Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1-3

По дисциплине: «ОСиСП»

Выполнил:

Студент 3 курса

Группы ПО-4(1)

Боровский М.В.

Проверил:

Дряпко А.В.

Брест 2021

Лабораторная работа № 1-3

Вариант 3:

**Цель**: приобрести практические навыки проектирования и разработки приложений с графическим пользовательским интерфейсом в ОС Windows.

**Задание к 1-2 лабораторной работе**: Игра «Змейка». Один уровень игры с начислением очков и 3 типами бонусов (начисляется разное количество очков за каждый тип бонуса).

**Задание к 3 лабораторной работе**:

**1.**Реализовать окно «О программе» в виде объекта динамической библиотеки about.dll с указанием автора программы, группы, курса и краткого описания разработанного приложения.

**2.**Реализовать расширения для приложения, позволяющие изменять оформление пунктов меню (шрифт, размер, начертание и т.д.). Соответствующие изменения должны происходить при выборе специального пункта меню. Создать как минимум три расширения такого типа.

Осуществить импорт указанной библиотеки и отображение соответствующего окна при выборе пункта меню «О программе».

**Код**:

**Snake.cs**:

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

using UnityEngine.UI;

public class Snake : MonoBehaviour

{

    private Vector2 \_diraction = Vector2.right;

    private List<Transform> \_segments = new List<Transform>();

    public Transform segmentPrefab;

    public int initialSize = 4;

    public int Score = 0;

    public GameObject ScoreText;

    public void Count()

    {

        Score++;

        ScoreText.GetComponent<Text>().text = "Score: " + Score.ToString();

        if(Score > 10)

        {

            Time.timeScale = Score/10;

        }

    }

    private void Start()

    {

        ResetState();

    }

    private void Update()

    {

            if(Input.GetKeyDown(KeyCode.W) && \_diraction != Vector2.down) {

                \_diraction = Vector2.up;

            }else if(Input.GetKeyDown(KeyCode.S) && \_diraction != Vector2.up) {

                \_diraction = Vector2.down;

            }else if(Input.GetKeyDown(KeyCode.A) && \_diraction != Vector2.right) {

                \_diraction = Vector2.left;

            }else if(Input.GetKeyDown(KeyCode.D) && \_diraction != Vector2.left) {

                \_diraction = Vector2.right;

            }else if(Input.GetKeyDown(KeyCode.R)) {

                SceneManager.LoadScene(1);

        }

    }

    private void  FixedUpdate()

    {

        for (int i = \_segments.Count - 1; i > 0; i--)

        {

            \_segments[i].position = \_segments[i - 1].position;

        }

        this.transform.position = new Vector3(

            Mathf.Round(this.transform.position.x) + \_diraction.x,

            Mathf.Round(this.transform.position.y) + \_diraction.y,

            0.0f

        );

    }

    private void Grow()

    {

        Transform segment = Instantiate(this.segmentPrefab);

        segment.position = \_segments[\_segments.Count - 1].position;

        \_segments.Add(segment);

        Count();

    }

    private void ResetState()

    {

        for (int i = 1; i < \_segments.Count; i++)

        {

            Destroy(\_segments[i].gameObject);

        }

            \_segments.Clear();

            \_segments.Add(this.transform);

            for (int i = 1; i < this.initialSize; i++)

            {

                \_segments.Add(Instantiate(this.segmentPrefab));

            }

            this.transform.position = Vector3.zero;

    }

    private void OnTriggerEnter2D(Collider2D other)

    {

        if(other.tag == "Food") {

        Grow();

        } else if (other.tag == "Obstacle") {

            Time.timeScale = 0f;

        }

    }

}

**PauseMenu.cs**:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class PauseMenu : MonoBehaviour

{

    public void MainMenu()

    {

        SceneManager.LoadScene(0);

    }

    public void Restart()

    {

        SceneManager.LoadScene(1);

        Time.timeScale = 0.5f;

    }

}

**MainMenu.cs**:

using System.Collections;

using System.Collections.Generic;

using UnityEngine.UI;

using UnityEngine.SceneManagement;

using UnityEngine;

using Snakedll;

public class MainMenu : MonoBehaviour

{

    public Text text;

    public About sdll = new About();

    public GameObject textbox;

    public Text starttext;

    public Text abouttext;

    public Text exittext;

    public void start(int index)

    {

        SceneManager.LoadScene(index);

    }

    public void About()

    {

        text.text = sdll.Information();

        textbox.SetActive(!textbox.activeSelf);

    }

    public void Exit()

    {

        Application.Quit();

    }

    public void Bold()

    {

        starttext.fontStyle = FontStyle.Bold;

        abouttext.fontStyle = FontStyle.Bold;

        exittext.fontStyle = FontStyle.Bold;

    }

    public void Default()

    {

        starttext.fontStyle = FontStyle.Normal;

        abouttext.fontStyle = FontStyle.Normal;

        exittext.fontStyle = FontStyle.Normal;

    }

}

**Food.cs**:

using UnityEngine;

public class Food : MonoBehaviour

{

    public BoxCollider2D gridArea;

    private void Start()

    {

        RandomizePosition();

    }

    private void RandomizePosition()

    {

        Bounds bounds = this.gridArea.bounds;

        float x = Random.Range(bounds.min.x, bounds.max.x);

        float y = Random.Range(bounds.min.y, bounds.max.y);

        this.transform.position = new Vector3(Mathf.Round(x), Mathf.Round(y), 0.0f);

    }

    private void OnTriggerEnter2D(Collider2D other)

    {

        if(other.tag == "Player") {

        RandomizePosition();

        }

    }

}

**Snakedll.cs**:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using UnityEngine;

namespace Snakedll

{

public class About

{

public string Information()

{

return "Данная игра сделана \n" +

"Боровским Максимам ПО-4\n" +

"Змейка их уже много но надо!";

}

}

}

**Результат**:







