

Gen AI Documentation

Date Used: February 2nd 2026

Tool disclosure: AI Agent on Visual Studio Code

Purpose of Use: To explore the functionality of AI generated code and learn how to vibecode.

Summary of Interaction: I used the AI agent to help write the code for my interactive game.

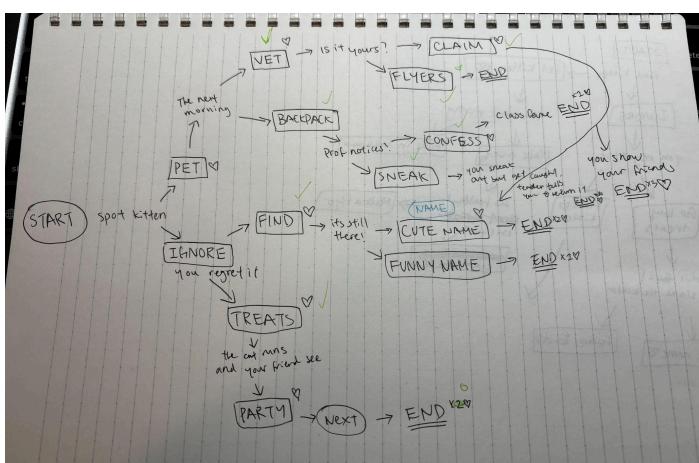
Human Decision Points: Each prompt changed and created new lines of code to create the game. I had to describe each scene and state for the game, and I also decided what each scene name would be for easy organisation.

Scope of GenAI Use: GenAI was used to write all of the code and comments. GenAI was not used to write the description of anything in the Read Me file.

Limitations or Misfires: The AI wasn't able to generate graphic elements, only simple drawings. The AI generated code would sometimes not work properly and I would need to troubleshoot, reprompt, or try different solutions.

Summary of Process (Human + Tool)

- Written outline of the game (decision tree)
- Prompting and iteration
- Play testing
- Moments of failure



```
← DESIGNING A TITLE SCREEN FOR A G... □
Restore Checkpoint

when the user clicks the button "go to the store to buy treats", go to scene:TREATS

Scene:TREATS
Text: "You found some fish treats! You're about to feed the kitten when it suddenly jumps up and snatches the fish. Oh no! The kitten is running towards the park!"

Next: Text:"huff You look up and see a group of familiar faces sitting around a small fluffy blob. Looking closer, it's your friends! I guess the little kitten just wanted a big party!" - end

✓ Added new scenes and updated routing
I'll add the TREATS scene path for when the user buys treats at the store.

✓ Edited constants.js +2 -0
✓ Edited game.js +18 -4
✓ Summarized conversation history
Let me check the current structure to fix the replacements:
```

Decision Points & Trade-offs

- I requested hearts on the canvas and the AI used the draw function to draw a heart using two circles and a triangle. It didn't look like a heart so I changed the heart graphic to a heart emoji instead.
- The AI was organizing the scenes in a confusing way so I decided to name each scene and have the AI use those as constant variables.
- Some buttons that were generated were not spaced well on the canvas so I had to prompt the AI to fix that.

Verification & Judgement

- I verified and judged each prompt outcome through playtesting, making sure everything aligned with my intentions

APPENDIX

On the start page, show a title screen with a big "Kitten Adoption Quest" title, pink and pastel style css. "Start Adventure" button, and "Instructions" button

- ✓ Read HTML, JavaScript, and CSS files

Let me read the rest of the `js start.js` file and `# style.css` to understand the full structure:

- ✓ Read JavaScript and CSS files

Now I'll update the `js start.js` file to show "Kitten Adoption Quest" with pink and pastel styling, and update the buttons to "Start Adventure" and "Instructions". I'll also update the `# style.css` to support pastel colors:

Let me fix that and use the correct structure:

- ✓ Edited `js start.js +13 -13`

Perfect! I've updated the start page with the following changes:

- ✓ **Title:** Changed to "Kitten Adoption Quest" with a larger font size (64px) and soft pink color
- ✓ **Background:** Updated to a pastel pink background (255, 220, 240)
- ✓ **Buttons:**

- "Start Adventure" button (was "START")
- "Instructions" button (was "INSTRUCTIONS")
- ✓ **Colors:** Applied a cohesive pastel pink and purple theme throughout:
 - Title text in soft pink
 - Button backgrounds in light pastel pink
 - Hover state with brighter pastel pink
 - Text in dark pastel purple

put some cute kitten paw prints in the background of the title page

- ✓ Read lines 1 to 60 of file

Now I'll add a helper function to draw paw prints and call it in the `drawStart()` function to create a cute background pattern:

- ✓ Edited `js start.js +3 -0`

Now I'll add the paw print drawing function at the end of the file:

- ✓ Read lines 120 to 141 from file

- ✓ Edited `js start.js +39 -0`

Done! I've added cute kitten paw prints to the background of your title page. Here's what I added:

- ✓ **Paw print function:** Created `drawPaw()` and `drawPawPrints()` functions that draw adorable paw prints with a main pad and 4 toe pads
- ✓ **Background placement:** Paw prints are scattered at various positions and sizes across the page, drawn before the title and buttons so they stay in the background
- ✓ **Pastel styling:** The paw prints are in semi-transparent pastel pink (255, 180, 220) to complement your pastel pink theme without overwhelming the design

The paw prints will appear playfully scattered across the background, giving your "Kitten Adoption Quest" title page a cute, thematic look!

center all the elements on the title page. add a short description under the title "A funny story with cat chaos!"

✓ Read lines 13 to 65 of start.js

Now I'll update the drawStart() function to add the description and adjust positioning for better centering:

✓ Edited `start.js +9 -3`

Now I need to update the button position references in the startMousePressed() function:

✓ Edited `start.js +2 -2`

Perfect! I've updated the title page with the following changes:

✓ **Centered layout:** All elements are now properly centered vertically with better spacing

- Title moved to y: 120
- Description added at y: 180
- Buttons repositioned to y: 300 and y: 410

✓ **New description:** Added "A funny story with cat chaos!" in a complementary pastel pink color (slightly darker than the title for contrast)

✓ **Updated button positions:** Updated both the visual display and the click detection in startMousePressed() to match the new layout

Everything is now centered and nicely spaced on your title page!



Claude Haiku 4.5 • 1x

move everything down so the tile is at 1/3 of the screen

✓ Updated title and button positions accordingly

✓ Edited `start.js +6 -6`

Done! The title is now positioned at 1/3 of the screen height, with the description and buttons proportionally spaced below it. Everything will scale responsively based on your canvas size.



have the paw prints be in a line, like a cat walked over the screen

✓ Read lines 150 to 180 of file

✓ Edited `start.js +18 -15`

Done! The paw prints now form a diagonal line across the screen, like a cat walked from the bottom-left to the top-right. They're evenly spaced along the path for a natural walking effect.



make it so that the paw prints are in a walking pattern, not a direct straight line

✓ Modified drawPawPrints for zigzag pattern

✓ Edited `start.js +9 -5`

when the game starts, the player's heart state should be at 0. the hearts can increase or decrease depending on the user choices. the hearts go up to 3 hearts. I want to show a heart meter at the top right of the screen.

- ✓ Read main.js and game.js files

Now let me read the rest of `game.js` to see if there's already a triggerRandomOutcome function:

- ✓ Read `game.js`, lines 100 to 129

Perfect! Now I'll add the heart system. I need to:

1. Add a global `hearts` variable in `main.js` initialized to 0
2. Reset it to 0 when the game screen is entered
3. Add a heart meter display function in `game.js`
4. Draw the heart meter at the top right of the game screen

Let me fix the array format:

- ✓ Edited `main.js +2 -1`
- ✓ Edited `main.js +4 -1`

Now let me update `game.js` to add the heart meter display:

- ✓ Edited `game.js +3 -0`
- ✓ Edited `game.js +50 -0`

Perfect! I've added a heart system to your game:

- ✓ **Global hearts variable:** Added `hearts` to `main.js`, initialized to 0
- ✓ **Heart meter display:** Shows 3 hearts in the top right corner of the game screen
- ✓ **Visual feedback:**

after the user clicks "start adventure" the next page should show a text box. the text box should read "a droplet of rain falls on your face...it's raining". There's a "next" button on the bottom right of the screen.

- ✓ Read `game.js` lines 1 to 50

- ✓ Edited `game.js +40 -17`

The mouse input function is already correct. Let me just verify the changes were made to the first two replacements:

- ✓ Read `game.js`, lines 1 to 80

I see there's a duplicate cursor feedback section. Let me clean that up:

- ✓ Read `game.js`, lines 65 to 120
- ✓ Edited `game.js +0 -6`

Perfect! I've updated the game screen with:

- ✓ **Story text box:** Displays "a droplet of rain falls on your face...it's raining" in a beige text box on the screen
- ✓ **Next button:** Moved to the bottom right (position 700, 700) with the label "Next"
- ✓ **Heart meter:** Still visible at the top right showing the player's hearts
- ✓ **Clean layout:** Removed the old "Game Screen" instructions text

When the player clicks "Start Adventure" from the title screen, they'll now see the story text with the Next button to continue.



Claude Haiku 4.5 • 1x

after clicking "next", the next screen's text box will say "You spot a kitten on the side of the road, it seems to be shivering. What now?". Two action buttons are underneath the text box: the first button says "Pet the kitten (+1) heart", the second button says "Walk past the kitten". Make the buttons kind of cute and rounded.

- ✓ Read lines 20 to 50 of main.js

Now let me add a game scene tracker to `main.js` and then update `game.js` to handle multiple scenes:

Let me read the exact content again:

- ✓ Read `main.js`, lines 18 to 30
- ✓ Edited `main.js` `+1 -0`

Now I also need to reset `gameScene` when entering the game screen:

- ✓ Edited `main.js` `+1 -0`

Now let me update `game.js` to handle multiple scenes:

- ✓ Read `game.js`, lines 1 to 60
- ✓ Edited `game.js` `+88 -18`

Perfect! I've updated your game with a scene-based system:

- ✓ **Scene 1:** Shows the rain intro with a "Next" button
- ✓ **Scene 2:** Shows the kitten encounter with two cute choice buttons:
 - "Pet the kitten (+1) heart" - adds 1 heart (max 3)
 - "Walk past the kitten" - no heart change
- ✓ **Cute button design:** Choice buttons have:

the background of every page should be the same pastel pink as the title page. the text box should be a slightly darker shade of the pink, and the size of the text box should fit the height of the text.

- ✓ Read lines 40 to 110 in `game.js`

Now I'll update the background color and the text box styling:

- ✓ Edited `game.js` `+26 -7`

I need to use a different approach since p5.js doesn't have a `textHeight()` function. Let me fix that:

Let me use the correct `multi_replace` format:

- ✓ Read lines 85 to 130 in `game.js`

Let me use `multi_replace`:

- ✓ Edited `game.js` `+11 -14`

Perfect! I've updated the game screen with:

- ✓ **Pastel pink background:** Changed from yellow to match the title page (255, 220, 240)
- ✓ **Darker pink text box:** Now uses a slightly darker shade (245, 200, 230) with a matching pink border
- ✓ **Dynamic text box height:** The text box now adjusts its height based on the text content instead of being a fixed 500px height:

- Short text: 150px height
- Medium text: 200px height
- Long text: 250px height

The result is a more cohesive, aesthetic design that maintains the cute pastel pink theme throughout all game pages!



the next button should be closely under the text box. the "hearts" should be a bigger text size. have show three outlines hearts next to the "Hearts:" to represent a heart bar

- ✓ Read specified lines from game.js

Now I'll update the button positions and heart meter:

- ✓ Edited game.js +11 -10

The gameMousePressed replacement already matched. Let me verify the changes:

- ✓ Read lines 15 to 30 and 125 to 165

Perfect! All the updates have been applied successfully:

✓ **Next button repositioned:** Moved from y: 700 to y: 380, so it now sits much closer right under the text box

✓ **Larger Hearts text:** Increased font size from 14 to 24 for better visibility

✓ **Outline hearts:** The hearts now display as outline hearts with:

- White/empty fill for unfilled hearts
- Red fill for earned hearts
- Gray stroke outline around all hearts to clearly show the heart bar

The heart meter now looks like a proper health/affection bar with three outline hearts that fill as the player earns them!



Claude Haiku 4.5 • 1

after clicking next, the screen should change to the next prompt/question

- ✓ Read lines 40 to 85 of game.js

I see - you want to add a third scene that appears after the player makes a choice. Let me add that:

- ✓ Edited game.js +45 -6

Perfect! I've added a third scene to the game flow:

✓ **Scene 1:** "a droplet of rain falls on your face...it's raining" → Click "Next"

✓ **Scene 2:** "You spot a kitten on the side of the road, it seems to be shivering. What now?" → Choose between two buttons

✓ **Scene 3 (NEW):** Shows the result of your choice:

- If you pet the kitten: "The kitten purrs happily. You've made a new friend!"
- If you walk past: "You walk on, leaving the kitten behind."
- Click "Next" to proceed to win/lose screen

The flow now progresses through each prompt as you click Next, and your choice about the kitten affects the message in scene 3!

combing scenes 1 and 2: "A droplet of rain falls on your face...it's raining. You see a small kitten on the side of the road, it seems to be shivering. What do you do?" have two buttons under the text box with the two choices i wrote before.

✓ Read lines 50 to 90 of game.js

✓ Edited `game.js +14 -17`

Let me check the exact current format:

✓ Read lines 130 to 290 in file

Great! I can see the first and last replacements already succeeded. Now I need to update the scene functions. Let me check what's currently there:

✓ Read `game.js`, lines 60 to 110

Perfect! Scene 1 is already correct. Now I just need to remove scene 2 (the duplicate kitten choice) and rename scene 3 to scene 2:

✓ Edited `game.js +1 -17`

Perfect! I've combined scenes 1 and 2 into a single opening scene:

Scene 1 (combined): "A droplet of rain falls on your face...it's raining. You see a small kitten on the side of the road, it seems to be shivering. What do you do?"

- Shows both choice buttons underneath: "Pet the kitten (+1) heart" and "Walk past the kitten"

Scene 2 (formerly scene 3): Shows the result of your choice

- If you pet the kitten: "The kitten purrs happily. You've made a new friend!"
- If you walk past: "You walk on, leaving the kitten behind."
- Click "Next" to proceed to win/lose

make sure there is a gap between the two choice buttons

✓ Adjusted button positioning and updated todo list

I'll add a short TODO and then patch the two choice buttons' positions and widths.

✓ Created 2 todos

✓ Edited `game.js +4 -4`

✓ Completed: *Update choice button layout (1/2)*

Done — I moved and narrowed the choice buttons so they no longer overlap:

- `choiceBtn1 : x=235, w=300`
- `choiceBtn2 : x=565, w=300`

Next: visually verify the gap in the running canvas; want me to run any checks or further tweak spacing or vertical position?

can you make the hearts out of the white and red heart emojis? blank hearts are white, filled hearts are red, and negative hearts are black.

- ✓ Evaluated heart representation and updated rendering

I'll update `drawHeartMeter` in `game.js` to render emoji hearts (white, red, black) and handle negative heart values. Applying the patch now.

- ✓ Edited `game.js` `+19 -15`

Done — the heart meter now uses emoji:

- White heart: ❤ (empty)
- Red heart: ❤ (filled)
- Black heart: 💔 (negative)

Behavior:

- If `hearts >= 0` then the first `hearts` slots show red, the rest white.
- If `hearts < 0` then the first `abs(hearts)` slots show black, the rest white.

Next: want me to clamp `hearts` changes (so it can go negative) or update any UI spacing further?

after the user clicks button 1 (pet the kitten), go to scene 2: "The kitten purrs happily and follows you home. I think it likes you! The next morning...". Button 1: "Take it to the campus vet" (+1 heart). Button 2: Hide it in your backpack."

- ✓ Modified game flow and created patch

I'm going to add button objects for the vet and backpack choices, track whether the kitten is with the player, and update scene rendering and mouse handling accordingly. I'll patch `game.js` now.

- ✓ Edited `game.js` `+21 -0`

- ✓ Edited `game.js` `+30 -27`

- ✓ Identified bug and repaired code structure

Applying two edits: 1) complete and close `drawScene1()` so it shows both choice buttons; 2) restore `drawStoryBox(storyText)` function by wrapping the orphaned text block in the function header and moving it back into place.

- ✓ Troubleshoot patch issues and applied code patch

- ✓ Edited `game.js` `+9 -0`

- ✓ Fixed code structure and updated story box

- ✓ Edited `game.js` `+11 -15`

- ✓ Edited `game.js` `+21 -2`

- ✓ Fixed game scene implementation

- ✓ Edited `game.js` `+2 -0`

- ✓ Updated task management and todo list

after the user clicks button 2 (walk past the kitten), go to scene 3: "You walk away and the kitten cries continue in the distance. Regret starts to hit during your lecture. After class, you sneak back to the alleyway...". Button 1: Find the kitten and scoop it up" (+1 heart).
Button 2: Go to the store to buy treats.

Name the scenes:

Scene 1 - start

Scene 2 - kitten

Scene 3 - ignore

- ✓ Modified game scenes and added content

I'm going to update `js game.js` : add `findBtn` and `storeBtn`, change scene transitions so walk-past goes to scene 3, move vet/hide result to scene 4, and add `drawScene3` (ignore) and `drawScene4` (result) plus mouse handling. I'll apply a focused patch to `js game.js`.

- ✓ Edited `js game.js +16 -0`
- ✓ Searched for text and implemented updates
- ✓ Edited `js game.js +2 -0`
- ✓ Edited `js game.js +38 -1`
- ✓ Read and modified game.js mouse handler logic
- ✓ Edited `js game.js +15 -5`

Done — updates applied:

- Walk-past now sets `gameScene = 3` (ignore).
- Added `findBtn` and `storeBtn` and `drawScene3()` to show the ignore story and those choices.
- Vet/hide now advance to `gameScene = 4` (results), and ignore choices (find/store) also advance to scene 4, which shows appropriate result text.
- `gameMousePressed()` updated to handle scenes 1–4 and adjust hearts/lastAction accordingly.

after the user clicks the button "walk past the kitten" in scene 1, go to scene IGNORE

- ✓ Summarized conversation history
- ✓ Updated scene files and planned modifications

I'll update the todo list and then edit `js game.js` to route "Walk past the kitten" to the IGNORE scene.

- ✓ Managed and tracked todo items for planning

Now I'll search `js game.js` for references to scene 3 and replace numeric checks with a named scene string, then patch the click handler.

- ✓ Searched for text and finalized patch

Patching `js game.js` to use the 'ignore' scene string for scene 3 checks and the walk-past assignment.

- ✓ Edited `js game.js +1 -1`
- ✓ Patching game function
- ✓ Edited `js game.js +2 -2`

Done — I updated the todo list and patched `js game.js` so the "Walk past the kitten" button goes to the IGNORE scene.

- Files changed:
 - `js game.js`:
 - draw routing: replaced numeric scene-3 check with `gameScene === 'ignore'`.
 - click handler: `choiceBtn2` (walk past) now sets `gameScene = 'ignore'`.
 - later input check for scene 3 updated to `gameScene === 'ignore'`.

Progress: routing now uses the 'ignore' scene string and the walk-past button advances to that scene. Next: open `index.html` in a browser and verify the sequence (Start → Walk past → IGNORE choices). Want me to run a quick visual check or open the page for you?

when I click the button "walk past the kitten" on scene 1, nothing happens. Make sure this button takes me to the next scene: IGNORE

✓ Debugging game scene issues

I'll inspect `js main.js` and `js game.js` to find why the walk-past click isn't advancing, then patch the code to ensure the button click sets `gameScene` to the IGNORE state and the draw/input checks use the same identifier. I'll also verify `isHover()` and `gameScene` initialization.

✓ Debugged button functionality and fixed issues

I'll patch `js main.js` to reset `gameScene` only when entering the `game` screen (tracking previous screen), preventing an immediate reset back to scene 1 after clicking. I'll update the file now.

✓ Edited `js main.js +8 -4`

Patch applied: `js main.js` now only resets `hearts` and `gameScene` when entering the `game` screen (uses `prevScreen`).

Progress: this prevents the immediate reset that made the "Walk past the kitten" click appear to do nothing. Please open `index.html` in your browser and test: Start → Walk past the kitten should now go to the IGNORE choices. Want me to run a quick browser verification command suggestion?

after the user clicks button 1 (pet the kitten), go to sceneKITTEN: "The kitten purrs happily and follows you home. I think it likes you! The next morning...". Button 1: "Take it to the campus vet" (+1 heart) . Button 2: Hide it in your backpack."

after the user clicks button 2 (walk past the kitten), go to sceneIGNORE: "You walk away and the kitten cries continue in the distance. Regret starts to hit during your lecture. After class, you sneak back to the alleyway...". Button 1: Find the kitten and scoop it up" (+1 heart). Button 2: Go to the store to buy treats.

✓ Troubleshoot screen handling and confirmed code placement

I'll read `js main.js` to confirm where `prevScreen` gets updated and fix it if it's in the wrong place.

✓ Updated screen handling and evaluated logic

I'll patch `js main.js` to move `prevScreen = currentScreen` inside `draw()` so the previous screen updates each frame. This will stop repeatedly resetting `gameScene` and allow scene changes to persist.

✓ Edited `js main.js +1 -1`

Fixed — `prevScreen` now updates inside `draw()`, so the game scene resets only on the frame you enter the game screen. This should allow clicks (pet / walk past) to change `gameScene` and persist.

Next step: reload the page and test the flow:

- Start → click "Pet the kitten" → verify overlay shows `scene: KITTEN` and vet/hide buttons appear.
- Start → click "Walk past the kitten" → verify overlay shows `scene: IGNORE` and find/store buttons appear.

If anything still fails, tell me the overlay's `scene` value and any console logs you see and I'll continue.



GPT-5 mini • 1x

after (take to campus vet)
New scene: CLAIM

Text: "You take the kitten to the campus vet. It gets checked over and it's healthy! The vet asks if it's your kitten..."

Button 1: "The kitty is mine!" --> scene: NAME
Button 2: "No, I just found it" --> scene: FLYER

- ✓ Managed and tracked todo items for planning

I'll update `constants.js` to add CLAIM, NAME, and FLYER scenes, then update `game.js` to add the new scene drawing and routing logic.

- ✓ Read `constants.js`

- ✓ Edited `constants.js +3 -0`

Now I'll read `game.js` to find where to add the scene buttons and routing logic.

- ✓ Read `game.js`, lines 40 to 80

I'll add buttons for the CLAIM scene after the existing buttons.

- ✓ Edited `game.js +17 -0`

Now I'll read the drawGame routing section to add the CLAIM scene check.

- ✓ Read `game.js`, lines 85 to 120

- ✓ Edited `game.js +2 -0`

Now I'll find where the vet button is clicked and update it to go to CLAIM scene instead of RESULT.

after clicking "the kitty is mine" button, add 1 heart, go to new scene: NAME

Text: "What should we name our new kitten friend?"

Button 1: Something cute --> Scene:CUTENAME
button 2: Something funny: FUNNYNAME

- ✓ Added hearts and updated scene routing.

I'll update `constants.js` to add CUTENAME and FUNNYNAME, then modify the NAME scene to show the naming choice buttons, add the heart increment, and create the new scene functions.

- ✓ Read `constants.js`, lines 1 to 10

- ✓ Edited `constants.js +2 -0`

Now I'll add buttons for the naming choices and update the claimBtn click handler to increment hearts:

- ✓ Read `game.js`, lines 60 to 90

- ✓ Edited `game.js +17 -0`

Now I'll update the drawSceneName function to show the naming choice buttons instead of a Next button:

- ✓ Read lines 240 to 270 and 275 to 305

- ✓ Edited `game.js +26 -3`

Now add the CUTENAME and FUNNYNAME scene draw functions after FLYER:

- ✓ Edited `game.js +16 -0`

Now add CUTENAME and FUNNYNAME to the drawGame routing:

after clicking "something cute"
 Go to scene:CUTENAME

Text: "Welcoming our newest family member...Mochi!"

NEXT: end title showing game results:
 3 hearts = PERFECT ENDING! Mochi lives happily with you, cat parent forever!

✓ Updated CUTENAME scene and added ending routing.

I'll update `constants.js` to add an ENDING scene, modify the CUTENAME text, and create an ending scene that displays results based on hearts.

✓ Edited `constants.js +1 -0`

Now update the `drawSceneCuteName()` function and add ENDING to drawGame routing:

✓ Edited `game.js +1 -1`

Now add the ENDING scene draw function after the FUNNYNAME scene:

✓ Edited `game.js +19 -0`

Now add ENDING to the drawGame routing:

✓ Edited `game.js +2 -2`

Now update the CUTENAME and FUNNYNAME click handlers to go to ENDING instead of RESULT:

✓ Edited `game.js +10 -4`

Done — I've added the ENDING scene with heart-based results:

remove the "lose" page. after the ending results, the game ends. and theres only a restart button taking you back to the start page

✓ Removed lose page functionality and added restart button

I'll update the ENDING scene to display a restart button that takes you back to the start page instead of triggering a win/lose outcome.

✓ Read `game.js`, lines 100 to 120

I'll add a restart button constant after the funnyNameBtn:

✓ Edited `game.js +9 -0`

Now update the `drawSceneEnding()` function to show the restart button instead of next button:

✓ Edited `game.js +2 -2`

Now update the ENDING scene click handler to go back to start instead of triggering win/lose:

✓ Edited `game.js +4 -4`

Done — the game now ends after the ENDING scene with a Restart button that takes you back to the start page. The win/lose pages are no longer used in the flow.

Changes made:

- Added `restartBtn` constant
- Updated `drawSceneEnding()` to display the restart button instead of next button
- Updated ENDING scene click handler to go back to `currentScreen = "start"` instead of calling `triggerRandomOutcome()`

Test it: Play through any ending → Click Restart → Back to the start page ready for a new game.

add a "RESTART" button on the last ending result scene that bring the user to the start page

- ✓ Checked implementation and confirmed restart button.

The restart button is already implemented from the previous request! Let me verify it's working correctly by checking the current state:

- ✓ Searched for text `restartBtn` in game.js

Perfect! The restart button is already there. Let me verify the implementation is complete:

- ✓ Read specified lines from game.js

The restart button is fully implemented! It's:

- Defined as a button constant at the top of `js game.js` (line 111)
- Displayed in the ENDING scene (line 371)
- Has a click handler that sets `currentScreen = "start"` (line 649)

The button should appear on the final ending results screen. If it's not showing, reload the page in your browser. Try playing through to an ending and you should see the Restart button appear below the ending text.

when the "hide in backpack" button is clicked, go to new scene:BACKPACK

Scene: BACKPACK

Text: "You hide the kitten in your backpack and hope nobody notices. Mid lecture, people are alerted by a strange purring sound..."

Button 1: Confess --> Scene:CONFESS
button 2: Sneak out --> scene:SNEAK

- ✓ Added BACKPACK, CONFESS, and SNEAK scenes

I'll add the BACKPACK, CONFESS, and SNEAK scenes to the game flow.

- ✓ Edited `js constants.js +3 -0`
- ✓ Edited `js game.js +17 -0`

Now I'll update the drawGame routing to include BACKPACK, CONFESS, and SNEAK scenes:

- ✓ Edited `js game.js +6 -0`

Now I'll add the scene draw functions. Let me find where to insert them:

- ✓ Read lines 190 to 210 and 220 to 250

I'll add the BACKPACK, CONFESS, and SNEAK scene draw functions after drawScene2. Let me find the end of drawScene2:

- ✓ Read `js game.js`, lines 250 to 280
- ✓ Edited `js game.js +48 -0`

Now update the hideBtn click handler to go to BACKPACK instead of RESULT:

Scene:CONFESS

Text: "You raise your hand and carefully take sleeping kitten out of your bag. The whole class starts laughing and taking pictures!"

Next button: Text: "You're now famous across campus as the Kitten Whisperer! It was a fun time, but maybe this kitten can find a better home" - end

- ✓ Updated CONFESS scene and modified Next button

- ✓ Edited `game.js +1 -1`

Now I'll update the CONFESS click handler to go to ENDING instead of RESULT, and update the ENDING scene to show different text based on `lastAction`:

- ✓ Edited `game.js +2 -2`

Now update the `drawSceneEnding()` function to display different text based on the `lastAction`:

Let me read the file to get the exact text and use `apply_patch` instead:

- ✓ Read specified lines in game.js

- ✓ Edited `game.js +3 -1`

Done — I've updated the CONFESS scene and ending:

Changes made:

- Updated `drawSceneConfess()` text to: "You raise your hand and carefully take sleeping kitten out of your bag. The whole class starts laughing and taking pictures!"
- Updated CONFESS click handler to go to `SCENES.ENDING` instead of RESULT
- Updated `drawSceneEnding()` to check if `lastAction === "confess"` first and display: "You're now famous across campus as the Kitten

Scene:SNEAK

Text: "You make a quiet and swift exit out the door. *THUD* You accidentally bump into security!

Next: Text: "You surrender to the security man's questioning and open your bag. Suddenly, the kitten jumps out and runs off! Well, maybe you'll see it again another day." - end

I'll add a new scene constant `SNEAK2`, update the draw routing, change `drawSceneSneak()` text, add `drawSceneSneak2()`, and update click handling so Next from SNEAK goes to SNEAK2, and Next from SNEAK2 goes to ENDING. I'll patch `constants.js` and `game.js`.

- ✓ Read `constants.js`
- ✓ Edited `constants.js +1 -0`
- ✓ Read `game.js`, lines 240 to 320
- ✓ Edited `game.js +9 -1`
- ✓ Updated click handler and routing in game.
- ✓ Edited `game.js +2 -0`
- ✓ Edited `game.js +2 -2`

Done — I added a SNEAK2 scene and wired it:

- `constants.js` : added `SNEAK2`.
- `game.js` :
 - `drawSceneSneak()` now shows the bumped-into-security text.
 - Added `drawSceneSneak2()` showing the bag-opens/kitten-runs-off text.
 - `drawGame()` now routes `SCENES.SNEAK` to `drawSceneSneak2()`.
 - Clicking Next on `SCENES.SNEAK` now advances to `SCENES.SNEAK2`; Next on `SCENES.SNEAK2` goes to the ending.

Scene: FLYER, add 1 heart

Text: "Let's find this kitten a good home! Time to put my GBDA skills to use and make a nice flyer for this little guy"

Next: Text: Your flyer went viral online and the kitten found a loving new home. Yay!" - end

✓ Modifying game logic

I'll patch `game.js` to increment hearts when the player chooses the FLYER path and update the FLYER scene texts and flow to end with the restart button. I'll modify `drawSceneFlyer` and `gameMousePressed` handling.

✓ Modified flyer scene and refined game mechanics

✓ Edited `game.js +9 -0`

✓ Edited `game.js +1 -1`

✓ Updated scene logic

✓ Edited `game.js +2 -0`

Done — changes applied:

- Increment hearts when the player chooses to post a flyer (in `gameMousePressed` when `foundBtn` is clicked).
- Updated `drawSceneFlyer()` text to the requested wording.
- Clicking Next on the FLYER scene now sets `lastAction = 'flyer'` and goes to the ENDING scene.
- `drawSceneEnding()` now shows the viral flyer message when `lastAction` is 'found' or 'flyer'.

Test sequence: Start → Pet → Take to vet → Choose "No, I just found it" → Flyer scene (hearts +1) → Next → Ending shows "Your flyer went viral..." with Restart button.

when the user clicks the button "find the kitten and scoop it up", go to new scene:FIND

Scene:FIND

Text: "Phew! The kitten is still there! You feel relieved and a bit guilty. Let's name this little kitten."

Button1: "Cute name"

Button2: "Funny name"

Button1 --> Text:"Mochi! Although you didn't take the cat home, your new daily schedule now includes feeding the kitten everyday on your way to class" - end

Button2 --> Text:"Mr.Fluffyorangelover sleeps soundly in his cardboard box. I guess you'll have to visit him pretty often and feed this little guy." - end

✓ Added FIND scene and related functionalities

I'll add the FIND scene path with naming choices that lead to different endings.

✓ Read `constants.js`

Let me fix the JSON formatting:

✓ Edited `constants.js +3 -0`

Now I'll update `game.js` to route the find button to FIND scene and add the scene draw functions:

✓ Read `game.js`, lines 680 to 700

✓ Edited `game.js +81 -3`

Done — I've implemented the FIND scene path with cute and funny naming options:

when the user clicks the button "go to the store to buy treats", go to scene:TREATS

Scene:TREATS

Text: "You found some fish treats! You're about to feed the kitten when it suddenly jumps up and snatches the fish. Oh no! The kitten is running towards the park!"

Next: Text:"huff You look up and see a group of familiar faces sitting around a small fluffy blob. Looking closer, it's your friends! I guess the little kitten just wanted a big party!" - end

- ✓ Added new scenes and updated routing

I'll add the TREATS scene path for when the user buys treats at the store.

- ✓ Edited `constants.js +2 -0`

- ✓ Edited `game.js +18 -4`

- ✓ Summarized conversation history

Let me check the current structure to fix the replacements:

- ✓ Searched for text and read game.js lines

Now let me apply the remaining corrections:

- ✓ Edited `game.js +4 -0`

- ✓ Edited `game.js +16 -0`

- ✓ Searched for text and read game.js lines

I see the issue. There's a syntax error and duplicate handler for FINDFUNNY. Let me fix this:

- ✓ Edited `game.js +2 -1`

change the instructions page to fit the game. make the background of the instructions a slightly darker pink than the start page.

- ✓ Read multiple JavaScript files and lines

Now I'll update the instructions page with game-specific content and a darker pink background:

- ✓ Edited `instructions.js +16 -10`

Done! Updated the instructions page with:

- **Darker pink background** (250, 200, 230) - slightly darker than the start page's (255, 220, 240)
- **Game-specific instructions** about helping the kitten, making choices, building affection hearts, and finding different endings
- **Title changed** to "How to Play"
- **Styling updated** to match the pastel pink aesthetic with darker text colors



change the description to fit the game.

- ✓ Read README file

- ✓ Edited `README.md +7 -4`

Done! Updated the README with a description that fits the Kitten Adoption Quest game, including the game concept, heart system, branching narratives, and updated learning goals that reflect the actual project features.