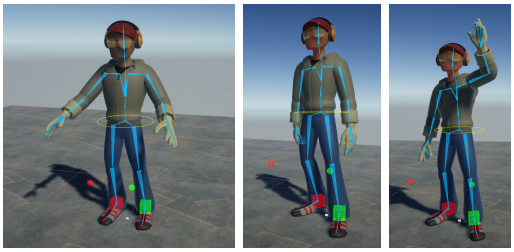
Character assignment 7 - Animating a walk cycle

UW CSE490 summer 2023
Instructor: Dave Hunt

Introduction

In this assignment we will use the control rig from the previous assignment to animate a walk cycle. We will then create a state machine with a basic character controller that incorporates the walk, idle and emote animations to enable player driven locomotion at run-time. Finally, we will place our character in the scene and record a short gif or video of the character walking.



Start from the AnimationTestScene with your character's control rig prefab that was built in assignment 6

Save it as a new Unity scene named *AnimationWalkScene.unity*

Part 1, Blocking in the main walk poses

It is helpful to think of a walk cycle in these four main poses: *contact, down, passing* and *up*. We will set keys on all rig controls for these main poses spaced 10 frames apart as a starting point.

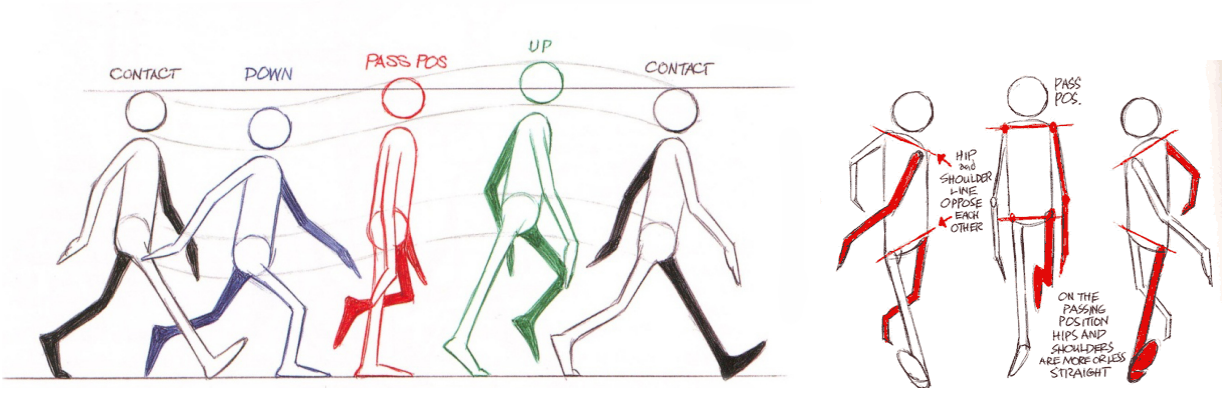
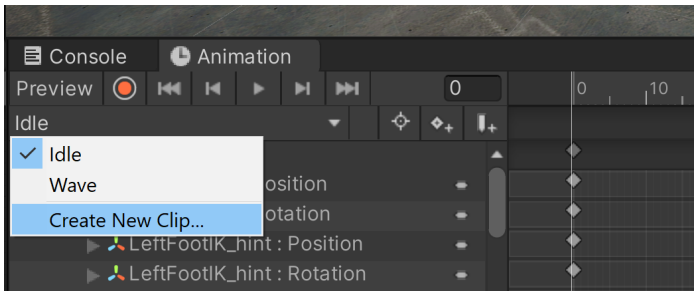


Image source: Animators Survival Kit, Richard Williams

1.1 Create a new animation clip

1. Select your character prefab in the scene Hierarchy
2. In the **Animation Window** click on the clip menu and Create New Clip...

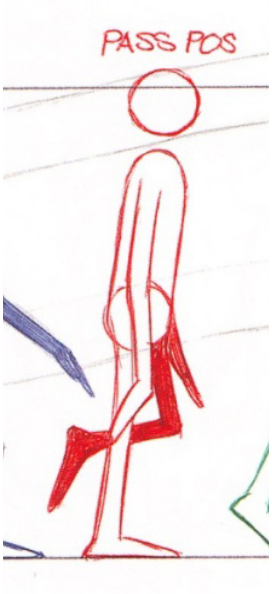


3. Name it *Walk* and save the clip in your character's animations folder `Assets/Assignments/Characters/Sly/Animations`
4. Lock the Animation Window (lock button is in the top-right corner)

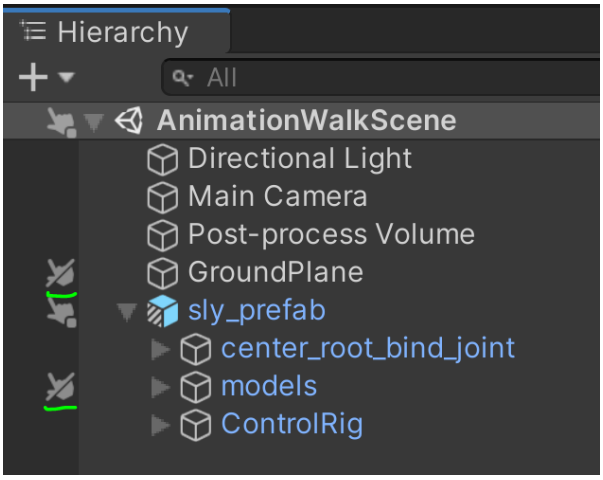
1.2 Set the Passing pose

It is good to start the walk cycle in a pose that is near to the idle pose. This will help it look smooth when blending animations in the player controller state machine. Since we made the idle pose with right foot slightly back it is best to start the walk in the right foot **passing pose**. (note that the drawings below are of the opposite side)

The passing pose is the most important pose in the walk cycle for expressing a character's mood and personality. How high the passing foot raises makes a big difference in the perceived speed, energy and attitude of the walk. Take some time to experiment with what works best for your character's passing pose. You may want to come back and adjust this pose again after all of the main poses are blocked in so you can see how the poses blend together.



1. Turn on Record mode in Animation Window (red circle button)
2. Set a key in the bind pose on all rig controls on frame 1
    a. In the scene Hierarchy alt-click *center_root_bind_joint* to expand all of the bones
    b. Shift-select all bones
    c. In the Scene view shift-select the rig controls for hips, feet and knees
    (or ctrl-select in Hierarchy, but don't select the other nodes in the control rig)
    d. In Inspector right-click Set Key on **position** and **rotation**
3. Rotate the camera to a side viewing angle
4. Move the rig controls and rotate the bones into the **passing** pose. Try to replicate the sketch as a pose on your character rig.



Tip: setting objects to not-pickable
Unity manual, picking and selecting

In the scene Hierarchy there are finger icons next to each GameObject to control if it is pickable in the Scene view. This is useful for animating because we only want to pick rig controls and bones.

In this screenshot the *GroundPlane* and character *models* are set to not-pickable in order to make it easier to interact with the rig when animating.

Discussion: Walking in place or using root motion?

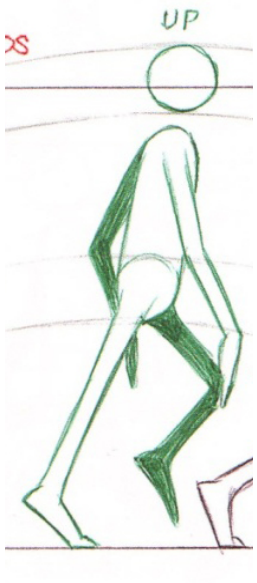
It is considered best-practice to animate walk cycles with the character translating forward in space. This makes it easier to keep the feet in contact with the ground so that there is no foot sliding when it is played back at run-time. The character controller extracts this forward translation as Root Motion and applies it to the prefab root in order to move the capsule collider forward.

Another way to animate walk cycles is where the character is **walking in place**. When

the animation clip has no forward translation the prefab root with the capsule collider needs to be moved forward manually by the character controller. This is often done by exposing a public variable for move speed that can be tuned to look visually in sync with the speed of the animation. It may take some experimentation to get the right speed so it looks like the feet are not sliding on the ground.

For this assignment we will be using the **walking in place** method. This will help us keep the implementation simple in the character controller, which will be preferable for an introductory level class like this. Just please make note that the more common way of doing this is where the walk cycle is animated to have forward translation.

1.3 Setting the Up pose



1. In Animation Window move forward in time to frame 10
2. With nothing selected press the "**k**" hotkey to key everything
3. Set the **up** pose
    a. Translate the hips upward slightly
    b. Move the left foot back with the heel lifting off the ground slightly. Try not to hyper-extend the knee, it should have at least a slight bend in order to not look rigid.
    c. Move the right foot forward and rotated downward
    d. Rotate the spine so the left shoulder aims more forward
    e. Rotate the neck so the head is looking forward
    f. Rotate the arms and hands to counterbalance the legs
4. Scroll the time slider to see how the two poses interpolate
5. View the poses from the side, front and back. Make sure the poses look good from all angles.

1.4 Setting the Contact pose

6. In Animation Window move forward in time to frame 20
7. With nothing selected press "**k**" to key everything
8. Set the **Contact** pose
    a. Translate the hips downward slightly
    b. Move the right foot forward and rotated upward so the heel contacts the ground
    c. Move the left foot back and rotated downward with the toes flat on the ground
    d. Rotate the spine to push the left shoulder more forward
    e. Swing the arms to counterbalance the legs
9. Scroll the time slider to see how the poses interpolate
10. Make sure the poses look good from all angles

1.5 Setting the Down pose

11. In Animation Window move forward in time to frame 30
12. With nothing selected press "**k**" to key everything
13. Set the **Down** pose
    a. Translate the hips downward slightly
    b. Set the right foot flat on the ground with the knee bent
    c. Pose the left foot with the tip of the toes just leaving the ground
    d. The arms should be rotated to their farthest extent in this pose
14. Scroll the time slider to see how the poses interpolate
15. Make sure the poses look good from all angles

1.6 Set the left side copies of each pose

It would be very nice to have tools for automatically mirroring poses on the rig so we can create similar looking main poses for the opposite side. This is often where technical artists come in on productions to develop tools to improve efficiency for animators. Instead, we will get some practice working with the rig by manually posing the opposite side ;)

1. Set the left side **Passing** pose on frame 40
    a. Flip back to the opposite side pose on frame 1 to check if it looks similar
2. Set the left side **Up** pose on frame 50

3. Set the left side **Contact** pose on frame 60
4. Set the left side **Down** pose on frame 70
5. Copy the first frame to the end to make the animation loop smoothly
    1. In Animation Window move to frame 1
    2. Switch to the Dopesheet tab in Animation Window
    3. Select the top key on frame 1 to select all keys and ctrl-c to **copy**
    4. Move forward in time to frame 80
    5. Use ctrl-v to **paste** the keys

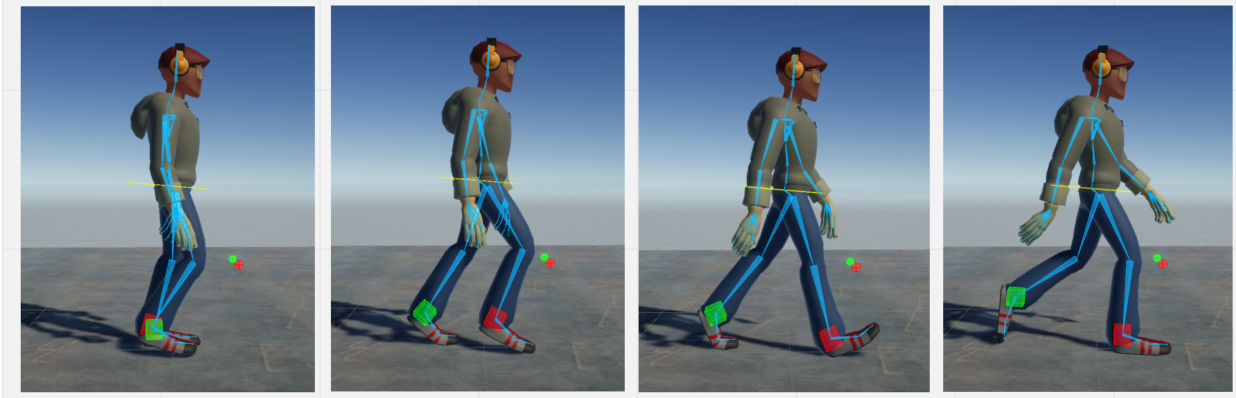
1.7 Refine the poses

1. Press Play in Animation Window and view it from all angles in the Scene view
2. Check the front view and make sure the poses feel balanced
3. Adjust the poses if they look off

End of part 1.

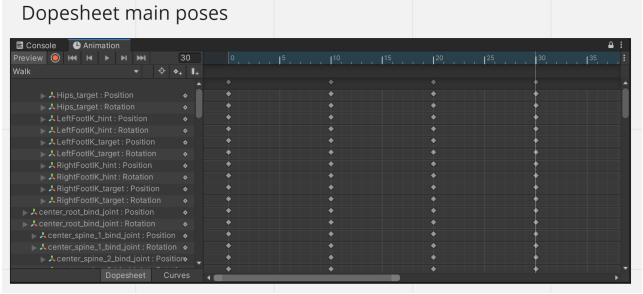
Save screenshots of your main walk poses to Miro to turn in with your assignment. Also add a screenshot of your Animation Window Dopesheet with 4 rows of keys where all bones and rig controls are keyframed.

Walk main poses



Passing | Up | Contact | Down

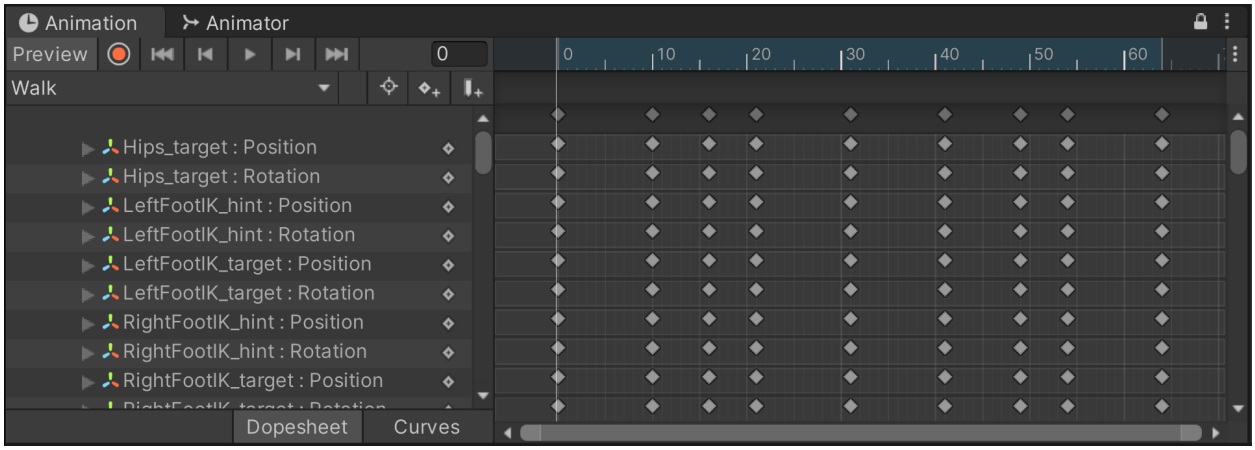
Dopesheet main poses



Part 2, Refining the timing and interpolation

2.1 Adjust the timing of the walk poses

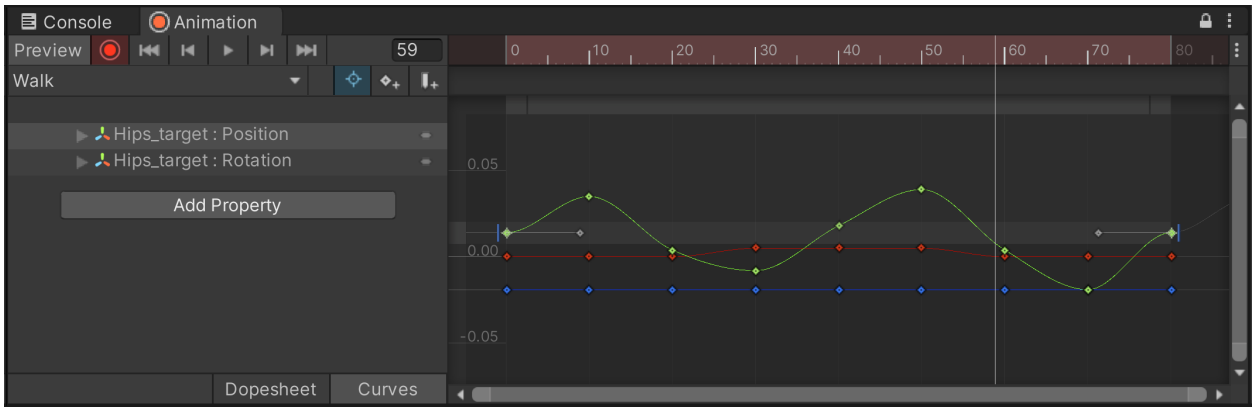
1. Decide if your character should walk at a faster or slower pace
2. Also look for where the poses should happen closer in time, for example between the contact and down poses
3. Adjust timing by moving complete rows of keys Dopesheet forward and back in time. For now keep keys aligned for the main poses, all keys at the same frame.



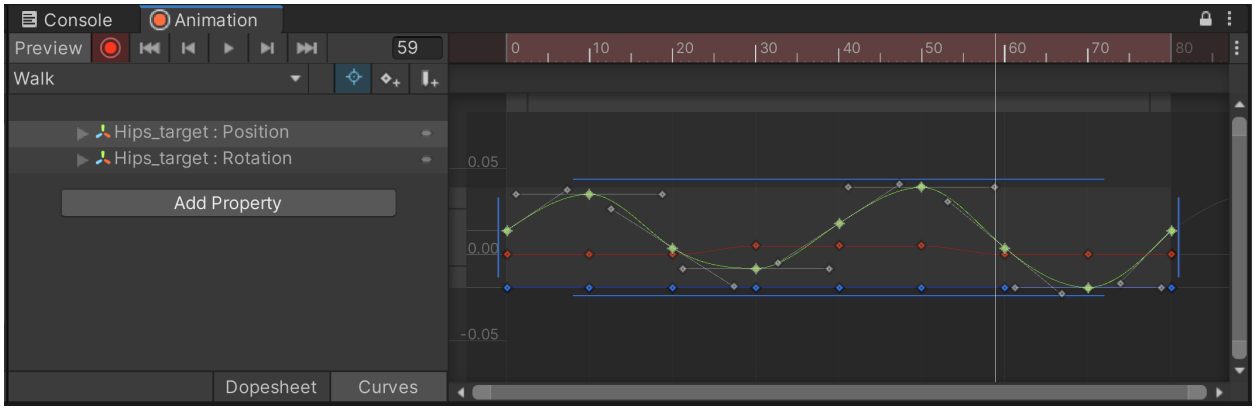
*Save a screenshot of the Dopesheet with timing for the walk keyframes and add it to Miro to turn in with your assignment.

2.2 Adjust the tangents to smooth out the animation curves

1. Press Play in Animation Window and check to see how the poses are interpolating. There is sometimes a hitch in timing of rotations or translations that looks unnatural. Fix these by modifying key tangents to influence the shape of the animation curve.



2. In the screenshot above the hips Position Y curve is not smooth at the clip boundary. This can be seen as a slight bump in the character's motion. Adjust the tangents of the start and end keys to make it loop smoothly.



3. Here is a screenshot where the hips Position Y curve after the tangents were adjusted
4. Continue checking all the bones and rig controls, smooth out any bumps in the motion by tuning the key tangents

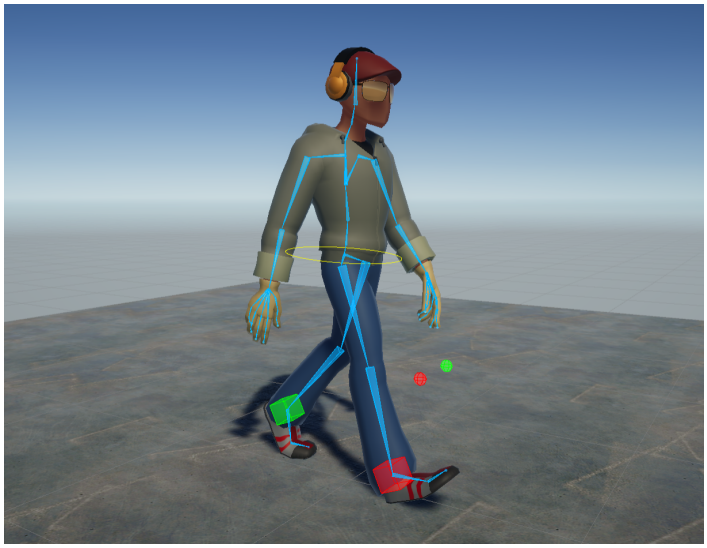
2.3 Add in-betweens

Finally, after fixing as much as possible with the existing keys and tangents we can now add a few more keys where we need more detail in the animation curves. These in-between keys don't need a key on all bones and rig controls, only on the channels that need to move.

1. Set keys on individual bones or rig controls where it's necessary to control the curves beyond what tangents can do
2. Check the knees and watch out for any hyperextensions or sudden pops where it goes straight and feels rigid. Smooth out any pops where the leg straightens out too quickly.

End of part 2.

Capture a gif or video of your walk cycle and add it to Miro to turn in with your assignment.

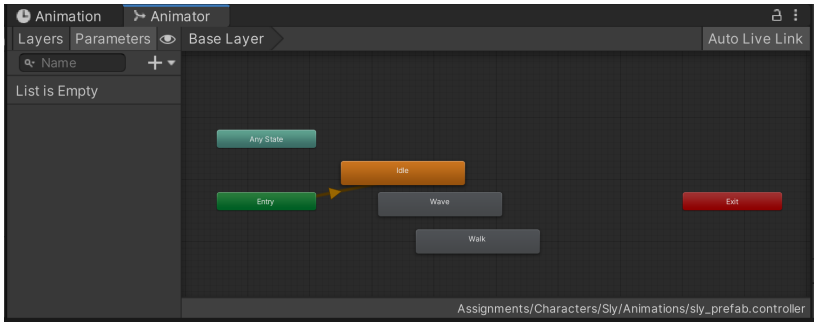


Part 3, Setting up a player controller state machine

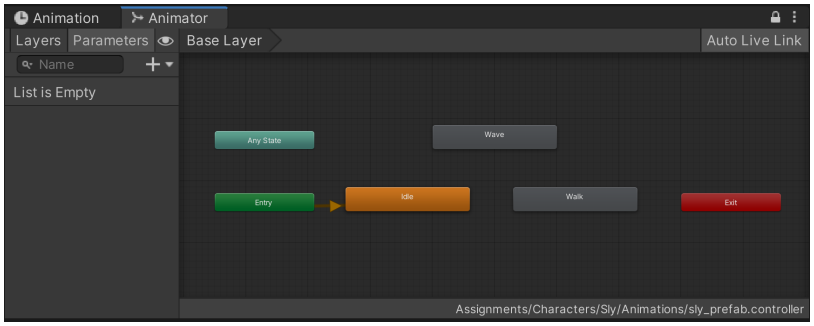
The **player controller** is the gameplay system that causes the character to move in response to player input. It uses a **state machine** to blend multiple clips such as walk and idle. Here we will set up a very basic player controller to enable our character to locomote in their world.

3.1 Open the Animator window

1. Window → Animation → Animator
2. Dock the **Animator** panel below Scene view where you can see both at the same time
3. Select your character prefab in scene Hierarchy and it displays their Animator Controller



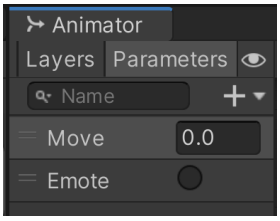
4. Lock the Animator window (lock button is at the top right corner)
5. It should contain the clips for *Idle*, *Walk* and your emote that were made in previous steps. If not then add them by dragging and dropping the clips into the Animator window.
6. Rearrange the clips to look more like this:



3.2 Set up the parameters

Parameters are gameplay variables that influence the conditions of transitions between states.

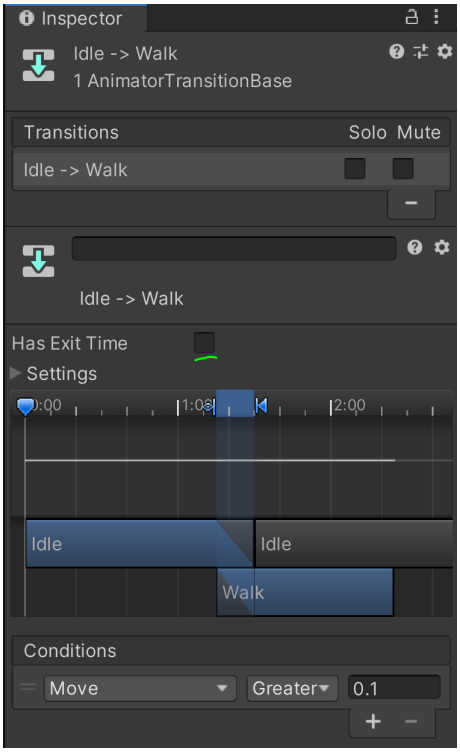
1. Add a parameter for moving
    a. Click the "**+**" button in the Parameters tab in the Animator window
    b. Choose type **float** and name it *Move*
2. Add a parameter for the emote
    a. Click the "+" button again
    b. Choose type **trigger** and name it *Emote*



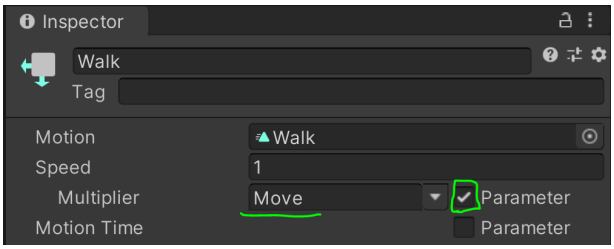
3.3 Set up the Idle to Walk transitions

The **default state** is colored orange and it will play first when the game starts. We can blend to different states by setting up **transitions**.

1. Right-click the Idle state and Make Transition
2. Drop the arrow that extends from *Idle* onto *Walk*
3. Select the transition arrow and set up the conditions in the Inspector
    a. Uncheck Has Exit Time
    b. Set the condition to **Move**, **Greater** than **0.1**

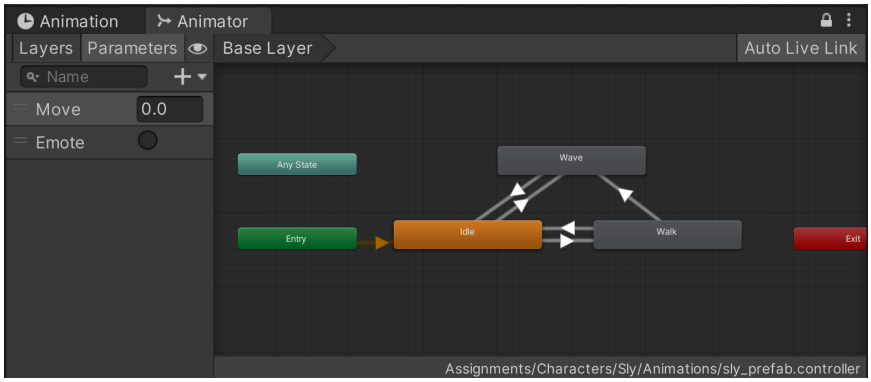


4. Add another transition back from *Walk* to *Idle*
5. Select the arrow and in the Inspector set the following options
    a. Has Exit Time = false
    b. Condition: Move, Less than 0.1
6. In the Walk state set Speed Multiplier to be driven by the Move parameter



3.4 Set up the transitions for the emote

1. Add a transition from *Idle* to the emote with these settings
    a. Has Exit Time = false
    b. Condition: Emote
2. Add a transition from *Walk* to the emote with these settings
    a. Has Exit Time = false
    b. Condition: Emote
3. Add a transition back from the emote to *Idle* with these settings
    a. Has Exit Time = true



3.5 Add the player controller script

1. Create a new C# script
    a. In Project view go to the Assets/Scripts folder
    b. Right-click the Scripts folder and Create → C# Script
    c. Name it *ThirdPersonPlayerController*
2. Assign the script to your character prefab
    a. Select your character prefab in the scene Hierarchy
    b. Drag and drop the ThirdPersonPlayerController script into the Inspector
    c. Apply the override to the prefab so the script will stay added
3. Double-click the script in Project view to open it in Visual Studio
4. Copy and paste the following code into the script and save it

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

public class ThirdPersonPlayerController : MonoBehaviour
{
    public float moveScale = 1.0f;
    public float moveSpeed = 2.0f;
    public float turnSpeed = 5.0f;
    private Animator animator;

    void Start()
    {
        animator = gameObject.GetComponent<Animator>();
    }

    void Update()
    {
        if (Input.GetKeyDown("1"))
        {
            animator.SetTrigger("Emote");
        }

        float moveInput = Input.GetAxis("Vertical");
```

```
        animator.SetFloat("Move", moveInput);

        if (moveInput > 0.1)
        {
            transform.Translate(0, 0, moveInput * moveSpeed * moveScale * Time.deltaTime);
        }
        transform.Rotate(0, Input.GetAxis("Horizontal") * turnSpeed * 50f * Time.deltaTime, 0);
    }
}
```

NOTE: Watch out for a copy/paste error that will cause your script to not work. There can't be an extra enter (new line) after the final } character. (This is because of how Google Docs copy/paste works)

3.6 Add collision and physics components to your character prefab

1. Add a RigidBody component to your character
    a. With your character prefab selected in the Hierarchy view click Add Component in the Inspector. In the search field type RigidBody
    b. In the RigidBody Constraints check Freeze Rotation X and Z.
    c. In the RigidBody component set Is Kinematic to true.
2. Add a CapsuleCollider component and fit it to your character model
    a. Set Center Y to the midpoint of your model. For my character it is 0.9
    b. Set the Height to fit your character. For my character it is 1.8
    c. Set the Radius to fit your character. For my character it is 0.3
3. On the Animator component set Apply Root Motion to true

3.7 Tune the player controller values to look good for your character

1. Press Play in the Unity editor
2. In the Game window press the arrow keys or WASD to move your character around
    a. If the game camera has its own controller using WASD you might need to turn that component off first
3. Select the character prefab and scroll to the ThirdPersonPlayerController component in the Inspector
4. Modify the values for Move Scale, Move Speed and Turn Speed until they look good based on how fast the walk animation plays
5. When you like how the values are set (before exiting Play mode) go to the context menu of the ThirdPersonPlayerController component (3 dots button) and Copy Component
6. Exit Play mode
7. Go to the context menu of ThirdPersonPlayerController and Paste Component Values
8. Apply the overrides to your character prefab

Part 4, Going for a walk in your world

4.1 Add your character prefab to your scene

1. Open animation test scene
2. Add your character prefab to the scene
3. Press Play and walk around the scene
    a. Check to make sure the speed of the walk feels right for the character

4.2 [optional] **Add a 3rd person follow camera with Cinemachine**

1. Open Package Manager and install the Cinemachine package
2. Here is the Cinemachine documentation for Free Look Camera
3. Follow this tutorial for setting up Free Look Camera for your character
4. Tune the camera settings to get a good view of your character

End of part 4.

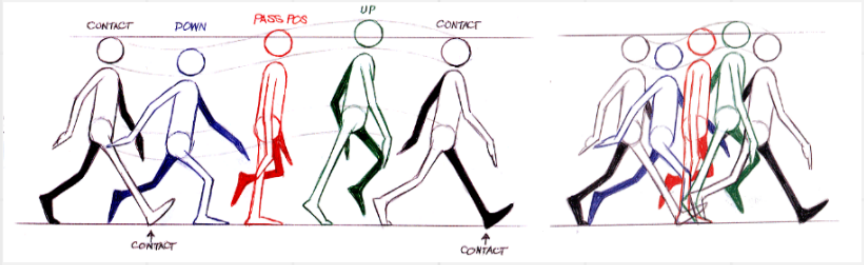
Capture a short animated gif or video of your character walking around the scene and add it to Miro to turn in with your assignment.

Grading rubric

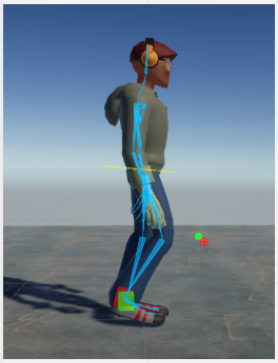
Criteria	**Achievement levels**			
	level 1	level 2	level3	level4
Passing pose format: images [Miro]	15 points: (0 incorrect) Left foot flat on the ground, right foot raised and passing the left, arms down	14 points: (1-2 incorrect) Left foot flat on the ground, right foot raised and passing the left, arms down	13 points: (3+ incorrect) Left foot flat on the ground, right foot raised and passing the left, arms down	0 points: files were missing
Up pose format: image [Miro]	15 points: (0 incorrect) Body is raised up, left toe is contacting the ground	14 points: (1 - 2 incorrect) Body is raised up, left toe is contacting the ground	13 points: (3+ incorrect) Body is raised up, left toe is contacting the ground	0 points: files were missing
Contact pose format: image [Miro]	15 points: (0 incorrect) Right heel contacts ground, left foot is raised up on toe	14 points: (1-2 incorrect) Right heel contacts ground, left foot is raised up on toe	13 points: (3+ incorrect) Right heel contacts ground, left foot is raised up on toe	0 points: files were missing
Down pose format: image [Miro]	15 points: (0 incorrect) Right foot is flat on ground, body is lowered down, arms are at widest extent	14 points: (1-2 incorrect) Right foot is flat on ground, body is lowered down, arms are at widest extent	13 points: (3+ incorrect) Right foot is flat on ground, body is lowered down, arms are at widest extent	0 points: files were missing
Dopesheet main poses format: image [Miro]	10 points: (0 incorrect) Four rows of keys, all properties have keys every 10 frames	9 points: (1-2 incorrect) Four rows of keys, all properties have keys every 10 frames	8 points: (3+ incorrect) Four rows of keys, all properties have keys every 10 frames	0 points: files were missing
Walk animation format: animated gif or video [Miro]	15 points: (0 incorrect) Motion looks like walking, is looping, the pace feels right for this character	14 points: (1-2 incorrect) Motion looks like walking, is looping, the pace feels right for this character	13 points: (3+ incorrect) Motion looks like walking, is looping, the pace feels right for this character	0 points: files were missing
Player control video format: animated gif or video [Miro]	15 points: (0 incorrect) Camera follows character as they walk	14 points: (1-2 incorrect) Camera follows character as they walk	13 points: (3+ incorrect) Camera follows character as they walk	0 points: files were missing

Character assignment 7 - walk

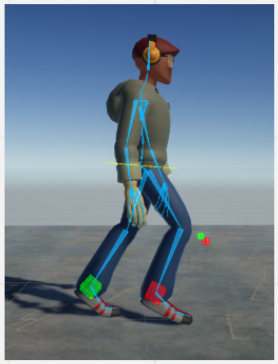
Walk reference - Animators Survival Kit, Richard Williams



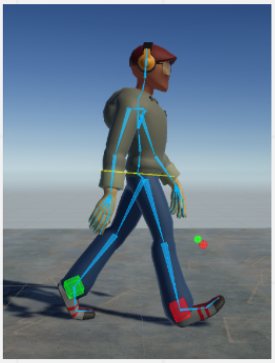
Walk main poses



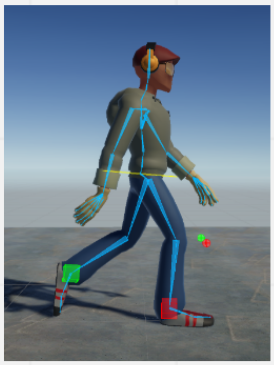
Passing



Up

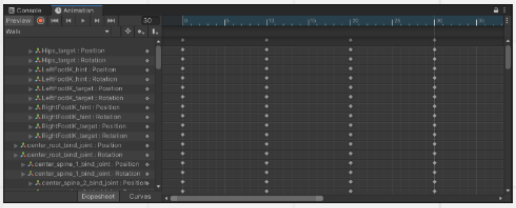


Contact

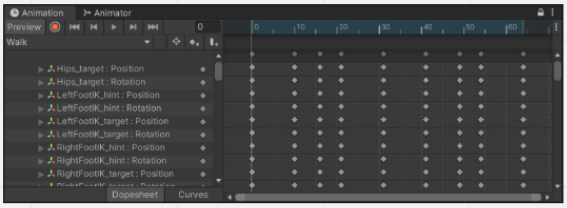


Down

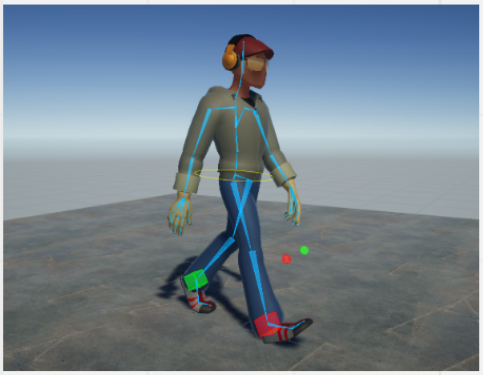
Dopesheet - walk keyframes



Dopesheet - timing



Walking cycle



Walking around in the CHOP zone