Шаблонний метод — це поведінковий патерн проектування, який визначає кістяк алгоритму, перекладаючи відповідальність за деякі його кроки на підкласи. Патерн дозволяє підкласам перевизначати кроки алгоритму, не змінюючи його загальної структури.

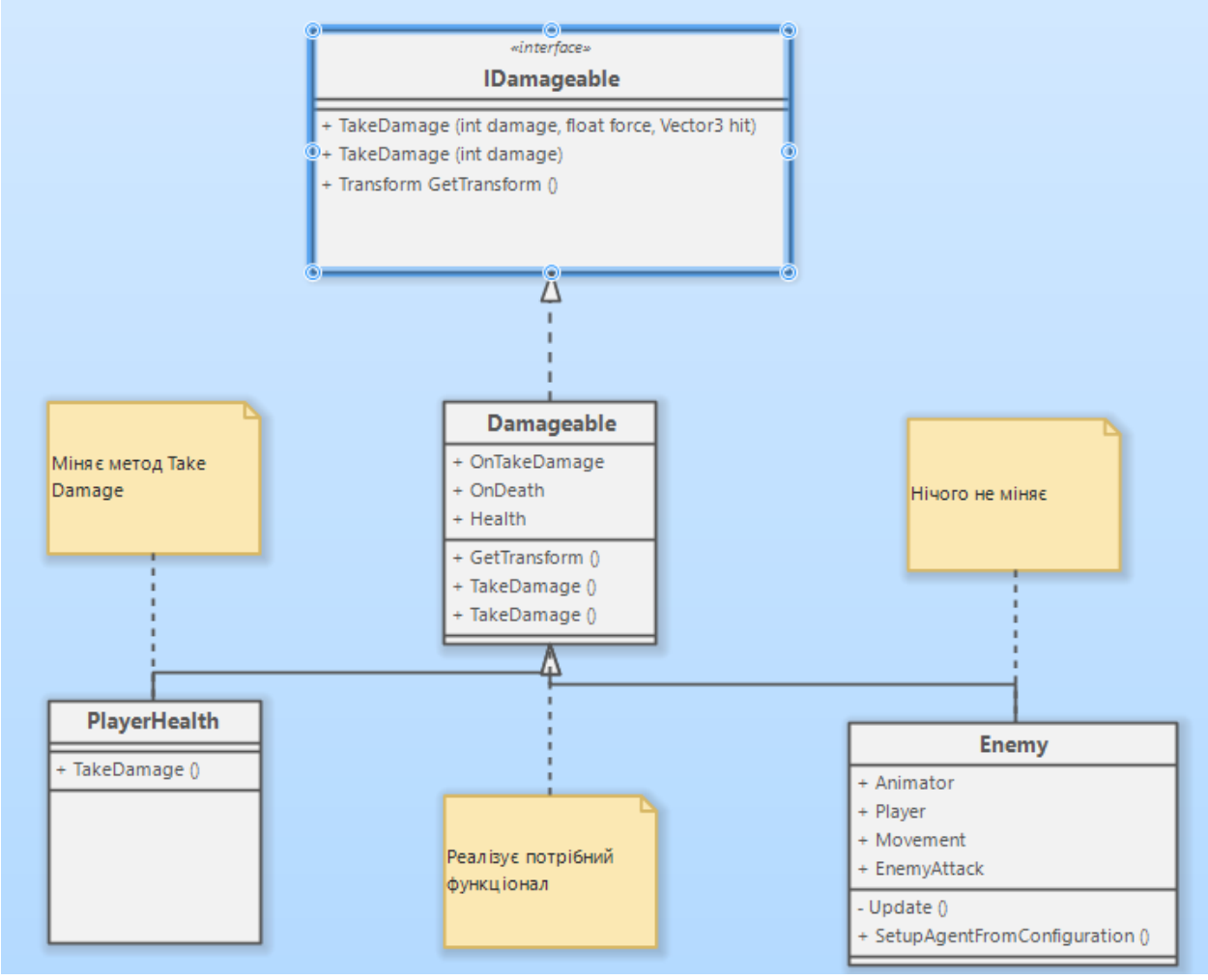
В моєму проекті є інтерфейс **IDamageable**.

Клас **Damageable** реалізує інтерфейс **IDamageable**, а після чого всі об’єкти, яким потрібні, успадковують клас **Damageable**. Цей клас в мене містить поле зі здоров’ям, а також інші потрібні методи та поля. Тобто він і є шаблонним.

Інші методи по типу **PlayerHealth**, успадковують клас **Damageable** та перероблюють деякі методи або додають нові.

Наприклад вищезгаданий **PlayerHealth**, переробляє лише метод **TakeDamage**, а інші метод та поля мають стандартну реалізацію. Також до прикладу клас Enemy, також успадковує **Damageable**, але нічого не переробляє.

Тобто в класі **Damageable** я виніс всі поля та методи, які повторюються.



Клас **Damageable** реалізує методи інтерфейсу та додає деякі потрібні поля виглядає так:

```
public class Damageable : MonoBehaviour, IDamageable
{
    public delegate void TakeDamageEvent(float force, Vector3 hit);
    public delegate void DeathEvent();
    public TakeDamageEvent OnTakeDamage;
    public DeathEvent OnDeath;
    public  int Health = 100;

    2 references
    public virtual Transform GetTransform()
    {
        return transform;
    }

    3 references
    public virtual void TakeDamage(int damage, float force, Vector3 hit)
    {
        TakeDamage(damage);

        OnTakeDamage?.Invoke(force, hit);
    }
    8 references
    public virtual void TakeDamage(int damage)
    {
        Health -= damage;


        if (Health <= 0)
        {

            OnDeath?.Invoke();

        }
    }
```

Клас **PlayerHealth** має трохи своєї логіки, але більшу частину бере від **Damageable.**

```
public class PlayerHealth : Damageable
{


    7 references
    public override void TakeDamage(int damage)
    {
        if(TryGetComponent(out CameraShake cameraShake))
        {
            cameraShake.FooCameraShake(damage / 20);
        }
        Health -= damage;

        if (Health <= 0)
        {
            Debug.Log($"Damage {damage}. You're Dead!");
            OnDeath?.Invoke();
        }
        else
        {
            Debug.Log($"Damage {damage}");
        }
    }


}
```

Клас **Enemy** не міняє логіки **Damageable.**

Скрипти можете знайти за шляхом:

Assets/Scripts/CommonCore/Damageable/Damageable.cs

Assets/Scripts/CommonCore/Damageable/IDamageable.cs

Assets/Scripts/CommonCore/PlayerScripts/PlayerHealth.cs

Assets/Scripts/Enemies/Base/Enemy.cs