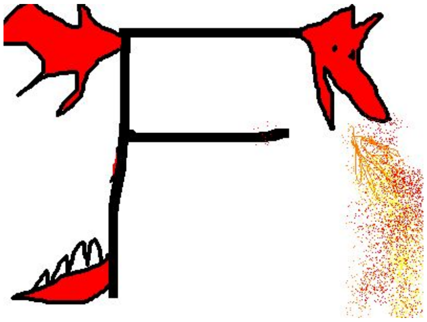
Dungeon Game

Fafnir



Mark Wagner

Jordan Sanders

Karamullah Agha

Table of Contents

Introduction

This document describes the design and planning phase of our project. Our project is a game that takes place in a dungeon. Your goal is to explore the dungeon and find treasure, but it won't be easy. You will have to explore a variety of different rooms, and sometimes you will have to solve different puzzles to access a room. Some of these rooms might contain treasure, or items you can use. The game will have a straight forward command line interface, where you would enter one of the options shown on screen to perform an action. Everytime you enter a room, you will be given information about the room, such number of doors in the room, what non player characters are present if any, whether there is treasure. There will be may be some rooms which can be locked and may require a key to enter, or a puzzle. You will be able to interact with some non-player characters as well, some of these characters might give you a task like solving a riddle and might reward you for it. While playing you will have to keep an eye on your health and hunger, if your health goes to 0 or hunger becomes 100, you will die. Food and health potions can be found randomly throughout the game. The game will end once you find all the treasure.

This document has the following sections:

- **Project Management:** Going over our team organization and risk management.
- **Development Process:** Going over Code review process, Communication tools, dealing with change in management, and Software Design.

Project Management

a. Team Organization

Mark Wagner: Team leader

Karamullah Agha: Design lead

Jordan Sanders: Quality assurance lead

b. Risk Management

I. Requirements/Design/Estimation.

1. **The team planned a project that is too large.**
    - We will not implement all of the originally planned features to the project.
2. **The team underestimated how long parts of the project would take.**
    - Reduce some of the features or simplify them if they are too complex and we are running out of time.
3. **Major changes to design are needed during implementation**
    - Make the necessary changes in order to implement the features.

II. People

1. **Addition or loss of team member**

- If there is a loss of team member we will report to the instructor immediately and continue with our project taking on additional roles.
- If we gain a team member, we will update them on the current status of the project. Go over some of the code with them and the design. Do whatever we can to get them to start contributing to the project.

2. **Unproductive Team Members(s)**

If a group member decides to not contribute to the project, we will first ask them for the reason. If they have a valid reason, such as being stuck on their part, we will try to help them out. If they are just being unproductive because they do not want to work, or are aiming for a low grade, we will report to the instructor and take on their job if they refuse to change. Do whatever we can to make sure the project does not slow down.

3. **Team member(s) lacking expected technical background**

Every member of our team is inexperienced with c++. We must all put in an effort to learn as we go.

4. **Major life events**

If a team member has a major life event, we will inform our instructor and do our best to complete the project without said member, provided that the major event interferes with development.

III. Learning and Tools

1. **Inexperience with new tools**

   It is expected that we will run into issues resulting from the team's lack of experience with c++ and Code::Blocks. As mentioned before, we will do our best to learn as we develop the project. We will remain in communication with each other to discuss solutions to the problems we encounter.

2. **Learning curve for tools steeper than expected**

   In this case we call a group meeting and work together to pick up on the tools, perhaps speak to the instructor for help.

3. **Tools don't work together in an integrated way**

   Depending on how far we are into the project. If we feel the tools just won't work, we might have to abandon them and find different ones. Continuing to try to make them work might just lead to us wasting time.

Development Process

a. Code Review Process

The Quality Assurance and Design Leads will review the code to ensure quality and performance. The Team leader will also review the code.

b. Communication Tools

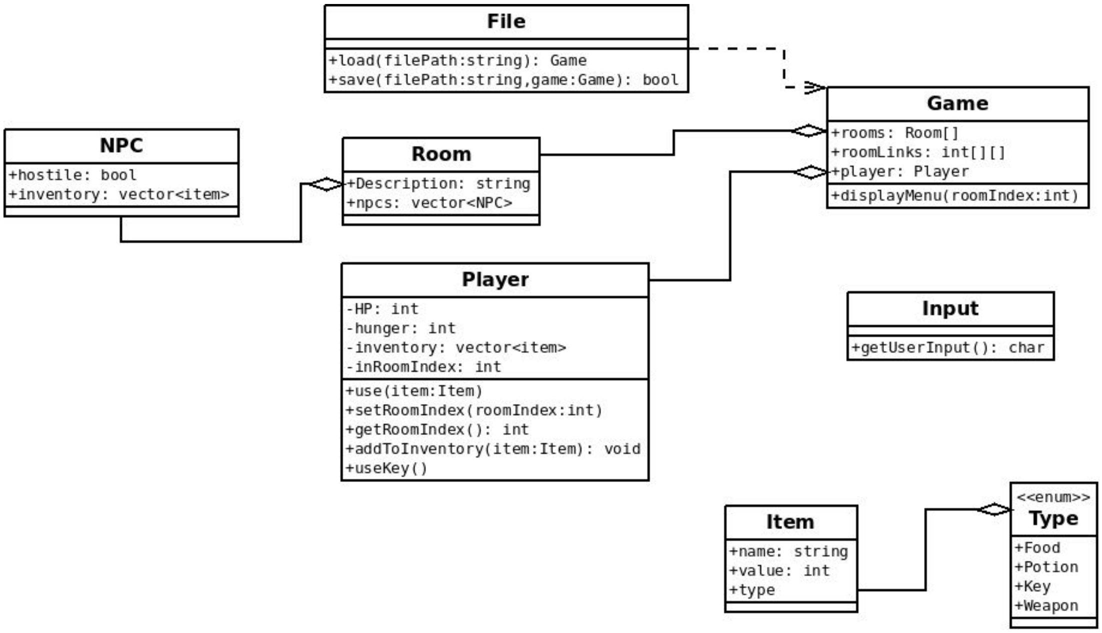
The team has a Slack group chat for communication. Email may also be utilized.

c. Change Management

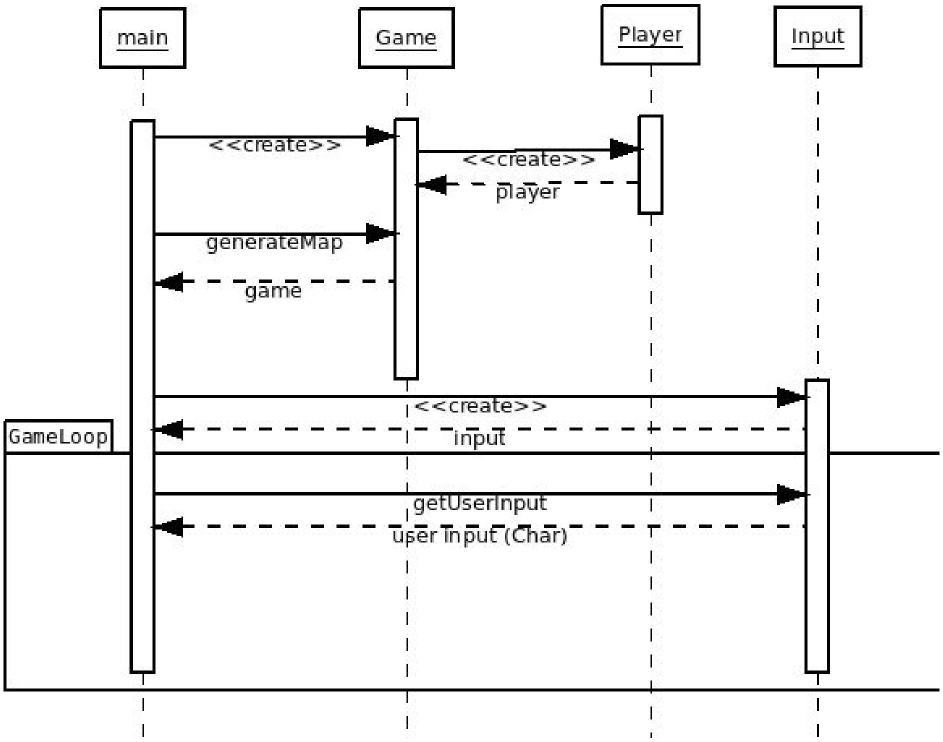
Major changes will require at least two group member's approval. Since we're using version control tools, rollbacks to previous versions will also be possible. (Again needing ⅔ consensus.)

d. Software Design

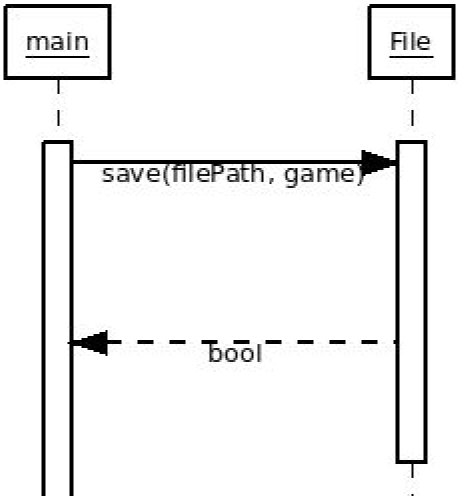
I. UML Diagram



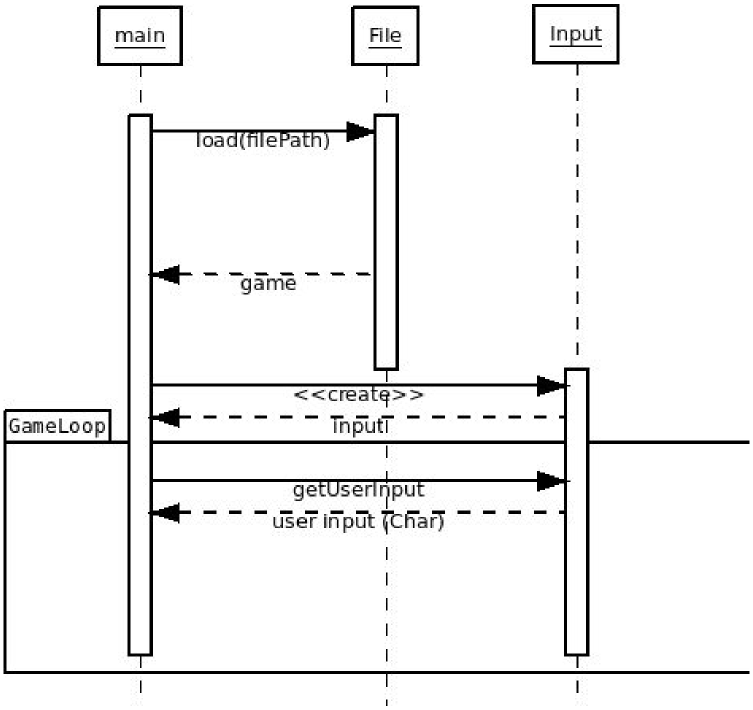
II. Game Starting



III. Saving the game



IV. Loading the game


main
File
Input
load(filePath)
game
<<create>>
GameLoop
Input
getUserInput
user input (Char)