CMSC 447
Software Design and Development

Fall 2023

# System Requirements Specification

The Boundless Hollow
System Requirements Specification
Kabeer Alabi, CJ Vittek, Nahim Kamruzzaman, Tae Hyung Kim, Collins Ufua

**Table of Contents**


| | Page |
|---|---|
| 1. Introduction | |
| 1.1 Purpose of This Document | 4 |
| 1.2 References | 4 |
| 1.3 Purpose of the Product | 4 |
| 1.4 Product Scope | 5 |
| 2. Functional Requirements | 6,7,8,9,10 |
| 3. Non-Functional Requirements | |
| 3.1 Customer Constraints | 11 |
| 3.2 External Interfaces | 11 |
| 3.3 Other | 11 |
| 4. Deliverables | 12, 13 |
| 5. Open Issues | 14 |

Appendix A – Agreement Between Customer and Contractor 15

Appendix B – Team Review Sign-off 16

Appendix C – Document Contributions 17, 18

## Document Versioning Control

| Version Number | Date | Changes from Previous Version |
|---|---|---|
| 1.0 | 10/02/2023 | Decided on primary requirements |
| 1.1 | 11/01/2023 | Creates user cases for functional requirements |
| 1.2 | 11/29/2023 | Decided on non-functional requirements by customers. |
| 1.3 | 12/16/2023 | Final Docs |

# 1. **Introduction**

## 1.1 Purpose of This Document

This System Requirements Specification (SRS) document serves as a comprehensive guide outlining the requirements and specifications for the development of the Boundless Hollow Project. The intended readership includes software developers, project managers, quality assurance teams, and any stakeholders involved in the design and implementation of the game. Intended for both technical and non-technical readers, the document provides a detailed overview of the project scope and purpose, functional and non-functional requirements, and system constraints. It aims to ensure a clear understanding of the project's objectives and features, facilitating effective communication and collaboration among the development team and relevant stakeholders. The content of the document encompasses critical information essential for guiding the entire development life cycle of the Boundless Hollow project.

## 1.2 References

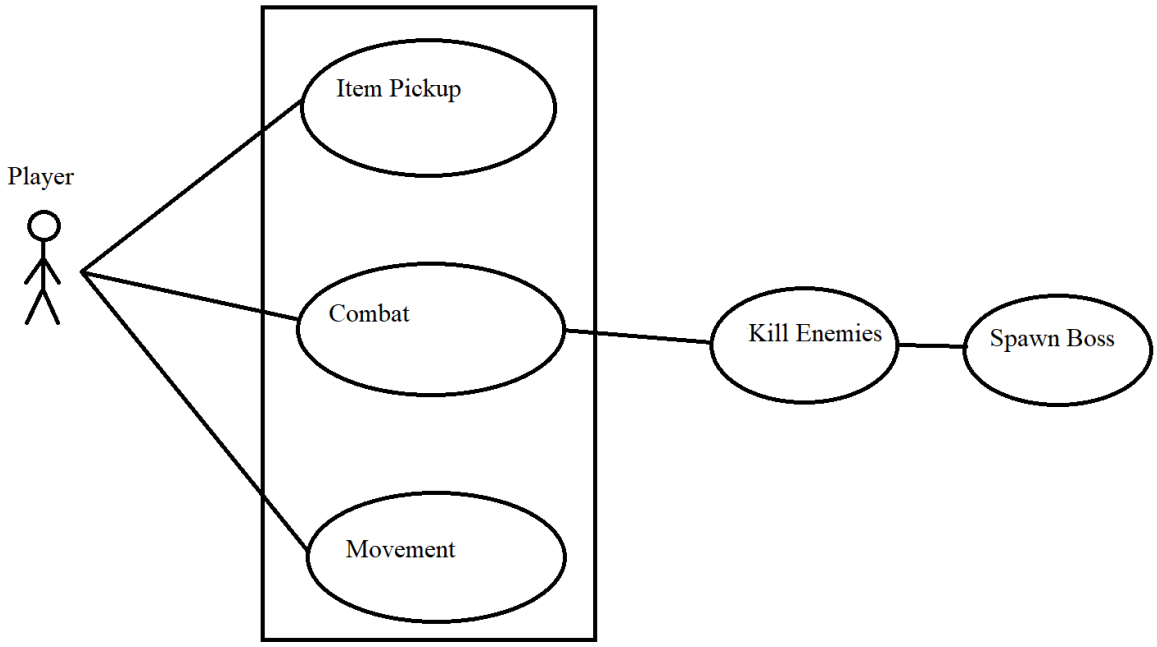
Unity Documentation: https://docs.unity.com/
Krita Documentation: https://docs.krita.org/en/index.html
Software Engineering 10th Edition: Ian Sommerville

## 1.3 Purpose of the Product

The Boundless Hollow is a 2D Medieval Dungeon Crawler designed to immerse players in an engaging gaming experience where they take on the role of a character navigating through a dungeon filled with enemies and challenges. The inspiration for this product stems from the desire to provide players with an exciting and strategic gaming environment. As it stands, players are seeking a gaming experience that combines exploration, combat, and progression. The project was initiated in response to the demand for a captivating dungeon crawler, leveraging the Unity engine for its development and Krita for visually appealing 2D art. The primary goal is to enable players to accomplish tasks such as defeating enemies, collecting items, and ultimately facing a formidable boss. By offering a rich gaming experience, the product aims to fulfill the need for an immersive and entertaining 2D medieval dungeon crawler.

## 1.4 Product Scope

### Top-level User Case Diagram




The Boundless Hollow is delineated by its boundaries, encapsulating the system to be developed within the Unity environment. The top-level use case diagram illustrates the system's scope, showcasing key functionalities and their relationships. Primary actors include the player, enemies, and the boss. Players engage in actions such as movement, combat, and item collection, forming the core interactions within the system. The diagram clarifies that the product excludes login/logout functionalities, focusing on the core gameplay mechanics.

The classes of users are well-defined, with players as the primary actors, engaging in combat, exploration, and item acquisition. Enemies are secondary actors, serving as obstacles and sources of drops. The boss, as a pivotal element, is triggered by the player's progression through defeating a specified number of enemies. The diagram provides a concise overview of how the system interfaces with the external entities, emphasizing the interconnectedness of player actions, enemy encounters, and the ultimate boss fight. This carefully curated scope ensures a focused and engaging gaming experience for the users.

2. **Functional Requirements**

| **Number** | UC-01 | |
|---|---|---|
| **Name** | Character Controls | |
| **Summary** | Player navigates the character through the dungeon, collecting items along the way. | |
| **Priority** | 4 (1 = lowest priority, 5 = highest priority) | |
| **Preconditions** | Player is in the game environment. | |
| **Postconditions** | Player moving animations execute, and player position is updated, and items are collected if they collide with the player. | |
| **Primary Actor(s)** | Player | |
| **Secondary Actor(s)** | None | |
| **Trigger** | Player input to move the character. | |
| **Main Scenario** | **Step** | **Action** |
| | Step 1 | Player presses the directional keys. |
| | Step 2 | Character moves accordingly. |
| | Step 3 | If a character overlaps with an item, it is collected. |
| **Extensions** | **Step** | **Branching Action** |
| | Step 4 | Alternative paths that the use case may take |
| **Open Issues** | Issue | |

| **Number** | UC-02 | |
|---|---|---|
| **Name** | Combat System | |
| **Summary** | Player engages in combat with goblins using different weapons. | |
| **Priority** | 5 (1 = lowest priority, 5 = highest priority) | |
| **Preconditions** | Player has a weapon and encounters a goblin. | |
| **Postconditions** | Goblin defeated, and the player gains experience points (EXP) dropped from a goblin. | |
| **Primary Actor(s)** | Player | |
| **Secondary Actor(s)** | Goblins | |
| **Trigger** | Player encounters a goblin. | |
| **Main Scenario** | **Step** | **Action** |
| | 1 | Weapon attack triggers in the direction the player is facing based on a timer. |
| | 2 | Weapon attack animation plays. |
| | 3 | If successful hit, goblin health decreases. |
| **Extensions** | **Step** | **Branching Action** |
| | 4 | If goblin health reaches zero and the player collects an EXP drop, go to Character Progression (UC-03). |
| | 5 | If unsuccessful hit, goblin counterattacks. |
| **Open Issues** | Issue 1 | Player invulnerability frame duration needs to be tweaked. |

| **Number** | UC-03 | |
|---|---|---|
| **Name** | Character Progression | |
| **Summary** | Player levels up and gains attributes after defeating goblins. Weapons get stronger as player levels up from EXP. | |
| **Priority** | 4(1 = lowest priority, 5 = highest priority) | |
| **Preconditions** | Player defeats a goblin and picks up EXP. | |
| **Postconditions** | Player levels up and weapons get stronger (gain additional abilities). Goblins also get stronger. | |
| **Primary Actor(s)** | Player | |
| **Secondary Actor(s)** | Goblins | |
| **Trigger** | Player defeats a goblin. | |
| **Main Scenario** | **Step** | **Action** |
| | Step 1 | Player's EXP increases. |
| | Step 2 | If EXP threshold is reached, the player levels up. |
| | Step 3 | Weapons gain increased damage and abilities. |
| **Extensions** | **Step** | **Branching Action** |
| | Step 4 | Goblin damage and health increases. |
| **Open Issues** | Issue 1 | The player is currently unable to choose their desired upgrade and functionality needs to be implemented. |

| **Number** | UC-04 | |
|---|---|---|
| **Name** | Enemy AI and Attack | |
| **Summary** | Enemies chase and attack the player in the game. | |
| **Priority** | 5 (1 = lowest priority, 5 = highest priority) | |
| **Preconditions** | Player health decreases if attacked by enemies. | |
| **Postconditions** | What will be true after the use case successfully “executes” | |
| **Primary Actor(s)** | Enemies | |
| **Secondary Actor(s)** | Player | |
| **Trigger** | Player enters the enemy's detection range. | |
| **Main Scenario** | **Step** | **Action** |
| | Step 1 | Player enters the enemy's detection range. |
| | Step 2 | Enemy pursues the player. |
| | Step 3 | If in range, enemy attacks the player. |
| **Extensions** | **Step** | **Branching Action** |
| | Step 4 | If the player's health reaches zero, a death screen appears. |
| **Open Issues** | Issue 1 | Boss AI is not implemented currently. |

| | | |
|---|---|---|
| **Number** | UC-05 | |
| **Name** | Creation of 2D Art Assets and Animations | |
| **Summary** | All visual elements, including characters, items, and enemy sprites, are created using Krita. Multiple frames are created for characters and enemies to provide fluid movements during gameplay. | |
| **Priority** | 4 (1 = lowest priority, 5 = highest priority) | |
| **Preconditions** | 2D art assets are created in Krita and available. | |
| **Postconditions** | 2D art assets and animation sprites are integrated into the game. | |
| **Primary Actor(s)** | Artists | |
| **Secondary Actor(s)** | Arts Assets | |
| **Trigger** | Start of the game development process and completion of 2D art assets. | |
| **Main Scenario** | **Step** | **Action** |
| | Step 1 | Artists create character, item, and enemy sprites. |
| | Step 2 | Sprites are integrated together to create frames for animations. |
| | Step 3 | Add sprites and animations onto scene. |
| **Extensions** | **Step** | **Branching Action** |
| | Step 4 | If additional assets or animations are needed, return to Step 1. |
| **Open Issues** | Issue | |

| | | |
|---|---|---|
| **Number** | UC-06 | |
| **Name** | Level Designing | |
| **Summary** | Basic level design with continuous enemy spawning. | |
| **Priority** | 4 (1 = lowest priority, 5 = highest priority) | |
| **Preconditions** | Tile assets for floors and walls are available. | |
| **Postconditions** | Playable level created with enemies spawning and player placed. with a camera on the player. | |
| **Primary Actor(s)** | Level Designer | |
| **Secondary Actor(s)** | None | |
| **Trigger** | Start of the level development process. | |
| **Main Scenario** | **Step** | **Action** |
| | Step 1 | Level designer arranges floor tiles to create a basic layout. |
| | Step 2 | Walls are added to define the play area. |
| | Step 3 | Continuous enemy spawning points are set. |
| **Extensions** | **Step** | **Branching Action** |
| | Step | |
| **Open Issues** | Issue | |

| | | |
|---|---|---|
| **Number** | UC-07 | |
| **Name** | HUD Design | |
| **Summary** | HUD displays the player's health and experience during gameplay. | |
| **Priority** | 4 (1 = lowest priority, 5 = highest priority) | |
| **Preconditions** | Player is in the game environment. | |
| **Postconditions** | HUD is updated to reflect current health and experience. | |
| **Primary Actor(s)** | Player | |
| **Secondary Actor(s)** | None | |
| **Trigger** | Start of the game. | |
| **Main Scenario** | **Step** | **Action** |
| | Step 1 | The Player's health and experience are displayed on the HUD. |
| | Step 2 | Player's health and experience are updated real time on the HUD. |
| **Extensions** | **Step** | **Branching Action** |
| | Step | |
| **Open Issues** | Issue | |

| | | |
|---|---|---|
| **Number** | UC-08 | |
| **Name** | Creating Menus | |
| **Summary** | Players can customize game options and start the game from the menu. | |
| **Priority** | 3 (1 = lowest priority, 5 = highest priority) | |
| **Preconditions** | Game is launched. | |
| **Postconditions** | Player preferences are set, and the game is started or paused. | |
| **Primary Actor(s)** | Player | |
| **Secondary Actor(s)** | None | |
| **Trigger** | Player selects "Start Game" or accesses the options menu. | |
| **Main Scenario** | **Step** | **Action** |
| | Step 1 | Player navigates the menu to access game options. |
| | Step 2 | Player customizes preferences (if desired). |
| | Step 3 | Player selects "Start Game." |
| **Extensions** | **Step** | **Branching Action** |
| | Step 4 | Player can access options from pause menu in game. |
| **Open Issues** | Issue 1 | Issues regarding the use case that need resolution |

| | | |
|---|---|---|
| **Number** | UC-09 | |
| **Name** | Documentation Update | |
| **Summary** | Documentation is updated as progress is made throughout the project. | |
| **Priority** | 3 (1 = lowest priority, 5 = highest priority) | |
| **Preconditions** | Changes or progress in the project. | |
| **Postconditions** | Updated documentation reflecting the latest project status. | |
| **Primary Actor(s)** | Project Manager | |
| **Secondary Actor(s)** | Development Team | |
| **Trigger** | Regular project updates or milestones. | |
| **Main Scenario** | **Step** | **Action** |
| | Step 1 | Review and identify changes in the project. |
| | Step 2 | Update relevant documentation. |
| **Extensions** | **Step** | **Branching Action** |
| | | |
| **Open Issues** | Issue | |

(This template was adapted from *Basic Use Case Template***,** by Alistair Cockburn, http://members.aol.com/acockburn/papers/uctempla.htm, accessed 1/17/08.)

## 3. **Non-Functional Requirements**

### 3.1 Customer Constraints

#### 3.1.1 NFR-001: System Platform
Priority: 5
Description: The system shall be developed as a desktop application.

#### 3.1.2 NFR-002: Operating System Compatibility
Priority: 5
Description: The system shall run on Microsoft Windows 10, version 1903 or later.

#### 3.1.3 NFR-003: Graphics Rendering
Priority: 4
Description: The system's graphics and art assets shall be rendered smoothly on a minimum screen resolution of 1920x1080 pixels.

### 3.2 External Interfaces

3.2.1 This is an offline game that is single-player and thus there is no need for the reading of data files from an external system. All the data required is included in the download of the game and is on the local machine.

### 3.3 Other

Place the remaining non-functional requirements here. Do not include any NFRs that are related to the user interface. These will be included in the User Interface Design Document. Also, do not include NFRs that relate to hardware or software. Hardware and software specifications will be part of the System Design Document. As stated earlier, if your customer has already restricted you to particular hardware or software, put these NFRs in the Customer Constraints section above.

4. **Deliverables**

| Document | Contents | Delivery Time | Document Type |
|---|---|---|---|
| System Requirements Specification | Contains all of the information pertaining to the product requirements | 12/18 | Paper Copy<br>PDF |
| System Design Document | Contains all the information pertaining to the design of the product and how they meet the requirements | 12/18 | Paper Copy<br>PDF |
| User Interface Design Document | Contains all of the information pertaining to the UI and how it works and we transition from scene to scene | 12/18 | Paper Copy<br>PDF |
| Testing Report | Contains all the information pertaining to what's tests were performed, how they were performed, who performed them, what issues arose and how they were fixed | 12/18 | Paper Copy<br>PDF |
| Source Code | Contains all the code needed to run the projects | 12/18 | C# Files |
| Executable | Loads the program into memory and start the process, allowing the instructions within | 12/18 | Executable File |

| | | | |
|---|---|---|---|
| | the file to be carried out | | |

## 5. **Open Issues**

Currently we have a few errors left. This includes the mace error and other projectile errors. To fix this issue maybe would take another 2 weeks to complete. However, we could also always expand our project by adding more items, enemies, or levels, which could take at least another month to add.

**Appendix A – Agreement Between Customer and Contractor**

When the customer and our team collectively sign off on this document, it will signify a mutual agreement on the specifications, design standards, and requirements outlined within this document. The customer acknowledges the proposed system requirements as described within this document. Simultaneously, our team commits to implementing and delivering a product that aligns with these agreed-upon specifications. This agreement serves as a foundation and understanding guide for the development process, fostering transparency and shared expectations between the customer and the development team.

In the event of future changes or updates to the document, a systematic procedure will be followed to ensure accountability. Any proposed changes should be submitted in writing. This will detail the modifications or additions required. This document will then undergo a review process involving relevant stakeholders from both the customer and the team. Once a consensus is reached, an updated version of the document will be created, and all parties will re-evaluate and confirm their agreement. This structured procedure aims to become an effective communication method, and accommodate the evolving project as needed, and maintain alignment between the customer's expectations and our developmental efforts.

| | |
|---|---|
| Kabeer Alabi | Kabeer Alabi |
| Print Name | Signature |
| CJ Vittek | CJ Vittek |
| Print Name | Signature |
| Nahim Kamruzzaman | Nahim Kamruzzaman |
| Print Name | Signature |
| Tae Hyung Kim | Tae hyung kim |
| Print Name | Signature |
| Collins Ufua | Collins Ufua |
| Print Name | Signature |
| Customer | Customer |
| Print Name | Signature |

**Appendix B – Team Review Sign-off**

After thorough review and discussion, all members of the team have carefully examined the document, and unanimously agreed regarding its content and format. This collaborative team effort has allowed us to incorporate diverse perspectives from one another, ensuring that the document accurately represents our shared understanding of the project requirements and design standards. We are confident that this agreement reflects the collective vision of the team.

| | |
|---|---|
| Kabeer Alabi | Kabeer Alabi |
| Print Name | Signature |
| CJ Vittek | CJ Vittek |
| Print Name | Signature |
| Nahim Kamruzzaman | Nahim Kamruzzaman |
| Print Name | Signature |
| Tae Hyung Kim | Tae hyung kim |
| Print Name | Signature |
| Collins Ufua | Collins Ufua |

## Appendix C – Document Contributions

This is the Document Contributions template from Blackboard. Remember that each team member must contribute to the writing (includes diagrams) for each document produced.

| Document Section | Team Member | Contributions |
|---|---|---|
| 1.1 Purpose of This Document | Collins | Drafted the initial purpose statement, highlighting the document's intent and summarizing its content. |
| 1.2 References | Collins | Compiled and listed all relevant references and sources used in creating the document, following the appropriate citation formats. |
| 1.3 Purpose of the Product | CJ Vittek | Described the purpose of the Boundless Hollow, emphasizing the player's experience and the project's justification. |
| 1.4 Product Scope | Tae Hyung | Outlined the product's scope using a top-level use case diagram, describing primary actors, system functionalities, and external systems. |
| 2. Functional Requirements | Nahim Kamruzzaman | Contributed to the creation of detailed use case descriptions for key functionalities, ensuring clarity and completeness. |
| 3. Non-Functional Requirements | Nahim Kamruzzaman | Formulated non-functional requirements, specifying customer constraints, external interfaces, and other essential non-functional aspects. |
| 4. Deliverables | CJ Vittek | Compiled and detailed the deliverables section, specifying the expected outputs and outcomes of the project. |
| 5. Open Issues | Kabeer Alabi | Identified and documented open issues, highlighting areas |

| | | |
|---|---|---|
| | | of concern or uncertainty that require further discussion and resolution. |