

Process & Decision Documentation

Project/Assignment Decisions

I created a simple pet-care simulator with a silly pet friend, using the coding example from class as a foundation. The story branches based on whether the player chooses to feed or play with the pet friend. Originally, there was a correct choice that'd lead to a good ending, but I decided not to show the pet's internal needs and make the outcome random instead (drawing inspiration from the coding example). I chose to remove the idea of a correct path to make the game experience and logic a little more experimental, interactive, and unpredictable.

Role-Based Process Evidence

- Started with the class example and adapted it into a simple pet care simulator with game, win, and lose states.
- Removed the start screen since it didn't feel necessary for this project.
- Implemented two main interaction buttons (Feed and Play) to create a simple branching choice system.
- The original idea was to build the game around a "correct choice" mechanic based on the pet's visible needs but ran into Github commit issues, resulting in timeshortage.
- Revised the mechanic to hide the pet's internal needs and randomize outcomes to remove a predictable winning path.
- Debugged image loading issues by restructuring preload() into main.js and separating asset loading from screen files.
- Iteratively simplified the logic to keep the system minimal while still maintaining a sense of unpredictability.
- Tested the game myself multiple times to ensure screen transitions, feedback text, and restart system functioned smoothly.

See appendix & commit history

Entry Header

Name: Kaitlyn Subcharoen

Primary responsibility for this work: Complete side quest

Goal of Work Session

The goal of this side quest was to build a branching interactive simulator using multiple game states (game, win, lose) across separate files. I worked on implementing player interaction (feed or play), refining screen transitions, simplifying the logic so that the pet's internal needs were hidden. I also debugged image loading and screen state issues to ensure the experience flowed smoothly.

Tools, Resources, or Inputs Used

- p5.js
- Coding example week3
- Lecture note on game stages
- Self-playtesting
- Procreate – drawing a pet friend

GenAI Documentation

No GenAI used for this task

Appendix

The screenshot shows a GitHub pull request interface. The title of the pull request is "Update main.js". The commit message is "Update main.js - ksubchar/ksubchar_sidequest_W3@bdd23e3". The commit was made by "ksubchar" 3 weeks ago. The file changed is "main.js". The diff shows the following changes:

```
// We store the "name" of the current screen as a string.  
// Only one screen should be active at a time.  
let currentScreen = "game"; // "game" | "win" | "lose"  
+ let petLoveImg;  
+ let petDoneImg;  
+ let petStartImg;  
+  
+ // Load pet images  
+ function preload() {  
+   petLoveImg = loadImage("assets/images/petLove.png");  
+   petDoneImg = loadImage("assets/images/petDone.png");  
+   petStartImg = loadImage("assets/images/petStart.png");  
}  
// -----  
// setup() runs ONCE at the beginning
```

Debugged image loading issues:

- I originally used the `preload()` function multiple times across different files.
- Asked my roommate if this is why the game stop showing up, and the answer was yes, so I debugged it by removing all `preload()` functions in other files and called it once in `main.js`

Filter files... win.js

1 file changed +6 -2 lines changed

```

13 13 // -----
14 14 // drawWin() is called from main.js
15 15 // only when currentScreen === "win"
16 + @@ -13,6 +13,7 @@
16 17     function drawWin() {
17 18         // Green-tinted background to communicate success
18 19         background(200, 255, 200);
19 + @@ -22,11 +23,14 @@
20 +         // Main success message
21 +         textSize(40);
22 -         text("Good job! Your pet loves you", width / 2, 300);
23 +         text("Good job! Your pet loves you", width / 2, 200);
24 +         // Instruction text
25 +         textSize(20);
26 +         text("Click or press R to keep taking care of your pet.", width / 2, 360);
27 +         text("Click or press R to keep taking care of your pet.", width / 2, 260);
28 +         // Pet Image
29 +         image(petLoveImg, 280, 300, 240, 320);
30 +     }
31 +
32 +     // Pet Image
33 +     image(petLoveImg, 280, 300, 240, 320);
34 +
35 +
36 // -----

```

Comments 0 Lock conversation

Building on the coding example, I changed the text to reflect the pet simulator and added an image for the win screen

Comparing e00a84e6dfa6e3db722ba4749adea2155575e0d... -1845695b22faa38204b39e0fdeee8ed8325f1251 - ksubchar/ksubchar... Update lose.js - ksubchar/ksubchar_sidequest_W3@2f64981

1 file changed +6 -2 lines changed

```

9 9 // -----
10 10 // drawLose() is called from main.js
11 11 // only when currentScreen === "lose"
12 + @@ -9,6 +9,7 @@
12 13     function drawLose() {
13 14         // Red-tinted background to communicate failure
14 15         background(255, 210, 210);
15 + @@ -18,11 +19,14 @@
16 +         // Main message
17 +         textSize(40);
18 -         text("Oh no, your pet ran away...", width / 2, 300);
19 +         text("Oh no, your pet ran away...", width / 2, 200);
20 +
21 +         // Instruction text
22 +         textSize(20);
23 +         text("Click or press R to take better care of your pet.", width / 2, 360);
24 +         text("Click or press R to take better care of your pet.", width / 2, 260);
25 +         // Pet Image
26 +         image(petGoneImg, 250, 280, 280, 400);
27 +
28 +         // Pet Image
29 +         image(petGoneImg, 250, 280, 280, 400);
30 +
31 +
32 // -----

```

Comments 0 Lock conversation

Same thing here as well, building on the coding example, I changed the text to reflect the pet simulator and added an image for the loose screen.