

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.

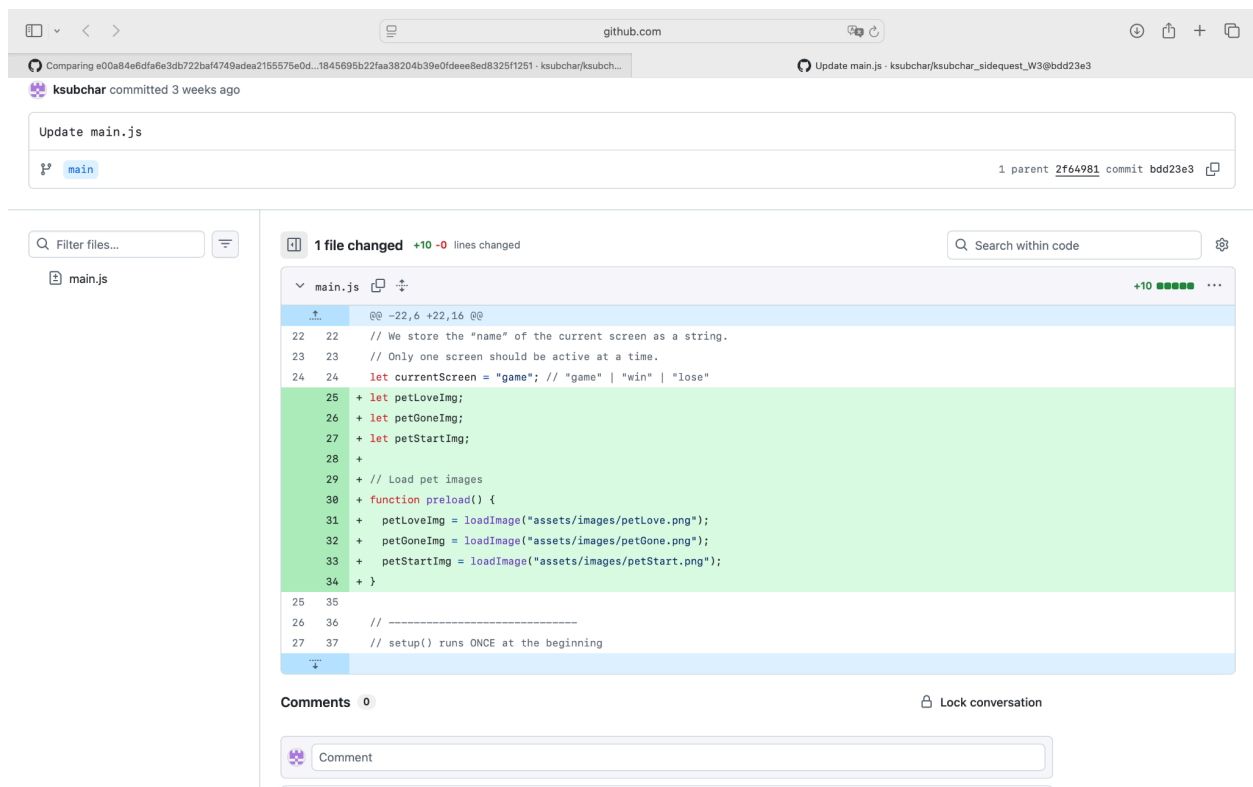
Tools, Resources, or Inputs Used

- 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



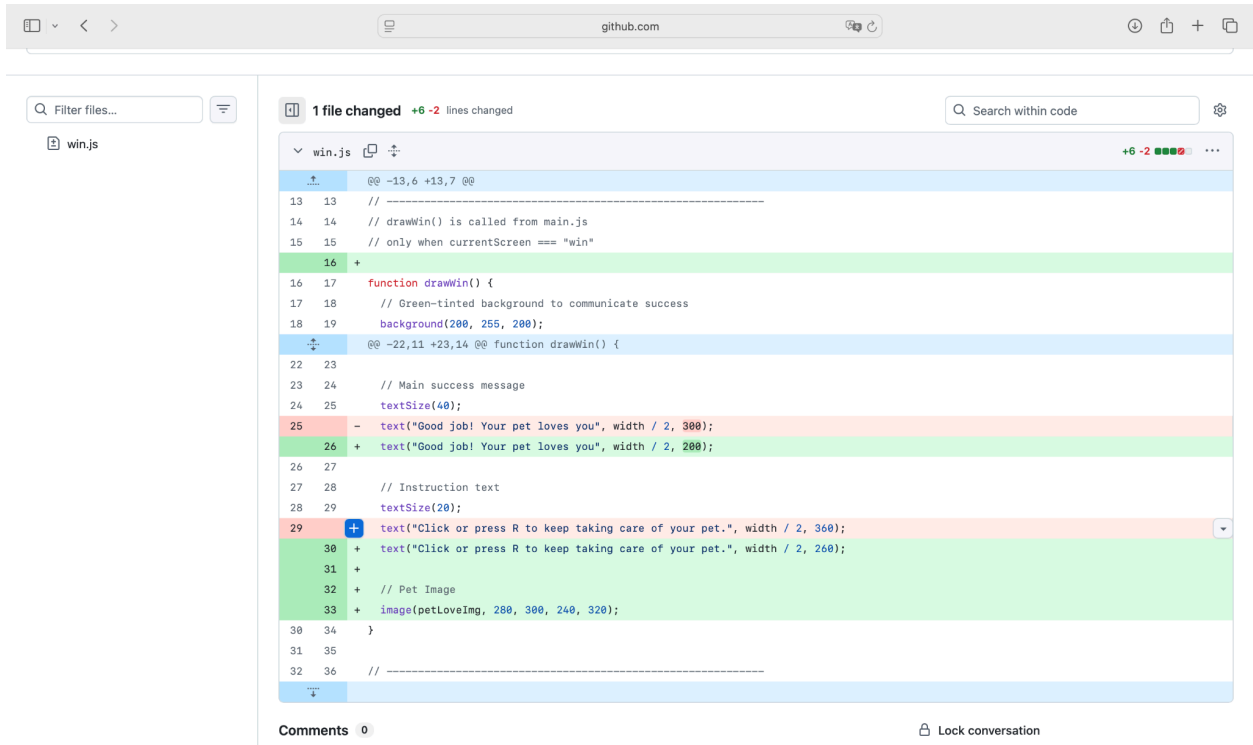
The screenshot shows a GitHub commit page for the file 'main.js'. The commit message is 'Update main.js'. The code changes are as follows:

```
@@ -22,6 +22,16 @@
22 22 // We store the "name" of the current screen as a string.
23 23 // Only one screen should be active at a time.
24 24 let currentScreen = "game"; // "game" | "win" | "lose"
25 + let petLoveImg;
26 + let petGoneImg;
27 + let petStartImg;
28 +
29 + // Load pet images
30 + function preload() {
31 +   petLoveImg = loadImage("assets/images/petLove.png");
32 +   petGoneImg = loadImage("assets/images/petGone.png");
33 +   petStartImg = loadImage("assets/images/petStart.png");
34 + }
25 35
26 36 // -----
27 37 // setup() runs ONCE at the beginning
```

The commit page also shows a search bar for filtering files and a search bar for searching within the code. The file 'main.js' is selected in the left sidebar. The commit message is 'Update main.js'. The commit hash is '2f64981'. The commit is linked to the parent commit 'bdd23e3'.

Debugged image loading issues:

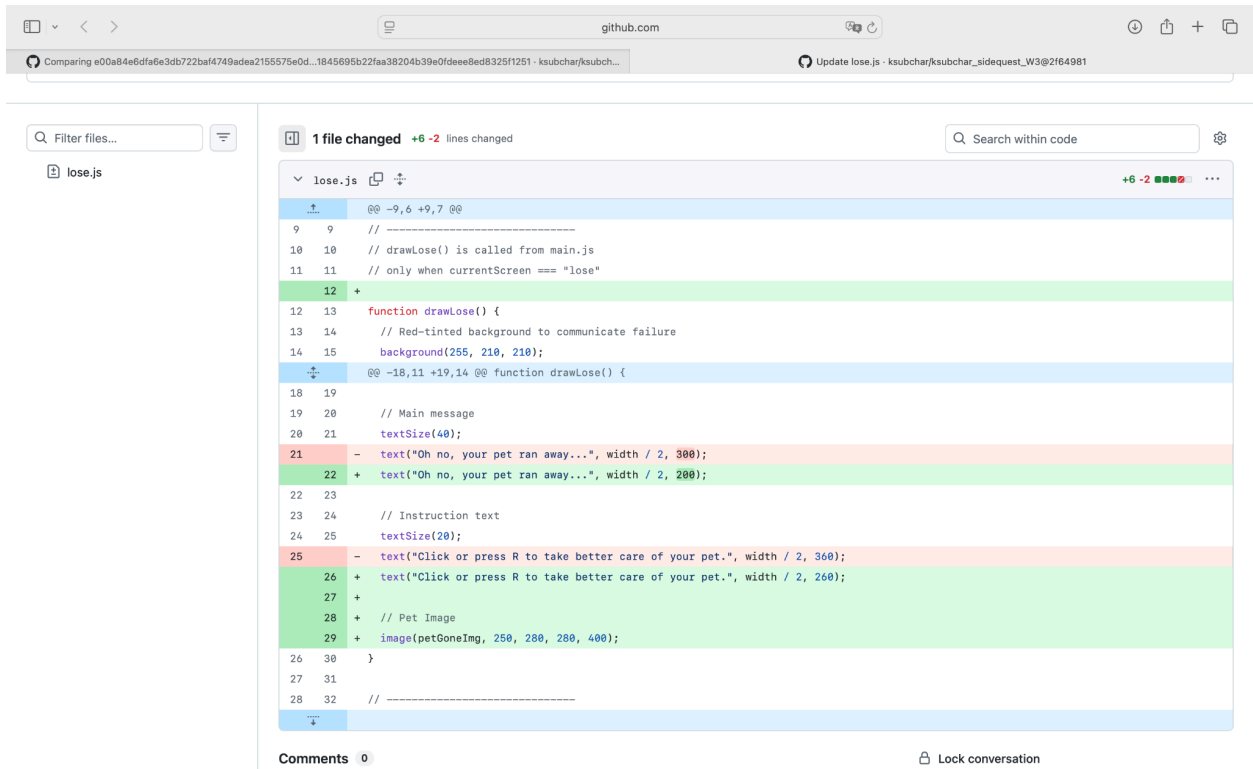
- I originally used the preload() function multiple times across different files.
- Asked my roommate if this is why the game stop showing up, and the answer was yes, so I debugged it by removing all preload() functions in other files and called it once in main.js



The screenshot shows a GitHub pull request interface. On the left, a sidebar contains a search bar labeled "Filter files..." and a file list with "win.js" selected. The main area displays a diff for "win.js", indicating "1 file changed +6 -2 lines changed". The code is shown with line numbers 13 to 36. Line 16 has a green background with a "+" sign. Line 25 has a red background with a "-" sign. Line 26 has a green background with a "+" sign. Line 29 has a red background with a "+" sign. Line 30 has a green background with a "+" sign. Line 31 has a green background with a "+" sign. Line 32 has a green background with a "+" sign. Line 33 has a green background with a "+" sign. Line 34 has a green background with a "+" sign. Line 35 has a green background with a "+" sign. Line 36 has a green background with a "+" sign. The code includes comments and function calls like "drawWin()", "background()", "text()", and "image()". At the bottom, there are "Comments 0" and a "Lock conversation" button.

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

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



The screenshot shows a GitHub pull request interface. On the left, a sidebar contains a search bar labeled "Filter files..." and a file list with "lose.js" selected. The main area displays a diff for "lose.js", indicating "1 file changed +6 -2 lines changed". The code is shown with line numbers 9 to 32. Line 12 has a green background with a "+" sign. Line 21 has a red background with a "-" sign. Line 22 has a green background with a "+" sign. Line 25 has a red background with a "-" sign. Line 26 has a green background with a "+" sign. Line 27 has a green background with a "+" sign. Line 28 has a green background with a "+" sign. Line 29 has a green background with a "+" sign. Line 30 has a green background with a "+" sign. Line 31 has a green background with a "+" sign. Line 32 has a green background with a "+" sign. The code includes comments and function calls like "drawLose()", "background()", "text()", and "image()". At the bottom, there are "Comments 0" and a "Lock conversation" button.

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

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.