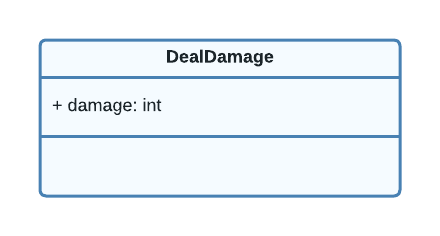
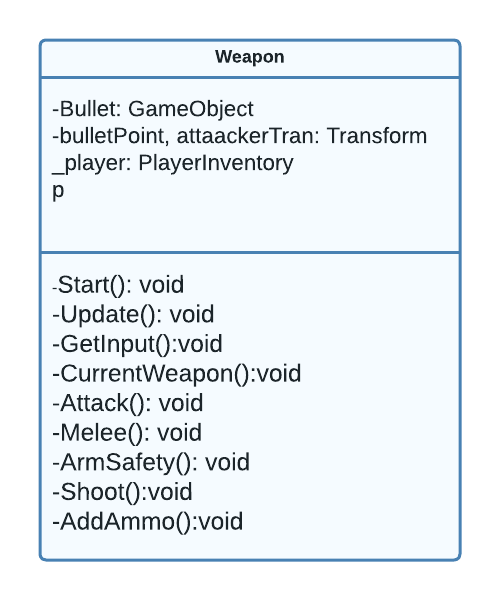
OOP – Documentation draft V1

* Project Concept
* Graphics and Sounds
* Programming
* Character Controller
* Enemies/Obstacles & Random Elements
* Non-MonoBehaviour class



* Instantiation



The weapon class is the system responsible for player attacks At the start it set the player’s bullets to the size of the magazine and gets a reference to the bullet spawn point. In Update is where most of the functions are executed in order. First checks the current weapon which is done with Enums Guns and Sword, on default current weapon is Gun but (KeyCode.Z and KeyCode.X) is used to switch between Gun and Sword respectively. Then Checks for player input for Attack, if the player wants to attack if the current weapon is Sword, it calls the Melee function else check the ArmSafetyOff function for the gun which calls the shoot function if some conditions are met. The Shoot function instantiates bullets prefabs from the attack transform position and gets the rigidbody to the prefab and adds velocity to it depending on the direction of the player then reduces the number of bullets the player has left.

* Methods – without arguments or returns
* Methods – appropriate use to organise code
* Methods – using inputs/arguments
* Methods – methods with non-void return type
* Use of collections such as Arrays or Lists.
* Bit flags
* While loop
* Foreach loop
* For loop
* Try/Catch
* Enums
* Inheritance and Polymorphism
* Encapsulation and Integrity
* OOP TechniquesDiagram

  Description automatically generated
* Encapsulation

Encapsulation is the process in which no direct access is granted to the data; instead, it is hidden. If you want to gain access to the data, you must interact with the object responsible for the data. (Clark, 2013)

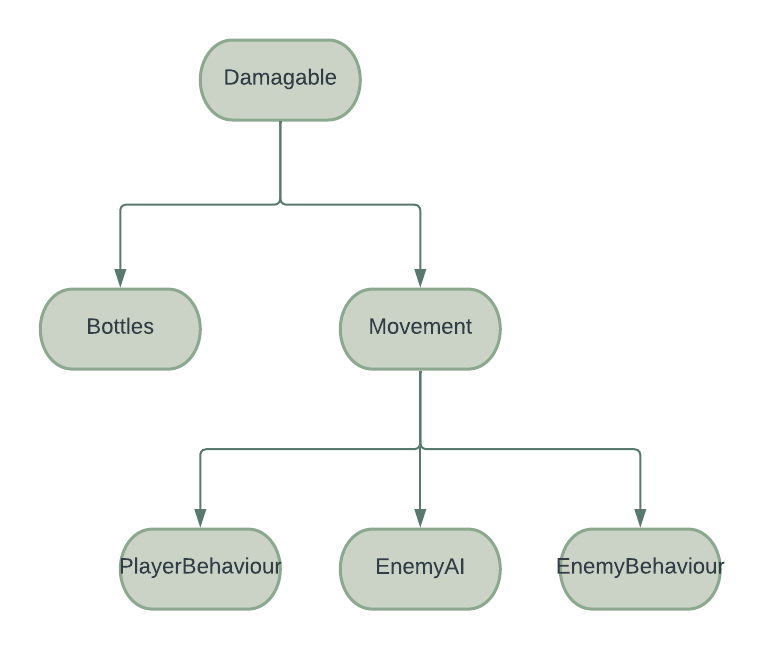
In this game demo, the player’s health has been used in multiple places the UIManager, Damage, Health Collectables are used to show the health bar, dealing damage to players’ health, adding to health respectively. So, the player inventory has a get property is used to allow external classes access to the private store health variable.

Inheritance and polymorphism work hand in hand together.

* Inheritance

Most objects are classified according to hierarchies. For example, you can classify all dogs together as having certain common characteristics such as having four legs and fur. It makes programming easier because it enables you to combine general characteristics into a parent object and inherits these characteristics in the child objects. (Clark, 2013)

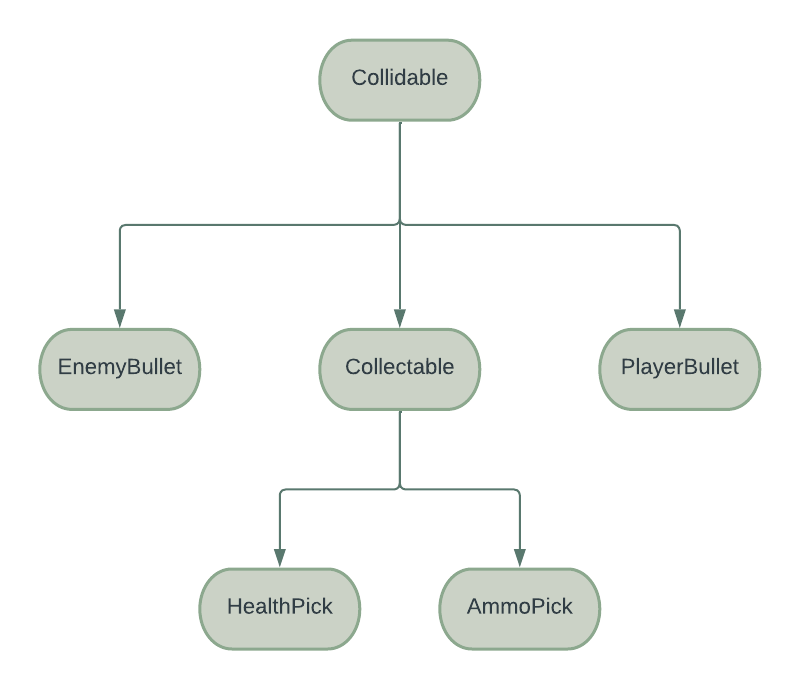
All classes created in unity inherit from Monobehaviour. The project makes use of inheritance to make some class inherit from some parent class is created. The main ones are Collidable and Damageable.



Damagable, a class that has the function of anything that is destroyable in the game demo which has two children’s classes Bottles and Movement, bottles are damageable but do not have the function to move on their own. Movement is a child to Damageable, which is a child of its own PlayerBehaviour, EnemyAI1, EnemyA2. Every moveable object in the game can be destroyed. The player has its own unique movement and is Damageable and Enemies have their own movements and can be destroyed in a unique way.

* Polymorphism

Talking polymorphism, Polymorphism is the ability of two objects to respond to the same request message in their own unique way (Clark, 2013). Polymorphism is the way that makes all this hierarchical structure works out using virtual and override.



Collectables, EnemyBullet, PlayerBullet all inherit from collidables. Responds to a function differently “OnCollide”. OnCollide is made virtual in the Collectable class which enables any child created to override the function EnemyBullet destroys Player when the function is triggered while PlayerBullet destroys anything destroyable exception from itself and finally Collectable awards player with gifts depending on its child variables.

* Summary
* References
* Appendix A – Existing Assets
* If you used any existing assets such as sprites, sounds, or music, including a list or table with names and URLs. Please ensure you have appropriate copyright permissions for anything you use.

Comments

I have a struct class that is like encapsulation its used in sending message damage

Abstract keyword that prevents a class to be used directly