Process & Decision Document

1. Short Description

For this Side Quest, I created a meditative exploration experience using p5.js.
The project focuses on a camera that scrolls through a world larger than the screen while using slow pacing and smooth motion to create an emotional, reflective atmosphere rather than a fast-paced game.

The player controls a small glowing object that moves through space while the camera follows with soft easing. Hidden symbols are scattered throughout the world and are gradually discovered as the camera reaches them.

The main technical focus was:

- Building a world larger than the visible canvas
- Implementing a smooth-follow camera using interpolation
- Creating layered environmental motion (parallax particles and drift)
- Designing subtle interactive discoveries instead of explicit goals

2. One Small Decision I Made

Originally, I made the camera directly follow the player's position.
However, this felt too rigid and mechanical, which did not match the reflective goal of the assignment.

Instead, I changed the camera to move using lerp-based smoothing so it slowly eases toward the player rather than snapping instantly.

This decision:

- Made movement feel calmer and more cinematic
- Helped emphasize emotion through motion rather than speed
- Reduced visual stress during exploration
- Supported the assignment's goal of pacing-driven experience design

3. GenAI Use Statement

GenAI was used to assist with:

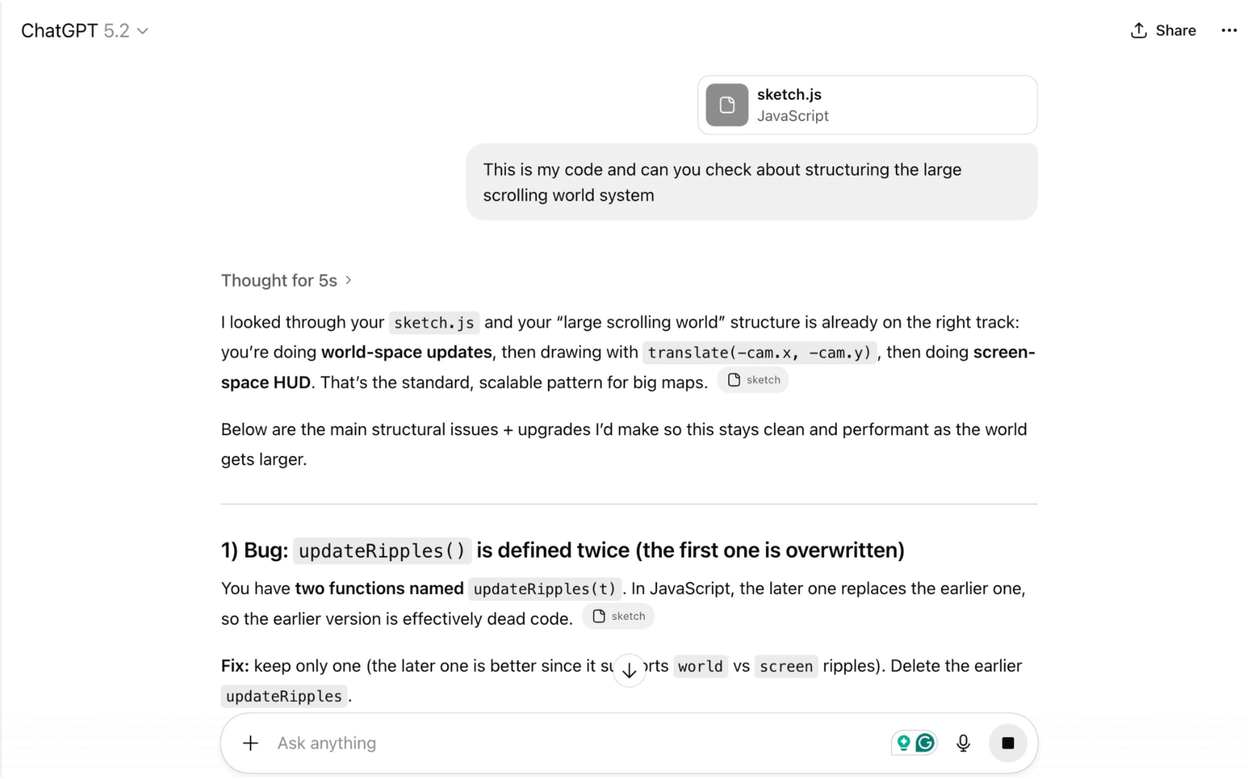
- Structuring the large scrolling world system
- Refining camera easing behavior
- Troubleshooting pacing and interaction logic

- Suggesting ways to implement hidden discovery objects

All generated suggestions were reviewed, tested, and adjusted.
The final parameters, layout decisions, and visual behavior were iterated manually during development.

No external assets were generated or imported; all visuals are drawn procedurally in code.

Appendix: GenAI Transcript



sketch.js
JavaScript

This is my code and can you check about structuring the large scrolling world system

Thought for 5s

I looked through your `sketch.js` and your "large scrolling world" structure is already on the right track: you're doing **world-space updates**, then drawing with `translate(-cam.x, -cam.y)`, then doing **screen-space HUD**. That's the standard, scalable pattern for big maps. sketch

Below are the main structural issues + upgrades I'd make so this stays clean and performant as the world gets larger.

---

1) Bug: `updateRipples()` is defined twice (the first one is overwritten)

You have **two functions named** `updateRipples(t)`. In JavaScript, the later one replaces the earlier one, so the earlier version is effectively dead code. sketch

Fix: keep only one (the later one is better since it su↓orts `world` vs `screen` ripples). Delete the earlier `updateRipples`.