

Давайте реалізуємо для гравця здоров'я! Створіть скрипт Health і застосуйте до Альфи

```
public int health;      //Здоровье объекта

//Метод для добавления здоровья
0 references
public void AddHealth(int add)
{
    health += add;

    //Корректируем здоровье, если его стало больше 100
    if (health > 100)
    {
        health = 100;
    }
}

//Метод вычитания здоровья
0 references
public void SubstractHealth(int subst)
{
    health -= subst;

    //Если здоровья меньше нуля - проигрыш
    if (health <= 0)
    {
        //В дальнейшем это замениться на загрузку
        //сцены с информацией о проигрыше
        Destroy(gameObject);
    }
}
```

В інспекторі встановлюємо здоров'я у 100!

Тепер необхідно реалізувати завдання шкоди роботі небезпечними ігровими об'єктами. Для початку напишемо скрипт DamageDetected



```

public int damage;           //Размер урона
public string damageTargetTag; //Тег объекта, которому должен наноситься урон
public bool resetPosition = false; //Если урон наносится игроку, определяют должен ли он
                                //вернуться в сохраненную позицию

Unity Message | 0 references
private void OnCollisionEnter2D(Collision2D col)
{
    //Проверяем прикосновение к объекту с заданным тегом
    if (col.gameObject.CompareTag(damageTargetTag))
    {
        //Считываем и сохраняем его компонент здоровья
        Health health = col.gameObject.GetComponent<Health>();
        health.SubstractHealth(damage); //Вычитаем нужное количество здоровья

        //Для некоторых объектов (например для падения вводу) нужно сбросить позицию игрока
        if (resetPosition)
        {
            //В домашнем задании функция Reload Position была объявлена как приватная
            //Пока вы не сделаете её публичной, доступа к ней не будет!
            col.gameObject.GetComponent<ReloadStartPosition>().ReloadPosition();
        }
    }
}

```

Застосовуємо скрипт до ворога та налаштовуємо будь-який бажаний рівень шкоди. Змінну `resetPosition` не чіпаємо. Зберігаємо зміни до префабу!

Застосовуємо скрипт до префабу платформи із шипами. Налаштовуємо бажаний рівень втрат. Змінну `resetPosition` не чіпаємо. Зверніть увагу, що якщо ця платформа тепер містить скрипт з методом `onCollision`, їй потрібно додати компонент `Rigidbody2D` з налаштуваннями статички.

Застосовуємо скрипт до префабу води. Налаштовуємо бажаний рівень втрат. Включаємо `resetPosition`. Зверніть увагу, що якщо ця платформа тепер містить скрипт з методом `onCollision`, їй потрібно додати компонент `Rigidbody2D` з налаштуваннями статички.

Запускаємо гру та перевіряємо правильність роботи скрипту!

