Lab6 – Character Animation and Motion Graphs

CSE 3541 Games and Animation, The Ohio State University

1. Requirements

In this project, we will use Unity 3D to write a simple game that contains a virtual human character doing different movements, including running, jumping, waving, and feign death. This lab assignment does not require much coding, but you must know what to do and how to use the animation features provided in Unity.

Example package You will be provided with an example package that contains most students for this lab assignment. In the animation folder, you will find multiple animation sequences. You can also find a texture grassland for the ground floor.

1.1. Basic setup (2 Points)

At this time, the scene is empty and your job is to complete it. Create a plane called grassland and assign the texture to it, with multiple tiling. Create a directional light source and use it to illuminate your character. Set the light source and the camera to be the children of the character object, so that when the character moves, the light and the camera can follow.

1.2. Basic Running (2 Points)

Your next job is to create an animation controller, assign it to the character, and implement the basic running feature by using two states: idle and run. Use the **IdleShort** sequence for the idle state and the Run sequence for the run state. Define a floating-point parameter called speed and use it to define the transition between these two states. Next, create a script and obtain the animator object using **GetComponent**<Animator>. Also use the attribute to require the GameObject to have an Animator component. In the ***Update*** function, get the forward speed assign it to the ***speed*** parameter using **SetFloat**. For this simple demo, we will use the Unity New Input System to set the speed. Use *W* and up-arrow on the keyboard. This will be a value of *1* which you can scale with a [SerializeField] parameter for your script. For a gamepad, you can use a more continuous range for the speed based on how far the joystick is pushed forward.

1.3. Running and turning (2 Points)

Next, we will replace the run state by a new blend-tree state called Real Run. Add three motion sequences into this new state and use a new floating-point parameter Direction to control their blending. Specifically, **RunLeft** has the direction value of *-1*, **Run** has the direction value *0*, and **RunRight** has the direction value *1*. Add to your input events for left and right. For example, while the *A*-key or left-arrow key is pressed (or left on the gamepad), assign a variable, horizontal to be -1 (or the amount of the gamepad). Similarly for the right direction. Once the key is released, set the horizontal value to zero. Use this variable to assign a value to the Direction parameter as: ***SetFloat****("Direction", horizontal, 0.25f,* ***Time.deltaTime***). The last two variables here are used to prevent the values from changing too fast. After this, you should be able to see the character running and making turns.

1.4. Other motions (3 Points)

Add the other sequences into the animation controller as Figure 1 shows. Use the *D*-key to control the transition from any state to dying; use *R*-key to control the transition from dying to reviving; use *H*-key to control the transition from idle to waving hello; and use *space*-key to control the transition from real run to jump. The other transitions are controlled by time. The state structure is shown in Figure 1. Note that jumping happens only when the character is running and the transition from waving to running can happen immediately if speed is large enough. Remember to turn the parameters off after the transition has already been triggered, or future transitions will be wrong. How to do it? (Hint: the transition is done after each Update call.)

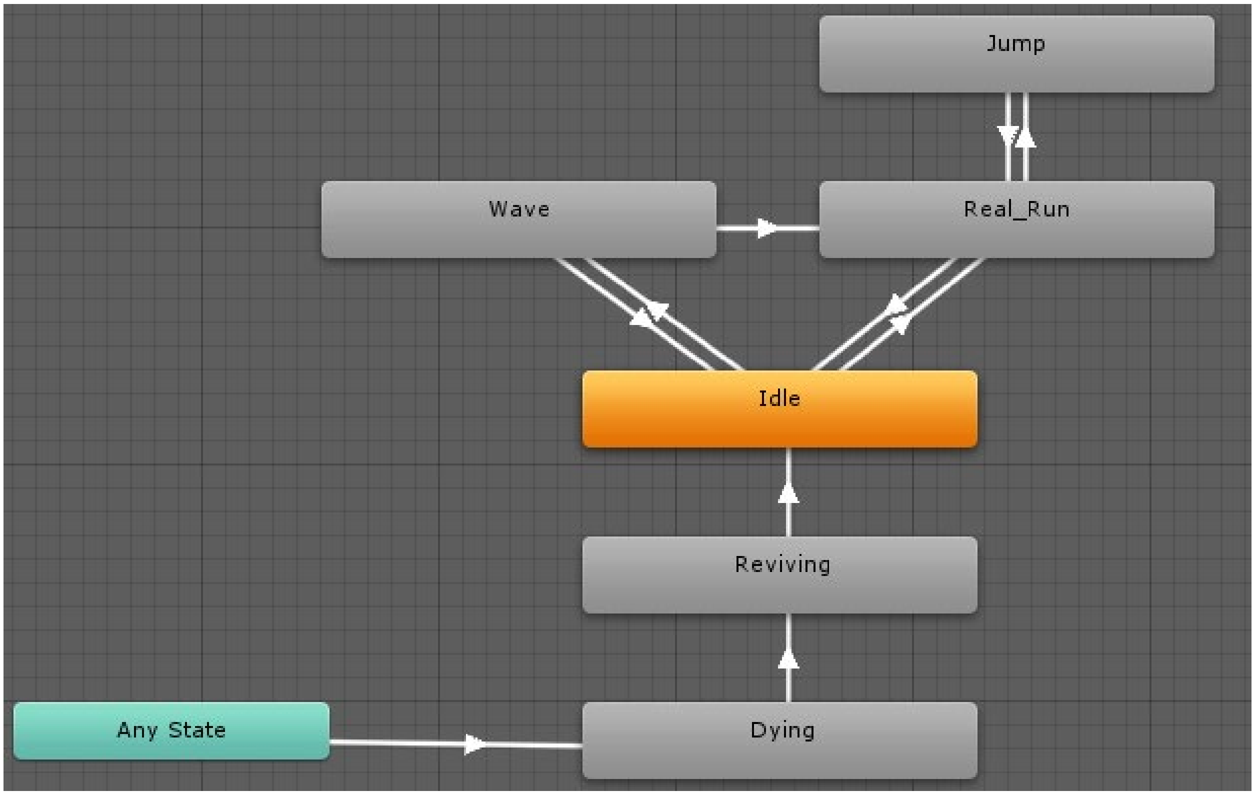

Figure 1: Animation control states.

1.5. Extra Credit

- Use Unity Record (or some other recording software) and create a video highlighting what you did, upload it to Youtube and include a link in your report. – 10%
- Add 2 other characters from the Mixamo website (https://www.mixamo.com/ - Characters in the upper-left) that have the same behavior. You can control all three at the same time by having them all subscribe to the input events. – 10%
- Add 2 other animations from the Mixamo website (https://www.mixamo.com/ - Animations in the upper-left) (e.g., search for dance or dancing) – 10%

Submission Guideline

Do not forget to save your scene file! After that, you can find all of your files, including the scene, the scripts, and the materials in the folder under the Project sub-window. Select your Scene, right click to export it and necessary files into a package. Save the package as Lastname_Firstname_Lab5. Submit via Carmen.