



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА - Российский технологический университет»

РТУ МИРЭА

---

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения  
(ИиППО)

## ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ

по дисциплине

«Моделирование сред и разработка приложений виртуальной и дополненной  
реальности»

Выполнил студент группы ИМБО-02-22

Ким К.С.

Принял старший преподаватель

Благирев М.М.

Практические работы выполнены «\_\_» \_\_\_\_\_ 2025 г.

(подпись студента)

«Зачтено» «\_\_» \_\_\_\_\_ 2025 г.

(подпись студента)

Москва 2025

# Проект

## Цель работы

Создание 2D игры на Unity.

Находим нужные ассеты для игры на Рисунке 1.

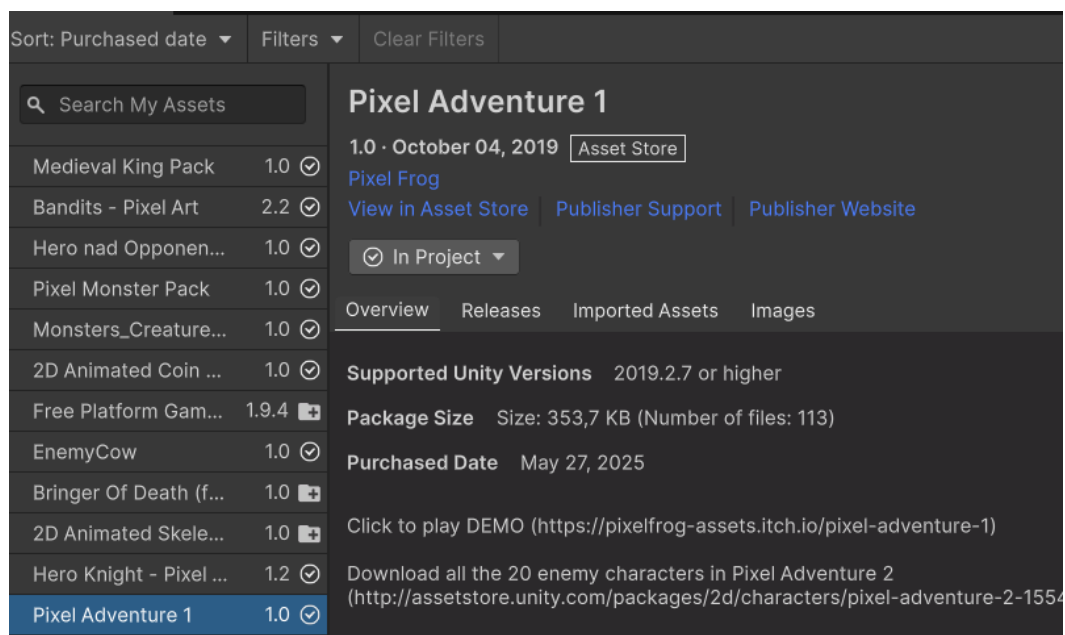


Рисунок 1 – Ассеты

Создали 1 уровень показано на Рисунке 2. В этом уровне надо собрать 10 монет.

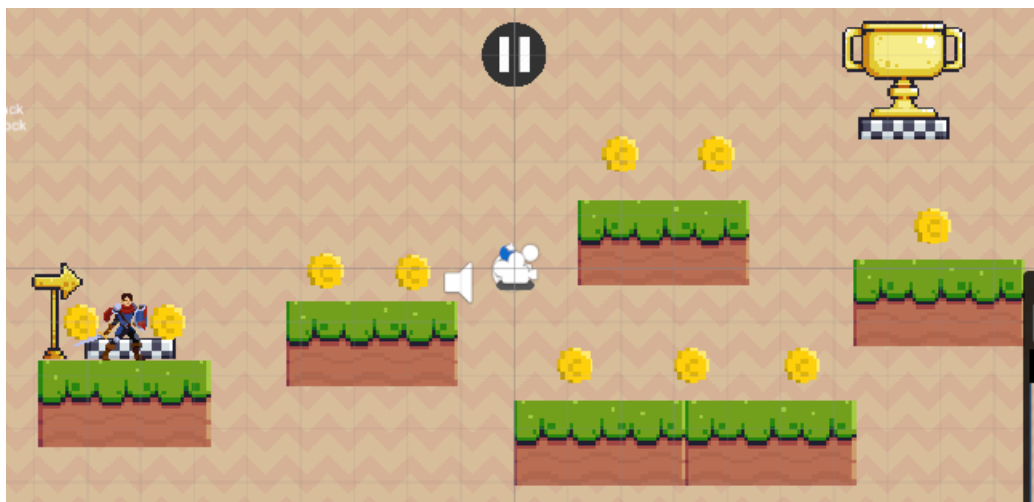


Рисунок 2 – 1 уровень

Создали 2 уровень показано на Рисунке 3. В этом уровне надо убить 5 NPC врагов.

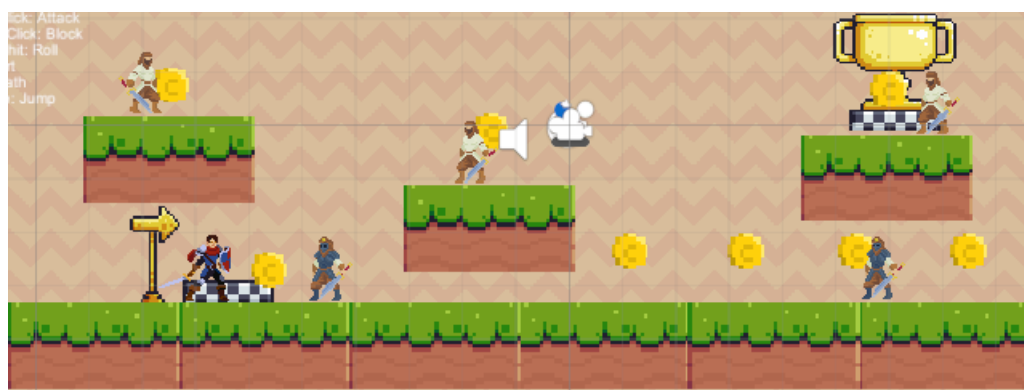


Рисунок 3 – 2 уровень

Создали 3 уровень показано на Рисунке 4. В этом уровне надо убить двух боссов.

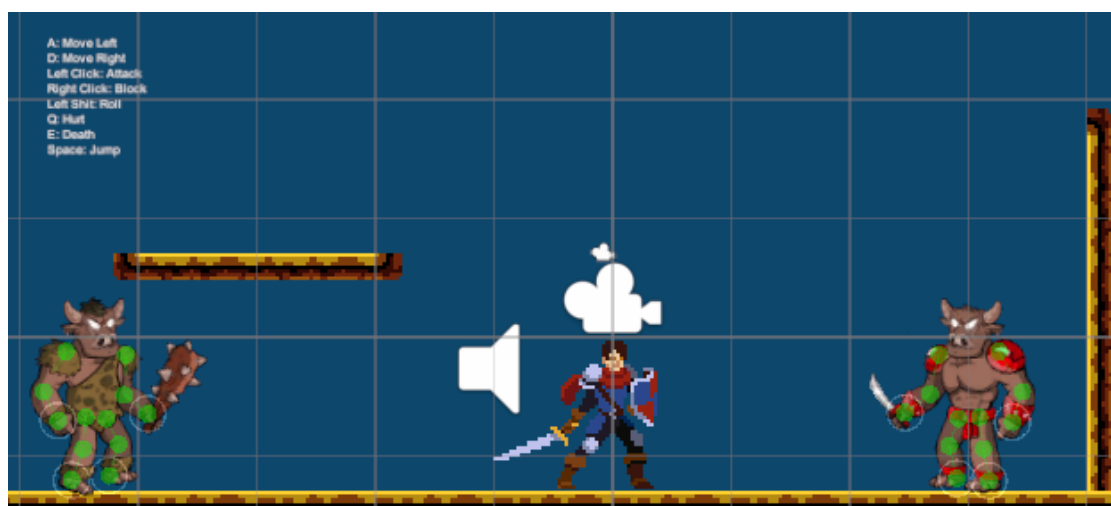


Рисунок 4 – 3 уровень

Создание меню на Рисунке 5.

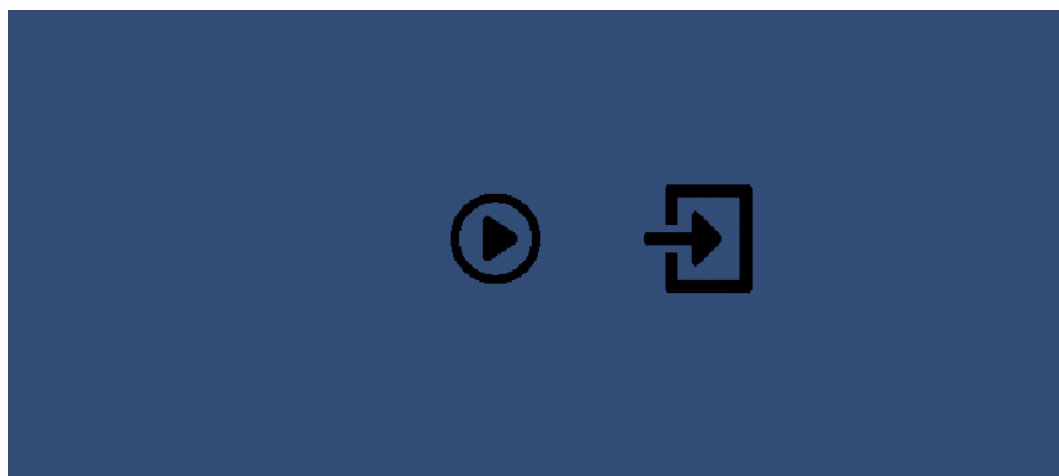


Рисунок 5 – Меню

Создание анимации героя на Рисунке 6.

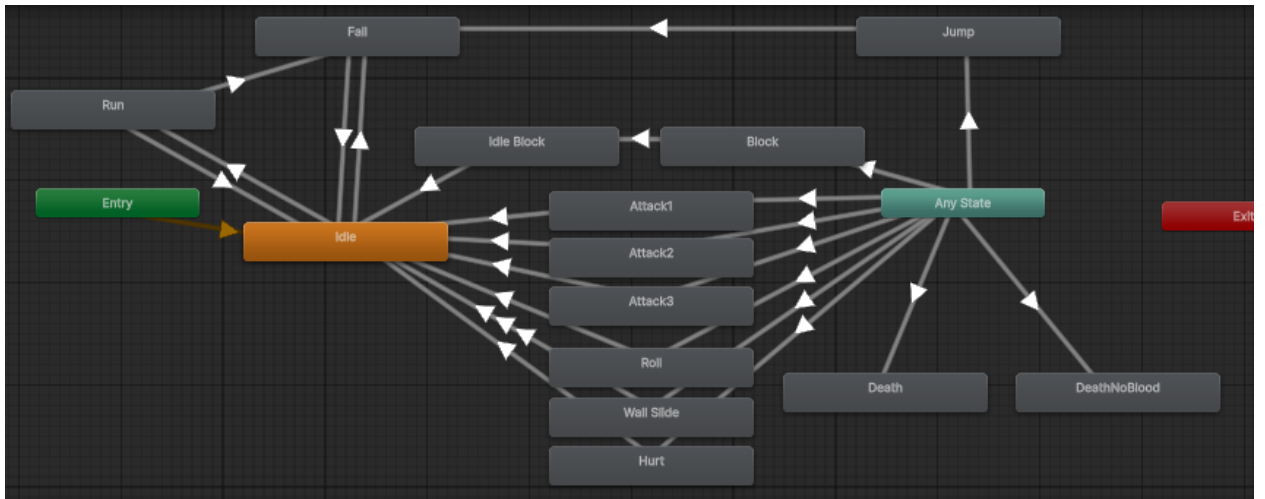


Рисунок 6 – Анимации героя

Создание PlayerController, взятие монеты и убийство врагов на Рисунках 7-10.

```

[RequireComponent(typeof(Rigidbody2D), typeof(Animator), typeof(SpriteRenderer))]
// Скрипт Unity (ссылка на ресурсы: 3) | Ссылка: 8
public class PlayerController : MonoBehaviour
{
    [Header("Movement Settings")]
    [SerializeField] float m_speed = 4.5f;
    [SerializeField] float m_jumpForce = 8f;
    [SerializeField] float m_rollForce = 6.0f;
    [SerializeField] float m_rollDuration = 8.0f / 14.0f;

    [Header("Combat Settings")]
    [SerializeField] int m_maxHealth = 5;
    [SerializeField] int m_damage = 25;
    [SerializeField] float m_attackRange = 0.8f;
    [SerializeField] Transform m_attackPoint;
    [SerializeField] LayerMask m_enemyLayer;
    [SerializeField] float m_invincibilityTime = 1f;

    [Header("Effects")]
    [SerializeField] bool m_noBlood = false;
    [SerializeField] GameObject m_slideDust;

    [Header("Audio Sources")]
    [SerializeField] AudioSource attackAudioSource; // Аудио для атаки
    [SerializeField] AudioSource hurtAudioSource; // Аудио для урона
    [SerializeField] AudioSource deathAudioSource; // Аудио для смерти

    // Components
    private Animator m_animator;
    private Rigidbody2D m_body2d;
    private SpriteRenderer m_spriteRenderer;
    private AudioSource m_audioSource;

    // Sensors
    private Sensor_HeroKnight m_groundSensor;
    private Sensor_HeroKnight[] m_wallSensors = new Sensor_HeroKnight[4];
}

```

Рисунок 7 – PlayerController.cs часть 1

```

Ссылка: 1
void HandleMovement()
{
    if (m_rolling) return;

    float inputX = Input.GetAxis("Horizontal");
    m_body2d.linearVelocity = new Vector2(inputX * m_speed, m_body2d.linearVelocity.y);

    // Flip sprite based on direction
    if (inputX > 0)
    {
        m_spriteRenderer.flipX = false;
        m_facingDirection = 1;
    }
    else if (inputX < 0)
    {
        m_spriteRenderer.flipX = true;
        m_facingDirection = -1;
    }
}

Ссылка: 1
void HandleCombat()
{
    if (Input.GetMouseButtonDown(0)) // Left click
    {
        if (m_timeSinceAttack > 0.25f && !m_rolling)
        {
            m_currentAttack = (m_timeSinceAttack > 1.0f) ? 1 : (m_currentAttack % 3) + 1;
            m_animator.SetTrigger("Attack" + m_currentAttack);
            m_timeSinceAttack = 0.0f;

            // Play attack sound
            if (attackAudioSource != null && !attackAudioSource.isPlaying)
            {
                attackAudioSource.Play();
            }
        }
    }

    // Blocking
    if (Input.GetMouseButtonDown(1)) // Right click
    {
        m_animator.SetTrigger("Block");
        m_animator.SetBool("IdleBlock", true);
    }
}

```

Рисунок 8 – PlayerController.cs часть 2

```

using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

// Скрипт Unity (1 ссылка на ресурсы) | Ссылка: 0
public class TakeItem : MonoBehaviour
{
    [SerializeField] private float coinCount = 0;
    [SerializeField] private Text coinText;
    [SerializeField] private AudioSource coinSound;

    // Сообщение Unity | Ссылка: 0
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Item") && collision.gameObject.activeSelf)
        {
            collision.gameObject.SetActive(false);
            CollectCoin(collision.gameObject);
        }
    }

    // Ссылка: 1
    private void CollectCoin(GameObject coinObject)
    {
        coinCount++;
        UpdateCoinUI();
        PlayCoinSound();
        Destroy(coinObject);

        if (coinCount >= 10)
        {
            LoadNextLevel();
        }
    }
}

```

Рисунок 9 – TakeItem.cs

```

using UnityEngine.SceneManagement;

// Скрипт Unity (ссылки на ресурсы: 5) | Ссылки: 0
public class BanditEnemy : MonoBehaviour
{
    public int health = 3;
    [SerializeField] private Text BanditText;
    // Ссылки: 1
    public void TakeDamage()
    {
        health--;
        Debug.Log($"Враг получил {1} урон. Здоровье: {health}");
        if (health <= 0)
        {
            // Вызов метода DefeatEnemy у игрока
            PlayerController player = FindAnyObjectByType<PlayerController>();
            if (player != null)
            {
                player.DefeatEnemy(); // Увеличивает счётчик убитых врагов
            }
            Debug.Log("Враг умер!");
            Destroy(gameObject);
        }
    }

    // Сообщение Unity | Ссылки: 0
    private void OnCollisionEnter2D(Collision2D collision)
    {
        if (collision.gameObject.tag == "Player")
        {
            TakeDamage();
        }
    }
}

```

Рисунок 10 – BanditEnemy.cs

## Вывод

Выполнен проект.