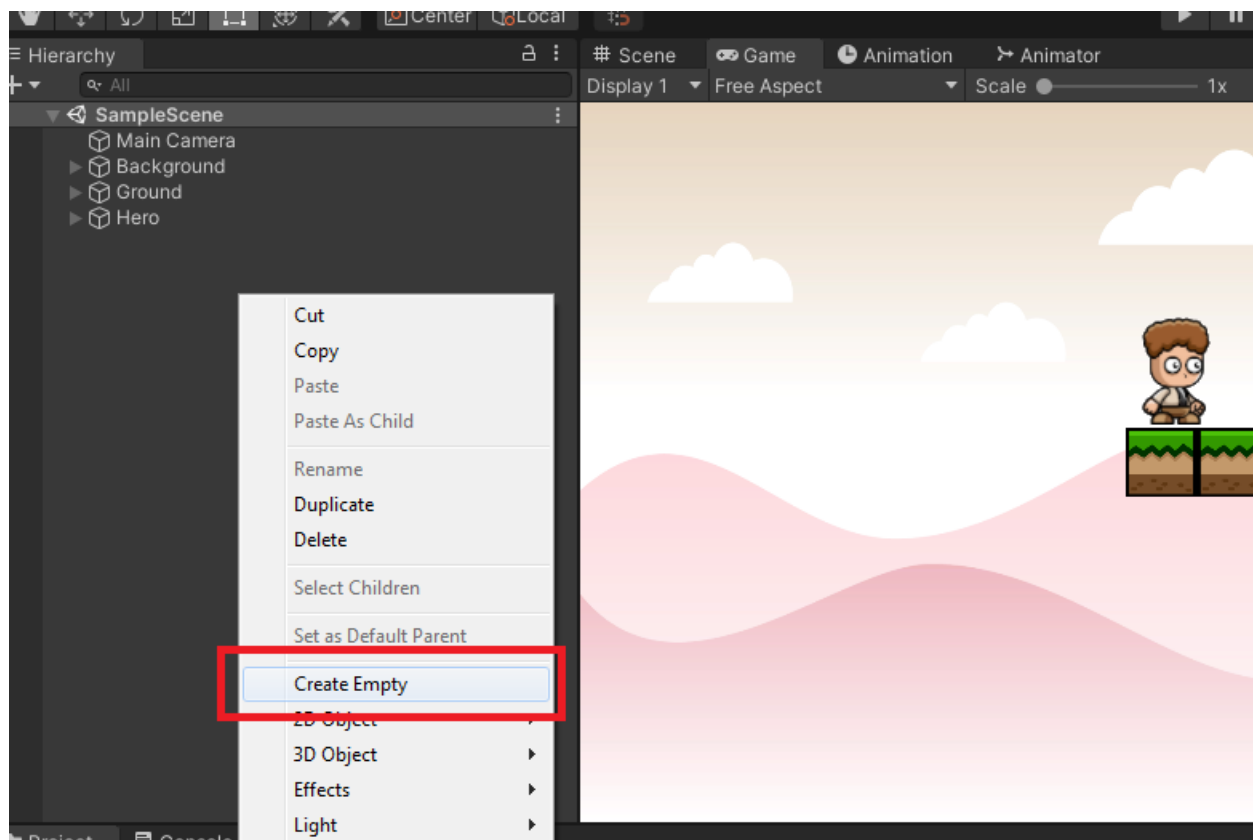


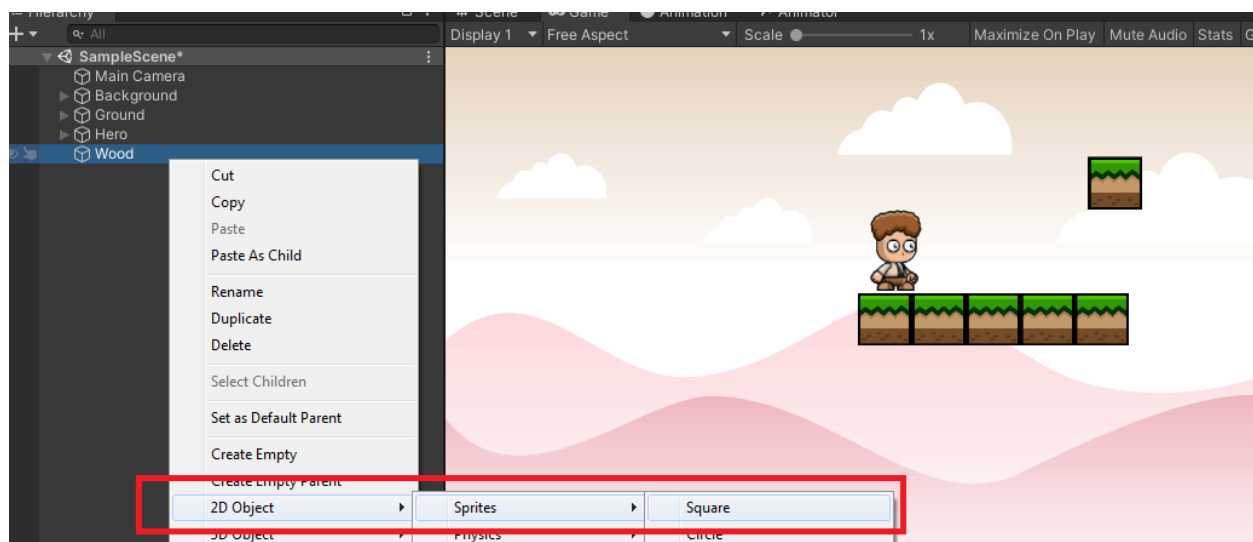
Лабораторная 4. Платформер. Враги

1. Создание неподвижного врага

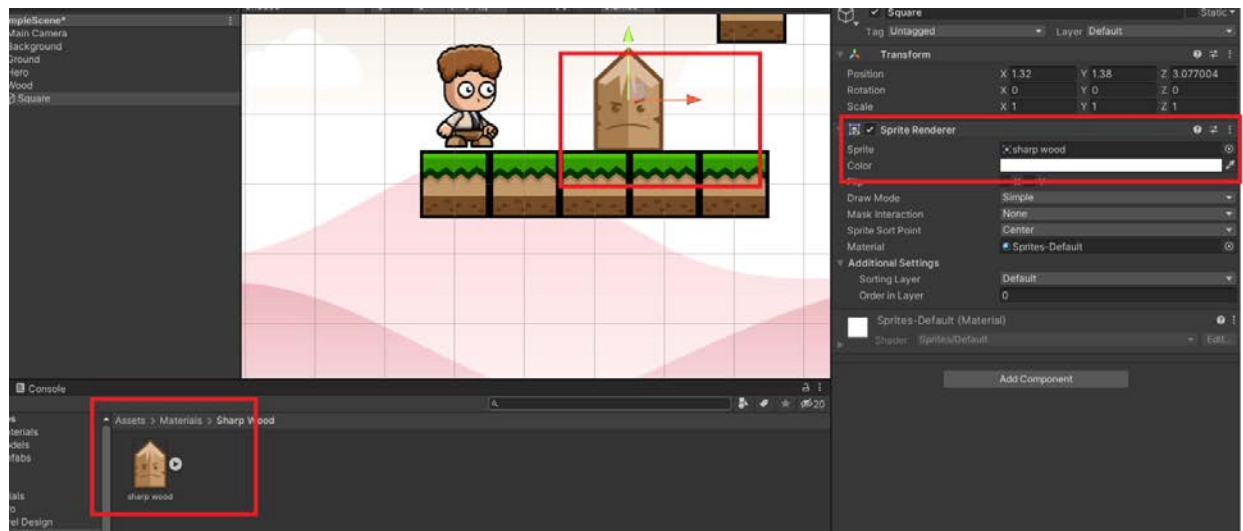
Создаем пустой объект



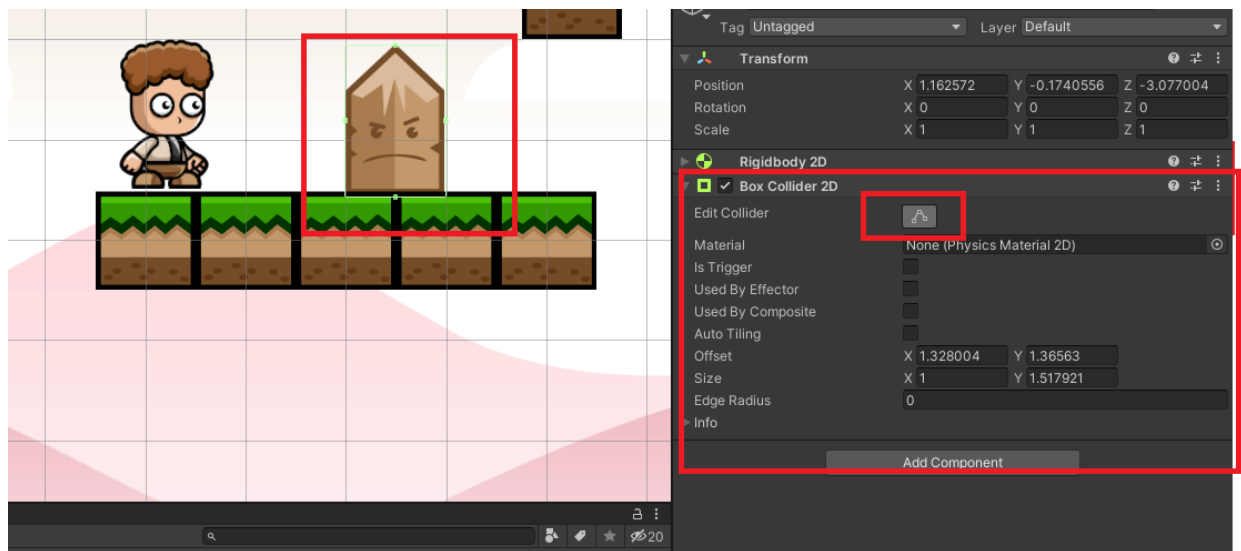
Переименовываем и добавляем спрайт



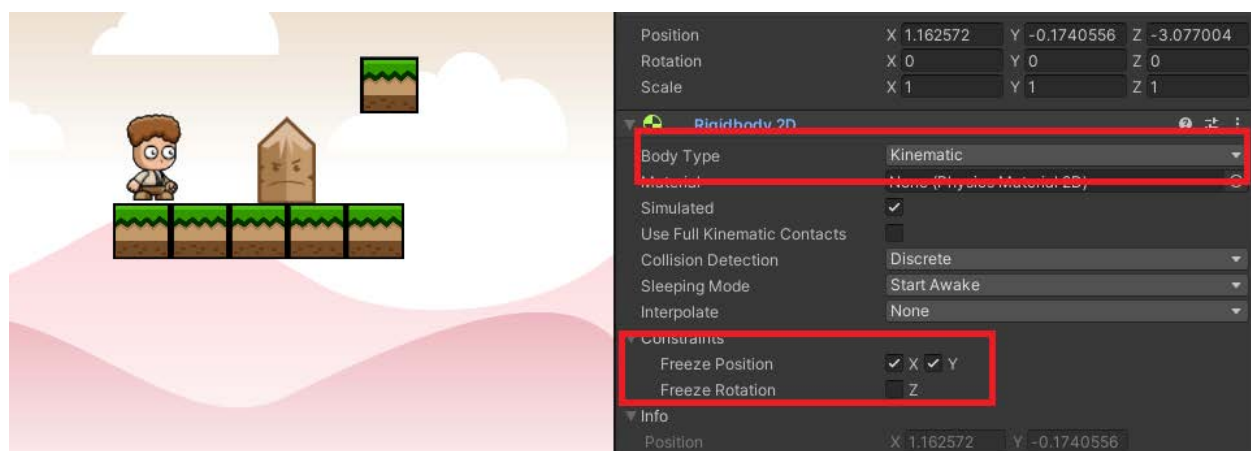
Добавляем картинку на объект



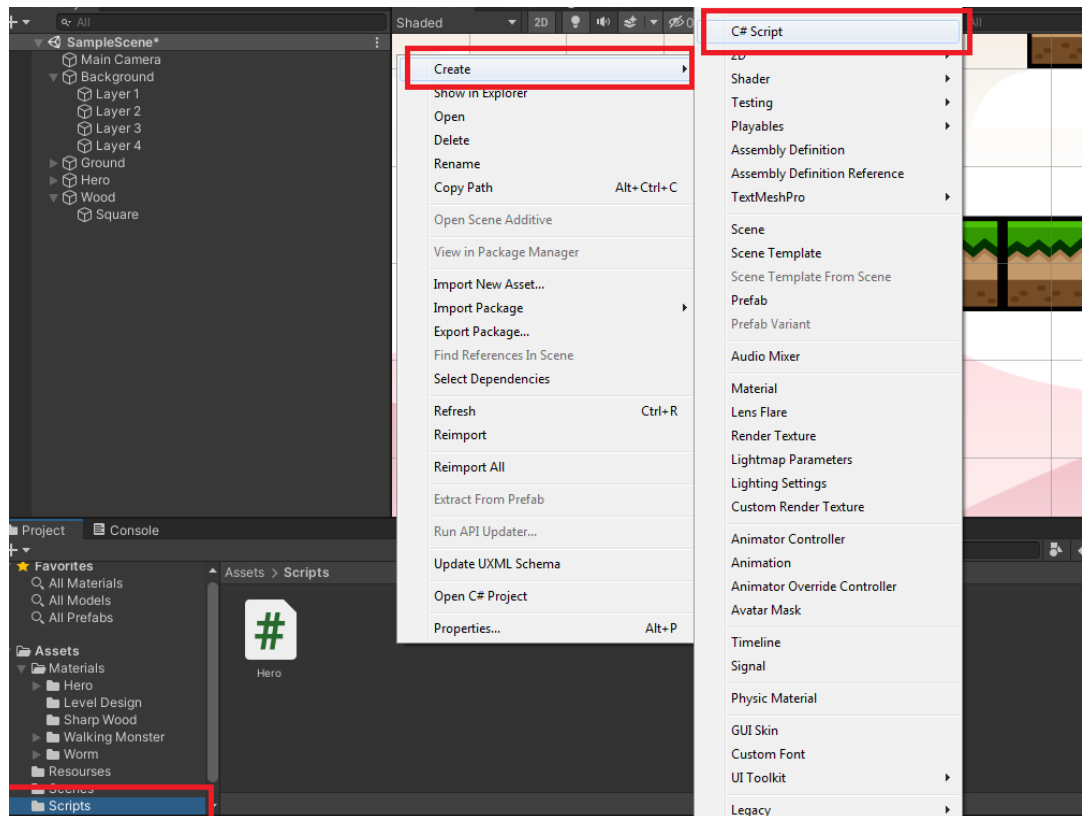
Добавляем Box Collider (на забываем уточнить место Box Collider)



Добавляем Rigidbody и Замораживаем координаты

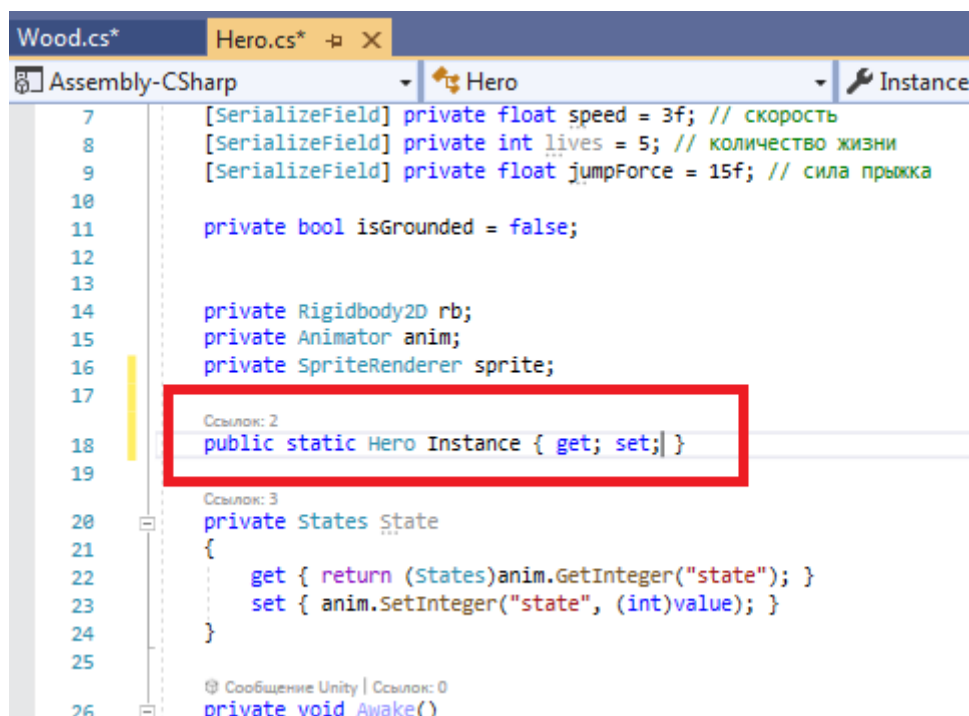


Создаем скрипт



Переходим к написанию скриптов.

вносим изменения в скрипт Hero - данное изменение позволит обращаться к данным класса не создавая его экземпляра (паттерн Singleton)



```

Ссылка: 3
public static Hero Instance { get; set; }

Ссылка: 3
private States state
{
    get { return (States)anim.GetInteger("state"); }
    set { anim.SetInteger("state", (int)value); }
}

Сообщение Unity | Ссылка: 0
private void Awake()
{
    sprite = GetComponentInChildren<SpriteRenderer>(); //обращение к спрайту
    anim = GetComponent<Animator>();
    rb = GetComponent<Rigidbody2D>();
    Instance = this;
}

ссылка: 1
54 private void Jump()...
58
ссылка: 1
59 private void CheckGround()...
65
66 ссылка: 1
67 public void GetDamage()
68 {
69     lives--;
70     Debug.Log(lives);
71 }
72

Ссылка: 5
73 public enum States
74 {
75     IDLE,
76     run,
77     jump
78 }

```

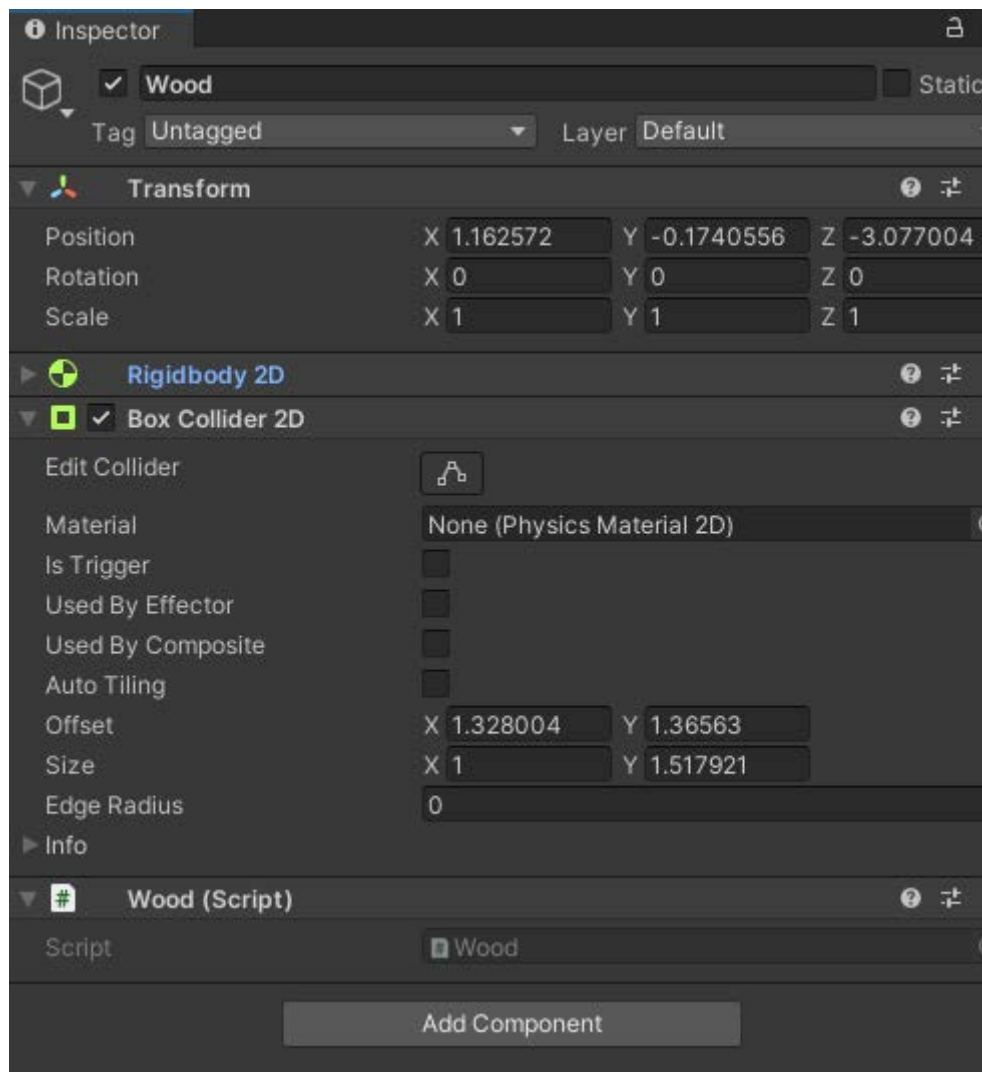
Создаем скрипт неподвижного врага

```

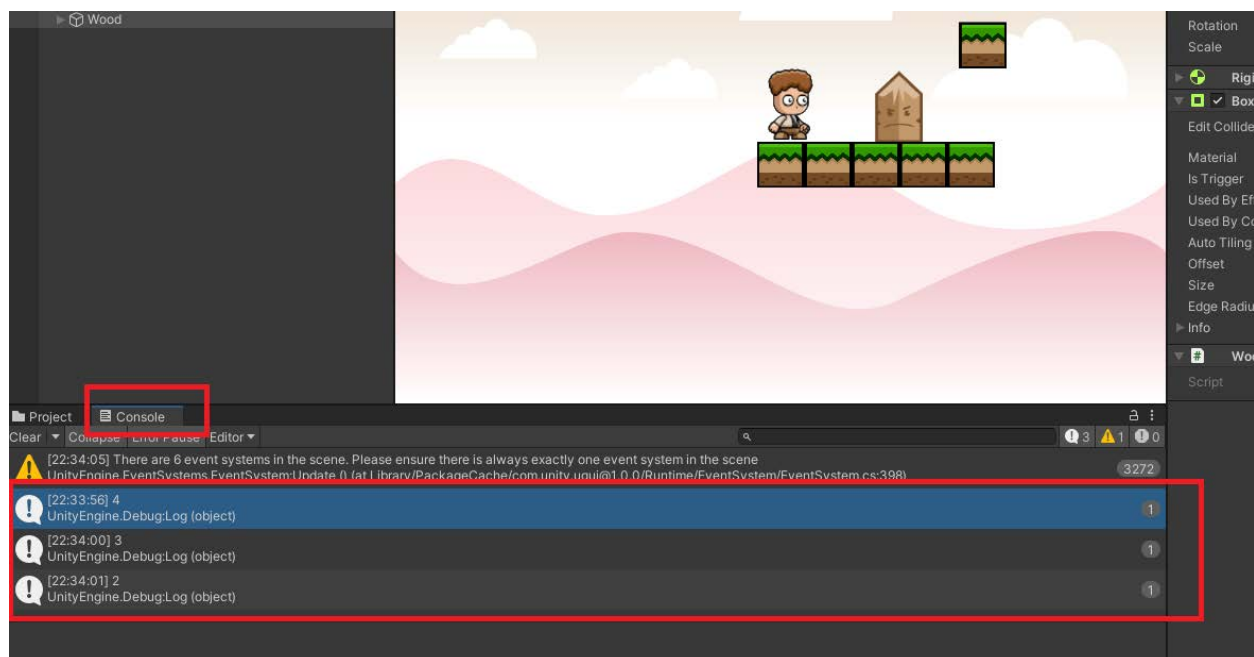
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 Сообщение Unity | Ссылка: 0
6 public class Wood : MonoBehaviour
7 {
8     Сообщение Unity | Ссылка: 0
9     private void OnCollisionEnter2D(Collision2D collision)
10    {
11        if (collision.gameObject == Hero.Instance.gameObject)
12        {
13            Hero.Instance.GetDamage();
14        }
15    }
16 }

```

Добавляем скрипт на объект



Проверяем работу - взаимодействия героя и неподвижного врага



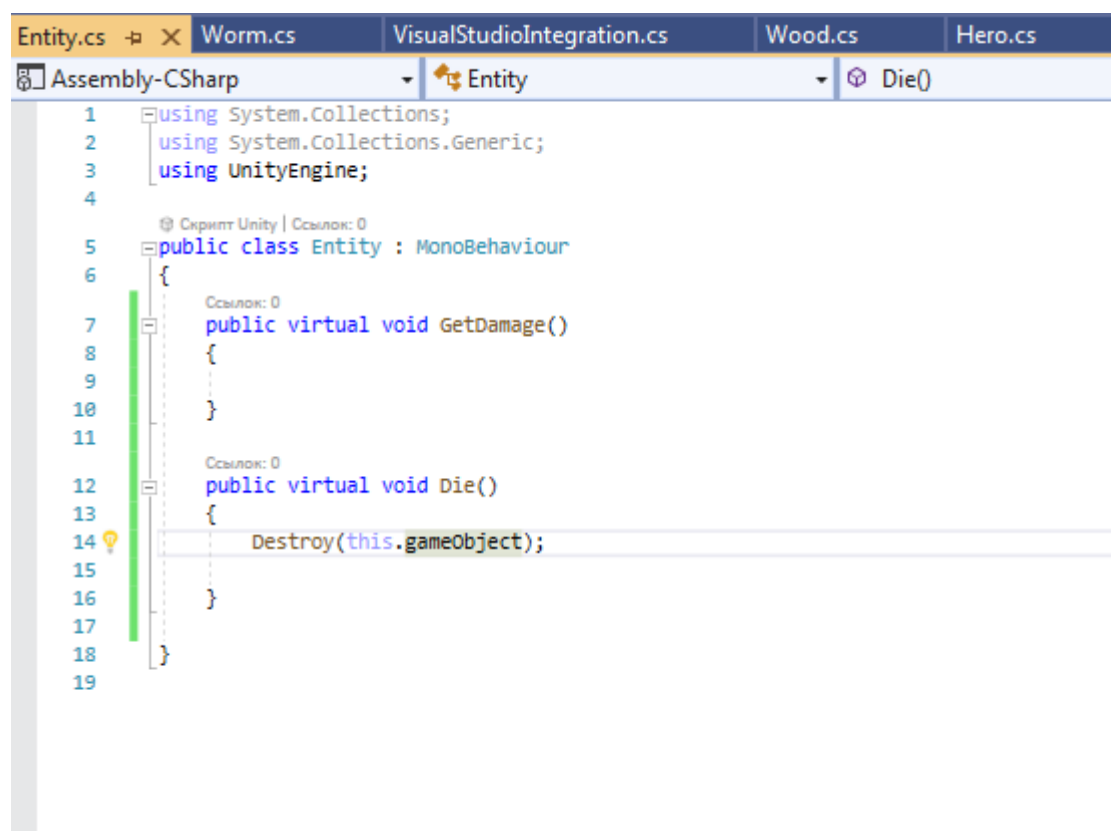
2. Анимированный неподвижный враг

Самостоятельно сделаем

1. Создаем так же как неподвижного врага
2. Добавляем анимацию
3. Добавляем скрипт аналогично как у неподвижного врага

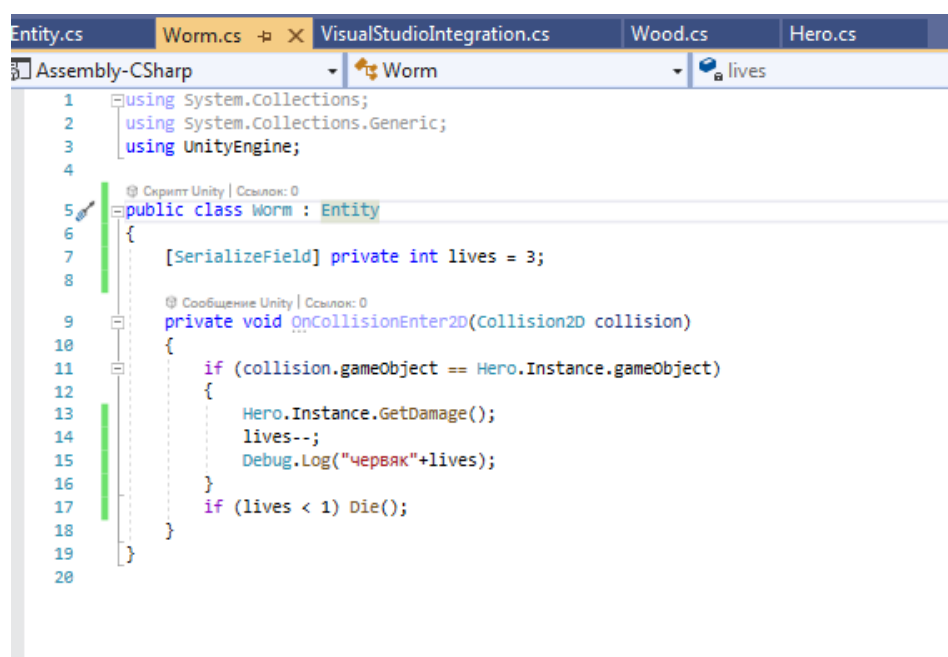
Продолжаем модифицировать код

Создаем новый скрипт - В котором создаем класс для ВСЕХ
сущностей



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Entity : MonoBehaviour
6 {
7     public virtual void GetDamage()
8     {
9     }
10
11     public virtual void Die()
12     {
13         Destroy(this.gameObject);
14     }
15 }
16
17
18
19
```

Переписываем скрипты



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Worm : Entity
6 {
7     [SerializeField] private int lives = 3;
8
9     private void OnCollisionEnter2D(Collision2D collision)
10     {
11         if (collision.gameObject == Hero.Instance.gameObject)
12         {
13             Hero.Instance.GetDamage();
14             lives--;
15             Debug.Log("червяк"+lives);
16         }
17         if (lives < 1) Die();
18     }
19 }
20
```

у Hero

```
59     private void CheckGround()[...]  
65  
66     public override void GetDamage()  
67     {  
68         lives--;  
69         Debug.Log(lives);  
70     }  
71 }  
72  
Ссылка: 5  
public class Hero
```

```
3 using UnityEngine;  
4  
5 public class Hero : Entity  
6 {  
7     [SerializeField] private float speed = 3f; // скорость  
8     [SerializeField] private int lives = 5; // количество жизни  
9     [SerializeField] private float jumpForce = 15f; // сила прыжка  
10  
11     private bool isGrounded = false;  
12  
13  
14     private Rigidbody2D rb;  
15     private Animator anim;  
16     private SpriteRenderer sprite;
```

3. Создаем подвижного врага

1. Создаем аналогично, как делали второй вариант. Добавляем Box Collider (на забываем уточнить место Box Collider), Добавляем Rigidbody и Замораживаем координаты
2. Пишем скрипт – основное отличие функция Move () – которая перемещает монстра


```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  ❏ Скрипт Unity | Ссылок: 0
6  public class WalkingMonster : Entity
7  {
8      private float speed = 3.5f;
9      private Vector3 dir;
10     private SpriteRenderer sprite;
11
12     ❏ Сообщение Unity | Ссылок: 0
13     private void Start()
14     {
15         dir = transform.right;
16     }
17
18     ❏ Сообщение Unity | Ссылок: 0
19     protected void Update()
20     {
21         Move();
22     }
23
24     ❏ Сообщение Unity | Ссылок: 0
25     protected void Awake()
26     {
27         sprite = GetComponentInChildren<SpriteRenderer>();
28     }
29
30     ссылка: 1
31     private void Move()
32     {
33         Collider2D[] colliders = Physics2D.OverlapCircleAll(transform.position + transform.up * 0.1f + transform.right * dir.x * 0.7f, 0.01f);
34
35         if (colliders.Length > 0 ) dir *= -1f;
36
37         transform.position = Vector3.MoveTowards(transform.position, transform.position + dir, Time.deltaTime);
38     }
39
40     ❏ Сообщение Unity | Ссылок: 0
41     private void OnCollisionEnter2D(Collision2D collision)
42     {
43         if (collision.gameObject == Hero.Instance.gameObject)
44         {
45             Hero.Instance.GetDamage();
46         }
47     }
48 }

```

САМОСТОЯТЕЛЬНО

Создать анимацию на перемещения