Лабораторная работа №26

Тема: разработка игрового проекта Bouncy

Цель: приобрести навыки в разработке игрового проекта Bouncy

Ход работы:

1. Выполнение работы

Создание игрового проекта Bouncy

1. Сцена

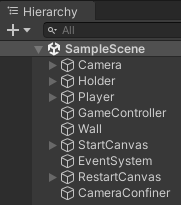


Рис. 26.1 Иерархия сцены

1. Создал скрипт GameController, хранящий цвета объектов и отвечающий за старт и рестарт игры.

Листинг скрипта GameController.cs

using UnityEngine;

using UnityEngine.SceneManagement;

public class GameController : MonoBehaviour

{

public Color colorPink;

public Color colorBlue;

public bool startGame, GameOver;

public GameObject StartCanvas, EndCanvas;

private Player player;

void Start()

{

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);

}

}

1. Создал скрипт Player, отвечающий за поведение шарика при столкновении с разными объектами.

Листинг скрипта Player.cs

using UnityEngine;

public class Player : MonoBehaviour

{

public enum colorType { blue, pink, }

public colorType currentColor;

public int index;

public Vector2 maxVelocity;

public LayerMask PlatformLayer;

public GameObject ps;

private int maxCount = 2;

private SpriteRenderer sr;

private Rigidbody2D rgbd;

private GameController controller;

void Start()

{

rgbd = GetComponent<Rigidbody2D>();

sr = GetComponentInChildren<SpriteRenderer>();

controller = FindObjectOfType<GameController>();

ColorChange(index);

}

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;

sr.color = controller.colorPink;

break;

}

}

public void OnCollisionEnter2D(Collision2D collision)

{

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

return;

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

GameOver();

}

void OnCollisionStay2D(Collision2D collision)

{

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

return;

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

GameOver();

}

void OnTriggerEnter2D(Collider2D collision)

{

if (collision.gameObject.layer != PlatformLayer)

return;

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

GameOver();

}

void OnTriggerStay2D(Collider2D collision)

{

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

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

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();

}

}

1. Создал скрипт Platform, отвечающий за установку типа платформы при старте игры.

Листинг скрипта Platform.cs

using UnityEngine;

public class Platform : MonoBehaviour

{

private SpriteRenderer sr;

private GameController controller;

void Start()

{

sr = GetComponentInChildren<SpriteRenderer>();

controller = FindObjectOfType<GameController>();

ChnageCOlorAndSetTag();

}

void ChnageCOlorAndSetTag()

{

if (Random.value < 0.5f)

{

sr.color = controller.colorBlue;

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

}

else

{

sr.color = controller.colorPink;

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

}

}

}

1. Создал скрипт Tiles.

Листинг скрипта Tiles.cs

using UnityEngine;

public class Tiles : MonoBehaviour

{

public void OnTriggerEnter2D(Collider2D collision)

{

if (collision.tag != "Player")

return;

if (collision.tag != transform.tag)

Debug.Log("dead");

}

}

1. Вывод

В ходе проделанной работы были приобретены навыки в разработке игрового проекта Bouncy.