~~We have~~ Scripting Movement

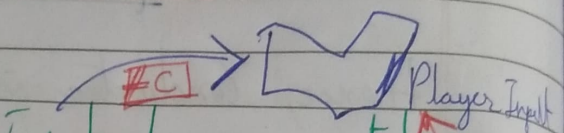
We have a C# class generated → Player Input

(1) Now, create another C# script "Input Manager" (which manages inputs)

In this code :- (Input Manager)

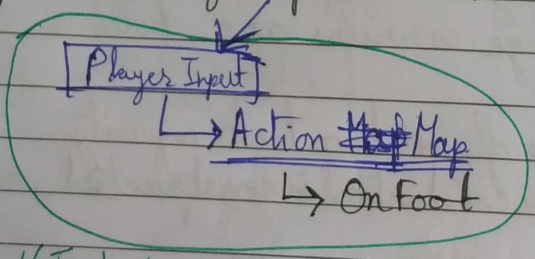
```
using Unity Engine.Input System; // Tells Unity we are using the NEW INPUT SYSTEM

public class InputManager : MonoBehaviour
{
```



- ~~public~~
  // Reference to Player Input class we generated
  private PlayerInput playerInput;

- // Reference to Player Input's Action Map
  private PlayerInput.OnFootActions onFoot



- // Initialize at beginning of scene
  void Awake() {
  playerInput = new PlayerInput();
  onFoot = playerInput.OnFoot;
  }

Referencing this class's Action Map

- private void Enable() {
  onFoot.Enable(); // → Enable onFoot
  }
- private void Disable() {
  onFoot.Disable(); // Disable onFoot
  }
}

(2) Then, create C# script "PlayerMovement" (which controls the player movement)

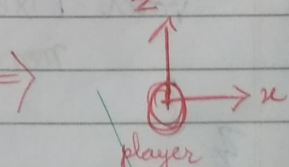
In this code :- (PlayerMovement)

```
public class PlayerMovement : MonoBehaviour
{
    // Player stats
    private CharacterController controller;
    private Vector3 playerVelocity;
    public float speed = 5f;

    // Initialize CharacterController
    void Start() {
        controller = GetComponent<CharacterController>();
    }
```

→ Get Component from Attached Gameobject

```
    // Recieve inputs from InputManager script
    // & apply them to character controller
    public void ProcessMove(Vector2 input) {
        Vector3 moveDirec = Vector3.zero; // Initialize to (0,0,0)
        moveDirec.x = input.x;
        moveDirec.z = input.y;
        controller.Move(transform.TransformDirection(moveDirec)
            * speed * Time.deltaTime);
    }
}
```



(input.x, 0, input.z)

→ Tells the controller to move towards moveDirec

→ rate of movement

→ Converts moveDirec from local → global space

③ Let's modify "Input Manager"

In this code: (Input Manager)

```
:
public class InputManager : MonoBehaviour
{
    private PlayerInput playerInput;
    private PlayerInput.OnFootActions onFoot;

    private PlayerMotor motor; //Reference to
    private PlayerMovement movement;

    void Awake(){
        playerInput = new PlayerInput();
        onFoot = playerInput.OnFoot;

        movement = GetComponent<PlayerMovement>();
    }

    //Process Input
    void FixedUpdate(){
        movement.ProcessMove(onFoot.Movement.ReadValue<Vector2>());
    }
}
```

(Same as previous)

Player Movement
↳ public void ProcessMove()

Action Map
OnFoot → Actions
- Movement
- Jump

Action Properties
▽ Action
Action Type [Value]
Control Type [Vector 2]

is of type <Vector 2>

Hence, we are reading Vector 2 value

from Movement Actions

classmate
Date
Page

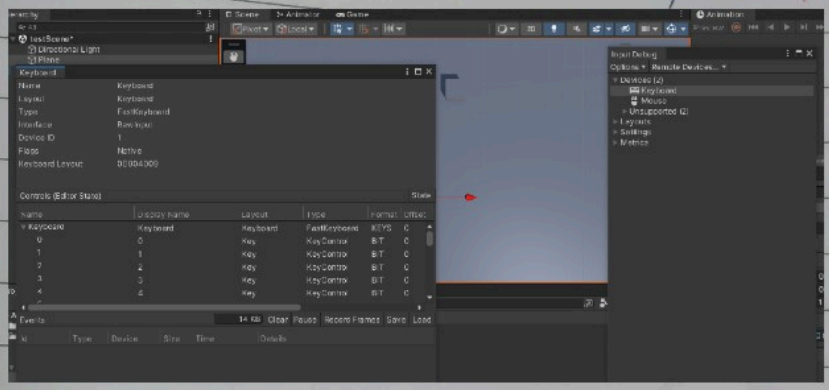
4) Attach both scripts to Player

Input Manager

Player Movement

We can debug our movement Inputs :-

Window → Analysis → Input Debugger



Hit Play & press keys to see the analysis