BitWith Bite

PYGAME MASTERY WORKSHEET

Sound System

Module 10 of 20 · Level ★☆☆ · Time: 30 min

SCORE ___ / 20

Name _____ Class _____ Date _____

AFTER THIS WORKSHEET YOU CAN

- ✓ Play music vs sound effects correctly
- ✓ Control volume for music and SFX
- ✓ Use fade in/out for smooth transitions
- ✓ Explain what channels are for

QUICK RECAP

pygame.mixer.music streams one background track at a time, while pygame.mixer.Sound loads short effects fully into memory so many can overlap via channels — both support independent volume control and smooth fade in/out.

Section A · Concept Check • BEGINNER 4 × 1 = 4

1 Which API is correct for looping background music?

- ◯ A. pygame.mixer.Sound
- ◯ B. pygame.mixer.music
- ◯ C. pygame.mixer.Channel ◯ D. pygame.draw

2 How many Sounds can overlap by default?

- ◯ A. Only 1
- ◯ B. Up to 8 (raiseable with set_num_channels)
- ◯ C. Unlimited always ◯ D. Exactly 2

3 What does loops=-1 mean in music.play(loops=-1)?

- ◯ A. Play once ◯ B. Loop forever
- ◯ C. Play in reverse ◯ D. An error

4 What does music.fadeout(800) do?

- ◯ A. Instantly stops the music
- ◯ B. Fades volume to silence over 800ms, then stops
- ◯ C. Raises the volume ◯ D. Loads a new track

Section B · Problem Solving • INTERMEDIATE 2 + 3×3 = 11

5 Volume is set with a value from 0.0 to _____.

6 A _____ is what a Sound automatically plays on when .play() is called.

7 Write the two lines to load and play a jump sound effect at 80% volume.

8 Why would a game sound "silent" during a busy moment with many effects at once?

9 Why load several variant Sound files for something like a hit/hurt sound?

Section C · Challenge • CHALLENGE 5

10 Explain why background music uses mixer.music instead of mixer.Sound, focusing on memory usage and how many tracks can play at once.

🧠 Reflection — the most useful thing I learned:

A ___/4 B ___/11 C ___/5 **Total ___/20**

Teacher's Signature

Parent's Signature

✂ ANSWER KEY — FOLD OR CUT BEFORE HANDING OUT

1-B 2-B 3-B 4-B | 5 1.0 6 Channel 7 = jump_sfx = pygame.mixer.Sound("jump.wav"); jump_sfx.set_volume(0.8); jump_sfx.play() 8 = the game has run out of free mixer channels (default 8), so new Sound.play() calls have nothing left to play on 9 = randomly picking among a few variants avoids the repetitive "same sound every time" feeling, since Pygame has no built-in pitch-shift | 10 = mixer.music streams from disk rather than loading the whole file into RAM, using far less memory for a long track, but only ONE music stream can play at a time; mixer.Sound loads fully into memory for instant, overlapping playback of short effects across multiple channels simultaneously