SPRAWOZDANIE

Zajęcia: Grafika i Multimedia

Prowadzący: prof. dr hab. inż. Vasyl Martsenyuk

|  |  |
| --- | --- |
| Laboratorium Nr 1  Data 14.10.2023  Temat: Modelowanie gry komputerowej "Roll and Ball"  Wariant 3 | Adam Kubliński  Informatyka  II stopień, stacjonarne,  1 semestr, gr.1A |

1. Polecenie: wariant 3 zadania

Kolor materialu „playera” – zielony,

obiekty „pick up” dwóch typów

1 typ obiekta „pick up” - capsule,

2 typ obiekta „pick up” - cylinder,

ilość obiektów „pick up” typu 1 - 7,

ilość obiektów „pick up” typu 2 - 5,

Kolor materialu obiektów „pick up” typu 1 – żółty,

Kolor materialu obiektów „pick up” typu 2 – brązowy,

Kolor materialu „ścian” - niebieski

Reguły gry:

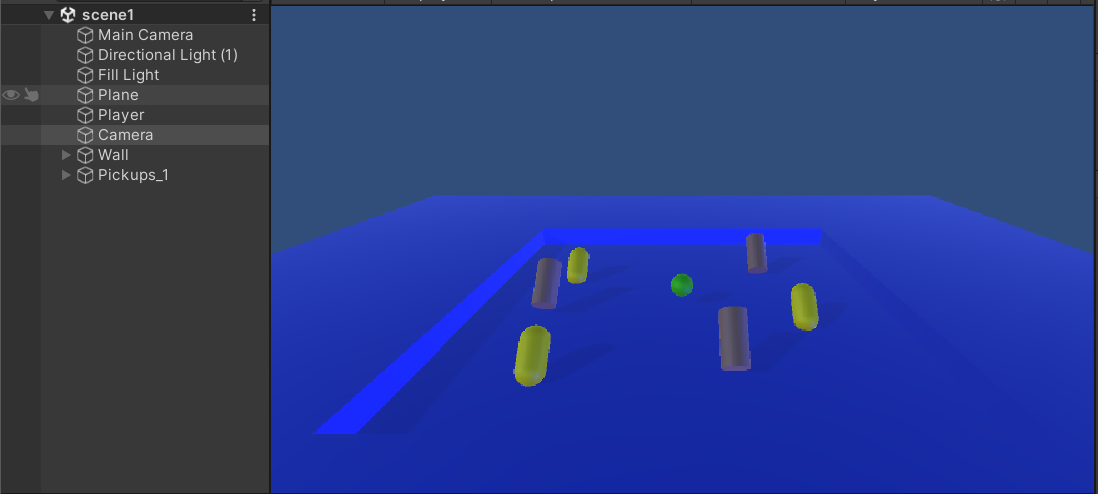
trafianie w obiekt typu 1 – 1 punkt

trafianie w obiekt typu 2 – 4 punkty

warunek zakończenia gry – 9 punktów

2. Opis programu opracowanego (kody źródłowe, rzuty ekranu)

2.1. Widok z gry



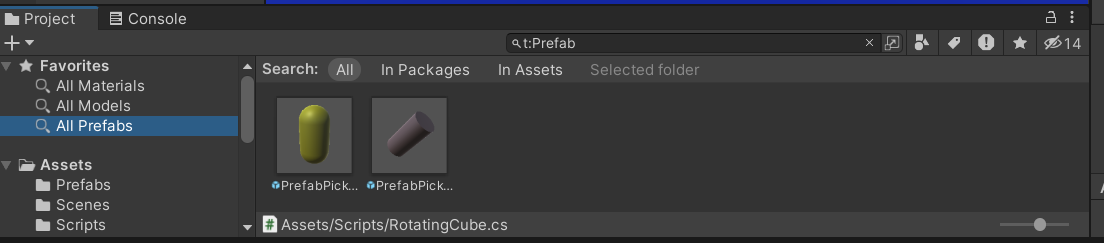
2.2 Stworzone obiekty typu „PreFab”

1 typ obiekta „pick up” - capsule,

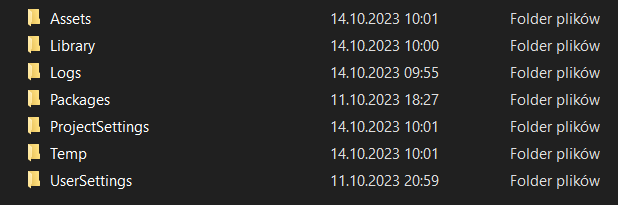
2 typ obiekta „pick up” - cylinder,

Kolor materialu obiektów „pick up” typu 1 – żółty,

Kolor materialu obiektów „pick up” typu 2 – brązowy,



2.3 Struktura folderów:



2.4 Skrypt „CameraScript”

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class CameraScript : MonoBehaviour

{

    public GameObject player;

    public Vector3 offset;

    void Start()

    {

        void LateUpdate()

        {

            transform.position = player.transform.position + offset;

        }

    }

}

2.5 Skrypt „NewBahaviourScript”

using UnityEngine;

public class NewBehaviourScript : MonoBehaviour {

    public float speed;

    private int count;

    void Start()

    {

        count = 0;

    }

    void Update () {

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

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

        Vector3 movement = new Vector3 (moveHorizontal, 0.0f, moveVertical);

        GetComponent<Rigidbody>().AddForce (movement \* speed \* Time.deltaTime);

        if(count >= 9)

        {

            Stop();

        }

    }

    void OnTriggerEnter(Collider other)

    {

        if(other.gameObject.tag == "PickupCapsule")

        {

            other.gameObject.SetActive(false);

            count += 1;

        }

        if(other.gameObject.tag == "PickupCylinder")

        {

            other.gameObject.SetActive(false);

            count += 4;

        }

    }

    void Stop()

    {

        Debug.Log("Game finished");

    }}

2.6 Skrypt „RotatingCube”

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class RotatingCube : MonoBehaviour

{

    void Update ()

    {

        transform.Rotate (new Vector3 (15, 30, 45) \* Time.deltaTime);

    }

}

3. Wnioski

Celem laboratorium było zapoznanie się i stworzenie gry Roll and Ball w oparciu o instrukcję i z zachowaniem wymagań według wariantu 3. Gra została zaprojektowana pod względem wizualnym zgodnie z wytycznymi. Zostały zaimplementowane skrypty definiujące reguły gry jak i sposób poruszania się obiektów. Po przetestowaniu gra działa prawidłowo.