

**BitWithBite**

PYGAME MASTERY WORKSHEET

# Final Capstone Planner

Module 20 of 20 • Level ★★★ • Time: 45 min

SCORE

--- / 20

Name \_\_\_\_\_ Class \_\_\_\_\_ Date \_\_\_\_\_

**AFTER THIS WORKSHEET YOU CAN**

- ✓ Write a game design document
- ✓ Plan a testing pass
- ✓ Package a game with PyInstaller
- ✓ Prepare a portfolio presentation

**QUICK RECAP**

The capstone ties together every module: plan with a design document, build core mechanics on the Module 18 Scene skeleton, integrate UI and sound, test deliberately, optimize, package with PyInstaller, and present the finished game with a README and gameplay GIF.

**Section A · Concept Check**

• ADVANCED

4 × 1 = 4

- |                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>1</b> Why write a game design document before coding?</p> <ul style="list-style-type: none"> <li><input type="radio"/> A. It's not necessary</li> <li><input type="radio"/> B. It defines concept/mechanics/controls upfront, avoiding scope creep</li> <li><input type="radio"/> C. Pygame requires a .gdd file</li> <li><input type="radio"/> D. It replaces testing</li> </ul> | <p><b>2</b> What should testing focus on besides the happy path?</p> <ul style="list-style-type: none"> <li><input type="radio"/> A. Only the main menu</li> <li><input type="radio"/> B. Edge cases like rapid input and boundary collisions</li> <li><input type="radio"/> C. Only sound effects</li> <li><input type="radio"/> D. Nothing else is needed</li> </ul> |
| <p><b>3</b> What does the capstone's starter skeleton reuse from Module 18?</p> <ul style="list-style-type: none"> <li><input type="radio"/> A. Sound effect files</li> <li><input type="radio"/> B. The Scene/SceneManager pattern</li> <li><input type="radio"/> C. Tile map data</li> <li><input type="radio"/> D. Font files</li> </ul>                                             | <p><b>4</b> What should a portfolio presentation include besides the executable?</p> <ul style="list-style-type: none"> <li><input type="radio"/> A. Nothing else</li> <li><input type="radio"/> B. A README and a gameplay GIF/screenshots</li> <li><input type="radio"/> C. Only the source .py file</li> <li><input type="radio"/> D. A paid license</li> </ul>     |

**Section B · Problem Solving**

• INTERMEDIATE

2 + 3×3 = 11

- 5** A design document should define win/lose conditions and player \_\_\_\_\_
- 6** Assets should be organized into separate folders like assets/images and assets/\_\_\_\_\_
- 7** List the 4 core modules (topics) whose techniques must combine to form a playable capstone gameplay loop.
- \_\_\_\_\_
- 8** Why should the design document be written before writing any code?
- \_\_\_\_\_
- 9** What is the PyInstaller command to package a finished capstone game as a single windowed executable?
- \_\_\_\_\_



## Section C · Challenge

• CHALLENGE

5

10 Outline your own capstone game in 4-5 sentences: what genre, what core mechanic, what makes it winnable/losable, and which 3 modules from this course you'll lean on most.

Reflection — the most useful thing I learned across the whole course: \_\_\_\_\_

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 controls   6 sounds   7 = e.g. Sprites, Movement System, Collision Detection, and Animation (or a similar reasonable combination from earlier modules)   8 = it forces clear scope decisions upfront (what the game actually is) before any code is written, preventing wasted work from an unclear or constantly-changing plan   9 = pyinstaller --onefile --windowed game.py | 10 = open-ended — grade based on whether the student names a genre, a clear core mechanic, a specific win/lose condition, and identifies 3 real modules (e.g. Movement, Collision, UI) that plausibly support their described game

[bitwithbite.com](https://bitwithbite.com)

Free STEM Learning Resources · Pygame Mastery · Final Capstone