**Лабораторное занятие № 26.**

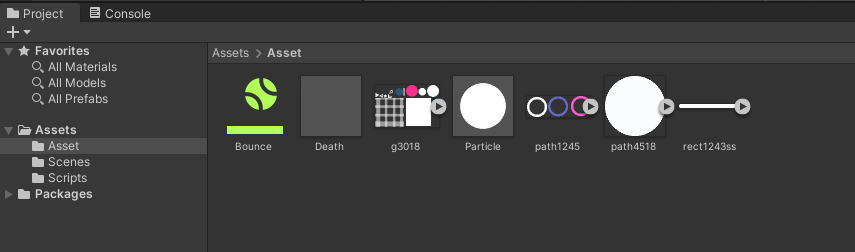
Лист

**Тема: Разработка игры “ Bounce”**

**Цель: Разработать игру “ Bounce ”**

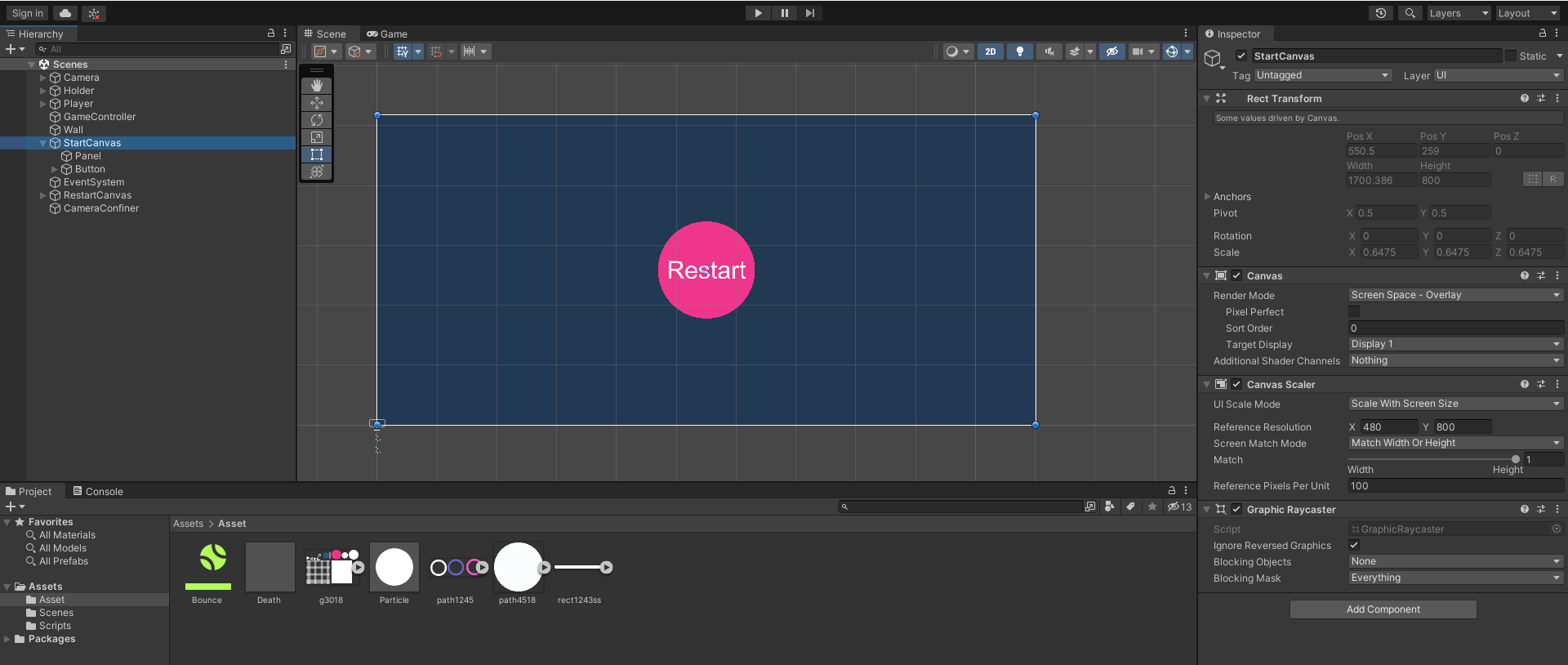
Ход работы:

1.Перекидываю ресурсы (Asset)



**Рис.26.1** – Asset.

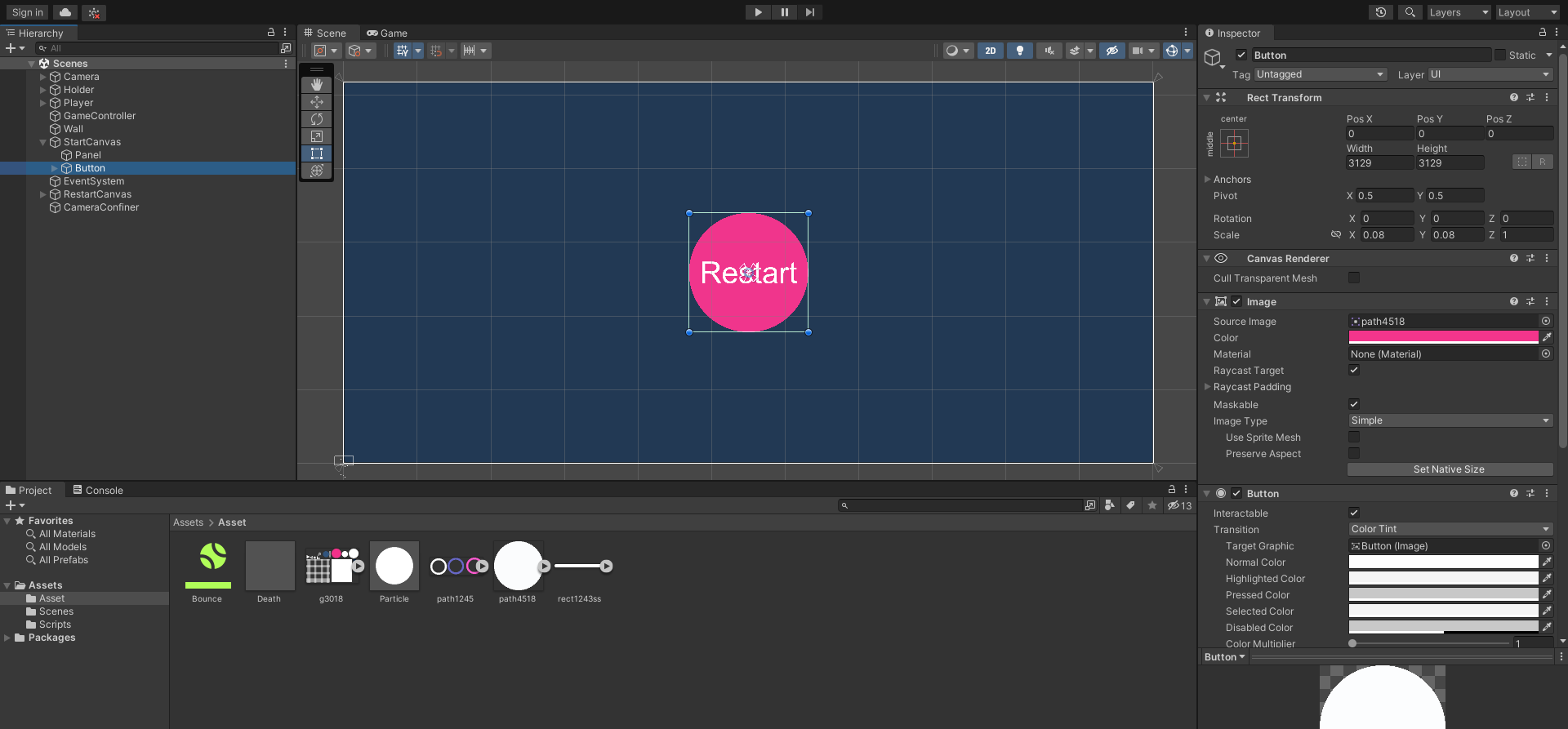
2.Создание панели UI.



**Рис.26.2** – Панель UI.

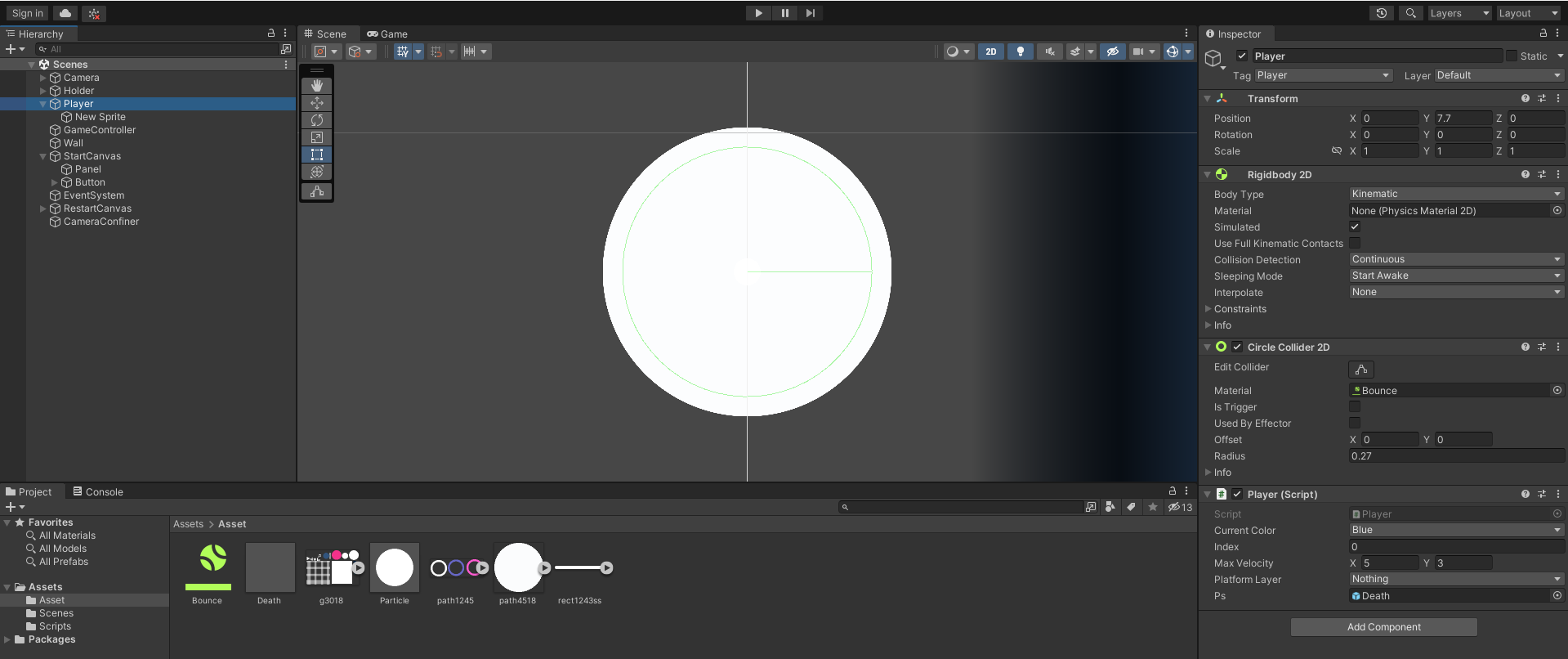
3.Создание кнопки «Start»

Лист



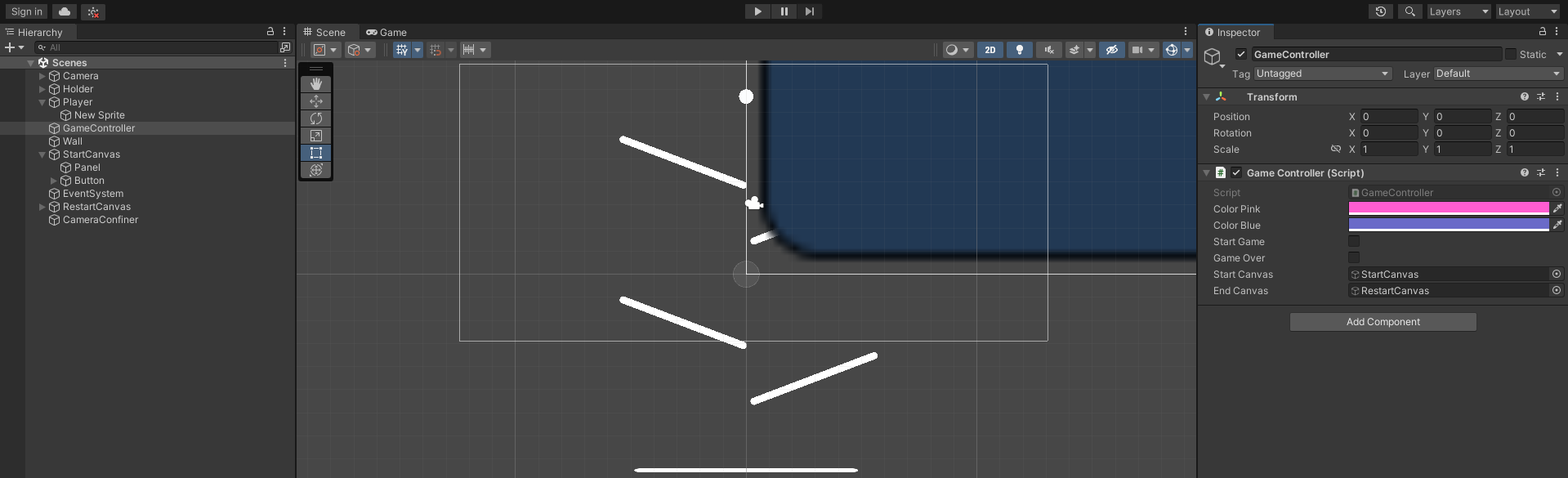
**Рис.26.3** – «Start».

4.Создание игрока «Player»



**Рис.26.4** – «Player».

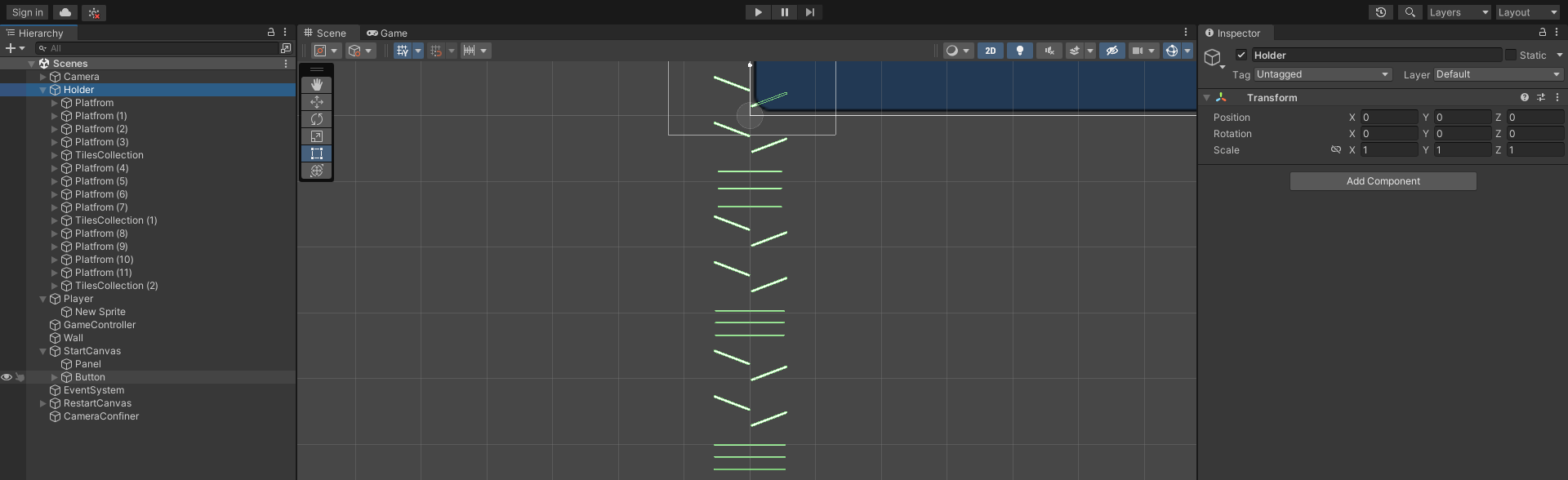
5.Создание Старта/Рестарта игры.



**Рис.26.5** – Старт/Рестарт.

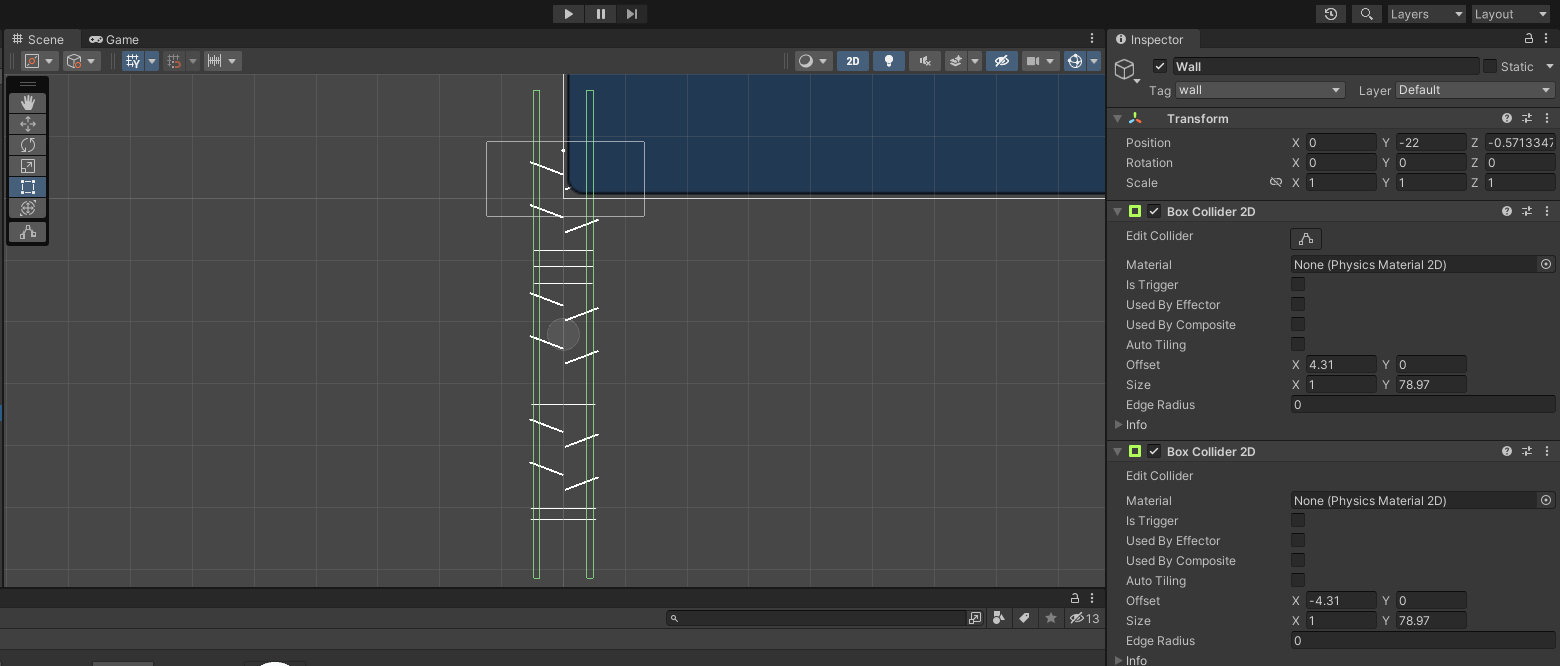
6.Создание Игрового поля.

Лист



**Рис.26.6** – Игровое поле.

7.Границы Игрового поля.



**Рис.26.7** – Границы Игрового поля.

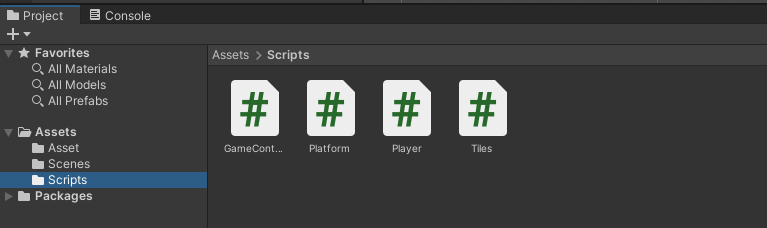
8.Inspector Рестарта игры.

Лист



**Рис.26.8** – Inspector рестарта.

9.Создание скриптов.



**Рис.26.9** – Scripts.

**Скрипты:**

GameController.cs

using UnityEngine.SceneManagement;

using UnityEngine;

[DefaultExecutionOrder(-100)]

public class GameController : MonoBehaviour

{

public Color colorPink;

public Color colorBlue;

public bool startGame, GameOver;

Player player;

public GameObject StartCanvas, EndCanvas;

// Start is called before the first frame update

void Start()

Лист

{

StartCanvas.SetActive(true);

EndCanvas.SetActive(false);

player = FindObjectOfType<Player>();

}

public void StartTheGame()

{

startGame = true;

player.ChangeTheRigidbody();

}

public void ShowEndCanvas()

{

EndCanvas.SetActive(true);

}

public void RestartScene()

{

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

}

}

Platform.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Platform : MonoBehaviour

{

SpriteRenderer sr;

GameController controller;

// Start is called before the first frame update

void Start()

{

sr = GetComponentInChildren<SpriteRenderer>();

controller = FindObjectOfType<GameController>();

ChnageCOlorAndSetTag();

}

void ChnageCOlorAndSetTag()

{

if (Random.value < 0.5f)

{

//blue

sr.color = controller.colorBlue;

transform.GetChild(0).tag = "blue";

}

else

{

//pink

sr.color = controller.colorPink;

transform.GetChild(0).tag = "pink";

}

}

// Update is called once per frame

void Update()

{

}

}

Player.cs

using System.Collections;

Лист

using System.Collections.Generic;

using UnityEngine;

public class Player : MonoBehaviour

{ public enum colorType { blue,pink,}

public colorType currentColor;

public int index;

int maxCount=2;

SpriteRenderer sr;

Rigidbody2D rgbd;

GameController controller;

public Vector2 maxVelocity;

public LayerMask PlatformLayer;

public GameObject ps;

// Start is called before the first frame update

void Start()

{

rgbd = GetComponent<Rigidbody2D>();

sr = GetComponentInChildren<SpriteRenderer>();

controller = FindObjectOfType<GameController>();

ColorChange(index);

}

// Update is called once per frame

void Update()

{ if (controller.GameOver) return;

if (Input.GetButtonDown("Fire1"))

{

if (controller.GameOver) return;

if (index<= maxCount)

{

index++;

if(index == maxCount ) { index = 0; }

ColorChange(index);

}

}

ClampVelocity();

}

void ClampVelocity()

{

Vector2 vel = rgbd.velocity;

if(vel.x > maxVelocity.x)

{

vel.x = maxVelocity.x;

}

if (vel.y > maxVelocity.y)

{

vel.y = maxVelocity.y;

}

rgbd.velocity = vel;

}

void ColorChange(int colorValue)

{

switch (colorValue)

{

case 0: currentColor = colorType.blue;

sr.color = controller.colorBlue;

break;

case 1:

currentColor = colorType.pink;

Лист

sr.color = controller.colorPink;

break;

default: Debug.LogError("unable to change the color");

break;

}

}

public void OnCollisionEnter2D(Collision2D collision)

{

// if (collision.gameObject.layer != PlatformLayer) return;

if (collision.collider.tag == "wall")

{

return;

}

if(collision.collider.tag == currentColor.ToString())

{

//good give point to player

}else if(collision.collider.tag != currentColor.ToString())

{

//game Oveer

Debug.Log("GameOver");

GameOver();

}

}

public void OnCollisionStay2D(Collision2D collision)

{

// if (collision.gameObject.layer != PlatformLayer) return;

if (collision.collider.tag == "wall")

{

return;

}

if (collision.collider.tag == currentColor.ToString())

{

//good give point to player

}

else if (collision.collider.tag != currentColor.ToString())

{

//game Oveer

Debug.Log("GameOver");

GameOver();

}

}

public void OnTriggerEnter2D(Collider2D collision)

{

if (collision.gameObject.layer != PlatformLayer) return;

if (collision.tag == currentColor.ToString())

{

//good give point to player

}

else if (collision.tag != currentColor.ToString())

{

//game Oveer

Debug.Log("GameOver");

GameOver();

}

} public void OnTriggerStay2D(Collider2D collision)

{

if (collision.gameObject.layer != PlatformLayer) return;

if (collision.tag == currentColor.ToString())

{

//good give point to player

}

else if (collision.tag != currentColor.ToString())

{

//game Oveer

Debug.Log("GameOver");

GameOver();

}

}

public void ChangeTheRigidbody()

{

rgbd.bodyType = RigidbodyType2D.Dynamic;

}

void GameOver()

{

controller.GameOver = true;

Instantiate(ps, transform.position, Quaternion.identity);

sr.enabled = false;

rgbd.bodyType = RigidbodyType2D.Kinematic;

controller.ShowEndCanvas();

}

}

Tiles.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Tiles : MonoBehaviour

{

public void OnTriggerEnter2D(Collider2D collision)

{ if (collision.tag != "Player")

return;

if(collision.tag != transform.tag)

Debug.Log("dead");

}

}

Лист