BitWith Bite

PYGAME MASTERY WORKSHEET

Working with Images

Module 4 of 20 · Level ★☆☆ · Time: 30 min

SCORE ___ / 20

Name _____ Class _____ Date _____

AFTER THIS WORKSHEET YOU CAN

✓ Load and convert images correctly ✓ Scale, rotate and flip sprites ✓ Manage transparency and alpha

✓ Organize an assets folder efficiently

QUICK RECAP

Images are loaded once into Surfaces with pygame.image.load(), converted for fast blitting, then transformed (scale/rotate/flip) via pygame.transform — which always returns a new Surface rather than modifying the original.

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

1 Why call .convert() or .convert_alpha() after loading?

- ◯ A. It's optional and does nothing
- ◯ B. It matches pixel format to the display for faster blitting
- ◯ C. It compresses the file
- ◯ D. It resizes the image

2 Which format supports real per-pixel transparency?

◯ A. JPG ◯ B. PNG ◯ C. BMP ◯ D. All equally

3 What does pygame.transform.rotate() return?

- ◯ A. Nothing, it mutates in place
- ◯ B. A brand-new Surface ◯ C. A file path
- ◯ D. An angle value

4 What does img.set_alpha(128) do?

- ◯ A. Rotates the image
- ◯ B. Makes the whole surface roughly 50% transparent
- ◯ C. Scales it to 128px ◯ D. Crops 128 pixels

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

5 pygame.transform._____(img, True, False) mirrors an image horizontally.

6 Loading images inside the game loop is bad because it's a slow _____ operation.

7 What's the difference between .convert() and .convert_alpha()?

8 Why should rotated animation frames be pre-generated instead of rotated live every frame?

9 Describe a good pattern for loading and storing all of a game's images at startup.

Section C · Challenge • CHALLENGE 5

10 You need a player sprite to face left and right. Explain two different ways to get a mirrored version, and which is more efficient at runtime.

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 flip 6 disk-read 7 = .convert() is for opaque images; .convert_alpha() preserves per-pixel transparency for images with alpha channels 8 = rotating every frame is CPU-expensive; pre-generating a list of rotated frames once trades a bit of memory for much better runtime speed 9 = load every image once at startup into a dictionary keyed by name, then reference that dictionary everywhere instead of reloading from disk | 10 = (a) load a separate pre-drawn left-facing image, or (b) use pygame.transform.flip(img, True, False) on the right-facing image once and cache the result — option (b) is more efficient since it avoids managing double the art assets and only flips once, not every frame