

코드쌤: 지혜야, 슬기야! 우리가 지금까지 만든 게임은 눈으로 보는 즐거움이 있었지? 멋진 그림도 그리고, 귀여운 캐릭터도 등장시키고, 글자도 써넣었잖아. 그런데 뭔가 하나 빠진 것 같지 않니?

지혜: 음... 소려요! 캐릭터가 점프할 때 “뽕!” 하는 소리라던가, 배경음악 같은 거요!

슬기: 맞아요! 게임에서 효과음이랑 신나는 음악이 나오면 훨씬 더 재미있잖아요! 탕! 승! 팡! 이런 소려요!



코드쌤: 바로 그거야! 오늘은 우리 게임에 소리(효과음)와 음악(배경음악)을 넣어보는 마법을 배울 거란다. 눈뿐만 아니라 귀도 즐거운 게임을 만드는 거지! 마치 영화에 음향 효과를 넣는 것처럼!

(코드 조각 1: 효과음(Sound) 로드하고 재생하기 - 있는 코드 그대로!)

```
import pygame

pygame.init() # 소리 쓰려면 이것도 꼭 필요해!

WINDOW_WIDTH = 600
WINDOW_HEIGHT = 300

display_surface = pygame.display.set_mode((WINDOW_WIDTH, WINDOW_HEIGHT))
pygame.display.set_caption("쿵쾅쿵쾅! 사운드 추가하기!")

# --- 효과음 담당! pygame.mixer.Sound ---

# 1. 효과음 파일 불러오기: "이 소리 파일을 준비해줘!"
# (sound_1.wav, sound_2.wav 파일이 'data' 폴더 안에 있다고 가정!)
# 만약 파일이 없으면 여기서 에러가 날 수 있어요!
sound1 = pygame.mixer.Sound("data/sound_1.wav") # 뽕! 하는 소리?
sound2 = pygame.mixer.Sound("data/sound_2.wav") # 탕! 하는 소리?

# 2. 효과음 재생하기: "이 소리, 지금 바로 틀어줘!"
sound1.play() # sound1 재생!

# 잠깐 기다렸다가 다음 소리 재생 (소리가 겹치지 않게)
pygame.time.delay(1000) # 1000 밀리초 = 1초 동안 잠시 멈춤

sound2.play() # sound2 재생!

pygame.time.delay(2000) # 2초 동안 잠시 멈춤

# 3. 효과음 볼륨 조절하기: "이 소리, 조금만 작게 틀어줘!"
```

```
sound2.set_volume(0.1) # 0.0 (안들림) ~ 1.0 (최대 볼륨) 사이의 값. 0.1은 10% 크기!  
# print("두 번째 효과음 볼륨을 줄였어요!") # 주석 처리된 print문은 설명에서 생략할게요.  
sound2.play() # 줄인 볼륨으로 다시 재생!
```

(음악 재생 및 게임 루프 코드는 원래 코드 그대로 이어집니다)

지혜: pygame.mixer.Sound("data/sound_1.wav")요? mixer는 뭔가 섞는다는 뜻인데... 소리를 섞는 기계 같은 건가요?

코드쌤: 맞아, 지혜야! pygame.mixer 모듈은 Pygame에서 소리와 음악을 다루는 DJ 같은 역할을 해. 그중에서 pygame.mixer.Sound()는 **짧은 효과음 파일(보통 .wav 형식)**을 불러와서 Sound 객체로 만들어준단다. 마치 DJ가 효과음 버튼에 소리를 하나씩 저장해두는 것과 같지. 여기서도 data 폴더 안에 sound_1.wav 파일이 정확히 있어야 컴퓨터가 찾을 수 있어.

슬기: 그럼 sound1.play()는 그 저장된 효과음 버튼을 “꼭!” 누르는 거네요! “뽕!” 하고 소리가 나겠어요!

코드쌤: 정확해! play() 메소드를 호출하면 해당 Sound 객체에 저장된 소리가 재생된단다. 그리고 sound2.set_volume(0.1)은 뭐하는 걸까?

지혜: set_volume이니까 소리 크기를 정하는 거겠죠? 0.1이면... 아주 작게?

코드쌤: 빙고! set_volume()은 소리의 크기를 조절하는데, 0.0이면 소리가 안 나고, 1.0이면 최대 크기야. 0.1은 최대 볼륨의 10% 크기로 소리를 내라는 뜻이지. 그래서 같은 소리라도 상황에 따라 크게 또는 작게 들리게 할 수 있어. 코드에서는 두 번째 소리(sound2)의 볼륨을 줄이고 다시 재생했네.

코드쌤: 자, 효과음은 짧고 간결하게 “뽕!”, “탕!” 하는 느낌이라면, 게임 내내 분위기를 살려주는 긴 음악도 필요하겠지? 그건 **배경음악(Music)**이라고 해.

(코드 조각 2: 배경음악(Music) 로드하고 재생하기 - 있는 코드 그대로!)

```
# ... (이전 효과음 코드는 그대로) ...
```

```
# --- 배경음악 담당! pygame.mixer.music ---  
# (music.wav 파일이 'data' 폴더 안에 있다고 가정!)  
# 만약 파일이 없으면 여기서 에러가 날 수 있어요!  
pygame.mixer.music.load("data/music.wav") # 배경음악 파일 불러오기  
  
# 배경음악 재생하기: "이 음악, 계속 반복해서 들어줘!"  
# play(반복횟수, 시작시간)  
# 반복횟수: -1은 무한 반복! 0은 한 번 재생, 1은 두 번 재생 (한 번 반복)  
# 시작시간: 음악의 몇 초부터 재생할지 (보통 0.0부터 시작)  
pygame.mixer.music.play(-1, 0) # 원래 코드에서는 시작 시간이 0으로 되어 있네요!  
  
pygame.time.delay(5000) # 5초 동안 음악을 들어보자!
```

```

# 배경음악 멈추기
pygame.mixer.music.stop()

# --- 게임 루프 ---
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    # 게임 창을 업데이트합니다.
    pygame.display.update()

# pygame 모듈을 종료하고 게임 창을 닫습니다.
pygame.quit()

```

슬기: pygame.mixer.music.load("data/music.wav")요? 아까 Sound랑 비슷한데 music이라고 되어 있네요! 이건 긴 노래 파일을 위한 건가 봐요!

코드샐: 맞아, 슬기! pygame.mixer.music은 하나의 긴 배경음악 파일(주로 .mp3나 .wav 형식)을 다루기 위한 특별한 DJ 부스라고 생각하면 돼. load()로 음악 파일을 불러오고, play()로 재생하는 건 Sound랑 비슷하지? 여기서도 data 폴더 안에 music.wav 파일이 있어야 해.

지혜: pygame.mixer.music.play(-1, 0)에서 -1은 뭐예요? “무한 반복”이라고 주석에 써있어요!

코드샐: 예리한데, 지혜! music.play()의 첫 번째 인자는 **반복 횟수**를 나타내. -1이라고 쓰면 “음악이 끝까지 재생되면 다시 처음부터, 또다시 처음부터... 게임이 끝날 때까지 무한히 반복해줘!”라는 마법 주문이야. 만약 0을 쓰면 한 번만 재생되고, 1을 쓰면 총 두 번 재생(한 번 반복)된단다. 두 번째 인자 0은 음악의 맨 처음(0초)부터 재생하라는 뜻이고.

슬기: 와! 그럼 게임하는 내내 신나는 음악이 계속 나오겠네요! 그리고 pygame.mixer.music.stop()은 음악 끄는 거고요!

코드샐: 정확해! stop()은 재생 중인 배경음악을 멈추는 버튼이야. 게임이 끝나거나, 다른 장면으로 넘어갈 때 배경음악을 바꾸거나 멈출 필요가 있겠지? 그때 사용한다. 지금 코드에서는 음악을 5초 동안 재생하고 멈추도록 되어 있네.

지혜: 선생님, 그럼 효과음이랑 배경음악은 동시에 나올 수 있는 거예요? DJ가 효과음 버튼도 누르고, 배경음악도 틀어놓는 것처럼요?

코드샐: 물론이지! Pygame의 mixer는 똑똑해서 여러 개의 효과음과 하나의 배경음악을 동시에 재생할 수 있어. 그래서 배경음악이 잔잔하게 흐르는 중에 캐릭터가 점프하면 “뿡!” 하는 효과음이 함께 들리게 할 수 있는 거야. 지금 이 코드에서는 효과음들이 먼저 순차적으로 재생되고, 그 후에 배경음악이 재생되고 멈추는 순서로 되어 있지만, 게임 루프 안에서 필요에 따라 효과음을 재생하면 배경음악과 함께 들릴 수 있단다.

슬기: 우와! 이제 제 게임에 캐릭터가 움직일 때마다 발자국 소리 “타닥타닥” 넣고, 점프할 땐 “슌!”, 아이템 먹으면 “반짝!” 하는 소리도 넣을 수 있겠어요! 배경음악은 용감한 모험 음악으로 하고요!

코드샴: 아주 멋진 생각이야, 슬기! 소리는 게임의 분위기를 만들고, 플레이어에게 중요한 정보를 전달하며, 게임을 훨씬 더 몰입감 있게 만들어준단다. 오늘 너희는 그 소리의 마법을 다루는 방법을 배운 거야!

지혜: 효과음은 `pygame.mixer.Sound()`, 배경음악은 `pygame.mixer.music`! 그리고 둘 다 파일 경로가 중요하고, `play()`로 재생하고! 기억할게요!

코드샴: 훌륭해! 오늘 배운 내용으로 다양한 소리 파일(.wav 추천!)을 구해서 게임에 넣어보고, 언제 어떤 소리가 나면 재미있을지 상상하면서 코드를 짜보렴. 너희들의 게임이 한층 더 생생해질 거야!