



Quick answers to common problems

Blender 2.5 Character Animation Cookbook

50 great recipes for giving soul to your characters by building high-quality rigs and understanding the principles of movement

Virgilio Vasconcelos

[PACKT] open source*
PUBLISHING community experience distilled

Blender 2.5 Character Animation Cookbook

50 great recipes for giving soul to your characters by building high-quality rigs and understanding the principles of movement

Virgilio Vasconcelos

[PACKT] open source 
PUBLISHING community experience distilled

BIRMINGHAM - MUMBAI

Blender 2.5 Character Animation Cookbook

Copyright © 2011 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: June 2011

Production Reference: 1130611

Published by Packt Publishing Ltd.
32 Lincoln Road
Olton
Birmingham, B27 6PA, UK.

ISBN 978-1-849513-20-3

www.packtpub.com

Cover Image by Virgilio Vasconcelos (virgiliovasconcelos@gmail.com)

Credits

Author

Virgilio Vasconcelos

Project Coordinator

Joel Goveya

Reviewers

Allan Brito

Martin Poirier

Proofreader

Aaron Nash

Acquisition Editor

Sarah Cullington

Indexer

Tejal Daruwale

Development Editor

Hyacintha D'Souza

Production Coordinator

Aparna Bhagat

Technical Editor

Aaron Rosario

Cover Work

Aparna Bhagat

About the Author

Virgilio Vasconcelos is an animator based in Brazil, who uses Blender as his 3D tool to produce animations. He is also a university professor, teaching digital 3D and 2D animation at Universidade Federal de Minas Gerais (UFMG). His specialties include character rigging and animation, and his first tryst with Blender was back in 2003. He has worked as lead 3D artist at Nitrocorpz Design Studio, and has several personal and commissioned productions recognized by the Blender community, being awarded and nominated for artistic categories in events such as Blender Conference and BlenderPRO.

You can watch his animations, read his blog and contact him at <http://www.virgiliovasconcelos.com>.

Acknowledgements

No book is the product of just the author—he just happens to be under the spotlight with his name on the cover. The contribution of a number of people was crucial to bring this book to fruition, and it would take far more space than I have available to thank each one individually.

A special note goes to Chaitanya Apte, Hyacintha D'Souza, and Joel Goveya from Packt Publishing, without whom this book wouldn't exist. Thank you for believing in me, and for all the wonderful guidance and professionalism throughout these months. You and the entire Packt Publishing team did an outstanding job to help produce a high quality publication.

I must also thank the coding wizards who don't get tired of making Blender such an amazing tool, which crosses the line of being just an open source graphics application to be a respected tool by all CG professionals, regardless of its license. I'm grateful to people such as Ton Roosendaal, the head of the Blender Foundation, who is the main man responsible for what Blender has become; artists such as Bassam Kurdali and Nathan Vegdahl, from whom I've learned a lot by studying their rigging approaches; and the active user community in forums and discussion lists, such as blenderbrasil-dev in Google Groups. Aside from Blender fellows, I'm also very thankful to the guys at Nitrocorpz Design Studio, where I gained more experience and knowledge working on projects to write about in this book.

Along with all these people, a book isn't worth without a reader. If you're reading this now I want to thank you and let you know that I've put a lot of effort into making something very useful for you and your projects.

About the Reviewers

Allan Brito is a Brazilian architect, specialized in information visualization. He lives and works in Recife, Brazil. He works with Blender 3D to produce animations and still images, for visualization and instructional material. Besides his work with Blender as an artist, he also has wide experience in teaching and researching about 3D modeling, animation, and multimedia.

He is an active member in the Blender users community, writing about Blender 3D and its development for websites in Brazilian Portuguese (<http://www.allanbrito.com>) and English (<http://www.blender3darchitect.com> and <http://www.blendernation.com>). Besides his two blogs, he has written three books about Blender, in both English and Brazilian Portuguese, covering topics such as architectural visualization, mechanical modeling, and general Blender guides.

To know more about the author, visit the website <http://www.blender3darchitect.com>, where he covers the use of Blender and other tools for architectural visualization.

I want to thank my wife Erica for the support during the review of this book.

Martin Poirier is a software developer with a Master's Degree in Computer Graphics, specialized in character animation and simulation. He's been involved with the FOSS Blender project since early 2003, soon after its open source debut, working on plenty of things here and there, but mostly on the transformation system. Lately, he has been developing a distributed rendering solution for Blender animations.

Martin has reviewed and contributed to the official *Blender 2.3 Guide*. He is also contributing to the wiki version of the Blender Manual (occasionally).

www.PacktPub.com

Support files, eBooks, discount offers and more

You might want to visit www.PacktPub.com for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why Subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

*This book is dedicated to my love, Suryara, who brings so much happiness to my days;
and my mom and sister, Leonora and Leticia, who support me unconditionally
when I'm trying to draw with a computer.*

This book would not have been possible without your love and understanding.

Table of Contents

Preface	1
Chapter 1: Get Rigging	5
Introduction	5
Defining good orientations for your bones	6
Using separate bone chains for different tasks	11
Customizing shapes and colors for your bones	18
Using corrective shape keys	27
Making an IK-FK switcher	33
Tips on weight painting your character	40
Chapter 2: Rigging the Torso	47
Introduction	47
How to create a stretchy spine	48
Rigging the pelvis	56
Making your character breathe	61
Controlling the neck and head	63
Chapter 3: Eying Animation	69
Introduction	69
How to control where your characters look at	69
The eyelids controllers	74
Controlling the pupils	82
Chapter 4: Poker Face? Facial Rigging	91
Introduction	91
Adding expressions using Shape Keys	92
Face controls with lattices	99
Creating the jaw controller	105
Controlling your tongue	109

Chapter 5: Hands Down! The Limbs Controllers	113
Introduction	113
Controlling fingers	114
Creating IK legs with a three-pivot foot	119
Stretch those limbs!	127
Setting up the shoulders	132
Cartoon bending for arms and legs	135
Different spaces for IK hands	139
Chapter 6: Blending with the Animation Workflow	145
Introduction	145
Animating in layers	146
Changing between FK and IK in a shot	153
Grasping and throwing objects	157
Silhouette and mirrored rendering	160
Tracking animation arcs	164
Using video for background reference	167
Working with linked assets and characters	171
Non-linear animation	177
Chapter 7: Easy to Say, Hard to Do: Mastering the Basics	181
Introduction	181
Adjusting and tracking the timing	182
Spacing: favoring and easing poses	186
Anticipating an action	192
Using squash and stretch	195
Breaking the symmetry	200
Chapter 8: Shake That Body: The Mechanics of Body Movement	205
Introduction	205
Animating a tennis serve	206
Heavy metal	213
Glory for your team: kicking the ball	220
Run, Forrest! (in cycles)	225
Chapter 9: Spicing it Up: Animation Refinement	233
Introduction	233
It's time for secondary actions	234
Hold, but not still: using moving holds	238
Animating characters with appendages	242
Like clay: refining with the AniSculpt technique	247

Chapter 10: Drama King: Acting in Animation	253
Introduction	253
In the blink of an eye	254
Walking with style	259
Talking heads (and bodies)	265
Appendix: Planning Your Animation	273
Introduction	273
Creating thumbnails with Grease Pencil	273
Naming conventions	276
Extremes, Breakdowns, Inbetweens, ones and twos	277
Index	281

Preface

This book offers clear, illustrative, and easy-to-follow recipes to create character rigs and animations for common situations. Bring your characters to life by understanding the principles, techniques, and approaches involved in creating rigs and animations; you'll be able to adapt them to your own characters and films.

What this book covers

Chapter 1, Get Rigging—It's about the essential concepts and recipes you need to know before creating the controllers for your character. Master them and you'll avoid lots of headaches in the future.

Chapter 2, Rigging the Torso—Here we'll begin rigging our character's torso. It's a crucial set of recipes where you'll learn how to control things such as the pelvis, the neck, and how to stretch them in a cartoony way.

Chapter 3, Eyeing Animation—Here's a chapter dedicated to controlling our character's eyes. The eyes are what our audience looks at the most, so we have to carefully create good controllers for that part of the body.

Chapter 4, Poker Face? Facial Rigging—This chapter is dedicated to teaching you how to enable our characters to talk and express their feelings through facial expressions.

Chapter 5, Hands Down! The Limbs Controllers—In this chapter, we'll see how to create all kinds of controllers for arms, legs, feet, fingers, and shoulders.

Chapter 6, Blending with the Animation Workflow—It's time for animation, and in this chapter we'll see some important concepts and techniques to get started on the right foot and work efficiently.

Chapter 7, Easy to Say, Hard to Do: Mastering the Basics—Here we'll see some very important principles of animation applied to our characters. These principles are crucial for virtually every piece of animation you'll need to create.

Chapter 8, Shake That Body: The Mechanics of Body Movement—In this chapter we'll mix everything we've learned until here and apply them to real world situations.

Chapter 9, Spicing it Up: Animation Refinement—Now that we have achieved proper movement in our characters, it's time to take them to the next level with refinements.

Chapter 10, Drama King: Acting in Animation—Animators don't just move puppets around; they make you believe the characters are alive. Here we'll see some recipes about why our characters move, instead of how.

Appendix, Planning Your Animation—The Appendix talks a bit more about some concepts related to animation and how you can prepare yourself to make the perfect shot.

Who this book is for


This book will be handy for those Blender users who already know the basics of adding, modeling, and rendering objects within the program, but are eager to learn how to turn a character's mesh into a more life like entity.


Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "Select your entire bone, open the **Specials** menu (press the W key), and choose **Subdivide**."

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "Open the file 001-Orientation.blend from this book's support files."

[ Warnings or important notes appear in a box like this.]

[ Tips and tricks appear like this.]

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to feedback@packtpub.com, and mention the book title via the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the **SUGGEST A TITLE** form on www.packtpub.com or e-mail suggest@packtpub.com.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at <http://www.PacktPub.com>. If you purchased this book elsewhere, you can visit <http://www.PacktPub.com/support> and register to have the files e-mailed directly to you. Alternatively, the author also maintains a copy of the code on his website at <http://virgiliovasconcelos.com/blender-animation-cookbook/>.

Downloading the color images of this book

We also provide a PDF file that has color images of the screenshots used in this book. The high resolution color images will help you better understand changes in the output. You can download this file from https://www.packtpub.com/sites/default/files/32030S_Color_Images.pdf.

The author also maintains a copy of the graphics as well as the other code files from this book at <http://virgiliovasconcelos.com/blender-animation-cookbook/>.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/support>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at questions@packtpub.com if you are having a problem with any aspect of the book, and we will do our best to address it.

1

Get Rigging

In this chapter, we will cover the following topics:

- ▶ Defining good orientations for your bones
- ▶ Using separate bone chains for different tasks
- ▶ Customizing shapes and colors for your bones
- ▶ Using corrective shape keys
- ▶ Making an IK-FK switcher
- ▶ Tips on weight painting you characters

Introduction

So, you've successfully modeled an awesome character in Blender. After hours of careful and detailed work you have built a very appealing protagonist with a good topology for your next animation, but there's an issue: how do we make it look more life like, and also, how do we make it move?

Since a character model can be made of thousands of vertices, moving them individually across the 3D space is virtually impossible. We need an easier way of moving our models, and this way is called rigging.

Rigging is the process of creating a series of controls (the "Rig") to deform another object, which is often a character mesh. It involves creating special objects that move selected groups of vertices at once. This is the principle behind Skeletal Animation, where objects called "bones" are used to control parts of our models.

In Blender, there is a special object called **Armature**, which can be described roughly as a set of related bones that are used to control a mesh. To use an analogy, bones are for armatures as vertices are for meshes. Armatures can be added within the 3D View by pressing *Shift + A* and choosing **Armature | Single Bone** on the menu. Similar to meshes, armatures also have an Edit Mode accessible through the *Tab* key, where you can add, change, and remove bones as you wish. Bones can also be linked, creating a chain of hierarchically related bones.

Rigging is often referred to as one of the most difficult subjects in 3D animation. When creating a character rig, there are many aspects that you have to keep in mind, and two of them should be observed as major guidelines:

- ▶ The rig must be simple enough to be used by the animator
- ▶ The rig must be complex enough to allow convincing movements for your characters

Finding an ideal balance between complexity of features and ease of use is the Holy Grail of character rigging. On one hand, if a rig is too simple it can be harder for the animator to give the character an "illusion of life". On the other hand, an extremely complex rig can be a nightmare: the animator should not require a tutorial to be able to start posing a character. It has to be straightforward enough to be used instinctively. Of course, a skilled animator should be able to achieve an amazing piece of animation even with a very simple rig, but the job of a character rigger is to make the animator's life easier.

Because every animation project has its own sorts of challenges and demands, there is no absolute right or *wrong* way to build a character rig. What we will see here are best practices that should apply to most situations. These recipes should be dealt with just as in a traditional cookbook: feel free to add spice to suit your personal taste.

Defining good orientations for your bones

When creating rigs for 3D characters in Blender, there is one mistake that is probably the most common of all, and it is also responsible for lots of headaches in the future: the orientations of the bone chains.

Every time we have to create a bone chain to allow our character to do a specific movement, some people (maybe in a hurry) often overlook this foundation of a good character rig. Since our characters and its bones live in a three-dimensional space, everyone familiar with 3D concepts should know that they are subject of the three world axes: X, Y and Z.

Along these concepts, we should be comfortable with the idea of "local" and "global" coordinates. **Global coordinates** are the ones relative to the scene: every scene has its Up or Down (Z axis), Left or Right (X axis), and Front or Back (Y axis) coordinates. Every object in a scene also has its own, or local, coordinates to allow easier transformations. To make an analogy with our world, "going East" would be the global coordinates while "turning right" refers to your local coordinates.

For instance, we should be allowed to bend a character forward regardless of its rotation and position relative to the scene. This "bending forward" would be too difficult to achieve using only the global coordinates; that's why we can use the local ones.

Getting ready

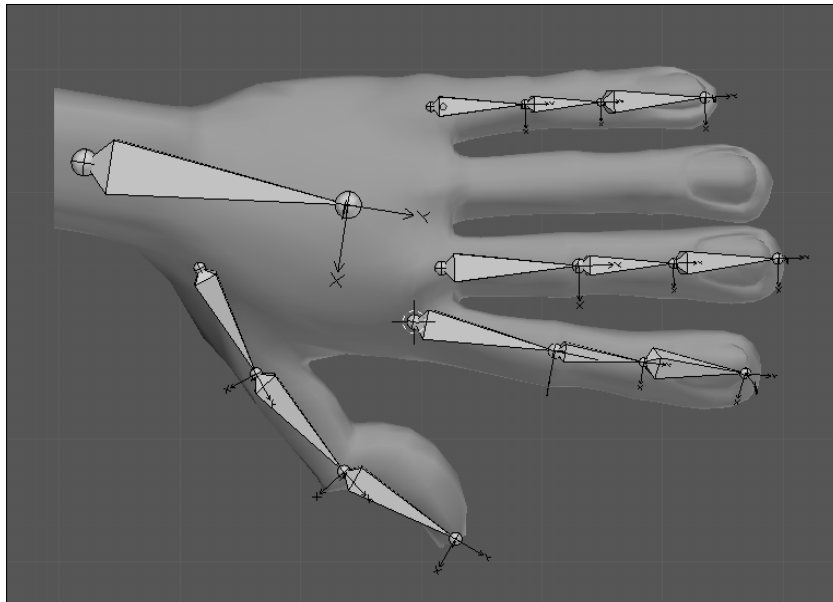
Using the concept of local coordinates, we have also to define some conventions such as which axis we are talking about when bending "forward". We have to pay attention to the same organization of the bones, where a chosen local axis (for instance, local X) would be the same for all "forward" transformations, be it a finger or a knee. The character **Otto**, which is used throughout this book, uses the X local axis for the most common transformations, such as for bending the elbows and knees, closing the fingers, or bending the torso forward. This makes it easier to pose our character without having to worry about which axis you should use: if in doubt, use X!

Here we'll see how to create and correct bone chains in order for them to be more coherent and easier to manipulate.

How to do it...

Let's suppose you want a chain with three bones for a finger:

1. Open the file `001-Orientation.blend` from this book's support files. You'll see a hand model with four of its finger bones already set. I've let the ring finger for you to rig, like in the next screenshot:



2. Position your 3D cursor where the first finger bone should be created: select your mesh, enter the Edit Mode (*Tab*), select a vertex or group of vertices at the base of the finger and press *Shift* + *S*. Choose **Cursor to Selected**.
3. Go back to Object Mode (*Tab*), select the armature, enter its Edit Mode (*Tab*) and press *Shift* + *A* to add a new bone under the cursor location.

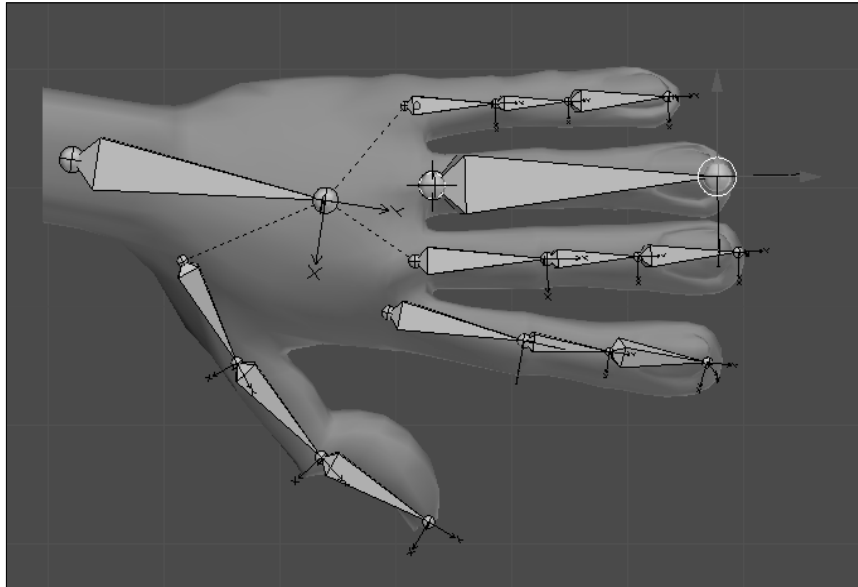


In order to view better what we're doing, I've enabled the X-Ray and Axes display modes, in the **Properties** panel, under the **Object Data** tab.

Now you just move the tip of this bone until the finger's first joint and extrude the bone two times, right? WRONG!

Extruding bones is just what one would normally do in order to create a bone chain, but that brings to our new bones some unwanted rotations.

4. Select the tip of your bone and move (press the *G* key) it until the tip of the finger, as seen in the next screenshot:



5. Select your entire bone, open the **Specials** menu (press the *W* key), and choose **Subdivide**. Select one of the bones and repeat this process.
6. Move (press the *G* key) the joints to the appropriate places of the finger, and the orientations will be consistent. Note that the X axis of each bone is always pointing toward us, while the other bones on the hand have their Z axes pointing up. We need them all consistent.

7. Select the bones you have just created, press *Ctrl + R*, and type 90. This will correct their rotations, making their Z axes point toward us (check the *Axis conventions* section at the end of this recipe to know more about this).
8. Select the hand mesh, hold *Shift*, select a bone, and press *Ctrl+ P*. Choose **With Automatic Weights** to get a basic deformation on the hand.
9. Now you can rotate the bones under their local X axis (*R + X + X*) to see what happens.

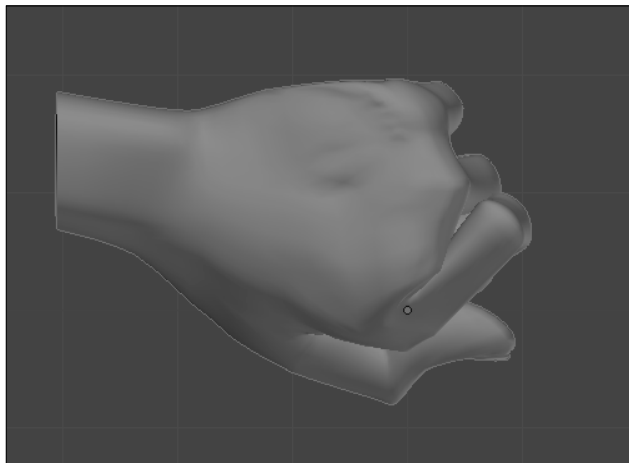


When you tell Blender to rotate (*R*), move (*G*), or scale (*S*) an object, you can use some key modifiers to tell it in which axis that transformation must happen. If you press *X*, *Y*, or *Z* one time, you're telling it that one of these global coordinates must be used. If you press the modifier key twice (*X + X*, *Y + Y*, or *Z + Z*) you're demanding that the transformation happen regarding that local coordinate.

How it works...

When using a correct bone orientation upon its creation, you avoid the need to correct the armature later. The orientation of the bones is often overlooked, and problems at this stage will be painfully noticed on later stages, when the animator tries to move the bones in a coherent way.

To demonstrate that, I've created the index finger bones with the usual "extrude" technique. Let's say the animator wants to close the fingers: this should be accomplished by selecting each bone and pressing *R* to rotate and *X* twice to select the local X axis. The next image shows the results of the previous action with the bones hidden, to demonstrate how the extruding technique on the index finger leads to unwanted results. We wanted to close the finger, not twist it like that. An animator would have a hard time trying to figure out which axis should be used for every finger. Talk about being counterproductive!



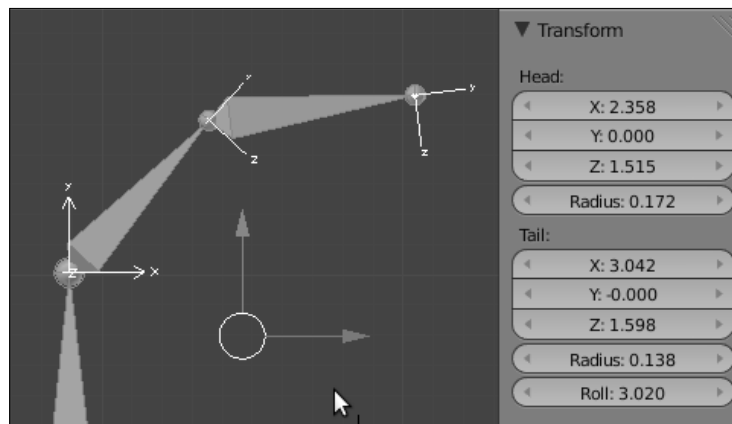
There's more...

What if you have already created bone chains with this orientation disorder? Do you have to recreate everything from scratch? No, there's hope for us all. This process is just to avoid the need for correcting the bones later, by creating them with the right orientations from the beginning. This means we can adjust the orientation by hand at any moment.

Correcting the orientation

You can always correct or define the orientation of a bone through its **Roll** value. In the Armature's Edit Mode, select the bone(s) that you want to correct, press **Ctrl + R**, and move your mouse. As with any transformation in Blender, you can use it along with **Ctrl** (to do it in steps of five degrees) or **Shift** (to get softer transformations).

You can manually view and set the **Roll** angle of a bone through the **Properties Panel** (press the **N** key), just below the **Radius** slider, as shown in the following screenshot:



Another shortcut for automatically correcting the **Roll** value is **Ctrl + N**, which tells Blender to guess what the best roll angles are based on their Z axes, which will all point to the same direction ("Up" or the cursor location).

Axis conventions

Another important thing to keep in mind when creating bones is axis conventions. It means you should always set the "front" of a bone to a given axis. This "front" is the default axis for a transformation, usually a rotation.

For instance, a humanoid character has some default movements, such as bending an elbow, knee, or finger. The bones for all these parts point in different directions, but you can set their roll values in a way that the animator's life becomes a little easier. A common approach is setting the X axis of a bone as the default transformation angle, so when the animator wants to bend and elbow, a knee, or a finger, it's just a matter of using the X local axis for that.

There's no need to wonder "what local axis should I use for this transformation?". In the case of our armature, we can select all the bones, press *Ctrl + R*, and type 90, so the bones' X axes point at us in the Front view.

This way you get consistence throughout your rig, which is a must in professional workflows. I've seen rigs where different fingers in a hand required a different default axes for bending. A nightmare!

Rigify

A very cool way to add new bones and chains is by using the **Rigify** add-on which comes bundled with Blender 2.5. You can enable it in the User Preferences Window (press *Ctrl + Alt + U*), in the Add-ons tab.

With this add-on enabled you can add predefined bone chains for body parts or even a full human body. The great advantage is that you don't need to worry about names or orientations, since they come configured. You only have to worry about correctly adjusting the preset chains to the proportion of your character mesh.

See also

Chapter 5: Controlling fingers

Using separate bone chains for different tasks

A useful approach when building rigs is to create more than one bone chain to accomplish different tasks. The idea behind this is to not overwhelm you with so many functions attached to one single bone, making the rig easier to understand and modify.

It is useful to separate the bone chains by their main functions to make things easier to manage: one chain that will only deform your character's mesh, one for creating **Inverse Kinematics (IK)** controllers, another for **Forward Kinematics (FK)** controllers, interface, helpers, and so on.

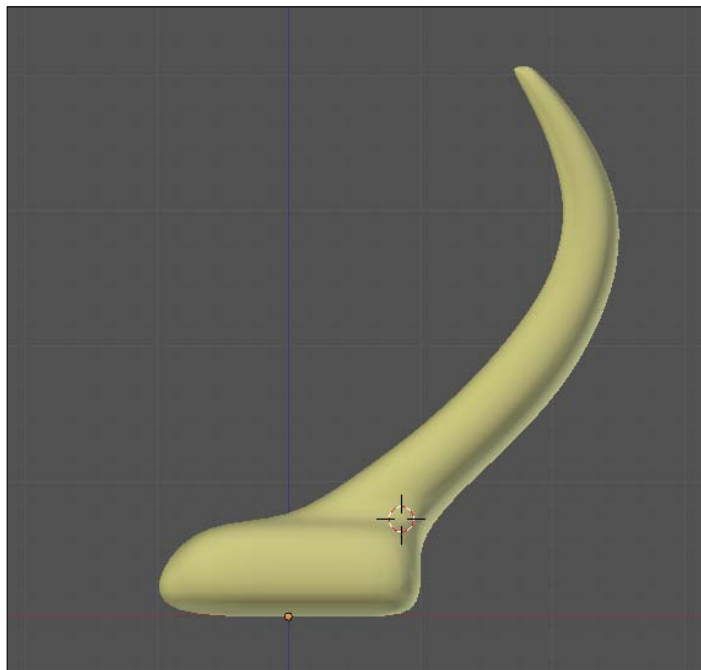
By creating them separately, you can make changes without breaking things in your rig. If you stack all the functions and constraints on one single chain, a little change can make a real mess. By separating them you can also make your rig more appealing and usable by defining custom shapes, colors, and hiding bones that shouldn't be touched by the animator.

Getting ready

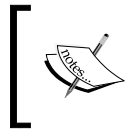
You need a mesh to be deformed by the bones you'll create. Open the file `001-Chains.blend` from this book's support files. It contains a tail-like mesh so you can follow this recipe to create separate chains, producing a scorpion-like movement.

How to do it...

1. Position the 3D cursor on the base of the tail, with a left mouse click in the 3D View, as seen in the next screenshot:

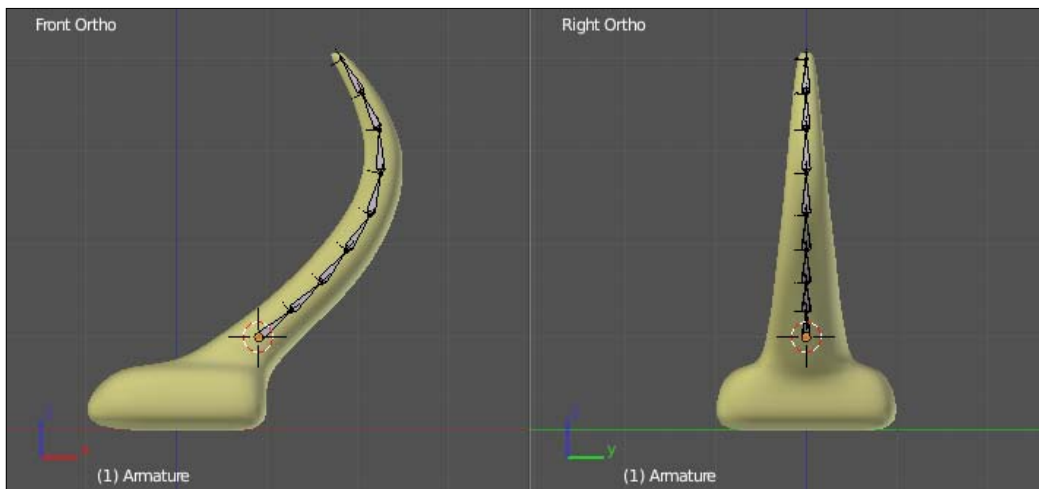


2. Press *Shift* + *A* and select **Armature | Single Bone** to create one bone which extends from the base to the tip of the tail. Enter in Edit Mode (press *Tab*), select its tip, and move (press *G*) it to the tip of the tail.



In order to better see the bones and their axes, go to the **Properties** window, under the **Object Data** panel, and enable the **X-Ray and Axes** properties.

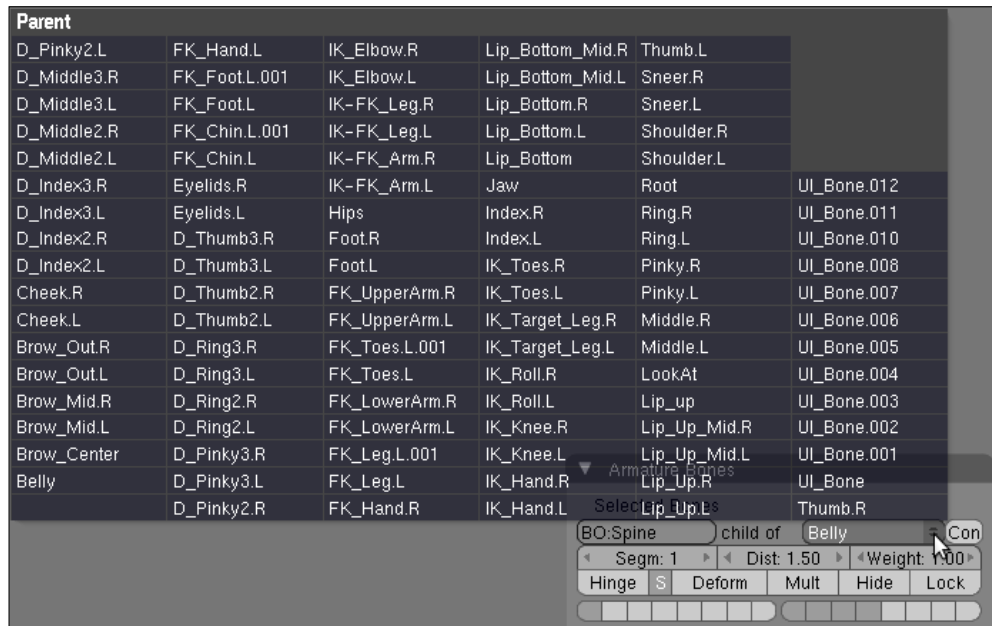
3. Select the bone, press **W**, and select **Subdivide**. Repeat this two more times in order to get eight bones. Select each joint and move (press **G**) it so it fits the tail nicely.
4. Select all the bones (the **A** key), press **Ctrl + R**, and type 90 so their orientation is set with their X axes pointing towards us. The X local axis will be the default to the front or back rotation.
5. Refer to the recipe called *Defining good orientations for your bones* if in doubt. You should end up with something similar to the next screenshot, showing the front and side views:



6. Still in Edit Mode, set the bones' names using the **Properties** panel (press the **N** key). For their names, use a prefix such as **D_**, which stands for "Deformation". That's the role of these bones: they're responsible for deforming our mesh. Good names can be **D_tail.1** to **D_tail.8**.



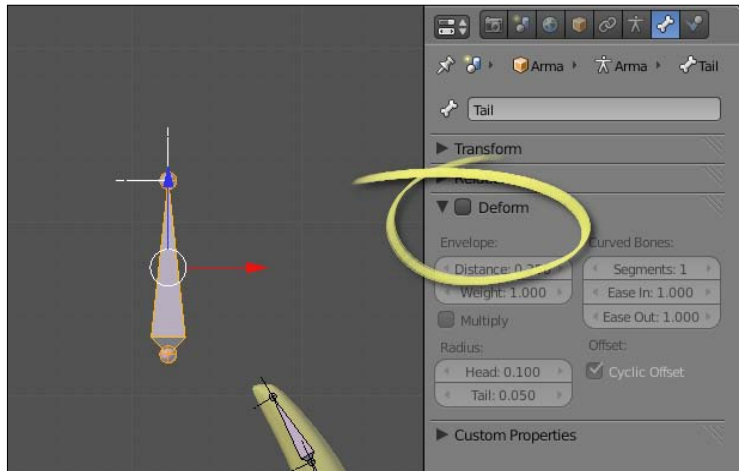
In Blender versions prior to 2.5, finding the name of a bone in the list displayed by the program could be a tough job, as seen in the next screenshot. Using prefixes are crucial to help you find the desired bone in a list and know its function without having to select it. With the arrival of Blender 2.5, finding a bone (or any object) by its name is much easier: just start typing in the appropriate field to narrow the selection options.



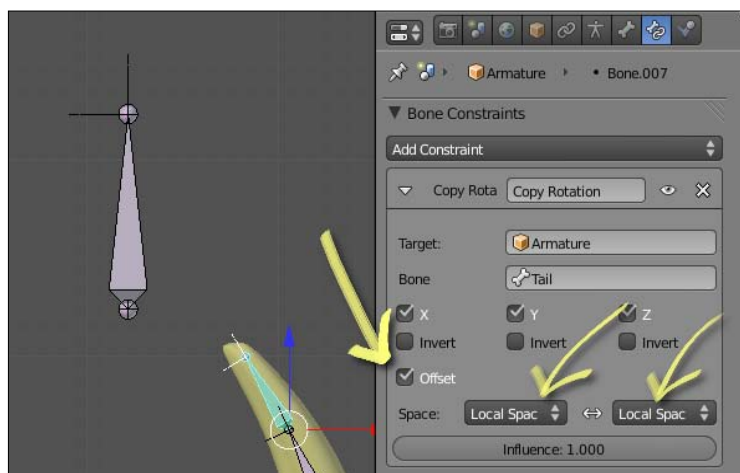
Now we're going to create the controller bone. This bone belongs to another "chain" of bones responsible for controlling the deformation ones. The controllers don't perform any mesh deformation by themselves. Although in this example this chain has just one bone for the sake of simplicity, more complex rigs can easily have dozens of them.

- Still in the armature's Edit Mode, place your cursor just above the tip of the tail and press **Shift + A** to add another bone. Press **Ctrl + R** and type 90 so that its orientation is the same as the deformation chain ones. Define this bone's name as **Tail**. The controller bones are usually named without prefixes in order to be friendlier to the animator, who will look out for **Tail** instead of **C_Tail**.

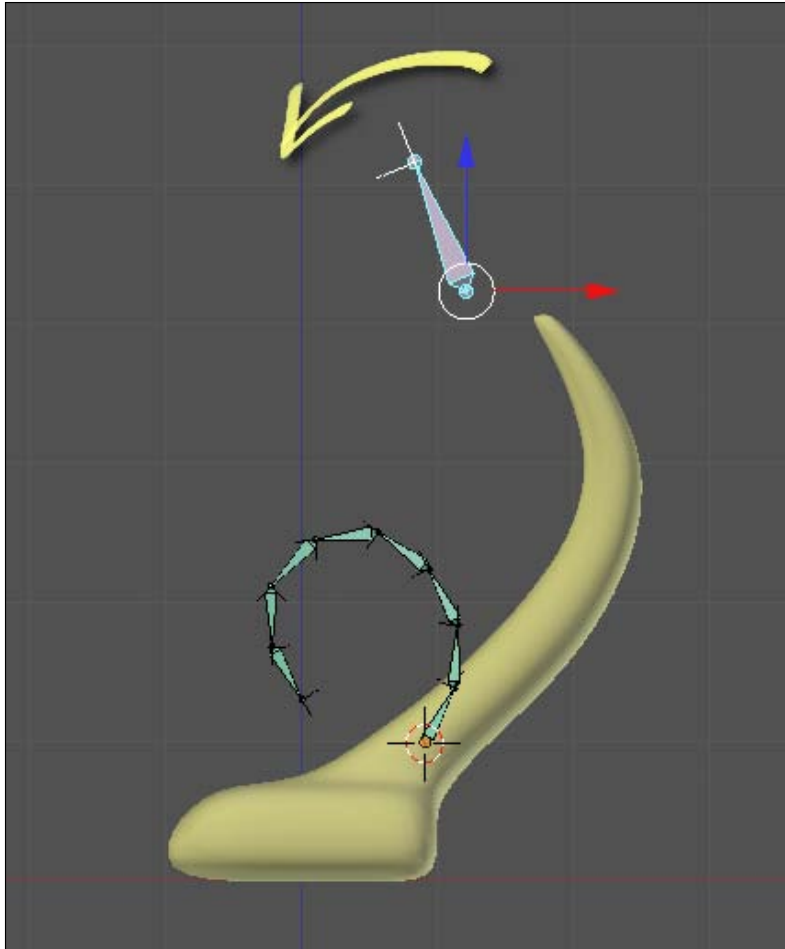
8. Disable the **Deform** option on the **Bones** tab in the **Properties** window, as seen in the next screenshot, so this bone will not perform any deformations on the mesh:



9. Now we're going to add constraints to control our deformation chain. Go to the armature's Pose Mode (press *Ctrl + Tab*). Then, select the bone Tail (which is the controller one), press *Shift* and then select the bone on the tip of the deformation chain, *D_tail.8*. Press *Ctrl + Shift + C* to bring up the **Constraints** menu and choose **Copy Rotation**.
10. This will make the bone on the deformation chain copy the rotations of the controller one, but you will notice that it will copy the absolute rotation (which is not what we want). To make this bone copy the transformation based on its own rotation, select it and go to the **Bone Constraints** panel, under the **Properties** window. Check the **Offset** option and select **Local Space** on the two drop-down lists, as seen in the next screenshot:



11. Apply the same constraint to all other bones of the deformation chain: select them and, lastly, the bone which has the constraint we want to copy. Go to the **Pose** menu on the window header and select **Constraints | Copy Constraints to Selected**. That will apply the same constraint to all bones of the deformation chain.
12. Still in Pose Mode, select the **Tail** bone and rotate it. You'll see that all the bones on the deformation chain follow its rotation like a real tail, as seen in the next screenshot:



13. Since the animators wouldn't need to see or move the bones on the Deformation chain, you should select and move (the **M** key) them to another (and invisible) bone layer. I usually move my deformation bones to the last layer, so you can do the same for yours. You should now also turn off the X-Ray option for this armature, since it's no longer needed.