

A WIRE PARAMETER AND REACTION MANAGER BASED BIPED CHARACTER SETUP AND RIGGING TECHNIQUE IN 3DS MAX FOR ANIMATION

Zeeshan Bhati, Ahmad Waqas, Mostafa Karbasi, Abdul Waheed Mahesar

Khulliyah of Information and Communication Technology
International Islamic University Malaysia

ABSTRACT:

In this paper a unique way of rigging a bipedal character is discussed inside 3Ds Max, using constraints, wire parameter setup and linking the controls. 3Ds Max is more commonly used for game character modeling and rigging. For this purpose a built-in Biped setup is present that is a pre-rigged structure, which can be easily applied on any bipedal character type easily and efficiently. However, the built-in Biped system lacks the functionality of manual control and motion retargeted manipulation, mostly found in custom rigs. This proposed system builds a custom rig using a unique approach of combining multiple tools and expressions.

KEYWORDS:

Biped Rigging, Rigging, 3Ds Max, Biped, Character Setup, Animation, Biped Animation

1. INTRODUCTION

Rigging is the most essential part of any animation process. In order to bring any object to life, an animator needs a set of controls and manipulators, which would allow him to select the each part and animate. Character rigging is essential and imperatively long process, involving several long tedious procedures. These controllers and manipulators usually consist of simple joints, locators, selection handles, splines curves, or even an independent graphical user interface (GUI) for control selection[1]. By connecting a rig to a model in a process called binding, the model mimics the motions of the rig like a puppet. The boredom of manually doing this process for each character and object in a project makes the pipeline of character animation more time-consuming, difficult and problematic [2][3].

Generally, a large portion of animators use a pre-generated rigs to animate their characters, such as the Biped tool in 3DS Max, Andrew Silke's Generi Rig for Maya, or Cryptic Studio's CrypticAR [4]. Similarly, the more advance auto-rigs used in the industry are The Setup Machine, Face Robot and MetaNode Rigging [5]. In this tutorial a unique approach is presented to design and develop a biped rig in 3D studio Max, using Bones, wire parameters, constraints, and linking the controls. The basic principles and requirements of creating a biped rig were used from the paper of Bhatti et al.[6]. The rigging process involves the knowledge of the various tools and modules inside 3Ds Max which can be learnt from the book [7].

2. BONE CREATION

The rigging process is initiated by first creating a hierarchical bone structure. First, create the character bone hierarchy according to the diagram, shown in Figure 1. The background information and understanding of 3Ds Max bones can be studied from [8]. Once the bone structure is created, then each bone of individual body part must be named for formal understanding and recognition in the subsequent steps. The naming convention followed for each bone is given in the figures below, for example, the individual bones of left leg for the biped character are named as Left_Leg, Left_Knee, Left_Ankel, Left_Ball and Left_Toe, as illustrated in Figure 2. The same names are used for right leg with word right preset.

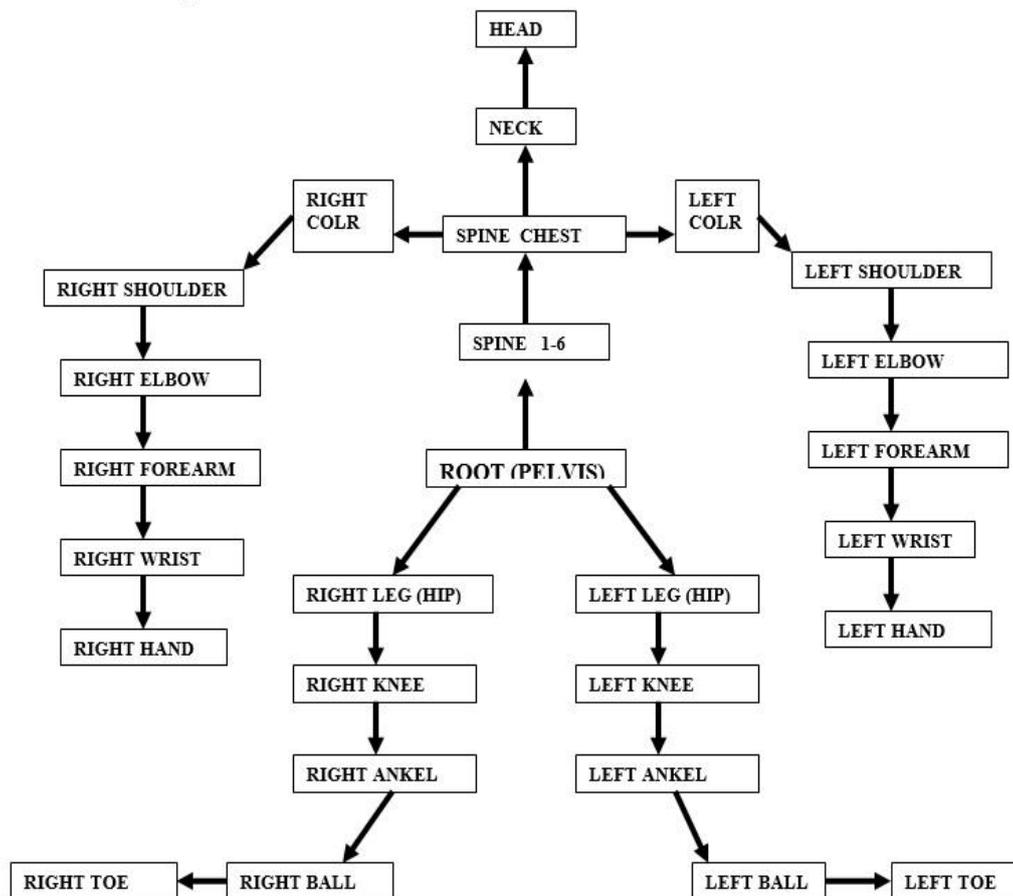


Figure 1: Hierarchical Bone Structure of Biped Character

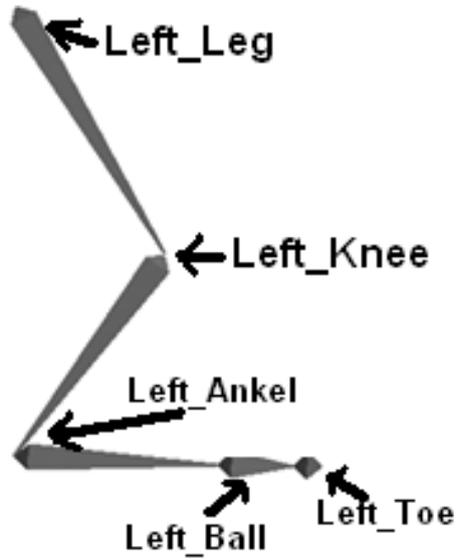


Figure 2: Bone structure of LeftLeg

Similarly, all the bones of the body are named and labeled as shown in Figure 3. Similar to legs, each arm is also labeled with left and right preset, to identify each body section.

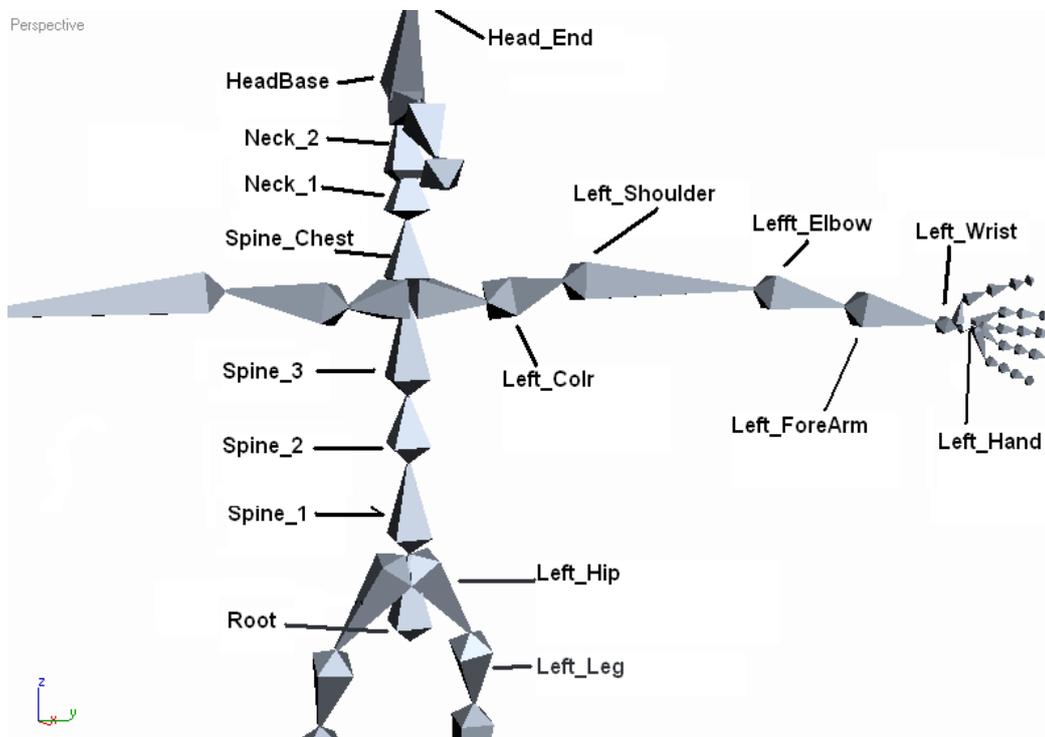


Figure 3: Bone Structure for the Spine and Left Arm

Each bone of hand and fingers are also named and label as shown in Figure 4.

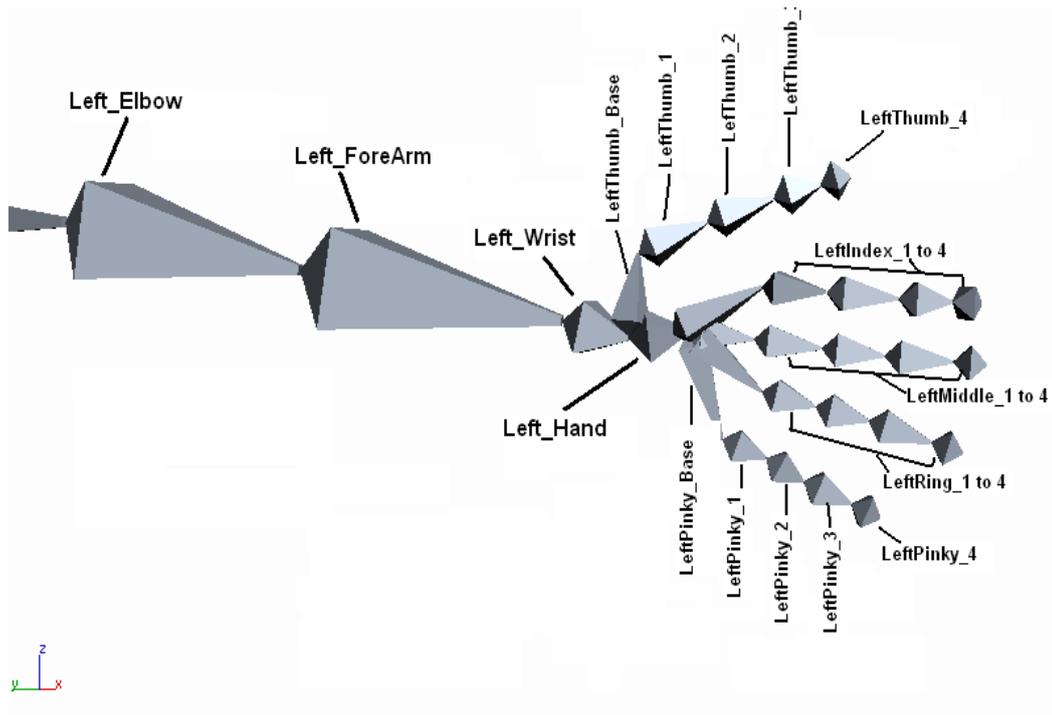


Figure 4: Naming of Bone Structure for the Hand

3. RIGGING THE LEGS AND FEET (CREATING CUSTOM CONTROLS):

Now we begin the process of rigging the biped character, the process initiates from the leg. The rigging of leg begins by creating the Inverse Kinematics (IK) setup between joints of the leg.

3.1. Create IK Solver Chains:

Here are the step by step process of creating the IK joint chain for legs.

- i. In Perspective view port, select the Left_Leg bone and go to main menu Animation → IK Solvers → HI Solver.
- ii. This allows you to create a basic IK chain. A rubber-band line appears between the bone and the mouse cursor.
- iii. Now click on the Left_Ankel bone. This will create a IK Chain from the Left_Leg bone to Left_Ankel bone.
- iv. Create another IK chain (HI Solver) from Left_Ankel bone to Left_Toe bone.

This extra IK chains let you keep the feet planted solidly, but also let you produce other types of motion, such as heel lift and toe rotation.

- i. Name the first IK solver as LeftAnkel_IKGOAL. Name the second IK as LeftFoot_IKGOAL.
- ii. With LeftFoot_IKGOAL still selected, go to the Motion panel > IK Solver Properties rollout > IK Solver Plane group, and set Parent Space to IK Goal.

This helps to resolve swivel (Rotation) issues that could occur when you add the knee rotation/swivel. Using the IK Goal setting, the swivel target's calculations for twist will be based on the end-effector rotation, which means the feet will stay flat unless you rotate them. Using the default Start Joint setting, on the other hand, the twist would be based on the parent joint's twist rotation. Thus, if the knee were to rotate, the foot would also start to twist or rotate, which you'll want to avoid later on.

3.2. Create the Foot Control Shapes

The next step involves, creating the control shapes for the foot, which allows the animator to easily select the foot and animate it.

- i. Goto Create Panel > Helpers and create a Point Helper (Box). Increase the size of box and make it the Size of the Foot in rectangle shape. Name the helper as Left_FootCTRL.
- ii. We will use this Left_FootCTRL helper to control the motion of the entire leg.

Warning: Make sure you don't do a Non-uniform scale (Scale an object in only one axis i.e. x-axis only) on the object itself, as that will cause Skewing/Rotation problems to objects that are parented to it.

- iii. Align this helper with the LeftBall bone and this position it accurately around the foot. Then go to Hierarchy Panel and choose Pivot Button, then click Affect only Pivot.
- iv. Turn on Pivot Snapping and Move the Pivot point of the LeftFoot_CTRL to the LeftAnkel_IKGOAL.
- v. This will allow us to move the foot in much natural way.
- vi. Create another point helper (box), and align it to the Ankel_IKGOAL. Name this helper FootRotateHELP.

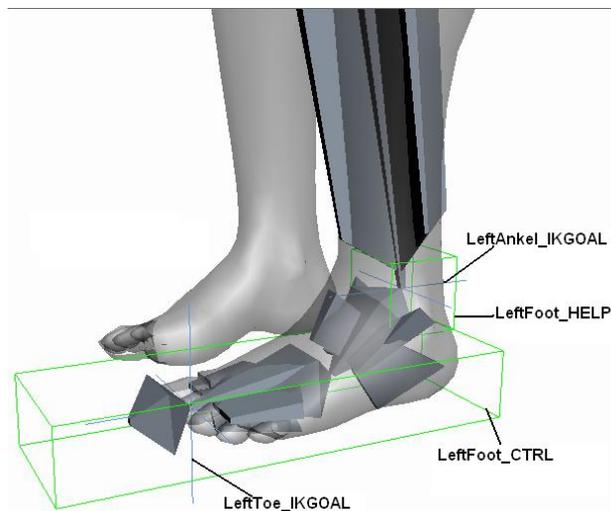


Figure 5: Left Foot Control Shapes and IK Solvers

3.3 Link the Helper Controls with IK GOAL:

Now, each of the helper created needs to be linked to the IK goal attribute of each IK controller.

- i. Select the LeftAnkel_IKGOAL, click the Link button on the toolbar and link this IK with the LeftFoot_CTRL.
- ii. Select the LeftToe_IKGOAL and link it with FootRotateHELP.

- iii. Select the FootRotateHELP and Link it with LeftFoot_CTRL.

Before proceeding further, test the helper Foot Control movements . When you Move the LeftFoot_CTRL the entire leg moves naturally and when you rotate the LeftFoot_CTRL the Foot rotates.

3.3. Add Custom Attributes for Knee Control:

The knee of the biped character is specifically controlled using an additional attribute for higher level manipulation. This is achieved by adding a custom attribute in the foot control helper and connecting it with IK swivel angle. First, create a custom attribute, here are the steps.

- i. Select the LeftFoot_CTRL, goto modify panel and add a Modifier “ Attribute Holder”.
- ii. With LeftFoot_CTRL selected, Goto main menu Animation > Parameter Editor

Apply following settings,
Add to Type = Selected Object’s Current Modifier
Parameter Type = float
UI Type = Spinner
Name = Knee_Twist
Range From = -120 to + 120
Default = 0
Click Add Button

3.4 Wiring Knee_Twist with IK Swivel Angel

The inbuilt wire parameter module of 3Ds Max, is a very powerful tool that allows any attribute of one object to be directly manipulated and controlled by another parameter belonging to an independent object. Here, this wire parameter tool is used to connect and drive the custom knew_twist attribute to the IK swivel angle.

- i. Select the LeftFoot_CTRL and Right Mouse Click (RMC) and select Wire Parameters from the Quad menu (At Bottom).
- ii.

A popup menu will appear and choose the Knee_Twist parameter as illustrated in Figure 6:



Figure 6: Wire parameter setting for knee twist

- iii. A rubber band will be attached to the point, Now select the LeftAnkel_IKGOAL. Another similar popup will appear from which choose Swivel Angel.



A Wire Parameter Dialog Box Will Appear.

- iv. Select one way connection: Left to Right, From the Three buttons in the middle, and Hit Connect as shown in
- v. Figure 7.

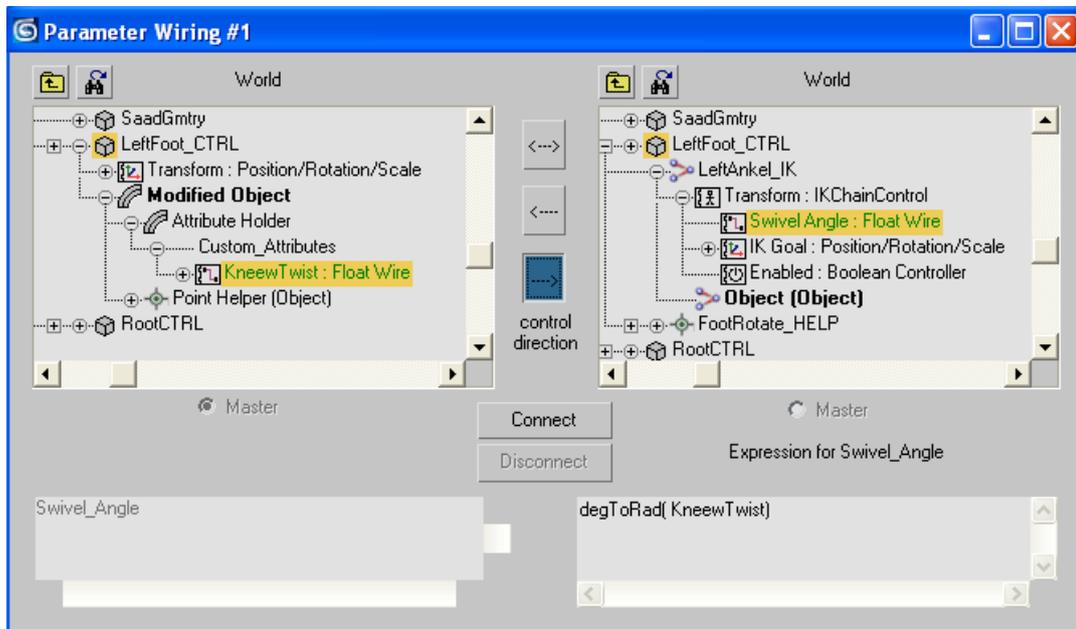


Figure 7: Wire Parameter dialog for connecting knee twist with IK swivel angle

- vi. Then check the Connection by Manipulating the Knee_Twist Value.

See the Over Shooting Rotation of the Knee. What is actually happening is not an Error, but rather an issue with how **3Ds Max** internally handles rotations. Currently, **3Ds Max** wants radians, that is, 6.28 Radians equals 360 degrees which means that if you change Knee_Twist from 0 to 3.14, you'll get a 180 or half spin. It's easier if you work in degrees. Next, you'll use a function called degToRad to convert Knee_Twist into radians.

- vii. In the Parameter Wiring dialog, change the wiring expression to: **degToRad(Knee_Twist)** as shown in the above Figure. Click Update.

Now Check the Rotation, it will seem Right now. Leg Rig is Complete. Check the Motion of the Leg. The same procedure is then done for the right leg as well.

4. RIGGING THE ARM AND HAND

The arm is the most visible and expressive part of the body. Therefore, the arm rig need to be able to incorporate such wide variety of motion gaits. The process of arm rigging begins with creating and linking the IK chain to the arm bones. Then, that IK chain is again constraint to the helper (similar to the Leg Rig) for easier manipulation and control of the arm.

4.1. Creating and Linking the IK Chain for Arm:

- i. Select the Left_Shoulder Bone and from main menu select Animation > IK Solvers > HI Solver and again click on the wrist bone to create an IK Chain from the Shoulder to the Wrist Bone.

- ii. Name the IK as LeftArm_IKGOAL.
- iii. From Create Panel , create a Point Helper (Box) and align it with the LeftArm_IKGOAL. Increase the helper size if necessary. Name the Helper as LeftArm_CTRL
- iv. Link the LeftArm_IKGOAL with the LeftArm_CTRL.

Check the Control by moving the helper. Make sure you press undo when you are done.

4.2 Creating Controls for the Hand and Fingers

Fingers, the most crucial and key components in any animation project, hold the maximum motion gestures and thus require extremely deep control over their motion parameters. For finger motion types, we used 6 types of general customized motion attributes. These attributes are Index finger roll, Middle finger roll, Ring finger roll, Pinky finger roll, Thumb roll and finally finger Spread parameters. For this purpose, 6 custom attributes are created using the same process as discussed earlier for the knee twist, as shown in Figure .

- i. Select the LeftArm_CTRL and add a modifier “Attribute Holder” to this Helper.
- ii. From Main Menu select Animation > Parameter Editor
- iii. Add new parameters for the each finger of the hand with following properties

Add to Type = Selected Object’s Current Modifier

Parameter Type = float

UI Type = Spinner

Range From = -5To 10

Default = 0

Name =

- a. Index_Roll
- b. Middle_Roll
- c. Ring_Roll
- d. Pinky_Roll
- e. Thumb_Roll
- f. Fingers_Spread (Range From = -10 To = 10)

Similarly add few more Parameters for the Wrist Movement and Elbow control.

Add to Type = Selected Object’s Current Modifier

Parameter Type = float

UI Type = Spinner

Range From = -120 to +120

Default = 0

Name =

- a. Wrist_Rotate
- b. Wrist_Spin
- c. Wrist_Bend
- d. Elbow_Twist

Add all of the above parameters in the LeftArm_CTRL.

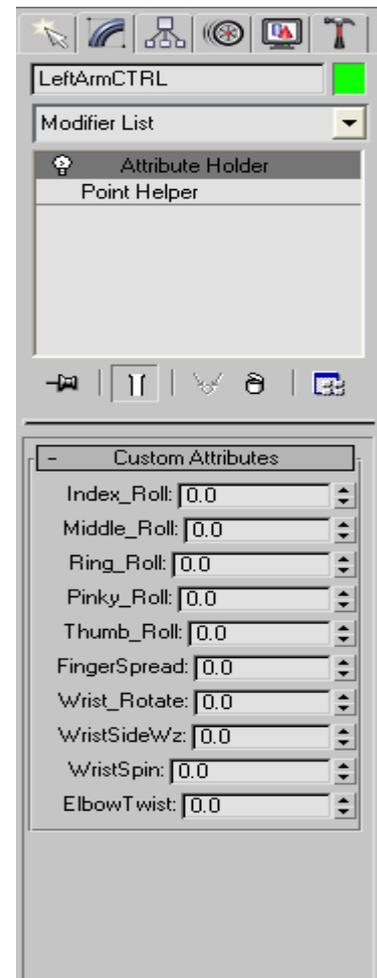


Figure 8: Custom Parameters for carious finger motions

4.3 Using Reaction Manager to connect the Hand Control Parameters with bones.

Now we are going to control the motion of the Fingers with our custom created parameters by using Reaction Manager.

Reaction manager works on the principle of Master and Slave. Here a Master Parameter controls one or more Slave Parameters. We first specify a Master Parameter and then we specify a slave parameter, which becomes the InitialState of the Master/Slave. Then we have to specify the Final State of the both, after which the controls are established.

- i. From main menu select Animation > Reaction Manager. A Window opens.
- ii. At the Top of the Window click the Add Master Button – shown in Figure 8, and in the view port select the LeftArm_CTRL.



Figure 8: Add Master option in the reaction manager

- iii. A popup will appear from which to select the parameter of the control as shown in the Figure 10:

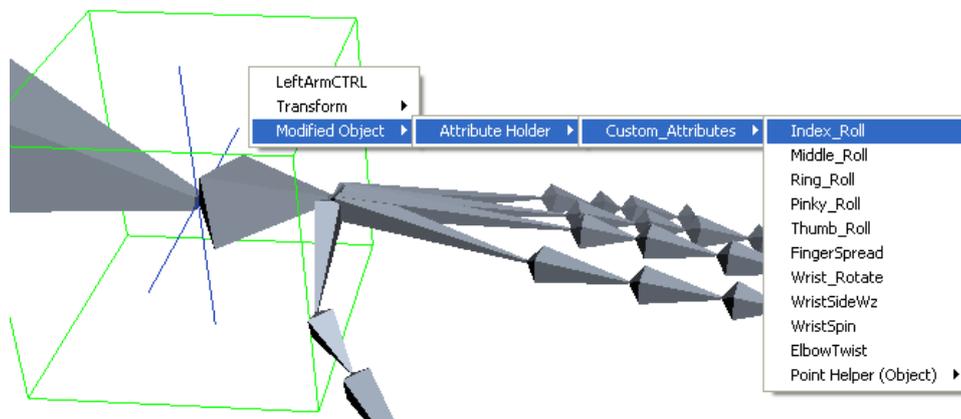


Figure 10: Selection of Master Parameter for the Index_Roll parameter

This will create the master parameter.

- iv. Now first check the Local Rotation Axis of the Bone Fingers, by first selecting the rotate tool and then select the Local axis from the drop down list on Main Toolbar highlighted in Figure 11.

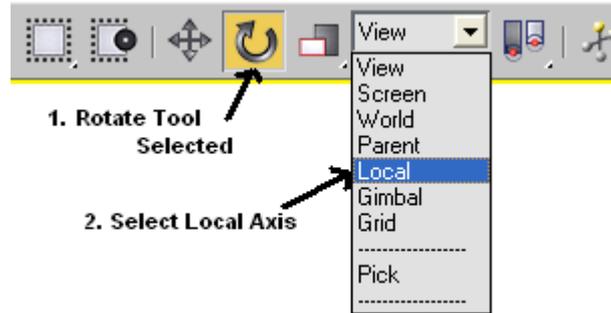


Figure11: Local Axis selection

- v. After checking that the bone finger will have to Rotate on Y-axis for it to close the finger, we will select all the 3 main bones of the Index Finger.
- vi. Then open the Reaction Manager again and Click on the Add Selected Button, as illustrated in Figure 12:



Figure 12: Add selected option in Reaction Manager

This will open another popup window, from which you have to select the slave parameter of the finger bones (Rotate Y) as in Figure:



Figure13: Popup to select the parameters of Selected Index Finger Bones

- vii. This step will create the Slave Parameter of the Selected Bones and It will also record the Initial State of the Bones.

Now we have to create the FinalState of the Bones.

- viii. Select the LeftArm_CTRL and in Modify Panel Move it Index_Roll Parameter to 10.
- ix. In Reaction Manager Turn On **Create Mode**, as illustrated in
- x. Figure 9.



Figure 9: Create Mode Selected in Reaction Manager

Create Mode Will allow us to Rotate the bones in Y-axis.

Note: If create mode is not On, then you will not be able to rotate the bones in Y direction.

- xi. Now In view port, select the Index_01 bone and Rotate it in Y – axis at approximately 70 degrees.
 Select Index_02 and Rotate it also in Y-axis at 70 degrees.
 Select Index_03 and Rotate it also in Y-axis at 70 degrees.

- xii. Back in Reaction Manager, Click on Create State Button, shown in Figure
- xiii.

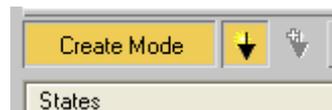


Figure 15: Create Mode Button Selected and Create State Button Select

This will Record the Final State of the Bones According to the Master’s Position. An important thing to note here is, to make sure to uncheck the create mode button in the end.

- xiv. Now go and check the Reaction between the LeftArm_CTRL / Index_Roll and Index Finger Bone. The reaction manager window should look similar to the one given in
- xv. Figure.
- xvi.

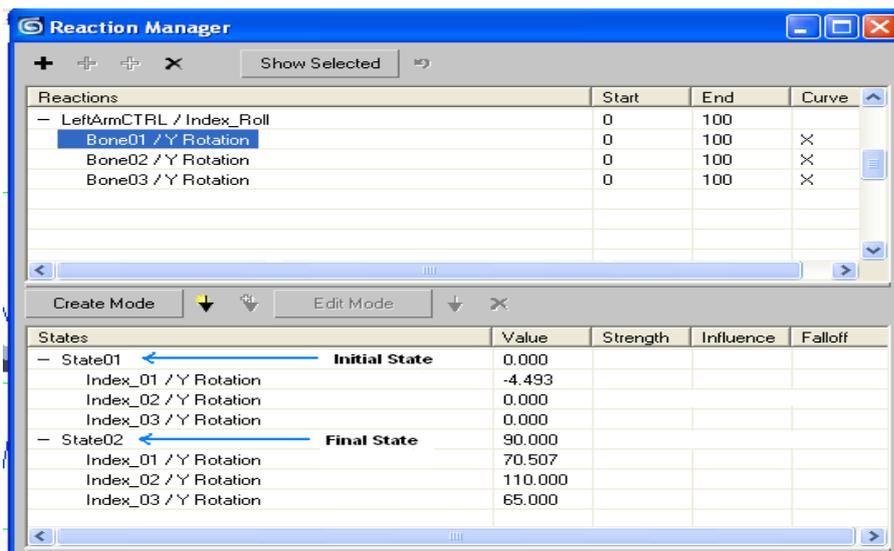


Figure 16: Showing initial and Final State of the Index Roll Parameter

xvii. Similarly create the third state which will be the finger bending upwards.

- a) Select the LeftArm_CTRL, Change Index_Roll = -5
- b) In Reaction Manager Click **Create Mode** Button.
- c) In View port Rotate in Y – axis the Index_01 bone to –25.
- d) Rotate in Y – axis the Index_02 bone to –25 and finally
- e) Rotate in Y – axis the Index_03 bone to –25 as well.
- f) Again in Reaction Manager, Click **Create State** Button.
- g) Click Create mode button again to uncheck it.

A very important this to keep in view is that, for every parameter of the LeftArm_CTRL, we will have to create a new Master, i.e.

Master for LeftArm_CTRL / Index_Roll
Master for LeftArm_CTRL / Middle_Roll
Master for LeftArm_CTRL / Ring_Roll

and so on. Then each finger bones will become slave under their respective Master.

xviii. All the Finger Bones will be controlled in the same manner.

1.1 Elbow and Wrist Controls:

Once the finger control process is complete, now the elbow and wrist parts of each hand are rigged. For elbow and wrist, Wire Parameter tool will be used. Here, the Reaction Manager can also be used, it depends on us. We will Wire the LeftArm_CTRL / Elbow_Twist Parameter with the LeftArm_IKGOAL / Swivel Angel, in the same way as we had wired the Knee_Twist of the Left Leg.

- i. Select the LeftArm_CTRL, RMC and choose Wire parameter.
- ii. From popup select Modified Objects > Attribute Holder > Custom Attributes > Elbow Twist.
- iii. With Rubber Band Active select the LeftArm_IKGOAL . another popup appears from which Select Transform > Swivel Angel.
- iv. In Wire Parameters Dialog Box choose One Way connection – Left to Right and click **Connect Button**.
- v. Then Write the Expression degToRad (Elbow_Twist) in the bottom right window. Click **Update**.

In the Same Way Wire or create Reaction between the **Wrist_Rotate**, **Wrist_Spin** and **Wrist_Bend** Parameters with the LeftWrist bone's **Rotate X, Y** and **Z** Attributes.

2. CREATING ROOT CONTROLS.

After the legs and hand rigging, now we move to other parts of the body. We begin with the root or pelvic controls. To create a root controller setup, follow the following steps.

- i. Create a point helper(Box). Align it with root Bone. Move it forward a distance.
- ii. Name this helper as Root_CTRL.
- iii. Link Root Bone with this Root_CTRL Helper.

2.1 Creating Spine Controls:

For the spine, again we will use some custom attributes to control the rotation, bending and twisting effect of the spine bones.

- i. Select the Root_CTRL, Add a Modifier “Attribute Holder”.
- ii. Add 3 Attributes to the Root_CTRL. From Parameter Editor.
 - Add to Type = Selected Object’s Current Modifier
 - Parameter Type = float
 - UI Type = Spinner
 - Range From = -120 to + 120
 - Default = 0
 - Name =
 - a. Spine_Rotate
 - b. Spine_Bend
 - c. Spine_Twist

Add all of the above parameters in the Root_CTRL.

- i. Select the Spine_01 bone and in Local Axis check the rotation axis of the spine bones that on which axis it will rotate and on which axis it will bend. Note: Make sure you are in Local Axis mode.
- ii. Use Reaction Manager to control the Spine control parameters of Root_CTRL.
Bones are : Spine_01 , Spine_02, Spine_03 and Spine_04
 - a. Spine_Rotate = Rotate Z
 - b. Spine_Bend = Rotate X
 - c. Spine_Twist = Rotate Y

3. Creating Head Control:

- i. Create another point helper, align it with neck bone and the move it forward so it lies in front of the head. Name it Head_CTRL.
- ii. Select the **Neck_01** Bone and go to main menu, Animation > Constraints > Orientation Constraint.
- iii. With rubber band active, select the Head_CTRL.
- iv. The Neck Bone Flips or twists itself. That is because now the rotation of the Neck_01 bone is being controlled by the Head_CTRL.

To Fix this, select the Neck_01 bone, go to Command Panel > Motion. Scroll down and in Orientation Constraint folder Check on Keep Initial Offset, as shown in

Figure 10. This Will Move the Neck_01 Bone to its original place and the constraint will also work.

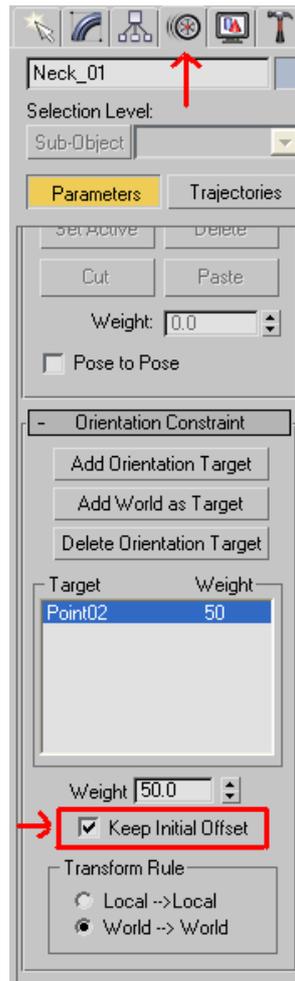


Figure 10: Check On Keep Initial Offset option.

4. DUPLICATING LEFT SIDE CONTROLS ONTO RIGHT SIDE

Once, the left leg and left arm have been rigged, we need to apply the entire process again on the right side. This can be little cumbersome and time consuming. A simple copy paste option will not work here also, due to complex wire parameter connections, and reaction manager relations with object and custom parameters. To avoid this, a unique technique of creating a duplicate of the rig and creating a clone replica of left side is done. First, create Right Leg and Left Arm Control Setup by Cloning the Left Leg and Left Arm Rig, as described in the steps below.

- i. Save The previous file as another name i.e, temp.max
- ii. Open this temp.max file.
- iii. unhide all objects in the scene, and delete and Geometry or model that u have.
- iv. Than delete the Following in same order:
 - a. Head Bones
 - b. Neck Bones
 - c. Head_CTRL
 - d. SpineChest Bone
 - e. Spine Bones All
 - f. Root Bone
 - g. Root_CTRL

- v. Now select every thin and group it.
- vi. Then mirror it in opposite direction and move it so that it will be in its proper place (Right Side).
- vii. Rename all the joints and control from Left to Right.
Tip: Use **Rename Objects Tool**, found in Main menu Tools > Rename Objects.
- viii. Save the file.
- ix. Open the Previous File and Merge this file (Temp.max) with it from File > Merge. Select all objects and merge.
- x. In this original file select the group which we have just merged, and position it accurately with the original body bones and then ungroup it.
- xi. Select the Right_color bone and link it with Spine_Chest Bone. Similarly Select the RightLeg bone and Link it with Root Bone.

This simple cloning process reduces the time of creating the right side rig again, and solves the problem of duplication through copy paste also.

5. CREATING CHARACTER

Finally, a main Character controller is created which is used to control the entire character rig. Through this main character controller, the animator can easily select and move the entire character anywhere in the scene and position it according to the scene requirements.

- i. Select the LeftArm_CTRL, RightArm_CTRL and Head_CTRL, and link them all with Root_CTRL.
- ii. Then Select All the 6 Controls again,
 - a. LeftArm_CTRL.
 - b. RightArm_CTRL
 - c. Head_CTRL
 - d. Root_CTRL
 - e. LeftLeg_CTRL
 - f. RightLeg_CTRL

From Main menu select Character > Create Character. This will make 3Ds Max recognize my character as a teal character.

- iii. Select the Human Doll symbol that has appeared at the grid, and in modify panel, click on **Set as Skin Pose** button. Click yes, on the conformation dialog box.

This will save your default starting pose, so whenever you move your controls, and change their position, all you have to do is select the Character Doll Symbol and in modify panel, Click on the **Assume Skin Pose** button and the character will go back to its default pose. The final fully rigged biped character is shown in

Figure.

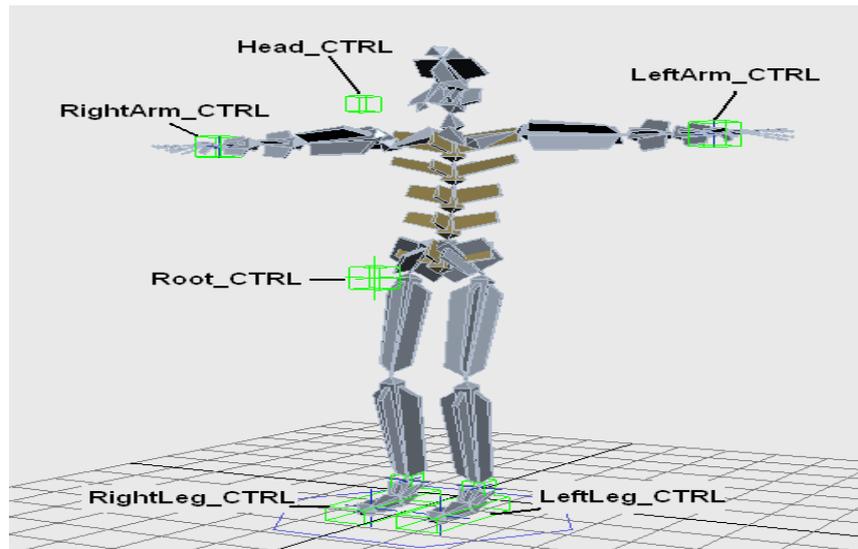


Figure18: Final complete Character Rig with Fins on each Bones

6. CONCLUSION

In this tutorial based paper, a complex technique of manually rigging a biped character inside 3Ds Max was discussed. The process involved using various custom control attributes, which were manipulated using wire parameter connection tool, with fingers driven through a higher level of reaction manager based relation between the finger bones and custom finger manipulation attributes. Similarly, the legs and arms motion is controlled through an Inverse Kinematics based setup with elbow, wrist and knee motion parameters being driven through custom set of parameters linked through wire parameters. The left side rig –containing left leg and left arm, were then cloned to right side, using a unique technique, saving time and effort. The final rig is an advance custom rig, having the capabilities of being able to perform vast animation gestures and poses. In future a graphical user interface is needed to be created. Along with making bones stretch and squash.

REFERENCES

- [1] E. ALLEN and K. MURDOCK, *Body Language: Advance 3D Character Rigging*, First Edit. Cybex. Wiley Publishing, INC, 2008.
- [2] Z. Bhati, A. Shah, A. Waqas, H. Abid, and M. Malik, “Template based Procedural Rigging of Quadrupeds with Custom Manipulators,” in *International Conference on Advanced Computer Science Applications and Technologies*, 2013, pp. 259–264.
- [3] Z. Bhatti and A. Shah, “Widget based automated rigging of bipedal character with custom manipulators,” *Proc. 11th ACM SIGGRAPH Int. Conf. Virtual-Reality Contin. its Appl. Ind. - VRCAI '12*, p. 337, 2012.
- [4] C. Wood, “Creating Custom Rigs In 3DS MAX (Thesis),” Southern Methodist University, 2008.
- [5] A. Turusov, “Modular Procedural Rigging,” Bournemouth University, 2009.
- [6] Z. Bhati, A. Shah, A. Waqas, and N. Mahmood, “Analysis of Design Principles and Requirements for Procedural Rigging of Biped and Quadrupeds Characters with Custom Manipulators for Animation,” *International Journal of Computer Graphics and Animation*, vol. 5, no. 1, pp. 47–67, 2015, arXiv:1502.06419.
- [7] M. Bousquet, *Model, Rig, Animate with 3Ds Max 7*. New Riders Publishing, 2005.
- [8] K. L. Murdock, *3Ds Max 9 Bible*. John Wiley & Sons., 2007.