

インターフェース

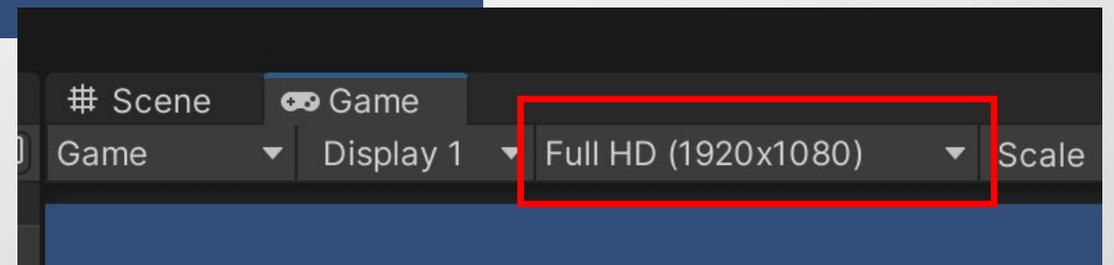
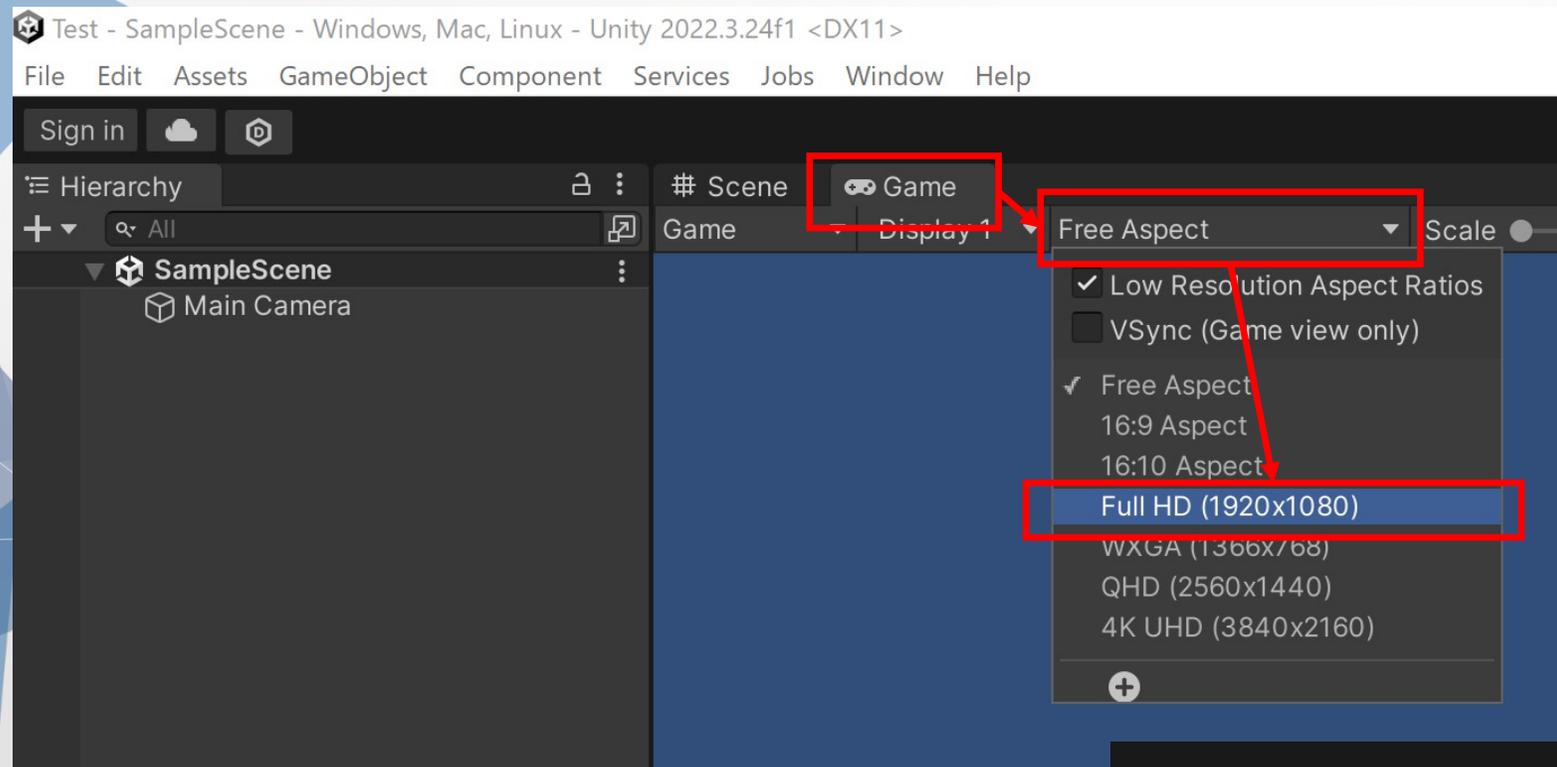


目次

1. インターフェースとは
2. Unityにおける使い方
3. 問題

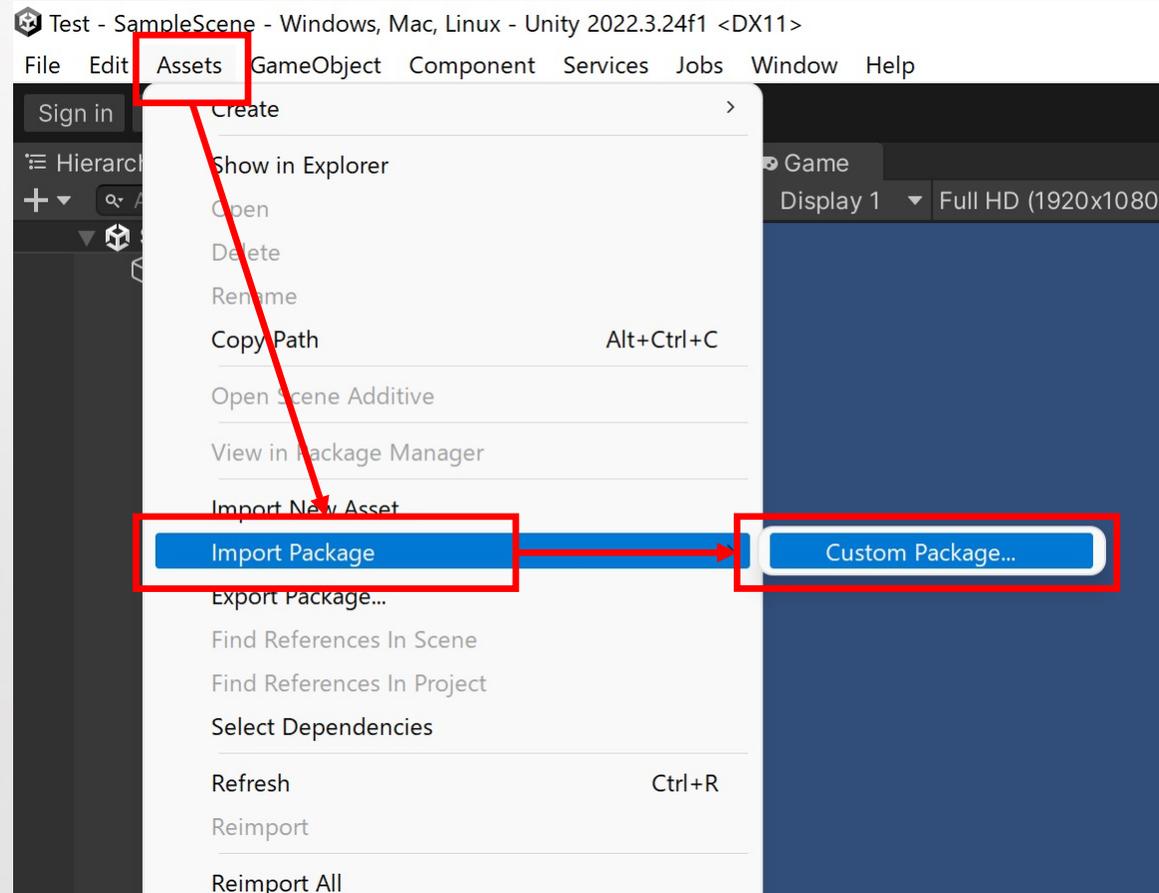
0. 前準備

プロジェクト開いた人は、以下の操作をお願いします。(画面の解像度の設定)



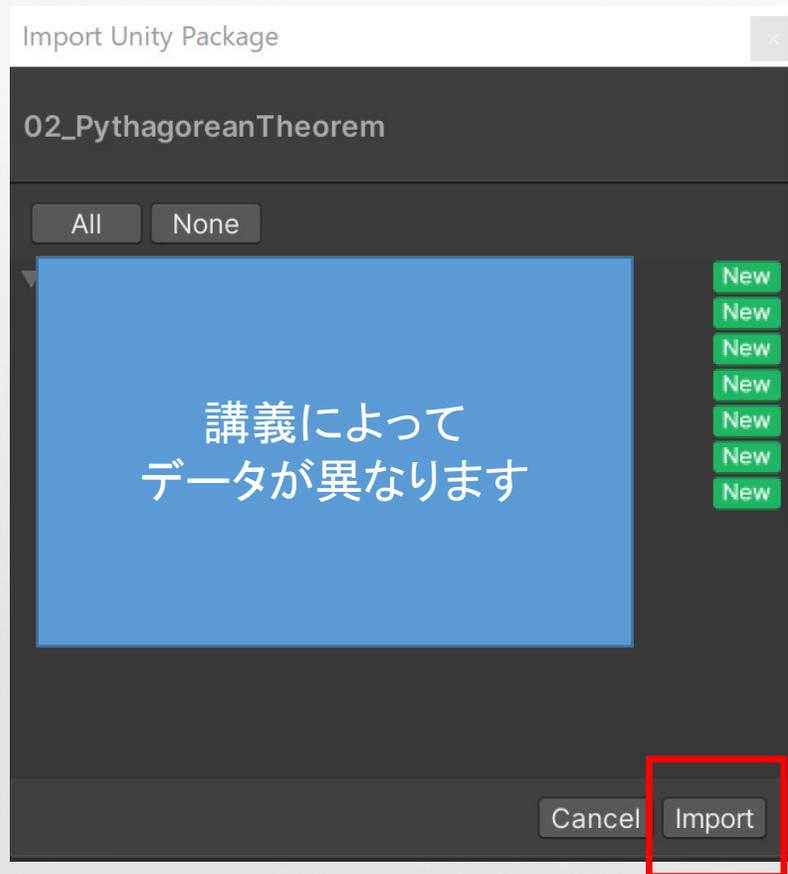
0. 前準備

次に、unitypackageというパッケージファイル(データが詰め込まれたファイル)を読み込みます。



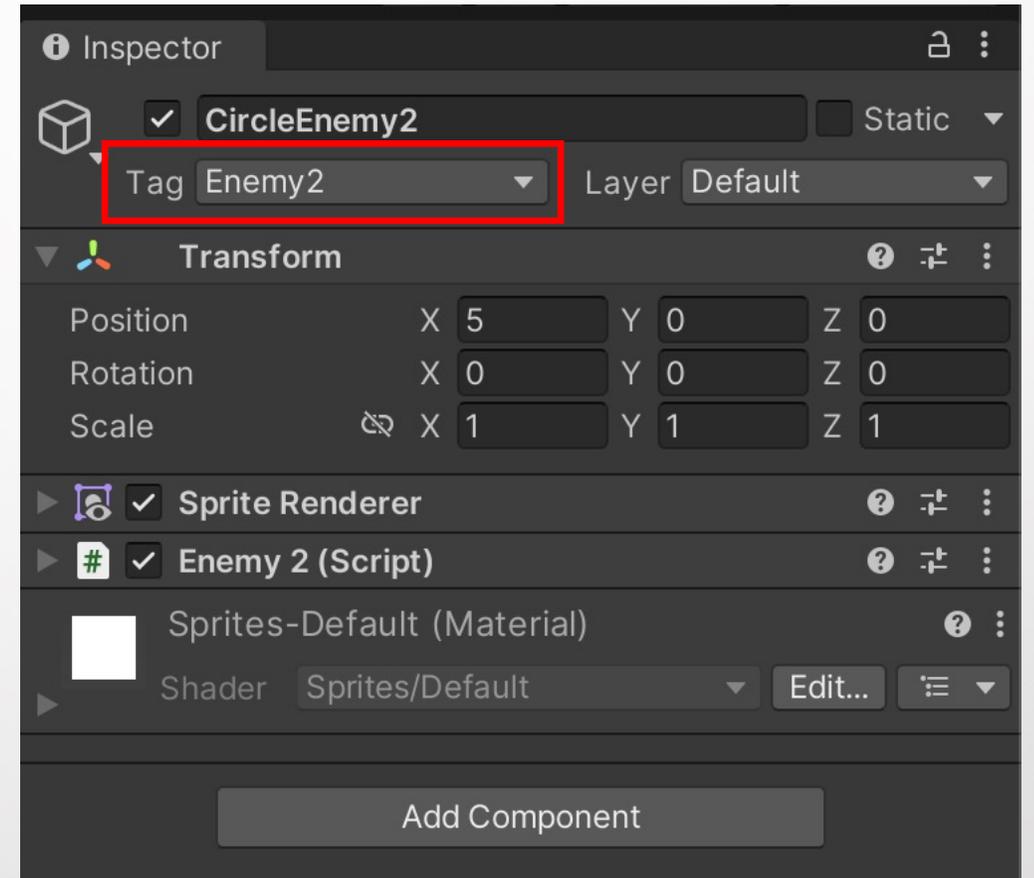
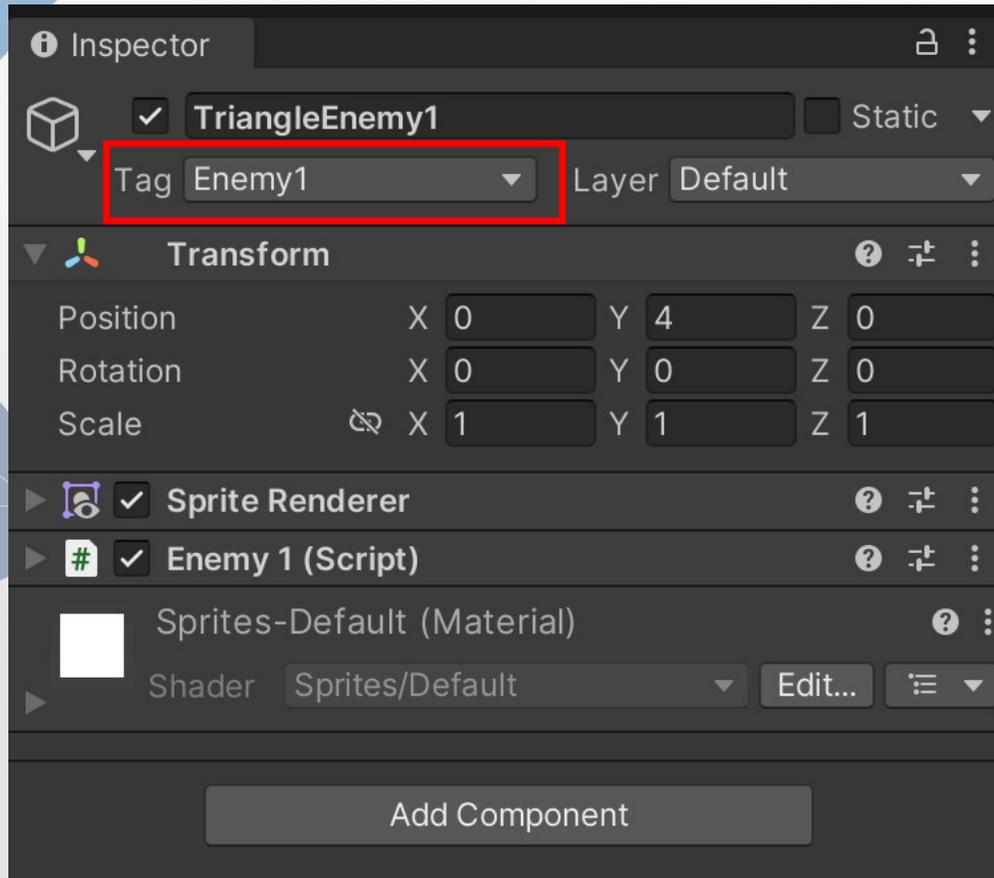
0. 前準備

ファイルを指定し、以下の画面が出たらImportを押すと、ファイルが展開されます。



0. 前準備

以下のようにタグを設定してください。



1. インターフェースとは

Interface : 境界面、接点

⇒プログラムでは2つのものをつなぐ仲介的な存在のこと。

<よく聞く言葉>

UI(ユーザーインターフェース)

プログラムと人間をつなぐ仲介役のこと。

⇒プログラム内のHPをテキストで人間に伝える

⇒次のシーンに移りたいからボタンを押すことでプログラムに伝える

GUI/CUI

⇒デスクトップとコマンドライン

1. インターフェースとは

C#におけるインターフェースとは？

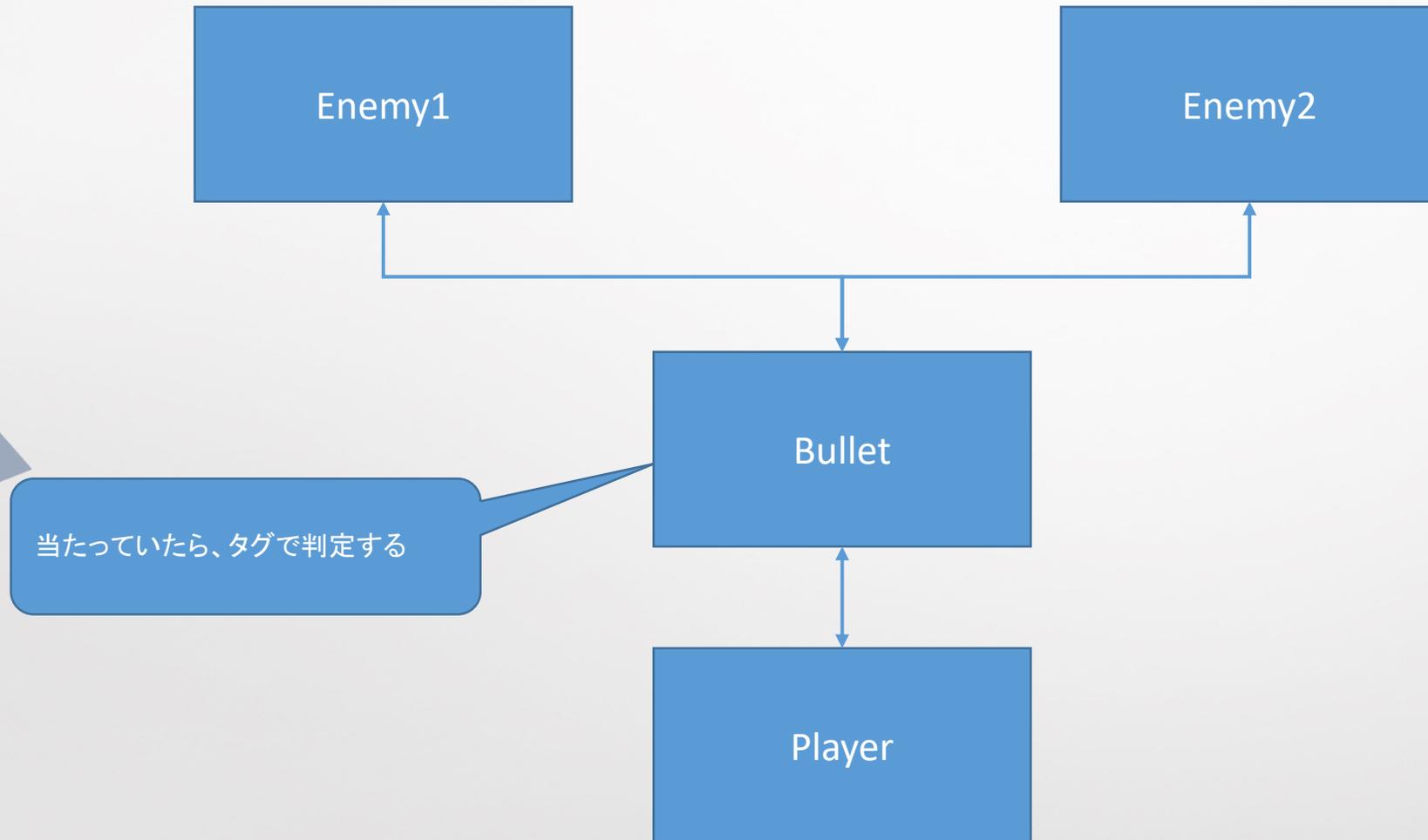
⇒クラスが実装すべきメンバーを定義する、いわゆる契約のような存在。

言葉だけで理解するのが難しいため実践してみましよう、Lessonのシーンを開いてください。

□がプレイヤーです。Wを押すと上のEnemy1クラスの敵に弾を発射します。
Dを押すと右のEnemy2クラスの敵に弾を発射します。
弾が当たった際に、Enemy1とEnemy2の「health」の変数に対して
マイナス1をして、0になったら敵をDestroyするプログラムを作成してください。

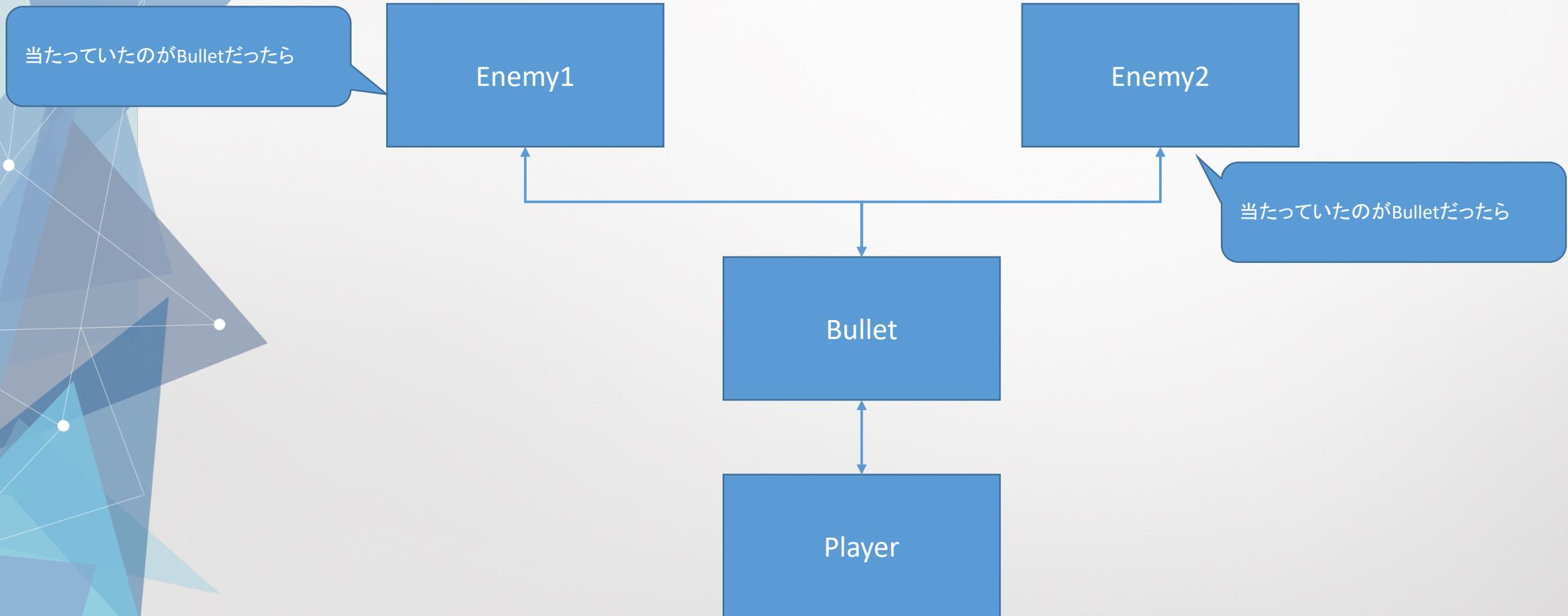
2. Unityにおける使い方

この場合、敵の種類が増えたら、Bulletが肥大化してしまう。。



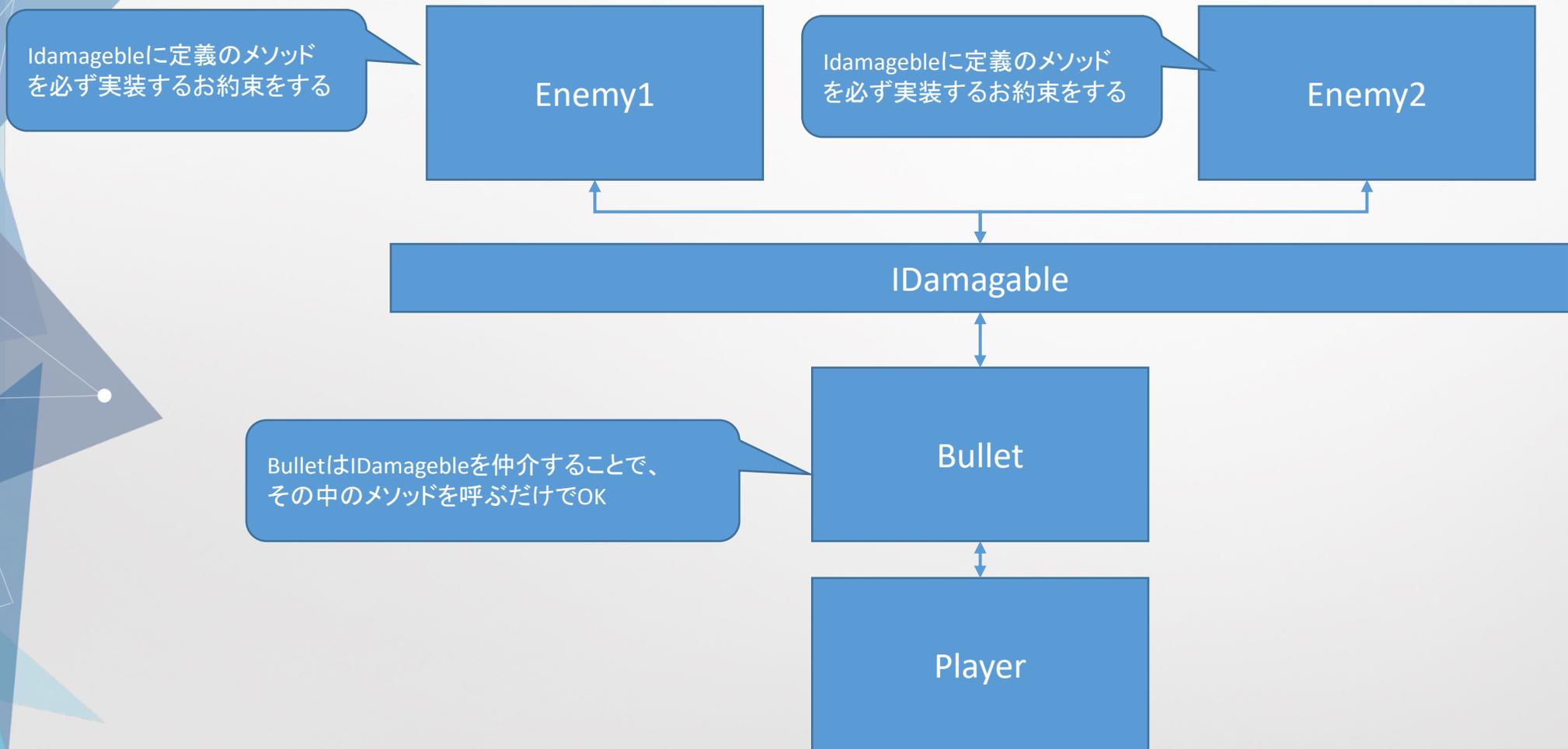
2. Unityにおける使い方

この場合、敵が増えたときに同じようなプログラムが増えてメンテが大変。。



2. Unityにおける使い方

インターフェースを使うと、以下のような実装が出来ます！



2. Unityにおける使い方

```
public interface I名前  
{  
    戻り値 メソッド名();  
}
```

上記のように派生先で必要になるメンバのみを宣言でき、継承することを前提とした型になります。

インターフェースは継承を前提としているため、メソッドは全て抽象メソッドです。

派生先のクラスではインターフェースで宣言されたメンバを必ず定義する必要があり、またそれらにはpublicアクセスとする必要があります。

インターフェースは基底クラスとは異なり一つのクラスに複数個継承できます。

3. 問題

<問題>

オリジナルの敵、Enemy3とEnemy4を作成し、インターフェースを継承して実装してみてください。