Process & Decision Documentation

Project/Assignment Decisions

I learned how to use GenAI to brainstorm ideas and help add on dynamic loops for obstacles, tiles, and/or words visually to a starter code for the JSON 2D array game in Side Quest 4.

GenAI Documentation:

Date Used: Feb 10, 2026

Tool Disclosure: ChatGPT 5.2

Purpose of Use: I used ChatGPT 5.2 to brainstorm dynamic obstacles to add as a loop for the 2D array game, explore how loops could communicate a challenge, and generate starter ideas for adjusting 2D visuals.

Summary of Interaction: The tool suggested different types of obstacles, tiles, and words and that could be modified to express them. It also provided simple starter examples that could be adjusted.

Human Decision Point(s): I made the final decisions for what kinds of dynamic loops to include in the visual design choices, how the objects should be placed, and which variables to change that would be appropriate to the assignment and what we have learned in class thus far.

Integrity & Verification Note: I checked GenAI for accuracy, bias, and made sure it was appropriate to fit for this assignment. I made sure the output was aligning to the class materials and verified it by rereading the assignment instructions and comparing them to GenAI suggestions. Any content that was beyond what we learned in class was removed and replaced.

Scope of GenAI Use: The scope of using GenAI was for the ideation process, organization, and early scaffolding.

Limitations or Misfires: The tool is good for storing information and re visiting it. However, it over-generalized the emotional expression, suggested certain features that were beyond the course material, and misunderstood specific constraints.

Summary of Process (Human + Tool)

- Initial ideation between drafts or versions
- Conversing with ChatPT 5.2
- Testing and revision

- Debugging and restricting code by myself
- Moments of rework: Using ChatGPT 5.2 too much to ideate the starter code led to extra revision of my work.

Decision Points & Trade-offs

- Options Considered: I considered putting different words other than “GO” and “EXIT” and different tiles.
- What Changed: The early concepts focused more on clarity than visuals. Through iteration, I prioritized conveying a clear message through variables to show how the dynamic loops can be used.

Verification & Judgement

- Rereading assignment criteria
- Testing the iterations manually
- Comparing code structure to example code
- Re-reading assignment criteria
- Ethical, representational, accessible, or global considerations
    - Treated as a support assistant for debugging and ideating, rather than taking full ownership.

Limitations, Dead Ends, or Open Questions

- Always be specific when sending in the prompts so it doesn’t misunderstand

Appendix

My prompt:

I need ideas for generating a level using arrays or JSON data. It has to use loops to dynamically place tiles, obstacles, or words, but stay within what we learned in class.

GenAI:

Here are some brainstormed level concepts that could be built using arrays or JSON data:

- Maze grid using 0 and 1 values for walls and floors
- Word maze where tiles contain short labels like “GO” or “EXIT”
- Obstacle tiles placed from an array of coordinates
- Pathfinding layout where words guide direction
- Simple dungeon layout with repeated tile patterns
- Puzzle grid where symbols mark key locations
- Start and end markers stored in JSON data

- Decorative tiles layered over a base grid
- Multiple levels stored as separate arrays
- Tile legend system (floor, wall, obstacle)

My prompt:

Can these ideas still work if I keep the Level class from the lecture and don’t add new mechanics?

GenAI:

Yes. You can keep the Level class exactly the same and extend the level by adding external data arrays.

For example:
- Keep the grid rendering inside the Level class
- Store obstacles in a separate array or JSON object
- Store words in another array
- Use loops in draw() to place them dynamically
- This maintains separation of data and rendering logic

My prompt:

How could I structure the level data in JSON so loops can place tiles and objects?

GenAI

Use a JSON level object that contains both the tile grid and extra level elements.

```
"grid": [
  [1,1,1,1],
  [1,2,0,1],
  [1,0,3,1],
  [1,1,1,1]
]

"obstacles": [
  { "r": 1, "c": 2 },
  { "r": 3, "c": 4 }
],

"words": [
  { "r": 1, "c": 2, "text": "GO" },
  { "r": 5, "c": 8, "text": "EXIT" }
]
```

Each object uses:

- `r` → row index
- `c` → column index
- `text` → label (for words only)