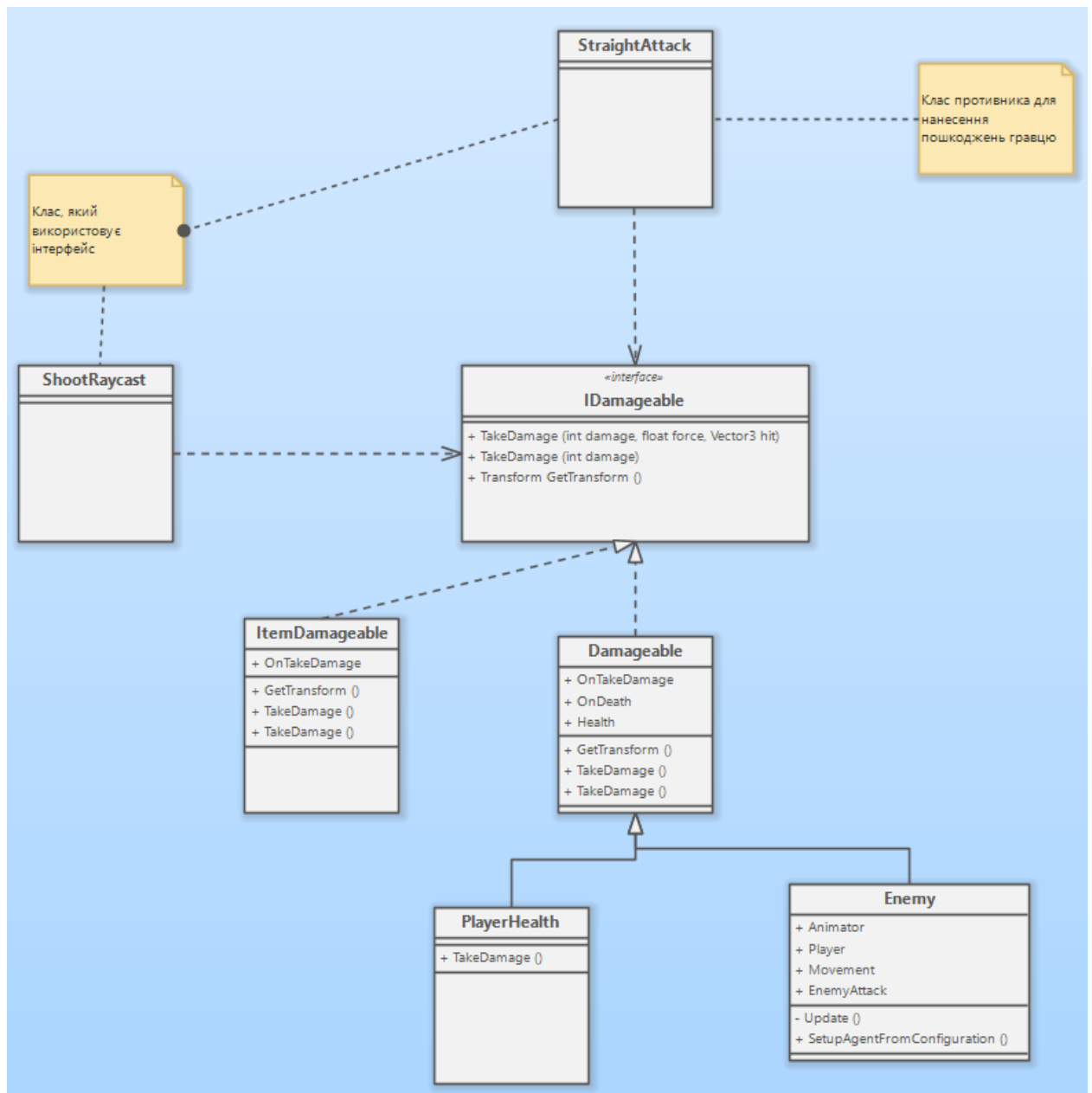


Фабричний метод - це породжувальний патерн проектування, який визначає загальний інтерфейс для створення об'єктів у суперкласі, дозволяючи підкласам змінювати тип створюваних об'єктів.

В моєму проєкті цей патерн використовується в багатьох випадках.

Наприклад клас **Damageable** в мене відповідає за нанесення урону об'єкту. Сам клас реалізує інтерфейс **IDamageable**, в якому винесені всі потрібні методи. Для різних противників та для нашого гравця є свої окремі класи які успадковуються з класу **Damageable**. Цей клас використовується як база, заради зменшення коду (зменшення копіпасту), тому що багато класів потребують однакову реалізацію деяких функцій з інтерфейса, але при цьому мають мати і свою власну логіку. Тому було прийнято рішення винести всі важливі функції в клас **Damageable**. Тобто цей клас також використовує шаблонний патерн.

Фабричний метод полягає в тому що, коли якийсь об'єкт намагається нанести пошкодження іншому, то для цього використовується інтерфейс **IDamageable**. Тобто інші класи не прив'язуються до класу **Damageable**, або подібним йому, а використовують інтерфейс **IDamageable**.



Цей патерн також використовується багато де в моєму проєкті, оскільки він є універсальним та добре підходить для реалізації різних типів.

Також хочу зазначити, що UML діаграми використанні тут не є точними, а лише показують основну структуру класів та успадковування, для кращого ознайомлення раджу подивитися скрипти на моєму гітхабі.

Скрипти можете знайти за шляхом:

Assets/Scripts/CommonCore/Damageable/Damageable.cs

Assets/Scripts/CommonCore/Damageable/IDamageable.cs

Assets/Scripts/Interactables/Common/ItemDamagable.cs

Assets/Scripts/CommonCore/PlayerScripts/PlayerHealth.cs

Assets/Scripts/Enemies/Base/Enemy.cs

Assets/Scripts/Weapon/ShootingWeapon/MainScripts/ShootRaycast.cs

Assets/Scripts/Enemies/Attack/StraightAttack.cs