



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА - Российский технологический университет»
РТУ МИРЭА

Институт Информационных Технологий
Кафедра Инструментального и прикладного программного обеспечения

Практическая работа №4

по дисциплине
«Моделирование сред и разработка приложений виртуальной и
дополненной реальности»

Студент группы ИКБО-04-22

Егоров Л.А.
(Ф.И.О. студента)

Принял

Костыренков А.О.
(Ф.И.О. преподавателя)

СОДЕРЖАНИЕ

1 ЭЛЕМЕНТЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ И ЧИСТОГО КОДА	3
1.1 Цель и задачи	3
1.2 Практическая часть	3
1.3 Заключение	4

1 ЭЛЕМЕНТЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ И ЧИСТОГО КОДА

1.1 Цель и задачи

Цель: изучить элементы ООП

Задачи:

- применить основные принципы ООП в работе над проектом.

1.2 Практическая часть

Основные принципы ООП (объектно-ориентированного программирования):

- Инкапсуляция – объединение данных и методов в одном классе, а также сокрытие внутренней реализации от внешнего вмешательства (например, через модификаторы доступа: `private`, `protected`, `public`);
- Наследование – возможность создания новых классов на основе существующих, что позволяет переиспользовать код и выстраивать иерархии (например, класс `Child` наследует свойства и методы класса `Parent`);
- Полиморфизм – способность объектов с одинаковым интерфейсом вести себя по-разному (например, переопределение методов в дочерних классах или использование интерфейсов).

```

public class PlayerMovement : MonoBehaviour
{
    [SerializeField] private Vector3 _characterVelocity;
    [SerializeField] private float _speed;
    [SerializeField] private LayerMask _groundCheckLayers;
    [SerializeField] private float _jumpHeight;
    [SerializeField] private float mouseSensitivity;
    private Transform cameraTransform;
    private CharacterController _characterController;
    private float _groundCheckDistance;
    private bool _isGrounded;
    private float xRotation;
    private void Awake()
    {
        _characterController = GetComponent<CharacterController>();
        _groundCheckDistance = _characterController.skinWidth + Physics.defaultContactOffset;

        cameraTransform = transform.GetChild(0);

        Cursor.lockState = CursorLockMode.Locked;
    }

    private void HandleCameraRotation()
    {
        float mouseX = Input.GetAxis("Mouse X") * mouseSensitivity * Time.deltaTime;
        float mouseY = Input.GetAxis("Mouse Y") * mouseSensitivity * Time.deltaTime;

        xRotation -= mouseY;
        xRotation = Mathf.Clamp(xRotation, -20f, 20f);

        cameraTransform.localRotation = Quaternion.Euler(xRotation, 0f, 0f);
        transform.Rotate(Vector3.up * mouseX);
    }
}

```

Рисунок 1.1 — Пример написанного класса

1.3 Заключение

В результате выполнения практической работы изучены принципы ООП и их применение для работы над проектом.