Project Report

Systems Programming – Spring 2025

Akif Yıldırım, 2022400108
Oğuz Semih Arık, 2022400036

April 15, 2025

1 Introduction

The Witcher Tracker project simulates a text-based assistant for Geralt of Rivia. The system manages ingredients, potions, trophies, and monster knowledge using linked list structures. The main objective is to provide a dynamic, modular command-based program capable of parsing inputs and updating the internal state accordingly.

2 Problem Description

Geralt can perform several actions:

- Loot ingredients.
- Learn new potion formulas or monster weaknesses.
- Brew potions using ingredients.
- Trade trophies for ingredients.
- Defeat monsters if prepared.
- Query total ingredients, potions, or trophies.

Constraints:

- All lists must be implemented using custom linked lists.

3 Methodology

The solution is based on string parsing techniques in C using `strtok`, `sscanf`, and `strstr`. The project uses multiple handler functions for different commands. Dynamic memory management is handled via `malloc` and `free`.

4 Implementation

4.1 Code Structure

- `main.c` — Contains all data structures, input parsing, and command handlers.
- `Makefile` — Compiles the program.

4.2 Key Functions

- `handleLoot()` — Adds looted ingredients.
- `handleLearn()` — Learns formulas or monster knowledge.
- `handleTrade()` — Trades trophies for ingredients.
- `handleBrew()` — Brews potions using ingredients.
- `handleQuestions()` — Handles all queries.

4.3 Sample Code

```c
void add(Item** head, const char* name, int quantity) {
    Item* item = find(*head, name);
    if (item) item->quantity += quantity;
    else {
        Item* newItem = malloc(sizeof(Item));
        strcpy(newItem->name, name);
        newItem->quantity = quantity;
        newItem->next = *head;
        *head = newItem;
    }

}
```

4.4 Challenges and Solutions

During the implementation of the Witcher Tracker project, we encountered several challenges. Below is a summary of the main difficulties and how we solved them.

- **Dynamic Linked List Management:** I had difficulties at first managing linked lists dynamically, especially with adding, searching, and removing nodes properly.
  Solution: With the help of ChatGPT, I learned how to properly use linked lists in C. I understood the logic of pointers and implemented generic functions like `find()`, `add()`, and `removeItem()` for safe data management.

- **String Parsing and Input Handling:** It was challenging to parse complex input commands like `Geralt trades 1 Harpy trophy for 8 Vitriol, 3 Rebis` correctly.
  Solution: I used a combination of `strstr`, `strtok`, and `sscanf` functions to split the string dynamically and extract necessary values without hardcoding.

- **Different Sorting Requirements:** Only "What is in" query required sorting items from most to least quantity, but other outputs had to be alphabetical.
  Solution: I created a separate function `printSortedByQuantity()` and used it only in relevant parts to meet the project requirements.

5 Results

Sample Input:

```
Geralt loots 5 Rebis, 4 Vitriol
Geralt learns Black Blood potion consists of 3 Vitriol, 2 Rebis, 1 Quebrith
Geralt brews Black Blood
Total ingredient?
```

Sample Output:

```
Alchemy ingredients obtained
New alchemy formula obtained: Black Blood
Alchemy item created: Black Blood
3 Rebis, 1 Vitriol
```

6 Discussion

Strengths:

- Clean modular code.
- Fully dynamic memory usage.
- Correct parsing of all commands.

Limitations:

- Some commands require precise spacing to be parsed correctly.

7 Conclusion

The Witcher Tracker project successfully simulates a text-based inventory and combat assistant for Geralt. It demonstrates C programming skills such as dynamic memory allocation, string parsing, and modular code structure.

8 References

- Ritchie, Dennis M., and Kernighan, Brian W. "The C Programming Language."
- UNIX Programming Concepts — Lecture Notes.

AI Assistants

During the development of this project, AI tools (ChatGPT) were used for:

- Debugging syntax errors.
- Getting suggestions for string parsing improvements.
- Learning linkedlist fundamentals for C.

All code is written and implemented by the students. AI usage was limited to understanding concepts and formatting.