Side quest: Week 3

Name: Min Htet Naing, dmnaing (21008098)

Project/Assignment Decisions (2–4 sentences)

I chose a simple Solo Levelling-inspired training story where the player decides which exercise to do each day (push-ups, running, or resting), and those choices branch the story into different scenes. I decided to track health and level across states, so the player must balance training and recovery instead of always picking the same option. I kept the visuals lightweight (buttons, HUD, simple effects) so the focus stayed on the requirement: a multi-file, multi-state decision tree with endings unlocked by a player stat.

GenAI Documentation (Used)

Date Used: February 2, 2026
Tool Disclosure: ChatGPT (GPT-5.2 Thinking)

Purpose of Use:
I used ChatGPT to help structure the project as a clean state machine across multiple files and to draft a simple branching decision tree that meets the Week 3 requirements (multiple game states + bonus stat tracking and endings). I also used it to get suggestions for small UI/visual improvements (HUD bars, background effects) while keeping the assignment scope manageable.

Prompt(s) Used:

- “Teach me step-by-step what to add/replace and how to make new files using my existing multi-file project template.”
- “My game becomes blank after level 4. What happened, and how do I fix it?”
- “Improve the CSS so the page looks cleaner.”

- “Make the game less text-only and more interesting without adding heavy assets.”

Summary of Interaction

ChatGPT suggested a state-based structure (start/instructions/hub/scenes/gate/ending) and recommended tracking Health + Level to control branching and unlock multiple endings. It also provided a file organization plan and basic UI ideas (buttons + HUD) to keep the project readable across multiple files. When a blank-screen bug happened, it pointed out that a missing state handler or missing screen function (e.g., win screen) can crash the draw loop and suggested using the browser console to confirm the error.

Human Decision Point(s)

- I decided the stat system would be **Health + Level** (not just one stat) because it better supports “training vs. rest” decision-making.
- I tuned the numeric values (health loss/gain, day count, and ending thresholds) after testing to make outcomes fair and repayable.
- I simplified visuals to keep performance stable and avoid needing external images/audio, focusing on meeting the rubric (multi-file + branching + stat-based endings).

Integrity & Verification Note

I verified the game by clicking through every branch, intentionally choosing extreme patterns (all training vs all rest), and confirming that stats persist correctly across scenes. I also tested that every state has a valid draw function and that script loading order in index.html does not break after changes. I checked the browser console when debugging the blank screen to confirm the exact runtime error and then fixed it.

Scope of GenAI Use

GenAI supported planning the decision tree, suggesting code structure patterns, and identifying common bugs. I assembled the final project, integrated the code into my existing file template, adjusted parameters, and tested/verified the final behavior myself.

Limitations or Misfires

Some suggested “extra effects” made the project feel too busy or increased the chance of bugs (script order/state mismatches), so I kept only lightweight enhancements. Also, an early version caused a blank screen due to entering a state without a matching screen handler, which I corrected by ensuring every state has a corresponding draw function and required files are loaded.