Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет

телекоммуникаций и информатики»

Кафедра ПМиК

Расчетно-графическое задание по дисциплине

Программирование мобильных устройств

На тему

Игровое приложение. 2D - платформер

Выполнил: студент 3 курса

Ф. ИВТ, группа: ИП-711

Мартасов И. О.

Проверил: доцент кафедры ПМиК

Нечта Иван Васильевич

Новосибирск, 2020

**Содержание**

1. **Описание Игры……………………………………………………………3**
2. **Результаты работы……………………………………………………….5**
3. **Приложение. Скрипты…………………………………………………...9**

**Описание игры**

**Сюжет:**

Главный герой игры – маленький лисенок. Он ходит по локациям и собирает предметы – вишенки и бриллианты. Также он умеет прыгать и ползать.

Цель игры – собрать все предметы (вишенки и бриллианты) и прийти к концу уровня невредимым.

**Описание игрового мира:**

В игре существует две локации:

Первая – Воздушный Остров. Локация изображена в светлых тонах, в ней присутствуют различные каменные постройки и несколько мини-островов в воздухе.

Вторая – Подземелье. В нем находятся движущиеся платформы, по которым игрок должен передвигаться. Сама локация изображена в темно-фиолетовом тоне.

Обе локации, помимо предметов, населяют враги – опоссумы и орлы. Опоссумы ходят по земле из стороны в сторону, орлы обитают на небе и не водятся в подземельях. Столкновение с врагами приводит к смерти главного героя и рестарту игры.

**Геймплей:**

Движение персонажа:

Вперед – стрелка вправо

Назад – стрелка влево

Кнопка прыжка – стрелка вверх

Кнопка ползка – стрелка вниз

У главного героя есть 3 жизни – в случае смерти – падении с острова или столкновение с врагом – жизнь теряется. По истечении всех жизней игра заканчивается плохой концовкой.

Если главный герой дошел до конца игры, но не собрал все предметы, то игра заканчивается т.н. нормальной концовкой.

Если же все 20 вишенок и 10 бриллиантов собраны и герой дошел до конца, то открывается хорошая концовка – игра пройдена.

**Техническое описание**

Платформер был сделан в среде Unity.

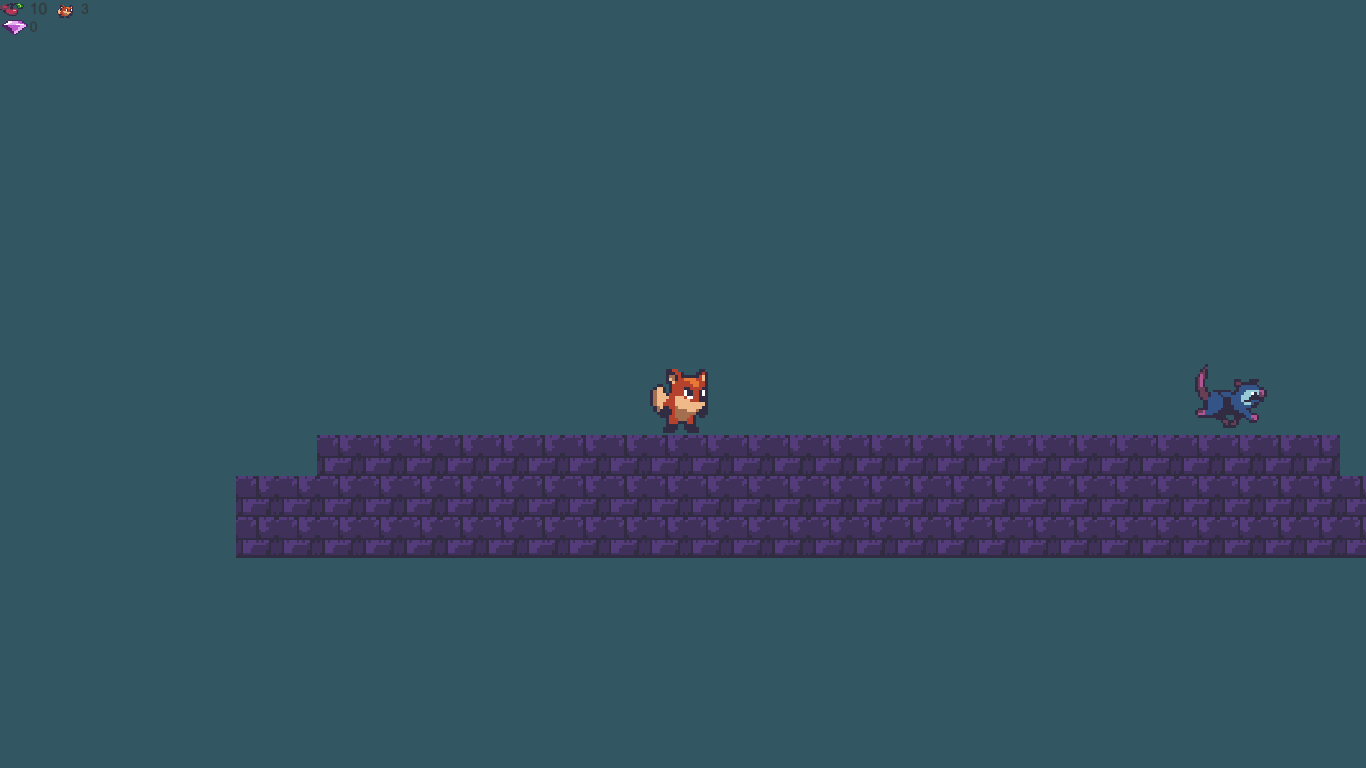
**Результаты работы**

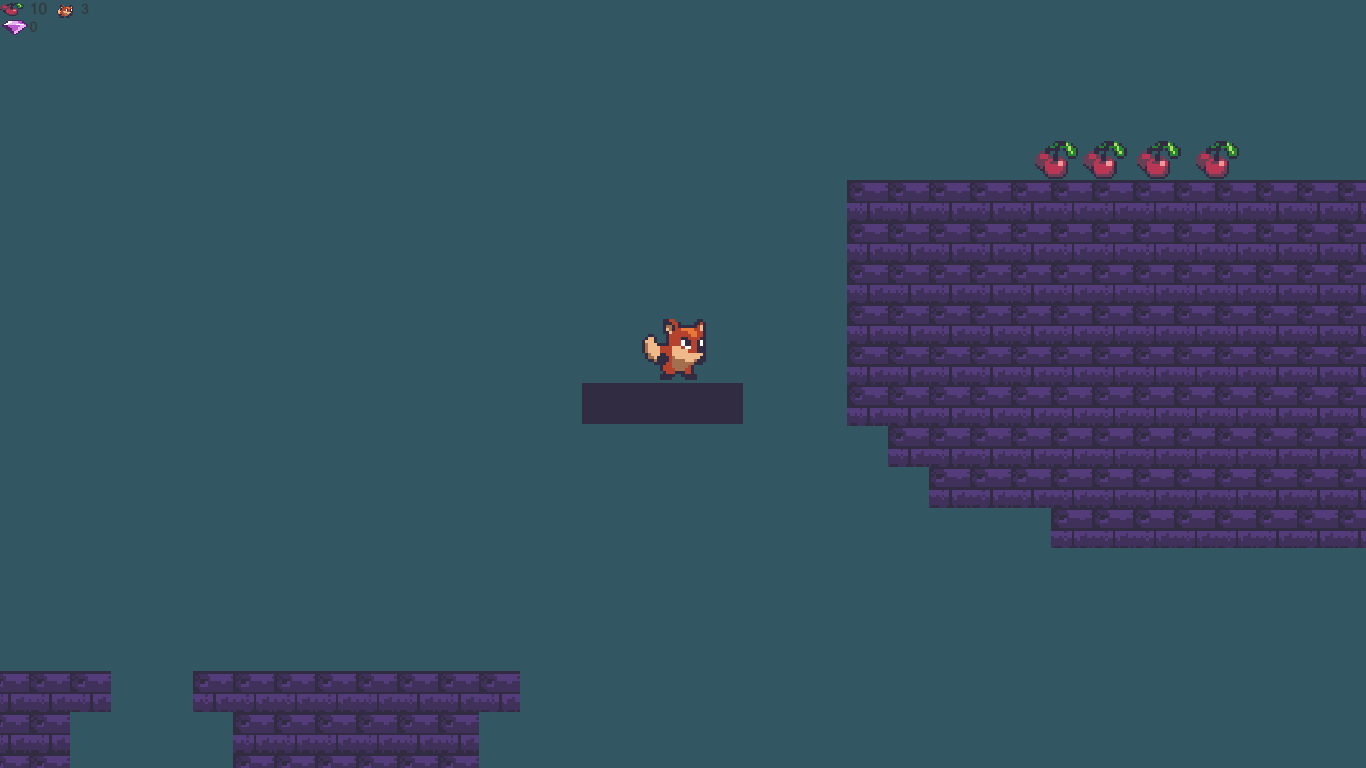
1. **Локация “Воздушный остров” **

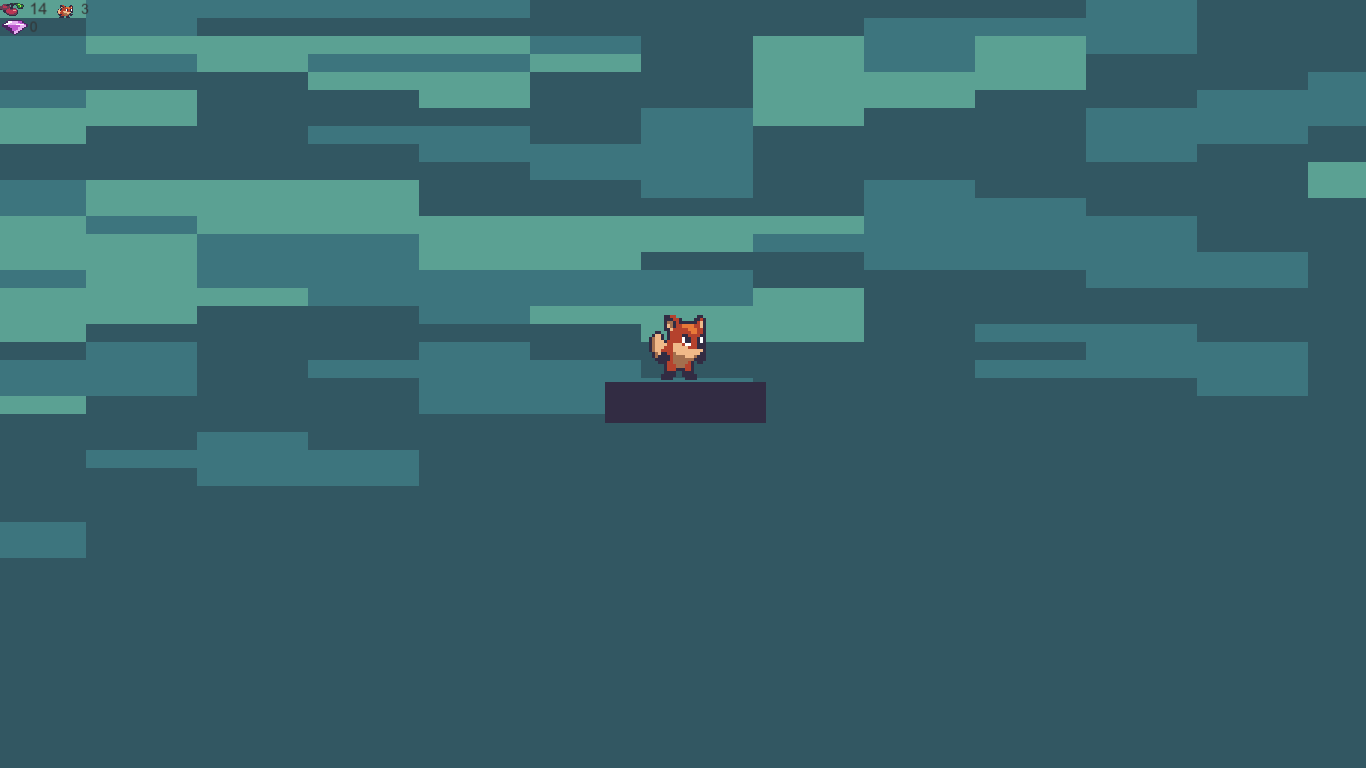
****

****

1. **Локация “Подземелье”**

****

****

****

**Приложение. Скрипты.**

1. **Playermove – взаимодействие героя с миром**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Collections.Specialized;

using System.Security.Cryptography;

using System.Threading;

using UnityEngine.SceneManagement;

using UnityEngine;

public class Playermove : MonoBehaviour

{

public CharacterController2D controller;

public Animator animator;

public float runSpeed = 20f;

float horizontalMove = 0f;

float SX, SY;

bool jump = false;

bool crouch = false;

void Start()

{

SX = transform.position.x;

SY = transform.position.y;

}

void Update()

{

horizontalMove = Input.GetAxisRaw("Horizontal") \* runSpeed;

animator.SetFloat("Run", Math.Abs(horizontalMove));

if(Input.GetButtonDown("Jump"))

{

jump = true;

animator.SetBool("isJumping", true);

}

if (Input.GetButtonDown("Crouch"))

{

crouch = true;

}

else if (Input.GetButtonUp("Crouch"))

{

crouch = false;

}

}

private void OnCollisionEnter2D(Collision2D collision)

{

if (collision.gameObject.name.Equals("Platform"))

{

this.transform.parent = collision.transform;

}

if (collision.gameObject.name == "DeadSpace" || collision.gameObject.name == "Opossum" || collision.gameObject.name == "Eagle")

{

Collect.TheCherry = 0;

CollectGem.TheGem = 0;

CountLives.Live -= 1;

if(CountLives.Live == 0)

{

SceneManager.LoadScene("BadEnding1");

}

else

{

SceneManager.LoadScene("SampleScene");

}

transform.position = new Vector3(SX, SY, transform.position.z);

}

}

private void OnCollisionExit2D(Collision2D collision)

{

if (collision.gameObject.name.Equals("Platform"))

{

this.transform.parent = null;

}

}

public void OnCrouch(bool isCrouch)

{

animator.SetBool("isCrouch", isCrouch);

}

public void OnLanding()

{

animator.SetBool("isJumping", false);

}

private void FixedUpdate()

{

controller.Move(horizontalMove \* Time.deltaTime, crouch, jump);

jump = false;

}

private void OnTriggerEnter2D(Collider2D collision)

{

if (collision.tag == "Cherry")

{

Collect.TheCherry += 1;

Destroy(collision.gameObject);

}

if (collision.tag == "Gem")

{

CollectGem.TheGem += 1;

Destroy(collision.gameObject);

}

}

}

1. **Scenemanager – переход между уровнями**

using System.Collections;

using System.Collections.Generic;

using UnityEngine.SceneManagement;

using UnityEngine;

public class Scenemanager : MonoBehaviour

{

private void OnCollisionEnter2D(Collision2D collision)

{

if (collision.gameObject.name == "EndLevel1")

{

SceneManager.LoadScene("Scene02");

}

if (collision.gameObject.name == "EndGame")

{

if(Collect.TheCherry == 20 && CollectGem.TheGem == 10)

{

SceneManager.LoadScene("GoodEnding");

}

else

{

SceneManager.LoadScene("BadEnding2");

}

}

}

}

1. **Platformmove – движение платформы**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Platformmove : MonoBehaviour

{

public Transform pos1, pos2;

public float speed = 1f;

public Transform startPos;

Vector3 nextPos;

void Start()

{

nextPos = startPos.position;

}

// Update is called once per frame

void Update()

{

transform.position = Vector3.MoveTowards(transform.position, nextPos, speed \* Time.deltaTime);

if (transform.position == pos1.position)

{

nextPos = pos2.position;

}

if (transform.position == pos2.position)

{

nextPos = pos1.position;

}

}

private void OnDrawGizmos()

{

Gizmos.DrawLine(pos1.position, pos2.position);

}

1. **Enemymove – движение врагов**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Enemymove : MonoBehaviour

{

public Transform pos1, pos2;

public float speed = 1f;

public Transform startPos;

Vector3 nextPos;

void Start()

{

nextPos = startPos.position;

}

// Update is called once per frame

void Update()

{

transform.position = Vector3.MoveTowards(transform.position, nextPos,speed \* Time.deltaTime);

if (transform.position == pos1.position)

{

nextPos = pos2.position;

transform.localScale = new Vector2(-1, 1);

}

if (transform.position == pos2.position)

{

nextPos = pos1.position;

transform.localScale = new Vector2(1, 1);

}

}

private void OnDrawGizmos()

{

Gizmos.DrawLine(pos1.position, pos2.position);

}

}

1. **Collect – счетчик вишенок**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class Collect : MonoBehaviour

{

public static int TheCherry;

public Text textCherry;

// Start is called before the first frame update

void Start()

{

textCherry = GetComponent<Text>();

}

// Update is called once per frame

void Update()

{

textCherry.text = "" + TheCherry;

}

}

1. **СollectGem – счетчик бриллиантов**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class CollectGem : MonoBehaviour

{

public static int TheGem;

public Text textGem;

// Start is called before the first frame update

void Start()

{

textGem = GetComponent<Text>();

}

// Update is called once per frame

void Update()

{

textGem.text = "" + TheGem;

}

}

1. **CountLive – счетчик жизней**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class CountLives : MonoBehaviour

{

public static int Live = 3;

public Text textLive;

// Start is called before the first frame update

void Start()

{

textLive = GetComponent<Text>();

}

// Update is called once per frame

void Update()

{

textLive.text = "" + Live;

}

}