

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра інформаційних систем та технологій**

**Лабораторна робота №2**

з дисципліни

"Розробка ігрових застосувань. Unity рішення"

на тему " Дослідження базового патерну ігрового рушія Unity на прикладі  
тривимірного ігрового застосунку"

Виконав:  
студент групи ІП-93  
Завальнюк М.  
Викладач:  
Катін П.Ю.

Київ – 2022

## **Зміст**

1. Мета, завдання та вхідні дані .....	3
Мета.....	3
Завдання .....	3
Вхідні дані.....	3
2. Створення проекту та програмування об'єктів .....	4
3. Додатки .....	7

## 1. Мета, завдання та вхідні дані

**Мета:** полягає у набутті знань, умінь та навичок з технології розроблення основ проекту з використанням обраної мови програмування у обраній парадигмі. Надається досвід створення репозиторію у системі контролю версій і знання елементів середовища розробки і основи вихідного коду для управління грою.

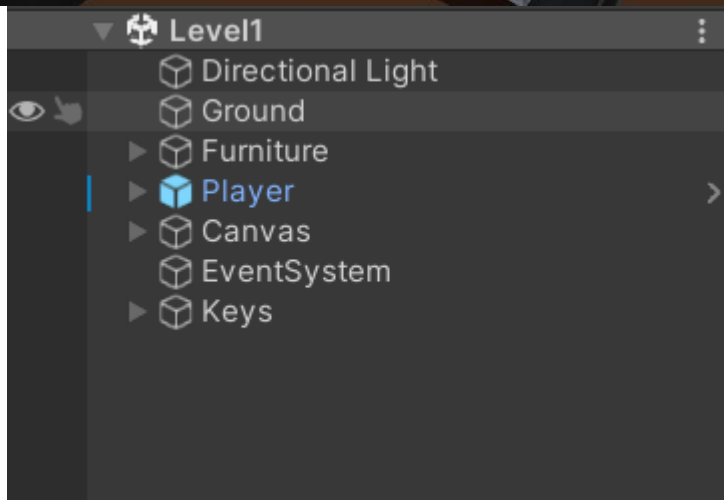
**Завдання:** створений проект IDE (3D) на основі рушія, що містить 2 сцени, ігровий персонаж. Можуть бути включені інші елементи. Розроблений і налагоджений скрипт для управління ігровим персонажем. Додані умови переходу між сценами. В якості ігрового персонажа використовується звичайний примітив. Достатньо продемонструвати рух ліворуч, праворуч, стрибки, коректну фізику, зупинку перед перешкодою, набір балів і перехід з одного рівня на інший. Можливо, за бажанням реалізувати інтерфейс гравця.

**Вхідні дані:**

- Прізвище – Завальнюк;
- Ім'я – Максим;
- Шифр групи – ІП-93;
- Назва факультету – ФІОТ;
- Варіант – 7.

## 2. Створення проекту та програмування об'єктів

Запустивши Unity, я створив 2D проект. Потім завантажив ассет відповідно до свого варіанту. Додавши об'єкти до сцени, отримав наступне:



Для програмування руху «гравця», руху камери та перехід між сценами створив папку Scripts, у ній C#-скрипти. Код можна переглянути у додатках.

### **3. Висновок**

Під час виконання даної лабораторної роботи, я ознайомився з базовим патерном 3D проєктів у Unity, навчився керувати камерою, створювати користувацький інтерфейс. Ознайомився із методами Unity та синтаксисом C#.

### 3. Додатки

Посилання на проекти – [GitHub](#).

Код **MovementPlayer.cs**:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class MovementPlayer : MonoBehaviour
{
    private CharacterController controller;

    private Vector3 velocity;
    private bool isGrounded;

    [SerializeField]
    private float speed = 5f;
    [SerializeField]
    private float rotationSpeed = 2f;
    [SerializeField]
    private Camera followCamera;
    [SerializeField]
    private float jumpForce = 1.0f;
    [SerializeField]
    private float gravity = -9.81f;

    private void Start()
    {
        controller = GetComponent<CharacterController>();
    }

    private void Update()
    {
        isGrounded = controller.isGrounded;
        if (isGrounded && velocity.y < 0)
        {
            velocity.y = 0f;
        }

        float horizontal = Input.GetAxis("Horizontal");
        float vertical = Input.GetAxis("Vertical");

        Vector3 movement = Quaternion.Euler(0,
followCamera.transform.eulerAngles.y, 0) * new Vector3(horizontal, 0, vertical);
        Vector3 direction = movement.normalized;
```

```

        controller.Move(direction * speed * Time.deltaTime);

        if (direction != Vector3.zero)
        {
            Quaternion desiredRotation = Quaternion.LookRotation(direction,
Vector3.up);
            transform.rotation = Quaternion.Slerp(transform.rotation,
desiredRotation, rotationSpeed * Time.deltaTime);
        }

        if (Input.GetKey(KeyCode.Space) && isGrounded)
        {
            velocity.y += Mathf.Sqrt(jumpForce * -3.0f * gravity);
        }
        velocity.y += gravity * Time.deltaTime;
        controller.Move(velocity * Time.deltaTime);

        Scene scene = SceneManager.GetActiveScene();
        if (scene.name == "Level1" && ScoreManager.scoreValue == 3)
SceneManager.LoadScene(1);
    }

    void OnTriggerEnter(Collider collision)
    {
        if (collision.gameObject.tag == "Keys")
        {
            ScoreManager.scoreValue += 1;
            Destroy(collision.gameObject);
        }
    }
}

```

Код **ScoreManager.cs**:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class ScoreManager: MonoBehaviour
{
    public static int scoreValue = 0;
    Text scoreObject;

    void Start()
    {
        scoreObject = GetComponent<Text>();
    }

    void Update()

```



```

    {
        scoreObject.text = "Score: " + scoreValue;
    }
}

```

Код **CameraController.cs**:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CameraController : MonoBehaviour
{
    [SerializeField]
    private Transform player;

    [SerializeField]
    private float distanceToPlayer = 3.0f;

    void Update()
    {
        transform.position = player.position - transform.forward *
distanceToPlayer;
    }
}

```