Mechanical Beasts of Burden

Paul Neale PEN Productions

Code DG1989

Learning Objectives

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

- Set up and rig complex mechanical rigs

- Use MAXScript to track velocities and acceleration of objects in Autodesk® 3ds Max® software

- Prepare and identify objects for rigging

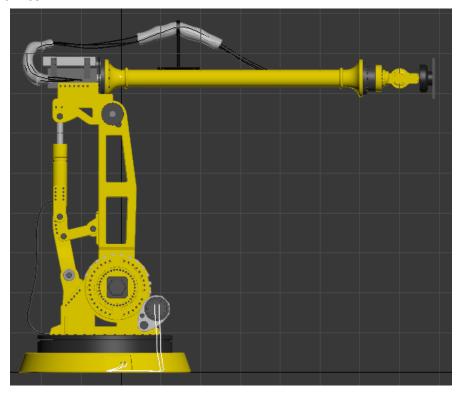- Problem-solve technical issues associated with rigging

About the Speaker

Paul Neale has been internationally known in the 3D Animation industry for almost two decades. His extensive involvement as Senior Director of Research and Development and Art Director of 3D has encompassed areas in TV series, feature film, special effects and high-profile games. Paul specializes in character rigging and modeling as well as writing plug-ins and scripted tools for system, software and production needs. In addition to his industry experience, Paul has been an Ontario College Professor for fifteen years where he brings his knowledge, professionalism and passion of 3D to his students. In 2008, Paul received Autodesk Masters Award for Contributions to CG Artistry. Paul Neale has been a Presenter for multiple Siggraph Master Classes as well as a Master Class at GDC. He has represented Autodesk as a regular Guest Speaker at trade shows and special events. Paul Neale has trained numerous companies over the years including Walt Disney Studios, UBIsoft.

Email: paul@penproductions.ca

AUTODESK **UNIVERSITY** 2013
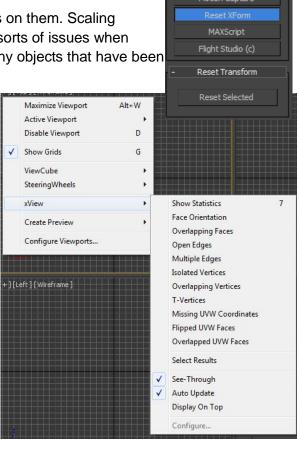
Setting up the Mesh:

Settings up and working with the mesh is an important step that needs to be addressed before any rigging will take place.

1. Check logical pivots in the mesh and realign if necessary. This can help with set up of joints and placing natural pivots. Also asses axes each pivot have freedom in and understand how the system is meant to work in the real world.
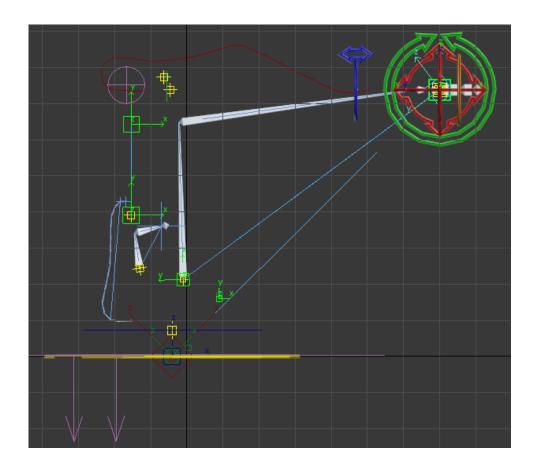
2. Check all objects for having scale values on them. Scaling objects at the object level can cause all sorts of issues when using rigging tools and animating rigs. Any objects that have been scaled need to have their transforms reset.

3. Attaching and or Grouping objects that don't need to have separated animation on them.

4. Check all mesh objects for errors that might affect the end animated result. Use xView in the viewport to display where issue reside on the model. One big issue that I see all the time are open edges where those edges should be welded. When skinning characters this can cause major headaches and make it impossible to smooth skin weights without creating holes when deforming.

Choosing the setup needed for animation:

To be able to setup a rig that will be useful in animation it needs to be first determined what are the most important issues that will arise and what will be the best work flows. In the current robot I determined that the important issue when working with this rig would be that the head could be positioned at the objects that needed to be picked up and worked with. With this in mind the best solution is to start with an IK system.

IK Systems have two basic options for solvers, HI Solver and HD Solver but the best one to use is the HI IK Solver since it is an History Independent solver and will not slow down over time. The HD solver is inaccurate as well as History Dependent which will make it slow down as the time line gets longer as well as create different solutions for any one set of values because it is reliant on what values came first.

HI Solvers have an issue that we will have to resolve and that is there is no good control over the root bone in the chain as far as axes that will be affected are concerned.
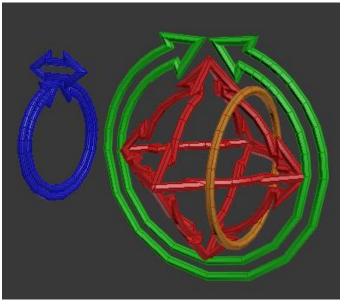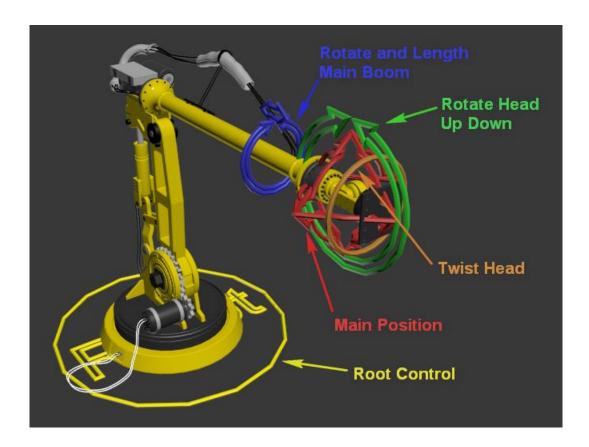
Choose Animation Controls:

Building a rig with one control is just about impossible from an animation stand point. Once you start animating it is always nice if each operation that the rig can do is separated into a unique control. This allows work flows and possibilities that will not work with a single control for everything.

For the head of the robot we will be using four distinct control objects and one at the base.

1. Root Control.
2. Main position control for head.
3. Rotate and Lengthen main boom.
4. Rotate head up and down.
5. Twist head.

These Five controls will allow for all operations that the robot is capable of. The way that the control affect the robot will directly affect the type of animations that are possible with the least amount of headaches.

Root Control:

The root control should control the entire rig for positioning it in the scene. This can also be setup to be able to scale the rig if needed. Scaling rigs should be avoided as the model should be created to scale.

Main Position Control:

The main position control will move the head of the robot and allow the animator to keep the head on target when picking up and placing objects. For this to work the boom needs to be controlled with an IK based system.

Rotate and Lengthen Main Boom:

These two operations have been added to one control since both control the same object and different transforms. Twisting the control will twist the main boom and moving it in and out will lengthen the boom.
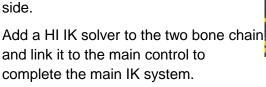
Rotate head up and down:

The rotate up down control rotates the motor on the end of the boom on and up and down motion. If the Boom has been twisted then the angle is in relation to the booms rotation.
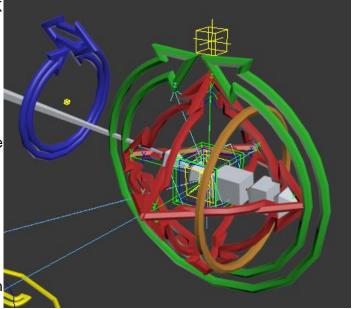
Twist Head:

The twist head control twist the final mountain plate on the end of the boom.

Rigging Methods:

One of the first tricks to setting up the IK solver is to get the base rotation to only rotate on one axis. To do this a point helper is added at the base of the bone system and setup with a Look At Constraint and looking at the main position control. Then use an Expose Transform node to get the rotation of the point helper around the Z axis an wire the base to Expose Transform Local Z axis. The bone system is then linked up to the base and will rotate with it as the main position control is moved side to side.

Add a HI IK solver to the two bone chain and link it to the main control to complete the main IK system.

The up down control needs some extra setup since it can't just be orientation constrained to the position control as it can't rotate on all axes. It does have to stay aligned to the world in and up and down rotation as the IK chain moves up and down. Another Expose Transform is used to extract the rotation of the root bone from the base and then the up down control is wired to it so that it counter rotates and stays parallel to the ground.
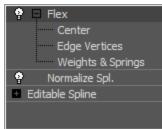
The length and twist control controls x position of a bone before the end of the IK chain.

Dyanmics:

Dynamics have been done in the simplest way possible for this rig using the Flex modifier for all three of the setups. With a few adjustments to Flex it is possible to get a realistic solutions that are fast and are real time in the viewport.

Boom Cable Flex:

The Boom Cable Flex modifier and cables have been setup by creating a single spline that is driven by Flex and then the three cables and padding around them have been skinned to the spline just using the spline as the bone. It has been setup this way for two reasons, first is that it will be faster since there is only one spline needing to calculate the dynamics and second we don't have to worry about the separate cables colliding with each other or passing through the padding that is around them.

The Flex spline is linked to the main boom bone so that it follows the rig. When flex is first added to the spline it will make the entire spine flex from the pivot point. This isn't the effect that we are after as we need to have both ends and the center of the spline fixed and the portions between flexing without stretching to much.

The spline has been created with as few vertices as possible to get the curve right. To ensure that the Flex modifier works well we need more vertices that are evenly spaced along the curve. Add a Normalize Spline modifier and set the Seg Length value so there are enough vertices for Flex to calculate a natural motion. This will vary depending on the length of the spline.

Enter the Edge Vertices SO mode and select a couple verts at each end and at the point where they are clamped in the middle. Vertices that are selected in the SO mode will not be part of the solve.
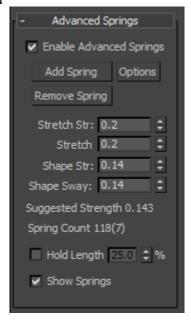
First turn off the Use Weights check box in the Parameters rollout. This will turn off the weights that are set from the pivot of the object.

Change the Euler solver to the more acurate Runge-Kutta 4 solution and set the samples to 4. You can adjust the samples up and down as needed. More samples will tend to make the solution more active and less can make it sluggish.

The Strength and Sway (Dampening) can be played with as needed once the rest of the solution is setup. Strength will make the spline spring back into shape faster and Sway, which should be called dampening will add friction to the solution. The Flex value can usually be left at a value of 1 but can be turned down if needed but not up as that can make the solve uncontrollable.

The trick to getting flex to work well is to set up springs that will help stop the spline from stretching to much. Without springs Flex each vertex will not pull on the others around it and will stretch in all directions.
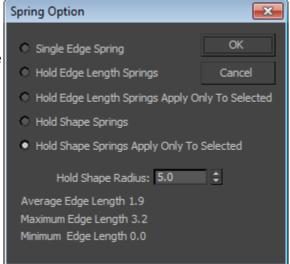
In the Advanced Springs rollout check the Enable Advanced Springs check box and press the Options button. Shape springs work well for holding the shape of the spline. Choose the Hold Shape Springs and set the Hold Shape Radius just a bit higher then the Maximum Edge Length that is listed. Springs will be created between each vertex. If you want more springs set the radius higher. This can create an unstable solution if to many springs are generated. Press the OK button.

At the bottom of the Advanced Springs rollout check on the Show Springs check box and then press the Add Springs button at the top. Inspect the spline and the springs can be seen.

Set the Shape Str (strength) and Shape Sway values to the Suggested Strength as a starting point. If these values are two high an unstable solution will be created.

At this point the time line can be scrubbed and the Flex solution viewed. Adjust the Sway and Strength in the Parameters rollout to get the result that is desired. If the object is moving to fast then turn the Flex amount down. This value can also be animated if needed.

Tracking Velocities.

There are several ways to setup tracking systems for velocity and acceleration, one of the easiest is to use a slider manipulator and a script controller.

Add a Slider Manipulator to the viewport and right click on the Value input spinner and open trackview. In trackview add a Float Script controller to the value track. Once in the script controller all that needs to be added is the node that will be tracked in the variables and then calculations can be made using the position of the object.

The At Time context can be used to grab the position of the tracked object at times other then the current time to be able to get the distance between them.

B
e

Acceleration Rate: 4.09405

Selected

Editor   Edit   View   Curves   Keys

Transform : Position/Rotat
Position : Position XYZ
X Position : Bezier F
Y Position : Bezier F
Z Position : Bezier F
Rotation : Euler XYZ
X Rotation : Bezier I
Y Rotation : Bezier I
Z Rotation : Bezier I
Scale : Bezier Scale
Object (sliderManipulator)
value : Float Script
minVal
maxVal
Custom Attribute

Script Controller : Head_Acceleration\value

Create Variable

Name: RotateUpDown_CNT

Create   Delete   Rename

Variable Parameters

Tick Offset: 0

Variables

F
NT
RotateUpDown_CNT
S
T

Expression

```
v1=at time (f-1) RotateUpDown_CNT.transform.pos
v2=RotateUpDown_CNT.transform.pos
v3=at time (f+1) RotateUpDown_CNT.transform.pos
d1=distance v1 v2
d2=distance v2 v3
a=(d2-d1)*30.0
```

Description

T = ticks   F = frames   S = secs   NT = normalized time

Assign Constant   Assign Track

Assign Controller   Assign Node

Assigned to:
Object: $RotateUpDown_CNT

Save   Load   Debug   Evaluate   Close

l
o
a
t

S
c
r
i
p
t

c
o
n
t
r
o
l
l
e
r

f

r