

Process and Decision Documentation

Name: Mackenzie Leu

Primary responsibility for this work: Code a game with levels

Tools, Resources, or Inputs Used: VSCode AI tool (GPT-5.2 mini)

GenAI Documentation

Date Used: Feb 9, 2026

Tool Disclosure: VSCode AI tool (GPT-5.2 mini)

Purpose of Use: Write/edit code

Summary of Interaction: AI edited and added to the preexisting code based on my specifications. Specifically it created a game that is maze based with a library theme.

Human Decision Point(s): Decided on the theme of the game, the layout/flow of the game, the layout of the instructions page. Dictated that the walls would be bookshelves and there would be a key component. Described to the agent what the screens would look like and adjusted as needed.

Integrity & Verification Note: Continuously checked the live site to ensure the changes being made were accurate to what I was aiming for and describing.

Scope of GenAI Use: GenAI was not used to make decisions on how I wanted the project to turn out, just implemented the changes I requested.

Limitations or Misfires: The title page did not work well initially as it was being "drawn" and therefore the screen kept generating new book shelves every second making it look glitchy. The library at the end took many changes to make it look the way that it does (Along with the clouds).

Summary of Process (Human + Tool)

For this project, I worked alongside AI for coding, while I stayed in control of the overall idea, design choices, and iteration. I started by deciding on the game concept myself. It would be a maze-style game where levels are generated, the player must collect a key, and there would be progress through multiple levels. I wanted it to be simple but still feel complete and visually themed.

The agent helped mainly with the technical setup, like generating levels, handling player movement, adding key and exit logic, and managing game states. From there, most of the process became very back-and-forth. I tested the game, noticed things that felt off, and asked for specific changes, like making the game full screen, adjusting the information for each level of the maze, adding a library theme, or fixing bugs that appeared when replaying the game.

A lot of the visual and experience-based decisions were human-led. I guided how the start screen, instructions page, and end screen should look, and refined details like button placement, animations, window shapes, and difficulty of the final level. The agent handled the implementation, but the direction came from my feedback and judgment. Overall, the tool helped with execution, but the creative decisions and final experience were shaped by me through iteration and testing.

Appendix

User: create a simple maze-style 2d game using javascript and p5.js. build each maze level using a 2d array or json grid where each cell represents a tile type: wall, path, player start, key, and exit. use nested loops to draw the maze dynamically from the level data and place the player, key, and exit based on the symbols in the grid. let the player move with arrow keys or wasd, blocking movement into walls. require the player to collect the key before the exit works. when the player reaches the exit with the key, automatically load a second maze level. keep it beginner friendly, clean, and well commented, and include two example maze levels in the code.

GitHub Copilot: I'll create a short TODO plan first, then inspect the existing project files to avoid overwriting anything.

Created 5 todos

Now I'll read the project's `index.html` and list the folder to see what files already exist.

Read

`[(file:///Users/kenzieleu/Documents/GitHub/mleu_sidequest_w4/Week4_Example04/index.html)]`

Listing the project directory to find existing scripts and levels.

Read `[(file:///Users/kenzieleu/Documents/GitHub/mleu_sidequest_w4/Week4_Example04)]`

I'll open `levels.json`, `level.js`, `player.js`, and `sketch.js` to review existing code.

Read

```
[(file:///Users/kenzieleu/Documents/GitHub/mleu_sidequest_w4/Week4_Example04/levels.json)
```

Read

```
[(file:///Users/kenzieleu/Documents/GitHub/mleu_sidequest_w4/Week4_Example04/level.js)
```

Read

```
[(file:///Users/kenzieleu/Documents/GitHub/mleu_sidequest_w4/Week4_Example04/player.js)
```

Read

```
[(file:///Users/kenzieleu/Documents/GitHub/mleu_sidequest_w4/Week4_Example04/sketch.js)
```

Now I'll patch levels.json, level.js, player.js, and sketch.js to add key tiles and key/exit logic.

Using "Apply Patch"

```
---
```

```
---
```

Using "Apply Patch"

```
---
```

```
---
```

Using "Apply Patch"

```

```

Using "Apply Patch"

Read

[(file:///Users/kenzieleu/Documents/GitHub/mleu_sidequest_w4/Week4_Example04/sketch.js)]

Using "Apply Patch"

```

```

Updating the todo list to mark implemented files as complete.

Completed: *Update HTML* (1/5)

Done — added collectible key logic and two example maze levels.

– ****What I changed****

- levels.json: added key tiles (value 4) to both example levels.
- level.js: added ``isKey(r,c)`` and ``clearKey(r,c)`` and draws a simple key icon.
- player.js: added ``hasKey`` flag and a small key badge on the player avatar.
- sketch.js: detects key pickup, requires the key to activate the exit, shows brief feedback if player tries to exit without a key, and resets key state on level load.

– ****How to run****

- Open the game by serving the folder and visiting it in your browser. From the project root run:

```
```bash
```

```
cd Week4_Example04
```

```
python3 -m http.server 8000
```

```
```
```

- Then open: <http://localhost:8000/>

- **Controls & behavior**
 - Move with arrow keys or WASD.
 - Collect the yellow key (tile) before the exit (glowing tile) will advance to the next level.
 - Two levels are included (progress wraps).

Would you like me to:

- Add a README note with run instructions?
- Add sound effects or a small level-complete animation?

Made changes.

User: make the size of the screen bigger. the wall blocks should look like bookshelves made of wood with different colour and sized books.

GitHub Copilot: I'll add a short todo plan, then update sketch.js to enlarge tiles and level.js to draw wooden bookshelves for wall tiles.

Starting: *Increase screen/tile size* (1/3)

Read

`[(file:///Users/kenzieleu/Documents/GitHub/mleu_sidequest_w4/Week4_Example04/sketch.js)]`

Read

`[(file:///Users/kenzieleu/Documents/GitHub/mleu_sidequest_w4/Week4_Example04/level.js)]`

Now I'll update sketch.js (increase tile size and HUD text) and level.js (replace wall drawing with wooden bookshelf rendering).

Using "Apply Patch"

...

...

Using "Apply Patch"

^^

^^

Progress update: increased tile size (TS = 64) and updated level.js so wall tiles render as wooden bookshelves with varied book widths, heights, and colors using a deterministic per-cell PRNG.

Next steps (if you want):

- I can fine-tune book spacing, colors, or add shelf shadows.
- I can run a quick visual test by opening the game in a headless preview or add instructions to serve locally.

To test locally, run:

```
^^bash
cd Week4_Example04
python3 -m http.server 8000
^^
```

Then open <http://localhost:8000/> in your browser.

Made changes.

User: make the "key none" visible by turning the text to be black

GitHub Copilot: I'll mark the small task in the todo list, then patch sketch.js to make the 'Key: none' HUD text black.

Updated todo list

Using "Apply Patch"

^^

^^

Completed: *Make 'Key: none' black* (1/1)

Done — I changed the HUD so `Key: none` uses black text in sketch.js. Would you like the HUD repositioned or styled further?

Made changes.

User: add a start screen with a start button and instructions button. the start screen should be library themed. call the game "escape the library". add a large key as the logo. the key should have teeth on it to make it look more realistic.

GitHub Copilot: I'll add a `menu` state and UI buttons in sketch.js, draw a library-themed start screen with a large key logo, and wire Start/Instructions/Back behavior. I'll update the todo list first.

Created 3 todos

Now I'll patch sketch.js to add the menu state, buttons, and drawing functions.

Using "Apply Patch"

...

...

Completed: *Add start screen* (1/3)

Progress update: I added a menu with Start and Instructions buttons, a library-themed background, and a large key logo with teeth in sketch.js. Buttons start the game or show instructions; Start loads level 1 and resizes the canvas.

Next you can:

- Test it by serving the folder and opening <http://localhost:8000/>
- Tell me if you want button styling adjusted, animated transitions, or the menu rendered in DOM/CSS instead of p5.

Made changes.

User: the library background of the start page should not move

GitHub Copilot: I'll update sketch.js to generate the menu background once into a `p5.Graphics`` buffer and reuse it, keeping the start screen background static. I'll patch ``showMenu``, ``drawMenu``, and add a helper ``makeMenuBackground()`` and a ``menuBG`` variable.

Using "Apply Patch"

^^^

^^^

Using "Apply Patch"

^^^

^^^

Done — the start screen background is now rendered once into a cached image and no longer moves.

Progress: I updated sketch.js to generate a static ``menuBG`` with deterministic shelves when showing the menu; ``drawMenu()`` now displays that cached image.

Next steps (optional):

- Add a subtle parallax or animated lights (if you want limited motion while keeping shelves static).
- Adjust colors, shelf count, or book density — tell me preferred style.

Made changes.

User: move escape the library down to be centred in between the self and capitalize properly to be "Escape the Library"

GitHub Copilot: I'll add a small todo and then patch sketch.js to capitalize the title and move it slightly lower (centred between top shelves and key logo).

Created 1 todo

Using "Apply Patch"

Read

[(file:///Users/kenzieleu/Documents/GitHub/mleu_sidequest_w4/Week4_Example04/sketch.js#1-1), lines 1 to 260]

Read

[(file:///Users/kenzieleu/Documents/GitHub/mleu_sidequest_w4/Week4_Example04/sketch.js#260-260), lines 260 to 382]

Updated todo list

Using "Apply Patch"

...

...

Completed: *Relocate and capitalize title* (1/1)

Done — I changed the menu title to "Escape the Library" and moved it down to be centered between the shelves and the key logo (sketch.js). Want the title font, size, or color adjusted further?

Made changes.

User: move the key to be above the title. the circle part of the key needs to be closer to the line part of the key

GitHub Copilot: I'll add a quick todo, then update sketch.js to place the key above the title and bring the key head closer to the shaft.

Starting: *Move key above title and tighten head* (1/1)

Using "Apply Patch"

^^

^^

Using "Apply Patch"

^^

^^

Completed: *Move key above title and tighten head* (1/1)

Done — the large key logo is now above the title and the circular head is closer to the shaft (sketch.js). Want the head even closer or a different key style (rounded teeth, shadow, or color tweaks)?

Made changes.

User: remove the instructions from the title page. you should only see the instructions on the instructions page. the instructions page should have the same background as the title but there is a large white rectangle in the centre. at the top there is a title saying "Instructions" and then the instructions are listed below. the rectangle should take up 75% of the screen

GitHub Copilot: I'll remove instructions from the title page and update the instructions screen to use the static background plus a centered white panel.

Starting: *Menu instructions cleanup* (1/1)

Using "Apply Patch"

^^

^^

Completed: *Menu instructions cleanup* (1/1)

Done — I removed the instructions from the title page and updated the Instructions screen to:

- Use the same static library background (`menuBG`).
- Show a centered white panel that fills 75% of the screen.
- Display the title "Instructions" at the top of that panel and the instruction list below.

Test it by serving the folder and opening <http://localhost:8000/> and clicking "Instructions".
Want different panel styling (shadow, semi-transparency) or more instruction text?

Made changes.

User: the title page is not loading now

GitHub Copilot: I'm going to scan sketch.js for syntax issues that could prevent the menu from rendering (e.g., trailing commas or missing references). I'll read the file contents.

Read

[](file:///Users/kenzieleu/Documents/GitHub/mleu_sidequest_w4/Week4_Example04/sketch.js)

I'll fix the corrupted `setup()` function in sketch.js by replacing it with the intended initialization code (levels, canvas, buttons, styles, and showMenu). This will restore the title page.

Using "Apply Patch"

^^

^^

I restored `setup()` in sketch.js — the menu should load again. Test by serving the folder and opening <http://localhost:8000/>.

If it still fails, tell me the browser console error (or paste it here) and I'll fix it.

Made changes.

User: remove "collect the key..." from the home page. make the instructions and start button change when hover

GitHub Copilot: I'll remove the subtext from the home page and add hover effects to the Start and Instructions buttons.

Starting: *Add button hover effects and remove home subtext* (1/1)

Using "Apply Patch"

^^^

^^^

Using "Apply Patch"

^^^

^^^

Completed: *Add button hover effects and remove home subtext* (1/1)

Done — I removed the instruction subtext from the home page and added hover effects to the Start and Instructions buttons (sketch.js). Hovering slightly changes background color and scales the button.

Test by reloading the page; let me know if you want a stronger hover effect (shadow, color change, or pointer cursor).

Made changes.

User: on the game screens remove the instruction text in the top left corner

GitHub Copilot: I'll add a short todo, then patch `drawHUD()` in `sketch.js` to remove the controls text and keep only the level indicator.

Starting: *Remove HUD controls text* (1/1)

Using "Apply Patch"

...

...

Completed: *Remove HUD controls text* (1/1)

Done — I removed the controls text from the in-game HUD; it now shows only the level and key status. Reload the page and start the game to verify.

Made changes.

User: make the levels full screen like the starting screen

GitHub Copilot: I'll update `sketch.js` to render levels full-screen by scaling and centering the level drawing instead of resizing the canvas. I'll patch `draw()`, `loadLevel()`, and `windowResized()` accordingly.

Starting: *Make levels full screen* (1/1)

Using "Apply Patch"

...

...

Completed: *Make levels full screen* (1/1)

Done — levels now render full-screen like the menu. I scaled and centered the level and player drawing (keeps tile-grid logic unchanged) and stopped resizing the canvas to match level pixel size.

Quick test:

```
```bash
cd Week4_Example04
python3 -m http.server 8000
```
```

Open <http://localhost:8000/>, click Start, and resize the window to verify the level scales to fill the view. Would you like margins/padding, or a fullscreen toggle button?

Made changes.

User: make the level number and the key collected text smaller so it fits in the gap between the edge of the canvas and the maze

GitHub Copilot: I'll update sketch.js: add globals to track level drawing transform, set them in `draw()`, and update `drawHUD()` to use a smaller font and place the level/key text in the gap next to the maze.

Using "Apply Patch"

```
```
```

```
```
```

Using "Apply Patch"

```
```
```

```
```
```

Done — HUD now uses smaller text and is positioned relative to the drawn maze so the level number and key status fit in the gap. Reload and test; tell me if you want the HUD moved to the other side, use icons, or further shrink the font.

Made changes.

User: move level and key status to the left

GitHub Copilot: I'll move the HUD text to the left edge of the canvas and update sketch.js.

Starting: *Move HUD to left edge* (1/1)

Using "Apply Patch"

^^^

^^^

Completed: *Move HUD to left edge* (1/1)

Done — the level number and key status are now fixed at the left edge (8px margin). Want them aligned vertically with the maze top or moved to the right edge instead?

Made changes.

User: make a third level. randomize the wall locations and key locations. make sure it is solvable

GitHub Copilot: I'm going to add a level generator to sketch.js and push a generated third level into `levelsData` before converting to `Level` instances. This will include a BFS check to ensure start->key and key->goal paths exist.

Using "Apply Patch"

^^^

^^^

Using "Apply Patch"

^^^

^^^

I added a third, randomized level generator to sketch.js that:

- Creates a 7x10 grid with randomized interior walls.
- Places the start at (1,1), a random key, and a random goal.
- Ensures solvability by checking start->key and key->goal with BFS.
- Appends the generated level to `levelsData.levels` if fewer than 3 levels exist.
- Falls back to a safe map if generation fails.

Test by serving and running the game; the Start button will load the game including the generated third level (progress wraps). Want different level size, wall density, or multiple randomized levels?

Made changes.

User: make the third level tougher. add more walls

GitHub Copilot: Starting: *Increase third-level difficulty* (1/1)

Using "Apply Patch"

^^^

^^^

Completed: *Increase third-level difficulty* (1/1)

Done — I increased the wall density for the generated third level by updating the call in sketch.js to:

- `generateSolvableLevel(7, 10, 0.42, 400)`

This makes the level noticeably tougher (more walls) and allows more attempts to produce a solvable layout. Test by starting the game and advancing to level 3 — tell me if you want it even denser or larger.

Made changes.

User: the game ends after the third level. bring the user to a end screen that says "Congrats! You escaped!"

GitHub Copilot: I'll update sketch.js to add a 'win' gameState, show a centered end screen message, and change `nextLevel()` to show the end screen when the player finishes the last level. I'll patch the file now.

Using "Apply Patch"

^^

^^

Done — when the player completes the final level the game now shows a centered end screen reading "Congrats! You escaped!" with a Menu button to return to the title.

Made changes.

User: this should be a completely different screen. make it have the outside of a library building

GitHub Copilot: I'll update sketch.js to render a distinct end screen (exterior library scene) and ensure the win state is displayed instead of the menu background. I'll patch the file now.

Searched for text `gameState === 'win'` (`*/Week4_Example04/*`), no results

Read

[\[\(file:///Users/kenzieleu/Documents/GitHub/mleu_sidequest_w4/Week4_Example04/sketch.js#1-1\), lines 1 to 260\]](#)

Read

[(file:///Users/kenzieleu/Documents/GitHub/mleu_sidequest_w4/Week4_Example04/sketch.js#260-260), lines 260 to 520]

I'll update sketch.js to (1) handle the 'win' gameState in `draw()` and (2) replace `drawWin()` with a distinct exterior-library scene and message.

Using "Apply Patch"

~

~

Using "Apply Patch"

~

~

Finished — the win screen is now a distinct exterior-library scene with sky, building, windows, door, steps, a 'LIBRARY' sign, and a congratulatory banner reading "Congrats! You escaped!".

Quick test: serve the project and finish level 3 to view the new end screen. Want animations, confetti, or a replay button added to this screen?

Made changes.

User: make the library more grand with pillars at the front, brick, and a nice triangle roof. instead of the menu button put a "Home" button. add clouds to the landscape

GitHub Copilot: I'll update the win screen drawing to add clouds, pillars, brick texture, and a triangular roof; and rename the end-screen button to "Home".

Using "Apply Patch"

^^

^^

Done — the end screen is now a grand exterior with clouds, brick facade, pillars, and a triangle roof. I also changed the end-screen button label to "Home" so it returns to the title.

Would you like animated clouds, a little confetti burst when the screen appears, or a replay button to restart level 1 directly?

Made changes.

User: the pillars need to go all the way to the roof. make the brick more subtle. use this as inspo

GitHub Copilot: I'll modify the building-drawing code in sketch.js to extend pillars to the roof and tone down the brick texture (smaller, more transparent bricks). I'll patch the relevant section now.

Using "Apply Patch"

^^

^^

I've updated the win screen:

- Pillars now extend up to (and slightly into) the roof so they visually meet it.
- Brick texture is subtler: smaller, lighter, semi-transparent bricks with softer spacing.

Reload and view the end screen to confirm — I can further tweak pillar width/position, brick color/opacity, or add column fluting if you want.

Made changes.

User: make the library skinnier and make the windows have arched tops. they should also be more subtle. put the congrats you escaped part below the library on the grass

GitHub Copilot: I'll update the win screen to make the building skinnier, add arched windows that are more subtle, and move the congratulations banner to the ground below the library.

Using "Multi-Replace String in Files"

...

...

Done — I've updated the end screen:

- The library building is now skinnier (narrower width).
- Windows have arched tops and are more subtle (darker, semi-transparent).
- "Congrats! You escaped!" banner is now below the library on the ground.

Reload and complete level 3 to see the updated victory screen!

Made changes.

User: move the home button down and make it hoverable. the clouds should be translucent and move along the sky. the windows should be one continuous shape

GitHub Copilot: Using "Multi-Replace String in Files"

...

...

Done — I've updated the end screen:

- Home button is positioned lower and has hover effects (color change and scale)
- Clouds are translucent and slowly drift across the sky
- Windows are now one continuous arched shape (seamless from arch top to rectangular bottom)

Reload and finish level 3 to see the animated victory screen!

Made changes.

User: make the roof taller, move the home button down more. the windows should be arched at the top to create a norman window

GitHub Copilot: Using "Multi-Replace String in Files"

...

...

Done — the end screen now has:

- A taller triangular roof
- Home button positioned lower (closer to bottom edge)
- Norman/Romanesque arched windows with proper semicircular tops

Reload and complete the game to see the updated victory screen!

Made changes.

User: make the clouds more random, the norman windows do not look right, the pillars should end right below the roof

GitHub Copilot: Using "Multi-Replace String in Files"

...

...

Done — I've updated the end screen:

- Clouds are now more random with varied sizes, heights, and spacing
- Norman windows are fixed with proper semicircular arched tops and rectangular bottoms
- Pillars now end right at the facade top, below the roof

Reload and complete the game to see the improvements!

Made changes.

User: spread out the clouds and randomize their shapes. add a couple more as well. make sure the window does not have a stroke but the fills of both parts are the same

GitHub Copilot: Using "Multi-Replace String in Files"

^^^

^^^

Done — the end screen now has:

- 7 clouds (up from 5) with randomized shapes and more spread out across the sky
- Windows with no stroke and consistent fill color for both the arched top and rectangular bottom

Reload and complete the game to see the updated victory screen!

Made changes.

User: once the game is complete and go back to the home page, the game should reset and be able to be played again

GitHub Copilot: Using "Replace String in File"

^^^

^^^

Done — the game now resets when returning to the home page. When you click "Home" from the win screen and then click "Start" again, it will begin from level 1 with a fresh game state.

Made changes.

User: when i play through a second time there is no key

GitHub Copilot: Using "Replace String in File"

^^^

^^^

Done — the game now recreates all levels when you start a new game, so keys will be restored and playable on subsequent playthroughs.

Made changes.