Process & Decision Documentation

I employed ChatGPT-5.2 to support my planning, debugging, and simplification process for this side quest. I began by outlining the assignment requirements and my initial idea for an interactive story directly in my code editor, then used GenAI to help reason through logic issues, file organization, and scope decisions. My process followed a simple human-AI collaboration: first, I identified what I wanted the sketch to accomplish (multiple states, player choice, and an outcome); second, I asked targeted questions to troubleshoot errors and clarify state management; third, I iterated based on testing results and my own understanding of the code; and finally, I refined the sketch to ensure it aligned with the expectations and constraints of a side quest.

Name: Jolin Li

Role(s): Side Quest 3

Primary responsibility for this work:
Designing the story logic and implementing interaction using JavaScript and p5.js.

Goal of Work Session:
My goal was to create a simple interactive story that:

- progressed through multiple screens
- allowed player choices
- tracked a basic stat across scenes
- led to a win or lose ending based on player decisions

Briefly describe what you were trying to accomplish during this phase of the assignment.

During this phase of the assignment, I was focused on building a functioning interactive story in JavaScript that used multiple screens and simple branching logic. My goal was to ensure player choices triggered clear state changes, tracked a basic stat across scenes, and led to a reliable win or lose outcome while keeping the scope manageable for a side quest.

GenAI Documentation

If GenAI was used (keep each response as brief as possible):

Date Used
February 2nd, 2026

Tool Disclosure
ChatGPT 5.2

Purpose of Use
I used GenAI to help debug loading issues, simplify my logic when my project became unstable, and reframe my story system into a smaller, more manageable structure.

Summary of Interaction
GenAI helped identify file-loading issues (such as incorrect script paths and conflicts between files), suggested a simplified story structure, and provided example logic for tracking a single stat across scenes.

Human Decision Point(s)
I coded the screens and I chose to reduce the number of scenes and stats after, deciding that clarity and stability were more important than complexity for this assignment.

Integrity & Verification Note
I checked GenAI output for accuracy, bias, appropriateness, and alignment with course expectations by carefully reviewing all suggestions against the assignment instructions and my existing code. I validated technical advice through direct testing in the browser and console, and questioned the tool's reasoning by asking it to explain its decision logic.

Scope of GenAI Use
The story content, final structure, and integration of files were completed by me. GenAI was used only for troubleshooting, simplification, and guidance.

Summary of Process (Human + Tool)

- Set up project files using the provided p5.js structure

- Attempted a more complex branching story

- Encountered loading and logic issues

- Used GenAI to debug and identify conflicts

- Simplified the story to two scenes and one stat

- Tested win/lose logic and finalized visuals

- Describe one or two key decisions you made:

I decided to simplify the interactive story I created into a small decision tree with two scenes and one tracked stat to keep the logic stable and manageable.

Verification & Judgement

- Tested the sketch multiple times in the browser to confirm choices correctly updated story state
- Checked console output to identify and fix logic or state errors
- Re-reviewed assignment requirements to ensure all criteria were met
- Simplified or adjusted logic when testing revealed instability

Appendix

User: My game isn’t loading when I click start.
ChatGPT: This may be caused by a missing function reference or script loading order. Check whether the reset function exists globally and confirm p5.js is loading from the correct directory.

User: Live Server says it can’t reload without a body tag.
ChatGPT: This indicates your HTML file is missing proper `<head>` and `<body>` tags. Add a standard HTML structure and reload using Live Server.

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Win or Lose</title>

    <link rel="stylesheet" href="style.css" />
  </head>

  <body>
    <!-- Load p5 FIRST (correct path) -->
    <script src="libraries/p5.min.js"></script>
    <!-- optional: only if you use sound -->
    <!-- <script src="libraries/p5.sound.min.js"></script> -->

    <!-- Your game files (order matters) -->
    <script src="main.js"></script>
    <script src="storyData.js"></script>

    <script src="start.js"></script>
    <script src="instructions.js"></script>
    <script src="game.js"></script>
    <script src="win.js"></script>
    <script src="lose.js"></script>

    <!-- IMPORTANT: do NOT include sketch.js -->
  </body>
</html>
```

User: Can we simplify this more?

```
let karma = 0;
let health = 3;

function applyChoice(choice) {
  karma += choice.karma;
  health += choice.health;

  if (health <= 0) currentScreen = "lose";
  else if (karma >= 2) currentScreen = "win";
  else currentNodeId = choice.next;
}
```

ChatGPT: Yes. You can simplify this by removing the extra stat and reducing the logic to a single tracked variable. Instead of updating both `karma` and `health`, keep only `karma` and use it to decide the final outcome when the story reaches its end. This keeps the logic clearer and avoids unnecessary branching while still showing player choice and state change.

```
let karma = 0;

function pick(choice) {
  karma += choice.karma;

  if (choice.next === "END") {
    if (karma >= 1) {
      currentScreen = "win";

    } else {

      currentScreen = "lose";

    }

  } else {

    currentNodeId = choice.next;

  }

}
```