**Міністерство освіти і науки України**

**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: Перевірив:

студент групи ІП-93 Катін П.Ю.

Дмитренко Р.В.

Номер у списку: 5

Київ 2022

Зміст

[1. Постановка задачі 3](#_Toc115112603)

[2. Виконання 3](#_Toc115112604)

[Хід роботи 3](#_Toc115112605)

[Скріншоти 4](#_Toc115112606)

[3. Висновок 5](#_Toc115112607)

[4. Додатки 5](#_Toc115112608)

# 1. Постановка задачі

**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.

**Варіант:** 5

**Завдання до 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, набір балів і перехід з одного рівня на інший.
* Можливо, за бажанням реалізувати інтерфейс гравця.

**Завдання відповідно до варіанту:**

1. Примітив: куб.

2. Набор асетів: <https://assetstore.unity.com/packages/3d/environments/landscapes/low-poly-simple-nature-pack-162153>

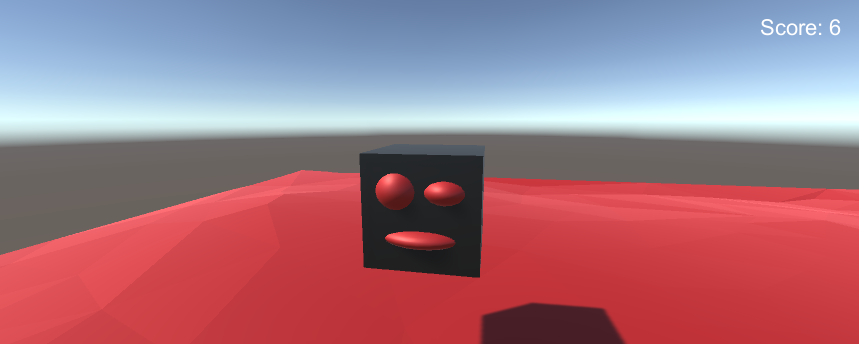
# 2. Виконання

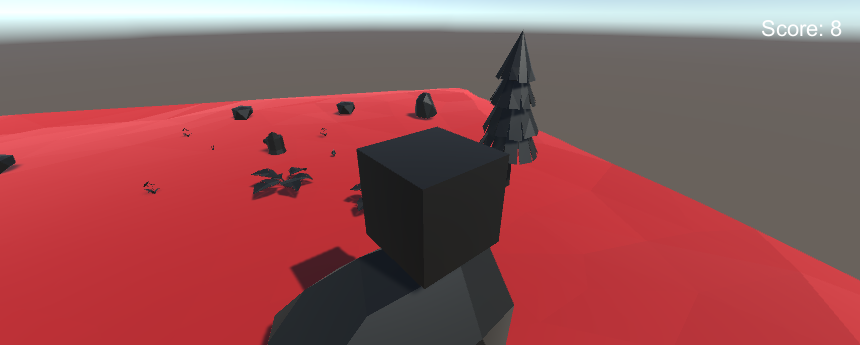
### Хід роботи

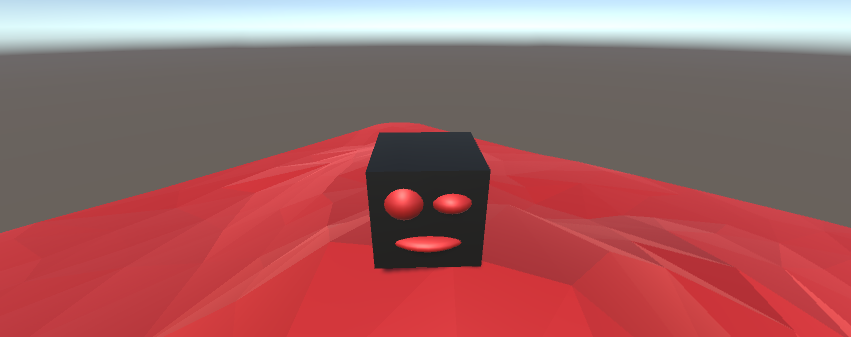
Для виконання даної лабораторної роботи, було створено проект. В ньому: об'єкт-гравець, ділянка, на якій він буде рухатись, створена за допомогою асетів відповідно до варіанту, лічильник, який встановлює кількість стрибків гравця для отримання очок та камера.

За допомогою скриптів, робимо так, щоб камерою можна було рухати прямо під час виконання сцени. Далі прикріплюємо цю камеру до гравця, щоб вона стежила за гравцем. Після цього, прописуємо базові рухи для гравця: вправо, вліво, вперед, назад, оберт на 360 і стрибок. В кінці, реалізовуємо, щоб при кожному стрибку лічильник збільшувався на 1 та, щоб при досягненні 10 стрибків, гра перемикалась на наступну сцену. Весь код реалізації програми наведений у додатках.

### Скріншоти







# 3. Висновок

Виконуючи дану лабораторну роботу, було набуто знання, уміння та навички з технології розроблення основ проекту з використанням обраної мови програмування у обраній парадигмі. Був отриманий досвід створення репозиторію у системі контролю версій і знання елементів середовища розробки і основи вихідного коду для управління грою.

# 4. Додатки

Код програми:  
CameraController.cs:  
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class CameraController : MonoBehaviour

{

    private float rotationY;

    private float rotationX;

    private Vector3 currentRotation;

    private Vector3 smoothVelocity = Vector3.zero;

    [SerializeField]

    private float sensitivity = 3.0f;

    [SerializeField]

    private Transform target;

    [SerializeField]

    private float distanceFromTarget = 3.0f;

    [SerializeField]

    private float smoothTime = 0.2f;

    [SerializeField]

    private Vector2 rotationXMinMax = new Vector2(-40, 40);

    void Update()

    {

        float mouseX = Input.GetAxis("Mouse X") \* sensitivity;

        float mouseY = Input.GetAxis("Mouse Y") \* sensitivity;

        rotationY += mouseX;

        rotationX += mouseY;

        rotationX = Mathf.Clamp(rotationX, rotationXMinMax.x, rotationXMinMax.y);

        Vector3 nextRotation = new Vector3(rotationX, rotationY);

        currentRotation = Vector3.SmoothDamp(currentRotation, nextRotation, ref smoothVelocity, smoothTime);

        transform.localEulerAngles = currentRotation;

        transform.position = target.position - transform.forward \* distanceFromTarget;

    }

}

PlayerMovement.cs:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class PlayerMovement : MonoBehaviour

{

    private CharacterController controller;

    private Vector3 playerVelocity;

    private bool groundedPlayer;

    [SerializeField]

    private float speed = 5f;

    [SerializeField]

    private float rotationSpeed = 10f;

    [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()

    {

        groundedPlayer = controller.isGrounded;

        if (groundedPlayer && playerVelocity.y < 0)

        {

            playerVelocity.y = 0f;

        }

        float horizontalInput = Input.GetAxis("Horizontal");

        float verticalInput = Input.GetAxis("Vertical");

        Vector3 movementInput = Quaternion.Euler(0, followCamera.transform.eulerAngles.y, 0) \* new Vector3(horizontalInput, 0, verticalInput);

        Vector3 movementDirection = movementInput.normalized;

        controller.Move(movementDirection \* speed \* Time.deltaTime);

        if (movementDirection != Vector3.zero)

        {

            Quaternion desiredRotation = Quaternion.LookRotation(movementDirection, Vector3.up);

            transform.rotation = Quaternion.Slerp(transform.rotation, desiredRotation, rotationSpeed \* Time.deltaTime);

        }

        if (Input.GetKey(KeyCode.Space) && groundedPlayer)

        {

            playerVelocity.y += Mathf.Sqrt(jumpForce \* -3.0f \* gravity);

            TextManager.scoreValue += 1;

        }

        Scene scene = SceneManager.GetActiveScene();

        if (scene.name == "SampleScene" && TextManager.scoreValue == 10) SceneManager.LoadScene(1);

        playerVelocity.y += gravity\* Time.deltaTime;

        controller.Move(playerVelocity \* Time.deltaTime);

    }

}

TextManager.cs:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class TextManager : MonoBehaviour

{

    public static int scoreValue = 0;

    Text score;

    // Start is called before the first frame update

    void Start()

    {

        score = GetComponent<Text>();

    }

    // Update is called once per frame

    void Update()

    {

        score.text = "Score: " + scoreValue;

    }

}