

---

# **Software Requirements Specification**

**for**

# **Lost in Peril**

**Version 1.2 approved**

**Prepared by Dennis McMeekan & Tim Mardesen**

**<CS 491>**

**April 14, 2019**

# Table of Contents

<b>Table of Contents</b>	<b>ii</b>
<b>Revision History</b>	<b>ii</b>
<b>1. Introduction</b>	<b>4</b>
1.1 Purpose	4
1.2 Document Conventions	4
1.3 Intended Audience and Reading Suggestions	4
1.4 Project Scope	4
1.5 References	5
<b>2. Overall Description</b>	<b>5</b>
2.1 Product Perspective	5
2.2 Product Features	6
2.3 User Classes and Characteristics	6
2.4 Operating Environment	6
2.5 Design and Implementation Constraints	6
2.6 User Documentation	6
2.7 Assumptions and Dependencies	7
<b>3. System Features</b>	<b>7</b>
3.1 Save/Load	7
3.2 Items/Upgrades	7
3.3 NPC/Boss with AI	8
3.4 Pause Menu	8
3.5 Free Movement	9
<b>4. External Interface Requirements</b>	<b>10</b>
4.1 User Interfaces	10
4.2 Hardware Interfaces	10
4.3 Software Interfaces	10
4.4 Communications Interfaces	11
<b>5. Other Nonfunctional Requirements</b>	<b>11</b>
5.1 Performance Requirements	11
5.2 Safety Requirements	11
5.3 Security Requirements	11
5.4 Software Quality Attributes	11
<b>6. Other Requirements</b>	<b>12</b>
<b>Appendix A: Glossary</b>	<b>12</b>
<b>Appendix B: Analysis Models</b>	<b>13</b>
<b>Appendix C: Issues List</b>	<b>16</b>

## Revision History

<b>Name</b>	<b>Date</b>	<b>Reason For Changes</b>	<b>Version</b>
Tim Mardesen	3/2/19	Updated To-Do	0.9
Tim Mardesen	4/14/19	Minimum System Requirements Updated	1.2

# 1. Introduction

This section describes the contents of this document, and to inform the reader of some reading conventions to keep in mind while reading.

## 1.1 Purpose

This document contains detailed descriptions of the requirements for “Lost in Peril”. This software is shipped as an executable that requires no other prerequisites for those who intend to play the game. Therefore, entire system of this application is covered in this document.

## 1.2 Document Conventions

Items that are highlighted are short snippets to take away from that specific section. These items are highlighted so that it is easier, as well as quicker to understand the gist of the current section and to help “skimmer” style readers.

## 1.3 Intended Audience and Reading Suggestions

This document is intended to be read by all persons who develop, market, distribute, and enjoy the video game. When questioning the reasoning behind certain decisions of the gameplay, deeper meaning behind the plot, or simply want more hints on how to beat certain levels, this is the document that should be referenced. Customers and/or the end users are encouraged to read this document, and after reading, the customer will have a full understanding of this software and how to use it. The developing team should reference this to create the first revision of this software. The testing team will need to verify that the details of this document match the actual program, and relay any and all inconsistencies to the developing team to be improved. Persons involved with distributing or marketing this software need to be familiar with the specifications to accurately describe the product to potential end users.

It is suggested to read the appendices first to encapsulate the ideas of the systems used to create this software. This will help understand the game's background, and how it was developed. From there on, read this document from top to bottom. This will help the reader see how this application works from downloading it, through working the features, and completing the story mode of the game.

## 1.4 Project Scope

“Lost in Peril” is a video game that allows players to challenge their skills against the obstacles of the game, as well as challenge their colleagues by obtaining scores that are higher than one another. By making the core of this game open source, we hope to gain the business advantage of having a larger customer support group that remains free, as it is open source material. A

goal that the owners of this application share is to have fans of this program give us feedback on items they like, and changes they'd like to see made.

In future releases of this software, multiplayer gaming and custom server hosting will be added to this document, including sections describing the dependencies needed, minimum experience of the user that is expected to manage a custom server, how to use the server, and tactics to make a server more popular in the “Lost in Peril” community.

## 1.5 References

Some links that are listed are not used in Revision 1 of the software, and are simply listed here for developer reference to create the next revision.

GNU License for Server - if players want to host their own server, must remain open source and distributed, and must also be forkable, keeping the same license. Title, Author, Version Number, Date, and source or location can be found by going to the links and researching.

JavaFX - Used for graphics and menus. - <https://docs.oracle.com/javase/8/javafx/api/toc.htm>

Jenkins - Makes continuous integration easier. - <https://jenkins.io/doc/> (future versions)

Apache Tomcat -To host global scores of all players. - <http://tomcat.apache.org/tomcat-7.0-doc/>

Jackson ObjectMapper - exports objects as JSON - <https://github.com/FasterXML/jackson-docs>

Tiled map designer - <https://doc.mapeditor.org/en/stable/>

Gradle - Build Automation Tool - <https://docs.gradle.org/current/userguide/userguide.html>

Maven - used inside Gradle compilation - <https://maven.apache.org/>

Eclipse Photon - IDE for development - <https://www.eclipse.org/photon/>

## 2. Overall Description

### 2.1 Product Perspective

“Lost in Peril” is a standalone, self contained program that run in Java. There will also be dockerized versions of this video game that will be distributed to include all possible dependencies for the game to play, meaning that this document will cover the entire application system, from supported environments onward.. This product serves the purpose of entertainment for those who enjoy the early 2000’s style of gaming. This entails a “top down” view, bosses that require clever, outsmarting techniques to defeat, and pixel styled artwork. Lost in Peril is a computer video game, designed to emulate the video games that many played in their earlier years, while adding a modern touch to the plot. The games that are being modeled

after include Zelda, Pokemon Mystery Dungeon, and Shining Force. Revision 1.0 will include single player mode and a tutorial.

## 2.2 Product Features

Multiplayer options, highscore rankings (global and local) Save/Loading feature, and game pausing are a few features that have instructions below on how to use. Other features, such as multiplatform support, dockerized application usage, and source code viewing are not listed below. Refer to APPENDIX B, IMAGE 2 for more high level specifications.

## 2.3 User Classes and Characteristics

The main player class (Randy) is played by the consumer of this product. After interacting with the Start screen and clicking Start new game or loading a previously saved game, the player can control Randy and navigate him around the maps, attacking NPCs and beating levels.

There are no other roles for this game in revision 1. Revision 2 will have a Administrator user class where an Admin can host their own server for Lost in Peril, where a dockerized server could be deployed to allow for game map development, and a local host for multiplayer LAN games.

## 2.4 Operating Environment

Lost in Peril requires the JRE (Java Runtime Environment) version 11 or higher, and is designed to be played on Any Linux, Windows, or MacOS operating system that has the Java 11 JRE installed. OpenGL 2.0 or higher is also required.

Supported Hardware:

- GeForce 6 (6xxx) series or higher
- Radeon R300 Series (AGP Slot 9000)
- Intel Celeron CPU or newer

<Future support for Android and iOS>

## 2.5 Design and Implementation Constraints

Java applications are not natively supported on Xbox, Playstation, or Nintendo systems. Since there is no support for these platforms, there is a restriction on the potential user base, as people who do not have a device from the list of supported devices cannot play this game.

Server availability is limited, as the internet connection to the server device is spotty. Without a trustworthy internet connection to the main server, multiplayer gaming is impossible.

## 2.6 User Documentation

The documents that will be delivered with our software, Lost in Peril, will include but not be limited to a user manual and also a tutorial level to provide the users with the basics on how to complete and play the game.

The README.txt will be supplied with the software that describes the steps needed to get the software installed and running, as well as the basic controls and strategy to beating “Lost in Peril”

## 2.7 Assumptions and Dependencies

Assuming the user follows the direct instructions of having Java Runtime Environment and having the capable memory and processing features on their device, there are not many third-party or commercial components that have yet been added to our software. <TBD>

It is also assumed that the intended end user can read, understand how to operate a computer system, and that the intended end user also has a keyboard and a mouse available to use for this software.

# 3. System Features

With the development of Lost in Peril, there are many features available to make the best gaming experience possible for the user. The explanation of these features is to provide a basis on what we plan to implement and provide, and in the future we are able to improve or add on many more features.

## 3.1 Save/Load Game

### 3.1.1 Description and Priority

Users will be given the option to save their game through another feature which is the pause menu, this allows each user to be able to resume the game at the point he or she saved if the user decides to exit. This then loads the current level, score, time, and health remaining for the user at a later time done through the main menu.

### 3.1.2 Stimulus/Response Sequences

Using a xml/json file that contains stats of characters, items, locations, timestamps, and current score the user can save the game at the point they wish, or load the game through the main menu.

### 3.1.3 Functional Requirements

This will be performed by using Jackson Java Parser.

REQ-1: User needs to manually click save.

REQ-2: Developer needs to implement Jackson Parser for Java.

## 3.2 Upgrades/Items

### 3.2.1 Description and Priority

Throughout the game, there will certain upgrades or items available that either will increase and/or increase the amount of damage Randy is able to take or deal to the NPC. These items include topics such as weapon upgrades, shields, and secret items that are just given as a easter egg to add more to the overall score. This will require very minimal added coding and precision due to the fact that it will be easy to add the effects of each item to either the user's score or character.

### 3.2.2 Stimulus/Response Sequences

The picking up of these items and upgrades will be quite easy to use for the user by simply pressing a certain control button that is still to be determined, such as the "X" key.

### 3.2.3 Functional Requirements

This will require through the usage of JavaFX, an item or upgrade will have to appear on the floor and be the conditions be set to not true, in terms of Randy having them picked up. One pick up by the user, the item or upgrade then needs to be set to true, and removed from being able to be picked up. Then, the needed attributes then be applied to Randy.

REQ-1: User will manually press the "X" key to add item or upgrade to character or score.

REQ-2: Software must maintain and remove items or upgrades as need necessary.

## 3.3 NPC/Boss with Artificial Intelligence

### 3.3.1 Description and Priority

The goal of the game is to conquer each level which requires that the user completely wipe out all enemies of each level, to ultimately reach the final boss fight in which a user then beats the game. These NPC's will have the ability to intelligently attack Randy, or the user, to make the game challenging. This will definitely be quite the challenge but is something that will take our software to level we wish to achieve.

### 3.3.2 Stimulus/Response Sequences

With these NPC's, there will be basic movements and motion that they will do to try and hurt the user to make the game challenging and enjoyable. Through this, the user will have to strategically avoid getting hurt and eliminating all the enemies of each level.

### 3.3.3 Functional Requirements

The NPC and Boss that are going to need be defeated by Randy through the storyline will require specific coding and precise measurements of how to react on each level to actually to damage to Randy will be done through JavaFX, the specifics on this are still to come. <TBD>

REQ-1: Each NPC will have to be programmed to hold basic statistics and abilities that will be able to harm Randy.



- REQ-2: Randy must be reactive to these NPC and be able to take damage and respond accordingly.

### **3.4 Pause Menu**

#### **3.4.1 Description and Priority**

The Pause Menu will be a clear high priority, as it allows access to several different features such as Saving and Exiting. This menu will have a clear and sleek look, which will then have the buttons appear that allow the user to access each feature.

#### **3.4.2 Stimulus/Response Sequences**

Through a key, such as “N” or “Tab” the user will then have a drop down menu which holds the options of save or exit. When these buttons are pressed by the user, the system will then access each of these latter functions of saving and exiting.

#### **3.4.3 Functional Requirements**

Similar to making a GUI, this will be done either through Java Swing or JavaFX to create a drop down user interface that responds on certain button actions by the user. In doing so, when the buttons are pressed to change the conditional to true, then the action must be applied to the game.

REQ-1: User will be able to manually save the game by pressing a save button.

REQ-2: Similar to the save button, user will be able to exit at any point feel need, whether their game is saved or not, to return the main menu.

### **3.5 Free Movement**

#### **3.5.1 Description and Priority**

Randy, the user, will be coded and have the ability to have free movement, which is something that will differ us from other common RPG like Shining Force. Other games with similar features have used a grid based system, which limits the ability for the user to connect with Randy, the character. This will be a slight challenge because it will be harder to allow the input strictly to determine character movements but is something needed to differ from other RPG.

#### **3.5.2 Stimulus/Response Sequences**

With JavaFX, our main character Randy will have free movement which will be coded and be done through straight user input. This may be challenging because of the fact that Randy's hitbox will almost always be moving and alternating since it won't be able to be recorded with a grid.

#### **3.5.3 Functional Requirements**

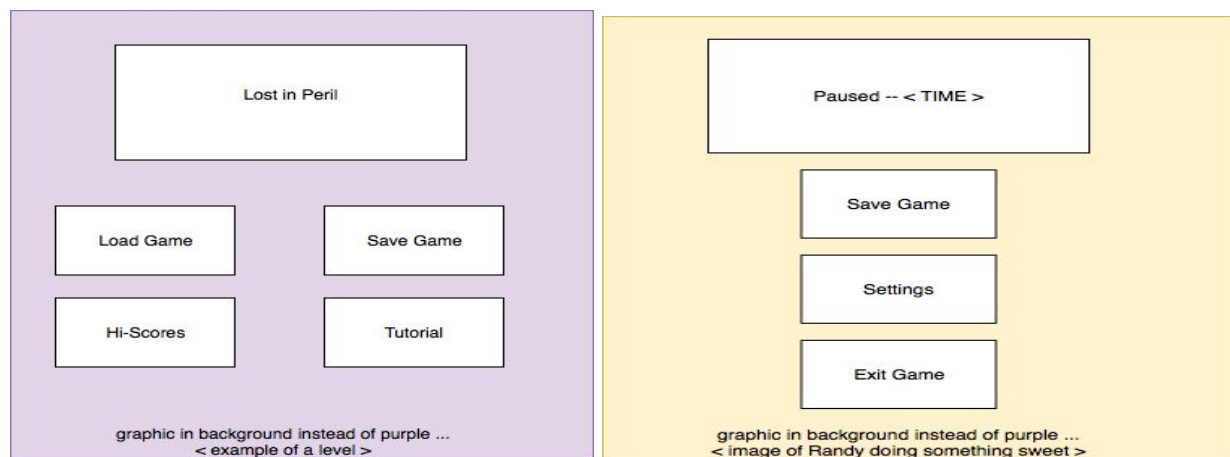
With the programming and creation of the NPC and Boss, it will allow us to determine the correct actions needed for Randy's movement and how to allow a common ground between both. Keeping the game challenging but also possible to do with skill. Will be done through JavaFX

- REQ-1: With the pressing of the “A, S, D, and W” keys, the user will be able to move Randy in the directions of Left, Down, Right, and Straight accordingly.

## 4. External Interface Requirements

### 4.1 User Interfaces

For the development of *Lost in Peril*, there will be at least two user interfaces that will absolutely need to be created for the user to have the best experience. One of these user interfaces will be the main menu, which allows the user to save and load, and the other will be a pause menu available mid game, which helps the user navigate through their items, saving the game, and exiting the game. Very basic models of these interfaces is provided below;



### 4.2 Hardware Interfaces

The hardware that will need to be used is an OS that is able to run Java Script and allow our file to run smoothly and efficiently, this includes an internal processor and memory storage. A keyboard for user input to allow the main character to move and interact with each level is also required.

### 4.3 Software Interfaces

With the ability to save and load our game, there will need to be a direct correlation between the json saving of the file and the user's system memory. When clicking the direct button of saving, we need to then make sure that the file be saved in memory but also in an organized manner, either allowing the user to name the save or know which one it is. There may be an implementation constraint in doing so, but we yet have figured out how to go about the code and the processing involved. <TBD>

## **4.4 Communications Interfaces**

In communicating between user and software developer, there will always be a support email available if any issues occur. Also, upon the implementation of our multiplayer feature there may be a FTP that needs to be involved to ensure that all data is able to be received and sent by each player accordingly. <TBD>

# **5. Other Nonfunctional Requirements**

## **5.1 Performance Requirements**

The performance of our program will mainly stay in the rational areas of java coding, meaning the ideally we will try to avoid anything that may cause performance issues on a basic processing unit. Through this, it is ideal that we always test our code at almost every level upon development to ensure the user the best ideal gaming experience. <TBD>

## **5.2 Safety Requirements**

Many physicians recommend to take breaks every 15 minutes to reduce the effects of video gaming. These effects include but are not limited to Video Game Addiction (VGA), Computer Vision Syndrome (CVS), insomnia, obesity, and eyelid twitch.

Stretching arms every 15 minutes is recommended as well to prevent carpal tunnel, which is caused by the hands being angled for a keyboard to an extended period of time. Good spine posture is also needed to play this game safely.

## **5.3 Security Requirements**

In the versions that allow for multiplayer use, a router and/or firewall is highly recommended to prevent unauthorized users from breaching your network. Without something filtering malicious internet traffic on your network, your computer system could be a target for malicious software installation.

Only download this software from the main website. Other websites that offer this software, as well as many other applications, are insecure and may clone this software in order to trick users into download this program on that website, but actually install spyware with this program. Do not download this software on websites other than the main page.

## **5.4 Software Quality Attributes**

In general, we want our product to be available to everyone in the general public, possibly making the game quite relatively cheap or having an open beta. Also, when the product is available to download it is a priority to ensure that the code and programming is working to it's

best available with no errors and is reliable. Possibly this means more testing on our end, but to ensure to the user they are getting the best gaming experience in Role Playing Games, the extra work is needed.

## **6. Other Requirements**

### Database or Storage:

With the implementation of a saved game, where the user can resume right where they left off even after closing the software, there will need to be some sort of database requirement where we are able to save this file for later use. Through xml or json file. **<TBD>**

## **Appendix A: Glossary**

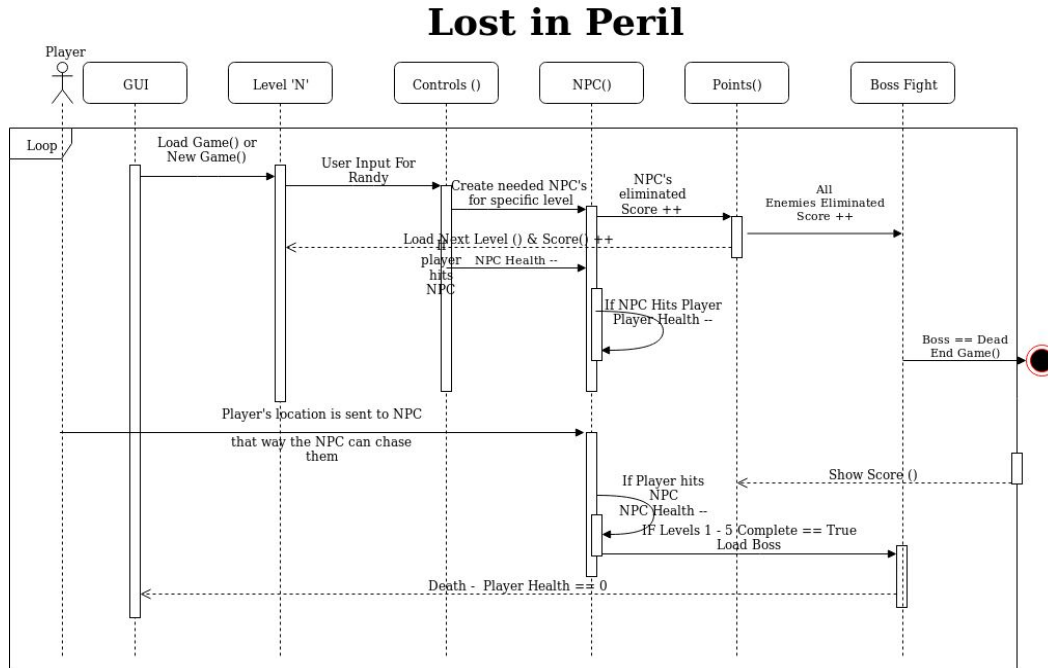
Lost in Peril: The name of the video game that this document describes.

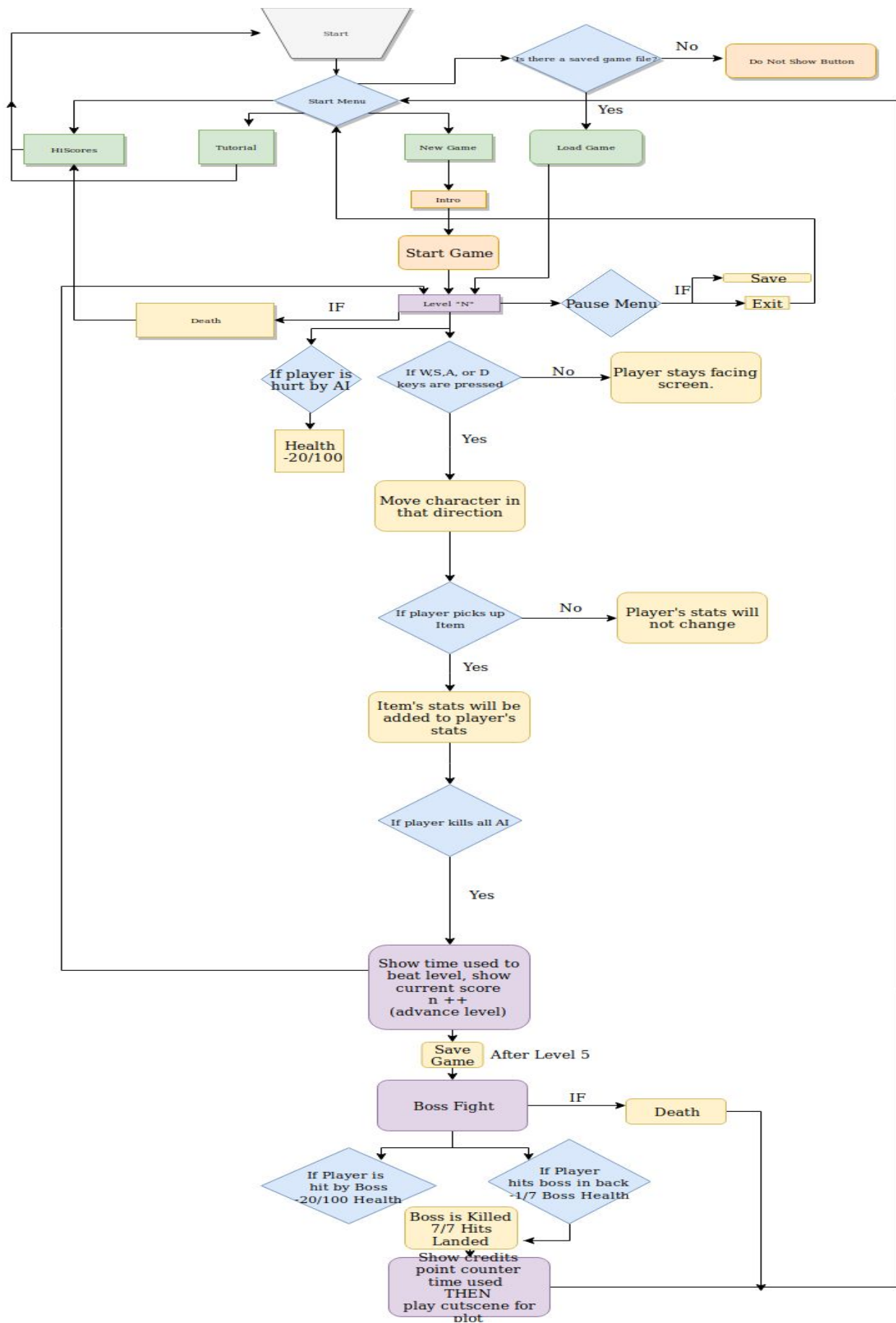
RPG: Role Playing Game

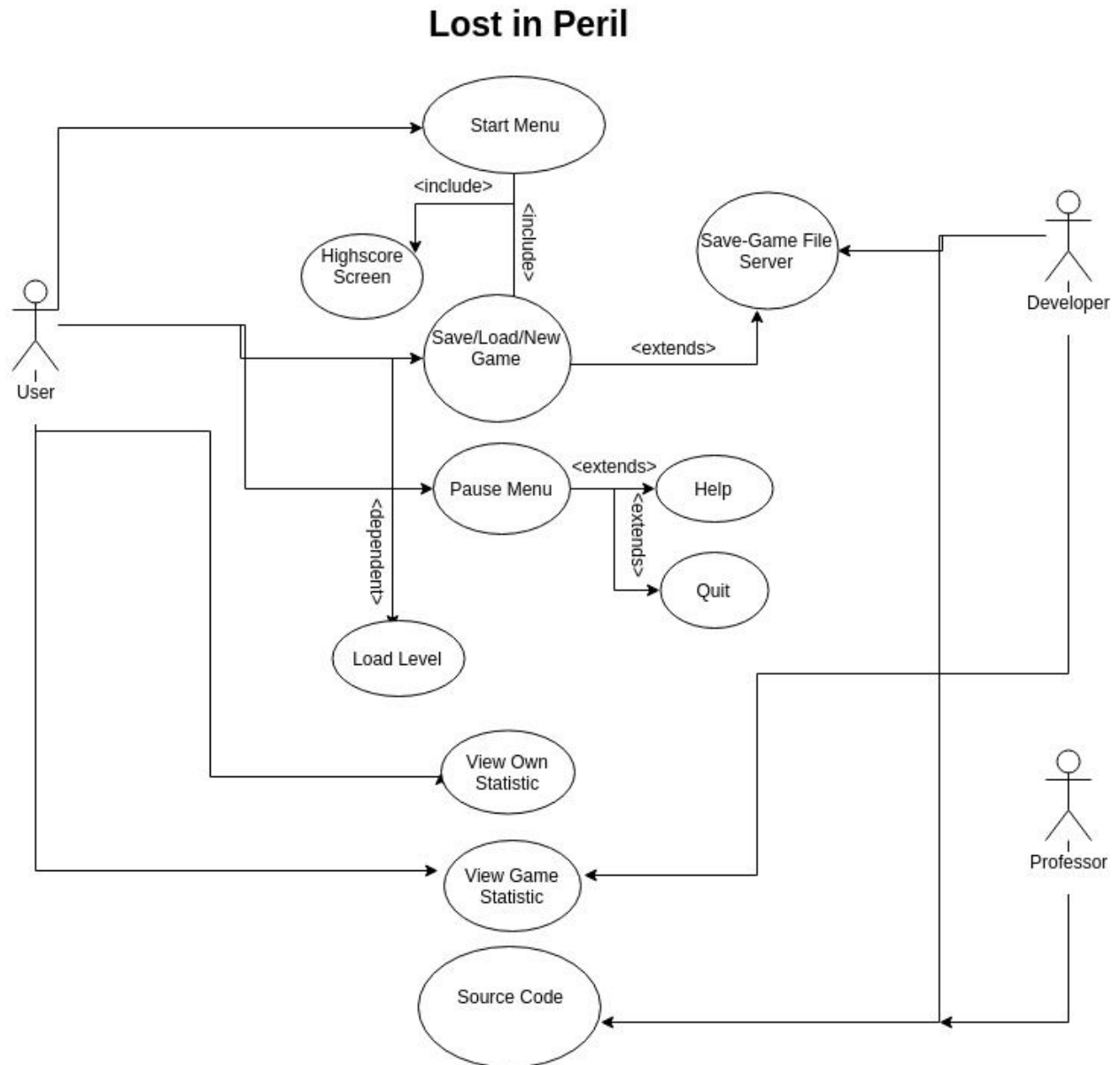
NPC: Nonplayer character

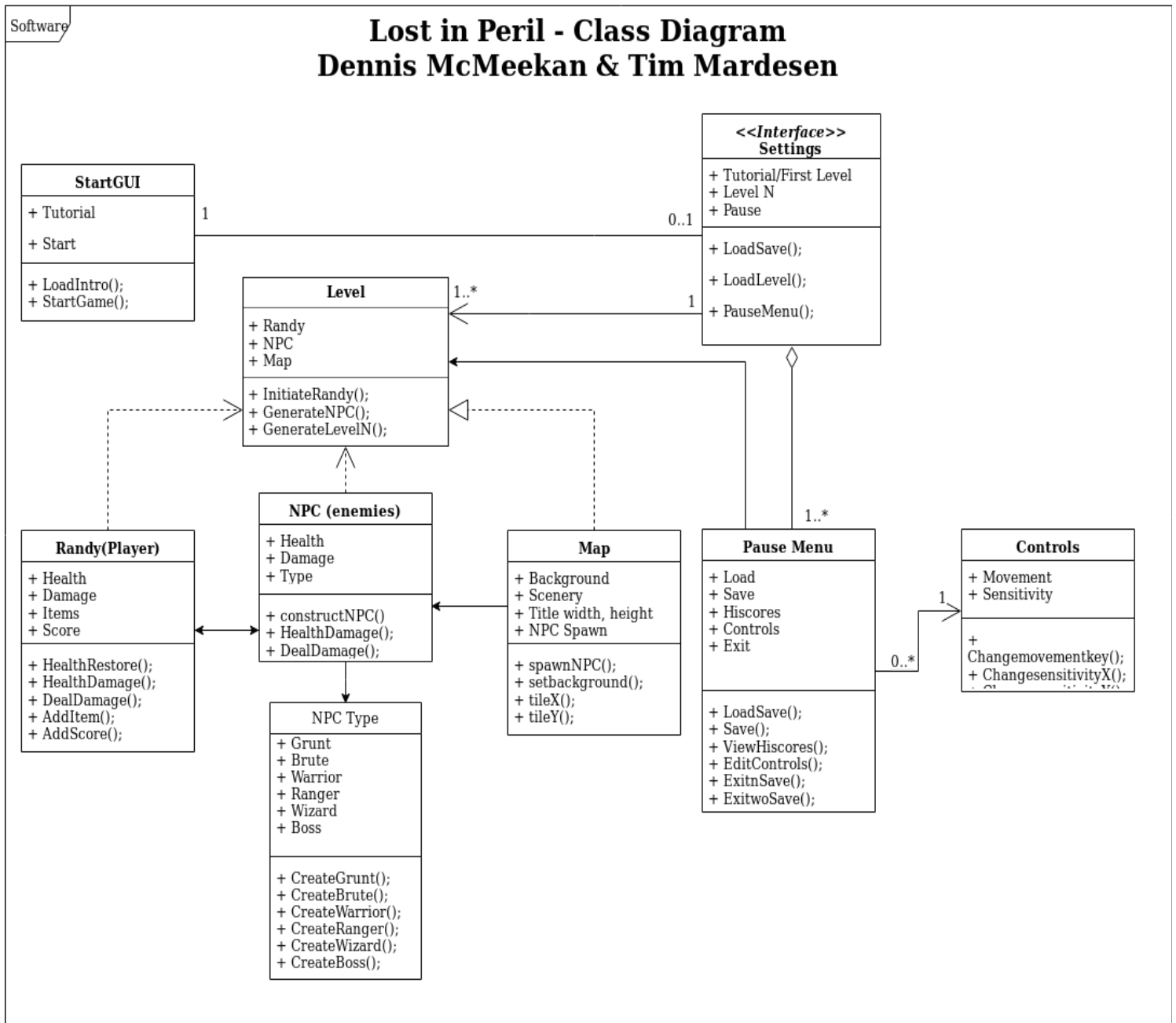
Randy: The main character the user plays

## Appendix B: Analysis Models











## **Appendix C: Issues List**

Update reference section (1.5)

Multiplayer support with TomCat server - TBD && sockets  
(main page to download software)

readme file to help install software & setup (Link to java jre)