Process & Decision Documentation

In Side Quest #4, I used ChatGPT-5.2 to help me build a game where a blob moves through obstacles to reach the next level. The original code had only a one-level map, in which the blob navigates a simple obstacle course of platforms. The user can control the blob by moving it left or right or by making it jump. Once the level was completed, nothing changed, and it just stayed the same.

I put this code into GenAI and asked it to add loops and arrays to create different "levels." I began by sending the code and asking it to loop through the maps. At first, I tried to change the code across all the pages in Visual Studio Code, but there were so many errors that the game didn't work. After much trial and error, I decided to restart. This experience demonstrated that AI will not always provide the exact code you want. Instead, it served as a tool to help me figure out how to make the game work and which parts needed to be changed. I learned that I mainly had to update the level.json file to make the code function properly.

After this, I was able to click "N," and the game would switch to a more difficult level, where the colour changed, and the number of platforms increased. I then wanted to add a third level, so I asked ChatGPT to create one that was even more challenging. It initially generated a map similar to Stage 1, so it wasn't much harder, and I decided not to keep it. Instead, I asked it to add more platforms to increase difficulty. ChatGPT also created unique names for each map, such as "Climb Higher" for the second stage and "Reach the Peak" for the third. I found this very creative and decided to keep it. I then asked ChatGPT to make the game automatically progress to the next level once the blob reaches the end of the current map, instead of requiring the user to click "N" each time.

Overall, I found that ChatGPT can help code a game, but errors and mistakes are common. Since it doesn't have direct access to the files, it isn't aware of all the details within the code. Using GenAI effectively requires feedback, trial and error, and guidance to get the code working properly.

Project/Assignment Decisions

Throughout this side quest, I learned how to iterate on and refine a simple starter code through trial and error with AI. I developed a deeper understanding of how to use loops and arrays in code to create repetition. I also learned how to analyze AI-generated code to improve gameplay and add more levels. Additionally, I gained insight into using ChatGPT as a support tool, guiding it with specific prompts and corrections to achieve

the desired result. I discovered that sometimes restarting from the beginning can produce better output when the code does not work.

GenAI Documentation

Date Used: February 9th, 2026

Tool Disclosure: ChatGPT-5.2

Purpose of Use: I used ChatGPT as a learning and support tool to change maps, use loops/arrays, and add levels.

Human Decision Point(s): I gave ChatGPT prompts to modify the code in order to keep code that worked and got rid of code that did not work.

Summary of Interaction: ChatGPT was used as a collaborative support tool to expand an existing game prototype. I provided the original code and prompted the AI to build upon it by adding multiple levels, more difficult platforms, and automatic loops.

Integrity & Verification Note: I opened a live demo in order to see if the code/game works. I constantly checked it whenever I updated the code and changed what was necessary.

Limitations, Dead-Ends, and Misfires: ChatGPT does not really know which codes to edit in order for the code to work.

Decision Points & Tradeoffs

- **Level complexity**: More levels make the game engaging, but also require more debugging and AI guidance. Adding too many levels too quickly can create errors and make testing harder.
- **Automating Level Progression**: Automatic progression improves gameplay flow but requires more complex coding and debugging. Manual progression is simpler but less smooth for the player.

Appendix

Link to chat: [ChatGPT Appendix]