[Just Survive Damage Module]

Architecture/Design Document

**Table of Contents**

[**1**](#_heading=h.30j0zll) **Introduction 3**

[**2**](#_heading=h.1fob9te) **Design Goals 4**

[**3**](#_heading=h.3znysh7) **System Behavior 4**

[**4**](#_heading=h.2et92p0) **Logical View 5**

[**4.1**](#_heading=h.tyjcwt) **High-Level Design (Architecture) 5**

[**4.2**](#_heading=h.3dy6vkm) **Mid-Level Design 6**

[**4.3**](#_heading=h.1t3h5sf) **Detailed Class Design 8**

[**5**](#_heading=h.4d34og8) **Process View 11**

[**6**](#_heading=h.3rdcrjn) **Development View 12**

[**7**](#_heading=h.2s8eyo1) **Physical View 12**

[**8**](#_heading=h.17dp8vu) **Use Case View 12**

Change History

**Version:** <1.0>

**Modifier:** <Vanessa>

**Date:** 03/04/2020

**Description of Change:** <Document started and set up>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Version:** <1.1>

**Modifier:** <Vanessa>

**Date:** 03/05/2020

**Description of Change:** <Added UML and interaction diagrams>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Version:** <1.2>

**Modifier:** <Vanessa>

**Date:** 03/05/2020

**Description of Change:** <Updated Physical View>

# Introduction

The purpose of this document is to describe the architecture and design for damage components and health system used in the game Just Survive, developed by Samurai Intellectuals. The damage component is used to deal damage to various actors and characters in game but also receive damage based on who’s dealing it with what gun, ammo, etc., and where they are hit.

# Design Goals

The design goals for this module will contain setting various damage types for each gun and ammo that is being used. As of right now the actors receive damage the same no matter what surface they’re hit, but this may change in the future. We will be implementing different damage types that will affect the strength of your projectiles attack based on enemy and gun types. As of now, each projectile deals equal amounts of damage across the game. The designers will also be responsible for creating UI for the Game Over screen that plays upon reaching zero health.

# System Behavior

The damage system spreads across a couple of different classes. It needs to have the projectile class call functions to deal damage based on what it interacts with. Each class we want to receive damage, such as our enemies, generator, player will have a set HP and a TakeAnyDamage function that will decrease their health upon OnHit being called. If a generator starts to take damage, an onscreen message notifies the player so that they’re aware. If zero HP is reached for a generator or player, a Game Over screen is called and displayed, allowing the player to exit the game.

# Logical View

The logical view describes the main functional components of the system. This includes modules, the static relationships between modules, and their dynamic patterns of interaction.

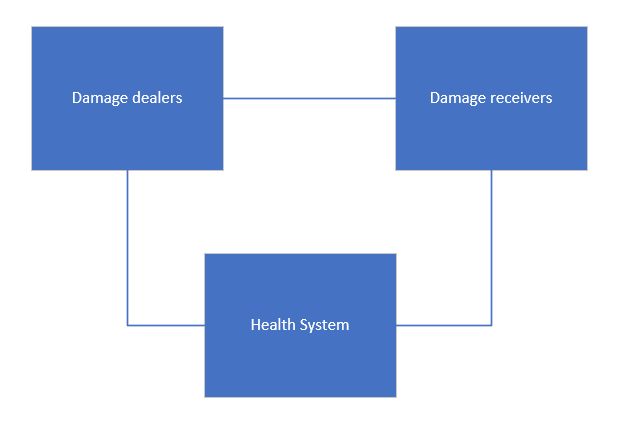
In this section the modules of the system are first expressed in terms of high level components (architecture) and progressively refined into more detailed components and eventually classes with specific attributes and operations.

## High-Level Design (Architecture of the Entire system)

The high-level view or architecture consists of <3> major components:

<Damage Dealt, Damage Received, Health System>

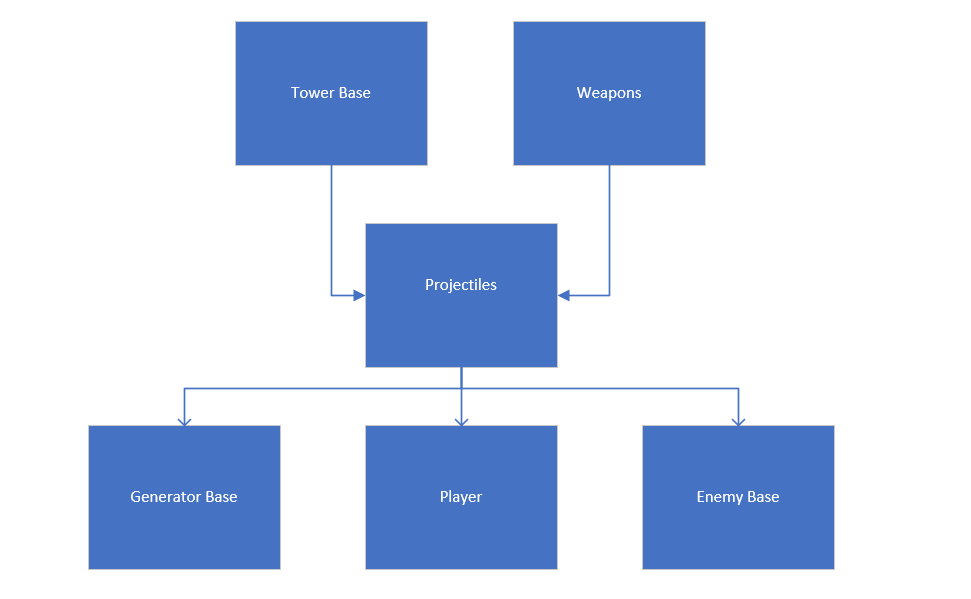
**Example:**

****

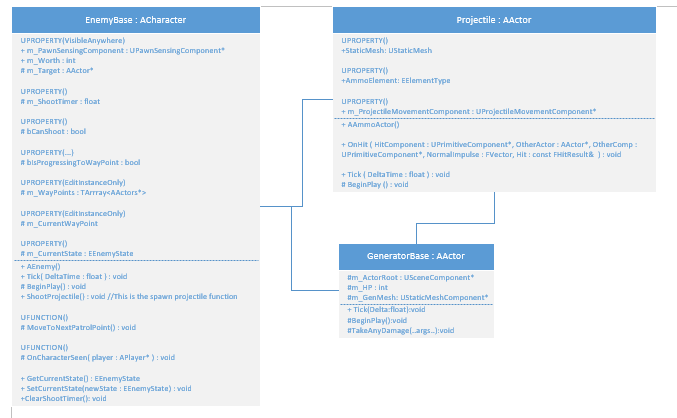
**System Architecture**

* The **Damage dealers** are responsible for setting how much damage is dealt and what they’re interacting with if it should be damaged or not. In our case, our projectile class is responsible for this.
* The **Damage receivers** are actors/characters such as our main player, AI, and the generator (target of the game). Upon being hit with a projectile call a TakeAnyDamage function and decrease their HP accordingly.
* The **Health System** works individually with each damage receiver class. Depending on their role in the game, if their HP reaches zero, a Game Over screen is called.

## Mid-Level Design of Module <Damage/Health Module>

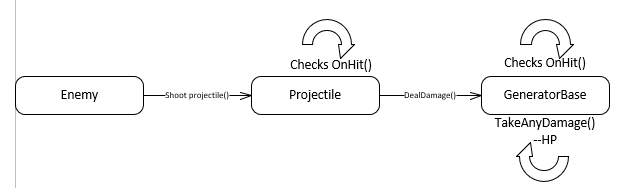
* **** The **Tower Base** and **Weapon Base** both contain and shoot projectiles
* The **Projectiles** are responsible for setting how much damage they deal, and dealing damage accordingly
* The **Generator Base** is the main target of the game and can receive damage from enemy base. Upon reaching zero HP, the game is over.
* The **Player** can take and deal damage using their weapon/ projectiles.
* The **Enemy Base** can take and receive damage as well. Enemies can take damage from towers or from other players but will be focused on finding and dealing damage to the generator.

## Detailed Class Design of Module <Damage/Health Module>



* Projectile directly interacts with both GeneratorBase and EnemyBase.
* EnemyBase interacts with GeneratorBase

# Process View of Module <Damage/Health>



# Physical View (Applies to Multiplayer)

The health variables will have to be networked and updated accordingly as they’ll affect gameplay the most. Health will be displayed as a UI element for the player.

# Use Case View

* Towers and weapons fire projectiles, dealing damage upon OnHit
* Enemies, players, generators receive damage based on who they’re hit by and what damage type is being dealt
  + Upon being hit will decrement their HP
  + Enemies at zero are destroyed
  + Player and generator at HP calls Game Over (for now)
    - Players will eventually want to be respawned to keep playing