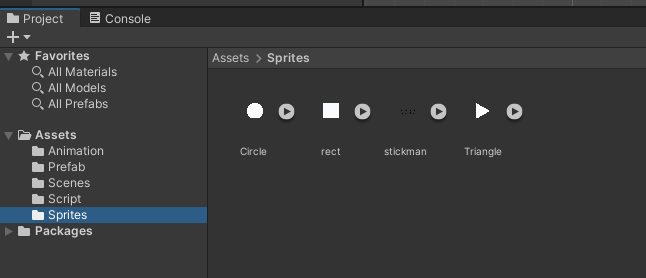
**Лабораторное занятие № 22.**

**Тема: Разработка игры “ Make Them Fall”**

**Цель: Разработать игру “ Make Them Fall”**

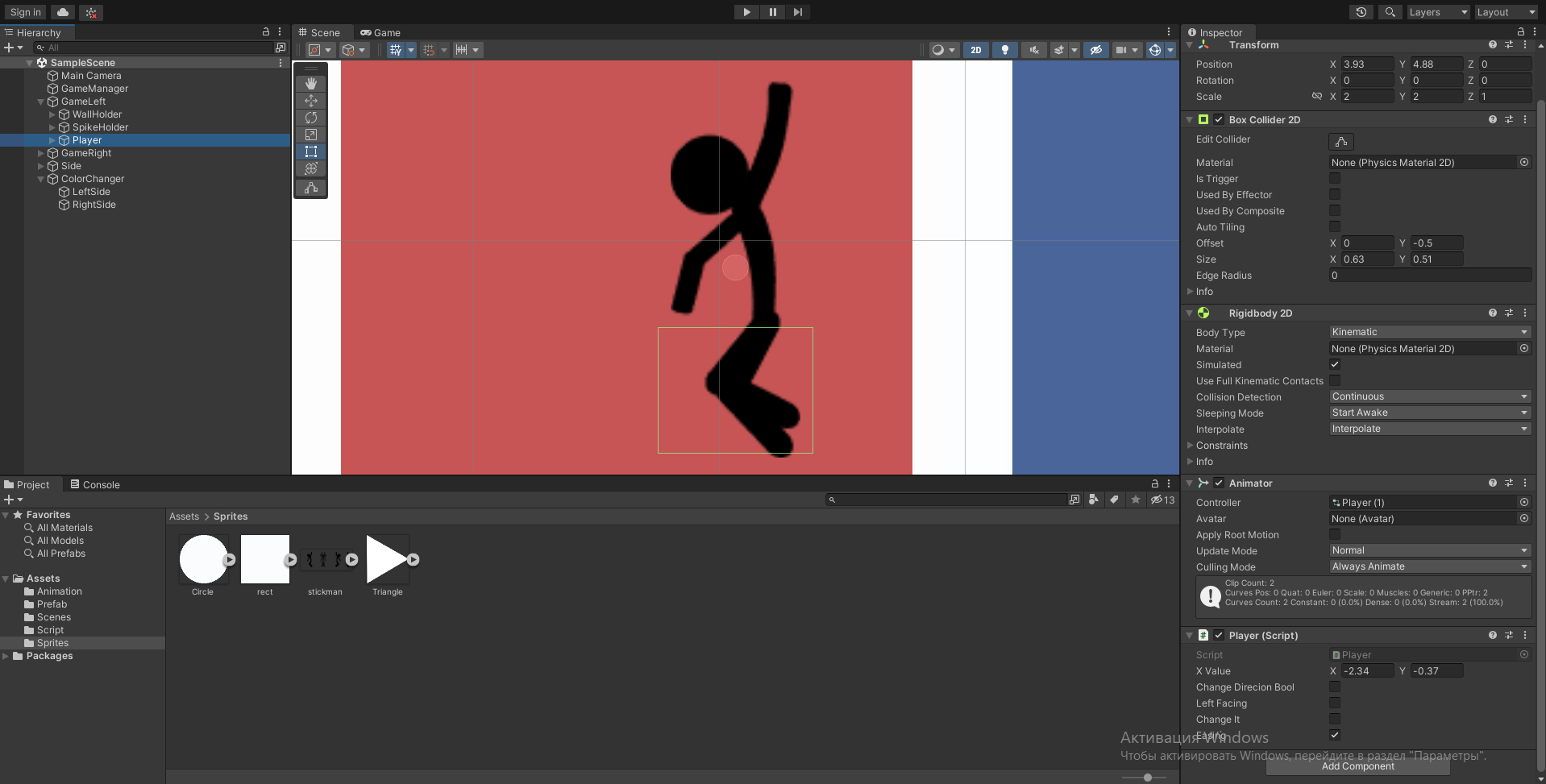
**Ход работы:**

1.Перекидываю ресусры (Sprites).



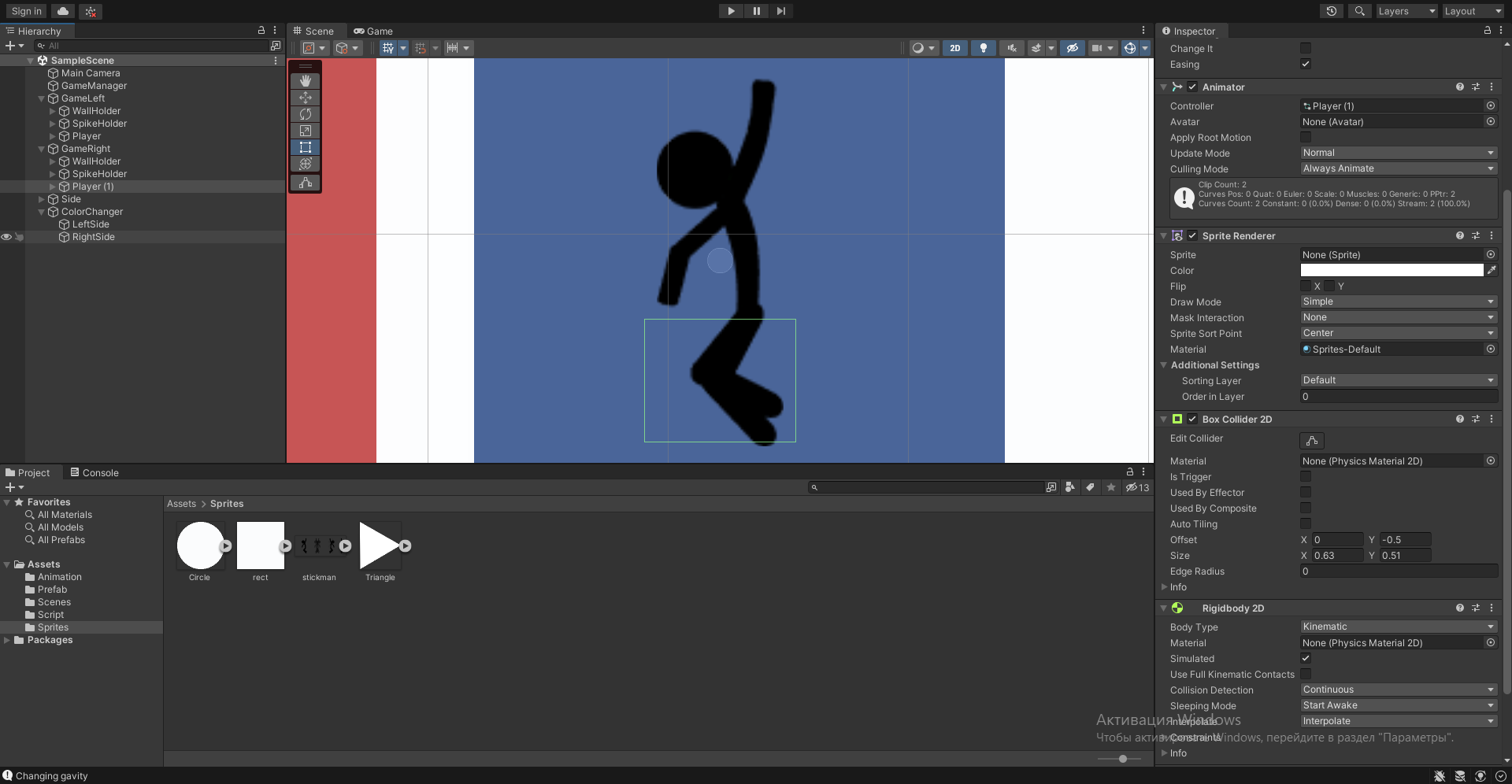
**Рис. 22.1** – Папка Sprites.

2.Создание игрока (Left Player).



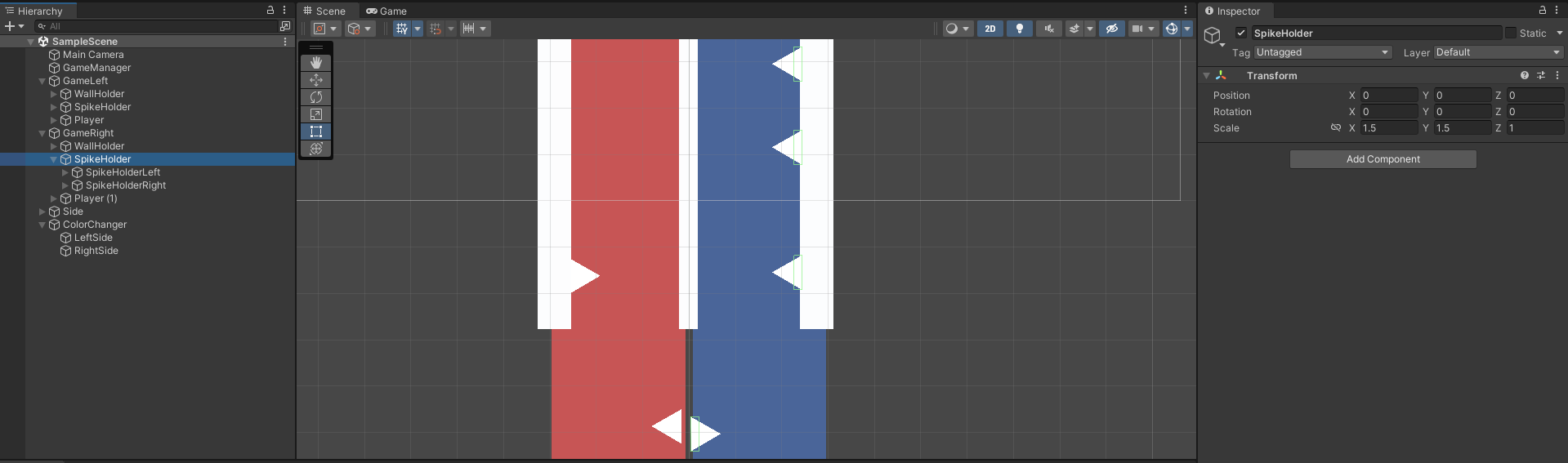
**Рис. 22.2** – Left Player.

3.Создание игрока (Right Player).



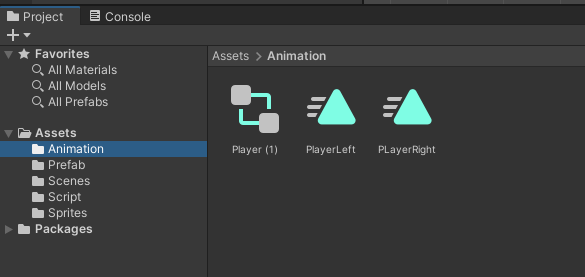
**Рис. 22.3** – Right Player.

4.Создание препятствий (Spike Holder).



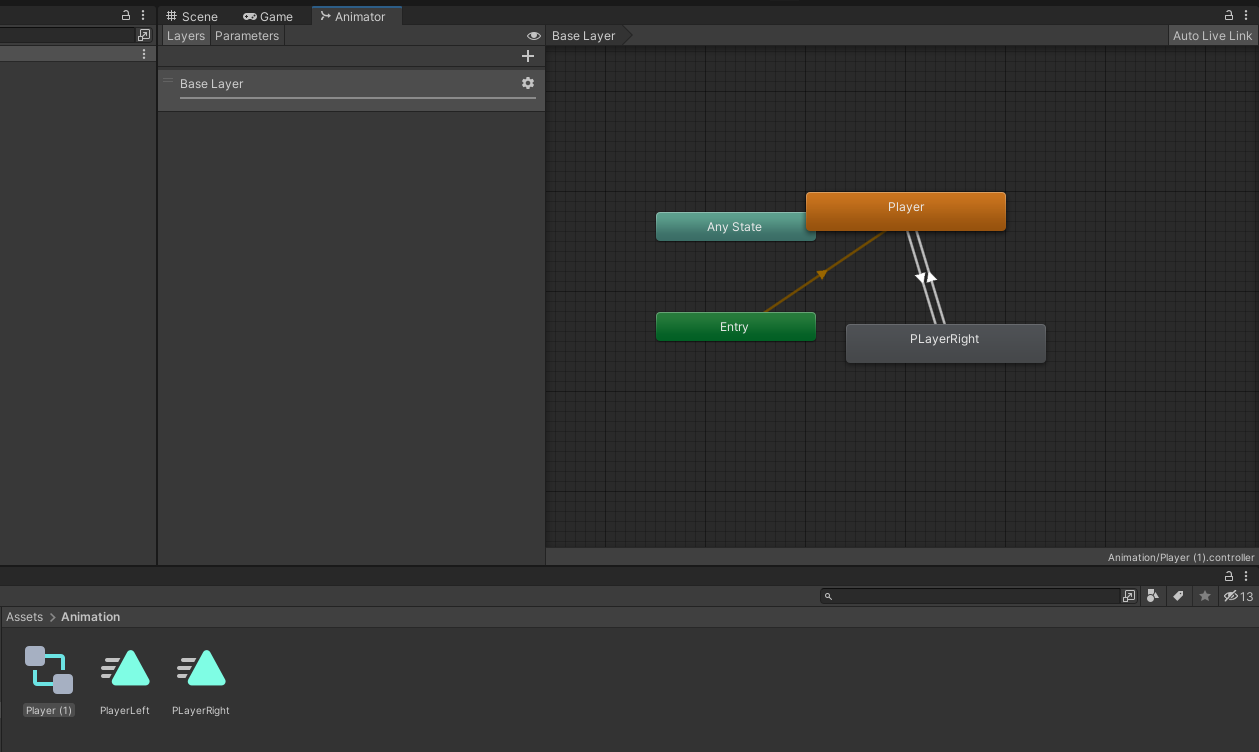
**Рис. 22.4** – Spike Holder.

5.Создание анимаций (Animation).



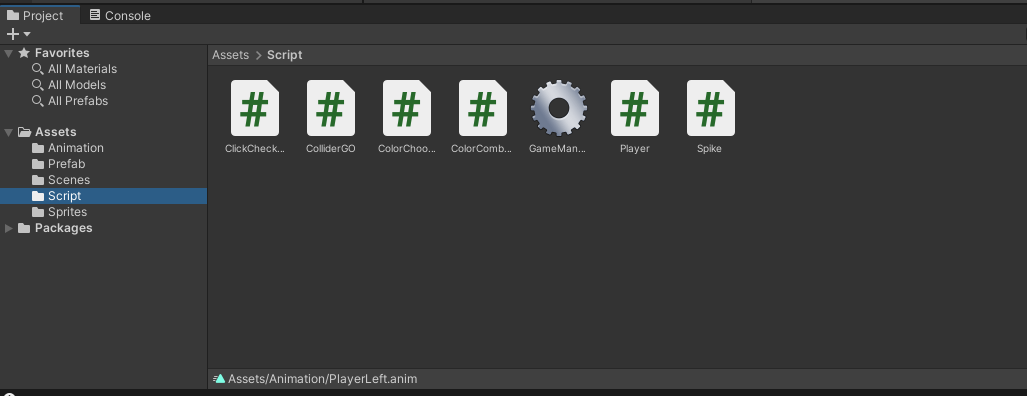
**Рис. 22.5** – Animation.

6.Создание анимации для игрока (Animation Player).



**Рис. 22.6** – Animation Player.

7.Создание Скриптов.



**Рис. 22.7** – Scripts.

**Скрипты:**

ClickChecker.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class ClickChecker : MonoBehaviour

{

void Update()

{

if (Input.GetMouseButtonDown(0))

{

Vector2 pos = Camera.main.ScreenToWorldPoint(Input.mousePosition);

RaycastHit2D hit = Physics2D.Raycast(pos, Vector2.zero);

if (hit)

{

if (hit.transform.name == "LeftSide")

{

hit.collider.GetComponent<ColliderGO>().CallPlayerMethod();

}

else if (hit.transform.name == "RightSide")

{

hit.collider.GetComponent<ColliderGO>().CallPlayerMethod();

}

else { Debug.Log("Unknown hit from click Checker"); return; }

}

}

}

}

ColliderGO.cs

using UnityEngine;

public class ColliderGO : MonoBehaviour

{

public Transform PlayerTransform;

public void CallPlayerMethod()

{

Debug.Log(PlayerTransform.name);

PlayerTransform.GetComponent<Player>().CallThisMethod();

}

}

CollorChooser.cs

using UnityEngine;

public class ColorChooser : MonoBehaviour

{

public GameObject[] imagesGO;

public ColorCombinatoin[] chooseColor;

void Awake()

{

int randomNo = Random.Range(0, chooseColor.Length );

imagesGO[0].GetComponent<SpriteRenderer>().color = chooseColor[randomNo].ColorToChoose[0];

imagesGO[1].GetComponent<SpriteRenderer>().color = chooseColor[randomNo].ColorToChoose[1];

}

}

ColorCombination.cs

using UnityEngine;

[System.Serializable]

public class ColorCombinatoin

{

public Color[] ColorToChoose;

}

GameManager.cs

using UnityEngine.SceneManagement;

using UnityEngine;

public class GameManager : MonoBehaviour

{

public void RestartScene()

{

SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);

}

}

Player.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Player : MonoBehaviour

{

public Vector2 xValue ;

public bool changeDirecionBool, leftFacing, changeIt,easing;

Animator animator;

Vector3 desiredPosition;

// Start is called before the first frame update

void Start()

{

DesiredPosition();

changeDirection();

animator = GetComponent<Animator>();

}

// Update is called once per frame

void Update()

{

if (changeIt)

{

if (easing)

{

transform.position = Vector3.Lerp(transform.position, desiredPosition, .1f);

}

else

{

transform.position = desiredPosition;

}

if (Mathf.Abs(desiredPosition.x - transform.position.x) <= 0.1f)

{

Debug.Log("Changing gavity");

changeIt = false;

}

}

animator.SetBool("ChangeDirection", changeDirecionBool);

}

public void CallThisMethod()

{

DesiredPosition();

changeDirection();

}

public void DesiredPosition()

{

desiredPosition = new Vector2(changeDirecionBool ? xValue.x : xValue.y, transform.position.y);

}

void changeDirection()

{

changeDirecionBool = !changeDirecionBool;

// transform.position = Vector3.Lerp(transform.position,desiredPosition,.1f);

if (changeDirecionBool != leftFacing)

{

changeIt = true;

leftFacing = changeDirecionBool;

}

Debug.Log(transform.position.x);

}

public void OnTriggerEnter2D(Collider2D collision)

{

if (collision.tag == "Spike")

{

Debug.Log("dead Player");

FindObjectOfType<GameManager>().RestartScene();

}

}

}

Spike.cs

using UnityEngine;

public class Spike : MonoBehaviour

{

public float moveSpeed;

public Vector2 minMaxYValue;

Vector2 pos;

void Update()

{ //Vector3.Up = new Vector3(0,1,0)

transform.position += Vector3.up \* moveSpeed \* Time.deltaTime;

if (transform.position.y > minMaxYValue.x)

{

pos = transform.position;

pos.x = transform.position.x;

pos.y = minMaxYValue.y;

transform.position = pos;

ShowOrHide();

}

}

void ShowOrHide()

{

if (Random.value > 0.5f)

{

Debug.Log("Disabling");

transform.GetChild(0).gameObject.SetActive(false);

return;

}

transform.GetChild(0).gameObject.SetActive(true);

}

}