Mocap Data to UDK

Matt Latas -- TECH 326

Goal: Being able to take Motion Capture data from Blade and bringing it through a modified Motion Builder pipeline to get back into the Unreal Development Kit to show the basics of Motion Capture in a game engine.

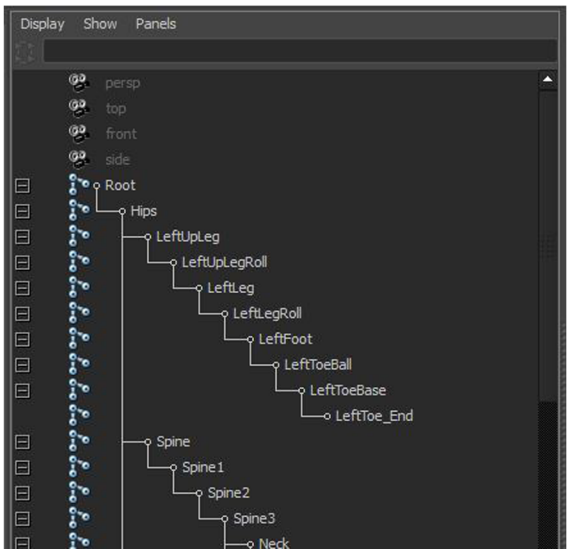
**It is assumed that the guides on Vicon, Motion Builder, and Importing Character Data have already been read.

1. Preparation:

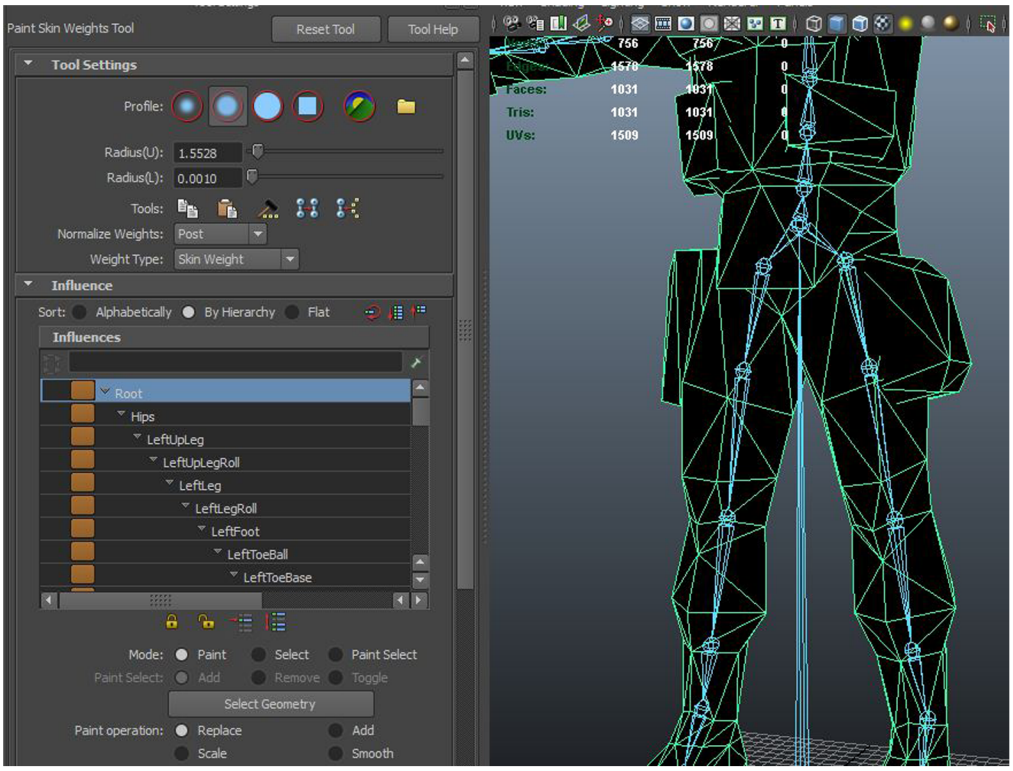
(The takes that are performed in Blade and then cleaned up in Motion Builder need no additional steps of preparation.)

In Maya, you have your character in the T Pose with joints already in place. Make sure that the joints are the same setup for taking a regular Maya character into Motion Builder (ie: Hips being the top level of the hierarchy and all of the joints are labeled so that they will import correctly into Motion Builder). You now need to add an additional joint before you can actually skin the character. UDK relies on an additional joint for animated meshes (named skeletal meshes in UDK) called the Root joint. This allows for translation input from player control if need be, but UDK can also allow Root movement if this will just be a skeletal mesh placed in the level and not actively controlled by a player.

In Maya, create a joint that is above the hip joint in the hierarchy. Make sure to place it at 0, 0, 0 world space coordinates, otherwise you will run into problems later.



After smooth skinning the model, you must paint out the weights of the Root joint completely. While it may not appear to be a problem in Maya or Motion Builder, UDK will default influenced pieces of geometry back to the root joint when it begins moving. In recent versions of Maya, the last color of the painting weights ramp (primarily black) automatically reassigns the influence. Because of this reassignment, painting weights with the new Root joint can be particularly troublesome, so make absolutely sure to completely take influence away from the Root joint.



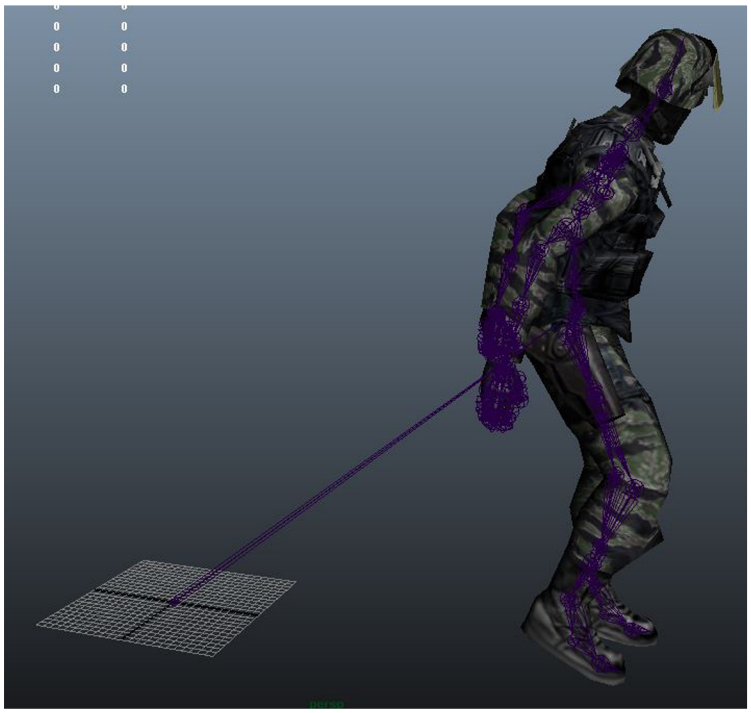
After you finish painting the remainder of the weights with the regular methods, you can the export the scene as an .FBX file type.

2. Motion Builder:

Once inside Motion Builder, merge the scenes the same way as you normally would without the take on the Maya character. When characterizing the data this time, drag the character asset onto the hips and not the Root joint. Motion Builder will get confused with the naming terminology if placed on the Root joint, and entirely disregard the rest of the skeleton.

Now that the actor is characterized, continue following the steps for taking a character through Motion Builder regularly. You'll notice that when you play the animation it will look as

though the character is tethered to the ground by its Root joint while it moves, and this is the expected result (it will look the same in Maya when you import it as well).



When adjusting the control rig, ignore the Root joint and instead fix the animation on the other levels of the rig. After it is cleaned up and you apply the skeleton, the Root will remain where it was. Save the file out as a .FBX and continue into Maya.

3. Back into Maya:

Import the new .FBX and make sure the animation runs smoothly. If it looks good, you are ready to import into UDK with one of two options: .FBX and .PSK/.PSA. Either one will get your animation and models into UDK, but they consist of slightly different pipelines and advantages/disadvantages.

.FBX Importation:

UDK can read raw .FBX files and create a skeletal mesh and animation based off of the data. The downfall here is that the import has a greater chance to fail, but it offers a much quicker solution of bringing in the data. This is the superior option if you only have one animation that needs to go into UDK.

.PSK/.PSA Importation:

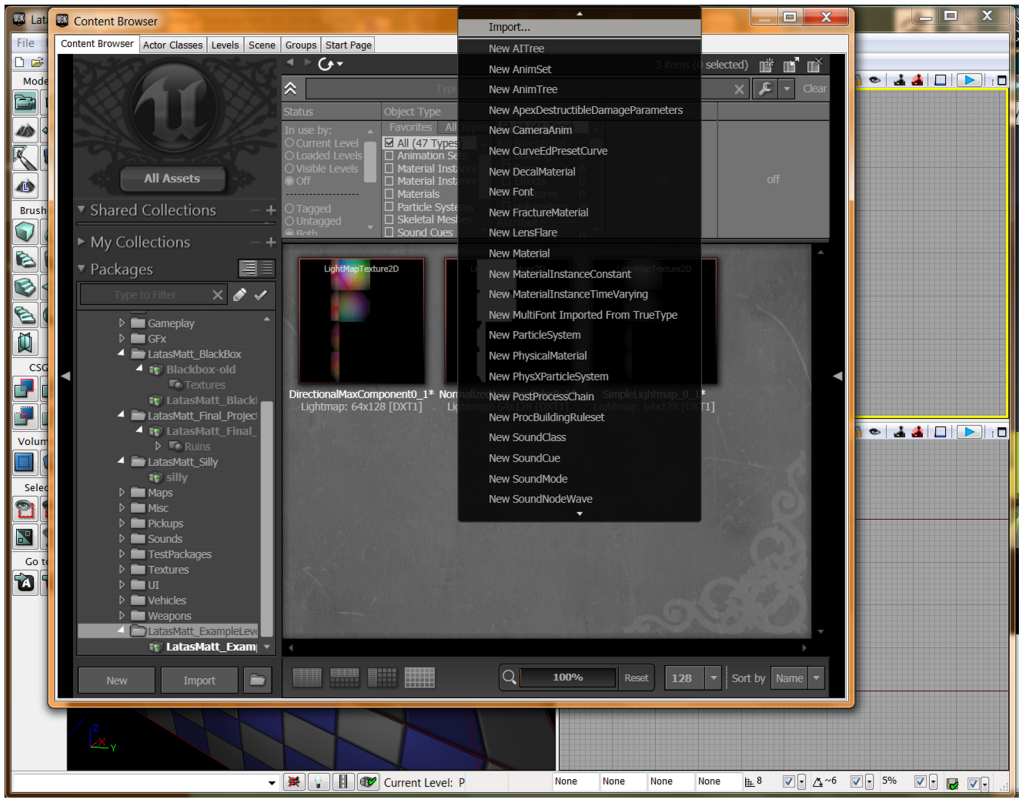
Done through a plugin called ActorX for Maya (and Max), this file type splits up the data into skeletal meshes (.PSK) and skeletal animations (.PSA). It is a slightly more involved process, but the advantage here is that you have individual control of the animations and meshes with joints. If you have multiple animations for one character that you're driving, then this is the superior method.

4a. Option 1 -- .FBX Import into UDK:

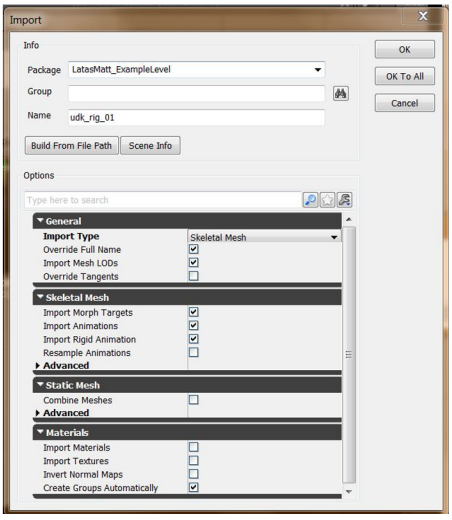
With this option, the first step is simple, and that is just to keep the .FBX file that you just opened handy. Place the file in your level's package folder for easier importation.

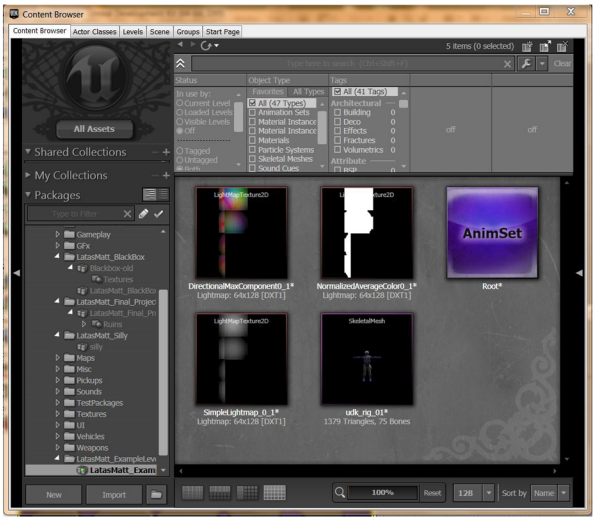
ex: C:\UDK\UDK-2011-03\UDKGame\Content\LatasMatt_ExampleLevel

Inside of UDK, open up your .UDK file and point your Content Browser to the folder that your level is under. Right click in an empty space and then click import at the top.



UDK will then ask where you want the file stored, what group you want it under (optional), and what you want to call it, as well as a few other file specific options. In the case of the .FBX, the options don't need to be tampered with (with .FBX, you don't need to assume Maya coordinates as you normally would when importing .OBJ files for instance).



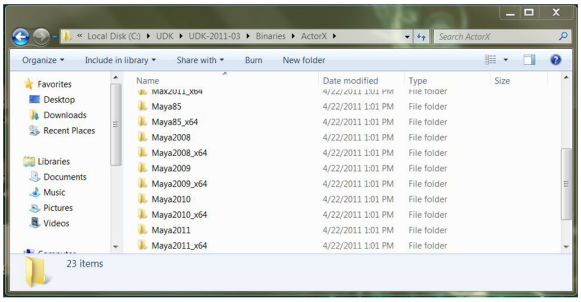


In your folder in the Content Browser, you will see that UDK has created two new assets. One is a skeletal mesh and the other is an Animset, which were both derived from the one file. Make note though that UDK will automatically name your animation Root. When importing multiple .FBX files, it is imperative that you rename them, otherwise UDK will overwrite the preexisting Animset.

4b. Option 2 -- .PSA/.PSK Import into UDK:

With this option, you need to do some additional work before leaving Maya. First of all, the ActorX plugin must be installed. On Monty machines this is automatic, but on a home machine you first must derive the file from the UDK folders. After downloading UDK, they provide you with a great deal of plugins for various software packages. Make sure to carefully choose the right one for the software you are using, plus if you are running the program in 64 bits or 32.

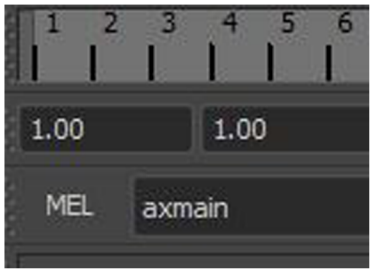
ex: C:\UDK\UDK-2011-03\Binaries\ActorX



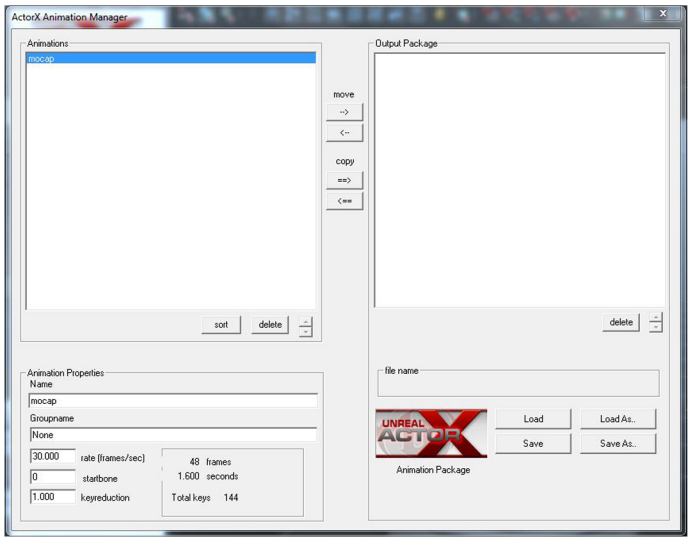
Copy the .MLL file they provide and place it in Maya's plugin folder. When Maya is loaded, make sure to activate the plugins from Maya's plugin manager, otherwise this won't work.

ex: C:\Program Files\Autodesk\Maya2011\bin\plug-ins

Now you can begin creating your .PSK/.PSA files. Begin by importing the .FBX file like usual and making sure it looks alright. The next step is to go to the MEL command line at the bottom of the screen and type in "axmain." Typing "axmesh" will open ActorX as a modified .OBJ exporter, whereas "axmain" opens ActorX as an animation and skeletal mesh exporter.



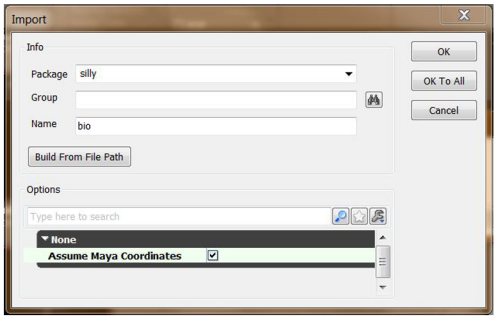
Browse to the location you'd like to save the .PSK file and name it, then press Save Mesh/Refpose. Now we just need to save the .PSA file. First digest the animation. Maya will quickly scrub through it. Then go into the animation manager from the plugin and rename it. This can be finicky, so if the name doesn't show up, click the top of the Animations column anyway, because the file will likely be there. Rename it and then click sort to get it to show up. Click on the arrows to move it to the output package and then save it out.



Just like the .FBX, place the files in the folder you're currently working in.

ex: C:\UDK\UDK-2011-03\UDKGame\Content\LatasMatt_ExampleLevel

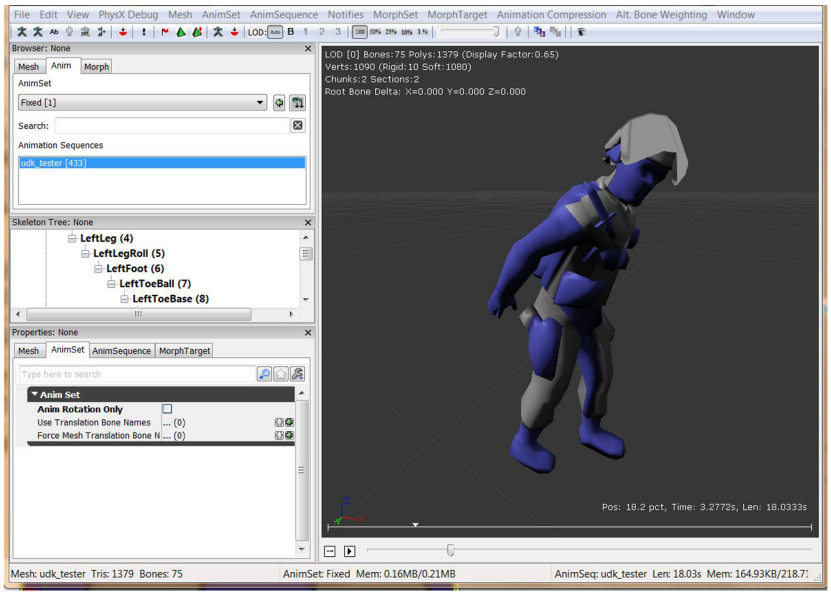
You will open it up similarly to the .FBX in that you must go to your folder in the Content Browser and click import. When importing the .PSK file, make sure to check the box to assume Maya coordinates, otherwise the mesh will be sideways (Maya treats the Y axis as up, and UDK treats the Z axis as up). This will create your skeletal mesh asset.



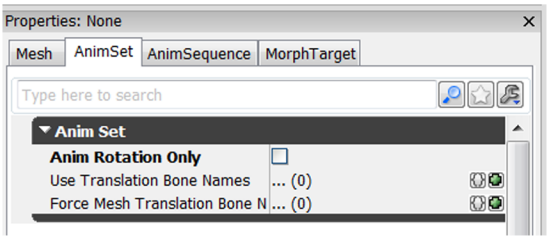
Do the same for the .PSA file and it will create an Animset for that animation file. Unlike the .FBX file, this one was named in Maya, and does not have the Root naming problem.

5. Checking and Activating Animations in UDK:

Save out your level and package files and then double click the skeletal mesh in the Content Browser, bringing up the Skeletal Mesh Editor. You will notice that there is now a tab for "anim" in the top right. Click that and select your Animset file from the drop down menu. The character may snap into place a little as the motion is applied to them. To test it, click the play button on the bottom of the Skeletal Mesh Editor and see how it turned out.

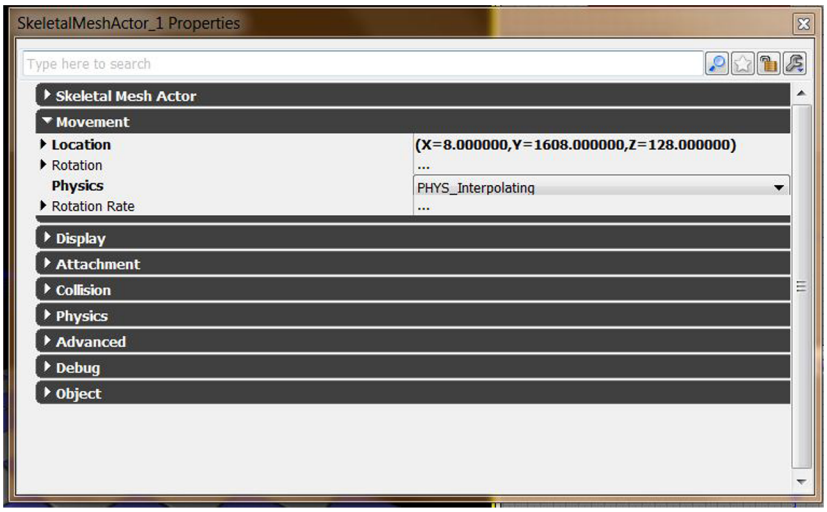


There are a few ways to adjust this, but for animations where you wanted the mesh to move and not just rotate, you can uncheck the box "Anim Rotation Only" in the bottom right under Animsets.

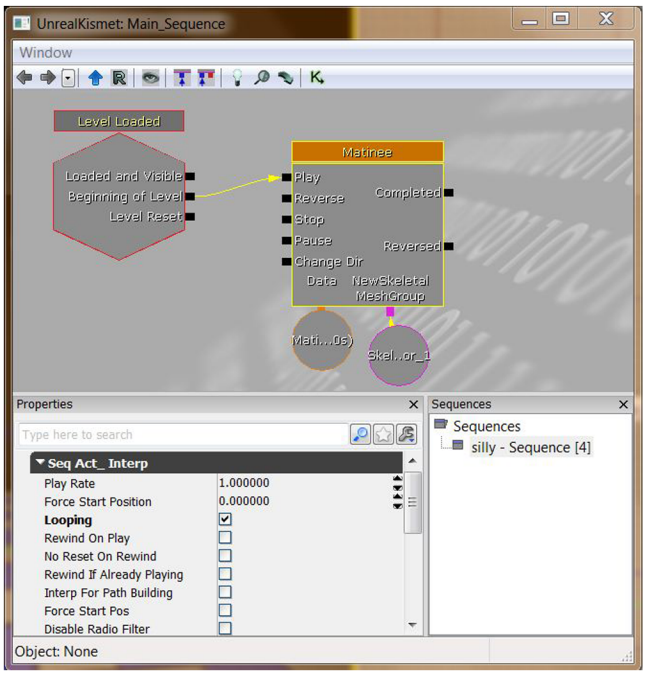


There is an alternate way to apply and import the motion into UDK. An Animset asset in UDK is a collection of animations that are all grouped together but can be accessed individually through Matinee. The alternative way to apply motion to a model is to just import the .PSK from before, but do not import the .PSA just yet. Instead, create a new blank Animset in your package, and name it something. Now in the Skeletal Mesh editor, go up to file and click import .PSA. You will import the motion file and set it directly to the newly created Animset. You can import as many .PSA files into a single Animset as you'd like. The advantage here is that you can stack many animations for one character in the same set. This is the superior method of importing animations and utilizing them for a single skeletal mesh, but for the sake of the tutorial, we will only deal with the one Animset.

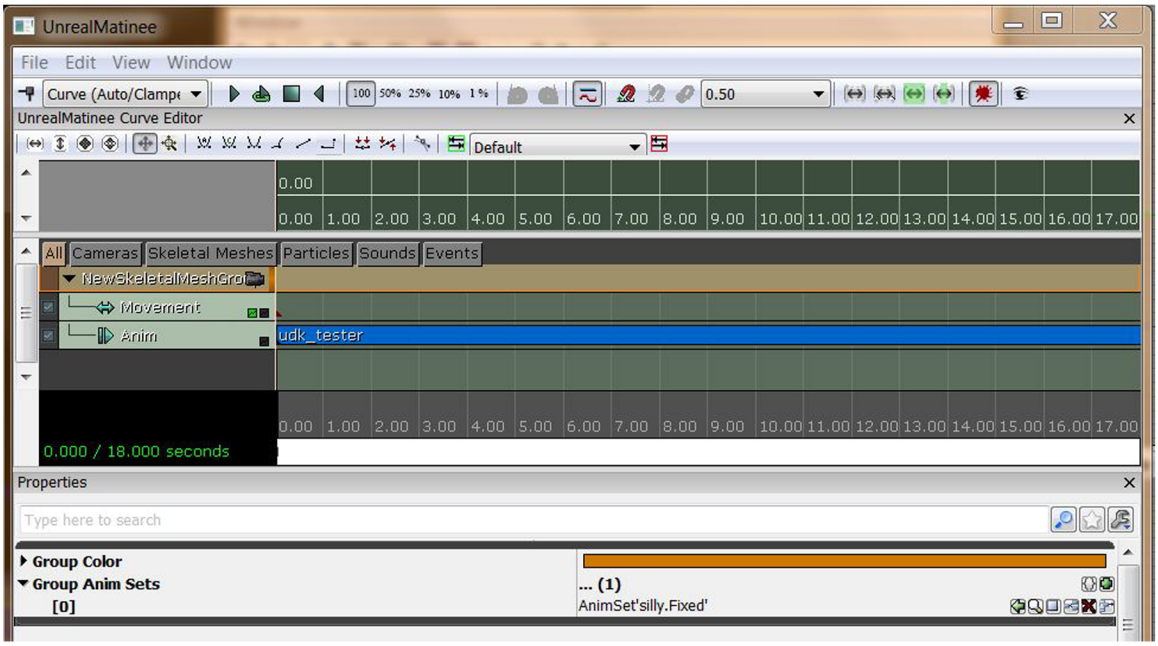
Now that we know our animations are working properly, we just have to activate them. There are myriad ways to activate your animation in a level depending on the situation, but the most important aspect of this is that the Matinee sequence is correct. Begin by dragging your skeletal mesh into your level somewhere (preferably with enough room to move around). Double click (or press F4) on the skeletal mesh, and under his Movement Tab, set the Physics mode to PHYS_Interpolating, otherwise UDK might have issues running the animation.



Go into Unreal Kismet and set up a basic sequence for a Level Load Condition to drive the skeletal mesh animation. Create a new Matinee event and plug Beginning of Level into the Play control of Matinee.

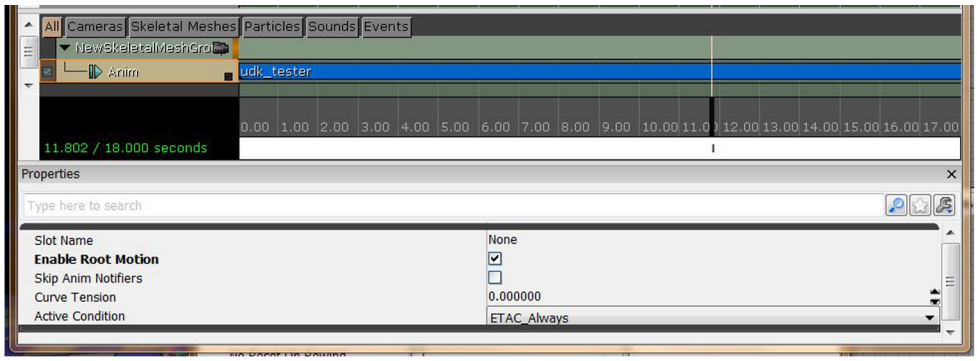


In Matinee, this step is the one that is often overlooked, but is vital to running the animation. Right click on the left hand pane and select a new skeletal mesh group. UDK will automatically give you the tracks for movement and animation. Click on the group itself to see its properties appear at the bottom. Where it says Group Anim Sets, click on the little plus icon to add another variable to the array.



Navigate your Content Browser to the Animset you want on the skeletal mesh and select it. In Matinee, press the green arrow button by the new array variable and you will have the Animset selected for use by this sequence. Click on the Animset track and press enter once, and UDK will ask you which sequence in your set you would like to use. Select the one you want to use and UDK will automatically set the keyframes for you. Be sure to drag the bar to the full length of the animation or it won't all play.

Optional side note: If you didn't enable Anim Rotation earlier in the Animset's properties, then you can individually tell Matinee to allow root motion in the anim track. If you choose this, you must also delete the movement track that was automatically generated with the file.



When you load the level, your animation should load immediately. If it was too quick, you can set the Matinee's properties to looping in Kismet to see it more clearly. Otherwise, congrats! Your motion capture data works in UDK!