Project Proposal

Paula Lozano Gonzalo

January 10th, 2025

Contents

1 Problem Description

Stardew Valley is a popular farming simulation game that challenges players to efficiently manage their in-game time, money, and resources. Many players, both new and experienced, struggle to plan their activities effectively or need quick access to information such as newly discovered items or preferred gifts for characters. These challenges (such as determining the best crops to plant based on the season, day, and growth time) along with managing the money, can lead to inefficiencies and frustration, especially for players who are less familiar with the game's mechanics.

2 Proposed Solution

This project proposes the development of a Stardew Valley Assistant, a software tool designed to optimize gameplay strategies. The assistant will provide personalized recommendations in the following areas:

1. **Crop Selection**: Suggest the best crops to plant based on the current in-game season, day, available funds, and expected profit margins.
2. **Budget Management**: Advise players on spending priorities to ensure they can afford necessary seeds and upgrades.
3. **Item Usage**: Provide information on the purpose of various items, such as crafting, gifting, or selling.
4. **Relationship Building**: Offer tips on improving relationships with characters, including their favorite gifts and schedules.
5. **Processing Advice**: Recommend the most profitable ways to process items, such as turning crops into artisan goods or refining raw materials.
6. **Resource Gathering**: Suggest efficient methods for collecting materials like wood, stone, ores, and forageables based on the player's progress.

3 Technical Overview

The assistant will be implemented as a mod using Stardew Valley's API for cross-device accessibility. The following technologies will be used:

1. **Modding within the Game**: Implement the assistant directly as a mod for Stardew Valley using the SMAPI framework. The mod will be written in C#, a language and API in which I have no experience.
2. **Data Handling**: Store game data using JSON or SQLite databases. I have some experience with SQLite but no prior experience with JSON databases.
3. **Natural Language Processing**: Integrate an LLM API to generate responses to user queries. I have some experience working with LLM APIs.

4 Milestone List

Jan 30: Finalize project requirements, gather game data, research and select libraries, and set up the project environment. (The next milestones might change order or change a little bit depending on the finalized requirements for the project) Identify which Stardew Valley data is necessary for the mod (e.g., crop schedules, seasons, player stats). Research modding libraries/tools (e.g., SMAPI for Stardew Valley, a C# database library like SQLite, and LLM APIs). Install SMAPI and set up a Stardew Valley modding environment. Choose and configure a database solution for player and game data. Create a repository (GitHub).
Feb 13: Mod setup for user input and game event storage. Create a basic mod that hooks into the game using SMAPI. Add functionality to listen to user inputs. Test message output from the mod back to the game. Research how to listen for in-game events.
Feb 27: Database integration, AI tree search and basic LLM setup. Integrate the database into your mod. Experiment with connecting the mod to an LLM API. Search algorithm for simple queries. (I need to think more about the search algorithm).
Mar 20: Advanced functionality and question-handling. Implement tree search or similar algorithms for complex questions (e.g., "What crops maximize profit by the end of the season?"). Expand input/output options for better communication with the player.
Apr 3: Testing and final optimization (there will be testing all along but this is where other users test the mod). Improve user feedback and error handling.

5 Validation Plan

Functional Testing: Ensure that core features like crop selection and budget management work as expected, and verify that the assistant can correctly respond to user queries.
Interface Testing: Verify that messages are sent when user wants and that there is an answer.
Usability Testing: Have others test the assistant to ensure ease of use and effectiveness in providing valuable support.