

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**DETAILED DESIGN SPECIFICATION
CSE 4317: SENIOR DESIGN II
FALL 2022**



**EUPORIE STUDIOS
PROJECT CALAMITY**

**MATTHEW GREEN
ROBERT KEMBEL
JACOB LANHAM
BRADLEY MIETTINEN**

REVISION HISTORY

| Revision | Date | Author(s) | Description |
|----------|------------|----------------|-----------------------|
| 0.1 | 9.30.2022 | JL | document creation |
| 1.0 | 10.21.2022 | MG, JL, RK, BM | document finalization |

CONTENTS

| | | |
|----------|---|-----------|
| 1 | Introduction | 6 |
| 2 | System Overview | 6 |
| 3 | Player Controller Subsystems | 7 |
| 3.1 | Layer Operating System | 7 |
| 3.2 | Layer Software Dependencies | 7 |
| 3.3 | Input System | 7 |
| 3.4 | Looking Handler | 9 |
| 3.5 | Movement Handler | 10 |
| 4 | Inventory Subsystems | 11 |
| 4.1 | Layer Operating System | 11 |
| 4.2 | Layer Software Dependencies | 11 |
| 4.3 | Inventory Handler | 12 |
| 4.4 | Item Handler | 13 |
| 5 | Combat Layer Subsystems | 14 |
| 5.1 | Layer Operating System | 14 |
| 5.2 | Layer Software Dependencies | 14 |
| 5.3 | Weapon Controller | 14 |
| 5.4 | Enemy Controller | 15 |
| 5.5 | Health System | 16 |
| 6 | Quest Layer Subsystems | 17 |
| 6.1 | Layer Operating System | 17 |
| 6.2 | Layer Software Dependencies | 17 |
| 6.3 | Quest Controller | 18 |
| 6.4 | Quest Inventory System | 20 |
| 6.5 | Quest Item | 22 |
| 7 | Environment Layer Subsystems | 24 |
| 7.1 | Layer Operating System | 24 |
| 7.2 | Layer Software Dependencies | 24 |
| 7.3 | Gravity | 25 |
| 7.4 | Landscape | 26 |
| 7.5 | Structures | 27 |
| 7.6 | Lighting | 28 |
| 7.7 | Flora/Fauna | 29 |
| 8 | Player Management Systems Subsystems | 30 |
| 8.1 | Layer Operating System | 30 |
| 8.2 | Layer Software Dependencies | 30 |
| 8.3 | Base PMS | 30 |
| 8.4 | Alpha PMS | 31 |
| 8.5 | Beta PMS | 32 |
| 8.6 | Suit Module | 33 |

| | | |
|----------|--|-----------|
| 8.7 | Weapon Upgrade Module | 34 |
| 8.8 | Crafting Module | 35 |
| 8.9 | Quests Module | 36 |
| 8.10 | Stash Module | 37 |
| 8.11 | Transportation Module | 38 |
| 8.12 | Market Module PMS | 39 |
| 8.13 | Memory Module | 40 |
| 9 | UI Layer Subsystems | 41 |
| 9.1 | UI Layer Operating System | 41 |
| 9.2 | UI Layer Software Dependencies | 41 |
| 9.3 | UI Controller | 41 |
| 9.4 | Quest UI | 43 |
| 9.5 | Player UI | 44 |
| 9.6 | Inventory UI | 45 |

LIST OF FIGURES

| | | |
|----|--|----|
| 1 | System Architecture | 7 |
| 2 | Player Controller: Input System Subsystem | 7 |
| 3 | Player Controller: Looking Handler Subsystem | 9 |
| 4 | Player Controller: Movement Handler Subsystem | 10 |
| 5 | Inventory: Inventory Handler Subsystem | 12 |
| 6 | Inventory: Item Handler Subsystem | 13 |
| 7 | Weapon Controller Subsystem | 14 |
| 8 | Enemy Controller Subsystem | 15 |
| 9 | Health Subsystem | 16 |
| 10 | Quest System: Quest Controller Subsystem | 18 |
| 11 | Quest System: Quest Inventory System Subsystem | 20 |
| 12 | Quest System: Quest Item Subsystem | 22 |
| 13 | Environment System: Gravity Subsystem | 25 |
| 14 | Environment System: Landscape Subsystem | 26 |
| 15 | Environment System: Structures Subsystem | 27 |
| 16 | Environment System: Lighting Subsystem | 28 |
| 17 | Environment System: Flora/Fauna Subsystem | 29 |
| 18 | Player Management Station: Base PMS Subsystem | 30 |
| 19 | Player Management Station: Alpha PMS Subsystem | 31 |
| 20 | Player Management Station: Beta PMS Subsystem | 32 |
| 21 | Player Management Station: Suit Module Subsystem | 33 |
| 22 | Player Management Station: Weapon Upgrade Module Subsystem | 34 |
| 23 | Player Management Station: Crafting Module Subsystem | 35 |
| 24 | Player Management Station: Quests Module Subsystem | 36 |
| 25 | Player Management Station: Stash Module Subsystem | 37 |
| 26 | Player Management Station: Transportation Module Subsystem | 38 |
| 27 | Player Management Station: Market Module Subsystem | 39 |
| 28 | Player Management Station: Memory Module Subsystem | 40 |
| 29 | UI Controller Subsystem Diagram | 41 |
| 30 | Quest UI Subsystem Diagram | 43 |
| 31 | Player UI Subsystem Diagram | 44 |
| 32 | Inventory UI Subsystem Diagram | 45 |

LIST OF TABLES

1 INTRODUCTION

This section provides the reader with an overview of Project Calamity. The basic components which make up the game are explained here. Project Calamity is a PC video game which takes input from the player via mouse and keyboard which in turn interacts with the game software to control the Player Character. Next there are a series of game components the player will interact with which make up the bulk of the functionality of the game.

The Environment includes all non-character assets which make up the world. Within the environment, the player will find various non-player character (NPC) inhabitants. These include enemies, allies, and bystanders such as shopkeepers which are used to add life to the environment. As the player explores the world, they will require a means of gathering resources - such as equipment - to face the challenges ahead. This will be done via a Quest System - at fixed points in the game, the player can choose to complete quests which will provide a reward of money, gear, or some other kind of item. The Quest System also tracks which quests are unfinished, in progress, or complete. The Inventory System will be implemented to allow both players and NPC's to carry and access a fixed number of items. The player can use this to equip and remove clothes, weapons, money, etc. as well as trade items. The Combat System controls how players and enemies fight within the game as well as how assets respond to damage. The User Interface is what the Player sees on screen as the Player Character interacts with each of the various game components. This includes everything from menus and dialogue boxes to health bars and navigation aides. Finally, Player Management Stations (PMS) are where many of the various layers and subsystems come together to be managed. The PMS are a set of terminals placed throughout the game which allow the player access to menus for managing quests, buying items, upgrading suits and weapons, saving the game, and much more.

2 SYSTEM OVERVIEW

The system consists of seven interacting layers. The Player Controller takes the input from the mouse and keyboard and implements or disperses those inputs to the relevant systems. The combat system controls all behavior related to combat between the Player, enemy characters and other damageable objects.

The Player Management system is the back-end of all the menus of the game which interface with other subsystems such as the Quest System for choosing quests and Inventory System for accessing items. The Environment controls all the visible characteristics of the game outside of the characters and UI. The UI controls all the visual elements superimposed onto the screen such as health, ammo and menus.

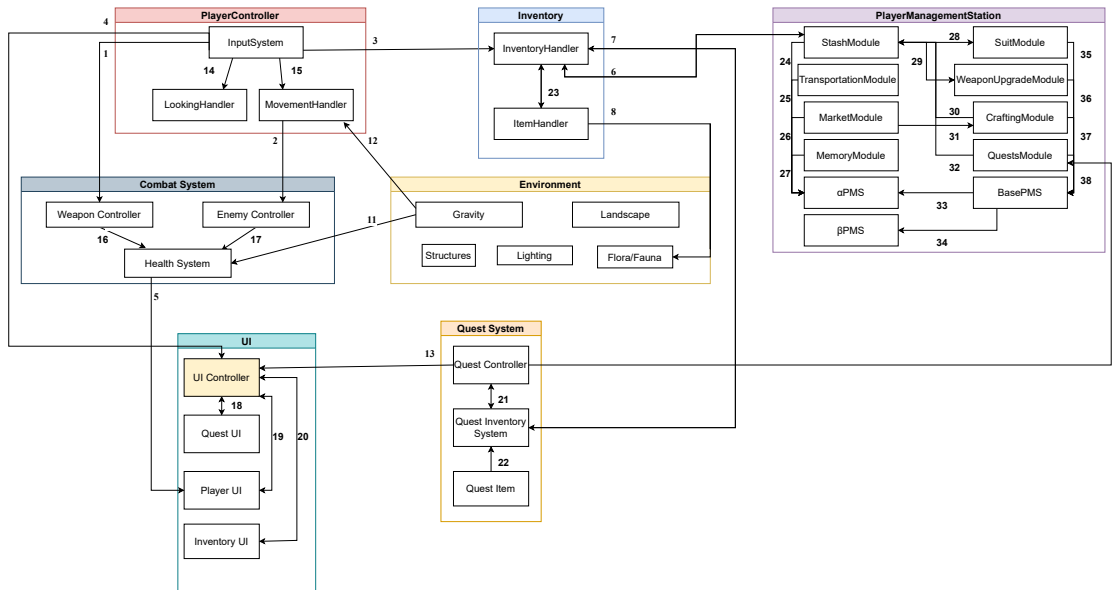


Figure 1: System Architecture

3 PLAYER CONTROLLER SUBSYSTEMS

The Player Controller handles all the inputs from the user and converts them to in-game actions such as moving, looking, and toggling menus. It also handles gravity and inertia acting on the player.

3.1 LAYER OPERATING SYSTEM

Windows 10 or 11.

3.2 LAYER SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.InputSystem, UnityEngine.EventSystem.

3.3 INPUT SYSTEM

This subsystem gets the input from the user's keyboard and mouse and passes the data to various other systems depending on what the user is attempting to control.

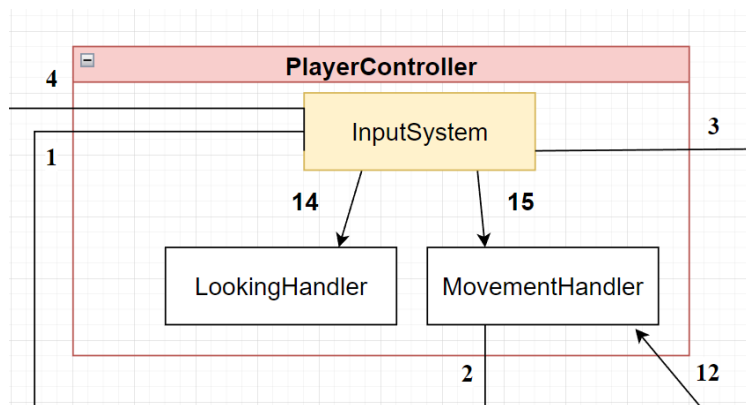


Figure 2: Player Controller: Input System Subsystem

3.3.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

3.3.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.InputSystem, UnityEngine.EventSystem.

3.3.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

3.3.4 SUBSYSTEM DATA STRUCTURES

N/A.

3.3.5 SUBSYSTEM DATA PROCESSING

N/A.

3.4 LOOKING HANDLER

Manages the player looking around his environment.

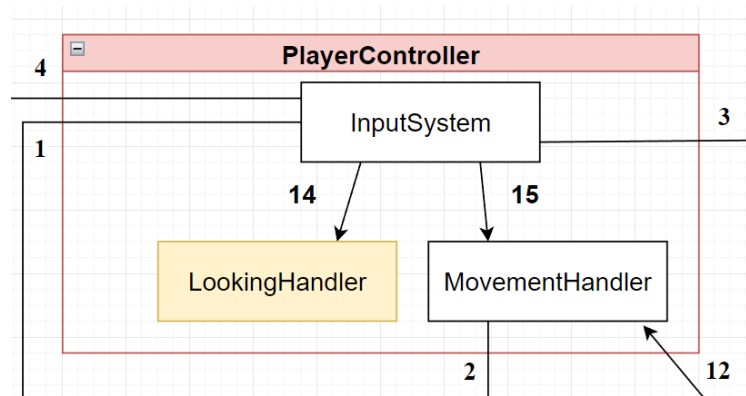


Figure 3: Player Controller: Looking Handler Subsystem

3.4.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

3.4.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.InputSystem.

3.4.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

3.4.4 SUBSYSTEM DATA STRUCTURES

N/A.

3.4.5 SUBSYSTEM DATA PROCESSING

N/A.

3.5 MOVEMENT HANDLER

Manages the player moving around his environment as well as handling gravitational acceleration and inertia.

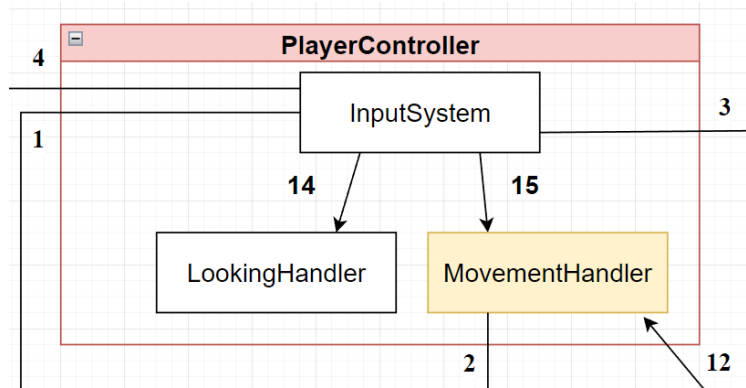


Figure 4: Player Controller: Movement Handler Subsystem

3.5.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

3.5.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.InputSystem.

3.5.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

3.5.4 SUBSYSTEM DATA STRUCTURES

N/A.

3.5.5 SUBSYSTEM DATA PROCESSING

N/A.

4 INVENTORY SUBSYSTEMS

Holds all the data on what the player has equipped and what is contained within said equipment. Also manages moving around items between and inside of equipment slots and containers.

4.1 LAYER OPERATING SYSTEM

Windows 10 or 11.

4.2 LAYER SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI, UnityEngine.EventSystem.

4.3 INVENTORY HANDLER

Holds all the data on what the player has equipped and what is contained within said equipment.

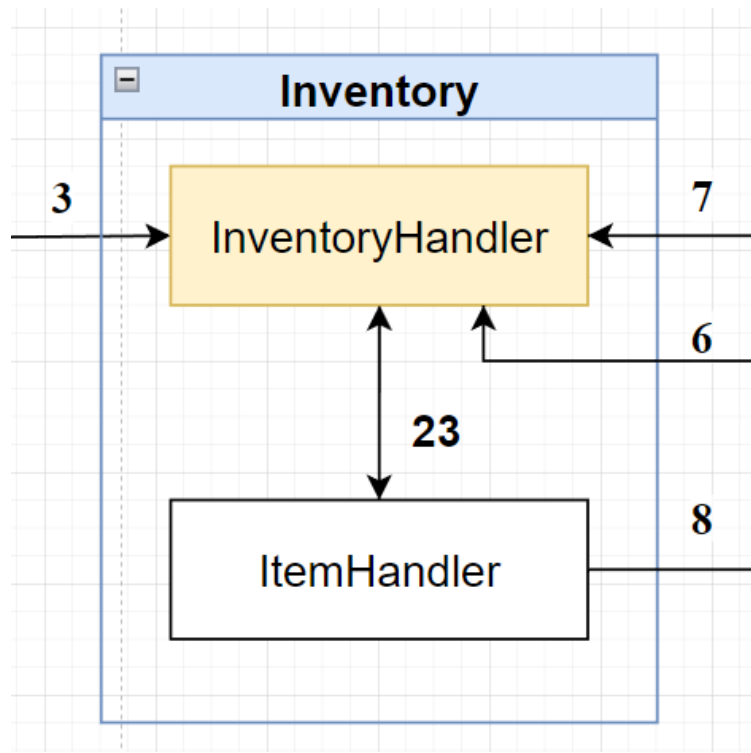


Figure 5: Inventory: Inventory Handler Subsystem

4.3.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

4.3.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.EventSystem.

4.3.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

4.3.4 SUBSYSTEM DATA STRUCTURES

A class called ContainerSO that inherits from UnityEngine.ScriptableObject storing data about a container. A Container class that has a ContainerSO field and performs all general logic and graphics logic using the data associated with that field. A class called Inventory that contains multiple fields for Containers.

4.3.5 SUBSYSTEM DATA PROCESSING

N/A

4.4 ITEM HANDLER

Manages moving around items between and inside of equipment slots and containers.

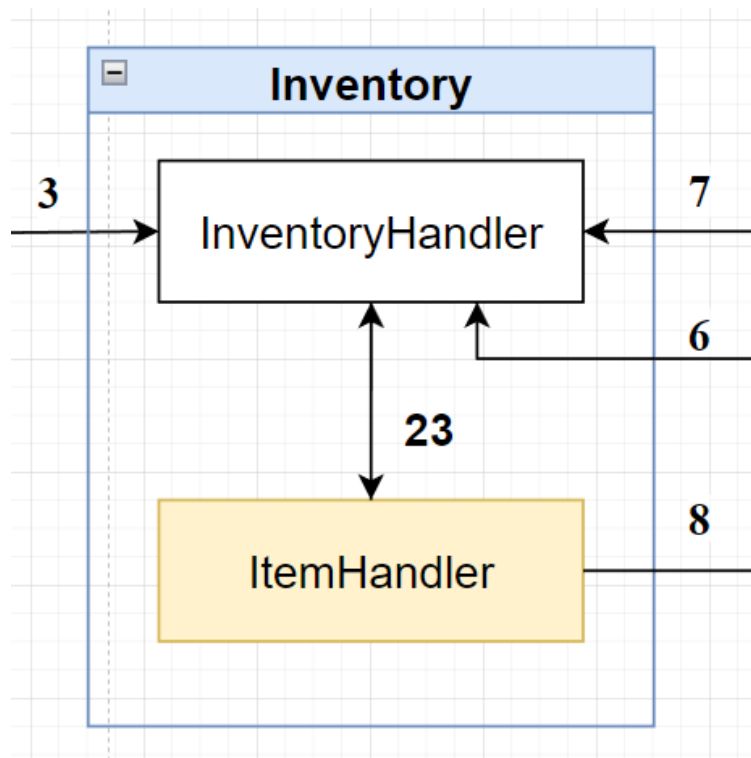


Figure 6: Inventory: Item Handler Subsystem

4.4.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

4.4.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI.

4.4.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

4.4.4 SUBSYSTEM DATA STRUCTURES

A class called ItemSO that inherits from UnityEngine.ScriptableObject that stores all data about an item. Then an Item class that contains a ItemSO field and performs all general logic and graphics logic using the data associated with that field.

4.4.5 SUBSYSTEM DATA PROCESSING

N/A

5 COMBAT LAYER SUBSYSTEMS

This is a pure software layer which manages the weapon, enemy and health behavior throughout the game. Any objects which utilize such behavior will instance and version of one of the following subsystems.

5.1 LAYER OPERATING SYSTEM

Windows 10 or 11.

5.2 LAYER SOFTWARE DEPENDENCIES

This layer uses Unity Engine to interface with the development engine, Unity Input System to access control inputs, Cinemachine to assist with camera control, and the Animation Rigging package to adjust rigging details like gun placement and inverse kinematics.

5.3 WEAPON CONTROLLER

Dicates the behavior of weapons in-game.

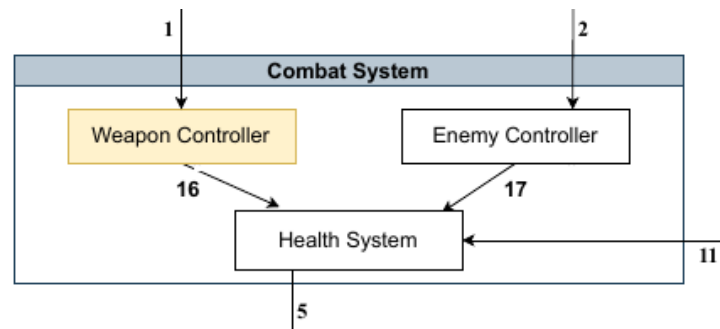


Figure 7: Weapon Controller Subsystem

5.3.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

5.3.2 SUBSYSTEM SOFTWARE DEPENDENCIES

The subsystem uses Cinemachine to alter the camera views while aiming the weapon. The Animation Rigging package was used to make modifications to the gun's placement on the Player Character and to implement reverse kinematics between the Player's hands and the gun. Unity Engine framework was used to interface with the game engine.

5.3.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

5.3.4 SUBSYSTEM DATA STRUCTURES

Instances of a Gun class implement each gun's behavior and how it interacts with its target. A Target class is used to implement the behavior of the target of a gun's projectile.

5.3.5 SUBSYSTEM DATA PROCESSING

N/A

5.4 ENEMY CONTROLLER

Dictates the behavior of enemies both in and out of combat.

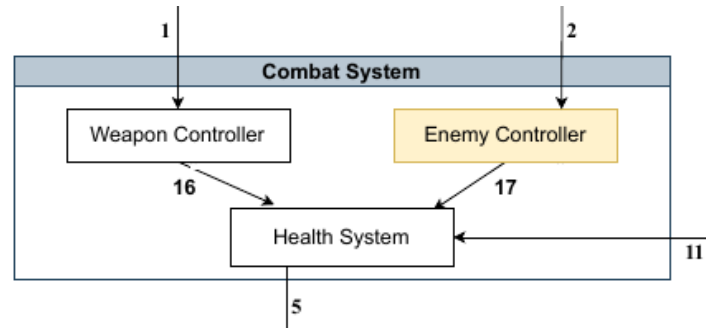


Figure 8: Enemy Controller Subsystem

5.4.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

5.4.2 SUBSYSTEM SOFTWARE DEPENDENCIES

The system uses Animation Rigging to alter the animation details of the character. The Unity Engine library is used to interface with the Engine.

5.4.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

5.4.4 SUBSYSTEM DATA STRUCTURES

The Enemy class controls the enemies behavior when it is in the presence of the Player and otherwise.

5.4.5 SUBSYSTEM DATA PROCESSING

N/A

5.5 HEALTH SYSTEM

Determines how health is managed for all objects.

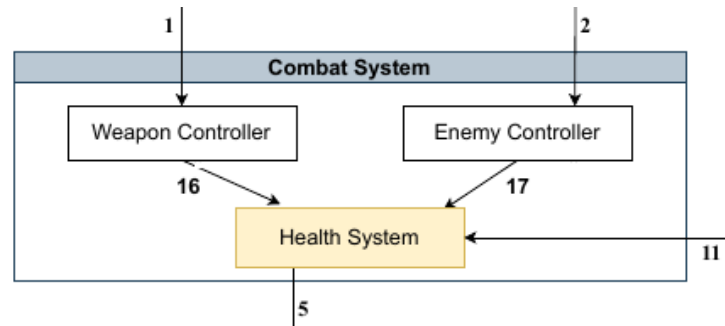


Figure 9: Health Subsystem

5.5.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

5.5.2 SUBSYSTEM SOFTWARE DEPENDENCIES

The Unity Engine library is used to interface this system with the game engine.

5.5.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

5.5.4 SUBSYSTEM DATA STRUCTURES

The Health class controls the health system for each damagable object in the game including enemies and the Player Character.

5.5.5 SUBSYSTEM DATA PROCESSING

N/A

6 QUEST LAYER SUBSYSTEMS

Handles the giving, receiving, tracking, removing of quests and distribution of rewards of completed quests and any progress-tracked activity.

6.1 LAYER OPERATING SYSTEM

Windows 10 or Windows 11.

6.2 LAYER SOFTWARE DEPENDENCIES

UnityEngine

6.3 QUEST CONTROLLER

Manages individual quest items, distribution of the quest rewards, handing out full quests, deleting quests from the quest inventory, adding quests to the quest inventory system, and forwarding current quests to the UI system.

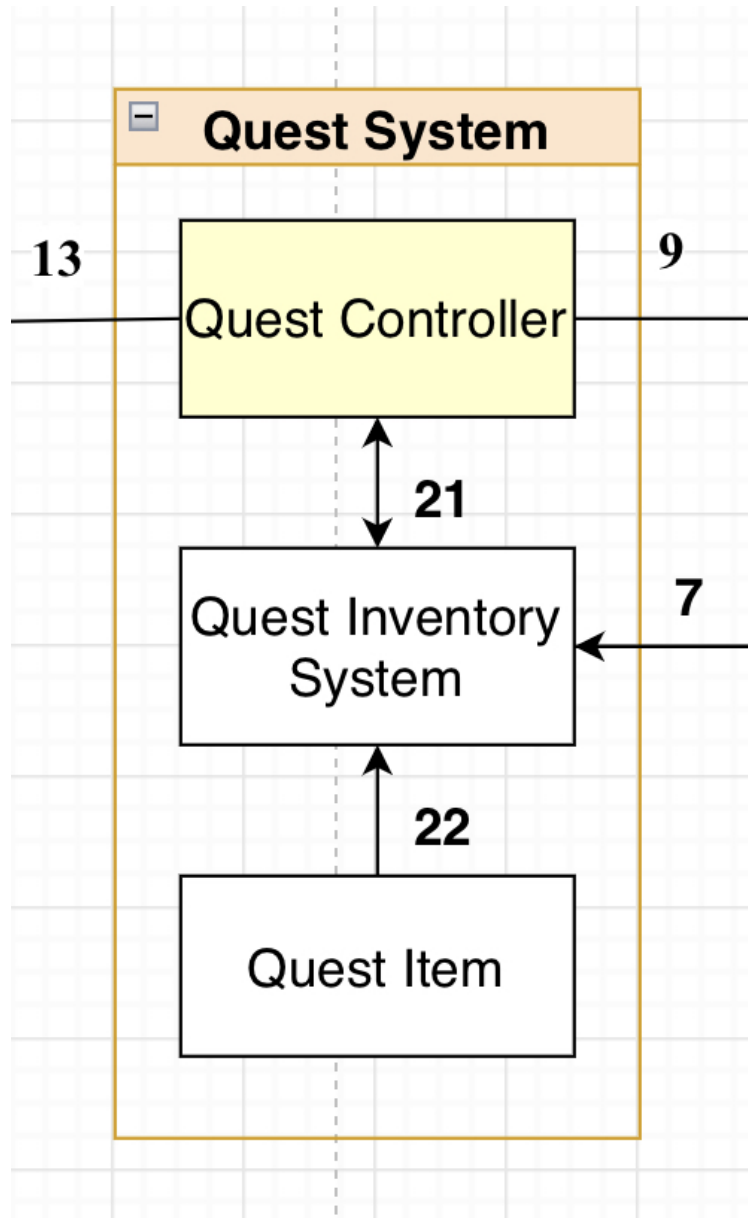


Figure 10: Quest System: Quest Controller Subsystem

6.3.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or Windows 11.

6.3.2 SUBSYSTEM SOFTWARE DEPENDENCIES

N/A

6.3.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

6.3.4 SUBSYSTEM DATA STRUCTURES

N/A

6.3.5 SUBSYSTEM DATA PROCESSING

N/A

6.4 QUEST INVENTORY SYSTEM

Holds quest items in groups to constitute a quest and forwards status of quests to the Controller.

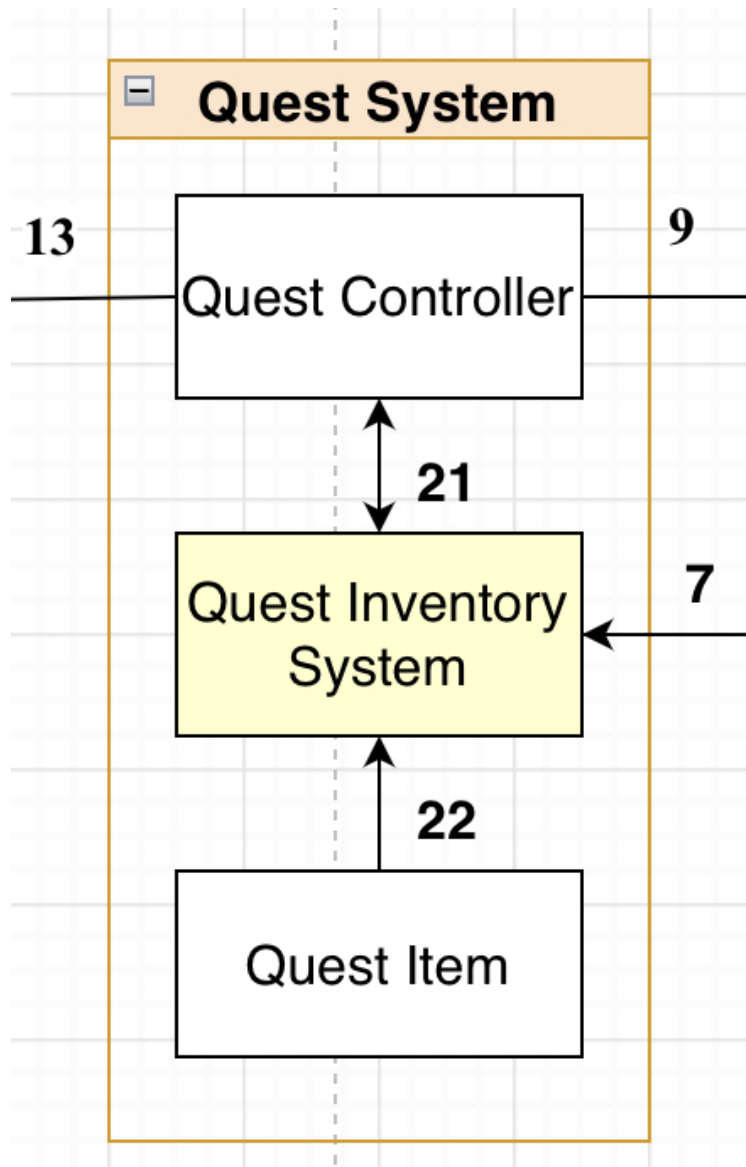


Figure 11: Quest System: Quest Inventory System Subsystem

6.4.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or Windows 11.

6.4.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine

6.4.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

6.4.4 SUBSYSTEM DATA STRUCTURES

N/A

6.4.5 SUBSYSTEM DATA PROCESSING

N/A

6.5 QUEST ITEM

Individual items that constitute quest steps.

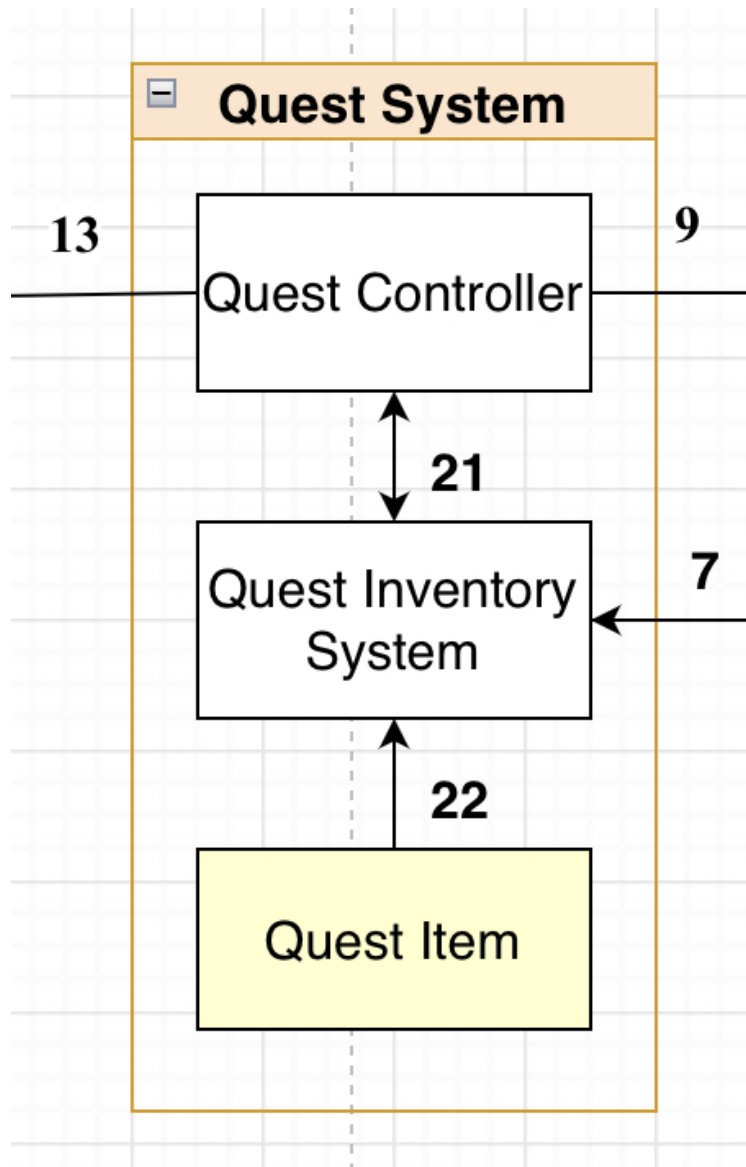


Figure 12: Quest System: Quest Item Subsystem

6.5.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or Windows 11.

6.5.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine

6.5.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

6.5.4 SUBSYSTEM DATA STRUCTURES

A simple structure containing the elements that make up a quest.

6.5.5 SUBSYSTEM DATA PROCESSING

N/A

7 ENVIRONMENT LAYER SUBSYSTEMS

Handles the lighting, gravity, wildlife, and any other items involved the environment.

7.1 LAYER OPERATING SYSTEM

Windows 10 or Widows 11.

7.2 LAYER SOFTWARE DEPENDENCIES

UnityEngine, World-Creator

7.3 GRAVITY

Applies gravity to objects interacting with the environment.

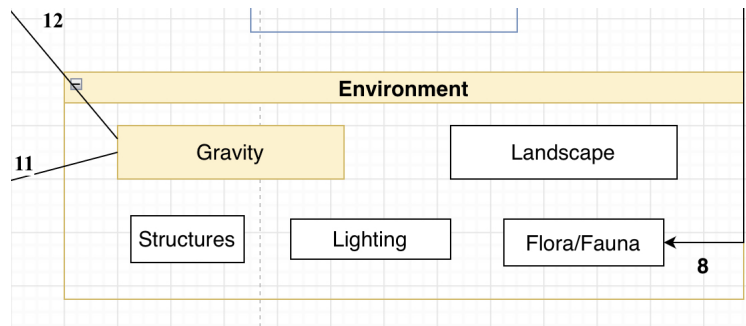


Figure 13: Environment System: Gravity Subsystem

7.3.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or Windows 11.

7.3.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine

7.3.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

7.3.4 SUBSYSTEM DATA STRUCTURES

N/A

7.3.5 SUBSYSTEM DATA PROCESSING

N/A

7.4 LANDSCAPE

Covers the natural or modified landscape of the environment.

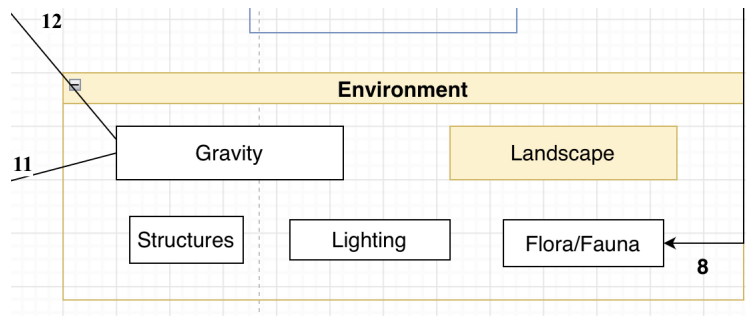


Figure 14: Environment System: Landscape Subsystem

7.4.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or Windows 11.

7.4.2 SUBSYSTEM SOFTWARE DEPENDENCIES

World-Creator, UnityEngine

7.4.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

7.4.4 SUBSYSTEM DATA STRUCTURES

N/A

7.4.5 SUBSYSTEM DATA PROCESSING

N/A

7.5 STRUCTURES

Any buildings, equipment, and things that are not natural or covered by another category are considered structures.

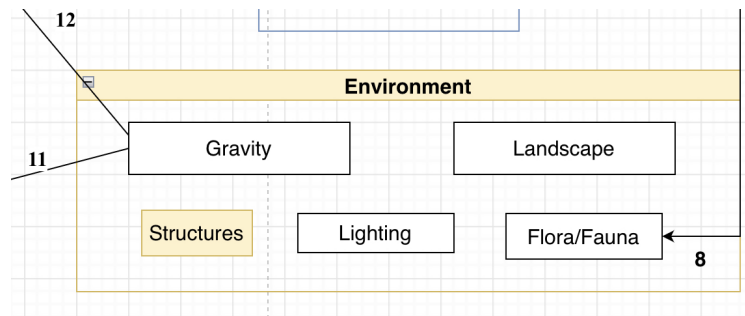


Figure 15: Environment System: Structures Subsystem

7.5.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or Windows 11.

7.5.2 SUBSYSTEM SOFTWARE DEPENDENCIES

N/A

7.5.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

7.5.4 SUBSYSTEM DATA STRUCTURES

N/A

7.5.5 SUBSYSTEM DATA PROCESSING

N/A

7.6 LIGHTING

General lighting, shadows, and reflections.

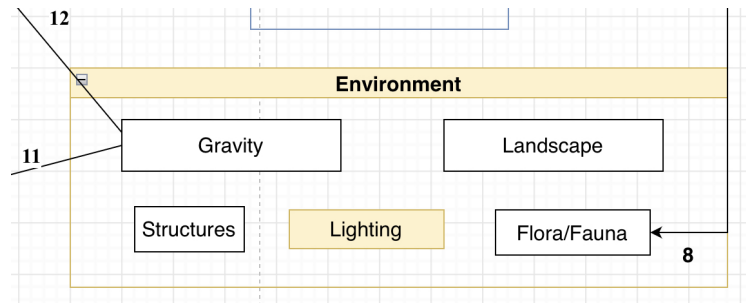


Figure 16: Environment System: Lighting Subsystem

7.6.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or Windows 11

7.6.2 SUBSYSTEM SOFTWARE DEPENDENCIES

N/A

7.6.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

7.6.4 SUBSYSTEM DATA STRUCTURES

N/A

7.6.5 SUBSYSTEM DATA PROCESSING

N/A

7.7 FLORA/FAUNA

Flowers, animals, and the like that can be collected for various purposes.

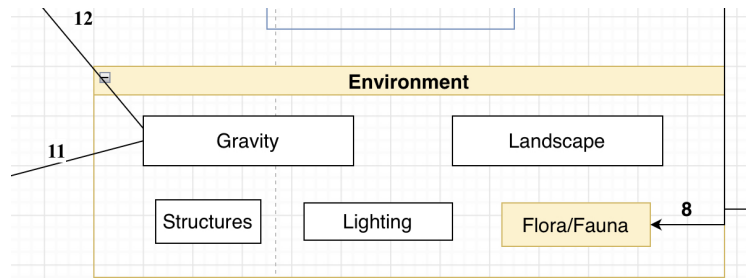


Figure 17: Environment System: Flora/Fauna Subsystem

7.7.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or Windows 11.

7.7.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine.

7.7.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

7.7.4 SUBSYSTEM DATA STRUCTURES

N/A

7.7.5 SUBSYSTEM DATA PROCESSING

N/A

8 PLAYER MANAGEMENT SYSTEMS SUBSYSTEMS

There are two different types of PMS, an alpha and beta version. Alpha PMS have access to all modules and functionality while the beta has less modules and reduced functionality. These stations allow the player to upgrade their suits and weapons. It also allows them to purchase recipes for crafting. Stash items.

8.1 LAYER OPERATING SYSTEM

Windows 10 or 11.

8.2 LAYER SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI, UnityEngine.EventSystem, UnityEngine.InputSystem.

8.3 BASE PMS

Defines the basic functionality of all Player Management Systems.

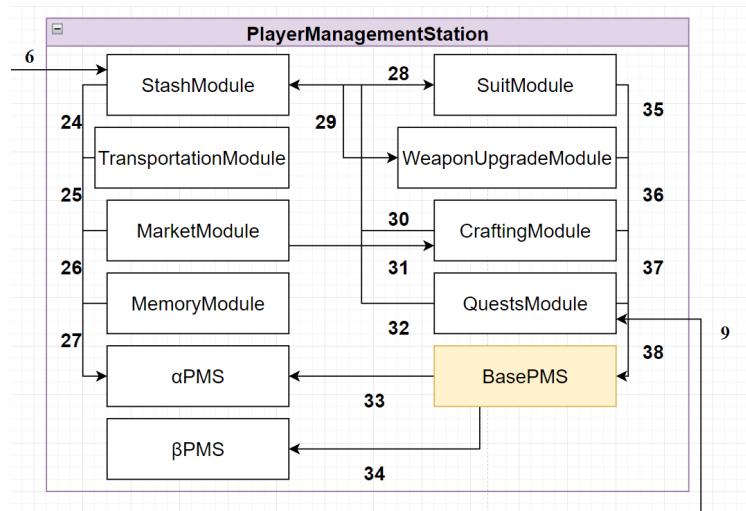


Figure 18: Player Management Station: Base PMS Subsystem

8.3.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

8.3.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI, UnityEngine.EventSystem, UnityEngine.InputSystem.

8.3.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

8.3.4 SUBSYSTEM DATA STRUCTURES

N/A.

8.3.5 SUBSYSTEM DATA PROCESSING

N/A.

8.4 ALPHA PMS

The major type of PMS that allows you to access all modules.

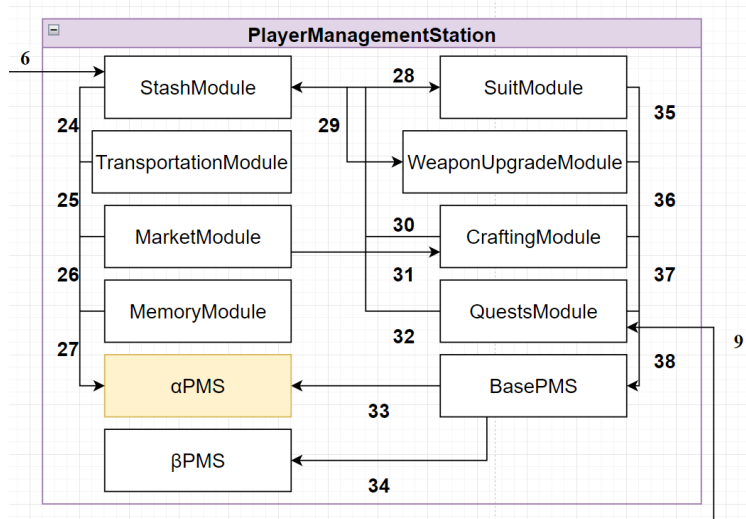


Figure 19: Player Management Station: Alpha PMS Subsystem

8.4.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

8.4.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI, UnityEngine.EventSystem, UnityEngine.InputSystem.

8.4.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

8.4.4 SUBSYSTEM DATA STRUCTURES

N/A.

8.4.5 SUBSYSTEM DATA PROCESSING

N/A.

8.5 BETA PMS

Has only access to half the modules.

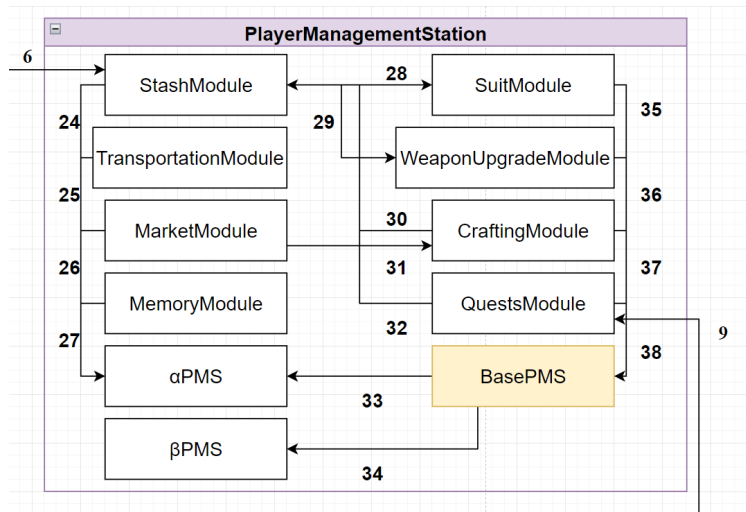


Figure 20: Player Management Station: Beta PMS Subsystem

8.5.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

8.5.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI, UnityEngine.EventSystem, UnityEngine.InputSystem.

8.5.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

8.5.4 SUBSYSTEM DATA STRUCTURES

N/A.

8.5.5 SUBSYSTEM DATA PROCESSING

N/A.

8.6 SUIT MODULE

Allows the player to upgrade their suit.

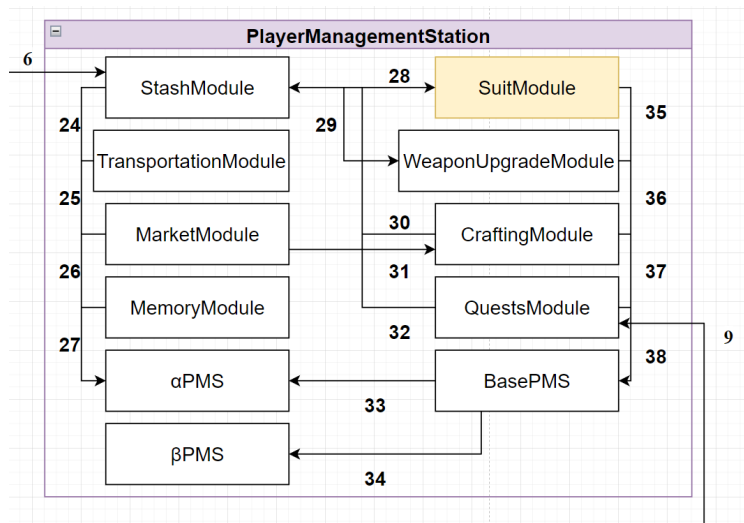


Figure 21: Player Management Station: Suit Module Subsystem

8.6.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

8.6.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI, UnityEngine.EventSystem, UnityEngine.InputSystem.

8.6.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

8.6.4 SUBSYSTEM DATA STRUCTURES

N/A.

8.6.5 SUBSYSTEM DATA PROCESSING

N/A.

8.7 WEAPON UPGRADE MODULE

Allows the player to upgrade their weapons.

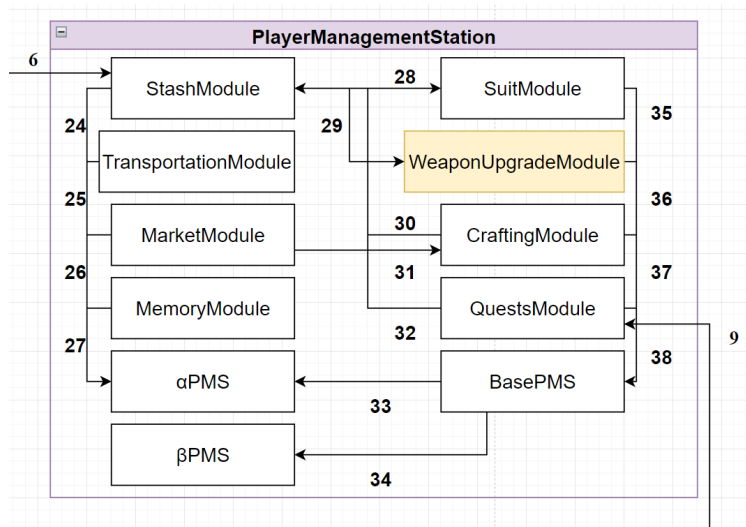


Figure 22: Player Management Station: Weapon Upgrade Module Subsystem

8.7.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

8.7.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI, UnityEngine.EventSystem, UnityEngine.InputSystem.

8.7.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

8.7.4 SUBSYSTEM DATA STRUCTURES

N/A.

8.7.5 SUBSYSTEM DATA PROCESSING

N/A.

8.8 CRAFTING MODULE

Allows the player to craft items.

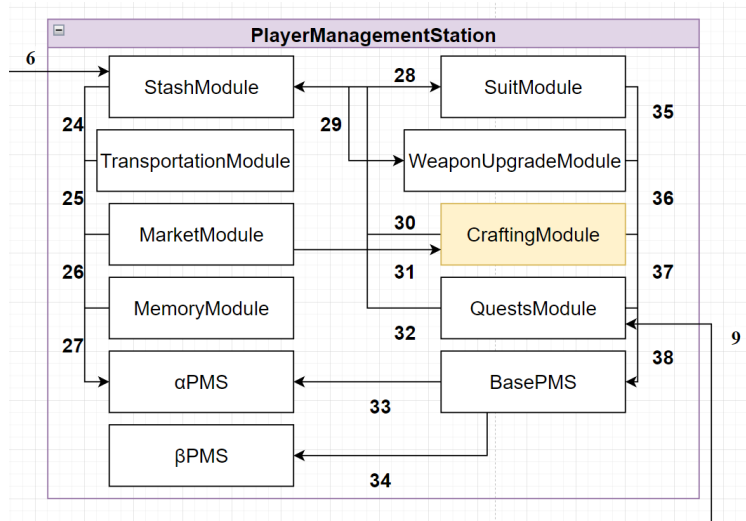


Figure 23: Player Management Station: Crafting Module Subsystem

8.8.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

8.8.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI, UnityEngine.EventSystem, UnityEngine.InputSystem.

8.8.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

8.8.4 SUBSYSTEM DATA STRUCTURES

Reference's Item, ItemSO, Container, and ContainerSO classes from the Inventory System to be able to store crafted items and remove ingredients from and to the player's inventory and stash.

8.8.5 SUBSYSTEM DATA PROCESSING

N/A.

8.9 QUESTS MODULE

Allows access to quests.

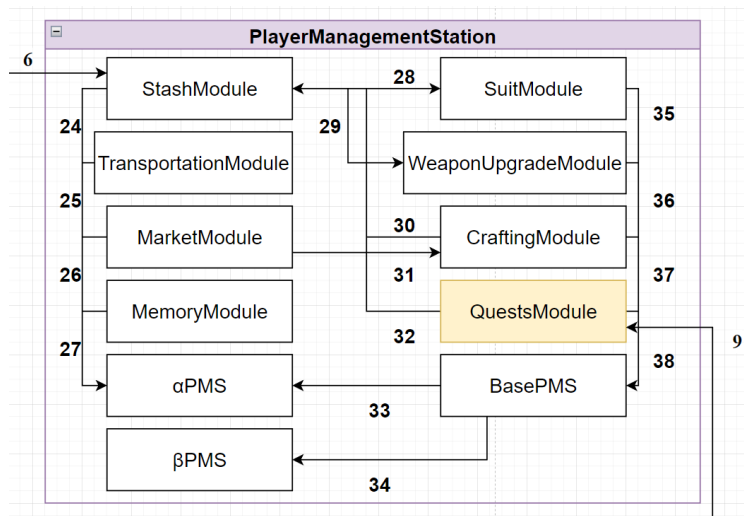


Figure 24: Player Management Station: Quests Module Subsystem

8.9.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

8.9.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI, UnityEngine.EventSystem, UnityEngine.InputSystem.

8.9.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

8.9.4 SUBSYSTEM DATA STRUCTURES

Reference's Item, ItemSO, Container, and ContainerSO classes from the Inventory System to be able to store items inside the player's stash and being able to turn in items for quest completion.

8.9.5 SUBSYSTEM DATA PROCESSING

N/A.

8.10 STASH MODULE

Allows the user to stash items inside.

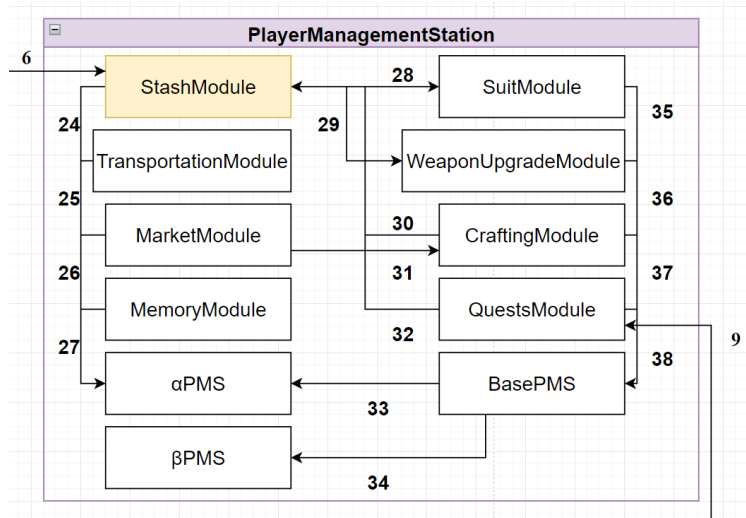


Figure 25: Player Management Station: Stash Module Subsystem

8.10.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

8.10.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI, UnityEngine.EventSystem, UnityEngine.InputSystem.

8.10.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

8.10.4 SUBSYSTEM DATA STRUCTURES

Reference's Item, ItemSO, Container, and ContainerSO classes from the Inventory System to be able to store within the cache and moving items to and from the stash to the player's inventory.

8.10.5 SUBSYSTEM DATA PROCESSING

N/A.

8.11 TRANSPORTATION MODULE

Allow the player to transport to other PMS that have a Transportation Module.

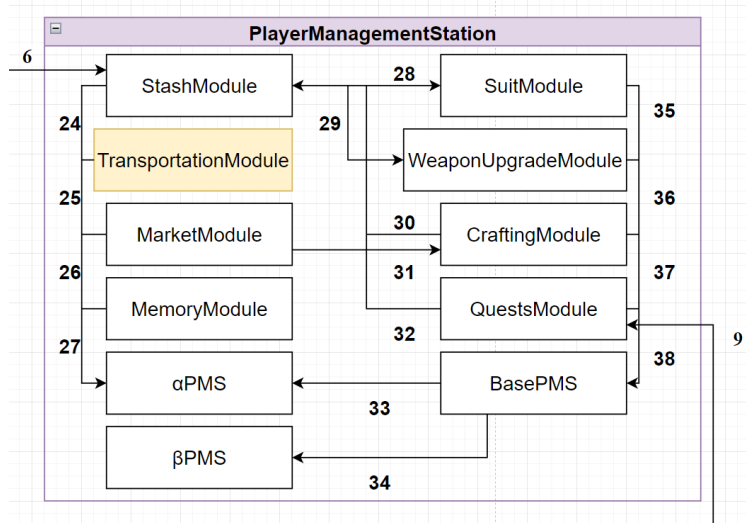


Figure 26: Player Management Station: Transportation Module Subsystem

8.11.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

8.11.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI, UnityEngine.EventSystem, UnityEngine.InputSystem.

8.11.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

8.11.4 SUBSYSTEM DATA STRUCTURES

N/A.

8.11.5 SUBSYSTEM DATA PROCESSING

N/A.

8.12 MARKET MODULE PMS

Defines the basic functionality of all Player Management Systems.

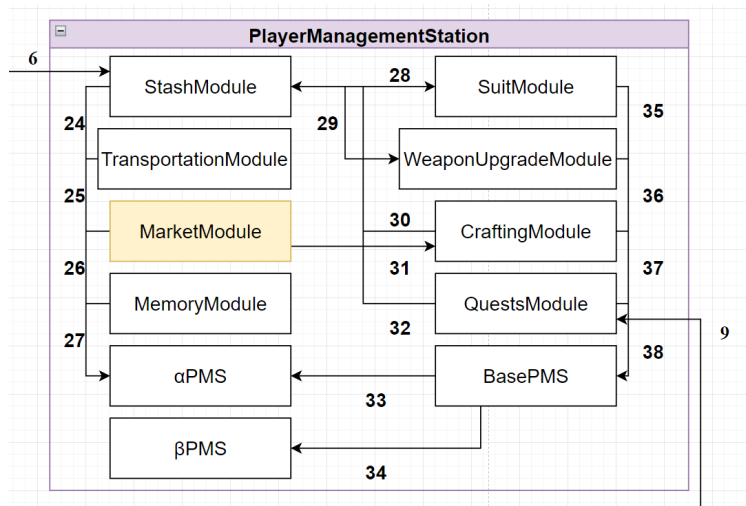


Figure 27: Player Management Station: Market Module Subsystem

8.12.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

8.12.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI, UnityEngine.EventSystem, UnityEngine.InputSystem.

8.12.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

8.12.4 SUBSYSTEM DATA STRUCTURES

Reference's Item, ItemSO, Container, and ContainerSO classes from the Inventory System to be able to store items inside the market and give items bought to the player's stash.

8.12.5 SUBSYSTEM DATA PROCESSING

N/A.

8.13 MEMORY MODULE

Allows the user to save, load, and exit the game.

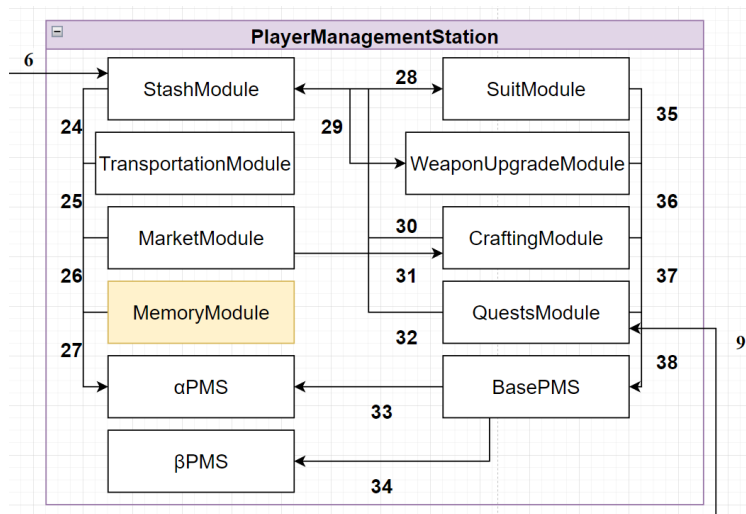


Figure 28: Player Management Station: Memory Module Subsystem

8.13.1 SUBSYSTEM OPERATING SYSTEM

Windows 10 or 11.

8.13.2 SUBSYSTEM SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI, UnityEngine.EventSystem, UnityEngine.InputSystem.

8.13.3 SUBSYSTEM PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

8.13.4 SUBSYSTEM DATA STRUCTURES

Uses a PlayerData struct that is XML serializable for easily loading and saving the player's current state into a XML file.

8.13.5 SUBSYSTEM DATA PROCESSING

N/A.

9 UI LAYER SUBSYSTEMS

The User Interface allows for the interaction of the player and the game's components through a visual medium. The UI primarily informs the player of in-game states such as player status, quest status as well as allowing for inventory interaction. The project is designed within the Unity game-engine and written in C# v9.0.

9.1 UI LAYER OPERATING SYSTEM

Windows 10 or 11.

9.2 UI LAYER SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI

9.3 UI CONTROLLER

The UI controller is software that manages the other UI components: Quest, Player and Inventory.

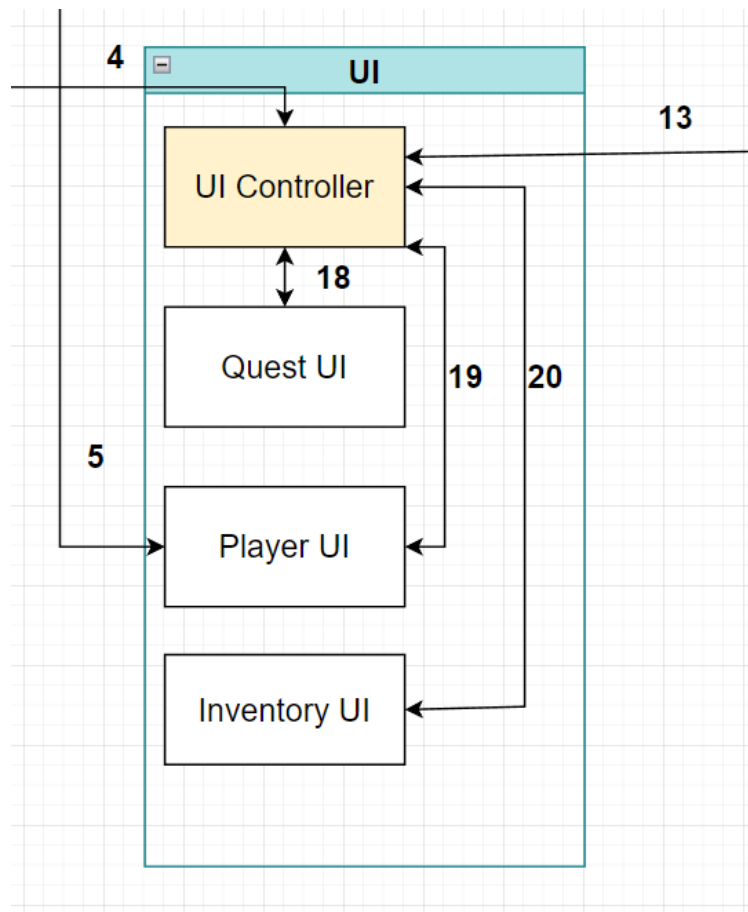


Figure 29: UI Controller Subsystem Diagram

9.3.1 UI CONTROLLER OPERATING SYSTEM

Windows 10 & 11.

9.3.2 UI CONTROLLER SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI

9.3.3 UI CONTROLLER PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

9.4 QUEST UI

The Quest UI is designed to allow the player to interact with and keep track of components related to the current task/quest.

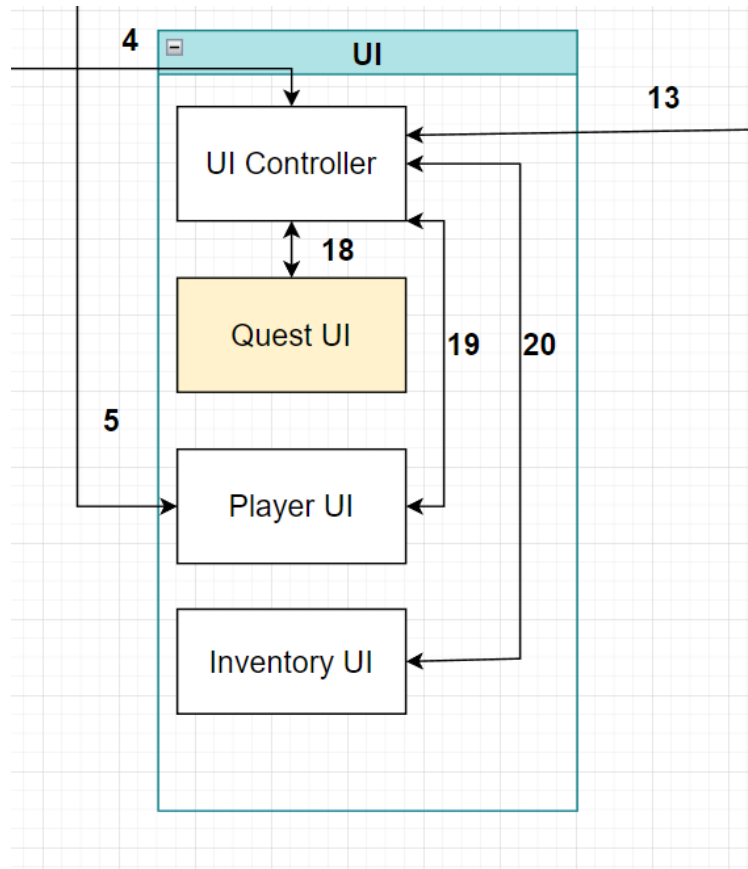


Figure 30: Quest UI Subsystem Diagram

9.4.1 QUEST UI OPERATING SYSTEM

Windows 10 & 11.

9.4.2 QUEST UI SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI

9.4.3 QUEST UI PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

9.5 PLAYER UI

The Player UI is designed with the purpose of displaying information related to the character's status as it relates to game mechanics. Primarily, the status of the characters health.

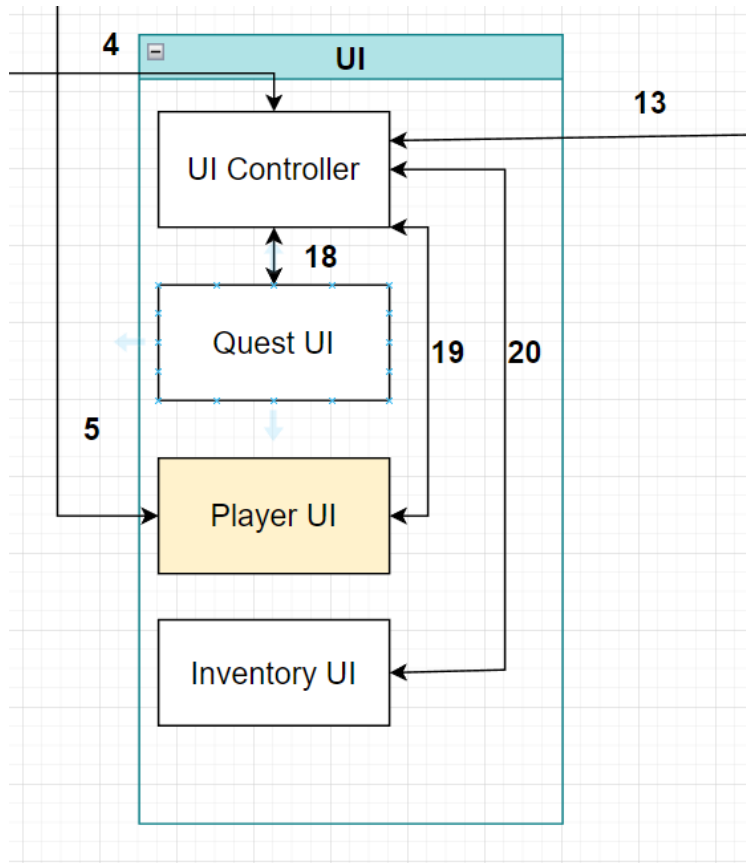


Figure 31: Player UI Subsystem Diagram

9.5.1 PLAYER UI OPERATING SYSTEM

Windows 10 & 11.

9.5.2 PLAYER UI SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI

9.5.3 PLAYER UI PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.

9.6 INVENTORY UI

The Inventory UI purpose is to allow for the player to visually interact with items collected within the inventory. Expected items include, but are not limited to, weapons, quest items and consumables. This system includes the display of items, inventory slot management, and inventory item interactions.

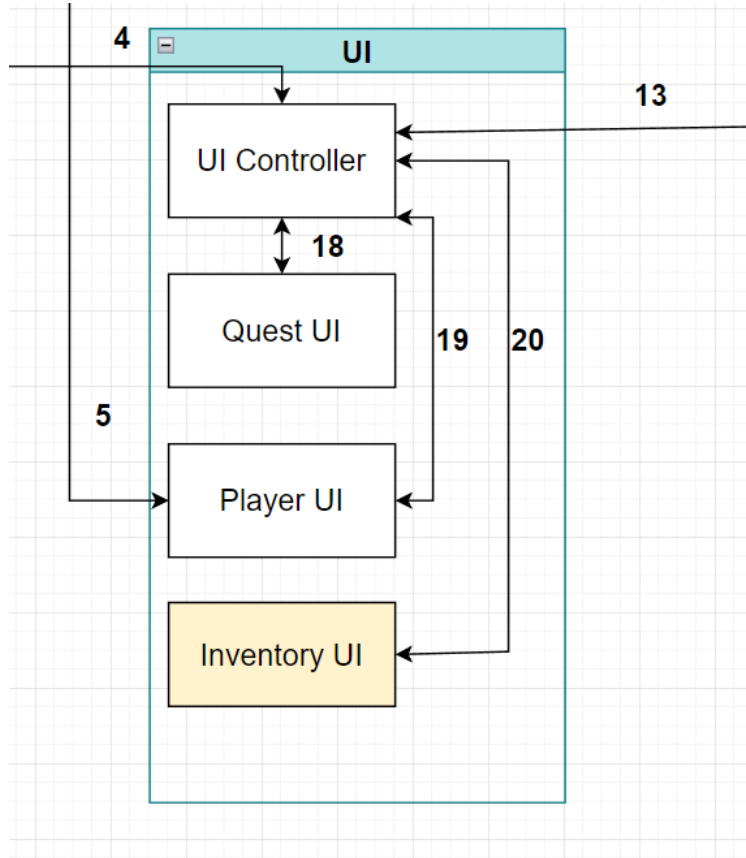


Figure 32: Inventory UI Subsystem Diagram

9.6.1 INVENTORY UI OPERATING SYSTEM

Windows 10 & 11.

9.6.2 INVENTORY UI SOFTWARE DEPENDENCIES

UnityEngine, UnityEngine.UI

9.6.3 INVENTORY UI PROGRAMMING LANGUAGES

C# v9.0 with reduced capabilities.