

**Приложение А**  
Листинг программы

```

Файл DialogAnimator.cs:
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class DialogAnimator : MonoBehaviour
{
    public Animator startAnim;
    public DialogManager dm;
    public void OnTriggerEnter2D(Collider2D other)
    {
        startAnim.SetBool("startOpen", true);
    }
    public void OnTriggerExit2D(Collider2D other)
    {
        startAnim.SetBool("startOpen", false);
        dm.EndDialogue();
    }
}

```

```

Файл Bullet.cs:
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Bullet : MonoBehaviour
{
    public float speed;
    public float lifetime;
    public float lifetime1;
    public float distance;
    public int damage;
    public LayerMask whatIsSolid;

    private Vector2 direction; // Направление движения пули

    public GameObject bulletEffect;

    private void Start()
    {
        // Уничтожаем пулю через заданное время
        Destroy(gameObject, lifetime1);
    }

    public void SetDirection(Vector2 dir)
    {
        direction = dir; // Устанавливаем направление
    }

    private void Update()
    {
        RaycastHit2D hitInfo =
        Physics2D.Raycast(transform.position, direction,
        distance, whatIsSolid);
        if (hitInfo.collider != null)
        {
            if (hitInfo.collider.CompareTag("Enemy"))

```

```

{
    hitInfo.collider.GetComponent<Enemy>().TakeDamage(damage);
    }
    Instantiate(bulletEffect, transform.position, Quaternion.identity);
    Destroy(gameObject);
    }

    // Двигаем пулю в заданном направлении
    transform.Translate(direction * speed * Time.deltaTime);
    }
}

```

```

Файл DialogManager.cs:
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class DialogManager : MonoBehaviour
{
    public Text dialogueText;
    public Text nameText;

    public Animator boxAnim;
    public Animator startAnim;

    public int x;
    public Button test;

    private Queue<string> sentences;

    private void Start()
    {
        sentences = new Queue<string>();
    }

    public void StartDialogue(Dialogue dialogue)
    {
        boxAnim.SetBool("boxOpen", true);
        startAnim.SetBool("startOpen", false);

        nameText.text = dialogue.name;
        sentences.Clear();

        foreach(string sentence in dialogue.sentences)
        {
            sentences.Enqueue(sentence);
        }
        DisplayNextSentence();
    }

    public void DisplayNextSentence()
    {
        if (sentences.Count == 0)
        {

```

```

        EndDialogue();
        if (x == 1)
        { test.gameObject.SetActive(true); }
        return;
    }
    string sentence = sentences.Dequeue();
    StopAllCoroutines();
    StartCoroutine(TypeSentence(sentence));
}

IEnumerator TypeSentence(string sentence)
{
    dialogueText.text = "";
    foreach(char letter in sentence.ToCharArray())
    {
        dialogueText.text += letter;
        yield return null;
    }
}

public void EndDialogue()
{
    boxAnim.SetBool("boxOpen", false);
}
}

```

Файл DialogTrigger.cs:

```

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

public class DialogTrigger : MonoBehaviour
{
    public Dialogue dialogue;
    public void TriggerDialogue()
    {

        FindObjectOfType<DialogManager>().StartDialogue(dialogue);
    }
}

```

Файл Dialogue.cs:

```

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

[System.Serializable]
public class Dialogue
{
    public string name;
    [TextArea(3, 10)]
    public string[] sentences;
}

```

Файл taxt.cs:

```

using System.Collections;

```

```

using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
public class taxt : MonoBehaviour
{
    public Text dialogueText;
    private Queue<string> sentences;
    public Dialogue dialogue;
    public Button dalee ;
    public Button Wputi;

    private void Start()
    {
        sentences = new Queue<string>();
        StartDialogue(dialogue);
    }

    public void StartDialogue(Dialogue dialogue)
    {
        foreach (string sentence in dialogue.sentences)
        {
            sentences.Enqueue(sentence);
        }
        DisplayNextSentence();
    }

    public void DisplayNextSentence()
    {
        if (sentences.Count == 0)
        {
            dalee.gameObject.SetActive(false);
            Wputi.gameObject.SetActive(true);
            return;
        }
        string sentence = sentences.Dequeue();
        StopAllCoroutines();
        StartCoroutine(TypeSentence(sentence));
    }

    IEnumerator TypeSentence(string sentence)
    {
        dialogueText.text = "";
        foreach (char letter in sentence.ToCharArray())
        {
            dialogueText.text += letter;
            yield return null;
        }
    }
}

```

Файл hil.cs:

```

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

public class hil : MonoBehaviour

```

```

{
    public Image a;
    public Image b;
    public Text text;
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.F))
        {
            hiler();
        }

        string[] stroka1 = File.ReadAllLines("зелья.txt");
        text.text = stroka1[0];
    }
    private void hiler()
    {
        string[] stroka1 = File.ReadAllLines("зелья.txt");
        if
        ((int.Parse(stroka1[0])>0)&&(FindObjectOfType<hp>().hp<2))
        {
            FindObjectOfType<hp>().hp += 1;
            stroka1[0] = (int.Parse(stroka1[0])-1).ToString();
            StreamWriter f1 = new
            StreamWriter("зелья.txt", false);
            for (int k = 0; k < stroka1.Length; k++)
            {
                f1.WriteLine(stroka1[k]);
            }
            f1.Close();

            if (FindObjectOfType<hp>().hp == 1)
            { a.gameObject.SetActive(true); }
            if (FindObjectOfType<hp>().hp==2)
            { b.gameObject.SetActive(true); }
        }
    }
}

```

Файл porajenie.cs:

```

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

public class porajenie : MonoBehaviour
{
    public Image a;
    public Image b;

    public Image imgObg1;
    public Sprite spriteImage1;
    public Sprite spriteImage2;

    public Button karta1;

    public GameObject prajenie1;

```

```

    public GameObject prajenie2;

    public int kol_vo = 0;

    bool par=false;

    private void Update()
    {
        string[] stroka = File.ReadAllLines("номер
        принятого задания.txt");

        string[] stroka1 =
        File.ReadAllLines("предметы.txt");

        if ((FindObjectOfType<hp>().hp <= 0
        )&&(par==false) )
        {
            stroka1[4]=(int.Parse(stroka1[4]) -
            kol_vo).ToString();

            StreamWriter f1 = new
            StreamWriter("предметы.txt", false);
            for (int k = 0; k < stroka1.Length; k++)
            {
                f1.WriteLine(stroka1[k]);
            }
            f1.Close();

            if (int.Parse(stroka[0]) == 6)
            {
                prajenie1.gameObject.SetActive(true);

                string[] stroka5 =
                File.ReadAllLines("карта.txt");
                if (stroka5[0] == "0")
                {
                    imgObg1.sprite = spriteImage1;
                }
                else
                {
                    imgObg1.sprite = spriteImage2;
                    karta1.gameObject.SetActive(true);
                }

                StartCoroutine(stop());
            }

            else {
                prajenie2.gameObject.SetActive(true);
                StartCoroutine(stop());
            }

            par = true;
        }
    }

    private IEnumerator stop()

```

```

{
    yield return new WaitForSeconds(0f);
    Time.timeScale = 0f;
}

```

```

public void заново()
{
    par = false;
    prajenie1.SetActive(false);
    prajenie2.SetActive(false);
    karta1.gameObject.SetActive(false);
    FindObjectOfType<hp>().hp = 2;
    { a.gameObject.SetActive(true); }
    { b.gameObject.SetActive(true); }
    Time.timeScale = 1f;
}
}

```

Файл zad\_po\_sbor.cs:

```

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

public class zad_po_sbor : MonoBehaviour
{
    public GameObject z1;
    public Text[] text;

    public Image imgObg;
    public Sprite spriteImage1;
    public Sprite spriteImage2;

    public Button karta;

    public GameObject pobeda;

    void Start()
    {
        string[] stroka = File.ReadAllLines("номер
        принятого задания.txt");

        if (int.Parse(stroka[0]) == 6)
        {
            z1.gameObject.SetActive(true);
        }
    }

    void Update()
    {
        string[] stroka = File.ReadAllLines("номер
        принятого задания.txt");

        string[] stroka1 =
        File.ReadAllLines("предметы.txt");
    }
}

```

```

if (int.Parse(stroka[0]) == 6)
{
    text[0].text = stroka1[4] + "/10";
}

```

```

if ((int.Parse(stroka[0]) == 6) &&
(int.Parse(stroka1[4]) >= 10))
{

```

```

    stroka1[4] = (int.Parse(stroka1[4]) -
10).ToString();

```

```

    StreamWriter f1 = new
    StreamWriter("предметы.txt", false);
    for (int k = 0; k < stroka1.Length; k++)
    {
        f1.WriteLine(stroka1[k]);
    }
    f1.Close();

```

```

    stroka[0] = "0";
    StreamWriter f = new StreamWriter("номер
    принятого задания.txt", false);
    for (int k = 0; k < stroka.Length; k++)
    {
        f.WriteLine(stroka[k]);
    }
    f.Close();

```

```

/////////
string[] stroka2 =
File.ReadAllLines("манеты.txt");
stroka2[0] = (int.Parse(stroka2[0]) +
300).ToString();

```

```

    StreamWriter f2 = new
    StreamWriter("манеты.txt", false);
    for (int k = 0; k < stroka2.Length; k++)
    {
        f2.WriteLine(stroka2[k]);
    }
    f2.Close();

```

```

    string[] stroka3 =
    File.ReadAllLines("выполнение зд.txt");
    stroka3[5] = (int.Parse(stroka3[5]) + 1).ToString();

```

```

    StreamWriter f3 = new
    StreamWriter("выполнение зд.txt", false);
    for (int k = 0; k < stroka3.Length; k++)
    {
        f3.WriteLine(stroka3[k]);
    }
    f3.Close();

```

```

    string[] stroka4 = File.ReadAllLines("панг.txt");

```

```

        stroka4[0] = (int.Parse(stroka4[0]) + 3).ToString();

        StreamWriter f4 = new StreamWriter("панг.txt",
false);
        for (int k = 0; k < stroka4.Length; k++)
        {
            f4.WriteLine(stroka4[k]);
        }
        f4.Close();

        pobeda.gameObject.SetActive(true);

        string[] stroka5 = File.ReadAllLines("капра.txt");
        if (stroka5[0] == "0")
        {
            imgObg.sprite = spriteImage1;
        }
        else
        {
            imgObg.sprite = spriteImage2;
            karta.gameObject.SetActive(true);
        }

        StartCoroutine(stop());
    }

}

private IEnumerator stop()
{
    yield return new WaitForSeconds(1f);
    Time.timeScale = 0f;
}
}

```

Файл dostun\_rang.cs:

```

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

public class dostun_rang : MonoBehaviour
{
    public Button[] btn;
    public Image imgObg;
    public Sprite[] spriteImage1;

    void Update()
    {
        string[] stroka = File.ReadAllLines("панг.txt");

        if (int.Parse(stroka[0]) > 0)
        {

```

```

            btn[7].gameObject.SetActive(false);
        }

        if
        ((int.Parse(stroka[0]) >= 1) && ((int.Parse(stroka[0]) <
5)))
        {
            imgObg.sprite = spriteImage1[0];
            btn[0].gameObject.SetActive(true);
        }
        if ((int.Parse(stroka[0]) >= 5) &&
((int.Parse(stroka[0]) < 12)))
        {
            imgObg.sprite = spriteImage1[1];
            btn[0].gameObject.SetActive(true);
            btn[1].gameObject.SetActive(true);
        }
        if ((int.Parse(stroka[0]) >= 12) &&
((int.Parse(stroka[0]) < 22)))
        {
            imgObg.sprite = spriteImage1[2];
            btn[0].gameObject.SetActive(true);
            btn[1].gameObject.SetActive(true);
            btn[2].gameObject.SetActive(true);
        }
        if ((int.Parse(stroka[0]) >= 22) &&
((int.Parse(stroka[0]) < 35)))
        {
            imgObg.sprite = spriteImage1[3];
            btn[0].gameObject.SetActive(true);
            btn[1].gameObject.SetActive(true);
            btn[2].gameObject.SetActive(true);
            btn[3].gameObject.SetActive(true);
        }
        if ((int.Parse(stroka[0]) >= 35) &&
((int.Parse(stroka[0]) < 51)))
        {
            imgObg.sprite = spriteImage1[4];
            btn[0].gameObject.SetActive(true);
            btn[1].gameObject.SetActive(true);
            btn[2].gameObject.SetActive(true);
            btn[3].gameObject.SetActive(true);
            btn[4].gameObject.SetActive(true);
        }
        if ((int.Parse(stroka[0]) >= 51) &&
((int.Parse(stroka[0]) < 70)))
        {
            imgObg.sprite = spriteImage1[5];
            btn[0].gameObject.SetActive(true);
            btn[1].gameObject.SetActive(true);
            btn[2].gameObject.SetActive(true);
            btn[3].gameObject.SetActive(true);
            btn[4].gameObject.SetActive(true);
            btn[5].gameObject.SetActive(true);
        }
        if (int.Parse(stroka[0]) > 70)
        {
            imgObg.sprite = spriteImage1[5];
            btn[0].gameObject.SetActive(true);

```

```

        btn[1].gameObject.SetActive(true);
        btn[2].gameObject.SetActive(true);
        btn[3].gameObject.SetActive(true);
        btn[4].gameObject.SetActive(true);
        btn[5].gameObject.SetActive(true);
        btn[6].gameObject.SetActive(true);
    }
}
}

```

Файл wbr\_ranga.cs:

```

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

public class wbr_ranga : MonoBehaviour
{
    public int r;
    public Image[] imgObg;
    public Button[] btn;

    public void RangF_clik()
    {
        btn[0].gameObject.SetActive(false);
        btn[1].gameObject.SetActive(false);
        btn[2].gameObject.SetActive(false);

        r = 1;
        imgObg[0].gameObject.SetActive(true);
        imgObg[1].gameObject.SetActive(false);
        imgObg[2].gameObject.SetActive(false);
        imgObg[3].gameObject.SetActive(false);
        imgObg[4].gameObject.SetActive(false);
        imgObg[5].gameObject.SetActive(false);
        imgObg[6].gameObject.SetActive(false);

        string[] stroka = File.ReadAllLines("выполнение
зд.txt");
        if (int.Parse(stroka[0]) == 0)
        {
            btn[0].gameObject.SetActive(true);
        }
        if (int.Parse(stroka[1]) == 0)
        {
            btn[1].gameObject.SetActive(true);
        }
        if (int.Parse(stroka[2]) == 0)
        {
            btn[2].gameObject.SetActive(true);
        }
    }

    public void RangE_clik()
    {
        btn[0].gameObject.SetActive(false);
        btn[1].gameObject.SetActive(false);

```

```

        btn[2].gameObject.SetActive(false);

        r = 2;
        imgObg[0].gameObject.SetActive(false);
        imgObg[1].gameObject.SetActive(true);
        imgObg[2].gameObject.SetActive(false);
        imgObg[3].gameObject.SetActive(false);
        imgObg[4].gameObject.SetActive(false);
        imgObg[5].gameObject.SetActive(false);
        imgObg[6].gameObject.SetActive(false);

```

```

        string[] stroka = File.ReadAllLines("выполнение
зд.txt");
        if (int.Parse(stroka[3]) == 0)
        {
            btn[0].gameObject.SetActive(true);
        }
        if (int.Parse(stroka[4]) == 0)
        {
            btn[1].gameObject.SetActive(true);
        }
        if (int.Parse(stroka[5]) == 0)
        {
            btn[2].gameObject.SetActive(true);
        }
    }

    public void RangD_clik()
    {
        btn[0].gameObject.SetActive(false);
        btn[1].gameObject.SetActive(false);
        btn[2].gameObject.SetActive(false);

        r = 3;
        imgObg[0].gameObject.SetActive(false);
        imgObg[1].gameObject.SetActive(false);
        imgObg[2].gameObject.SetActive(true);
        imgObg[3].gameObject.SetActive(false);
        imgObg[4].gameObject.SetActive(false);
        imgObg[5].gameObject.SetActive(false);
        imgObg[6].gameObject.SetActive(false);

        string[] stroka = File.ReadAllLines("выполнение
зд.txt");
        if (int.Parse(stroka[6]) == 0)
        {
            btn[0].gameObject.SetActive(true);
        }
        if (int.Parse(stroka[7]) == 0)
        {
            btn[1].gameObject.SetActive(true);
        }
        if (int.Parse(stroka[8]) == 0)
        {
            btn[2].gameObject.SetActive(true);
        }
    }

    public void RangC_clik()
    {
        btn[0].gameObject.SetActive(false);

```

```

    btn[1].gameObject.SetActive(false);
    btn[2].gameObject.SetActive(false);

    r = 4;
    imgObg[0].gameObject.SetActive(false);
    imgObg[1].gameObject.SetActive(false);
    imgObg[2].gameObject.SetActive(false);
    imgObg[3].gameObject.SetActive(true);
    imgObg[4].gameObject.SetActive(false);
    imgObg[5].gameObject.SetActive(false);
    imgObg[6].gameObject.SetActive(false);

    string[] stroka = File.ReadAllLines("выполнение
зд.txt");
    if (int.Parse(stroka[9]) == 0)
    {
        btn[0].gameObject.SetActive(true);
    }
    if (int.Parse(stroka[10]) == 0)
    {
        btn[1].gameObject.SetActive(true);
    }
    if (int.Parse(stroka[11]) == 0)
    {
        btn[2].gameObject.SetActive(true);
    }
}
public void RangB_клик()
{
    btn[0].gameObject.SetActive(false);
    btn[1].gameObject.SetActive(false);
    btn[2].gameObject.SetActive(false);

    r = 5;
    imgObg[0].gameObject.SetActive(false);
    imgObg[1].gameObject.SetActive(false);
    imgObg[2].gameObject.SetActive(false);
    imgObg[3].gameObject.SetActive(false);
    imgObg[4].gameObject.SetActive(true);
    imgObg[5].gameObject.SetActive(false);
    imgObg[6].gameObject.SetActive(false);

    string[] stroka = File.ReadAllLines("выполнение
зд.txt");
    if (int.Parse(stroka[12]) == 0)
    {
        btn[0].gameObject.SetActive(true);
    }
    if (int.Parse(stroka[13]) == 0)
    {
        btn[1].gameObject.SetActive(true);
    }
    if (int.Parse(stroka[14]) == 0)
    {
        btn[2].gameObject.SetActive(true);
    }
}
public void RangA_клик()
{

```

```

    btn[0].gameObject.SetActive(false);
    btn[1].gameObject.SetActive(false);
    btn[2].gameObject.SetActive(false);

    r = 6;
    imgObg[0].gameObject.SetActive(false);
    imgObg[1].gameObject.SetActive(false);
    imgObg[2].gameObject.SetActive(false);
    imgObg[3].gameObject.SetActive(false);
    imgObg[4].gameObject.SetActive(false);
    imgObg[5].gameObject.SetActive(true);
    imgObg[6].gameObject.SetActive(false);

    string[] stroka = File.ReadAllLines("выполнение
зд.txt");
    if (int.Parse(stroka[15]) == 0)
    {
        btn[0].gameObject.SetActive(true);
    }
    if (int.Parse(stroka[16]) == 0)
    {
        btn[1].gameObject.SetActive(true);
    }
    if (int.Parse(stroka[17]) == 0)
    {
        btn[2].gameObject.SetActive(true);
    }
}
public void RangS_клик()
{
    btn[0].gameObject.SetActive(false);
    btn[1].gameObject.SetActive(false);
    btn[2].gameObject.SetActive(false);

    r = 7;
    imgObg[0].gameObject.SetActive(false);
    imgObg[1].gameObject.SetActive(false);
    imgObg[2].gameObject.SetActive(false);
    imgObg[3].gameObject.SetActive(false);
    imgObg[4].gameObject.SetActive(false);
    imgObg[5].gameObject.SetActive(false);
    imgObg[6].gameObject.SetActive(true);

    string[] stroka = File.ReadAllLines("выполнение
зд.txt");
    if (int.Parse(stroka[18]) == 0)
    {
        btn[0].gameObject.SetActive(true);
    }
    if (int.Parse(stroka[19]) == 0)
    {
        btn[1].gameObject.SetActive(true);
    }
    if (int.Parse(stroka[20]) == 0)
    {
        btn[2].gameObject.SetActive(true);
    }
}

```



```

    }
}
Файл Zadanija.cs:
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.IO;

public class Zadanija : MonoBehaviour
{
    public Image imgObg;
    public Sprite[] spriteImage1;

    public Переход perehod;

    public void Z1_clic()
    {
        if (FindObjectOfType<wbr_ranga>().r == 1)
        {
            perehod.levelToLoad = 13;

            string[] stroka = File.ReadAllLines("номер
            принятого задания.txt");
            stroka[0]=1.ToString();
            StreamWriter f1 = new StreamWriter("номер
            принятого задания.txt", false);
            for (int k = 0; k < stroka.Length; k++)
            {
                f1.WriteLine(stroka[k]);
            }
            f1.Close();

            string[] stroka1 =
            File.ReadAllLines("посещённые локации.txt");
            stroka1[0] = 1.ToString();
            StreamWriter f11 = new
            StreamWriter("посещённые локации.txt", false);
            for (int k = 0; k < stroka1.Length; k++)
            {
                f11.WriteLine(stroka1[k]);
            }
            f11.Close();
        }
        if (FindObjectOfType<wbr_ranga>().r == 2)
        {
            perehod.levelToLoad = 14;
            string[] stroka = File.ReadAllLines("номер
            принятого задания.txt");
            stroka[0] = 4.ToString();
            StreamWriter f1 = new StreamWriter("номер
            принятого задания.txt", false);
            for (int k = 0; k < stroka.Length; k++)
            {
                f1.WriteLine(stroka[k]);
            }
            f1.Close();
        }
        if (FindObjectOfType<wbr_ranga>().r == 3)

```

```

    {
        perehod.levelToLoad = 5;
    }
    if (FindObjectOfType<wbr_ranga>().r == 4)
    {
        perehod.levelToLoad = 5;
    }
    if (FindObjectOfType<wbr_ranga>().r == 5)
    {
        perehod.levelToLoad = 5;
    }
    if (FindObjectOfType<wbr_ranga>().r == 6)
    {
        perehod.levelToLoad = 5;
    }
    if (FindObjectOfType<wbr_ranga>().r == 7)
    {
        perehod.levelToLoad = 5;
    }

    int s = 0;
    for (int i = 1; i < 8; i++)
    {
        if (FindObjectOfType<wbr_ranga>().r == i)
        {
            imgObg.sprite = spriteImage1[s];

        }
        s += 3;
    }
    imgObg.gameObject.SetActive(true);
}
public void Z2_clic()
{
    if (FindObjectOfType<wbr_ranga>().r == 1)
    {
        perehod.levelToLoad = 12;

    }
    if (FindObjectOfType<wbr_ranga>().r == 2)
    {
        perehod.levelToLoad = 14;

        string[] stroka = File.ReadAllLines("номер
        принятого задания.txt");
        stroka[0] = 5.ToString();
        StreamWriter f1 = new StreamWriter("номер
        принятого задания.txt", false);
        for (int k = 0; k < stroka.Length; k++)
        {
            f1.WriteLine(stroka[k]);
        }
        f1.Close();
    }
    if (FindObjectOfType<wbr_ranga>().r == 3)
    {
        perehod.levelToLoad = 5;
    }

```

```

    }
    if (FindObjectOfType<wbr_ranga>().r == 4)
    {
        perehod.levelToLoad = 5;
    }
    if (FindObjectOfType<wbr_ranga>().r == 5)
    {
        perehod.levelToLoad = 5;
    }
    if (FindObjectOfType<wbr_ranga>().r == 6)
    {
        perehod.levelToLoad = 5;
    }
    if (FindObjectOfType<wbr_ranga>().r == 7)
    {
        perehod.levelToLoad = 5;
    }

    int s = 1;
    for (int i = 1; i < 8; i++)
    {
        if (FindObjectOfType<wbr_ranga>().r == i)
        {
            imgObg.sprite = spriteImage1[s];

            }
            s += 3;
        }
        imgObg.gameObject.SetActive(true);
    }
    public void Z3_clic()
    {
        if (FindObjectOfType<wbr_ranga>().r == 1)
        {
            perehod.levelToLoad = 13;

            string[] stroka = File.ReadAllLines("номер
            принятого задания.txt");
            stroka[0] = 2.ToString();
            StreamWriter f1 = new StreamWriter("номер
            принятого задания.txt", false);
            for (int k = 0; k < stroka.Length; k++)
            {
                f1.WriteLine(stroka[k]);
            }
            f1.Close();

            string[] stroka1 =
            File.ReadAllLines("посещённые локации.txt");
            stroka1[0] = 1.ToString();
            StreamWriter f11 = new
            StreamWriter("посещённые локации.txt", false);
            for (int k = 0; k < stroka1.Length; k++)
            {
                f11.WriteLine(stroka1[k]);
            }
            f11.Close();
        }
    }

```

```

    if (FindObjectOfType<wbr_ranga>().r == 2)
    {
        perehod.levelToLoad = 15;

        string[] stroka = File.ReadAllLines("номер
        принятого задания.txt");
        stroka[0] = 6.ToString();
        StreamWriter f1 = new StreamWriter("номер
        принятого задания.txt", false);
        for (int k = 0; k < stroka.Length; k++)
        {
            f1.WriteLine(stroka[k]);
        }
        f1.Close();

        string[] stroka1 =
        File.ReadAllLines("посещённые локации.txt");
        stroka1[1] = 1.ToString();
        StreamWriter f11 = new
        StreamWriter("посещённые локации.txt", false);
        for (int k = 0; k < stroka1.Length; k++)
        {
            f11.WriteLine(stroka1[k]);
        }
        f11.Close();
    }
    if (FindObjectOfType<wbr_ranga>().r == 3)
    {
        perehod.levelToLoad = 5;
    }
    if (FindObjectOfType<wbr_ranga>().r == 4)
    {
        perehod.levelToLoad = 5;
    }
    if (FindObjectOfType<wbr_ranga>().r == 5)
    {
        perehod.levelToLoad = 5;
    }
    if (FindObjectOfType<wbr_ranga>().r == 6)
    {
        perehod.levelToLoad = 5;
    }
    if (FindObjectOfType<wbr_ranga>().r == 7)
    {
        perehod.levelToLoad = 5;
    }

    int s = 2;
    for (int i = 1; i < 8; i++)
    {
        if (FindObjectOfType<wbr_ranga>().r == i)
        {
            imgObg.sprite = spriteImage1[s];

            }
            s += 3;
        }
        imgObg.gameObject.SetActive(true);
    }

```

```

    public void kr()
    {
        imgObg.gameObject.SetActive(false);
    }
}

```

Файл книга.cs:

```

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

public class книга : MonoBehaviour
{
    public GameObject[] scrol;
    public Image img;
    public Sprite[] sprites;
    public Image fon;
    public Sprite[] sprites_fon;
    public Button exit;
    public Button exit1;

    void Start()
    {
        string[] stroka =
File.ReadAllLines("использовать.txt");

        if (int.Parse(stroka[0]) == 0)
        {
            fon.sprite = sprites_fon[0];
            exit1.gameObject.SetActive(true);
            img.gameObject.SetActive(false);
        }
        if (int.Parse(stroka[0]) == 1)
        {
            fon.sprite = sprites_fon[1];
            exit.gameObject.SetActive(true);
            scrol[0].gameObject.SetActive(true);
            img.sprite = sprites[3];
        }
        if (int.Parse(stroka[0]) == 2)
        {
            fon.sprite = sprites_fon[2];
            exit.gameObject.SetActive(true);
            scrol[1].gameObject.SetActive(true);
            img.sprite = sprites[0];
        }
    }

    public void button_click(int i)
    {
        img.sprite = sprites[i];
    }
}

```

Файл Draggable1.cs:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.IO;
using UnityEngine.EventSystems;

public class Draggable1 : MonoBehaviour,
IDragHandler, IBeginDragHandler
{
    private Canvas canvas;
    private RectTransform rectTransform;
    private Vector3 originalPosition;

    private Vector2 offset;
    void Awake()
    {
        rectTransform =
GetComponent<RectTransform>();
        canvas = GetComponentInParent<Canvas>();
        originalPosition =
rectTransform.anchoredPosition;
    }

    public void OnBeginDrag(PointerEventData
eventData)
    {
        // Сохранение оригинальной позиции при
        // начале перетаскивания

        // Сохраняем смещение между курсором и
        // объектом
        RectTransform rectTransform =
GetComponent<RectTransform>();
        originalPosition =
rectTransform.anchoredPosition;
        offset = eventData.position -
RectTransformUtility.WorldToScreenPoint(eventData.
pressEventCamera, rectTransform.position);
    }

    public void OnDrag(PointerEventData eventData)
    {
        Vector2 position;

        RectTransformUtility.ScreenPointToLocalPointInRecta
ngle(canvas.transform as RectTransform,
eventData.position, canvas.worldCamera, out position);

        Vector2 newPosition = eventData.position - offset;
        RectTransform rectTransform =
GetComponent<RectTransform>();
        rectTransform.position = newPosition;
    }
}

```

```

    public Vector3 GetCurrentPosition()
    {
        return rectTransform.anchoredPosition;
    }
}

```

Файл Draggable2.cs:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.IO;
using UnityEngine.EventSystems;

```

```

public class Draggable2 : MonoBehaviour,
IDragHandler, IBeginDragHandler

```

```

{
    private Canvas canvas;
    private RectTransform rectTransform;
    private Vector3 originalPosition;

    private Vector2 offset;
    void Awake()
    {
        rectTransform =
GetComponent<RectTransform>();
        canvas = GetComponentInParent<Canvas>();
        originalPosition =
rectTransform.anchoredPosition;

    }

```

```

    public void OnBeginDrag(PointerEventData
eventData)
    {
        // Сохранение оригинальной позиции при
начале перетаскивания

```

```

        // Сохраняем смещение между курсором и
объектом
        RectTransform rectTransform =
GetComponent<RectTransform>();
        originalPosition =
rectTransform.anchoredPosition;
        offset = eventData.position -
RectTransformUtility.WorldToScreenPoint(eventData.
pressEventCamera, rectTransform.position);
    }

```

```

    public void OnDrag(PointerEventData eventData)
    {
        Vector2 position;

```

```

RectTransformUtility.ScreenPointToLocalPointInRecta
ngle(canvas.transform as RectTransform,
eventData.position, canvas.worldCamera, out position);

```

```

;

        Vector2 newPosition = eventData.position - offset;
        RectTransform rectTransform =
GetComponent<RectTransform>();
        rectTransform.position = newPosition;
    }

```

```

    public Vector3 GetCurrentPosition()
    {
        return rectTransform.anchoredPosition;
    }
}

```

Файл JewelryManager1.cs:

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.IO;
using UnityEngine;
using UnityEngine.EventSystems;
using UnityEngine.UI;
using static System.Net.Mime.MediaTypeNames;

```

```

public class JewelryManager1 : MonoBehaviour
{
    public GameObject[] jewelryPrefabs; // массив
префабов
    public Button[] buttons;
    private Dictionary<int, int> quantities = new
Dictionary<int, int>();
    public Transform parentTransform;
    public Vector2 spawnPosition;

```

```

    public List<Draggable1> draggableJewelry = new
List<Draggable1>();

```

```

    void Start()
    {
        LoadQuantities();
        LoadJewelryPositions();
        for (int i = 0; i < buttons.Length; i++)
        {
            int index = i;
            buttons[i].onClick.AddListener(() =>
OnJewelryButtonClick(index));
        }
    }

```

```

    void Update()
    {
        // Проверяем, нажата ли клавиша Backspace
if (Input.GetKeyDown(KeyCode.Backspace))
        {
            RemoveSelectedJewelry();
        }
    }

```

```

    }

    void LoadQuantities()
    {
        string[] lines = File.ReadAllLines("растения.txt");
        for (int i = 0; i < lines.Length; i++)
        {
            if (int.TryParse(lines[i], out int quantity))
            {
                quantities[i] = quantity;
            }
        }
    }

    void LoadJewelryPositions()
    {
        if (File.Exists("positions_kotel.txt"))
        {
            string[] lines =
            File.ReadAllLines("positions_kotel.txt");
            foreach (string line in lines)
            {
                string[] parts = line.Split('|');
                if (parts.Length == 3)
                {
                    string name = parts[0].Substring(0,
                    parts[0].Length - 7);
                    float x = float.Parse(parts[1]);
                    float y = float.Parse(parts[2]);

                    GameObject prefab =
                    System.Array.Find(jewelryPrefabs, p => p.name ==
                    name);
                    if (prefab != null)
                    {
                        GameObject jewelryInstance =
                        Instantiate(prefab, parentTransform);

                        jewelryInstance.GetComponent<RectTransform>().anchoredPosition = new Vector2(x, y);

                        jewelryInstance.AddComponent<Draggable1>(); //
                        добавляем компонент для перетаскивания

                        draggableJewelry.Add(jewelryInstance.GetComponent
                        <Draggable1>()); // добавляем в список
                    }
                }
            }
        }

        public void OnJewelryButtonClick(int index)
        {

```

```

            if (quantities.ContainsKey(index) &&
            quantities[index] > 0)
            {
                GameObject jewelryInstance =
                Instantiate(jewelryPrefabs[index], parentTransform);

                jewelryInstance.GetComponent<RectTransform>().anchoredPosition = spawnPosition;
                quantities[index]--; // Уменьшаем количество
                UpdateQuantityFile();

                jewelryInstance.AddComponent<Draggable1>();

                draggableJewelry.Add(jewelryInstance.GetComponent
                <Draggable1>());
            }

            void UpdateQuantityFile()
            {
                using (StreamWriter writer = new
                StreamWriter("растения.txt"))
                {
                    foreach (var quantity in quantities)
                    {
                        writer.WriteLine(quantity.Value);
                    }
                }

                private void OnApplicationQuit()
                {
                    SavePositions(); // Сохраняем позиции при
                    выходе из приложения
                }

                public void SavePositions()
                {
                    File.WriteAllText("positions_kotel.txt",
                    string.Empty);
                    foreach (var draggable in draggableJewelry)
                    {
                        if (draggable != null)
                        {
                            string positionData =
                            $"{draggable.gameObject.name}|{draggable.GetCurrent
                            tPosition().x}|{draggable.GetCurrentPosition().y}\n"; //
                            Добавляем размер
                            File.AppendAllText("positions_kotel.txt",
                            positionData); // Сохраняем позицию в файл
                        }
                    }
                    draggableJewelry.Clear(); // Очищаем список
                    после сохранения
                }

                private void RemoveSelectedJewelry()
                {

```

```

        PointerEventData pointerData = new
        PointerEventData(EventSystem.current) { position =
        Input.mousePosition };
        List<RaycastResult> results = new
        List<RaycastResult>();
        EventSystem.current.RaycastAll(pointerData,
        results);

        foreach (RaycastResult result in results)
        {
            Draggable1 draggable =
            result.gameObject.GetComponent<Draggable1>();
            if (draggable != null)
            {
                int index =
                GetJewelryIndex(draggable.gameObject.name);
                if (index != -1)
                {
                    // Увеличиваем количество в словаре
                    quantities[index]++;

                    // Обновляем файл после изменения
                    количества
                    UpdateQuantityFile();

                    // Удаляем объект из списка
                    draggableJewelry и уничтожаем его
                    draggableJewelry.Remove(draggable);

                    string[] stroka1 =
                    File.ReadAllLines("крафт.txt");

                    if
                    ((Math.Round(draggable.GetCurrentPosition().x, 2) ==
                    437) &&
                    (Math.Round(draggable.GetCurrentPosition().y, 2) ==
                    146))
                    {
                        stroka1[0] = 0.ToString();

                    }
                    if
                    ((Math.Round(draggable.GetCurrentPosition().x, 2) ==
                    625.2) &&
                    (Math.Round(draggable.GetCurrentPosition().y, 2) ==
                    144.8))
                    {
                        stroka1[1] = 0.ToString();
                    }
                    if
                    ((Math.Round(draggable.GetCurrentPosition().x, 2) ==
                    351.3) &&
                    (Math.Round(draggable.GetCurrentPosition().y, 2) ==
                    23.9))
                    {
                        stroka1[2] = 0.ToString();

```

```

                    }
                    if
                    ((Math.Round(draggable.GetCurrentPosition().x, 2) ==
                    528.8) &&
                    (Math.Round(draggable.GetCurrentPosition().y, 2) ==
                    23.3))
                    {
                        stroka1[3] = 0.ToString();
                    }
                    if
                    ((Math.Round(draggable.GetCurrentPosition().x, 2) ==
                    712.8) &&
                    (Math.Round(draggable.GetCurrentPosition().y, 2) ==
                    23.4))
                    {
                        stroka1[4] = 0.ToString();
                    }
                    if
                    ((Math.Round(draggable.GetCurrentPosition().x, 2) ==
                    437) &&
                    (Math.Round(draggable.GetCurrentPosition().y, 2) == -
                    107.5))
                    {
                        stroka1[5] = 0.ToString();
                    }
                    if
                    ((Math.Round(draggable.GetCurrentPosition().x, 2) ==
                    627) &&
                    (Math.Round(draggable.GetCurrentPosition().y, 2) == -
                    107.6))
                    {
                        stroka1[6] = 0.ToString();
                    }

                    StreamWriter f = new
                    StreamWriter("крафт.txt", false);
                    for (int i = 0; i < stroka1.Length; i++)
                    {
                        f.WriteLine(stroka1[i]);
                    }
                    f.Close();

                    Destroy(draggable.gameObject);
                    break;
                }
            }
        }
        private int GetJewelryIndex(string name)
        {
            for (int i = 0; i < jewelryPrefabs.Length; i++)
            {
                if (name.StartsWith("парт" + (i +
                1).ToString() + "(Clone)"))
                {
                    return i;
                }
            }
            return -1;

```

```
}
```

```
}
```

Файл JewelryManager2.cs:

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.IO;
using UnityEngine;
using UnityEngine.EventSystems;
using UnityEngine.UI;
using static System.Net.Mime.MediaTypeNames;

public class JewelryManager2 : MonoBehaviour
{
    public GameObject jewelryPrefab; // единственный префаб
    public Button button; // единственная кнопка
    private int quantity; // количество
    public Transform parentTransform;
    public Vector2 spawnPosition;
    private List<Draggable2> draggableJewelry = new List<Draggable2>();

    void Start()
    {
        LoadQuantity();
        LoadJewelryPositions();
    }

    button.onClick.AddListener(OnJewelryButtonClick);

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Backspace))
        {
            RemoveSelectedJewelry();
        }
    }

    void LoadQuantity()
    {
        string[] lines =
        File.ReadAllLines("предметы.txt");
        if (lines.Length >= 5 && int.TryParse(lines[4],
        out quantity)) // берем 5 строку
        {
            Debug.Log($"Quantity loaded: {quantity}");
        }
    }

    void LoadJewelryPositions()
    {
        if (File.Exists("positions_kotel_krist.txt"))
        {
```

```
string[] lines =
File.ReadAllLines("positions_kotel_krist.txt");
foreach (string line in lines)
{
    string[] parts = line.Split('|');
    if (parts.Length == 3)
    {
        string name = parts[0].Substring(0,
        parts[0].Length - 7); // обрезаем "(Clone)"
        float x = float.Parse(parts[1]);
        float y = float.Parse(parts[2]);
        if (jewelryPrefab.name == name)
        {
            GameObject jewelryInstance =
            Instantiate(jewelryPrefab, parentTransform);

            jewelryInstance.GetComponent<RectTransform>().anchoredPosition = new Vector2(x, y);

            jewelryInstance.AddComponent<Draggable2>();

            draggableJewelry.Add(jewelryInstance.GetComponent
            <Draggable2>());
        }
    }
}

public void OnJewelryButtonClick()
{
    if (quantity > 0)
    {
        GameObject jewelryInstance =
        Instantiate(jewelryPrefab, parentTransform);

        jewelryInstance.GetComponent<RectTransform>().anchoredPosition = spawnPosition;
        quantity--;
        UpdateQuantityFile();

        jewelryInstance.AddComponent<Draggable2>();

        draggableJewelry.Add(jewelryInstance.GetComponent
        <Draggable2>());
    }
}

void UpdateQuantityFile()
{
    var lines = File.ReadAllLines("предметы.txt");
    if (lines.Length >= 5)
    {
        lines[4] = quantity.ToString(); // обновляем 5
        строку
        File.WriteAllLines("предметы.txt", lines);
    }
}
```

```

private void OnApplicationQuit()
{
    SavePositions();
}

public void SavePositions()
{
    using (StreamWriter writer = new
StreamWriter("positions_kotel_krist.txt"))
    {
        foreach (var draggable in draggableJewelry)
        {
            if (draggable != null)
            {
                string positionData =
$"{{draggable.gameObject.name}}|{{draggable.GetCurrent
tPosition().x}}|{{draggable.GetCurrentPosition().y}}\n";
                writer.Write(positionData);
            }
        }
        draggableJewelry.Clear();
    }

    private void RemoveSelectedJewelry()
    {
        PointerEventData pointerData = new
PointerEventData(EventSystem.current) { position =
Input.mousePosition };
        List<RaycastResult> results = new
List<RaycastResult>();
        EventSystem.current.RaycastAll(pointerData,
results);
        foreach (RaycastResult result in results)
        {
            Draggable2 draggable =
result.gameObject.GetComponent<Draggable2>();
            if (draggable != null)
            {
                quantity++; // увеличиваем количество
                UpdateQuantityFile();
                draggableJewelry.Remove(draggable);

                string[] stroka1 =
File.ReadAllLines("кпафрт.txt");

                if
((Math.Round(draggable.GetCurrentPosition().x, 2) ==
437) &&
(Math.Round(draggable.GetCurrentPosition().y, 2) ==
146))
                {
                    stroka1[0] = 0.ToString();

                }
                if
((Math.Round(draggable.GetCurrentPosition().x, 2) ==

```

```

625.2) &&
(Math.Round(draggable.GetCurrentPosition().y, 2) ==
144.8))
            {
                stroka1[1] = 0.ToString();
            }
            if
((Math.Round(draggable.GetCurrentPosition().x, 2) ==
351.3) &&
(Math.Round(draggable.GetCurrentPosition().y, 2) ==
23.9))
            {
                stroka1[2] = 0.ToString();
            }
            if
((Math.Round(draggable.GetCurrentPosition().x, 2) ==
528.8) &&
(Math.Round(draggable.GetCurrentPosition().y, 2) ==
23.3))
            {
                stroka1[3] = 0.ToString();
            }
            if
((Math.Round(draggable.GetCurrentPosition().x, 2) ==
712.8) &&
(Math.Round(draggable.GetCurrentPosition().y, 2) ==
23.4))
            {
                stroka1[4] = 0.ToString();
            }
            if
((Math.Round(draggable.GetCurrentPosition().x, 2) ==
437) &&
(Math.Round(draggable.GetCurrentPosition().y, 2) == -
107.5))
            {
                stroka1[5] = 0.ToString();
            }
            if
((Math.Round(draggable.GetCurrentPosition().x, 2) ==
627) &&
(Math.Round(draggable.GetCurrentPosition().y, 2) == -
107.6))
            {
                stroka1[6] = 0.ToString();
            }

            StreamWriter f = new
StreamWriter("кпафрт.txt", false);
            for (int i = 0; i < stroka1.Length; i++)
            {
                f.WriteLine(stroka1[i]);
            }
            f.Close();

            Destroy(draggable.gameObject);
            break;
        }
    }
}

```



```

    }
}

```

Файл scitkolrastenij.cs:

```

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

public class scitkolrastenij : MonoBehaviour
{
    public Text[] text;

    void Update()
    {
        string[] stroka =
File.ReadAllLines("растения.txt");
        string[] stroka1 =
File.ReadAllLines("предметы.txt");

        for (int i = 0; i < stroka.Length; i++)
        {
            text[i].text = stroka[i];
        }
        text[15].text = stroka1[4];
    }
}

```

Файл Slot.cs:

```

using System.Collections;
using System.Collections.Generic;
using System.IO;
using Unity.VisualScripting;
using UnityEngine;
using UnityEngine.EventSystems;
using static UnityEngine.RuleTile.TilingRuleOutput;

public class Slot : MonoBehaviour, IDropHandler
{
    public float targetX; // Задайте значение x
    public float targetY; // Задайте значение y
    public string y;
    public int x;
    public void OnDrop(PointerEventData eventData)
    {
        string[] stroka = File.ReadAllLines("крафт.txt");

        Draggable1 item =
eventData.pointerDrag.GetComponent<Draggable1>();

        Draggable2 item2 =
eventData.pointerDrag.GetComponent<Draggable2>();

        if (item != null)
        {
            if (((y == "Slot1") && (int.Parse(stroka[0]) ==
0)) || ((y == "Slot2") && (int.Parse(stroka[1]) == 0)) ||
((y == "Slot3") && (int.Parse(stroka[2]) == 0)) || ((y

```

```

== "Slot4") && (int.Parse(stroka[3]) == 0)) || ((y ==
"Slot5") && (int.Parse(stroka[4]) == 0)) || ((y ==
"Slot6") && (int.Parse(stroka[5]) == 0)) || ((y ==
"Slot7") && (int.Parse(stroka[6]) == 0)))
{

```

```

// Устанавливаем позицию элемента в
заданные координаты (targetX, targetY)
RectTransform itemRectTransform =
item.GetComponent<RectTransform>();
itemRectTransform.anchoredPosition = new
Vector2(targetX, targetY);

```

```

if (y == "Slot1")
{
    if (item.name == "паст9(Clone)")
    {
        x = 1;
    }
    else if (item.name == "паст13(Clone)")
    { x = 2; }
    else if (item.name == "паст2(Clone)")
    { x = 3; }
    else
    { x = -1; }

    stroka[0] = x.ToString();
}

```

```

if (y == "Slot2")
{
    if (item.name == "паст14(Clone)")
    {
        x = 1;
    }
    else if (item.name == "паст4(Clone)")
    { x = 2; }
    else if (item.name == "паст15(Clone)")
    { x = 3; }
    else
    { x = -1; }

    stroka[1] = x.ToString();
}

```

```

if (y == "Slot3")
{
    if (item.name == "паст3(Clone)")
    {
        x = 1;
    }
    else
    { x = -1; }

    stroka[2] = x.ToString();
}

```

```

if (y == "Slot4")

```

```

{
    if (item.name == "паст(Clone)")
    {
        x = 1;
    }
    else
    { x = -1; }

    stroka[3] = x.ToString();
}

if (y == "Slot5")
{
    if (item.name == "паст7(Clone)")
    {
        x = 1;
    }
    else if (item.name == "паст6(Clone)")
    { x = 2; }
    else if (item.name == "паст1(Clone)")
    { x = 3; }
    else
    { x = -1; }

    stroka[4] = x.ToString();
}
if (y == "Slot6")
{
    if (item.name == "паст10(Clone)")
    {
        x = 1;
    }
    else
    { x = -1; }

    stroka[5] = x.ToString();
}
if (y == "Slot7")
{
    if (item.name == "паст8(Clone)")
    {
        x = 1;
    }
    else
    { x = -1; }

    stroka[6] = x.ToString();
}

StreamWriter f = new
StreamWriter("кпафr.txt", false);
for (int i = 0; i < stroka.Length; i++)
{
    f.WriteLine(stroka[i]);
}
f.Close();
}
}

```

```

if (item2 != null)
{
    if (((y == "Slot1") && (int.Parse(stroka[0]) ==
0)) || ((y == "Slot2") && (int.Parse(stroka[1]) == 0)) ||
((y == "Slot3") && (int.Parse(stroka[2]) == 0)) || ((y
== "Slot4") && (int.Parse(stroka[3]) == 0)) || ((y ==
"Slot5") && (int.Parse(stroka[4]) == 0)) || ((y ==
"Slot6") && (int.Parse(stroka[5]) == 0)) || ((y ==
"Slot7") && (int.Parse(stroka[6]) == 0)))
    {

        // Устанавливаем позицию элемента в
заданные координаты (targetX, targetY)
        RectTransform itemRectTransform2 =
item2.GetComponent<RectTransform>();
        itemRectTransform2.anchoredPosition = new
Vector2(targetX, targetY);

        if (y == "Slot1")
        {
            x = -1;

            stroka[0] = x.ToString();
        }

        if (y == "Slot2")
        {
            x = -1;

            stroka[1] = x.ToString();
        }

        if (y == "Slot3")
        {
            x = -1;

            stroka[2] = x.ToString();
        }

        if (y == "Slot4")
        {

            x = 1;

            stroka[3] = x.ToString();
        }

        if (y == "Slot5")
        {
            x = -1;

            stroka[4] = x.ToString();
        }
        if (y == "Slot6")
        {
            x = -1;

```

```

        stroka[5] = x.ToString();
    }
    if (y == "Slot7")
    {
        x = -1;

        stroka[6] = x.ToString();
    }

    StreamWriter f = new
    StreamWriter("крафт.txt", false);
    for (int i = 0; i < stroka.Length; i++)
    {
        f.WriteLine(stroka[i]);
    }
    f.Close();
}
}
}

```

Файл sosdanie.cs:

```

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

```

```

public class sosdanie : MonoBehaviour
{

    public Button[] dt;

    private void Start()
    {
        string[] stroka2
        File.ReadAllLines("зелья_кнопки.txt");

        for (int i = 0; i < stroka2.Length; i++)
        {
            if (int.Parse(stroka2[i]) > 0)
            {
                dt[i].gameObject.SetActive(true);
            }
        }
    }

    public void Sosdanie()
    {
        string[] stroka = File.ReadAllLines("крафт.txt");
        string[] stroka2
        File.ReadAllLines("зелья_кнопки.txt");
        int n = 0;
        for (int i = 0; i < stroka.Length; i++)
        {
            if (int.Parse(stroka[i]) == 1)

```

```

        {
            n += 1;
        }
    }
    if (n == 7)
    {
        dt[0].gameObject.SetActive(true);
        stroka2[0] = (int.Parse(stroka2[0]) + 1).ToString();
        n = 0;

        GameObject[] objects =
        GameObject.FindObjectsOfType<GameObject>();

        foreach (GameObject obj in objects)
        {
            if (obj.name == "паст9(Clone)")
            {
                // Получаем координаты объекта
                double objX =
                Math.Round(obj.transform.position.x, 2);
                double objY =
                Math.Round(obj.transform.position.y, 2);
                // Проверяем, совпадают ли координаты
                if (objX == 1397 && objY == 686)
                {
                    // Удаляем первый найденный объект
                    Destroy(obj);
                    break; // Выходим из цикла после
                    удаления
                }
            }
        }
        foreach (GameObject obj in objects)
        {
            if (obj.name == "паст14(Clone)")
            {
                // Получаем координаты объекта
                double objX =
                Math.Round(obj.transform.position.x, 2);
                double objY =
                Math.Round(obj.transform.position.y, 2);

                // Проверяем, совпадают ли координаты
                if (objX == 1585.2 && objY == 684.8)
                {
                    // Удаляем первый найденный объект
                    Destroy(obj);
                    break; // Выходим из цикла после
                    удаления
                }
            }
        }
        foreach (GameObject obj in objects)
        {
            if (obj.name == "паст3(Clone)")
            {
                // Получаем координаты объекта

```

```

        double          objX          =
Math.Round(obj.transform.position.x, 2);
        double          objY          =
Math.Round(obj.transform.position.y, 2);

        // Проверяем, совпадают ли координаты
        if (objX == 1311.3 && objY == 563.9)
        {
            // Удаляем первый найденный объект
            Destroy(obj);
            break; // Выходим из цикла после
удаления
        }
    }
}

foreach (GameObject obj in objects)
{
    if (obj.name == "паст(Clone)")
    {
        // Получаем координаты объекта
        double          objX          =
Math.Round(obj.transform.position.x, 2);
        double          objY          =
Math.Round(obj.transform.position.y, 2);

        // Проверяем, совпадают ли координаты
        if (objX == 1488.8 && objY == 563.3)
        {
            // Удаляем первый найденный объект
            Destroy(obj);
            break; // Выходим из цикла после
удаления
        }
    }
}

foreach (GameObject obj in objects)
{
    if (obj.name == "паст7(Clone)")
    {
        // Получаем координаты объекта
        double          objX          =
Math.Round(obj.transform.position.x, 2);
        double          objY          =
Math.Round(obj.transform.position.y, 2);

        // Проверяем, совпадают ли координаты
        if (objX == 1672.8 && objY == 563.4)
        {
            // Удаляем первый найденный объект
            Destroy(obj);
            break; // Выходим из цикла после
удаления
        }
    }
}

foreach (GameObject obj in objects)
{
    if (obj.name == "паст10(Clone)")

```

```

    {
        // Получаем координаты объекта
        double          objX          =
Math.Round(obj.transform.position.x, 2);
        double          objY          =
Math.Round(obj.transform.position.y, 2);

        // Проверяем, совпадают ли координаты
        if (objX == 1397 && objY == 432.5)
        {
            // Удаляем первый найденный объект
            Destroy(obj);
            break; // Выходим из цикла после
удаления
        }
    }
}

foreach (GameObject obj in objects)
{
    if (obj.name == "паст8(Clone)")
    {
        // Получаем координаты объекта
        double          objX          =
Math.Round(obj.transform.position.x, 2);
        double          objY          =
Math.Round(obj.transform.position.y, 2);

        // Проверяем, совпадают ли координаты
        if (objX == 1587 && objY == 432.4)
        {
            // Удаляем первый найденный объект
            Destroy(obj);
            break; // Выходим из цикла после
удаления
        }
    }
}

StreamWriter f = new StreamWriter("крафт.txt",
false);
for (int i = 0; i < stroka.Length; i++)
{
    stroka[i] = "0";
    f.WriteLine(stroka[i]);
}
f.Close();
}

if ((int.Parse(stroka[0]) == 2)&&
(int.Parse(stroka[1]) == 2) && (int.Parse(stroka[4]) ==
2) && (int.Parse(stroka[3]) == 1))
{
    dt[1].gameObject.SetActive(true);
    stroka2[1] = (int.Parse(stroka[1]) + 1).ToString();

    GameObject[] objects =
GameObject.FindObjectsOfType<GameObject>();

    foreach (GameObject obj in objects)
    {

```

```

        if (obj.name == "паст13(Clone)")
        {
            // Получаем координаты объекта
            double objX =
Math.Round(obj.transform.position.x, 2);
            double objY =
Math.Round(obj.transform.position.y, 2);
            // Проверяем, совпадают ли координаты
            if (objX == 1397 && objY == 686)
            {
                // Удаляем первый найденный объект
                Destroy(obj);
                break; // Выходим из цикла после
удаления
            }
        }
    }
    foreach (GameObject obj in objects)
    {
        if (obj.name == "паст4(Clone)")
        {
            // Получаем координаты объекта
            double objX =
Math.Round(obj.transform.position.x, 2);
            double objY =
Math.Round(obj.transform.position.y, 2);

            // Проверяем, совпадают ли координаты
            if (objX == 1585.2 && objY == 684.8)
            {
                // Удаляем первый найденный объект
                Destroy(obj);
                break; // Выходим из цикла после
удаления
            }
        }
    }
    foreach (GameObject obj in objects)
    {
        if (obj.name == "паст(Clone)")
        {
            // Получаем координаты объекта
            double objX =
Math.Round(obj.transform.position.x, 2);
            double objY =
Math.Round(obj.transform.position.y, 2);

            // Проверяем, совпадают ли координаты
            if (objX == 1488.8 && objY == 563.3)
            {
                // Удаляем первый найденный объект
                Destroy(obj);
                break; // Выходим из цикла после
удаления
            }
        }
    }
    foreach (GameObject obj in objects)

```

```

        {
            if (obj.name == "паст6(Clone)")
            {
                // Получаем координаты объекта
                double objX =
Math.Round(obj.transform.position.x, 2);
                double objY =
Math.Round(obj.transform.position.y, 2);

                // Проверяем, совпадают ли координаты
                if (objX == 1672.8 && objY == 563.4)
                {
                    // Удаляем первый найденный объект
                    Destroy(obj);
                    break; // Выходим из цикла после
удаления
                }
            }
        }
        StreamWriter f = new StreamWriter("крафт.txt",
false);
        for (int i = 0; i < stroka.Length; i++)
        {
            stroka[i] = "0";
            f.WriteLine(stroka[i]);
        }
        f.Close();
    }
    if ((int.Parse(stroka[0]) == 3) &&
(int.Parse(stroka[1]) == 3) && (int.Parse(stroka[4]) ==
3) && (int.Parse(stroka[3]) == 1))
    {
        dt[2].gameObject.SetActive(true);
        stroka2[2] = (int.Parse(stroka2[2]) + 1).ToString();

        GameObject[] objects =
GameObject.FindObjectsOfType<GameObject>();

        foreach (GameObject obj in objects)
        {
            if (obj.name == "паст2(Clone)")
            {
                // Получаем координаты объекта
                double objX =
Math.Round(obj.transform.position.x, 2);
                double objY =
Math.Round(obj.transform.position.y, 2);
                // Проверяем, совпадают ли координаты
                if (objX == 1397 && objY == 686)
                {
                    // Удаляем первый найденный объект
                    Destroy(obj);
                    break; // Выходим из цикла после
удаления
                }
            }
        }
    }
    foreach (GameObject obj in objects)
    {

```

```

        if (obj.name == "паст15(Clone)")
        {
            // Получаем координаты объекта
            double objX
Math.Round(obj.transform.position.x, 2);
            double objY
Math.Round(obj.transform.position.y, 2);

            // Проверяем, совпадают ли координаты
            if (objX == 1585.2 && objY == 684.8)
            {
                // Удаляем первый найденный объект
                Destroy(obj);
                break; // Выходим из цикла после
удаления
            }
        }
    }
    foreach (GameObject obj in objects)
    {
        if (obj.name == "паст(Clone)")
        {
            // Получаем координаты объекта
            double objX
Math.Round(obj.transform.position.x, 2);
            double objY
Math.Round(obj.transform.position.y, 2);

            // Проверяем, совпадают ли координаты
            if (objX == 1488.8 && objY == 563.3)
            {
                // Удаляем первый найденный объект
                Destroy(obj);
                break; // Выходим из цикла после
удаления
            }
        }
    }
    foreach (GameObject obj in objects)
    {
        if (obj.name == "паст1(Clone)")
        {
            // Получаем координаты объекта
            double objX
Math.Round(obj.transform.position.x, 2);
            double objY
Math.Round(obj.transform.position.y, 2);

            // Проверяем, совпадают ли координаты
            if (objX == 1672.8 && objY == 563.4)
            {
                // Удаляем первый найденный объект
                Destroy(obj);
                break; // Выходим из цикла после
удаления
            }
        }
    }
}

```

```

        StreamWriter f = new StreamWriter("крафт.txt",
false);
        for (int i = 0; i < stroka.Length; i++)
        {
            stroka[i] = "0";
            f.WriteLine(stroka[i]);
        }
        f.Close();
    }

    StreamWriter f2 = new
StreamWriter("зелья_кнопки.txt", false);
    for (int i = 0; i < stroka2.Length; i++)
    {
        f2.WriteLine(stroka2[i]);
    }
    f2.Close();
}

public void sele_клик(int n)
{
    string[] stroka1 = File.ReadAllLines("зелья.txt");
    string[] stroka2 =
File.ReadAllLines("зелья_кнопки.txt");

    if (n == 0) {
        stroka1[0] = (int.Parse(stroka1[0]) + 1).ToString();
        stroka2[0] = 0.ToString();
    }
    if (n == 1)
    {
        stroka1[1] = (int.Parse(stroka1[1]) + 1).ToString();
        stroka2[1] = 0.ToString();
    }
    if (n == 2)
    {
        stroka1[2] = (int.Parse(stroka1[2]) + 1).ToString();
        stroka2[2] = 0.ToString();
    }
    StreamWriter f1 = new StreamWriter("зелья.txt",
false);
    for (int i = 0; i < stroka1.Length; i++)
    {
        f1.WriteLine(stroka1[i]);
    }
    f1.Close();

    StreamWriter f2 = new
StreamWriter("зелья_кнопки.txt", false);
    for (int i = 0; i < stroka2.Length; i++)
    {
        f2.WriteLine(stroka2[i]);
    }
}

```

```

    }
    f2.Close();

    dt[n].gameObject.SetActive(false);
}
}

```

Файл invent.cs:

```

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

public class invent : MonoBehaviour
{
    public Text[] text;

    public Image img;
    public Sprite[] sprites;
    public Переход perehod;
    public Button ispols;
    private void Start()
    {
        img.sprite = sprites[0];
        ispols.gameObject.SetActive(false);
    }

    void Update()
    {
        string[] stroka =
File.ReadAllLines("растения.txt");
        string[] stroka1 =
File.ReadAllLines("предметы.txt");
        string[] stroka2 =
File.ReadAllLines("манеты.txt");
        string[] stroka3 = File.ReadAllLines("карта.txt");
        string[] stroka4 = File.ReadAllLines("зелья.txt");
        string[] stroka5 =
File.ReadAllLines("количество.txt");

        for (int i = 0; i < stroka.Length; i++)
        {
            text[i].text=stroka[i];
        }

        for (int i = 0; i < stroka1.Length; i++)
        {
            text[i+ stroka.Length].text = stroka1[i];
        }

        for (int i = 0; i < stroka2.Length; i++)
        {
            text[i+ stroka.Length+ stroka1.Length].text =
stroka2[i];
        }

        for (int i = 0; i < stroka3.Length; i++)
        {

```

```

            text[i+stroka.Length+ stroka1.Length+
stroka2.Length].text = stroka3[i];
        }

        for (int i = 0; i < stroka4.Length; i++)
        {
            text[i+ stroka.Length+ stroka1.Length+
stroka2.Length+ stroka3.Length].text = stroka4[i];
        }

        for (int i = 0; i < stroka5.Length; i++)
        {
            text[i + stroka.Length + stroka1.Length +
stroka2.Length + stroka3.Length+ stroka4.Length].text
= stroka5[i];
        }

        public void button_clic(int i)
        {
            //для смены номера сцены на кнопке
использовать
            ispols.gameObject.SetActive(false);
            img.sprite=sprites[i];

            string[] stroka1 =
File.ReadAllLines("предметы.txt");
            string[] stroka3 = File.ReadAllLines("карта.txt");
            if ((i == 1) && (int.Parse(stroka3[0])>0))
            {
                ispols.gameObject.SetActive(true);
                perehod.levelToLoad = 16;
            }
            img.sprite = sprites[i];
            if ((i == 2) && (int.Parse(stroka1[0]) > 0))
            {
                ispols.gameObject.SetActive(true);
                perehod.levelToLoad = 10;
            }

            string[] stroka =
File.ReadAllLines("использовать.txt");
            stroka[0]=0.ToString();
            StreamWriter f1 = new
StreamWriter("использовать.txt", false);
            for (int k = 0; k < stroka.Length; k++)
            {
                f1.WriteLine(stroka[k]);
            }
            f1.Close();

        }
        img.sprite = sprites[i];
        if ((i == 4) && (int.Parse(stroka1[1]) > 0))
        {
            ispols.gameObject.SetActive(true);
            perehod.levelToLoad = 14;
        }
        img.sprite = sprites[i];
        if ((i == 5) && (int.Parse(stroka1[2]) > 0))

```

```

    {
        ispols.gameObject.SetActive(true);
        perehod.levelToLoad = 10;

        string[] stroka =
File.ReadAllLines("использовать.txt");
        stroka[0] = 1.ToString();
        StreamWriter f1 = new
StreamWriter("использовать.txt", false);
        for (int k = 0; k < stroka.Length; k++)
        {
            f1.WriteLine(stroka[k]);
        }
        f1.Close();
    }
    img.sprite = sprites[i];
    if ((i == 6) && (int.Parse(stroka1[3]) > 0))
    {
        ispols.gameObject.SetActive(true);
        perehod.levelToLoad = 10;

        string[] stroka =
File.ReadAllLines("использовать.txt");
        stroka[0] = 2.ToString();
        StreamWriter f1 = new
StreamWriter("использовать.txt", false);
        for (int k = 0; k < stroka.Length; k++)
        {
            f1.WriteLine(stroka[k]);
        }
        f1.Close();
    }
}
}

```

Файл kupl\_prod.cs:

```

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

public class kupl_prod : MonoBehaviour
{
    public void k_p(int i)
    {
        string[] stroka = File.ReadAllLines("Купля-
продажа.txt");
        stroka[0]=i.ToString();
        StreamWriter f1 = new StreamWriter("Купля-
продажа.txt", false);
        for (int k = 0; k < stroka.Length; k++)
        {
            f1.WriteLine(stroka[k]);
        }
        f1.Close();
    }
}

```

Файл magas.cs:

```

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

public class magas : MonoBehaviour
{
    public GameObject[] scrol;
    public Text text1;
    public Image img;
    public Sprite[] sprites;
    public Button[] bt;
    void Start()
    {
        string[] stroka = File.ReadAllLines("Номер
сцены.txt");
        string[] stroka1 = File.ReadAllLines("Купля-
продажа.txt");
        string[] stroka2 = File.ReadAllLines("цена в
торговой лавке.txt");
        string[] stroka3 = File.ReadAllLines("цена
продажи.txt");
        string[] stroka4 = File.ReadAllLines("цена в
магазине магии.txt");

        if ((int.Parse(stroka[0]) == 6) &&
(int.Parse(stroka1[0]) == 0))
        {
            scrol[0].SetActive(true);
            img.sprite=sprites[0];
            bt[0].gameObject.SetActive(true);
            text1.text= stroka2[0];
        }
        if ((int.Parse(stroka[0]) == 6) &&
(int.Parse(stroka1[0]) == 1))
        {
            scrol[1].SetActive(true);
            img.sprite = sprites[1];
            bt[1].gameObject.SetActive(true);
            text1.text = stroka3[0];
        }
        if ((int.Parse(stroka[0]) == 7))
        {
            scrol[2].SetActive(true);
            img.sprite = sprites[2];
            bt[2].gameObject.SetActive(true);
            text1.text = stroka4[0];
        }
    }
}

```

Файл pokupca.cs:

```

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

```



```

using UnityEngine.UI;
using System.IO;

public class pokupca : MonoBehaviour
{
    public int num;

    public Text text0;
    public Text text1;
    public Image img;
    public Sprite[] sprites;

    private void Start()
    {
        num = 0;

    }

    void Update()
    {
        string[] stroka = File.ReadAllLines("манеты.txt");
        text0.text = stroka[0];
    }

    public void button_click(int i)
    {
        num = i;
        img.sprite = sprites[i];

        string[] stroka1 = File.ReadAllLines("Номер
сцены.txt");
        if ((int.Parse(stroka1[0]) == 6))
        {
            string[] stroka = File.ReadAllLines("цена в
торговой лавке.txt");
            text1.text = stroka[i];
        }
        if ((int.Parse(stroka1[0]) == 7))
        {
            string[] stroka = File.ReadAllLines("цена в
магазине магии.txt");
            text1.text = stroka[i];
        }
    }

    public void pokur()
    {
        string[] stroka = File.ReadAllLines("манеты.txt");
        string[] stroka1 =
File.ReadAllLines("предметы.txt");
        string[] stroka2 =
File.ReadAllLines("количество.txt");

        if (int.Parse(stroka[0]) >= int.Parse(text1.text))
        {

```

```

            stroka[0] = (int.Parse(stroka[0]) -
int.Parse(text1.text)).ToString();

            text0.text = stroka[0];

            StreamWriter f1 = new
StreamWriter("манеты.txt", false);
            for (int k = 0; k < stroka.Length; k++)
            {
                f1.WriteLine(stroka[k]);
            }
            f1.Close();

            if (num == 0)
            {
                stroka1[0] = (int.Parse(stroka1[0]) +
1).ToString();

                StreamWriter f = new
StreamWriter("предметы.txt", false);
                for (int k = 0; k < stroka1.Length; k++)
                {
                    f.WriteLine(stroka1[k]);
                }
                f.Close();

            }
            if (num == 1)
            {
                stroka1[2] = (int.Parse(stroka1[2]) +
1).ToString();

                StreamWriter f = new
StreamWriter("предметы.txt", false);
                for (int k = 0; k < stroka1.Length; k++)
                {
                    f.WriteLine(stroka1[k]);
                }
                f.Close();

            }
            if ((num >= 2) && (num <= 27))
            {
                stroka2[num - 2] = (int.Parse(stroka2[num -
2]) + 1).ToString();

                StreamWriter f = new
StreamWriter("количество.txt", false);
                for (int k = 0; k < stroka2.Length; k++)
                {
                    f.WriteLine(stroka2[k]);
                }
                f.Close();

            }
        }
    }

    public void pokur_magii()
    {

```

```

        string[] stroka = File.ReadAllLines("манеты.txt");
        string[] stroka1 =
File.ReadAllLines("предметы.txt");
        string[] stroka2 = File.ReadAllLines("карта.txt");
        string[] stroka3 = File.ReadAllLines("зелья.txt");

        if (int.Parse(stroka[0]) >= int.Parse(text1.text))
        {

            stroka[0] = (int.Parse(stroka[0]) -
int.Parse(text1.text)).ToString();

            text0.text = stroka[0];

            StreamWriter f1 = new
StreamWriter("манеты.txt", false);
            for (int k = 0; k < stroka.Length; k++)
            {
                f1.WriteLine(stroka[k]);
            }
            f1.Close();

            if (num == 0)
            {
                stroka2[0] = (int.Parse(stroka2[0]) +
1).ToString();

                StreamWriter f = new
StreamWriter("карта.txt", false);
                for (int k = 0; k < stroka2.Length; k++)
                {
                    f.WriteLine(stroka2[k]);
                }
                f.Close();

            }
            if (num == 1)
            {
                stroka3[0] = (int.Parse(stroka3[0]) +
1).ToString();

                StreamWriter f = new
StreamWriter("зелья.txt", false);
                for (int k = 0; k < stroka3.Length; k++)
                {
                    f.WriteLine(stroka3[k]);
                }
                f.Close();

            }
            if ((num == 2))
            {
                stroka1[1] = (int.Parse(stroka1[1]) +
1).ToString();

                StreamWriter f = new
StreamWriter("предметы.txt", false);
                for (int k = 0; k < stroka1.Length; k++)
                {

```

```

                    f.WriteLine(stroka1[k]);
                }
                f.Close();
            }
            if ((num == 3))
            {
                stroka1[3] = (int.Parse(stroka1[3]) +
1).ToString();

                StreamWriter f = new
StreamWriter("предметы.txt", false);
                for (int k = 0; k < stroka1.Length; k++)
                {
                    f.WriteLine(stroka1[k]);
                }
                f.Close();
            }
        }
    }
}

```

Файл prodaja.cs:

```

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

public class prodaja : MonoBehaviour
{
    public Text[] text;

    public int num;

    public Text text0;
    public Text text1;
    public Text text2;
    public Image img;
    public Sprite[] sprites;

    void Start()
    {
        num = 0;
    }

    void Update()
    {
        string[] stroka =
File.ReadAllLines("растения.txt");
        string[] stroka1 =
File.ReadAllLines("предметы.txt");
        string[] stroka2 = File.ReadAllLines("зелья.txt");

        for (int i = 0; i < stroka.Length; i++)
        {
            text[i].text = stroka[i];

```

```

    }

    text[stroka.Length].text = stroka1[4];

    for (int i = 1; i < stroka2.Length; i++)//зелья в
    файле со 2 строки
    {
        text[i + stroka.Length + 1-1].text = stroka2[i];
    }
}

public void button_click(int i)
{
    num = i;
    img.sprite = sprites[i];
    string[] stroka = File.ReadAllLines("цена
продажи.txt");
    text1.text = stroka[i];
}

public void prod()
{
    string[] stroka = File.ReadAllLines("манеты.txt");
    string[] stroka1 =
File.ReadAllLines("растения.txt");
    string[] stroka2 =
File.ReadAllLines("предметы.txt");
    string[] stroka3 = File.ReadAllLines("зелья.txt");

    if (int.Parse(text[num].text) >=
int.Parse(text2.text))
    {

        stroka[0] = (int.Parse(stroka[0]) +
(int.Parse(text1.text)* int.Parse(text2.text))).ToString();

        text0.text = stroka[0];

        StreamWriter f1 = new
StreamWriter("манеты.txt", false);
        for (int k = 0; k < stroka.Length; k++)
        {
            f1.WriteLine(stroka[k]);
        }
        f1.Close();

        if ((num >= 0) && (num <= 14))
        {
            stroka1[num] = (int.Parse(stroka1[num]) -
int.Parse(text2.text)).ToString();

            StreamWriter f = new
StreamWriter("растения.txt", false);
            for (int k = 0; k < stroka1.Length; k++)
            {
                f.WriteLine(stroka1[k]);
            }
            f.Close();

```

```

    }

    if (num == 15)
    {
        stroka2[4] = (int.Parse(stroka2[4]) -
int.Parse(text2.text)).ToString();

        StreamWriter f = new
StreamWriter("предметы.txt", false);
        for (int k = 0; k < stroka2.Length; k++)
        {
            f.WriteLine(stroka2[k]);
        }
        f.Close();
    }

    if ((num >= 16) && (num <= 17))
    {
        stroka3[num-15] = (int.Parse(stroka3[num-
15]) - int.Parse(text2.text)).ToString();

        StreamWriter f = new
StreamWriter("растения.txt", false);
        for (int k = 0; k < stroka3.Length; k++)
        {
            f.WriteLine(stroka3[k]);
        }
        f.Close();
    }
}

}

}

```

Файл Draggable.cs:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.IO;
using UnityEngine.EventSystems;

public class Draggable : MonoBehaviour,
IDragHandler, IBeginDragHandler
{
    private Canvas canvas;
    private RectTransform rectTransform;
    private Vector3 originalPosition;

    void Awake()
    {
        rectTransform =
GetComponent<RectTransform>();
        canvas = GetComponentInParent<Canvas>();
        originalPosition =
rectTransform.anchoredPosition;
    }

```

```

    public void OnBeginDrag(PointerEventData
eventData)
    {
        // Сохранение оригинальной позиции при
        начале перетаскивания
        originalPosition =
        rectTransform.anchoredPosition;
    }

    public void OnDrag(PointerEventData eventData)
    {
        Vector2 position;

        RectTransformUtility.ScreenPointToLocalPointInRecta
ngle(canvas.transform as RectTransform,
eventData.position, canvas.worldCamera, out position);
        rectTransform.anchoredPosition = position;
    }

    public Vector3 GetCurrentPosition()
    {
        return rectTransform.anchoredPosition;
    }
}

```

Файл inwentari.cs:

```

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

public class inwentari : MonoBehaviour
{
    public Text[] text;
    public GameObject plus;
    public bool pl=true;

    void Update()
    {
        string[] stroka =
        File.ReadAllLines("количество.txt");

        for (int i = 0; i < stroka.Length; i++)
        {
            text[i].text = stroka[i];
        }
    }

    public void Plus()
    {
        if (!pl)
        {
            plus.gameObject.SetActive(false);
            pl = true;
        }
    }
}

```

```

    else
    {
        plus.gameObject.SetActive(true);
        pl = false;
    }
}

```

Файл JewelryManager.cs:

```

using System.Collections;
using System.Collections.Generic;
using System.IO;
using UnityEngine;
using UnityEngine.EventSystems;
using UnityEngine.UI;
using static System.Net.Mime.MediaTypeNames;

public class JewelryManager : MonoBehaviour
{
    public GameObject[] jewelryPrefabs; // массив
префабов украшений
    public Button[] buttons;
    private Dictionary<int, int> quantities = new
Dictionary<int, int>();
    public Transform parentTransform;
    public Vector2 spawnPosition;

    private List<Draggable> draggableJewelry = new
List<Draggable>();

    void Start()
    {
        LoadQuantities();
        LoadJewelryPositions();
        for (int i = 0; i < buttons.Length; i++)
        {
            int index = i;
            buttons[i].onClick.AddListener(() =>
OnJewelryButtonClick(index));
        }
    }

    void Update()
    {
        // Проверяем, нажата ли клавиша Backspace
        if (Input.GetKeyDown(KeyCode.Backspace))
        {
            RemoveSelectedJewelry();
        }

        // Проверяем, нажата ли клавиша Tab для
поворота украшения
        if (Input.GetKeyDown(KeyCode.Tab))
        {
            RotateSelectedJewelry();
        }
    }
}

```

```

        // Проверяем, нажаты ли клавиши + и - для
изменения размера
        if (Input.GetKeyDown(KeyCode.Equals)) //
Клавиша + на большинстве раскладок
        {
            ChangeSelectedJewelrySize(1.1f); //
Увеличиваем размер
        }
        else if (Input.GetKeyDown(KeyCode.Minus))
        {
            ChangeSelectedJewelrySize(0.9f); //
Уменьшаем размер
        }

        // Проверяем, нажата ли клавиша Tab для
поворота украшения по окружности
        if (Input.GetKeyDown(KeyCode.Q))
        {
            RotateSelectedJewelry1();
        }
    }

    void LoadQuantities()
    {
        string[] lines =
File.ReadAllLines("количество.txt");
        for (int i = 0; i < lines.Length; i++)
        {
            if (int.TryParse(lines[i], out int quantity))
            {
                quantities[i] = quantity;
            }
        }
    }

    void LoadJewelryPositions()
    {
        if (File.Exists("positions.txt"))
        {
            string[] lines =
File.ReadAllLines("positions.txt");
            foreach (string line in lines)
            {
                string[] parts = line.Split('|');
                if (parts.Length == 5) // Изменили на 4 для
учета размера
                {
                    string name = parts[0].Substring(0,
parts[0].Length - 7);
                    float x = float.Parse(parts[1]);
                    float y = float.Parse(parts[2]);
                    float size = float.Parse(parts[3]);
                    float angle = float.Parse(parts[4]);
                    float size1 = size;
                    if (size < 0)
                    {
                        size1 *= -1;

```

```

        }

        GameObject prefab =
System.Array.Find(jewelryPrefabs, p => p.name ==
name);
        if (prefab != null)
        {
            GameObject jewelryInstance =
Instantiate(prefab, parentTransform);

jewelryInstance.GetComponent<RectTransform>().anc
horedPosition = new Vector2(x, y);
            jewelryInstance.transform.rotation =
Quaternion.Euler(0, 0, angle); // Устанавливаем угол
            jewelryInstance.transform.localScale =
new Vector3(size, size1, 1); // Устанавливаем размер

jewelryInstance.AddComponent<Draggable>(); //
добавляем компонент для перетаскивания

draggableJewelry.Add(jewelryInstance.GetComponent
<Draggable>()); // добавляем в список
        }
    }

    public void OnJewelryButtonClick(int index)
    {
        if (quantities.ContainsKey(index) &&
quantities[index] > 0)
        {
            GameObject jewelryInstance =
Instantiate(jewelryPrefabs[index], parentTransform);

jewelryInstance.GetComponent<RectTransform>().anc
horedPosition = spawnPosition;
            quantities[index]--; // Уменьшаем количество
UpdateQuantityFile();
            jewelryInstance.AddComponent<Draggable>();

draggableJewelry.Add(jewelryInstance.GetComponent
<Draggable>());
        }
    }

    void UpdateQuantityFile()
    {
        using (StreamWriter writer = new
StreamWriter("количество.txt"))
        {
            foreach (var quantity in quantities)
            {
                writer.WriteLine(quantity.Value);
            }
        }
    }

```

```

private void OnApplicationQuit()
{
    SavePositions(); // Сохраняем позиции при
    выходе из приложения
}

public void SavePositions()
{
    File.WriteAllText("positions.txt", string.Empty);
    foreach (var draggable in draggableJewelry)
    {
        string positionData =
        $"{draggable.gameObject.name}|{draggable.GetCurrentPosition().x}|{draggable.GetCurrentPosition().y}|{draggable.transform.localScale.x}|{draggable.transform.rotation.eulerAngles.z}\n"; // Добавляем размер
        File.AppendAllText("positions.txt", positionData); // Сохраняем позицию в файл
    }
    draggableJewelry.Clear(); // Очищаем список после сохранения
}

private void RemoveSelectedJewelry()
{
    PointerEventData pointerData = new
    PointerEventData(EventSystem.current) { position =
    Input.mousePosition };
    List<RaycastResult> results = new
    List<RaycastResult>();
    EventSystem.current.RaycastAll(pointerData,
    results);

    foreach (RaycastResult result in results)
    {
        Draggable draggable =
        result.gameObject.GetComponent<Draggable>();
        if (draggable != null)
        {
            int index =
            GetJewelryIndex(draggable.gameObject.name);
            if (index != -1)
            {
                // Увеличиваем количество в словаре
                quantities[index]++;

                // Обновляем файл после изменения
                количества
                UpdateQuantityFile();

                // Удаляем объект из списка
                draggableJewelry и уничтожаем его
                draggableJewelry.Remove(draggable);
                Destroy(draggable.gameObject);
                break;
            }
        }
    }
}

```

```

private int GetJewelryIndex(string name)
{
    for (int i = 0; i < jewelryPrefabs.Length; i++)
    {
        if (name.StartsWith("укр" + (i + 1).ToString() +
        "(Clone)"))
        {
            return i;
        }
    }
    return -1;
}

private void RotateSelectedJewelry()
{
    PointerEventData pointerData = new
    PointerEventData(EventSystem.current) { position =
    Input.mousePosition };
    List<RaycastResult> results = new
    List<RaycastResult>();

    EventSystem.current.RaycastAll(pointerData,
    results);

    foreach (RaycastResult result in results)
    {
        Draggable draggable =
        result.gameObject.GetComponent<Draggable>();
        if (draggable != null)
        {
            draggable.transform.localScale = new
            Vector3(-draggable.transform.localScale.x,
            draggable.transform.localScale.y, 1); // Поворачиваем
            в противоположную сторону
            break; // Поворачиваем только первое
            найденное украшение
        }
    }
}

private void RotateSelectedJewelry1()
{
    PointerEventData pointerData = new
    PointerEventData(EventSystem.current) { position =
    Input.mousePosition };
    List<RaycastResult> results = new
    List<RaycastResult>();
    EventSystem.current.RaycastAll(pointerData,
    results);

    foreach (RaycastResult result in results)
    {
        Draggable draggable =
        result.gameObject.GetComponent<Draggable>();
        if (draggable != null)
        {
            draggable.transform.Rotate(0, 0, 90); //
            Поворачиваем на 90 градусов
        }
    }
}

```

```

        break; // Поворачиваем только первое
        найденное украшение
    }
}
}
private void ChangeSelectedJewelrySize(float
scaleFactor)
{
    PointerEventData pointerData = new
    PointerEventData(EventSystem.current) { position =
    Input.mousePosition };
    List<RaycastResult> results = new
    List<RaycastResult>();
    EventSystem.current.RaycastAll(pointerData,
    results);
    foreach (RaycastResult result in results)
    {
        Draggable draggable =
        result.gameObject.GetComponent<Draggable>();
        if (draggable != null)
        {
            draggable.transform.localScale *=
            scaleFactor; // Изменяем размер
            break; // Изменяем только первое
            найденное украшение
        }
    }
}
}
}

```

Файл Enemy.cs:

```

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

public class Enemy : MonoBehaviour
{
    public int health;
    public GameObject deathEffect;

    private void Update()
    {
        if (health <= 0)
        {
            Instantiate(deathEffect, transform.position,
            Quaternion.identity);

            FindObjectOfType<Rang>().heal = 0;

            Destroy(gameObject);
        }
    }
    public void TakeDamage(int damage)
    { health -= damage; }
}

```

Файл Exit.cs:

```

using System.Collections;

```

```

using System.Collections.Generic;
using UnityEngine;

```

```

public class Exit : MonoBehaviour
{
    public void ExitGame()
    {
        Application.Quit();
    }
}

```

Файл Gan.cs:

```

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

```

```

public class Gan : MonoBehaviour
{
    public float offset;
    public GameObject bullet;
    public Transform shotPoint;
    private float timeBtwShots;
    public float startTimeBtwShots;
    // Укажите объект, к которому будут
    прикрепляться пули
    public Transform parentObject;

```

```

    private void Update()
    {
        Vector3 difference =
        Camera.main.ScreenToWorldPoint(Input.mousePosition) -
        transform.position;
        float rotZ = Mathf.Atan2(difference.y,
        difference.x) * Mathf.Rad2Deg;
        transform.rotation = Quaternion.Euler(0f, 0f, rotZ
        + offset);
        if (timeBtwShots <= 0)
        {
            if (Input.GetMouseButtonDown(0))
            {
                // Создаем пулю с фиксированным
                вращением
                GameObject newBullet = Instantiate(bullet,
                shotPoint.position, Quaternion.Euler(0f, 0f, 0f));

                // Устанавливаем родителя для новой пули
                newBullet.transform.SetParent(parentObject);

                // Задаем направление полета для пули
                Bullet bulletScript =
                newBullet.GetComponent<Bullet>();

                bulletScript.SetDirection(difference.normalized); //
                Передаем направление

```

```

                timeBtwShots = startTimeBtwShots;
            }
        }
        else

```

```

        {
            timeBtwShots -= Time.deltaTime;
        }
    }
}
Файл Karta.cs:
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.IO;
using UnityEngine.UI;

public class Karta : MonoBehaviour
{
    public Button[] bt;
    public Image imgObj1;
    public Sprite[] sprites;
    void Start()
    {
        string[] stroka = File.ReadAllLines("посещённые
локации.txt");
        if (int.Parse(stroka[0])==1)
        {
            bt[0].gameObject.SetActive(true);
            imgObj1.sprite = sprites[0];
        }
        if (int.Parse(stroka[1]) == 1)
        {
            bt[1].gameObject.SetActive(true);
            imgObj1.sprite = sprites[1];
        }
    }
}

```

```

Файл Konec_igr.cs:
using System.Collections;
using System.Collections.Generic;
using System.IO;
using UnityEngine;
using UnityEngine.UI;

public class Konec_igr : MonoBehaviour
{
    public GameObject endddd;

    bool prod = false;

    private void Update()
    {
        string[] stroka = File.ReadAllLines("выполнение
зд.txt");

        int z21 = 0;
        for (int i = 0; i < stroka.Length; i++)
        {

```

```

            if (int.Parse(stroka[i]) == 1)
            {
                z21++;
            }
        }
        if ((z21== 21) && (prod == false))
        {
            endddd.gameObject.SetActive(true);
            StartCoroutine(stop());
            prod = true;
        }
    }
    private IEnumerator stop()
    {
        yield return new WaitForSeconds(3f);
        Time.timeScale = 0f;
    }

    public void prodolj()
    {
        Time.timeScale = 1f;
        endddd.gameObject.SetActive(false);
    }
}
Файл Kristall.cs:
using System.Collections;
using System.Collections.Generic;
using System.IO;
using UnityEngine;
public class Kristall : MonoBehaviour
{
    public GameObject kristall_pref;
    public Transform parentTransform;
    private void Awake()
    {
        // Получаем родителя текущего объекта и
        // записываем его в переменную
        parentTransform = transform.parent;
    }
    private void OnTriggerEnter2D(Collider2D other)
    {
        if (other.CompareTag("Player"))
        {
            string[] stroka =
File.ReadAllLines("предметы.txt");
            stroka[4]=(int.Parse(stroka[4])+1).ToString();
            FindObjectOfType<porajenie>().kol_vo += 1;

            StreamWriter f = new
StreamWriter("предметы.txt", false);
            for (int i = 0; i < stroka.Length; i++)
            {
                f.WriteLine(stroka[i]);
            }
            f.Close();
            // Задаем массив возможных позиций
            Vector2[] positions = new Vector2[]
            {
                new Vector2(453f, 171f),
                new Vector2(-1104f, -553f),
                new Vector2(1533f, -540f)
            }

```



```

    };

    // Генерируем случайный индекс
    int randomIndex = Random.Range(0,
positions.Length);

    GameObject prefab = kristall_pref;
    if (prefab != null)
    {
        GameObject jewelryInstance =
Instantiate(prefab, parentTransform);

jewelryInstance.GetComponent<RectTransform>().anchoredPosition = positions[randomIndex]; //
Устанавливаем позицию
    }

    Destroy(gameObject);
}
}
}

```

Файл LoadingScreen1.cs:

```

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

public class LoadingScreen1 : MonoBehaviour
{
    public Slider progressBar;
    public string sceneToLoad;

    private void Start()
    {
        StartCoroutine(LoadSceneAsync());
    }

    IEnumerator LoadSceneAsync()
    {
        AsyncOperation operation =
SceneManager.LoadSceneAsync( sceneToLoad );
        while (!operation.isDone)
        {
            float progress = Mathf.Clamp01(
operation.progress/0.9f);
            progressBar.value = progress;
            yield return null;
        }
    }
}

```

Файл MusicManager.cs:

```

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

```

```

public class MusicManager : MonoBehaviour
{
    public AudioClip[] musicClips;
    public AudioSource audioSource;
    private void Awake()
    {
        if (FindObjectsOfType<MusicManager>().Length
> 1)
        {
            Destroy(gameObject);
            return;
        }
        DontDestroyOnLoad(gameObject);
        audioSource =
gameObject.AddComponent<AudioSource>();
        audioSource.loop = true;
        PlayMusic(0);
    }
    public void PlayMusic(int index)
    {
        if (index < musicClips.Length)
        {
            audioSource.clip = musicClips[index];
            audioSource.Play();
        }
    }
    public void SetVolume(float volume)
    {
        audioSource=GetComponent<AudioSource>();//
audioSource.volume = volume;
    }

    public void TogglePause()
    {
        if (audioSource.isPlaying) audioSource.Pause();
        else audioSource.UnPause();
    }
}

```

Файл MusicSelectionUI.cs:

```

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

public class MusicSelectionUI : MonoBehaviour
{
    private MusicManager musicManager;

    private void Start()
    {
        musicManager =
FindObjectOfType<MusicManager>();
    }
    public void SelectMusic(int index)
    {
        if (musicManager != null)
        {
            musicManager.PlayMusic(index);
        }
    }
}

```

```

    }
}
public void TogglePause()
{
    if (musicManager != null)
    {
        musicManager.TogglePause();
    }
}
}
Файл Oto_rang.cs:
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.IO;

public class Oto_rang : MonoBehaviour
{
    public GameObject[] images;
    public Text rang;
    void Update()
    {
        string[] stroka = File.ReadAllLines("rang.txt");

        if (int.Parse(stroka[0]) > 0)
        {
            rang.gameObject.SetActive(true);
        }

        if
((int.Parse(stroka[0])>0)&&(int.Parse(stroka[0])<=3))
        {
            rang.text = "F";
        }
        if ((int.Parse(stroka[0]) >= 5) &&
(int.Parse(stroka[0]) <= 9))
        {
            rang.text = "E";
        }
        if ((int.Parse(stroka[0]) >= 12) &&
(int.Parse(stroka[0]) <= 18))
        {
            rang.text = "D";
        }
        if ((int.Parse(stroka[0]) >= 22) &&
(int.Parse(stroka[0]) <= 30))
        {
            rang.text = "C";
        }
        if ((int.Parse(stroka[0]) >= 35) &&
(int.Parse(stroka[0]) <= 45))
        {
            rang.text = "D";
        }
        if ((int.Parse(stroka[0]) >= 51) &&
(int.Parse(stroka[0]) <= 63))
        {
            rang.text = "A";
        }
    }
}

```

```

    }
    if ((int.Parse(stroka[0]) >= 70) &&
(int.Parse(stroka[0]) <= 84))
    {
        rang.text = "S";
    }
    if ((int.Parse(stroka[0]) == 1) ||
(int.Parse(stroka[0]) == 5) || (int.Parse(stroka[0]) == 6)
|| ((int.Parse(stroka[0]) >= 12) &&(int.Parse(stroka[0])
<= 14)) || ((int.Parse(stroka[0]) >= 22) &&
(int.Parse(stroka[0]) <= 25)) || ((int.Parse(stroka[0]) >=
35) && (int.Parse(stroka[0]) <= 39)) ||
((int.Parse(stroka[0]) >= 51) && (int.Parse(stroka[0])
<= 56)) || ((int.Parse(stroka[0]) >= 70) &&
(int.Parse(stroka[0]) <= 76)))
    {
        images[0].gameObject.SetActive(true);
    }
    if ((int.Parse(stroka[0]) == 2) ||
(int.Parse(stroka[0]) == 7) || (int.Parse(stroka[0]) == 8)
|| ((int.Parse(stroka[0]) >= 15) && (int.Parse(stroka[0])
<= 17)) || ((int.Parse(stroka[0]) >= 26) &&
(int.Parse(stroka[0]) <= 29)) || ((int.Parse(stroka[0]) >=
40) && (int.Parse(stroka[0]) <= 44)) ||
((int.Parse(stroka[0]) >= 57) && (int.Parse(stroka[0])
<= 62)) || ((int.Parse(stroka[0]) >= 77) &&
(int.Parse(stroka[0]) <= 83)))
    {
        images[0].gameObject.SetActive(true);
        images[1].gameObject.SetActive(true);
    }
    if ((int.Parse(stroka[0]) == 3) ||
(int.Parse(stroka[0]) == 4) || ((int.Parse(stroka[0]) >= 9)
&& (int.Parse(stroka[0]) <= 11)) ||
((int.Parse(stroka[0]) >= 18) && (int.Parse(stroka[0])
<= 21)) || ((int.Parse(stroka[0]) >= 30) &&
(int.Parse(stroka[0]) <= 34)) || ((int.Parse(stroka[0]) >=
45) && (int.Parse(stroka[0]) <= 50)) ||
((int.Parse(stroka[0]) >= 63) && (int.Parse(stroka[0])
<= 69)) || (int.Parse(stroka[0]) >= 84))
    {
        images[0].gameObject.SetActive(true);
        images[1].gameObject.SetActive(true);
        images[2].gameObject.SetActive(true);
    }
}
}

```

```

Файл PausMenu.cs:
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.IO;

public class PausMenu : MonoBehaviour
{
    public Image imgObg;
    public Sprite spriteImage1;
}

```

```

public Sprite spriteImage2;
public Button karta;
public bool gameIsPause = false;
public GameObject pauseMenuUI;
public void Start()
{
    string[] stroka = File.ReadAllLines("kapra.txt");
    if (stroka[0] == "0")
    {
        imgObg.sprite = spriteImage1;
    }
    else
    {
        imgObg.sprite = spriteImage2;
        karta.gameObject.SetActive(true);
    }
}
public void Button_menu()
{
    Pause();
}
public void Button_resum()
{
    Resume();
}

void Resume()
{
    pauseMenuUI.SetActive(false);
    Time.timeScale = 1f;
    gameIsPause = false;
}
void Pause()
{
    pauseMenuUI.SetActive(true);
    Time.timeScale = 0f;
    gameIsPause=true;
}
}
Файл playerControler.cs:
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.Threading.Tasks;
public class playerControler : MonoBehaviour
{
    public float speed;
    public float jumpForce;
    private float moveInput;
    private Rigidbody2D rb;
    private bool facingRight=true;
    private bool isGrounded;
    public Transform freePos;
    public float checkRadius;
    public LayerMask whatIsGround;
    private Animator anim;
    private void Start()
    {
        anim = GetComponent<Animator>();

```

```

        rb = GetComponent<Rigidbody2D>();
    }
    private void FixedUpdate()
    {
        moveInput = Input.GetAxis("Horizontal");
        rb.velocity = new Vector2(moveInput*speed,
        rb.velocity.y);

        if (facingRight==false && moveInput>0 )
        { Flip(); }
        else if (facingRight==true && moveInput<0 ) {
        Flip(); }

        if(moveInput==0 ) {
            anim.SetBool("isRunning", false);
        }
        else
        {
            anim.SetBool("isRunning", true);
        }
    }
    private void Update()
    {
        isGrounded =
        Physics2D.OverlapCircle(freePos.position,
        checkRadius, whatIsGround);
        if (isGrounded == true &&
        ((Input.GetKeyDown(KeyCode.Space)) ||
        (Input.GetKeyDown(KeyCode.UpArrow)) ||
        (Input.GetKeyDown(KeyCode.W))))
        {
            rb.velocity = Vector2.up * jumpForce;
            anim.SetTrigger("taceOf");
        }
        if (isGrounded==true)
        {
            anim.SetBool("isJumping",false);
        }
        else
        {
            anim.SetBool("isJumping", true);
        }

        if (isGrounded == true &&
        ((Input.GetKeyDown(KeyCode.DownArrow) ||
        (Input.GetKeyDown(KeyCode.S))))))
        {
            anim.SetBool("isPrised", true);
        }
        else { anim.SetBool("isPrised", false); }
    }
    void Flip()
    {
        facingRight=!facingRight;
        Vector3 Scaler = transform.localScale;
        Scaler.x *= -1;
        transform.localScale = Scaler;
    }
}

```

```

    }
Файл Rang.cs:
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.IO;
using UnityEngine.UI;
using UnityEngine.UIElements;

public class Rang : MonoBehaviour
{
    public GameObject pauseMenuUI;
    public Text text1;
    public int heal;
    public bool b;

    public void Start()
    {
        heal = 10;
        b=true;
    }
    void Update()
    {
        if (((FindObjectOfType<hp>()).hp <= 0) || (heal ==
0))&&(b==true))
        {
            b = false;
            StartCoroutine(HandleGameOver());
        }
    }
    private IEnumerator HandleGameOver()
    {
        // Ждем секунд перед остановкой времени и
        отображением меню
        yield return new WaitForSeconds(1f);

        // Останавливаем время
        Time.timeScale = 0f;
        string[] stroka = File.ReadAllLines("панг.txt");

        if (FindObjectOfType<hp>().hp < 2)
        {
            text1.text = "F";
            stroka[0] = "1";
        }
        else
        {
            text1.text = "E";
            stroka[0] = "5";
        }
        using (StreamWriter f = new
StreamWriter("панг.txt", false))
        {
            for (int k = 0; k < stroka.Length; k++)
            {
                f.WriteLine(stroka[k]);
            }
        }
        // Отображаем меню паузы

```

```

        pauseMenuUI.SetActive(true);
    }
}

Файл Restart_game.cs:
using System.Collections;
using System.Collections.Generic;
using System.IO;
using UnityEngine;
using UnityEngine.UI;
public class Restart_game : MonoBehaviour
{
    public GameObject endddd;
    public void restart_game()
    {
        string[] stroka3 = File.ReadAllLines("время
скрытия.txt");//
        string[] stroka4= File.ReadAllLines("выполнение
зд.txt");//
        string[] stroka6 =
File.ReadAllLines("зелья.txt");//
        string[] stroka7 =
File.ReadAllLines("зелья_кнопки.txt");//
        string[] stroka8 =
File.ReadAllLines("использовать.txt");//
        string[] stroka9 =
File.ReadAllLines("капта.txt");//
        string[] stroka10 =
File.ReadAllLines("количество.txt");//
        string[] stroka11 =
File.ReadAllLines("крафт.txt");//
        string[] stroka12 = File.ReadAllLines("Купля-
продажа.txt");//
        string[] stroka13 =
File.ReadAllLines("манеты.txt");//
        string[] stroka14 = File.ReadAllLines("номер
принятого задания.txt");//
        string[] stroka15 = File.ReadAllLines("Номер
сцены для муз.txt");//
        string[] stroka16 = File.ReadAllLines("Номер
сцены.txt");//
        string[] stroka17 =
File.ReadAllLines("посещённые локации.txt");//
        string[] stroka18 =
File.ReadAllLines("предметы.txt");//
        string[] stroka19 =
File.ReadAllLines("панг.txt");//
        string[] stroka20 =
File.ReadAllLines("растения.txt");//
        File.WriteAllText("positions.txt", string.Empty);
        File.WriteAllText("positions_kotel.txt",
string.Empty);
        File.WriteAllText("positions_kotel_krist.txt",
string.Empty);

        StreamWriter f3 = new StreamWriter("время
скрытия.txt", false);
        for (int k = 0; k < stroka3.Length; k++)
        {

```

```

        stroka3[k] = "03.05.2025 17:41:54";
        f3.WriteLine(stroka3[k]);
    }
    f3.Close();
    StreamWriter f4 = new
StreamWriter("выполнение зд.txt", false);
    for (int k = 0; k < stroka4.Length; k++)
    {
        if ((k >= 0) && (k < 6))
        { stroka4[k] = 0.ToString(); }
        else
        { stroka4[k] = 1.ToString(); }
        f4.WriteLine(stroka4[k]);
    }
    f4.Close();
    StreamWriter f6 = new StreamWriter("зелья.txt",
false);
    for (int k = 0; k < stroka6.Length; k++)
    {
        stroka6[k] = 0.ToString();
        f6.WriteLine(stroka6[k]);
    }
    f6.Close();
    StreamWriter f7 = new
StreamWriter("зелья_кнопки.txt", false);
    for (int k = 0; k < stroka7.Length; k++)
    {
        stroka7[k] = 0.ToString();
        f7.WriteLine(stroka7[k]);
    }
    f7.Close();
    StreamWriter f8 = new
StreamWriter("использовать.txt", false);
    for (int k = 0; k < stroka8.Length; k++)
    {
        stroka8[k] = 0.ToString();
        f8.WriteLine(stroka8[k]);
    }
    f8.Close();
    StreamWriter f9 = new StreamWriter("карта.txt",
false);
    for (int k = 0; k < stroka9.Length; k++)
    {
        stroka9[k] = 0.ToString();
        f9.WriteLine(stroka9[k]);
    }
    f9.Close();

    StreamWriter f10 = new
StreamWriter("количество.txt", false);
    for (int k = 0; k < stroka10.Length; k++)
    {
        stroka10[k] = 0.ToString();
        f10.WriteLine(stroka10[k]);
    }
    f10.Close();
    StreamWriter f11 = new
StreamWriter("крафт.txt", false);
    for (int k = 0; k < stroka11.Length; k++)

```

```

    {
        stroka11[k] = 0.ToString();
        f11.WriteLine(stroka11[k]);
    }
    f11.Close();
    StreamWriter f12 = new StreamWriter("Купля-
продажа.txt", false);
    for (int k = 0; k < stroka12.Length; k++)
    {
        stroka12[k] = 0.ToString();
        f12.WriteLine(stroka12[k]);
    }
    f12.Close();
    StreamWriter f13 = new
StreamWriter("манеты.txt", false);
    for (int k = 0; k < stroka13.Length; k++)
    {
        stroka13[k] = 0.ToString();
        f13.WriteLine(stroka13[k]);
    }
    f13.Close();
    StreamWriter f14 = new StreamWriter("номер
принятого задания.txt", false);
    for (int k = 0; k < stroka14.Length; k++)
    {
        stroka14[k] = 0.ToString();
        f14.WriteLine(stroka14[k]);
    }
    f14.Close();
    StreamWriter f15 = new StreamWriter("Номер
сцены для муз.txt", false);
    for (int k = 0; k < stroka15.Length; k++)
    {
        stroka15[k] = 0.ToString();
        f15.WriteLine(stroka15[k]);
    }
    f15.Close();
    StreamWriter f16 = new StreamWriter("Номер
сцены.txt", false);
    for (int k = 0; k < stroka16.Length; k++)
    {
        stroka16[k] = 0.ToString();
        f16.WriteLine(stroka16[k]);
    }
    f16.Close();
    StreamWriter f17 = new
StreamWriter("посещённые локации.txt", false);
    for (int k = 0; k < stroka17.Length; k++)
    {
        stroka17[k] = 0.ToString();
        f17.WriteLine(stroka17[k]);
    }
    f17.Close();
    StreamWriter f18 = new
StreamWriter("предметы.txt", false);
    for (int k = 0; k < stroka18.Length; k++)
    {
        stroka18[k] = 0.ToString();
        f18.WriteLine(stroka18[k]);
    }

```

```

    }
    f18.Close();
    StreamWriter f19 = new StreamWriter("панг.txt",
false);
    for (int k = 0; k < stroka19.Length; k++)
    {
        stroka19[k] = 0.ToString();
        f19.WriteLine(stroka19[k]);
    }
    f19.Close();
    StreamWriter f20 = new
StreamWriter("растения.txt", false);
    for (int k = 0; k < stroka20.Length; k++)
    {
        stroka20[k] = 0.ToString();
        f20.WriteLine(stroka20[k]);
    }
    f20.Close();

    Time.timeScale = 1f;
    if (endddd.gameObject != null)
    {
        endddd.gameObject.SetActive(false);
    }
}
}

```

Файл sborrastanii.cs:

```

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

public class sborrastanii : MonoBehaviour
{
    public Button[] bt;

    void Update()
    {
        string[] vremia = File.ReadAllLines("время
скрытия.txt");
        for (int i = 0; i < vremia.Length; i++)
        {
            DateTime d1 = DateTime.Now;
            TimeSpan raznosti = d1 -
DateTime.Parse(vremia[i]);
            if (raznosti.TotalMinutes > 15)
            {
                bt[i].gameObject.SetActive(true);
            }
        }
    }
    public void ButtonClik(Button clickedButton)
    {
        string buttonName = clickedButton.name;//
проверка какая из одинаковых кнопок нажата

```

```

        string[] stroka =
File.ReadAllLines("растения.txt");

        int x = 0;
        for (int j = 0; j < stroka.Length; j++)
        {
            if ((buttonName == bt[0 + x].name) ||
(buttonName == bt[1 + x].name))
            {
                stroka[j] = (int.Parse(stroka[j]) +
1).ToString();
            }
            x += 2;
        }
        StreamWriter f = new
StreamWriter("растения.txt", false);
        for (int i = 0; i < stroka.Length; i++)
        {
            f.WriteLine(stroka[i]);
        }
        f.Close();//счёт собранных растений
        string[] vremia = File.ReadAllLines("время
скрытия.txt");
        DateTime d = DateTime.Now;

        for (int k=0; k < vremia.Length; k++)
        {
            if (buttonName == bt[k].name)
            {
                vremia[k] = d.ToString();
                bt[k].gameObject.SetActive(false); ;
            }
        }
        StreamWriter f1 = new StreamWriter("время
скрытия.txt", false);
        for (int i = 0; i < vremia.Length; i++)
        {
            f1.WriteLine(vremia[i]);
        }
        f1.Close();
    }
}

```

Файл Spravka.cs:

```

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

public class Spravka : MonoBehaviour
{
    public string chmFilePath;

    public void OpenCHM()
    {
        Process.Start(chmFilePath);
    }
}

```

Файл vosvr\_vr.cs:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class vosvr_vr : MonoBehaviour
{
    public void Button_vosvr()
    {
        Time.timeScale = 1f;
    }
}
Файл zadanieposborules.cs:
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.IO;
public class zadanieposborules : MonoBehaviour
{
    public GameObject z1;
    public GameObject z2;
    public Text[] text;
    public Image imgObg;
    public Sprite spriteImage1;
    public Sprite spriteImage2;
    public Button karta;
    public GameObject pobeda;
    void Start()
    {
        string[] stroka = File.ReadAllLines("номер
        принятого задания.txt");
        if (int.Parse(stroka[0]) == 1)
        {
            z1.gameObject.SetActive(true);
        }
        if (int.Parse(stroka[0]) == 2)
        {
            z2.gameObject.SetActive(true);
        }
    }
    void Update()
    {
        string[] stroka = File.ReadAllLines("номер
        принятого задания.txt");
        string[] stroka1 =
        File.ReadAllLines("растения.txt");
        if (int.Parse(stroka[0]) == 1)
        {
            text[0].text = stroka1[8] + "/2";
            text[1].text = stroka1[9] + "/2";
        }
        if (int.Parse(stroka[0]) == 2)
        {
            text[2].text = stroka1[7] + "/2";
            text[3].text = stroka1[12] + "/2";
        }
        if ((int.Parse(stroka[0]) == 1) &&
        (int.Parse(stroka1[8]) >= 2) && (int.Parse(stroka1[9])
        >= 2))
        {

```

```

        stroka1[8] = (int.Parse(stroka1[8]) -
        2).ToString();
        stroka1[9] = (int.Parse(stroka1[9]) -
        2).ToString();

        StreamWriter f1 = new
        StreamWriter("растения.txt", false);
        for (int k = 0; k < stroka1.Length; k++)
        {
            f1.WriteLine(stroka1[k]);
        }
        f1.Close();

        stroka[0] = "0";
        StreamWriter f = new StreamWriter("номер
        принятого задания.txt", false);
        for (int k = 0; k < stroka.Length; k++)
        {
            f.WriteLine(stroka[k]);
        }
        f.Close();
        string[] stroka2 =
        File.ReadAllLines("манеты.txt");
        stroka2[0] = (int.Parse(stroka2[0]) +
        100).ToString();

        StreamWriter f2 = new
        StreamWriter("манеты.txt", false);
        for (int k = 0; k < stroka2.Length; k++)
        {
            f2.WriteLine(stroka2[k]);
        }
        f2.Close();

        string[] stroka3 =
        File.ReadAllLines("выполнение зд.txt");
        stroka3[0] = (int.Parse(stroka3[0]) +
        1).ToString();
        StreamWriter f3 = new
        StreamWriter("выполнение зд.txt", false);
        for (int k = 0; k < stroka3.Length; k++)
        {
            f3.WriteLine(stroka3[k]);
        }
        f3.Close();
        string[] stroka4 =
        File.ReadAllLines("панг.txt");
        stroka4[0] = (int.Parse(stroka4[0]) +
        1).ToString();

        StreamWriter f4 = new
        StreamWriter("панг.txt", false);
        for (int k = 0; k < stroka4.Length; k++)
        {
            f4.WriteLine(stroka4[k]);
        }
        f4.Close();
        pobeda.gameObject.SetActive(true);

```

```

        string[] stroka5 =
File.ReadAllLines("капра.txt");
        if (stroka5[0] == "0")
        {
            imgObg.sprite = spriteImage1;
        }
        else
        {
            imgObg.sprite = spriteImage2;
            karta.gameObject.SetActive(true);
        }
        StartCoroutine(stop());
    }
    if ((int.Parse(stroka[0]) == 2) &&
(int.Parse(stroka1[7]) >= 2) && (int.Parse(stroka1[12])
>= 2))
    {
        stroka1[7] = (int.Parse(stroka1[7]) -
2).ToString();
        stroka1[12] = (int.Parse(stroka1[12]) -
2).ToString();

        StreamWriter f1 = new
StreamWriter("пастения.txt", false);
        for (int k = 0; k < stroka1.Length; k++)
        {
            f1.WriteLine(stroka1[k]);
        }
        f1.Close();

        stroka[0] = 0.ToString();
        StreamWriter f = new StreamWriter("номер
принятого задания.txt", false);
        for (int k = 0; k < stroka.Length; k++)
        {
            f.WriteLine(stroka[k]);
        }
        f.Close();
        string[] stroka2 =
File.ReadAllLines("манеты.txt");
        stroka2[0] = (int.Parse(stroka2[0]) +
100).ToString();

        StreamWriter f2 = new
StreamWriter("манеты.txt", false);
        for (int k = 0; k < stroka2.Length; k++)
        {
            f2.WriteLine(stroka2[k]);
        }
        f2.Close();

        string[] stroka3 =
File.ReadAllLines("выполнение зд.txt");
        stroka3[2] = (int.Parse(stroka3[2]) +
1).ToString();

        StreamWriter f3 = new
StreamWriter("выполнение зд.txt", false);
        for (int k = 0; k < stroka3.Length; k++)

```

```

        {
            f3.WriteLine(stroka3[k]);
        }
        f3.Close();

        string[] stroka4 =
File.ReadAllLines("панг.txt");
        stroka4[0] = (int.Parse(stroka4[0]) +
2).ToString();

        StreamWriter f4 = new
StreamWriter("панг.txt", false);
        for (int k = 0; k < stroka4.Length; k++)
        {
            f4.WriteLine(stroka4[k]);
        }
        f4.Close();
        pobeda.gameObject.SetActive(true);

        string[] stroka5 =
File.ReadAllLines("капра.txt");
        if (stroka5[0] == "0")
        {
            imgObg.sprite = spriteImage1;
        }
        else
        {
            imgObg.sprite = spriteImage2;
            karta.gameObject.SetActive(true);
        }
        StartCoroutine(stop());
    }
}
private IEnumerator stop()
{
    yield return new WaitForSeconds(3f);
    Time.timeScale = 0f;
}
}

```

Файл zadani\_sell.cs:

```

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

public class zadani_sell : MonoBehaviour
{
    public GameObject z1;
    public GameObject z2;
    public Text[] text;

    public Image imgObg;
    public Sprite spriteImage1;
    public Sprite spriteImage2;
    public Sprite spriteImage3;

```



```

public Sprite spriteImage4;
public Button karta;

public GameObject pobeda;

public Переход perehod;

void Start()
{
    string[] stroka = File.ReadAllLines("номер
    принятого задания.txt");

    if (int.Parse(stroka[0]) == 4)
    {
        z1.gameObject.SetActive(true);
        perehod.levelToLoad = 5;
    }
    if (int.Parse(stroka[0]) == 5)
    {
        z2.gameObject.SetActive(true);
        perehod.levelToLoad = 5;
    }
}

void Update()
{
    string[] stroka = File.ReadAllLines("номер
    принятого задания.txt");

    string[] stroka1 = File.ReadAllLines("зелья.txt");

    if (int.Parse(stroka[0]) == 4)
    {
        text[0].text = stroka1[1] + "/" + 1;
    }

    if (int.Parse(stroka[0]) == 5)
    {
        text[1].text = stroka1[2] + "/" + 1;
    }

    if ((int.Parse(stroka[0]) == 4) &&
        (int.Parse(stroka1[1]) >= 1))
    {
        stroka1[1] = (int.Parse(stroka1[1]) -
        1).ToString();

        StreamWriter f1 = new
        StreamWriter("зелья.txt", false);
        for (int k = 0; k < stroka1.Length; k++)
        {
            f1.WriteLine(stroka1[k]);
        }
        f1.Close();

        stroka[0] = "0";
        StreamWriter f = new StreamWriter("номер
        принятого задания.txt", false);

```

```

        for (int k = 0; k < stroka.Length; k++)
        {
            f.WriteLine(stroka[k]);
        }
        f.Close();
        string[] stroka2 =
        File.ReadAllLines("манеты.txt");
        stroka2[0] = (int.Parse(stroka2[0]) +
        225).ToString();

        StreamWriter f2 = new
        StreamWriter("манеты.txt", false);
        for (int k = 0; k < stroka2.Length; k++)
        {
            f2.WriteLine(stroka2[k]);
        }
        f2.Close();

        string[] stroka3 =
        File.ReadAllLines("выполнение зд.txt");
        stroka3[3] = (int.Parse(stroka3[3]) +
        1).ToString();

        StreamWriter f3 = new
        StreamWriter("выполнение зд.txt", false);
        for (int k = 0; k < stroka3.Length; k++)
        {
            f3.WriteLine(stroka3[k]);
        }
        f3.Close();

        string[] stroka4 =
        File.ReadAllLines("панг.txt");
        stroka4[0] = (int.Parse(stroka4[0]) +
        2).ToString();

        StreamWriter f4 = new
        StreamWriter("панг.txt", false);
        for (int k = 0; k < stroka4.Length; k++)
        {
            f4.WriteLine(stroka4[k]);
        }
        f4.Close();

        pobeda.gameObject.SetActive(true);

        string[] stroka5 =
        File.ReadAllLines("капра.txt");
        if (stroka5[0] == "0")
        {
            imgObg.sprite = spriteImage1;
        }
        else
        {
            imgObg.sprite = spriteImage2;
            karta.gameObject.SetActive(true);
        }
    }
}

```

```

        StartCoroutine(stop());
    }

    if ((int.Parse(stroka[0]) == 5) &&
(int.Parse(stroka1[2]) >= 1) )
    {

        stroka1[2] = (int.Parse(stroka1[2]) -
1).ToString();

        StreamWriter f1 = new
StreamWriter("зелья.txt", false);
        for (int k = 0; k < stroka1.Length; k++)
        {
            f1.WriteLine(stroka1[k]);
        }
        f1.Close();

        stroka[0] = 0.ToString();
        StreamWriter f = new StreamWriter("номер
принятого задания.txt", false);
        for (int k = 0; k < stroka.Length; k++)
        {
            f.WriteLine(stroka[k]);
        }
        f.Close();
        string[] stroka2 =
File.ReadAllLines("манеты.txt");
        stroka2[0] = (int.Parse(stroka2[0]) +
200).ToString();

        StreamWriter f2 = new
StreamWriter("манеты.txt", false);
        for (int k = 0; k < stroka2.Length; k++)
        {
            f2.WriteLine(stroka2[k]);
        }
        f2.Close();

        string[] stroka3 =
File.ReadAllLines("выполнение зд.txt");
        stroka3[4] = (int.Parse(stroka3[4]) +
1).ToString();

        StreamWriter f3 = new
StreamWriter("выполнение зд.txt", false);
        for (int k = 0; k < stroka3.Length; k++)
        {
            f3.WriteLine(stroka3[k]);
        }
        f3.Close();

        string[] stroka4 =
File.ReadAllLines("панг.txt");

```

```

        stroka4[0] = (int.Parse(stroka4[0]) +
2).ToString();

        StreamWriter f4 = new
StreamWriter("панг.txt", false);
        for (int k = 0; k < stroka4.Length; k++)
        {
            f4.WriteLine(stroka4[k]);
        }
        f4.Close();
        pobeda.gameObject.SetActive(true);

        string[] stroka5 =
File.ReadAllLines("капра.txt");
        if (stroka5[0] == "0")
        {
            imgObg.sprite = spriteImage3;
        }
        else
        {
            imgObg.sprite = spriteImage4;
            karta.gameObject.SetActive(true);
        }

        StartCoroutine(stop());
    }
}

private IEnumerator stop()
{
    yield return new WaitForSeconds(3f);
    Time.timeScale = 0f;
}

public void start()
{
    Time.timeScale = 1f;
}
}

```

Файл zagruska.cs:

```

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

public class zagruska : MonoBehaviour
{

    private Animator anim;

    void Start()
    {

        StartCoroutine(gar1());
    }
}

```

```

    }
    private IEnumerator gar1()
    {
        yield return new WaitForSeconds(7.8f);

        anim = GetComponent<Animator>();
        SceneManager.LoadScene(1);
        anim.SetTrigger("fade");

    }
}

```

Файл Переход.cs:

```

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

public class Переход : MonoBehaviour
{
    private Animator anim;
    public int levelToLoad;
    private void Start()
    {
        anim = GetComponent<Animator>();
    }

    public void FadeToLevel()
    {
        anim.SetTrigger("fade");
    }

    public void OnFadeComplete()
    {
        SceneManager.LoadScene(levelToLoad);
    }
}

```

Файл враг.cs:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using static UnityEngine.RuleTile.TilingRuleOutput;

public class враг : MonoBehaviour
{
    [SerializeField] private float x;
    [SerializeField] private float x1;
    [SerializeField] private float y;
    [SerializeField] private float moveSpeed = 2f;
    [SerializeField] private Vector2 zoneSize = new
Vector2(5f, 5f);
    private UnityEngine.Transform player;
    [SerializeField] private bool playerInRange = false;
    private Vector3 initialPosition;
    private Animator animator; // Аниматор врага

```

```

    void Start()
    {
        player =
GameObject.FindWithTag("Player").transform;
        initialPosition = transform.position;
        animator = GetComponent<Animator>(); //
Получаем компонент Animator
    }

    void Update()
    {
        if (player.position.x > x && player.position.y < y
&& player.position.x < x1)
        {
            MoveTowardsPlayer();
        }
        else
        {
            ReturnToInitialPosition();
        }
    }

    void MoveTowardsPlayer()
    {
        Vector2 direction = (player.position -
transform.position).normalized;
        Vector3 targetPosition = transform.position + new
Vector3(direction.x * moveSpeed * Time.deltaTime, 0,
0);

        if (IsWithinZone(targetPosition))
        {
            transform.position = targetPosition;
            UpdateAnimationAndRotation(direction); //
Обновляем анимацию и поворот
        }
    }

    bool IsWithinZone(Vector3 position)
    {
        float halfWidth = zoneSize.x / 2;
        float halfHeight = zoneSize.y / 2;
        return position.x >= initialPosition.x - halfWidth
&& position.x <= initialPosition.x + halfWidth &&
        position.y >= initialPosition.y - halfHeight
&& position.y <= initialPosition.y + halfHeight;
    }

    void ReturnToInitialPosition()
    {
        if (Vector3.Distance(transform.position,
initialPosition) > 0.1f)
        {
            if (transform.position.x > initialPosition.x)
            {
                transform.localScale = new Vector3(1, 1, 1);
            }
            if (transform.position.x < initialPosition.x)

```

```

    {
        transform.localScale = new Vector3(-1, 1, 1);
    }

    transform.position =
    Vector3.MoveTowards(transform.position,
    initialPosition, moveSpeed * Time.deltaTime);

    if (transform.position == initialPosition)
    {
        animator.SetBool("isRunning", false); //
Остановка анимации бега
    }
}

private void UpdateAnimationAndRotation(Vector2
direction)
{
    animator.SetBool("isRunning", true); // Запуск
анимации бега

    // Поворачиваем врага в сторону движения
    if (direction.x > 0)
    {
        transform.localScale = new Vector3(-1, 1, 1); //
Поворот вправо
    }
    else if (direction.x < 0)
    {
        transform.localScale = new Vector3(1, 1, 1); //
Поворот влево
    }
}

private void OnCollisionEnter2D(Collision2D
collision)
{
    if (collision.collider.CompareTag("Player"))
    {
        hp playerController =
collision.collider.GetComponent<hp>();
        playerController.boy(); // Наносим урон
игроку

        // Получаем Rigidbody2D игрока
        Rigidbody2D playerRigidbody =
collision.collider.GetComponent<Rigidbody2D>();
        if (playerRigidbody != null)
        {
            // Вычисляем направление от врага к
игроку
            Vector2 knockbackDirection =
(collision.transform.position -
transform.position).normalized;

            // Устанавливаем компонент Y для
отбрасывания

```

```

            float knockbackX = knockbackDirection.x *
400f; // Настройте силу отталкивания по вашему
усмотрению
            float knockbackY = 400f; // Сила
отталкивания вверх

            // Применяем силу отталкивания
            playerRigidbody.AddForce(new
Vector2(knockbackX, knockbackY),
ForceMode2D.Impulse);
        }
    }
}

```

Файл hp.cs:

```

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

public class hp : MonoBehaviour
{
    public int hp;
    public Image a;
    public Image b;

    public void boy()
    {
        hp--;
        if (hp == 1)
        { a.gameObject.SetActive(false); }
        if (hp == 0)
        { b.gameObject.SetActive(false); }
    }
}

```

Файл переходизмузыки.cs:

```

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

public class переходизмузыки : MonoBehaviour
{
    private Animator anim;

    private void Start()
    {
        anim = GetComponent<Animator>();
    }

    public void FadeToLevel()
    {
        anim.SetTrigger("fade");
    }

    public void OnFadeComplete()

```

```

{

    //считывание номера сцены из файла
    StreamReader f = new StreamReader("Номер
сцены для муз.txt");
    int n = int.Parse(f.ReadToEnd());
    f.Close();
    SceneManager.LoadScene(n);
}

public void OnFadeComplete1()
{

    //считывание номера сцены из файла
    StreamReader f = new StreamReader("Номер
сцены.txt");
    int n = int.Parse(f.ReadToEnd());
    f.Close();
    SceneManager.LoadScene(n);
}
}

```

Файл переходизторговли.cs:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using System.IO;
public class переходизторговли : MonoBehaviour
{
    private Animator anim;

    private void Start()
    {
        anim = GetComponent<Animator>();
    }

    public void FadeToLevel()
    {
        anim.SetTrigger("fade");
    }

    public void OnFadeComplete()
    {
        //считывание номера сцены из файла
        StreamReader f = new StreamReader("Номер
сцены.txt");
        int n = int.Parse(f.ReadToEnd());
        f.Close();
        SceneManager.LoadScene(n);
    }
}

```

Файл переходсзаписью.cs:

```

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

```

```

using System.IO;

```

```

public class переходсзаписью : MonoBehaviour
{

```

```

    private Animator anim;
    public int levelToLoad;
    private void Start()
    {
        anim = GetComponent<Animator>();
    }

```

```

    public void FadeToLevel()
    {
        anim.SetTrigger("fade");
    }

```

```

    //получение индекса
    Scene currentScene
    =SceneManager.GetActiveScene();
    int sceneIndex =currentScene.buildIndex;
    StreamWriter f = new StreamWriter("Номер
сцены.txt",false);
    f.WriteLine(sceneIndex.ToString() );
    f.Close();
}

```

```

    public void FadeToLevel1()
    {
        anim.SetTrigger("fade");
    }

```

```

    //получение индекса
    Scene currentScene =
    SceneManager.GetActiveScene();
    int sceneIndex = currentScene.buildIndex;
    StreamWriter f = new StreamWriter("Номер
сцены для муз.txt", false);
    f.WriteLine(sceneIndex.ToString());
    f.Close();
}

```

```

    public void OnFadeComplete()
    {

        SceneManager.LoadScene(levelToLoad);
    }
}

```

Файл расшифровка.cs:

```

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

```

```

public class расшифровка : MonoBehaviour
{

```

```

    public Text text1;
    public Text text2;
    public Image imgObj;

```

```

public Sprite spriteImage1;
public Sprite spriteImage2;

public Image imgObg2;
public Sprite spriteImage21;
public Sprite spriteImage22;
public Button karta;

public GameObject pobeda;
public GameObject porajenie;
public void Gotovo()
{
    if ((text1.text == "Великие начинания даже не
надо обдумывать, надо взяться за дело, иначе,
заметив трудность отступишь.") && (text2.text == "-
Гай Юлий Цезарь"))
    {
        string[] stroka1 =
File.ReadAllLines("манеты.txt");
        stroka1[0] = (int.Parse(stroka1[0]) +
100).ToString();

        StreamWriter f1 = new
StreamWriter("манеты.txt", false);
        for (int k = 0; k < stroka1.Length; k++)
        {
            f1.WriteLine(stroka1[k]);
        }
        f1.Close();

        string[] stroka2 =
File.ReadAllLines("выполнение зд.txt");
        stroka2[1] = (int.Parse(stroka2[1]) +
1).ToString();

        StreamWriter f2 = new
StreamWriter("выполнение зд.txt", false);
        for (int k = 0; k < stroka2.Length; k++)
        {
            f2.WriteLine(stroka2[k]);
        }
        f2.Close();

        string[] stroka3 =
File.ReadAllLines("панр.txt");
        stroka3[0] = (int.Parse(stroka3[0]) +
1).ToString();

        StreamWriter f3 = new
StreamWriter("панр.txt", false);
        for (int k = 0; k < stroka3.Length; k++)
        {
            f3.WriteLine(stroka3[k]);
        }
        f3.Close();
        pobeda.SetActive(true);
        string[] stroka =
File.ReadAllLines("карта.txt");

```

```

        if (stroka[0] == "0")
        {
            imgObg.sprite = spriteImage1;
        }
        else
        {
            imgObg.sprite = spriteImage2;
            karta.gameObject.SetActive(true);
        }
    }

    else {
        porajenie.SetActive(true);

        string[] stroka =
File.ReadAllLines("карта.txt");
        if (stroka[0] == "0")
        {
            imgObg2.sprite = spriteImage21;
        }
        else
        {
            imgObg2.sprite = spriteImage22;
            karta.gameObject.SetActive(true);
        }
    }

    Time.timeScale = 0f;
}

public void zanovo()
{
    porajenie.SetActive(false);
    karta.gameObject.SetActive(false);
    text1.text = "";
    text2.text = "";
    Time.timeScale = 1f;
}
}

```

Файл Bullet\_enemy.cs:

```

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

public class Bullet : MonoBehaviour
{
    public float speed;
    public float lifetime;
    public float lifetime1;
    public float distance;
    public int damage;
    public LayerMask whatIsSolid; // Лейер для
"огонёк"
    public LayerMask whatIsWrag; // Лейер для
"кислота_паук"

```

```

private Vector2 direction; // Направление движения
пули

public GameObject bulletEffect;

private void Start()
{
    // Уничтожаем пулю через заданное время
    Destroy(gameObject, lifetime1);
}

public void SetDirection(Vector2 dir)
{
    direction = dir; // Устанавливаем направление
}

private void Update()
{
    RaycastHit2D hitInfo =
Physics2D.Raycast(transform.position, direction,
distance);

    if (hitInfo.collider != null)
    {
        string objectName = gameObject.name;

        // Проверяем, если это "огонёк(Clone)"
        if (objectName == "огонёк(Clone)" && ((1 <<
hitInfo.collider.gameObject.layer) & whatIsSolid) != 0)
        {
            if (hitInfo.collider.CompareTag("Enemy"))
            {
                hitInfo.collider.GetComponent<Enemy>().TakeDamag
e(damage);
            }

            Instantiate(bulletEffect, transform.position,
Quaternion.identity);
            Destroy(gameObject);
        }

        // Проверяем, если это
"кислота_паук(Clone)"
        if ((objectName == "кислота_паук(Clone)"||
objectName == "огонь_сал(Clone)") && ((1 <<
hitInfo.collider.gameObject.layer) & whatIsWrag) != 0)
        {
            if (hitInfo.collider.CompareTag("Player"))
            {
                hitInfo.collider.GetComponent<hp>().boy();
            }

            Instantiate(bulletEffect, transform.position,
Quaternion.identity);
            Destroy(gameObject);
        }
    }
}

```

```

// Двигаем пулю в заданном направлении
transform.Translate(direction * speed *
Time.deltaTime);
}
}

```

Файл casanije\_wrag.cs:

```

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

```

```

public class casanije_wrag : MonoBehaviour
{
    public int x = 0;
    private void OnCollisionEnter2D(Collision2D
collision)
    {
        if (collision.collider.CompareTag("Player"))
        {
            hp playerController =
collision.collider.GetComponent<hp>();
            playerController.boy(); // Наносим урон
игроку

            if (x == 0) {
                // Получаем Rigidbody2D игрока
                Rigidbody2D playerRigidbody =
collision.collider.GetComponent<Rigidbody2D>();
                if (playerRigidbody != null)
                {
                    // Вычисляем направление от врага к
игроку
                    Vector2 knockbackDirection =
(collision.transform.position -
transform.position).normalized;
                    float knockbackX = knockbackDirection.x *
400f;
                    float knockbackY = 400f;
                    playerRigidbody.AddForce(new
Vector2(knockbackX, knockbackY),
ForceMode2D.Impulse);
                }
            }
        }
    }
}

```

Файл result\_z7.cs:

```

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

```

```

public class result_z7 : MonoBehaviour
{
    public Button gild;
}

```

```

public Button kart;

public GameObject pobeda;
public GameObject belca;

void Start()
{
    string[] stroka_v =
File.ReadAllLines("враги.txt");
    if (stroka_v[2] == 1.ToString())
    {
        string[] stroka_z = File.ReadAllLines("номер
принятого задания.txt");
        stroka_z[0] = 7.ToString();
        StreamWriter fz = new StreamWriter("номер
принятого задания.txt", false);
        for (int k = 0; k < stroka_z.Length; k++)
        {
            fz.WriteLine(stroka_z[k]);
        }
        fz.Close();
    }
    else {
        Destroy(belca.gameObject);
    }

    string[] stroka = File.ReadAllLines("номер
принятого задания.txt");
    if( stroka[0] == 7.ToString())
    {
        gild.gameObject.SetActive(true);
    }

}

void Update()
{
    string[] stroka_z = File.ReadAllLines("номер
принятого задания.txt");
    string[] stroka_v =
File.ReadAllLines("враги.txt");
    if ((stroka_v[2] == 0.ToString())&&(stroka_z[0]
== 7.ToString()))
    {
        pobeda.gameObject.SetActive(true);

        string[] stroka5 =
File.ReadAllLines("карта.txt");
        if (stroka5[0] == "1")
        {
            kart.gameObject.SetActive(true);
        }

        string[] stroka = File.ReadAllLines("номер
принятого задания.txt");
        stroka[0] = 0.ToString();

```

```

        StreamWriter f = new StreamWriter("номер
принятого задания.txt", false);
        for (int k = 0; k < stroka.Length; k++)
        {
            f.WriteLine(stroka[k]);
        }
        f.Close();

        string[] stroka2 =
File.ReadAllLines("манеты.txt");
        stroka2[0] = (int.Parse(stroka2[0]) +
400).ToString();

        StreamWriter f2 = new
StreamWriter("манеты.txt", false);
        for (int k = 0; k < stroka2.Length; k++)
        {
            f2.WriteLine(stroka2[k]);
        }
        f2.Close();

        string[] stroka3 =
File.ReadAllLines("выполнение зд.txt");
        stroka3[6] = 1.ToString();

        StreamWriter f3 = new
StreamWriter("выполнение зд.txt", false);
        for (int k = 0; k < stroka3.Length; k++)
        {
            f3.WriteLine(stroka3[k]);
        }
        f3.Close();

        string[] stroka4 =
File.ReadAllLines("панг.txt");
        stroka4[0] = (int.Parse(stroka4[0]) +
3).ToString();

        StreamWriter f4 = new
StreamWriter("панг.txt", false);
        for (int k = 0; k < stroka4.Length; k++)
        {
            f4.WriteLine(stroka4[k]);
        }
        f4.Close();

        StartCoroutine(stop());
    }

}

private IEnumerator stop()
{
    yield return new WaitForSeconds(3f);
    Time.timeScale = 0f;
}
}

```



```

Файл result_z8.cs:
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.IO;
using System;
using static UnityEngine.Rendering.DebugUI.Table;
using UnityEngine.SceneManagement;

```

```

public class result_z8 : MonoBehaviour

```

```

{
    public Text text_jz;
    public GameObject licor;
    public Text text_licor;

    public Button gild;
    public Button kart_pob;
    public Button kart_por;

    public GameObject pobeda;
    public GameObject porajenie;
    public GameObject pauc;
    public Image serdca;

    bool par = false;
    public Image a;
    public Image b;
    void Start()
    {
        string[] stroka_v =
        File.ReadAllLines("враги.txt");
        if (stroka_v[0] == 1.ToString())
        {
            string[] stroka_z = File.ReadAllLines("номер
            принятого задания.txt");
            stroka_z[0] = 8.ToString();
            StreamWriter fz = new StreamWriter("номер
            принятого задания.txt", false);
            for (int k = 0; k < stroka_z.Length; k++)
            {
                fz.WriteLine(stroka_z[k]);
            }
            fz.Close();
        }
        else
        {
            Destroy(pauc.gameObject);
            serdca.gameObject.SetActive(false);
        }
        string[] stroka = File.ReadAllLines("номер
        принятого задания.txt");
        if (stroka[0] == 8.ToString())
        {
            gild.gameObject.SetActive(true);
        }
    }
    void Update()

```

```

{
    if (FindObjectOfType<Enemy>() != null)
    {
        text_jz.text =
        (FindObjectOfType<Enemy>().health).ToString();
    }

    string[] vremia = File.ReadAllLines("z8-
    9_время.txt");
    DateTime d1 = DateTime.Now;
    TimeSpan raznosti = d1 -
    DateTime.Parse(vremia[0]);
    if (raznosti.TotalMinutes > 30)
    {
        licor.gameObject.SetActive(true);
        text_licor.gameObject.SetActive(false);
    }
    else
    {
        licor.gameObject.SetActive(false);
        text_licor.gameObject.SetActive(true);
        DateTime d2 = DateTime.Now; ;
        text_licor.text = (TimeSpan.FromMinutes(30) -
        (d1 -
        DateTime.Parse(vremia[0]))).ToString(@"hh\:mm\:ss")
        ;
    }

    if ((FindObjectOfType<hp>().hp <= 0) && (par
    == false))
    {
        porajenie.gameObject.SetActive(true);

        string[] stroka5 =
        File.ReadAllLines("капра.txt");
        if (stroka5[0] == "1")
        {
            kart_por.gameObject.SetActive(true);
        }

        par = true;
        StartCoroutine(stop());
    }

    string[] stroka_z = File.ReadAllLines("номер
    принятого задания.txt");
    string[] stroka_v = File.ReadAllLines("z8-9.txt");
    if ((int.Parse(stroka_v[0]) > 0) && (stroka_z[0] ==
    8.ToString()))
    {
        pobeda.gameObject.SetActive(true);
    }

```

```

        string[] stroka5 =
File.ReadAllLines("карта.txt");
        if (stroka5[0] == "1")
        {
            kart_pob.gameObject.SetActive(true);
        }

        string[] stroka = File.ReadAllLines("номер
принятого задания.txt");
        stroka[0] = 0.ToString();
        StreamWriter f = new StreamWriter("номер
принятого задания.txt", false);
        for (int k = 0; k < stroka.Length; k++)
        {
            f.WriteLine(stroka[k]);
        }
        f.Close();

        string[] stroka2 =
File.ReadAllLines("манеты.txt");
        stroka2[0] = (int.Parse(stroka2[0]) +
500).ToString();

        StreamWriter f2 = new
StreamWriter("манеты.txt", false);
        for (int k = 0; k < stroka2.Length; k++)
        {
            f2.WriteLine(stroka2[k]);
        }
        f2.Close();

        string[] stroka3 =
File.ReadAllLines("выполнение зд.txt");
        stroka3[7] = 1.ToString();

        StreamWriter f3 = new
StreamWriter("выполнение зд.txt", false);
        for (int k = 0; k < stroka3.Length; k++)
        {
            f3.WriteLine(stroka3[k]);
        }
        f3.Close();

        string[] stroka4 =
File.ReadAllLines("панг.txt");
        stroka4[0] = (int.Parse(stroka4[0]) +
3).ToString();

        StreamWriter f4 = new
StreamWriter("панг.txt", false);
        for (int k = 0; k < stroka4.Length; k++)
        {
            f4.WriteLine(stroka4[k]);
        }
        f4.Close();

        string[] stroka_lik = File.ReadAllLines("z8-
9.txt");
        stroka_lik[0]=0.ToString();

```

```

        StreamWriter f_lik = new StreamWriter("z8-
9.txt", false);
        for (int k = 0; k < stroka_lik.Length; k++)
        {
            f_lik.WriteLine(stroka_lik[k]);
        }
        f_lik.Close();

        StartCoroutine(stop());
    }
}
private IEnumerator stop()
{
    yield return new WaitForSeconds(3f);
    Time.timeScale = 0f;
}

public void zanovo()
{
    par = false;
    porajenie.SetActive(false);
    FindObjectOfType<hp>().hp = 2;
    FindObjectOfType<Enemy>().health = 10;
    { a.gameObject.SetActive(true); }
    { b.gameObject.SetActive(true); }
    Time.timeScale = 1f;
    SceneManager.LoadScene(19);
}
}

```

```

Файл result_z9.cs:
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.IO;
using System;
using static UnityEngine.Rendering.DebugUI.Table;
using Unity.VisualScripting;
using UnityEngine.SceneManagement;

public class result_z9 : MonoBehaviour
{
    public Text text_jz;
    public GameObject licor;
    public Text text_licor;

    public Button gild;
    public Button kart_pob;
    public Button kart_por;

    public GameObject pobeda;
    public GameObject porajenie;
    public GameObject pauc;
    public Image serdca;

```

```

bool par = false;
public Image a;
public Image b;

void Start()
{
    string[] stroka_v =
File.ReadAllLines("враги.txt");
    if (stroka_v[1] == 1.ToString())
    {
        string[] stroka_z = File.ReadAllLines("номер
принятого задания.txt");
        stroka_z[0] = 9.ToString();
        StreamWriter fz = new StreamWriter("номер
принятого задания.txt", false);
        for (int k = 0; k < stroka_z.Length; k++)
        {
            fz.WriteLine(stroka_z[k]);
        }
        fz.Close();
    }
    else
    {
        Destroy(pauc.gameObject);
        serdca.gameObject.SetActive(false);
    }
    string[] stroka = File.ReadAllLines("номер
принятого задания.txt");
    if (stroka[0] == 9.ToString())
    {
        gild.gameObject.SetActive(true);
    }
}

void Update()
{
    if (FindObjectOfType<Enemy>() != null)
    {
        text_jz.text =
(FindObjectOfType<Enemy>().health).ToString();
    }

    string[] vremia = File.ReadAllLines("z8-
9_время.txt");
    DateTime d1 = DateTime.Now;
    TimeSpan raznosti = d1 -
DateTime.Parse(vremia[1]);
    if (raznosti.TotalMinutes > 45)
    {
        licor.gameObject.SetActive(true);
        text_licor.gameObject.SetActive(false);
    }
    else
    {
        licor.gameObject.SetActive(false);
        text_licor.gameObject.SetActive(true);
        DateTime d2 = DateTime.Now; ;

```

```

        text_licor.text = (TimeSpan.FromMinutes(45) -
(d1 -
DateTime.Parse(vremia[1]))).ToString(@"hh\:mm\:ss")
;
    }

    if ((FindObjectOfType<hp>().hp <= 0) && (par
== false))
    {
        porajenie.gameObject.SetActive(true);

        string[] stroka5 =
File.ReadAllLines("капра.txt");
        if (stroka5[0] == "1")
        {
            kart_por.gameObject.SetActive(true);
        }

        par = true;
        StartCoroutine(stop());
    }
    string[] stroka_z = File.ReadAllLines("номер
принятого задания.txt");
    string[] stroka_v = File.ReadAllLines("z8-9.txt");
    if ((int.Parse(stroka_v[1]) > 0) && (stroka_z[0]
== 9.ToString()))
    {
        pobeda.gameObject.SetActive(true);

        string[] stroka5 =
File.ReadAllLines("капра.txt");
        if (stroka5[0] == "1")
        {
            kart_pob.gameObject.SetActive(true);
        }
        string[] stroka = File.ReadAllLines("номер
принятого задания.txt");
        stroka[0] = 0.ToString();
        StreamWriter f = new StreamWriter("номер
принятого задания.txt", false);
        for (int k = 0; k < stroka.Length; k++)
        {
            f.WriteLine(stroka[k]);
        }
        f.Close();

        string[] stroka2 =
File.ReadAllLines("манеты.txt");
        stroka2[0] = (int.Parse(stroka2[0]) +
600).ToString();

        StreamWriter f2 = new
StreamWriter("манеты.txt", false);
        for (int k = 0; k < stroka2.Length; k++)
        {
            f2.WriteLine(stroka2[k]);
        }
        f2.Close();

```

```

        string[] stroka3 =
File.ReadAllLines("выполнение зд.txt");
        stroka3[8] = 1.ToString();

        StreamWriter f3 = new
StreamWriter("выполнение зд.txt", false);
        for (int k = 0; k < stroka3.Length; k++)
        {
            f3.WriteLine(stroka3[k]);
        }
        f3.Close();

        string[] stroka4 =
File.ReadAllLines("панг.txt");
        stroka4[0] = (int.Parse(stroka4[0]) +
4).ToString();//

        StreamWriter f4 = new
StreamWriter("панг.txt", false);
        for (int k = 0; k < stroka4.Length; k++)
        {
            f4.WriteLine(stroka4[k]);
        }
        f4.Close();

        string[] stroka_lik = File.ReadAllLines("z8-
9.txt");
        stroka_lik[1] = 0.ToString();
        StreamWriter f_lik = new StreamWriter("z8-
9.txt", false);
        for (int k = 0; k < stroka_lik.Length; k++)
        {
            f_lik.WriteLine(stroka_lik[k]);
        }
        f_lik.Close();

        StartCoroutine(stop());
    }
    private IEnumerator stop()
    {
        yield return new WaitForSeconds(3f);
        Time.timeScale = 0f;
    }

    public void zanovo()
    {
        par = false;
        porajenie.SetActive(false);
        FindObjectOfType<hp>().hp = 2;
        FindObjectOfType<Enemy>().health = 15;
        { a.gameObject.SetActive(true); }
        { b.gameObject.SetActive(true); }
        Time.timeScale = 1f;

        SceneManager.LoadScene(20);
    }

```

```

    }

    Файл sbor_predmeta.cs:
    using System;
    using System.Collections;
    using System.Collections.Generic;
    using System.IO;
    using UnityEditor;
    using UnityEngine;

    public class sbor_predmeta : MonoBehaviour
    {
        private void OnTriggerEnter2D(Collider2D other)
        {
            if (other.CompareTag("Player"))
            {

                string[] stroka = File.ReadAllLines("z8-9.txt");

                string objectName = gameObject.name;

                if (objectName == "Ликорис") { stroka[0] =
(int.Parse(stroka[0])+1).ToString(); }
                else if (objectName == "Огненный кристалл")
                { stroka[1] = (int.Parse(stroka[1]) + 1).ToString(); }

                StreamWriter f = new StreamWriter("z8-9.txt",
false);
                for (int k = 0; k < stroka.Length; k++)
                {
                    f.WriteLine(stroka[k]);
                }
                f.Close();
                string[] stroka1 = File.ReadAllLines("z8-
9_время.txt");
                DateTime d = DateTime.Now;

                if (objectName == "Ликорис") { stroka1[0]
=d.ToString(); }
                else if (objectName == "Огненный кристалл")
                { stroka1[1] = d.ToString(); }

                StreamWriter f1 = new StreamWriter("z8-
9_время.txt", false);
                for (int k = 0; k < stroka1.Length; k++)
                {
                    f1.WriteLine(stroka1[k]);
                }
                f1.Close();

                gameObject.SetActive(false);
            }
        }
    }

```