CSE 3541, Lab Assignment 5

Motion Graph, due on 04/03/2015 11:59PM

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 As before, you will be provided with an example package that contains most stuffs for this lab assignment. In the animation folder, you will find multiple animation sequences. In the skybox folder, you will find one skybox called `Sunny2`. (This skybox is part of the whole skybox package, which can be imported when you create a new project.) You can also find a texture `grassland` for the ground floor.

1.a. 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. Add a skybox component to the camera and assign `Sunny2` to that component. 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.b. 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>`. In the `Update` function, get the forward speed using `Input.GetAxis("Vertical")` and assign it to the `speed` parameter using `SetFloat`.

1.c. 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 -1, `Run` has the direction value 0, and `RunRight` has the direction value 1. In the script, get the turning direction using `Input.GetAxis("Horizontal")`. Assign it 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.d. Other motions (4 Points) Add the other sequences into the animation controller as Figure 1 shows. Use "1" to control the transition from `any state` to `dying`; use "2" to control the transition from `dying` to `reviving`; use "3" to control the transition from `idle` to `waving`; and use whitespace 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,**

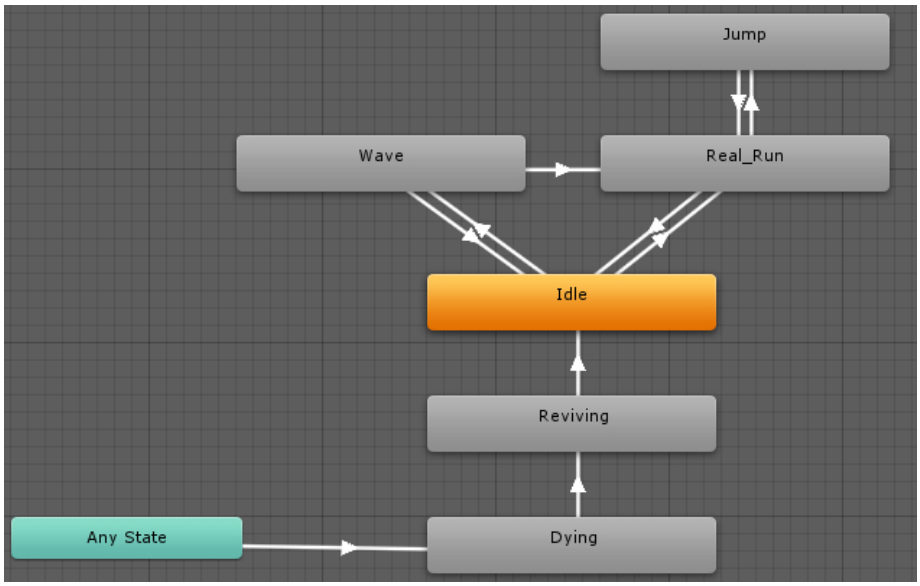

Figure 1: Animation control states.

or future transitions will be wrong. How to do it? (Hint: the transition is done after each `Update` call.)

2 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 subwindow. Select them all and right click to export them into a package. Save the package as lastname_firstname_Lab5. This will create a lastname_firstname_Lab5.unitypackage in the folder you are using. Since the file will be big this time, we will use Carmen for submission.