

ГУАП

КАФЕДРА № 42

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ \_\_\_\_\_  
ПРЕПОДАВАТЕЛЬ

доцент  
должность, уч. степень, звание

подпись, дата

Т.А. Суетина  
\_\_\_\_\_  
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

**СПРАЙТОВАЯ АНИМАЦИЯ**

по курсу: МУЛЬТИМЕДИЙНЫЙ ПРАКТИКУМ

РАБОТУ ВЫПОЛНИЛИ

СТУДЕНТ ГР. № \_\_\_\_\_ 4128

\_\_\_\_\_  
подпись, дата

Воробьев В.А.  
\_\_\_\_\_  
инициалы, фамилия

## **1 Исходная информация**

### **1.1 Цель работы**

Изучить основы создания анимированного 2D персонажа в Unity.

### **1.2 Задание**

Используя атлас спрайтов создать 2D анимацию персонажа с основными элементами движения - бег, прыжок.

## 2 Результаты работы

### 1.2 Задание

Скриншоты, относящиеся к разработке изображены на рисунках 1-3. Результаты (скриншоты, сделанные во время передвижения) представлены на рисунках 4-6. Скрипт контроллера персонажа представлен в Приложении А – он управляет перемещениями персонажа, его скоростью и направлением.



Рисунок 1 – Атлас спрайтов персонажа

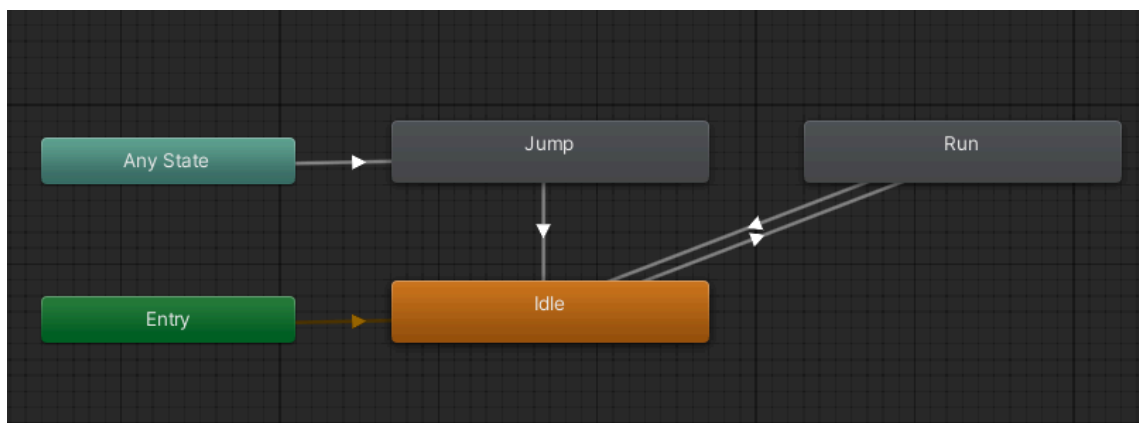


Рисунок 2 – Аниматор

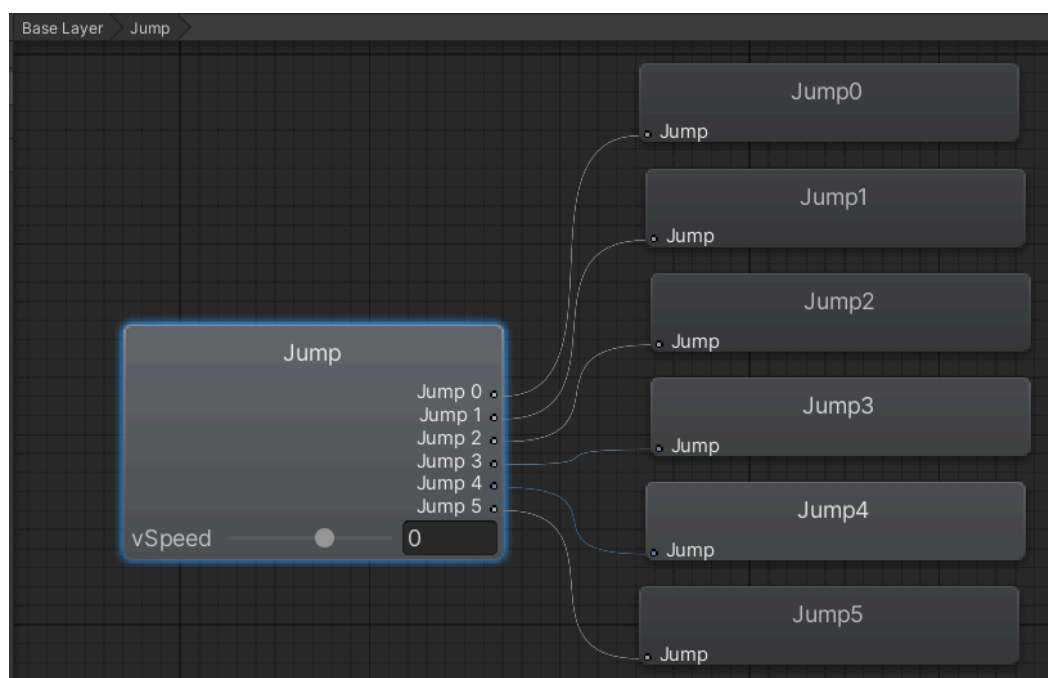


Рисунок 3 – Blend Tree для прыжка

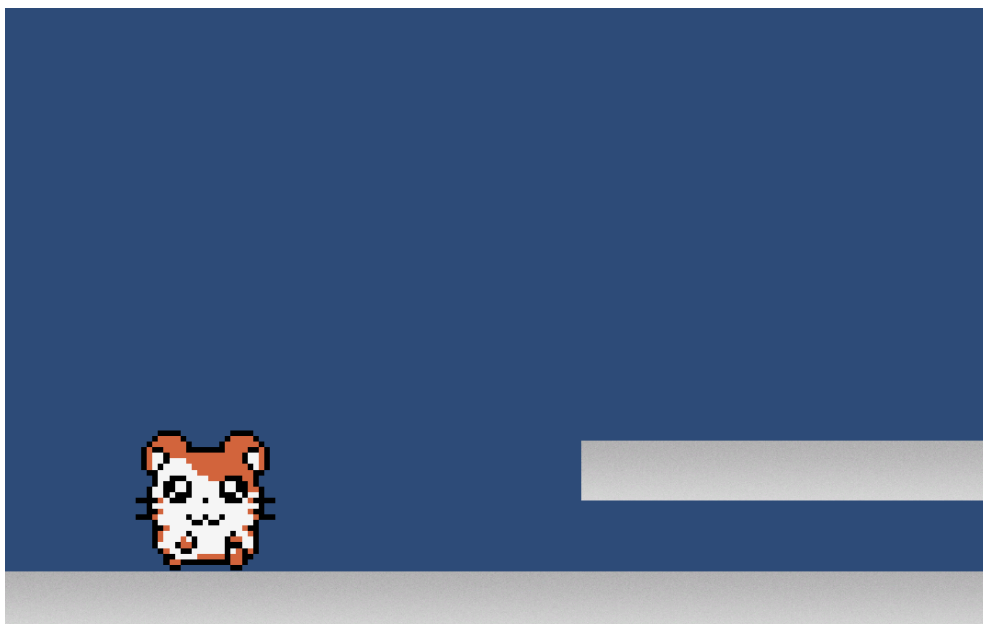


Рисунок 4 – Анимация «Idle»

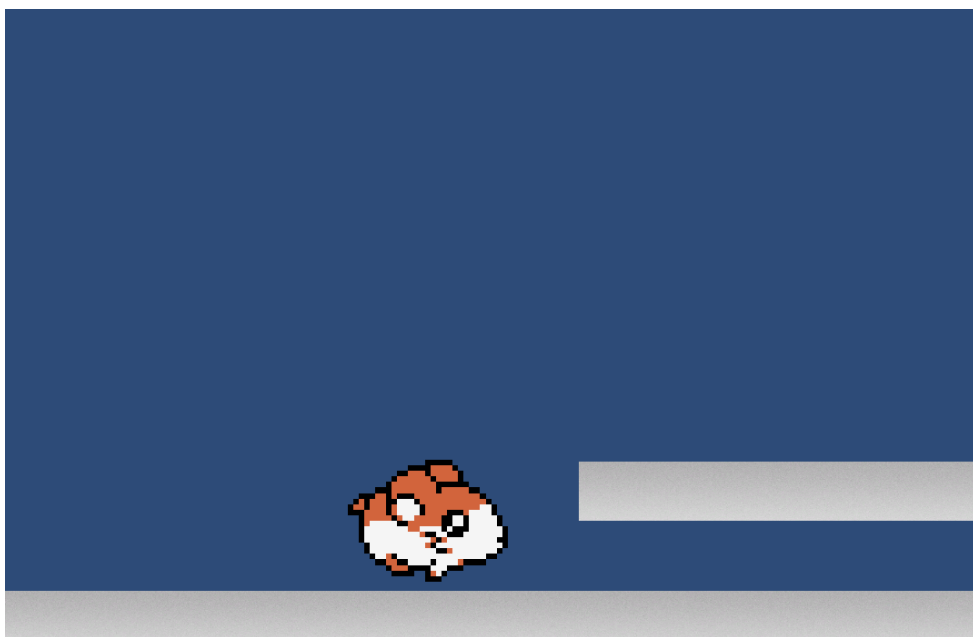


Рисунок 5 – Анимация «Run»

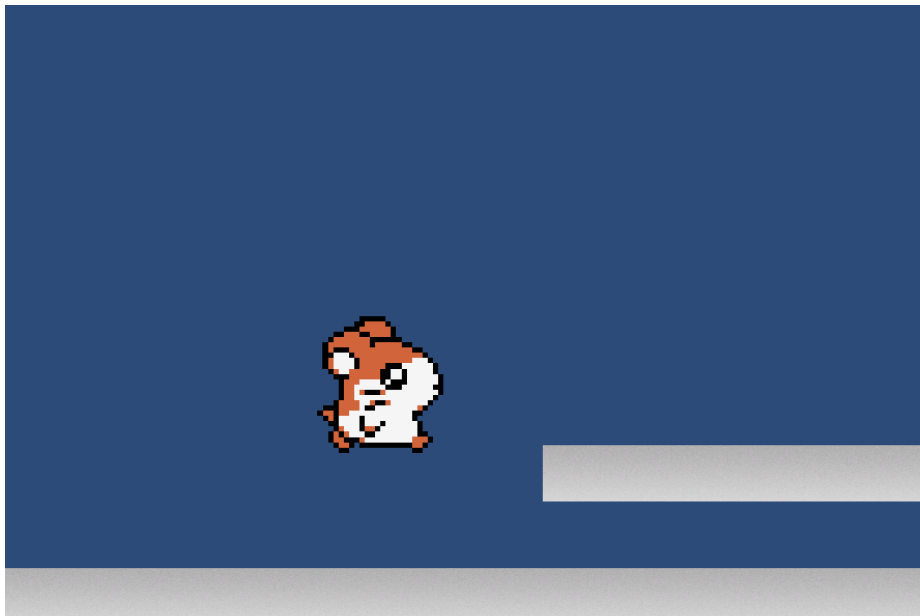


Рисунок 6 – Анимация «Jump»

## ЛИСТИНГ КОНТРОЛЛЕРА ПЕРСОНАЖА

```
using UnityEngine;
using System.Collections;

public class CharacterControllerScript : MonoBehaviour
{
    public float maxSpeed = 10f;
    private bool isFacingRight = true;
    private Animator anim;
    private bool isGrounded = false;
    public Transform groundCheck;
    private float groundRadius = 0.2f;
    public LayerMask whatIsGround;

    private void Start()
    {
        anim = GetComponent<Animator>();
    }

    private void FixedUpdate()
    {
        isGrounded = Physics2D.OverlapCircle(groundCheck.position, groundRadius,
        whatIsGround);
        anim.SetBool ("Ground", isGrounded);
        anim.SetFloat ("vSpeed", GetComponent<Rigidbody2D>().velocity.y);
        if (!isGrounded)
            return;
        float move = Input.GetAxis("Horizontal");

        anim.SetFloat("Speed", Mathf.Abs(move));

        GetComponent<Rigidbody2D>().velocity = new Vector2(move * maxSpeed,
        GetComponent<Rigidbody2D>().velocity.y);

        if(move > 0 && !isFacingRight)
            Flip();
        else if (move < 0 && isFacingRight)
            Flip();
    }

    private void Update()
    {
        if (isGrounded && Input.GetKeyDown (KeyCode.Space))
        {
            anim.SetBool("Ground", false);
            GetComponent<Rigidbody2D>().AddForce(new Vector2(0, 600));
        }
    }

    private void Flip()
    {
        isFacingRight = !isFacingRight;
        Vector3 theScale = transform.localScale;
        theScale.x *= -1;
        transform.localScale = theScale;
    }
}
```