

Process & Decision Documentation

I used ChatGPT-5.2 to support my planning, debugging, and simplification process for this side quest. I began by reviewing the assignment requirements and the provided Playable Maze example in VS Code, then used GenAI to help reason through how to adapt the example while keeping the structure simple and stable. My workflow followed a human-AI collaboration: first, I identified the required outcomes (JSON-based level generation, loop-based tile placement, and automatic level progression); second, I asked targeted questions to troubleshoot loading and logic issues; third, I iterated through testing and refining the code myself; and finally, I simplified the implementation to ensure it clearly met the side quest criteria without unnecessary complexity.

Name: Jolin Li

Role(s): Side Quest 4

Primary responsibility for this work:

Designing and implementing a data-driven maze system using JavaScript and p5.js, including level generation, collision logic, and level progression.

Goal of Work Session:

My goal was to modify the provided Playable Maze example so that:

- levels were generated using JSON data rather than hard-coded grids
- loops dynamically placed tiles (walls)
- multiple levels could be loaded automatically when the player reached the goal

Briefly describe what you were trying to accomplish during this phase of the assignment.

During this phase of the assignment, I focused on transforming the original static maze into a generated system driven by JSON instructions. I aimed to preserve the original player movement and collision logic while simplifying level creation through reusable loops. I also ensured that a second level loaded automatically once the player reached the goal tile in the first level.

GenAI Documentation

If GenAI was used (keep each response as brief as possible):

Date Used
February 9th, 2026

Tool Disclosure
ChatGPT 5.2

Purpose of Use

I used GenAI to help debug file-loading issues, reason through why certain level layouts were behaving incorrectly, and simplify the generation logic so it aligned clearly with the assignment requirements.

Summary of Interaction

GenAI helped identify issues related to script loading order, canvas initialization, and level structure. It also suggested a simplified approach to procedural generation using rows, columns, and wall segments defined in JSON and placed using loops.

Human Decision Point(s)

I decided to remove unnecessary features and focus only on dynamically generated tiles. This decision was made to keep the system readable, stable, and clearly aligned with the side quest requirements.

Integrity & Verification Note

I reviewed all GenAI suggestions critically and tested each change directly in the browser. I verified that level generation, collision, and automatic level switching worked as expected and compared the final implementation against the assignment instructions.

Scope of GenAI Use

The final level layouts, player behavior, and overall structure were implemented by me. GenAI was used only for troubleshooting, simplification, and guidance.

Summary of Process (Human + Tool)

- Reviewed the provided Playable Maze example
- Replaced static grid layouts with JSON-based level descriptions
- Used loops to generate empty grids and place walls dynamically
- Debugged loading and canvas issues with GenAI support

- Simplified level layouts to avoid visual and logical errors
- Confirmed automatic level progression worked correctly
-

Describe one or two key decisions you made:

One key decision I made was to simplify the system by focusing solely on dynamically generated wall tiles instead of adding additional mechanics like words or obstacles. This allowed me to clearly demonstrate procedural generation and automatic level loading without introducing unnecessary complexity that could compromise stability.

Verification & Judgement

- Tested both levels multiple times to confirm walls and paths behaved correctly
- Verified the player could not move through walls
- Confirmed the second level loaded automatically upon reaching the goal
- Checked console output for errors after each change
- Re-reviewed the side quest requirements to ensure all criteria were met

Appendix

User: My maze isn't loading properly when I open it in Live Server.

ChatGPT: This can happen if `resizeCanvas()` is called before `createCanvas()`. Make sure you create a canvas in `setup()` before resizing it based on the level dimensions.

User: Thank you it works now.

User: Level 2 looks wrong - the walls form big stripes and block everything. How can I fix it?

{

"levels": [

```
{  
  "name": "Level 1 (Generated)",  
  "rows": 8,  
  "cols": 12,  
  "start": { "r": 1, "c": 1 },  
  "goal": { "r": 6, "c": 10 },
```

```
  "walls": {  
    "border": true,  
    "lines": [  
      { "r": 2, "c1": 2, "c2": 9 },  
      { "c": 5, "r1": 2, "r2": 6 }  
    ]  
  },
```

```
  "words": [  
    { "r": 1, "c": 3, "text": "JSON" },  
    { "r": 3, "c": 3, "text": "LOOP" },  
    { "r": 5, "c": 2, "text": "WIN" }  
  ],  
},
```

```
{
```

```
"name": "Level 2 (Generated)",  
"rows": 10,  
"cols": 14,  
"start": { "r": 1, "c": 1 },  
"goal": { "r": 8, "c": 12 },  
  
"walls": {  
    "border": true,  
    "lines": [  
        { "r": 3, "c1": 1, "c2": 10 },  
        { "r": 6, "c1": 3, "c2": 12 },  
        { "c": 8, "r1": 2, "r2": 8 }  
    ]  
,  
    "words": [  
        { "r": 1, "c": 4, "text": "LEVEL" },  
        { "r": 2, "c": 4, "text": "TWO" },  
        { "r": 4, "c": 2, "text": "AUTO" }  

```

ChatGPT: This is likely because long wall segments are being generated without gaps. Try shortening horizontal and vertical wall lines and spacing them out so a path remains through the maze.

User:

Do I need words or obstacles for this assignment?

ChatGPT:

The requirement is to dynamically generate tiles, obstacles, or words. Generating tiles (walls) using loops and JSON already satisfies the requirement, so words are optional.