

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
Федерального государственного бюджетного образовательного учреждения
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет _____ ИТР _____

Кафедра _____ ПИИ _____

ЛАБОРАТОРНАЯ РАБОТА №2

По Компьютерной графике

Руководитель

Привезенцев Д.Г.

(фамилия, инициалы)

(подпись)

(дата)

Студент ПИИ - 121

(группа)

Ермилов М.В.

(фамилия, инициалы)

(подпись)

(дата)

Муром 2023

Лабораторная работа №2

Тема: Разработка игры 2D Platformer

Ход работы:

1. Создание проекта Unity. 2D

					МИ ВлГУ 09.03.04						
Изм.	Лист	№ докум.	Подпись	Дата							
Разраб.		Ермилов М.В.						Лит.	Лист	Листов	
Провер.		Привезенцев Д.Г.								2	5
Реценз.								ПИН-121			
Н. Контр.											
Утверд.											

Код проекта на юнити:

```
public class Character : MonoBehaviour
{
    new private Rigidbody2D rigidbody;
    private Animator animator;
    private SpriteRenderer sprite;
    public enum CharState { Idle, Run, Jump, Fall };
    private bool IsGrounded;
    [SerializeField]
    public double lives;
    public float speed;
    public int jumpForce;

    private CharState State
    {
        get { return (CharState)animator.GetInteger("State"); }
        set { animator.SetInteger("State", (int)value); }
    }

    private void FixedUpdate()
    {
        CheckGround();
    }

    public void ReceiveDamage(Unit enemy)
    {
        lives--;
        if (lives <= 0)
            Die();
        else
        {
            Vector2 dir = (transform.position - enemy.transform.position).normalized;
            rigidbody.velocity = Vector2.zero;
            rigidbody.AddForce(transform.up * 8.0f, ForceMode2D.Impulse);
        }
    }

    public virtual void Die()
    {
        Destroy(gameObject);
    }

    private void CheckGround()
    {
        Collider2D[] colliders = Physics2D.OverlapCircleAll(transform.position, 0.3f);
        IsGrounded = colliders.Length > 1;
        float YVelocity = rigidbody.velocity.y;
        if (!IsGrounded && YVelocity > 0)
            State = CharState.Jump;
        else if (!IsGrounded && YVelocity < 0)
            State = CharState.Fall;
    }

    private void Awake()
    {
        rigidbody = GetComponent<Rigidbody2D>();
        animator = GetComponent<Animator>();
        sprite = GetComponentInChildren<SpriteRenderer>();
    }

    void Run()
    {
        if (IsGrounded) State = CharState.Run;
        Vector3 direction = transform.right * Input.GetAxis("Horizontal");
        sprite.flipX = direction.x < 0;
    }
}
```

```

        transform.position = Vector3.MoveTowards(transform.position, transform.position
+ direction, speed * Time.deltaTime);
    }

    void Jump()
    {
        rigidbody.AddForce(transform.up * jumpForce, ForceMode2D.Impulse);
    }

    void Update()
    {
        if (Input.GetButton("Horizontal")) Run();
        if (IsGrounded && Input.GetButtonDown("Jump")) Jump();
        if (IsGrounded) State = CharState.Idle;
    }

    void Start()
    {
        sprite = GetComponentInChildren<SpriteRenderer>();
        animator = sprite.GetComponent<Animator>();
    }
}

public class Obstacle : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
    }

    void OnTriggerEnter2D( Collider2D collision)
    {
        Unit unit = collision.GetComponent<Unit>();
        if (unit && unit.is Character)
        {
            unit.ReceiveDamage(this);
        }
    }

    // Update is called once per frame
    void Update()
    {
    }
}

public class Camera : MonoBehaviour
{
    private Transform player;
    private float damping = 3f;
    private Vector2 Offset = new Vector2(3.0f, 2.0f);
    // Start is called before the first frame update
    void Start()
    {
        player = GameObject.FindGameObjectWithTag("Player").transform;
    }

    // Update is called once per frame
    void Update()
    {
        Vector3 target = new Vector3(player.position.x + Offset.x, player.position.y +
Offset.y, transform.position.z);
        Vector3 currentPosition = Vector3.Lerp(transform.position, target, damping *
Time.deltaTime);
        transform.position = target;
    }
}

```

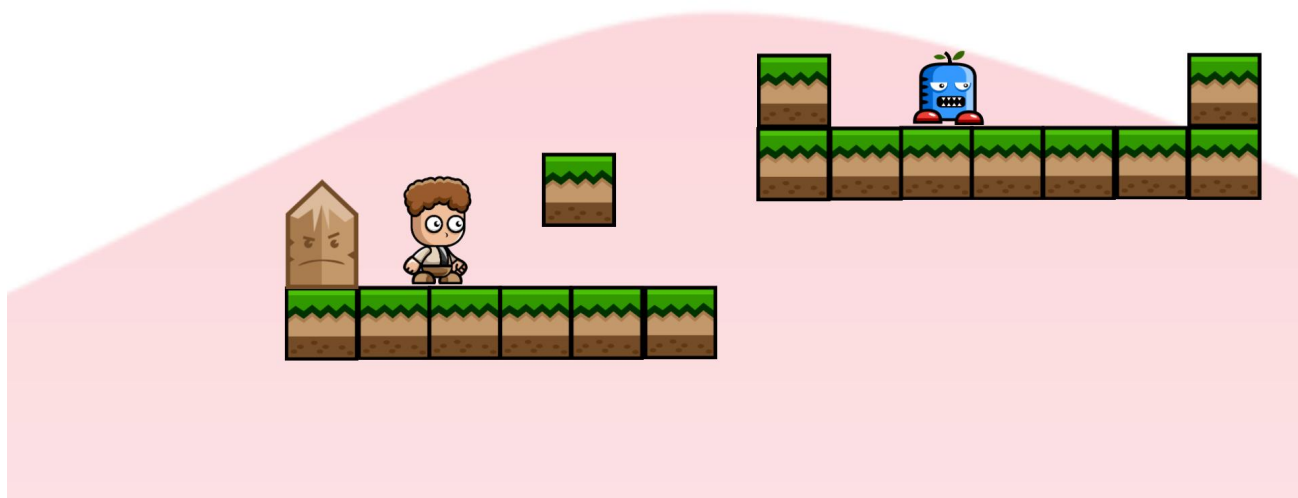


Рис 1 - пример работы игры

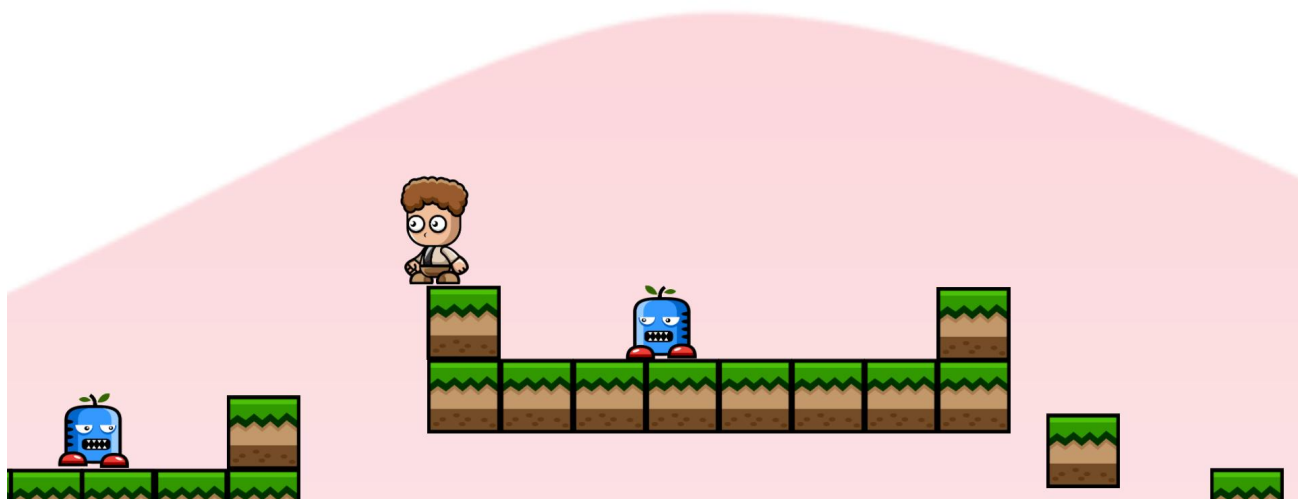


Рис 2 - пример работы игры