

# Documentation - Tomas Aquistapace

## Game developer

### Blue Gravity Studios

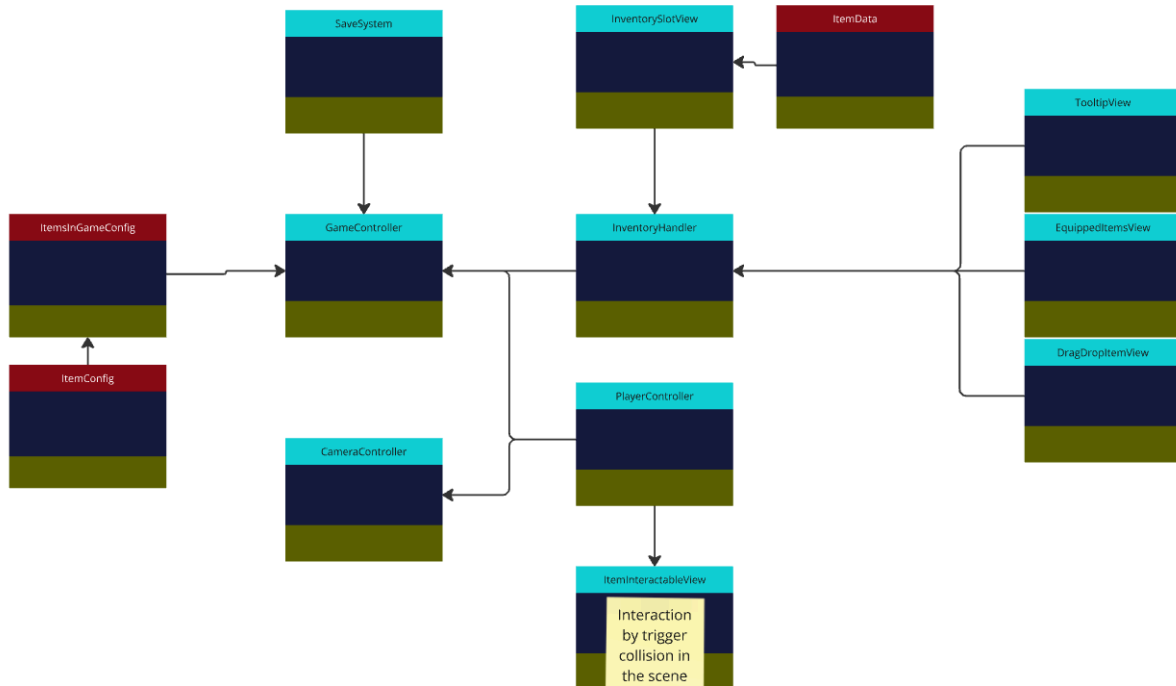
This document will discuss the operation and interactions of the different scripts in the program and how they interact with each other.

This small graphic shows which scripts are active in the project, how they are connected to each other, and their dependencies.

All the code and development was developed by me and no assets were downloaded from the unity asset store.

Only free artwork from the [Kenney](https://kenney.nl/) website was used.

Page used for making the graph: Miro



**Camera Controller:**

It is a small script that is responsible for moving the camera and following the target assigned to it, which in this case is the "player" object.

**PlayerController:**

Its functionality is to move and control the player using the system input.

**GameController:**

This script is in charge of starting the different scripts and passing an initial information of the items that are available in the scene, as well as loading the saved game.

**SaveSystem:**

This static class script is responsible for saving and loading the player's inventory information.

**ItemsInGameConfig:**

This script is responsible for storing the items found in the scenario so that this data can be used to load the player's inventory in an orderly fashion.

**ItemConfig:**

It is the container of the important information of each item found in the game; image, name, description and item type.

**InventoryHandler:**

It is in charge of managing and controlling the passage of item information. It is also the one who controls and manages different subsystems for the inventory items.

**TooltipView:**

In charge of displaying information of the selected item and being able to use or remove it from it.

**DragDropItemView:**

It has the functionality to move objects and show a preview image of the item being moved.

**EquippedItemsView:**

It has the functionality to contain and manage the objects in the player's hands; and to only allow a specific type of items to be added in those same slots.

**InventorySlotView:**

It is in charge of storing and displaying the object in a specific space of the inventory.

**Item Data:**

It is the class used to contain the information of the item without having to use or move any ScriptableObject.