using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class MouseLook : MonoBehaviour

{

    //the position of the x and y

    float mouseX = 0f;

    float mouseY = 0f;

    public float mouseSensetivity = 400f;

    public Transform playerBody;    //the camera

    float xRotation = 0f;

    // Start is called before the first frame update

    void Start()

    {

        Cursor.lockState = CursorLockMode.Locked;

        Cursor.visible = false;

    }

    // Update is called once per frame

    void Update()

    {

        mouseMovement();

        rotationCam();

    }

    void mouseMovement(){

        //Gets the mouse inputs

        //Time.deltaTime is independednt of framrate so the mosue movement is conbsitant no amtter how amny Frames are being rendered

        mouseX = Input.GetAxis("Mouse X") \* mouseSensetivity \* Time.deltaTime;

        mouseY = Input.GetAxis("Mouse Y") \* mouseSensetivity \* Time.deltaTime;

    }

    void rotationCam(){

        //x rotation of the player object

        playerBody.Rotate(Vector3.up \* mouseX);

        //y rotation of the camera object and clamping the xRotation

        xRotation -= mouseY;

        xRotation = Mathf.Clamp(xRotation, -90f, 90f);

        transform.localRotation = Quaternion.Euler(xRotation, 0f, 0f);

    }

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class PlayerMovement : MonoBehaviour

{

    public CharacterController controller;  //The ingame player

    public float speed = 10f;

    public float x = 0f;

    public float z = 0f;

    //ground cehck and distance check

    public Transform groundCheck;

    public float groundDistance = 0.4f;

    public LayerMask groundMask;

    bool isGrounded;

    Vector3 velocity;   //speed

    public float gravity = -20f;    //gravity

    //jumping

    public float jumpHeight = 2f;

    // Update is called once per frame

    void Update()

    {

        //keep this order

        playerInput();

        checkGround();

        jump();

        direction();

    }

    void playerInput(){

        //player input

        x = Input.GetAxis("Horizontal");    //mouse left right

        z = Input.GetAxis("Vertical");      //mouse up down

    }

    void checkGround(){

        //checks ground

        isGrounded = Physics.CheckSphere(groundCheck.position, groundDistance, groundMask);

        if(isGrounded && velocity.y < 0){

            velocity.y = -2f;   //to make sure the velocity doesnt increase for ever

        }

    }

    void direction(){

        //direction

        Vector3 move = transform.right \* x + transform.forward \* z;

        //moves the player by the amount velocity set

        controller.Move(move \* speed \* Time.deltaTime);

        velocity.y += gravity \* Time.deltaTime;

        controller.Move(velocity \* Time.deltaTime);

    }

    void jump(){

        //jumping

        if(Input.GetButtonDown("Jump") && isGrounded)

        {

            velocity.y = Mathf.Sqrt(jumpHeight \* -2 \* gravity); //sets the hight of the jump

        }

    }

}

public class LockMouse : MonoBehaviour

{

    // Start is called before the first frame update

    void Start()

    {

        //this locks the mosue to the center of the screen and also makes the mouse invisible

        Cursor.lockState = CursorLockMode.Locked;

        Cursor.visible = false;

    }

}

public class UnlockMouse : MonoBehaviour

{

    // Start is called before the first frame update

    void Start()

    {

        //this unlocks the mouse and makes the mouse visible

        Cursor.lockState = CursorLockMode.None;

        Cursor.visible = true;

    }

}

public class FlashlightOnOff : MonoBehaviour

{

    public bool isOn = false;

    public GameObject lightSource;  //the flash lgiht

    public bool failSafe = false;   //For delay

    // Update is called once per frame

    void Update()

    {

        toggleFlashlight();

    }

    public void toggleFlashlight()

    {

        //This if checks to see if the F key is pressed

        if(Input.GetButtonDown("FKey")){

            //this if checks to see if the flashlight is on or off and to see if the faile safe is on

            if(isOn == false && failSafe == false){

                failSafe = true;

                lightSource.SetActive(true);    //Enables the flashlight

                isOn = true;

                StartCoroutine(FailSafe()); //calls the delay

            }

            if(isOn == true && failSafe == false){

                failSafe = true;

                lightSource.SetActive(false);   //turns of the flashlight

                isOn = false;

                StartCoroutine(FailSafe());     //calls the delay

            }

        }

    }

    IEnumerator FailSafe(){

        yield return new WaitForSeconds(0.05f); //adds a delay of 0.05 seconds

        failSafe = false;   //sets failsafe to false

    }

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;   //Calling UI Libraries

using TMPro;

public class Interact : MonoBehaviour

{

    public GameObject InteractUI;   //the sphere around the alter

    public GameObject ThingToPopUp; //the E to pop up

    public bool failSafe = false;

    public bool isOn = false;

    //Trigger to see if player is in the sphere of IteractUI

    public void OnTriggerEnter(Collider other) {

        InteractUI.SetActive(true); //shows the UI Element E

    }

    //Trigger to see if player is in the sphere of IteractUI

    public void OnTriggerExit(Collider other) {

        InteractUI.SetActive(false);    //Hides the UI element E

    }

    public void Update() {

        activateUI();   //Calls activateUI

    }

    public void activateUI()

    {

        //This if checks to see if the E key is pressed

        if(Input.GetKeyDown(KeyCode.E) && InteractUI.activeInHierarchy) {

            if(isOn == false && failSafe == false) {

                failSafe = true;

                InteractUI.SetActive(false);    //sets the state of InteractiveUI

                ThingToPopUp.SetActive(true);   //sets the state of ThingToPopUp

                isOn = true;

                StartCoroutine(FailSafe()); //calls the delay

            }

            if(isOn == true && failSafe == false) {

                failSafe = true;

                InteractUI.SetActive(true);     //sets the state of InteractiveUI

                ThingToPopUp.SetActive(false);  //sets the state of ThingToPopUp

                isOn = false;

                StartCoroutine(FailSafe()); //calls the delay

            }

        }

    }

    IEnumerator FailSafe() {

        yield return new WaitForSeconds(0.05f); //adds a delay of 0.05 seconds

        failSafe = false;   //sets failsafe to false

    }

}

public class ZMenu : MonoBehaviour

{

    // Update is called once per frame

    void Update()

    {

        //This if checks to see if the Z key is pressed

        if(Input.GetKeyDown(KeyCode.Z)) {

            SceneManager.LoadSceneAsync("Menu");    //Loads the main menu scene

        }

    }

}

public class Quit : MonoBehaviour

{

    public void Exit() {

        Application.Quit(); //Quits the game

        // Just used to see it in editor. Can be removed.

        Debug.Log("Quit!");

    }

}