Process & Decision Document

Student: Qianhui (Kiki) Tan
Role: Creator

Project / Assignment Decisions

(Side Quests – Individual Work)

A key design decision in this assignment was to express **panic** primarily through movement and interaction rather than visual detail alone. I chose to exaggerate panic behavior in early iterations (e.g., high speed, strong jitter, unstable edge behavior) to clearly surface problems before refining them in later commits. This approach allowed me to make concrete, observable improvements—such as stabilizing wall collisions and improving emotional readability—rather than attempting to “design perfectly” from the start.

Role-Based Process Evidence

Entry Header

Name: Kiki Tan
Role(s): Designer & Developer
Primary responsibility for this work:
Designing and implementing emotional movement, interaction mechanics, and iterative refinement using p5.js.

Goal of Work Session

The goal was to redesign a simple animated blob so that its behavior clearly communicates panic, and to explore a small “mischief” interaction where the blob disrupts objects in its environment. I aimed to document this process through multiple GitHub commits that demonstrate exploration, failure, refinement, and polish.

Tools, Resources, or Inputs Used

- **p5.js** (JavaScript creative coding library)
- **GitHub** (version control and process evidence via commits)
- **ChatGPT (GenAI)** for coding and debugging support
- Manual testing and visual observation in the browser

GenAI Documentation

Date Used

2026 Jan 26

Tool Disclosure

ChatGPT (text-based large language model)

Purpose of Use:
GenAI was used primarily as coding support during development. This included rewriting and extending existing p5.js code, helping structure functions, suggesting parameter adjustments, and improving code comments for clarity. It was not used to generate the project concept or emotional intent.

Summary of Interaction:
Through iterative prompts, GenAI assisted with modifying and expanding my existing codebase. This included proposing alternative implementations for movement logic, boundary handling, and collision-based interactions, as well as rewriting code segments to improve readability and maintain consistent commenting style. GenAI also helped identify technical issues (such as edge-sticking or unstable motion) and suggest code-level fixes rather than conceptual ideas.

Human Decision Point(s):
All design and implementation decisions remained human-driven. I selected which code changes to integrate, manually tuned all parameters through testing, and chose when to intentionally retain or remove flawed behaviors to support iteration. GenAI-generated code was treated as a draft or reference and was frequently modified, simplified, or partially discarded based on observed behavior.

Integrity & Verification Note:
All GenAI-assisted code was reviewed and tested in the browser. Behavior was validated through repeated interaction and observation, and changes were cross-checked against assignment requirements. No code was included without manual verification and adjustment.

Scope of GenAI Use:
GenAI did not define the project’s goals, emotional framing, or interaction design decisions. It did not independently generate the full codebase or finalize mechanics. Core logic, tuning, and evaluation were completed by me.

Limitations or Misfires:
Some suggested implementations required additional adjustment to fit the existing code structure or produced unintended side effects (e.g., overly strong reactions or mismatched visual cues). These cases required human judgement and iterative correction.

Summary of Process (Human + Tool)

The project progressed through a series of small, intentional commits: establishing an environmental baseline, exploring exaggerated panic behavior, refining motion for readability, adding a mischief mechanic, and finally balancing the system for clarity and presentation. GenAI supported ideation and reflection, while all evaluation, tuning, and final decisions were made through testing and observation.

Decision Points & Trade-offs

- **Exaggeration vs. Control:**
  Early panic behavior was intentionally chaotic to surface problems before refinement.
- **Expressiveness vs. Simplicity:**
  Facial features were kept minimal to avoid over-design while still communicating emotion.
- **Mischief Frequency vs. Readability:**
  Object stealing was reduced and limited in duration to prevent visual overload.

Verification & Judgement

Decisions were evaluated through:

- Repeated manual testing
- Comparing emotional readability across commits
- Observing unintended behaviors (e.g., edge-sticking, overly cheerful expressions)
- Re-checking assignment goals and bonus requirements

Limitations, Dead Ends, or Open Questions

This project prioritised panic over other emotional states. Additional work could explore how the same system adapts to emotions like curiosity or joy. More complex object behaviors were intentionally avoided to keep the system lightweight and focused on emotional communication.

Appendix — Full GenAI Conversation Transcript (Simulated)

Note: This is a simulated transcript reconstructed for transparency and verification, aligned with the project's commit-by-commit workflow. It reflects the kinds of prompts, code-focused assistance, and iterative decision-making used during development.

Commit 1 — Environment Baseline (Walls + Objects)

User:
I have a basic p5.js blob animation. I need to redesign it for a specific emotion later, and also add a small map with objects. For the first commit, I want environment only. Can you help me modify the code but keep the original comment style?

GenAI:
Yes. For a clean baseline, keep the blob animation unchanged and only add:

1. simple "walls" (rect border),
2. an `objects` array with `makeObject()` and `drawObjects()`.
   I'll keep your comment style and avoid changing blob deformation logic so later behavior changes are easy to observe.

User:
Ok. Please provide the updated `sketch.js` with comments.

GenAI:
Provided an updated `sketch.js` that adds `drawWalls()`, `makeObject()`, and `drawObjects()` while keeping the blob's noise-based breathing untouched. Also added a small on-screen text line identifying this as "Commit 1 baseline".

- Blob still animates and stays centered
- Walls render correctly
- Objects appear within bounds
- No interactions yet (expected)

Commit 2 — First Panic Movement Pass (Intentionally Rough)

User:
Now I want panic movement: blob should flee the mouse. But I want this first attempt to be intentionally intense and imperfect so I can refine later. Can you help implement that?

GenAI:
Yes. We can treat the mouse as a “threat” inside a fear radius. Add velocity, acceleration, and jitter. For an intentionally rough version, set acceleration and max speed high, and use simple boundary constraints (`constrain`) that may cause edge-sticking (which you can fix in the next commit).

User:
Please update `sketch.js`. Keep comments consistent.

GenAI:
Provided code changes:

- added `vx/vy`, `accel`, `maxSpeed`, `fearRadius`, `jitter`
- added `updatePanicMotion()` called in `draw()`
- left objects static (no mischief yet)
- updated on-screen text to “Commit 2: panic movement”

User:
How do I verify it’s working?

GenAI:

- Move mouse near blob: it should accelerate away
- Move mouse away: it should drift slowly
- Expect roughness: near walls it may “stick” due to `constrain` (this is intentional for iteration)

Commit 3 — Refine Panic (Fix Edge-Sticking + Improve Readability)

User:
Commit 2 works but it feels buggy near edges and too chaotic. I want it to feel like panic, not broken physics. Can you help refine?

GenAI:
Yes. Two main fixes:

1. replace `constrain()` boundary handling with bounce-based collision (reverse velocity with damping)

2. reduce max speed and jitter, add damping so it calms down when safe
   For readability, scale deformation/breath speed with fear, and optionally add a simple face that looks away from the mouse.

User:
I’m okay adding a simple face. Let’s do the refined movement + a minimal face.

GenAI:
Provided `sketch.js` update:

- bounce-based wall collisions (prevents edge-sticking)
- reduced `maxSpeed`, reduced shake, added `damping`
- fear-scaled wobble/breathing speed
- simple eyes + mouth

User:
What should I test?

GenAI:

- Blob rebounds off walls instead of pinning
- Panic triggers within fear radius
- Far from mouse: motion gradually calms
- Face should not overpower the emotion; if mouth reads as “happy,” revise mouth shape next iteration

User:
Cool, I’ll commit as a refinement.

Commit 4 — Mischief v1 + Facial Adjustment (Panic Face Fix)

User:
Ok the face is kind of funny—it reads like a smile. Also I need the bonus: blob bumps objects and sometimes steals them. Can we put both in Commit 4?

GenAI:
Yes, that’s reasonable and easy to justify: “mischief mechanic + emotional readability correction.”
For mischief:

- give each object `vx/vy` so bumping is visible
- on overlap: apply impulse to push objects

- allow “steal” only for small objects: switch to `stolen=true`, orbit around blob with offset angle
  For face fix: replace curved mouth with an open-mouth “gasp” and enlarge eyes under fear.

User:
I want it to be obviously interactive, even if it’s a bit messy. So steal chance can be a bit high now; we’ll balance later. Please implement.

GenAI:
Provided updated code:

- objects now have velocities + friction + bounce
- collision applies push impulse
- small objects have chance to become stolen and orbit blob
- stolen objects change color (yellow)
- face updated to open mouth + wide eyes (panic)
- on-screen UI includes “Stolen: X”

User:
What known issues should I expect so I can refine next?

GenAI:

- If steal chance is high, many objects may become stolen quickly, reducing scene readability
- Orbiting objects may clutter the blob silhouette
- This is fine for v1; balance and pacing can be handled in Commit 5

Commit 5 — Balance + Polish (Presentation-Ready)

User:
Mischief works but it’s too chaotic: stealing happens too often and it becomes hard to read the scene. I want a final commit to balance and polish for GitHub Pages.

GenAI:
Recommended changes:

1. reduce `stealChance` so stealing feels like a surprise
2. add a time-limited stolen state (objects drop after N frames)
3. slightly calm panic shake for readability
4. tighten on-screen instructions for public demo

User:
Implement those changes. I want the final version stable and readable.

GenAI:
Provided updated code:

- lowered `stealChance`
- added `life` timer for stolen objects; when expired, objects drop back with some velocity
- slightly reduced jitter and refined face scaling
- simplified UI text for final presentation

User:
How do I verify final quality?

GenAI:

- Panic still reads clearly (flee + fear-scaled deformation + gaze away)
- Bumping is consistent and visible
- Stealing happens occasionally, not constantly
- Stolen objects return to environment (restores variety)
- Page loads reliably (hard refresh if caching)