



PYGAME MASTERY WORKSHEET



Collision Detection

Module 9 of 20 • Level ★★★ • Time: 35 min

SCORE

___ / 20

Name _____ Class _____ Date _____

AFTER THIS WORKSHEET YOU CAN

- ✓ Detect Rect and circle collisions
- ✓ Explain hitboxes vs hurtboxes
- ✓ Use trigger zones for events
- ✓ Explain collision optimization basics

QUICK RECAP

Rect collision (collidect) is the fast default; circle collision compares center distance to summed radii; mask collision is pixel-perfect but slow. Hitboxes deal damage, hurtboxes receive it, and trigger zones detect overlap without physical response.



Section A • Concept Check

• INTERMEDIATE

4 × 1 = 4

1 Which is the fastest, most common collision check?

- ☐ A. Mask collision
- ☐ B. Rect collision (collidect)
- ☐ C. Pixel array scan ☐ D. Ray casting

2 How is circle collision detected?

- ☐ A. Comparing bounding rects
- ☐ B. Comparing distance between centers to the sum of radii
- ☐ C. Comparing pixel alpha values
- ☐ D. Comparing colors

3 What does a hurtbox represent?

- ☐ A. The area that deals damage
- ☐ B. The vulnerable area that receives damage
- ☐ C. The player's score ☐ D. The screen boundary

4 What is a trigger zone?

- ☐ A. A Rect that detects overlap without physical collision response
- ☐ B. A type of sprite animation
- ☐ C. A sound effect ☐ D. A camera setting



Section B • Problem Solving

• INTERMEDIATE

2 + 3×3 = 11

5 pygame.sprite.collide_mask checks collision at the _____ level.

6 A hitbox _____ damage; a hurtbox receives it.

7 Write the if-condition to check if player.rect overlaps enemy.rect.

8 Why is mask collision slower than rect collision?

9 Describe one way to optimize collision checks when there are hundreds of objects.



Section C · Challenge

• CHALLENGE

5

10 A circular coin sprite uses rect collision and feels "wrong" — the player can collect it while visibly not touching it. Explain why this happens and how switching to circle collision fixes it.

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-A | 5 pixel 6 deals 7 = `if player.rect.colliderect(enemy.rect):` 8 = it compares actual non-transparent pixels between two images every check, far more computation than comparing four rectangle edge values 9 = only check objects that are near each other (e.g. grid/spatial partitioning or a broad distance check) instead of comparing every object against every other object each frame | 10 = a rectangular bounding box includes the empty corner space around a round sprite, so the player can trigger a "collision" while only touching that empty corner area; circle collision compares actual center-to-center distance against the combined radii, matching the coin's real round shape

bitwithbite.com

Free STEM Learning Resources · Pygame Mastery · Collision Detection