

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ”

Расчетно-графическая работа
“раннер на Unity 3D”

Выполнил: студент группы ИП-014

Обухов А.И.

Работу проверил: старший преподаватель кафедры ПМиК

Павлова У.В.

Новосибирск 2023

Оглавление:

Цель работы:	2
Листинг программы:	3
Приложение:	15
Вывод:	18

Цель работы:

Разработать кроссплатформенную игру, где игрок бежит по бесконечно-генерируемому коридору, собирая монеты, обегая и перепрыгивая препятствия: машины и шлагбаумы, пока время, отведенное на игру, не закончится. Игра заканчивается по истечению минуты или при коллизии объектов персонажа и препятствия.

Расположение префабов-препятствий и монет на сцене фиксировано. Игрок управляет персонажем кнопками на клавиатуре (перемещает его вправо, влево или совершает прыжок), время игры - минута, за это время необходимо собрать как можно больше монет или пробежать большую дистанцию. Время, дистанция и количество собранных монет отображаются в верхнем левом углу экрана. Игра заканчивается досрочно, если происходит столкновение персонажа с префабом-машиной или шлагбаумом. Главной особенностью раннера является то, что фигурка игрока по факту стоит на месте, а все действие происходит специальным объектом под названием Destroyer, который удаляет префабы-препятствий и карты в целом и спавнит их впереди игрока.

Листинг программы:

MainMenuFunctions.cs

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

public class MainMenuFunction : MonoBehaviour
{
    void Start()
    {

    }

    void Update()
    {

    }

    public void PlayGame()
    {
        SceneManager.LoadScene(1);
    }
}
```

PlayerMove.cs

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

public class PlayerMove : MonoBehaviour
{
    public float moveSpeed = 3;
    public float leftRightSpeed = 4;
    public static bool canMove = false;
    public bool isJumping = false;
    public bool comingDown = false;
```

```

public GameObject playerObject;

void Update()
{
    transform.Translate(Vector3.forward * Time.deltaTime * this.moveSpeed,
Space.World);
    if (canMove == true)
    {
        if (Input.GetKey(KeyCode.A) || Input.GetKey(KeyCode.LeftArrow))
        {
            if (this.gameObject.transform.position.x > LevelBoundary.leftSide)
            {
                transform.Translate(Vector3.left * Time.deltaTime *
leftRightSpeed);
            }
        }

        if (Input.GetKey(KeyCode.D) || Input.GetKey(KeyCode.RightArrow))
        {
            if (this.gameObject.transform.position.x < LevelBoundary.rightSide)
            {
                transform.Translate(Vector3.left * Time.deltaTime * leftRightSpeed
* -1);
            }
        }

        if (Input.GetKey(KeyCode.W) || Input.GetKey(KeyCode.RightArrow) ||
Input.GetKey(KeyCode.Space))
        {
            if (isJumping == false)
            {
                isJumping = true;
                playerObject.GetComponent<Animator>().Play("Jump");
                StartCoroutine(JumpSequence());
            }
        }
    }
}

```

```

    if (Input.touchCount > 0)
    {
        Touch touch = Input.GetTouch(0);
        if (touch.position.x < Screen.width / 2)
        {
            if (this.gameObject.transform.position.x > LevelBoundary.leftSide)
            {
                transform.Translate(Vector3.left * Time.deltaTime *
leftRightSpeed);
            }
        }
        if (touch.position.x > Screen.width / 2)
        {
            if (this.gameObject.transform.position.x <
LevelBoundary.rightSide)
            {
                transform.Translate(Vector3.left * Time.deltaTime *
leftRightSpeed * -1);
            }
        }

        if (Input.touchCount >= 2)
        {
            if (isJumping == false)
            {
                isJumping = true;
                playerObject.GetComponent<Animator>().Play("Jump");
                StartCoroutine(JumpSequence());
            }
        }
    }

    if (isJumping == true)
    {
        if (comingDown == false)

```

```

        {
            transform.Translate(Vector3.up * Time.deltaTime * 5,
Space.World);
        }
        if (comingDown == true)
        {
            transform.Translate(Vector3.up * Time.deltaTime * -5,
Space.World);
        }
    }
}

```

```

IEnumerator JumpSequence()
{
    float jumpTime = 0.5f;
    yield return new WaitForSecondsRealtime(jumpTime);
    comingDown = true;
    yield return new WaitWhile(() => transform.position.y > -3.19f);
    isJumping = false;
    comingDown = false;
    if (canMove)
    {
        playerObject.GetComponent<Animator>().Play("Standard Run");
    }
}

```

CollectableControl.cs

```

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

```

```

public class CollectableControl : MonoBehaviour
{
    public static int score = 0;
    public GameObject scoreDisplay;
    public GameObject coinEndDisplay;

    // Update is called once per frame
    void Update()
    {
        if (scoreDisplay != null)
        {
            scoreDisplay.GetComponent<TextMeshProUGUI>().text =
score.ToString();
            coinEndDisplay.GetComponent<TextMeshProUGUI>().text =
score.ToString();
        }
    }
}

```

CollectCoin.cs

```

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

public class CollectCoin : MonoBehaviour
{
    public AudioSource coinFX;

    void OnTriggerEnter(Collider other)
    {
        coinFX.Play();
        CollectableControl.score += 1;
        this.gameObject.SetActive(false);
    }
}

```


LevelDistance.cs

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

```
public class LevelDistance : MonoBehaviour
{
```

```
    public GameObject distanceDisplay;
    public GameObject distanceEndDisplay;
    public int distanceRan;
    public bool addingDistance = false;
```

```
    // Update is called once per frame
```

```
    void Update()
```

```
    {
        if (addingDistance == false)
        {
            addingDistance = true;
            StartCoroutine(AddingDistance());
        }
    }
}
```

```
IEnumerator AddingDistance()
```

```
{
    distanceRan += 1;
    distanceDisplay.GetComponent<TextMeshProUGUI>().text =
distanceRan.ToString();
    distanceEndDisplay.GetComponent<TextMeshProUGUI>().text =
distanceRan.ToString();
```

```
    yield return new WaitForSeconds(0.25f);
```

```

        addingDistance = false;
    }

}

RotateObject.cs
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class RotateObject : MonoBehaviour
{
    public int rotateSpeed = 1;
    void Update()
    {
        transform.Rotate(0, rotateSpeed, 0, Space.World);
    }
}

```

```

EndRunSequence.cs
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class EndRunSequence : MonoBehaviour
{
    public GameObject liveCoins;
    public GameObject liveDistance;
    public GameObject liveTime;
    public GameObject endScreen;
    public GameObject fadeOut;

    // Start is called before the first frame update
    void Start()
    {
        StartCoroutine(EndSequence());
    }
}

```

```

    }

IEnumerator EndSequence()
{
    yield return new WaitForSeconds(3);
    liveCoins.SetActive(false);
    liveDistance.SetActive(false);
    liveTime.SetActive(false);
    endScreen.SetActive(true);
    yield return new WaitForSeconds(3);
    fadeOut.SetActive(true);
    yield return new WaitForSeconds(2);
    SceneManager.LoadScene(0);
}
}

```

GenerateLevel.cs

```

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

public class GenerateLevel : MonoBehaviour
{

    public GameObject[] section;
    public int zPos = 40;
    public bool creatingSection = false;
    public int sectionNumber;

    // Update is called once per frame
    void Update()
    {
        if (creatingSection == false)
        {
            creatingSection = true;
            StartCoroutine(GenerateSection());
        }
    }
}

```

```

    }

    IEnumerator GenerateSection()
    {
        sectionNumber = Random.Range(0, 3);
        Instantiate(section[sectionNumber], new Vector3(0, 0, zPos),
Quaternion.identity);
        zPos += 40;
        yield return new WaitForSeconds(1);
        creatingSection = false;
    }
}

```

LevelBoudary.cs

```

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

public class LevelBoundary : MonoBehaviour
{
    public static float leftSide = -5;
    public static float rightSide = 2.3f;
    public float internalLeft;
    public float internalRight;

    // Update is called once per frame
    void Update()
    {
        internalLeft = leftSide;
        internalRight = rightSide;
    }
}

```

LevelStarter.cs

```

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

public class LevelBoundary : MonoBehaviour
{
    public static float leftSide = -5;
    public static float rightSide = 2.3f;
    public float internalLeft;
    public float internalRight;

    // Update is called once per frame
    void Update()
    {
        internalLeft = leftSide;
        internalRight = rightSide;
    }
}

```

ObstacleCollision/cs

```

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

public class LevelBoundary : MonoBehaviour
{
    public static float leftSide = -5;
    public static float rightSide = 2.3f;
    public float internalLeft;
    public float internalRight;

    // Update is called once per frame
    void Update()
    {

```

```

        internalLeft = leftSide;
        internalRight = rightSide;

    }
}

```

TimeControl.cs

```

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

public class TimeControl : MonoBehaviour
{
    public int time = 60;
    public GameObject playerModel;
    public GameObject player;
    public GameObject levelControl;
    public GameObject timeDisplay;

    void Start()
    {
        StartCoroutine(Timer());
    }

    IEnumerator Timer()
    {
        while (time > 0)
        {
            time--;
            timeDisplay.GetComponent<TextMeshProUGUI>().text =
time.ToString();
            yield return new WaitForSeconds(1);
        }
    }
}

```

```
playerModel.SetActive(false);
player.GetComponent<PlayerMove>().enabled = false;
PlayerMove.canMove = false;
levelControl.GetComponent<EndRunSequence>().enabled = true;
levelControl.GetComponent<LevelDistance>().enabled = false;

}
}
```

Приложение:



Рис.1 Начальный экран



Рис.2 Геймплей

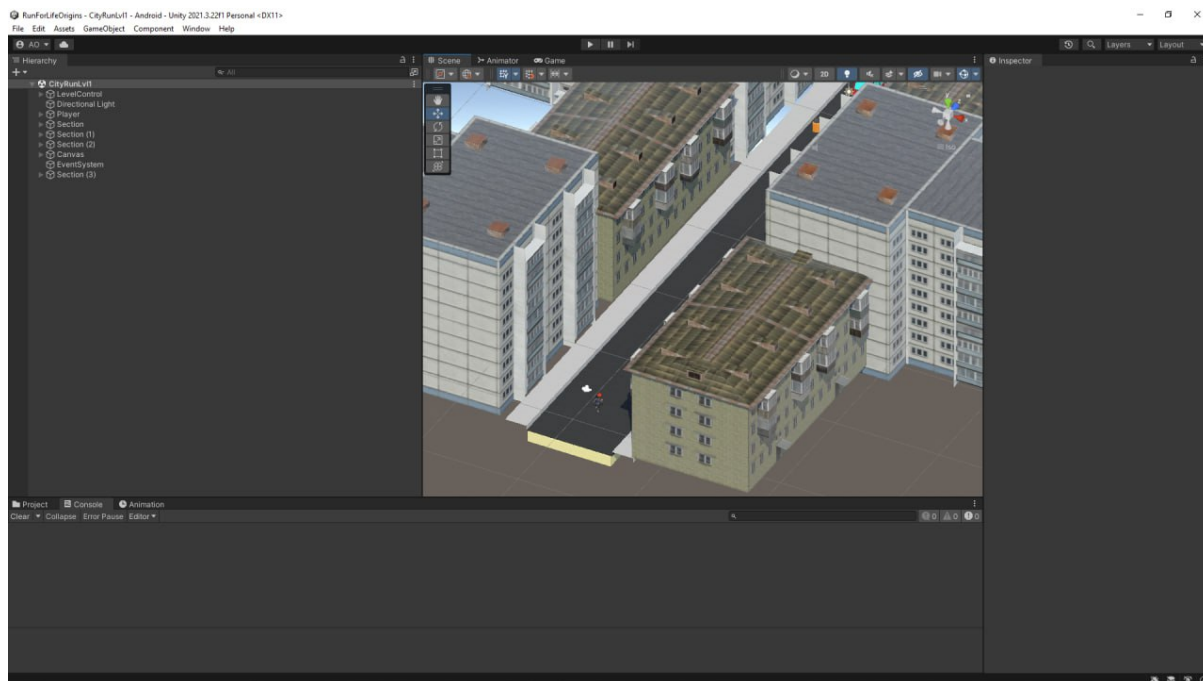


Рис.3 Сцена сверху

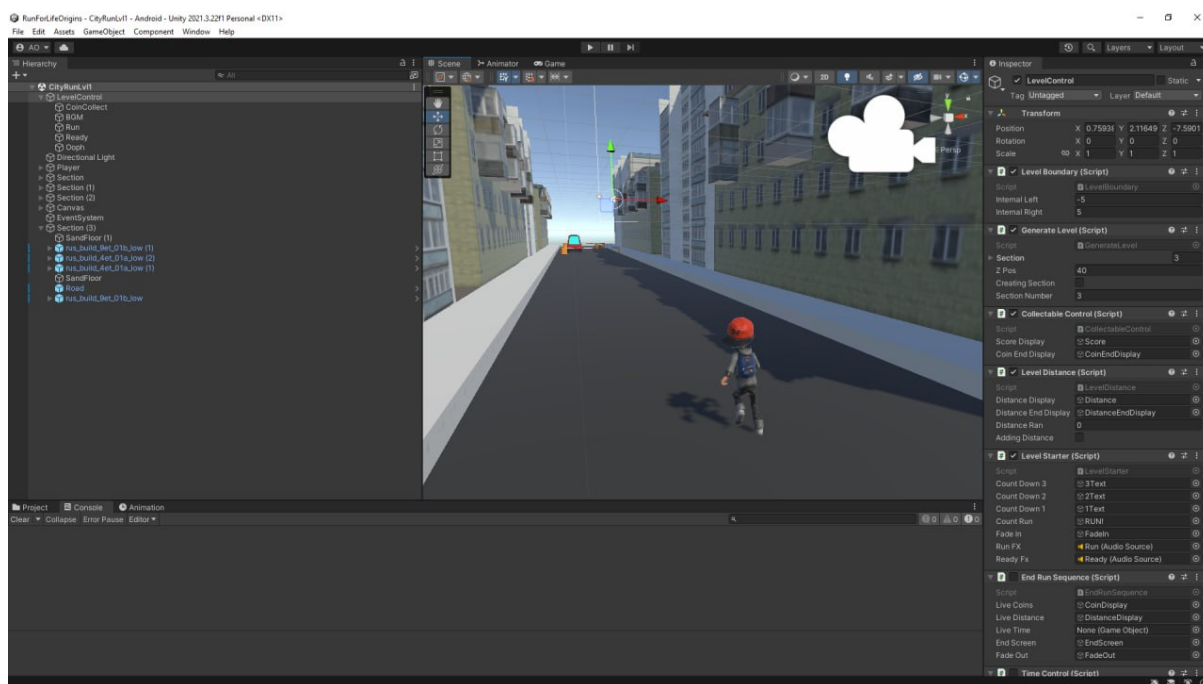


Рис.4 Объект-камера прикреплен к объекту-персонажу игрока



Рис.5 Конец игры

Вывод:

В ходе выполнения расчетного-графического задания на языке C# и движка Unity 3D, было создано простое приложение при помощи инструментов, позволяющих создавать игры в 2D и 3D пространстве, была изучена особенность работы с текстурами, объектами, префабами и спрайтами. Данное приложение можно также запускать и на андроид, что делает его кроссплатформенным.