Inventory Component 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:** 2/3/2020

**Description of Change:** File started and created. Created weapon data and implement entire inventory component.

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

**Version:** <1.1>

**Modifier:** Anthony

**Date:** 2/3/2020

**Description of Change:** Made projectiles spawn from player by adding a socket to the weapon and changing the weapon from a skeletal mesh to a static mesh.

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

**Version:** <1.2>

**Modifier:** Anthony

**Date:** 2/8/2020

**Description of Change:** Fixing bugs and displayed weapon data in UI.

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

# Introduction

This document is for the inventory component that was developed for Samurai Intellectuals game Just Survive.

The purpose of this document is to describe the inventory component system for all major stakeholders, answering any inquiries that may arise. The stakeholders include:

* Game Designers
* Game Programmers

# Design Goals

* The inventory component should simplify the interaction between the player and items in the world.
* It should organize the weapons that the player chooses to equip and carry in a simple way and display weapons and weapon related resources in the UI based on a weapon data structure.

# System Behavior

**Inventory Component** contains a list of weapons received by the player as they interact with weapons in game.

**Weapon Base** uses a weapon data struct set in editor. The weapon data struct determines things like fire rate, clip size, current ammo, weapon damage etc. The UI uses the weapon data struct to determine what information to display in the HUD

**Player** contains a reference to inventory component. Adds each weapon it picks up to the inventory component. Refers back to active weapon in inventory to check if player can shoot, reload etc.

# Logical View

## High-Level Design (Architecture of the Inventory System)

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

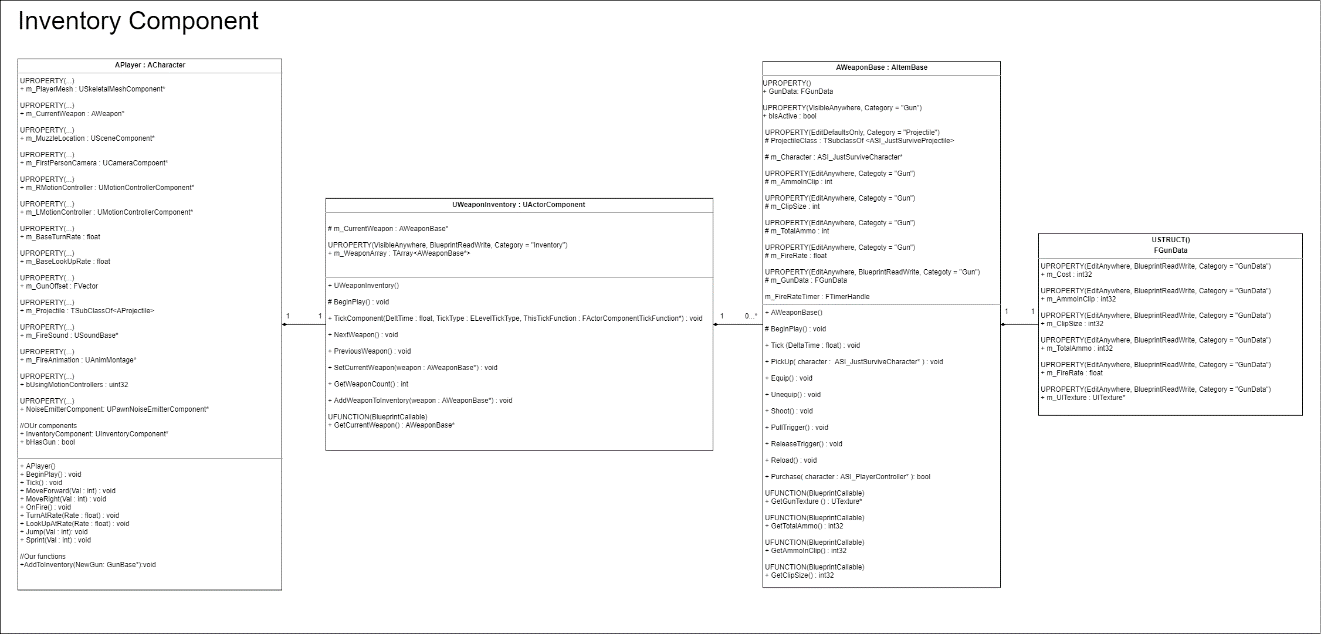
InventoryComponent

Player

WeaponBase

UGunData

## Detailed Class Design of Module <Inventory Component>



# Process View of Module Inventory Component

**5.1 Player Shooting weapon**

AWeaponBase

InventoryComponent

ASI\_JustSurviveCharacter

ASI\_PlayerController

OnFire()

GetCurrentWeapon()

ReturnCurrentWeapon

PullTrigger()

Check if Gun has enough ammo

Shoot()

**5.2 HUD Displaying Gun Data**

ASI\_JustSurviveHUD

AWeaponBase

InventoryComponent

ASI\_JustSurviveCharacter

UWorld

GetPlayer()

GetInventoryComponent()

GetCurrentWeapon()

Return GunDataStruct

GetGunData ()

# Physical View (Applies to Multiplayer)

The Inventory Component will be replicated to each player from the server.

# Use Case View

* **Player Interacts with Weapon in world and adds it to Inventory**

Player raytraces to weapon and presses “E” to add it to his or her inventory. The weapon is added to an array found in inventory component which each player has a reference to.

* **Player Shoots currently equipped weapon in inventory**

Player clicks mouse. Shoot function is called on player controller which calls OnFire() on the active player. Player retrieves the current weapon from inventory component and calls the PullTrigger(). The current weapon pointer has the address of the weapon the player is holding and it uses a timer and the fire rate set in its Gun Data struct to determine how fast it should spawn bullets.