**Software Requirements and Design Document**

**For**

**Group 4**

Version 1.0

**Authors**:

Brenden G

Stephen J

Benjamin W

Logan L

# Overview

“Beneath the Manor” is a rouge-like dungeon crawl game with 2d graphics and randomly generated levels. The player will navigate through rooms attempting to reach the deepest point in the maze while avoiding enemies and collecting items. The player’s top-down view will encompass one room at a time, including doors or openings that lead to other rooms. The layout of each floor (what rooms border each other, and the location of rooms with stairs leading to other floors) will be randomly generated.

In addition to the game itself, the game will be downloadable from a website that also includes tutorial and background flavor information about the game. The website will have a home, faq, download, and guide page. All of which will have content that will relate to the game.

# Functional Requirements

1. Graphically display the character in the window, and implement the character movement. (High priority)
2. Generate the random layout of each floor. (High Priority)
3. Generate the random layout of each room. (High Priority)
4. Make sure the character stays within bounds. (High Priority)
5. Letting the user use the arrow keys to move the character throughout the room. (Medium Priority)
6. Implementing items that can be picked up by the character. (High Priority)
7. Display a homepage for the website with a clickable button that downloads the runnable java jar file. (High Priority)
8. Connect the homepage using a nav bar to the other pages such as guide, download, and faq. (Medium Priority)
9. Provide parallax properties to website containers. This means that there is dynamic movement on the page while scrolling. (Low Priority)
10. Provide a guide to the game and answers to questions about the game in the form of an FAQ and Guide page. (Low Priority)

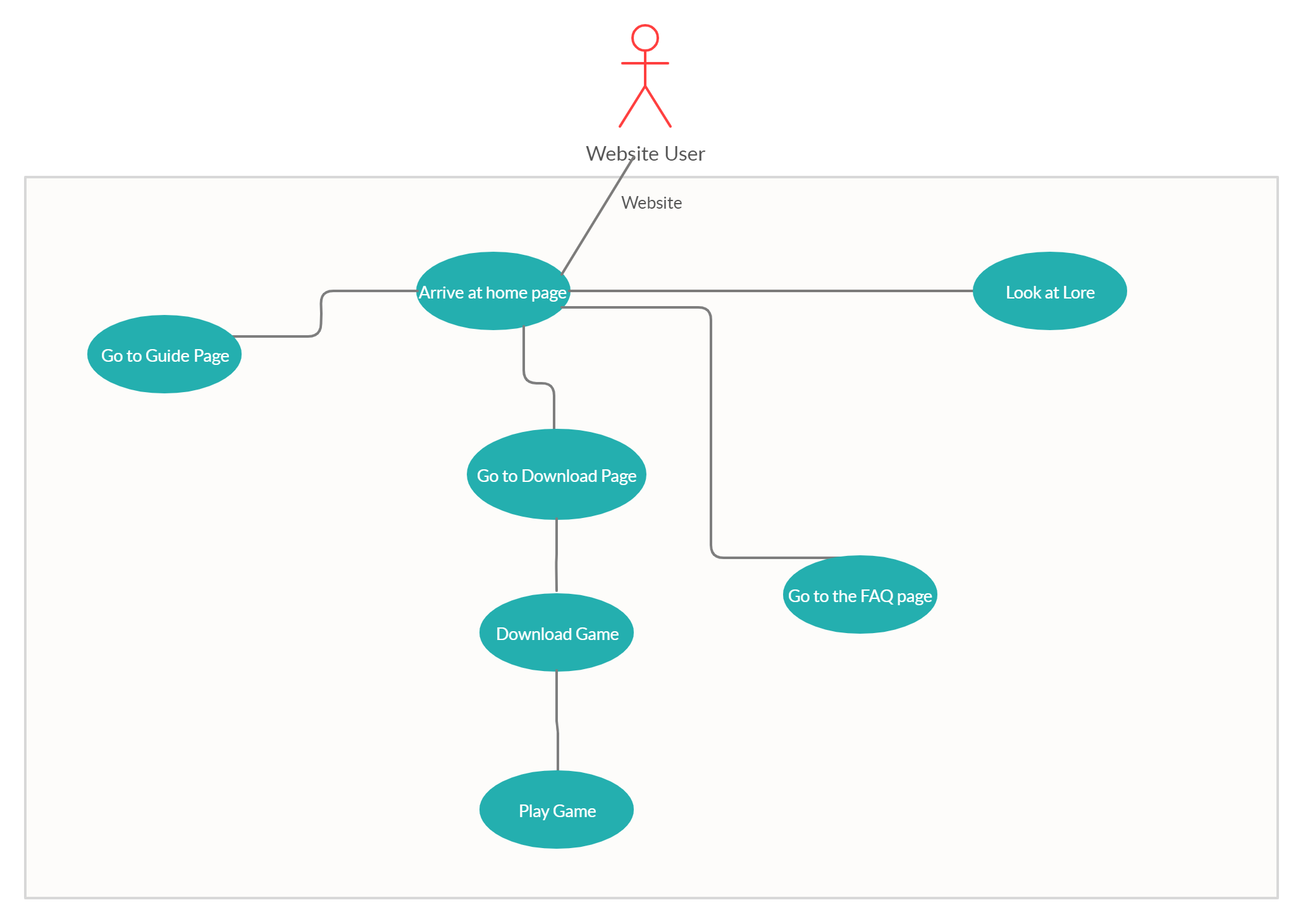
# Non-functional Requirements

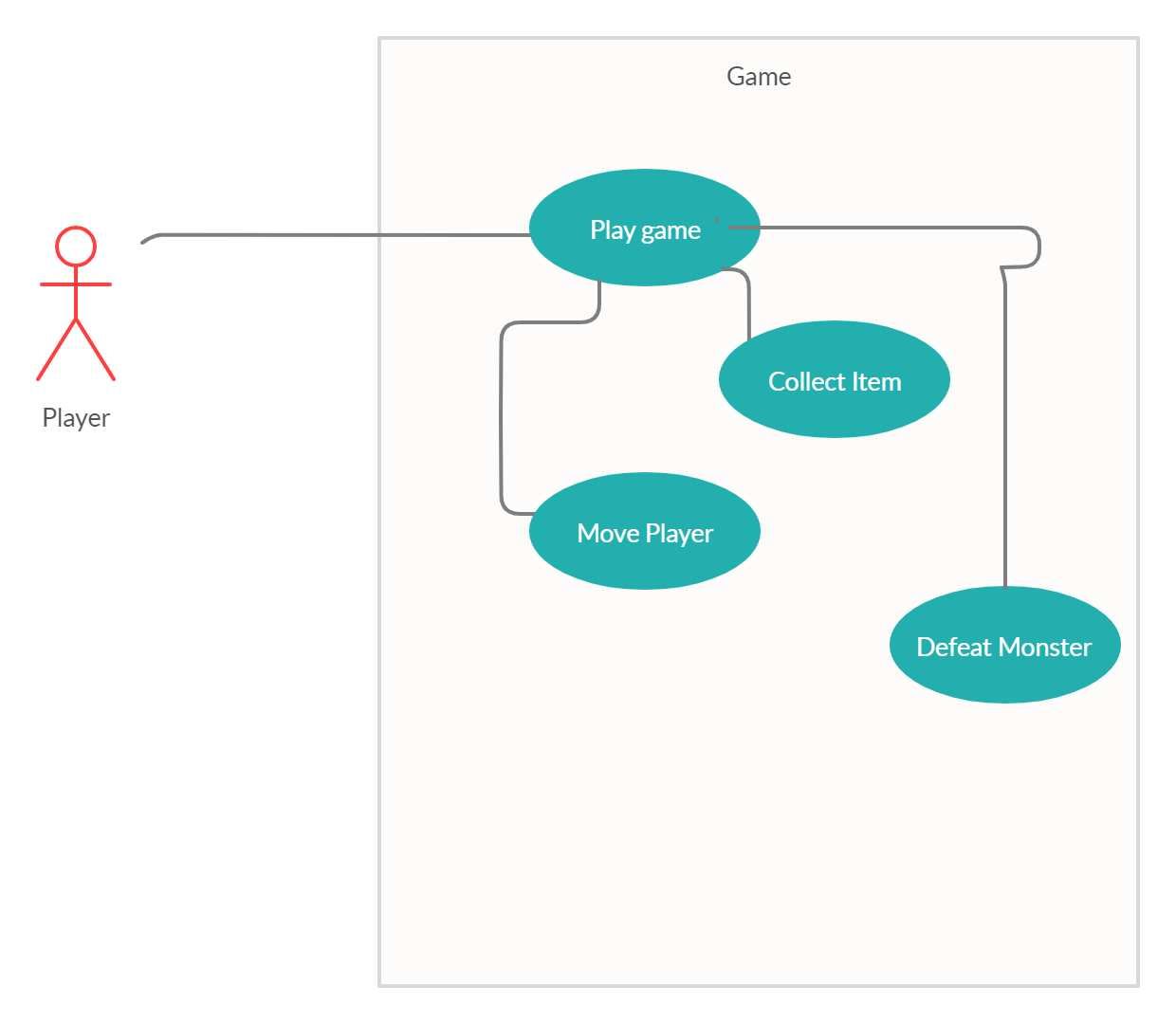
*List the* ***non-functional requirements*** *of the system (any requirement referring to a property of the system, such as security, safety, software quality, performance, reliability, etc.) You may provide a brief rationale for any requirement which you feel requires explanation as to how and/or why the requirement was derived.*

1. Having very little lag when the user is moving the character from space to space.
2. Generating new, random rooms quickly.
3. Scalability, for the website. The hosting platform Amazon Web Services (AWS) is a pay for what you use model. For example, if we got a million hits we pay for the exact number of people visiting our website and not just some flat fee.
4. Safety, for the website. In the near future once the website is complete we are going to add the free SSL certificate provided to make the website secure. This will encrypt data and provide users with a safer experience.

# Use Case Diagram

*This section presents the* ***use case diagram*** *for the system under development. The use case diagram should contain all the use cases and relationships between them needed to describe the functionality to be developed. If you discover new use cases between two increments, update the diagram for your future increments.*

Website Use Case Diagram**

Game Use Case Diagram

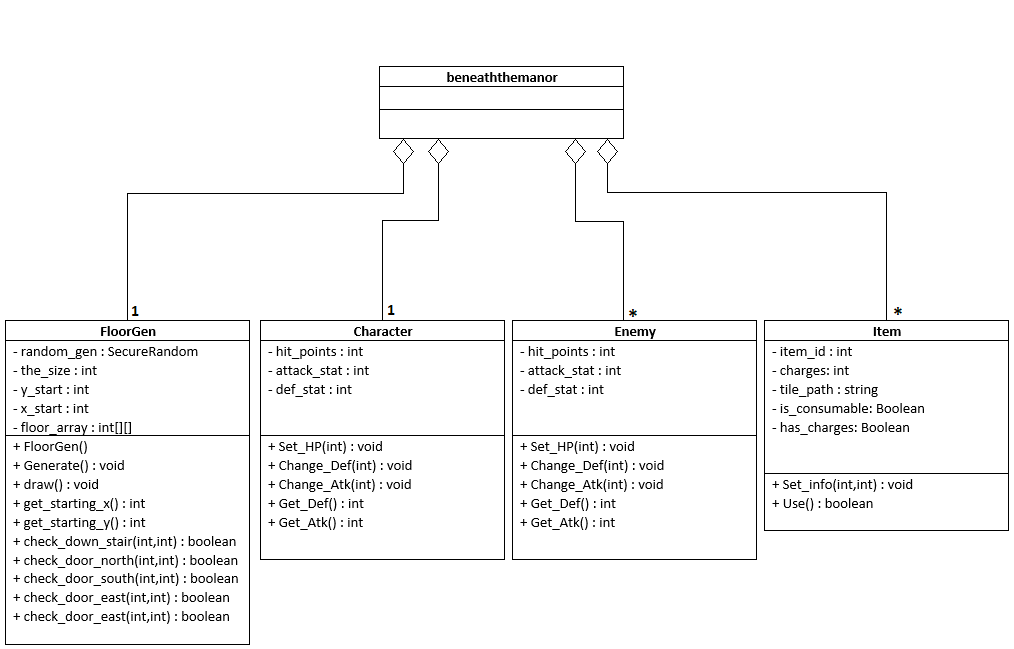
# Class Diagram and/or Sequence Diagrams

*This section presents a high-level overview of the anticipated system architecture using a* ***class******diagram*** *and/or* ***sequence diagrams****.*

*If the main* ***paradigm*** *used in your project is* ***Object Oriented*** *(i.e., you have classes or something that acts similar to classes in your system), then draw the* ***Class Diagram******of the entire system and Sequence Diagrams for the three (3) most important use cases in your system.***

*If the main* ***paradigm*** *in your system is* ***not Object Oriented*** *(i.e., you* ***do not*** *have classes**or anything similar to classes in your system) then only draw* ***Sequence Diagrams****,* ***but for all the use cases of your system.*** *In this case, we will use a modified version of Sequence Diagrams, where instead of objects, the lifelines will represent the functions in the system involved in the action sequence.*

***Class Diagrams*** *show the* ***fundamental objects/classes*** *that must be modeled with the system to satisfy its requirements and* ***the relationships*** *between them. Each class rectangle on the diagram* ***must also include the attributes and the methods of the class*** *(they can be refined between increments). All the* ***relationships between classes and their multiplicity*** *must be shown on the class diagram.*

**

*A* ***Sequence Diagram*** *simply depicts* ***interaction******between objects*** *(or* ***functions -*** *in our case - for non-OOP systems) in a sequential order, i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.*

# Operating Environment

The game is intended for personal computers. The game is written in Java, so it should run on any operating system that can support the java runtime environment. The website is intended to run on all modern desktop browsers and mobile browsers as well. Website is written using HTML, CSS, and Javascript. Bootstrap CSS libraries were used as well.

# Assumptions and Dependencies

*List any assumed factors (as opposed to known facts) that could affect the requirements stated in this document. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.*

The user knows how to run a jar file.

The user has the correct version of Java Runtime.

We assume the user has access to the link via internet to the website through Amazon Web Services hosting platform.

We are dependent on the Bootstrap and AOS open source libraries to be properly working once we are finished.