Side Quest 3 - Process & Decision Documentation

Name: Amy Truong
Date Used: Monday, feb 2nd
Tool disclosure: CHATGPT 5.2
Purpose of use: Debugging

Summary of process:
I used chatgpt to help me to upload my code in VS code to github. I followed its directions which did not end up successful. I had to iterate different prompts, describing that it didn't work and copying the output of my error code that occurred after I followed the instructions it gave me.

Decision points & tradeoffs:
One key decision I made was how to resolve ongoing GitHub upload and deployment errors. I initially followed ChatGPT's debugging instructions, but when those steps did not resolve the issue, I redirected the process by manually reviewing my repository setup, branch state, and GitHub Pages configuration. This choice was made to avoid repeating the same troubleshooting steps and to regain control over the debugging process. I also chose to limit GenAI use strictly to technical debugging rather than creative decisions, ensuring that all narrative and scene design choices remained my own.

Verification & Judgement:
I evaluated my decisions by repeatedly running the project locally, checking for console errors, and confirming that the project successfully uploaded to GitHub and displayed correctly via GitHub Pages. I also re-read the assignment requirements to ensure that both the repository and live project links met submission criteria.

Limitations, Dead Ends:
Some debugging steps suggested by ChatGPT did not resolve the issue and led to repeated errors. To move forward, I prioritized functionality over fully diagnosing the root cause of the Git configuration problems. As a result, some technical details about the errors remain unresolved.

Appendix:

Prompt:
"How do I upload my p5.js project from VS Code to GitHub?"

Response:
"First, make sure your project folder is open in VS Code. Open the terminal and run `git init` to initialize Git. Then run `git add .` and `git commit -m "Initial commit"`. Create a new repository on GitHub (without a README), copy the repository URL, and connect it using `git remote add origin <repo-url>`. Finally, push your code using `git push -u origin main`."

Prompt:
“I ran the commands but it didn’t work. It says it can’t access the repository.”

Response:
“This error usually means the remote URL is incorrect. Run `git remote -v` to check the URL. If it’s wrong, update it using `git remote set-url origin <correct-repo-url>`, then try pushing again.”

Prompt:
“It says ‘remote origin already exists’.”

Response:

“That means the remote was already added. You don’t need to add it again. Use `git remote set-url origin <repo-url>` to replace the existing URL instead.”

Prompt:
“Now it says ‘main cannot be resolved to branch’.”

Response:
“This usually happens if the branch hasn’t been created yet or there hasn’t been a commit. Make sure you’ve committed your files. Then run `git branch -M main` to rename your branch and try pushing again.”

Prompt:
“It still doesn’t work.”

Response:
“You can try pushing explicitly with `git push -u origin HEAD:main`. This forces Git to push your current branch to `main` on the remote.”