

**Руководство программиста**

ПМ.03 Ревьюирование программных продуктов

|  |  |
| --- | --- |
| Пинаев Алексей Владимирович | |
| *(Ф.И.О. обучающегося)* | |
| 09.02.07 Информационные системы и программирование | |
| *(специальность)* | |
|  | |
| Учебная группа | ИСПк-301-51-00 |

Киров, 2022 г.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class Hero : MonoBehaviour

{

[SerializeField] float speed = 3f;

[SerializeField] float jumpForce = 1f

bool Ground = false;

Rigidbody2D rb;

SpriteRenderer sprite;

void Start()

{

}

// Update is called once per frame

void Update()

{

if (Input.GetButton("Horizontal"))

Run();

if (Ground && Input.GetButton("Jump")) // вызывает функции Run() и Jump() с каждым кадром обновления

Jump();

}

void Awake()

{

rb = GetComponent<Rigidbody2D>();

sprite = GetComponentInChildren<SpriteRenderer>(); // объявления полей, передач ссылок на этот объект

}

void Run()

{

Vector3 dir = transform.right \* Input.GetAxis("Horizontal");

transform.position = Vector3.MoveTowards(transform.position, transform.position + dir, speed \* Time.deltaTime); // работа бега

sprite.flipX = dir.x < 0.0f;

}

void Jump()

{

rb.AddForce(transform.up \* jumpForce, ForceMode2D.Impulse); // работа прыжка

}

void Grounded()

{

Collider2D[] collider = Physics2D.OverlapCircleAll(transform.position, 1f); // предназначена для того чтобы главный персонаж мог стоять на платформах

Ground = collider.Length > 1;

}

void FixedUpdate()

{

Grounded(); //вызывается с фиксированной частотой не зависимо от FPS

}

private void OnTriggerEnter2D(Collider2D collision) // логика сбора ягод и логика смерти

{

if (collision.tag == "smert")

{

SceneManager.LoadScene(0);

}

if (collision.tag == "Cherry")

{ Collect.TheCherry += 1;

Destroy(collision.gameObject);

}

else if (collision.tag == "qwe")

{

SceneManager.LoadScene(0);

}

if (collision.tag == "qwe")

{

Collect.TheCherry = 0;

}

if (collision.tag == "speed")

{

if (transform.position.y > 10f)

Destroy(gameObject);

transform.position += new Vector3(14, speed \* Time.deltaTime, 0); //логика телепортирующих шаров

Destroy(collision.gameObject);

}

}

}