

**BitWithBite**

PYGAME MASTERY WORKSHEET

User Interface

Module 11 of 20 • Level ★★★ • Time: 35 min

SCORE

___ / 20

Name _____ Class _____ Date _____

AFTER THIS WORKSHEET YOU CAN

- ✓ Render text with `pygame.font` ✓ Build clickable buttons ✓ Implement a pause screen overlay
- ✓ Draw an HUD health bar

QUICK RECAP

Text is rendered into a Surface and blitted like an image. Buttons check `Rect.collidepoint()` against mouse clicks. Pause screens freeze gameplay updates while drawing a translucent overlay, and the HUD (health/score/timer) is drawn in fixed screen coordinates every frame.

**Section A - Concept Check**

• INTERMEDIATE

4 × 1 = 4

- 1 How does Pygame put text on screen?
 - ☐ A. `pygame.draw.text()` draws it directly
 - ☐ B. A Font renders text into a Surface, then it's blitted
 - ☐ C. Text is a window property
 - ☐ D. Only images can show text
- 2 How do you detect a button click?
 - ☐ A. Check the button's `Rect.collidepoint()` on `MOUSEBUTTONDOWN`
 - ☐ B. Pygame has a built-in Button widget
 - ☐ C. Buttons only work via keyboard
 - ☐ D. It's not possible in Pygame
- 3 How is a pause screen commonly implemented?
 - ☐ A. Stop gameplay updates, keep drawing an overlay on the last frame
 - ☐ B. Restart the whole game
 - ☐ C. Delete all sprites ☐ D. It isn't possible
- 4 Does the HUD scroll with the camera?
 - ☐ A. Yes, always
 - ☐ B. No, it's drawn in fixed screen coordinates
 - ☐ C. Only the score scrolls
 - ☐ D. It depends on FPS

**Section B - Problem Solving**

• INTERMEDIATE

2 + 3×3 = 11

- 5 `pygame.font.Font(None, 48).render(text, True, color)` — the True argument enables _____
- 6 A health bar's foreground width scales by `current_hp` _____ `max_hp`.
- 7 Write the check for whether the mouse click at `event.pos` landed inside a button's rect.

- 8 Describe the 3 steps to draw a health bar (background rect, foreground rect, outline).

- 9 Why should text surfaces be cached instead of re-rendered every frame when the text doesn't change?



Section C · Challenge

• CHALLENGE

5

10 Explain why a pause overlay Surface needs pygame.SRCALPHA when created, and what would happen visually if you forgot 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-A 3-A 4-B | 5 anti-aliasing (smooth edges) 6 / (divided by) 7 = if
button.rect.collidepoint(event.pos): 8 = draw a full-width background rect in a dark color, draw a
foreground rect scaled to width*(current_hp/max_hp) in a bright color on top, then draw a border
outline rect around both 9 = re-rendering text every frame is wasted CPU work since the
Font.render() call is relatively expensive; cache the Surface and only re-render when the text
actually changes | 10 = pygame.SRCALPHA allows the surface to store per-pixel transparency, which
is required for a translucent black overlay; without it, the fill color would render fully opaque,
completely blacking out the game screen behind the "Paused" text instead of dimly showing it through

bitwithbite.com

Free STEM Learning Resources · Pygame Mastery · User Interface