Донецкий Национальный Технический Университет

Лабораторная работа № 1

«Создание игры для мобильного устройства»

Выполнил:

ст. группы ПИ-13

Лысенко А. С.

Проверил:

ст. препод. Костин В.И.

Покровск 2017

Задание: Создать игру для мобильного устройства (КПК, Smartphone и т.п.), используя любой язык программирования для мобильных устройств (C#, C++, J2ME, Python и т.п.). В игре обязательно должен выполняться подсчёт очков. Рекордное количество очков (10 рекордов) должно сохраняться в файле и по выбору пользователя выводиться в виде таблицы рекордов в отсортированном по количеству набранных очков виде. Если пользователь в результате игры набрал рекордное количество очков, то ему необходимо предоставить возможность ввести своё имя, а после ввода имени показать таблицу рекордов, в которой выделяется только что добавленная запись.

Игра заключается в том, чтобы выбрать цвет, показывающий на кубике сверху, в центре экрана. Подсказка: выбирайте самый темный из предложенных цвет.

В игре 3 сцены: главная(main), игровая(play) и справка (howTo), показывающая правила игры. Сцена игровая(play), содержит в себе панель проигрыша (panel lost), которая выплывает при проигрыше игрока.

Игра разрабатывалась в программной среде Unity, с использованием языка программирования C#.

Скриншоты сцен:

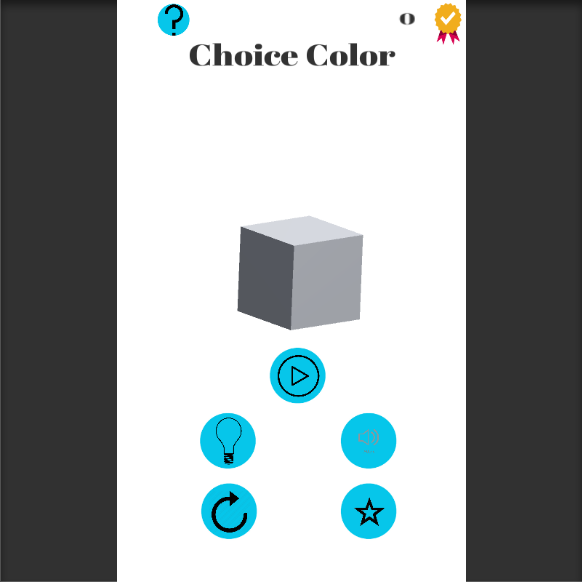


Рис. 1 сцена main

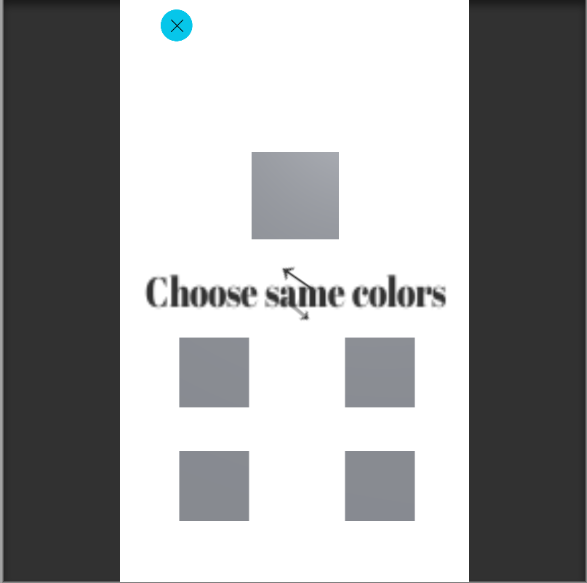


Рис. 2 сцена howTo

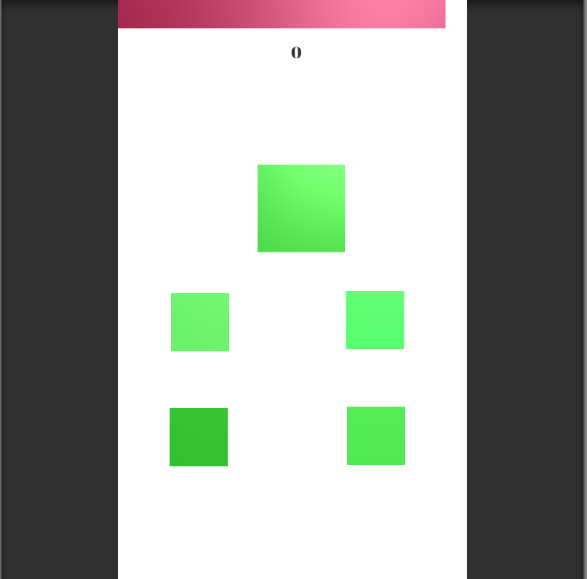


Рис. 3 сцена play

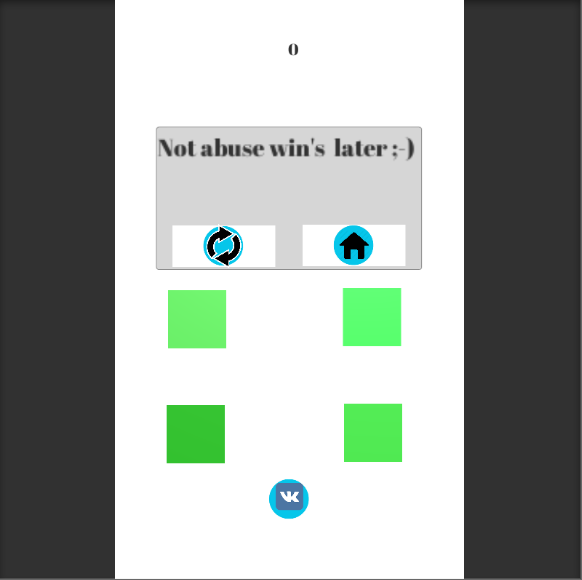


Рис. 3 сцена play, окно проигрыша (panel lost)

Скрипты, которые были использованы при написании игры:

Buttons.cs – отвечает за изменение цвета кнопок при нажатии и их функционал.

CubeFloat.cs – отвечает за движение куба в главной(main) сцене игры.

DontDestroy.cs – отвечает за переход между уровнями, на правильного цвета куб налаживаем звук, если звуки равны, уровень изменяется.

GameCntrl.cs – отвечает за генерацию цвета кубиков.

MaxScore.cs – отвечает за отображение счета в главной(main) сцене игры.

RandCol.cs – отвечает за изменение цвета кубика в главной(main) сцене игры.

RightOne.cs – отвечает за определение одного правильного кубика.

Timer.cs – отвечает за время, за которое нужно успеть выбрать кубик.

Buttons.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Buttons : MonoBehaviour

{

public GameObject m\_on, m\_off;

public Sprite layer\_blue, layer\_red;

public string action;

private void Start()

{

if (gameObject.name == "Music")

{

if (PlayerPrefs.GetString("Music") == "no")

{

m\_on.SetActive(false);

m\_off.SetActive(true);

}

else

{

m\_on.SetActive(true);

m\_off.SetActive(false);

}

}

}

void OnMouseUp()

{

GetComponent<SpriteRenderer>().sprite = layer\_blue;

}

private void OnMouseUpAsButton()

{

if (PlayerPrefs.GetString("Music") != "no")

{

GameObject.Find("Click Audio").GetComponent<AudioSource>().Play();

}

switch (gameObject.name)

{

case "Play":

Application.LoadLevel("Play");

break;

case "Rating":

Application.OpenURL("https://play.google.com/store/apps/details?id=com.DeusComp.ChoiceColor");

break;

case "Replay":

Application.LoadLevel("play");

break;

case "Home":

Application.LoadLevel("main");

break;

case "Vk":

Application.OpenURL("http://vk.com");

break;

case "How To":

Application.LoadLevel("howTo");

break;

case "Close":

Application.LoadLevel("main");

break;

case "Music":

if (PlayerPrefs.GetString("Music") != "yes")

{

PlayerPrefs.SetString("Music", "yes");

m\_on.SetActive(true);

m\_off.SetActive(false);

}

else

{

PlayerPrefs.SetString("Music", "no");

m\_on.SetActive(false);

m\_off.SetActive(true);

}

break;

}

}

}

CubeFloat.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class CubeFloat : MonoBehaviour {

public float speed, tilt;

public Vector3 target;

void Update ()

{

transform.position =

Vector3.MoveTowards(transform.position,

target, Time.deltaTime \* speed);

if(transform.position == target && target.y != 0.1f)

target.y = 0.1f;

else if (transform.position == target && target.y == 0.1f)

target.y = 1.8f;

transform.Rotate(Vector3.up \* speed);

}

}

DontDestroy.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class DontDestroy : MonoBehaviour

{

void Start ()

{

if (GameObject.FindGameObjectsWithTag("Audio").Length == 1)

DontDestroyOnLoad(gameObject);

else

Destroy(gameObject);

}

void Update () {

}

}

GameCntrl.cs

using UnityEngine;

using UnityEngine.UI;

using System.Collections;

public class GameCntrl : MonoBehaviour

{

public GameObject pLost;

public GameObject colBlock;

public Vector3[] positions;

private GameObject block;

private GameObject[] blocks = new GameObject[4];

public int rand, count;

private float rCol, gCol, bCol;

public Text score;

private static Color aColor;

[HideInInspector]

public bool next, lose;

void Start()

{

count = 0;

next = false;

lose = false;

rand = Random.Range(0, positions.Length);

for (int i = 0; i < positions.Length; i++)

{

blocks[i] = Instantiate(colBlock, positions[i], Quaternion.identity) as GameObject;

if (rand == i)

block = blocks[i];

}

block.GetComponent<RandCol>().right = true;

}

void Update()

{

if (lose)

playerLose();

if (next && !lose)

nextColors();

}

void nextColors()

{

if (PlayerPrefs.GetString("Music") != "no")

{

GetComponent<AudioSource>().Play();

}

count++;

score.text = count.ToString();

aColor = new Vector4(Random.Range(0.1f, 1f), Random.Range(0.1f, 1f), Random.Range(0.1f, 1f), 1);

GetComponent<Renderer>().material.color = aColor;

next = false;

if (count < 3)

{

rCol = 0.2f;

gCol = 0.2f;

bCol = 0.2f;

}

else if (count >= 3 && count < 5)

{

rCol = 0.1f;

gCol = 0.1f;

bCol = 0f;

}

else if (count >= 5)

{

rCol = 0f;

gCol = 0f;

bCol = 0.5f;

}

// New colors for blocks

rand = Random.Range(0, positions.Length);

for (int i = 0; i < positions.Length; i++)

{

if (i == rand)

blocks[i].GetComponent<Renderer>().material.color = aColor;

else

{

float r = aColor.r + Random.Range(0.1f, rCol) > 1f ? 1f : aColor.r + Random.Range(0.1f, rCol);

float g = aColor.g + Random.Range(0.1f, gCol) > 1f ? 1f : aColor.g + Random.Range(0.1f, gCol);

float b = aColor.b + Random.Range(0.1f, bCol) > 1f ? 1f : aColor.b + Random.Range(0.1f, bCol);

blocks[i].GetComponent<Renderer>().material.color = new Vector4(r, g, b, aColor.a);

}

}

}

void playerLose()

{

print("Player lose");

if(PlayerPrefs.GetInt("maxScore") < count)

PlayerPrefs.SetInt("maxScore",count);

pLost.SetActive(true);

if (PlayerPrefs.GetString("Music") != "yes") {

pLost.GetComponent<AudioSource>().mute = true; } } }

MaxScore.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine.UI;

using UnityEngine;

public class MaxScore : MonoBehaviour

{

void Start ()

{

GetComponent<Text>().text = PlayerPrefs.GetInt("maxScore").ToString();

}

}

RandCol.cs

using UnityEngine;

using System.Collections;

public class RandCol : MonoBehaviour

{

public bool main = false, right = false;

private static Color aColor;

void Awake()

{

if (main)

aColor = new Vector4(Random.Range(0.1f, 1f), Random.Range(0.1f, 1f), Random.Range(0.1f, 1f), 1);

}

void Start()

{

if (main || right)

GetComponent<Renderer>().material.color = aColor;

else

GetComponent<Renderer>().material.color = new Vector4(aColor.r + Random.Range(0.1f, 0.3f),

aColor.g + Random.Range(0.1f, 0.3f), aColor.b + Random.Range(0.1f, 0.3f), aColor.a);

}

}

RightOne.cs

using UnityEngine;

using System.Collections;

public class RightOne : MonoBehaviour

{

private GameObject mainCube;

void Start()

{

mainCube = GameObject.Find("Main Cube");

}

void OnMouseDown()

{

if (GetComponent<Renderer>().material.color == mainCube.GetComponent<Renderer>().material.color)

mainCube.GetComponent<GameCntrl>().next = true;

else

mainCube.GetComponent<GameCntrl>().lose = true;

}

}

Timer.cs

using UnityEngine;

using System.Collections;

public class Timer : MonoBehaviour

{

public Color col, defCol;

public GameObject mCube;

private Color lastCol;

void Start()

{

lastCol = mCube.GetComponent<Renderer>().material.color;

}

void Update()

{

if (!mCube.GetComponent<GameCntrl>().lose)

{

if (transform.position.x < -5.58f)

Destroy(gameObject);

if (transform.position.x < -0.53f)

GetComponent<Renderer>().material.color = Color.Lerp(GetComponent<Renderer>().material.color, col, Time.deltaTime);

transform.position -= new Vector3(0.03f, 0, 0);

}

if (mCube.GetComponent<Renderer>().material.color != lastCol)

{

lastCol = mCube.GetComponent<Renderer>().material.color;

transform.position = new Vector3(0, transform.position.y, 0);

GetComponent<Renderer>().material.color = defCol;

}

}

void OnDestroy()

{

if (mCube)

mCube.GetComponent<GameCntrl>().lose = true;

}

}