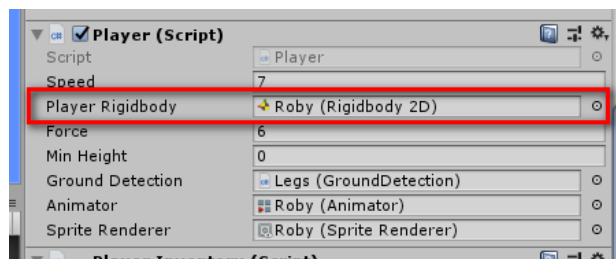


Дуже багато помилок у програмі відбувається через забування програмістом заповнити public поля відповідними об'єктами та компонентами. виправити це можна за допомогою порівняння з null.

Найчастіше подібні перевірки роблять у методі Start, щоб на момент безпосереднього звернення до змінних з посиланнями точно не виникло помилки.

Подібні перевірки не потрібно робити для примітивних публічних змінних (int, float і т.д.). Вони за замовчуванням вже дорівнюють нулю.

Наприклад, у скрипті Player є публічне поле Rigidbody2D



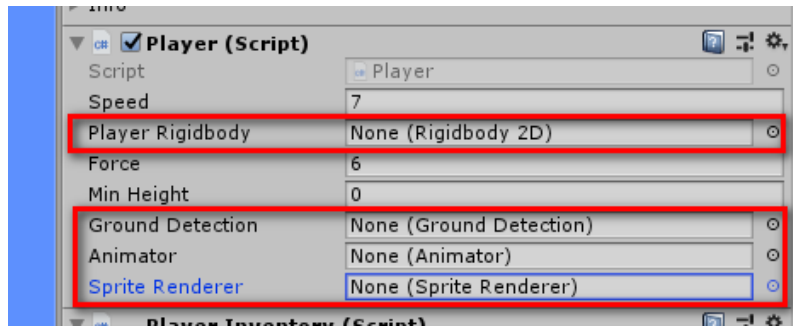
Для перевірки заповненості цього поля методом start робимо таку перевірку:

```
private void Start()
{
    if (playerRigidbody == null)
    {
        playerRigidbody = GetComponent<Rigidbody2D>();
    }
}
```

Метод GetComponent споживає більше ресурсів процесора, однак у цій ситуації є гарним вирішенням проблеми!

Пропишіть всі перевірки так, щоб навіть при подібному стані компонента Player гра працювала без помилок:



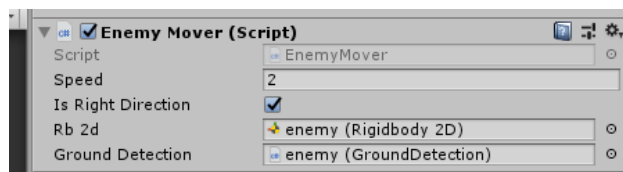


П.С. Якщо в майбутньому вам знадобиться отримати доступ до компонента, що знаходиться у вкладеному об'єкті, зробити це можна за допомогою методу GroundDetection. Наприклад:

```
GetComponentInChildren<GroundDetection>()
```

У скрипті EnemyMover зараз зчитування компонента GroundDetection проводиться за замовчуванням через спосіб Start. Переробіть це рішення так, щоб GroundDetection стало публічним полем. Реалізуйте захист від помилки із заповненням поля у методі старт після перевірки його відсутності!

Те саме зробіть із змінною типу Rigidbody2D. Повинно вийти приблизно так:



Переконайтеся, що ворог і персонаж перебувають у префабах і в них збережені всі зміни!

