**Name: Sofiya Sherif**

**Script**

**Pistol**

using UnityEngine;

// I wrote this part of the code and modified the values

public class Handgun : MonoBehaviour

{

    [Header("Ammo Settings")]

    public int magCapacity = 10;       // Max bullets in a single magazine

    public int reserveCapacity = 30;  // Max bullets in reserve

    public float fireCooldown = 0.5f; // Time between shots

    public float reloadTime = 0.5f;   // Time taken to reload

    private float actionCooldown = 0.5f; // Cooldown between switching actions

    public float maxRange = 100f;     // Effective shooting range

    [Header("Effects")]

    public ParticleSystem hitEffect;   // Effect displayed on bullet impact

    public ParticleSystem muzzleEffect;

    public GameObject muzzleLight;

    [Header("Cartridge Settings")]

    public Transform ejectionPoint;   // Where the cartridge is ejected

    public GameObject cartridgePrefab;

    public float ejectionForce = 5f;

    [Header("Gun Stats")]

    public Animator gunAnimator;

    public AudioSource fireSound;

    public int damage = 10;  // Damage dealt per shot

    public bool isReadyToShoot = true;

    private bool isReloading = false;

    private int bulletsInMag;

    private int bulletsInReserve;

    private float fireCooldownTimer;

    void Start()

    {

        bulletsInMag = magCapacity;

        bulletsInReserve = reserveCapacity;

        isReadyToShoot = true;

        muzzleLight.SetActive(false);

    }

    void Update()

    {

        // Ammo management

        bulletsInMag = Mathf.Clamp(bulletsInMag, 0, magCapacity);

        bulletsInReserve = Mathf.Clamp(bulletsInReserve, 0, reserveCapacity);

        // Shooting input

        if (Input.GetButtonDown("Fire1") && isReadyToShoot && !isReloading)

        {

            actionCooldown = fireCooldown;

            Fire();

        }

        // Reload input

        if (Input.GetKeyDown(KeyCode.R))

        {

            actionCooldown = reloadTime;

            ReloadWeapon();

        }

        // Update cooldown timer

        if (fireCooldownTimer > 0)

        {

            fireCooldownTimer -= Time.deltaTime;

        }

    }

// I added the fire method for the bullets

    void Fire()

    {

        if (bulletsInMag > 0 && fireCooldownTimer <= 0)

        {

            isReadyToShoot = false;

            fireSound.Play();

            muzzleEffect.Play();

            muzzleLight.SetActive(true);

            gunAnimator.SetBool("shoot", true);

            // Handle shooting logic

            RaycastHit hit;

            if (Physics.Raycast(Camera.main.transform.position, Camera.main.transform.forward, out hit, maxRange))

            {

                if (hit.collider.CompareTag("Enemy"))

                {

                    var enemyHealth = hit.collider.GetComponent<EnemyHealth>();

                    enemyHealth?.TakeDamage(damage);

                }

                Instantiate(hitEffect, hit.point, Quaternion.LookRotation(hit.normal));

            }

            // Eject cartridge

            GameObject cartridge = Instantiate(cartridgePrefab, ejectionPoint.position, ejectionPoint.rotation);

            cartridge.GetComponent<Rigidbody>().AddForce(ejectionPoint.right \* ejectionForce, ForceMode.Impulse);

            StartCoroutine(ResetEffects());

            StartCoroutine(AllowAction());

            bulletsInMag--;

            fireCooldownTimer = fireCooldown;

        }

        else

        {

            Debug.Log("Out of ammo!");

        }

    }

// I wrote this part of the code

    void ReloadWeapon()

    {

        if (isReloading || bulletsInReserve <= 0)

            return;

        int bulletsNeeded = magCapacity - bulletsInMag;

        int bulletsReloaded = Mathf.Min(bulletsNeeded, bulletsInReserve);

        bulletsInMag += bulletsReloaded;

        bulletsInReserve -= bulletsReloaded;

        gunAnimator.SetBool("reload", true);

        StartCoroutine(ResetEffects());

        StartCoroutine(ReloadCooldown());

    }

    IEnumerator ReloadCooldown()

    {

        isReloading = true;

        isReadyToShoot = false;

        yield return new WaitForSeconds(reloadTime);

        isReloading = false;

        isReadyToShoot = true;

    }

    IEnumerator ResetEffects()

    {

        yield return new WaitForSeconds(0.1f);

        gunAnimator.SetBool("shoot", false);

        gunAnimator.SetBool("reload", false);

        muzzleLight.SetActive(false);

    }

    IEnumerator AllowAction()

    {

        yield return new WaitForSeconds(fireCooldown);

        isReadyToShoot = true;

    }

}

**Cursor Controller**

using UnityEngine;

public class CursorManager : MonoBehaviour

{

    private void Awake()

    {

        // Ensure the cursor is visible and not restricted

        Cursor.visible = true;

        Cursor.lockState = CursorLockMode.None;

    }

}

**Death Main**

using UnityEngine;

using UnityEngine.SceneManagement;

// I wrote this part of the code to end the game

public class MenuController : MonoBehaviour

{

    // Method to load the game scene

    public void LoadGameScene()

    {

        SceneManager.LoadScene("Game");

    }

    // Method to exit the application

    public void ExitGame()

    {

        Application.Quit();

    }

}