



**Національний Технічний Університет України КПІ**

Факультет Інформатики та Обчислювальної Техніки  
Кафедра інформатики та програмної інженерії

**Лабораторна робота №2**

З дисципліни «Розробка ігрових застосунків. Unity рішення»

**Перевірила:**

Викладач

Катін Павло Юрійович

**Виконав:**

Студент групи ІО-93

Ващенко І.О.

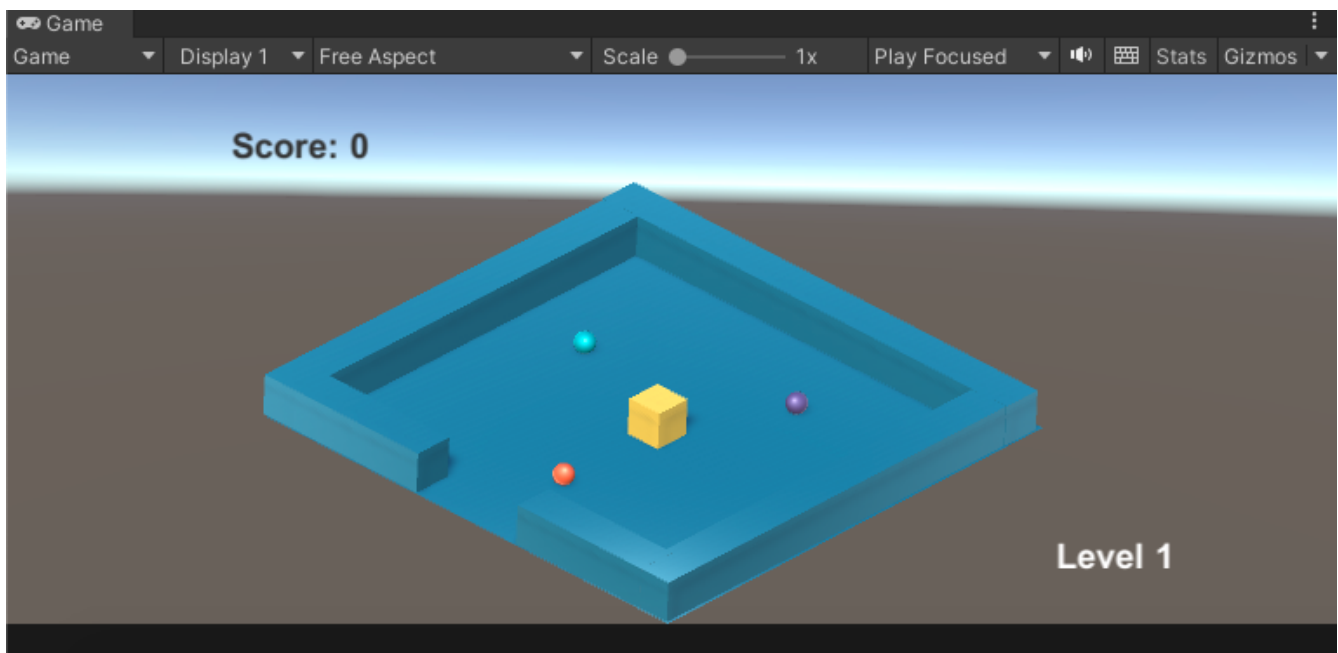
## Завдання лабораторної роботи

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.

Проект розташовано у репозиторій на GitHub, основна мета полягає у дослідженні і підтвердженні володіння обраною IDE (3D) і технологією розподіленої системи контролю версій.

## Виконання лабораторної роботи

- 1) Створимо проект. Для цього у Unity Hub ми тиснемо кнопку “New project” та введемо всі дані.
- 2) Створюємо дизайн гри, де ми розміщуємо нашого головного грака та додаткові об'єкти для гри.



- 2) Також додаємо скрипт, де ми визначимо поведінку нашого об'єкту в залежності від ситуації. Код виглядає наступним чином:

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

```

using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class Player : MonoBehaviour
{
    public float speed = 5f;
    private Rigidbody _rb;

    public Text scoreText;

    private int _score = 0;

    private void Awake()
    {
        _rb = GetComponent<Rigidbody>();
    }

    private void FixedUpdate()
    {
        float h = Input.GetAxis("Horizontal");
        float v = Input.GetAxis("Vertical");

        transform.Translate(Vector3.forward * Time.fixedDeltaTime * speed * h);
        transform.Translate(Vector3.right * Time.fixedDeltaTime * speed * -v);
    }

    private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.tag == "CollectrSphere")
        {
            _score++;
            Destroy(other.gameObject);
            if(_score != 5 ) {
                scoreText.text = "Score: " + _score;
            }
            else {
                scoreText.text = "You win!";
            }
        }
        if(this.CompareTag("Player") && other.CompareTag("Finish"))
        {
            SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
        }
    }
}

```

1-створюємо поля для пересування гравця, це speed. І будемо використовувати посилання на компонент Rigidbody

2-метод Awake всередині якого будемо встановлювати значення для поля \_rb. Тобто отримуватимемо компонент Rigidbody

3-FixedUpdate отримуємо при натискання користувача по горизонталі та вертикалі.

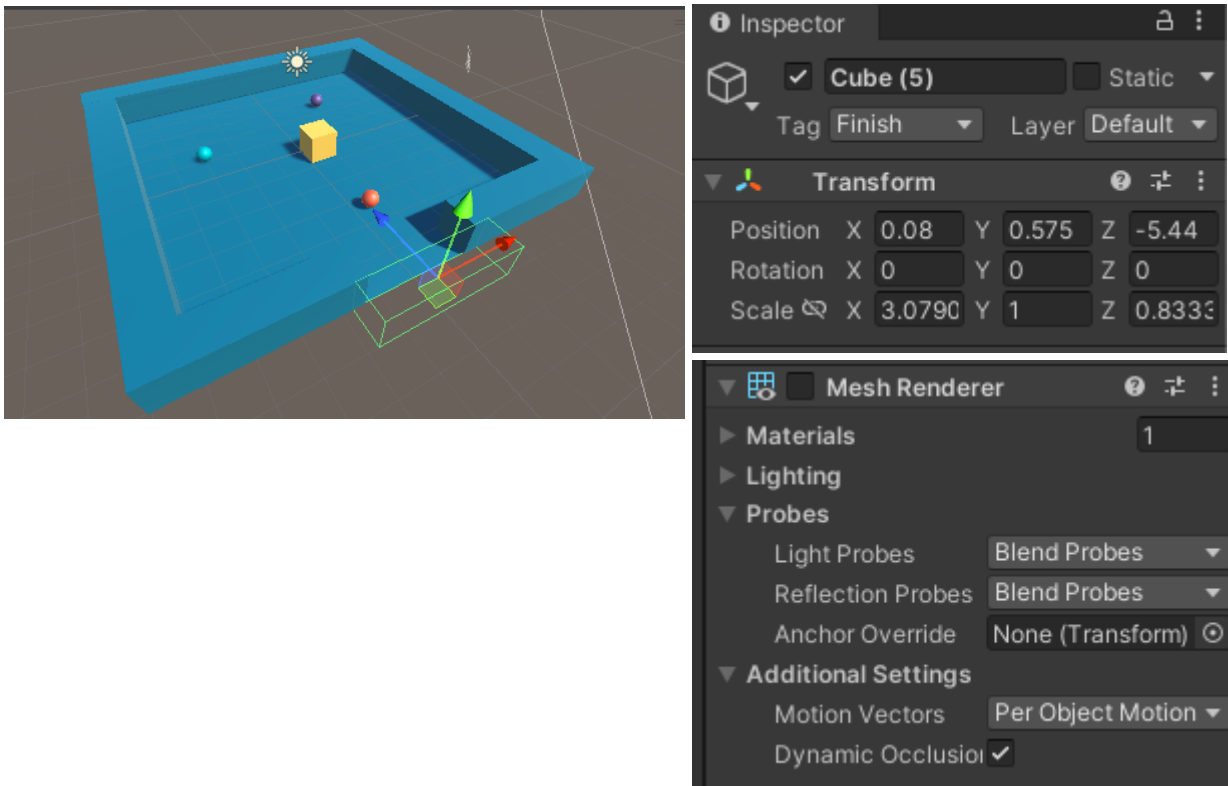
4-у методі OnTriggerEnter він буде спрацьовувати щоразу, коли гравець торкатиметься з тригерами (тобто з нашими об'єктами). Приписуємо умову, де ми

звертаємося до того об'єкта, якому доторкнулися, записуємося в параметрі `other`, звертаємось до нього як ігрового об'єкта та звертаємось до його тегу.

5-За допомогою методу `Destroy` ми видаляємо об'єкт з яким зіткнулися.

6-щоб працювати з текстом, необхідно підключити бібліотеку `UnityEngine.UI` - за допомогою цієї бібліотеки ми отримали доступ додаткових класів, які можуть посилатися до таких компонентів як текст.

Для цієї гри було створено дві сцени та два нескладних рівня. Щоб гравець міг перейти на іншу сцену, я застосувала новий об'єкт зробивши його прозорим за допомогою `Mesh Renderer` і тег `Finish`.



## Висновок:

У цій лабораторній роботі ми навчилися розробленню основ проекту з використанням обраної мови програмування у обраній парадигмі. Також ми отримали основні навички розробки з використанням IDE ігрового рушія.