



PYGAME MASTERY WORKSHEET



Camera System

Module 15 of 20 • Level ★★★ • Time: 35 min

SCORE

--- / 20

Name _____ Class _____ Date _____

AFTER THIS WORKSHEET YOU CAN

- ✓ Build a camera offset system ✓ Implement smooth camera follow ✓ Add screen shake and zoom
- ✓ Build parallax backgrounds

QUICK RECAP

A camera is just an (x,y) offset subtracted from world positions before drawing. Smooth follow lerps toward the target, screen shake adds temporary random offset, zoom scales a rendered surface, and parallax scrolls background layers at a fraction of the camera's speed.



Section A • Concept Check

• INTERMEDIATE

4 × 1 = 4

- | | |
|--|--|
| <p>1 What fundamentally is a 2D camera in Pygame?</p> <ul style="list-style-type: none"> ☐ A. A built-in pygame.Camera class ☐ B. An offset subtracted from world positions before drawing ☐ C. A special sprite type ☐ D. A hardware requirement | <p>2 Why lerp the camera instead of snapping instantly?</p> <ul style="list-style-type: none"> ☐ A. Snapping is impossible ☐ B. Lerp eases the camera gradually, feeling smoother ☐ C. It's required for collisions ☐ D. It's faster |
| <p>3 How does parallax create the illusion of depth?</p> <ul style="list-style-type: none"> ☐ A. Background layers scroll slower than the foreground ☐ B. Using 3D models ☐ C. Everything scrolls at the same speed ☐ D. Disabling the camera | <p>4 How can you achieve a zoom effect in Pygame?</p> <ul style="list-style-type: none"> ☐ A. Render to a smaller surface, then scale it to fill the window ☐ B. A built-in zoom() function ☐ C. It's impossible in 2D ☐ D. Change the frame rate |



Section B • Problem Solving

• INTERMEDIATE

2 + 3×3 = 11

- 5** Screen shake adds a temporary _____ offset to the camera.
- 6** A depth factor of 0.2 means a background layer moves at _____% of the camera's movement.
- 7** Write the apply() method that converts a world position to a screen position using camera.offset.
- _____
- 8** Why should a mini-map use a smaller scale rather than the same camera offset math as the main view?
- _____
- 9** Describe why screen shake magnitude usually decays over its duration rather than staying constant.
- _____



Section C · Challenge

• CHALLENGE

5

10 Your game has 3 parallax layers (mountains depth 0.2, trees depth 0.5, gameplay depth 1.0) plus screen shake on hit. Explain how shake should combine with the camera offset so ALL layers shake together realistically, not just the gameplay layer.

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-A 4-A | 5 random 6 20 7 = `def apply(self, pos): return pos - self.offset` 8 = the mini-map needs to fit the whole level in a small corner of the screen, so positions must be scaled down by a much larger factor than the 1:1 main camera view uses 9 = decaying magnitude makes the shake feel like it's settling down naturally after an impact, rather than an abrupt, jarring on/off toggle | 10 = compute the shake offset once per frame (a small random vector), then ADD it to `camera.offset` itself (not to each layer separately) — since every layer's draw position is derived from the same shared `camera.offset`, adding shake to that one value automatically shakes every layer together in sync