

**BitWithBite**

PYGAME MASTERY WORKSHEET

**Enemy AI**

Module 14 of 20 • Level ★★★ • Time: 40 min

SCORE

___ / 20

Name _____ Class _____ Date _____

AFTER THIS WORKSHEET YOU CAN

- ✓ Implement patrol movement
- ✓ Implement line-of-sight chasing
- ✓ Build a finite state machine
- ✓ Design boss phase transitions

QUICK RECAP

Enemy AI is built from simple behaviors — patrol between two points, chase when within a detection radius, attack when close — organized by a finite state machine that keeps the enemy in exactly one clearly-defined state at a time.

**Section A · Concept Check**

• ADVANCED

4 × 1 = 4

1 What is a finite state machine, for an enemy?

- ☐ A. A hardware chip
- ☐ B. Always in exactly one named state, with clear transition rules
- ☐ C. A random number generator
- ☐ D. A collision check

3 Why does straight-line chasing break around obstacles?

- ☐ A. It doesn't, it always works
- ☐ B. It moves directly toward the player, ignoring walls
- ☐ C. Pygame disables it automatically
- ☐ D. It requires pathfinding by default

2 What does `.normalize()` give you on a direction vector?

- ☐ A. A unit vector (length 1) in that direction
- ☐ B. The distance between two points
- ☐ C. A random direction
- ☐ D. The player's health

4 What commonly triggers a boss's next phase?

- ☐ A. A fixed timer only
- ☐ B. Remaining health dropping below a threshold
- ☐ C. The player's score
- ☐ D. Frame rate dropping

**Section B · Problem Solving**

• INTERMEDIATE

2 + 3×3 = 11

5 A patroller reverses _____ when it reaches either endpoint.**6** The FSM state that fires when distance to player is very small is usually _____.**7** Write the condition that switches an enemy from "chase" back to "patrol" when the player gets far enough away.**8** Describe a cheaper alternative to full A* pathfinding for many 2D games.**9** What data does a boss typically track to decide when to attack again?

**Section C · Challenge**

• CHALLENGE

5

10 Design a 4-state FSM (name the states) for a guard enemy that patrols, notices the player, chases, attacks when close, and gives up if the player escapes far enough. Describe the transition condition between each pair of states.

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-A 3-B 4-B | 5 direction 6 attack 7 = `elif dist > 300: state = "patrol"` 8 = hand-placed waypoint graphs or simple wall-following heuristics, which solve most practical 2D pathing needs without full A* complexity 9 = an `attack_timer/cooldown` that counts down each frame, triggering the next attack once it reaches zero and then resetting | 10 = states: patrol (walks back and forth) → chase (triggered when distance < detection radius) → attack (triggered when distance < a close attack range) → back to chase if the player backs off past attack range but stays within detection range → back to patrol if distance exceeds detection radius entirely, meaning the player escaped and the enemy gives up