

CMSC 447  
Software Design and Development  
Fall 2023

**System Design Document**

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

**Table of Contents**

	<u>Page</u>
1. Introduction	
1.1 Purpose of This Document	2
1.2 References	2
2. System Architecture	
2.1 Architectural Design	2,3
2.2 Decomposition Description	3,4
3. Persistent Data Design	
3.1 Database Descriptions	4
3.2 File Descriptions	4
4. Requirements Matrix	4,5
Appendix A – Agreement Between Customer and Contractor	6
Appendix B – Peer Review Sign-off	7
Appendix C – Document Contributions	8

## Introduction

### 1.1 Purpose of This Document

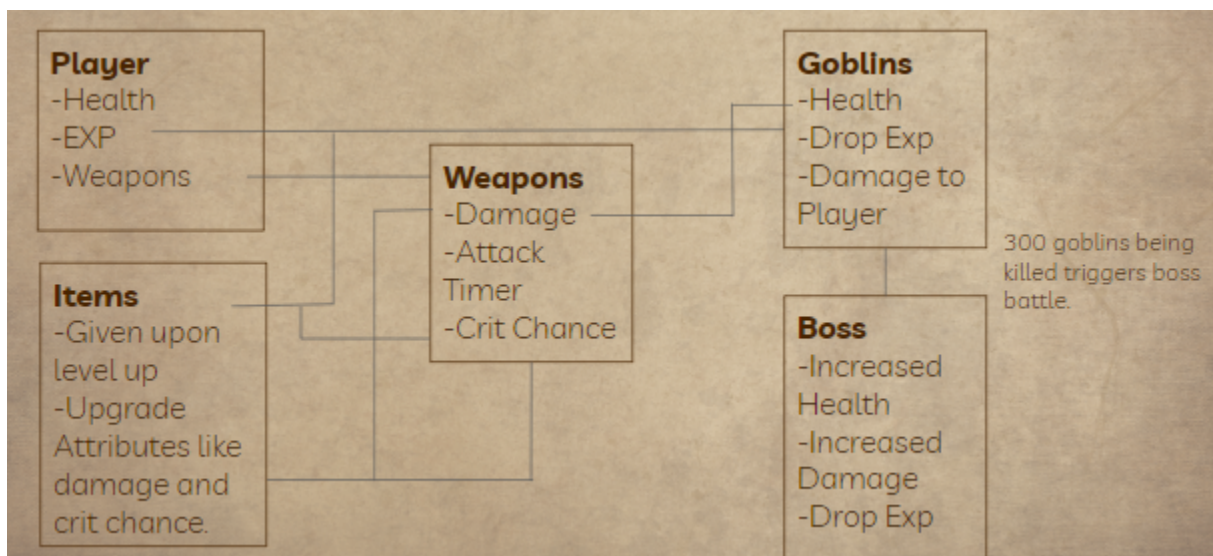
The purpose of this document is to show the design process we followed in the creation of our game The Boundless Hollow. We will go through how each part of the game was designed and how they interact by discussing the system architecture that was used and will be needed, the persistent data design which will go over the needed files for our project and how they are constructed, and finally the requirements matrix which will show how each part of the design fulfills the necessary requirements. This document is intended to be read by those working on the system design that need to know how each system functions and the purpose they fulfill.

### 1.2 References

### System Requirements Specifications

## 2. System Architecture

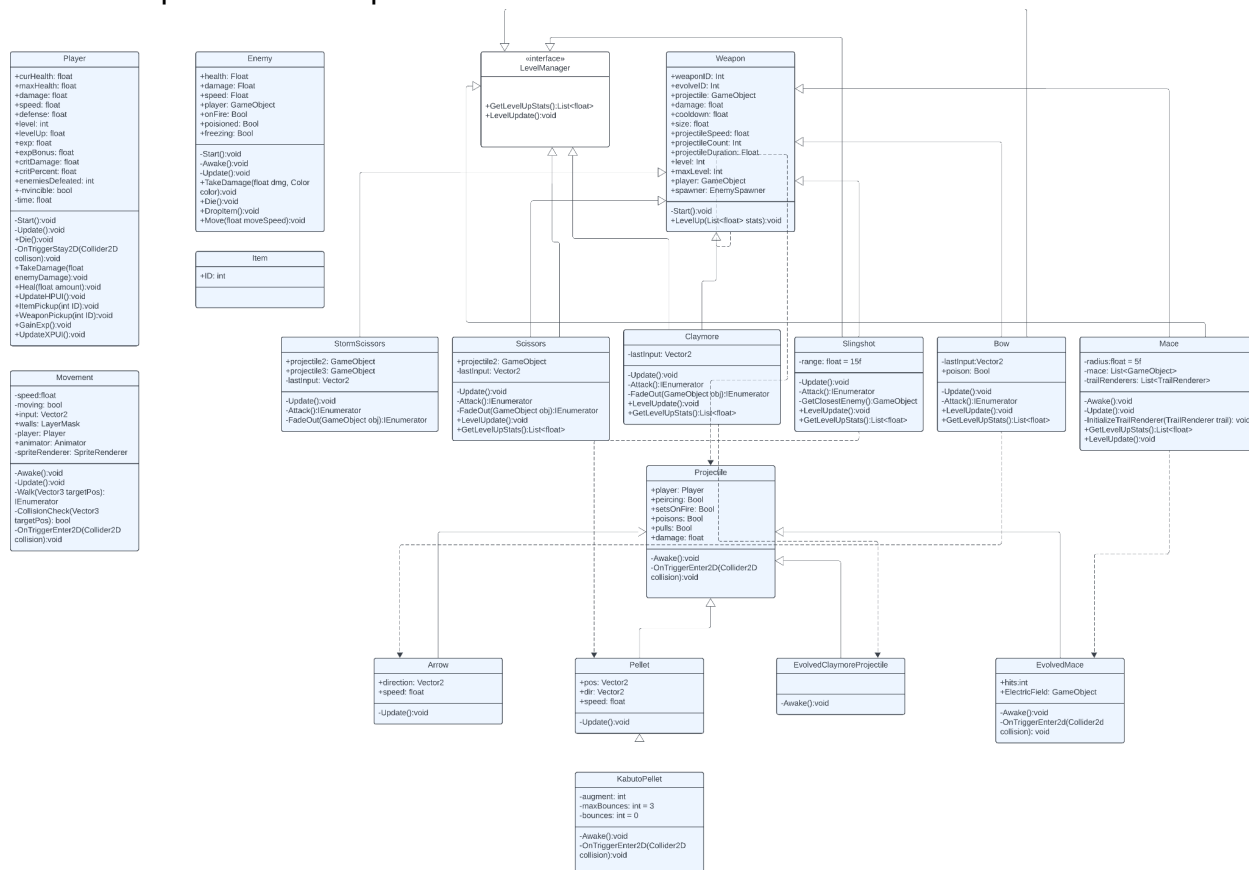
### 2.1 Architectural Design



The game works by having players fight an infinite horde of goblings. The player's only available inputs are for directional movement. All weapons function automatically without player input. There are five weapons with nine levels and each has an evolved form. Along with weapons, there are items the player can collect that increase various statistics like maximum health, speed, damage, critical damage, etc. The player can acquire both items and weapons from leveling up. To level up, the player must kill enemies, which upon death will drop experience crystals for the player to collect, which will level the player up after reaching a certain experience point threshold. Along with

normal enemies, after every three hundred enemies that are killed by the player, a boss enemy will spawn as an increased challenge for the player to face.

## 2.2 Decomposition Description



There are four main systems in play for the game to work: the Player, Weapons, Items, and Enemies. The Player and Movement components are the two most important components for the game to function. The Player object stores all data and interacts with all other objects in the game, from enemies to items to weapons. The Player object also controls taking damage, and interacting with weapon objects. The Movement object is what allows for the player's only form of interacting with the game. The player can only move with either the arrow keys, or with w, a, s, and d keys and all of that is controlled in the Movement object. The Enemy object is what controls enemy AI behaviors. The AI continuously walks towards the player and when it contacts the player's hitbox, it deals damage to the player. Along with this, enemies drop experience point items on the ground for the player to pick up when they die.

Weapon and Projectile objects work hand in hand with one another for weapons to be functional. All weapons function without player input. There are five weapon objects in the game, being the Claymore, Bow, Mace, Slingshot, and Scissors. All of these weapons have evolved forms, after leveling up enough using the LevelManager object, but the only evolved weapon requiring a new object is the Scissors which evolve into the StormScissors object. Every weapon will spawn a Projectile object that has

different functionality based on the weapon object that spawned it, along with any additional functionality found in the inherited projectile objects. The Arrow, Pellet, KabutoPellet, EvolvedClaymoreProjectile, and EvolvedMace objects all provide additional functionality to the projectiles spawned by the Bow, Slingshot, Claymore and Mace objects respectively.

For additional clarity on how each weapon functions, the Claymore object will spawn a stationary slashing projectile in the direction the player is moving and it will fade away over the next second. The Mace object will spawn a persistent projectile(s) that spins around the player with varying speeds based on the weapon's level. The Bow weapon will spawn a volley of arrows in the direction the player is moving and those arrows will travel at a fixed speed until they either hit an enemy or ten seconds have passed. The Slingshot spawns pellets that travel in the direction of the nearest enemy and if there are no enemies alive, it does nothing. Scissors spawn a repeated chain of stationary opened and closed scissor projectiles that all do damage to enemies. Each weapon has nine levels and an evolution that increases the lethality of that weapon.

### 3. Persistent Data Design

#### 3.1 Database Descriptions

No databases are used or required in this product.

#### 3.2 File Descriptions

No external files are used in this product

### 4. Requirements Matrix

Requirements Matrix		
Functional Requirement ID#	Functional Requirement Use Case	System Components
UC-01	Character Controls	Player and Movement Objects
UC-02	Combat System	Weapon and Projectile Systems
UC-03	Character Progression	Player Level Up system
UC-04	Enemy AI and Attack	Enemy System

Requirements Matrix		
Functional Requirement ID#	Functional Requirement Use Case	System Components
UC-01	Character Controls	Player and Movement Objects
UC-02	Combat System	Weapon and Projectile Systems
UC-03	Character Progression	Player Level Up system
UC-05	Creation of 2D Art Assets and Animation	Done through Krita
UC-06	Level Designing	Done in Unity Editor
UC-07	HUD Design	Done in Unity Editor
UC-08	Creating Menus	Done in Unity Editor

## Appendix A – Agreement Between Customer and Contractor

By signing this document both parties agree to all of the designs specified above and will ensure they do what is necessary so that they are all completed and perform up to given specifications. The customer will ensure they do their best to provide any necessary resources that are either deemed necessary or would help to ensure the completion of the product. The team agrees to ensure work is completed in a timely fashion and that resources are used in an efficient manner. Both parties also agree that a reasonable amount of time will be given between a request and having that request fulfilled.

If any changes wish to be made to this document then there must be a meeting between both parties in which all desired changes will be discussed. This discussion will require going over things such as why this change is important and why adding will improve the final product. It will also include the necessary resources needed to ensure the change and be implemented which will include time, money, and labor at the minimum. After a change has been agreed upon it will be added to the document and both parties will sign off on it.

Kabeer Alabi

Print Name

CJ Vittek

Print Name

Nahim Kamruzzaman

Print Name

Tae Hyung Kim

Print Name

Collins Ufua

Print Name

Customer

Print Name

*Kabeer Alabi*

Signature

*CJ Vittek*

Signature

*Nahim Kamruzzaman*

Signature

*Tae hyung kim*

Signature

*Collins Ufua*

Signature

*Customer*

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

Print Name

CJ Vittek

Print Name

Nahim Kamruzzaman

Print Name

Tae Hyung Kim

Print Name

Collins Ufua

*Kabeer Alabi*

Signature

*CJ Vittek*

Signature

*Nahim Kamruzzaman*

Signature

*Tae hyung kim*

Signature

*Collins Ufua*

## **Appendix C – Document Contributions**

CJ Vittek - Section 1.1, section 1.2, Appendix A

Collins Ufua - Section 2.1

Kabeer Alabi - Section 2.2, Appendix B

Nahim Kamruzzaman - Section 3

Tae Hyung Kim - Section 4