

Process & Decision Documentation

Project/Assignment Decisions

Throughout the Week 2 Side Quest, I led video game design strategy decisions, edited variables throughout the code, and developed illustrations to create a mini game that features a moving blob and environment that express the emotion of panic. Below is a thorough and transparent discussion of my collaboration with AI along with ethics and process recommendations for future GenAI use.

GenAI Documentation

Date Used: January 25, 2026

Tool Disclosure: Claude Haiku 4.5 via GitHub Copilot (VS Code extension)

Purpose of Use: I used GenAI as a coding assistant, inputting my high-level ideas (e.g. “evoke nervous feeling and spiraling organic shape”) and technical specifications (e.g. font family changes, RGB colour codes, and heights and widths in pixels) then subsequently approving Generative AI outputs.

Summary of Interaction: I utilized the tool within VS Code, prompting requests in changing blob properties and physics, setting background and foreground images from my assets folder, adjusting canvas and platforms size, and adjusting text. I guided these interactions, providing specifications, code hints from my previous HTML and JavaScript knowledge and iterating over the code myself to debug.

Human Decision Point(s): I did preliminary design research and moodboarding, setting the theme of the to “panic”. This decision was inspired by my childhood video game selection “Don’t Starve” and “Luigi’s Mansion” – a dark, moody environment, with black and white (cartoon-esque) elements.

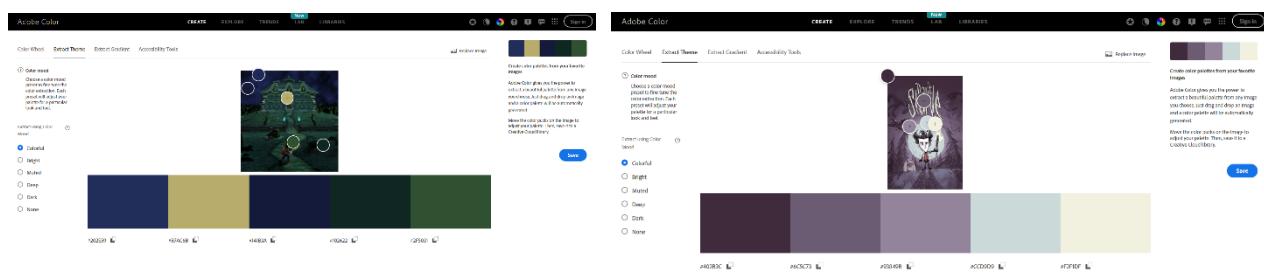


Figure 1: Adobe Color Screengrabs of Extracting Color Themes from “Luigi’s Mansion” and “Don’t Starve”

I independently led all design direction, including crafting the CoPilot prompts and illustrating the background and foreground by hand on Adobe Fresco. Additionally, I ideated the idea of separating the background and foreground and adding creative copywriting “Retreat behind the trees... or confront your fear and reach the moon” to create a more immersive web experience.

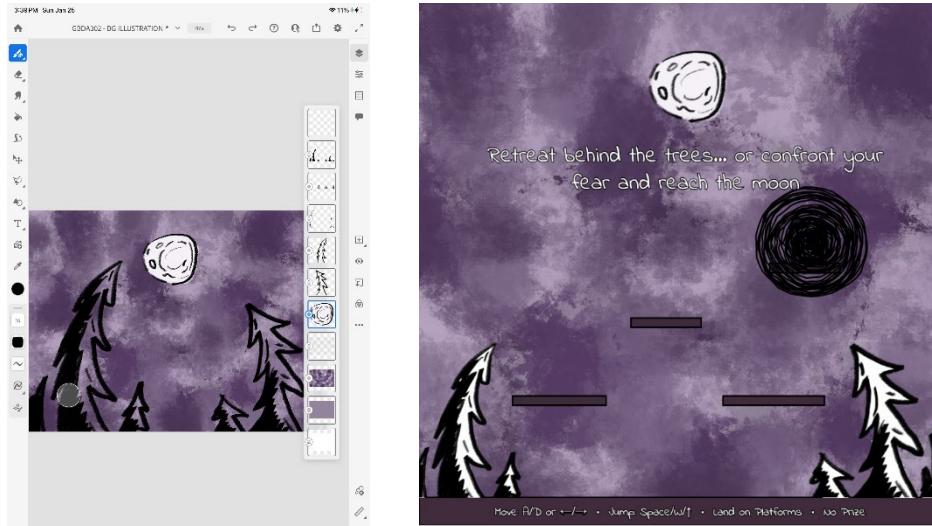


Figure 2: Adobe Fresco Illustration (Iteration #1) | Figure 3: Final GitHub website*

* To add stakes and further storytelling, I introduced foreground trees as hiding elements to the environment and positioned the moon as the player’s goal. As the blob progresses upward through the canvas, obstacles increase in game mechanics, and the blob grows in both size and ‘nervousness’ (noise).

I modified the database ideation as I found that information was left out due to its hyper-fixate on newer prompts or keywords, I modified its improvement of my assignment process to incorporate more human decision making as the instructions were very procedural and AI-oriented.

Integrity & Verification Note: I thoroughly reviewed the Generative AI guidelines in the syllabus to ensure my use of GenAI upheld academic integrity, including the verification of bias and plagiarism. I also took a hands-on approach, editing generated outputs to improve accuracy and to debug malformed code.

Scope of GenAI Use: To generate code snippets that alter the blob, canvas, and mechanics, as well as delete lines of pre-existing code that I did not find useful.

Limitations, Dead-Ends, and Misfires: At times, outputs required regeneration, especially when completing aesthetic changes, such as adjusting the blob appearance and behaviours and text adjustments. I found that position changes along the x- and y-axes were often inaccurate or difficult to prompt engineer. When I encountered dead-ends, I shifted my approach by querying

Google or applying a trial-and-error approach using my limited knowledge of HTML and JavaScript.

Summary of Process (Human + Tool)

- Research (Human)
- Moodboard (Human)
- Illustration (Human)
- Prompt Engineering (Human)
- Adjusting Blob Appearance (Human + AI)
- Adjusting Blob Mechanics (Human + AI)
- Adjusting Environment (AI)
- Adding Text (Human + AI)
- Adjusting Illustrations (Human)
- Fine-Tuning Environment (Human + AI)

Decision Points & Trade-offs

- Decision Point: Use AI to generate and code text
 - Decided to write my own text to personalize the experience and increase human development in storytelling
 - Decided to experiment with my own knowledge to add additional text
 - AI was unable to create text according to specifications
- Decision Point: Use AI to generate environment change ideas
 - Decided to ideate completely on my own (human judgement), using inspiration from childhood games
 - Using AI would have sacrificed idea quality, decision autonomy, and limited creativity
- Decision Point: How to approach the emotion of panic
 - Decided to approach by setting trees bowing inward (creating depth perspective and evoking feelings of uneasiness)
 - Decided to change the appearance entirely of the blob to capture the emotion (scribbles, canny)
 - Added background images to further personalization (inspired by the background in “Luigi’s Mansion”)

Verification and Judgment

- Used “Live Server” extension in order to verify accuracy and bias in real-time

- Supplied multiple prompts with several iterations to ensure code was differentiated from other potential video games
 - Using keywords and quotations ("") with intention will reduce inaccurate outputs and improve accuracy outputs (like incorrect naming conventions)
 - Using specific function names like "drawBlobCircle()" in order to guide accuracy and reduce redundancy

Appendix

Please find a copy of my full transcript of my conversation with the GenAI below, aligning with transparency and verification standards.

make the drawblobcircle function not a solid blue shape, instead make it look like a circle scribble that is black - evoking nervous feeling and spiraling organic shape (circle-like outer form but inside is a noisy spiral) - add text beside each line of changed code for easy debugging via human

Let me read more to find the drawBlobCircle function:

Read [sketch.js](#), lines 150 to 195

Perfect! Now I can see the [drawBlobCircle](#) function. I'll modify it to create a black circle scribble with a noisy spiral effect that evokes a nervous, organic feeling. I'll add comments beside each changed line for debugging.

Edited

sketch.js+29-15

Done! I've transformed the [drawBlobCircle](#) function from a solid blue blob to a black circle scribble with these key changes:

What changed:

- **Removed fill, added black stroke** for a hand-drawn scribble appearance
- **4 spiral layers** that create an inward spiraling motion, evoking nervous energy
- **Jitter noise** added on top of the wobble for that unsettled, jittery feeling
- **Spiral shrinking** makes the inner loops tighter, creating an organic spiral effect
- **Every line has a comment** explaining what it does for easy debugging

The result is a black circle scribble with nested spirals inside that creates that nervous, anxious vibe—like the blob is trembling or thinking intensely!

Now I'll implement the dynamic spiral count based on platform level and make spirals span beyond the shape:

Let me fix the JSON formatting:

Edited

sketch.js+2-2

Perfect! I've updated the drawBlobCircle function with these key changes:

What changed:

1. **Dynamic spiral count:** Starts at 8 spirals on the floor, then adds 15 more spirals for each platform level you jump to
 - Floor (level 0): 8 spirals
 - 1st platform: 23 spirals
 - 2nd platform: 38 spirals
 - 3rd platform: 53 spirals
 - etc.
2. **New calculatePlatformLevel() function:** Detects which platform the blob is currently on and returns the level (0, 1, 2, 3...)
3. **Spirals now span beyond the shape:** Instead of all spiraling inward (concentrated in middle), the spirals now expand outward using a sine wave curve that oscillates—early spirals go outward, mid spirals are at max extension, and later spirals fold back inward. This creates a distributed, radiating effect that extends well beyond the base circle.
4. **Every change is clearly commented** for easy debugging!

Now I'll update the code to load the background image and use it:

Edited

sketch.js+29-1

Done! I've updated the sketch to use the background image from [bg.png](#). Here's what changed:

Key updates:

1. **Added `bgImage` variable** to store the loaded image
2. **Loads the image in `setup()`** using `loadImage('assets/bg.png')`
3. **Replaced the color background** with image drawing that:
 - o Calculates aspect ratios to maintain the image's proportions
 - o Scales the image to fill as much of the canvas as possible without distortion
 - o Centers the image on the canvas
 - o Every step is clearly commented for debugging

The image will now fill your canvas while preserving its original aspect ratio!

Edited

sketch.js+22-0

Done! I've added two foreground images that overlay as the topmost layer:

What changed:

1. **Added `fg1Image` and `fg2Image` variables** to store the foreground images
2. **Loaded both images in `setup()`** from `fg1.png` and `fg2.png`
3. **Drew them at the end of `draw()` function** (after the blob and HUD) so they appear on top like z-index: 999
4. **Positioned them:**
 - o Bottom left: 20px from left edge, 20px from bottom
 - o Bottom right: 20px from right edge, 20px from bottom
 - o Both with 200px height, maintaining their original aspect ratios
5. **Every line is commented** for easy debugging

The blob can now hide behind these foreground elements!

Edited

sketch.js+1-1

Done! Jump height increased by 1.125x: -11.0 → -12.375

Edited

sketch.js+11-2

Done! The spiral count now dynamically increases as the blob moves toward the top of the canvas:

What changed:

- Calculates `distanceFromTop = floorY3 - b.y` to measure how far up the blob has traveled
- Spiral count: $50 + (\text{distanceFromTop} / 50) * 10$
 - At floor: 50 spirals
 - 50px up: 60 spirals
 - 100px up: 70 spirals
 - And so on...

The style and all visual effects remain exactly the same—only the spiral count changes based on vertical position.

Edited

sketch.js+13-4

Done! The "Retreat..." text now:

- **Centered on X-axis** (middle of canvas at 335px)
- **Same Y position** (40px)
- **Font**: Indie Flower
- **Size**: 17.5 (1.25x larger than 14)
- **Color**: White (no stroke)
- All changes are clearly commented for debugging

decrease the container width of the indie flower text so that it is contained within the canvas not spilling out (but all content is still readable)

ensure the "indie flower" text is in the center of the canvas and add 20px top padding

Edited

sketch.js+3-3

Done! The "Indie Flower" text is now:

- **Centered horizontally** on the canvas ($\text{width} / 2$)
- **20px top padding** from the top of the canvas
- **vertically aligned to top** (`textAlign(CENTER, TOP)`) for consistent padding
-
-

Claude Haiku 4.5 • 1x

I see the text is currently centered at width / 2. To remove 50px left padding, I need to shift it 50px to the right:

Edited

sketch.js+2-2

Done! Removed 50px left padding from the indie flower text by shifting it 50px to the right (width / 2 + 50).

-
-

move the sans-serif txt that says "Move:..." to just below the indie flower text

Edited

sketch.js+2-2

Done! Moved the "Move:..." text to y=260 (just below the indie flower text at y=200) and centered it on the X-axis to align with the narrative text above.