Practical Gaming 2024

# Anna Kovalenko

# T00243622

# Big Adventure

# Gameplay

W – Move Forward

A – Move Left

D – Move Right

S – Move Back

Space – Jump

R – Through a ball in a basketball mini game

T – Talk with NPCs

F – Open treasure chests

Mouse Left – Hit Enemies, Sell items

Mouse Right – Defence

Mouse position is a position of the camera

Teleports to move between game locations

Inventory system to store collectable items

Collectable Items:

1. Apples
2. Bones
3. Diamonds
4. Fish
5. Keys
6. 6.Rocks

Other items:

1. Mushrooms – to give damage to the player
2. Heart – to give health to the player

NPCs:

1. Haruko - at the start/end of the game
2. Sellers – to sell collectable items and get experience (located in the castele city)
3. Enemies – to fight with the player
4. Animals – to make game more real

The player should explore the world by collecting items, killing enemies, talking and selling items to npcs. The player goal is to find the lost flower on the last level.

# Coding

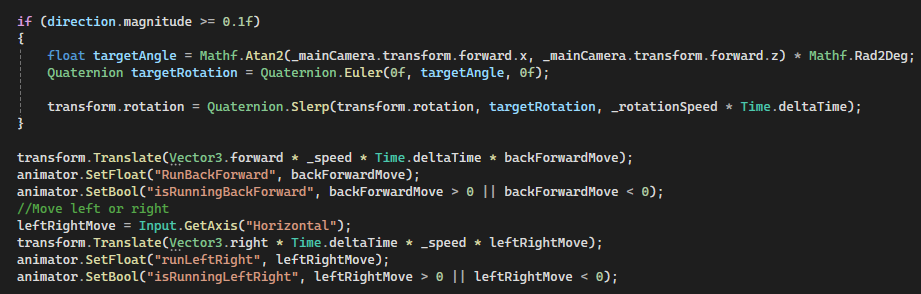
Under each of the following headings, please describe the concept, why is it or isn’t it useful/needed, where do you implement in your project, you may provide screenshots or cut and past code segments etc..

* Frame Rate Independence

Frame Rate Independence happens when game runs at the same speed as a frame. Here some examples of it:

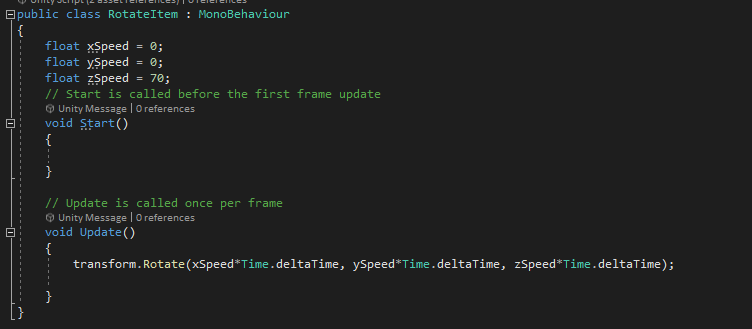
1. PlayerController Script

The Frame Rate Independence happens when player moving.



1. RotateItem Script

The Frame Rate Independence happens when item is rotating

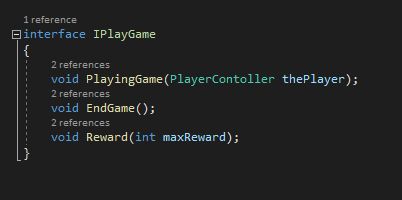


* Interfaces

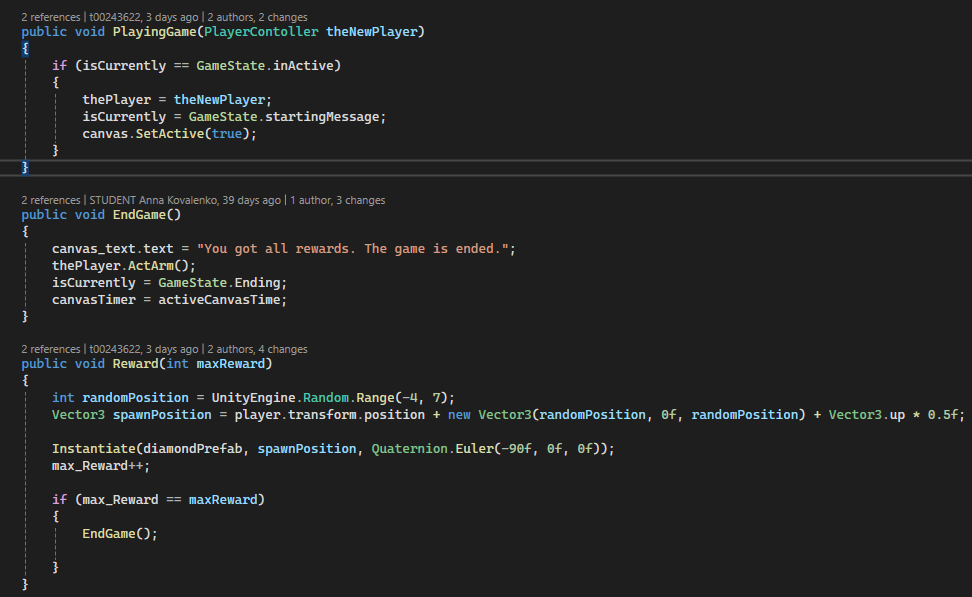
Interface is a way for objects to communicate with each ether without knowing all information about the objects. Here are some examples:

1. IPlayGame Interface

The interface helps to communicate the player and the basketball mini game with each other.



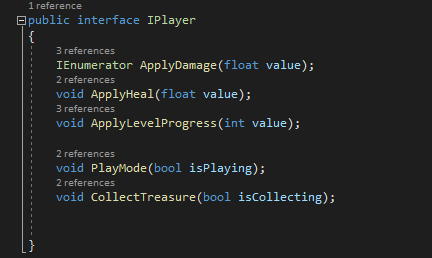






1. IPlayer Interface

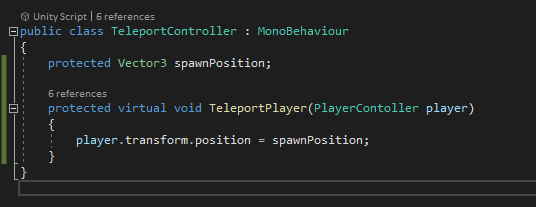
The Interface helps to communicate the player with the NPC\_Controller and with the Damage Script.

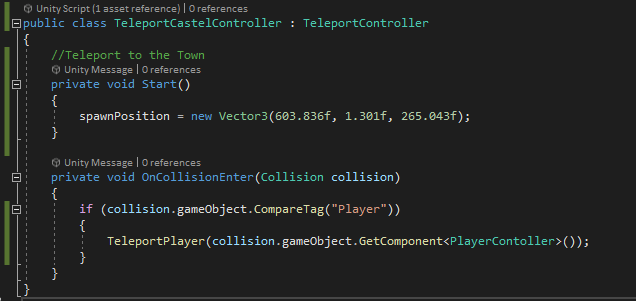


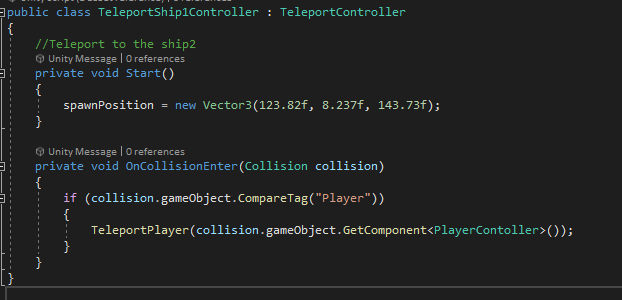
* Inheritance

Inheritance is a concept that allows classes inherit variables and methods from a base class. Here is my example:

TeleportController is a base class and TeleportCastelController, TeleportShipController are classes that inherit variable “spawnPosition” and method “TeleportPlayer” into their scripts.



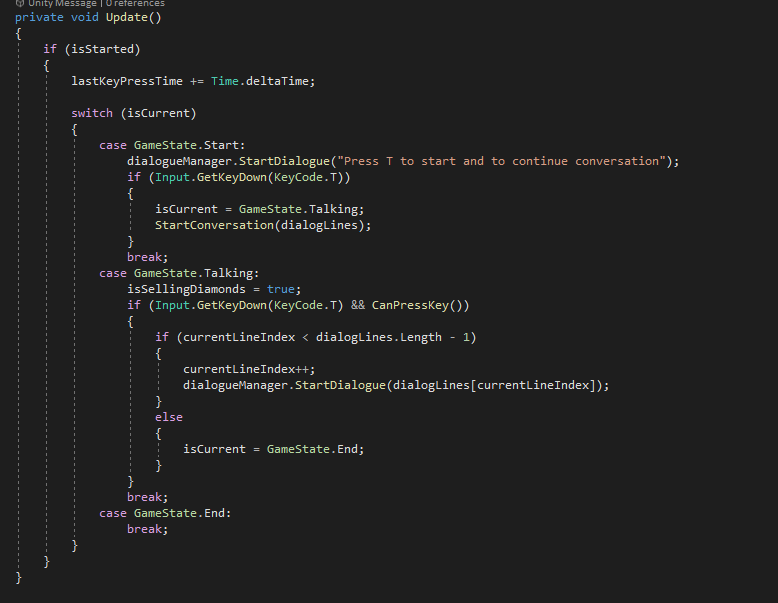




* Case pattern

Case Pattern is a way to have your code clean and understandable, handling different scenarios that can happen. Here is my example:

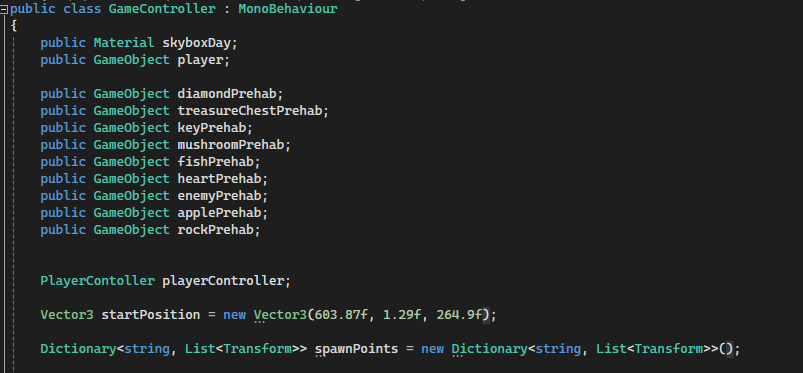
NPC\_Controller Script



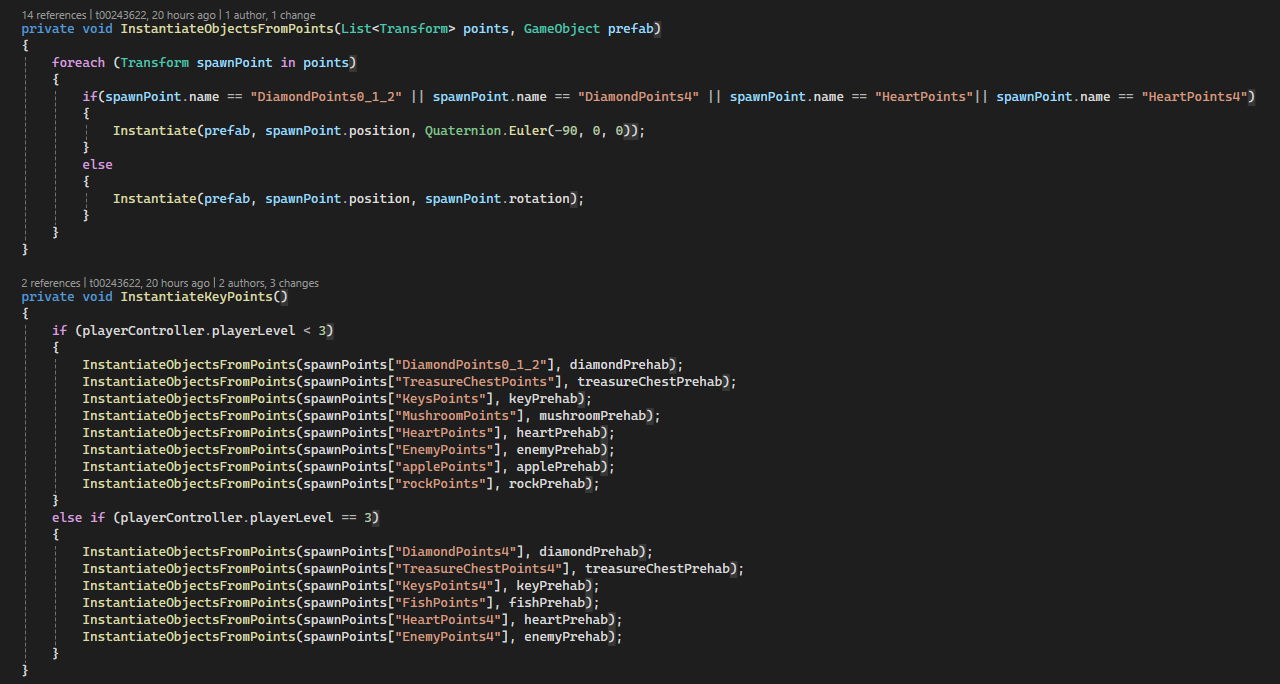
* Observer Pattern

Observer Pattern is a pattern that maintains a list of objects and work with it. Here is my example:

GameController Scrirt has a dictionary that collect all the spawn points in my game, print the quantity of the objects collected and instantiate them on the map.



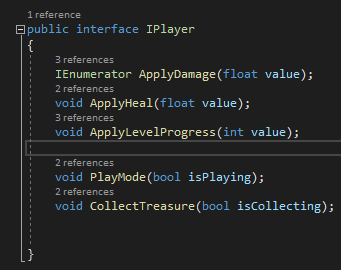


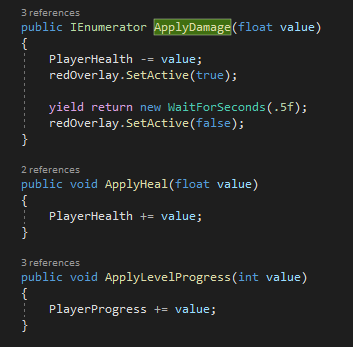


* Polymorphism

Polimorphism is when you manipulate children by using the base class's name. Here is my example:

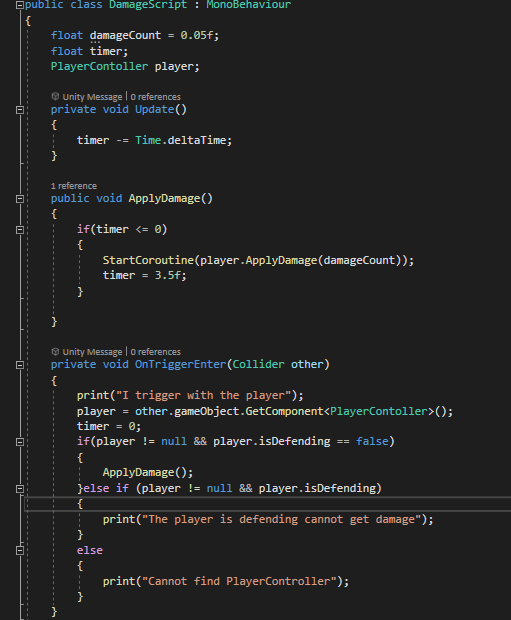
The Interface IPlayer has 5 methods and the Player Controller Scripts use these methods.





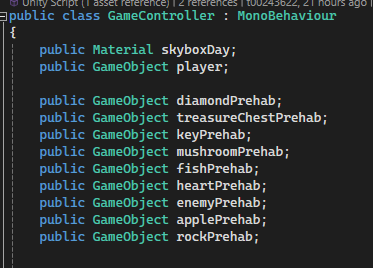
* Communication between scripts/game objects

Communication between scripts happen here, between Damage Script and Player Controller. If it triggers with the player it takes PlayerController Script and applies damage to the player it it can take the PlayerController Script.



* Instantiation and Prefabs

The GameController Script has 9 Prefabs and instantiate thes on the map when game is started.



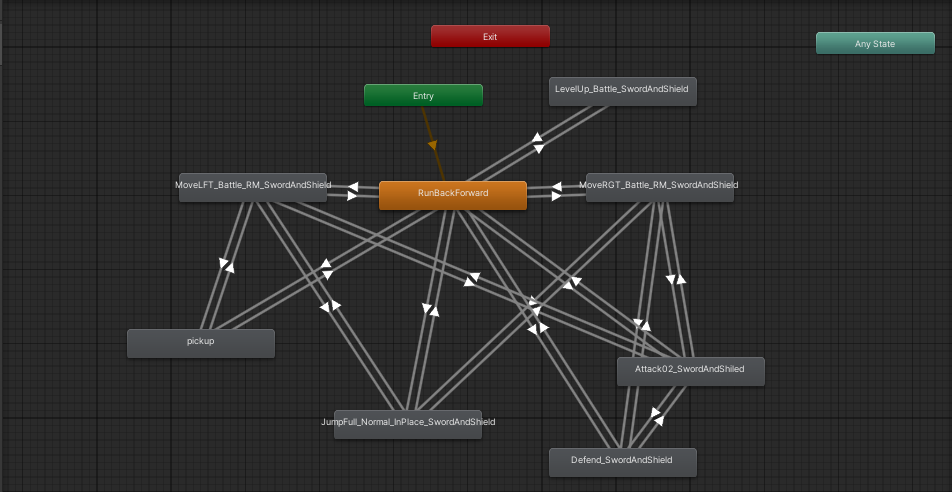


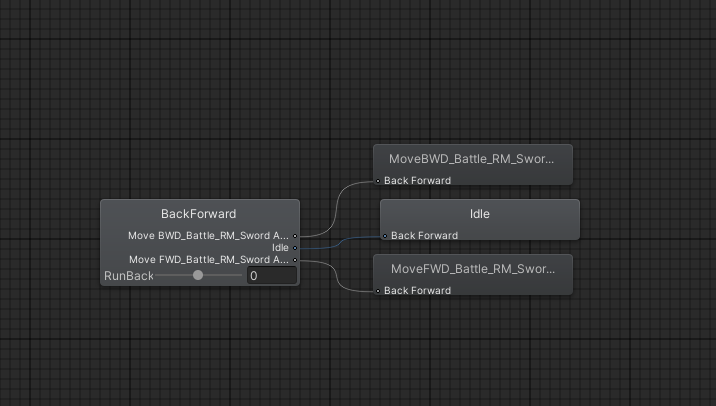
* Magic Numbers

My project does not have magic numbers because it is numerical value without explanation, make code hard to understand and hard to find and change the value in the future.

* Model Animation

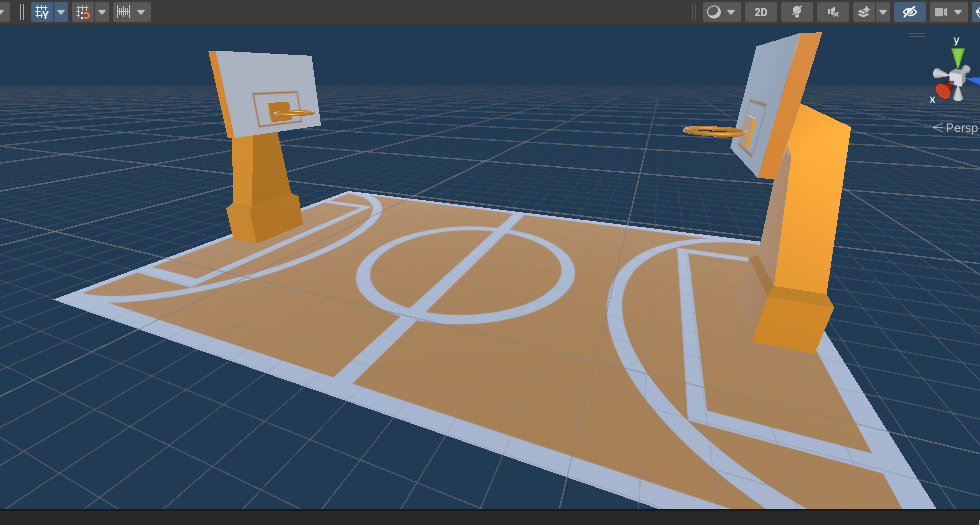
The main player of my game has a animation. Here is my animation model:

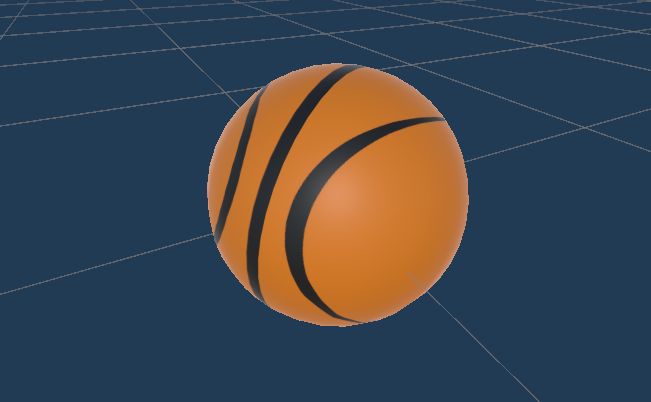


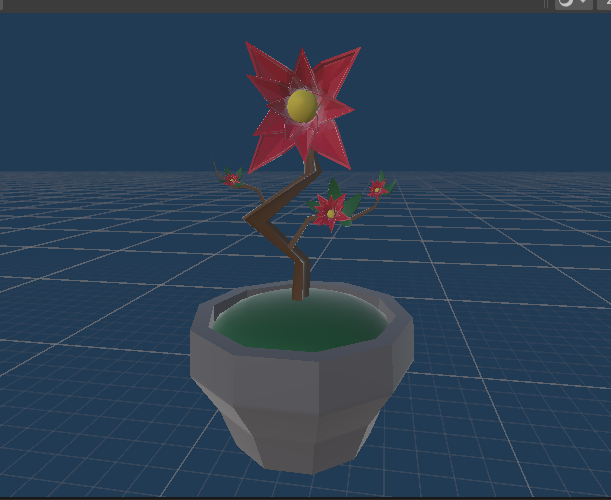


* Self made models and or animations

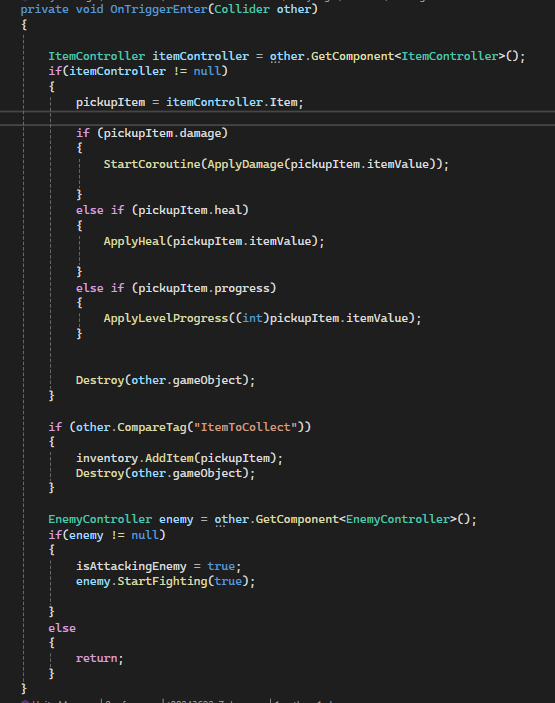
All the 3 models are made by myself in Maya 3D.







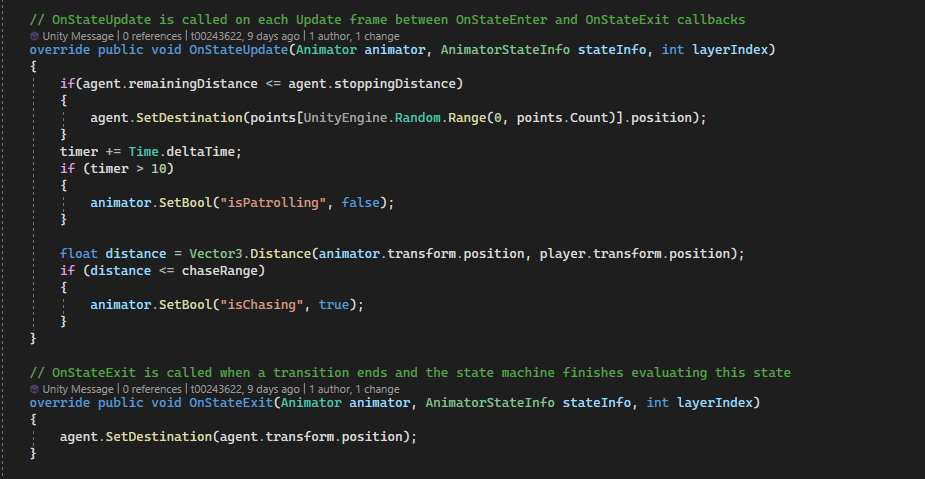
* Interactions between objects/scripts
  + PlayerController Script



* Propper code placement

I take patrolBehavious Script as a proper code placement for enemy animation.



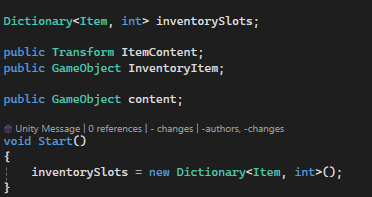


* Code repetition

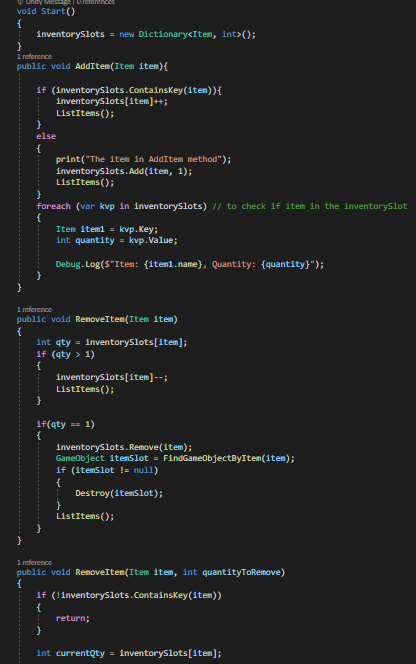
The project avoids code repetition because it makes the code harder to maintain and understand. This approach can lead to longer loading times, as the program is more complex with repeated code. By keeping code non-repetitive, the project aims for easier maintenance and faster loading.

* Feature 1

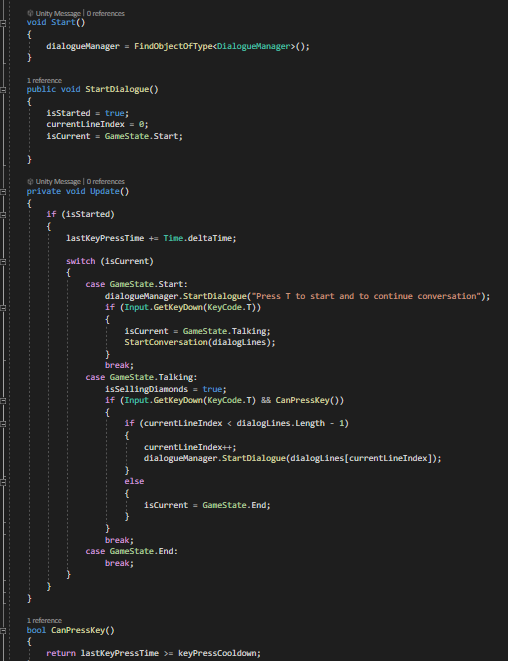
Using a dictionary instead of a list or array is better because it lets you access data directly by using descriptive keys, making code easier to read. This is faster and more efficient, especially with large amounts of data.



* Feature 2
  + Inventory System

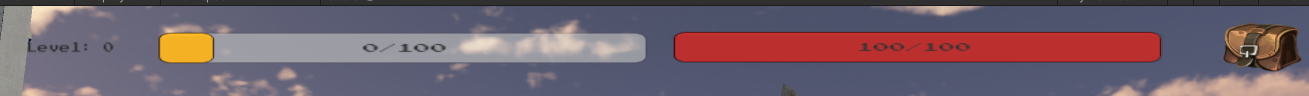


* Feature 3
  + Dialog System with NPCs



* Feature 4

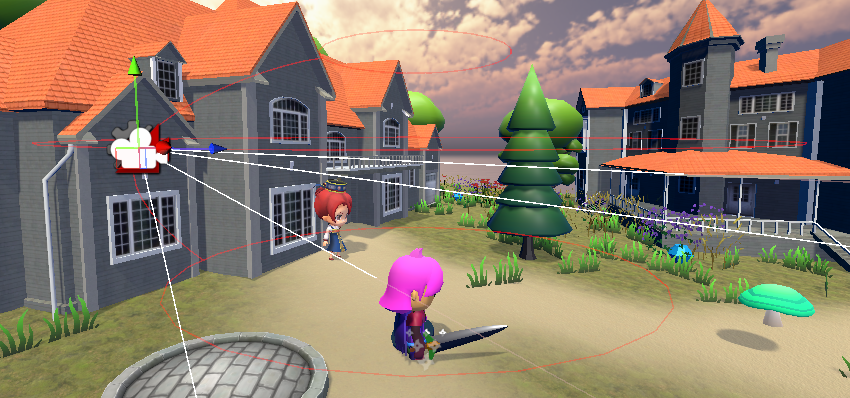
User Interface





* Feature 5

Camera Controller (3rd person view)



* Feature 6

Red Overlay when player get hit.

