In my development process I tried to follow a systematic approach to ensure efficient and organized work while building a project. Initially, I analyzed the project requirements and objectives, trying to understanding the core problem that needed to be solved, and what will be the necessaries mechanics to program, and also determining possible assets that I'm going to need for visuals, audio, including the ones provided.

After understanding the project's scope, I proceeded to structure into smaller tasks what I need to do, and also what I could do to bring more fun and juice to the game.

Right after creating my Unity project, I initiated a Git repository and used a .gitignore file to exclude unnecessary files from version control. With that I can keep control of my work, and split in different commits new additions, having a more maintainable project.

Before diving into the full implementation, I analyzed my references, mainly Stardew Valley, and I created quick prototypes to visualize the desired features and functionalities. This helped in validating concepts and making early adjustments as needed.

Started the implementation I focusing on integrating the core systems first, like the player interaction with the game using the mouse, and character movement, and creating the logic behind the NPCs, the store, and inventory, and then gradually adding the visual elements. Additionally, I revisited the usage of Tilemaps, which I hadn't utilized in a long time.

To enhance maintainability and flexibility, I aimed to write expandable code. Also, I tried to use a pattern like SOLID, unfortunately by the time I had I decided to step back in it, because it was being very time consume to guarantee that the code was following the pattern.

To speed up the process I brought some scripts I already have, like my Singleton class creator, that converts child classes in Singleton, and my Enhanced Button system for adding audio to button clicks.

In some situations, to speed up repetitive tasks, I created automation scripts, like automatically generating Scriptable Objects for the game items, already setting information like name, sprite, type, and value.

While I couldn't extensively profile for performance issues due to time constraints, I did my best to optimize where I could, mainly reducing instantiation and destruction of UI elements. But also, I acknowledged areas for potential improvement, like the character movement and environment collisions, that are very basic and I know I need to learn more about collisions in Tilemaps. Other point is the portrait in the inventory with render texture and a second camera, the inventory system that could be further enhanced for reusability e best usage by the player, and considered replacing strings with String Builders for better memory handling.

In conclusion, I am satisfied with my performance within the given timeframe. Working on project in Unity, a familiar environment, and working with 2D, that is not so usual for me, was comforting and fun, and I enjoyed the creation process. Despite what could be better, I believe I delivered a complete demo with the desired features.