Git Initiate

(if unity project doesn't exist)

Start Unity 3D project, ThirdPersonMove
Save and exit the Project
Rename project folder to ThirdPersonMovex
Create Repo named ThirdPersonMove in Git
git clone https://github.com/rayhere/ThirdPersonMove.git
cd ThirdPersonMove
Drag the project files from ThirdPersonMovex into ThirdPersonMove folder
git add .
git commit -a -m "2nd commit, project initiated"
git push

第三人稱角色移動 | Unity新手教學

https://www.youtube.com/watch?v=-Q-g44IgX48

Step1

Required Package
Input System
Cinemachine
ProBuilder

Create

- ☐ Create Empty named Level >
    - ☐ Create 3DObject > Plane > apply Material named Ground
- ☐ Empty named ThirdPersonPlayer >
    - ☐ 3DObject > Capsule named Player > Pos.y 1.5 Scale.y 1.5 > add CharacterController > add PlayerInput > Create Actions [PlayerInput] named PlayerInputAction > apply PlayerInputAction in Action [PlayerInput]
        - ☐ 3DObject > Cube named Eyes > Pos 0, 0.6, 0.2 > Scale 0.6, 0.1, 1
    - ☐ Create Cinemachine > FreeLookCamera named PlayerCam > drag Player [Capsule] to Follow and LookAt [CinemachineFreeLook] > VerticalFOV 60 > Y Axis > InputAxisValue Invert checked > X Axis > InputAxisValue Invert unchecked > Orbits > BindingMode World Space > X Axis > Speed 80 > Orbits MiddleRig Height 1.5 > TopRig Radius 2, MiddleRig Radius 10, BottomRig Radius 2, MiddleRig Body Damping X,Y,Z to 0 > Aim Tracked Object Offset Y to 1.25 > BottomRig Body Damping X,Y,Z to 0 > Aim Tracked Object Offset Y to 1.25
    - ☐

Create 2 script
PlayerController.cs
ControllerMovement3D.cs

Drag
PlayerController.cs
ControllerMovement3D.cs
Into Player [GameObject] in ThirdPersonPlayer
Drag MainCamera into ControllerMovement3D.cs

Code

PlayerController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.InputSystem;

public class PlayerController : MonoBehaviour
{
    private ControllerMovement3D _controllerMovement; //To Grab
ControllerMovement3D.cs
    private UnityEngine.Vector3 _moveInput;

    private void Awake()
    {
        //Grab ControllerMovement3D from the Object with this script, so
we don't have to create SerializeField;
        _controllerMovement = GetComponent<ControllerMovement3D>();
    }

    public void OnMove(InputValue value)
    {
        Vector2 input = value.Get<Vector2>();
        _moveInput = new Vector3 (input.x, 0f, input.y);
    }

    private void Update()
    {
        if (_controllerMovement == null) return;

        _controllerMovement.SetMoveInput(_moveInput);
        _controllerMovement.SetLookDirection(_moveInput);
    }
}
```

ControllerMovement3D.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.TextCore.Text;

public class ControllerMovement3D : MonoBehaviour
{
    [Header("Movement")]
    [SerializeField] private float _moveSpeed = 5f;
    [SerializeField] private float _turnSpeed = 10f;
    [SerializeField] private GameObject _mainCamera;
    // Used to synchronize the rotation of the main camera with the
character's position.

    private float _speed = 0f;
    private bool _hasMoveInput;
    private Vector3 _moveInput;
    private Vector3 _lookDirection;

    private CharacterController _characterController;

    private void Start()
    {
        _characterController = GetComponent<CharacterController>();
    }

    // This method is used to set the movement input for the character.
    public void SetMoveInput(Vector3 input)
    {
        // Check if the player presses the key or not.
        // It checks if the player has pressed any keys. If the input
magnitude is greater than 0.1,
        // it sets '_hasMoveInput' to true to avoid stick drag or
floating-point error. Otherwise, it sets it to zero.
        _hasMoveInput = input.magnitude > 0.1f;
        _moveInput = _hasMoveInput ? input : Vector3.zero;
    }

    // to make the character actually rotate
```

```
    public void SetLookDirection(Vector3 direction)
    {
        // We only get axis x and z because the camera only moves
horizontally.
        // Rotate the player.
        _lookDirection = new Vector3(direction.x, 0f,
direction.z).normalized;
    }

    private void FixedUpdate() {
        _speed = 0;

        // If player not moving
        float targetRotation = 0f;

        if (_moveInput.magnitude < 0.1f)
        {
            _moveInput = Vector3.zero; // make movement to zero if
magnitude is too small
            return;
        }

        // Move character
        if (_moveInput != Vector3.zero)
        {
            _speed = _moveSpeed; // If player is moving
        }

        targetRotation =
Quaternion.LookRotation(_lookDirection).eulerAngles.y +
_mainCamera.transform.rotation.eulerAngles.y;
        UnityEngine.Quaternion rotation = UnityEngine.Quaternion.Euler(0,
targetRotation, 0);
        transform.rotation =
UnityEngine.Quaternion.Slerp(transform.rotation, rotation, _turnSpeed *
Time.fixedDeltaTime); // Smooth the rotation

        _moveInput = rotation * Vector3.forward;
        _characterController.Move(_moveInput * _speed *
Time.fixedDeltaTime); // Let CharacterController move the character
```

```
        }
}
```

第三人稱角色移動 | Unity新手教學

https://youtu.be/-Q-g44IgX48?si=QkH7__LkWqc5J170&t=374

Step2