

The project's systems doesn't have an strict dependence, they can perfectly work without knowing each other, apart from the main one:

GameController

- All functionalities are centralized here, It knows every core system
- It also works as an Initializator, the whole app flow starts here

ShopController

- Handles the item buy / sell functionality
- It also works as an item handler

WardrobeController

- Handles the equip / unequip functionality

PlayerController

- Handles the player core behaviours (movement / animations / visuals / interactions)

Detector

- Handles the detection of another entity beneath a determined radius

Every system is working as intended, managing only what it was made for.

Personal Assessment:

As soon as I read the requirements document I knew how I would develop the project, so it went quite straightforward. I tried to work in every system one by one, not mixing behaviours nor multitasking, as I finish a behaviour I went ahead with the next one, and I think that is reflected in the commits.

Making the systems as requested while working with many asset sources is a bit complicated due to mismatches between art pieces. As time was short, I decided to adapt the character parts based on the body pieces I found in the source, that's why I ended just with just 3 item types.

I am truly proud of the final product, though the time span was short I managed to implement the main systems first, so I had plenty of time searching for good looking assets to improve its visuals, as well as code reviewing and polishing, trying to reuse as much code as I could and make it as clean as possible.

I had some little problems as the project advanced, such as a few bugs that I could solve without further inconveniencies, and allowed me to add more useful code, for example the Interactable interface, which later eased the entities dialog box implementation.