

## Дополнительные системы

```
[RequireComponent(typeof(AudioSource))]  
public class ComputerAudio : MonoBehaviour  
{  
    public AudioClip powerOnSound;  
    public AudioClip powerOffSound;  
    public AudioClip keyPressSound;  
    public AudioClip sendSound;  
    public AudioClip receiveSound;  
    public AudioClip errorSound;  
  
    private AudioSource audioSource;  
  
    void Awake()  
    {  
        audioSource = GetComponent<AudioSource>();  
        audioSource.spatialBlend = 0.8f; // 3D звук  
    }  
  
    public void PlaySound(AudioClip clip)  
    {  
        if (clip != null)  
            audioSource.PlayOneShot(clip);  
    }  
}
```

# Программирование компьютера

```

using UnityEngine;
using UnityEngine.UI;

public class ComputerTerminal : MonoBehaviour
{
    public GameObject computerUI; // Ссылка на Canvas
    public InputField inputField; // Поле ввода
    public Text outputText;       // Вывод сообщений
    public Button sendButton;     // Кнопка отправки

    private bool isActive = false;

    void Start()
    {
        computerUI.SetActive(false);
        sendButton.onClick.AddListener(SendData);
    }

    public void ToggleComputer()
    {
        isActive = !isActive;
        computerUI.SetActive(isActive);

        if (isActive)
        {
            Cursor.lockState = CursorLockMode.None; // Разблокируем курсор
            inputField.Select(); // Фокус на поле ввода
        }
        else
        {
            Cursor.lockState = CursorLockMode.Locked; // Блокируем курсор
        }
    }

    private void SendData()
    {
        string message = inputField.text;
        if (!string.IsNullOrEmpty(message))
        {
            outputText.text += "> " + message + "\n";
            inputField.text = ""; // Очищаем поле ввода
            SimulateNetworkResponse(message); // Имитация ответа
        }
    }

    private void SimulateNetworkResponse(string message)
    {
        // Имитация задержки сети
        Invoke("ShowResponse", 1.5f);
    }

    private void ShowResponse()
    {
        outputText.text += "< Ответ сервера: Данные получены >\n";
    }
}

```

## Взаимодействие игрока с компьютером

```
using UnityEngine;

public class PlayerInteraction : MonoBehaviour
{
    public float interactionDistance = 3f;
    public KeyCode interactKey = KeyCode.E;
    public ComputerTerminal computer;

    void Update()
    {
        if (Input.GetKeyDown(interactKey))
        {
            // Проверяем дистанцию до компьютера
            float distance = Vector3.Distance(transform.position, computer.transform.position);

            if (distance <= interactionDistance)
            {
                computer.ToggleComputer();
            }
        }
    }
}
```

## Имитация сети

```

public class NetworkSimulator : MonoBehaviour
{
    public float minDelay = 0.5f;
    public float maxDelay = 2f;
    public float packetLossChance = 0.1f; // 10% потерь

    public void SendData(string data, System.Action<string> callback)
    {
        StartCoroutine(SendDataCoroutine(data, callback));
    }

    private IEnumerator SendDataCoroutine(string data, System.Action<string> callback)
    {
        float delay = Random.Range(minDelay, maxDelay);
        yield return new WaitForSeconds(delay);

        if (Random.Range(0f, 1f) > packetLossChance)
        {
            callback("Ответ сервера: " + data);
        }
        else
        {
            callback("Ошибка: Пакет потерян");
        }
    }
}

```

## скрипта для UI

```

using UnityEngine;
using UnityEngine.UI;

public class UIManager : MonoBehaviour
{
    public InputField inputField;
    public Text outputText;

    public void OnButtonClick()
    {
        outputText.text = inputField.text;
    }
}

```