

Process & Decision Documentation

GBDA 302 | Side Quest 4 | Katelyn Nguyen

Project/Assignment Decisions

For this side quest, a significant change wasn't necessarily made but I was able to better understand how to adjust the code primarily because of the comments.

```
4  A Level represents ONE maze grid loaded from levels.json.
5
6  Tile legend (from your original example):
7  0 = floor
8  1 = wall
9  2 = start
10 3 = goal
11
12 Responsibilities:
13 - Store the grid
14 - Find the start tile
15 - Provide collision/meaning queries (isWall, isGoal, inBounds)
16 - Draw the tiles (including a goal highlight)
17 */
```

I began by looking through all the provided files from the Week 4 examples and found the most help from the *levels.js* file. The coding comments explained how the mazes were built and the way each level had a different maze layout. This was based on the numbering. The "Tile legend" made it clear what purpose each number served and how it built the arrays. But, everything only clicked after I viewed the *levels.json* and saw the grid levels/2D array of numbers.

I found it difficult to look at the arrays straight on because all I could see were numbers. Similar to colour by number pages, I colour coded each number and this revealed the maze path with the start and exit points. This made it much easier to visualize the maze and see how it's made up by the numbers.



For the third level, I took the existing coding from level 2 and simply tinkered with the numbers; changing 0s to 1s and vice versa to create new walls or path openings and assigning a different end point, until I was content with the new maze formation.

For the last level I wanted to see if I could make the maze taller and I did so by adding 3 additional rows to the nested array. As a result, this extended the maze in length and I had more available space to add in new walls and paths.

At one point, I mistyped and accidentally added in another 3 instead of a one or zero. Meaning my maze had multiple end goals rather than a single goal. With this, I thought the level wouldn't function properly and the player would remain stuck on level 4, but surprisingly this wasn't the case. After testing, I realized having more than one goal didn't have an effect on the function of the mazes. Instead it just provided multiple ways to complete the final level which I thought was a simple, yet cool find.

Entry Header

Name: Katelyn Nguyen

Role(s): Code modifier / Level designer

Primary responsibility for this work: Modify and create additional levels to maze game.

Goal of Work Session

- Understand the maze and levels structure
- Adjust and design new maze layouts through the numeric arrays

Tools, Resources, or Inputs Used

- Example 4 from Week 4 code examples
- Week 4 - part 2 lecture slide

GenAI Documentation

No GenAI used for this task

Summary of Process (Human + Tool)

- Looked through files and read comments
- Experimented with numbers
- Designed and tested new maze layouts
- Discovered the possibility of new end goals

Decision Points & Trade-offs

Describe one or two key decisions you made:

- Created new maze layouts by modifying existing levels code
- Added multiple end goals which didn't break the game logic

Verification & Judgement

- Playtesting the levels
- Confirming the player could move to next level after reaching goal
 - Ensuring final level looped back to first starting level
- Reviewing side quest prompt/instructions

Limitations, Dead Ends, or Open Questions

- Although I made the maze taller I didn't explore making it wider
- I thought adding multiple start points would change where the player began each time the maze loaded, however, this didn't have any impact