Rigging in 3Ds Max – Fundamentals

Austin Durose

22/06/15

Contents

RIGGING IN 3DS MAX

Overview

Rigging is an awesome game of confusing puzzle solving. Even through there is complete freedom in creativity and many ways to solve a problem, usually there is only one ultimate solution that would make everything right as rain and fun to animate. Our aim is to create a skeleton structure that has an expressive range of motion and that is easily accessible by the animator – anything we think the animator would want to do with the rig would need to have some form of controller to avoid them snooping around for way to manipulate and (more than likely) mess up our beautiful creation.

I'm approaching this with the aim to tell you what you need to know to get into rigging, with no rubbish fillers, just what I think to be the important stuff. We will not go straight into making a skeleton for a person, but instead give the understanding of what tools we have and how we might want to use them.

The first section in this tutorial will be the 'Fundamentals'. In this section I will aim to describe the different constraint techniques I use. These are very important to understand and will help us create and effective rig to animate with.

Orthographic Vs Perspective

I live in the world of the orthographic. It helps me see things with more precision, and a great feature of Max is that you can Orbit the camera in an orthographic view. When initially creating bones, every point you create will land on the grid space, so make sure you are in a Top or Front view when plotting out your bones.

Schematic View

I find the schematic view window very useful to see what is parented to what, how our bone chain looks and where our control objects are. Keep this open if you can, as it will help clarify what is going on.

Something to note is that bones don't have to be connected by bones. You can simply parent the upper arm bones of your character to their spine bones even though they do not physically touch. Though I wouldn't parent the upper arm bones to the spine. That would be silly. You need a clavicle bone, obviously.

Abbreviations

I may use some abbreviations. Get over it.

LMB - Left Mouse Button

RMB - Right Mouse Button

MMB - Middle Mouse Button

FUNDAMENTALS

Bone Tools

(Animation>Bone Tools)

Keep this open when plotting your bones. In a blank scene only 'Bone Edit Mode', 'Create Bones' and 'Refine' can be selected. This is silly. I can't edit or refine a bone if there isn't one, so I select 'Create Bones' and LMB click in as many places on my grid as I wish. When I am done I will RMB to finish the creation, ending the bone chain with an 'End' bone.

In the 'Bone Tools' panel, play around with the bones you have created and see what happens.

- 'Refine' will break the bone you have just created wherever you click.
- 'Reassign Root' will make any bone in the chain the new root bone.
- 'Delete' just gets rid without connecting bones.
- 'Remove' gets rid and connects the bones.
- 'Mirror' will create a duplicate of the bones you have selected.
- 'Create End' will add an 'End' bone to the bone selected (as long as that node doesn't have a child node in the hierarchy).
- 'Connect' will create a bone from the tip of the bone initially selected to the base of the bone selected, it would also connect the added bones to the original chain.

Bone Edit Mode

'Bone Edit Mode' allows you to move a bone without effecting its children. Create a two bone chain (UpperArm, LowerArm, End). When moving the LowerArm Bone without 'Bone Edit Mode' selected, it will rotate the UpperArm, and keep the LowerArm's orientation relative to its parent. All bones keep their length, but this is not always what we need.

With 'Bone Edit Mode' selected, we can move the LowerArm bones position without it effecting any other bone position. The length and orientation of the selected bone and its parent change to accommodate any changes that need to be made to fit the bone to the model you have.

Gimbal and Freeze Transforms

With any chain of bones, select one and rotate it any way that you like. On the tool bar at the top of your screen you will see the 'Reference Coordinate System' (RCS) drop down. Change it to 'Local'. This will give you clean rotation handles relative to the bone or any other object you have selected. Change it to 'Parent' and you will get the rotation handles relative to the parent bones (if you zero out the X, Y, Z coordinates at the bottom, it will become straight with its parent).

Rotate the selected bone and select 'Gimbal' from the RCS. You will see that the X, Y and Z rotation handles are at strange angles to each other. This is the actual axis used by Max to figure out rotation of that bone relative to the parent. We could Alt+RMB to bring up the 'Transform' Quad Menu and select Freeze Transform. A window would appear to ask if you would like to continue and yes. Yes you would.

You will now see that the Gimbal rotation handles are unified and exactly the same as the 'Local' rotation handles. This isn't imperative at the bone creation stage, but will be very important when creating control shapes for your animator to use.

Constraints Quad Menu

And now for the fun stuff, but before I go into the constraints I use, the easiest way to use these is to make sure they are easy to grab. Every time you use a Rotation or Position based constraint, you would likely want to follow these nine easy steps.

1. Select bone.
2. Go to Motion Panel>Assign Controller
3. Select Transform wanted, ie Position
4. Select 'Assign Controller' and select 'Position List'
5. Open the newly created 'Position List' and select 'Available'
6. Select 'Assign Controller' and select 'Position Constraint'
7. In the 'Position List' attribute, select 'Position Constraint' and 'Set Active'
8. Below that, 'Add Position Target' and select the target you want your bone to be constrained to.
9. Deselect 'Add Position Target', lest you wish to add more targets.

INSTEAD!!!! You could go to 'Customize>Customize User Interface', go to the Quads tab, Change 'Default Viewport Quad' to 'Animation', select a quadrant from the four squares, I use the lower right labelled 'set'. In the 'Menus' list to the bottom left find 'Constraints' and drag it into the 'set' quad menu to the right.

Now whenever you Alt+RMB, you can go to 'set>constraints' and select any constraint you need! Simply select your target and all of the steps above are done automatically!

METHODS

Position Constraint

Create a bone. Select both the bone and end bone and shift drag it to the left to create a duplicate. Call these BoneA, BoneA_end, BoneB and BoneB_end. Select BoneB and position constrain it to the BoneA_end. The bone you have selected will jump to that new position as though it was a child in the chain. However, this is no child! In the Schematic View we can see that these are completely separate chains. BoneB will now follow the position of BoneA_end, but not it's rotation. We can freely rotate BoneB, but cannot move it.

Orientation Constraint

In the same scene, create two points: PointA and Point B. Make them boxes for ease of viewing. Link PointB to PointA, so as you move and rotate PointA, Point B moves relative to it. Position Constraint BoneA to PointA and PointB to BoneA_end. Now as you rotate PointA, PointB only rotates on the spot. Orient Constraint Bones A and B to Points A and B respectively. Now as you rotate the points, the bones will move as though they are part of one chain.

LookAt Constraint

In a new scene, create a new bone and two points. Quick Align (Alt+A) the points, one to the bone and one to the end bone. Position Constraint the bone to its point, and LookAt Constraint it to the second point. As you move the first point, the bone will move, but still point to the second one. As we move the second one, the bone will continue to aim for it. Up nodes

Path Constraint

Create a line and a point for each of the two points already in the scene. Quick Align the lines to the original points, and Path Constrain the new points to the lines. Select the Path Points and Alt+RMB to bring up the Animation Quad. Go to Set>Delete Selected Animation to remove the animation added automatically. In the Path Parameters in the Motion panel (Position), change '% Along Path' to 50.

Link the two original points to the closest path point, and Quick Align them to centre them. Now we can use the '% Along Path' to manipulate the bones, but because the bones are not directly path constrained but linked via other points, we can still manipulate their positions manually.

Linked Xform

Select the lower line and its vertices. RMB to bring up the default quad menu and in the top left quad select 'Bezier' to give the vertices Bezier handles to manipulate the line. Create a new point for every vertex of the lower line and Quick Align each vertex to its relative point.

Select one vertex of the line and in the modifier list add a 'Linked Xfom' modifier. Pick the relative point as a control object. Then add an 'Edit Spline' modifier to the stack and select another vertex, add another 'Linked Xform' and select that vertex's relative point as the control object. Do this for every vertex in the line, and now we can use these points to manipulate the line. This is very useful if we have multiple bones that need to follow a smooth curve, such as in wings.

Inverse Kinematic Solver

In a new scene, create a two bone chain and name them Upper, Lower and End. Make sure there is a slight bend in this. if there isn't, use the Bone Edit Mode to pull out the Lower bone a little, or simply rotate the Lower bone a little.

Selecting the Upper bone, go to Animate>IKSolvers>HI Solver and select the End bone. This has created an IK Chain. In the Motion Panel, you may want to adjust the Goal Display to be able to see the IK Chain point clearly.

Moving the IK Chain point will bend the arm starting from the End Bone, with no effect on the position of the Upper Bone. Moving the Upper Bone will bend the arm but have no effect on the End Bone. Rotating either the IK Chain or Upper Bone will have no effect. Create a new Point and Quick Align (position only) to the Lower Bone, then move it out a little. Select the IK Chain and in the Motion Panel>IK Solver Properties select 'Pick Target' then the new point created. Now as we move that point, the 'elbow' between the Upper and Lower Bones will point towards it.

We cannot link this point directly to the IK Chain, because the chain is dependent on the point, and doing this would mess stuff up royally. Instead, if we create another point and Quick Align it to the End Bone, then link both the IK Chain and the Swivel Target Point to the new point, we can use the new point to control both the position and swivel of the arm.

IK/FK Arm Rig Example

1. Create and Rename Bones
    a. Create a two bone chain and rename them as ‘bn_Upper’, ‘bn_Lower’ and ‘be_End’. Select all three objects and shift drag them to duplicate them twice, one to the left, the other to the right. Double click the Upper of the middle chain to select all of its children and go to Tool>Rename Objects. Untick ‘Base Name’ and tick ‘Suffix’ in the box next to Suffix put ‘_Real’ and click rename.
    b. The duplicated bones already have a three-digit suffix (001, 002, etc.), so when renaming these, I would need to tick ‘Remove Last’ and put the number of digits to three. Give the chain on the left the suffix ‘_IK’ and to the right ‘_FK’.
    c. We will be using the _IK and _FK arms to drive the _Real arm.
2. Differentiate the bones.
    a. Select the _Real bones and in the Animation>BoneTools, double the width and height to make theses bones ticker than the others.
    b. Select the _FK and in the Animation>Bone Tools tick ‘Front Fin’. Select the _IK and tick ‘Back Fin’.
    c. Select all Upper bones and quick align to the middle one. Now all bones are on top of each other, but we should have a good understanding of which bone is which.
    d. Create a point (box), rename it ‘pnt_Shoulder’ and quick align the position to the top of the Upper bones. Select all Upper bones and Position Constraint them to pnt_Shoulder.
3. Open the Schematic Veiw and select the _IK bones, then Alt+Q to isolate this selection.
    a. Selecting bn_Upper_IK, go to Animate>IK Solver>HI Solver and select be_End_IK. Rename IK Chain to ‘ik_armIK’.
    b. Create a circle called ‘ctrl_wristIK’ and Quick Align (position and rotation) to be_End_IK. In the Hierarchy panel, select ‘Affect Object Only’ and rotate the circle locally 90 degrees on the Y axis.
    c. Link the ‘ik_armIK’ to ‘ctrl_wristIK’, select the control and Alt+RMB>Transform>Freeze Transform.
    d. Lastly, Orient Constraint ‘be_End_IK’ to ‘ctrl_wristIK’. Now as we move the control, the arm moves, and as we rotate the control, the be_End_IK rotates with it.
    e. You may add in the Swivel Target as mentioned above.
4. Open the Schematic Veiw and select the _IK bones, then Alt+Q to isolate this selection.
    a. In the schematic view, select the ctrl_wristIK and Ctrl+V to duplicate it (make sure it is a copy, not instance). Call this ‘ctrl_wristFK’ and in the Layer Explorer make it visible by selecting the greyed out bulb.
    b. In the ‘ctrl_wristFK’ Modifier panel, make the Radius smaller than it currently is. Duplicate this twice calling one ‘ctrl_upperFK’ and the other ‘ctrl_lowerFK’, then Quick Align (position and rotation) to their relevant bones.
    c. In the Schematic Veiw, link these together in the Upper>Lower>Wrist order. Then link this chain to the ‘pnt_Shoulder’ point.
    d. Now Orientation Constrain the bones to their controls, then Freeze Transform the controls. Now the controls can control the bones.
5. Un-isolate all, then selecting each of the _Real bones, Orientation Constraint them to the others, Upper to Upper, Lower to Lower, End to End. Do all IK first, then FK second.
    a. Now move/rotate the separate controls to make a clear gap between the IK and FK bones. We will see that the _Real bones stay perfectly half way between the two sets.

b. As we select any _Real bone and to go the Motion panel, we will see the constraints in the Rotation Parameters. We would notice that there are two targets, the _IK and _FK, both are weighted to 50 (%). Select one of these and change the Weight using the roller below. As the weight increases, the _Real bone will fit more to that position or orientation, as it decreases, the bone will back away. We need to wire these rollers to a custom attribute on the controls.

6. Select all controls and add an Attribute Holder to the modifier stack. This modifier is in italics to show it is instanced across all selected controlers.
    a. Go to Animation>Parameter Editor and create a new parameter as described below.
    b. Type:Float, Name:IK/FK, Range:00 to 10
    c. You will have to select one single controller to click 'Add', then that new parameter will be available across all controls.
    d. Selecting one control, RMB>Wire Parameters. A popup will appear, go Modified Object>Attribute Holder>Custom Attributes>IK/FK, then select bn_Upper_Real and another popup will appear. Go Transform>Rotation>Orientation Constraint>Orientation Weight 0.
    e. Now the Parameter Wiring dialogue box has opened. In the left we see the attribute selected first (IK/FK) and in the right we see the attribute selected second (Orientation Weight 0). We need the IK/FK to drive the Orientation.
    f. Select the arrow from left to right and select Connect. Playing with the IK/FK Control, we can see that it only does part of the job. The control goes from 0 to 10, so we need to increase the power of the wiring by multiplying it by 10. In the Parameter Wiring dialogue box, we can see that the two attributes are connected so that the Weight is equal to IK/FK. If we type 'IK_FK*10' under the right column, and 'Update' it gets closer, but isn't perfect. This is because we still have 50% influence from the other bone.
    g. In the Parameter Wiring dialogue box select 'Orientation Weight 1' from the right column and connect. If we type 'IK_FK*10' under the right column again, we will see no change in the _Real bone's rotation as we use the IK/FK Control. This is because both weights are going from 0 to 100 at the same time. We need to reverse one of these.
    h. If we want IK to be at 0 on our controller and FK to be at 10, we would need to reverse the effect of our IK wiring.
    i. Again in the Parameter Wiring dialogue box, select FK/IK from the left and Orientation Weight 0 from the right and type '100-(IK_FK*10)' under the right column. Test it out, and the bn_Upper_Real should swing fully between the bn_Upper_IK (0) to bn_Upper_FK (10).
7. To do the same to the Lower and End bones, select them in the scene, then hit the refresh box over the right column in the Parameter Wiring dialogue box.
    a. Find the Transform>Rotation>Orientation Constraint attributes and wire them to the IK/FK attribute to the left, keeping in mind the expressions we used to amend the values.
8. Play around with the controls, everything should work fine, then select the controls and Alt+RMB>Transform to Zero to bring everything back to their frozen positions.

Rope Rig Example

1. Create a spline and a few controllers.

    a. Create a spline in a slight ‘S’ shape with only three vertices. Shift Drag it to the left to create a duplicate.
    b. Create three large points (box) and snap them to the three vertices of the original spline, rotating them to fit the contour of your spline. Select both splines and add an ‘Edit Spline’ attribute to select both of the bottom verts, then use a Linked Xform to connect them to the bottom point, do the same with the middle and top vertices, Linked Xforming them to the middle and top points.
2. Create Path points
    a. Figure out how many bones you would like in your spline; I have chosen five. Creating six smaller points (box), Path Constrain them to the first spline at 20% intervals from 0% to 100%.
    b. Create five more points and Path Constrain them to the second spline at 20% intervals from 10% to 90%. These will be the bones up-nodes and will help us rotate the spine.
3. Create the bones.
    a. Create a small bone and duplicate it, and its end bone, five times.
    b. Each bone base needs to be Position Constraint to one of the smaller points with its end bone constrained to the next one. Then each bone needs to have a LookAt Constraint to the point containing its end bone.
    c. This LookAt Constraint needs to be given an up-node which are constrained to the second path.
4. Now using the three larger points as controls, we can manipulate the spline and the bones will follow uniformly.