

Process & Decision Documentation

Side Quest 4 Prompt: **Generate a level using arrays or JSON data; use loops to dynamically place tiles, obstacles, or words.** Bonus: Add a second level that loads automatically when the first is finished..

Project/Assignment Decisions

I used Claude to generate the new levels to add onto example 4's code from the tutorials. It generated me 2 new maze layouts, to add onto the level.json file. The game now has 4 levels in total that load automatically when completing. Even if it was a simple task, I had to rely on GenAi. I am not familiar with Javascript, as I only had courses that taught me Dr.Racket and Python. I used this as an opportunity to help me see the similarities between both languages, to help me better understand for future assignments when thinking about the logic behind the coding.

Side Quests and A4 (Individual Work)

I added 2 arrays to the level.json file from example 4 of the tutorials.

Role-Based Process Evidence

The screenshot shows a conversation between a user and Claude. The user asks Claude to generate a level using arrays or JSON data, and to use loops to dynamically place tiles, obstacles, or words. Claude responds with a generated index.html file and a modified level.json file. The index.html file includes a meta charset="utf-8" and a title "Example 4 - Playable Maze (JSON + Classes)". It also contains script tags for preload.js, sketch.js, and player.js. The level.json file is updated to include two new levels: Level 3 and Level 4. The legend at the bottom defines symbols for floor (0), wall (1), start (2), and goal (3).

Dynamically generating levels with arrays

- Level 2 has different layout or new tile types

Example 5: Typing Tutor/Word Game

An educational typing game where:

- JSON contains words or sentences to type
- Loop displays falling words from top
- Type correctly to clear them
- Complete all words to advance to level 2
- Level 2 has faster speed or longer words

Which type of game interests you most? I'm ready to work with whatever template you provide!

Perfect! I can see you have a nice playable maze game with 2 levels already. You want me to add 2 more levels (for a total of 4 levels) that will load automatically when completed.

Here's what I'll do:

Changes needed:

1. Update `levels.json` - Add 2 new maze layouts
 - Level 3: Medium difficulty with more complex pathways
 - Level 4: Harder difficulty with longer path. ↓ more dead ends

Reply...
+ Sonnet 4.5 ↗

Claude is AI and can make mistakes. Please double-check responses.

Pasted content

```
/*Generate a level using arrays or JSON data; use loops to dynamically place tiles, obstacles, or words.** Bonus: Add a second level that loads automatically when the first is completed.
```

```
index.html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta content="width=device-width, initial-scale=1" />
    <title>Example 4 - Playable Maze (JSON + Classes)</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <!-- Load p5 first -->
    <script src="https://cdn.jsdelivr.net/npm/p5@1.9.0/lib/p5.min.js"></script>
    <!-- Load your classes BEFORE sketch.js so they exist when sketch runs -->
    <script src="level.js"></script>
    <script src="player.js"></script>
    <!-- Then load the main sketch (preload/setup/draw) -->
    <script src="sketch.js"></script>
  </body>
</html>
```

```
jsonconfig.json
...
json
{
  "include": [
    "js",
    "libaries/*.js",
    "c:\Users\David\wmcopsci.cs50-p5-1.9.3\p5Types\global.d.ts"
  ]
}

level.js
```
js
/*
Level represents ONE maze grid loaded from levels.json.

Title legend (from your original example):
0 = floor
1 = wall
2 = start
3 = goal
```
A Level represents ONE maze grid loaded from levels.json.
```

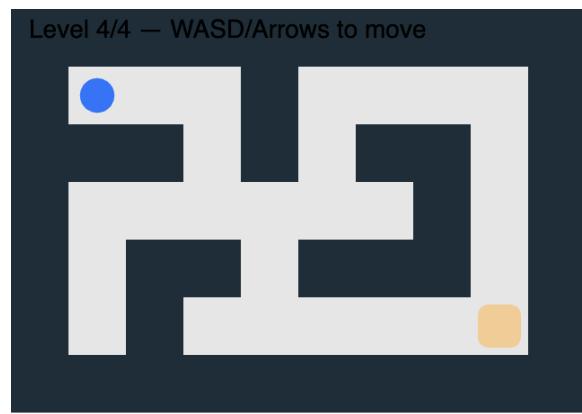
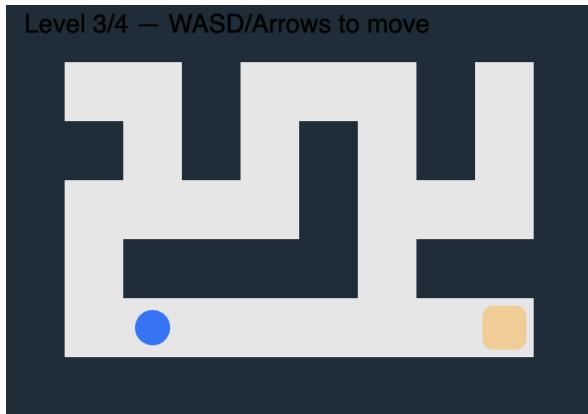
Title legend (from your original example):
0 = floor
1 = wall
2 = start
3 = goal

Screenshot of a dark-themed code editor (VS Code) showing the file `levels.json`. The file contains a JSON object with an array of levels. Each level is represented by a 2D grid of integers (0, 1, or 3). The first few lines of the JSON are:

```

1  {
2    "levels": [
3      [
4        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
5        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
6        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
7        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
8        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
9        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
10       ],
11      ],
12      [
13        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
14        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
15        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
16        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
17        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
18        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
19        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
20        ],
21      ],
22      [
23        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
24        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
25        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
26        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
27        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
28        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
29        ],
30      ],
31      [
32        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
33        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
34        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
35        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
36        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
37        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
38        ],
39      ],
40    }
  
```

The code editor interface includes an Explorer sidebar, a Chat panel on the right, and a status bar at the bottom.



Entry Header

Name: Marine-Chloe Vandame

Primary responsibility for this work: Adding levels (arrays) to the game

Goal of Work Session

- Understanding how all the file and functions work together
- Adding levels to the game

Tools, Resources, or Inputs Used

- Claude (Sonnet 4.5)
- Lecture Notes
- Prior code as base of the game (provided from Dr. Karen Cochrane & David Han)
- Playtesting feedback

GenAI Documentation

Date Used: February 12, 2026

Tool Disclosure: Claude (Sonnet 4.5)

Purpose of Use: Generate 2 new arrays to add to the level.json file

Summary of Interaction: I provided example 4's code and prompted to add the arrays necessary for the additional 2 new levels. I later further inquired about what a loop is and the similarities between Python and Java Script

Human Decision Point(s):

- Prompted what I wanted: to add 2 new levels to the current code (example 4)
- Tried to further understand the coding logic of Java Script with my previous knowledge of Python

Integrity & Verification Note: I play tested to check for any bug or unpredicted outcomes. I also looked over the code provided in class and the generated code, to check if it had the same format and indentations.

Scope of GenAI Use: GenAi did not have the base code for the game, that was code already provided by the professors. Meaning, it did not contribute to the conceptualization of the game or the game/code logic.

Limitations or Misfires: I don't believe there were any misfires due to my simple and direct prompts, that the GenAi correctly generated with the first iterations. No further correction was needed.

Summary of Process (Human + Tool)

- Explained what additions were needed to the code files.
- Inquired what a loop was and how it was used in this assignment, to make sure for assignment accuracy.
- Asked about the similarities of arrays(JavaScript) and lists(Python)

Verification & Judgement

I made sure that the alterations aligned with the Side Quest requirements and play tested to assure that the code was functioning, prior and after the edits were made.

Limitations, Dead Ends, or Open Questions

Due to my lack of proficiency in Java Script, such as not knowing the format requirements or what are the names of the functions of this language, I had to rely on GenAi. In attempts to fast track and not properly learn Javascript, I will have to continue to rely on GenAi to provide me with support when coding.

Appendix

<https://claude.ai/share/d4ff540a-188b-4875-9e75-cec9a05394bb>