

Class 5 Browser Game Exercise

Intro to Vibe Coding Name _____

Goal: make a playable browser game, keep the Python original, and explain one code change. Copy only text between COPY FROM HERE and STOP HERE.

1 Inspect and plan

Open your Python project in VS Code and your AI tool. If you need a starting project, ask Mr. Meyer. For a new game, describe a player, goal, obstacle, and ending.

PROMPT 1: INSPECT THE ORIGINAL

COPY FROM HERE

Read my existing Python game and explain its goal, controls, scoring or damage rules, endings, and distinctive visuals. Identify the files and settings that create those behaviors. Suggest a browser port using JavaScript and Canvas. Do not edit any files yet.

STOP HERE

Goal _____

Controls _____

Scoring or damage _____

Endings and restart _____

Visual or special feature to preserve _____

2 Build the browser version

PROMPT 2: BUILD THE PORT

COPY FROM HERE

Use the plan above to build a browser version in a new web subfolder. Leave my Python files unchanged. Use index.html, styles.css, and game.js with JavaScript and Canvas. Keep the original controls, rules, artwork, animations, and special features. Reuse local assets where possible. For a new game, use the design I supplied and one complete playable arena. Include clear instructions, start, pause, endings, and restart. Pause when the tab loses focus. Keep movement consistent at different frame rates and stop game controls from scrolling the page. Make the layout fit the browser window. Use local assets and no frameworks, accounts, or backend services. Put editable settings near the top of game.js. Start a local web server serving only web and give me the browser URL plus exact restart commands for my computer. Explain any feature you could not preserve. Show me the game loop, a score or state update, and one setting I can edit. Check the full round, then ask me to test the controls myself.

STOP HERE

3 Compare and investigate

Play both versions. Mark each check OK or FIX. For a new game, compare with your design. AI can help explain code, but test the real controls yourself.

- Movement and visuals: controls, artwork, and reactions match. _____
- Score and damage: the rules and hit protection work. _____
- Endings and restart: win/lose works and round values reset. _____
- Pause and tab change: movement and timer stop, then resume. _____
- Resize and keyboard: play stays usable and arrows do not scroll. _____
- Relaunch: your saved files reopen without missing assets. _____

ONE DIRECT CODE EXPERIMENT

Open web/game.js. Find the loop and a state update. Change one setting yourself, save, refresh the browser, and test. Use a hard refresh if an old version remains.

Setting and old/new values _____

My prediction _____

What happened _____

PROMPT 3: FIX ONE DIFFERENCE

Replace the bracketed words with your observation before sending.

COPY FROM HERE

When I [action], I expect [result], but [actual result] happens. Explain the cause and fix this one behavior. Keep the other game rules and visuals intact. Tell me how to test the fix and check restart.

STOP HERE

STRETCH: CHOOSE ONE AFTER THE CORE WORKS

- Touch controls: add buttons and test them in a mobile browser view.
- Personal best: save a score in this browser; test refresh and reset.
- Game presentation: improve resizing, instructions, or a title screen.

My browser URL / launch command _____

Exit: show your game, one code edit, and a feature you checked. Save all files. Your localhost address works on your own computer; public hosting comes next.