Challenge 2 - Character Animation

Refer to “About the Challenges and Solutions” in Session 1 for more information about readings of this type.

Scenario

Now, it’s time to add animation to both the Player character and the Enemy ACS-17 robot.

Challenge

To familiarize you with multiple forms of animation control, you’ll be both editing the state machine in the Animator for the Player and directly manipulating the animations for the Enemy.

Tasks to Complete

Player Animation
The Player already has an animation state machine in the Animator that was pulled from Unity Standard Assets. Modify it in the following ways:

- Disable the transitions to Airborne and Crouching
- Add an InCover state that is entered when the parameter inCover == true
- Make sure the state returns to Grounded when InCover==false
- Inside of the InCover state create a Blend Node that blends appropriately between the Left Cover Sneak, Cover Idle, and Right Cover Sneak animations based of the value of the property Creep. These animations can be found in Project > Adventure Player by Unity > Cover Animations from Mixamo.

Enemy Animation
The Animator for the EnemyBot (ACS-17) has all states in it, but we want to implement this animation without using the Transitions in the Animator. Instead, please use the Animator.CrossFade() method. Note: The Animator for the Enemy is on the ACS17 child GameObject, not the root-level Enemy GameObject.

- Make changes to the ACS17AnimationManager script that is already attached to the ACS17 GameObject.
- Read public mode and turnDir information from EnemyNav on the parent GameObject to know which state of the Animator to switch to:
- When mode == (idle || wait), show the ACS_Idle state

- When mode == move, show the ACS_Forward state
- When mode == (preMoveRot || postMoveRot), show ACS_TurnLeft or ACS_TurnRight based on turnDir.
- Adjust the speed of each ACS animation to match the movement of the Enemy
- Try using the anim.CrossFade() method with a transition time of 0. It will make things easier.

Bonus Challenge – Enemy

- Try changing the transition time of the CrossFade() method to something like 0.25f. This will require you to only call CrossFade() once for each transition to a new state (otherwise, the CrossFade will continually restart and appear to freeze the animation).

Start by downloading the Unity project files for this Challenge available in the Session Resources. Download the zipped file, unzip it to a local folder, and open the project using Unity version 2017.4 LTS.

When you're finished, complete the Self-Evaluation coursework before continuing to the instructor's solution.