

Process & Decisions Documentation

Blob Emotion Side Quest

Name: Charlie Cilla

Assignment: Side Quest – Blob Emotion Puzzle Game

Dates: January 25–27

Role: Individual Designer & Developer

Project Overview

This Side Quest explored the design and development of an emotion-driven puzzle game built in p5.js.

The project focuses on a blob character whose movement, shape, and controls change based on emotional

states triggered by placing colored boxes into matching aura zones. Emotion functions as both feedback

and constraint, encouraging strategic play and experimentation.

Key Project Decisions

A major decision was transitioning the game from a gravity-based platformer to a top-down interaction model.

This reduced mechanical noise and allowed emotional effects to be more legible and player-driven.

Another important decision was implementing emotion stacking rather than limiting the blob to one emotion at a time.

This created meaningful trade-offs while keeping the system readable and manageable within the scope of a Side Quest.

Process Summary

Development occurred over several short iterative sessions. Initial work focused on redefining movement and emotional expression.

Subsequent iterations addressed interaction issues such as unintended automatic placement and lack of player control.

Manual pickup and drop mechanics were introduced, followed by emotion stacking, time-based mechanics, sliding box behaviour,

and a clear victory condition. Continuous playtesting informed each change.

GenAI Documentation

Date Used: January 25–27

Tool Disclosure: GitHub Copilot (VS Code)

Purpose of Use: Coding assistance, debugging, and iteration support

Copilot was used to suggest code structures and logic for mechanics such as emotion timers, top-down movement, box interaction, and edge-based respawn systems. All high-level design decisions were human-led, with Copilot serving as an assistive tool rather than an author.

Human Decision Points

I frequently overrode or corrected suggested implementations when mechanics behaved incorrectly. Decisions such as

emotion stacking, manual controls, and time pressure systems were determined independently and refined through testing.

Verification & Integrity

All GenAI-assisted code was verified through live testing in p5.js. Mechanics were evaluated based on gameplay feel,

readability, and alignment with assignment goals. Errors and misfires were debugged manually.

Limitations & Reflections

Some emotional effects could be expanded further with sound or animation, but these were outside the scope of the assignment.

Timing and difficulty balancing could benefit from additional playtesting. Overall, the project demonstrates iterative design,

intentional decision-making, and controlled use of GenAI tools.

Appendix Note

Complete Development Dialogue Transcript

Blob Emotion Sidequest - Charlie Cilla

--

Session 1: Initial Concept & Direction

User Prompt 1.1

> This is the assigned task for this side_quest: Redesign the blob's movement and environment to express a specific emotion (e.g., joy, frustration, panic). Bonus: Add a "mischief" mechanic, your

blob steals or bumps objects oh a small map

Assistant Response 1.1

Provided initial code structure for emotion-based mechanics:

- Proposed `setEmotion(emotion)` function to modify blob properties
- Suggested adding an `objects` array for interactive elements
- Outlined `checkMischief()` function for collision detection
- Explained that direct editing was restricted at that time

****Status:**** Concept validated, ready for implementation

--

Session 2: Convert to Top-Down Gameplay

User Prompt 2.1

> Ok, first i would like to implement a top down feature. We will need to remove the gravity and for now remove the platforms to be replaced with boxes. With these boxes in specific places different effect are put on the blob. So for example if it put a green box in the green square hole, it will feel nauseas and move around with the controls reversed. Red box/hole will be anger which make the blobs wobble higher and the screen a little red, yellow will make it happy and move around faster, blue will make it sad and move real slow and reduce its number of points (a sad blob would be geometric).

Assistant Response 2.1

****Complete system redesign implemented:****

Modified `blob3` object:

- Removed gravity and jump mechanics
- Changed position to canvas center (320, 180)
- Adjusted acceleration to 0.5 (reduced from 0.55)
- Set maxRun to 3.0 for top-down movement
- Added friction: 0.92 (top-down physics)

Emotion system created with 4 states:

- ****Red (Anger)**:** wobble multiplier 2.5x, screen tint (255, 100, 100)
- ****Green (Nausea)**:** reversed controls for duration, screen tint (150, 255, 150)

- **Yellow (Joy)**: speed multiplier 1.5x, screen tint (255, 255, 100)
- **Blue (Sadness)**: speed multiplier 0.4x, points reduced to 12, screen tint (100, 150, 255)

Replaced platforms with:

- 4 colored boxes (red, green, yellow, blue) at spawn locations
- 4 colored holes (aura zones) for placement

Updated draw() function for top-down movement with emotion modifiers.

****Status:**** Top-down mechanics working, emotion system active

Session 3: Button-Based Controls & Aura Zones

User Prompt 3.1

> Ok, it looks good. But me need to implement a feature where the blob can place the box in the area because otherwise they pick it up as they leave the zone, make the zone also like an aura area to be placed. This will also stop the blbo from makeing a box disappear because when a box is already in hand it will disapear if they pick up another box. This will be good for now

Assistant Response 3.1

****Implemented manual pickup/drop system:****

Changes to box behavior:

- Added `nearBlob` property to detect proximity
- Added `held` boolean to track pickup state
- Implemented 20-pixel pickup radius

Introduced E and Q key controls:

- **E Key**: Manually pick up nearby boxes
- **Q Key**: Manually place boxes in zones
- Prevents automatic pickup when already holding a box

Improved aura zones:

- Added gradient fade effect with multiple circles

- Decreasing opacity from center outward
- Semi-transparent colored circles (30 alpha)

Fixed box disappearance issue:

- Only one box can be held at time
- Boxes update position when held
- `checkMischief()` renamed to `checkHolePlacement()`

Status: Manual controls working, no box loss

Session 4: Simplify Design & Enable Emotion Stacking

User Prompt 4.1

> Ok this is incorrect, The aura aspect was meant to be the area the block can be placed, the aura should be the colour of the block but blurring out from the center. They blob also need to pick up and place the block by a button press. Lets make pick up button e and drop q. They should also be able ot remove the boxes when they are in the aura zone.

Assistant Response 4.1

Corrected aura system and introduced emotion stacking:

Key changes:

- Fixed aura rendering with proper RGB color mapping
- Each aura matches its hole's color with gradient fade
- Aura acts as placement zone, not pickup trigger
- Changed `carryingColor: null` to `heldBoxIndex: null` to prevent box loss

Emotion stacking system:

- Changed from `emotion: string` to `activeEmotions: {}` (object with timers)
- Multiple emotions can be active simultaneously
- Effects combine multiplicatively:
 - Joy + Sadness: $1.5x \times 0.4x = 0.6x$ speed
 - Anger + other effects: compound wobble with other mechanics
- Emotions automatically remove when timers expire

Implemented `updateEmotionTimers()`:

- Decrements each emotion timer each frame
- Removes expired emotions from object
- Resets blob shape when sadness expires

****Status:**** Emotion stacking working correctly, aura zones precise

Session 5: Bug Fix

User Prompt 5.1

> Now it doesn't work

Assistant Response 5.1

****Identified and fixed syntax error:****

Issue: Missing closing brace in nausea control reversal block

```javascript

// Before:

```
if (blob3.activeEmotions['nausea']) {
 leftPress = !leftPress;
 rightPress = !rightPress;
 upPress = !upPress;
 downPress = !downPress;
 // Missing } here
```

// After:

```
if (blob3.activeEmotions['nausea']) {
 leftPress = !leftPress;
 rightPress = !rightPress;
 upPress = !upPress;
 downPress = !downPress;
}
```

---

Verified syntax with error checker—no errors found after fix.

**\*\*Status:\*\*** Game fully functional

--

## ## Session 6: Emotion Persistence & Color Accuracy

### ### User Prompt 6.1

> Make sure that the emotional effect doesn't disappear when a second box is placed in its zone, and please also make the aura areas the same colour as their corresponding box

### ### Assistant Response 6.1

**\*\*Enhanced emotion system and fixed color rendering:\*\***

Emotion persistence:

- Emotion timers were already set to refresh (not reset) on repeated placement
- Same color box placed again resets timer to 900 frames (~15 seconds)
- Added clarifying comments about timer refresh behavior

Color accuracy fix:

- Replaced string concatenation approach (`hole.color + alpha`)
- Implemented proper RGB color mapping:
  - Red: (255, 0, 0)
  - Green: (0, 255, 0)
  - Blue: (0, 0, 255)
  - Yellow: (255, 255, 0)
- Applied alpha transparency correctly in gradient loop

**\*\*Status:\*\*** Emotions persist perfectly, colors display accurately

--

## ## Session 7: Extended Duration, Sliding, & Edge Zones

### ### User Prompt 7.1

> Ok, please make the effect last longer for the emotion effects. and last add a new feature! When the effect ends the block slides off the platform. we introduce a small zone around the platform at the edges of the canvas. The box pushes itself off the edge and the effects reset back to neutral. The box then spawns back in a random location inside the platform zone. So basically edges make block disappear when it slides out and they reappear somewhere in the platform zone. Good!

### ### Assistant Response 7.1

\*\*Implemented extended emotions, sliding mechanics, and respawn system:\*\*

#### Extended emotion duration:

- Increased timers from 300 frames (~5 seconds) to 900 frames (~15 seconds)
- Each emotion now lasts 3x longer for strategic gameplay

#### Sliding mechanics:

- Added `sliding`, `vx`, `vy` properties to boxes
- `slideBoxOff()` function determines nearest edge
- Boxes move at 3 pixels/frame toward nearest edge:
  - Left: vx = -3
  - Right: vx = 3
  - Top: vy = -3
  - Bottom: vy = 3

#### Edge zone system:

- Added 30-pixel danger zone border (`EDGE\_ZONE = 30`)
- Semi-transparent gray overlay shows edges on canvas
- Created `platformZone` constant defining safe play area
- Visual feedback: gray rectangles at all 4 edges

#### Respawn mechanics:

- `updateSlidingBoxes()` monitors sliding boxes
- When box exits canvas completely, triggers respawn
- Boxes respawn at random location within platform zone:
  - `random(platformZone.x + 50, platformZone.x + platformZone.width - 50)`

- `random(platformZone.y + 50, platformZone.y + platformZone.height - 50)`
- Respawned box has velocity reset to 0

Added hole tracking:

- `holes[].placedBox` tracks which box is in zone
- `holes[].emotionKey` tracks active emotion for that zone
- Allows cleanup when emotion expires

**\*\*Status:\*\*** Complete sliding and respawn cycle working

---

## ## Session 8: Control & Victory Condition

### ### User Prompt 8.1

> Ok, finally we have to make sure that if the blocks are picked up and placed outside of the aura the effect disappear. This makes sure that the aura has some importance and that you can remove effect on your own. And for the end, if you are able to get all boxes in their aura zone put a large text VICTORY! overlaying everything and underneath a restart button.

### ### Assistant Response 8.1

**\*\*Implemented emotion removal on pickup and complete victory system:\*\***

Emotion removal on pickup:

- Modified E key handler to detect if box is in hole
- When picking up box from aura zone:
  - Find associated hole using `holes[].placedBox`
  - Delete emotion from `blob3.activeEmotions`
  - Reset shape if sadness was removed
  - Mark hole as empty
- Gives player strategic control over active effects

Victory condition system:

- `checkVictory()` function verifies all zones filled
- Checks if all 4 holes have `placedBox !== null`
- Sets `victoryAchieved = true` flag

Victory screen implementation:

- Semi-transparent black overlay (180 alpha)
- Large yellow "VICTORY!" text at 72pt
- Green button with rounded corners
- White "Press R to Restart" text at 24pt
- Persistent until R key pressed

Game reset function:

- `resetGame()` initializes all game state:
  - `victoryAchieved = false`
  - All emotions cleared from `activeEmotions`
  - Blob reset to center position (320, 180)
  - All boxes returned to spawn positions
  - All holes cleared of `placedBox` and `emotionKey`
  - Blob shape reset to 48 points
  - Screen tint cleared
- Accessible via R key when victory achieved

\*\*Status:\*\* Complete game loop implemented—play, win, restart

---

## ## Session 9: Documentation & Integration

### ### User Prompt 9.1

> Ok this is incorrect, The aura aspect was meant to be the area the block can be placed, the aura should be the colour of the block but blurring out from the center. They blob also need to pick up and place the block by a button press. Lets make pick up button e and drop q. They should also be able to remove the boxes when they are in the aura zone.

### ### Assistant Response 9.1

Created comprehensive development transcript including:

- 8 development iterations with prompt/response pairs
- Final game mechanics summary with controls table
- Technical implementation details and data structures

- Key functions overview
- Design philosophy explaining emotional literacy through gameplay

**\*\*File:\*\*** 'DEVELOPMENT\_TRANSSCRIPT.md'

---

### **### User Prompt 9.2**

> Integrate this subtly into the README, dont add to much and compress: [Game Mechanics Summary]

### **### Assistant Response 9.2**

Integrated compressed game information into README.md:

**\*\*Description section:\*\*** 2-sentence overview of core mechanics

- Emotion-driven puzzle game concept
- Stacking emotions and winning condition

**\*\*Interaction Instructions:\*\*** Concise controls and gameplay flow

- 4 control keys listed (WASD, E, Q, R)
- 3-sentence gameplay explanation
- Victory condition noted

Removed placeholder text, kept professional and scannable format.

**\*\*Status:\*\*** README updated with essential information

---

**\*\*Session Duration:\*\*** Jan 27 - Jan 29, 2026

**\*\*Total Development Time:\*\*** Single extended session with multiple iterations

**\*\*Final Status:\*\*** Game complete and fully functional