

МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

**Факультет Информационных технологий
Кафедра Информатики и информационных технологий**

**направление подготовки
09.03.02 «Информационные системы и технологии»**

ЛАБОРАТОРНАЯ (ПРАКТИЧЕСКАЯ РАБОТА)

Дисциплина: Разработка игровых приложений

Тема: Итоговый проект

Выполнил(а): студент(ка) группы 221-377

Егорова Елизавета Вячеславовна

(Фамилия И.О.)

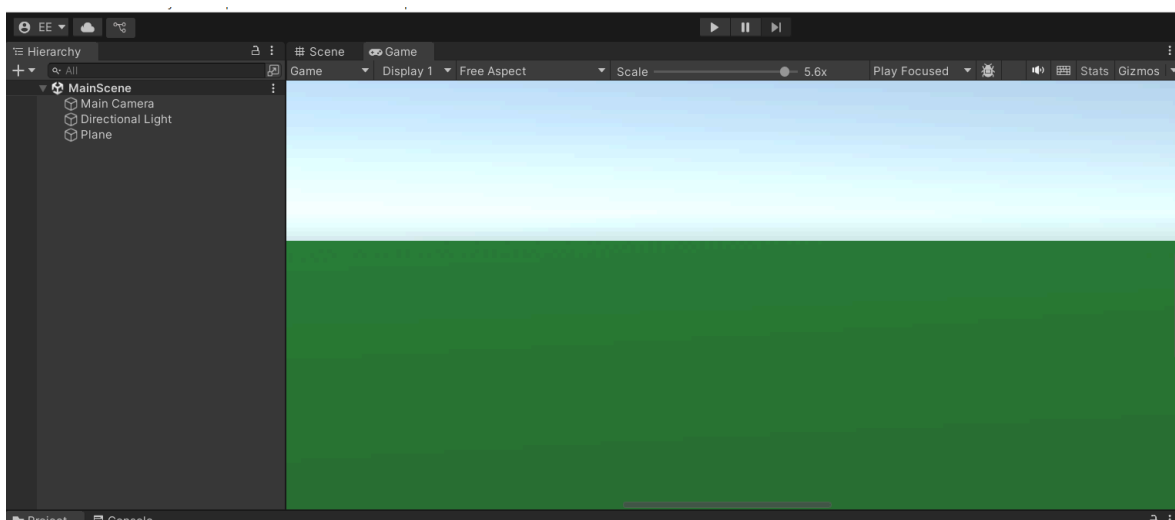
Дата, подпись _____
(Дата) (Подпись)

Проверил: _____
(Фамилия И.О., степень, звание) (Оценка)

Дата, подпись _____
(Дата) (Подпись)

**Москва
2024**

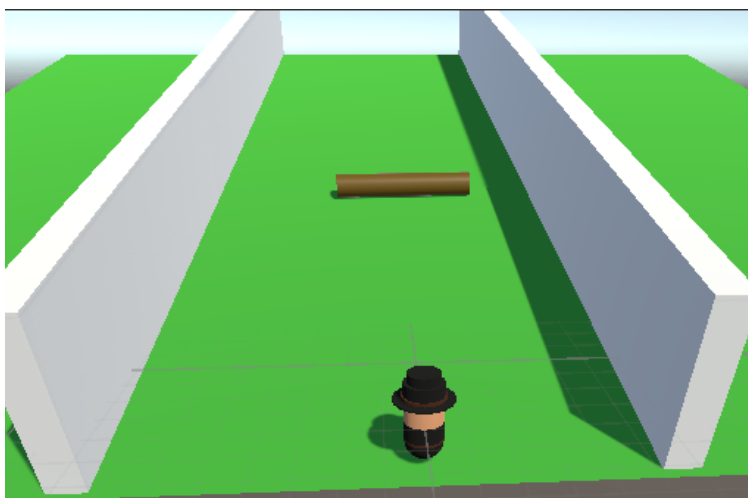
Создание поля:



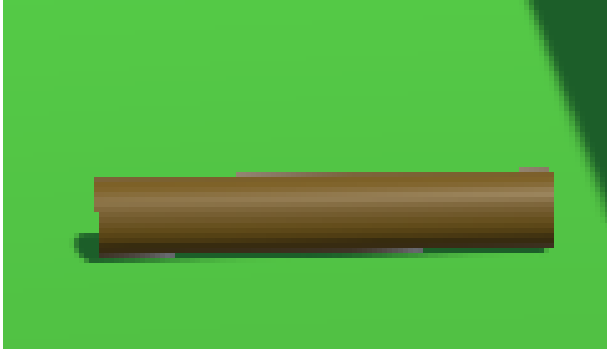
Главный герой:



Границы игрового поля:



Бревно:



Код спавна бревен:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Spawner : MonoBehaviour
{
    [SerializeField] GameObject Deck;
    float a = 1;
    float b = 3;

    private void Start()
    {
        StartCoroutine(Spawn());
    }

    IEnumerator Spawn()
    {
        while (true)
        {
            yield return new WaitForSeconds(Random.Range(a-=0.01f, b-=0.002f));

            GameObject NewDeck = Instantiate(Deck, new
            Vector3(Random.Range(-2.9f, 2.9f), 15, 10), Deck.transform.rotation);
```

```

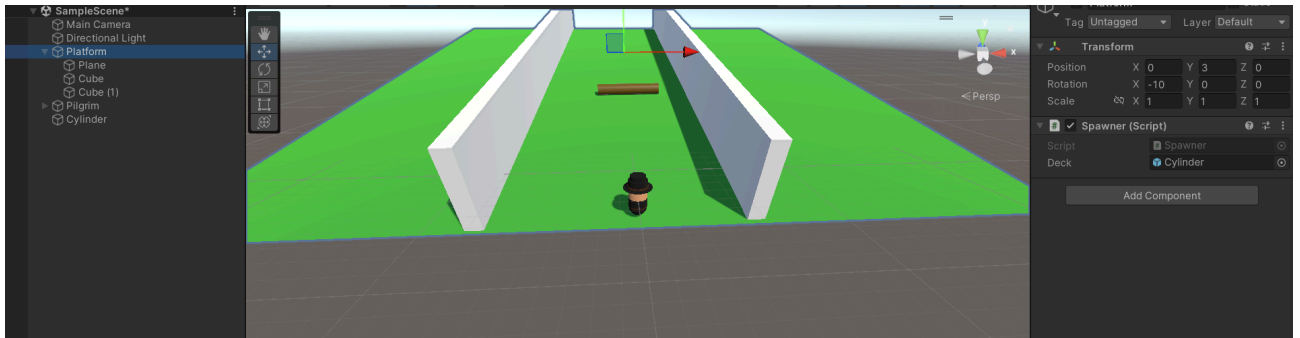
    }

}

}

```

Присвоение скрипта игровой платформе:



Скрипт движения игрока:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class PlayerCtrl : MonoBehaviour
{
    [SerializeField] float force;
    Vector3 Move = new Vector3(1, 0, 0);
    bool is_ground = false;
    Rigidbody rb;

    private void Awake()
    {
        rb = GetComponent<Rigidbody>();
    }

    private void FixedUpdate()

```

```
{  
    if (Input.GetKey(KeyCode.Space) && is_ground) {  
        rb.AddForce(Vector2.up * forse);  
    }  
    if (Input.GetKey(KeyCode.A) && is_ground)  
    {  
        rb.velocity -= Move;  
    }  
    if (Input.GetKey(KeyCode.D) && is_ground)  
    {  
        rb.velocity += Move;  
    }  
}
```

```
private void OnCollisionEnter(Collision collision)  
{  
    if (collision.collider.tag == "deck") {  
        SceneManager.LoadScene(0);  
    }  
}
```

```
private void OnCollisionStay(Collision collision)  
{  
    if (collision.collider.tag == "ground")  
    {  
        is_ground = true;  
    }  
}
```

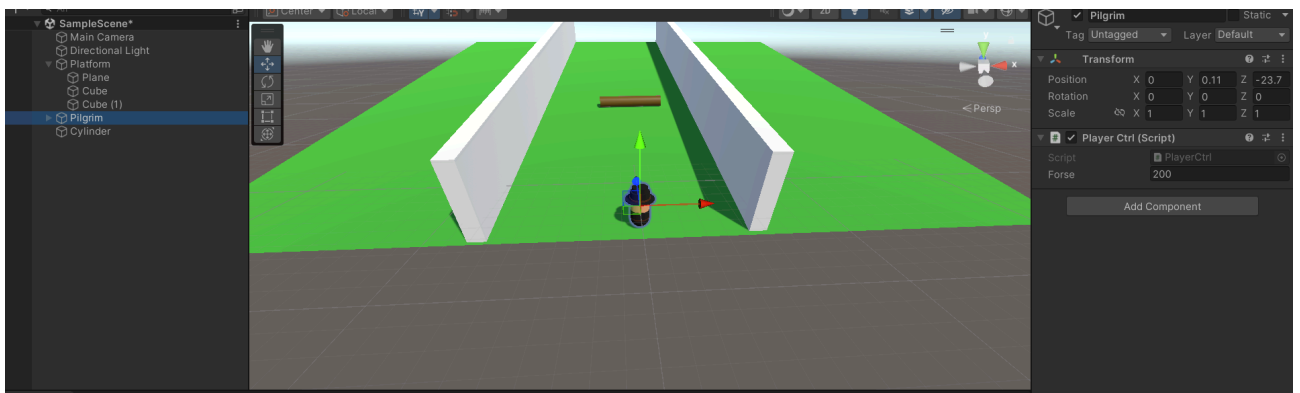
```
private void OnCollisionExit(Collision collision)
```

```

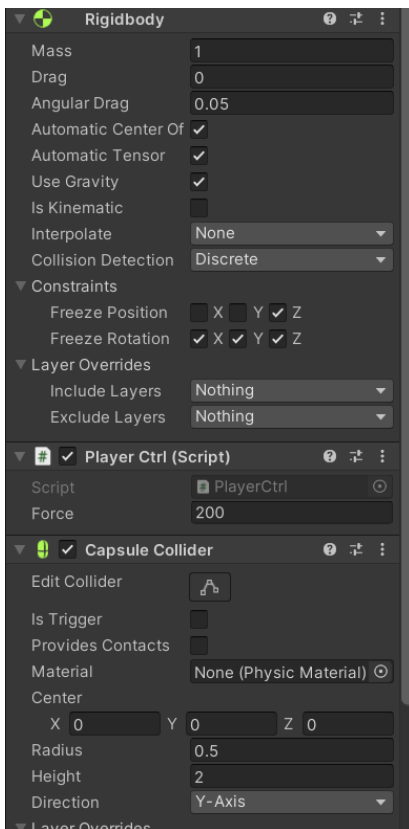
{
    if (collision.collider.tag == "ground")
    {
        is_ground = false;
    }
}
}

```

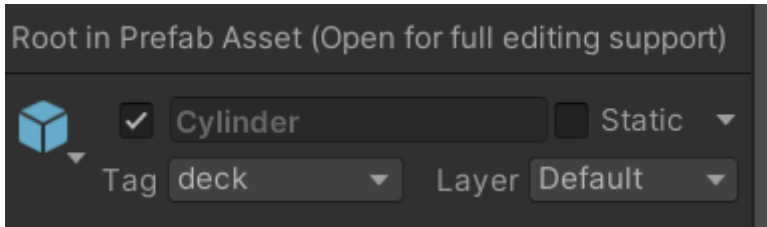
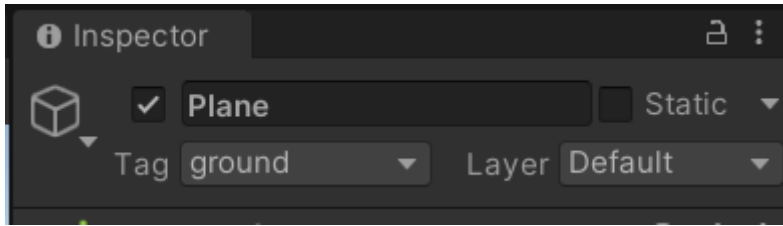
Присваивание скрипта игроку:



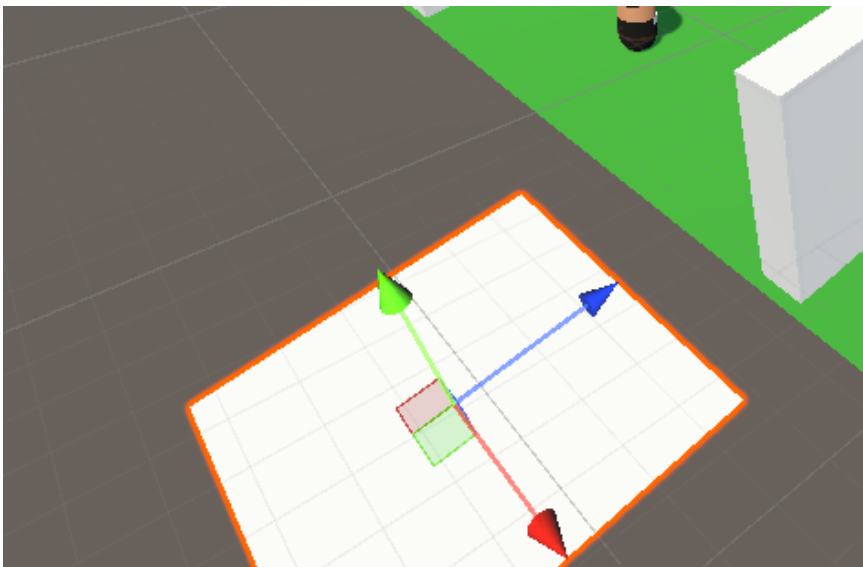
Настройки движения персонажа:



Создание тегов для бревнышка и площади:



Создаем платформу, на которые будут падать ребра:



Написание скрипта для удаления деревяшек из памяти:

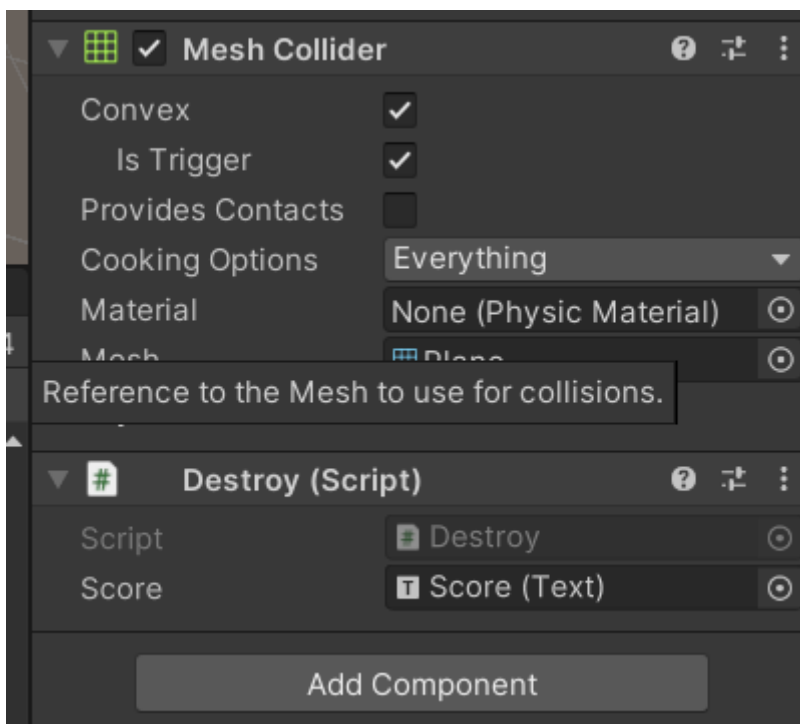
```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
using UnityEngine.UI;
```

```

public class Destroy : MonoBehaviour
{
    [SerializeField] Text Score;
    int score;

    private void OnTriggerEnter(Collider other)
    {
        if(other.tag == "deck")
        {
            Destroy(other.gameObject);
            score++;
            Score.text = score.ToString();
        }
    }
}

```



Итоговый результат:

