Chapter 6. Rigging the Low Resolution Mesh

In this chapter, we will cover the following recipes:

- Building the character's Armature from scratch
- Perfecting the Armature to also function as a rig for the Armor
- Building the character's Armature through the Human Meta-Rig
- Building the animation controls and the Inverse Kinematic
- Generating the character's Armature by using the Rigify add-on

Introduction

To be able to animate our character, we have to build the rig, which in Blender is commonly referred to as an **Armature**, and this is the *skeleton* that will deform the **Gidiosaurus** low resolution mesh.

The rigging process in Blender can be accomplished basically in two different ways:

- By building the **Armature** by hands from scratch
- By using the provided **Human Meta-Rig** or the **Rigify** add-on

Building the **Armature** manually by hand can be a lot of work, but in my opinion, is the only way to really learn and understand how a rig works; on the other hand, the **Rigify** add-on gives several tools to speed up and automate the rig creation process, and this in many occasions, can be very handy.

Building the character's Armature from scratch

So, the first recipe of this chapter is about the making of the **Armature** by hands for our **Gidiosaurus**.

Getting ready

In this first recipe, we are going to build by hands the **basic rig**, which is the skeleton made only by the **deforming bones**.

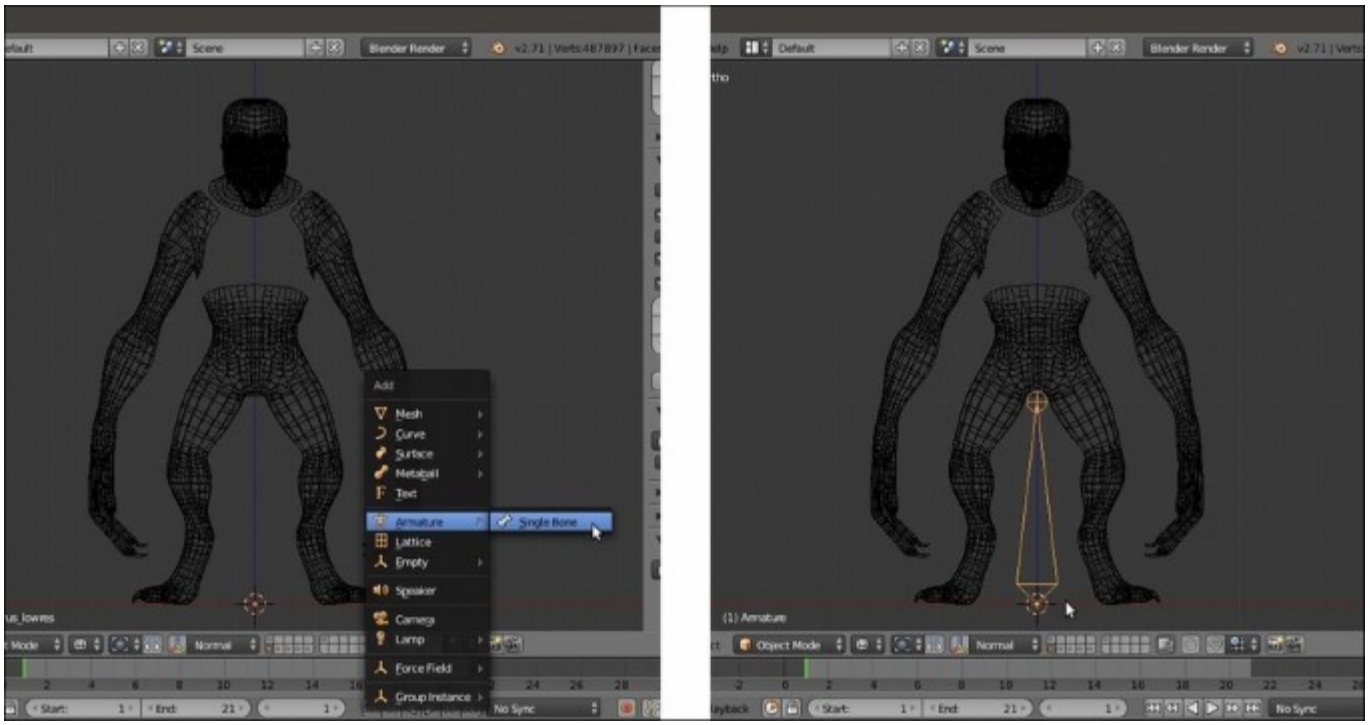
However, first, let's prepare a bit the file to be worked:

1. Start Blender and open the `Gidiosaurus_unwrap_final.blend` file.
2. Disable the **Textured Solid** and **Backface Culling** items in the 3D view **Properties** sidepanel, join the 3D window with the **UV/Image Editor** window, and click on the **11th** scene layer to have only the **Gidiosaurus** mesh visible in the viewport.
3. Go to the **Object** window under the **Display** subpanel and enable the **Wire** item. This will be useful in the process in order to have an idea of the mesh topology when in **Object Mode** and **Solid** viewport shading mode. However, for the moment, press the *Z* key to go in the **Wireframe** viewport shading mode.
4. Press *1* on the numpad to go in the **Front** view, and press *5* on the numpad again to switch to the **Ortho** view.
5. Save the file as `Gidiosaurus_rig_from_scratch_start.blend`.

How to do it…

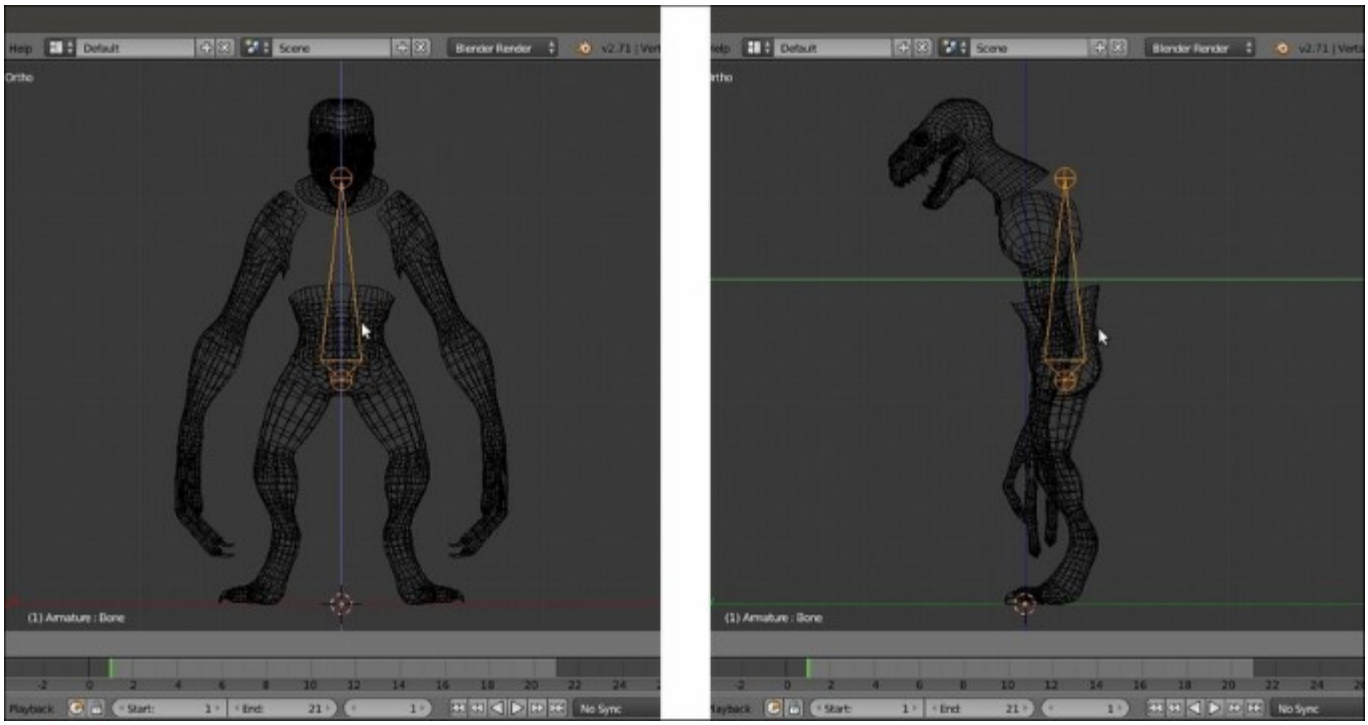
Let's start:

1. Be sure that the **3D Cursor** is at the origin pivot point of the **Gidiosaurus** mesh. Put the mouse pointer in the 3D view, press *Shift* + *A*, and in the **Add** pop-up menu, select **Armature** | **Single Bone**:



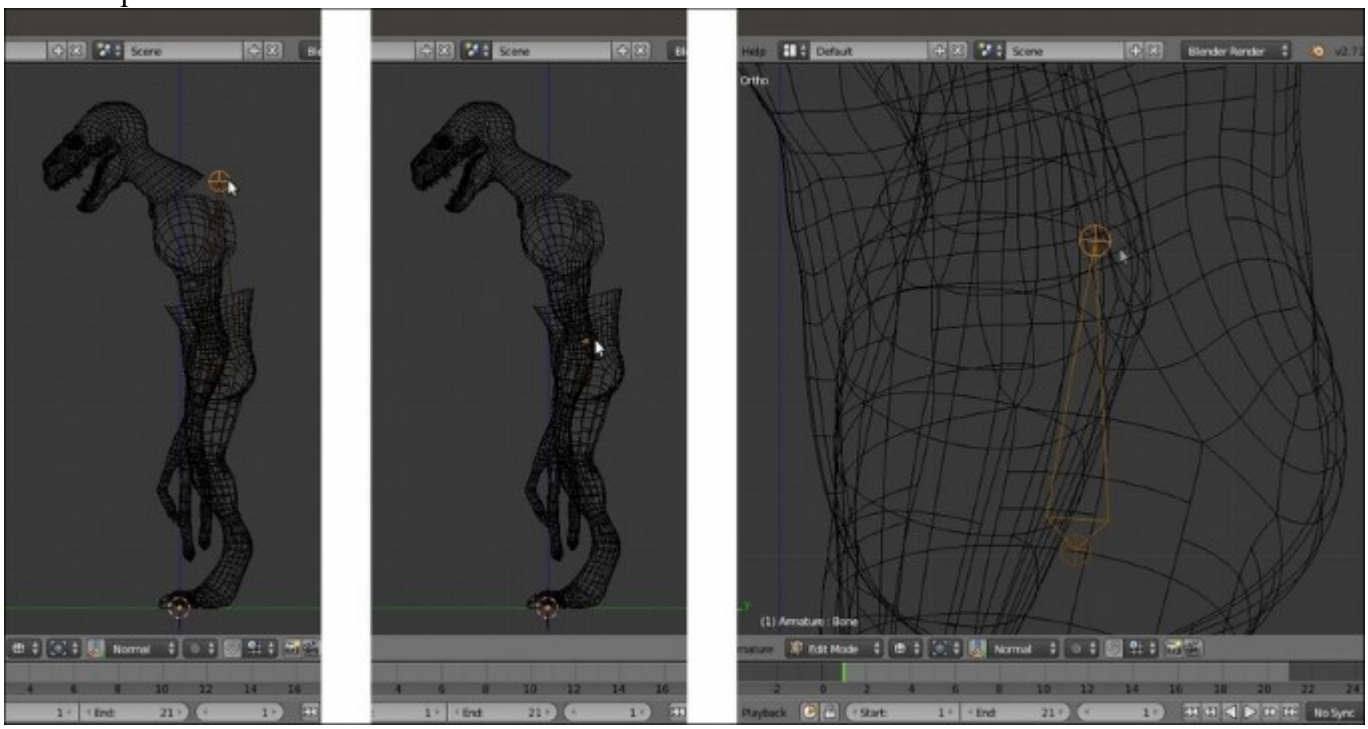
Adding the first Armature's bone

2. Press *Tab* to go into **Edit Mode** and select the whole bone by right-clicking on its **central part**; move the bone upwards to the **Gidiosaurus's** hips area (*G* | *Z* | *Enter* or left-click to confirm), and then go in the **Side** view (*3* key on the numpad) and center its position by moving it on the *y* axis:



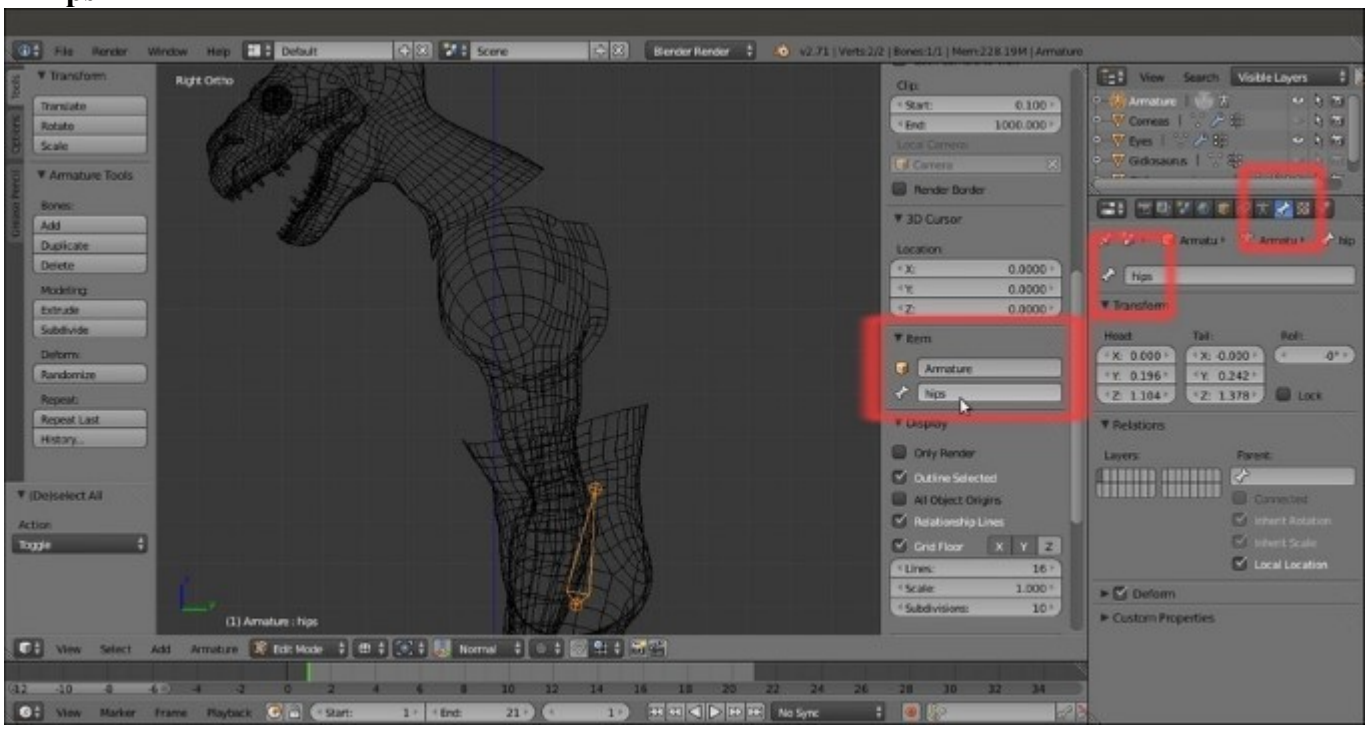
Positioning the bone in Edit Mode

3. Right-click on the **Head** of the bone to select it and by pressing *G* to move it, scale the bone size to fit the pelvis area:



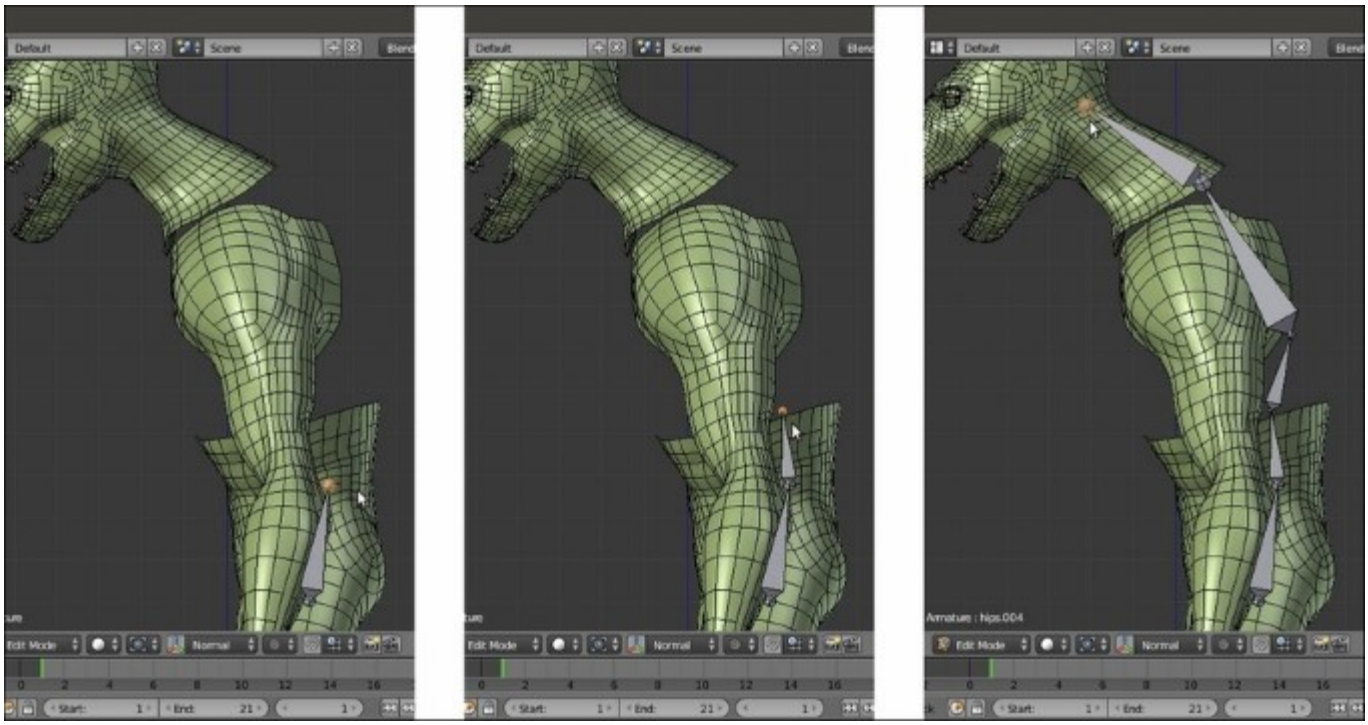
Scaling the bone in Edit Mode

4. Go to the **Item** subpanel under the 3D view **Properties** sidepanel, or in the **Bone** window under the main **Properties** panel to the right-hand side of the screen, and rename **Bone** (default name) as **hips**:



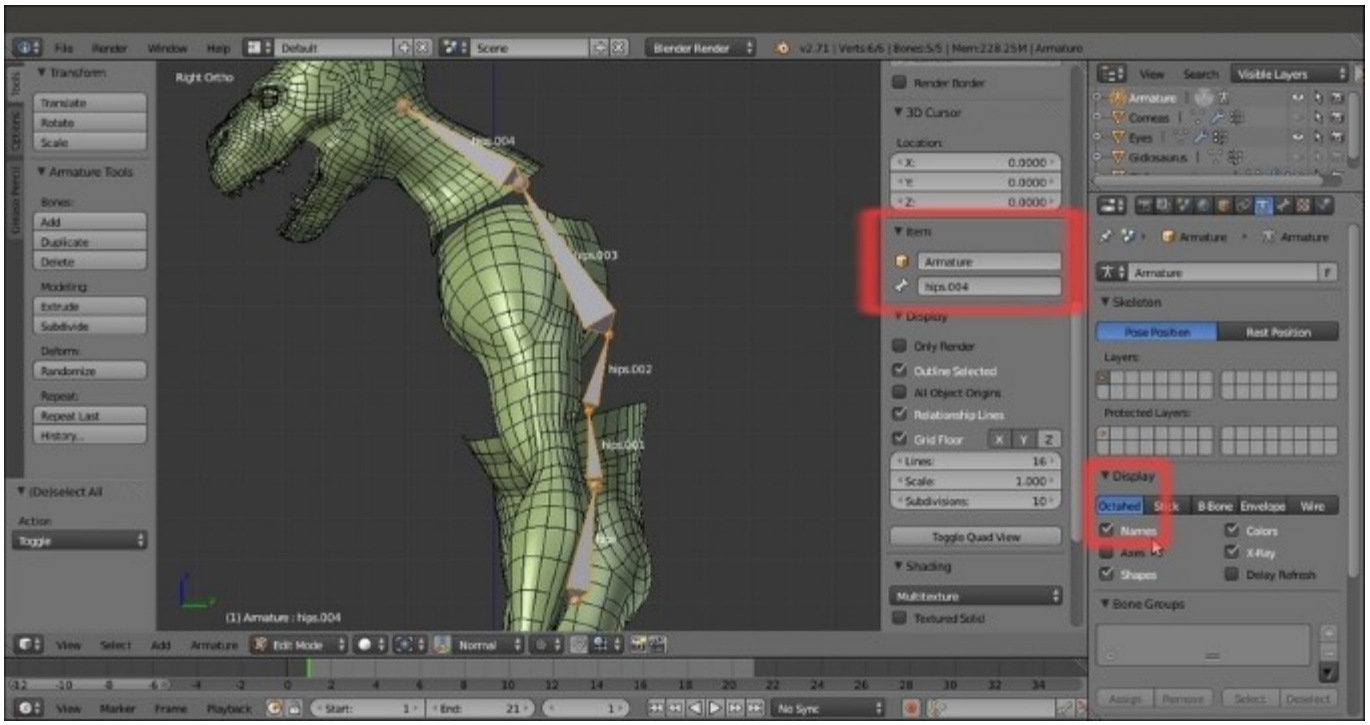
Renaming the bone

5. Press *Z* to go in the **Solid** viewport shading mode, and then go to the **Object Data** window and enable the **X-Ray** item under the **Display** subpanel.
6. With the tip of the bone selected (the **Head**), press the *E* key to extrude it. By this process, and by following the wire topology visible on the mesh as a guide, go upwards to build the **Gidiosaurus spine** (2 bones), **chest** (1 bone), and **neck** (1 bone); as much as possible, try to place the **Heads** (the tips/joints) of the bones aligned with the transversal edge-loops on the mesh's *articulation*:



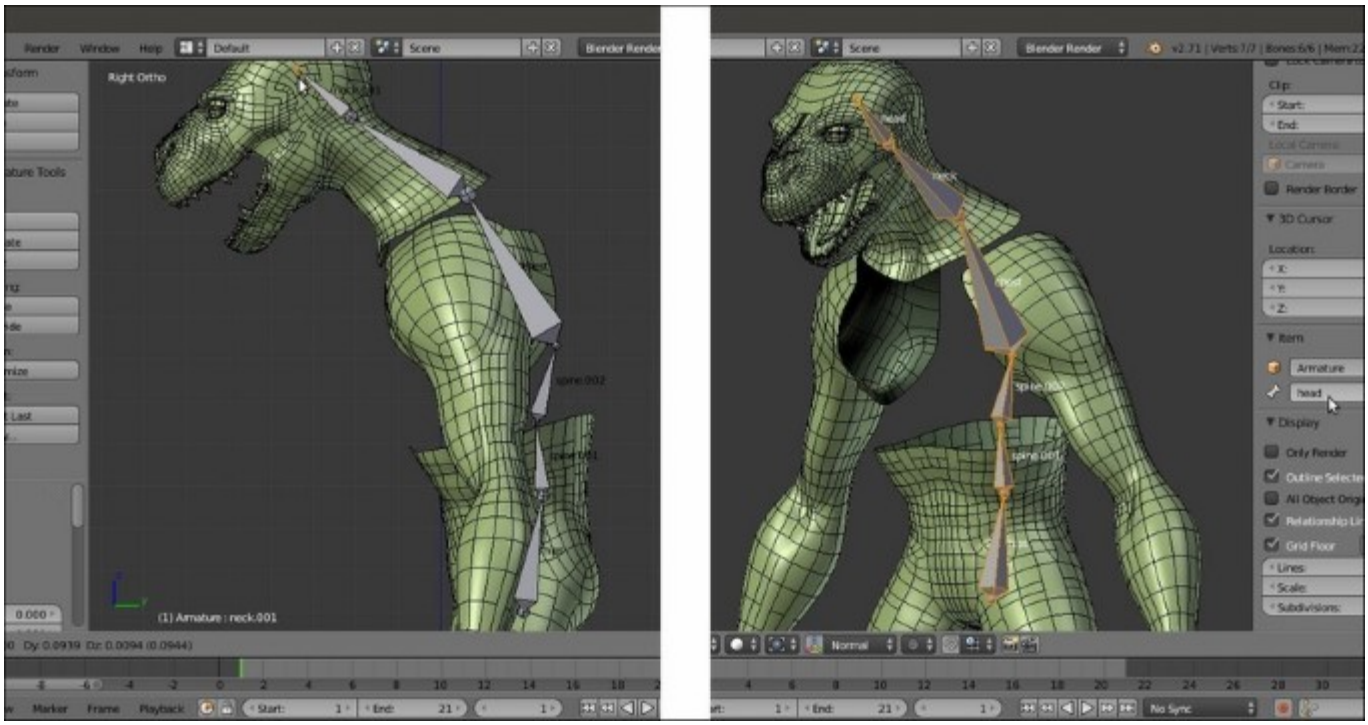
Extruding the bone to build the spine

7. Go again to the **Object Data** window under the **Display** subpanel, and enable the **Names** item (in the following screenshot, all the bones have been selected just to highlight them and their respective names). As you can see in the screenshot, the extruded bones get their names from the previous one, so we have **hips**, then **hips.001**, **hips.002**, and so on:



The bones' names

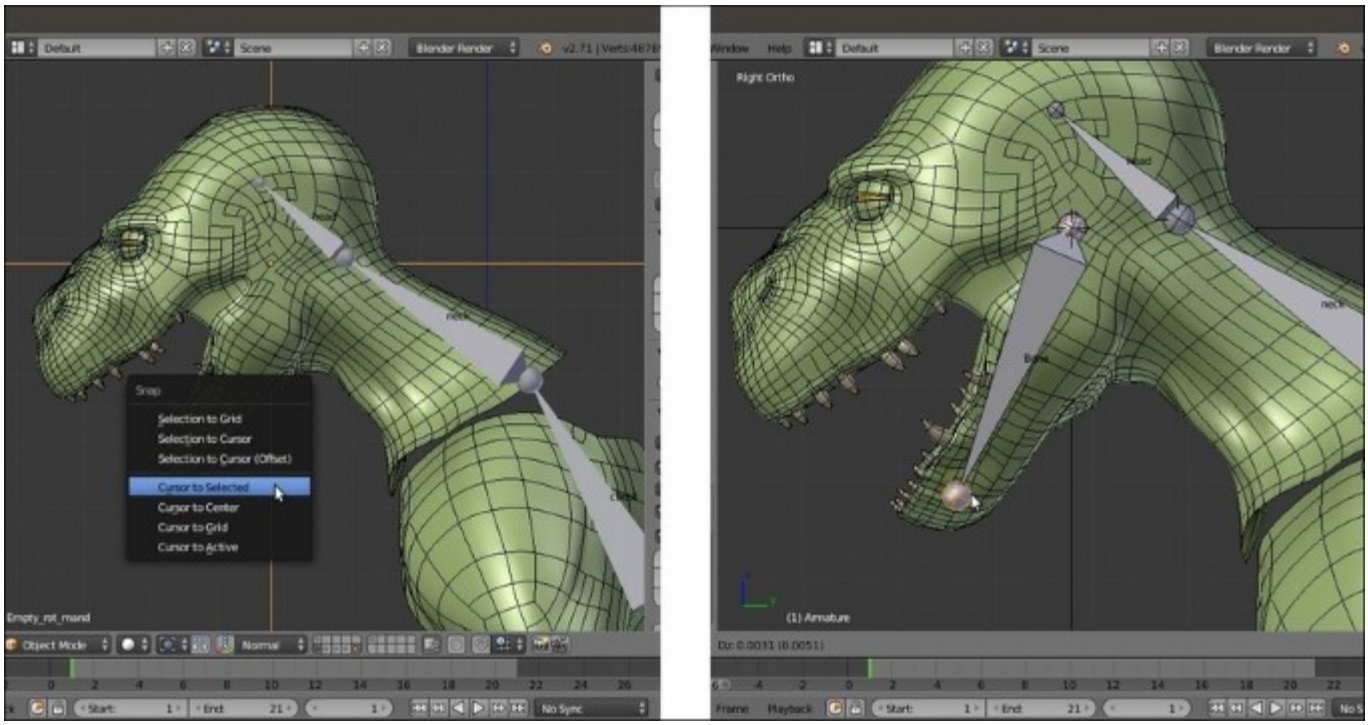
8. Select the **hips.001** bone and rename it **spine.001**; select the **hips.002** bone and rename it **spine.002**.
9. Select the **hips.003** bone and rename it **chest**; select the **hips.004** bone and rename it **neck**.
10. Select the tip of the **neck** bone and extrude it; rename the new bone (**neck.001**) as **head**:



The renamed bones and the head bone

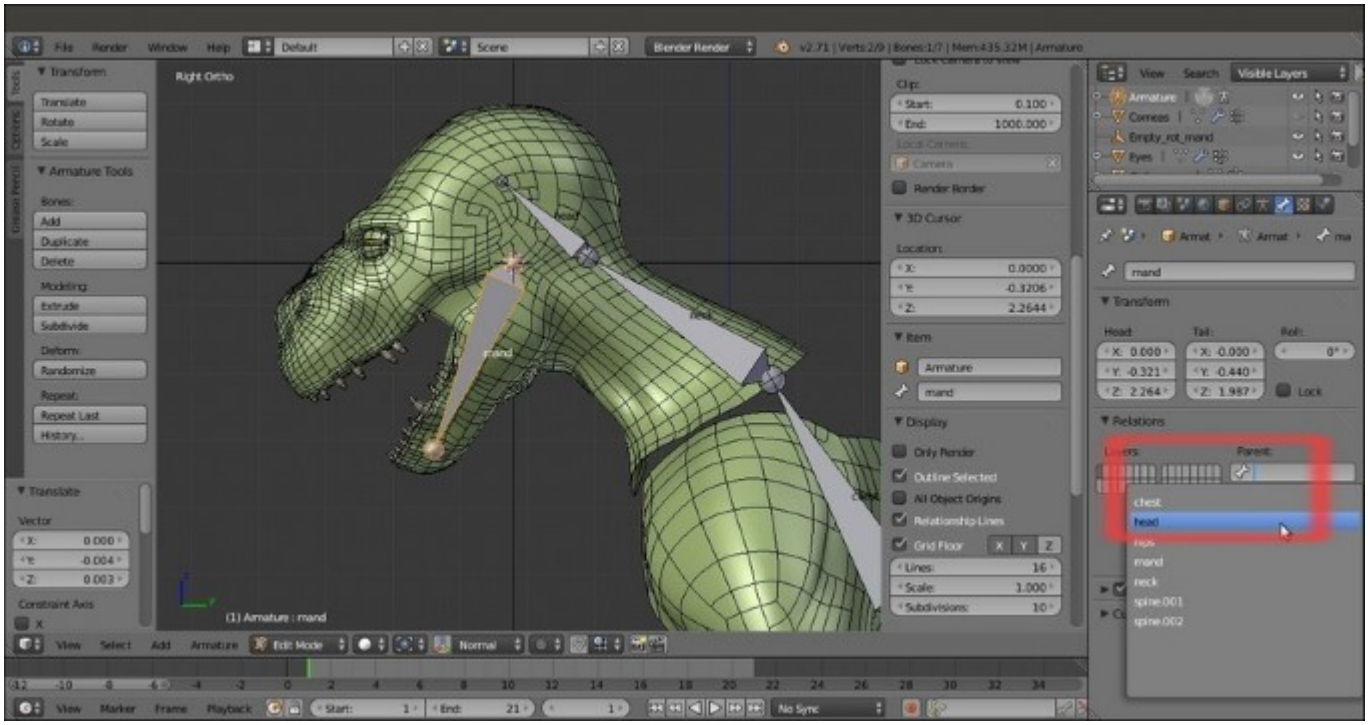
So, now we have built the **spine - neck** – **head** part of the **Armature**; actually, one thing is still missing: the bone to animate the **mandible**.

11. Press *Tab* to get out of **Edit Mode**. In the **Side** view, enable the **15th** scene layer on the 3D viewport toolbar, in order to show the **Empty_rot_mand** object; select it and press *Shift* + *S* to call the **Snap** pop-up menu. Then, select the **Cursor to Selected** item.
12. Reselect the **Armature** and go again into **Edit Mode**. Press *Shift* + *A* to add a new bone; move its **Head** to resize and fit it inside the **mandible** of the **Gidiosaurus**:



The mandible's bone

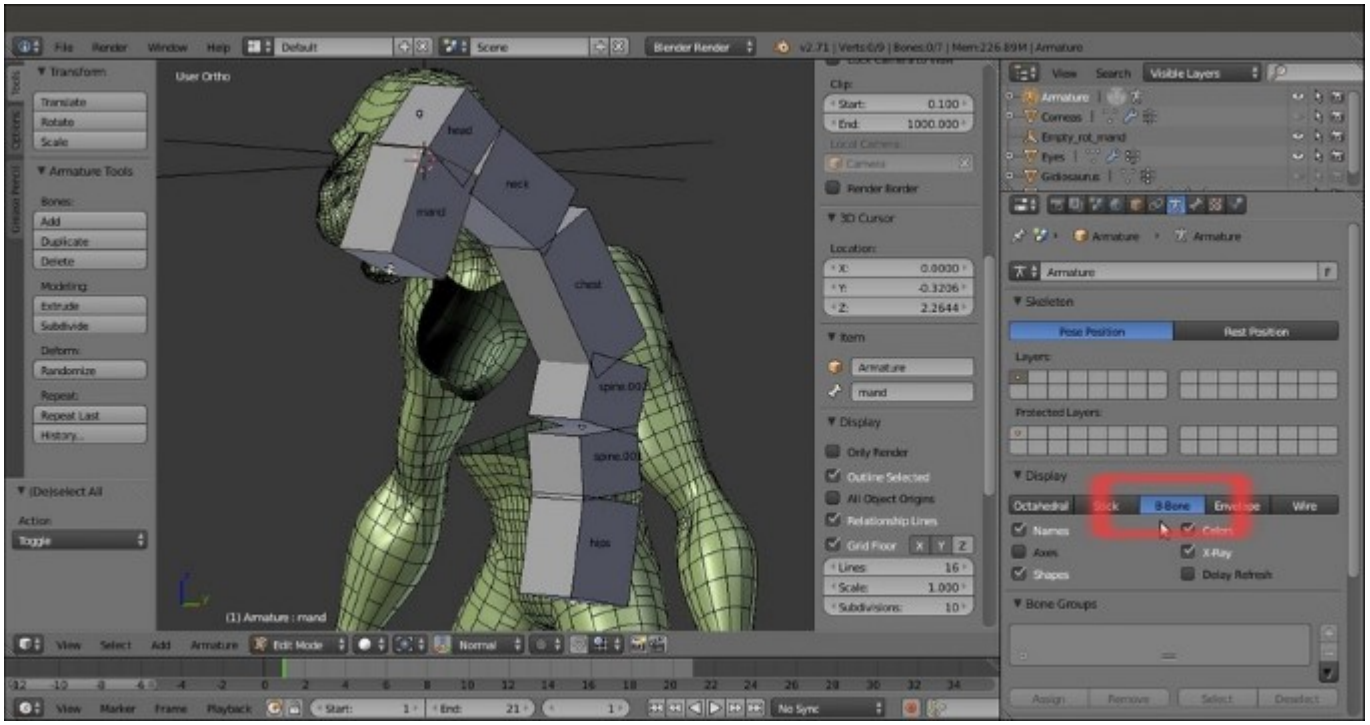
13. Rename it **mand** and in the **Bone** window under the main **Properties** panel, in the **Relations** subpanel, click on the **Parent** slot to select the **head** item from the pop-up menu with the bones list. Leave the **Connected** item **unchecked**:



The Parent slot and the pop-up menu to select the parent bone

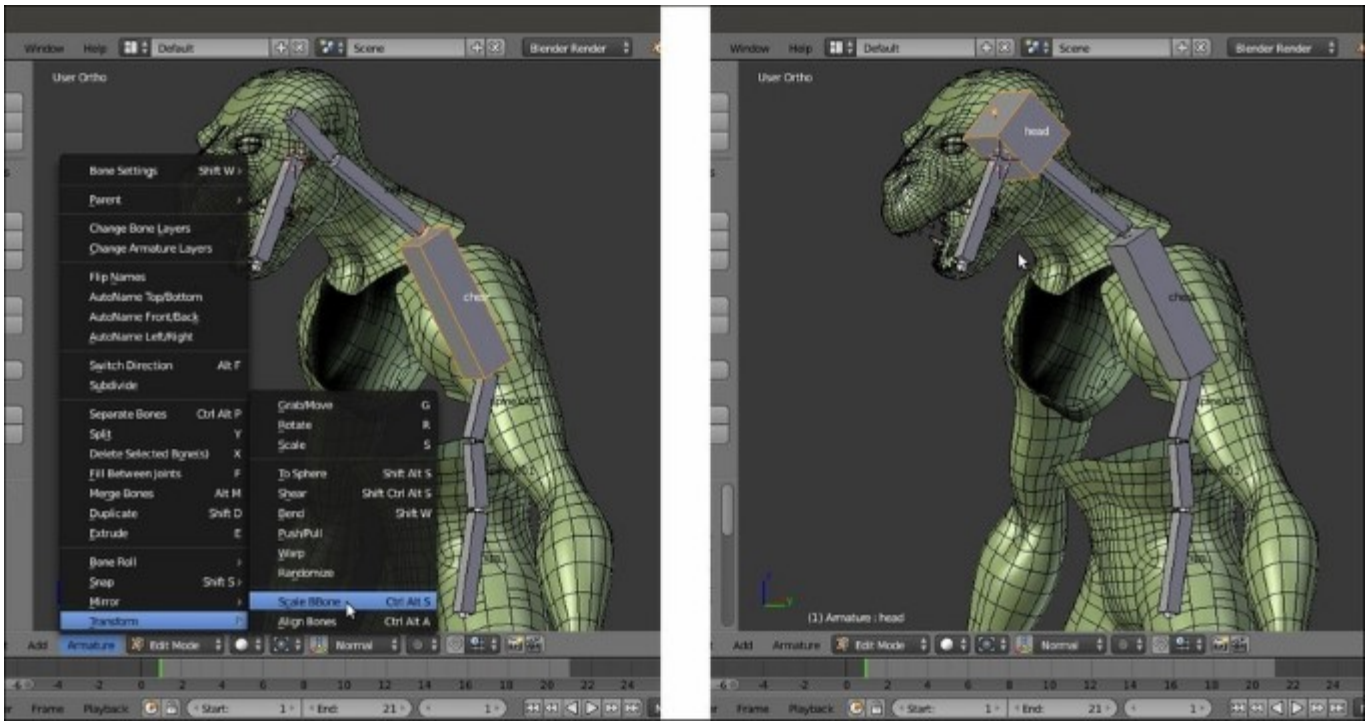
At this point, we can already see some particular setting to be applied to the bones.

14. Go to the **Object Data** window under the **Properties** panel and in the **Display** subpanel, switch from the default **Octahedral** to the **B-bone** button:



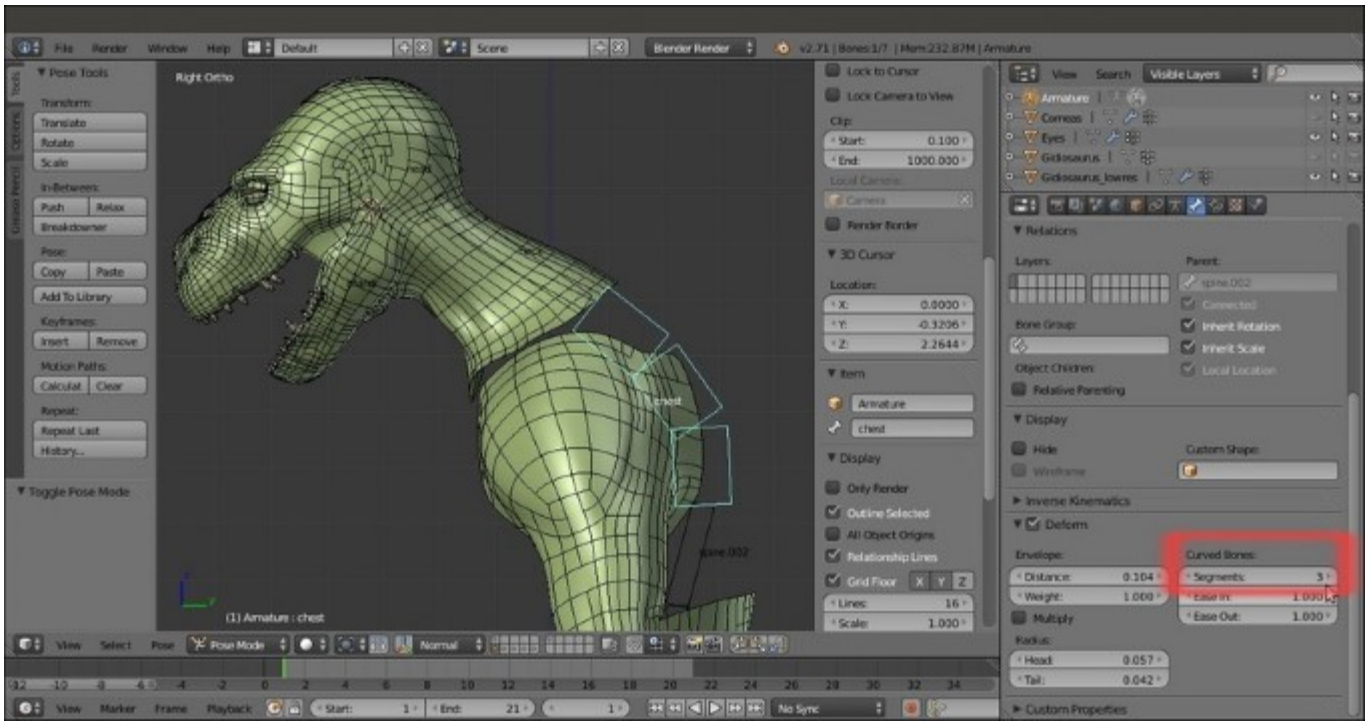
The bones visualized as B-bones

15. Press *A* to select all the bones, and then press *Ctrl* + *Alt* + *S* (or go to the **Armature** item in the window toolbar, and then go to **Transform** | **Scale Bbone**) and scale the **B-bones** to **0.200** (hold the *Ctrl* key to constrain the scaling values; the B-bones scaling works both in **Edit Mode** and **Pose Mode**).
16. Select only the **chest** bone and scale it bigger to **2.500**; select the **head** bone and scale it to **4.000**:



The B-bones scaled for better visualization

17. Go to the **Object** window and under the **Display** subpanel, click on the **Maximum Draw Type** slot (set to **Textured** by default) and switch it to **Wire**.
18. Press *Ctrl* + *Tab* to switch the **Armature** directly from **Edit Mode** to **Pose Mode**. Right-click on the **chest** bone to select it and go to the **Bone** window under the main **Properties** panel; in the **Deform** subpanel, set **Segments** under the **Curved Bones** item to **3**:

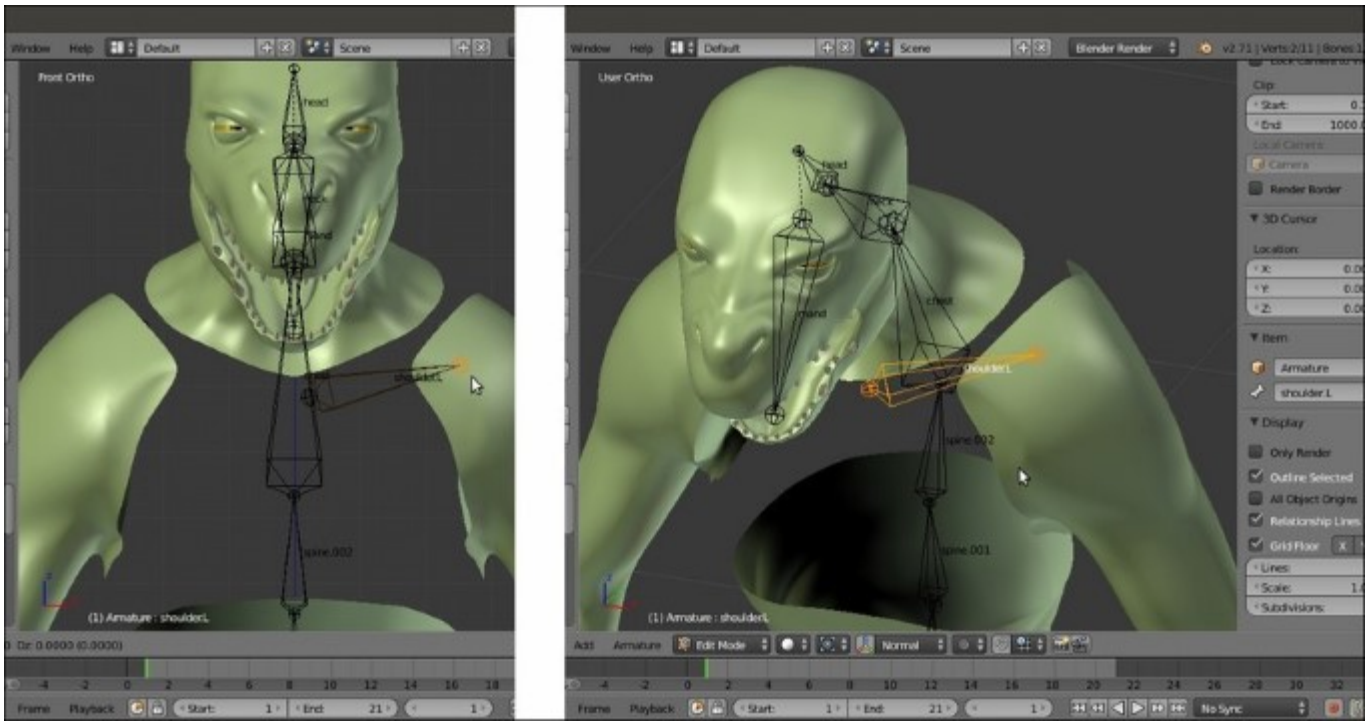


The chest B-bone with 3 curved segments

19. Select the **spine.002** and **spine.001** bones and set **Segments** to **2**. Select the **neck** bone and set **Segments** to **3**.
20. Select the **Gidiosaurus** mesh, go to the **Object** window, and disable the **Wire** item under the **Display** subpanel:

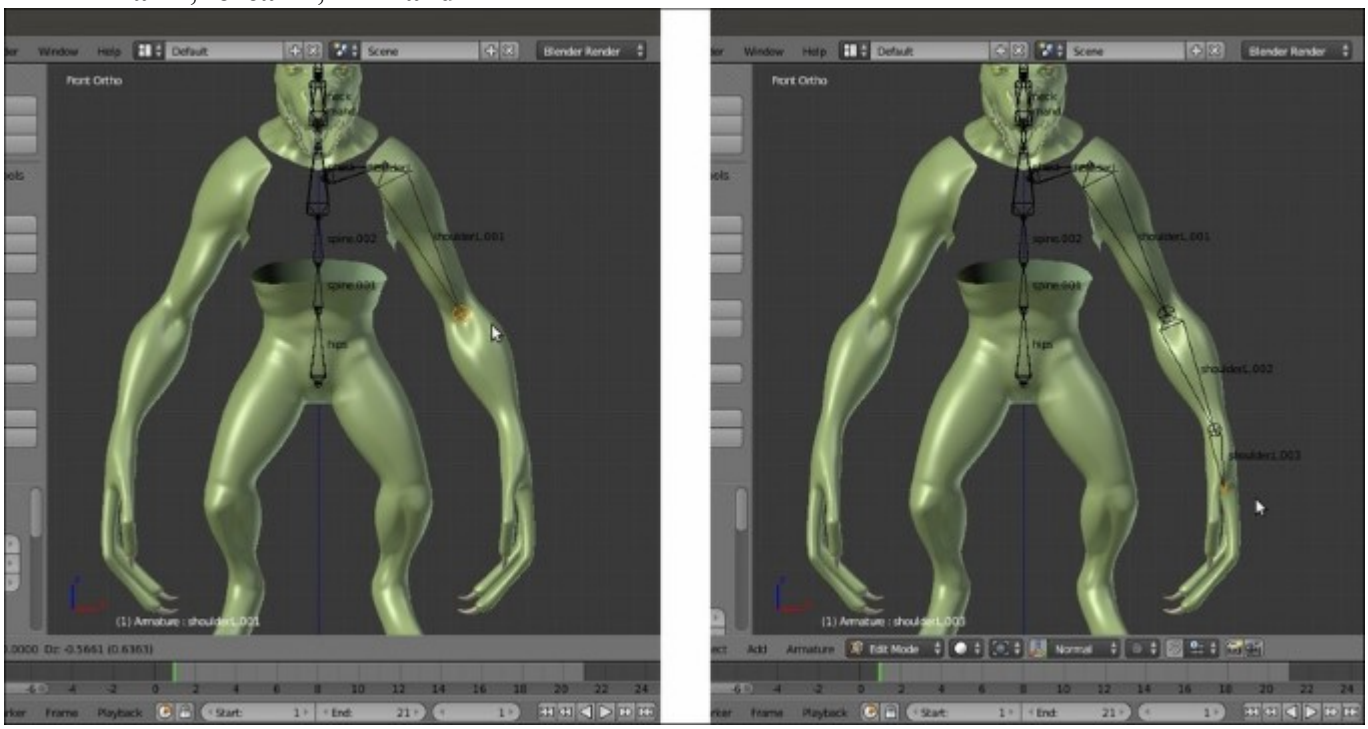
The rig so far

21. Press *Ctrl* + *Tab* to go out of the **Pose Mode**, and then *Shift* + *S* | **Cursor to Selected** to put the **3D Cursor** at the *rig/mesh/center of the scene* pivot point.
22. Press *Tab* to go into **Edit Mode** and press the *1* key on the numpad to go in the **Front** view; go to the **Object Data** window, under the **Display** subpanel, and switch back from **B-bone** to **Octahedral** (even if the visualization mode is different, the bones set as **B-Splines** still keep their *curved* properties in **Pose Mode**).
23. Press *Shift* + *A* to add a new bone at the cursor position. Move and resize it to put it as the **clavicle** bone—almost horizontal and slightly backward oriented, on the left-hand side of the rig. Rename it **shoulder.L** and in the **Parent** slot under the **Relations** subpanel, select the **chest** item:



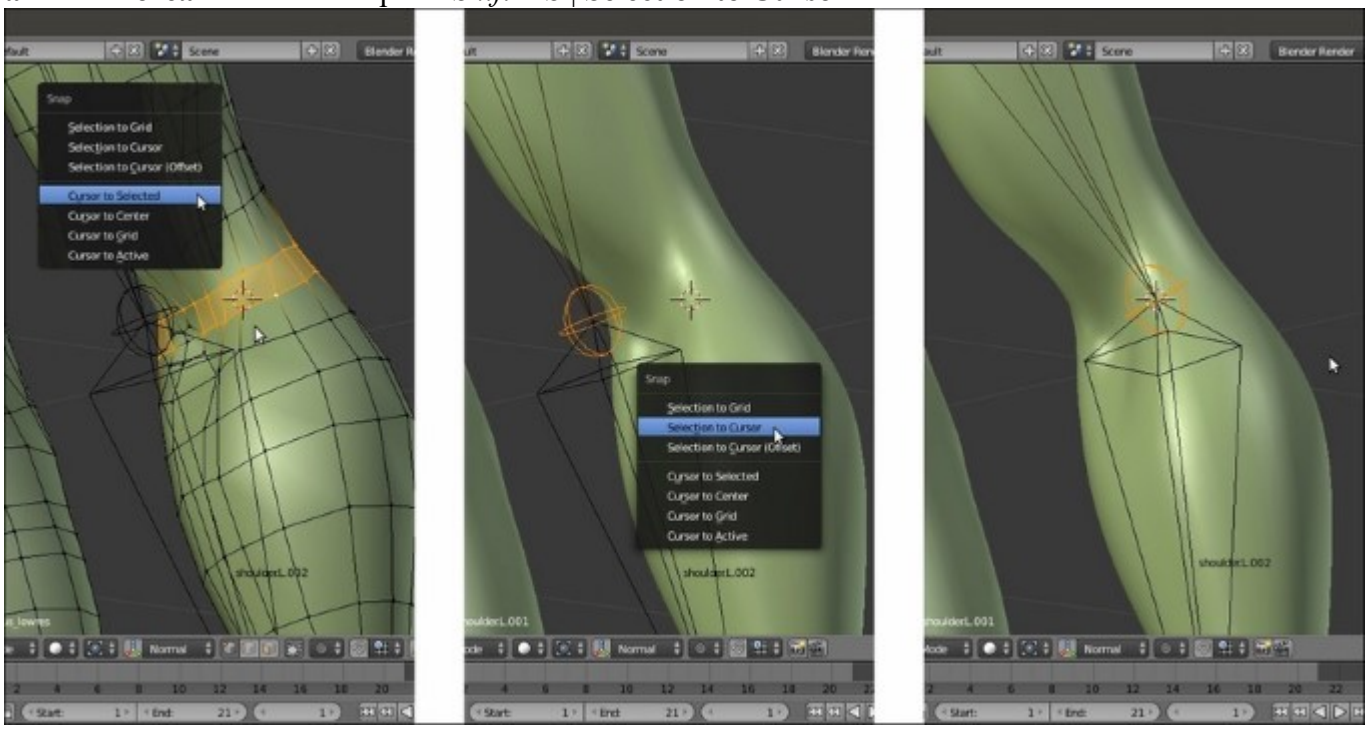
The shoulder.L bone

24. In the **Front** view, select the **Head** of the **shoulder.L** bone and extrude it **3** times to build the bones for **arm**, **forearm**, and **hand**:



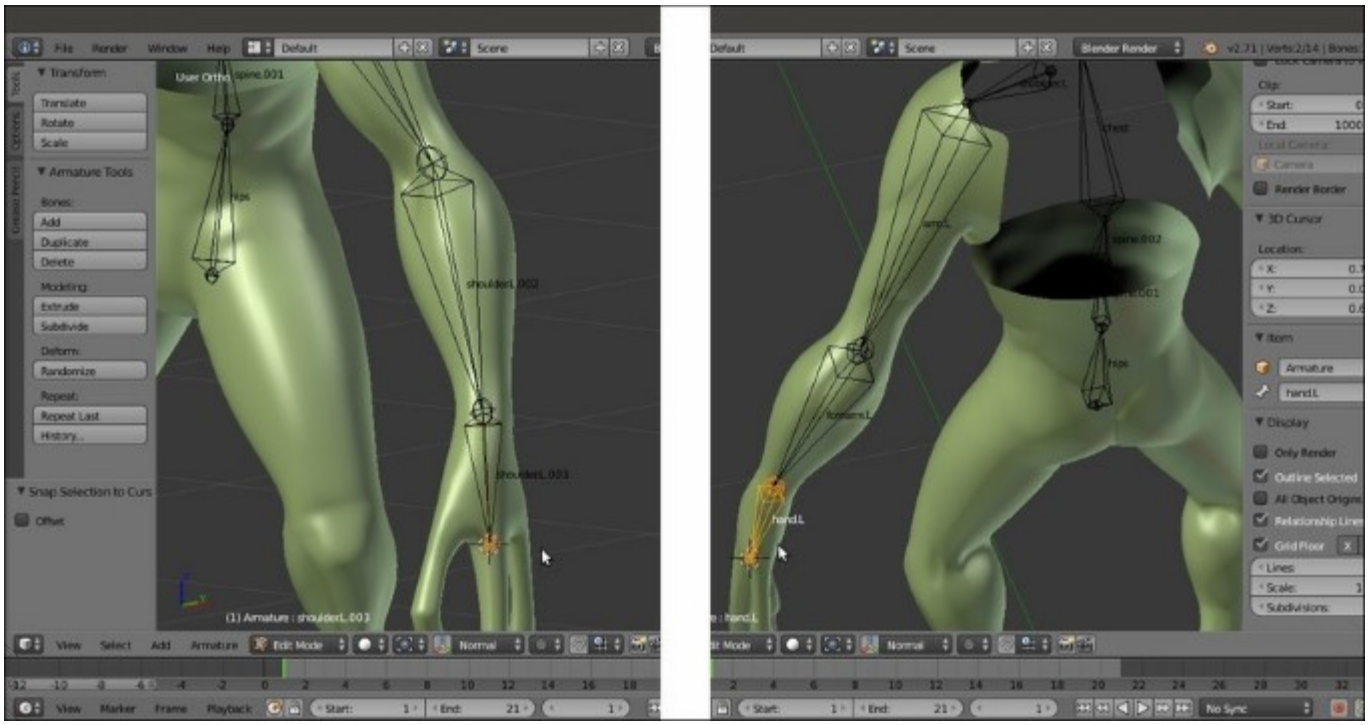
Extruding the shoulder.L bone to obtain the skeleton's bones for the arm

25. Now, exit **Edit Mode** and right-click to select the **Gidiosaurus** mesh; enter **Edit Mode**, select one or more edge-loops at the **elbow** level, and press *Shift* + *S* | **Cursor to Selected**.
26. Get out of **Edit Mode**, select the **Armature**; go into **Edit Mode**, select the joint between the **arm** and **forearm** bones and press *Shift* + *S* | **Selection to Cursor**:



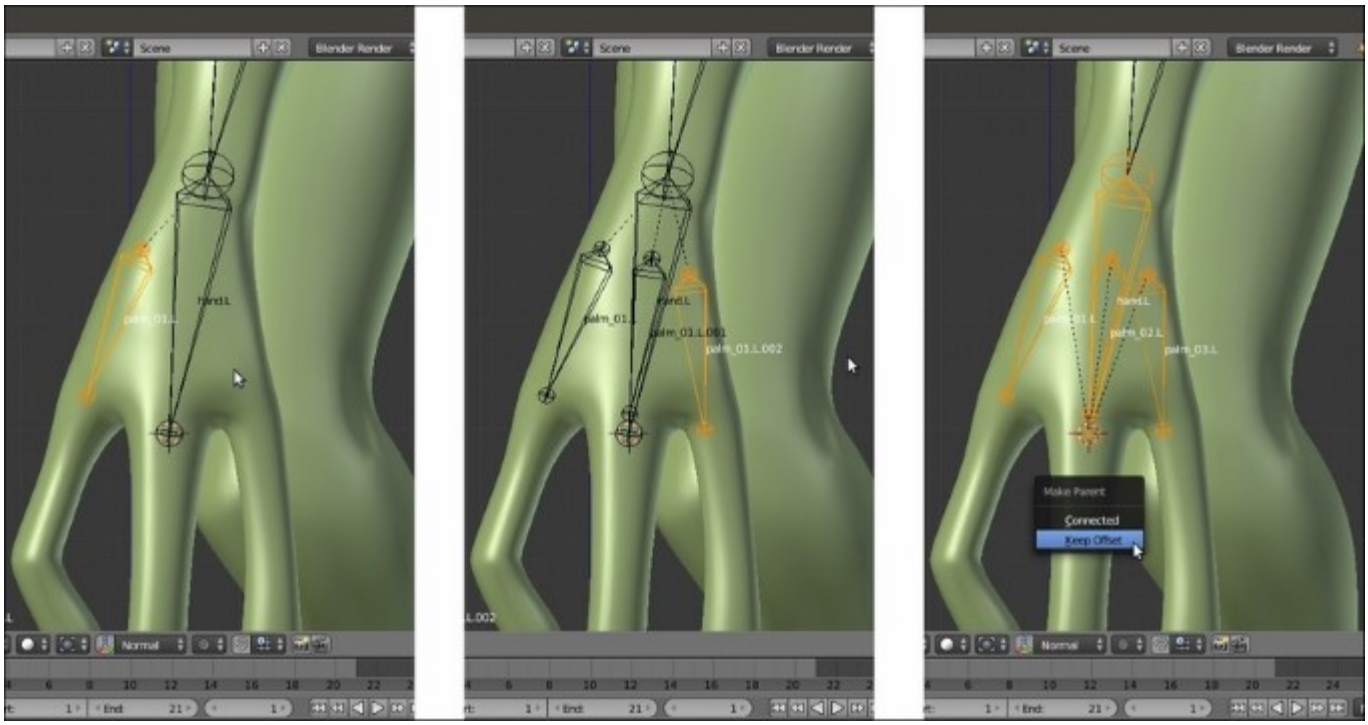
Placing the elbow joint

27. This is the easiest way to correctly align the rig joints with the mesh edge-loops. Do the same for the joint of the **wrist** and the bone of the **hand**; rename the bones as **arm.L**, **forearm.L**, and **hand.L**:



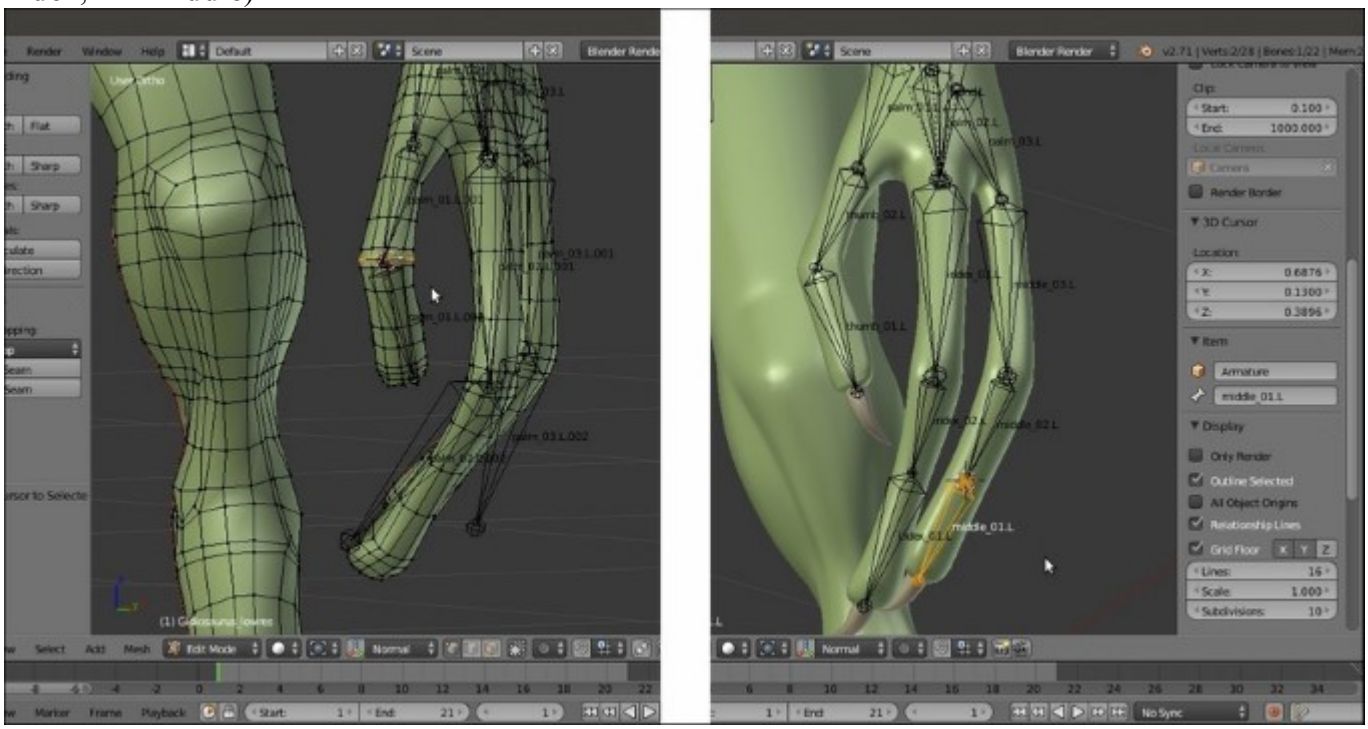
Fixing the position of the wrist joint and hand's bone

28. Select the **hand.L** bone and use *Shift* + *D* to duplicate it; scale it smaller (*S* | **0.600** | *Enter*), rename it **palm_01.L**, and move it above the joining of the **palm** with the **thumb**. Use *Shift* + *D* to duplicate it **2** more times and move the new bones above the joining of the other two **fingers**; rename them **palm_02.L** and **palm_03.L**.
29. Use *Shift* to select the three **palm** bones and, as the last one, the **hand.L** bone; press *Ctrl* + *P* | **Keep Offset** to parent them (**not connected**) to the latter one:



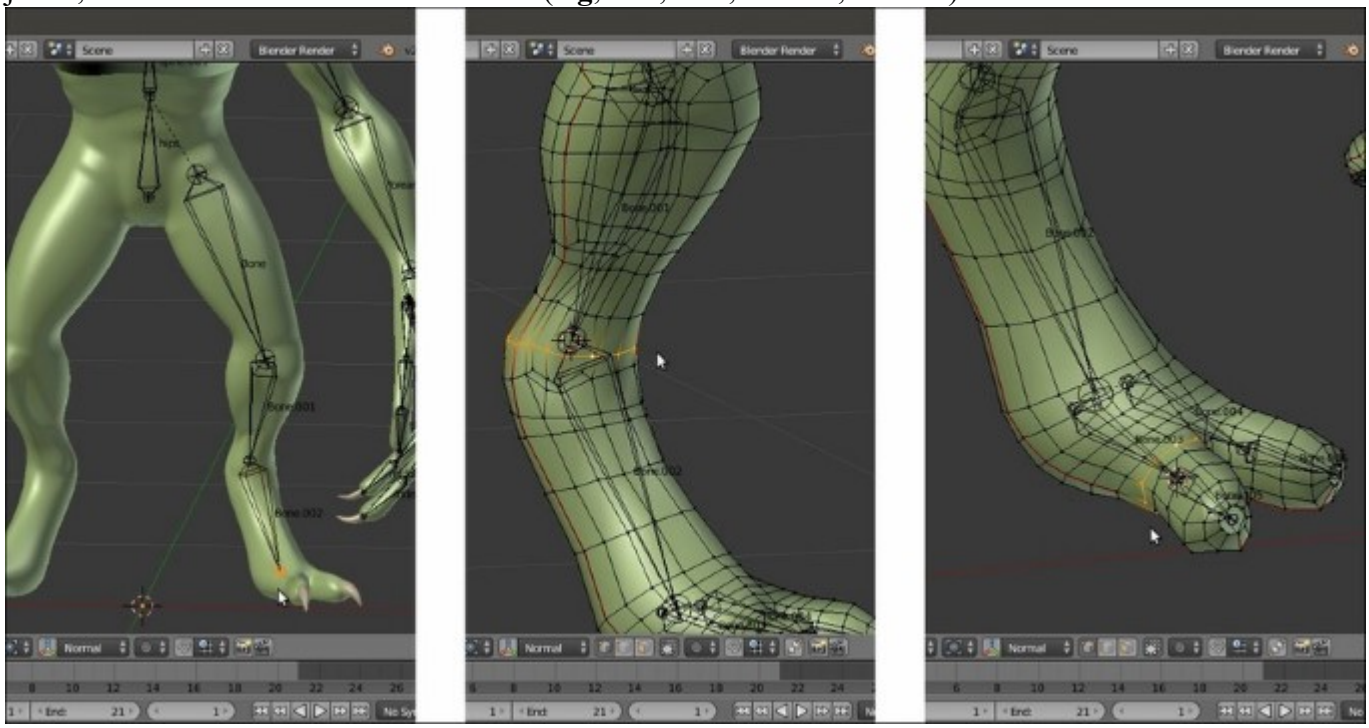
Adding the palm bones

30. Select (individually) the **Heads** of each **palm** bones and extrude the bones for the **fingers**; center their joints with the **3D Cursor**/**Snap** menu method and rename them properly (**thumb**, **index**, and **middle**):



The bones for the fingers

31. Again with the **3D Cursor** at the rig pivot point, add a new bone and shape it to fit inside the left **thigh**, the **Tail** at the top, close to the **hips** bone, and the **Head** at the **knee** location; select it and use *Shift* to select the **hips** bone, and then press *Ctrl* + *P* | **Keep Offset**. Extrude the bone's **Head** three times to build the **leg – foot** skeleton.
32. Extrude also the bones for the **toes** and repeat the previously described process to center the joints, and then rename all the new bones (**leg**, **calf**, **foot**, **toe inn**, and **ext**):



The bones for the leg and toes

In the preceding screenshots, you can see that we have hidden the **talons** vertices in **Edit Mode** (*H* key), in order to have the possibility to easily select the last edge-loops on **fingers** and **toes**.

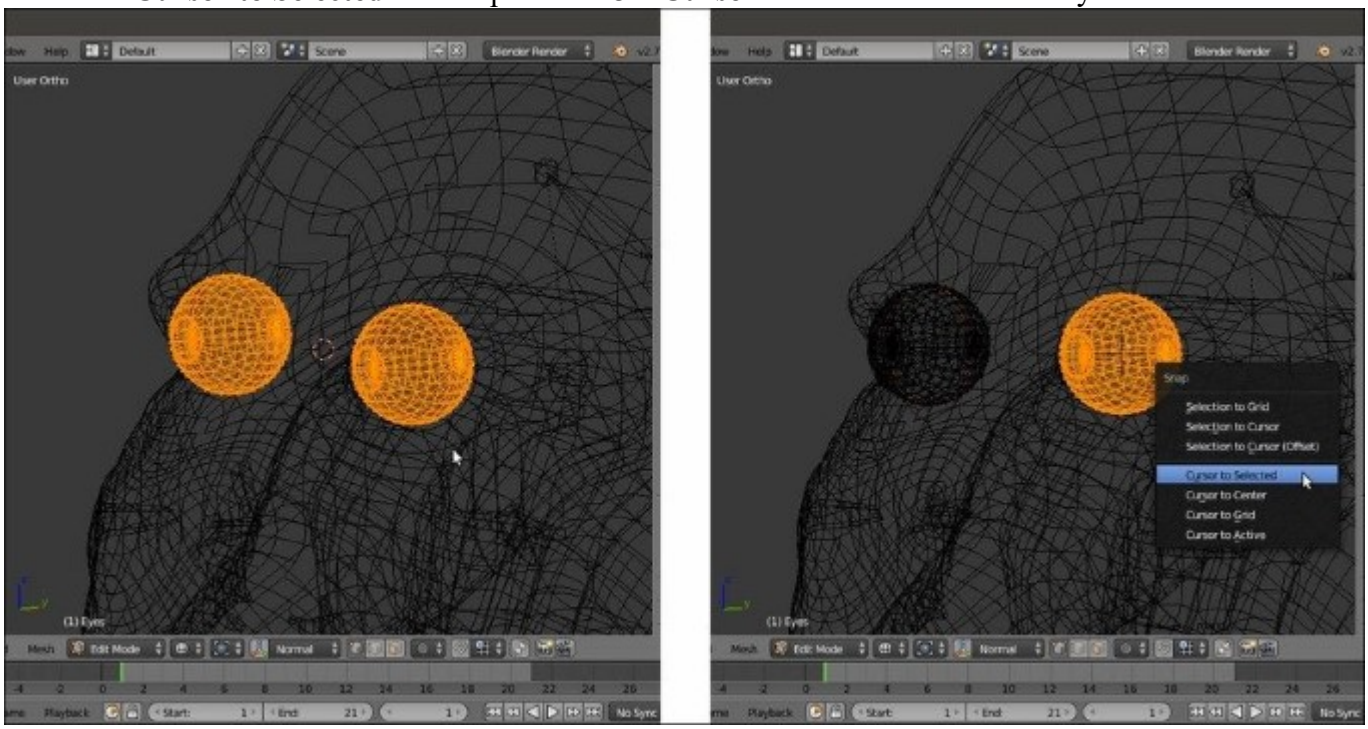
33. Save the file as `Gidiosaurus_rig_from_scratch_01.blend`.

Building the rig for the secondary parts

Now that we have completed the main body rigging system, it's time to build the rig for **eyes**, **eyelids**, and **tongue**:

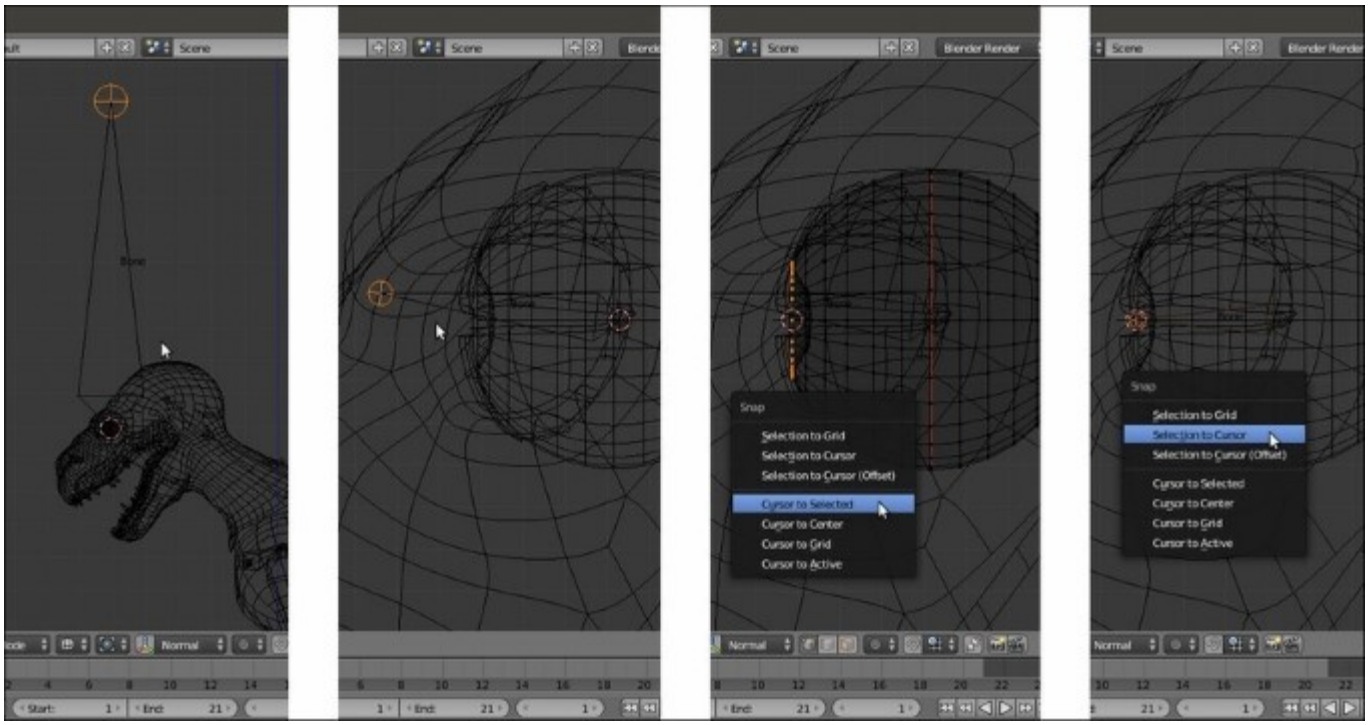
1. Get out of **Edit Mode** and select the **Eyes** item in the **Outliner**; press the dot (.) key on the numpad to center the view on the selected object, the *Z* key to go in **Wireframe** viewport shading mode, and *Tab* to go into **Edit Mode**.

2. Press the *A* key to select all the **eye** vertices and then box-deselect (the *B* key and the middle mouse button) the vertices of the **right eye**; use *Shift* + *S* to call the **Snap** pop-up menu and select the **Cursor to Selected** item to place the **3D Cursor** at the center of the left eye mesh:



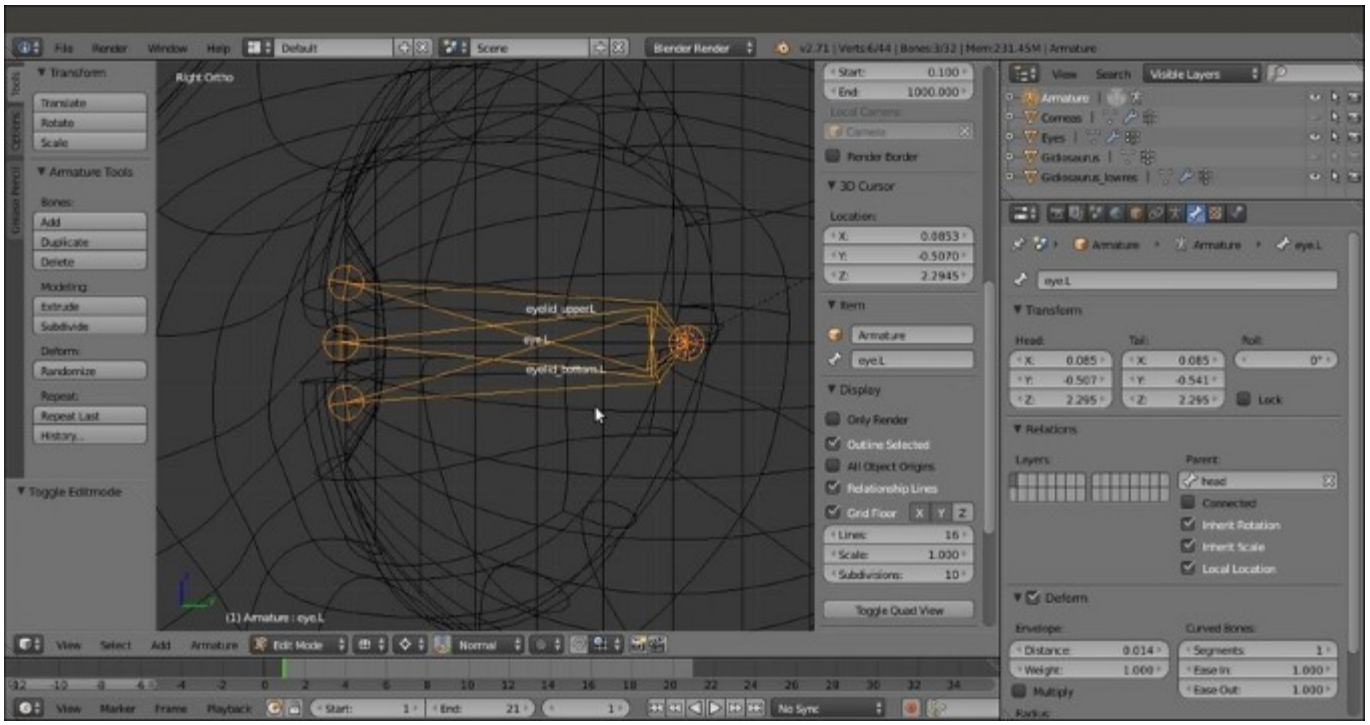
Placing the 3D Cursor at the center of the selected vertices

3. Get out of **Edit Mode**, press the *3* key on the numpad to go in the **Side** view and reselect the **Armature** item in the **Outliner**; press *Tab* to go into **Edit Mode**, and then use *Shift* + *A* to add a new bone at the cursor position. Press *G* to grab the already selected **Head** of the new bone and move it close to the center of the **eye** to resize it smaller.
4. Get out of **Edit Mode** and select the **Eyes** item; enter **Edit Mode** and deselect all the vertices except for the external last **iris** edge-loop. Then, press *Shift* + *S* | **Cursor to Selected** and get out of **Edit Mode**.
5. Again, select the **Armature**, go into **Edit Mode**, be sure that the **Head** of the new bone is still selected, and press *Shift* + *S* | **Selection to Cursor**:



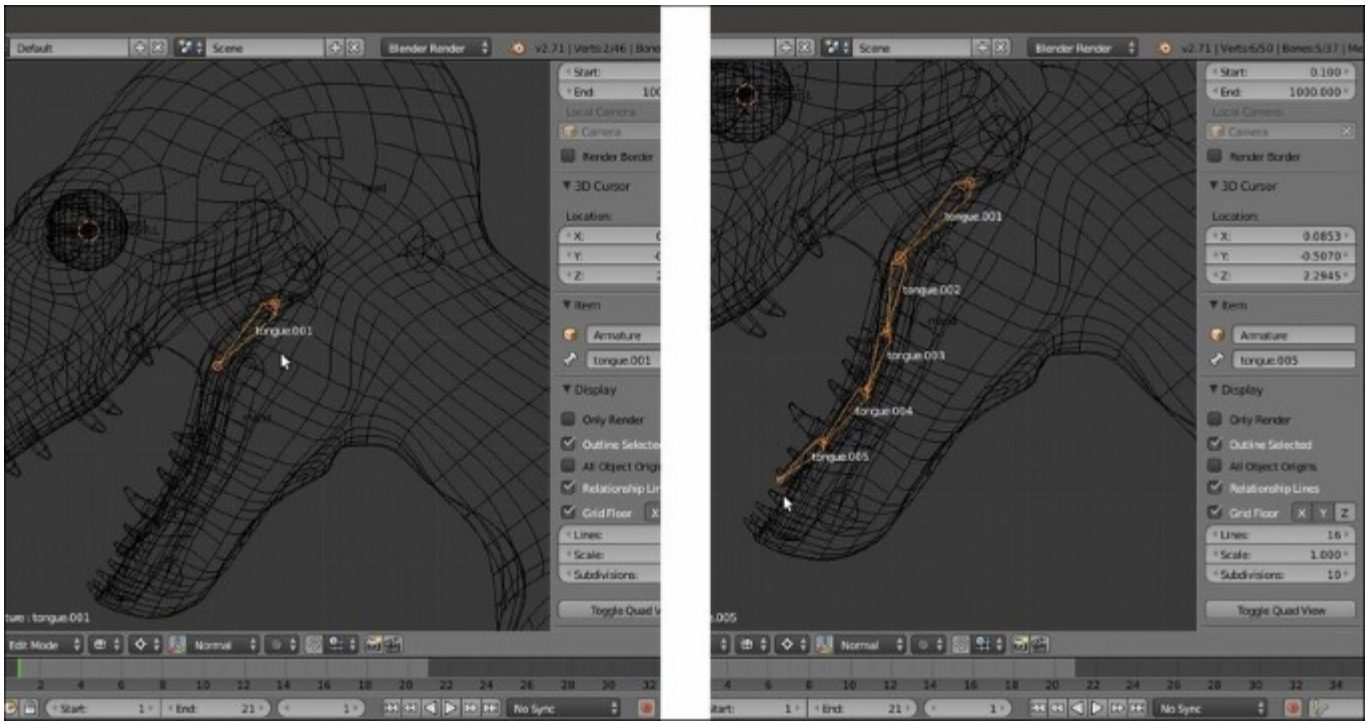
Placing the bone's head at the iris center location

6. Rename the new bone as **eye.L** and in the **Relations** subpanel under the **Bone** window, parent it to the **head** bone (not **Connected**), or use *Shift* to select the **eye.L** and **head** bones and press *Ctrl* + *P* | **Keep Offset**.
7. Now, select the **Tail** of the **eye.L** bone and press *Shift* + *S* | **Cursor to Selected** to put the **3D Cursor** on it, and then press the period (.) key to switch the **Pivot Point** around the **3D Cursor**; select the whole **eye.L** bone and use *Shift* + *D* to duplicate it, and soon after, click with the right mouse button to leave the duplicated bone untouched; rotate it **10** degrees clockwise on the cursor position (*Shift* + *D* | right-click | *R* | *X* | **10** | *Enter*).
8. Rename the new bone as **eyelid_upper.L**.
9. Reselect the whole **eye.L** bone and repeat the duplication procedure; rotate the new duplicate **10** degrees counterclockwise (*Shift* + *D* | right-click | *R* | *X* | **-10** | *Enter*).
10. Rename the new bone as **eyelid_bottom.L** (in the following screenshot, all the three new bones—**eyelid_upper.L**, **eye.L**, and **eyelid_bottom.L**—have been selected just to enhance their visibility):



The bones for the eye and eyelids

11. Now, duplicate the **head** bone, resize it smaller, and move it to the joining of the **tongue** with the **inner mouth**; rename it from **head.001** to **tongue.001** and in the **Relations** subpanel, change its parenting from **neck** to **mand**.
12. Select the **Head** of the **tongue.001** bone and press the *E* key to extrude **4** new bones:



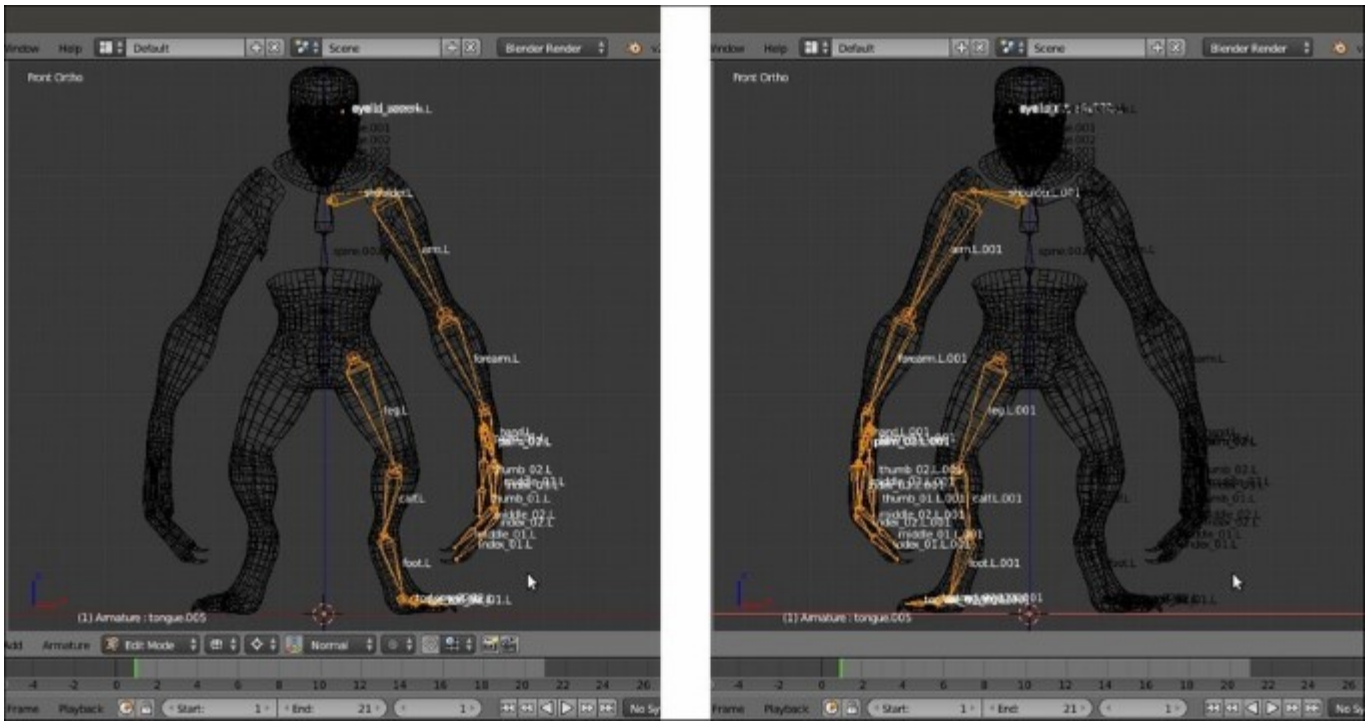
The tongue bones

13. Rename them accordingly, and then use *Shift* to select from the **tongue.001** to **tongue.005** bones and press *Ctrl* + *R*; move the mouse pointer horizontally to *roll* them on their *y* axis by **180°** (hold *Ctrl* to constrain the rolling to intervals of **5** degrees; alternatively, the roll value can also be set by typing it in the **Roll** button in the **Transform** subpanel under the 3D viewport **Properties** sidepanel).

Completing the rig

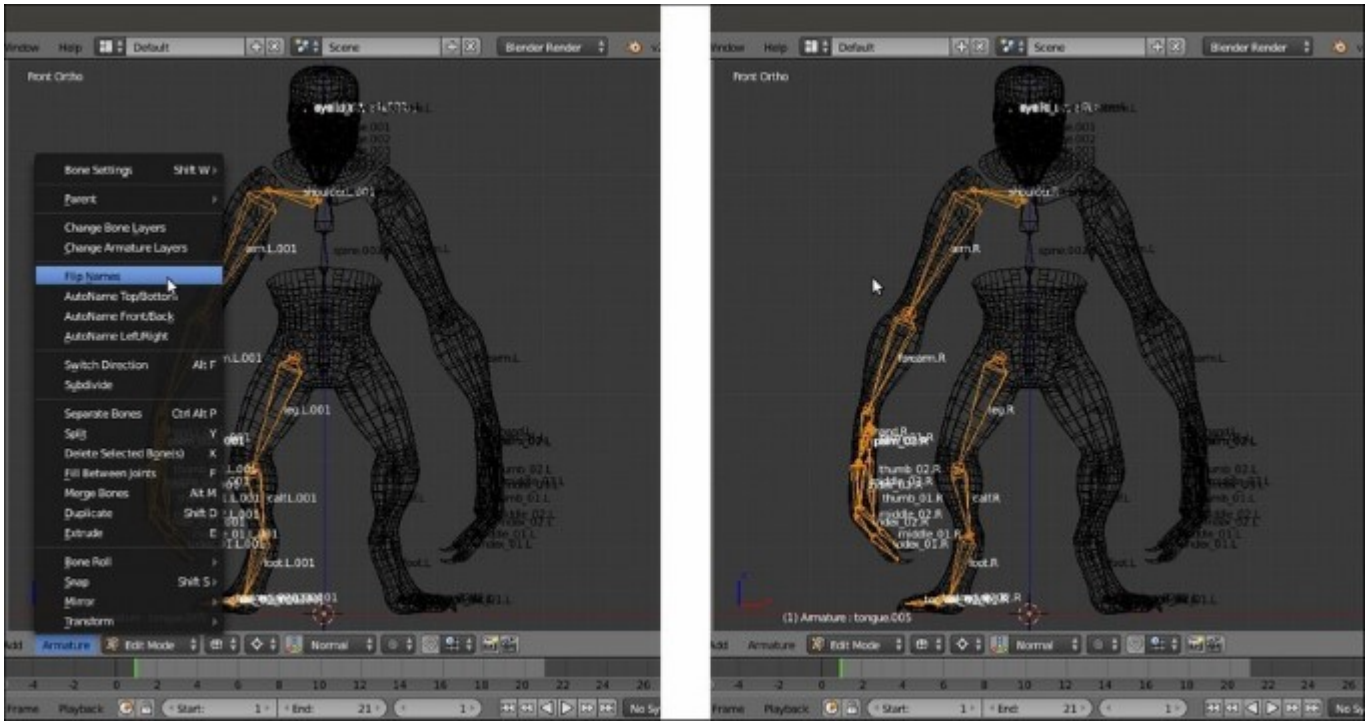
At this point, the basic rig building process is almost done, even if it is only for the left-half part of the mesh:

1. Get out of **Edit Mode** and press *Shift* + *S* | **Cursor to Selected** to place the **3D Cursor** at the median pivot point of the **Armature**.
2. Go back into **Edit Mode**, select only the left-half part bones and *not* the median ones (meaning: leave the **hips**, **spine**, **neck**, **head**, **mouth**, and **tongue** bones unselected), press *Shift* + *D* to duplicate them, and then right-click with the mouse button; press *Ctrl* + *M*, then the *X* key to mirror the duplicated bones on the *x* axis, in order to build the missing right-half part of the rig:



Mirroring the duplicated bones on the x axis

3. With the duplicated bones still selected, go to the **3D window toolbar** and click on the **Armature** item; in the pop-up menu, select the **Flip Names** item to automatically rename them with the correct **.R** suffix:



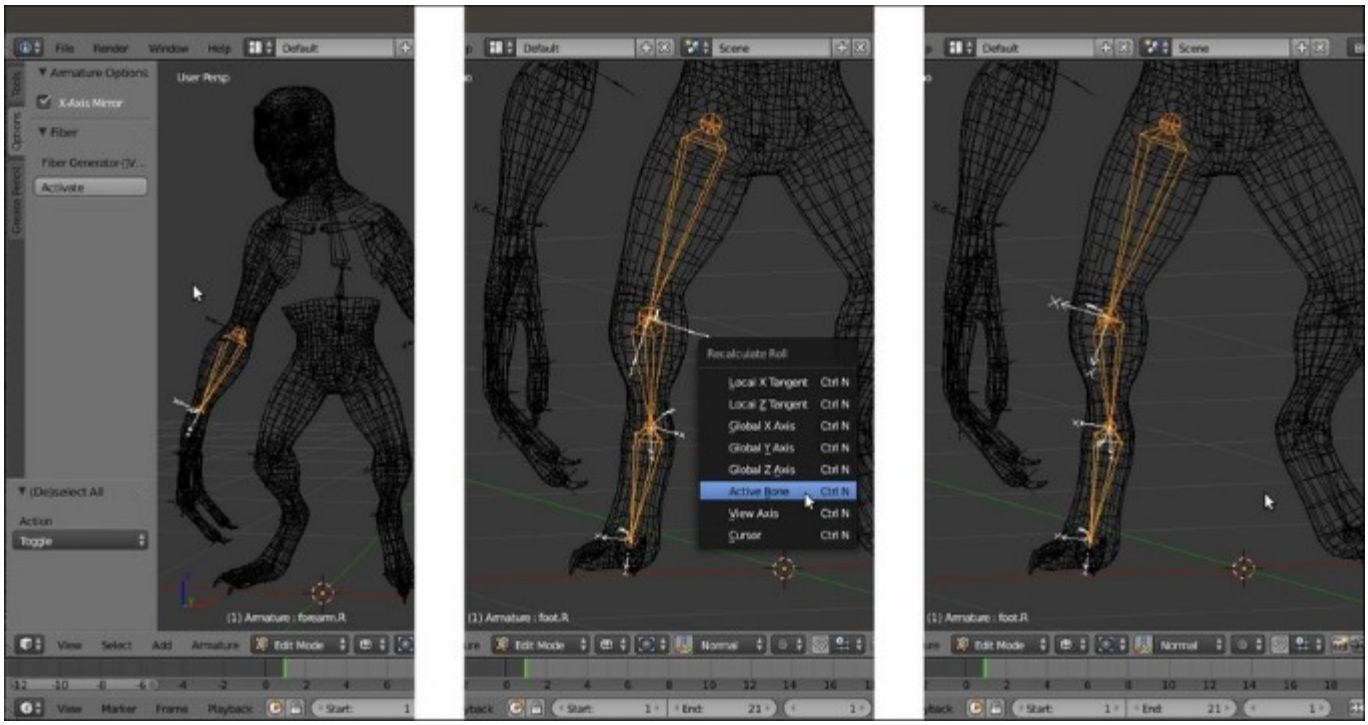
As a very last thing for this recipe, we must verify that the alignment of the bones, especially the last duplicated ones, is correct and, just in case, recalculate the roll rotation, that is, the rotation around the *y* axis of the bone itself.

4. In the **Object Data** window, under the **Display** subpanel, check the **Axes** item to make the bones orientation axes visible (only in **Edit Mode** and **Pose Mode**) in the 3D view.
5. Select all the bones and press *Ctrl* + *N* to recalculate the rolling of all of them; in the **Recalculate Roll** pop-up menu, there are several different options: because basically the *z* axis of the bones must match from the left to the right side of the whole rig, with the **Armature** (and the mesh) oriented along the *y* global axis, as in our **Gidiosaurus** case, the first top item, **Local X Tangent**, can be a good start.

   By the way, it is good practice to not trust this automated procedure alone, because sometimes it can give inconsistent results; so, do the following:
6. After the recalculation, check that the axes of each bone are actually correctly orientated in a consistent way; effectively, there are some bones that didn't get consistently oriented, meaning that their *x* and *z* local axes are oriented differently from the other bones.
7. In this case, select the incorrectly oriented bone, press *Ctrl* + *R*, and move the mouse to change the rolling; press the *Ctrl* key to constrain the rolling to intervals of **5** degrees. Alternatively, select the wrong bones, and then use *Shift* to select one bone that is correctly oriented and press *Ctrl* + *N* | **Active Bone** to copy the rolling from the last selected bone.

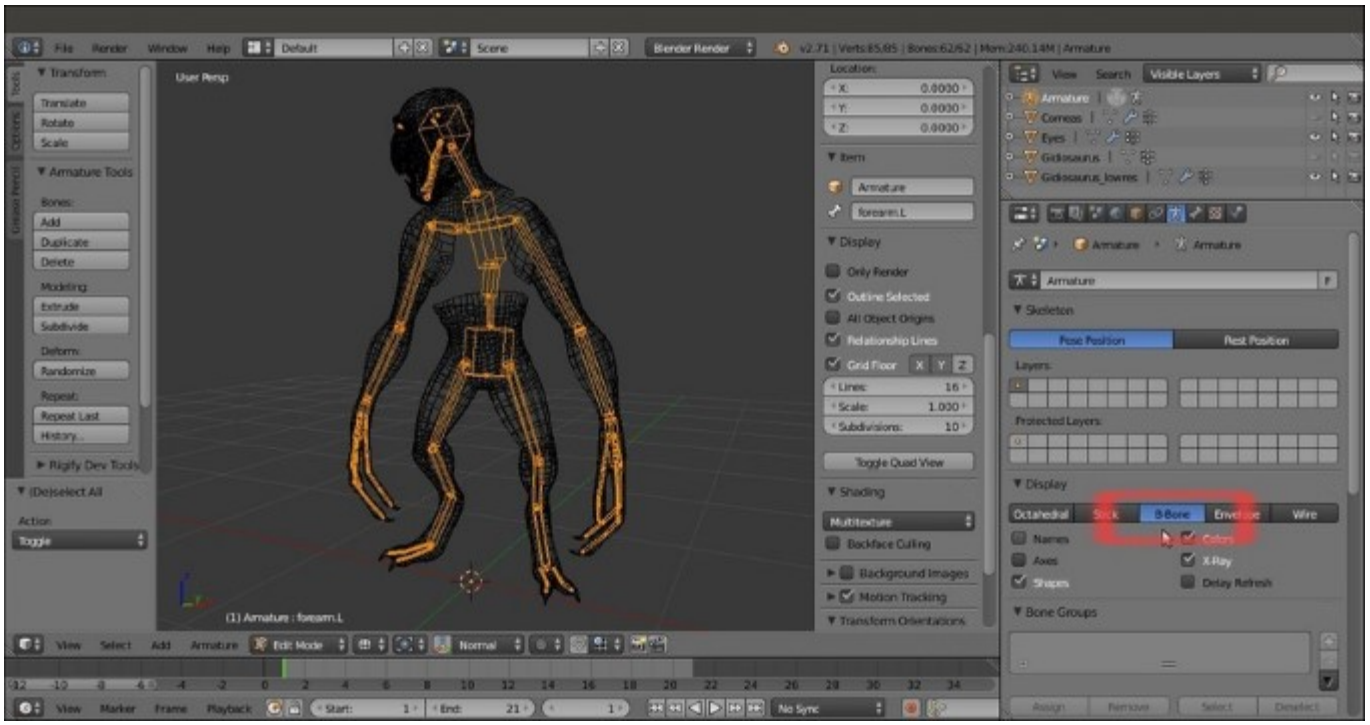
   By enabling the **X-Axis Mirror** item in the **Armature Options** tab under the **Tool Shelf**, you can recalculate only the bones of one side; the other side bones will follow automatically.

   If you want to make sure the bones' orientations are correct and everything is going to work in animation, just go into **Pose Mode** and rotate one bone, for example **leg.L**, and then click on the *Copies the current pose of the selected bones to copy/paste buffer* button (*Ctrl* + *C*), which is the first left one of the last three buttons to the right-hand side of the viewport toolbar; then, select the symmetrical bone, **leg.R**, and click on the last right button to paste the flipped pose (*Ctrl* + *Shift* + *V*); if the **leg.R** bone rotates correctly, then the orientation is OK:



Recalculating the roll of incorrectly oriented bones

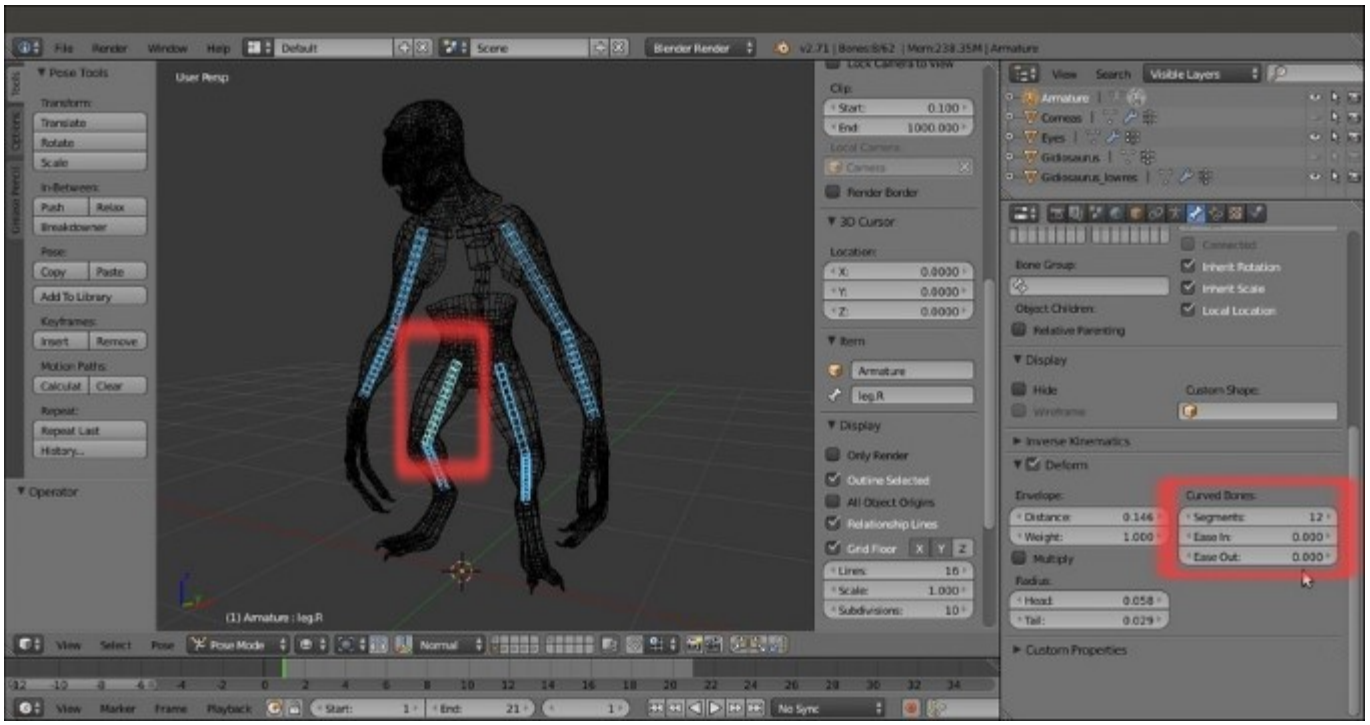
8. Now, go to the **Object Data** window under the main **Properties** panel and under **Display**, switch again the bones visualization from **Octahedral** to **B-bone**; select the bones and by pressing *Ctrl + Alt + S*, scale the **B-bones** smaller or bigger, depending on the visual effect you want to obtain:



The almost completed Armature in B-bones visualization

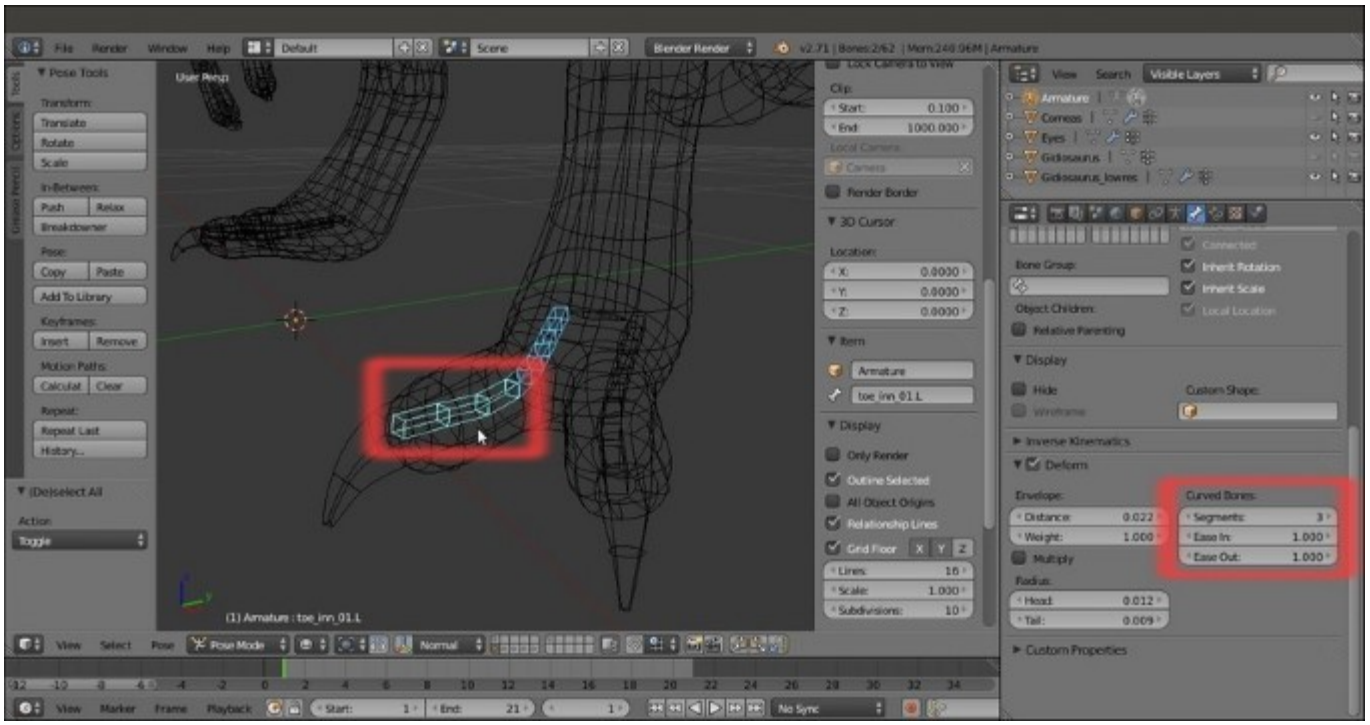
9. Press *Ctrl* + *Tab* to pass directly from **Edit Mode** to **Pose Mode** and select the **forearm.L** bone; in the **Deform** subpanel under the **Properties** panel, set the **Segments** for the **Curved Bones** to **12**, and then set the **Ease In** and **Ease Out** values to **0.000**.
10. Repeat this for the **forearm.R** bone and also for the **calf.L** and **calf.R** bones; repeat also for the **arm.L**, **arm.R**, **leg.L**, and **leg.R** bones.

    In the following screenshot, all the eight **B-bones** have been selected to make them more visible. By the way, the highlighted **leg.R** bone is the active one and shows the **Curved Bones** setting in the highlighted **Deform** subpanel to the right-hand side of the screen.



The Segments setting for the leg bone

11. Select the **toe_inn_02.L** bone and in the **Deform** subpanel under the **Properties** panel, set the **Segments** to **6** and leave the **Ease In** and **Ease Out** values to **1.000**.
12. Repeat this for the **toe_ext_02.L** bone; then, do the same also to the **toe_inn_02.R** and **toe_ext_02.R** bones.
13. Select the **toe_inn_01.L** bone and set the **Segments** to **3**; leave the **Ease In** and **Ease Out** values to **1.000**.
14. Repeat for the **toe_ext_01.L** bone; then, do the same also to the **toe_inn_01.R** and **toe_ext_01.R** bones:



The Segments setting for the toes bones

15. Save the file.

How it works…

Although it's often a really time consuming task, the handmade rigging is quite self-explicative; it is, however, better to explain some of the concepts behind this.

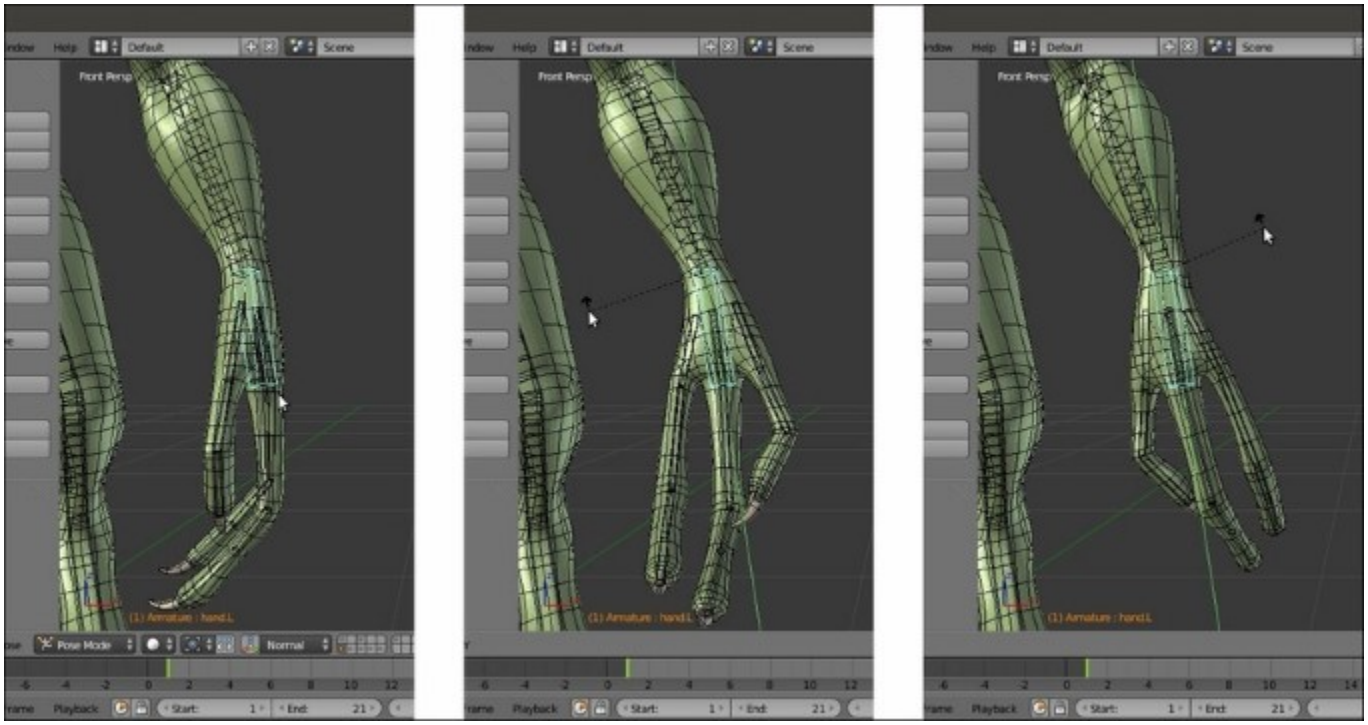
The proper renaming of the bones is important, considering that each deforming bone will affect a vertex group sharing the same name on the mesh; although in some cases, as for the **tongue** bones, the bone naming process can be automated in some way, usually it is better to spend time in giving meaningful names to each bone, in order to avoid mistakes in the following skinning process.

It's also very important to build the hierarchy of the bones so that a bone at a higher level can lead all of the children bones, as it would be in a real skeleton (that is, for example, the **hand** bone leads all the **fingers** bones, the **forearm** bone leads the **hand** bone, and so on).

Parenting a bone and then obtaining the others by extruding and/or duplicating simplifies the work because an extruded bone is automatically parented to the bone it has been extruded from, and a duplicated bone obviously inherits the parenting of the original one; in the case of the **tongue.001** bone, extruding the others has given us a chain with bones automatically parented and named as **tongue.002**, **tongue.003**, **tongue.004**, and **tongue.005**.

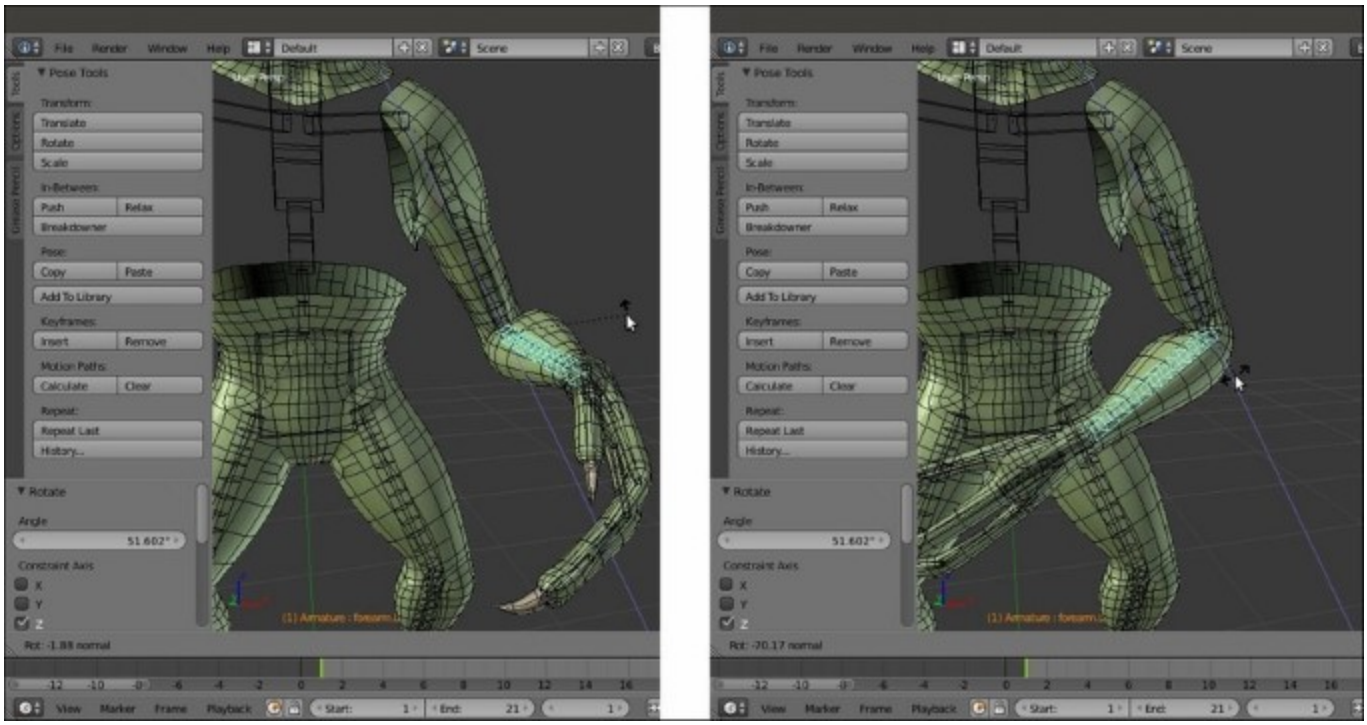
B-bones are both a visualization mode for the bones and a way of working; **B-bones**, in fact, can work inside a chain as splines, which means that the bones are curved according to the number of **Segments** and the values of the **Ease In** and **Ease Out** items. For the bones of the **arms** and **legs**, we have set the **Ease In** and **Ease Out** values to **0.000** (default is **1.000**; maximum is **2.000**), in order to have the B-bones rotating only on their **y** axis but remaining straight along their length, and hence, mimic the twisting by not only the rotation (*pronation* and *supination* of the lower arm) of both the *Ulna-Radius* and *Tibia-Fibula* articulation complexes, but also the (limited) rotation of *Femur* and *Humerus*.

In some way, **B-bones** can work as a kind of simulation for a very basic muscle system; in the following screenshot, you can see their effect on the skinned mesh for the **forearm** by rotating the **hand.L** bone on the local *y* axis (to enhance the visibility of the mesh surface's modifications, the *wireframe over solid drawing* item has been enabled in the **Display** subpanel under the **Object** window):



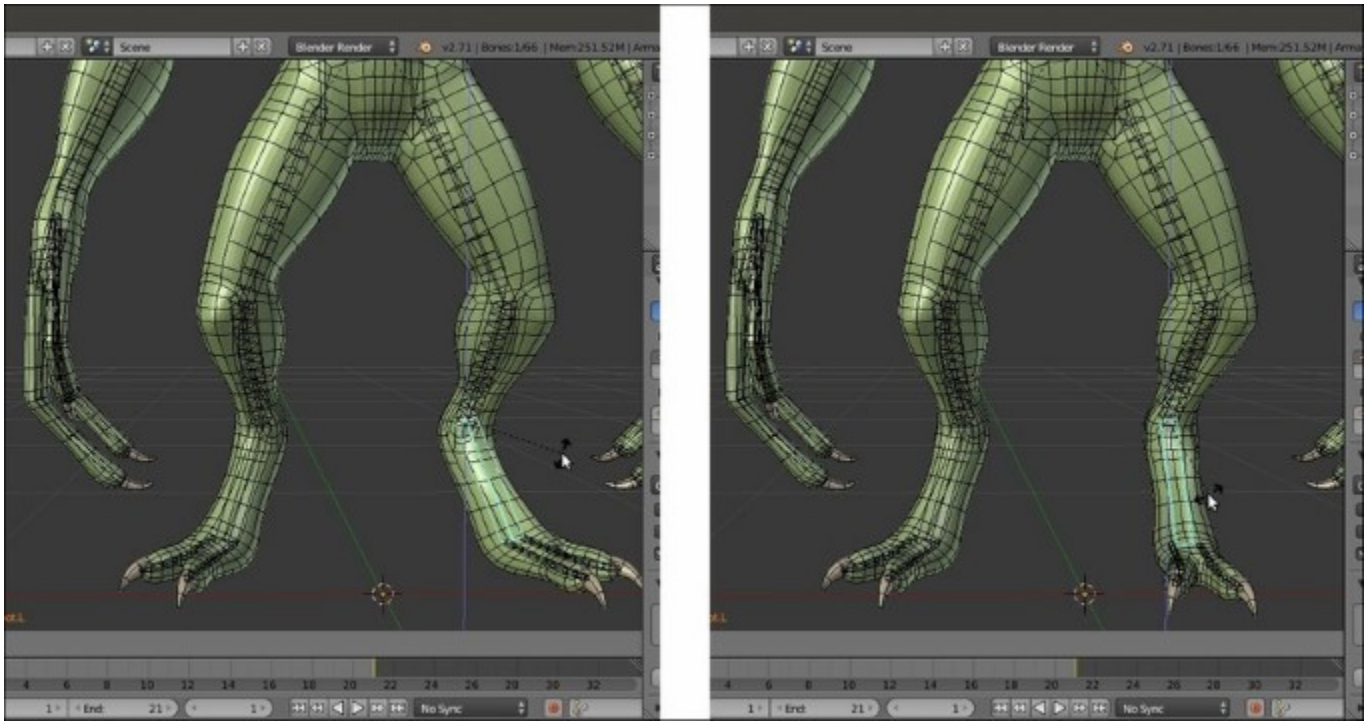
The effect of the rotation of the hand bone on the forearm B-bone and skinned mesh

Here is the effect of the rotation of the **forearm.L** bone on the **Gidiosaurus** high **arm**:



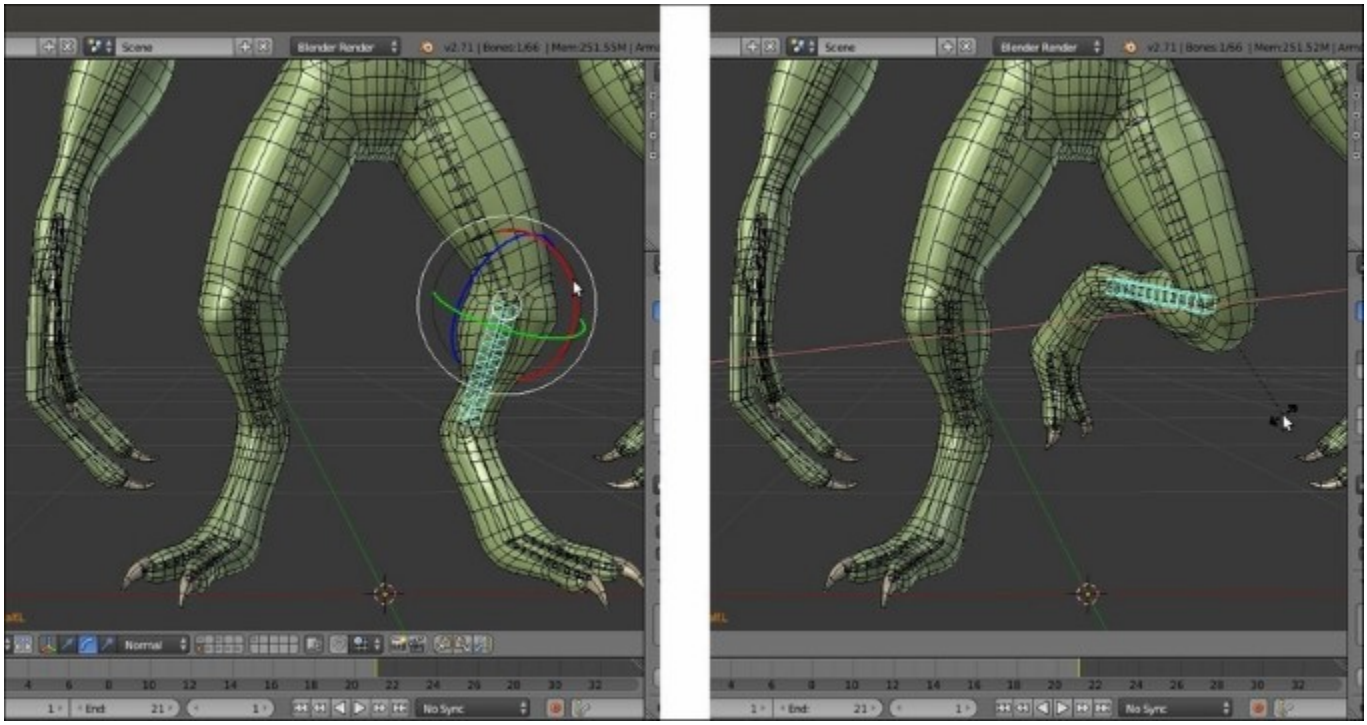
The effect of the rotation of the forearm bone on the upper arm b-bone and skinned mesh

The effect acts on the **shin** as well, by rotating the **foot.L** bone on the global *z* axis:



The same effect obtained on the calf b-bone by rotating the foot on its local y axis

Also, the same effect acts on the **thigh** by rotating the **calf.L** bone:



The same effect obtained on the leg b-bone by rotating the calf

Note that the **Gidiosaurus** is a **digitigrade biped humanoid**: the bones that, from our plantigrade point of view, look like the **foot** are actually the **toes**, while the almost vertical structure that we would call an **ankle** is the real **foot** (this is a very common condition among the majority of the terrestrial animals, both still alive and extinct).

Perfecting the Armature to also function as a rig for the Armor

So, in the previous recipe, we have built the body deforming **Armature** for the **Gidiosaurus** character.

However, the **Gidiosaurus** is an (almost) evolved and a civilized creature, and being also a warrior, it wears a metallic **Armor**; this armor will need to be later parented to the rig as well in order to be animated.

Some of the bones that we have already created will be perfect to skin the **Armor** object too, by assigning the right vertex group to the right mesh part (for example, the **head** vertex group for the **Helm** or the **chest** vertex group for the **Breastplate**). However, because the **Armor** is made also by different parts that cannot be simply driven by the already existing bones (for example, the **belts**, **Vambraces**, and especially **Groinguard**), some modification and/or addition to the rig must be done anyway.

Getting ready

Start from the previously saved `Gidiosaurus_rig_from_scratch_01.blend` file:

1. Enable the **13th** scene layer to show the **Armor** object.
2. Select it and go to the **Object Modifiers** window under the main **Properties** panel. Expand the **Subdivision Surface** modifier tab and click on the *Display modifier in viewport* button, the one with the eye icon, to disable it.
3. Go to the **Outliner** and click on the arrow icon to the side of the **Armor** item to make it unselectable.
4. Click on the arrow icon to the side of the **Gidiosaurus_lowres** item to make it unselectable as well.
5. Save the file as `Gidiosaurus_rig_from_scratch_02.blend`.

How to do it…

Let's start by adding bones dedicated to the **Armor**:

1. Go into **Edit Mode** and select the **forearm.L** bone; use *Shift* + *D* to duplicate it and rename it **vanbrace.L**. Press *M* and in the **Change Bone Layers** pop-up, click on the **2nd** button to move the duplicated bone to that bone layer.
2. Do the same for the **forearm.R** bone (**vanbrace.R**) and for the **calf.L** (**greave.L**) and **calf.R** bones (**greave.R**).
3. Now, go to the **Object Data** window and click on the **2nd** button under the **Layers** item in the **Skeleton** subpanel, in order to show only the four duplicated bones in the 3D viewport; press *Tab* to get out of **Edit Mode**.
4. Select the **vanbrace.L** bone and go to the **Bone** window under the **Deform** subpanel; under the **Curved Bones** item, set back the **Segments** and **Ease In** and **Ease Out** values to default, that is, **1**, **1.000** and **1.000**.
5. Go back into **Edit Mode** and click on the **Connected** item under the **Relations** subpanel.

6. Get out of **Edit Mode** and go to the **Bone Constraints** window under the main **Properties** panel (not to be confused with the **Object Constraints** window); click on the **Add Bone Constraint** button and select a **Copy Rotation** constraint from the pop-up menu (the bone turns light green, in order to show that it has a constraint assigned now).
7. In the **Target** field, select **Armature**; in the **Bone** field, select the **forearm.L** item; in the **Space** fields, select **Pose Space** for both.

   Alternatively, for steps 6 and 7, select the **forearm.L** bone and then use *Shift* to select the **vanbrace.L** bone. Hence, press *Shift* + *Ctrl* + *C* to call the **Add Constraint (with Targets)** pop-up menu and select the **Copy Rotation** item. This will automatically add the **Copy Rotation** constraint to the **vanbrace.L** bone, with the first select bone (**forearm.L**) as a target; the other setting must be enabled and/or tweaked in the constraint subpanel instead.
8. Click again on the **Add Bone Constraint** button and this time, select an **Inverse Kinematics** constraint (the bone turns yellow, in order to show that an **IK solver** has been assigned). In the **Target** field, select the **Armature** item, in the **Bone** field, select the **hand.L** bone, and set the **Chain Length** to **1**; deselect **Stretch** and select **Rotation**, lowering the weight to the minimum (that is **0.010**):

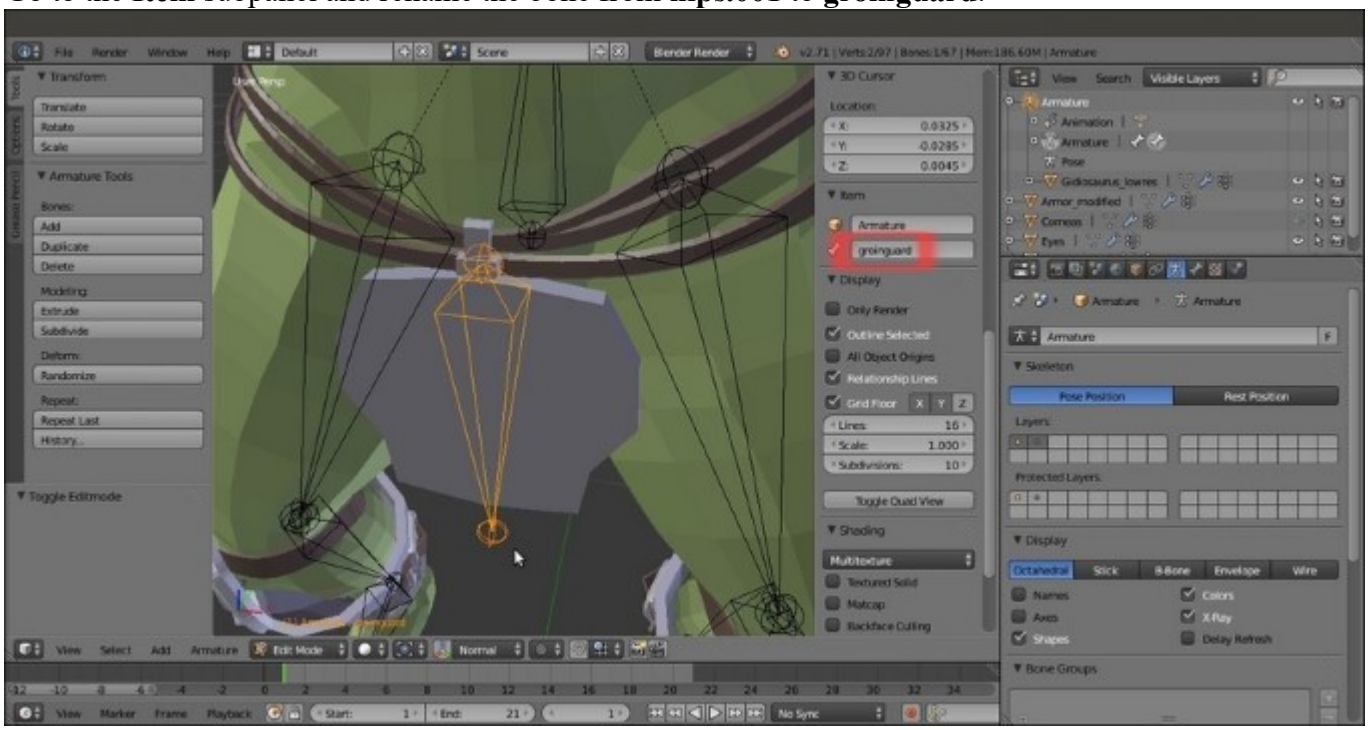
The constraints assigned to the forearm.L b-bone

9. Repeat the steps from 4 to 8 for the other three duplicated bones (obviously, setting the appropriate bones as targets for each pair of constraints; the target bone for the **IK** constraint assigned to the **greave** bones is the respective foot bone).

   The rig can now drive the **vambraces** and **greaves**; let's see the **knee guards** and **Groinguard**.
10. First, switch the **Armature** visualization back to **Octahedral**, then go into **Edit Mode**, select the **hips** bone and use *Shift* + *D* to duplicate it; in the **Side** view, rotate the duplicate **170**

degrees, then move it on the **Groinguard** part of the armor, in order to have the **Head** of the bone placed to the joint of the **plate** with the **ties**; select the **Tail** of the **groinguard** bone and scale it smaller to fit the part.

11. To position the bone more precisely, go to the **Transform** subpanel under the **Properties** 3D view sidepanel and set the following values for the **Head** (of the bone): **X** = **0.001**, **Y** = **0.020**, and **Z** = **1.147**; for the **Tail** set the following values: **X** = **0.001**, **Y** = **0.022**, and **Z** = **0.873**.
12. Go to the **Item** subpanel and rename the bone from **hips.001** to **groinguard**:

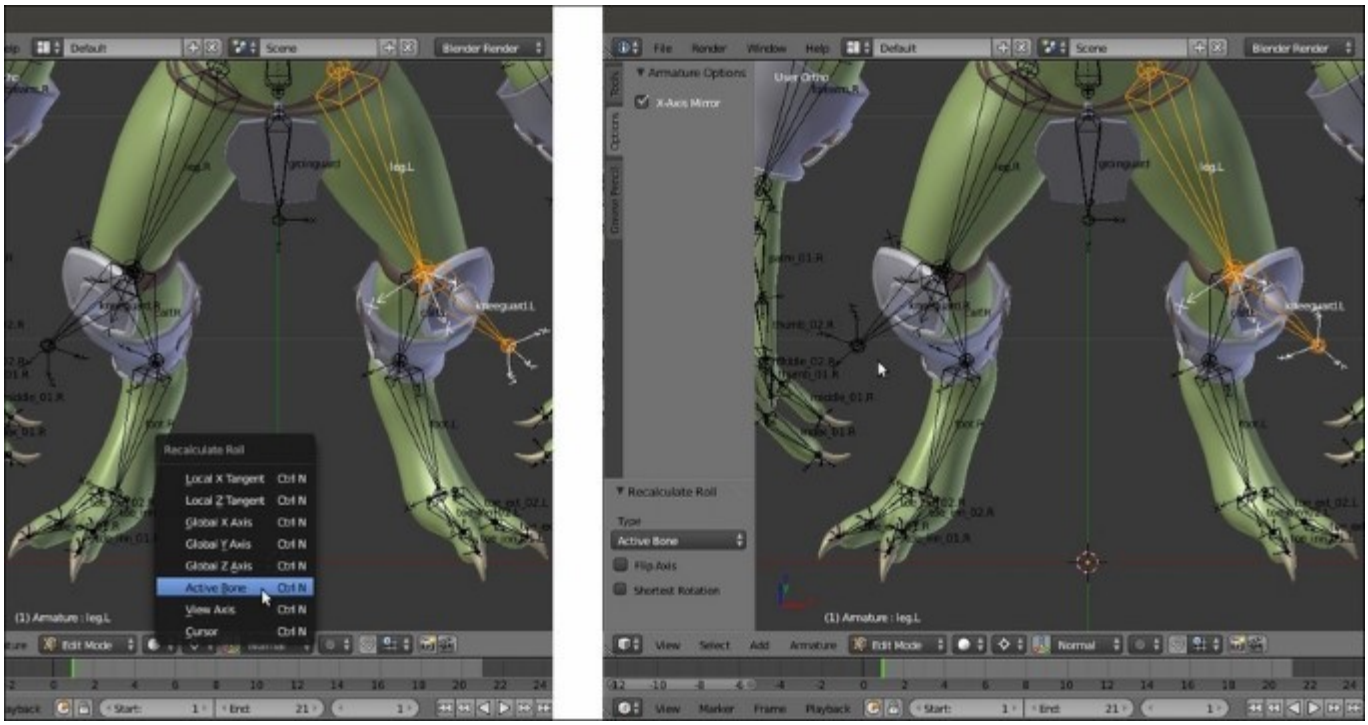


The groinguard bone

13. Go to the **Bone** window in the main **Properties** panel, and under the **Relations** subpanel, click on the **Parent** empty slot to select the **hips** item.
14. Now, select the joint of the **leg.L** bone with the **calf.L** bone and press *Shift* + *S* | **Cursor to Selected**; press *Shift* + *A* to add a new bone and rescale it smaller.
15. Select the whole new bone and use *Shift* to select the **calf.L** bone. Then, go in the 3D view toolbar and click on the **Armature** item; go to **Transform** | **Align Bones** (or else, press the *Ctrl* + *Alt* + *A* keys) to align the new bone as the **calf.L** one.
16. Enable the widget (*Ctrl* + spacebar), set the **Transform Orientation** to **Normal**, and the rotation pivot on the **3D Cursor**. Then, rotate the new bone **110** degrees on the normal *x* axis (the red wheel of the widget, or else *R* | *X* | *X* | **110** | *Enter*):

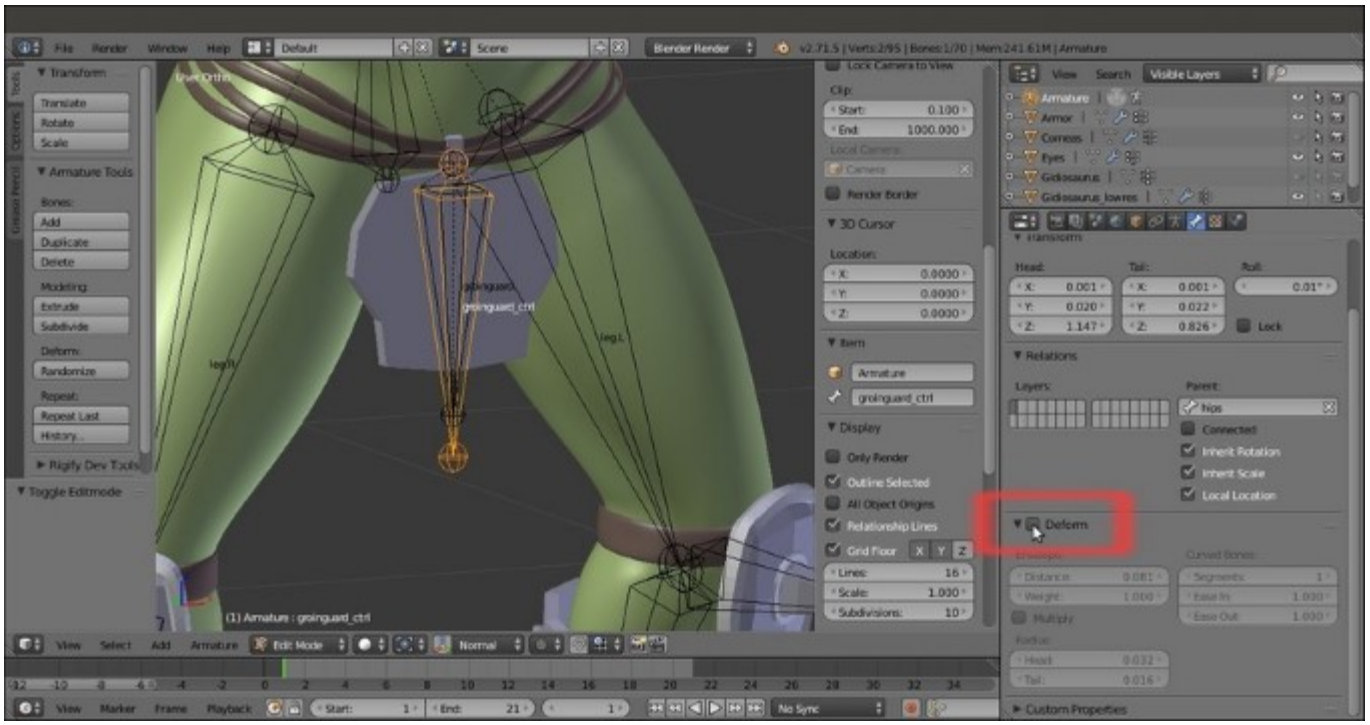
Aligning the new bone

17. Go into **Object Mode** and press *Shift* + *S* | **Cursor to Selected** to place the **3D Cursor** at the median pivot point of the **Armature**; go back into **Edit Mode**, press *Shift* + *D* to duplicate the new bone, then *Ctrl* + *M* | *X* to mirror it on the other side.
18. Rename the new bones as **kneeguard.L** and **kneeguard.R**; enable the axis visibility and recalculate the roll by the *Ctrl* + *N* | **Active Bone** tool:



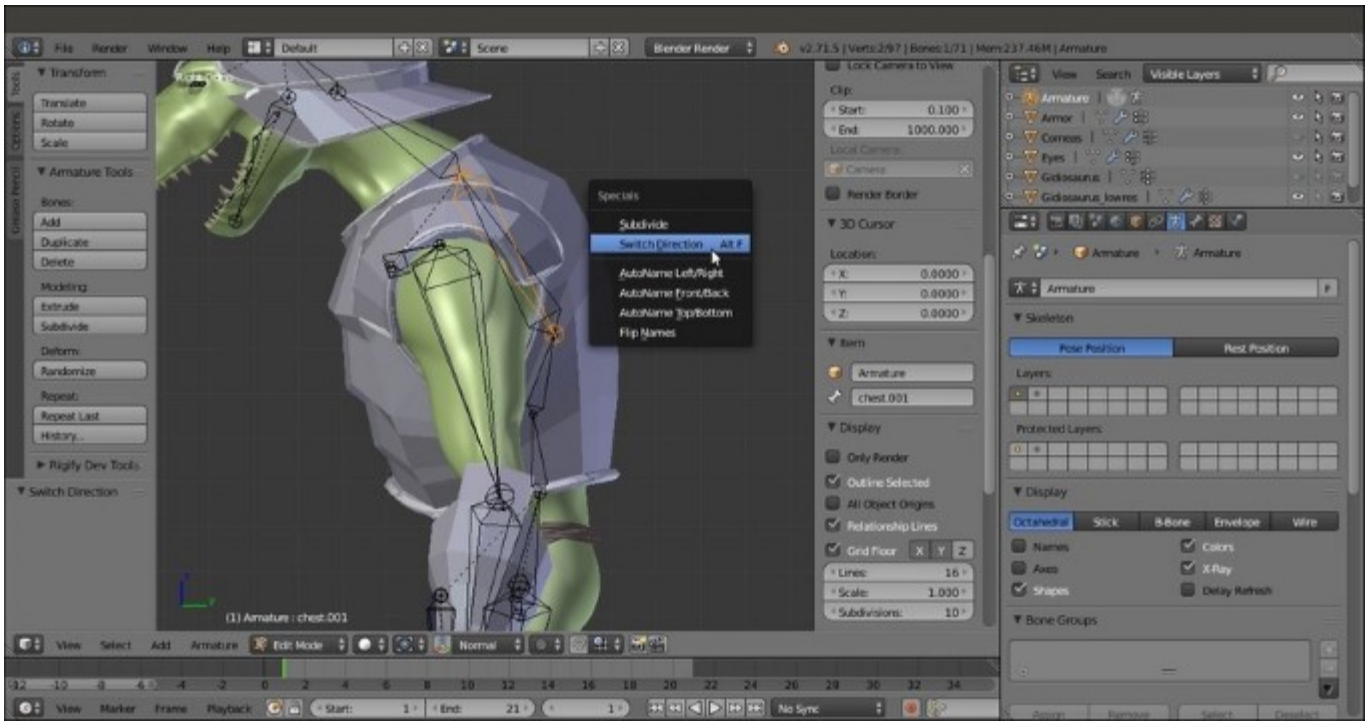
Recalculating the roll angle of the kneeguard bones

19. Parent the **kneeguard.L** bone to the **leg.L** bone and the **kneeguard.R** bone to the **leg.R** one (not connected).
20. Select the **groinguard** bone and use *Shift + D* to duplicate it, and then scale the duplicated bone a little bit bigger and rename it as **groinguard_ctrl**; uncheck the box of the **Deform** subpanel under the **Bone** window:



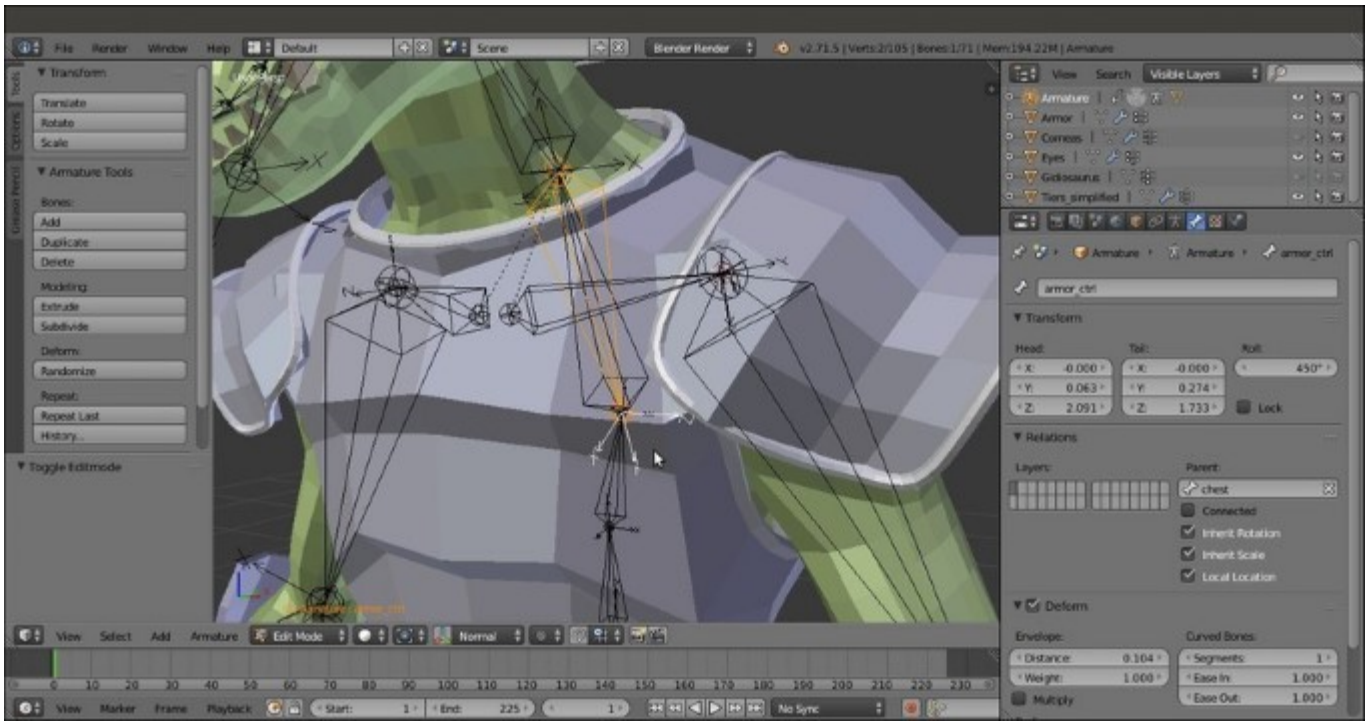
Creating a control bone for the groinguard bone

21. Select the **groinguard** bone, go to the **Relations** subpanel, and click in the **Parent** field to select the **groinguard_ctrl** bone.
22. Get out of **Edit Mode** and in **Pose Mode**, select the **groinguard_ctrl** bone.
23. Go to the **Bone Constraints** window under the main **Properties** panel; click on the **Add Bone Constraint** button and select a **Locked Track** constraint from the pop-up menu.
24. In the **Target** field, select the **Armature** item; in the **Bone** field, select the **kneeguard.L** item. Set the **Head/Tail** value to **0.500**: **To** (*Axis that points to the target object*) = **-X** and **Lock** (*Axis that points upward*) = **Y**. In the **Constraint Name** field, rename it as **Locked Track.L**.
25. Add a new **Locked Track** constraint and repeat everything as in the previous one, except in the **Bone** field, select the **kneeguard.R** item; rename it as **Locked Track.R**.
26. Add a **Damped Track** constraint: **Target** = **Armature**, **Bone** = **kneeguard.L**, **Head/Tail** = **0.728**, **To** = **Y**, and **Influence** = **0.263**. Rename it as **Damped Track.L**.
27. Add a new **Damped Track** constraint and repeat everything as in the previous one, except again in the **Bone** field, select the **kneeguard.R** item; rename it as **Damped Track.R**.
28. Just to be sure, save the file!
29. Go back into **Edit Mode** and in the **Side** view, select the **chest** bone and use *Shift* + *D* to duplicate it. Press *W* to call the **Specials** pop-up menu and select the **Switch_Direction** item, or else press *Alt* + *F* directly:



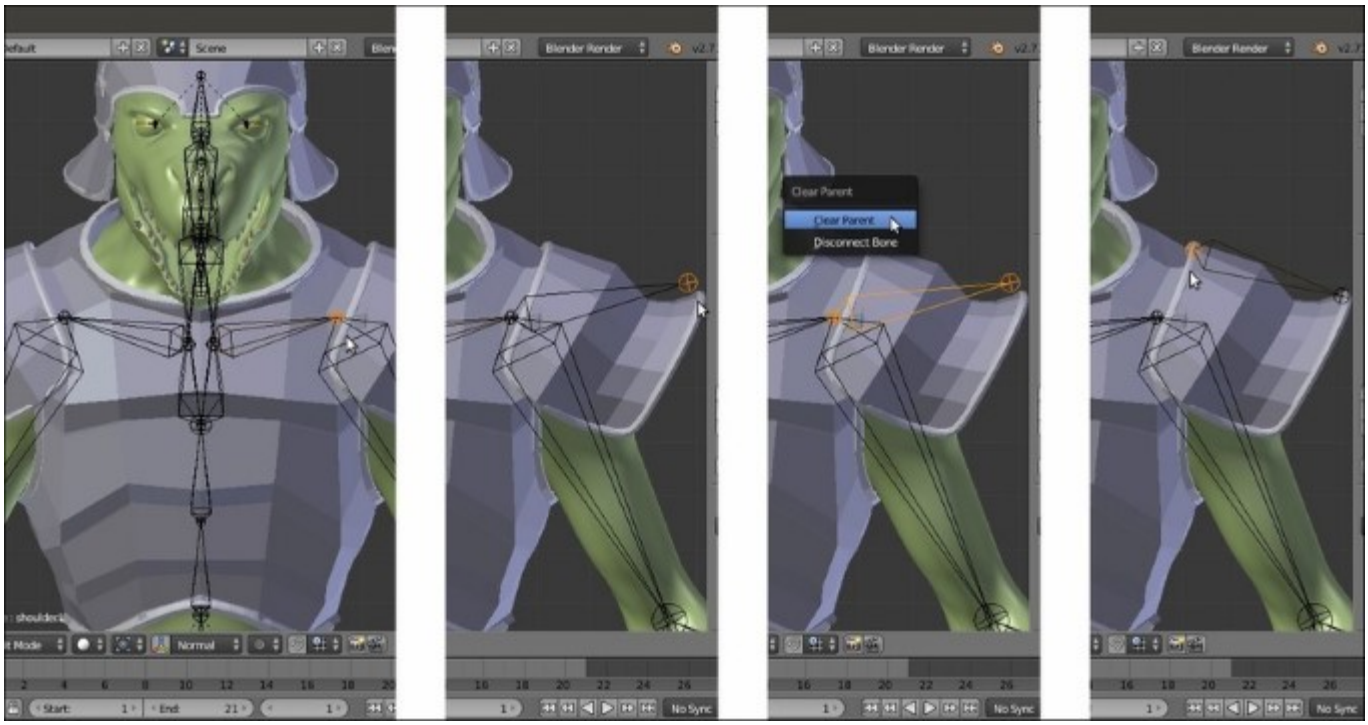
The Specials pop-up menu for the bones

30. Go to the **Bone** window and click on the **Parent** slot under the **Relations** subpanel to select the **chest** item (not connected); then, go to the Deform subpanel and set the **Segments** under **Curved Bones** to **1**. Rename the new bone as **armor_ctrl**.
31. Press *Ctrl* + *R* to roll the **armor_ctrl** bone, in order to be sure that its local *x* axis is pointing towards the front of the model; this is important to make the **Transformation** constraints, which we'll add later, work properly:



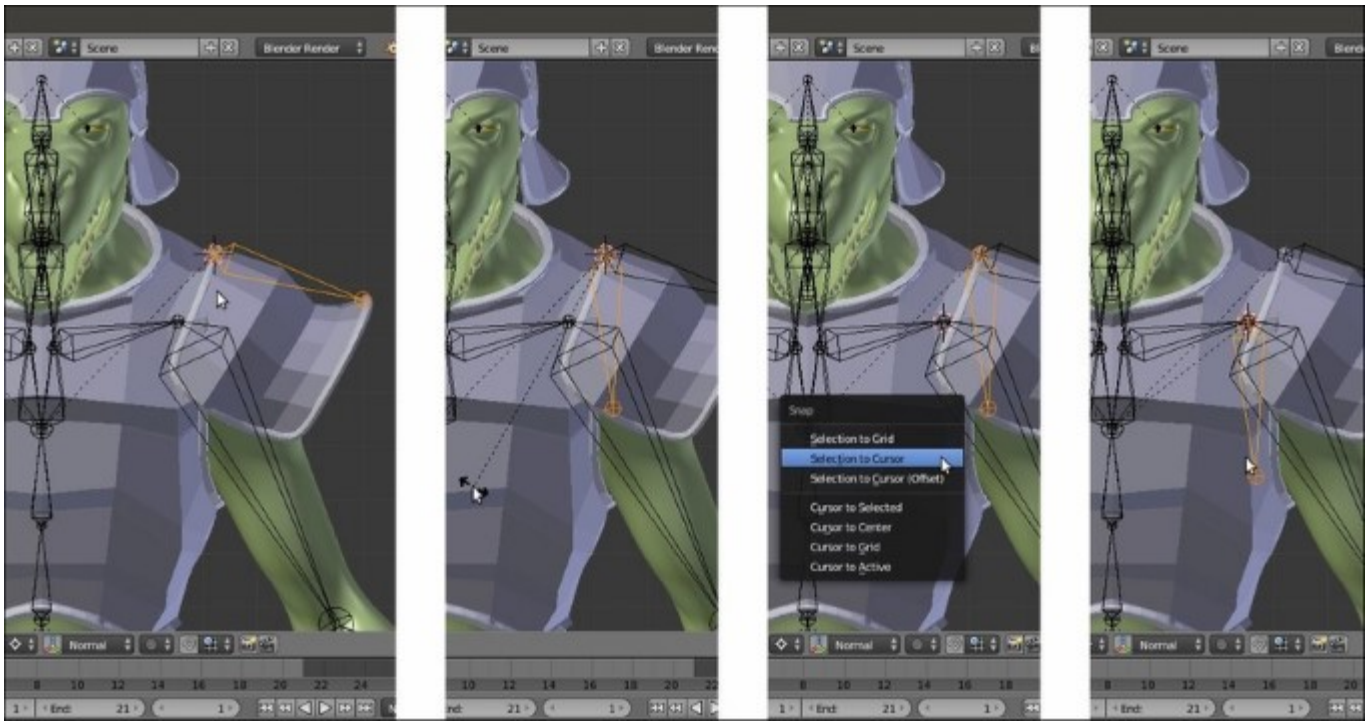
The armor control bone

32. Go in the **Front** view. Note that the **X-Axis Mirror** item in the **Armature Options** panel under the **Tool Shelf** is still enabled; select the **Tail** of the **shoulder.L** bone and extrude a new bone going towards the external edge of the armor **spaulder**. Then, select the extruded bone, press *Alt* + *P* | **Clear Parent**, and move its **Head** to be positioned above the *joint* of the **spaulder** with the **chest** plate.



Creating the bone for the spaulder

33. Rename the extruded bone and the corresponding mirrored one as **spaulder.L** and **spaulder.R**; parent them to the **armor_ctrl** bone (enable the **Keep Offset** item).
34. Use *Shift* to select the **spaulder.L** and **arm.L** bones and press *Ctrl* + *N* | **Active Bone**; do the same with the **spaulder.R** and **arm.R** bones.
35. Now, put the **3D Cursor** at the **spaulder.L** bone's **Head** location, and then set the **Pivot Point** to the **3D Cursor** in the 3D window toolbar. Use *Shift* + *D* to duplicate the **spaulder.L** bone and rotate the duplicate **70** degrees (in the **Front** view, *R* | **70** | *Enter*).
36. Place the **3D Cursor** at the **shoulder.L** bone's **Tail** location, select the duplicated bone, and press *Shift* + *S* | **Selected to Cursor**. Rename the duplicated bone and the mirrored one as **rotarmor.L** and **rotarmor.R**. Go to the **Relations** subpanel and set the **rotarmor.L** bone as the child of the **arm.L** bone and the **rotarmor.R** bone as the child of the **arm.R** bone. Disable the **Deform** item for both of them:



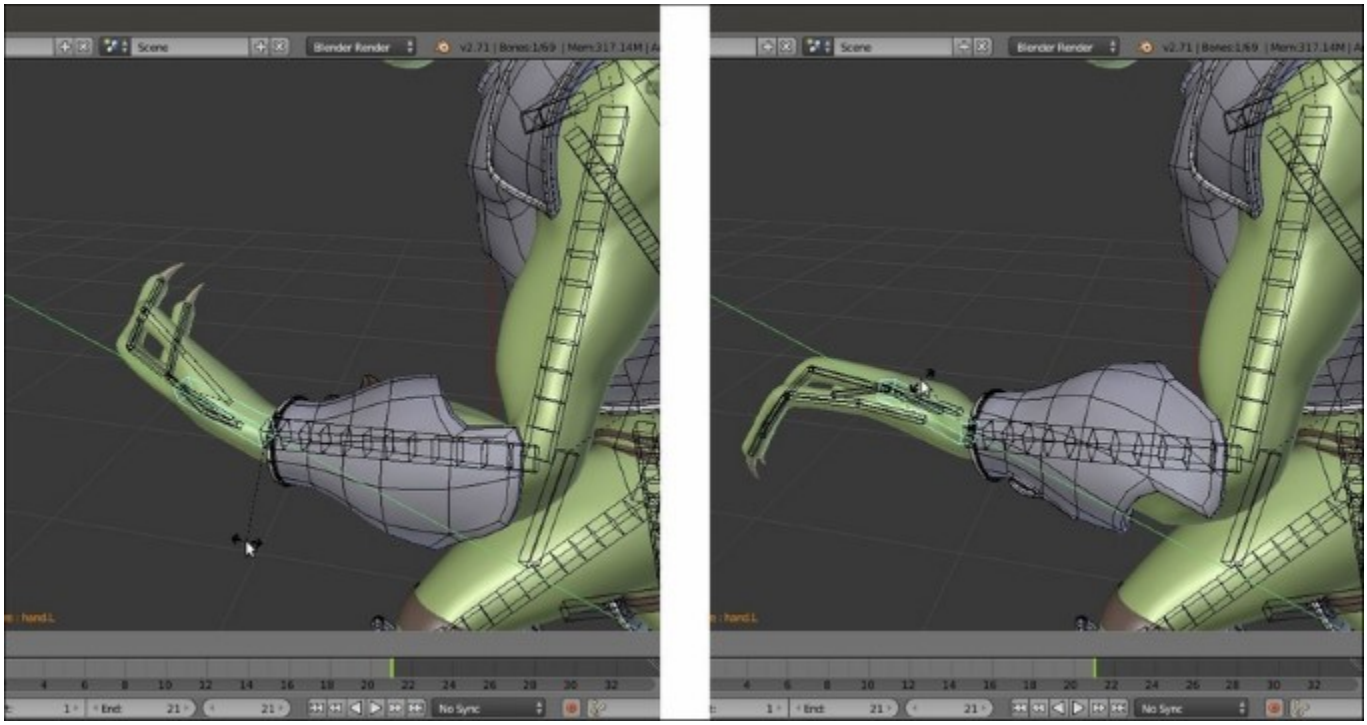
Using the 3D Cursor and the Snap menu to exactly place the bones

37. Go into **Pose Mode**. Select the **spaulder.L** bone and in the **Bone Constraints** window, assign a **Copy Rotation** constraint: **Target** = **Armature**, **Bone** = **arm.L**, **Space** = **Pose Space** to **Pose Space**, and **Influence** = **0.200**.
38. Select the **spaulder.R** bone and repeat with the **Bone** = **arm.R** target.
39. Now, select the **armor_ctrl** bone and assign a **Transformation** constraint. Set **Target** = **Armature**, **Bone** = **rotarmor.L**, **Source** = **Rot**, and **Z Max** = 20°; **Source To Destination Mapping** = switch **X** with **Z**; **Destination** = **Rot**, **X Max** = 4°, and **Space** = **Pose Space** to **Pose Space**. Rename the constraint as **Transformation_rot.L** and collapse the panel.
40. Assign a second **Transformation** constraint; set everything as in the previous one, except for the target **Bone** = **rotarmor.R**, **Source** = **Rot**, **Z Min** = **-20°**, and **Destination X Min** = **-4°**. Rename the constraint as **Transformation_rot.R** and collapse it.
41. Assign a third **Transformation** constraint; set everything as in the first one, except do not switch **X** with **Z**, set **Destination** = **Loc** and **Z Max** = **0.050**. Rename the constraint as **Transformation_move.L** and collapse it.
42. Assign a fourth **Transformation** constraint; set everything as in the second one, except do not switch **X** with **Z**; set **Destination** = **Loc** and **Z Min** = **0.050**. Rename the constraint as **Transformation_move.R** and collapse it.
43. Save the file.

How it works…

We couldn't directly use the **forearm** and **calf** bones to rig the **vanbraces** and **greaves** parts because being subdivided **B-bones**, they would *curve* these **armor** parts along the length as they actually do by

deforming organic parts as the **forearms** and **shins**, and this would look awkward, as you can see in the following screenshot:

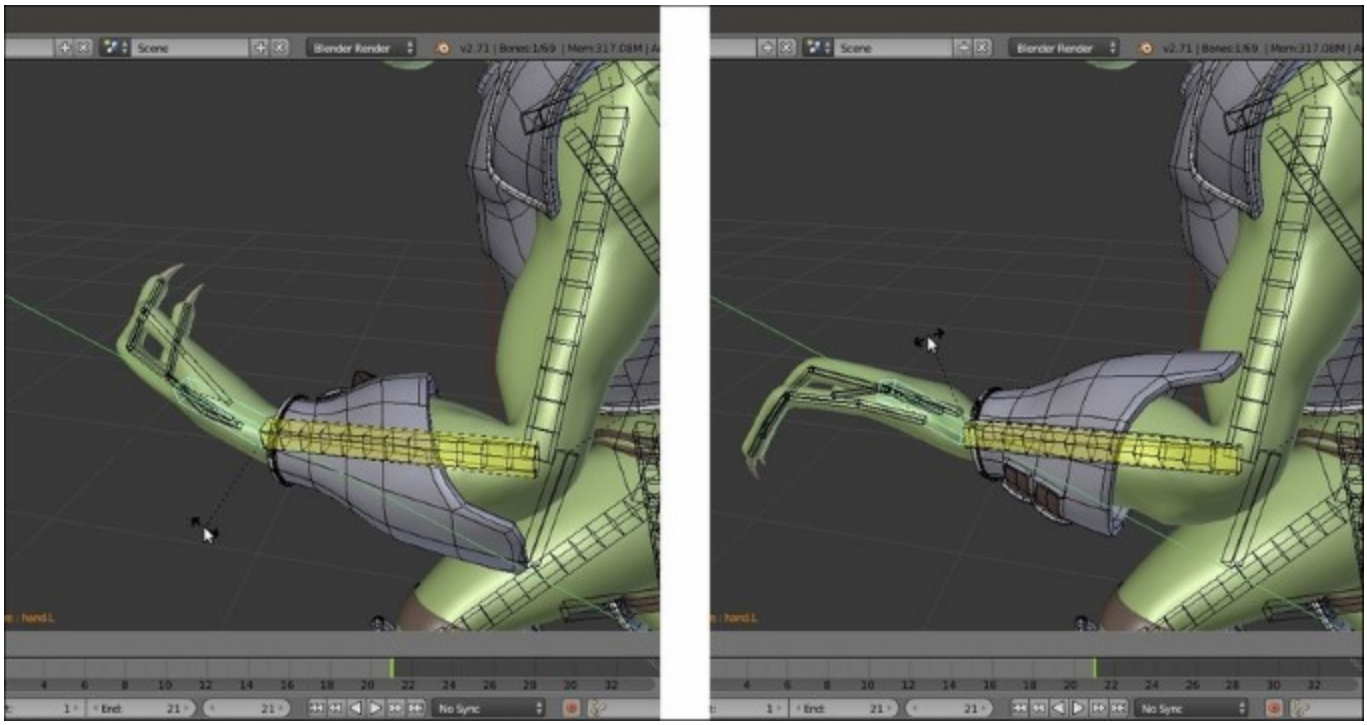


B-bones erroneously deforming stiff objects

Instead, we just duplicated the bones, restored **Segments** and **Ease In** and **Ease Out** to default values, and assigned **2** bone constraints (note that, as already mentioned, the bones have a **Bone Constraints** panel of their own, which is different from the **Object Constraints** one).

The **Copy Rotation** constraint, as the name itself explains, copies the rotation in space of the target **B-bone**; the position inside the chain is granted because the duplicated bones, although not connected, are children of the same bones as the original ones.

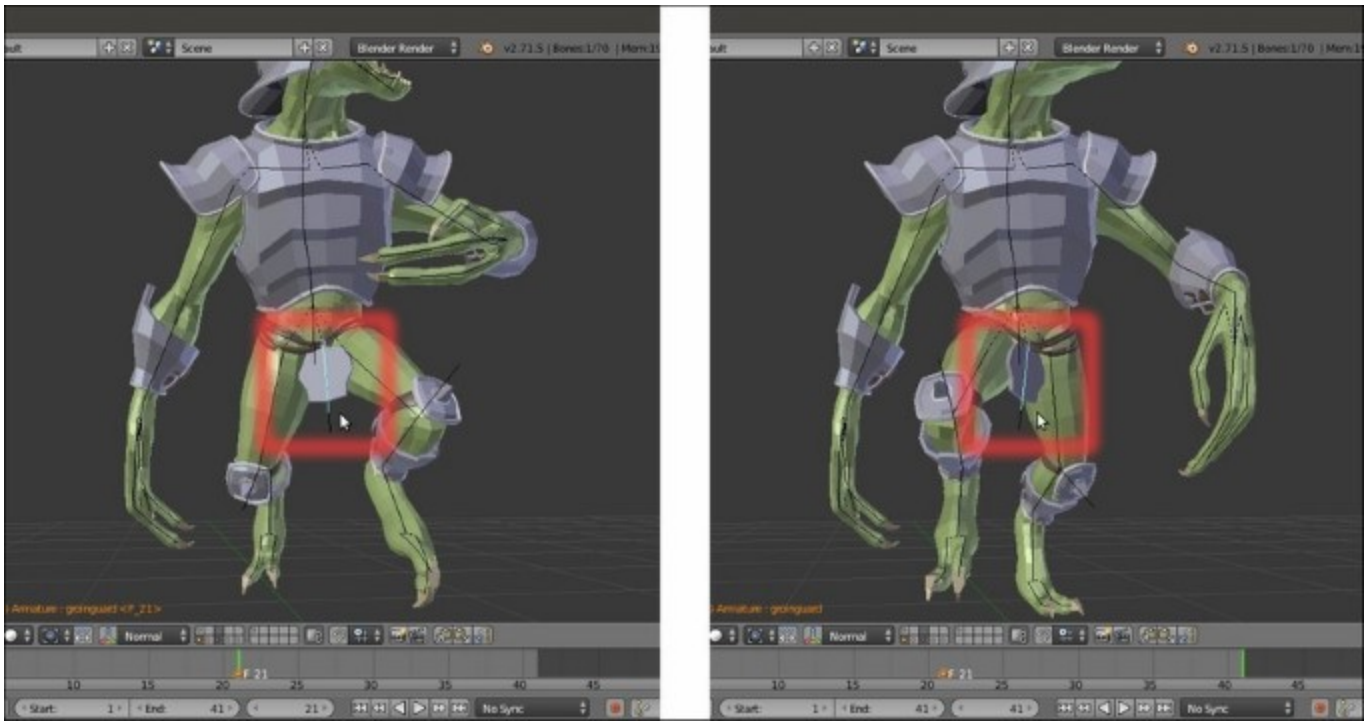
The **Inverse Kinematics** constraint—in this case, is used simply to track the **local y** rotation of the **hand** bone in order to rotate correctly on its *y* axis— is necessary because the **Copy Rotation** constraint doesn't seem to read the **local y** rotation of a subdivided **B-bone** (besides the technical details, it makes sense because that's actually not a rotation in space):



The correct rotation of the stiff armor parts

The constraints assigned to the **groinguard_ctrl** bone are a cheap, but quite an effective, way to fake a rigid body simulation for the **plate** that—in actions, for example, a walk cycle—should interact by colliding with the **Gidiosaurus thighs**. The **Locked Track** constraints, targeted to the **leg** bones, automatically rotate the **plate** according to the **thighs** movements, and the **Dumped Track** constraints, targeted to the **leg** bones as well but with a low influence, add a swinging movement.

The **groinguard** bone, actually the one affecting the **armor plate**, is the child of the **groinguard_ctrl** bone, and so it inherits the constraint's movements but can be used to refine, tweak, or modify the final animation of the plate by hands:



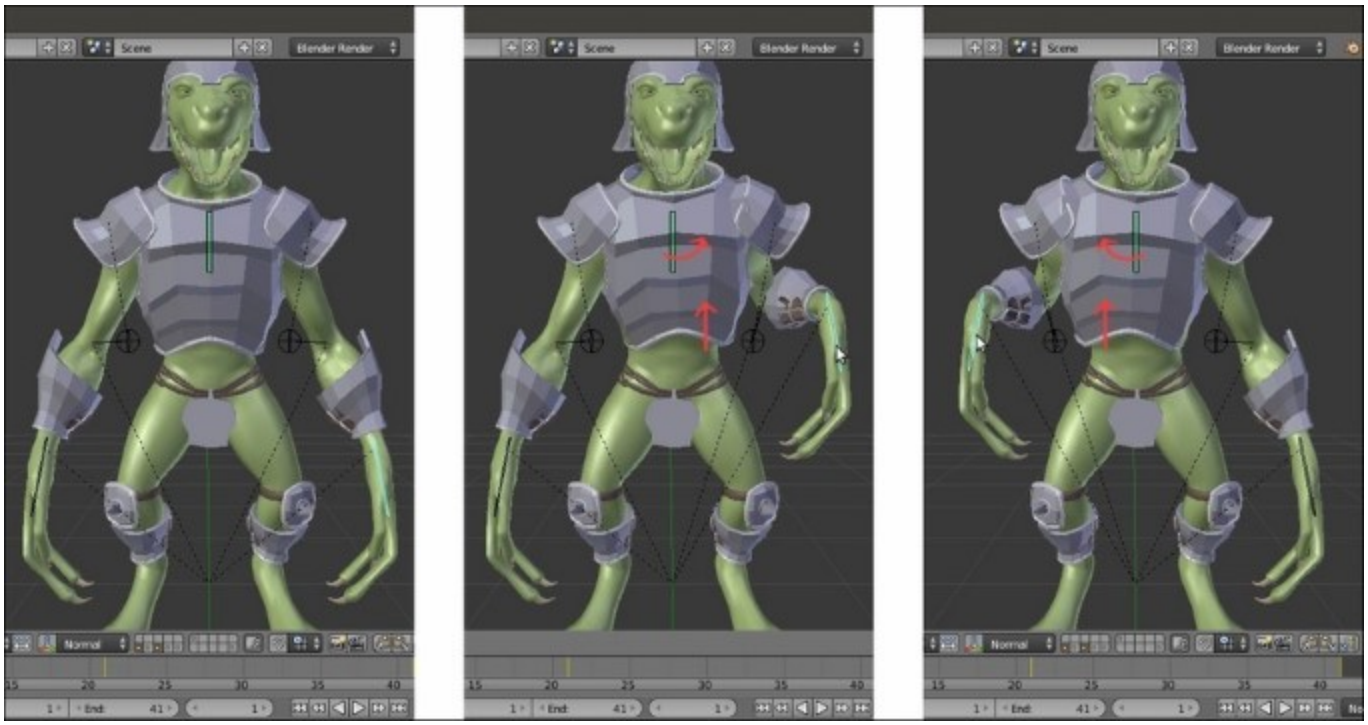
The groinguard bone (and plate) automatically rotating during the walk cycle

The **armor_ctrl** bone is the bone controlling the **armor's Breastplate**; it's the child of the **chest** bone, so it inherits the rotation of the **chest,** but has four **Transformation** constraints.

By using as an input the rotation angle of the **rotarmor.L** and **rotarmor.R** bones (which are children themselves of the **arm.L** and **arm.R** bones), the constraints give to the **Armor chest plate** a slight rotation on the vertical axis and a lateral swinging, driven by the oscillations of the **Gidiosaurus arms**, and simulating of the character's **shoulders** colliding with the **armor plate** during the walk.

Also, the **spaulders** are, in turn, partially rotated by bones with the **Copy Rotation** constraints targeted to the **arms,** but with quite a low influence.

Although better appreciated in motion, the following screenshot will show you the effects as the **arms** rotate backward:



The rotation and swinging of the armor chest plate according to the arms' movements

Building the character's Armature through the Human Meta-Rig

In the previous long and quite complex recipe, we hand-built the deforming elements of an average basic rig for the **Gidiosaurus** character; actually, in Blender, there are other tools to build rigs, particularly meant to facilitate the task, and we'll see them in this recipe and in the following ones.

Now, we are going to take a look at the **Human Meta-Rig** tool.

Getting ready

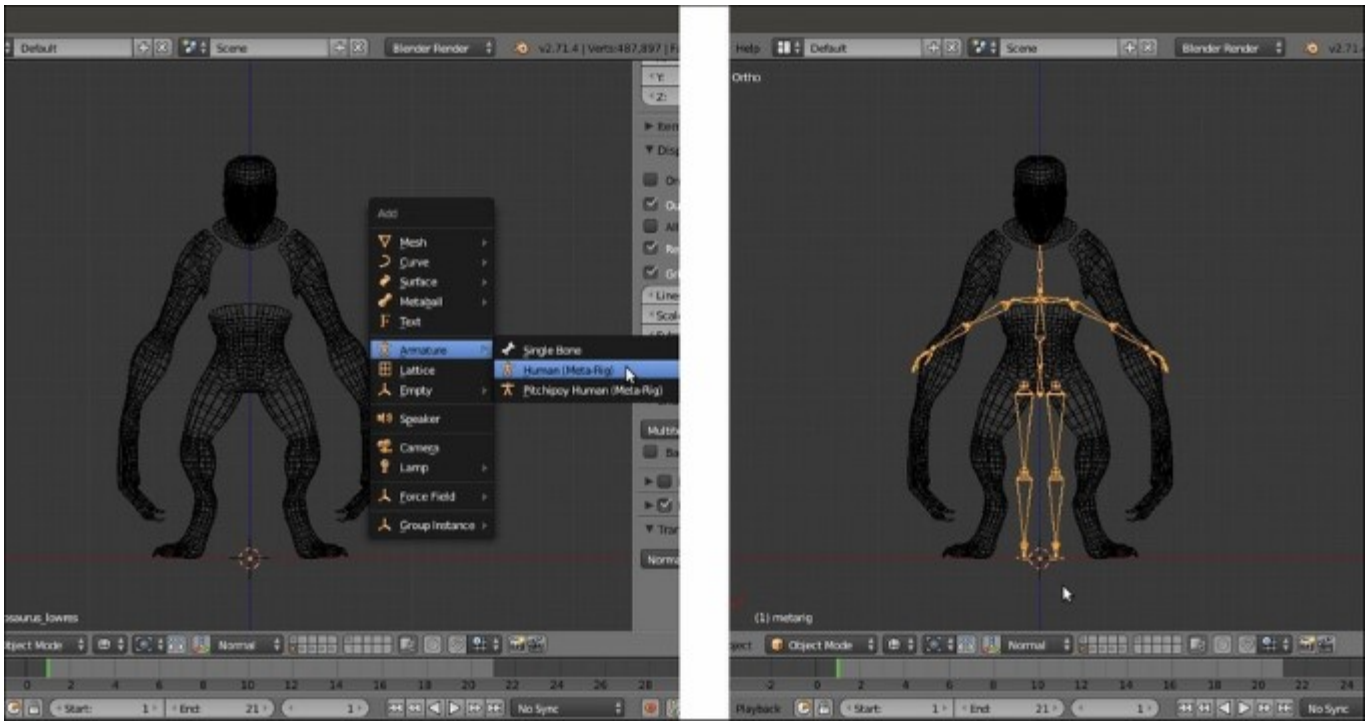
To be able to use the **Human Meta-Rig** tool, we must first enable the proper add-on:

1. Start Blender and press *Ctrl* + *Alt* + *U* to call the **User Preferences** panel. Go to the **Add-ons** tab and under **Categories** on the left-hand side, click on the **Rigging** item. Go to the right-hand side of the panel and check the box to the side of the **Rigging: Rigify** add-on to enable it.
2. Click on the **Save User Settings** button at the bottom-left of the panel and then close it. Because we are starting a rig from scratch again, load the `Gidiosaurus_unwrap_final.blend` file.
3. Disable the **Textured Solid** and **Backface Culling** items in the 3D view **Properties** panel, join the 3D window with the **UV/Image Editor** window, and click on the **11th** scene layer to have only the **Gidiosaurus** mesh visible.
4. Go to the **Object** window and under the **Display** subpanel, enable the **Wire** item; this will be useful in the process to have an idea of the mesh topology when in **Object Mode** and in **Solid** viewport shading mode. However, for the moment, press *Z* to go in the **Wireframe** viewport shading mode.
5. Press *1* on the numpad to go in the **Front** view and *5* on the numpad again to switch to the **Ortho** view.
6. Save the file as `Gidiosaurus_meta_rigging.blend`.

How to do it…

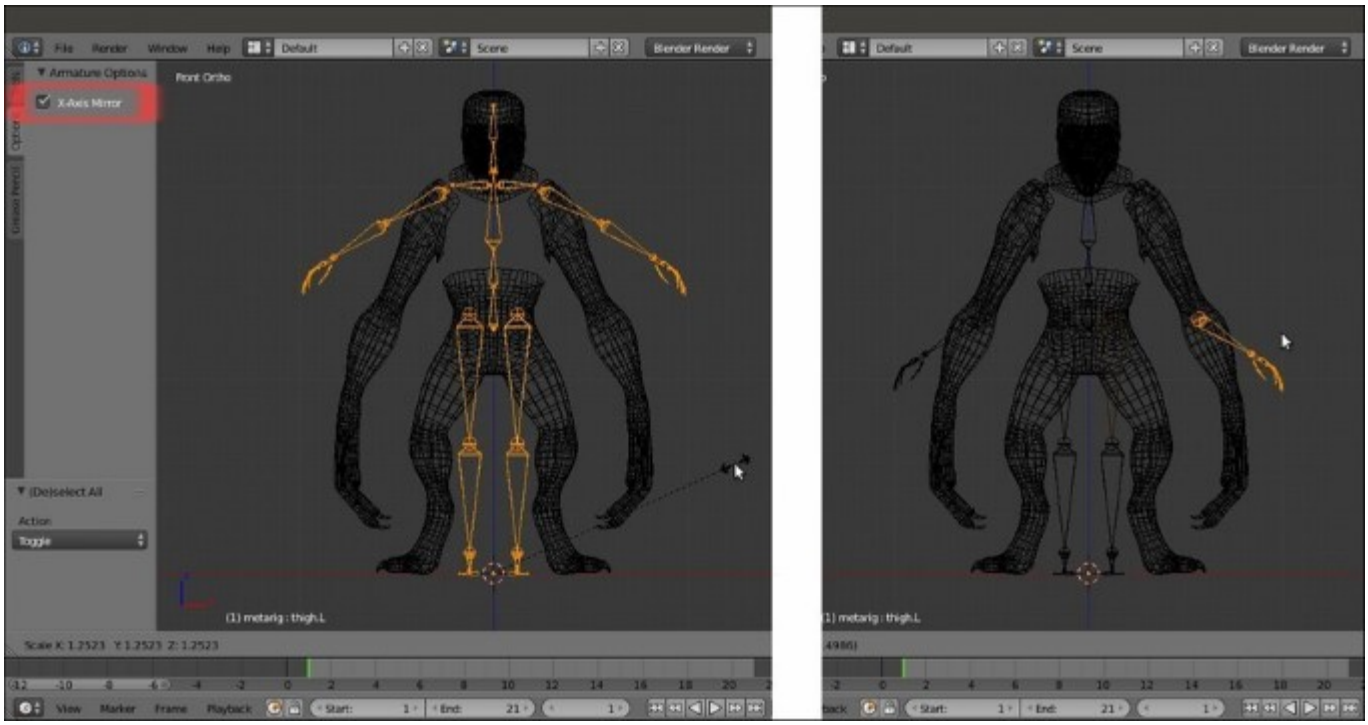
Let's go with the **metarig** itself:

1. Ensure that the **3D Cursor** is at the origin pivot point of the **Gidiosaurus** mesh. Put the mouse cursor in the 3D viewport, press *Shift* + *A*, and in the pop-up menu, select **Armature** | **Human (Meta-Rig)**; a biped **Armature**, automatically named **metarig** in the **Outliner**, appears at the **3D Cursor** location:



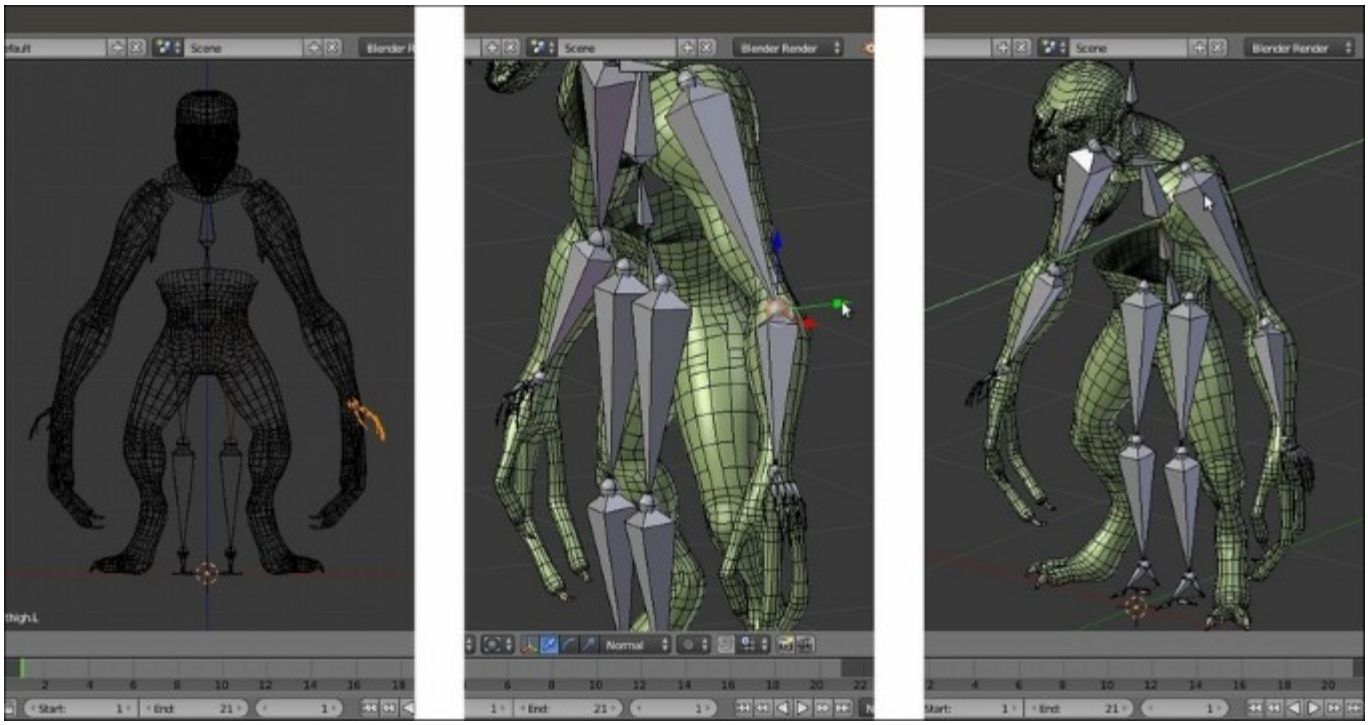
The Human (Meta-Rig) menu and the rig

2. Press the *Tab* key to enter **Edit Mode** and go to the **Options** tab that appeared under the **Tool Shelf** on the left-hand side of the screen; check the box to enable the **X-Axis Mirror** tool under the **Armature Options** item.
3. First, press the period (.) key to set the pivot point around the **3D Cursor** and scale the whole armature bigger while still in **Edit Mode**, and then start to edit locations and proportions of the bones of the **metarig** to fit inside the **Gidiosaurus** shape:



Tweaking the proportions of the bones of the metarig

4. Select the single joints to move them on the right location according to the mesh topology; to do this in a more exact way, just use the snap technique explained in steps 25 and 26 of the *How to do it…* section of the *Building the character's Armature from scratch* recipe. Because of the **X-Axis Mirror** tool we enabled, it's enough to operate only on one side of the **metarig**:



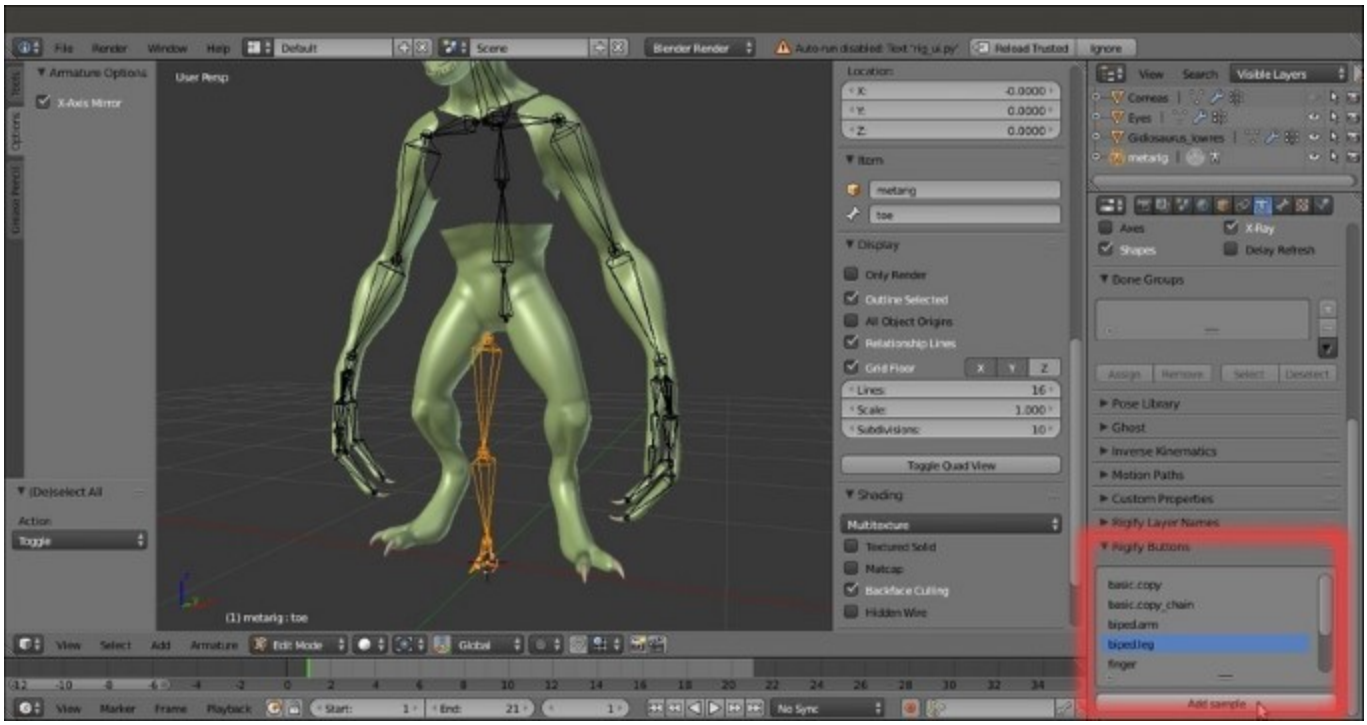
Further tweaking of the bones in Edit Mode

5. Delete the bones that you don't need, for example the extra **fingers** (consider that the **Gidiosaurus** has only three **fingers** in each **hand**), use *Shift + D* to duplicate the bones to be added, for example for the **toes**, and add new bones where missing, for example for the **jaw**, and then parent them. In short, just edit the rig as usual. Again, it should be enough to do all these operations just on one side of the rig:

The completed skeleton rig

6. Save the file.

We can also add premade rigging sets, for example a whole new **leg**, **spine**, or **arm**, by going, with the **metarig** still in **Edit Mode**, to the **Rigify Buttons** subpanel under the **Armature** window in the main **Properties** panel. Select the desired item to be added to the rig and click on the **Add sample** button; the new part gets added to the rig's **pivot point** location and must be moved to the right place and tweaked, rotated, and scaled as needed. Also, the new bones must be named with the correct **.R** or **.L** suffix and the top chain bone must be parented to the bottom **metarig** bone; for example, in the case of a **biped.leg** part addition, the **thigh** bone must be parented (*Ctrl + P* | **Keep Offset**) to the **hips** bone:



Adding premade rig to the skeleton

How it works…

The **Human metarig** is actually only the first part of a more complex and complete auto-rigging system named **Rigify**, and this we'll see in the next recipe. However, even used by itself, it gives us a readymade humanoid skeleton to be simply tweaked to fit the character's shape: a good shortcut to quickly build the **Armature** rig considering that, at least in its basic form, all the bones are already properly connected and named with the **.L** and **.R** suffices.

Building the animation controls and the Inverse Kinematic

Whether we built the **Gidiosaurus** deforming rig part by hands from scratch or by the **Human Meta-rig**, we must now add the necessary constraints and controls to allow the animators to easily manipulate the character.

Note

Note that once the mesh is skinned, the rig, as it is at this point, can actually already work by directly selecting the interested bones and rotating them in **Forward Kinematics**; however, to simplify the animator's work (and complicate our life a little bit more), it's good practice to add the **Inverse Kinematic** constraints and the control bones.

Getting ready

Let's start by opening the `Gidiosaurus_rig_from_scratch_02.blend` file; as usual, enter **Edit Mode** to ensure that the **X-Axis Mirror** item in the **Armature Options** subpanel under the **Tool Shelf** is enabled.

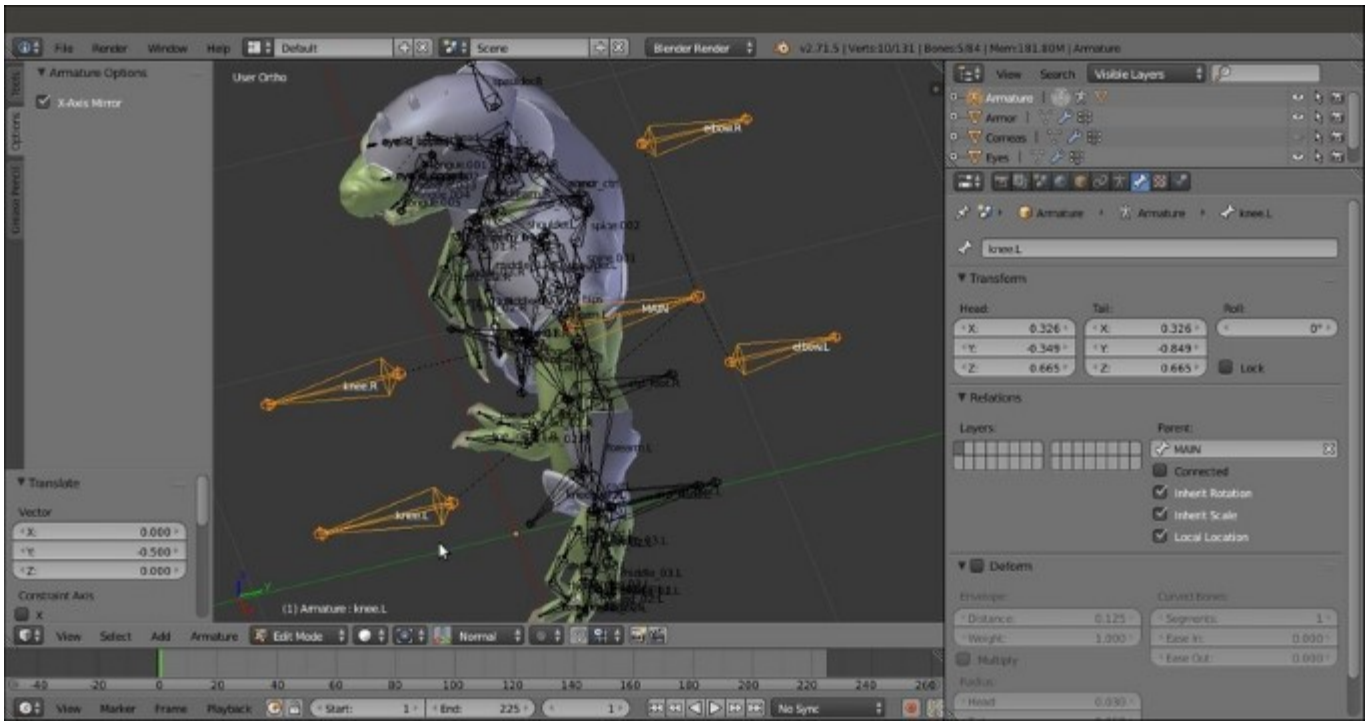
How to do it…

We now need to create the control bones; we can do it by extruding from the bones they will drive:

1. Press the *3* key on the numpad to go in the **Side** view and, if not already, select the **Armature**; if necessary, in the **Display** subpanel, change the visualization of the bones from **B-Bone** to **Octahedral**.
2. While still in **Edit Mode**, use *Shift* to select the joints of the **hand** with the **forearm** and the **calf** with the **foot** (it's enough only on one side) and extrude them going backwards (**0.400** along global *y* axis).
3. Rename the new extruded bones as **ctrl_hand.L**, **ctrl_hand.R**, **ctrl_foot.L**, and **ctrl_foot.R** respectively. Deselect the **Deform** item and unparent them all.
4. Select the **Head** of the **hips** bone and repeat: rename the extruded bone as **MAIN**.
5. Select the **hips** bone and in the **Relations** subpanel, parent it as a child of the **MAIN** bone:

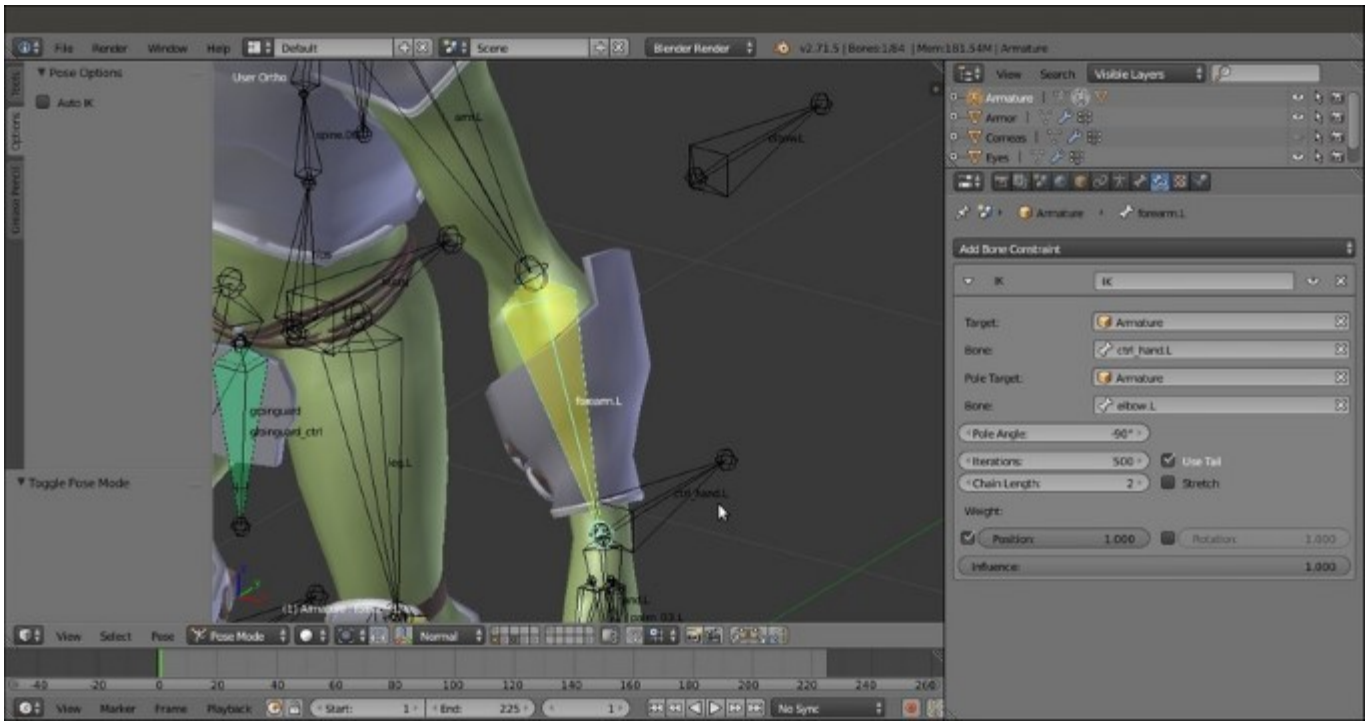
Extruding the control bones part 1

6. Select the **elbow** joint (between the **forearm** and **arm**) and extrude a new bone backwards; rename the extruded bone and the mirrored one as **elbow.L** and **elbow.R**. Disable the **Deform** item and parent them (**Keep Offset**) to the **MAIN** bone. Move them backwards by **0.500** along the global *y* axis.
7. Select the **knee** joint (between the **thigh** and **calf**) and extrude forward; rename the new bones as **knee.L** and **knee.R**. Disable the **Deform** item and parent them (**Keep Offset**) to the **MAIN** bone as well. Move them forward by **-0.500** along the global *y* axis;



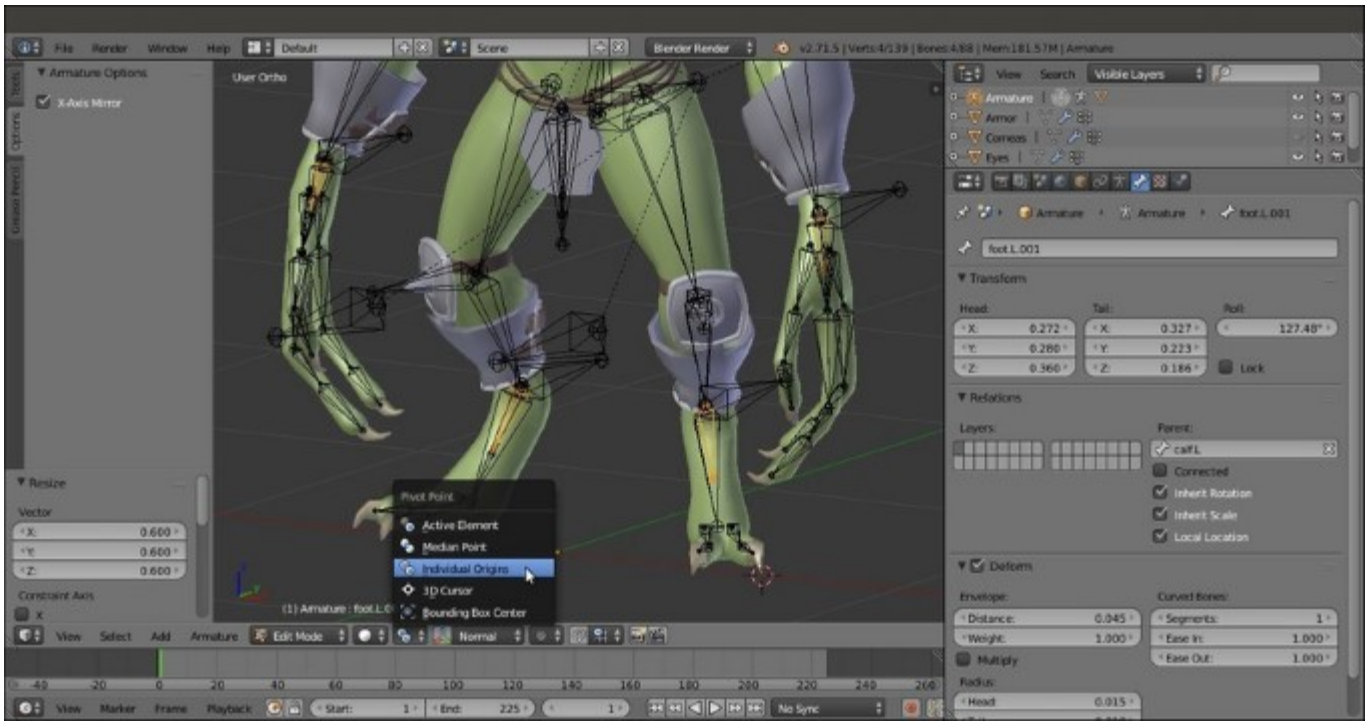
Extruding the control bones part 2

8. Go into **Pose Mode** and select the **forearm.L** bone; go to the **Bone Constraints** window and assign an **Inverse Kinematics** constraint. Set **Target** = **Armature**, **Bone** = **ctrl_hand.L**, **Pole Target** = **Armature**, **Bone** = **elbow.L**, **Pole Angle** = **-90°**, and **Chain Length** = **2**, and deselect **Stretch**. Repeat the process for the **forearm.R** bone:



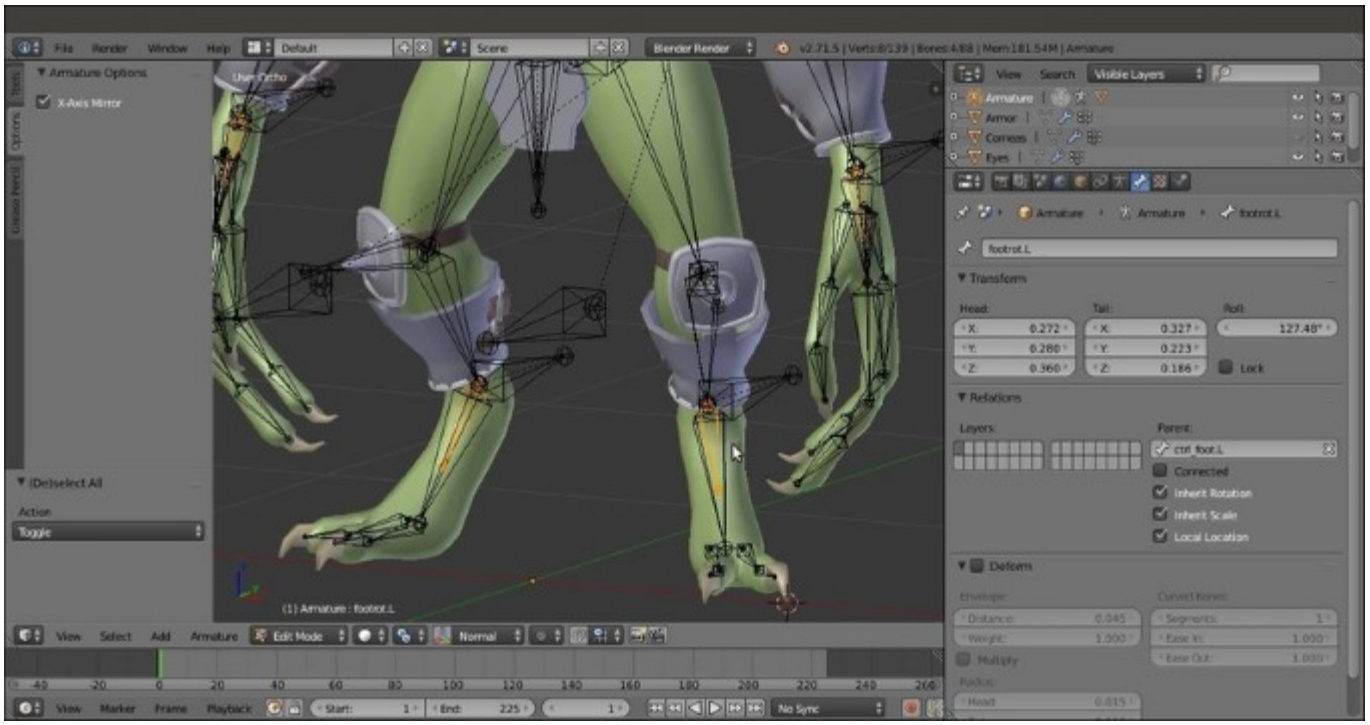
Assigning the IK constraint to the forearm.L bone

9. Do the same for the **calf.L** and **calf.R** bones, using the **ctrl_foot.L** and **ctrl_foot.R** bones as targets and the **knee.L** and **knee.R** bones as poles, but set the **Pole Angle** to **90°** for both.
10. Now, go back into **Edit Mode**, select the **hand** and **foot** bones, and use *Shift + D* to duplicate them. Click on the **Pivot Point** button on the 3D view toolbar, select the **Individual Origins** item, and then scale the duplicated bones smaller to **0.600**:



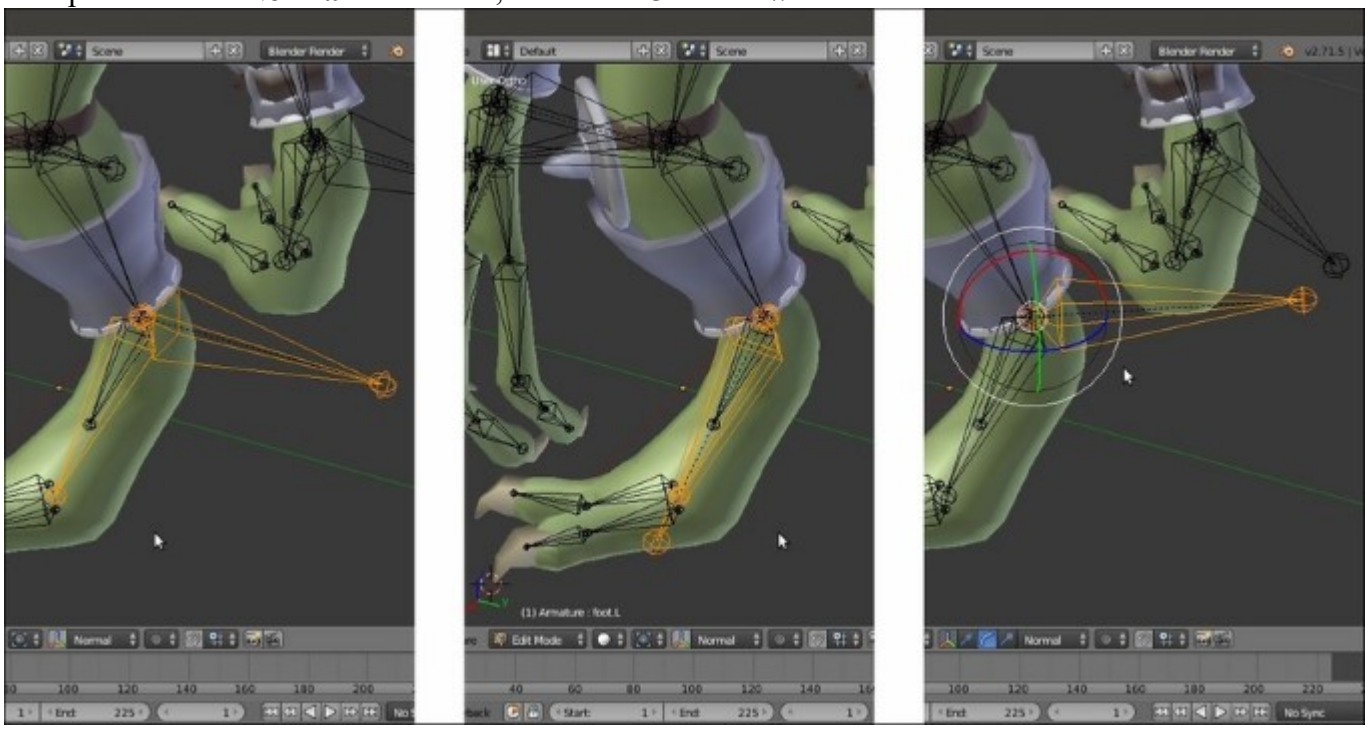
Scaling the bones smaller on their individual origin

11. Deselect the **Deform** item for all of them, and then rename them as: **handrot.L**, **handrot.R**, **footrot.L**, and **footrot.R**.
12. In the **Relations** subpanel (or by the *Ctrl* + *P* | **Keep Offset** shortcut), parent **handrot.L** to **ctrl_hand.L**, **handrot.R** to **ctrl_hand.R**, **footrot.L** to **ctrl_foot.L**, and **footrot.R** to **ctrl_foot.R**:



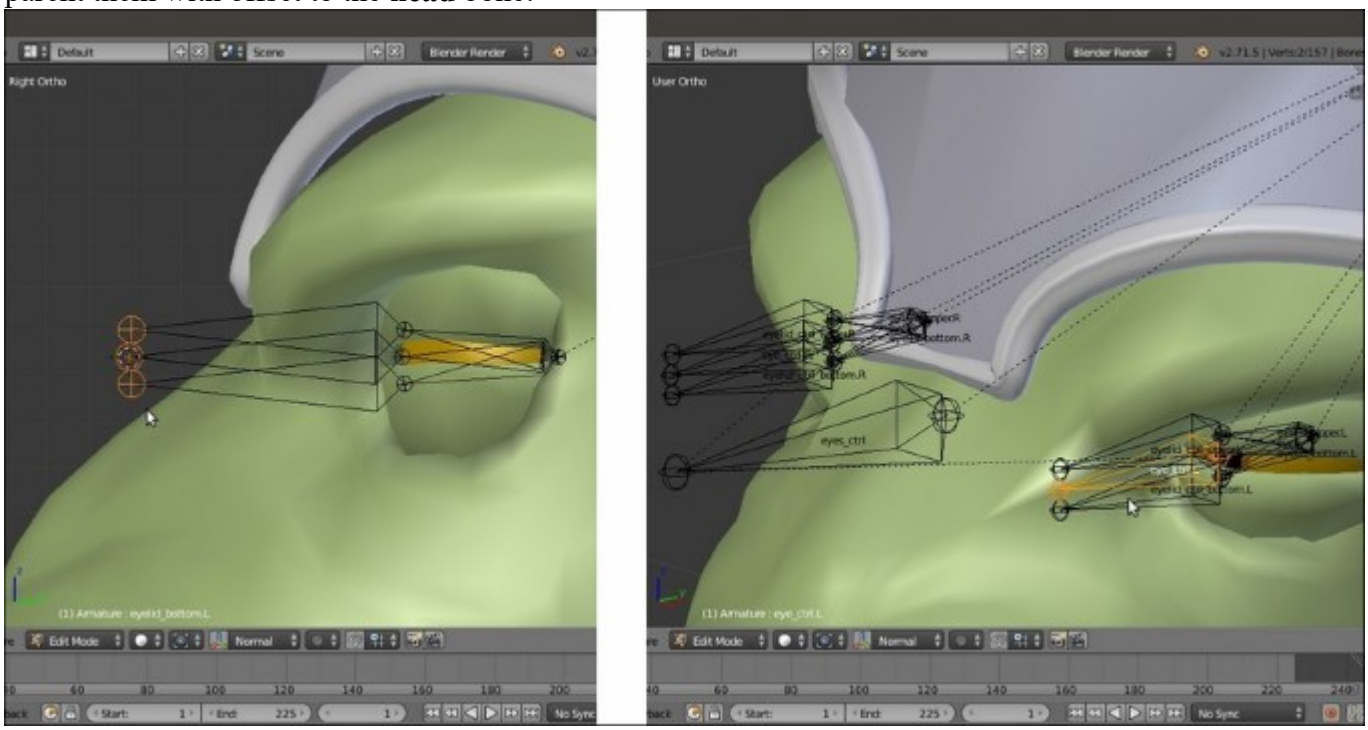
Using the Parent slot under the Relations subpanel

13. Use *Shift* to select the **ctrl_foot.L** bone and the **foot.L** bone and press *Ctrl* + *Alt* + *A* to align the first one with the active one; then, select only the **ctrl_foot.L** bone, and by the toolbar widget manipulator set to **Normal** orientation, rotate it **245°** on the *x* axis:



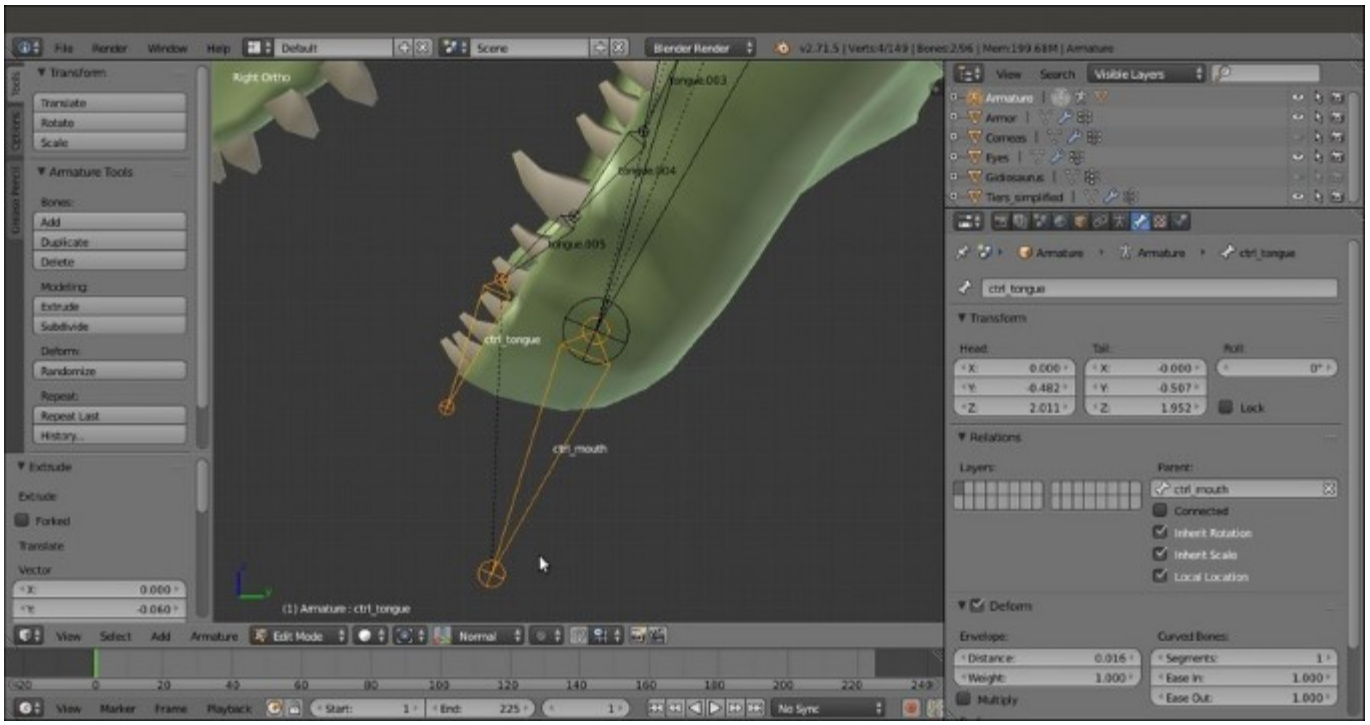
Rotating the bone on the Normal orientation by the widget

14. Go into **Pose Mode** and select the **hand.L** bone; assign a **Copy Rotation** bone constraint with **Target** = **Armature** and **Bone** = **handrot.L**, and set **Space** = **Pose Space** to **Pose Space**.
15. Repeat for the other **hand** bone and **feet**.
16. Select the **Tails** of the **eyelid_upper.L**, **eyelid_bottom.L**, and **eye.L** bones and extrude forward by **0.0600** along the *y* axis; rename them as **eyelid_ctrl_upper.L**, **eyelid_ctrl_bottom.L**, and **eye_ctrl.L** and the same names with the **.R** suffix for the mirrored ones.
17. Add a new bone in the middle front of the **eyes**, rename it **eyes_ctrl**, and parent it with offset to the **head** bone; then, select the **eye_ctrl.L** and **eye_ctrl.R** bones and parent them with offset to the **eyes_ctrl** bone.
18. Select the **eyelid_upper.L**, **eyelid_upper.R**, **eyelid_bottom.L**, and **eyelid_bottom.R** bones and parent them with offset to the **head** bone:



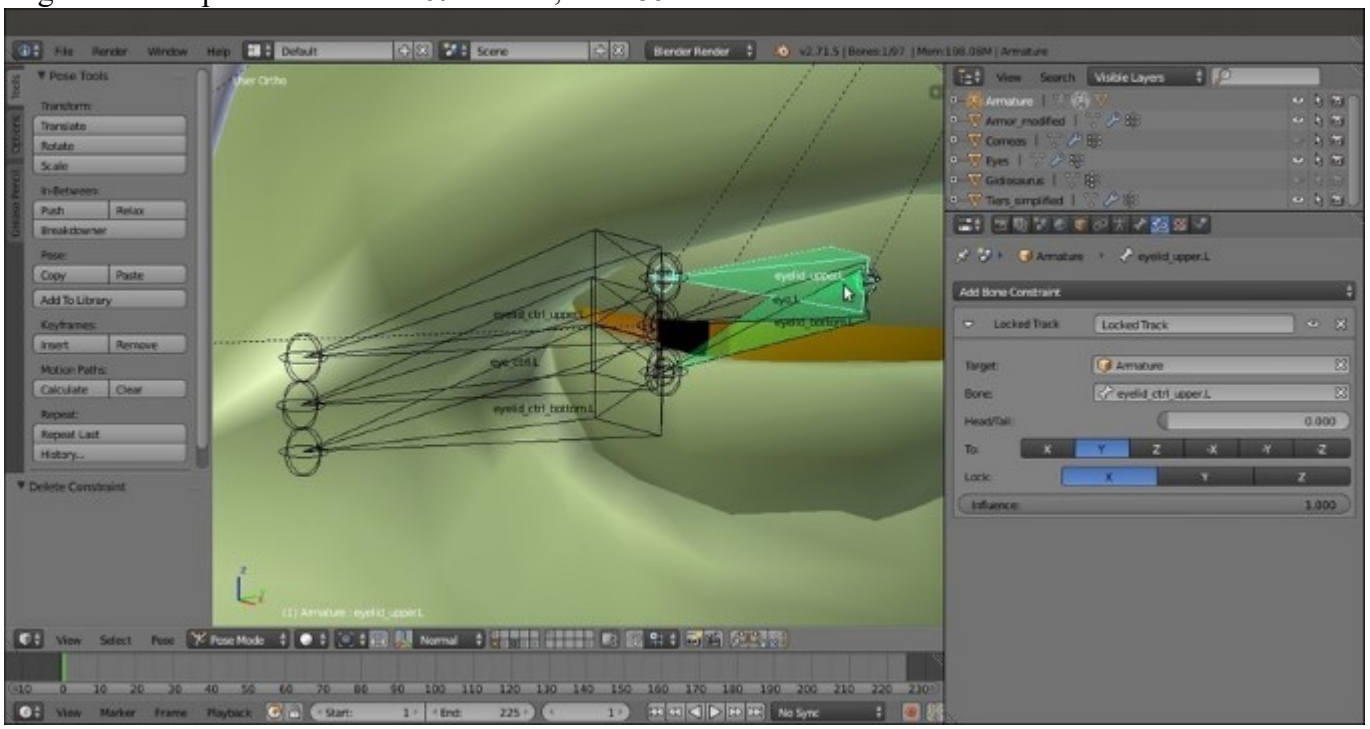
The eyes control rig

19. Select the **Tails** of the **mand** and **tongue.005** bones and extrude; rename the extruded bones as **ctrl_mouth** and **ctrl_tongue**. Parent with offset the **ctrl_tongue** bone to the **ctrl_mouth** bone and this latter bone to the **head** bone:



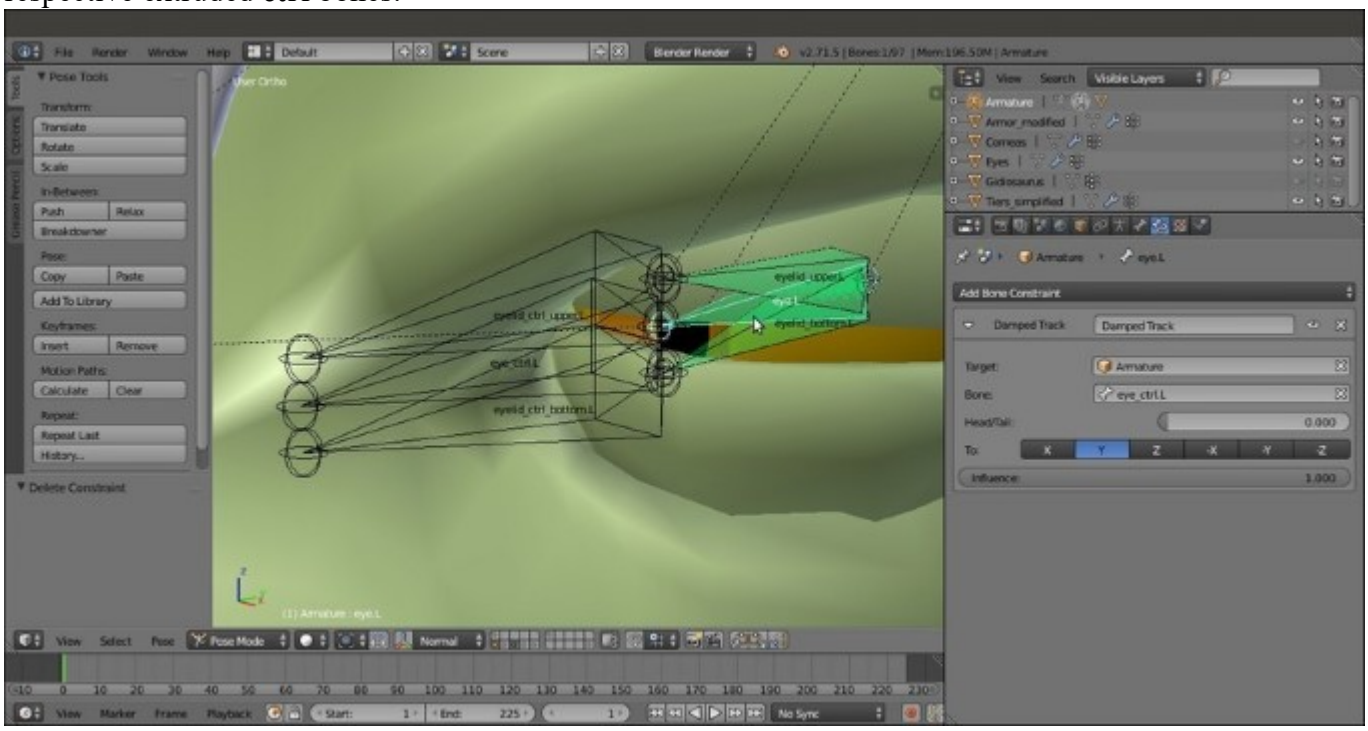
Extruding the control bones for the tongue and jaw

20. Go into **Pose Mode** and assign **Locked Track** constraints to the **eyelid_upper** and **bottom** with target to the respective extruded **ctrl** bones; set **Lock** to **X**:



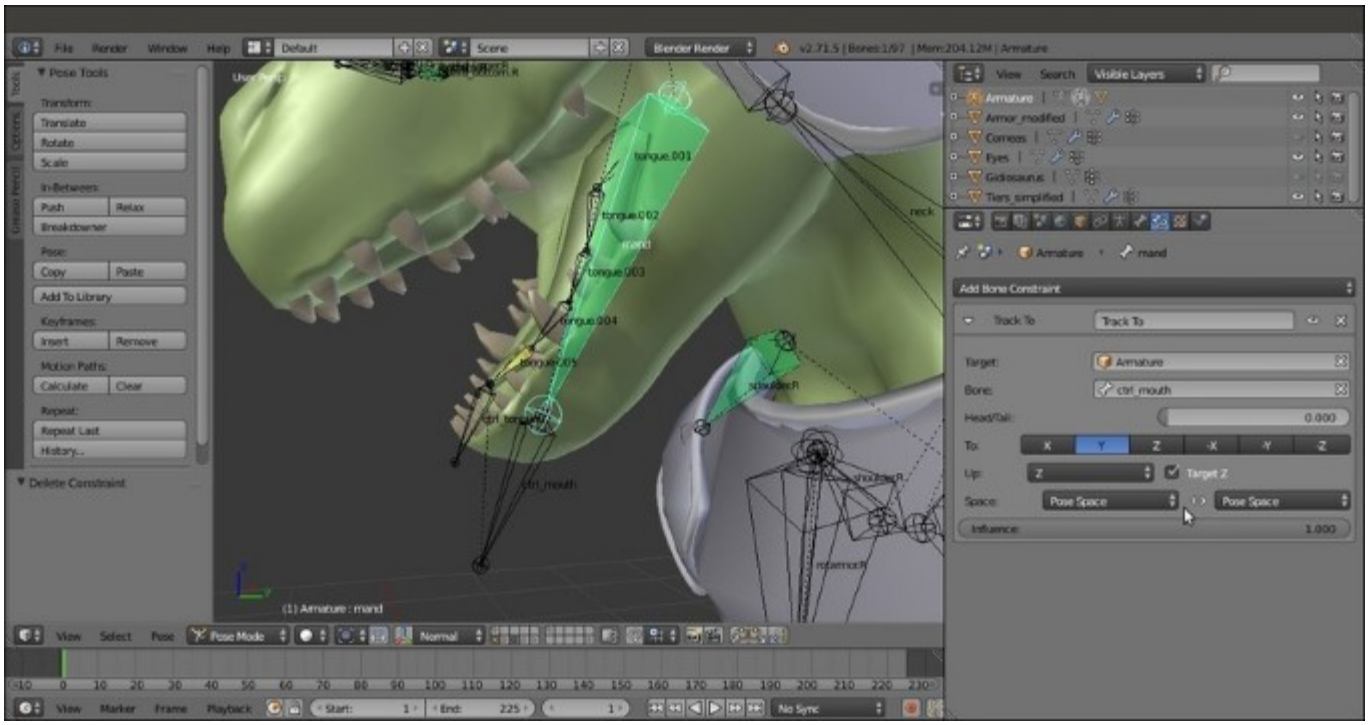
Assigning the Locked Track constraints for the eyelid's controls

21. Assign **Damped Track** constraints to the **eye.L** and **eye.R** bones, again with target to the respective extruded **ctrl** bones:



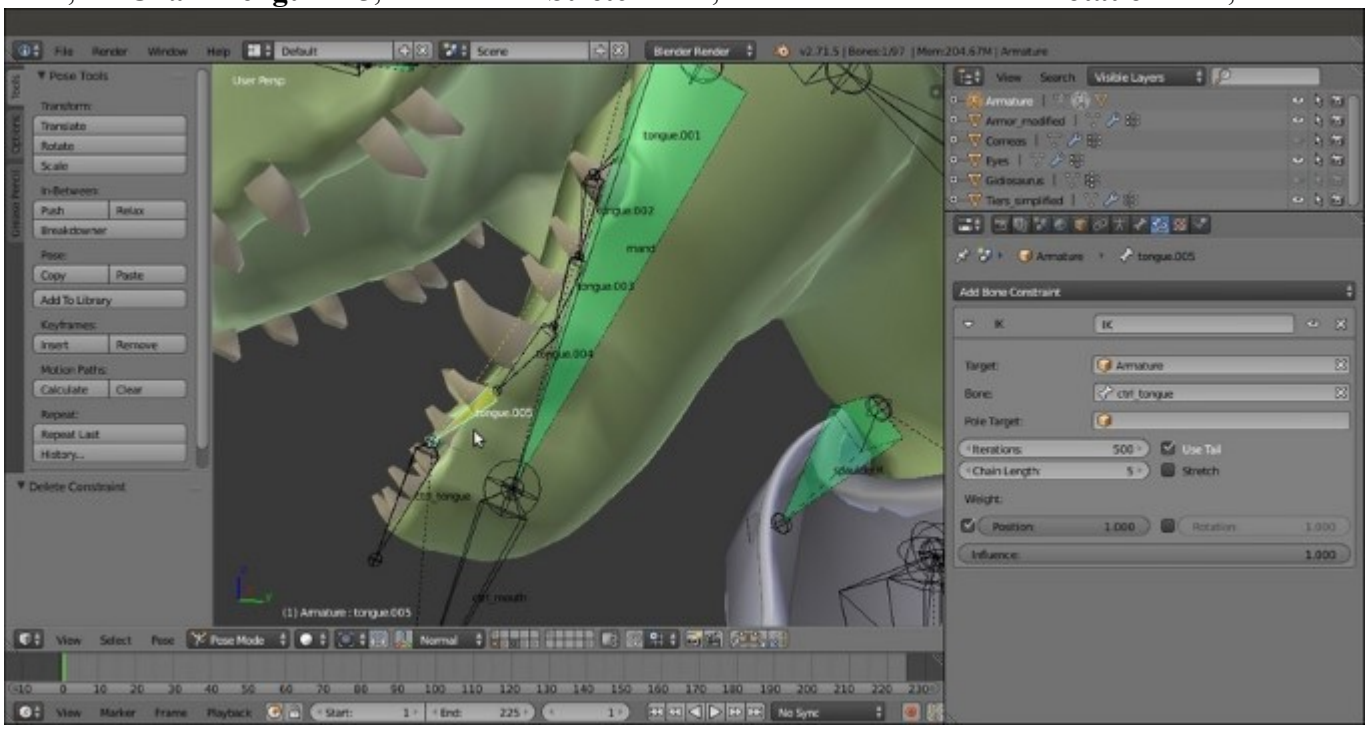
Assigning Damped Track constraints for the eye's controls

22. Assign a **Track To** constraint to the **mand** bone with target to the **ctrl_mouth** bone; check the **Target Z** item box and set **Space** = **Pose Space** to **Pose Space**:



Assigning a Track To constraint to the mand bone

23. Assign an **Inverse Kinematics** constraint to the **tongue.005** bone with target to the **ctrl_tongue** bone; set **Chain Length** to **5**, deselect the **Stretch** item, and then enable also the **Rotation** item;

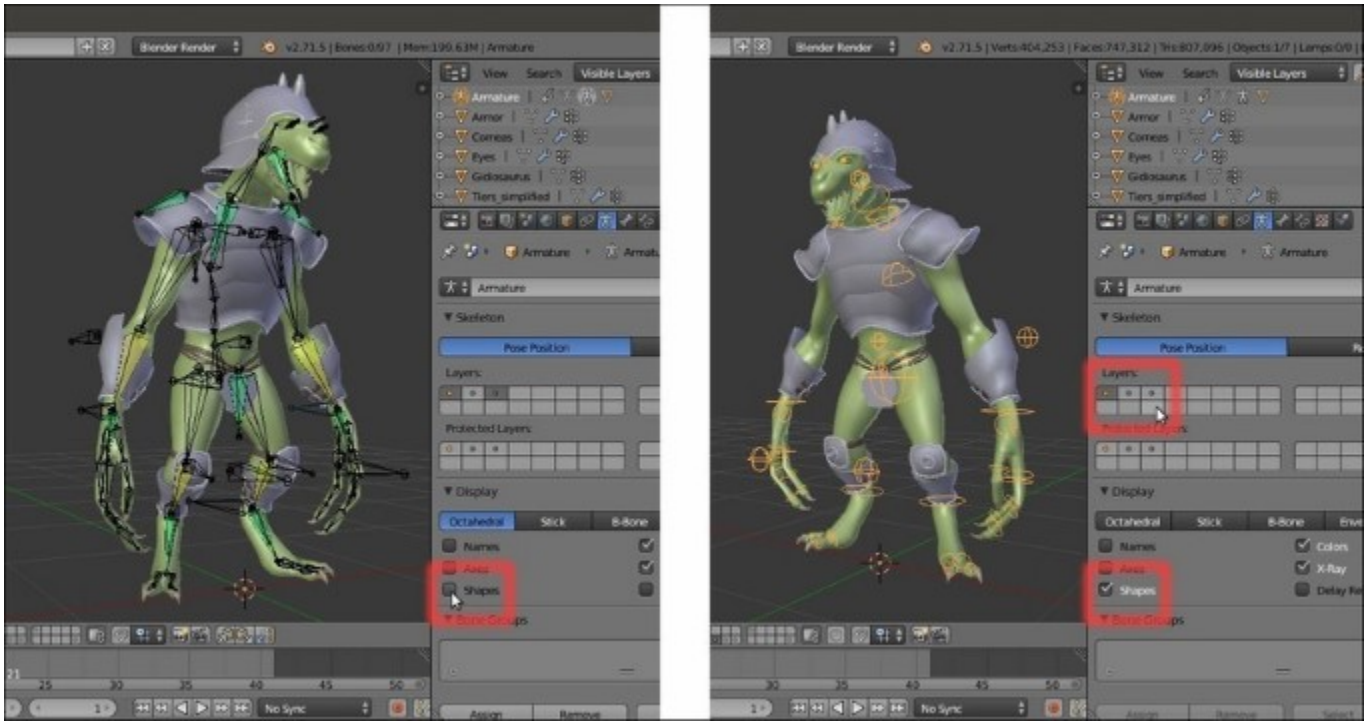


The IK constraint for the bone's chain of the tongue

At this point, the main controls for the **Gidiosaurus** rig are made; still something is missing, for example, the controls to drive **fingers** or/and **toes** bones as a whole, and also a *muscle system* layer of bones with the **Stretch To** constraints that can be added to improve the realism of the model. However, this latter option is quite a complex matter and, for the moment, we will stop here (maybe in another book).

The very last thing to do is to assign **Custom Shapes** (usually, simple meshes located on the last scene layer) to the control and animatable bones widget, and move the rest of the bones to the third **Armature** layer to be out of view.

To see the completed rig with the **Custom Shapes** assigned to the control bones, load the `Gidiosaurus_rig_from_scratch_03.blend` file;



The rig with and without Custom Shapes and with the deformation bones hidden on the third (disabled) Armature layer

See also

- http://www.blender.org/manual/rigging/index.html

Generating the character's Armature by using the Rigify add-on

We have already seen that the **Human Meta-Rig** armature is part of the **Rigify** add-on. It is a tremendously useful Python script, coded by Nathan Vegdhal, that we enabled two recipes ago, and in this recipe, we are going to use that to build the final rig for the **Gidiosaurus**.

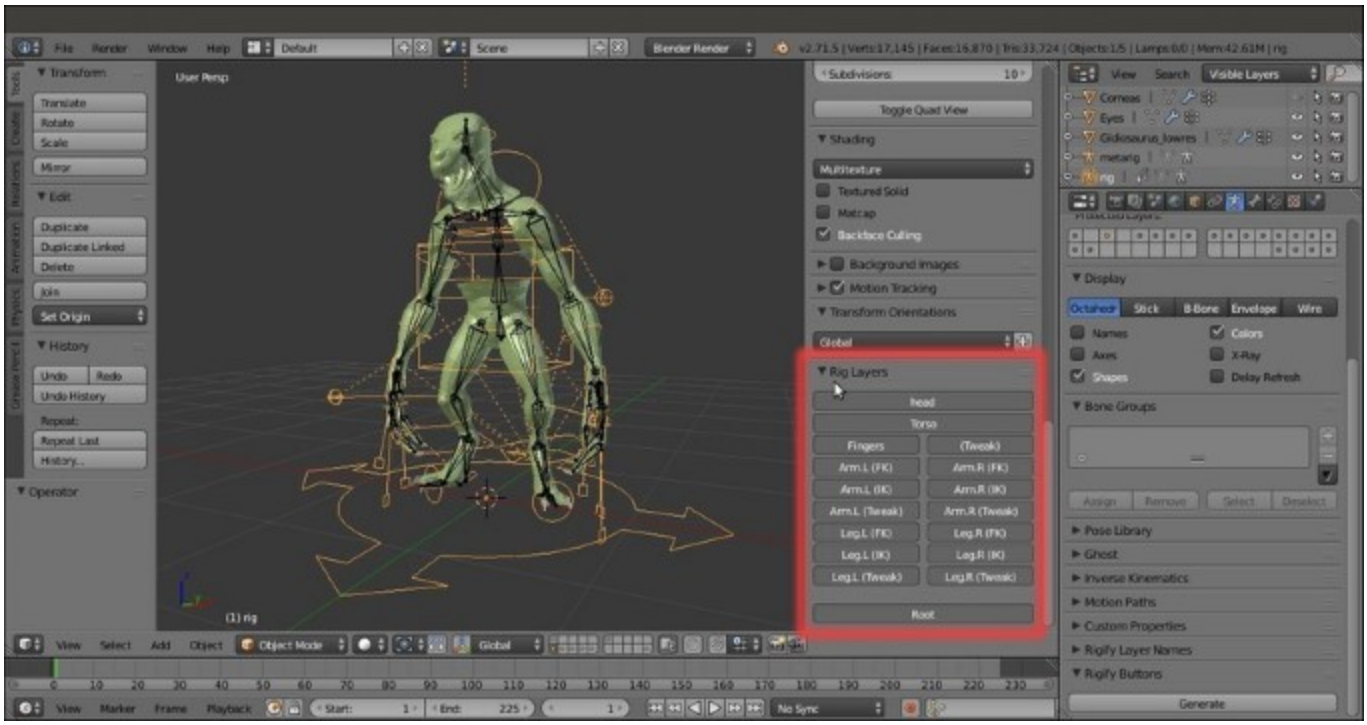
Getting ready

The preparation steps to use the **Rigify** add-on are the same as we did in the *Building the character's Armature through the Human Meta-Rig* recipe: after we have enabled the add-on in the **User Preferences** panel, we load the `Gidiosaurus_unwrap_final.blend` file, add the **Human metarig** to the scene, and then tweak the bone's position, rotation' and size in **Edit Mode** to fit the character's shape and topology.

Also, because the rig generated by the **Rigify** add-on uses some Python script, in the **User Preferences** panel, we must enable the **Auto Run Python Scripts** item (in the **File** | **User Preference** | **File** tab, click on the **Auto Run Python Scripts** checkbox).

How to do it…

At this point, in **Object Mode**, we can go to the bottom of the **Armature** window under the main **Properties** panel and click on the **Generate** button in the **Rigify Buttons** subpanel at the bottom of the **Armature** window; the add-on will automatically generate a new **rig** (simply named **rig** in the **Outliner**) using the **metarig** skeleton as an input and adding all the necessary **IK** constraints, the bone's widget controls (generated and located in the last scene layer), and also placing the different bones on different **Armature** layers that are easily accessible through the Python interface created by the script in the 3D window **Properties** sidepanel on the right-hand side (the **Rig Layers** subpanel):



The generated rig with the Rig Layers subpanel

Keep the **metarig** and move it to another layer, just in case we need to do some editing to it in the future; in fact, by testing the generated rig, sometimes you discover that something must be changed to work in a different way. In this case, it is enough to modify the **metarig** and generate the rig again by the add-on that automatically reuses the elements of any already existing rig and the bone's widgets on the last scene layer.

Keep in mind that the generated rig can (and often must) be edited later anyway; after the rig generation, save the file as `Gidiosaurus_rigify_01.blend`.

How it works…

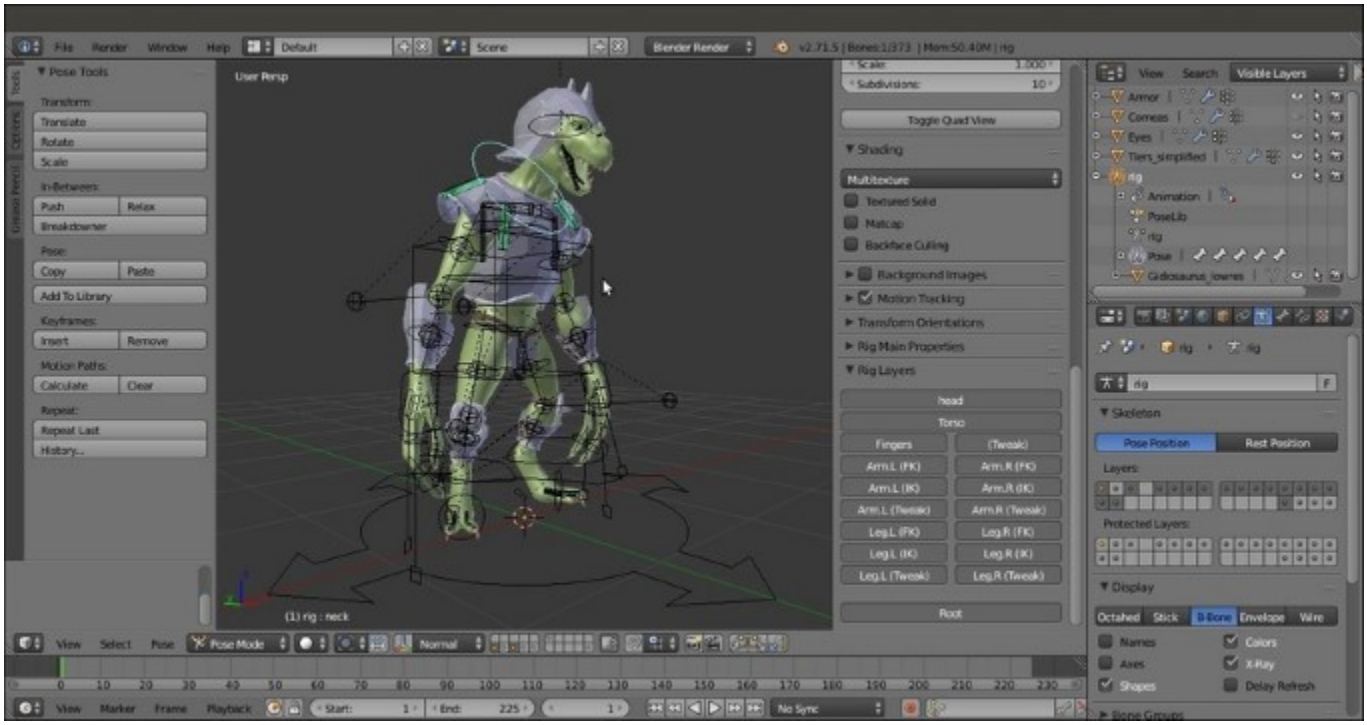
Being conceived to build a rig for a *generic biped humanoid* character, the **Rigify** add-on doesn't generate everything you need automatically: in our case, bones for the **jaw**, **tongue**, **eyes**, and **eyelids** must be added by hands after the rig regeneration and as explained in the *Building the character's Armature from scratch* recipe.

The choice to let face-rig elements, at least initially, out of the **Rigify** add-on has been intentional by Vegdhal, who thinks that a face-rig tool would probably be better as a separate add-on. By the way, in the last Blender releases, it is available, in the **Armature** menu, a **Pitchipoy human rig** option, which is an addition to the **Rigify** script that should help in the face's rig construction (http://pitchipoy.tv/?p=2026).

Also, at least for the moment, the **Rigify** add-on doesn't accept custom rig parts, but only the premade parts that we can add to the **metarig** by the **Add Sample** button under the **Rigify Buttons** subpanel in **Edit Mode**; for example, the premade leg rig (**biped.leg**) has only one bone and not two for the toes, as would be necessary for the **Gidiosaurus** character, but in any case, once the final rig is generated by the script, all the necessary additions and modifications can be (quite) easily made by hand.

Obviously, to modify the generated rig, knowing how a rig works in Blender is mandatory: you can rest upon the *Building the character's Armature from scratch*, *Perfecting the Armature to also function as a rig for the Armor*, and *Building the animation controls and the Inverse Kinematic* recipes in this chapter.

In the following screenshot, you can see the **Rigify**-generated rig modified with all the additional bones for the **Armor**, **eyes**, **mouth**, and **tongue**, with the necessary added constraints and the **two toed feet** bones; the file is saved as `Gidiosaurs_rigify_02.blend`:



The final total rig

See also

- http://blenderartists.org/forum/showthread.php?200371-Rigify-Auto-rigging-system-new-and-improved

Chapter 7. Skinning the Low Resolution Mesh

In this chapter, we will be covering the following recipes:

- Parenting the Armature and Mesh using the Automatic Weights tool
- Assigning Weight Groups by hand
- Editing Weight Groups using the Weight Paint tool
- Using the Mesh Deform modifier to skin the character
- Using the Laplacian Deform modifier and Hooks

Introduction

In the previous chapter, we saw the **rigging** stage, that is, how to build the character's rig (which in Blender is called an **Armature**) that will be used to deform the mesh for animations. In this chapter, instead, we are going to see quicker and more effective ways to do the **skinning** that is a necessary step to bind the bones of the **Armature** to the mesh's vertices so that they can be deformed.

To allow an **Armature** to deform a **Mesh**, they must be parented with some kind of relation; in Blender, usually you must select the **Mesh** and then *Shift* select the **Armature** and press *Ctrl + P* to parent them with different options.

This automatically makes the **Mesh** object a child of the **Armature** object and assigns the **Armature** modifier to the **Mesh**. In fact, the parenting would not be strictly necessary; it would be enough to assign an **Armature** modifier to the mesh and manually select the rig as a deforming object, but it's a good habit to use the *Ctrl + P* parenting to have the rig as a parent of the mesh, also in **Object Mode**. This way, whenever you move the **Armature** in **Object Mode**, the mesh will follow it automatically.

For the examples in these recipes, to skin the **Armature** to the **Gidiosaurus** mesh, we are going to use the final version of the rig we have built with our hands: the one saved as `Gidiosaurus_rig_from_scratch_02.blend`.

Anyway, if you want to put this to practice, in this chapter, with a more complex and complete **Rigify** armature (`Gidiosaurus_rigify_02.blend`), the procedure is exactly the same. In this case, even if not strictly necessary, remember that you can enable the **30th Armature layer** (in total there are **32**) to show the deforming bones; instead, disable the visibility of all the other bone layers also by the Python button interface in the **Rig Layers** subpanel under the 3D window **Properties** side panel: