 크기를 가지는 사운드 데이터.

## ■ 채널

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



배경음악  
구현

# Pico2d.py 사운드 에러 수정

```
def open_canvas(w=int(800), h=int(600), sync=False):  
    global window, renderer  
    global canvas_width, canvas_height  
    global debug_font  
    global audio_on  
  
    canvas_width, canvas_height = w, h  
  
    # all the initialization needs to be check for working  
    SDL_Init(SDL_INIT_EVERYTHING)  
    IMG_Init(IMG_INIT_JPG | IMG_INIT_PNG | IMG_INIT_TIF | IMG_INIT_WEBP)  
    TTF_Init()  
  
    Mix_Init(MIX_INIT_MP3 | MIX_INIT_OGG)  
  
    ret = Mix_OpenAudio(44100, MIX_DEFAULT_FORMAT, MIX_DEFAULT_CHANNELS, 1024)  
    if -1 == ret:  
        print('WARNING: Audio functions are disabled due to speaker or sound problems')  
    else:  
        audio_on = True
```



```
class Grass:
    def __init__(self):
        self.image = load_image('grass.png')
        self.bgm = load_music('football.mp3')
        self.bgm.set_volume(64)
        self.bgm.repeat_play()
```

# Music 관련 function 들

- 음악 파일(mp3, ogg)의 로딩
  - load\_music(파일이름)
- 음악 연주 기능
  - repeat\_play() - 반복재생
  - play(n) - n 번 재생
  - set\_volume(v) - 소리 크기 0~128
  - get\_volume() - 현재 소리 크기
  - stop() - 중단
  - pause() - 일시정지
  - resume() - 재개



# 이펙트 사운드 구현





```
def update(frame_time):  
    boy.update(frame_time)  
    for ball in balls:  
        ball.update(frame_time)  
  
    for ball in balls:  
        if collide(boy, ball):  
            balls.remove(ball)  
            boy.eat(ball)  
  
    for ball in big_balls:  
        if collide(grass, ball):  
            ball.stop()
```



```
class Boy
    def __init__(self):
        self.x, self.y = 0, 90
        self.frame = random.randint(0, 7)
        self.life_time = 0.0
        self.total_frames = 0.0
        self.dir = 0
        self.state = self.RIGHT_STAND
        if Boy.image == None:
            Boy.image = load_image('animation_sheet.png')
        if Boy.eat_sound == None:
            Boy.eat_sound = load_wav('pickup.wav')
            Boy.eat_sound.set_volume(32)

    def eat(self, ball):
        self.eat_sound.play()
```

# Wav 관련 function 들

- 이펙트 파일(wav)의 로딩
  - load\_wav(파일이름)
- 이펙트 재생 기능
  - repeat\_play() - 반복재생
  - play(n) - n 번 재생
  - set\_volume(v) - 소리 크기 0~128
  - get\_volume() - 현재 소리 크기