Animation Methods in Autodesk® 3ds Max®

Chris Murray – Autodesk Media and Entertainment

DG5918-V

Learning Objectives
At the end of this class, you will be able to:

- Effectively understand where and how to change animation controllers in 3ds Max
- Combine controllers together for sophisticated animated sequences
- Use constraints to accomplish precise movement on splines.
- Create wired relationships between object attributes to control movement and parametric object properties.

About the Speaker

I'm trained as an animator (3D variety) but consider myself a creative technologist. My daily role is a technical "evangelist" for Autodesk; I create marketing deliverables for my specific product expertise--3dsMax.

I've done all kinds of production work from games/simulation, broadcast, web3d, and medical. At the same time, I've maintained a lifelong relationship with learning. As a consultant trainer I have provided key training for numerous clients from Fortune 500's, Game studios, schools and others.

At one point I took the time to author a book, Mastering 3D Studio Max. Additionally, I've also authored numerous tutorials and end-user guides. I have a Masters in Fine Arts (MFA) in Computer Animation from Miami International University of Art and Design.

Email: chris.murray@autodesk.com

Follow me on Twitter: @chrismmurray

Link In to me: http://www.linkedin.com/in/chrismmurray (Just mention AU2012 in your request)
Best practices for character rigging

Character Rigging

Overview
This document describes possible work flows and usage of tools in 3DS Max for rigging character in Game, TV and film production.

Character pipelines vary in may different ways based on what the needs of the production are. This document should be used as a guild line for setting up a character based production.

Modeling
Modeling a character can be done with many methods. 3DS Max has focused the tool set on poly modeling as the main organic modeling method using TurboSmooth to tessellate the geometry to get a smoother result.

Edge Loops and Rings
Creating a great looking model using editable poly and edit poly can be achieved in several ways. Creating a great looking model and having it easily deform with a character rig is more demanding.

When modeling a character edge loops should try to follow the natural flow of curves in the characters joints and volumes. It is also best to take into consideration that the final character will have to deform with the character rig so adding extra edge loops that are not necessary to achieve the form of the model but are necessary for the deformation of the character should be taken into account.

Symmetry
If the design allows, creating characters with Symmetry will speed up work flow with rigging and skinning a character. If time is short on a project and animation has to start right away this should be considered. Mirroring skin weights and Skin Morph shapes means that you only need to work on half of the character. If the character will be symmetrical then using the symmetry modifier is an easy way to ensure that the left and right sides are identical. Be sure to set the threshold as low as possible so that it doesn't weld vertices together that are not on the center line.

Collapsing to a Mesh for Speed
Edit Poly and Editable Poly models are slower in the view port then an Editable Mesh models. Speed has been increased dramatically in Max 8 for the view ports however this is still a good habit to be in. Once modeling is complete the model should be converted. No data is lost in this conversion and if any more modeling needs to be done the model can be converted back to a poly. Never collapse TurboSmooth or MeshSmooth into the model since this will increase the amount of vertices that need to be skinned and also slow the speed of the rig down for the animators. Modifiers that should be collapsed are all mapping modifiers, symmetry, edit poly modifiers and any other parametric modifiers that will not be animated. Keeping the stack clean makes for faster models and rigs and cause less problems in production.
**XRefs**

XRefs are a useful way to share the model as it is being created. Completed characters can be referenced into scenes using Object Xref. It is best when using this feature that the modifiers be merged into the scene and the materials are left as references. Modifiers are needed for the animation process so they need to be in the scene so that parameters can be changed. Materials are best left as references so that any changes to the materials on the character will update across the scenes the character is used in.

**Referencing to rig**

Once the main shape and joint placements of the character are complete the rigging can begin. XRef the character model into a new file and start the rigging process. Use Object XRef since this will allow access to the mesh so that modifiers can be added. This referencing can be left in place to keep the elements of the character broken up into separate files making it easier to manage the assets.

**Rigging**

Game rigs and feature film rigs have become far more similar in the past few years. Game engines can handle more data than they ever have been able to and technology for creating games is starting to show up in productions that don't have to deal with real time but want to work faster.

**Animation Modifiers**

Most modifiers allow a certain amount of animation to be performed on them. Here are a selection of modifiers that are used most in creating character rigs and animation. The modifiers listed have been created with character and creatures in mind but can be used in many different ways.

<table>
<thead>
<tr>
<th>Animation modifiers.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skin</td>
<td>Skin is the primary deformation modifier for binding models to skeletal structures. It uses two methods depending on the level of detail that you need. An envelope based system that is very quick and easy to setup and adjust and a weighted vertex system that allows for painting of vertex weight on a per bone basis.</td>
</tr>
<tr>
<td>Skin Wrap</td>
<td>Skin Wrap can be used as a secondary deformer to skin or as a primary deformation method on an object. Skin Wrap differs from Skin in that it uses the vertices of other objects to deform the mesh.</td>
</tr>
<tr>
<td>Skin Morph</td>
<td>Skin Morph is a morphing solution that uses joint angles to drive target shapes. This used to be done by using Morph targets driven by expressions. Skin Morph makes this easy by allowing the user to create shapes to correct joints or create muscle bulging and then mirror the result to the opposite side of the character.</td>
</tr>
</tbody>
</table>
## Animation modifiers

<table>
<thead>
<tr>
<th>Animation modifiers</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flex</td>
<td>Flex is designed to handle simple giggles in the surface of a mesh. Flex can be placed above Skin in the stack and soft selections can be passed to it so that only small areas of the mesh are affected. Springs can be added to help keep the volume of the object and give a more realistic feel to the effect. Although Flex is a dynamic solution it shouldn't be confused with soft body dynamics or cloth. Flex doesn't allow for complex collisions but has the advantage that it doesn't need to be calculated.</td>
</tr>
<tr>
<td>Point Cache</td>
<td>Point Cache will capture any vertex animation to a file. This allows for turning off all other animated modifiers in the stack which will dynamically increase the speed of the scene. The Point Cache file can be loaded onto other meshes with the same topology. This tool is often used for baking animation before render time.</td>
</tr>
<tr>
<td>Morpher</td>
<td>Morpher is primarily used for facial animation but can be used in many other applications. Morpher allows for the model to be remodeled into new shapes and then applied back to the original and animated. Most facial work is done with way.</td>
</tr>
<tr>
<td>Attribute Holder</td>
<td>Attribute Holder is a blank modifier where custom attribute UI items can be added. UI items can be added to any modifier or base object but this can make for cluttered interfaces. Using the Attribute Holder modifier starts you with a clean UI.</td>
</tr>
<tr>
<td>Linked Xform</td>
<td>Linked Xforms allows for vertices to be parented to other objects in the scene. Soft selections can be used to soften the effect.</td>
</tr>
<tr>
<td>Spline IK Control</td>
<td>Spline IK Control modifier was designed to be used with the Spline IK Solver but can be used on it's own on shape objects instead of Linked Xform. It has the advantage that it back transforms the vertices and is quicker to set up then Linked Xform.</td>
</tr>
</tbody>
</table>

## Controllers

Animation controllers are the heart of 3DS Max for controlling animation. Knowing when to use which controller to get the most out of your character rig takes time and an intimate understanding of what they can do.

The main controllers that need to be understood to create character rigs are.

<table>
<thead>
<tr>
<th>Controllers</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controllers</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| List Controllers     | *Allows for other controllers to be layered so that more than one type of input can be controlling a single animation track.*  
                         *Nested list controllers can also be created to layer layers of controllers. Any type of controller can be placed in a list controller.* |
| Parameter Wires      | Allows only one input  
                         Max Script can be used to create expressions around the input value.  
                         Wires can also be bidirectional by saving the animation in one track but allowing both the master and slave to be manipulated. |
| Expression controllers| Expression controllers are very fast but have a limited language.  
                         Any number of inputs can be used but have to be either a float value or a point 3 value. Nodes are now also allowed in expression controllers in Max 8. No other types are allowed.  
                         Expression controllers don’t use the Max Script syntax but instead have their own language. Max script can be used to create expression controllers. |
| Script controllers   | Script controllers are very flexible because they use Max Script as their language. Script controllers can have an unlimited number of inputs or variables that can output and control just about anything Max script can do. For this reason you have to be careful that you are not trying to force a script controller to exceed what it was designed for.  
                         Any max script calls can be used with any number of inputs of any type available in Max. |
| Constraints          | Constraints are direct ways of setting up relationships between objects in the scene.  
                         They are fast to calculate and fast to set up for simple aligning of objects. Constraints can have multiple target objects that can be weighted for customizing the result.  
                         There are seven types of constraints. |
| Instanced controllers| Instancing controllers is a direct way of connecting one track of animation to another. It is very fast because it is the same controller on both tracks.  
                         This also produces a kind of bidirectional connection. Both tracks would show the keys. |
### Controllers

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limit Controllers</td>
<td>Limit controllers will limit any keyable float controller values from going higher or lower then what you need during animation. This is a great way to set ground levels so that feet don't pass through the floor or to limit mechanical systems for rotating to far.</td>
</tr>
<tr>
<td>Spring Controllers</td>
<td>Spring controllers are a fast way of adding dynamics to the position of an object. World X,Y and Z values can be assigned different values to get the solution that is needed for the animation. Spring controllers are a real dynamic solution and should be turned off while animating so they don't slow the rig down. They are also dependent on scene length and can get very slow at higher frame ranges.</td>
</tr>
</tbody>
</table>

### Helper Objects for rigging.

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point Helper</td>
<td>Point helpers are non-rendering nodes used for aiding in animation and character rigging. They have four display options that can be mixed together to help differentiate one set of helpers from another, or you can turn them all off so they don't show up at all.</td>
</tr>
<tr>
<td>Extract Transform helper</td>
<td>Extract Transform is a point helper that aids in rigging by extracting transformation data from an object. Data such as local and world rotations, local and world positions as well as distance between objects and local bounding box information are returned. These values can then be applied to other objects in the scene via one of the several different controllers like parameter wires.</td>
</tr>
</tbody>
</table>

### Twist Bones

Twist bones are used to simulate the skin sliding over the surface of our muscles and bones. It is necessary to create this effect with extra bones to help keep volume in the deforming bones. Twist bones are mostly used in the arms and legs of characters especially in the forearms where there is a large amount of twist along its length. Twist bones are easily set up using the extract transform helper and parameter wires.

### Stretchy Bones

Stretchy bones can be used to simulate the sliding of skin as well the the stretching and compressing of muscles. Stretchy bones have the option of using squash and stretch and will automatically change volume as the length of the bones changes. Using setup like this can make is very easy to mimic the
complex structures that drive a living creature. Stretchy Bones are also used for cartoon characters that
need to squash and stretch or for limbs that the length needs to be cheated to get the desired pose.

Off Loading Animation
Character rigs can become very complex with many controls and deformation systems to drive the final
mesh. In some cases it is best to break the rig into parts so that each has a unique task to perform.
Creating an animation rig and a skinning rig can reduce the amount of overhead that is needed to perform
the two tasks.

Animators don't always need to see the final deforming mesh. If the deforming mesh is very complex and
requires many layers of deformation to produce the end result it will be far to slow for an animator to work
with.

Create two rigs instead, one that is just for the animators and doesn't have all the deformation bones and
layers and a second that doesn't have controls but has all the objects in place for deforming the
character.

Use the Save/Load animation tool to make it easy to save the animation from one rig and apply it to
another.

Custom Attributes
Custom Attributes can either be created with the Parameter Editor or through Max script. A custom
attribute can create custom control interfaces for driving character rigs or just for a place to store data for
use by other tools. Using just the Parameter Editor allows for many of the Max Script UI elements to be
created and added to object in the scene, modifier, material or even a controller. Using Max Script and
writing the custom attributes yourself allows for any UI to be created that can be created in a rollout
floater or a dialog. You can also add any type of code to the custom attributes in the form of functions,
event handlers and just about anything that Max Script can to so extend what custom attributes can do.
For more information about what is possible look up Scripted Plugins in the Max Script help.

Keeping Rigs Fast
Character rigs and mechanical rigs can get very complex. Because Max allows for just about anything to
be done it is possible to do things that will create a character rig that is slow and hard to use. Here are
some of the things to look for to keep the speed of your rigs as fast as possible.

Choosing the right controllers for job is an important consideration when building rigs. Here is a
breakdown of what they do and how best to use them.

Constraints
Constraints are a very fast way to connect transforms of object to others. There are three categories of
constraints, Transform, Position and Rotation. Constraints cannot control individual float values however
so this limits what they can be used for. Constraints allow multiple inputs that can be weighted to achieve
different results.

Expression Controllers
Expressions controllers are the fastest to calculate of the expression based controllers so they should
used whenever possible. Although constraints have many options and can be used in several different
ways they are limited to position and rotation of objects and are limited in what they can do. Expression
controllers can be used on any track including point three controllers like positions. They also provide a
large selection of predefined functions that can be used in a user defined expression. Expression controllers however cannot perform some of the tasks that constraints can and take longer to setup. Both controllers are easily scripted for automated tasks. Multiple inputs can be used in the form of variables for the expression which makes them flexible for many situations.

**Wire Parameters**
Wire parameters are quick and simple to set up and are best used in areas that one value needs to be connected to another with a simple expression. Wire parameters only allow one input value so they are limited in the expressions that can be created. Wire parameters use Max Script syntax but don't accept all functions and commands. They are fast to setup and are easily scripted. Wire parameters are not as fast as constraints or expression controllers but if they are used without complex expressions in them they shouldn't slow your rigs down.

**Script Controllers**
Script controllers are very fast to calculate. Script controllers are like expression controllers now where you add variables and connect those variables to the tracks of animation that you want to use in your expressions. This means that script controllers don't have to search the scene for the nodes that they will be referenced based on naming so changes to names on scene objects will not affect them. Nodes as well as tracks can be referenced as a variable and any Max script commands and functions can be used within them. For this reason script controllers are the most flexible of all the controllers but can still cause slow rigs if attention to good coding practices is not followed.

Controllers are calculated many times per frame so keeping the expressions as short as possible can help to keep the speed of the controller as fast as possible. Avoid try()catch() statements for error checking code as this can be very slow. Avoid executing strings to build command lines or recreating variables over and over. Some functions in Max script can be slower then others, complex math functions can slow down a script controller considerably if not handled correctly.

Script controllers don't stop you from trying to control an object that the script controller is not on. Although this is possible is shouldn't be done. Often the expression will not calculate at render time or other errors will happen that will break the script.

**Reaction Controllers (Reaction Manager)**

Reaction controllers are set up through the Reaction Manager. These controllers are fast to calculate and allow for a more hands on feel for setting up the effect the controller will have one the values. Only one input is allowed and expressions can't be used. Reaction controllers also use a function curve to control the rate that the output value is affected by the input value. This helps in creating a very controllable solution.

**Game, TV and Film Rigs**
Game, TV and Film rigs don't differ as much as they used to. Computer speed and game consoles have brought game rigs to new heights and only differ now in a couple of areas. Feature film rigging that has traditionally been the hardest is now learning form the game market how to find news ways to rig characters.
There are different considerations to take into account when rigging characters and objects for productions. First we will have a look at the major considerations for game rigs.

**Game Rigging**
Character rigging for gaming has changed the most over the last few years, advancements in game consoles and engines have allowed for far more to be past to the game engine. Character rigs don't have to as simple as only the game can handle, rigs can be as complex as any feature film rig to drive the base structure of the character. What is exported to the game is what needs to be considered so the rig needs to be designed so that only the needed bones are in the correct hierarchy for the game engine and there are not more then it can handle. Skinning of the rig also needs to be taken into consideration, game engines will have a bone per vertex amount that limits what can be achieved compared to rigging for non-real time applications. Usually a limit of 3 to 4 bones per vertex is the limit but the amount of bones can vary a lot depending on the game engine.

One of the biggest ways that games rigs are limited is in the way that they can be deformed. Deformation needs to take place at the object transform level and not at the vertex level. This means that tools like Skin Morph and Skin Wrap and other deformation methods like Flex or FFD's can't be exported to games usually. Support for vertex deformation other then the Skin modifier would usually be done in game and not exported.

For final export to the game engine the transforms of the nodes that are used in deformation of the character's mesh are calculated and passed onto the bones in the game engine. It is also possible to remove any unneeded nodes that are in the hierarchy but are not used by Skin.

**Broadcast Rigging**
Broadcast rigging can be one of the hardest to right. It isn't possible to know what the characters will have to do from one show to the next for TV series, and TV commercials are done so quickly there isn't always time for adding needed features that will make animation easier.

**TV Series**
TV series rigs have to be created with as much flexibility as possible because it isn't always known what the character will be doing form show to show and season to season. The rigs also have to be as simple as possible so there is the least amount of problems with them. As always time and money is short so a balance has to be struck that will meet the needs of the production and fit the budget. How ever a series can take a considerable amount of time to generate just one season so rigs can be updated and new tools can be created to further enhance what the rigs are capable of. Reuse of character rigs should also be taken into consideration, character rigs can be automated using Max Script as well the methods that are used to create the rigs can allow for re-purposing them from one character to another with the least amount of re-building involved.

**TV Commercials**
Character rigs for TV commercials can very greatly. Usually it is known what the character will be doing for the commercial unless there will be a series of them that have not been written. This means corners can be cut to achieve the necessary effect of the character with the least amount of work. For instance, if a character is only ever seen from the waist up modeling and rigging of the character from the waist down doesn't need to be done. Also the part the character plays in the commercial might only require a limited amount of movement so advanced deformation methods might not be needed.
Feature Film Rigging
Feature film character rigs will often be the most complex; there is usually more time and resources to allow for complex setups and tools to be created for working with the rigs and characters. Because of the complexity of character and the need for advanced dynamic simulations for hair cloth and muscles, feature rigs will often be created with a layered approach. What this means is having several different versions of the character for the stages of production.

At first glance it can seem as though it is more difficult to work with rigs using this method as it does increase the steps involved from the animation to the final rendered version of the character. Because of the complexity of a character like this a single rig that carries all the necessary elements would be very slow and prone to errors and problems in a large production.

Animation Rig
For the animators a very fast rig is needed so that optimum view port speed and play back can be achieved. The animation rig usually consists of a cut up version of the character mesh so that no deformations are taking place in the scene that would slow it down. Animators also need an easy to use set of controls for the character with tools for helping the animation process accessible with the least about of work.

Deformation Rig
For deforming the character mesh a more complex setup is needed to handle muscle deformations, joint corrections and simple dynamic simulations. This rig however doesn't need any of the controls for the animator so it can be simplified by removing them and only having an FK joint system. The animation from the animation rig is saved as world transforms and loaded back on to the deformation rig. Another advantage of a system like this is the animation can be loaded onto different version of the deformation rig depending on the needs of the scene and what the character will be doing.

Simulation and Render Rig
Simulation and render rig is the simplest it can be. Baking the animation off the deformation rig using the point cache modifier and loading it onto a mesh that has no rig simplifies the character and allows for cloth and hair simulations to take place without the over head of bones, IK solvers, skin and expressions having to drive the mesh first. Often passes on simulations will be done the same way if hair needs to collide with cloth or other dynamics. Doing a pass on one dynamics simulation then baking the result before doing the next pass will speed up the simulations and reduce any possible problems. Once all dynamics are generated all the mesh objects that will be rendered can be baked with a point cache modifier reducing the processes needed for rendering the final image.

Autodesk [and other products] are either registered trademarks or trademarks of Autodesk, Inc., in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product offerings and specifications at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document. © 2012 Autodesk, Inc. All rights reserved.