



PYGAME MASTERY WORKSHEET

Movement System

Module 8 of 20 • Level ★★★ • Time: 40 min

SCORE

___ / 20

Name _____ Class _____ Date _____

AFTER THIS WORKSHEET YOU CAN

- ✓ Apply velocity/acceleration each frame
- ✓ Implement gravity and jumping
- ✓ Add double jump and dash
- ✓ Clamp movement to screen boundaries

QUICK RECAP

Movement feel comes from velocity, acceleration and friction applied every frame, plus gravity pulling objects down and a negative-velocity jump that gravity naturally arcs back down — with double jump and dash layered on top.

Section A • Concept Check • INTERMEDIATE $4 \times 1 = 4$

- 1 What is "speed" relative to a velocity vector?
 - ☐ A. The x-component only
 - ☐ B. Its magnitude, ignoring direction
 - ☐ C. Same as acceleration
 - ☐ D. Always constant
- 2 How is double jump typically implemented?
 - ☐ A. A separate physics engine
 - ☐ B. A jumps_left counter reset only when grounded
 - ☐ C. Doubling gravity
 - ☐ D. A third-party library
- 3 Why is jump velocity negative in Pygame?
 - ☐ A. It's a typo
 - ☐ B. Negative Y moves up the screen
 - ☐ C. Negative numbers move faster
 - ☐ D. It disables gravity
- 4 What does friction do with no movement input?
 - ☐ A. Increases velocity
 - ☐ B. Gradually decays velocity toward zero
 - ☐ C. Instantly stops the character
 - ☐ D. Doubles gravity

Section B • Problem Solving • INTERMEDIATE $2 + 3 \times 3 = 11$

- 5 Position updates each frame as `pos += _____ * dt`.
- 6 Gravity is applied as a constant _____ added to vertical velocity every frame.
- 7 Write the pattern for resetting jumps_left only when the player is on_ground.

- 8 What's the purpose of a cooldown timer on a dash ability?

- 9 Why should screen boundary clamping happen after movement is applied, not before?



Section C · Challenge

• CHALLENGE

5

10 Explain why acceleration-based movement (ramping up to max speed) feels better than instantly snapping velocity to max speed.

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 velocity (vel) 6 acceleration 7 = if on_ground: jumps_left = MAX_JUMPS 8 = it prevents the dash from being spammed repeatedly, forcing a delay between uses 9 = movement must be applied first so the new position exists to be clamped; clamping first would use the stale old position and let the object briefly overshoot before being corrected | 10 = instant snapping feels robotic/twitchy because there's no sense of weight or momentum; gradual acceleration mimics real physical objects building up speed, which reads as more natural and controllable