

Process & Decision Document

1. Short Description

For this Side Quest, I built a grid-based navigation game using p5.js. The goal was to generate a level using arrays (JSON-style data) and use loops to dynamically render tiles, obstacles, and collectible words.

The player moves one tile at a time and must collect all word tiles before reaching the goal. A second level loads automatically after the first is completed.

The main technical focus was:

- Storing level layouts in structured data
- Using nested loops to render tiles
- Adding simple collision detection
- Implementing automatic level switching

2. One Small Decision I Made

Originally, I considered placing word objects separately from the grid. Instead, I decided to encode special tiles (WORD and GOAL) directly inside the 2D array.

This decision:

- Simplified level readability
- Reduced redundant coordinate tracking
- Made it easier to scale additional levels

I also added a small movement throttle (one move per key press) to prevent accidental overshooting when keys are held down. This improved control precision.

3. GenAI Use Statement

GenAI was used to assist with:

- Structuring the JSON level format
- Debugging movement logic
- Implementing automatic level transitions
- Refining collision logic

All generated code was reviewed, tested, and modified to ensure understanding and functionality.

Appendix: GenAI Transcript

ChatGPT 5.2 ▾

🔗 Share ...

 **sketch.js**
JavaScript

My player movement in a p5.js grid game is too fast when I hold down the arrow keys.
The player keeps sliding across tiles instead of moving one tile at a time.
How can I modify my movement logic so that the player only moves once per key press instead of continuously while the key is held down?



Thought for a few seconds >

If you're checking `keyIsDown()` inside `draw()`, p5 will read the key as "down" every frame (60 times/second by default), so your player "slides." The usual fix is: **move only inside** `keyPressed()` (an event that fires once per press), not inside `draw()`.

In the `sketch.js` you uploaded, I only see the `Level` + grid rendering (no player input code yet).

 `sketch`

So here are the two clean patterns you can drop into your game.



+ Ask anything

