Round Manager Module

Architecture/Design Document

**Table of Contents**

[1 Introduction 3](#_Toc131876418)

[2 Design Goals 4](#_Toc131876419)

[3 System Behavior 4](#_Toc131876420)

[4 Logical View 5](#_Toc131876421)

[4.1 High-Level Design (Architecture) 5](#_Toc131876422)

[4.2 Mid-Level Design 6](#_Toc131876424)

[4.3 Detailed Class Design 8](#_Toc131876425)

[5 Process View 11](#_Toc131876426)

[6 Development View 12](#_Toc131876427)

[7 Physical View 12](#_Toc131876428)

[8 Use Case View 12](#_Toc131876429)

Change History

**Version:** <1.0>

**Modifier:** Anthony

**Date:** 2/9/2020

**Description of Change:** File started and created

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

**Version:** <1.1>

**Modifier:** Anthony

**Date:** 2/12/2020

**Description of Change:** Worked on spawning enemies based on Round Manager logic.

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

**Version:** <1.2>

**Modifier:** Anthony

**Date:** 2/13/2020

**Description of Change:** Fixing enemy movement as they spawn

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

**Version:** <1.3>

**Modifier:** Anthony

**Date:** 2/17/2020

**Description of Change:** Making waves scalable and debugging other Round Manager bugs.

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

# Introduction

This document is for the round manager system that was developed for Samurai Intellectuals game Just Survive. The round manager is a way of organizing the flow of enemies, enemy groups and waves during gameplay.

# Design Goals

* The round manager should be expandable based on designers choosing.
* It should be modular so that designers can adjust for game balancing purposes.
* The design should be organized based on a collection of blueprints representing each type of enemy group and wave.
* The round manager should act as an interface that exists directly in the level.

# System Behavior

**Round Manager** contains a list of waves, a spawn location and a timer to spawn the next wave. The round manager begins the process of spawning enemies on startup. It keeps track of how many active enemies exist in the current wave. The round manager checks the destroyed enemy count against the max number of enemies in the current wave (sum of all enemies in each enemy group of the wave) to spawn the next wave upon completion of the current one.

**Wave** contains a list of enemy groups. The wave calls the spawn all enemies function on each enemy group.

**Enemy Group** contains a list of enemies. It uses timer to spawn each of the enemies it is responsible for. Calculates all current active enemies to report back to round manager.

**Enemy** has a reference to the round manager and notifies it when it is destroyed.

Each enemy contains a reference to the round manager and notifies it when they are destroyed.

# Logical View

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

The high-level view or architecture consists of 4 major components:

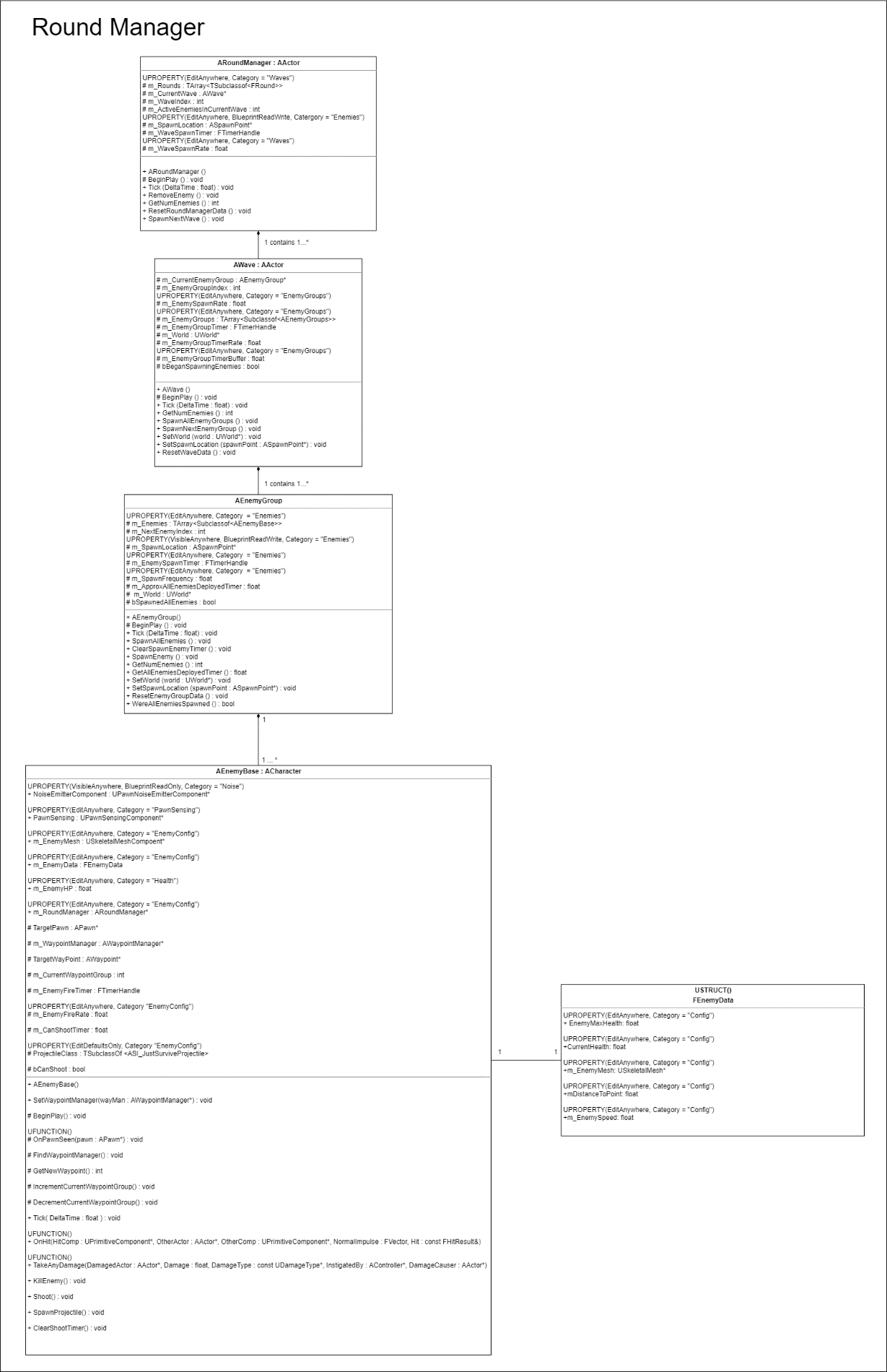
Wave

Round Manager

Enemy

EnemyGroup

## Detailed Class Design of Module <Round Manager>



# Process View of Module Round Manager

**5.1 Enemy is Destroyed and Communicates to Round Manager to spawn next Wave**

AWave

ARoundManager

AEnemyGroup

AEnemyBase

RemoveEnemy()

If Active Enemies == 0

Increment Wave Index

SpawnNextWave ()

SpawnAllEnemyGroups ()

SpawnNextEnemyGroup ()

SpawnAllEnemies ()

SpawnNextEnemy ()

# Physical View (Applies to Multiplayer)

All round manager functions will be performed server side.

# Use Case View

* **At Game Start**

The Round Manager calculates the total number of active enemies that will be spawned during this wave and spawns all enemy groups of the current wave. Wave uses a timer to spawn all of the enemy groups it is responsible for and spawns all enemies of the current active enemy group. The enemy group uses its timer to spawn all enemies it is responsible for. When an enemy is destroyed it reports back to the Round manager to reduce the number of active enemies.