

Process & Decision Documentation

This document is used to make your design and development process visible. At this stage of your academic career, you are expected not only to produce finished work, but to articulate how decisions were made, how ideas changed, and how collaboration (for the assignments that include group work) unfolds

In professional and co-op contexts, employers do not only evaluate your final projects in your portfolio. They often ask candidates to explain their process, justify trade-offs, reflect on iteration, and describe their roles within a team.

You will need to submit a modification of this document for every group assignment (A1 – A3) and a shorter version for your individual assignments (Side Quests and A4).

For A1 – A3, this is a group document submitted once per group. Each group member must clearly document their own role and responsibilities. Different roles will naturally produce different design processes.

Project/Assignment Decisions

Use this section to document key decisions that shape the direction, scope, or outcome of the assignment. The purpose is to make your judgement and reasoning visible, not to record every action taken.

What you include here should reflect decisions that mattered: moments where you chose one path over another, changed direction, or adapted to constraints.

The expected length and detail depend on the assignment.

Side Quests and A4 (Individual Work)

Keep this section brief, typically 2 to 4 sentences.

Focus on:

- One significant decision or change you made
- Why you made it
- What effect it had on the work

Examples:

- Simplifying a mechanic so it functioned correctly
- Changing an approach after something failed

- Deciding not to pursue an idea due to time or technical limitations

You are not expected to document every alternative or iteration

Role-Based Process Evidence

Entry Header

Name: Katherine Chen

Role(s): Coder

Primary responsibility for this work: Completing Side Quest 2 (redesigning blob and environment with anxious emotion)

Goal of Work Session

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

- I was trying to redesign the original sidequest 2 code example 3 into the emotion that I chose: anxiety
- The red, neon yellow, and orange colours and erratic fast movements were chosen to represent a highly panicked and anxious environment

Tools, Resources, or Inputs Used:

- GenAI: ChatGPT Free Version

GenAI Documentation

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

Date Used: Sun Jan 25th, 2025

Tool Disclosure: Chat GPT Free Version

Purpose of Use: To quickly understand variable names in the original sidequest 2 code example 3

Summary of Interaction: I only used Chatgpt to quickly let me know exactly which variables in the code were related to the sidequest objectives for my chosen emotion, as well as which rgb colour codes suited it best. I did not use it for decision making on exactly what values or parameters to set, as I chose those myself.

Human Decision Point(s): GenAI gave me a list of variables affected, as well as what values to set it to to affect the blob and its environment. I only looked at which sections were affected and did not take into account any of the values it gave me. In fact, I didn't even use all the variables it gave to me, I just decided myself based off trial and error in live

server which ones I wanted to apply to my work. It's the same with rgb, I just took the colour codes as suggestions and did my own tweaks.

Integrity & Verification Note: To see if GenAI was giving me the correct affected variables and colours, I tested it out myself through altercations of different values.

Scope of GenAI Use: GenAI did not contribute to my final decision making as to how I wanted the product to look. It only told me what variable did contributed to which visual function and I played around with the rest myself.

Limitations or Misfires: I would say that some of the rgb colours it provided me for anxiety were not what I was looking for. While many things like colour representation can be up to interpretation, I still felt like colours like "blue" were more suited to calming environments rather than panic.

Summary of Process (Human + Tool)

- I first asked ChatGPT to generate me which parts of the original sketch.js code was affected by the sidequest prompt, then chose which sections to alter myself without completely following what the AI told me
- After playing around with everything (eg. Adding platforms, changing blob freq and speed, ect.), I finalized the values
 - I made many platforms and some of different weight to make it feel "overwhelming"
- I also asked ChatGPT to generate for me a list of rgb values I could use for an anxious emotion and chose which colours I personally thought suited it best
 - I chose red, orange, and neon yellow because I thought it best suited anxiety (and because anxiety was orange in inside out 2)

Decision Points & Trade-offs

- I didn't choose the original blue rgb recommended to me by chatgpt, but instead opted for the red, yellow, orange palette

Verification & Judgement

Instead of having to commit to github every time I made a change, I ran the code through live server and played around with different values until I was satisfied with how the blob and environment represented my emotion.

Limitations, Dead Ends, or Open Questions

While I did my best with the colour choise, it can be misinterpreted as anger rather than anxiety.


Appendix

Please include a full transcript of your conversation with the GenAI. GenAI transcripts are used for transparency and verification only and are not assessed for writing quality or completeness.

Note: If GenAI is used in another language, you must include both the original language and a translated transcript in the appendix

Prompt	Answer
<pre>which parts of this code: // Y-position of the floor (ground level) let floorY3; // Player character (soft, animated blob) let blob3 = { // Position (centre of the blob) x: 80, y: 0, // Visual properties r: 26, // Base radius points: 48, // Number of points used to draw the blob wobble: 7, // Edge deformation amount wobbleFreq: 0.9, // Time values for breathing animation t: 0, tSpeed: 0.01, // Physics: velocity vx: 0, // Horizontal velocity vy: 0, // Vertical velocity // Movement tuning accel: 0.55, // Horizontal acceleration maxRun: 4.0, // Maximum horizontal speed gravity: 0.65, // Downward force</pre>	<p>Great question — this code is actually <i>perfect</i> for an “emotion through movement” requirement. The key idea is: you don’t need new systems, you reinterpret and tweak the ones you already have so they <i>feel</i> emotional.</p> <p>Below I’ll break this into (1) which parts of the code matter, and (2) how you’d redesign them to express emotion, with concrete examples like <i>joy</i>, <i>frustration</i>, and <i>panic</i>.</p>
	<p>1 Parts of your code that communicate “emotion”</p> <p>These are the <i>emotion levers</i> in your sketch:</p> <p>A. Movement physics (MOST IMPORTANT)</p> <p>These directly affect how the blob <i>feels</i> to control.</p> <pre>accel maxRun gravity jumpV frictionAir frictionGround</pre> <p>Why they matter:</p> <ul style="list-style-type: none">• Acceleration → confidence vs hesitation• Max speed → excitement vs heaviness• Gravity → floaty joy vs oppressive weight• Jump strength → freedom vs struggle

<pre> jumpV: -11.0, // Initial jump impulse // State onGround: false, // True when standing on a platform // Friction frictionAir: 0.995, // Light friction in air frictionGround: 0.88, // Stronger friction on ground }; // List of solid platforms the blob can stand on // Each platform is an axis-aligned rectangle (AABB) let platforms = []; function setup() { createCanvas(640, 360); // Define the floor height floorY3 = height - 36; noStroke(); textFont("sans-serif"); textSize(14); </pre>	<ul style="list-style-type: none"> • Friction → slippery chaos vs sticky resistance <p>👉 Changing just these can completely change the emotional tone.</p>
<pre> // Create platforms (floor + steps) platforms = [{ x: 0, y: floorY3, w: width, h: height - floorY3 }, // floor { x: 120, y: floorY3 - 70, w: 120, h: 12 }, // low step { x: 300, y: floorY3 - 120, w: 90, h: 12 }, // mid step { x: 440, y: floorY3 - 180, w: 130, h: 12 }, // high step { x: 520, y: floorY3 - 70, w: 90, h: 12 }, // return ramp]; // Start the blob resting on the floor </pre>	<h3>B. Platform layout (environmental emotion)</h3> <pre> platforms = [{ x, y, w, h }]; </pre> <p>Why they matter:</p> <ul style="list-style-type: none"> • Spacing = generosity vs cruelty • Height = accessibility vs intimidation • Width = safety vs anxiety <p>The environment can feel:</p> <ul style="list-style-type: none"> • welcoming • hostile • overwhelming • playful • claustrophobic
<pre> // Create platforms (floor + steps) platforms = [{ x: 0, y: floorY3, w: width, h: height - floorY3 }, // floor { x: 120, y: floorY3 - 70, w: 120, h: 12 }, // low step { x: 300, y: floorY3 - 120, w: 90, h: 12 }, // mid step { x: 440, y: floorY3 - 180, w: 130, h: 12 }, // high step { x: 520, y: floorY3 - 70, w: 90, h: 12 }, // return ramp]; // Start the blob resting on the floor </pre>	<h3>C. Blob animation (emotional body language)</h3> <pre> r wobble wobbleFreq tSpeed points </pre> <p>Why they matter:</p> <ul style="list-style-type: none"> • High wobble → nervous / excited • Slow breathing → calm / sad • Fast jitter → panic • Smooth edges → relaxed • Chaotic edges → unstable

<pre>blob3.y = floorY3 - blob3.r - 1; }</pre>	<p>This is <i>non-verbal emotion</i>.</p>
<pre>function draw() { background(240); // --- Draw all platforms --- fill(200); for (const p of platforms) { rect(p.x, p.y, p.w, p.h); } // --- Input: left/right movement --- let move = 0; if (keyIsDown(65) keyIsDown(LEFT_ARROW)) move -= 1; // A or ← if (keyIsDown(68) keyIsDown(RIGHT_ARROW)) move += 1; // D or → blob3.vx += blob3.accel * move; // --- Apply friction and clamp speed --- blob3.vx *= blob3.onGround ? blob3.frictionGround : blob3.frictionAir; blob3.vx = constrain(blob3.vx, - blob3.maxRun, blob3.maxRun); // --- Apply gravity --- blob3.vy += blob3.gravity; // --- Collision representation --- // We collide using a rectangle (AABB), // even though the blob is drawn as a circle let box = { x: blob3.x - blob3.r, y: blob3.y - blob3.r, w: blob3.r * 2, h: blob3.r * 2, }; }</pre>	<h3>D. Control responsiveness</h3> <pre>blob3.vx += blob3.accel * move; blob3.vx *= friction...</pre> <p>This controls:</p> <ul style="list-style-type: none"> • “I press a key → does the blob <i>immediately</i> respond?” • Lag = frustration • Over-response = panic
<pre>blob3.vx *= blob3.onGround ? blob3.frictionGround : blob3.frictionAir; blob3.vx = constrain(blob3.vx, - blob3.maxRun, blob3.maxRun); // --- Apply gravity --- blob3.vy += blob3.gravity; // --- Collision representation --- // We collide using a rectangle (AABB), // even though the blob is drawn as a circle let box = { x: blob3.x - blob3.r, y: blob3.y - blob3.r, w: blob3.r * 2, h: blob3.r * 2, };</pre>	<h2>2 Mapping emotions to code changes</h2> <p>Below are clear, defensible redesigns you could literally describe in a project write-up.</p>
<pre>blob3.vx *= blob3.onGround ? blob3.frictionGround : blob3.frictionAir; blob3.vx = constrain(blob3.vx, - blob3.maxRun, blob3.maxRun); // --- Apply gravity --- blob3.vy += blob3.gravity; // --- Collision representation --- // We collide using a rectangle (AABB), // even though the blob is drawn as a circle let box = { x: blob3.x - blob3.r, y: blob3.y - blob3.r, w: blob3.r * 2, h: blob3.r * 2, };</pre>	<h3> Emotion Example: JOY</h3> <p>Design goal: The blob feels light, playful, and eager to move.</p> <p>Code changes:</p> <pre>accel: 0.7 maxRun: 6.0 gravity: 0.45 jumpV: -13.5 frictionGround: 0.92</pre> <p>Animation tweaks:</p> <pre>wobble: 10 tSpeed: 0.02</pre> <p>Environment changes:</p> <ul style="list-style-type: none"> • Platforms are wide • Gaps are forgiving • Vertical movement is rewarded

```
// --- STEP 1: Move horizontally, then
resolve X collisions ---
box.x += blob3.vx;
for (const s of platforms) {
  if (overlap(box, s)) {
    if (blob3.vx > 0) {
      // Moving right → hit the left side of
      a platform
      box.x = s.x - box.w;
    } else if (blob3.vx < 0) {
      // Moving left → hit the right side of
      a platform
      box.x = s.x + s.w;
    }
    blob3.vx = 0;
  }
}
```

```
// --- STEP 2: Move vertically, then
resolve Y collisions ---
box.y += blob3.vy;
blob3.onGround = false;

for (const s of platforms) {
  if (overlap(box, s)) {
    if (blob3.vy > 0) {
      // Falling → land on top of a
      platform
      box.y = s.y - box.h;
      blob3.vy = 0;
      blob3.onGround = true;
    } else if (blob3.vy < 0) {
      // Rising → hit the underside of a
      platform
      box.y = s.y + s.h;
      blob3.vy = 0;
    }
  }
}
```

```
// --- Convert collision box back to
blob centre ---
blob3.x = box.x + box.w / 2;
```

```
{ x: 200, y: floorY3 - 60, w: 180, h:
12 }
```

Emotional effect:

- Floaty jumps feel like *delight*
- High wobble reads as *bounciness*
- Speed feels playful, not stressful

💬 “The blob expresses joy through buoyant physics and exaggerated, bouncy motion.”



Emotion Example: FRUSTRATION

Design goal:

The blob struggles. Inputs feel unreliable. The world resists you.

Code changes:

```
accel: 0.35
maxRun: 3.0
gravity: 0.9
jumpV: -9.0
frictionGround: 0.80
```

Animation tweaks:

```
wobble: 3
tSpeed: 0.005
```




Environment changes:

- Platforms are **narrow**
- Slightly too far apart
- Lots of head bonks






```
{ x: 260, y: floorY3 - 110, w: 60, h:
12 }
```






Emotional effect:

- Slow acceleration feels unresponsive
- Heavy gravity makes jumps fail *just barely*
- Low wobble = tense, clenched body

<pre> blob3.y = box.y + box.h / 2; // Keep blob inside the canvas horizontally blob3.x = constrain(blob3.x, blob3.r, width - blob3.r); // --- Draw the animated blob --- blob3.t += blob3.tSpeed; drawBlobCircle(blob3); // --- HUD --- fill(0); text("Move: A/D or </> • Jump: Space/W/↑ • Land on platforms", 10, 18); } // Axis-Aligned Bounding Box (AABB) overlap test // Returns true if rectangles a and b intersect function overlap(a, b) { return (a.x < b.x + b.w && a.x + a.w > b.x && a.y < b.y + b.h && a.y + a.h > b.y); } // Draws the blob using Perlin noise for a soft, breathing effect function drawBlobCircle(b) { fill(20, 120, 255); beginShape(); for (let i = 0; i < b.points; i++) { const a = (i / b.points) * TAU; // Noise-based radius offset const n = noise(cos(a) * b.wobbleFreq + 100, sin(a) * b.wobbleFreq + 100, b.t,); </pre>	<p> “The blob’s movement conveys frustration through heavy gravity, limited jump power, and restrictive level design.”</p> <p> Emotion Example: PANIC</p> <p>Design goal: The blob feels unstable, overreactive, and hard to control.</p> <p>Code changes:</p> <pre> accel: 1.2 maxRun: 7.5 gravity: 0.7 jumpV: -15 frictionGround: 0.97 </pre> <p>Animation tweaks:</p> <pre> wobble: 14 wobbleFreq: 2.0 tSpeed: 0.04 points: 32 </pre> <p>Environment changes:</p> <ul style="list-style-type: none"> • Platforms at uneven heights • Little time to react • Tight landings <p>Emotional effect:</p> <ul style="list-style-type: none"> • Tiny inputs cause huge motion • Blob jiggles violently • Player feels loss of control <p> “Exaggerated acceleration and chaotic deformation create a sense of panic and instability.”</p>
	<p>3 If this is for a written submission</p> <p>Your golden sentence formula:</p>

<pre> const r = b.r + map(n, 0, 1, - b.wobble, b.wobble); vertex(b.x + cos(a) * r, b.y + sin(a) * r); } endShape(CLOSE); } </pre>	<p><i>“Emotion is communicated through physics, responsiveness, and environmental layout rather than narrative.”</i></p> <p>Then explain:</p> <ol style="list-style-type: none"> 1. Which variables you changed 2. What the player <i>feels</i> 3. Why that feeling matches the emotion
<pre> // Jump input (only allowed when grounded) function keyPressed() { if ((key === " " key === "W" key === "w" keyCode === UP_ARROW) && blob3.onGround) { blob3.vy = blob3.jumpV; blob3.onGround = false; } } </pre> <p>/* In-class tweaks for experimentation:</p> <ul style="list-style-type: none"> • Add a new platform: <pre>platforms.push({ x: 220, y: floorY3 - 150, w: 80, h: 12 });</pre> • “Ice” feel → frictionGround = 0.95 • “Sand” feel → frictionGround = 0.80 <p>*/</p> <p>can help me with this project requirement: Redesign the blob’s movement and environment to express a specific emotion (e.g., joy, frustration, panic).</p>	<p>If you want, tell me:</p> <ul style="list-style-type: none"> • 🧠 which emotion you’re leaning toward • 📝 whether this is for a design write-up, crit, or grading rubric <p>and I’ll help you phrase it <i>exactly</i> how instructors like to see it.</p>
<p>what are the rgb colours codes for anxiety?</p>	<p>Ooo, good design question — <i>anxiety</i> is a really specific vibe 😬</p>

	<p>There’s no single “correct” RGB, but anxiety usually reads as muted, cold, sickly, or high-contrast colors.</p> <p>Here are commonly used RGB palettes for anxiety, with explanations so you can justify them in your project.</p>
	<p> Classic “anxiety” color choices (RGB)</p> <p> Muted blue-gray (uneasy, tense) <code>rgb(90, 110, 130)</code></p> <p>Feels cold, distant, emotionally tight.</p>
	<p> Sickly green (nervous / uncomfortable) <code>rgb(140, 170, 120)</code></p> <p>Often used for nausea, unease, stress.</p>
	<p> Dirty yellow (warning / dread) <code>rgb(180, 170, 90)</code></p> <p>Feels “off” — not cheerful yellow, more unsettling.</p>
	<p> Desaturated purple-gray (mental stress) <code>rgb(120, 110, 140)</code></p> <p>Reads as introspective, heavy, uneasy.</p>

	<div> High-anxiety / panic tones</div> <div> Dark red (danger, pressure) <code>rgb(120, 40, 40)</code></div> <div> Near-black blue (claustrophobic) <code>rgb(30, 35, 50)</code></div>
	<div> Suggested combos (ready to drop into your sketch)</div> <div>Background (anxious atmosphere) <code>background(90, 110, 130);</code></div> <div>Platforms (oppressive, heavy) <code>fill(70, 80, 90);</code></div> <div>Blob (contrasting, nervous) <code>fill(200, 160, 80); // uneasy yellow-orange</code></div> <div><p>This contrast makes the blob feel <i>out of place</i>, which reads as anxiety.</p></div>
	<div> How to explain this in a write-up</div> <div><p>You can literally say:</p><p>“Muted blue-gray and sickly yellow tones were chosen to create visual unease and emotional tension, reinforcing the theme of anxiety.”</p><p>That’s instructor-approved language 😊</p></div>
	<div><p>If you want, tell me:</p><ul style="list-style-type: none">• what you’re coloring (background / blob / platforms), and</div>

	<ul style="list-style-type: none">• how intense the anxiety should be (subtle vs panic) and I'll lock in a perfect 3-color palette for you.
--	--