Character Control - Unity Taster Session

# Objective:

By the end of this session, you will have a fully animated 3D character that can walk and run using keyboard input in Unity.

# Setting Up Unity (5 minutes)

* + - Open Unity Hub and log into an existing Unity account or create a new one.
    - Once logged in, click New Project inside Unity Hub.
    - Select the 3D (Core) Template and name your project 'CharacterControl'.
    - Click Create Project and wait for Unity to load.

# Downloading & Importing a Mixamo Character (10 minutes)

## Download from Mixamo

* + - Go to [https://www.mixamo.com/](http://www.mixamo.com/) and log in (create an Adobe account if needed).
    - Choose a character (e.g., Xbot).
    - Click Download, selecting: Format: FBX for Unity, Pose: T-Pose.
    - Download Walk, Run, and Idle animations separately, ensuring 'In Place' is checked and “Without Skin” is selected.

## Import into Unity

* + - Drag and drop your character and animation files into Unity's Assets folder.
    - Click on your character file, go to the Materials tab, and set Location to Use External Materials (Legacy)
    - Now, go to the Rig tab, and set Animation Type to Humanoid, then click Apply. Repeat this step for the three animation files as well.
    - Next, go to the Animation tab of each of the three animations, and tick the Loop Time option, so they will keep looping when we play them instead of playing just once.
    - Place your character in the scene by dragging and dropping them in.
    - Add a Character Controller component to the character as this will be used later for the movement.
    - Ensure the Character Controller capsule’s Center matches the center of your character by changing its Y coordinate to 1.

# Setting Up the Animator (10 minutes)

## Create the Animator Controller

* + - Right-click in the Assets folder and create a new Animator Controller (name it 'CharacterAnimator').
    - Drag it onto your character in the Hierarchy.
    - Open the Animator Controller.
    - Drag your Idle, Walk, and Run animations into the Animator window.
    - Right-click the Idle animation -> Set as Default State.
    - On the left-hand side, open the Parameters tab and create a new Float parameter named 'Speed'.
    - Create transitions between animations: Idle -> Walk, Walk -> Run, Run -> Walk, Walk -> Idle.
    - Untick the Has Exit Time option for each transition, so they happen quicker.
    - Click on each transition and create a new condition based on the 'Speed' parameter. You should make the following conditions:

- Idle -> Walk transition happens when Speed > 0.1

- Walk -> Run transition happens when Speed > 0.5

- Run -> Walk transition happens when Speed < 0.5

- Walk -> Idle transition happens when Speed < 0.1

# Coding the Character Movement (15 minutes)

* + - In the Assets folder, right-click -> Create -> C# Script. Name it 'CharacterMovement'.
    - Double-click to open it in Visual Studio.
    - Write the following movement code in the script:

using UnityEngine;

public class CharacterMovement : MonoBehaviour

{

private Animator animator;

private CharacterController controller;

public float walkSpeed = 2f;

public float runSpeed = 5f;

public float rotationSpeed = 180f;

void Start()

{

controller = GetComponent<CharacterController>();

animator = GetComponent<Animator>();

}

void Update()

{

float moveZ = Input.GetAxis("Vertical");

float moveX = Input.GetAxis("Horizontal");

float currentSpeed = Input.GetKey(KeyCode.LeftShift) ? runSpeed : walkSpeed;

controller.Move(transform.forward \* moveZ \* currentSpeed \* Time.deltaTime);

transform.Rotate(Vector3.up, moveX \* rotationSpeed \* Time.deltaTime);

float animSpeed = Mathf.Abs(moveZ) \* (currentSpeed / runSpeed);

animator.SetFloat("Speed", animSpeed);

}

}

* + - Select the character in the Hierarchy and drag the script asset to the Inspector window to attach it.
    - (Optional) For a third-person camera view, place the camera behind the character and make it a child component of the character object by dragging it onto that object.

Hint: You may need to untick “Apply Root Motion” on the character’s Animator component, to stop the camera swaying as they run.