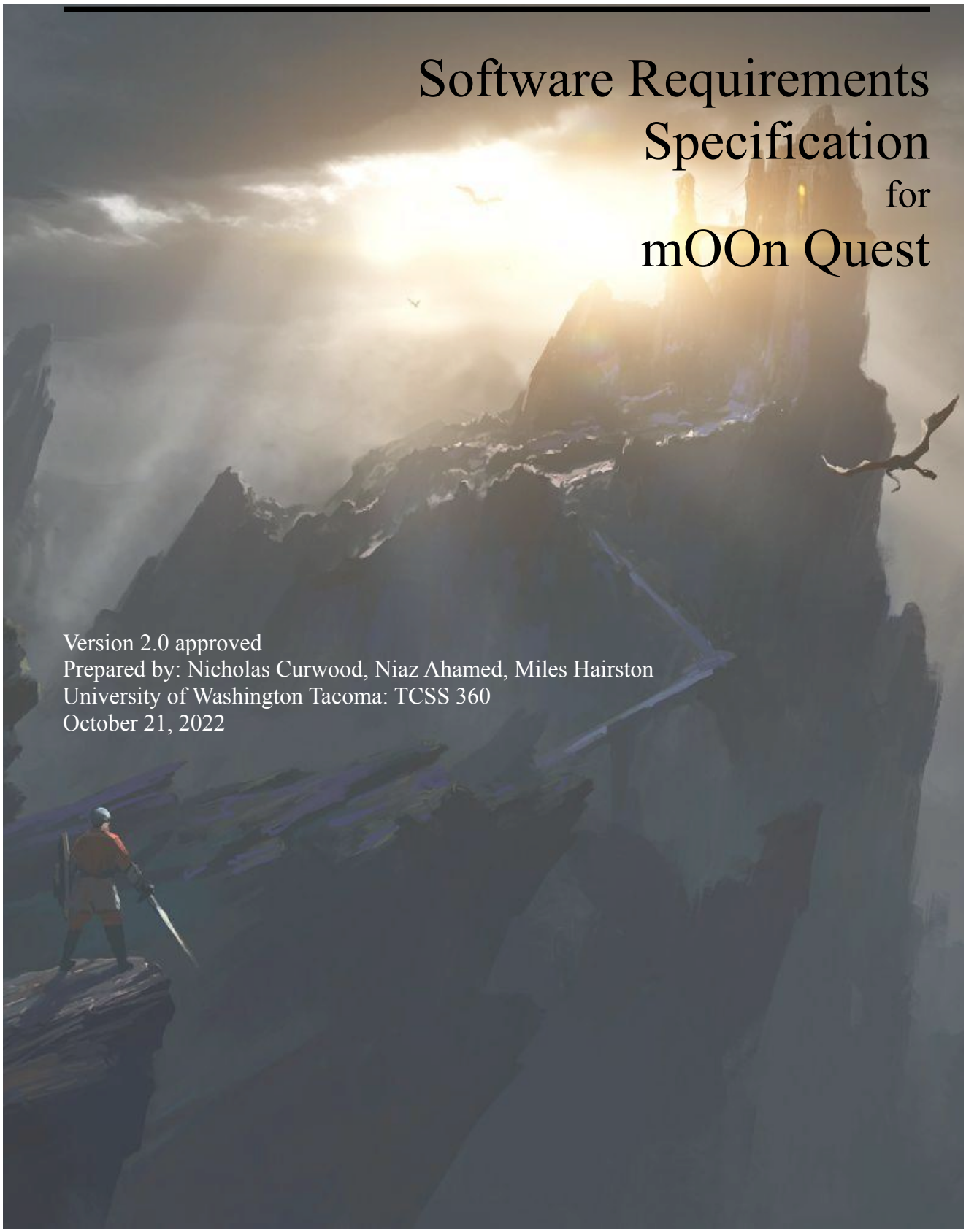

Software Requirements Specification
for
mOOn Quest

Version 2.0 approved
Prepared by: Nicholas Curwood, Niaz Ahamed, Miles Hairston
University of Washington Tacoma: TCSS 360
October 21, 2022

Table of Contents

1. Introduction **2**
1.1. Purpose…… 2
1.2. Document Conventions…… 2
1.3. Intended Audience and Reading Suggestions…….. 2
1.4. Project Scope…… 2
1.5. References…… 2-4

2. Overall Description **4**
2.1. Product Perspective…….. 4
2.2. Product Features……... 5
2.3. User Classes and Characteristics…….. 5
2.4. Operating Environment…… 5
2.5. Design and Implementation Constraints…….. 5
2.6. User Documentation……6
2.7. Assumptions and Dependencies……... 6

3. System Features **7**
3.1. The Maze…….. 7
3.2. Title Screen……... 7
3.3.Characters……...7-8
3.3.1. Heroes……..8-9
3.3.2. Monsters…… 9-10
3.3.3. Guardians……10-11
3.4. Save/Load State…….. 11
3.5. Difficulty level……12
3.6. Monsters spawns……12
3.7. Potions…… 12
3.8 Combat……12
3.9 Overboard Damage……12
3.10 Winning …… 12

4. External Interface Requirements **12**
4.1. User Interfaces……12
4.2. Software Interfaces……13

5. Other Nonfunctional Requirements **13**
5.1. Performance Requirements……13
5.2. Security Requirements……13
5.3. Software Quality Attributes……13

6. Other Requirements **14**
Appendix A: Glossary……... 14

Appendix B: Issues List……15

1. Introduction

1.1. Purpose

This software requirements and specifications document describes both the functional and non-functional requirements of version 1.0 of mOOn Quest. The purpose of this document is to provide an abstract overview of the implementations necessary to successfully and efficiently construct the Mythical Adventure game.

1.2. Document Conventions

Unless explicitly stated, all requirements are deemed high priority to ensure the success of the project. This document is a living document and will be used to plan both functional and non-functional requirements as they arise during project development.

1.3. Intended Audience and Reading Suggestions

This document will mostly serve the project developers as a checklist of functionalities that must be implemented in the end project. However, this document also serves a second purpose of providing the assignment manager (Tom Capaul) with an understanding of the direction that will be taken by the project developers. This document will both contain a list of functionalities as well as any assumptions made about the end user of the product.

1.4. Project Scope

This project will be an introduction to team based software development. The goal of this project is split into two parts of inequivalent importance. The first goal measures the success of the developers by completing the project and all of its functionalities. The second and most important goal is to properly organize developer contributions such that the end result is both legible and well-parsed, reflecting the developers ability to work together.

1.5. References

1. Oracle. *Code Conventions for the Java Programming Language*, http://web.archive.org/web/20140222045352/http:/www.oracle.com/technetwork/java/codeconv-138413.html
2. Tom, Capaul. *Course Project,* *https://canvas.uw.edu/courses/1589305/assignments/7702568?module_item_id=16759752*
3. Wikipedia. *Maze Generation Algorithm, Randomized Depth-First Search,* https://en.wikipedia.org/wiki/Maze_generation_algorithm#Recursive_implementation

4. PNG or JPG to SVG

https://www.pngtosvg.com/

5. Maxim Shoustin. ImagePanel
https://stackoverflow.com/questions/19125707/simplest-way-to-set-image-as-jpanel-background

Images:
Picture of Tom Cruise's Face:
https://www.ebay.com/itm/234591852232?chn=ps&_trkparms=ispr%3D1&amdata=enc%3A15hVpc9tLTnyppY_4jOT2mg25&norover=1&mkevt=1&mkrid=711-117182-37290-0&mkcid=2&mkscid=101&itemid=234591852232&targetid=1584571731043&device=c&mktype=&googleloc=1027744&poi=&campaignid=15275224983&mkgroupid=131097072938&rlsatarget=pla-1584571731043&abcId=9300697&merchantid=223123042&gclid=Cj0KCQiAkMGcBhCSARIsAIW6d0C9C4mvakwamM0aMf4k0xD7SsKqvLKmswqovlBizDwQH3yXVZJzFPEaAs4SEALw_wcB
https://www.istockphoto.com/photo/3d-illustration-medieval-tavern-gm1322864653-408704821

Picture of Knight:
https://www.pinterest.com/pin/28921622590561202/

Picture of Assassin:
https://www.deviantart.com/diablo7707/art/Baek-ji-middle-agesdark-assassin-render-814364890

Picture of DireWolf:
https://creaturequest.fandom.com/wiki/Dire_Wolf

Picture of Ogre:
https://www.pngwing.com/en/free-png-trmtd

Picture of Goblin:
https://www.pngwing.com/en/free-png-icpnv

Picture of Dragon:
https://www.coolstuffinc.com/p/125566

Picture of Hydra:
https://www.seekpng.com/ipng/u2w7y3y3e6i1w7a9_hydra-hydra-greek-mythology-png/

Picture of Cerberus:
https://monster.fandom.com/wiki/Cerberus

Thumbnail taken for swamp:
https://www.youtube.com/watch?v=Xf0P5dpC2-0

Volcano:
https://www.reuters.com/business/environment/alert-level-raised-hawaiis-mauna-loa-volcano-after-eruption-usgs-2022-11-28/

Underworld:
https://www.nytimes.com/2022/12/03/science/mauna-loa-volcano-eruption-hawaii.html

Picture of Graveyard for Death Panel
http://creaturespotlight.blogspot.com/2016/06/10-graveyard-encounters.html

Tavern for Win Panel:
https://www.istockphoto.com/photo/3d-illustration-medieval-tavern-gm1322864653-408704821

Forest:
https://www.youtube.com/playlist?list=PLga5e6HS7U3aosjXCvQc6KYoFgfYoGu1Y

Save Icon:

https://www.iconspng.com/image/25574/mono-save

Help Icon:
https://www.presentermedia.com/powerpoint-clipart/question-mark-button-symbol-pid-6177

BlackHole:
https://unsplash.com/s/photos/black-hole

Difficulty:
https://www.alamy.com/stock-photo-mossy-wall-bricks-24305874.html

Entrance:
https://www.istockphoto.com/photo/old-stone-entrance-wall-in-green-garden-gm482804311-37293348

Brown Fabric Background for Inventory and Save and Help
https://clickprops.co.uk/product/rembrandt-brown-profabric-photography-backdrop-for-studios/?v=7516fd43adaa

Map Background For Map Panel:
https://wallpaperset.com/vintage-map-wallpaper

Potions:
https://forgottenrealms.fandom.com/wiki/Potion_of_vitality

2. Overall Description

2.1. Product Perspective

mOOn Quest is a standalone game that will feature a turn-based battle mechanic and a randomly generated maze map for every iteration of the game. The game will also allow players to save their game to a text file and draw from this save in later instances of the game. The game will feature monsters, guardians, and heroes with certain special moves and attacks that pertain to the class that they occupy. The character attributes and their effectiveness will be defined by querying against a database of all the character information. The entirety of the game will be played through a GUI window that will offer ease of gameplay for the user. In figure 1, a UML class diagram displays the different interconnected classes that compose these major features.

2.2. Product Features

2.2.1: Overview

This game will feature several heroes and monsters, unique maps every run, battles between the hero and monsters, inventory for item usage, a save and load game mechanic, and room-specific items such as health potions and bear traps.

2.2.2: Bosses

To progress through the game, the hero will need to defeat four special boss monsters called guardians that each have their own unique abilities that will present a challenge to the hero. To aid in the reaffirmation of these bosses' importance, they have been given costume backgrounds that pertain to their characteristics. The Cerberus is fought in a rift between the woodlands and the underworld, the Hydra is fought in the swamplands, and the Red Dragon is fought at the peak of a volcano.

2.2.3 The Maze

The Maze is unique every time as well as the contents of each room. Only one thing is certain between game runs, there is an entrance and that entrance can not be used to exit until all four bosses have been vanquished. The Maze has three difficulties, each a little more difficult than the last. More difficult iterations of the maze are larger and contain more difficult base monsters.

2.2.4 Combat

Combat is turn based with each round of turns consisting of at least one hero turn and one monster turn. A hero can be given multiple turns if the speed of the hero is over two, three, or even four times greater than the enemy's. The hero is given the options of a basic attack, a special attack, or the usage of an item. The usage of items does not consume a turn. Finally, if the hero deals damage to the enemy that brings their health below zero, the hero has a chance to heal up to that much extra damage. If the hero kills the enemy and another hero turn remains, the hero can continue attacking for a chance to heal even more health.

2.3. User Classes and Characteristics

This project will be geared towards a singleplayer model. The player is the favored user class and their enjoyment of the game is paramount. For this to be achieved, the player must have an operating system that is compatible with Oracle OpenJDK version 19. Serialization works well with GSON 2.1+

2.4. Operating Environment

The software will need to operate on an operating system with access to the keyboard for shortcuts to the GUI. The software will be stored within a repository within github and operate on JDK version 19. The program will utilize GSON for all of its serialization and deserialization for the save and load.

2.5. Design and Implementation Constraints

The database we will be using is SQLite and can not contain especially cumbersome data. Additionally, all commits must be tested before merging with the main branch as well as

proper documentation. When a functionality is finished, it must be delivered to one or more of the other teammates.

2.6. User Documentation

The help option will be made available to the user at the beginning of the game and will display a detailed description of the various rules and features of the game. If the user needs this help option again, it is also made available outside of combat while traversing the maze.

2.7. Assumptions and Dependencies

The assumption is made that the user understands English and can accurately locate buttons on their keyboard. The user is capable of viewing the GUI displayed on the screen of the operating system. The user has access to the internet in order to retrieve the file from github to run on a local machine. The user has an operating system that can run Oracle OpenJDK version 19. Finally, the assumption is made that the user is capable of hearing audio from the game, however it is not a requirement.

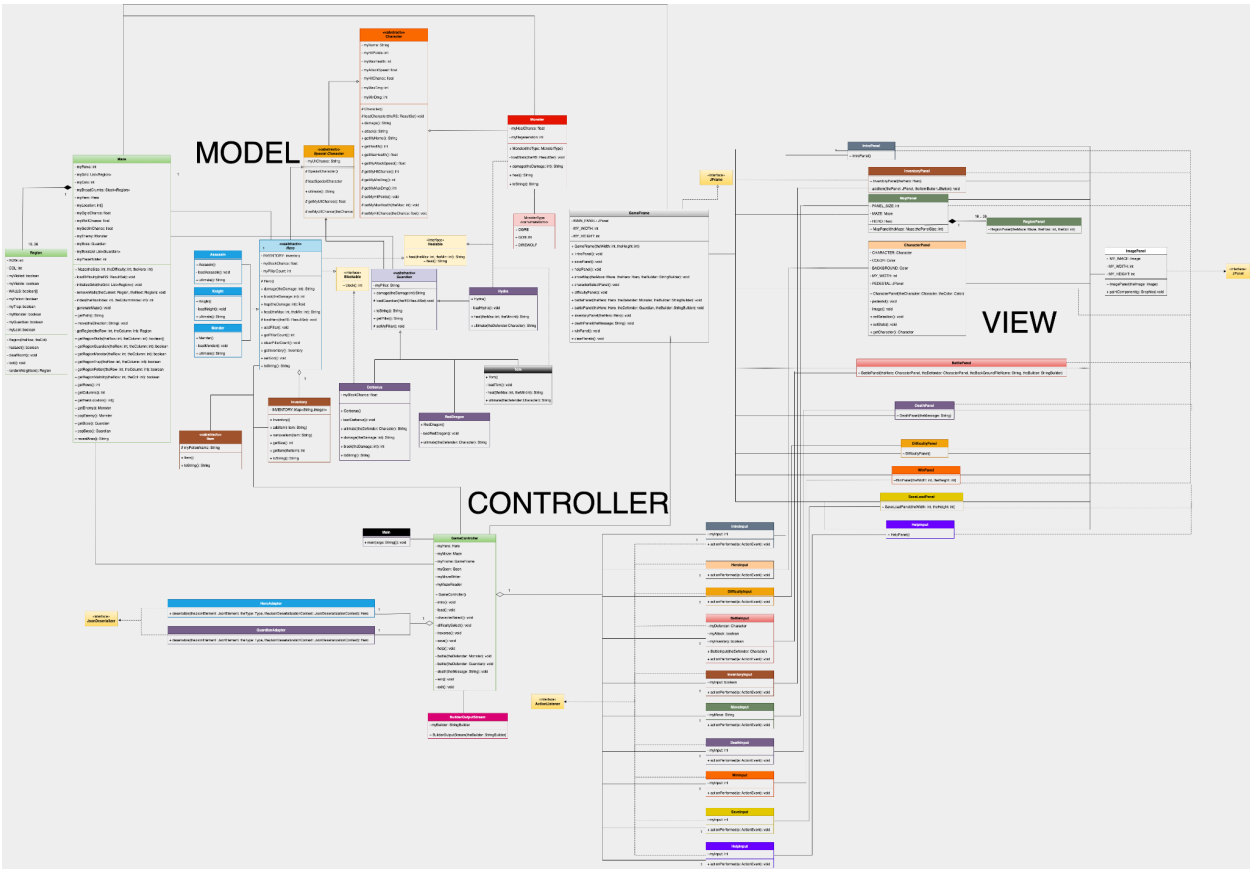

Figure 1: UML Class Diagram Draft

3. System Features

3.1. The Maze

The maze will be randomly generated and will contain region nodes that have a chance to contain items, hazards, or enemies. The maze's size and contents will be determined by the user when a new game is selected from the GUI. The maze will not allow the player to move in directions that are inaccessible and to reinforce this, the buttons will be grayed out and disabled.

3.2. Title Screen

This is the screen the player will see every time upon playing the game. Through this interface, the player can choose to start a new game, play from saved data, or view the help page.

3.3. Characters

The game will have a variety of characters of which the player will interact with. This includes monsters and guardians throughout the maze. A character has the following characteristics: health points, attack speed, chance to hit, and damage range. From the characters branches monsters and special characters. These special characters have special abilities and are categorized as heroes and guardians.

3.3.1. Heroes

A player chooses a hero to play as. These characters each have their own unique special skill and a chance to block an attack. They will serve to be a means of diversifying play styles.

3.3.1.1. Knight

Health points	150
Attack speed	2
Chance to hit	0.85
Chance to block damage	0.5
Minimum damage	45
Maximum damage	75

Special skill	Fatal slash
Fatal slash minimum damage	75
Fatal slash maximum damage	175
Fatal slash chance to hit	40

3.3.1.2. Mender

Health points	100
Attack speed	6
Chance to hit	0.9
Chance to block damage	0.3
Minimum damage	35
Maximum damage	45
Special Skill	Rejuvenate (heal)
Heal range	40-85
Chance to proc heal	0.95

3.3.1.3. Assassin

Health points	100
Attack speed	9
Chance to hit	0.9
Chance to block damage	0.4
Minimum damage	10
Maximum damage	80
Special Skill	Rhythm Echo

Rhythm Echo chance to proc additional attack	0.4

3.3.2. Monsters

These are the most common characters that the player will encounter as they play through the game. They all possess the ability to regenerate (at varying degrees). These will serve to provide a basic challenge to the player.

3.3.2.1. Ogre

Health points	250
Attack speed	2
Chance to hit	0.6
Minimum damage	30
Maximum damage	70
Chance to heal	0.1
Heal	45

3.3.2.2. Goblin

Health points	100
Attack speed	5
Chance to hit	0.8
Minimum damage	20
Maximum damage	40
Chance to heal	0.4
Heal	15

3.3.2.3. Dire Wolf

Health points	150
Attack speed	4
Chance to hit	0.8
Minimum damage	30
Maximum damage	50
Chance to heal	0.4
Heal range	30-50

3.3.3. Guardians

These are the characters that the player will only encounter at a Tower of OO. They all possess special skills and will serve as a worthy challenge in battle. One guardian will be assigned to each Tower. These will serve to be a difficult challenge for the player.

3.3.3.1. Hydra

Health points	250
Attack speed	3
Chance to hit	0.7
Minimum damage	30
Maximum damage	50
Chance to proc special skill	0.4
Special skill: Regeneration	Heal: 80-100

3.3.3.2. Tom

Health points	300
Attack speed	4
Chance to hit	0.65
Minimum damage	40

Maximum damage	60
Chance to proc special skill	0.7
Special Skill: mOOn Lecture	Heal: 30-50 Damage: 40-60

3.3.3.3. Red Dragon

Health points	325
Attack speed	2
Chance to hit	0.55
Minimum damage	30
Maximum damage	60
Chance to proc special skill	0.8
Special Skill: Incinerate	Damage: 85

3.3.3.4. Cerberus

Health points	250
Attack speed	5
Chance to hit	0.6
Minimum damage	30
Maximum damage	50
Chance to proc special skill	0.35
Special Skill: Multi-bite	Damage: 50-65

3.4. Save/Load State

The player will have a method of saving the game while not in combat in the maze. The player can thereafter load from the savestate to continue playing the game where they left off from the intro screen.

3.5. Difficulty level

The player can choose between three difficulty levels that will allow the player to experience mazes with varying sizes and monsters.

3.6. Monsters spawns

The player can encounter various monsters in the woodlands, however, the monsters available in random encounters depend on the difficulty level. The player can find these monsters at random while exploring the maze.

3.7. Potions

The player can encounter potions that spawn at random in the maze. The health potion can heal the player a considerable amount when they find themselves in a bind while in or exiting combat. The vision potion will help the player see all 8 adjacent squares - good for when there is a fork in the road and an indecisive player.

3.8. Combat

The player has the opportunity to engage in combat with 7 different adversaries. When within combat, the player has a choice to hit a basic attack with a high land rate, a special attack with a lesser land rate, or access inventory with no cost to the turn counter.

3.9. Overboard Damage

In the interest of keeping the game difficult yet interesting, monsters can deal a great deal of damage if they land their shots. The player must make difficult decisions in order to maximize overboard damage when low on health. Pushing the enemy's health bar below zero offers the hero a chance to heal up to the amount of overboard damage dealt.

3.10. Winning

Beating the game is a special achievement, but why let it stop there? The game will let the player keep playing with the items and health they won with. This allows the player to gradually escalate from easy mode to hard with items they earned from successful previous runs.

4. External Interface Requirements

4.1. User Interfaces

Our interface will receive input through buttons in the view that are wired to action listeners in the controller. We will have a variety of different panels to make sure that the entire game is playable from within GUI. It will begin with an intro panel, this will prompt the user if they want a new game, to load a game, or to get help understanding the game. Following the new game, we have the character select; they have an option to choose a Knight, a Mender,

or an Assassin. Next we choose the difficulty that determines the size of the maze and its contents. From there, we will begin our journey in the map GUI. The user can access their inventory, save the game, find the help button, and move from here. Once a player encounters an enemy, the battle panel activates. If the player dies, activate a unique death panel depending on what they've died by. If the player exits with all the pillars of OO, display the win panel and ask them if they want to keep playing with all of their acquired inventory.

4.2. Software Interfaces

The product will connect to a SQLite database that will store and maintain player data for loading heroes at the beginning and enemies as they are encountered throughout the maze. The game will run on Oracle OpenJDK version 19 and will be made accessible through github. The product will utilize GSON for its serialization and deserialization.

5. Other Nonfunctional Requirements

5.1. Performance Requirements

There are no concrete requirements for performance, however there should not be a delay that surpasses 3 seconds. Such a wait is unreasonable and could be improved upon by sparsing out operations amongst different calls and actions. Rather than loading the entire game at once, it could be loaded as the need arrives.

5.2. Security Requirements

In terms of security, the main concern to the project is the proper encapsulation of objects, methods, and variables such that all visibilities are reviewed and deemed necessary for the program to function properly. All necessary security is at the developer level, thus, all users may be treated the same. Maintain the tightest encapsulation and reduce the amount of public methods to the model especially.

5.3. Software Quality Attributes

Each program component must be tested by the developer and reviewed by another developer before being merged to the main branch. The GUI must contain buttons and shortcuts to suit different play styles. Additionally, the GUI must include a help button to properly define the rules of the game. Finally, the game must remain unique throughout each iteration of the game. In order to maintain replayability, the maze will be regenerated in a random pattern so that each user experience can be different from the last.

6. Appendix and Revision History

Appendix A: Glossary

- GUI: Graphical user interface
- JDK: Java development kit
- SQLite: a database
- Panel: referring to JPanel and it holds an optionally interactable image for the view

Appendix B: Issues List

- Lack of formal unit testing poses possible unforeseen problems.

Revision History

Name	Date	Reason For Changes	Version
Nick	Dec 14, 2022	Project changed a great deal and SRS was no longer relevant.	2.0