

Introduction to Character Animation

Blender Summer of Documentation

Copyright © 2006 by Ryan Dale

BLENDER OPEN CONTENT LICENSE

Terms and Conditions for Copying, Distributing, and Modifying

Items other than copying, distributing, and modifying the Content with which this license was distributed (such as using, etc.) are outside the scope of this license.

1. You may copy and distribute exact replicas of the OpenContent (OC) as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the OC a copy of this License along with the OC. You may at your option charge a fee for the media and/or handling involved in creating a unique copy of the OC for use offline, you may at your option offer instructional support for the OC in exchange for a fee, or you may at your option offer warranty in exchange for a fee. You may not charge a fee for the OC itself. You may not charge a fee for the sole service of providing access to and/or use of the OC via a network (e.g. the Internet), whether it be via the world wide web, FTP, or any other method.
2. You may modify your copy or copies of the OpenContent or any portion of it, thus forming works based on the Content, and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
 - a) You must cause the modified content to carry prominent notices stating that you changed it, the exact nature and content of the changes, and the date of any change.
 - b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the OC or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License, unless otherwise permitted under applicable Fair Use law.
 - c) The name of the Copyright Holder or contributors to the OC may not be used to endorse or promote products derived from this work without specific prior written permission.

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the OC, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the OC, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. Exceptions are made to this requirement to release modified works free of charge under this license only in compliance with Fair Use law where applicable.

3. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to copy, distribute or modify the OC. These actions are prohibited by law if you do not accept this License. Therefore, by distributing or translating the OC, or by deriving works herefrom, you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or translating the OC.

NO WARRANTY

4. BECAUSE THE OPENCONTENT (OC) IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE OC, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE OC "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK OF USE OF THE OC IS WITH YOU. SHOULD THE OC PROVE FAULTY, INACCURATE, OR OTHERWISE UNACCEPTABLE YOU ASSUME THE COST OF ALL NECESSARY REPAIR OR CORRECTION.

5. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MIRROR AND/OR REDISTRIBUTE THE OC AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE OC, EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This PDF contains a snapshot of the Blender Summer of Documentation tutorial, "Introduction to Character Animation". This PDF was generated on 19 Sept 2006.

For the most current version (with corrections and clarifications, and links), please see the Blender wiki at:

http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation

Enjoy!
-Ryan

BSoD/Introduction to Character Animation

< BSoD

Contents

- 1 Welcome!
- 2 Feedback
- 3 Introductory material
- 4 Modeling
- 5 Materials and lighting
- 6 Rigging
- 7 Animation
- 8 Finished product

Welcome!

This tutorial is the result of about a month of intense work as part of the Blender Summer of Documentation. I created it with the hope of introducing many new users to Blender, as well as providing a resource for experienced Blender users.

Even if you've never heard of Blender before, you will be able to do this tutorial.

Enjoy!

-Ryan

Marenzelleria 05:46, 26 July 2006 (CEST)

Note: The review in the latest Blenderart (<http://www.blenderart.org/>) 6 magazine (*thanks Blenderart!*) refers to three sections of the tutorial that were referred to in my original outline:

- . envelopes vs. vertex groups
- . custom bone shapes
- . using the stride bone

If you're looking for these somewhat advanced topics, they aren't in the current version of the tutorial yet. They're on the way.

-Ryan

Feedback

Questions? Comments? Suggestions? Visit the support thread (<http://blenderartists.org/forum/showthread.php?t=74704>) at blenderartists.

Introductory material

- . Introduction: What to expect from this tutorial.

Modeling

- . Setting up the mesh - Start up Blender, add a plane, and set up symmetrical editing.
- . Creating the mouth - start shaping and extruding the mouth.
- . Face and eyes - Keep extruding the face, and add eyes.
- . Finishing the head - Finish off the head.
- . Neck shoulders and arms - Create the upper body by extruding neck, shoulders, and arms.
- . Torso legs and feet - Create the torso and lower body.

Materials and lighting

- . Lighting - Add lights to the scene, and do a test render.
- . Materials and textures - Add materials to the skin and eyes, and add textures to the skin.

Rigging

- . Upper body: building the armature - Build a skeleton for the upper body.
- . Upper body: Weight painting - Configure the skeleton for the upper body.
- . Lower body: building the armature and weight painting - Build and configure the skeleton for the lower body.
- . Final rig adjustment. - Make some final improvements to the rig.

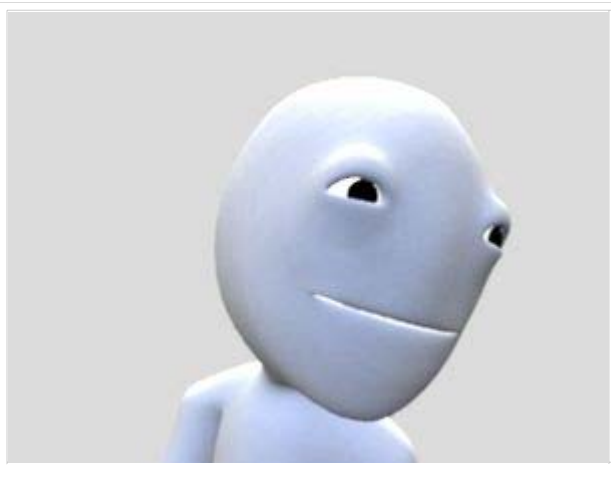
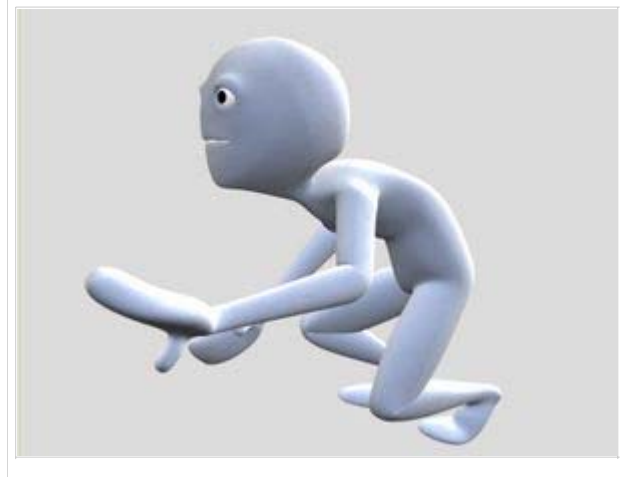
Animation

- . Animating a simple action - Animate a "Wave" action using the Action editor.
- . Animating a walkcycle - Animate a walkcycle using the Action editor.
- . Mixing actions with the NLA - Combine the wave action and the walkcycle action together in the NLA Editor.
- . Facial expressions - Create a set of facial expressions.
- . Adding sound and lip syncing - Use the facial expressions to lip sync to sound.
- . Final animation - Mix everything together into a final animation and render the output.

Summary - A recap of the tutorial, as it was first modified on 7th and 8th September 2006.

Finished product

Here are some static poses of the character you'll create in this tutorial. The end product of the tutorial is 5-second animation of the character, including a walkcycle and lip sync.



Ready? **Start with the Introduction . . .**

Categories: Introduction to Character Animation | Blender Summer of Documentation

BSoD/Introduction to Character Animation/Intro

From BlenderWiki

< BSoD | Introduction to Character Animation

Blender Summer of Documentation: Contents | Manual | Blender Version 2.42

Contents

- 1 About this character animation tutorial
 - 1.1 Why should I do this tutorial?
 - 1.2 How long will it take?
 - 1.3 Isn't this information available elsewhere?
- 2 Format of the tutorial
 - 2.1 Main body
 - 2.2 Side notes
 - 2.3 Keyboard commands, mouse controls, and menu items
 - 2.4 Notes
 - 2.5 Downloadable files
- 3 Prerequisites
- 4 Caveats

About this character animation tutorial

This tutorial is designed to teach you the more advanced tools available in Blender. It's written in the spirit of "Gus the Gingerbread Man" tutorial: no prior knowledge is assumed. In other words you'll be able to follow this tutorial without any prior Blender experience.

Why should I do this tutorial?

Upon completion, you will have a fully rigged character (with facial expressions and a skeleton) and an animation - lip synched to sound, starring your character. You'll end up with the tools and knowledge to make your own character and make it do whatever you want. It's your one-stop-shop for learning many of the tools in Blender.

How long will it take?

Depending on your prior experience and how quickly you work, it could take anywhere from a single day to a month. Take your time and don't worry about finishing it within some time limit.

It's not about the destination, it's about the journey: don't try to speed through this tutorial. Take the time to learn Blender -- it will pay back in the long run, when you'll be able to finish your own projects quickly and efficiently.

Isn't this information available elsewhere?

It sure is! I've repeated things that are found throughout the Blender manual, in other tutorials, and around the web. If I've used someone else's work for reference, I've cited that in the text. The advantage of this tutorial is that it compiles all of that knowledge into a single step-by-step guide.

Format of the tutorial

Main body

The main body of the tutorial consists of steps to carry out and accompanying screenshots. Individual steps are indicated by bulleted items.

- Step one, with some explanatory text
- Step two
- Step three, and why we did it that way.

If a screenshot is too small, you can always click on it for a larger view. In the text, I've tried to explain *why* you're doing a particular step in as well as how to do it.

Side notes

To accomodate the widest range of skills, I've tried to assume as little as possible on the part of the reader. To prevent beginning users from becoming lost and more advanced users from getting bored, I will introduce new topics and concepts as side notes that beginners can read and advanced users can easily skip over.

Keyboard commands, mouse controls, and menu items

At the beginning of the tutorial, I've tried to include the hotkey or menu item for each command. It's easy enough to skip over them if you know them already, but they'll be helpful if you take a break and come back later. You won't have to page back looking for the hotkey if you forget.

Later on in the tutorial, I won't spell out often-used commands, since you'll have already learned them.

Notes

Periodically you'll see a note like this:

A note on notes:

Notes like this one will warn you of common pitfalls, explain why some steps were taken, and to periodically remind you to save

About side notes

This is a side note, which will introduce new commands or concepts. It is a way of separating out explanatory material from the step-by-step format of main body of the tutorial.

These side notes will contain a brief explanation, relevant hotkeys or menu items, and a link where you can find more information.

Downloadable files

At the end of each major section, you can find a .blend file to download. If you get stuck, or want to skip some sections, feel free to download the file and start from that point.

Prerequisites

I've assumed little to no prior knowledge of Blender. As such, I take the time at the beginning of the tutorial to explain

common tools and commands. An experienced user can cruise right through the first few sections quickly, while a beginning user should take the time to read the explanations.

As supplementary material, check out the one-page Blender QuickStart Guide.

Don't worry about memorizing all those commands. At each step throughout the tutorial, I'll remind you of the hotkeys or menu commands. With enough practice, you'll memorize the commands that are worth memorizing without even trying.

Caveats

Please keep in mind that this is only one way of doing things. There are many different work flows, and I'm only showing you the way I do it.

I've tried to strike a balance between keeping the character simple enough so that a beginner can follow along, but complex enough so that you can learn more advanced techniques. Feel free to add more detail, make more complex materials and textures, or make a more complex rig. Consider this tutorial as a guideline for your own creations.

Next: Setting up the mesh

Previous: Index

Back to Index

Retrieved from "http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Intro"

- This page was last modified 08:49, 15 September 2006.

BSoD/Introduction to Character Animation/Setting up the mesh

From BlenderWiki

< BSoD | Introduction to Character Animation

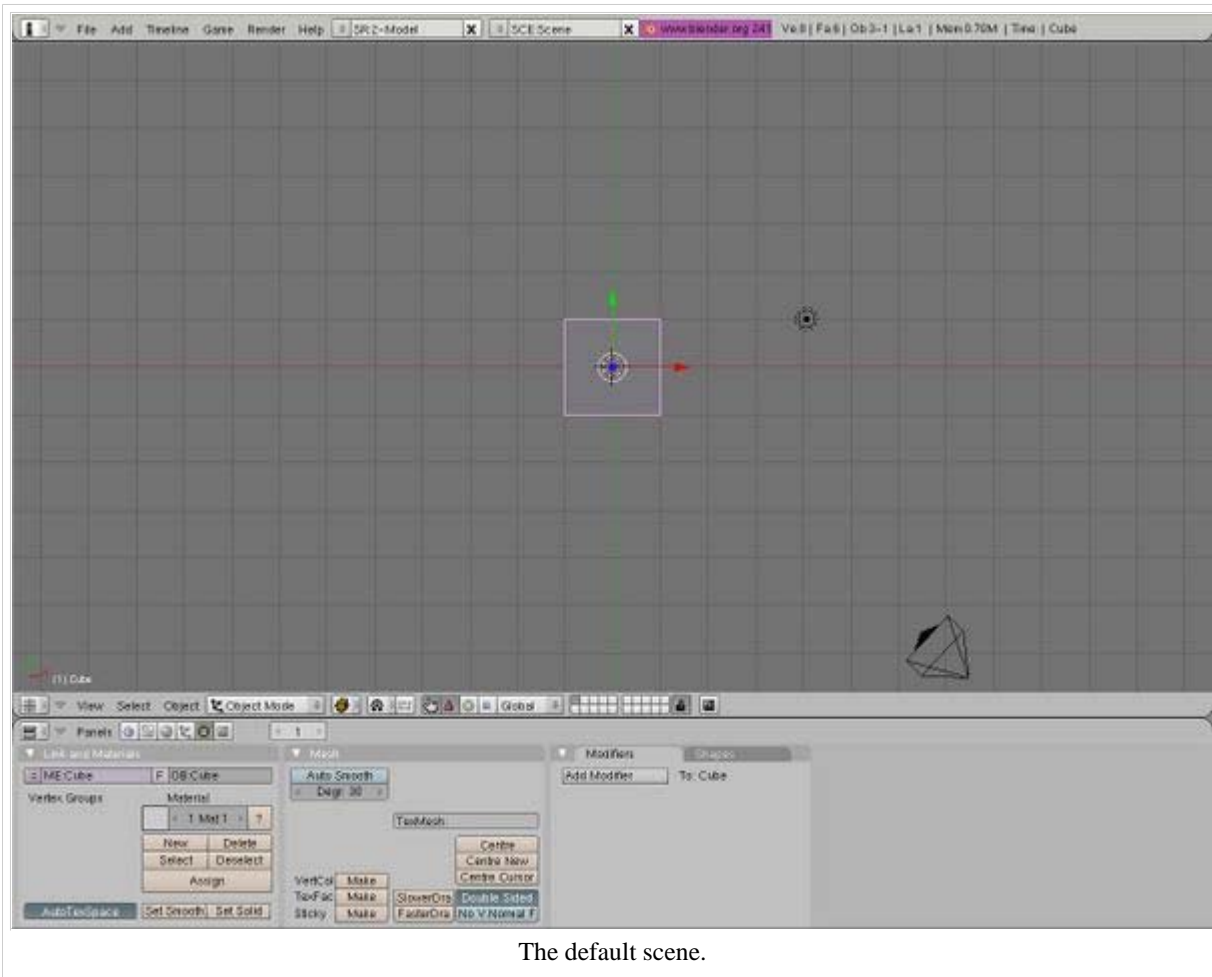
Contents

- 1 Welcome to Blender!
- 2 Vital functions
 - 2.1 Selecting
 - 2.2 Undo
 - 2.3 Saving
 - 2.4 Loading
- 3 A fresh start
- 4 Add a plane
- 5 Mirror the plane

Welcome to Blender!

If you haven't already, please take the time to read the Introduction for important information regarding this tutorial.

You can download Blender from www.blender.org (<http://www.blender.org>) . Blender is available for many operating systems: Windows, OSX, Linux, Solaris, and FreeBSD. Instructions for installation also an be found there. Once it's installed and started up, you'll see something like this:



The default scene.

In the default screen, moving from the top down, there is:





1. A top **header** containing the File menu, Add menu, and others
2. The **3D Window** with some objects
3. The header for the 3D Window
4. The header for the Buttons Window
5. The Buttons Window

Note: When you start Blender, a console window is opened. This is useful for feedback when doing more advanced operations. For now, you can safely ignore this window. Don't close it, though, or it will close the other Blender window as well.

Let's get started!

Vital functions








Note: A note on abbreviations used in this tutorial:

- RMB  stands for "right mouse button".
- LMB  stands for "left mouse button".
- MMB  stands for "middle mouse button".
- MW  stands for "mouse wheel".
- NumPad 3 refers to the "3" key on the NumPad.
- 3 refers to the "3" key on the top of the keyboard.

Keyboard commands like G, Ctrl Alt R, Ctrl I should be self-explanatory.

Selecting

In Blender, the most frequent action you perform is making a selection. So, there are many different ways to make a selection:

- RMB  - make a single selection.
- Shift RMB  - add to existing selection.
- B - bounding box select. Draw a rectangle with the mouse, then use LMB  to confirm or RMB  to cancel.
- B B (hit it twice) brush select. Use LMB  to "paint" over vertices you want to select, MMB  to deselect, or RMB  to cancel.
- A - select/deselect all

More info: Manual/Selection

Undo


It's probably a good idea to let you know early on that, like many programs, Blender has the ability to undo:

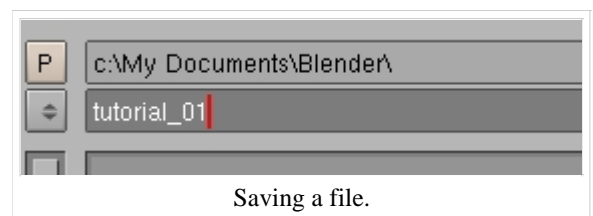
- Press Ctrl Z to **undo**. Everything we just deleted is returned to the scene.
- Press Ctrl Y to **redo**. Back to the fresh scene!

By default, Blender has 32 undo steps, so you can keep pressing Ctrl Z to do multiple undos. Or, for more control, you can use Alt U for a menu that acts much like the History in Photoshop where you can select which step to undo back to.

Saving

Now is a good time to **save** your file. Blender's file sizes are not very large, so you can save often.

- Press F2 to save. The 3D Window turns into a big Save dialog box. Navigate by clicking on the "P" button to move up one directory, or by clicking on directory names which are in white text.
- LMB  on the file name text box ("untitled.blend" by default) to be able to edit the text (**Saving a file**)
- Type in a file name. It's a good idea to use a number at the end, because Blender will auto-increment the filename for you



(see below).



- Press *Save file* to save. You are returned to the 3D Window.

To illustrate the useful feature of auto-incrementing (I wish more programs had this!),

- Press **F2** again for the Save dialog
- Press **+** to increment the filename. If it was "Tutorial_01", it will now be "Tutorial_02".
- Click *Save file* or hit **Enter** to save.

Using this feature, you can quickly save a file with the combination **F2** - **+** - **Enter**.

Loading

- Show the Load dialog with **F1**
- MMB  on the file you want to open, or select it with LMB  and click *Load file*.

A fresh start

The default scene has a cube, a camera, and a lamp. Let's delete all that and start from scratch.

- Since the default scene has the cube already selected press **A** to deselect all, then **A** again to select all objects.
- Press **X** to delete selected objects. Ah, a fresh scene!

Add a plane

Now we will add a **Mesh** object (a Mesh is the basis for most 3D models). Since we're modeling a character, it will be useful to only have to create one side, and automatically create the other side. To do this, we'll **Mirror** the mesh.

- Change to Front View (press NumPad **1** in the 3D Window). Note that this is different than the **1** key in the row of numbers at the top of the keyboard. We will use the NumPad keys frequently, so If you have a keyboard without a separate NumPad (for example many laptops lack a separate NumPad), then follow these quick instructions to reassign the top row of numbers to act like the NumPad numbers.

NumPad Views

NumPad **1** - Front View
NumPad **3** - Side View
NumPad **7** - Top View
NumPad **0** - Camera
NumPad **5** -
Perspective/Orthogonal

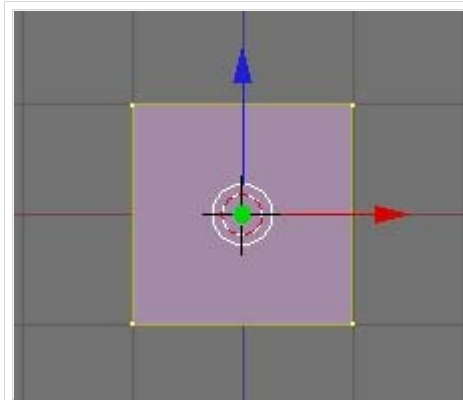
More info:

[Manual/PartI/Navigating_in_3D_Space](#)

- Add a plane (**SPACE** >> *Add* >> *Mesh* >> *Plane*). Two things determine the position and orientation of an object when you add it: 1) the **3D Cursor** (the red-and-white circle that acts as a reference point) and 2) your current view. We switched to Front View so that the plane would not be crooked when we added it. We could have added any one of those mesh types, but a plane is the simplest and most straightforward to work with.



Add a plane.



The new plane.

- When the plane was added, Blender starts us out in **Edit Mode**. Edit Mode is where you can edit the vertices (the yellow and pink dots) that make up a mesh.

Object vs Edit Modes

Use **Object Mode** any time you want to do something to the *whole object* at once, like moving it around in the scene.

Use **Edit Mode** any time you want to do something to *part* of a mesh, like editing vertices. There are many options in this mode, for more info see [Manual/PartII/Edit_Mode](#).

Switch between the two with **TAB**.


Note: For the remainder of the tutorial, I have turned off the **Transform Widget** (the thing with the three colored arrows). The Transform Widget is a graphical way of moving objects, but I prefer to use the much quicker hotkeys (more on these later). Turning off the Widget also makes these screenshots less cluttered. To turn off the Transform Widget, press the button with the pointing hand on the bottom of the 3D Window

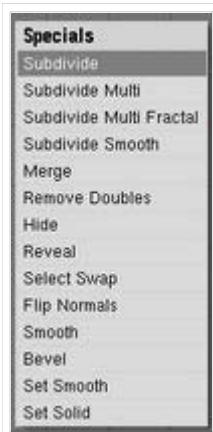


Turn off the Transform Widget by pressing the pointing hand button.

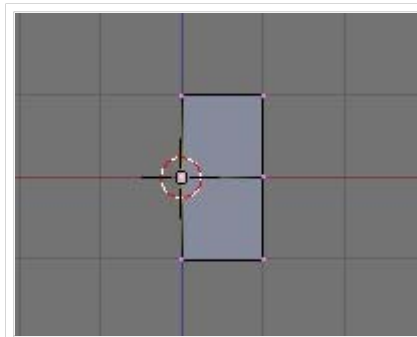
- Subdivide the plane once (**W** >> *Subdivide*, as in *Subdividing the plane*). Subdividing adds vertices to the mesh.

Then delete the left half of the plane:

- **A** to deselect all
- **B** to switch the mouse cursor to border select mode. Drag a rectangle around the vertices all the way on the left with **RMB**  to border-select them.
- **X** to delete the vertices. If we didn't subdivide, we'd only be left with two vertices now.
- You should have something like *Half a subdivided plane*.




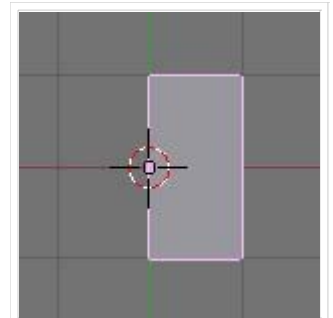
Subdividing the plane.



Half of a once-subdivided plane.

Mirror the plane

- Press **TAB** to switch from Edit Mode to **Object Mode**. The plane is outlined in pink, showing that it is currently selected (*The half-plane in Object Mode*).
- Go to the Edit Buttons (Press the  context button in the Buttons Window, or press **F9**).



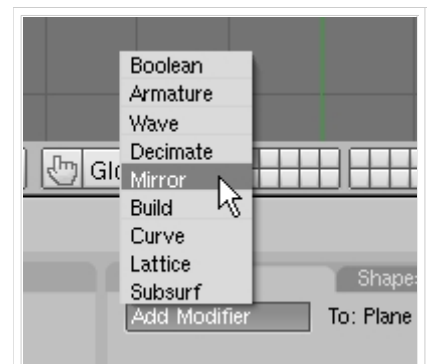
The half-plane in Object Mode.

Note: There are too many buttons to show all at once, so the Buttons Window is divided into sections. You can access these different sections by pressing the different **Context Buttons** along the top of the Buttons Window.



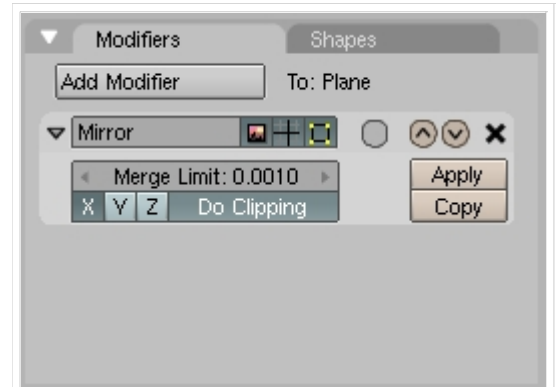
Context Buttons in the Buttons Window. Here, the Edit Buttons selected.

- In the Edit Buttons, find the *Modifier* tab. Click *Add Modifier* (*Add a Mirror modifier*).
- Select *Mirror*.



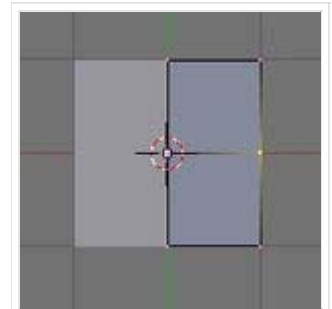
Add a Mirror modifier.

- In the Mirror modifier, make sure *X* is selected
- **Important:** Make sure to click *Do Clipping*, as in *The Mirror modifier*. This will prevent vertices from crossing the mirror axis.
- DO NOT hit *Apply*. We'll apply the effects of the Mirror modifier later when we're all done modeling.



The *Mirror* modifier, applied to the half-plane. Note that *Do Clipping* is selected.

- Go back into Edit Mode (`TAB`). Your mirrored plane should look like *Mirrored plane*.



Mirrored plane.

Summary:

We created a simple mesh, deleted half of it, and then mirrored it. Now we can model only one half of the character, and the other half will automatically update.

Next: Creating the mouth

Previous: Introduction

Back to Index

Retrieved from

["http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Setting_up_the_mesh"](http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Setting_up_the_mesh)

- This page was last modified 17:08, 19 August 2006.

BSoD/Introduction to Character Animation/Creating the mouth

From BlenderWiki



< BSoD | Introduction to Character Animation

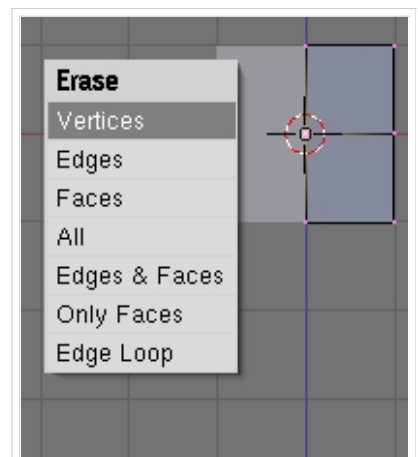
Contents

- 1 Shaping the outline of the mouth
- 2 Transforms
 - 2.1 Grab
 - 2.2 Rotate
 - 2.3 Scale
 - 2.4 Standard transform
 - 2.5 Modify a transform
- 3 Extruding the mouth
- 4 Loop cutting the mouth
- 5 Manipulating the 3D View
 - 5.1 Centering the view
- 6 Shaping the mouth
- 7 Set smooth and recalculating normals
- 8 Adding a subsurf modifier
- 9 Controlling Modifiers in Edit Mode

Shaping the outline of the mouth

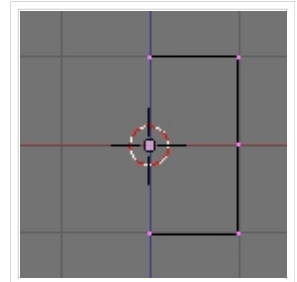
This plane will become the edge of the mouth. We'll extrude it out and add some more vertices. But first, we need to make a hole in the plane.

- In Edit Mode, delete the center vertex (RMB  to select, X to delete). It's tough to see, since the Object Center of the plane (indicated by the filled pink circle) is directly on top of it. Click RMB  on top of the Object Center, and the center vertex behind it should become selected.





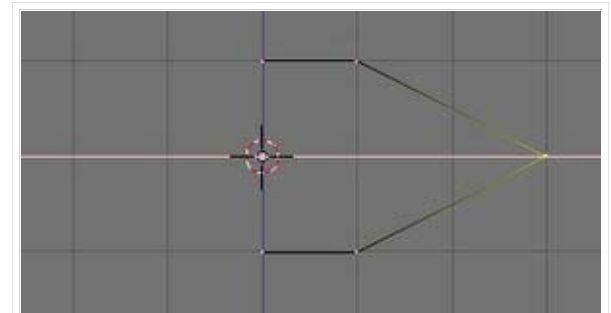
Deleting the center vertex (which is hidden behind the Object Center marker).

It looks like the plane is no longer mirrored. In fact, it still is, you just can't see it. By deleting that center vertex, we've also deleted the **faces** that contained the deleted vertex. It turns out that in our simple plane, all the faces contained the center vertex, so all the faces were deleted. Note that the **edges** around the outside are still there. In the following steps, we'll be making more faces from those outside edges.



Mirrored plane with center vertex deleted.

- Select the middle vertex on the right side with RMB .
- Move the middle-right vertex out 2 units or so in the X direction:
 - **G** to enter **Grab** mode
 - **X** to constrain the motion in the X direction
 - **2** to move 2 units to the right, as in *Moving the middle vertex*. In this way, you can type in exact numbers for any particular transform.
 - LMB  (or **Enter**) to confirm.



Moving middle vertex.

Transforms

Moving the vertex was the first transform we've done so far, and it was a grab (also called move or translate) transform. The following transform commands are **EXTREMELY IMPORTANT**:

Grab

G to grab.

Rotate


R to rotate.

Scale

S to scale.

These transforms all work pretty much the same way:



Standard transform

- Enter a transform mode by using one of the following:
 - **G** for grab
 - **R** for rotate
 - **S** for scale
- Move the mouse
- LMB  to confirm

- RMB  or **ESC** to cancel

Modify a transform


You can modify a transform while in transform mode, usually to either to constrain the transform along an axis or to make the transform easier to make. While in transform mode, use one of these modifiers:

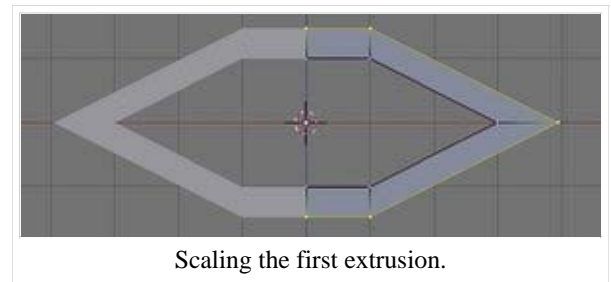
- **X** to constrain to the X axis
- **Y** to constrain to the Y axis
- **Z** to constrain to the Z axis
- Hold **MMB** , move the mouse to highlight the axis you want to transform along, let go of **MMB**  and continue the transform.
- Hold **Ctrl** to snap the transform to the grid.
- Hold **Shift** to slow down the transform for fine control

Note: For some transforms it matters where the mouse is when you start the transform.

Extruding the mouth

Now let's form more of the mouth.

- Select all vertices in the mesh with **A**.
- Enter **Extrude** mode (**E**>>Only Edges).
- Extrude Mode automatically puts you in Grab mode. Move the mouse around to see what Extrude does.
- Switch to **Scale** mode by pressing **S**. **Note:** It matters where the mouse cursor is when you press **S**.
- Scale up the extruded vertices by moving the mouse away from the Object Center so it looks something like *Scaling the first extrusion*
- **LMB**  to confirm the scaling.



This was the first extrusion we've done so far. Along with selecting and transforming, extruding is one of the most common actions when modeling in 3D. When you enter Extrude mode, new vertices are created directly on top of the vertices you had selected to extrude, and Grab mode is activated. Here, we didn't want Grab mode so we switched to Scale mode instead. A useful thing to remember is that the newly extruded vertices remain selected when you exit Extrude mode.


Extruding


E to enter Extrude Mode

Common menu choices are "Only Edges" or "Region".

Grab mode is automatically entered. *Optional: use **S** to switch to Scale mode or **R** to switch to Rotate mode).*

LMB  to confirm the extrude.


RMB  to cancel the extrude.

Important!: Extruding creates vertices on top of the vertices you selected to extrude, **even if you cancel the extrude** with **RMB** . This can be a source of trouble for new users.

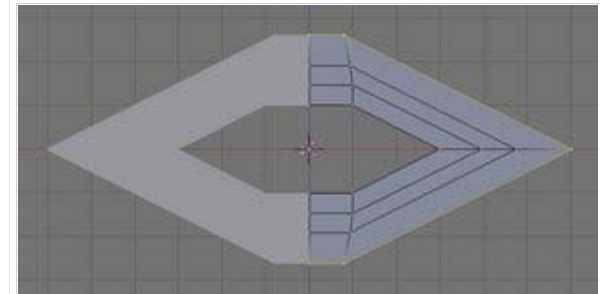
If you want to get rid of those extra vertices, undo the extrude with **Ctrl** **Z**. If it's been a while since you extruded and Undo won't work, use **W**>>*Remove Doubles*. This merges all

vertices that are directly on top of each other.

The newly extruded vertices remain selected after you exit Extrude mode.

- Repeat the extrusion and scaling two more times (for a total of three) by using:
 - **E** >> Only Edges to extrude
 - **S** to scale
 - LMB  to confirm.

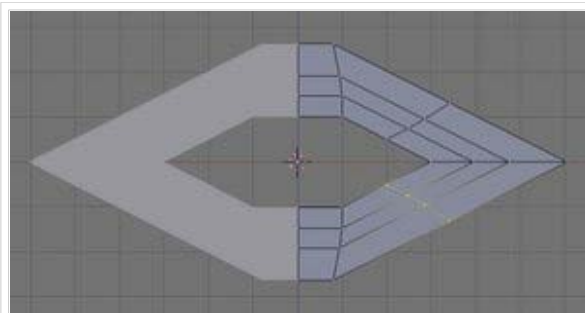
The result should look something like *After two more extrusions*.



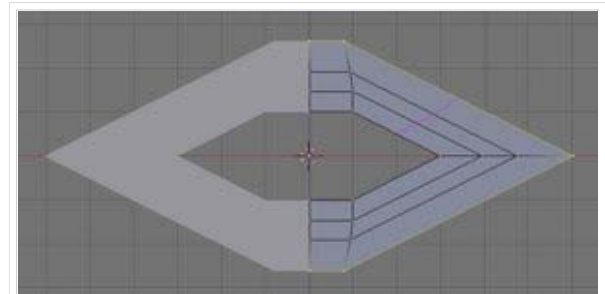
After two more extrusions.

Loop cutting the mouth

The Extrude tool, which we just used, is typically used to create additional vertices while expanding the mesh. There's a different type tool to use if





After the second loop cut. Yellow vertices were just created from the loop cut.




Setting up the first loop cut. Note purple line indicating which edges will be cut.

we want to add more vertices but don't want to expand the mesh any more: the Loop Cut. To make a **Loop Cut**,

- Enter Loop Cut mode with **Ctrl R**
- You'll see a purple line which shows approximately where the cut will be made.
- Move the mouse around until you see the purple line as in *Setting up the first loop cut*.
- LMB  to confirm the selected loop.
- Move the mouse around to see that you can place the new cut, indicated by the sliding yellow vertices, anywhere you'd like. However, we want the cut to be exactly in the middle of the loop.
- MMB  to make the cut exactly in the middle of the loop.
- Make another loop cut on the bottom as well, to end up with something like *After the second loop cut*.


Manipulating the 3D View


Important: Use MMB  to rotate the view around and get just the right angle.



When you want to be precise about moving some vertices, switch to one of the NumPad views (NumPad **1**, NumPad **3**, NumPad **7**) and move the vertices from that view. If you are in Front view, for example, when you move the vertices, you will ONLY be able to move them left/right and up/down . . . NOT forward/back. Similarly, in Side view, you can only move forward/back and up/down . . . not left/right.

Centering the view

Sometimes when you rotate the view with MMB , it seems like you're rotating around the wrong center, and this can get frustrating. There's an easy way to fix this:



- Move the 3D Cursor to where you want the view to be centered by clicking LMB .
- Center the view on the 3D Cursor with **C**. Now the view will be rotated around the 3D Cursor.

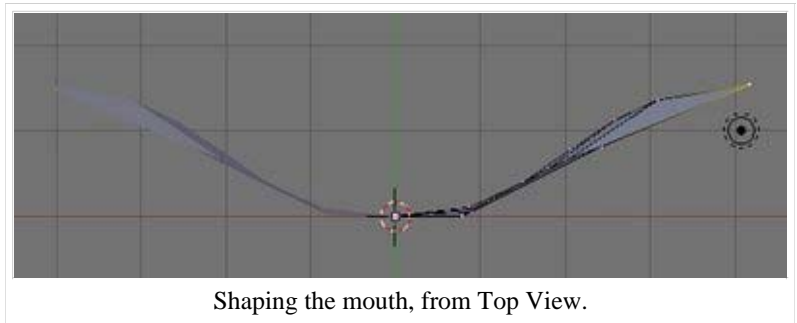
And another way, if you don't want to move the 3D Cursor:

- Make a selection
- Press NumPad **.** to center the view on the current selection.


Shaping the mouth

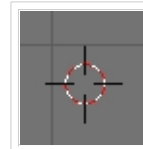
Let's give the mouth some shape.

- In Front View (NumPad **1**), select the right-most vertex
- **O** to enable **proportional editing**. Proportional editing transforms nearby vertices even though they're not selected. After starting a transform (grab, rotate, or scale), the mouse cursor will turn into a circle to outline the sphere of influence. You can use MW  to adjust the size of the sphere of influence while transforming.
- Switch to top view (NumPad **7**).
- Move the vertex up and in a little, using **G**, to look something like *Shaping the mouth*. I ended up making the sphere of influence pretty large with MW  (so the outer edge of the circle was almost touching the Object Center) to form this mouth shape in a single move.



The 3D cursor is a handy tool. It acts as a reference point for transforms and determines where new objects are placed.

LMB  to position the cursor
Shift **S** for the Snap menu



The 3D-cursor.


More info: [Manual/PartII/3DCursor](#)

If everything disappears . . .: If you press a number key at the top of the keyboard by mistake and everything disappears, press the **`** key (the one next to the 1 key, it also has a ~ on it) to get it back.

This happened because the numbers at the top of the keyboard let you view individual layers.

The Plane was added to Layer 1 by default and there's nothing

Proportional Editing

O to toggle proportional editing
MW  to change the influence
(or Alt NumPad **+** and
Alt NumPad **-** after **G**, **R** or

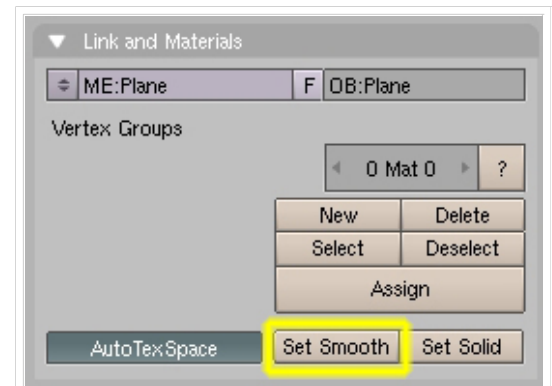
on Layer 7. So if you press **7**, it shows just Layer 7 and it seems like the plane disappeared. Just press the **~** key to show all layers at once.

S)

More info:
Manual/PartII/Advanced_Mesh_Model

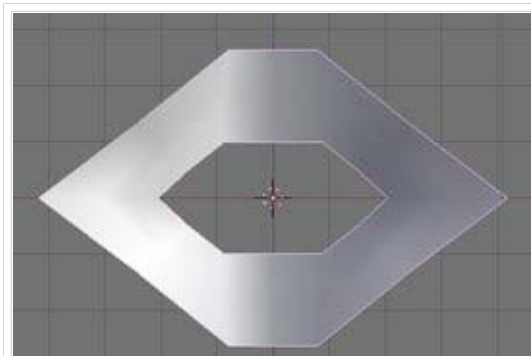
Set smooth and recalculating normals

- **TAB** to switch from Edit Mode to Object Mode. See how the mouth is sort of blocky? Let's change that.
- The mouth should still be selected.
- Find the *Set Smooth* button in the Edit Buttons (Buttons Window, Edit context . . . or **F9** as a shortcut) as in *Set Smooth button*.

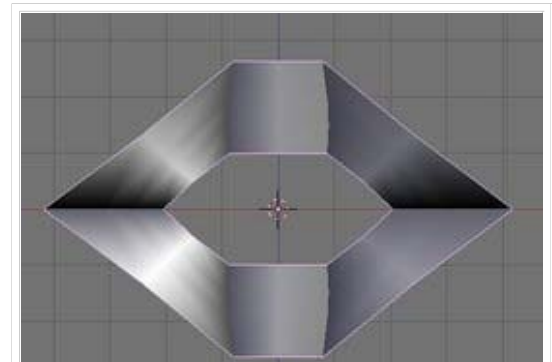


Set smooth button, in the Edit buttons.

See those ugly black lines in *Wrong normals*? Sometimes this happens when you do several extrudes. You can read about the details of why it happens, here:



Correct normals (after recalculating normals).



Wrong normals. To fix this, select all vertices in Edit Mode and hit **Ctrl N** to recalculate normals.

Manual/PartII/Subsurfaces.

Here's how to fix it:

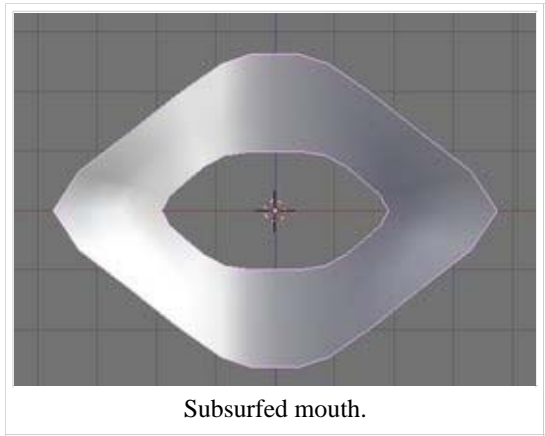
- **TAB** to switch to Edit Mode.
- **A** to select all vertices.
- **Ctrl N** to *Recalculate Normals*.
- **TAB** to get back to Object Mode.
- The result should look like *Correct normals*.

Adding a subsurf modifier

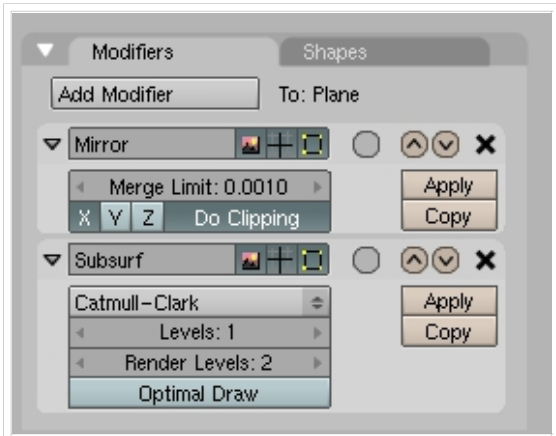
The corners of the mouth are still sort of sharp. One way to smooth it out would be to add many more vertices to round out the corners. There's another, better way: it's called

Subdivision surfacing, or Subsurf for short.

Subsurf is a fancy way of getting a smooth-looking object from a relatively coarse base mesh. It makes your model look better without needing lots of vertices. Luckily, it's quite easy to do in Blender.



Subsurf'ed mouth.

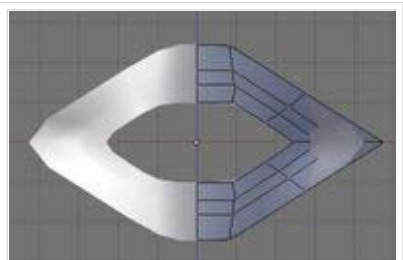


Modifier stack, with the *Subsurf* modifier underneath the *Mirror* modifier.

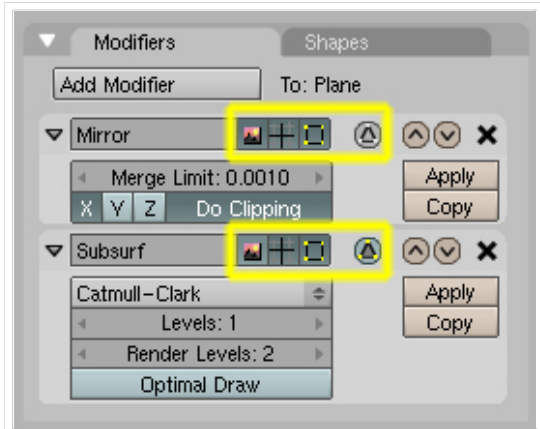
- With the mesh still selected, add a *Subsurf* modifier (*Modifier stack*) with the default settings.
- The result should look something like *Subsurf'ed mouth*.
- You can make the mouth look even smoother by increasing the *Levels* under the Subsurf Modifier. It's a tradeoff, though: Subsurf takes computing power. Setting the Levels too high will slow down your computer. It won't be an issue for a mesh this simple, but you will notice a difference with more complex meshes.
- You can learn more about Subsurf here: [Manual/PartII/Subsurfaces](#)

Controlling Modifiers in Edit Mode

Each time you create a modifier, it's added to the *modifier stack*. Each modifier applied to a mesh can be seen in this stack (see *Modifier stack*). Currently, this mesh has two modifiers: a *Mirror* modifier and a *Subsurf* modifier. Take a look at the icons next to the modifier name (outline in yellow, *Modifier details*).

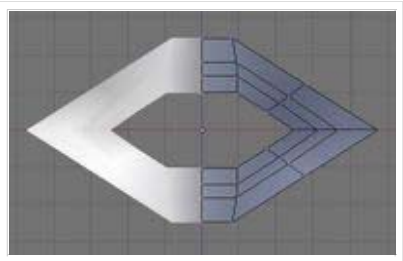


Subsurf **on** in Edit Mode (Default).



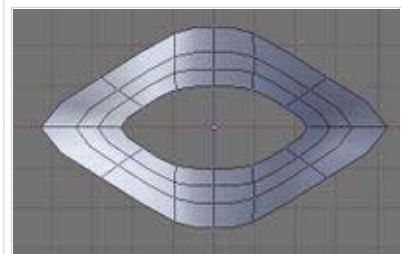
Modifier details.

Check out the right-most of the three buttons ("*Enable modifier during editmode*") and the gray circle to the right of the three buttons ("*Apply modifier to editing cage during Editmode*"). These buttons are great for tweaking a mesh. They turn a Modifier on and off when you're in Edit Mode. Try this:



Subsurf **off** in Edit Mode.

- Switch to Edit Mode (**TAB**)
- By default, Subsurf is turned on in Edit Mode. Note how from Front View, some vertices are hidden under the subsurface -- they don't follow the surface.



This is just something to be aware of when using Subsurf in Edit Mode. Think of the vertices as a cage, and the subsurface like a sheet tossed over the cage. By moving the cage, you'll move the sheet.

- Turn Subsurf off by clicking the right-most button in the Modifier (the one that says "Enable modifier during Editmode" when you hover the mouse over it). Sometimes it's useful to turn Subsurf off temporarily while you're modeling.
- Turn Subsurf back on with that same button.
- Now click the gray circle next to the buttons ("Apply modifier to editing cage"). Now, all the vertices are on the subsurface. It's no longer a cage with a sheet draped over it, now we can move the subsurface directly.
- Click the gray circle again to remove the Subsurf modifier from the editing cage. Now we're back to the default view.

Which view you use is a matter of personal preference. I tend to switch a lot between them as I'm modeling. Sometimes one view is better than another for certain circumstances, which we'll see later.

Don't forget to save a version with `F2` - NumPad `+` - Enter !

Summary: Great! You've learned the primary tools for modeling: Grab, Scale, Rotate, and Extrude. These skills will be very important in the next part of the tutorial.

We started the mouth and then smoothed it using Set Smooth and a Subsurf Modifier, and saw how to apply the modifier to Edit Mode.

Next:Face and eyes

Previous: Setting up the mesh

Back to Index

Retrieved from

["http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Creating_the_mouth"](http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Creating_the_mouth)

- This page was last modified 16:06, 14 September 2006.

BSoD/Introduction to Character Animation/Face and eyes

From BlenderWiki


< BSoD | Introduction to Character Animation


Contents

- 1 Extruding the rest of the face
- 2 Making room for the eyes
- 3 Adding an eye
- 4 Duplicating and Mirroring the eye
- 5 Move the eyes to a different layer
- 6 Adjusting the axes of the eyes
- 7 Forming the eye socket
- 8 Why don't we want too many vertices?

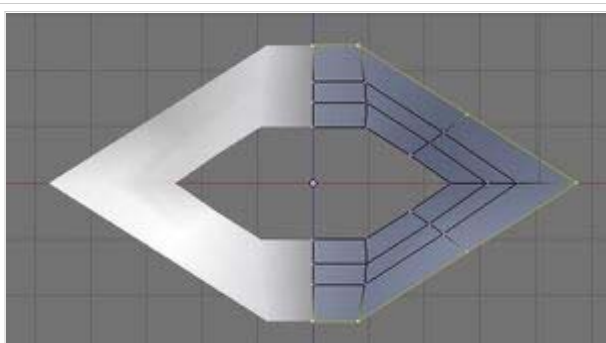
Extruding the rest of the face

Now that the mouth is mirrored, smoothed, and subsurfed, we'll make the rest of the face. This will basically consist of extruding, scaling, and shaping. For now, I'm going to be modeling the face with *Subsurf off* in Edit Mode.

- First, use **Alt** RMB  on the outer edge loop. This selects all the vertices in the edge loop.


If loop select doesn't work in GNU/Linux using **Alt RMB **


: If you're using Gnome, default configuration, you'll have to use one of the other selection methods instead. Alternately, some window managers will ignore the Alt-click if another modifier is pressed; try holding both the Windows and Alt key (with a right click) at the same time.



Outer loop selected.

Edge-loop and face-loop selection

Alt RMB  to select an edge loop
SHIFT-ALT-RMB to add edge loop to selection



Ctrl **Alt** RMB  to select a face loop

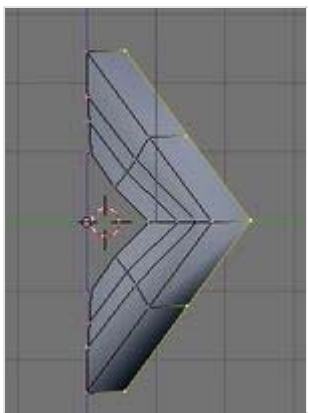
Shift **Ctrl** **Alt** RMB  to add face loop to selection

More info:

Manual/PartII/Edge_and_Face_Tools

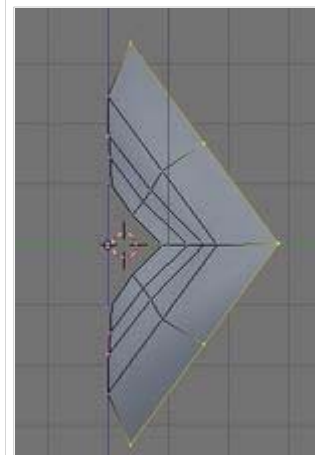
Now let's extrude that edge loop.

- NumPad **3** for Side View.
- **O** to turn off proportional editing for now.
- **E** to extrude
- Move the new vertices back a little (*Extrude edgeloop back . . .*)
- LMB  to confirm the move
- With the just-extruded vertices still selected, press **S** to scale the new loop up, LMB  to confirm (*. . . and scale it up*).



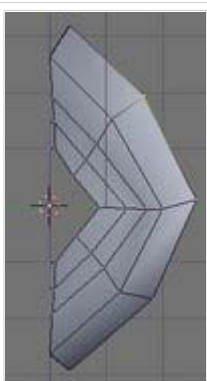
Extrude the edgeloop back .

..

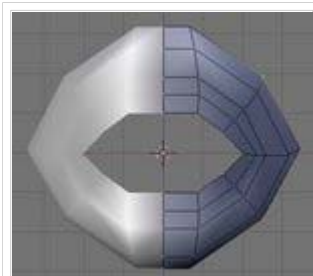


. . . and then scale it up.

- Enable proportional editing again with **O**, and give the face some shape by moving around some vertices with **G**. Use both Front View (NumPad **1**) and Side View (NumPad **3**) to shape the face (see two images below).




Extrude #1, after
shaping




Extrude #1, after shaping

Tips for shaping meshes: A couple tips to help you on your way . . .

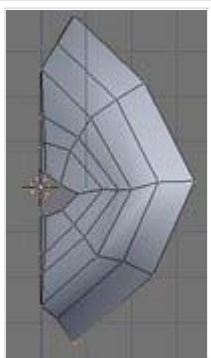
- Proportional editing (**O**) is great for shaping meshes. Don't forget to use MW  to change the size of the

sphere of influence. Every transform you make may need a different size sphere of influence. I tend to switch proportional editing on and off all the time, and I'm always changing the size of its influence.

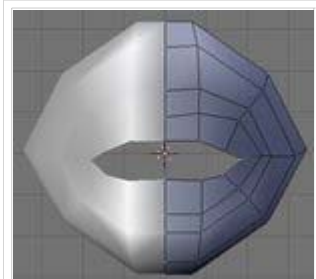
- The images below took a couple minutes of tweaking, with maybe 50 or so transforms (mostly **G**).
- When shaping like this, it's easiest to make liberal use of **MMB**  to get the right angle.
- **Ctrl Z** (undo) is your friend!

- After you're done shaping, disable proportional editing again with **O**.
- Just like you did above, extrude the outer edge twice more for a total of three times.

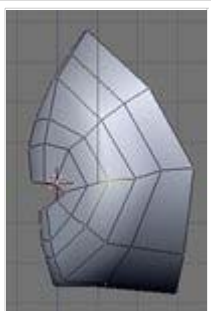
The images below show the results of a couple minutes of shaping the mesh. Note how I closed the mouth and gave the face some shape.



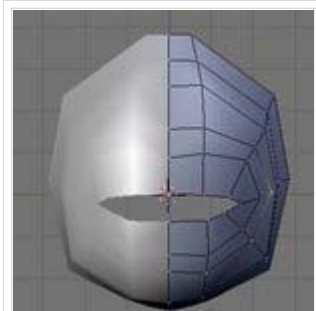
Extrude #2



Extrude #2



Extrude #3




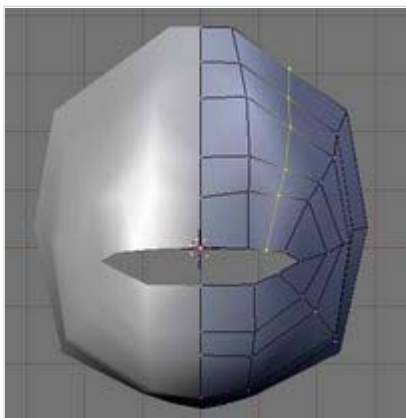
Extrude #3

This is where you can give your character some, well, character. I'm going to keep it simple in order to focus on the tools rather than the modeling. For example, I'm not going to worry about making a nose or ears for this character. Since you move the mouse differently than I do and will therefore shape the face differently than me, from here on out the model in this tutorial might look a little different from yours.

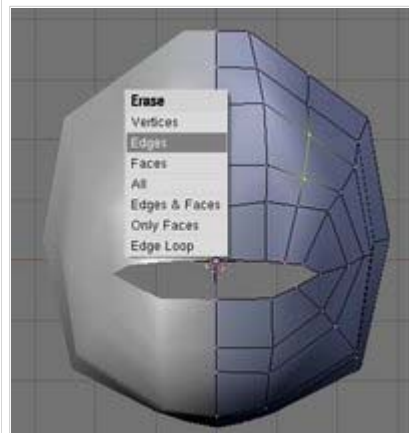
Making room for the eyes

The mesh will take a little prep work to make the eye fit properly. Here are the steps I took to make an eye socket:

- Make a loop cut (**Ctrl** **R** and MMB ) as in *Loop-cutting for the eye socket*. This will give us some more vertices to work with.
- Select the two vertices as shown in *Deleting edge*. Pay attention to the menu: this time, instead of deleting the selected *vertices*, we'll delete the selected *edge*. You do this by choosing **X**>>*Edges*. The vertices will remain, but the edge connecting them as well as the faces containing the edge, will be deleted.
- This makes a hole in the mesh



Loop-cutting **Ctrl** **R** for the eye socket.



Deleting edge to make room for the eye.


Deleting

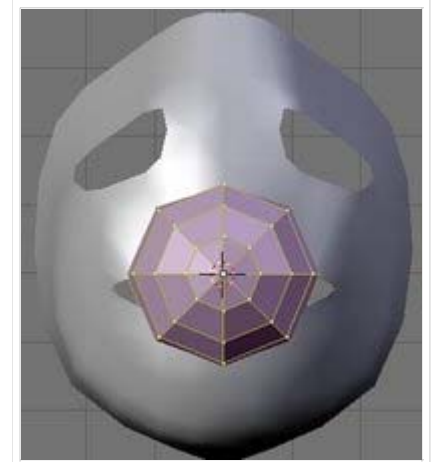
After pressing **X** to delete, you can select which part of the selection to delete from the menu.

- *Vertices* will delete the vertices, as well as any faces or edges containing those vertices.
- *Edges* will delete the edges and any faces containing those edges, but will leave vertices alone.
- *Faces* will delete just the faces and will leave edges and vertices alone
- *All* will get rid of **everything** in the mesh - including unselected regions!.
- *Edges & Faces* will leave you with a bunch of vertices floating in space.
- *Only Faces* will leave you a wireframe, without any faces.
- *Edge Loop*, which is a fairly new feature, is a great tool - it will remove an entire edge

Adding an eye

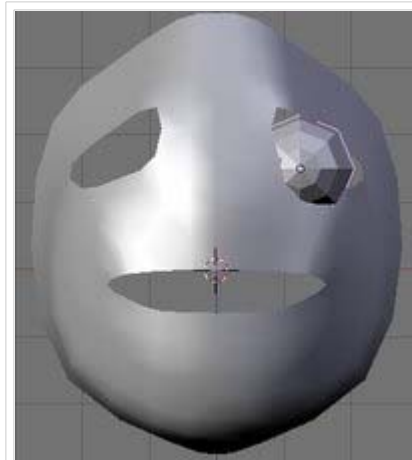
Let's add an eye. We want the eye to be separate from the face, so it can rotate freely. This won't be possible if we add vertices to the existing mesh. Instead, we will add another mesh object.

- Switch to Object Mode (**TAB**).
- Add a UV Sphere (**SPACE** >> Add >> Mesh >> UV Sphere). More info on mesh types can be found at Manual/PartII/Basic_Objects. To save yourself work, use the mesh type that best matches the object you're trying to model. I chose a UV Sphere instead of an icosphere because it will be easier to make the pupil using the UV Sphere's rings.
- You will be prompted for the number of Segments and Rings. Change both Segments and Rings to 8. You can either click the arrow buttons or LMB  on the number box to type in a value. You can use however many Segments and Rings you'd like; in this case I'm using 8. We can always subsurf it to make it look as smooth as we want. If the UV sphere has too many vertices, it may get difficult to work with. The default of 32 is too many - we don't need that kind of complexity for an eye.



Adding a UV Sphere (8 segments and 8 rings)

- Adding a mesh automatically puts you in Edit Mode. Switch back to Object Mode (**TAB**). We need to move the eye (it's clearly not in the right spot!), but if you move the vertices of the eye while in Edit Mode, the Object Center will stay where it was created. For reasons that will become clear later, we want the Object Center to remain in the center of the eye, so we have to switch Object Mode to move the object as a whole.
- Use **G** to position and **S** to scale the eyeball to where you ultimately want it to be. You'll need to change the view quite a bit to get the right size and location. In subsequent steps, we'll make the face mesh fit the eye - so don't worry about any gaps now between the eye and the eyelids.
- The result should look something like *Positioning the eye*.
- Just like we did for the face, use *Set Smooth* (in the Edit Buttons) and add a *Subsurf* modifier to make the eye smooth.

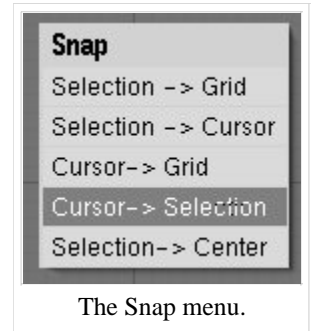


Positioning the eye, using **G** to grab and **S** to scale.


Duplicating and Mirroring the eye

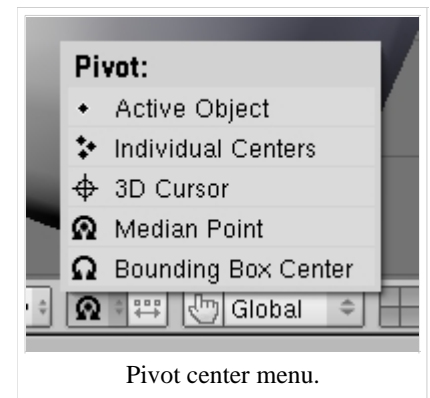
When you're happy with where the eye is, then we can make a duplicate and mirror it across the X-axis. However, we will **not** use a Mirror modifier for this. Why not?

Well, we don't want to mirror the eye in order to model it symmetrically. It's already symmetrical. Instead, we want a separate object for the other eye so that the eyes to be able to move around independently of each other. It's certainly possible to add another UV Sphere and position it and scale it exactly the same way as we just did, but there's an easier and more powerful way to make a second eye.

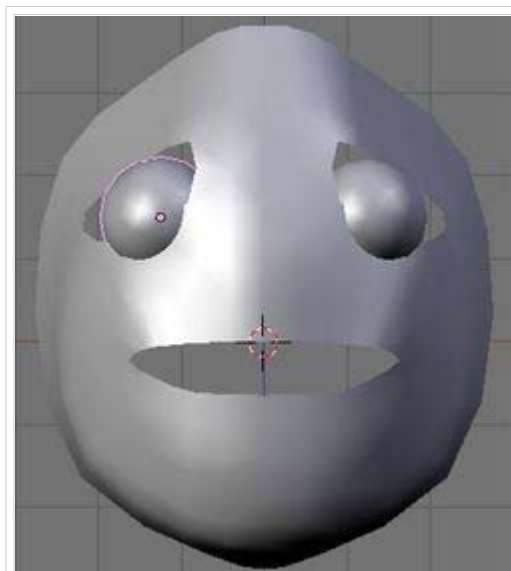


- Make sure you're in Object Mode.
- Make sure the 3D cursor is in the middle of the face. If you haven't moved it, it should still be there. If not,
 - Select the face mesh
 - Use the **Snap** tool **Shift S** >> *Cursor to Selection* to snap the cursor to the center of the face (*The Snap menu*).
- Select the eye.

- From the Pivot Center menu at the bottom of the 3D Window (*Pivot center menu*), select *3D cursor*. This will change the reference point to the 3D Cursor. The default was the median point of all items selected. In other words, we're forcing the center of rotation or the center of mirroring to be wherever the 3D Cursor is.
- With the eye still selected, use **Shift D** to Duplicate the eye. Just like in extruding, Grab mode is automatically activated. Try moving the mouse to see that we now have a duplicate eye.
- RMB  to cancel the automatic move mode. Just like extruding, **the duplicate object is still there, even though we cancelled the move**. The duplicate eye is still selected, and it's right on top of the original eye.



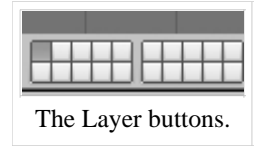
- **Ctrl M** >> *X Local* to mirror the duplicated eye.
- From the Pivot Center menu (*Pivot center menu*), go back to the default mode, *Median Point*.
- It should now look something like *Mirrored, duplicate eye*. The duplicate eye has been mirrored across the plane of the 3D Cursor, and there was no annoying moving and scaling to deal with.



Mirrored, duplicate eye.

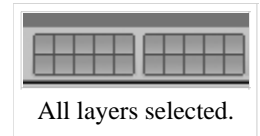
Move the eyes to a different layer



Up until adding the eyes, we only had one object to worry about (the face). Now, to make things easier later down the road, we'll move the eyes to a different layer. You access Blender's layer system either with the matrix of buttons in the bottom of the 3D Window (*Layer buttons*), or with the number keys on the top row of the keyboard. The default scene had only Layer 1 active, so the face and the eyes were both added to Layer 1.

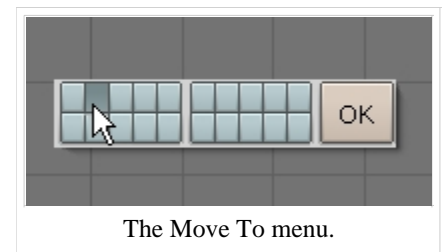


Let's move the eyes to Layer 2.

- Press `~` (the ~ key, to the left of the 1 key) to view all layers. Note how the Layer buttons all turn dark, indicating they are all selected (*All Layers selected.*)



- Select both eyes (RMB  and `Shift` RMB .
- Access the Move To menu with `M` (*The Move To menu*).
- Either press `2` or click the button for Layer 2, as shown in *The Move To menu*.

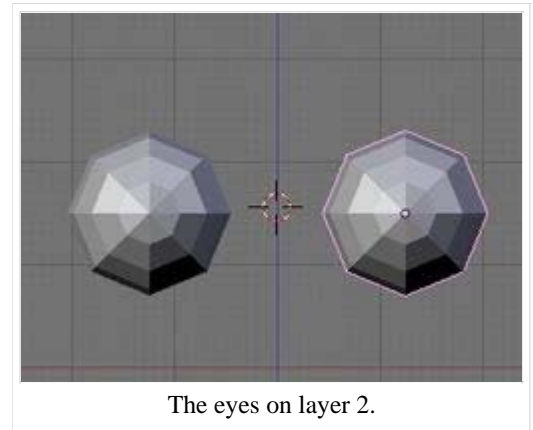


- Either press `Enter` or the *OK* button to confirm. Simply moving the mouse cursor away from the menu will cancel the move.
- Do the same thing with the face, to make sure it is on Layer 1 (Select the face, `M`, `1`, `Enter`).
- To view just the eyes, press `2`, or click the Layer 2 button in the Layers buttons. To view just the face, press `1`, or click the Layer 1 button in the Layers buttons.
- To view multiple layers, use `Shift`: `1`, then `Shift` `2` will show both Layer 1 and Layer 2. Alternatively, hold `Shift` while clicking the Layer buttons.

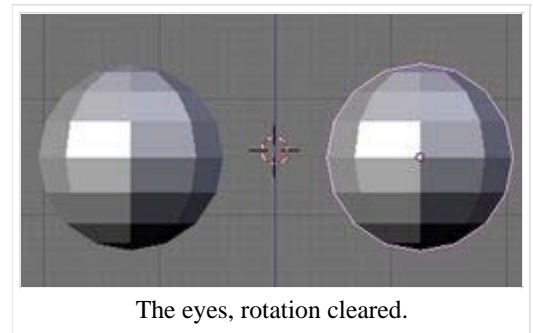
Adjusting the axes of the eyes

Mirroring the eyes across the X-axis was a useful way of making sure they were symmetrical. However, it also made the axes of the eyes different from each other. Later on when we add Track To constraints, we'll want the axes of both the eyes to be identical (this will become clearer later).

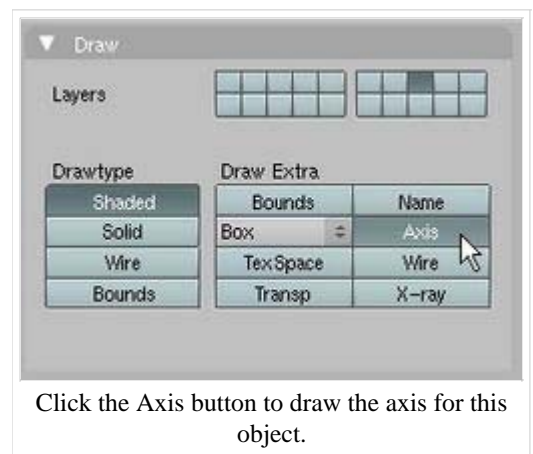
- View Layer 2 alone by using **2**.
- Select one of the eyes.



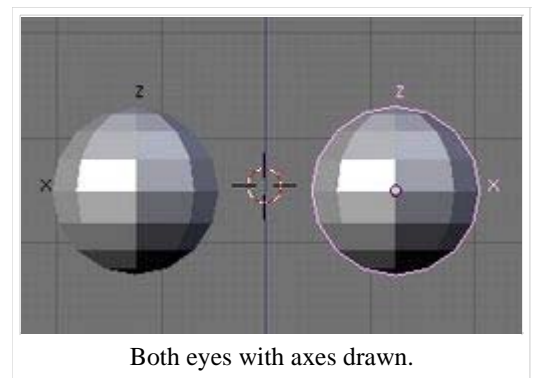
- Clear the rotation of the object with **Alt R**. This resets the rotation so the axes of the eyes are pointing straight up, so we're all on the same page.



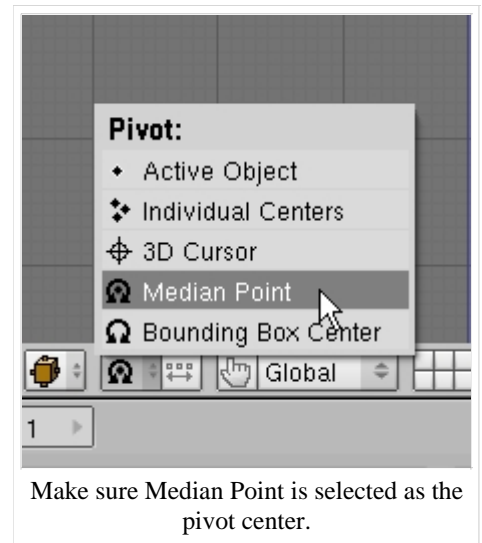
- Under the Object buttons in the Buttons window, look for the *Draw* panel and press the *Axis* button. This draws the axes (X,Y, and Z) for this object.
- Select the other eye and enable drawing of the axis for it as well.



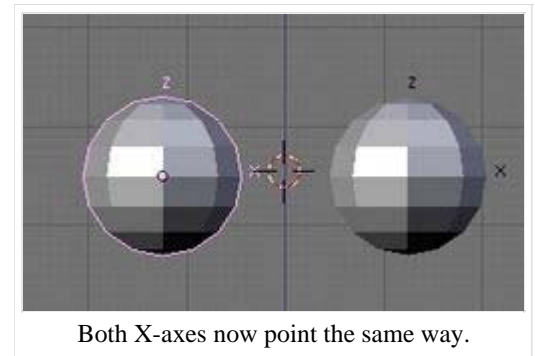
- Notice how the Z axis is pointing up for both eyes, but the X-axis points to the left in one eye and to the right in the other eye. The reason this happened is that we mirrored the second eye across the X-axis - so the new X-axis direction is a mirror image of the original one. We want both X-axes pointing the same way. To do this, we will mirror the second eye in place instead of across the 3D Cursor like we did earlier.



- Select the second eye.
- Make sure the *Pivot center* is set to *Median Point*.

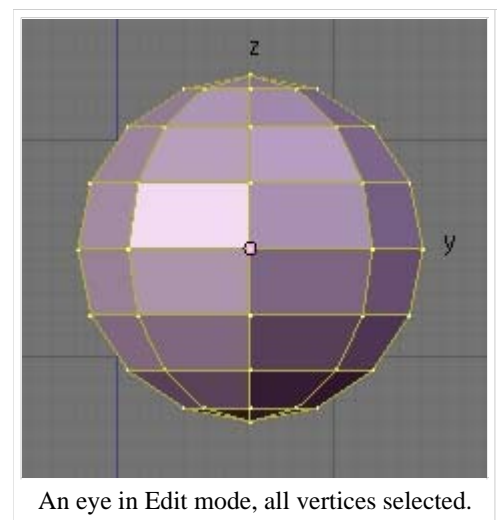


- Mirror the second eye in place with **Ctrl M** >> *X Local*. Now the X-axes of both eyes point the same way.



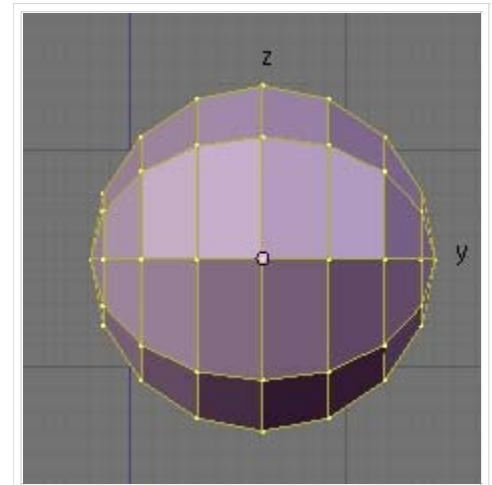
Next, we want to rotate the eyes so that the "poles" are pointing forward so we can easily make the pupils of the eyes. We want to leave the axis alone, since we just fixed it. To rotate the eye without changing the object's axis, we need to enter Edit mode.

- Select an eye and enter Edit mode.
- Switch to Side view (**3**).
- Select all vertices with **A** (you may have to hit it twice).



- Rotate the vertices 90 degrees with **R - 9 - 0 - Enter**.
- Switch to Object mode, select the other eye, and rotate it 90 degrees in Side view as well (it may be hidden behind the other eye, that's OK - your rotation commands will still work).

OK, we've rotated the eyes correctly to compensate for the mirroring and made their poles point toward the front. Make sure you're viewing both the eyes and the face for the next series of steps (remember, **`** views all layers).

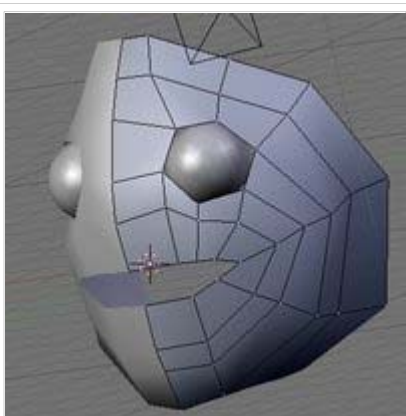


Vertices rotated 90 degrees in Side view.

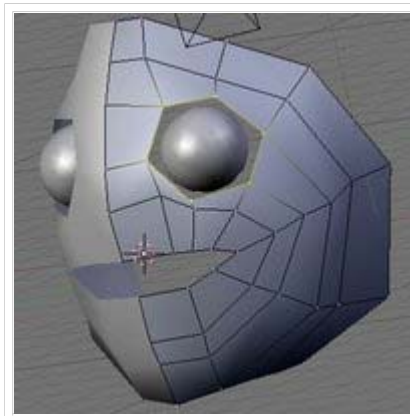
Forming the eye socket

Now it's time for more vertex-moving to accommodate our character's new eyes. We'll have to add some more vertices to work with, and seal off the holes surrounding the eyes. The goal is to have enough geometry around the eyes that the eyelids can close - we'll be making the character blink as well as have other expressions that involve the eyes.

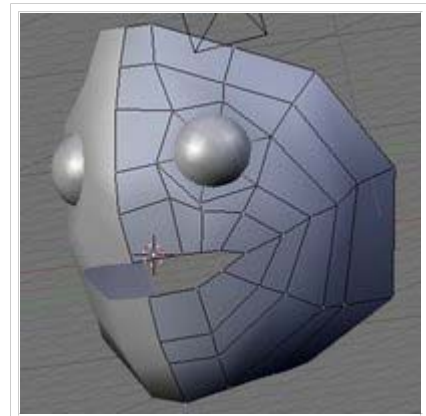
- Switch to the face's Edit Mode (**TAB**).
- Move the vertices surrounding the eye into more of a circular shape as in *Forming the eye 1*.
- Select the eye socket with **Alt RMB** and scale it up a little with **S** as in *Forming the eye 2*.
- With the eye socket still selected, extrude it and scale it down (**E** >> Edges Only, **S**). Scale it down far enough that it goes *into* the eye. There shouldn't be any visible holes now, as in *Forming the eye 3*.



Forming the eye 1 (moving the vertices around the eye into a circle).



Forming the eye 2 (scaling up the ring).

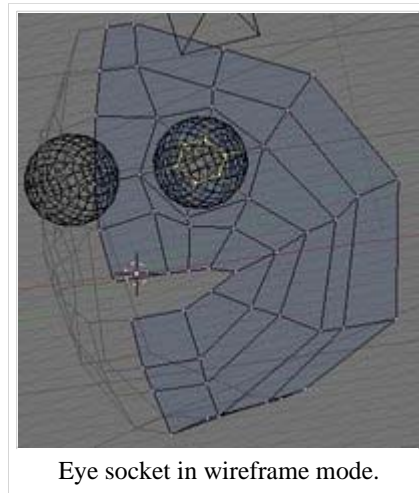
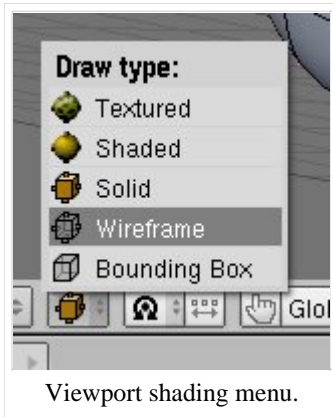


Forming the eye 3 (extruding and scaling down the extruded ring).

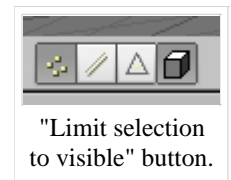
The ring that you just scaled down isn't visible. How to see it?

- Press **Z** to enter **Wireframe** mode (*Eye socket in wireframe mode*). Pressing **Z** again will get you back to **Solid** mode. Alternatively, select the Wireframe option from the Viewport Shading menu at the bottom of the 3D Window.

OR...



- ... Press the "Limit selection to visible" button at the bottom of the 3D Window. This button is only available in Edit Mode.



Up until now, we've been using the default Solid draw mode. That's when you see the solid faces of the mesh. By default, vertices that are behind the visible faces are invisible and you can't select them. This makes the interface faster (less vertices to calculate) and keeps the view uncluttered. Often, however, we need to see inside of a mesh. That's when you use either Wireframe draw mode or toggle Back-Face Visibility.

Viewport Shading

Z to switch between Solid and Wireframe draw modes.

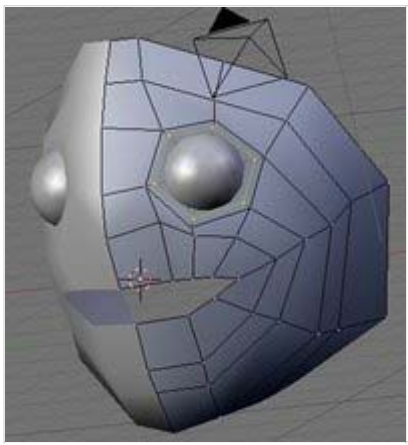
Or use the Viewport Shading menu to switch draw modes.

Press the "Limit selection to visible" button to toggle the visibility of vertices behind faces.

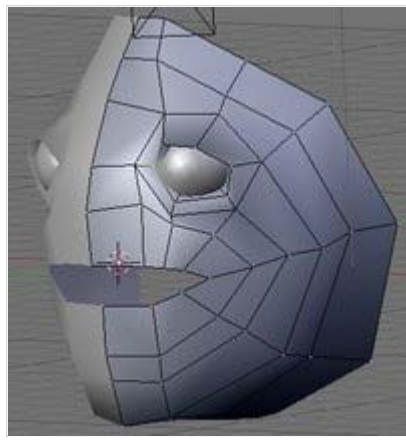
More info: [Manual/PartII/Draw types](#)

OK, now for a little more shaping to make some eyelids.

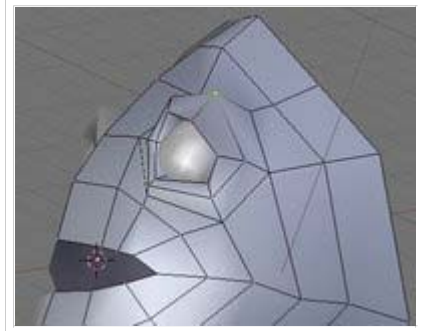
- Loop-cut the ring around the eye socket (**Ctrl R**) to get some more vertices to work with (*Loop cutting the eye socket*).
- Pull those new vertices forward and down a little using **G** to start forming the eyelid, as in *Forming the eyelid 1*.
- Make another loop cut (**Ctrl R**) between the old and new rings to get even more vertices, and move them around to give the eyelid some more body. To do this, pull the top vertices up a little bit, and the bottom vertices down a little bit, something like *Forming the eyelid 2*.



Loop cutting the eye socket



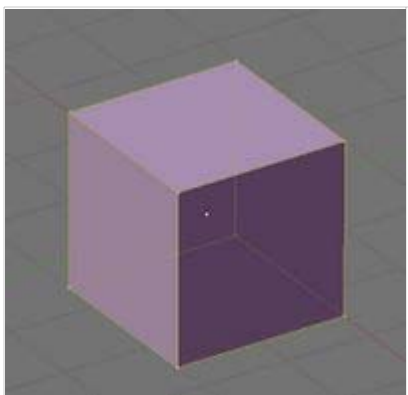
Forming the eyelid 1. The new vertices from the loop cut are moved to form the beginning of an eyelid.



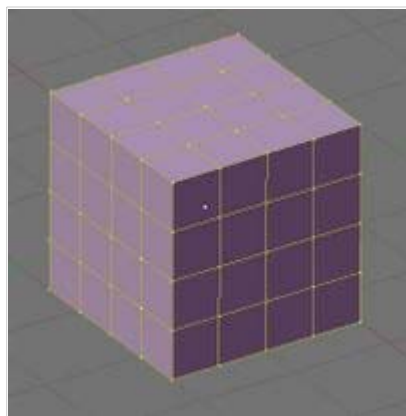
Forming the eyelid 2. A new loop cut was made, and the new vertices are moved to give the eyelids more body.

Why don't we want too many vertices?

Throughout this tutorial I talk about adding only as many vertices as we need. Generally speaking we want to have only enough vertices as we need to get the shape we're modeling. For example, we only need 8 vertices to model a cube . . . any more vertices are unnecessary.



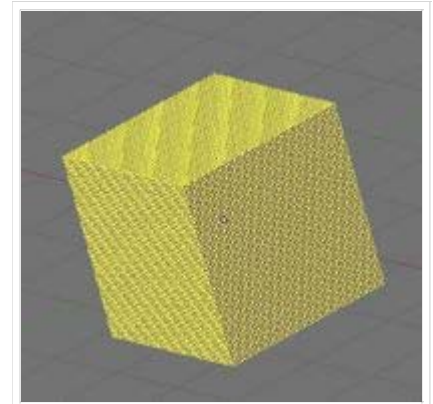
Cube with exactly enough vertices to form the cube shape (8 vertices).



Cube with unnecessary vertices (98 vertices).

You might think that more vertices are bad because it will slow down your computer. Typically, this is not the case. For example, I had to subdivide the cube several times to get a vertex count of almost 25,000 vertices before I saw noticeable slowdown on my computer when I rotated the view. We'd be hard-pressed to make a single character with that many vertices.

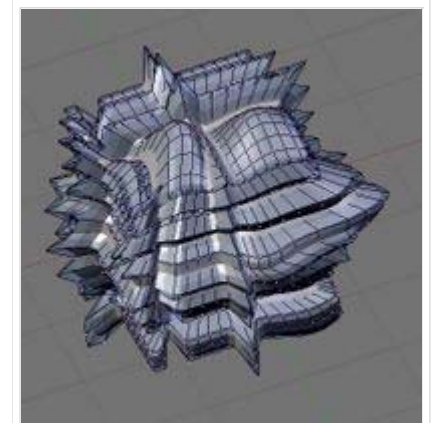
So if computing power is not limiting, what is? Brain power. When you're modeling, you generally have to select individual vertices and move them into place. The more vertices, the more selecting and moving you have to do. Imagine trying to make the cube into a truncated pyramid by scaling down the top. It's simple to do with an 8-vertex cube, but it would take more time to select the vertices with the 98-vertex cube or the 24,578-vertex cube. So it's a matter of convenience.



Cube with enough unnecessary vertices for a noticeable lag when rotating the view (24,578 vertices).

The more complex the shape, however, the more vertices are needed. For example, the object to the right needs just about all 1,538 vertices to make the shape.

In the end, it's up to you. These are just general guidelines. The point is, you don't have to go out of your way to reduce the number of vertices in your mesh to the absolute minimum. If you think you need extra vertices, add them!



A complex object that needs most of the 1,538 vertices to make the shape.

Don't forget to save a version with `F2` - NumPad `+` - Enter !

Summary: We extruded the mouth further to form the face, added an eye, then duplicated and mirrored the eye using the 3D Cursor as a reference point. After a few more tweaks to the mesh, we now have a face with eyes and eyelids.

Now we can finish making the head.

Next: Finishing the head

Previous: Creating the mouth

Back to Index

Retrieved from "http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Face_and_eyes"

-
- This page was last modified 20:20, 14 September 2006.

BSoD/Introduction to Character Animation/Mouth and head

From BlenderWiki


< BSoD | Introduction to Character Animation

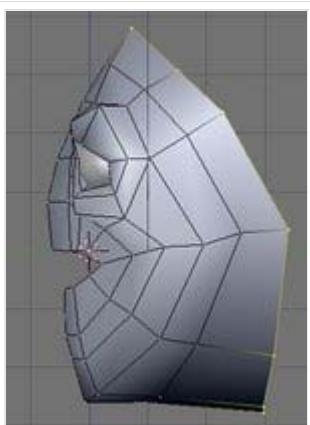
Contents

- 1 Modeling the rest of the head
- 2 Making the inside of the mouth
- 3 Closing the mouth
- 4 Final adjustments
- 5 The finished face

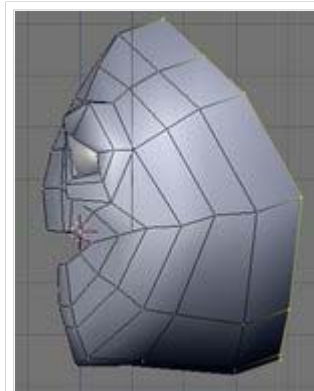
Modeling the rest of the head

Now we'll finish modeling the head. It will consist of several extrusions and moving lots of vertices around.

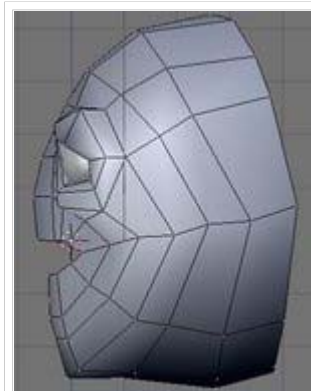
- **Alt** **RMB**  to loop-select the outside edge, as in *Head vertices to extrude*.
- **E** to extrude, as in *Head extrusion 1*.
- Move vertices so they're a little closer to the same plane (*Adjusting vertices*).




Head vertices to extrude.

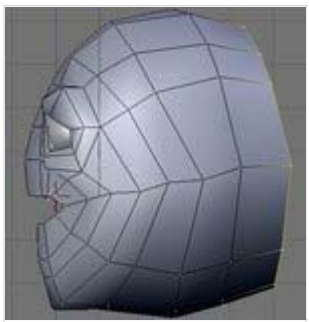


Head extrusion 1.

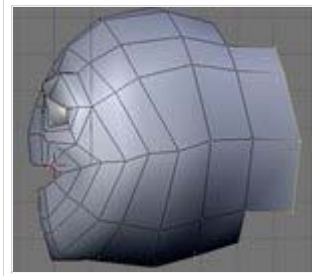


Adjusting vertices.

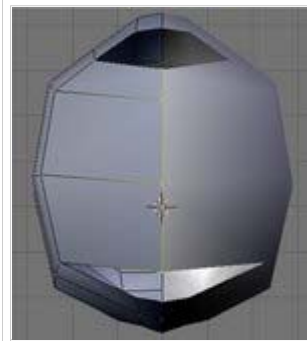
- **Alt** **RMB**  to loop-select the new outside edge, and extrude this new edge as in *Head extrusion 2*.
- Now let's do a third extrusion, but not on the whole edge. This time, select just the middle four vertices and extrude them straight backward as in *Head extrusion 3*.
- Switch to Rear View (**Ctrl** **NumPad 1**). Adding the **Ctrl** to the hotkey for Front View (**NumPad 1**) gives us the opposite (rear) view.
- Move the newly extruded vertices toward the center of the head with **G**. Thanks to the *Mirror* modifier and *Do Clipping*, they can't go any farther than the center plane (*Merging the back of the head*). Once they get to the center line, they snap there and essentially merge with the centerline.



Head extrusion 2.

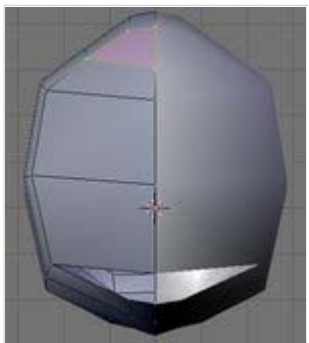


Head extrusion 3.



Merging the back of the head.

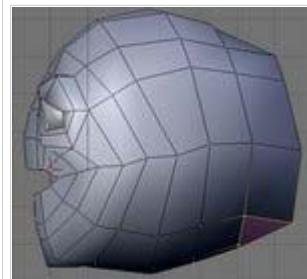
- Close off the holes by making faces: select four vertices at a time (**RMB**) and create a face out of those vertices (**F**). See *Finishing head 1*, *Finishing head 2*, and *Finishing head 3*.



Finishing the head 1.



Finishing the head 2.

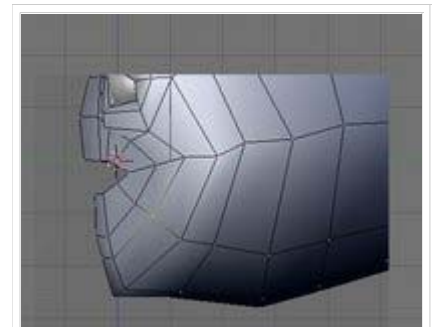


Finishing the head 3.


Making the inside of the mouth

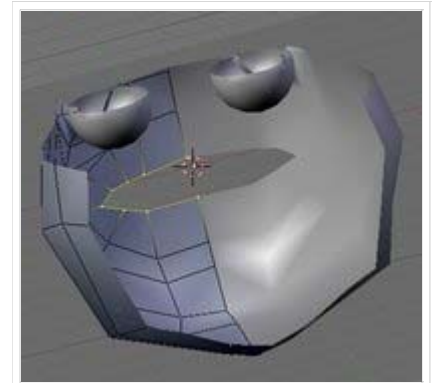
The next step is to close up the mouth. To do that we'll have to extrude the lips inward to make a "pocket" that will form the inside of the mouth. We'll use a new feature, **Alt B**, to work on the inside of the mesh.

- Switch to side view (NumPad 3).
- Clip the view by pressing **Alt B** and dragging a box around the lower front part of the head, **LMB** to confirm. (*Clipping the view*). Clipping the view is a valuable tool for working with complex meshes. It is a way of hiding parts of a mesh you don't need to see. It might take a little practice to figure out how to get the view you want, but once you figure it out it's very helpful. You can always hit **Alt B** again to restore the view.



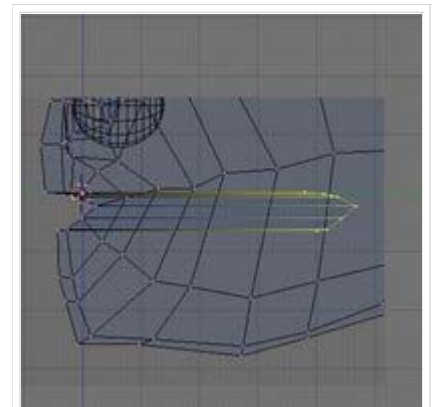
Clipping the view.

- Now rotate the view to see the inside of the head.
- Loop-select the lips with **Alt** RMB , as in *Lips selected in clipped view*.



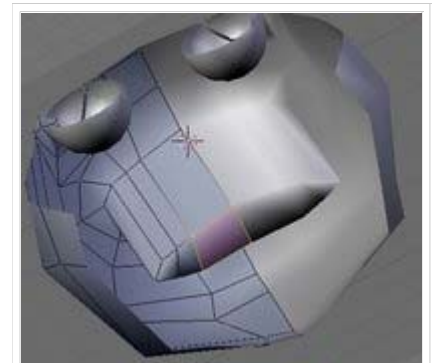
Lips selected in clipped view.

- Switch to Side View in Wireframe mode (NumPad **3**, then **Z**). This is so we can extrude backward and see how far back we're going.
- Extrude the edgeloop straight back (*Extruding lips backward*).



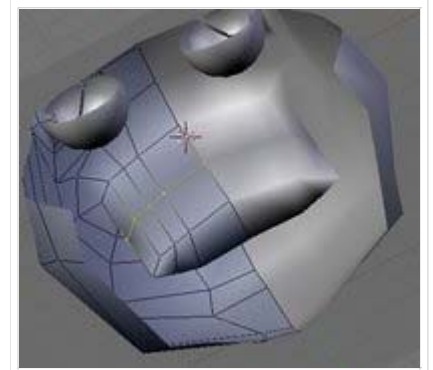
Extruding lips backward.

- Switch back to Solid mode so we can easily see the faces we're about to make (**Z**).
- Starting from one end of the extruded edge loop, select four vertices at a time and press **F** to make a face out of them. Make several faces to close off the back of the mouth, as in *Closing off the back of the mouth*.



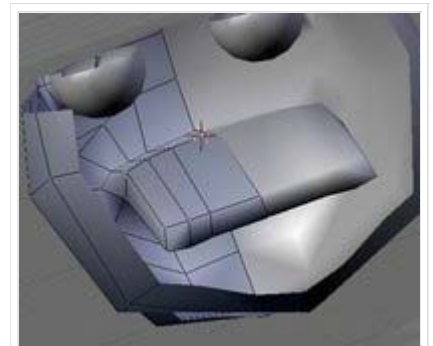
Closing off the back of the mouth.

- Make a loop-cut (**Ctrl** **R**) around the inside of the mouth, as in *Loop cutting the inside of the mouth*. This will allow us to give the mouth enclosure more volume.



Loop cutting the inside of the mouth.

- Shape the inside of the mouth, as in *Shaping the inside of the mouth*:
 - Enlarge the newly loop-cut vertices
 - Narrow the newly created faces at the back of the mouth
 - Lower the back of the mouth.

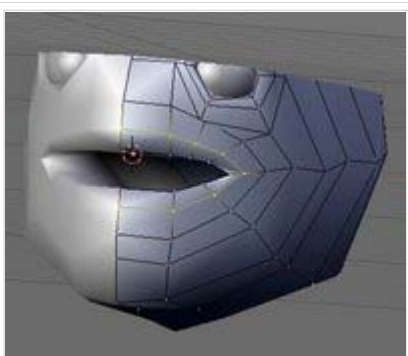


Shaping the inside of the mouth.

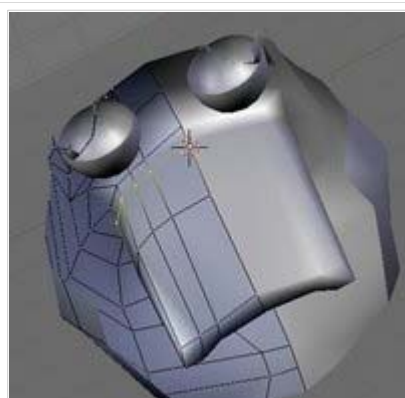
Closing the mouth

- Make two more loop cuts:
 - One on the front of the face (*A loop cut on the front*)
 - One on the inside of the mouth (*Another loop cut on the inside of the mouth*).

By having several vertex loops close to each other like this, the transition between face, lips, and mouth will be more distinct.



A loop cut on the front.

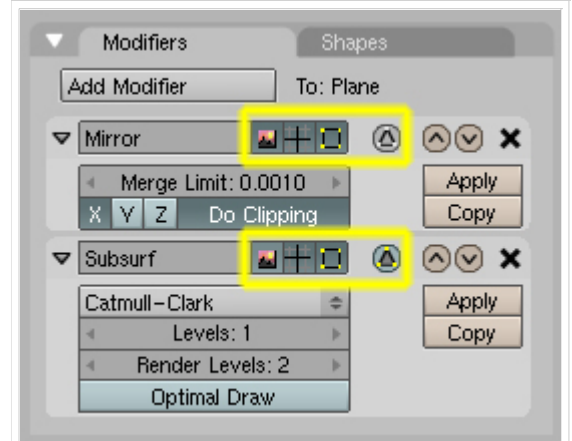


Another loop cut on the inside of the mouth.

Now we're ready to close the mouth. This is basically just grabbing vertices and moving them, but there are a couple of things you should be aware of:

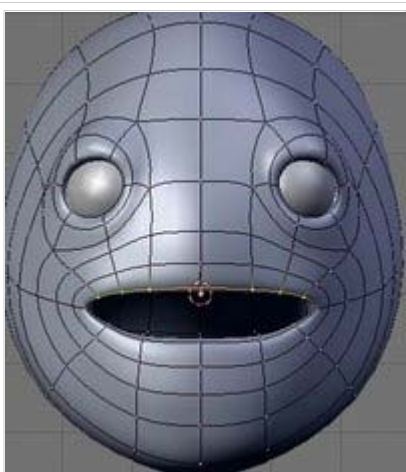
The final character is going to have the *Subsurf* modifier applied (it will be "subsurfed") to make it look nice and smooth. Meshes look different depending on whether they are subsurfed or not. It's possible that the mouth might look closed with Subsurf turned off, but once we turn on Subsurf the mouth opens a little. The trick is to make sure the mouth is closed when the mesh is subsurfed. To do this:

- Use the *Subsurf* modifier buttons and apply it to the editing cage (remember the gray circle on the right side of the Subsurf Modifier? See *Modifier stack buttons*). If you don't turn on subsurf in edit mode, you'll find that it's difficult to know when the mouth is fully closed.
- Now select vertices and start closing the mouth. As you're closing the mouth, you'll have to move the vertices in several edgeloops. You'll get a better shape that way.
- Try to make the expression of the face "bored". We will be forming mouth shapes and expressions later on, and it's best to start with a face with no expressions - so no smile or frown or anything on the shape we're building now.

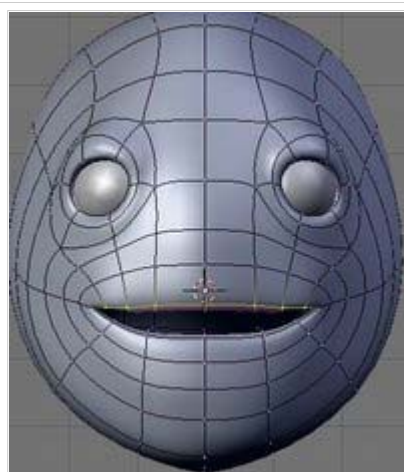


Modifier stack buttons.

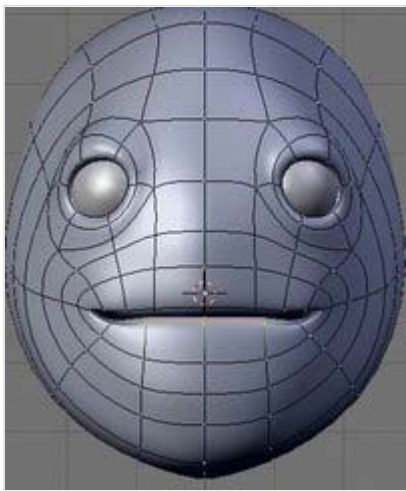
The series of images below shows the progressive closing of the mouth, with *Subsurf* turned on in Edit Mode. When moving the lips, I grabbed the vertices just inside the mouth, not just the lip vertices. First the top lip was moved down, then the bottom lip moved up, and then the sides of the mouth were brought in a little bit.



Closing the mouth 1 (Lip vertices, as well as nearby vertices inside the mouth, are selected.)



Closing the mouth 2 (moved the top lip, along with the vertices just inside the mouth, downward).



Closing the mouth 3 (moved the bottom lip upward).



Closing the mouth 4 (moved the corners of the mouth down and inward a little to get rid of the faint smile).

Final adjustments

- Turn off *Subsurf* in Edit Mode, by using the buttons in the Subsurf Modifier. Now we can see the underlying editing cage, as in *Subsurf modifier turned off*. Note that the upper lip of the editing cage appears to be pulled down past the lower lip. You can't even see some of the lower lip vertices, because they're covered by the upper lip.

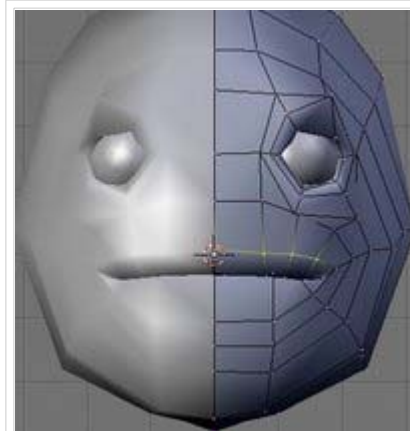
That's OK! A subsurfed surface is always smaller than its editing cage (picture a sheet hanging from the vertices of the editing cage). We want the mouth to be closed when it's subsurfed -- in other words, we want the subsurfed faces to touch each other.

In order to do that we have to overcompensate by pulling the un-subsurfed upper lip over the lower lip vertices. By editing the mesh as we did earlier with *Subsurf* applied to the editing cage, we didn't have to guess how far to pull the upper lip down. We were able to edit it directly.

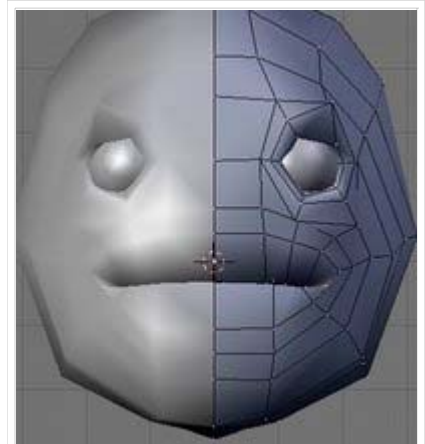
Now with *Subsurf* still turned off, you can see a dark area around the upper lip. This dark area is often caused by extreme stretching or bending of the editing cage. This often happens if you shape your mesh exclusively with *Subsurf* turned on.

To fix it, either add more vertices or move nearby vertices closer. Luckily, we made that loop cut a couple steps back to give ourselves more vertices to work with.

- Move the second row of vertices in the upper lip down a little (as in *Fixing the extreme angles*), this will smooth out the mesh.



Fixing the extreme angles on the upper lip by moving nearby vertices.

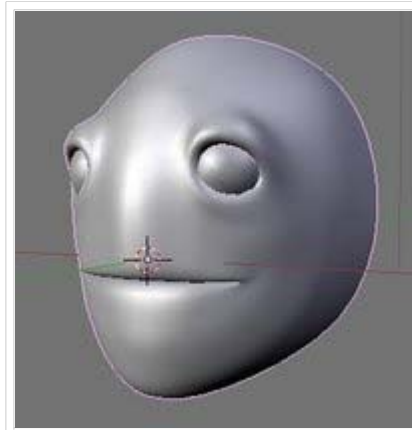


Subsurf modifier turned off. The upper lip appears to be pulled over the lower lip.

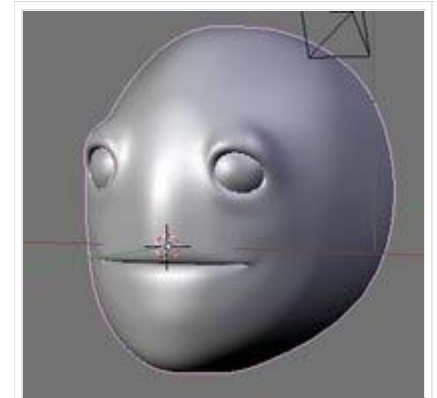
The finished face

The finished face looks like *Finished face 1* and *Finished face 2*. The difference between the two that (1) is in orthographic view, and (2) is in perspective view. You can switch between the views with NumPad 5.

Orthographic view is the default view. In orthographic view, all planes in the view are perpendicular to each other - there's no perspective, no "vanishing points", no taking distance into account. Ortho view is handy for when you want to transform vertices only in one plane, when you want to model mechanical objects, or when you're trying to be more exact in how the vertices are moved.



Finished face 2 (Perspective view).



Finished face 1 (Ortho view).

Perspective view is a "real-life" kind of view - distance is taken into account. It's also the default view for the camera, so when you render a project the final image will be in perspective view. This view more realistic, and is useful for getting the final shape of objects.

Don't forget to save a version with F2 - NumPad + - Enter !

Summary: That's it! Now you have a face with enough geometry that you can animate facial expressions. Next, we'll create a body for the character, before moving on to rigging and animation. Get ready for more extruding and moving!

You can download the .blend file for the finished head here:
Media:Tutorial_head.blend

Next: [BSoD/Introduction_to_Character_Animation/Neck_shoulders_and_arms](#)

Previous: [Face and eyes](#)

Back to Index

Retrieved from "http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Mouth_and_head"

■ This page was last modified 15:46, 27 July 2006.

BSoD/Introduction to Character Animation/Neck shoulders and arms

From BlenderWiki


< BSoD | Introduction to Character Animation

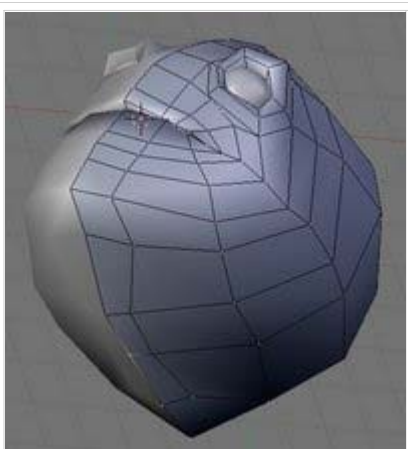
Neck and shoulders

Starting from the head created in the last section of the tutorial, we'll extrude the rest of the body.

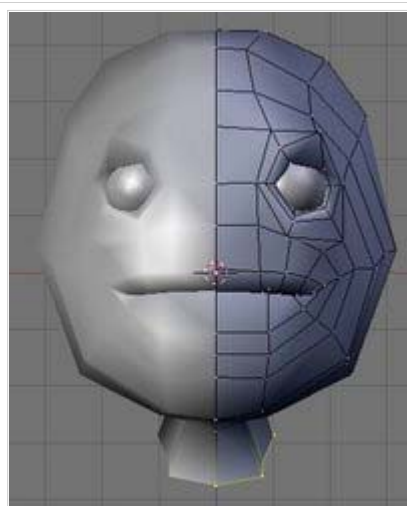
Note: In this section of the tutorial we're only worried about the geometry: which vertices are connected to which and how many vertices there are. So don't worry about extruding vertices exactly as they are in the screenshots.

In the next section, there will be plenty of time for shaping the body.

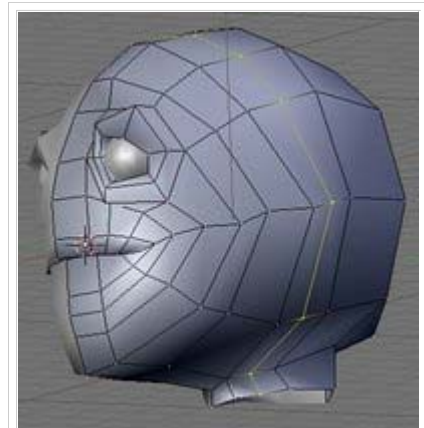
- Delete the vertex shown in *Delete vertex to extrude neck*.
- **Alt** RMB  to loop-select the surrounding vertices, and extrude downward to form the neck as in *Extrude the neck*.
- Shape the neck vertices so they're more even with each other, and loop cut (**Ctrl** **R**) as shown in *Reshape neck and loop cut around the head*. This will give us more vertices to work with when creating the shoulder.



Delete vertex to extrude neck.



Extrude the neck.

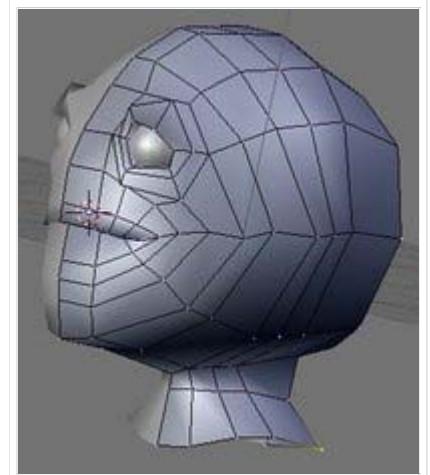


Reshape neck and loop cut around the head.

During this tutorial we've been making many loop cuts. Most of the time, the way I know to tell you to make a loop cut is because after some trial and error, it looks like the next couple steps are going to need some extra vertices to make things come out correctly. For example, I tried making the shoulder *without* the loop cut we just made, and found it would be difficult. So I backed up a couple steps and made the loop cut, and that's where we are now.

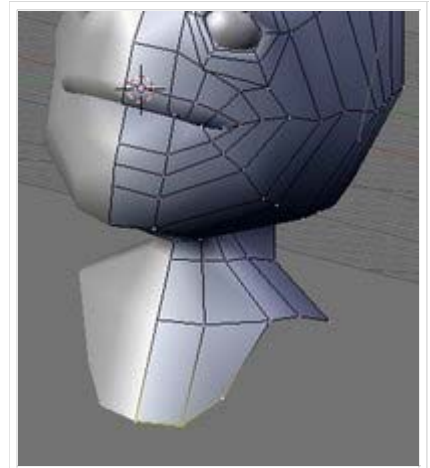
When you're making your own model, generally you'll try to work with the vertices you have, and then make loop cuts or add vertices when you find you can't make the shapes you want with the vertices you have. This cuts down on unnecessary complexity of the mesh.

- Extrude the neck and shape it as in *Extruding the neck and reshaping 2*.



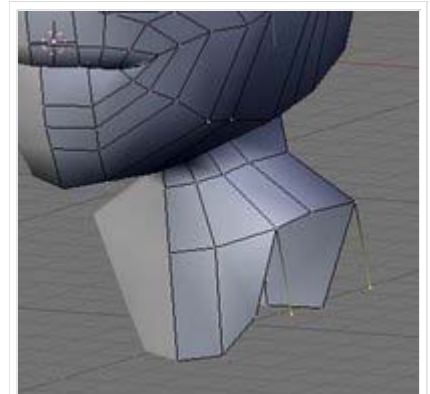
Extruding the neck and reshaping 2.

- Extrude the front three vertices down, as in *Extruding the chest* as well as the back three vertices (you can see them extruded in *Building the shoulder 1*).



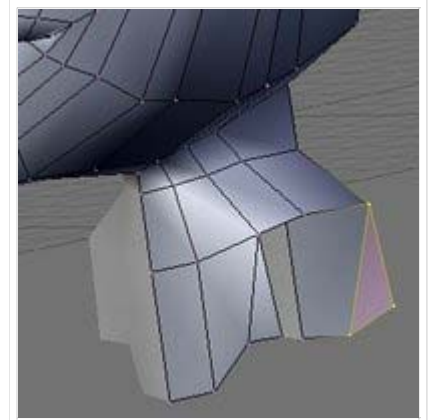
Extruding the chest.

- Extrude the top corners of the shoulder down (*Building the shoulder 1*).



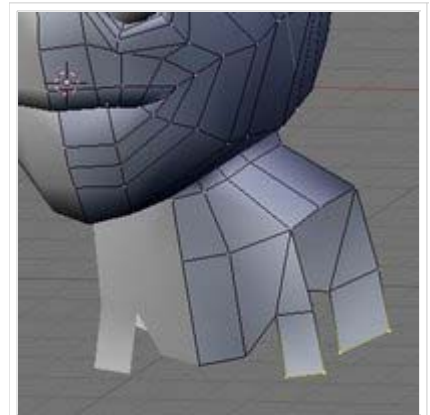
Building the shoulder 1.

- Make triangle faces out of the newly extruded vertices and the neighboring vertices (*Building the shoulder 2*).



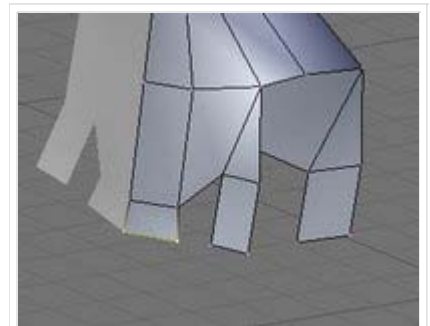
Building the shoulder 2.

- Extrude the bottoms of the triangles down (*Building the shoulder 3*).



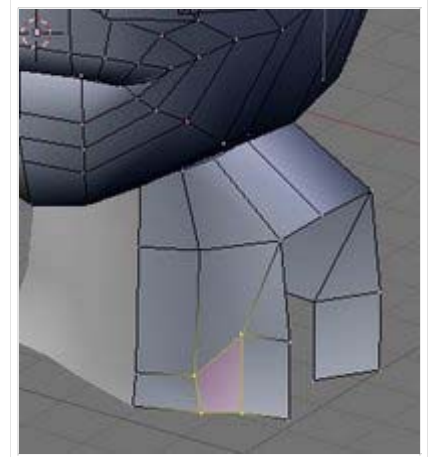
Building the shoulder 3.

- Extrude the front middle two vertices down (*Building the shoulder 4*).



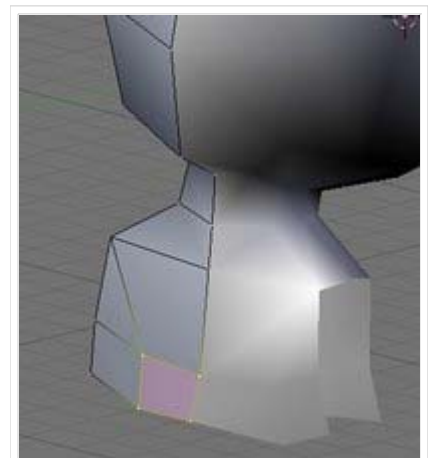
Building the shoulder 4.

- Make a quad face in the obvious spot (*Building the shoulder 5*).



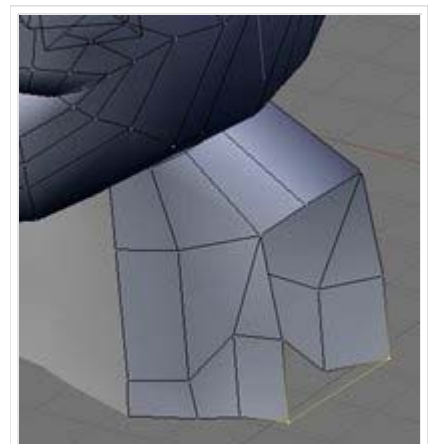
Building the shoulder 5.

- Make a quad face on the back to close the mesh (*Building the shoulder 6*).



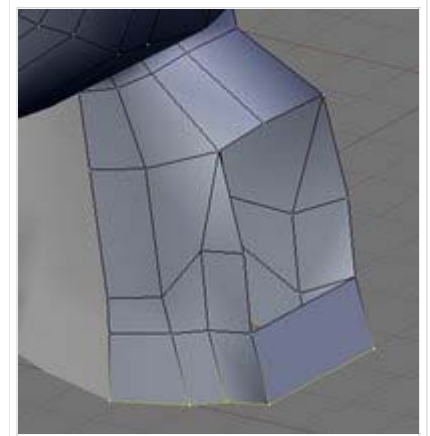
Building the shoulder 6.

- Connect the front and back below the armpit by selecting the 2 vertices and hitting F to make a 2-vertex face - that is, an edge (*Building the shoulder 7*).



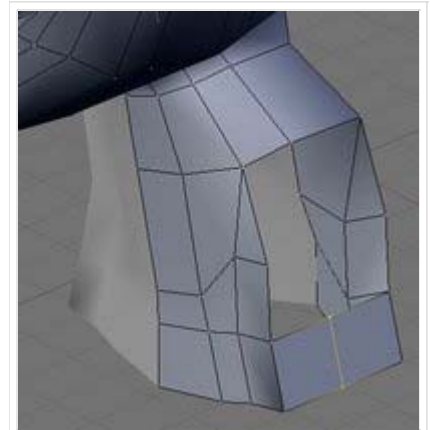
Building the shoulder 7.

- Extrude the entire bottom edge downward (*Building the shoulder 8*).



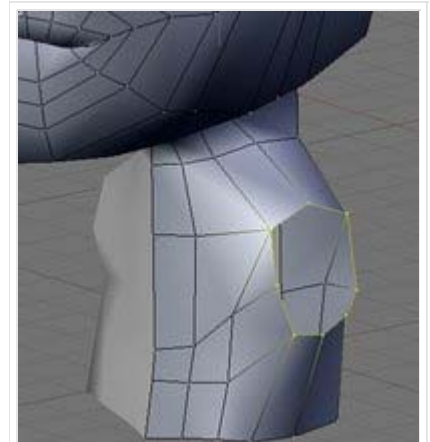
Building the shoulder 8.

- Loop-cut the newly created face (*Building the shoulder 9*).



Building the shoulder 9.

- Finally, reshape the shoulder area so that the empty hole forms the shape of the arm (*Shaping the root of the arm*).



Shaping the root of the arm.

Extruding the arm

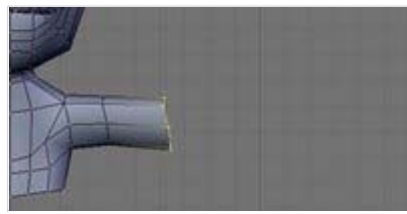
Next, we'll make the arm by extruding, moving, and scaling rings of vertices.

- Select the ring of vertices shown in *Shaping the root of the arm*.

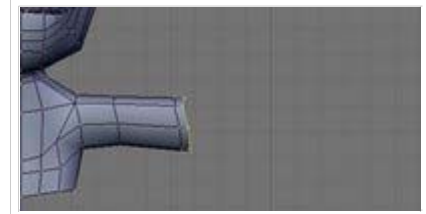
- Extrude the vertices, move them to the right, and scale them a little.
- Extrude, move, and scale for a total of 6 extrusions (*Extruding the arm 1-6*).
- Note that we're putting an extra loop around the elbow. That will help it bend better when we start animating.



Extruding the arm 1.



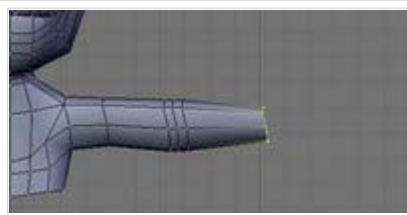
Extruding the arm 2.



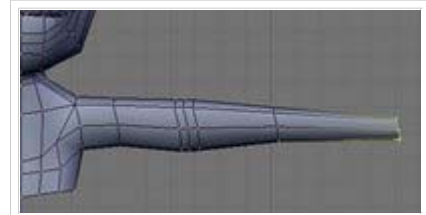
Extruding the arm 3.



Extruding the arm 4.



Extruding the arm 5.

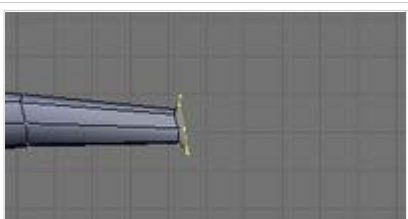


Extruding the arm 6.

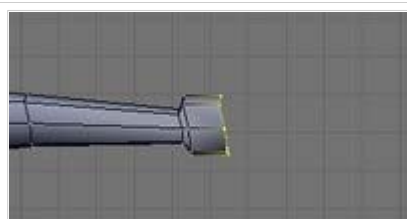
Making the hand

Now we'll make the hand. It'll be a simple "mitten" type hand since we're trying to keep it simple here . . .

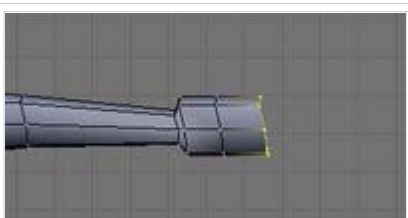
- Extrude the end of the arm several times to make the hand (*Extruding the hand 1-4*).



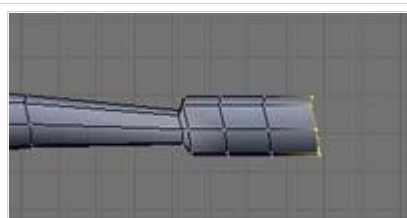
Extruding the hand 1.



Extruding the hand 2.

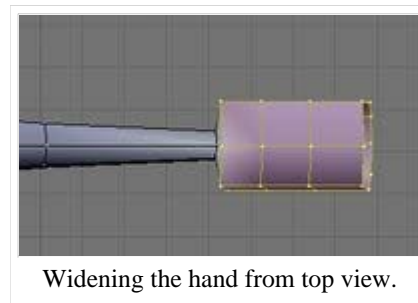


Extruding the hand 3.



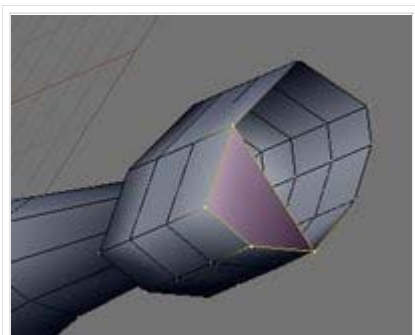
Extruding the hand 4.

- Widen the hand (NumPad **7** for top view, **S** to scale, **Y** to constrain scale axis), as in *Widening the hand from top view*.

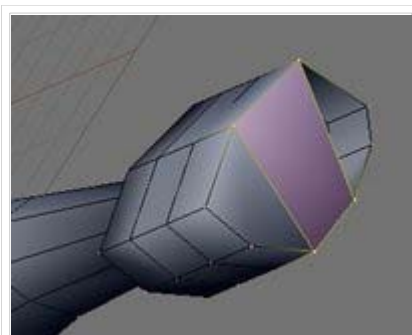


Widening the hand from top view.

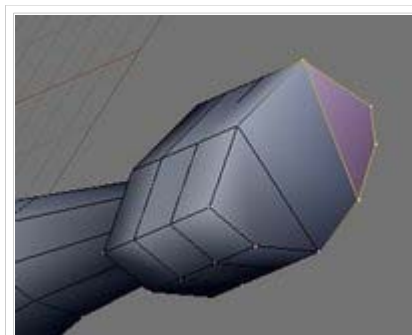
- Close off the end of the hand with faces (**F**), *Closing off the hand*).



Closing off the hand 1.

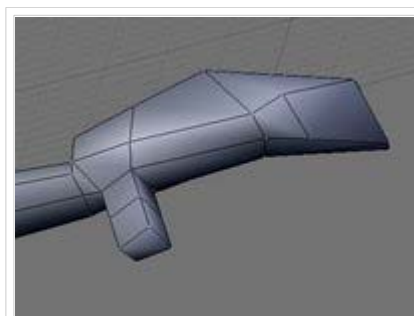


Closing off the hand 2.

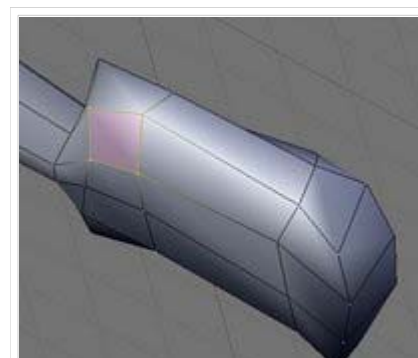


Closing off the hand 3.


- Now spend some time shaping the hand. I prefer to turn *Subsurf* on in Edit Mode when doing this kind of work.
- While you're shaping, make room for the thumb to be extruded. I made sure there was a square face for the thumb to be extruded (*Shaped hand, with square face to extrude thumb from*).
- Extrude the thumb from the hand, as in *Thumb extruded from hand*. Up until this point, everything we have extruded has been an edge. This is the first time we're extruding a face, so we'll get some options from an Extrude menu. Select **Region** from the extrude menu. This is the most commonly used option, and you can experiment with the other options to see what they do.



Thumb extruded from hand.

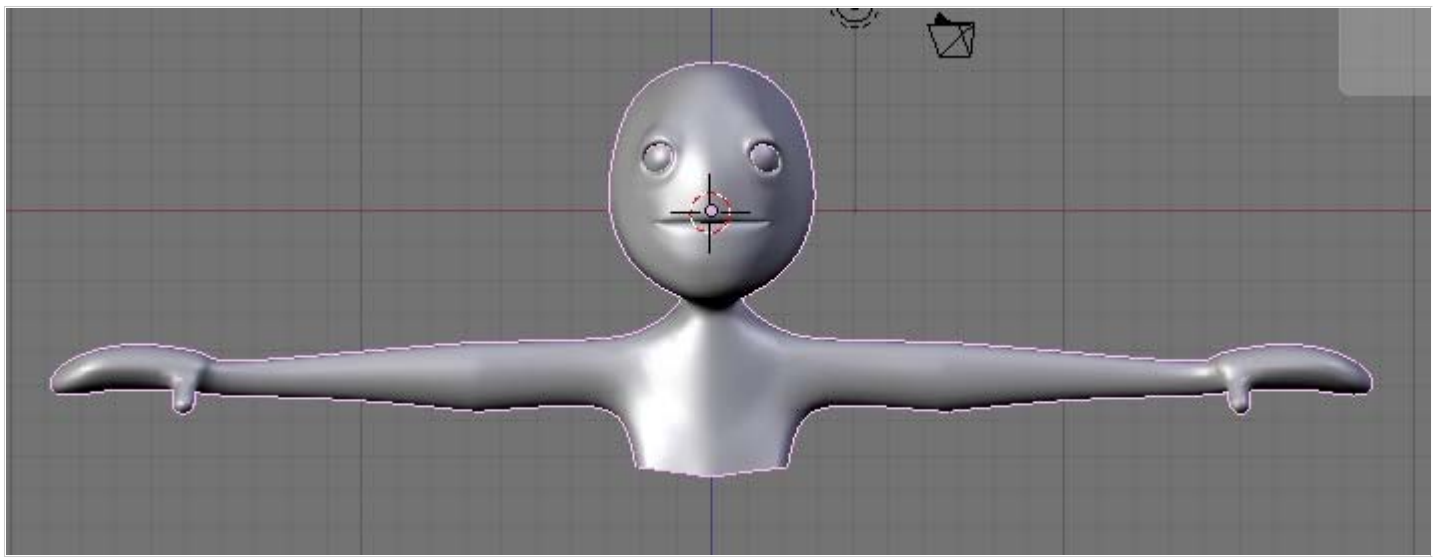


Shaped hand, with square face to extrude thumb from.

Note that when you extrude, there is a orange line that the extruded region moves along. This is so that the extruded faces have the same orientation as the "parent" face. To cancel this constrained move, **MMB**  to get into free move mode.}}

Don't forget to save a version with **F2 - NumPad **+** - **Enter**!**

The character so far is shown below. Next, we'll create the rest of the torso and make some legs.



The character so far.

Next: Torso, legs, and feet

Previous: Finishing the head

Back to Index

Retrieved from

["http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Neck_shoulders_and_arms"](http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Neck_shoulders_and_arms)

- This page was last modified 15:48, 27 July 2006.

BSoD/Introduction to Character Animation/Torso legs and feet

From BlenderWiki

< BSoD | Introduction to Character Animation

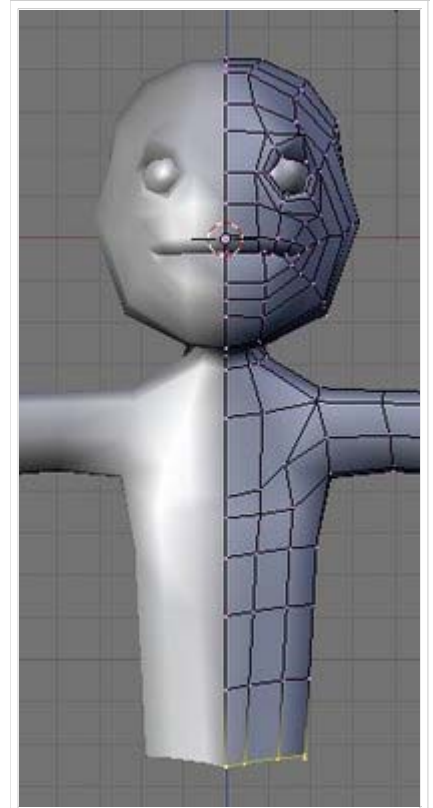
Contents

- 1 Creating the torso
- 2 Shaping the torso
- 3 Extruding the legs
- 4 Creating the feet
- 5 The finished model, with .blend file

Creating the torso

The bulk of the torso is very easy - it's just a couple of extrudes. The trick is to shape it later on to give it some mass, so it doesn't look like a square chunk.

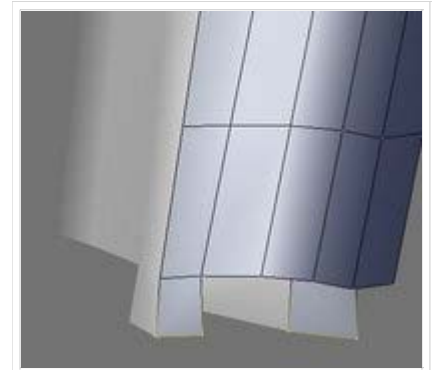
- Extrude the lower edge loop several times to form the torso. Here, I've extruded three times (*Extruding the torso*).



Extruding the torso.

Now we have to make room for the legs. It gets tricky to explain, just follow the images:

- Extrude the front-most edge and the back-most edge once (*Making room for the legs 1*).



Making room for the legs 1.

- Extrude the remaining vertices in the original loop once - including one of the vertices you just extruded from (*Making room for the legs 2*).



Making room for the legs 2.

- Extrude the front-most and back-most edges once more (*Making room for the legs 3*).



Making room for the legs 3.

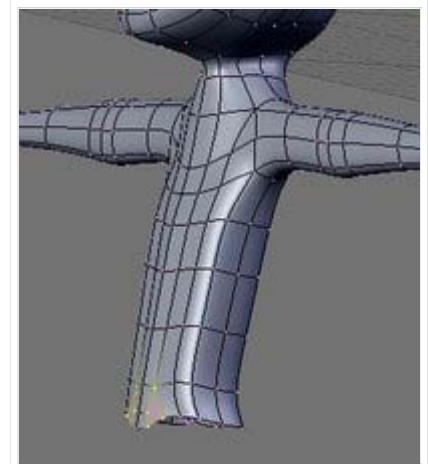
- Connect the newly extruded edge loops with faces - one in the front (*Making room for the legs 4*) and one in the back. If you get a message saying "Error: The selected vertices form a concave quad", try moving the vertices around a little, and see this page for why it happened: [BSoD/Introduction_to_Character_Animation/Concave quads](#)





Making room for the legs 4.

Shaping the torso

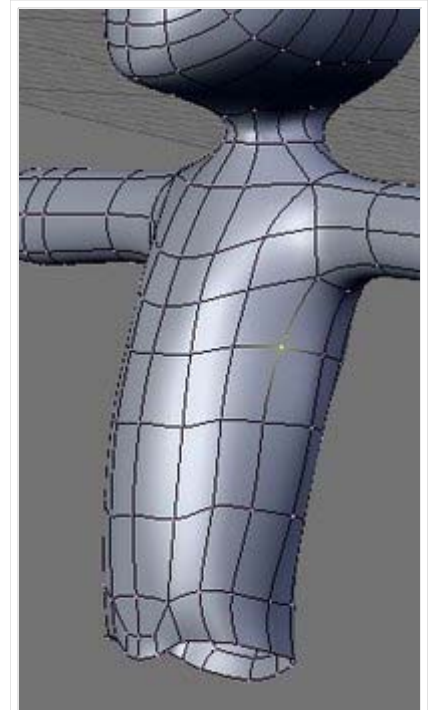
Now it's time to shape the torso. Remember to move the view around a lot, use perspective view and ortho view (NumPad 5), and use proportional editing (O). Here's what my character looked like before shaping:



Before shaping the torso - looks like a block of cheese!

And after a few mins of shaping . . . which involved about 150 vertex moves with RMB  - G - LMB .

It takes time!: Shaping the mesh takes a long time. It takes a lot of practice to figure out which views work best, when to use proportional editing (or when to turn it off), and how far to move vertices. The more time you spend on it, the better you'll get and the faster it will go next time.

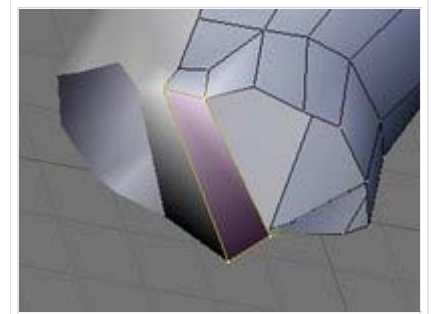


After shaping the torso - maybe cheesy, but not like a block . . .

Extruding the legs

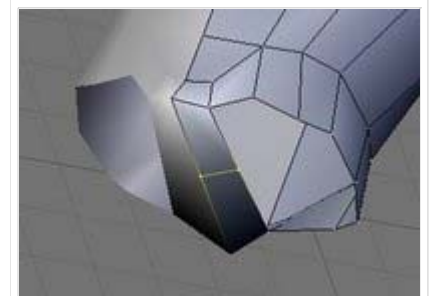
We have to close off the bottom of the torso before extruding the legs.

- Make a face connecting the front and back, as in *Closing off the bottom of the torso.*



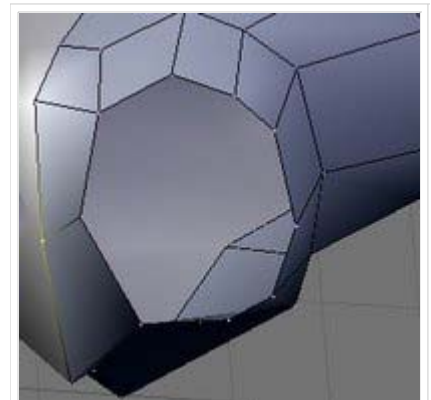
Closing off the bottom of the torso.

- Loop-cut (**Ctrl** **R**) the new face, as in *Loop cutting the bottom of the torso.*



Loop cutting the bottom of the torso.

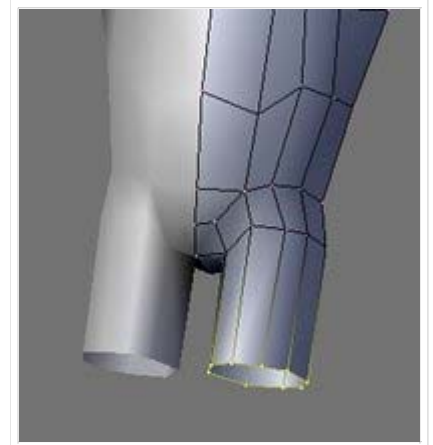
- Now shape the root of the leg so it's a little more circular (*Shaping the root of the leg.*).



Shaping the root of the leg.

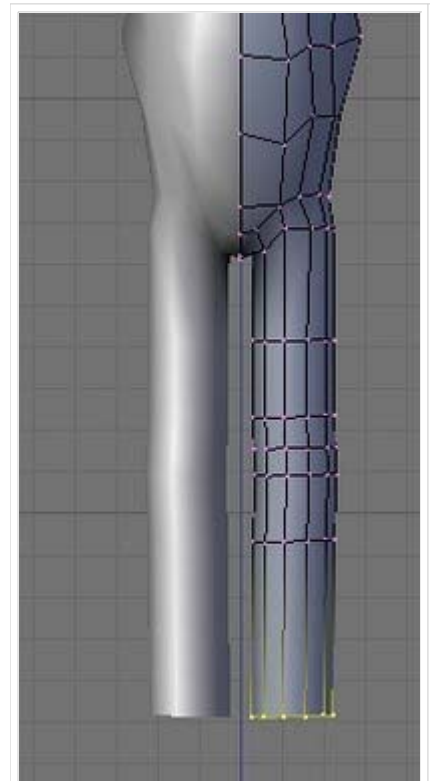
Extruding the legs is pretty straightforward. Making the feet is a little different from the hands - mostly because of the 90 degree angle the feet make with the legs.

- Extrude the vertex ring at the root of the leg, and shape it so the vertices are more or less on the same plane (*Extruding the leg 1*).



Extruding the leg 1.

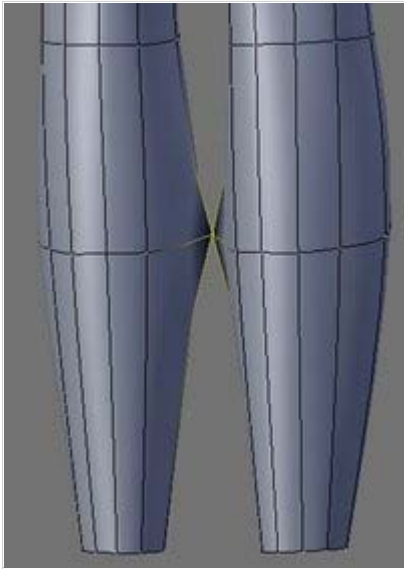
- Extrude the legs some more. Make sure you have three edge loops close together for the knees so that the leg will bend better when animating (*Extruding the leg 2*).



Extruding the leg 2.

- Spend some time to shape the legs (*Shaping the legs*).

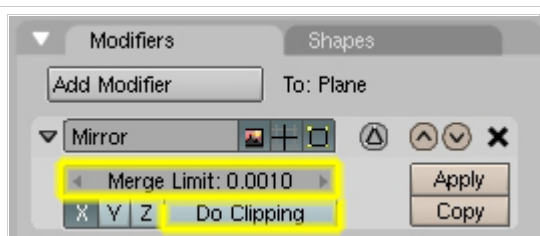
Problems with fused vertices when shaping: When shaping the legs, you might run into a problem like this, where the vertex you're moving suddenly snaps to the plane of the mirror:



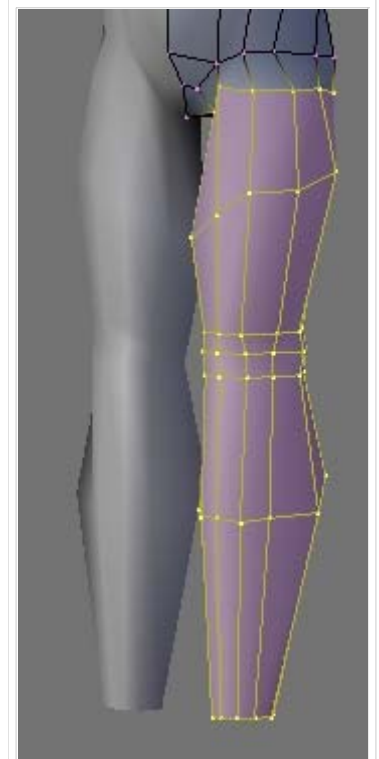
Fused vertices.

This happens because *Do Clipping* is enabled. The vertex I was moving in this case got too close to the mirror plane, so *Do Clipping* thought it should be snapped to the plane. There are two ways to fix this:

1. Turn off *Do Clipping*, move the vertex where you want it to go, then turn *Do Clipping* back on again.
2. Keep *Do Clipping* on, but decrease the *Merge Limit* value. This value determines how close a vertex can get to the mirror plane before being snapped to it. If this value is zero, the vertex has to be *right* on the plane for it to be snapped.

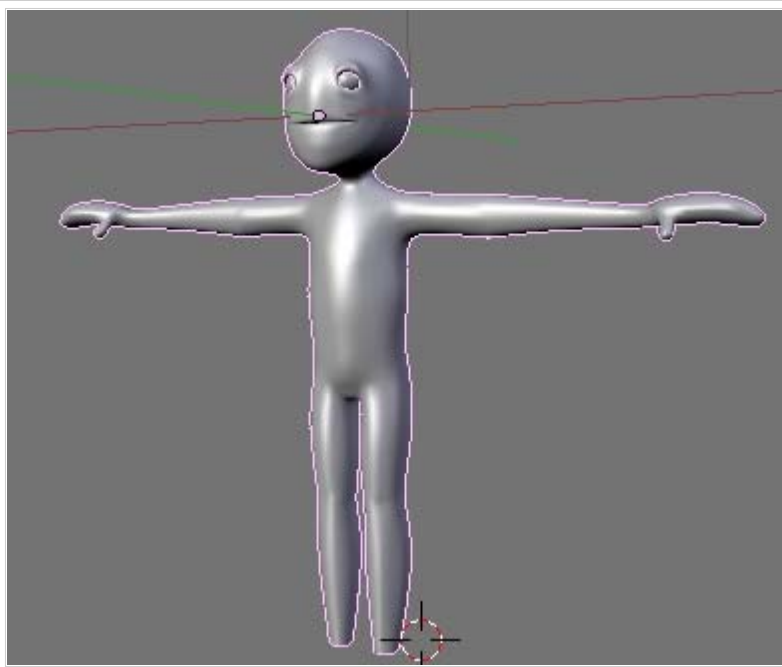


Mirror modifier panel.



Shaping the legs.

We've got legs! Here is the character so far:



The character so far.

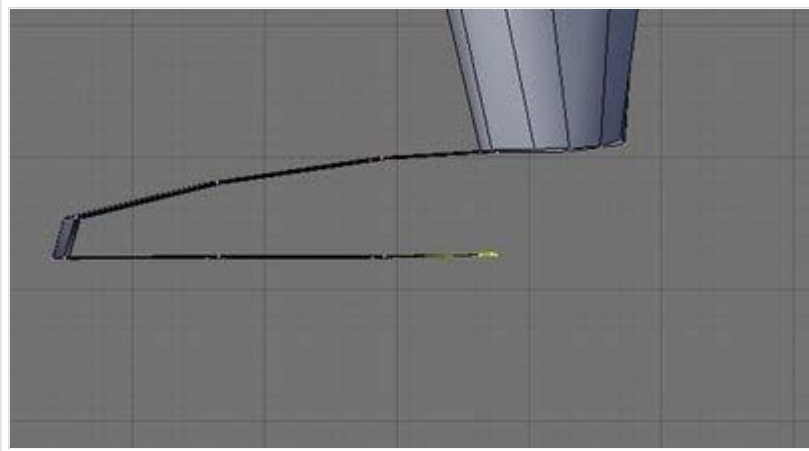
Creating the feet

- Select the three front vertices by the ankle and extrude them. *Extruding the feet 1* shows the extrusion from an oblique view, but it's probably easiest to do the extrusion in side view (NumPad 3).

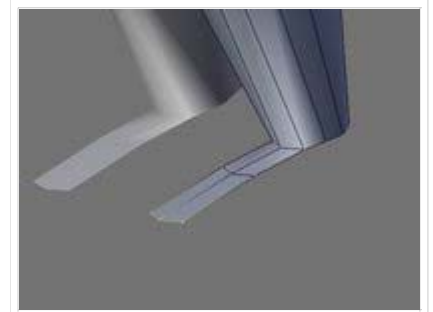


Extruding the feet 1.

- Key
 ext
 the
 thro
 ver
 as
 in
 in
 Ext
 the
 feet
 2
 and
 Ext
 the
 feet
 3.



Extruding the feet 3.

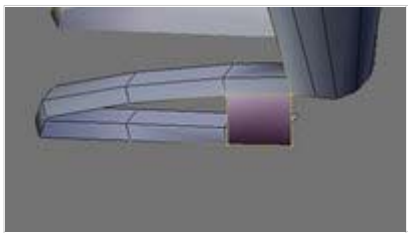


Extruding the feet 2.

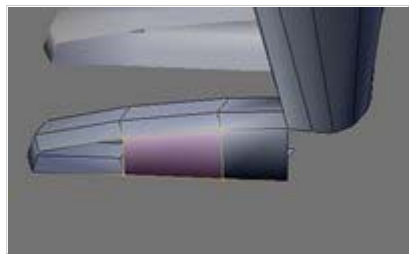
Note,
 in
*Extruding
 the
 feet
 3,*
 I've
 extruded
 the
 vertices
 along
 the
 sole
 of
 the
 foot
 so
 that
 they
 more
 or
 less
 line
 up
 with
 the
 vertices
 on
 the
 top
 of
 the
 foot
 where
 it
 meets
 the
 ankle
 (you
 may

have to click on the image to get a larger view).

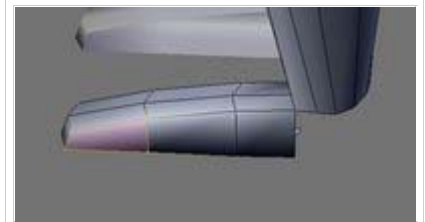
- Now start filling in faces on the feet by selecting four vertices at a time and hitting **F** to make a face. The sequence of 6 images below shows this process.



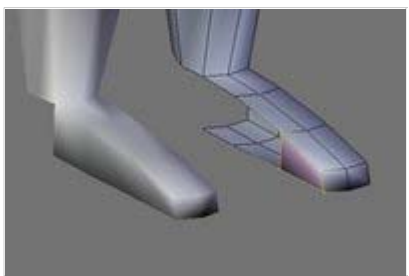
Faces on the feet 1.



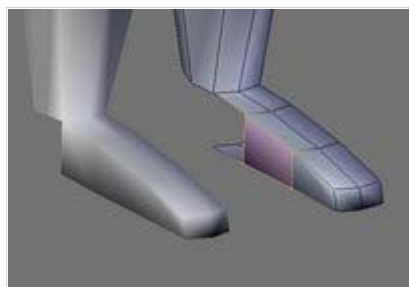
Faces on the feet 2.



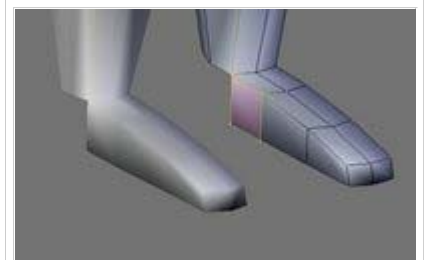
Faces on the feet 3.



Faces on the feet 4.

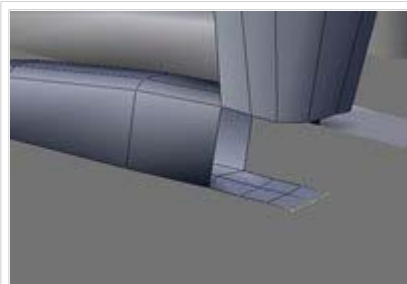


Faces on the feet 5.

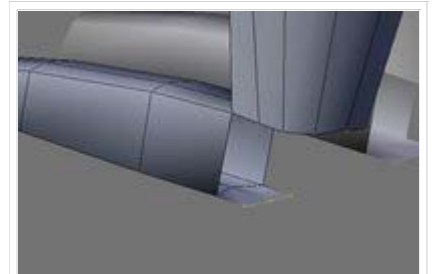


Faces on the feet 6.

- Extrude vertices from the sole back to make the heel (*Forming the heel 1* and *Forming the heel 2*). Each extrude should line up with vertices in the leg, because we're going to make faces using these extruded vertices.

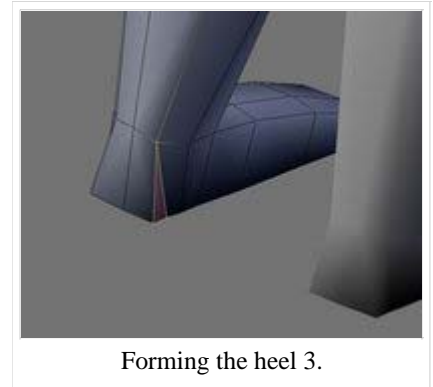


Forming the heel 2.



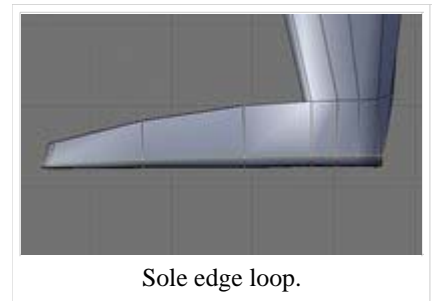
Forming the heel 1.

- Make faces with **F** to fill in the heel. Note that there's a single triangle *Forming the heel 3* - there weren't an even number of edge loops on the legs. That's OK, if it ends up being a problem later in animation we can fix it.

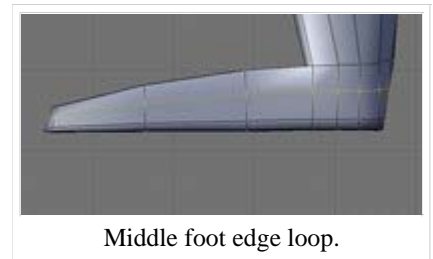


Now add edge loops to make sufficient vertices to shape the foot.

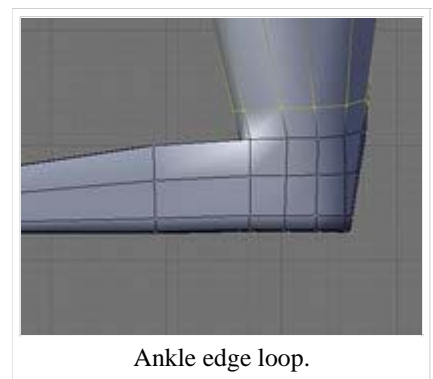
- **Ctrl R** to make an edge loop near the sole of the foot (*Sole edge loop*)).



- **Ctrl R** to make an edge loop around the middle of the foot (*Middle foot edge loop*). These edge loops will give the foot better shape.

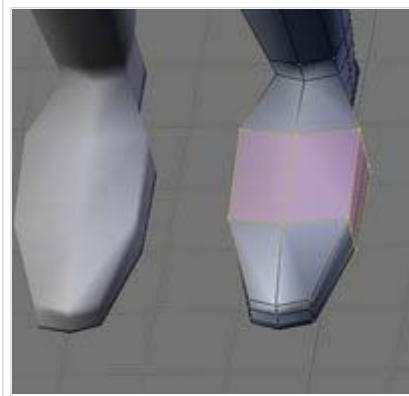


- **Ctrl R** to make an edge loop around the ankle. This will allow the transition from the lower leg to the foot to be a little sharper.

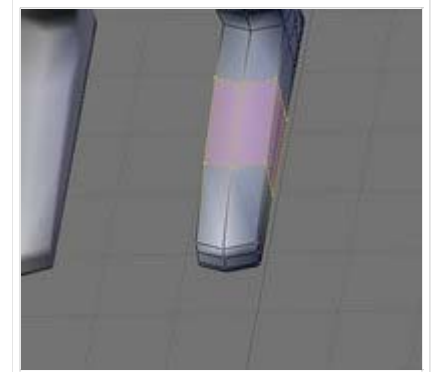


- Shaping the foot . . . you can of course shape however you want. You're probably beginning to develop your own style of shaping, these images are just the way I did it for this character.

- I selected a face loop (CTRL-ALT-MMB), S to scale, MMB to constrain to a single axis, LMB to confirm (*Shaping the foot 1 and 2*).



Shaping the foot 2.



Shaping the foot 1.

- Then I pushed and pulled vertices to shape the foot. This took me a little while to get right (*Shaping the foot 3*).

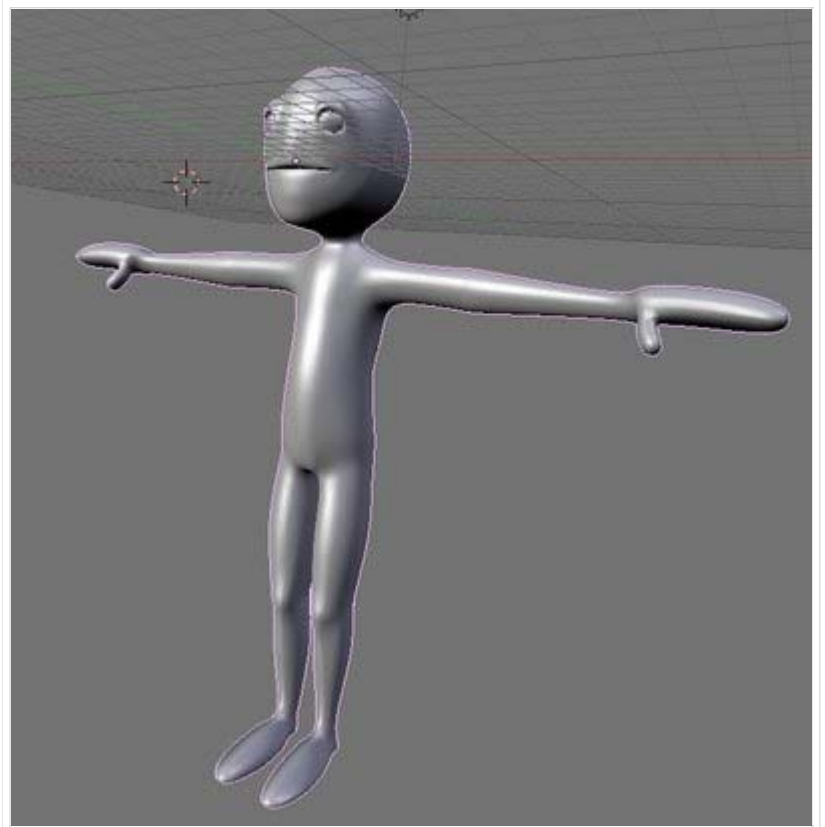


Shaping the foot 3.

The finished model, with .blend file

Congratulations! If you made it this far, you've probably learned a lot about how to model in Blender.

I spent some more time cleaning up and shaping the mesh. Here's the finished model, ready for materials, rigging, and animating:



The finished body.

Summary: We created the torso, legs, and feet in a series of extrudes and vertex movements.

Here's the file available for download:
Media:Tutorial_body.blend

Next: Lighting

Previous: Neck, shoulders, and arms

Back to Index

Retrieved from

["http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Torso_legs_and_feet"](http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Torso_legs_and_feet)

- This page was last modified 23:28, 25 July 2006.

BSoD/Introduction to Character Animation/Lighting

From BlenderWiki

< BSoD | Introduction to Character Animation

Contents

- 1 Lighting and rendering
- 2 Add a camera
- 3 Add a Lamp
- 4 Test Render
- 5 Rendering


Lighting and rendering

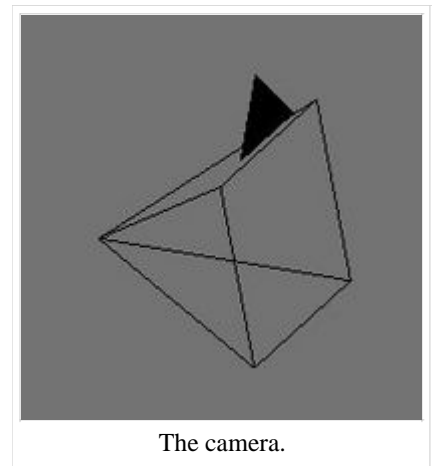
With a character modeled, now we can work on setting up the lighting and outputting the scene to an image ("rendering").

The default Blender scene has a cube, a lamp, and a camera. In the first step of this tutorial, we deleted all of that so we had a fresh scene. Now we'll add a camera to the scene so we can render it, and lamps for lighting.

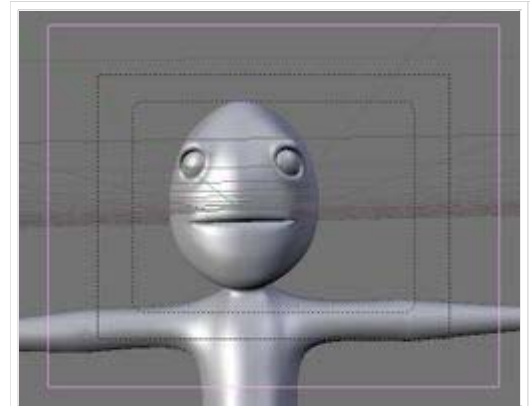
Add a camera

A **camera** is a special kind of object in Blender. The camera's view will determine the view of the final output, whether it's an image or an animation.


- Move the 3D Cursor somewhere where you can see it (LMB ).
- Switch to Front View (NumPad 1).
- Add a camera with Space >> *Camera*. This will add a camera wherever the 3D Cursor is, but it doesn't matter where that happens to be. We're going to move the camera. Note there's now another object in the scene (*The camera*). The square part is the front of the camera, and a black arrow points up so you know which direction the camera is facing.

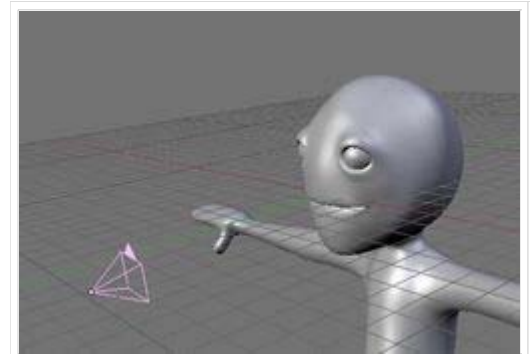


- Switch to **camera view** with `NumPad 0`. You know you're in camera view when you see the rectangular outlines near the center of the 3D Window (*Camera view*). The middle outline shows the boundary of the camera's view. The outer solid line is the camera object itself.




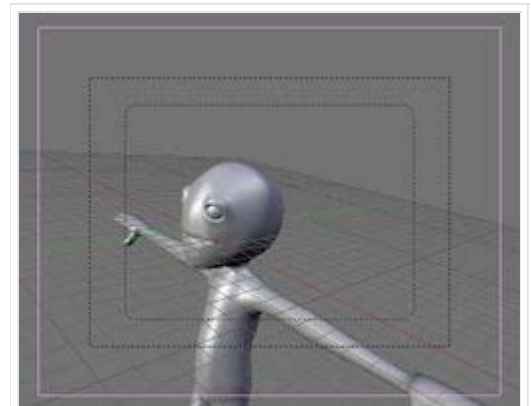
Camera view.

- To get out of camera view, use another view control (`MMB`  or one of the NumPad keys). Do this now, and move the view in the 3D Window to a view you want the camera to point at (*Moving to a new view*).




Moving to a new view.

- Snap the camera to this view with `Ctrl Alt NumPad 0`. This automatically puts you in camera view (*New camera view*).
- Select the camera object if it isn't already selected (`RMB`  on the outer solid line)




New camera view, using
`Ctrl Alt NumPad 0`.


- Move the camera by using **G**, and zoom the camera using **G**-**MMB** .

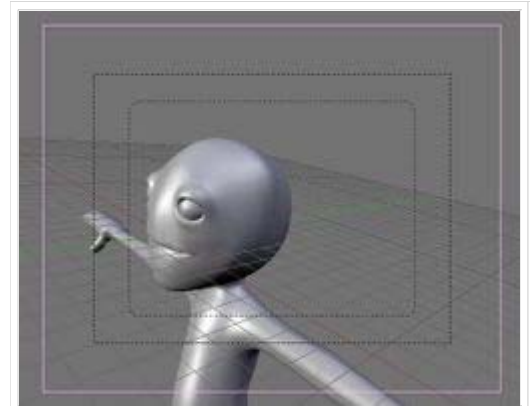
Camera keyboard controls

NumPad 0 - camera view.

Ctrl Alt NumPad 0 - snap camera to current view.

G- **MMB**  while in camera view to zoom

MMB  or **NumPad 0** views to exit camera view.



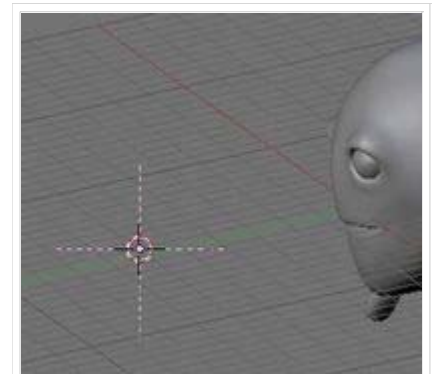
Same camera view, but zoomed with **G**-

MMB .

Add a Lamp

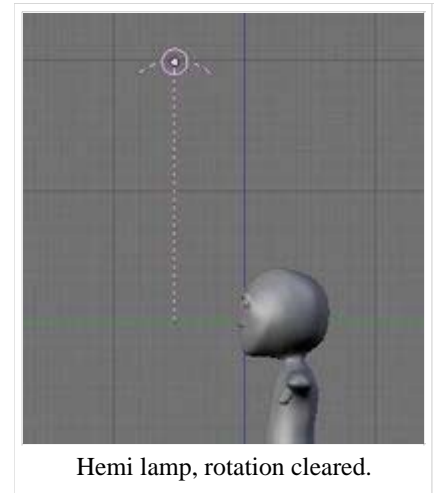
Now we will add a light to the scene. This will be a very simple lighting setup: a single lamp will be used.

- Switch out of camera view.
- Add a **Hemi** lamp at the 3D Cursor with **Space**>>**Lamp**>>**Hemi** (*The new Hemi lamp*). "Hemi" is short for "hemispherical", and simulates the uniform light of the sky. Since it simulates an infinitely large light source, the *position* of a Hemi lamp doesn't matter, only its *rotation*. You can read about this and other lamp types at [Manual/PartV/Lamp_Types](#).



The new Hemi lamp.

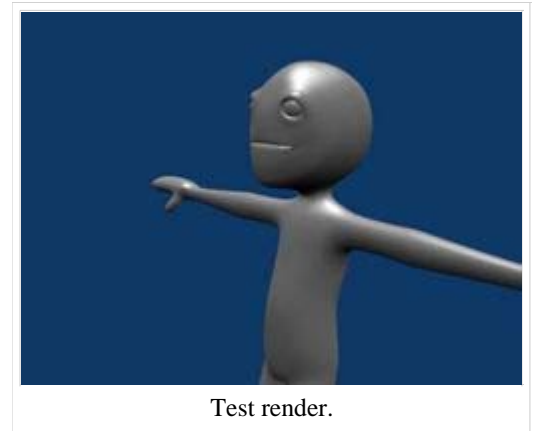
- With the Hemi lamp still selected, press **Alt R** to **clear rotation** (*Hemi lamp, rotation cleared*). This resets the lamp to point straight downward. Its direction is indicated by the long dashed line: now it's pointing straight down.
- You can move the Hemi lamp anywhere you'd like, I ended up moving it upward and out of the way. Remember for a Hemi lamp, only the rotation matters.



Test Render

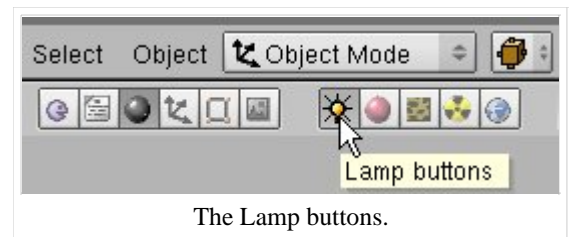
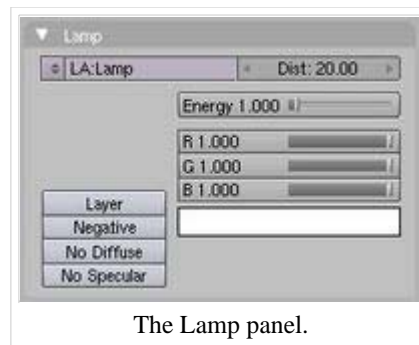
With at least one light in the scene, we can now do a test **render**. Rendering is the process of calculating how light bounces off of each object and turning those calculations into an image on the screen. Depending on the complexity of your scene, rendering can take a long time. For now, we'll just use the default settings for rendering.

- Press **F12** to render. A new window comes up, the render window, showing you what the final product looks like.
- If you'd like to save the image, press **F3** for the save window.
- Press **ESC** or close the render window to return to the main Blender window.

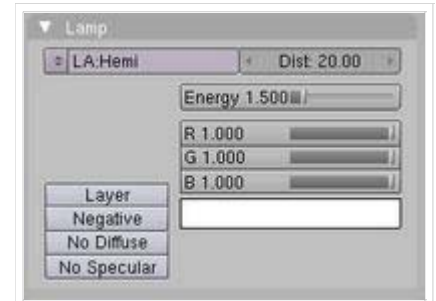


Let's make the lamp a little brighter.

- Select the Hemi lamp.
- In the Lamp buttons (*The Lamp buttons*), look for the Lamp panel.



- Name the lamp ("Lamp" by default) to Hemi (*Settings for the Hemi lamp*).
- Change the **Energy** to 1.5 either by moving the slider or by clicking on the number and typing it in. This will make the Hemi lamp a little bit brighter. Feel free to play around with the Energy settings. Even a small change (like from 1.0 to 1.5) is noticeable.



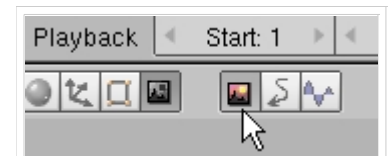
Settings for the Hemi lamp.

- Try out your lighting by rendering with **F12**.



Rendering

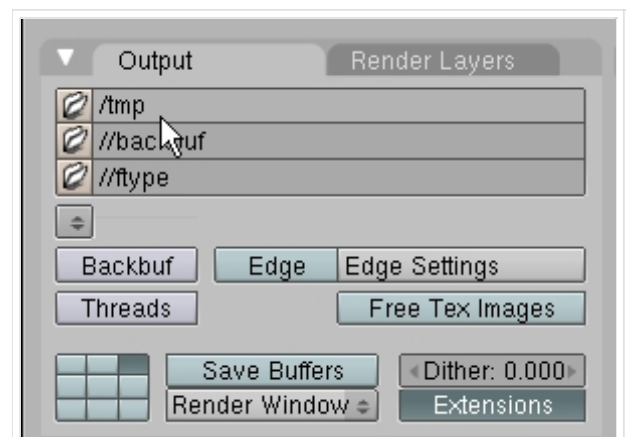
Settings for rendering can be found in the Render buttons.



Render buttons.

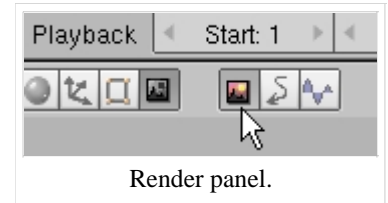
Of particular note are:

You can indicate where you want the renders to be save in the top row of the Output panel, although this is more for animations than for stills. You can always use **F3** in the render window to save the render to disk. A file browser window will open for you to choose where to save.



Output panel.

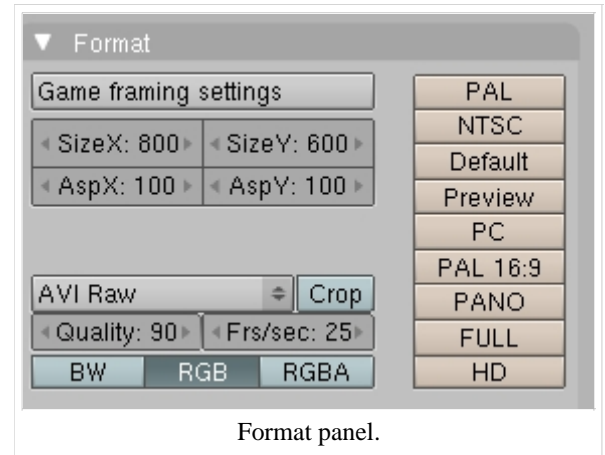
The OSA button stands for oversampling, also known as anti-aliasing. Anti-aliasing is a way of smoothing out edges and making the image look better at the expense of computation time. Disable this button to speed up rendering if you are just doing quick renders, but enable it for any image you want to look nice. The numbers below the OSA button indicate the number of samples: more samples results in a better image, but takes longer to render.




The Render button does the same thing as pressing **F12**.

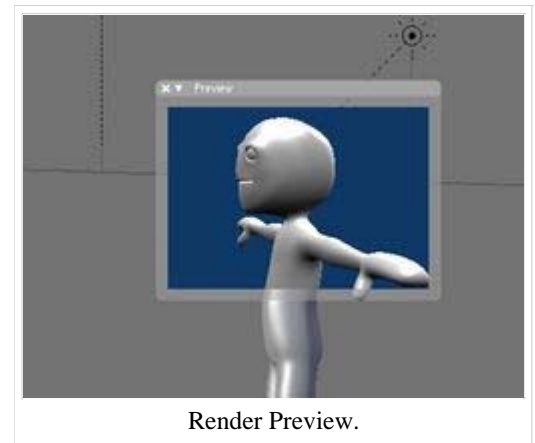
In the *Format* panel you can set the size of the rendered image with the *SizeX* and *SizeY* number boxes.

You can also set the image type from the menu. This is the type of image that will be saved when you hit **F3** after a render.



Instead of pressing **F12** all the time while experimenting to find the right lighting setup or materials, try using the Render Preview:

- In the 3D Window, press **Shift P**.
- Wait a moment for the preview to update (depends on the speed of your computer)
- Move the view with **MMB**  and watch the preview update.



Summary: We added a camera and some lights, and did some test renders to check the lighting. In the next section of the tutorial, we'll add materials to the skin and eyes to get rid of that ugly shiny gray look.

Next: [BSoD/Introduction_to_Character_Animation/Materials_and_textures](#)

Previous: [Torso, legs, and feet](#)

Back to Index

Retrieved from "http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Lighting"

- This page was last modified 23:30, 25 July 2006.

BSoD/Introduction to Character Animation/Materials and textures

From BlenderWiki

< BSoD | Introduction to Character Animation

Contents

- 1 Materials
- 2 To add a material to an object
- 3 Skin material color
- 4 Skin material shaders
- 5 Add skin textures
- 6 Multiple materials on a single object
 - 6.1 Create an eyeball material
 - 6.2 Create a Pupil material
- 7 Rotate the eyes outward
- 8 Change the inside of the mouth to a different material

Materials

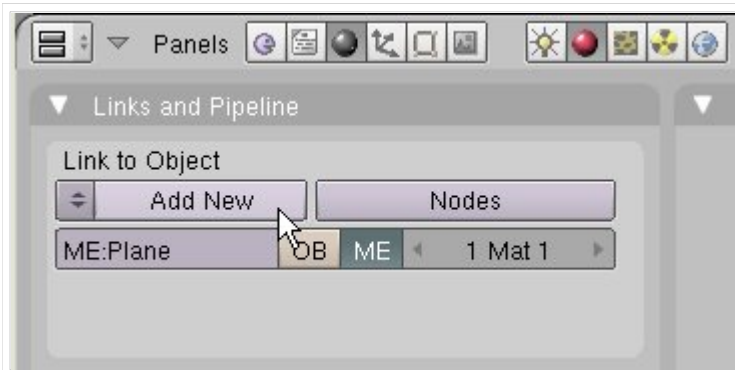
Next, we'll add **materials** to make the character more interesting. A material is a combination of colors, textures, and settings that tell Blender how an object reacts to light.

For details, see [Manual/PartIII/Materials_in_practice](#). Or, follow along by doing:

To add a material to an object

Let's add a material to the character mesh.

- Make sure the body is selected
- In the Buttons window, first click on the gray sphere in the header (or use the shortcut **F5**). These are the **Shader** buttons. There are so many Shader buttons that they won't all fit on one screen, so more buttons show up on the header.
- Click on the red sphere to get to the Material buttons, as shown in *Adding a material*. You may notice that there are other buttons next to the red sphere - these are for lights, textures, radiosity, and settings for the "world". More on these later.
- In the Links and Pipeline panel under the Materials buttons click the *Add new* button (*Adding a material*). Alternatively, click the arrows next to the text box that says *Add New* and select (you guessed it) *Add New*.



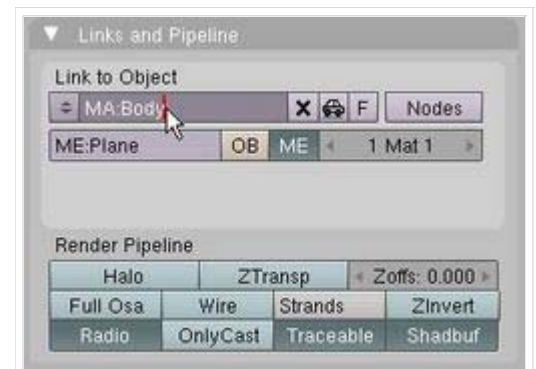
Adding a material.

As soon as you clicked Add New, a whole lotta stuff just popped up in the Materials window. These are the material controls, which we'll tweak to get just the right look.

Skin material color

Now we can start changing material settings.

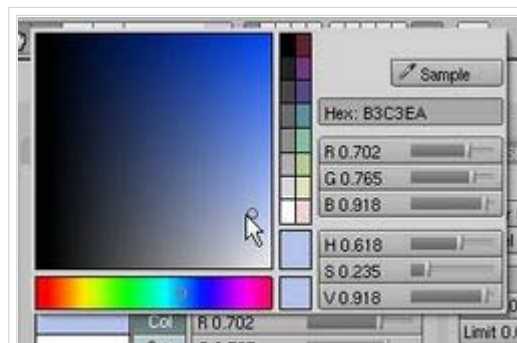
- Name the material by clicking in the text box in the Links and Pipeline panel. The default name is "Material"; I called this material "Body" (*Name changed to Body*).



Name changed to Body in the Links and Pipeline panel.

- Choose a color by clicking the (by default, white) box next to the Col button in the Material tab (as in *Click on the indicated box*). A color chooser pops up.

- Choose a color or type in a hex code for a precise color. Move the mouse away from the color chooser to accept the color change. I used a light blue (hex code B3C3EA). Note how the character mesh in the 3D Window turns the color of the material.



Color chooser.

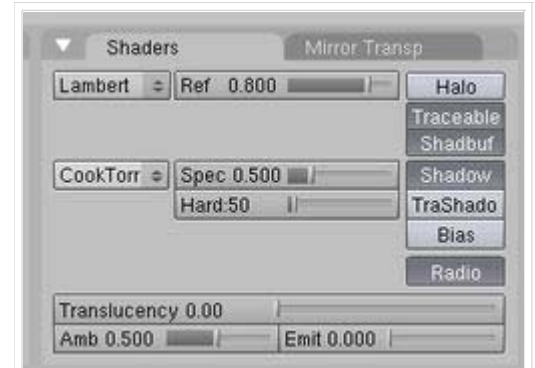


Click on the indicated box to change the material color.

Skin material shaders

Shaders determine how the material will react to light. You can adjust the shaders so you get a bright shiny material or a dull one.

- In the Material buttons, go to the *Shaders* panel (*Shaders panel*).
- Feel free to play with the different Diffuse and Specular shaders.



Shaders panel, default settings.

In this case, I made the following choices to make a sort of skin texture that was not too shiny and not too dull:

- Diffuse Shader: Oren-Nayer
 - Ref = 0.8 (Reflection)
 - Rough = 0 (Roughness)
- Specular Shader: WardIso
 - Spec = 0.1 (Intensity of specular)
 - rms = 0.255 (how focused the specular is)

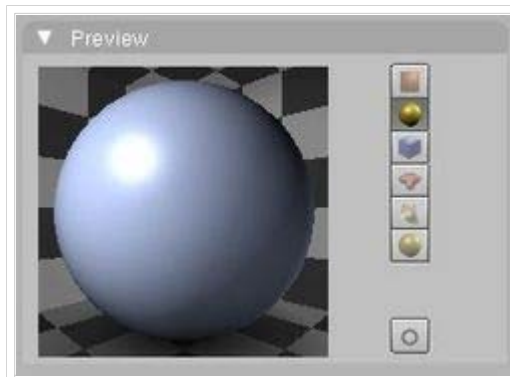
You can check out what the material looks like on different objects by pressing the buttons to the right of the Material Preview panel. Default is a Plane, but you can try out the other objects.



Shaders panel, new settings.



Material applied to a cube.



Material applied to a sphere.

For more on materials, check out this excellent shader discussion (http://www.newcottage.com/index.php?section=tutorials&subsection=tutorials/shading_1).

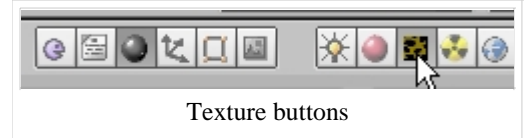
Add skin textures

A **texture** is an image that is associated with a material. A texture can affect different parts of a material - like specularity (how shiny it is), color, reflectance, or even how bumpy the surface is (a "bump map").

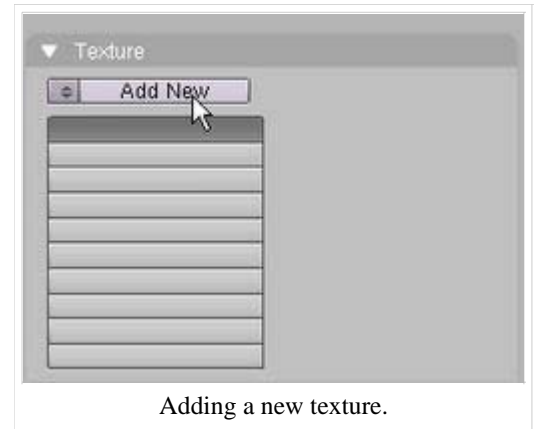
With a grayscale texture, lighter shades affect the material more, and darker shades affect the material less. In other words, the texture image affects the material depending on where the image is light and where it's dark.

We will add a texture to the skin material to make it look less like plastic.

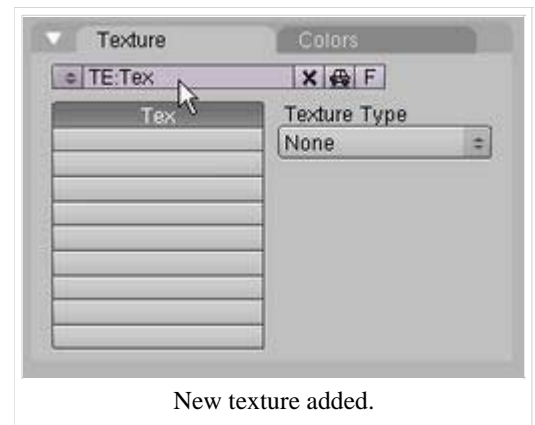
- In the Buttons window and in the Shading context, click on the button indicated in *Texture buttons* to get to the Texture buttons. You can also press F6 as a shortcut.



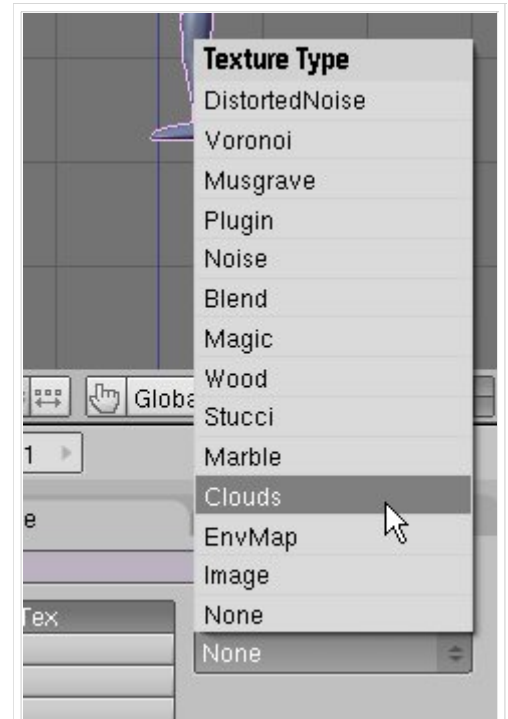
- Click on *Add New* (*Adding a new texture*) to add a texture.



A new texture, called "Tex", is occupying the first of 10 texture slots. This means that you can have up to 10 textures on any single material. Each one of those textures can be a different type, and can affect a different part of the material.

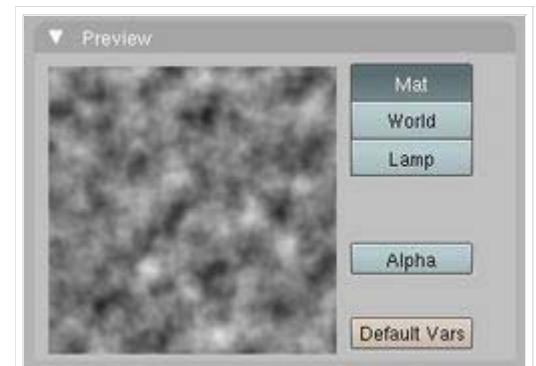


- Next, choose a type of texture. By default, the type is "None". Pretty boring.
- Choose *Clouds* (*Choosing a Clouds texture*).



Choosing a *Clouds* texture.

- The new *Clouds* texture is shown in the panel on the left side of the window (*The Noise texture*). Note also that a settings panel popped up as well, but we'll accept the default settings and leave that alone.



A *clouds* texture.

- Name this texture *Clouds* (we'll be adding two more cloud textures later).



Rename the texture to "*Clouds*".

- Now go back to the Material buttons (the red sphere, or F5). Note that in the Material Preview, the material is now blue with pink clouds (*Textures affect color by default*). That's because by default, a new texture affects the color of a material. Also by default, the color that is affected happens to be a really ugly magenta.

Here's what's happening: Wherever the Clouds texture we just added is lighter, there is correspondingly more magenta applied to the material. Conversely, where the Clouds texture is darker, there is correspondingly less color applied. Let's change the texture settings to make the texture affect the roughness or bumpiness of the skin instead of affecting the color.



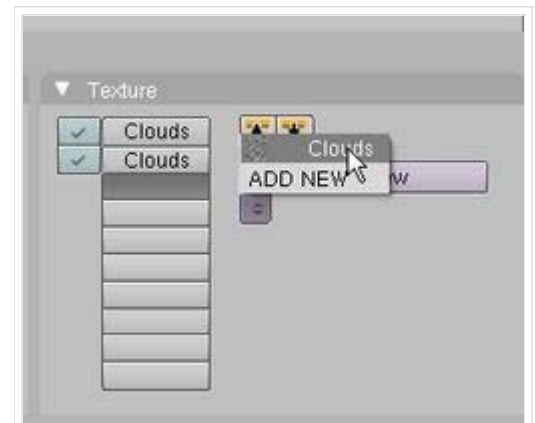
Textures affect color by default.

- Go to the Texture tab in the Materials buttons (*Texture panel in the Materials buttons*).



Texture panel in the Materials buttons.

- Select another texture channel button in the Texture panel, and assign the Clouds texture to that channel by choosing it in the menu to the right.
- Select a third texture channel button, and assign the Clouds texture to that as well. We've created one Clouds texture, and we'll reuse it three times for the skin texture. You'll see shortly why we're adding it three times to the material.

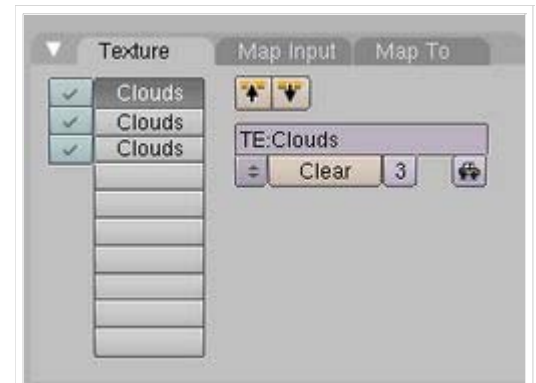


Adding the Clouds texture to other texture channels.

- Select the first texture channel. Make sure there's a check in the box next to the texture name. The check means that means the texture will be applied to the material, and the pressed button means that when we switch to the Map Input and Map To panels, we will be making changes to this texture channel.

The way to look at this is this:

- We created one Clouds texture in the Texture buttons. This texture can be used in any material.
- We added the Clouds texture to three separate texture channels in the Skin material. We could have put any texture in any one of those texture channels.
- We selected the first texture channel to make it active. We will now make changes to the settings of this texture channel to define how it will affect the material.



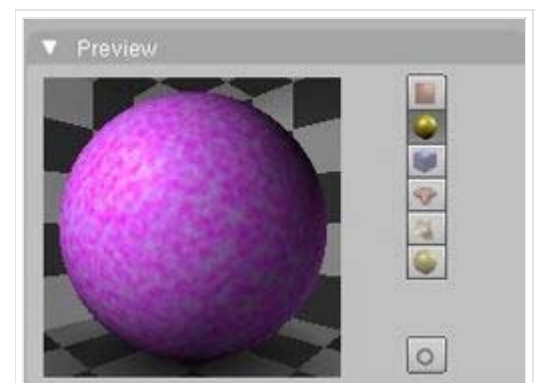
Texture panel in the Materials buttons.

- Now click on the Map Input tab, also in the Material buttons (*The Map Input panel*). These are settings that determine the relative size and direction of the texture, *before* it gets to the material. We want to shrink the pattern of the cloud texture so it will look better on the character.



The Map Input panel.

- In the "SizeX" number box, change the scaling factor to 5.
- Do the same for "SizeY" and "SizeZ". **Important:** larger numbers mean the pattern shrinks. Negative numbers mean the pattern gets larger. We are effectively shrinking the texture by a factor of 5 (*Result of scling the Clouds1 texture*).
- See [Manual/PartIV/Map_Input](#) for more details on the Map Input panel.



Result of scaling the Clouds1 texture.

- Click on the Map To tab (*The Map To panel, default settings*). You can see where the color of the texture is controlled: the color box that is by default pink. Don't bother changing this, because we're going to disable the texture from affecting the color.



The Map To panel, default settings.

- Click on the *Col* button to turn off Color.



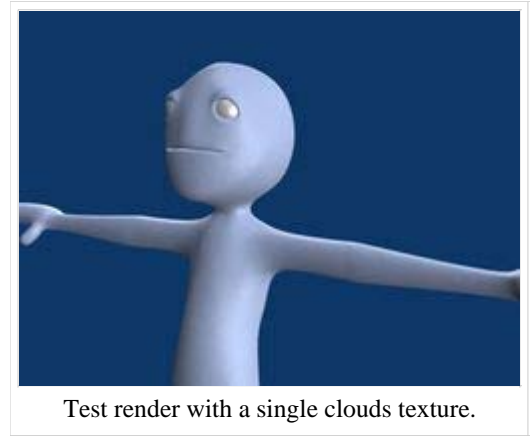
The Map To panel, new settings.

- Click on the *Nor* button to turn on Normals. The Preview reflects this change (*Preview of the clouds texture*). This makes the Clouds texture into a **bump map**: where the texture is white, there will be a raised bump. Where it is black, there will be a depression.



Preview of the clouds texture on the material.

- Try rendering to see the effect of this texture with [F12](#).



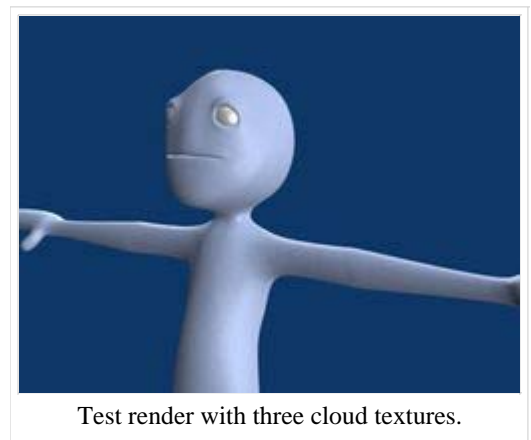
See [Manual/PartIV/Map_To](#) for more details on the Map To tab.

Now, change the second Clouds texture into a bump map.

- Go back to the Texture panel under the Material buttons.
- Select the second texture channel.
- In the Map Input panel, instead of changing SizeX, SizeY, and SizeZ to 5, change them to 25. That means that this channel will be shrinking the Clouds texture more than the first channel.
- In the Map To panel, deselect Col and select Nor, as before.

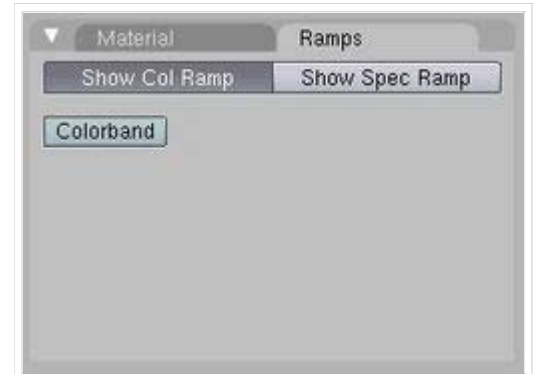
Change the third cloud texture in the same way, except change SizeX, SizeY, and SizeZ all to 50.

You should end up with a material that has three different sizes of the Clouds texture applied to it. Here's a render of it:



To finish the skin material, we'll add a color ramp. This affects how light reacts to the material around the edges, and will give the skin a nice look.

- In the Material buttons, click on the *Ramps* tab (*Ramps panel*). You can learn more about ramps here.



Ramps panel.

- Click on the *Colorband* button. Lots of controls pop up.
- By default, there are 2 colors on the colorband: black, on the left, and teal, on the right. This creates a gradient of color. There is also a gradient of transparency (or **alpha**): on the left is completely transparent (alpha = 0), and on the right is completely opaque (alpha = 1). There are currently two locations along the colorband, location 0 (the black) and location 1 (the teal).



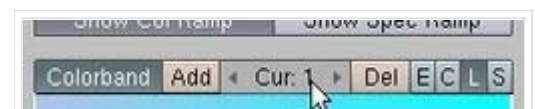
Showing the Colorband.

- Change the color of the first location from black to the same color as the skin material. I entered the hex code B3C3EA.
- Change the alpha (the "A" slider) to 1.0. This makes the location on the colorband completely opaque.



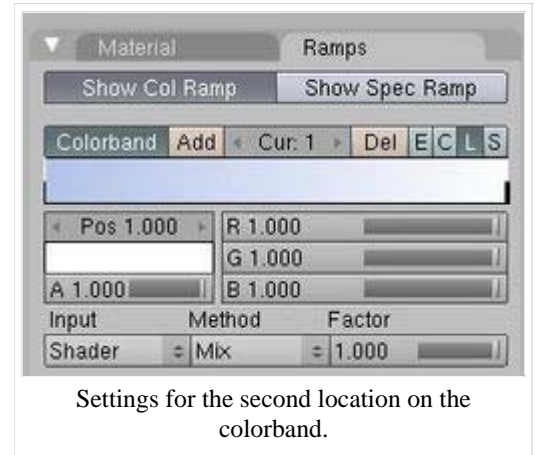
Settings for the first location on the colorband.

- Now switch to the second location on the colorband by clicking on the *Cur* button in the Ramps panel. This takes us to the location where we can change the right side of the colorband.

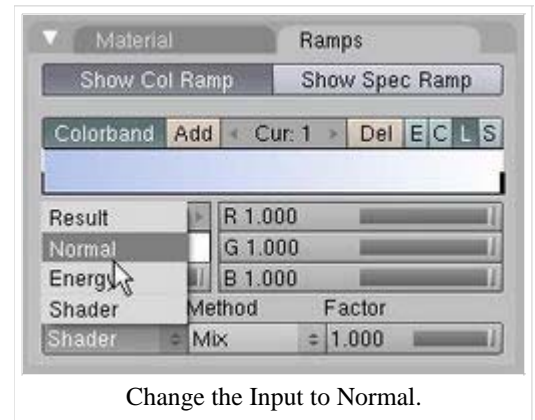


Switch to the next location on the colorband.

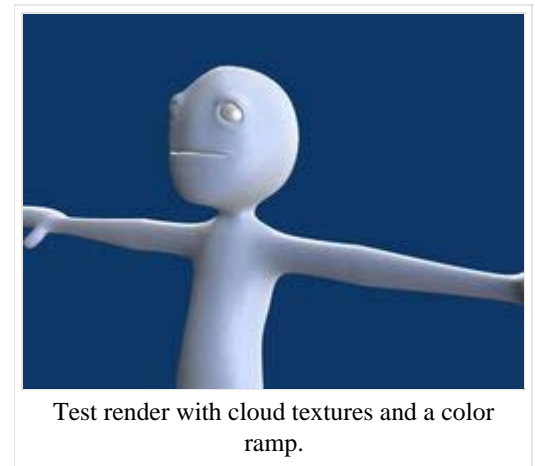
- Change the color from teal to full white.



- Change the Input to Normal in the Input menu (*Change the Input to Normal*). This makes the colorband react to the direction light is coming in on the character.



- Do a test render with **F12** to see the effect of the Color Ramp.

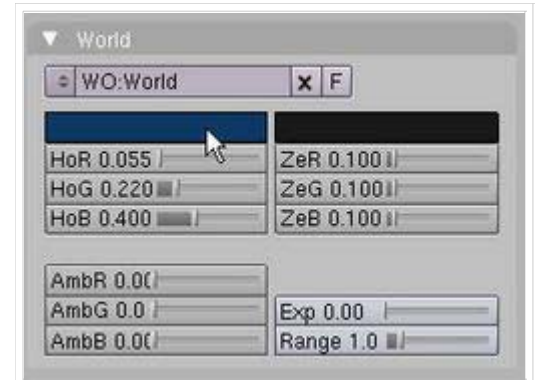


Let's change the color of the background. To do this,

- Go to the Shader buttons.
- Click on the **World** buttons (*World buttons*). Here, we can make changes to the background color of our render.



- Click on the color box to change the color (*The World panel*). I changed the blue to a light gray (hex# DCDCDC)



The World panel.

- Test render!



Test render after changing the World background color.

file up to this point:Media:Tutorial-body-in-progress.blend

Multiple materials on a single object

Now, let's add some materials to the eyes. To do this, we're going to create two new materials, and apply *both* of them to different parts of the same object: an eyeball material, and a pupil material.

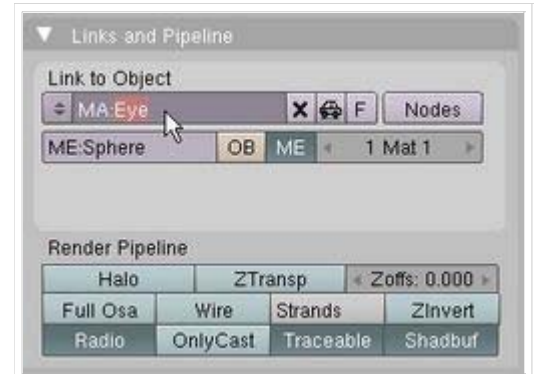
Create an eyeball material

- Select one of the eyes.
- In the Material buttons (**F5**), add a new material by clicking on the arrows next to the material name and choosing ADD NEW.



Add a new material to the eye.

- Name the new material "Eye"



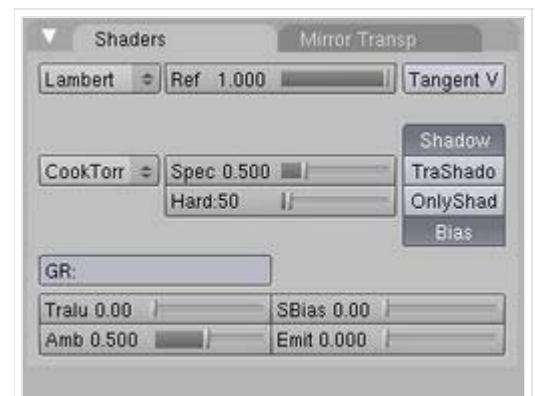
Rename the material to "Eye".

- Change the color to white.



Change the color to white.

- In the Shaders panel, I left the default settings alone, except for increasing the diffuse shader's Ref (Reflectance) to 1.0. (*Shader settings for the Eye material*). This makes the eye a bright white when rendered by reflecting all the light that comes to it.



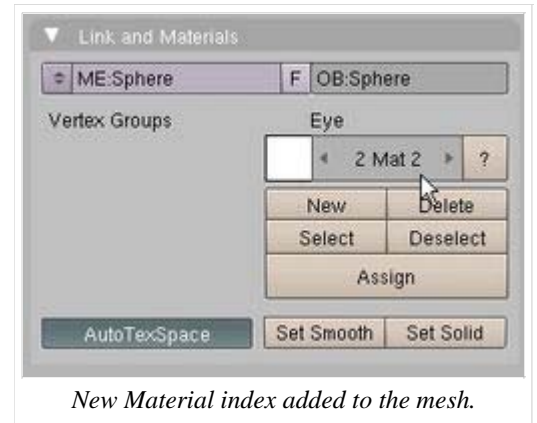
Shader settings for the Eye material.

Create a Pupil material

Here comes the tricky part. We're going to add a second material (a black Pupil material) to the eye . . . but only to the faces of the eye mesh that make up the pupil. The rest of the eye will retain the bright white Eye material we just assigned it.



Link and Materials panel in Edit Buttons, default.



New Material index added to the mesh.

BSoD/Introduction to Character Animation/Upper body armature

From BlenderWiki

< BSoD | Introduction to Character Animation

Contents

- 1 Rigging basics
- 2 For "real" armatures . . .
- 3 Apply the Mirror modifier
- 4 Add Armature
- 5 Extruding the spine
- 6 Extruding the arms symmetrically
 - 6.1 Naming the bones
 - 6.1.1 Bone naming conventions
- 7 Testing the rig and adjusting the arms for Auto-IK
- 8 Apply the Armature modifier to the Mesh

Rigging basics

Think about how you would animate a character. You definitely wouldn't want to move each vertex, frame by frame. Instead, we'll create a skeleton, complete with bones, and then move the bones in order to move the character. This process is called **rigging**.

In Blender, the skeleton is called an **Armature**. The idea is this:

- An Armature is composed of many Bones.
- Each Bone can have some vertices assigned to it.
- When you move a Bone, just the vertices assigned to it will move, and all the other vertices will stay still.
- The goal is to set up the Bones in the Armature so that neighboring Bones move neighboring vertices smoothly and naturally.

Just to give you a heads up, first we'll create the upper body armature, and once that's done we'll move to the lower body. Legs and feet take a little more work to do right, but hopefully by the time you get there you will be experienced from working on the upper body.

For "real" armatures . . .

This, by design, is a very simple armature to demonstrate the basics. If you are interested in making your character easier to control, see the Rigging BSoD project. There you will find lots of information on rigs suitable for the most demanding applications. Here we're just doing a simple rig to illustrate Blender's tools.

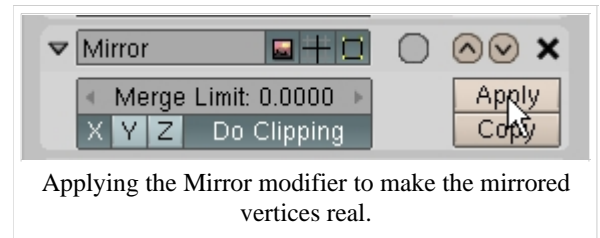
Apply the Mirror modifier

Up until now, we've been operating on one side of the mesh, and the other side has been automatically updated by the

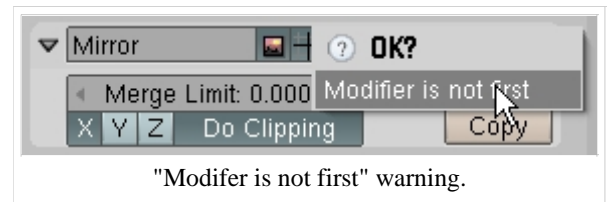
Mirror modifier. The Mirror modifier created virtual vertices on the other side of the mesh, and before we can attach an armature to the mesh we need to make those vertices real.

Important: Make sure the character mesh is the way you want it. We're about to leave the Mirror modifier behind for good. You can always change your mesh later, but it won't be as easy as using the Mirror modifier.

- With the character mesh selected, go to the Edit buttons
- In the Mirror modifier, click *Apply* (*Applying the Mirror modifier to make the mirrored vertices real*)



- You may get an error message like "Modifier is not first" (*"Modifer is not first"*). This refers to the stacking order of the modifiers; just click on the message and the Mirror modifier will automatically be moved to the top of the stack for you.



- The next message you see is "Applying will delete mesh sticky, keys, and vertex groups" (*Second warning message*). Click on that message to accept. We haven't made any such groups yet, so it'll be OK. We're about to make a LOT of vertex groups, though -- this is why we're applying the Mirror modifier now instead of later!!!



Add Armature

Time to add the Armature. In order to take advantage of symmetrical modeling tools, we want the center of the armature to be on the centerline of the character mesh. To do this,

- Select the mesh and enter Edit Mode.
- Select a vertex on the center line of the mesh.
- **Shift S** for the snap menu
- *Cursor to Selection* to snap the cursor to the selected vertex
- The cursor is now on the center line (*3D Cursor on centerline of mesh*).



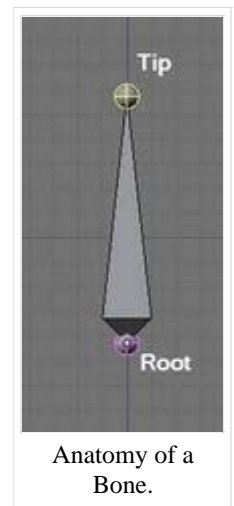
3D Cursor on centerline of mesh.

- Switch to Object Mode and Front View (NumPad 1).
- Add an Armature object with SPACE >>Add>>Armature.



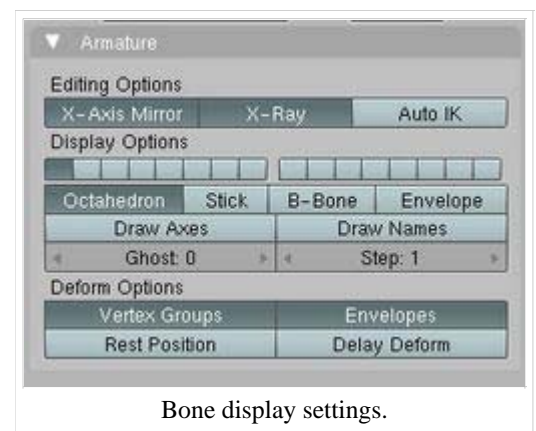
We've just added an armature, and it's composed of a single bone. Just like with meshes, Armatures have Object and Edit modes, and when you add an armature you automatically enter Edit mode. You can tell you're in Edit mode when you see yellow (selected) or pink (unselected) endpoints of bones.

- Each Bone has a ROOT (the blunt end) and a TIP (pointy end) (*Anatomy of a Bone*). At least, that's how they are represented in the default Octahedron draw type. We'll use other draw types later, where the root and tip aren't so obvious.



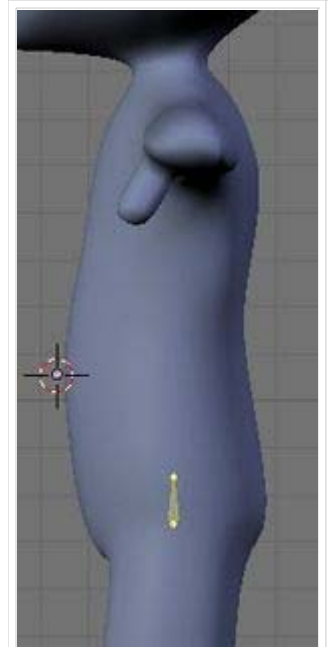
There are a couple of settings to make before we start building the rest of the rig. With the Armature still selected, these settings can be found in the Edit Buttons.

- **Important:** In the Armature panel under the Edit buttons, make sure *X-Axis Mirror Edit* is selected. This will allow us to symmetrically extrude bones to make the skeleton.
- Also make sure *X-Ray* is selected. This makes the bones visible through the mesh, and makes editing easier - we don't have to keep switching to wireframe view to see the bones inside the body (*Bone display settings*).



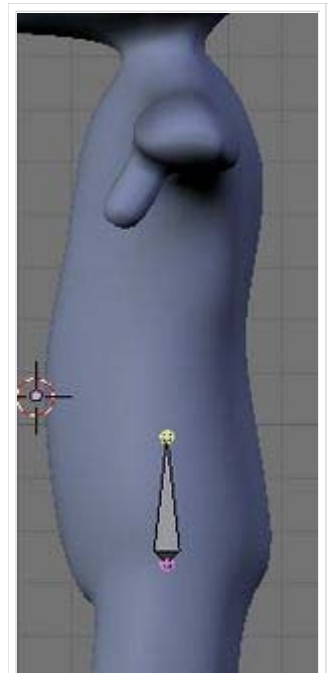
Extruding the spine

- Select the bone you added. You select an entire bone by clicking on the middle of it. You can also select a bone by selecting its root and then shift-selecting its tip.
- Switch to Side View and move the bone so it's closer to the pelvis and near the center of the body, roughly aligned with the neck.



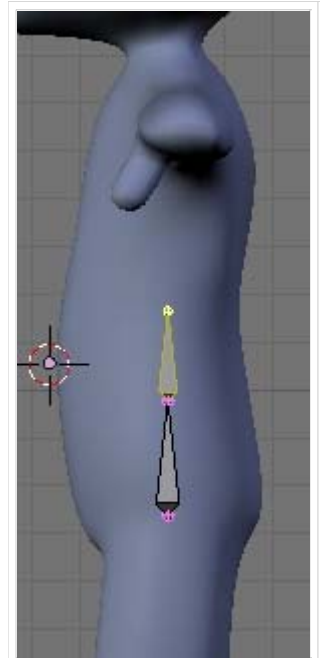
Moved bone closer to pelvis.

- Select just the tip.
- Grab the tip and move it up a bit to enlarge the bone in the Z-axis (**G** and **Z**) as in *Lengthening the spine1 bone*.



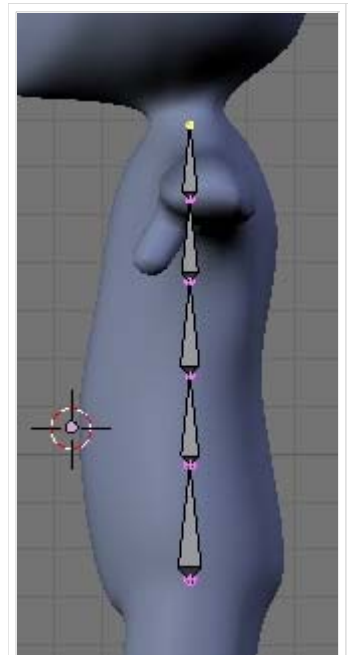
Lengthening the spine1 bone.

- With the tip still selected, do a constrained extrude along the Z axis with **E**, **Z** to make the second bone in the chain (*Extruding the spine 1*).



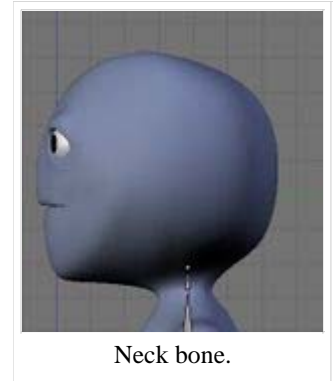
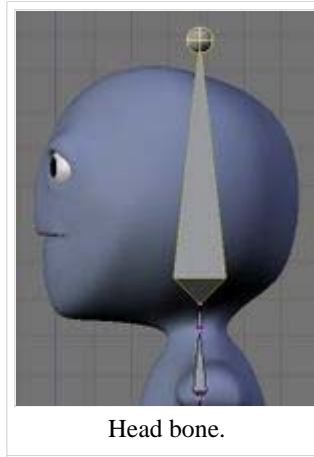
Extruding the spine 1.

- Do three more Z-constrained extrudes so you have a total of 5 spine bones (*Extruded spine*). The rib cage doesn't bend in a human, so this spine is a little more flexible than it would be in a real human.



Extruded spine.

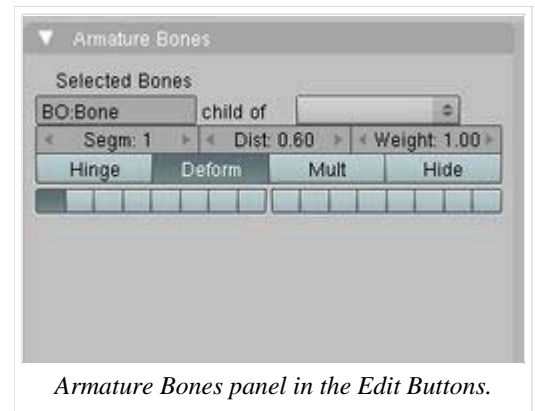
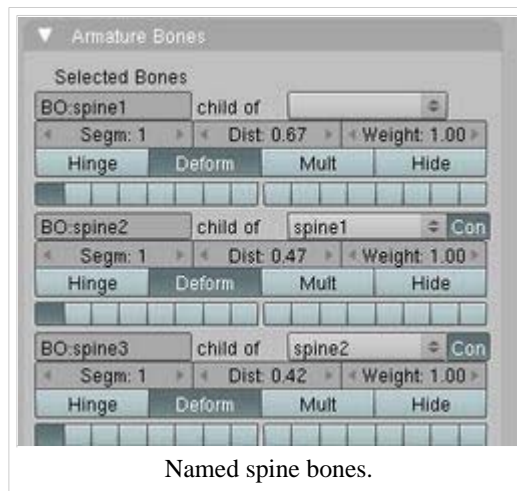
- Extrude a neck and a head bone. Note the head bone extends out the top of the head a little bit. That's so we can easily select the bone even if X-Ray is turned off for the armature.



We should name these bones. The names will appear in other parts of the interface, so it's useful to give them meaningful names now to avoid confusion later.

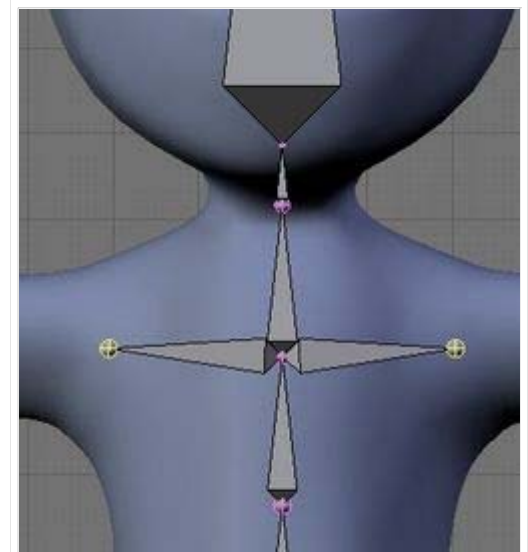
- With a bone selected, you can change the name of the bone in the Edit buttons Armature Bones panel (*Armature Bones panel in the Edit buttons*). I selected the bottom bone and changed the name to *spine1*.

- Select the each of the other bones in the spine and name them something meaningful. I named them, from the bottom, *spine1*, *spine2*, *spine3*, *spine4*, *spine5*, *neck*, and *head*. When they are all selected, they all show up in the Edit buttons (*Named spine bones*).



Extruding the arms symmetrically

- Go back to Front View (NumPad **1**)
- Select the tip of *spine4*.
- Something new: **Shift E** to symmetrical extrude and create a shoulder bone, as in *Symmetrically extrude the shoulder bones*. Symmetrical extrude only works if we have X-Axis Mirror Edit mode on, which we (conveniently enough!) turned on a couple steps back.



Symmetrically extrude the *shoulder* bones with **Shift E**.

- With the tip of the shoulder still selected, do another **Shift E** symmetrical extrude to make an upper arm bone (*Symmetrically extrude the upper arm bones*).



Symmetrically extrude the upper arm bones.

- Continue symmetrically extruding to make the lower arm . . .



Symmetrically extrude the lower arm bones.

- . . .the hand bone . . .



Symmetrically extrude the hand bones.

- . . .and two fingers bones.



Symmetrically extrude the finger bones (2).

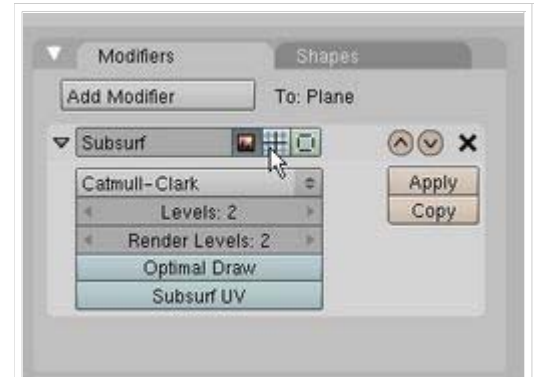


Symmetrically extrude the finger bones (1).

We have all the arm bones, now lets position them a little better. It's easiest to do this if we can see roughly where the vertices are in the mesh. Since we're working in the Armature's Edit mode, we can't view vertices in the mesh's Edit mode

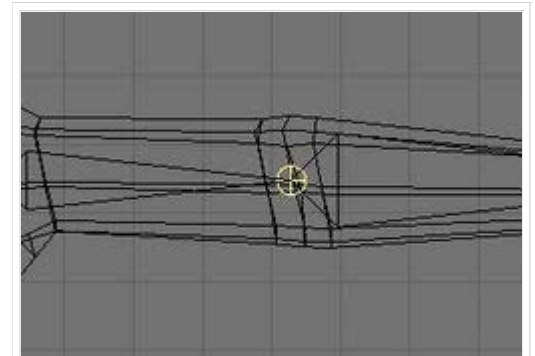
as well. Instead,

- **TAB** to exit the armature's edit mode.
- Select the character mesh.
- Turn off "Subsurf in interactive view" in the Subsurf modifier. This will allow us to see the true base mesh, and line up the bones accordingly (try Wireframe mode without doing this step to see what I mean).



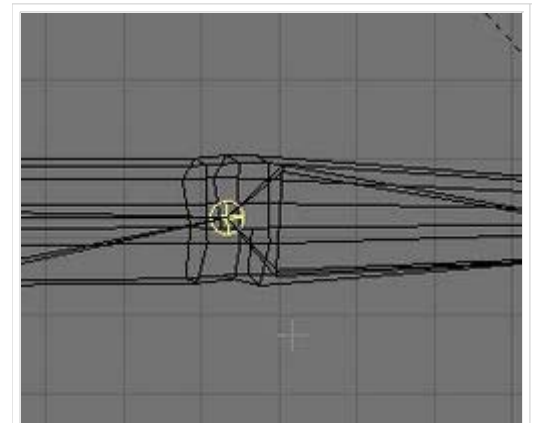
Disable Subsurf in the 3D Window.

- Switch to Wireframe view with **Z**.
- Select the armature again, and enter Edit Mode (**TAB**).
- Adjust the elbow so that it falls within the three elbow vertices in the mesh (*Adjusting the elbow from Front View*)



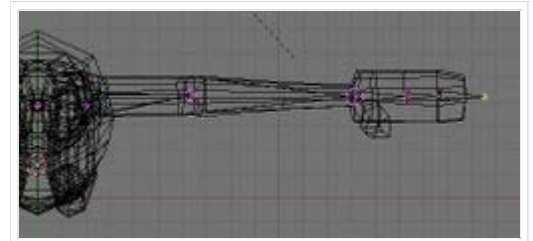
Adjusting the elbow, from Front View.

- Do the same from Top View (*Adjusting the elbow from Top View*).



Adjusting the elbow from Top View.

- While you're in Top View, make sure all the arm bones fall within the mesh (*Top view of arm bones*).
- **TAB**, select character mesh, turn Subsurf back on in interactive view, hit **Z** again for shaded mode
- Select the armature again and enter Edit Mode.



Top view of arm bones.

Naming the bones

Now for some naming. We have to be careful about naming these bones.

Bone naming conventions

In order to have some very handy X-Axis mirror tools work, we need to name symmetrical bones something like "bone.L" for the left bone, and "bone.R" for the equivalent right bone.

You could also use "bone.l" and "bone.r", or "Left.Bone" and "Right.Bone" . . . the symmetrical tools are pretty smart that way **ONLY** as long as both symmetrical bones have the same naming convention.

To be consistent, I'm going to use:

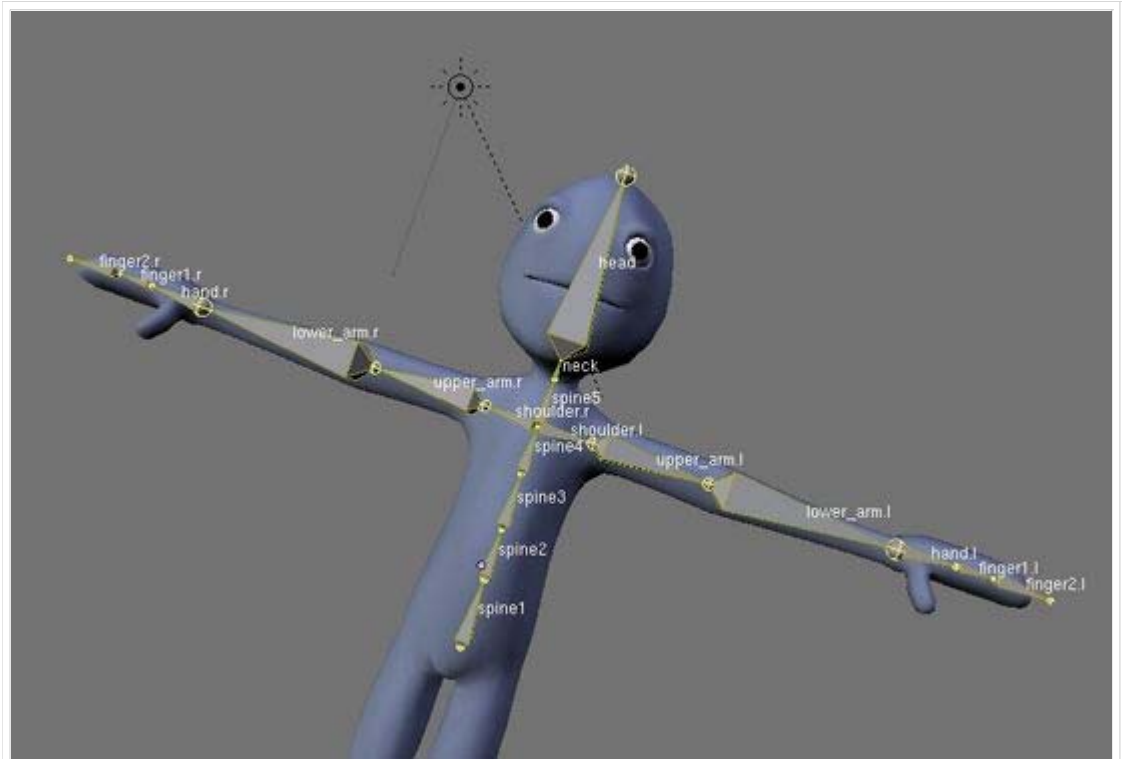
- All lowercase letters for the bone names, including "l" and "r" for left and right
- Two-word bones will have an underscore separating the names (upper_arm.l)
- The **LEFT** side is the **CHARACTER'S** left side. In front view, this often means you have to think a second before deciding which is left or right.
- Change the names of the bones to something that makes sense. Just make sure that the left and right bones are symmetrical. Under the Edit buttons (same place you turned on X-Ray and X-Axis Mirror Edit), you can turn on *Draw Names*. This will display the names of the bones, and is helpful to see if you missed naming any bones.

Bone names, below, is a screenshot of the names of the bones I used. You'll probably have to click on the image to view it full size and see the bone names. It's at a strange angle so that all bone names are clearly visible. The names I used were:

- shoulder.l
- upper_arm.l
- lower_arm.l
- hand.l
- finger1.l
- finger2.l

and

- shoulder.r
- upper_arm.r
- lower_arm.r
- hand.r
- finger1.r
- finger2.r

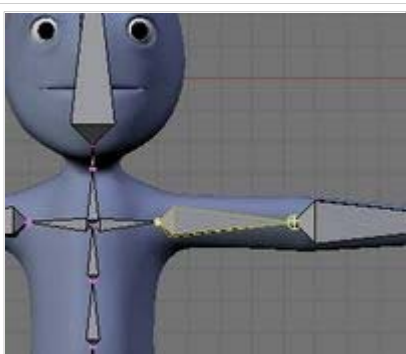


Bone names, upper body

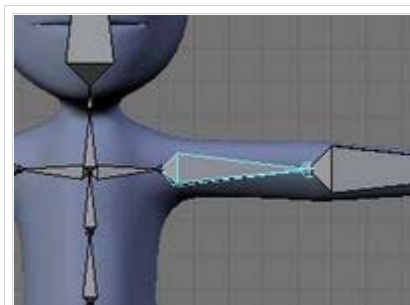
Testing the rig and adjusting the arms for Auto-IK

By now, you're familiar with Object mode and Edit mode. We're going to use another mode that's specific to armatures: **Pose mode**.

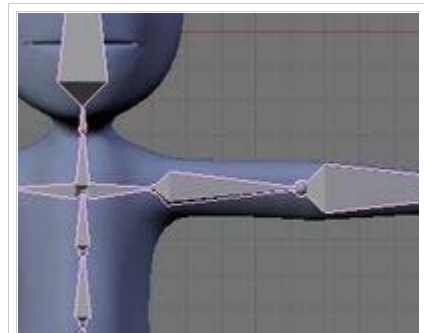
- With the armature selected, press **Ctrl** **TAB**. This essentially substitutes Object Mode for Pose Mode. In other words, you can now press **TAB** and switch between Pose mode and Edit Mode. If you need to get Object mode back, press **Ctrl** **TAB** again, and you can then switch between Object and Edit mode with **TAB**. You know you are in Pose mode when you select a bone and it is outlined in light blue.



upper_arm.l selected in Edit mode.



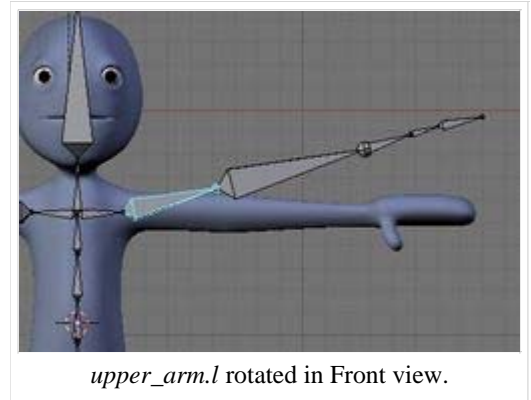
upper_arm.l selected in Pose mode.



Armature selected in Object mode (individual bones cannot be selected in Object mode).

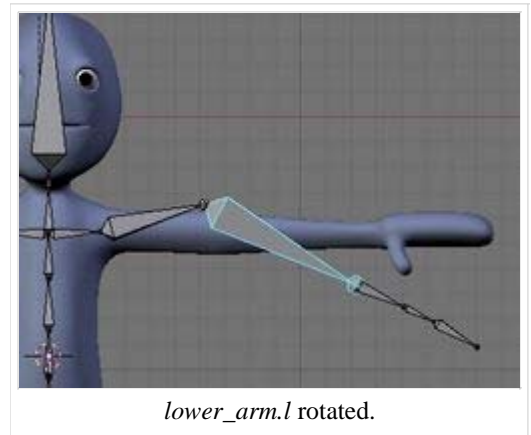
For armatures, **Edit mode** is used to construct the armature. **Object mode** is to move the entire armature as a whole. **Pose mode** is used for, well, posing. In Pose mode, you can grab, rotate, and scale each bone individually.

- Try selecting the upper arm in Pose mode and rotating it in Front view. Notice how all bones "downstream" of it rotate as well.

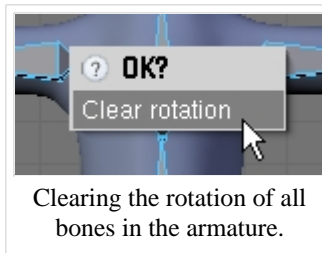


- Now select the lower arm and rotate it in Front view. The upper arm stayed in place, but the lower arm and everything "downstream" rotated. This is the essence of **parenting**.

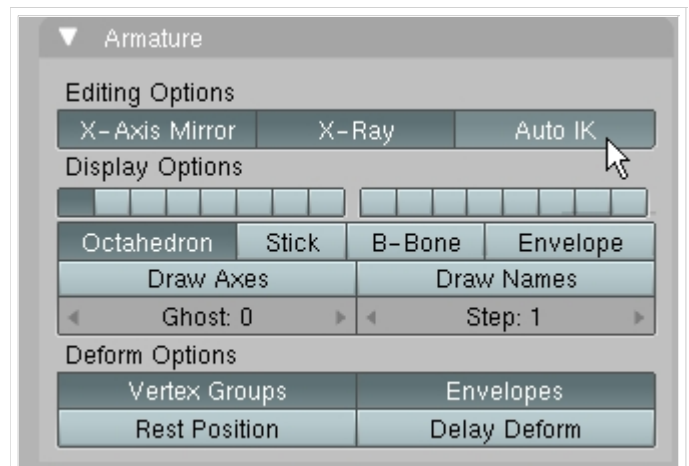
That is, the upper arm is the **parent** of the lower arm. The lower arm is in turn the parent of the hand bone. Another way to say that is that the hand is the child of the lower arm. These parent-child relationships were automatically created when we extruded the bones. The extruded bone becomes the child of whatever it was extruded from. That's the reason we started from the lower spine and extruded upward, as well as starting at the shoulder and extruding toward the fingers.



- Clear the rotation of all bones by using **A** twice to select all, then **Alt R** to clear rotation. The bones are now reset to their original rotations. You'll end up using this command a lot, along with the related command **Alt G**, which clears location.

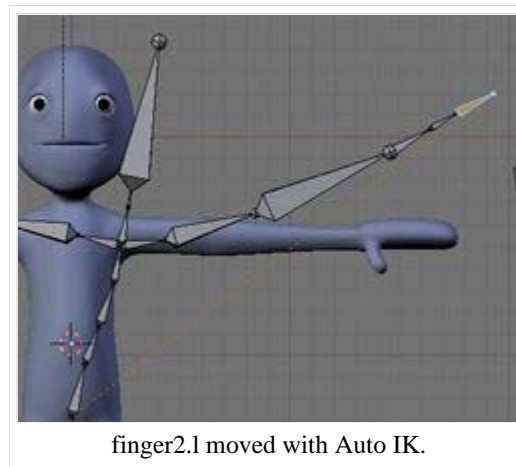


- Turn on *Auto IK* in the Armature panel, under the Edit buttons.



Turn on Auto IK in the Armature panel.

- Select the tip of the arm and move it with **G**. Note that it moves much differently now!

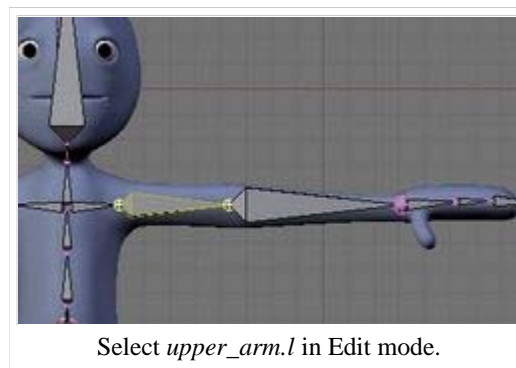


A little explanation: Forward Kinematics, or FK, is the way of moving bones that we first used. That is, rotate the upper arm, and its children (and children's children!) follow along. The opposite of FK is Inverse Kinematics (IK), where we move a child and the parents follow along. In reality, there is some fancy math going on in the background that tries to point the chain of bones toward the target. What's the target? For Auto-IK, it's whatever bone you have selected. In this case, the target is the *finger2.l* bone. What's the chain? It's the lineage of bones going all the way back to the great-great-great-(etc)-grandparent.

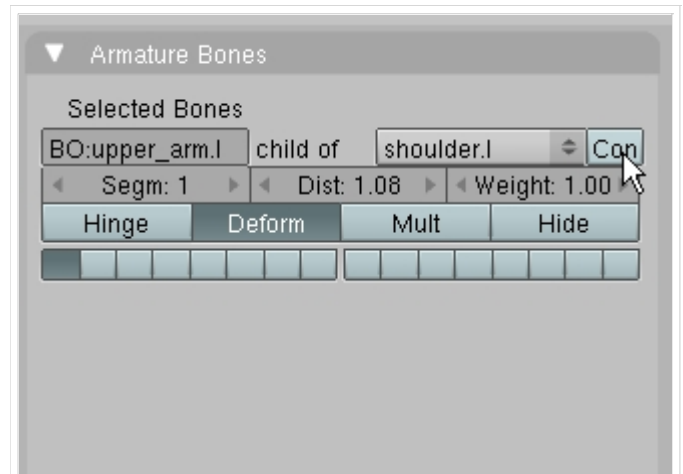
In our armature, when we moved the finger bone, all the bones in the chain tried to point to wherever we moved it. An orange line showed up, connecting the *finger2.l* bone to the *spine1* bone. The orange line points to the root of the chain: *spine1* is the highest parent of *finger2.l*, and the chain is everything between *spine1* and *finger2.l*.

It would be nice if the spine didn't move so much when we moved the arm. We'll fix this by essentially breaking the IK chain at the shoulder so only the arm moves and the spine stays still.

- Go into Edit mode of the armature with **TAB**. Note that even though you may have just moved some bones around in Pose mode, upon entering Edit Mode everything goes back to the way it was. In Edit Mode, you're viewing the bones as they are in **Rest position**, and once you go back out to Pose mode, your posed armature will return.
- Select the *upper_arm.l* bone in Edit mode.

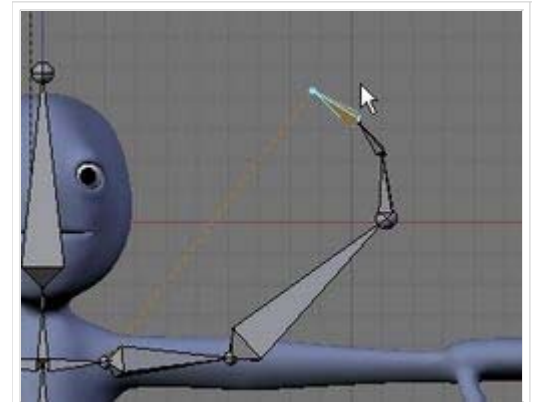


- In the Armature Bones panel under the Edit buttons, deselect the *Con* button. In this panel, the *child of:* menu indicates that this bone, *upper_arm.l*, is the child of *shoulder.l*. We want to keep that relationship, but we'd like to allow *upper_arm.l* to be disconnected from *shoulder.l* . . . and therefore break the IK chain. *Con* stands for Connected. By deselecting this button, we disconnected the *upper_arm.l* bone from the *shoulder.l* bone.



Deselect the "Con" button to disconnect *upper_arm.l* from *shoulder.l*.

- To test this new setting, switch to Pose mode (**TAB**).
- Reset the armature by pressing **A** twice to select all bones, **Alt G** to clear locations, and **Alt R** to clear rotations.
- Now, move the *finger2.l* bone again. Much different! The orange line now points to the root of the chain, which is the *upper_arm.l* bone.



Moving the *finger2.l* bone after breaking the IK chain at *upper_arm.l*.

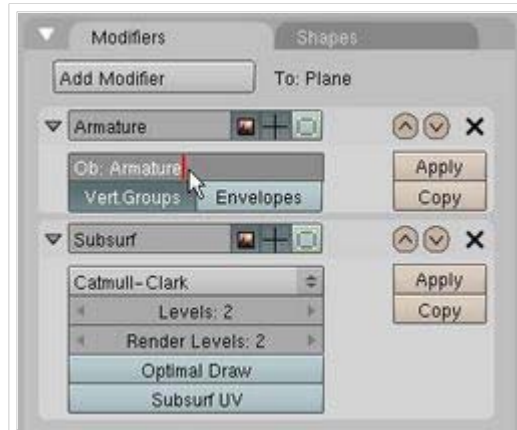
While we were able to extrude bones symmetrically, we have to make changes to the settings separately. To disconnect the *upper_arm.r* bone from *shoulder.r* in the same way,:

- Switch to the armature's Edit mode
- Select *upper_arm.r*.
- Deselect the *Con* button.
- Test the armature.

Apply the Armature modifier to the Mesh

Now that we have an armature, it's time to attach it to the mesh. To do this, we'll add an Armature Modifier to the mesh.

- Select the character mesh, in Object Mode
- In the Edit Buttons, choose *Add Modifier>>Arm*
- In the *Ob:* text box, enter the name of the armature object (*Enter the name of the armature object into the armature modifier*). The default armature name is, logically enough, "Armature". To double check the name of your armature, select the armature and look for the *OB:* text box in the Edit buttons, like in *Checking the name of the armature*.
- In the Armature Modifier, make sure only *Vertex Groups* buttons are selected. We will not be using *Envelopes* in this tutorial.



Enter the name of the armature object into the armature modifier, and make sure only Vertex Groups is selected (Envelopes should be unselected).



Checking the name of the armature (select it and check the name that shows up here).

Now Blender knows that we want the armature to affect the character mesh. Next, we need to tell Blender exactly what vertices to move when we move, say, *upper_arm.l*.

Summary: We added an armature object, and sequentially extruded bones to make an armature for the upper body. We made some changes to allow for the use of Auto-IK, should we choose to use it later on. Next up: weight painting!

Next: Upper body: weight painting

Previous: Materials and textures

Back to Index

Retrieved from

"http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Upper_body_armature"

-
- This page was last modified 08:36, 24 August 2006.

BSoD/Introduction to Character Animation/Upper body weight painting

From BlenderWiki

< BSoD | Introduction to Character Animation

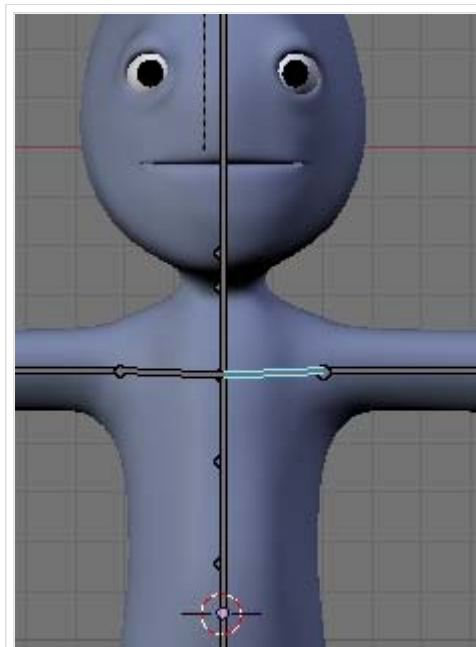
Contents

- 1 Weight Paint mode
- 2 Weight painting the upper arm
- 3 Tips for weight painting
- 4 Weight paint the rest of the arms
- 5 Weight painting the head and torso
- 6 Parenting the eyes to the head bone
- 7 Testing the rig

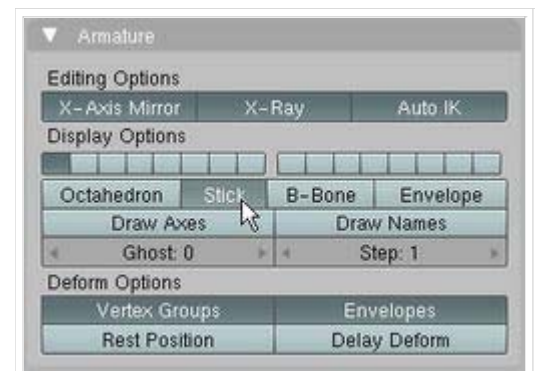
Weight Paint mode

We're going to tell Blender which vertices in the mesh to move whenever we move a bone in the armature. While we could manually choose vertices and assign them to groups, a more powerful way to do this is with Weight Paint mode.

- First, let's make the bones a little less obtrusive. Select the armature.
- In the Armature panel under the Edit buttons, choose *Stick* as the draw type. This changes the bones from bulky octahedrons (which are a useful shape for building armatures) into thin sticks that don't take up as much space on the screen.



Armature in Stick draw type.

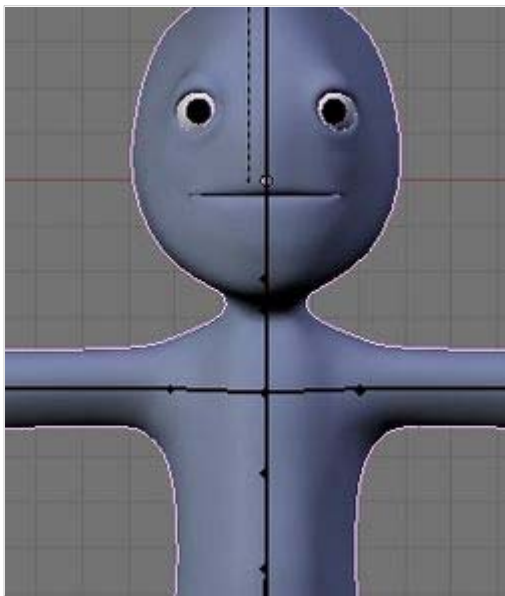


Change the armature draw type to Stick.

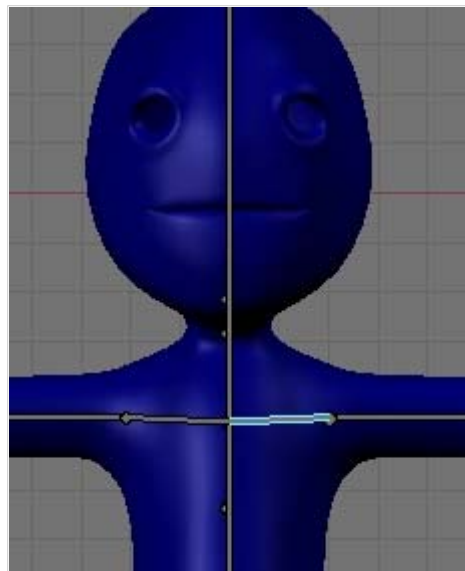
Note: In order to select bones in the armature while weight painting, the armature has to be in Pose mode.

- Make sure the armature is in Pose mode.
- Select the character mesh.
- Switch to the mesh's **Weight Paint** mode with **Ctrl TAB**. Similar to Pose mode for armatures, we've substituted Weight Paint mode for Object mode.

Notice the mesh turns dark blue. As you'll see, weight painting uses colors to represent information about groups of vertices.

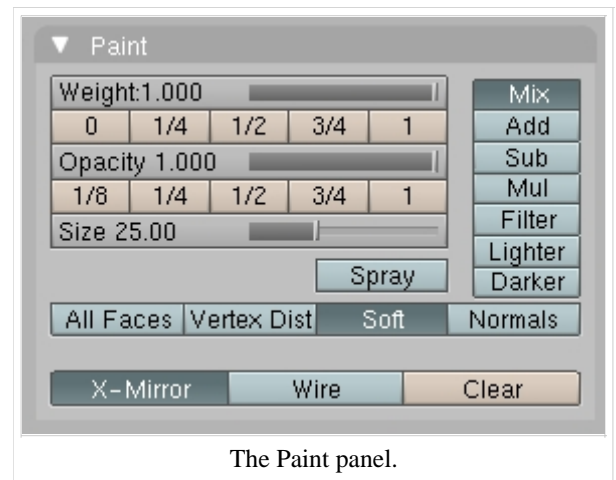


Character mesh, in Object mode.



Character mesh, in Weight Paint mode.


- In the Edit buttons a new panel is now available: the Paint panel. Make the following settings:
 - Change *Weight* to 1.0. This determines the color of the brush.
 - Change *Opacity* to 1.0. This is how much "paint" is applied with each brush stroke.
 - Turn off *All Faces*. This gives us a little more control over what gets painted.
 - Turn off *Vertex Dist*. This will also give us more control.
 - Leave *Soft* on, but this is mostly personal preference. For what we're doing, it just makes the painting look a little smoother.
 - **Important:** Turn *X-Mirror* on. This will let us take advantage of our symmetrically-named bones, so we'll only have to paint one side!

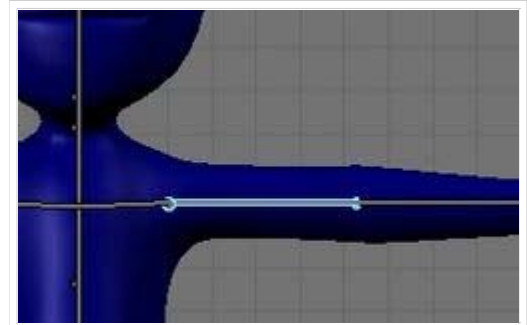


The Paint panel.


The **Weight** in weight painting refers to **the strength of a bone's influence**. With the *upper_arm.l* bone selected, we want the vertices in the upper arm of the mesh to be influenced. So we paint them!

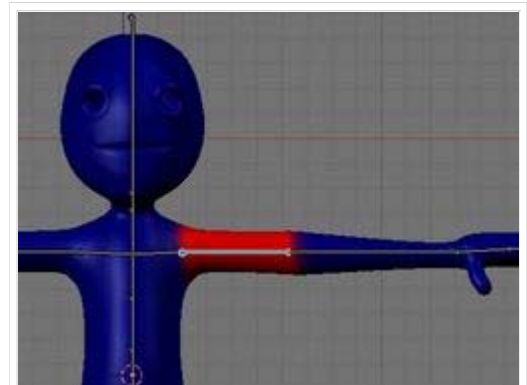
Weight painting the upper arm

- Select the *upper_arm.l* bone with RMB . Even though we're in the mesh's weight paint mode, we can still select bones because we told the mesh we're going to be using this armature using the Armature modifier.



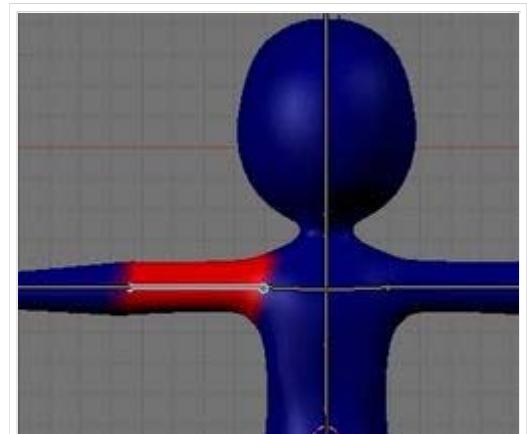
upper_arm.l selected while in the mesh's Weight paint mode.

- Using LMB , carefully paint over the the *upper_arm.l* bone with the brush. The mesh now appears red, indicating those vertices have a weight of 1.0 when it comes to being affected by *upper_arm.l*. Other colors indicate other weights, but for the purposes of this tutorial we will use all weights of 1.0.



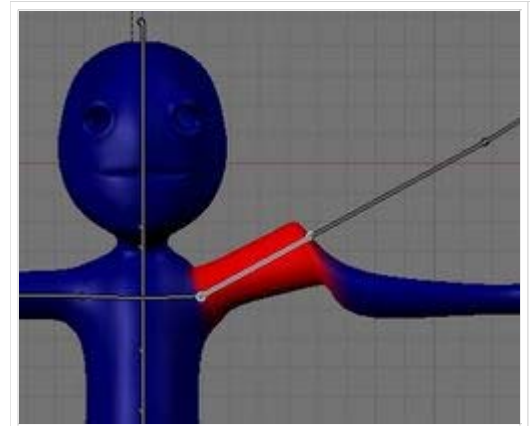
Weight painting vertices for *upper_arm.l*, from Front view.

- Rotate the view to get a good look at the back of the mesh. Ctrl NumPad 1 is a quick way to do this (Rear view). Paint the back side of *upper_arm.l* as well.



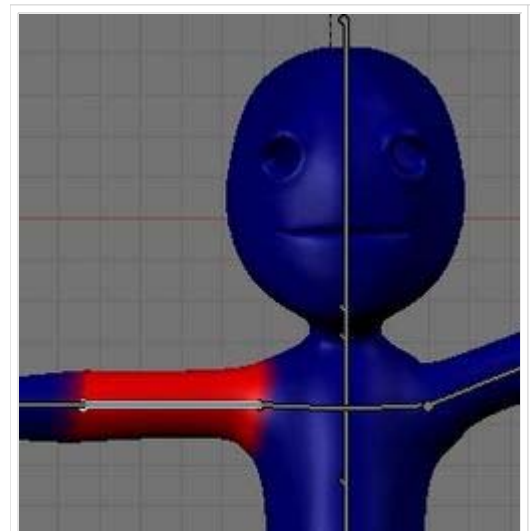
Weight painting vertices for *upper_arm.l*, from Rear view.

- Select the *upper_arm.l* bone and rotate it (**r**). The mesh deforms along with the bone.



Rotating *upper_arm.l*: the mesh move along.

- Select *upper_arm.r* on the other side. Notice the vertices for this bone are already painted! That's because we named the bones symmetrically (l and r) and enabled X-Mirror in the Paint panel. Blender found the bone on the other side and mirrored the weights for us.



upper_arm.l selected.

- Select *upper_arm.l* again and clear its rotation with **Alt r**.

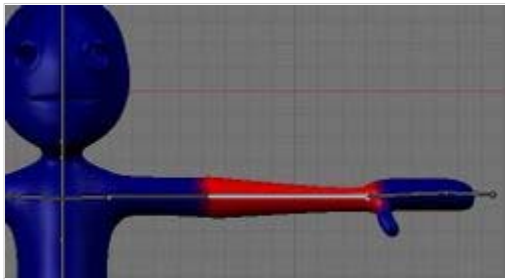
Tips for weight painting

Here are some tips to keep in mind while weight painting.

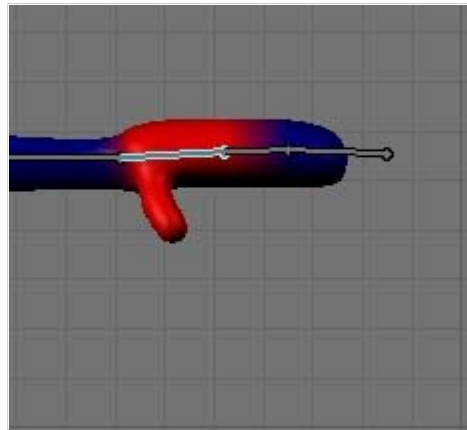
- The vertices you paint will be assigned to the bone you have selected.
- Make sure you paint over only the vertices you want to move with the bone you have selected. A little bit bleeding over into an adjacent bone's area may be OK.
- If you make a mistake, **Ctrl Z** only undoes the last step, not multiple steps. I find that the best way is to press the *Clear* button in the Paint panel. This clears all weights for the selected bone. **Important:** Clearing the weights does NOT happen symmetrically! If you aren't happy with *upper_arm.l* and clear it to start over on that bone, you have to select *upper_arm.r* and clear the weights on that bone as well. Otherwise you'll end up with residual weights on *upper_arm.r* that will make things unsymmetrical.
- It doesn't matter if your weights don't look exactly like those pictured here, as long as you're happy with the resulting deformation of the mesh when you move the bones around.

Weight paint the rest of the arms

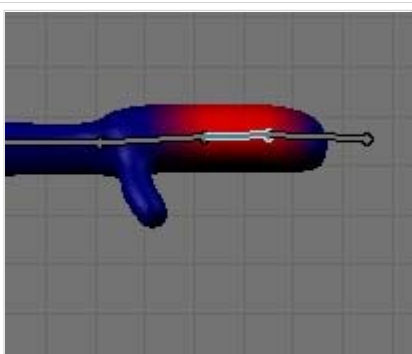
- Select *lower_arm.l* and weight paint it. Don't forget to rotate around to the other side of the mesh to paint the back of the lower arm as well.
- Continue on down to the hand and finger bones, first selecting the bone and then weight painting from the front and back.



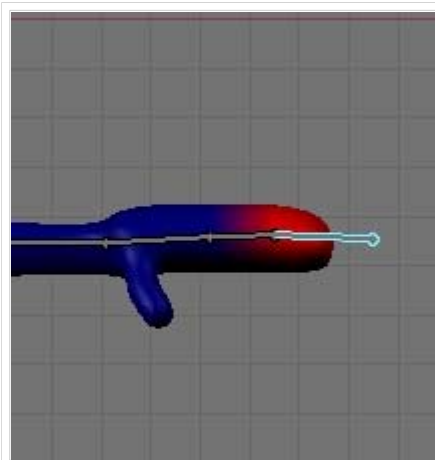
Weights for *lower_arm.l*.



Weights for *hand.l*.



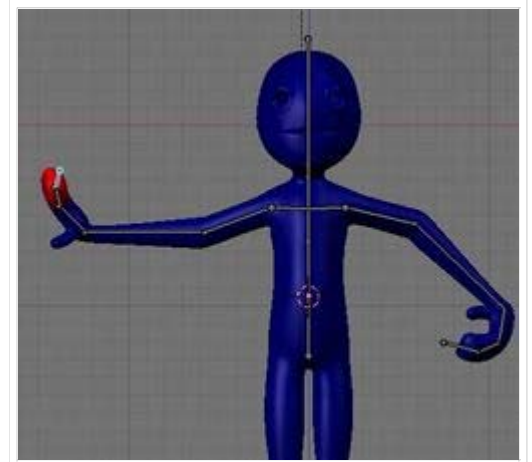
Weights for *finger1.l*.



Weights for *finger2.l*.

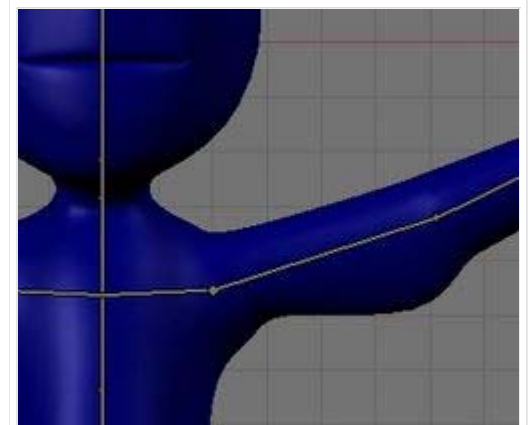
If you make a mistake: If you make a mistake while weight painting, **Ctrl** **Z** only undoes one step. The most straightforward way to fix a mistake is to reset the weights for the bone you have selected by pressing *Clear* in the Paint panel, and redo the weights for that bone.

- Now select the *finger2* bones and move them. If Auto-IK is enabled, the arms should move along with them. Auto-IK is a property of the armature, so if it's not enabled, you have to switch to Object mode, select the armature, and turn on Auto-IK.



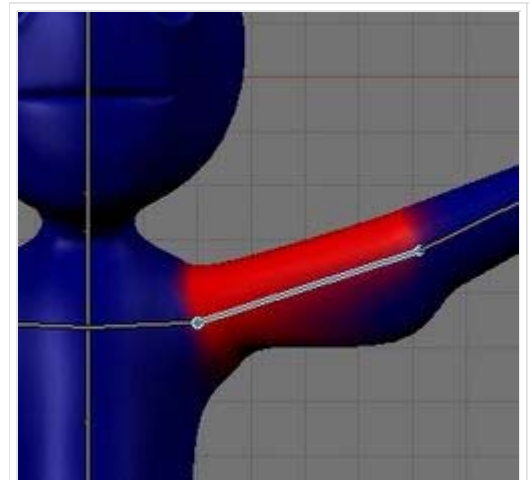
The arms now move!.

Sometimes you'll miss painting a vertex at all. When you go to move the bones, you might see something like this:



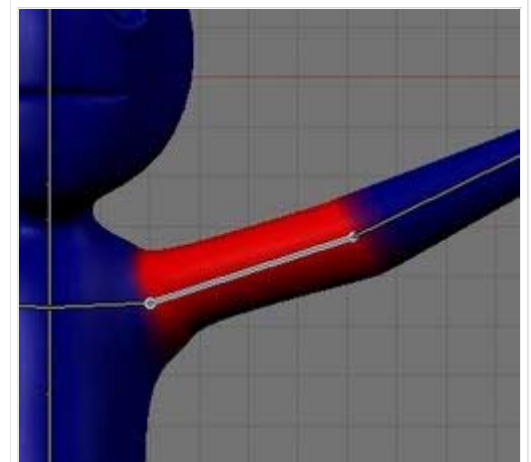
Arm isn't deforming correctly.

To correct it, select the bone that should move those vertices . . .



Select the bone . . .

and then paint the uncooperative vertices. They should snap back into place where they should be.



And paint the offending vertices.

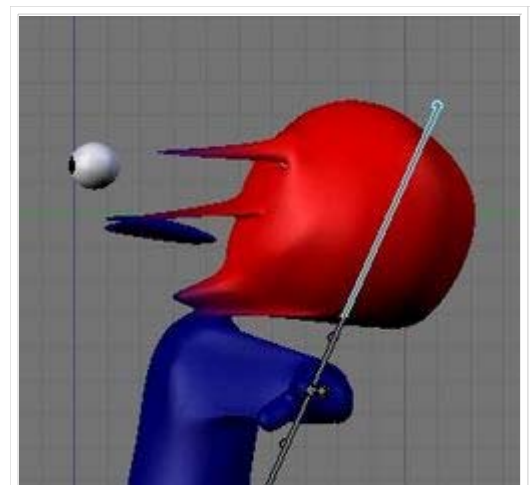
Depending on how you modeled your character, it may take some playing around with in Weight Paint mode to get the bones to deform the character correctly.

- To reset the bone rotation and location, we have to exit out of Weight Paint mode with **Ctrl** **TAB**
- Then select the armature.
- Use the **A** - **A** - **Alt** **G** - **Alt** **R** combination to clear the location and rotation of all bones.
- Re-select the character mesh.
- Switch back to Weight Paint mode (**Ctrl** **TAB**), with the armature reset to rest position.

Weight painting the head and torso

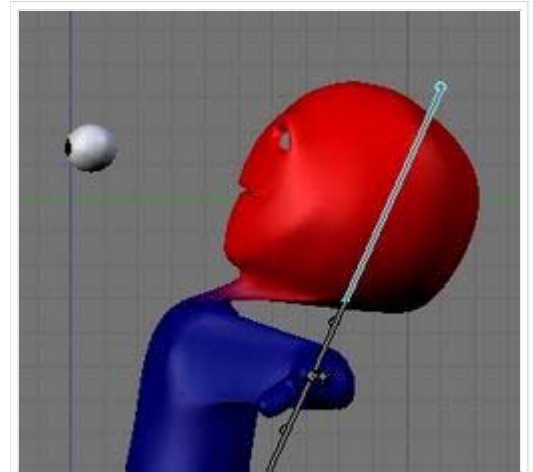
Now let's weight paint the head and torso.

- Switch to Weight Paint mode for the character mesh.
- Select the *head* bone.
- Paint away! Don't forget the front and back.
- With the head bone still selected, move it backward in Side view with **g**. Whoa! What's going on?
- By moving the head, we're able to see that we missed some vertices. It turns out that these are the vertices on the inside of the head that we couldn't see before: the inner faces of the eye sockets, and the inside of the mouth.



Moving the head bone back.

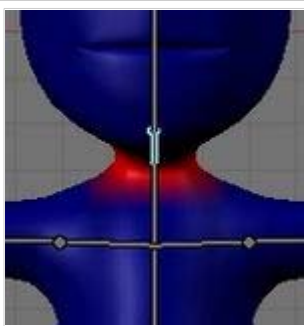
- To fix this, with the head bone still selected just paint over the left-behind vertices. By the way, don't worry about the eyes yet. We'll fix that momentarily.
- Reset the bones (**Ctrl** **TAB** , select armature, **a** - **a** - **Alt** **g** - **Alt**)
- Go back to Weight Paint mode of the character mesh.



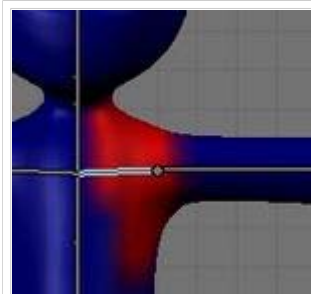
Paint over the vertices left behind after the head was moved.

Select the spine bones and the neck bone, and start weight painting.

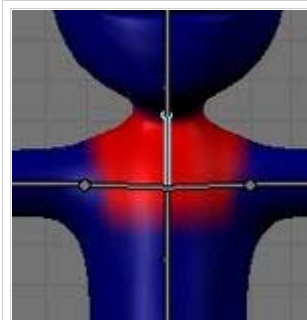
My weights looked like this:



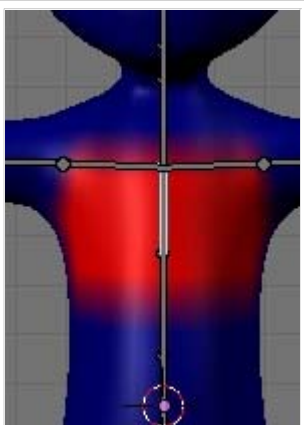
neck weights.



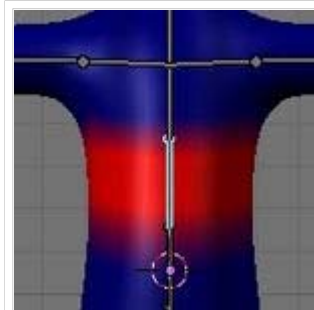
shoulder.l weights.



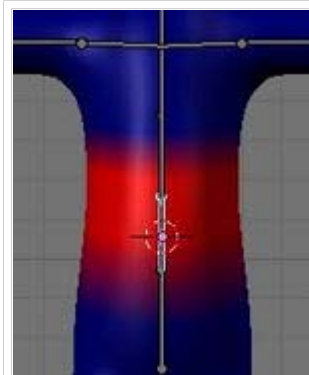
spine5 weights.



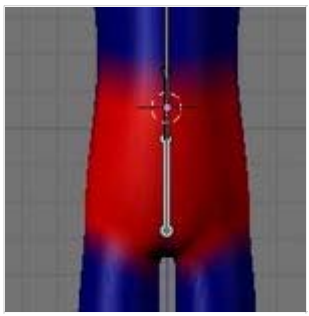
spine4 weights.



spine3 weights.



spine2 weights.



spine1 weights.

Parenting the eyes to the head bone

When we moved the head, the eyes were getting left behind. We couldn't weight paint them to the *head* bone while we were weight painting the body mesh, because the eyes are separate mesh objects.

It's possible to apply an Armature modifier to the eyes, select the *head* bone, and weight paint the eyes to the *head* bone. However, this isn't the best way to do it.

Here's why: We're going to eventually make the eyes track a target. To do so, they need to swivel around their object centers. If we weight painted the eyes to the head, when we moved the head it would pull the eye vertices away from the object center, making the eye tracking all funky.

Instead, we'll parent the entire eye object to the head bone -- not just its component vertices.

- Switch out of Weight Paint mode.
- Make sure the armature is in Pose mode.
- Select one of the eyes.
- With the eye still selected, shift-select the head bone.
- Parent the eye to the *head* bone with **Ctrl** **P** >> *Bone*.

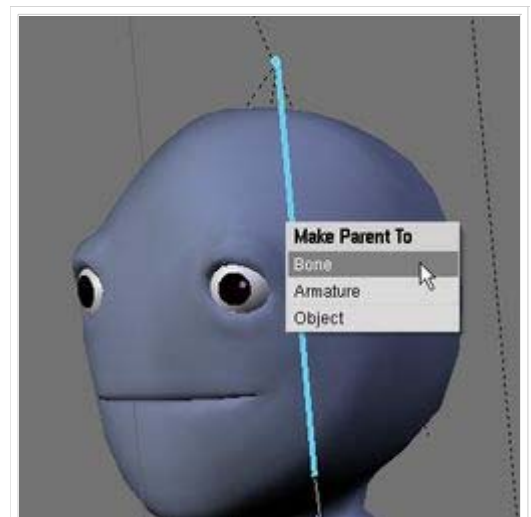
Do the same thing with the other eye:

- Select the eye
- Shift-select the *head* bone.
- **Ctrl** **P** >> *Bone* to parent to the *head* bone.

Now try moving or rotating the head. The eyes should follow along now.

Testing the rig

All right! Switch out of Weight Paint mode, and select the armature. You can now pose your character! Try out some test poses and test renders to play around with your new creation!



Parenting the left eye to the *head* bone.

Very handy tip: So far, we've been using **R** to rotate. Try hitting **R**-**R** (hit it twice in a row). This enables **trackball** rotation, and makes positioning bones much easier than regular old rotation.

It'll probably take some practice to get used to the controls. Try these different methods of moving bones to see how they react differently:

- Turn Auto-IK ON.
- Select a bone and move it
- Select a bone and rotate it. Notice how Auto-IK only kicks in when you move, but not when you rotate.
- Turn Auto-IK OFF
- Select a bone and move it
- Select a bone and rotate it. Notice they do the same thing, EXCEPT when you try to move the upper arm. Moving the upper arm detaches it from the shoulder. Well, that's easy to fix: don't move the upper arm, just rotate it!



Test render.

Summary: By weight painting, we told Blender which vertices to move when we moved the bones. The character finally moves! Next, we'll build the lower body armature and do some more weight painting.

You can download the .blend file up to this point: [Media:BSoD-ItCA-upper_body_weight_paint.blend](#).

Next: Lower body: armature and weight painting

Previous: Upper body: armature

Back to Index

Retrieved from

["http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Upper_body_weight_painting"](http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Upper_body_weight_painting)

- This page was last modified 11:49, 7 September 2006.

BSoD/Introduction to Character Animation/Lower body armature

From BlenderWiki

< BSoD | Introduction to Character Animation

Building the lower body armature

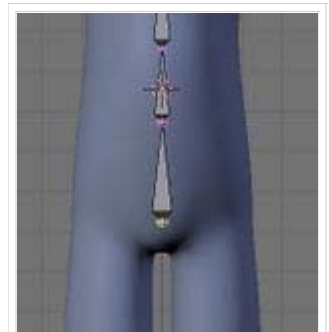
We'll add the lower body onto the existing upper body armature. Now that you've had some practice with armatures on the upper body, we can get into a slightly more complex leg setup. Keep in mind this is a purposefully simple rig, see BSoD/Introduction_to_Rigging/Leg_Rigs for more advanced rigs with better control.

- Select the armature and switch to Edit mode (**TAB**).
- Switch the draw type back to *Octahedron*. This will allow us to better see which direction the bones are facing.
- Double check that *X-Axis Mirror* is still selected.



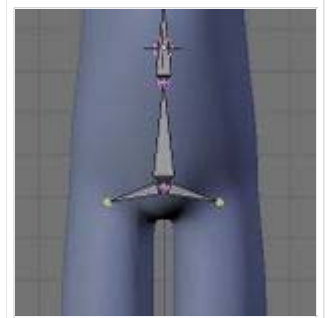
Armature panel settings.

- Select the root of the *spine1* bone.



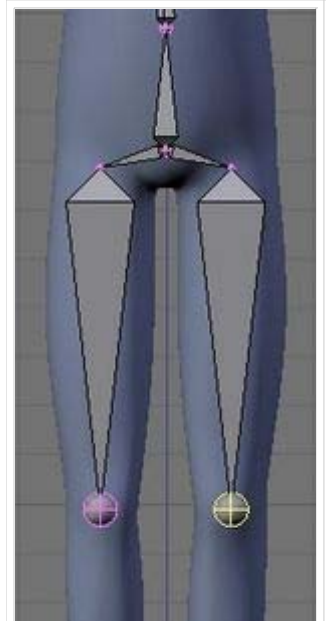
Select the root of *spine1*.

- Symmetrically extrude the hips using **Shift E**.



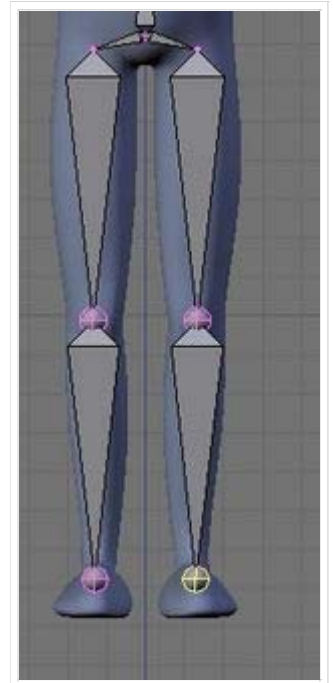
Symmetrically extrude the hips.

- Symmetrically extrude the upper legs with **Shift E**.



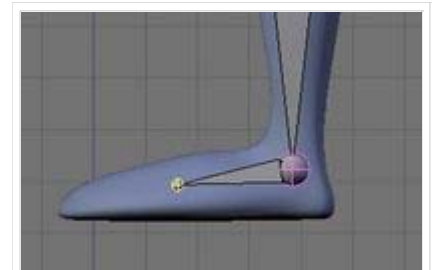
Symmetrically extrude the upper legs.

- Symmetrically extrude the lower legs.



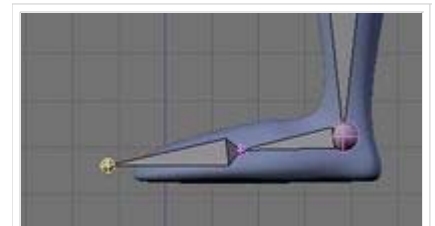
Symmetrically extrude the lower legs.

- Switch to side view (NumPad 3).
- Extrude a foot bone . . .



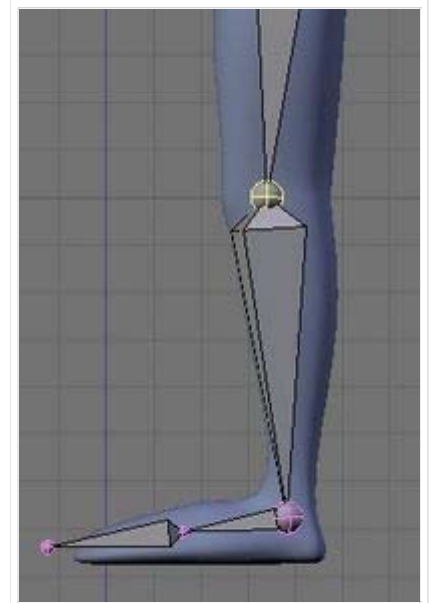
Symmetrically extrude the foot.

- . . . and a toe bone.



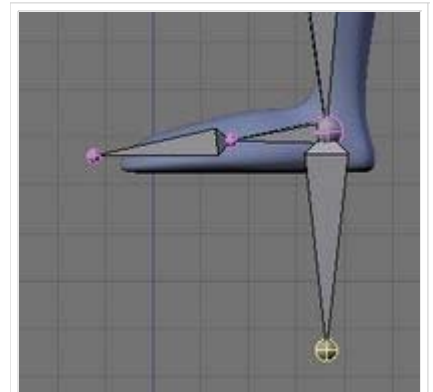
Symmetrically extrude the toe.

- In Side view, select the tip of the upper leg/bottom of the lower leg and move it forward a little to give the knee some bend. This will make the IK setup we're about to make work better - it will be difficult to get it to bend the knee backwards if it initially has some bend in the right direction.



Give a little bend to the knee.

- Select the tip of the lower leg, and extrude it downward. This bone won't deform the mesh, but it will act as the "handle" for the leg's IK setup. This will all become more clear later.



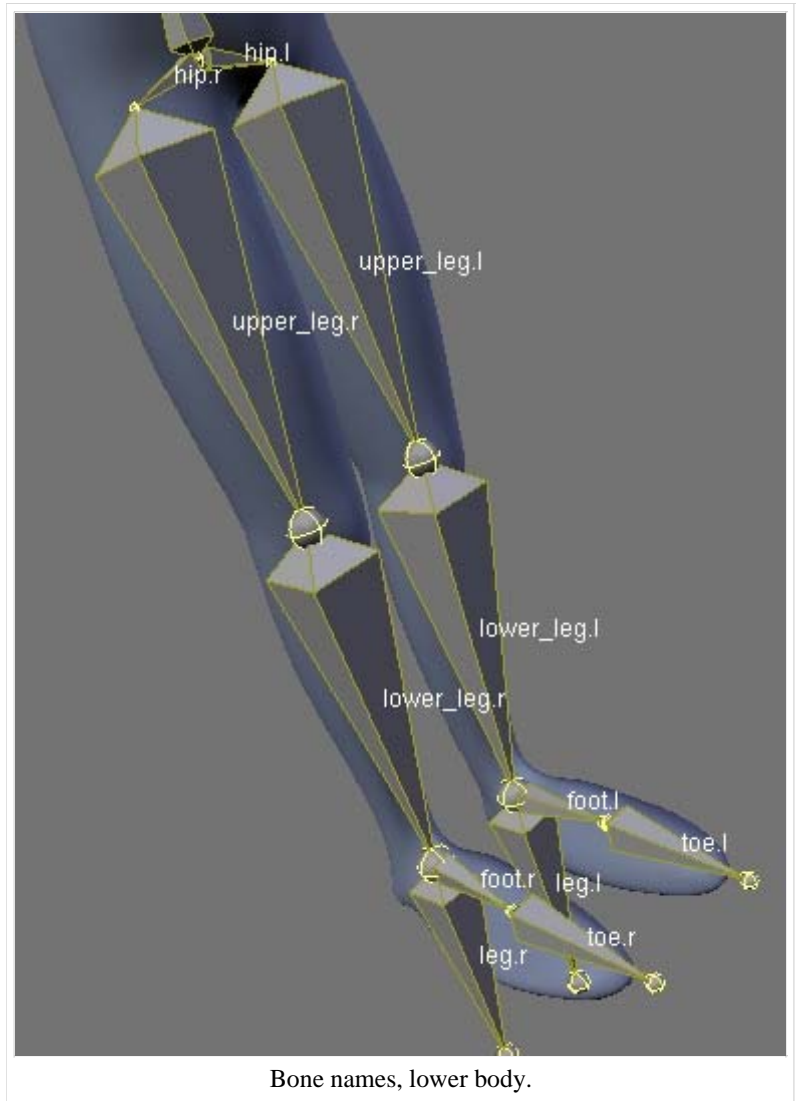
Extruding the foot controller.

The bones are all set up now. It's time to adjust some settings. Remember that settings will not automatically update on the other side like extruding bones did, so we'll have to make changes to the settings on each side separately.

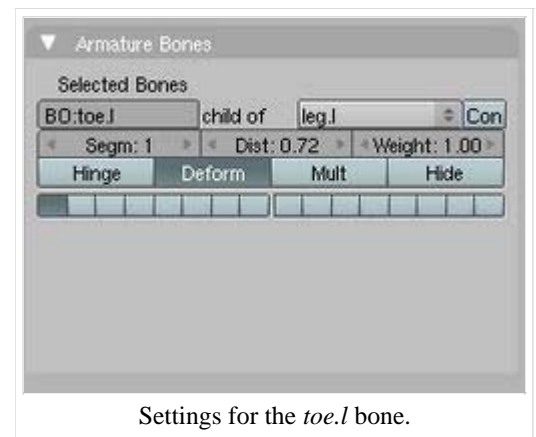
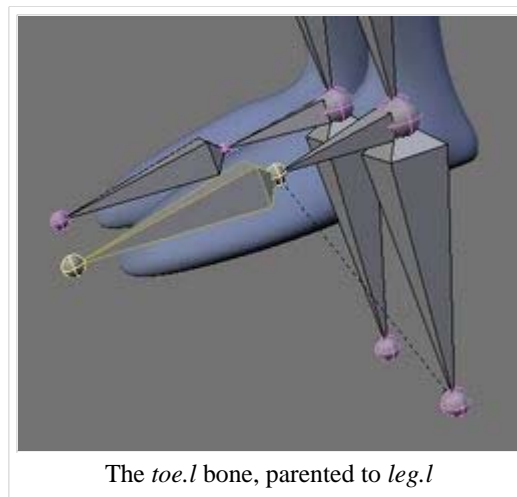
- Name the bones. I named them
- *hip.l*
- *upper_leg.l*
- *lower_leg.l*
- *foot.l*
- *toe.l*
- *leg.l*

and

- *hip.r*
- *upper_leg.r*
- *lower_leg.r*
- *foot.r*
- *toe.r*
- *leg.r*



- Select *toe.l*.
- In the Armature Bones panel, first deselect the *Con* button to disconnect the bone from its current parent.
- Then make it the child of *leg.l* by selecting "leg.l" from *toe.l*s Armature Bones panel. If you don't disconnect it first, when you make it the child of *leg.l* the toe bone will snap to the tip of the *leg.l* bone. If this happens, undo with

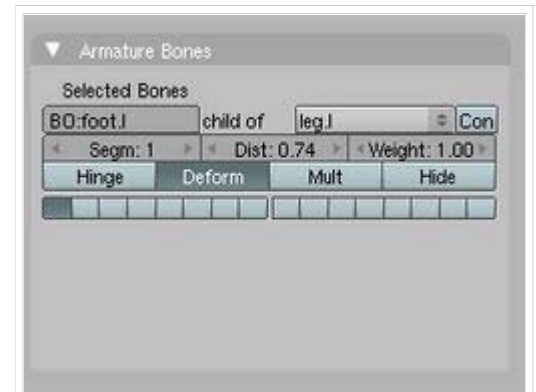


Ctrl | Z and disconnect the *toe.l* first.

- After parenting, notice that there's a dotted line that goes from *toe.l* to *leg.l*, indicating that *toe.l* is parented to *leg.l*.

Select *foot.l* and do the same thing:

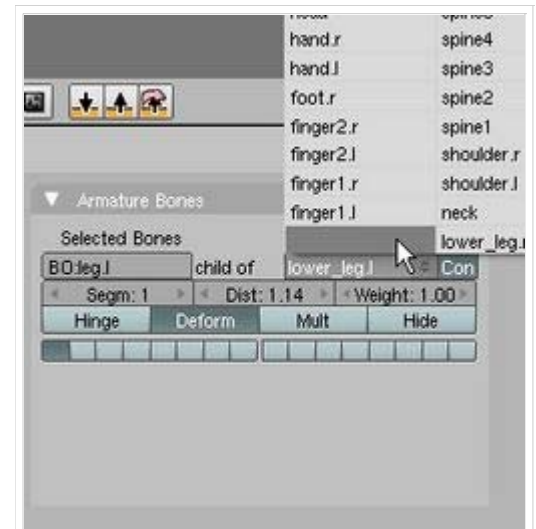
- Disconnect *foot.l* from its parent by clicking the *Con* button, then *foot.l* a child of *leg.l*. You can't see the dotted line connecting child and parent because it's hidden by the bones.



Settings for *foot.l*.

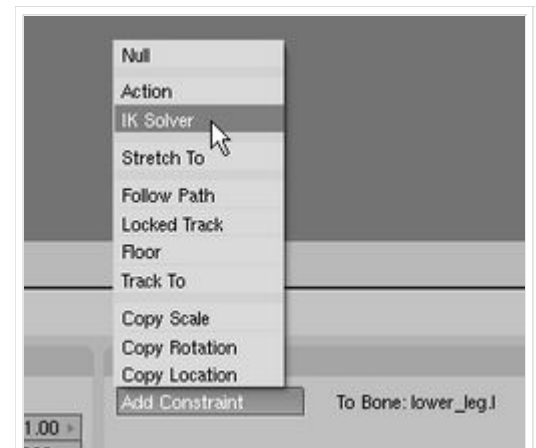
- Select *leg.l* and make it the child of "none", or the blank spot at the bottom of the "child of" menu. The "Con" button will disappear . . . if there's no parent, there's nothing for this bone to be connected to.

By making *leg.l* have no parent, it means that no bone higher up in the hierarchy will move *leg.l*. The only way it will move is if we move it directly. As you'll see, this is useful for keeping the feet on the ground.



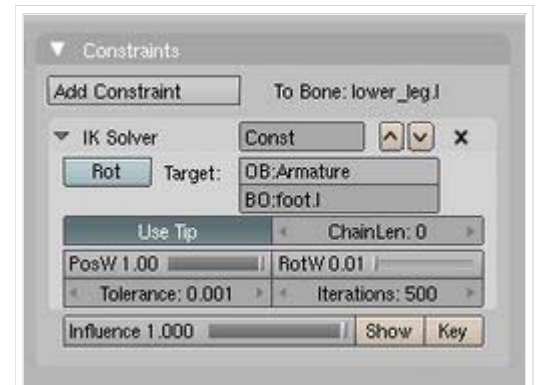
Make the *leg.l* bone be parented to none.

- Switch to Pose mode (TAB).
- Select the *lower_leg.l* bone.
- Add an IK Solver to *lower_leg.l*. To do this, in the Constraints panel under the Edit buttons click the *Add Constraint* button and choose *IK Solver*. This adds a **constraint** to the selected bone. Note that in the Constraints panel, there are now settings for the IK Solver constraint.



Add an IK Solver constraint to the *lower_leg.l* bone.

- In the Constraints panel, type in "Armature" in the OB: text box.
- Type in "foot.l" in the BO: text box. We just told the constraint that the target will be the *foot.l* bone in the armature object called Armature.



The *lower_leg.l* IK Solver constraint.

A constraint is a way of sort of programming how a bone should act. For example, one type of constraint on that menu is a *Track To* constraint. You tell the constraint what object you want the bone to track, and then whenever you move that object, the bone will point to it.

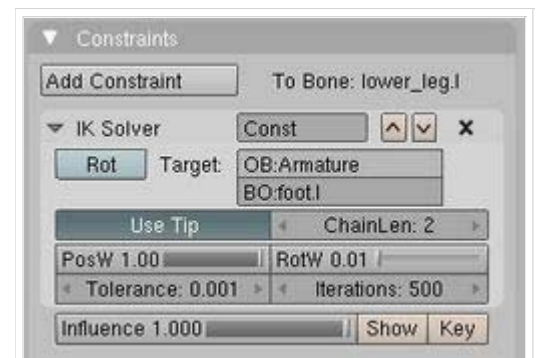
Well, the IK Solver constraint is similar. We'll tell the constraint what it's Target should be, and it will try to point the bone and its whole chain at that target by solving lots of equations. You'll have to wait and see till it's set up to see the advantage over Auto-IK . . .

- Important: Make sure Auto IK is OFF in the Armature panel.
- Try moving *leg.l*, the controller for the leg. The whole armature moves along, spine and everything attached to it. That's not right! Remember we had a similar problem with the arms, and fixed it by disconnecting the shoulder in order to break the chain. This time, though, there's a setting in the IK Solver constraint made for fixing just this sort of thing.



IK Solver doesn't work right with the default settings . . .

- Select the *lower_leg.l* bone again.
- In the Constraint panel under the IK Solver constraint, change *ChainLen* to 2. By default, ChainLen is zero, (or undefined) meaning that the IK Solver involves the entire chain no matter how long it is. By changing it to 2, we told the IK Solver that the chain is only 2 bones long - in other words, only the *lower_leg.l* bone and the *upper_leg.l* bone.



Changing *ChainLen* to 2.

- Try selecting the *leg.l* bone again and moving it. Just the leg should move now.

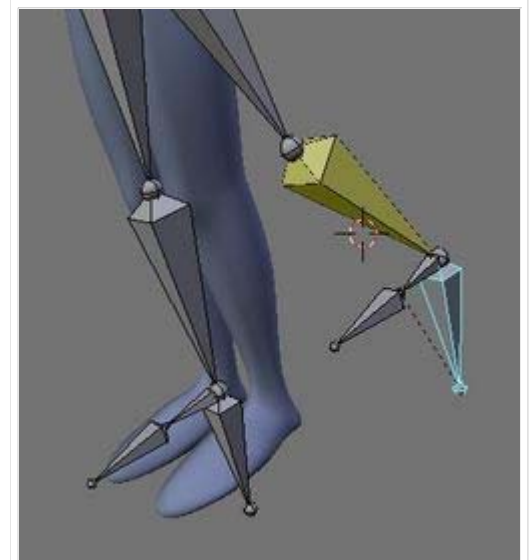
Now do the same thing to the right side of the body:

- In Edit mode, disconnect *toe.r* and make it the child of *leg.r*.
- Disconnect *foot.r* and make it the child of *leg.r*.
- Make *leg.r* the child of none.
- Switch to Pose mode.
- Add an IK Solver constraint to *lower_leg.r*. Make the target *foot.r* in the object *Armature*. **Alternative:** a faster way to add an IK constraint is by selecting the Target (*foot.r*), then shift-selecting the bone (*lower_leg.r*) and pressing **Ctrl** **I**.
- Change the ChainLen to 2.

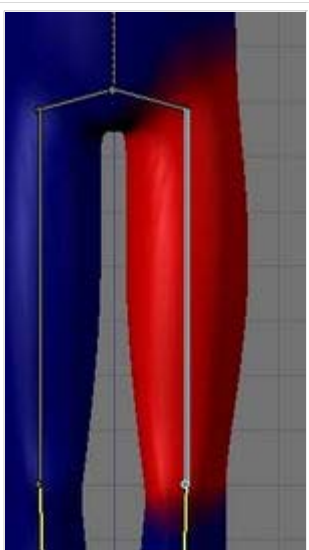
Weight paint the lower body

From working on the upper body, you know how to weight paint. Let's do the same thing for the lower body.

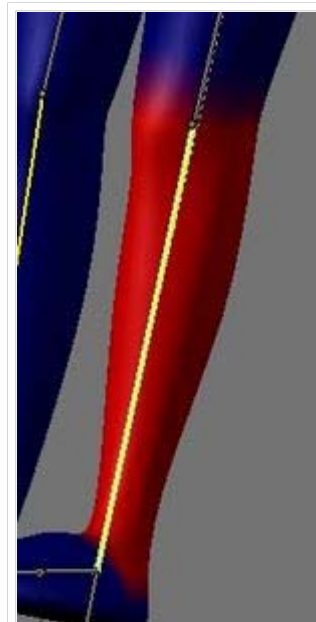
- In Pose mode, clear the location and rotation of all bones with **A - A - Alt R - Alt G**.
- In the Armature panel, change the draw type back to Stick.
- Select the character mesh.
- **Ctrl** **TAB** to get into Weight Paint mode.
- Double check:
 - *X-Mirror* is selected (important!)
 - All Faces and Vertex Dist are unselected
 - Weight is set to 1.0
 - Opacity is set to 1.0
- Select each bone and start weight painting, remember to rotate the view around to paint all vertices. I ended up not using the *hip* bones directly - the character was deforming fine without them, so there are no weights shown here.



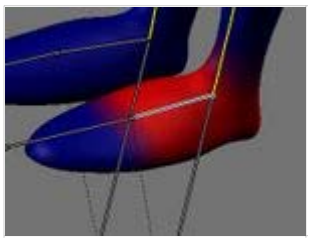
Now moving *leg.l* works correctly.



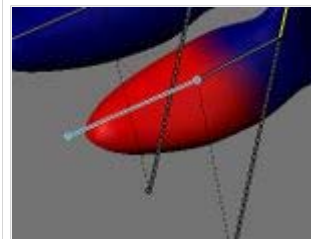
upper_leg.l weights.



lower_leg.l weights.



foot.l weights.



toe.l weights.

Test poses

Try out the new functionality with some test poses, and render with [F12](#).



Test pose

Summary: We constructed the lower body armature, added IK constraints, and weight painted it. Next we'll tweak the rig to make it easier to control.

Next: Final rig adjustments

Previous: Upper body: weight painting

Back to Index

Retrieved from

["http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Lower_body_armature"](http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Lower_body_armature)

- This page was last modified 21:55, 15 September 2006.

BSoD/Introduction to Character Animation/Final rig adjustments

From BlenderWiki

< BSoD | Introduction to Character Animation

Contents

- 1 Final adjustments
- 2 Adding a master bone
- 3 Eye tracking
- 4 Tips for using the rig

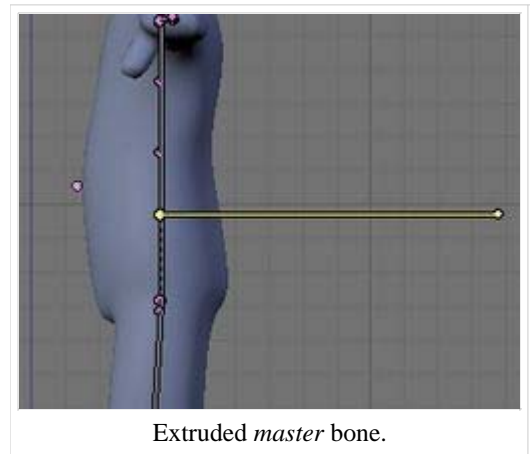
Final adjustments

In this section, we'll make some final adjustments to the rig to make it easier to control. Specifically, we'll add:

- a master bone so we can move the entire character at once.
- knee-lock targets so we can control where the knees point.
- an eye target bone so we can easily change which direction the eyes are looking.

Adding a master bone

- Select the armature and enter Edit mode.
- Select the tip of *spine1* and extrude it backward with E.
- In the Armature Bones panel of the Edit buttons, name this bone something meaningful. I named it "*master*".
- Make this new bone the child of none, so that the only way this bone will move is if we move it directly.

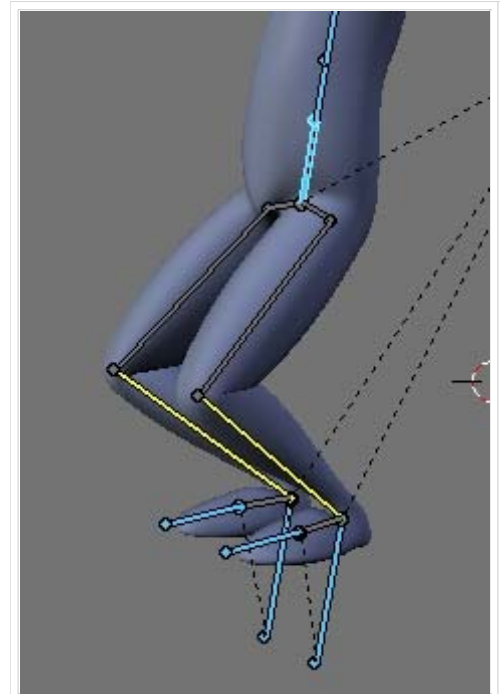


This bone will be the highest parent of all the other bones in the armature. The *master* bone will be the one to move when we want to move the whole body at once, like in a jump.

- For each bone (*spine1*, *leg.l*, and *leg.r*)
- Select the bone.
- Choose *master* in the *Child of* menu to parent it to the master bone.

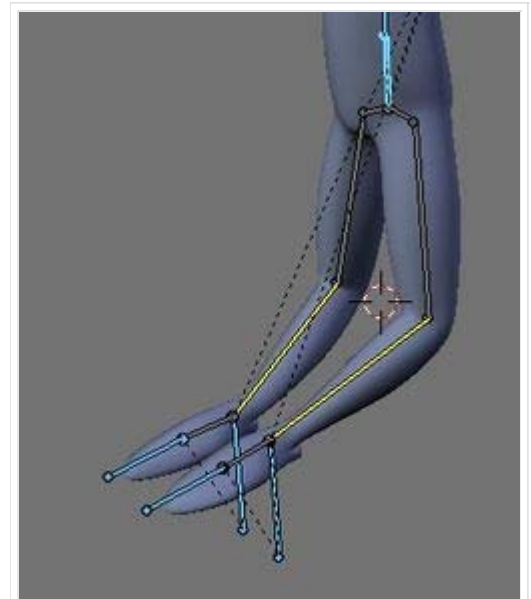
Note that dotted lines are now drawn from each bone to the *master* bone. Now when we move *master* in Pose mode, the entire armature and body will move.

- Here's something that needs fixing: in Pose mode, grab the *spine1* bone and move it around. When you move it forward and down, you get something like this.



Moving *spine1* down and forward: works OK.

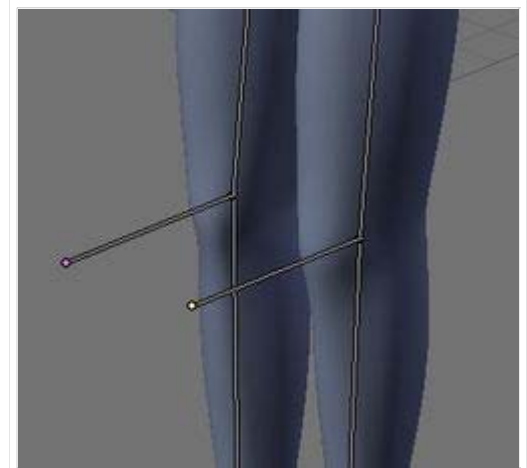
- But when you move it back and down, you get something like this. Ouch!



The IK constraints don't work well in this direction.

The IK constraints on the legs are doing their job. They're trying to keep the leg pointed at the leg target, and they don't know that knees only bend one way. To fix this, we'll add another constraint to the upper leg bone to keep it pointing in the direction we tell it.

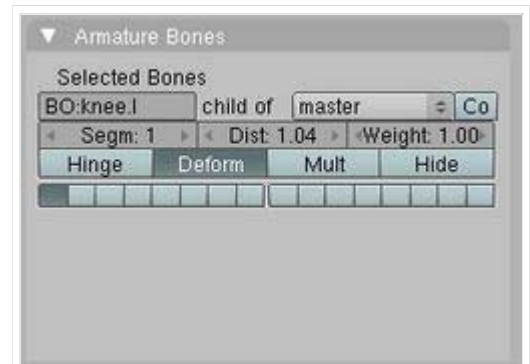
- Clear location and rotation of all bones (**A** - **A** - **Alt** | **G** - **Alt** | **R**).
- Switch to the armature's Edit mode.
- Select the tip of the upper leg bone (at the knee) and symmetrically extrude a bone forward.



Extruded *knee* bones (view rotated for clarity).

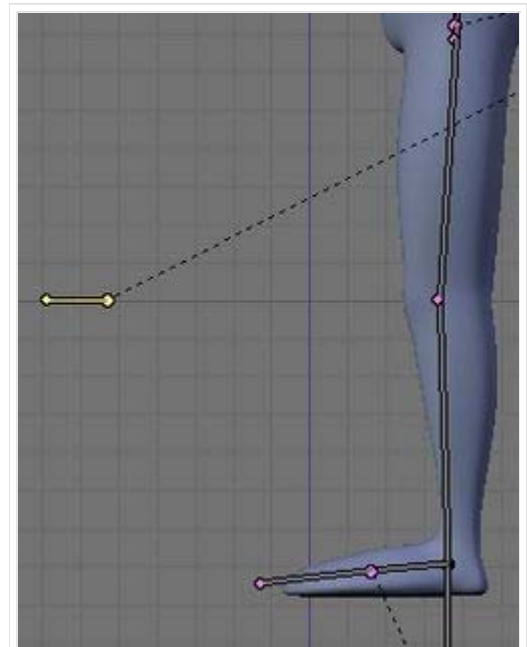
For each of the two new bones (remember you have to change these settings individually, they're not automatically symmetrical):

- Name the bone (I called them *knee.l* and *knee.r*).
- Disconnect the bone (Click the *Con* button).
- Make it the child of *master*.



Settings for the *knee.l* bone.

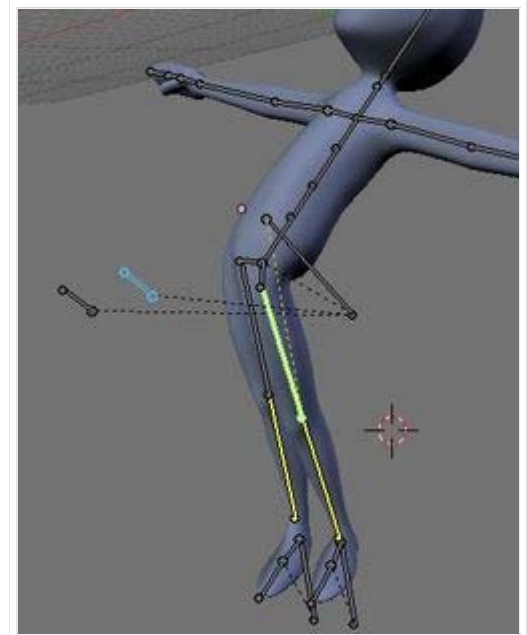
- In Side view, move the knee bones away from the actual knees. These bones will be acting as targets for the upper legs, and if they're too close the upper legs might not know what to point to. I also scaled them down a little with **S** so they're less in the way. Remember, with X-Axis Mirror selected, you only have to move and scale one bone, and the other will automatically update.



Knee bones moved away from the mesh.

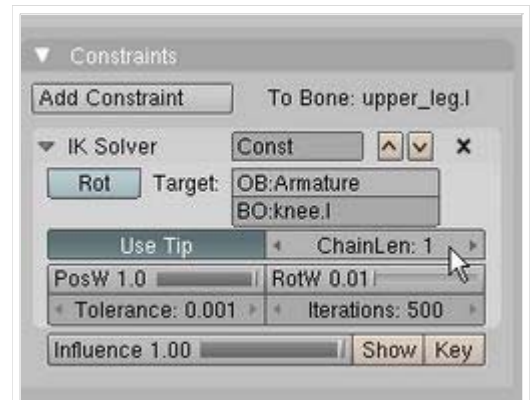
Now we'll add another IK Constraint to the upper legs, telling them to try to point to the new knee bones.

- Switch to Pose mode.
- First select *knee.l*, then shift-select *upper_leg.l*.
- Add an IK constraint with **Ctrl I**. Whoa, the armature goes crazy! That's because by default, the ChainLen is 0, which means the IK constraint tries to point as long of a chain as it can toward the knee bone. That chain includes *hip.l* and *spine1*, so it tries to point them to the *knee.l* target as well.



Armature directly after adding an IK constraint to *upper_leg.l*.

- We only want a single bone chain (just *upper_leg.l*) to point to the knee, so change the ChainLen of the IK constraint on *upper_leg.l* to 1.

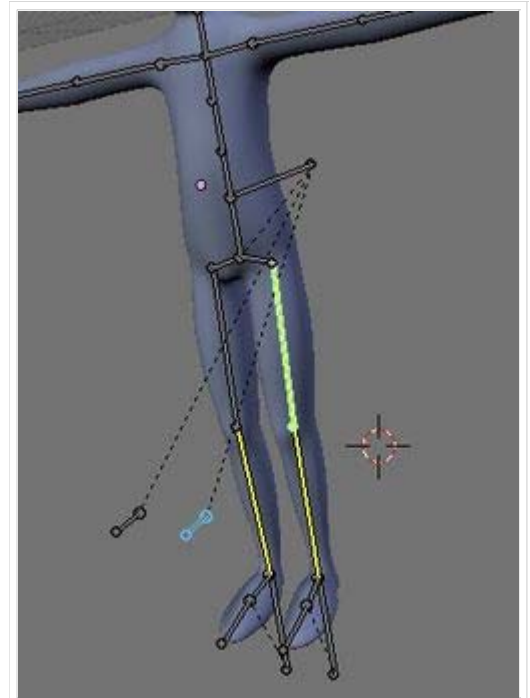


Change the ChainLen to 1 to fix the armature.

- The armature now acts normally.

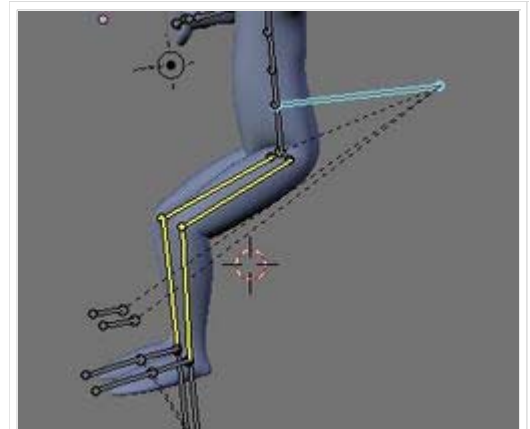
Do the same thing for the other leg:

- Select *knee.r*, then shift-select *upper_leg.r*.
- Add an IK constraint with **Ctrl I**.
- Change the ChainLen to 1.



The armature now acts normally.

Test out the new knee bones by moving the *spine1* bone around, especially down and back. Much better!

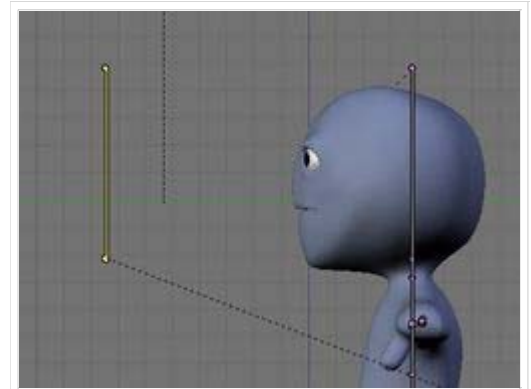


Knees now behave properly when *spine1* is moved down and back.

Eye tracking

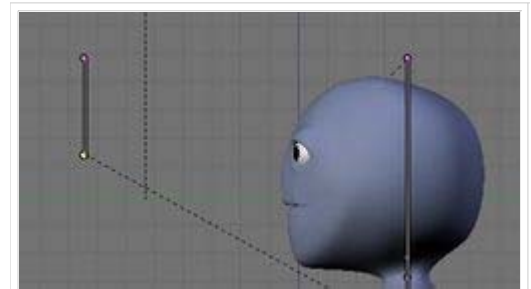
While we could rotate the eyes individually when animating, it will be a lot easier to have them track an object. It's possible to have the eyes track any object, but it will be easier to animate if we make that object a bone in the existing armature.

- In Pose mode, clear the rotation and location of all bones.
- Switch to the armature's Edit mode.
- Select the head bone.
- Duplicate it with **Shift D**, and move it in front of the head.
- Name the bone *eyes* or something meaningful.
- Make *eyes* a child of *master*. Note we didn't have to disconnect the bone this time, a bone is automatically disconnected when it's duplicated from an existing bone.



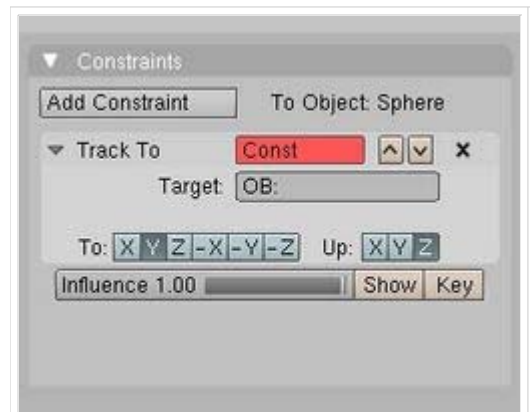
The *eyes* bone.

- Move the bottom (the root) of the bone to eye level. The root of the bone is where the eyes will point to (by the way, you could switch to Octahedron draw type for the bones to better see where the root is).



Adjusted *eyes* bone.

- Press **TAB** to switch to Pose mode (which is a substitute for Object mode) and select the character's left eye. It may take a couple of attempts to select the eye instead of the body. You can always switch to Layer 2 (with **2**), to see just the eyes, select the eye, then return to view all layers (**`**). The eye mesh is in Object mode.
- Switch to the Object buttons in the Button window (**F7**). There is a *Constraints* panel here, just like there was for the armature in Pose mode.
- In the Constraints panel, add a Track To constraint.

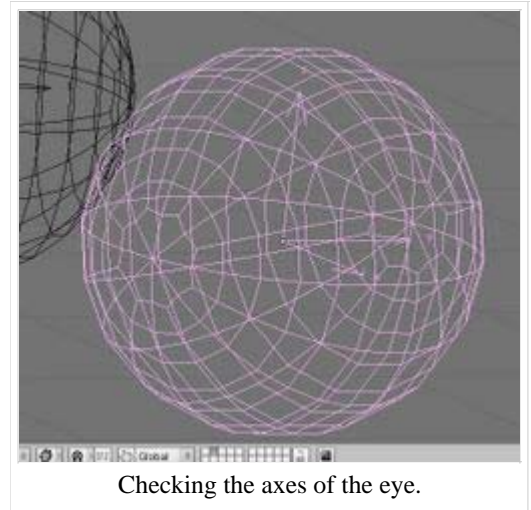


A Track To constraint added to the character's left eye in Object mode.

We want to tell this constraint that we want it to point the eye at the *eyes* bone in the armature. We'll also have to tell it which axis of the eye should be considered "up", and which axis should point toward the *eyes* bone. To figure this out, we need to remind ourselves of rotation of the eyes.

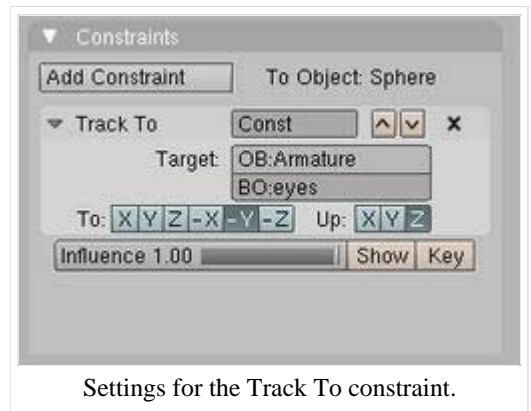
- In the *Draw* panel under the Object buttons, make sure *Axis* is selected.

- I had to switch to wireframe mode and just view Layer 2 to see the axes, and even then it was a little tough to see. Here, the Z axis is pointing up, the X axis is pointing to the right, and the Y axis is pointing back. In this case, I want to tell the Track To constraint that Z is up. The axis arrows point in a positive direction, and the Y axis is pointing back, so I want negative Y to point toward the *eyes* bone.
- Armed with the knowledge of which axis you want up and which you want to track with, view all layers and switch back to shaded mode (**Z**).



Checking the axes of the eye.

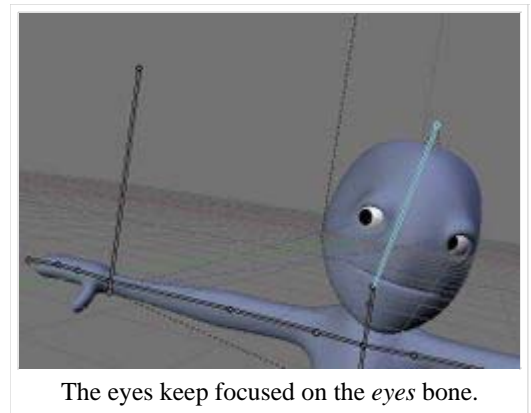
- In the *OB:* text box of the eye's Track To constraint, make the target object "Armature".
- Make the target bone "eyes" or whatever you have named the eyes bone (in the *BO:* text box that appears).
- Select "-Y" in the *To:* buttons. This is the axis we want to point to the *eyes* bone.
- Select "Z" in the *Up:* buttons. This tells the constraint which direction to point up.



Settings for the Track To constraint.

Do the same thing to the other eye:

- Select the other eye.
 - Double check the axes and the direction you want it to point (both eyes should be the same, otherwise go back and check out the instructions on rotating the eye).
 - Add a Track To constraint.
 - Make the target "Armature" and the "eyes" bone.
 - Select the *To:* and *Up:* axes as appropriate (I used -Y as the *To:* and Z as the *Up:*).
-
- Select the armature in Pose mode and move the *eyes* bone around. The eyes should track the bone!



The eyes keep focused on the *eyes* bone.

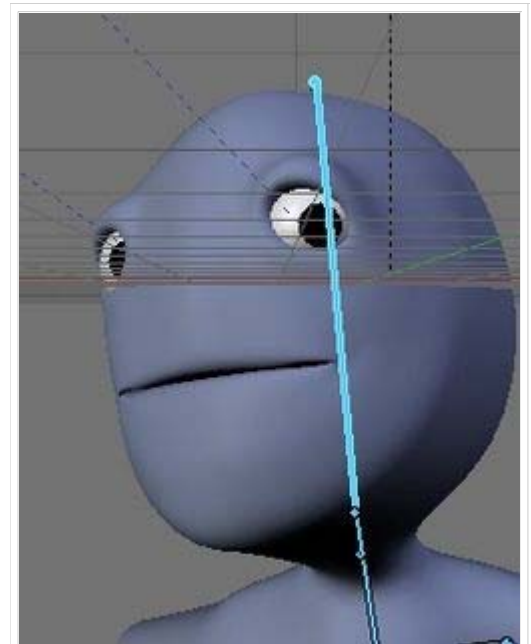
Tips for using the rig

A note on Auto IK: While it can be very useful for broadly positioning the arms, for the most part, you'll want to leave Auto IK off.

If you have Auto IK **on**, the *eyes*, *knee* bones, and *leg* bones won't work.

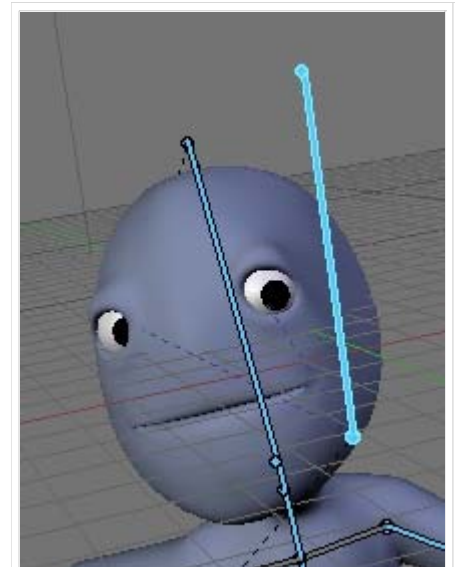
Some tips for each bone, starting from the top:

- *head*: use trackball rotation (**R** - **R**) to swivel the head around.



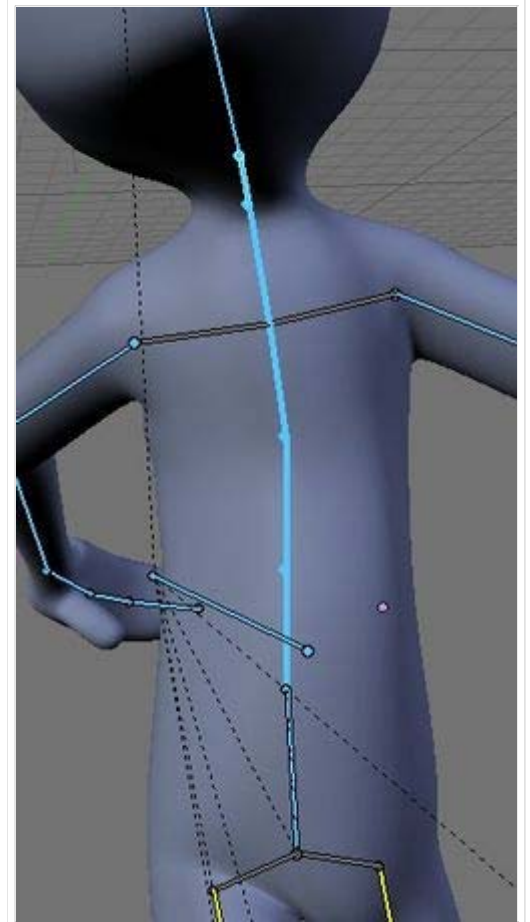
Rotate the *head* bone.

- *eyes*: move this bone around to control what the eyes are looking at. Move this bone farther away from the face to correct cross-eyes.



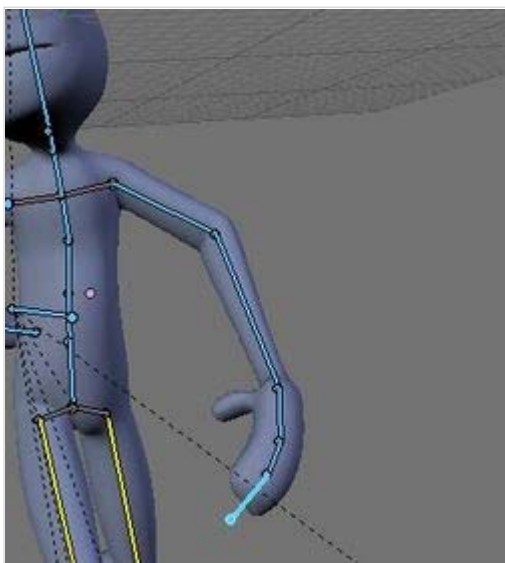
Point the eyes with the *eyes* bone.

- *neck* and *spine* bones (except *spine1*): Rotate individually for subtle bending of the body, or use Auto IK on the *head* bone for extreme bending of the spine.

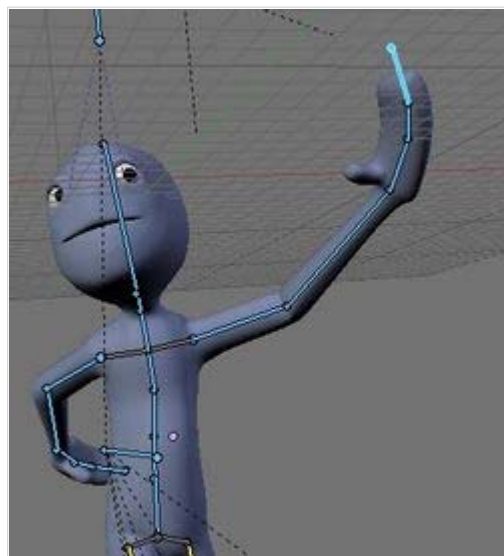


Rotate the *spine* bones individually.

- *arm* bones: Turn on Auto IK and move the *finger2* bones for broad sweeping motions of the arms.

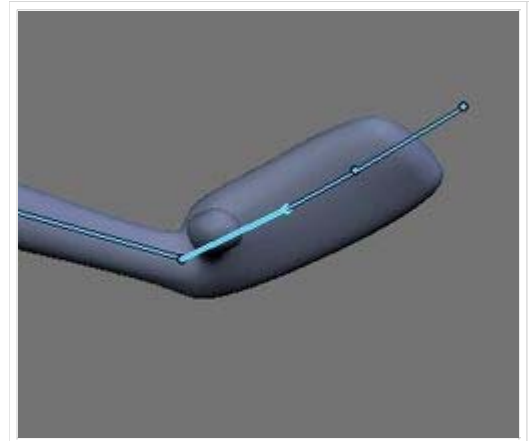


Move with Auto IK on *finger2* bones . . .



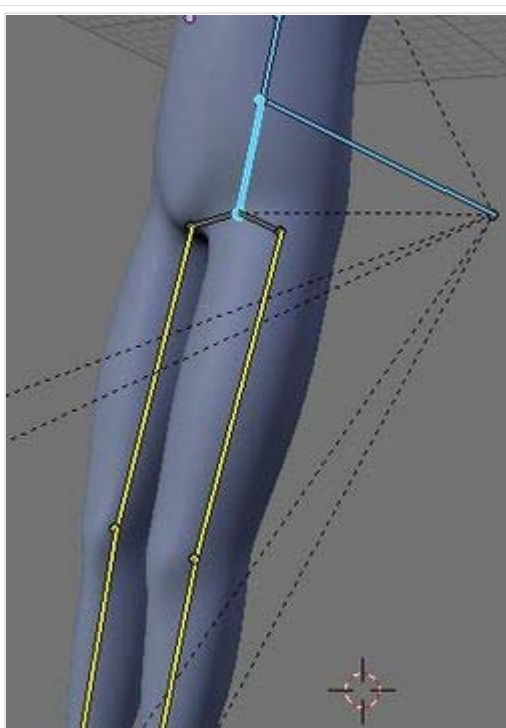
. . . for broad motions.

- For more control, turn off Auto IK and rotate the bones in the arm individually. This is the only way to get sharp angles in the arm joints. Don't forget to turn off Auto IK when you're done with it.

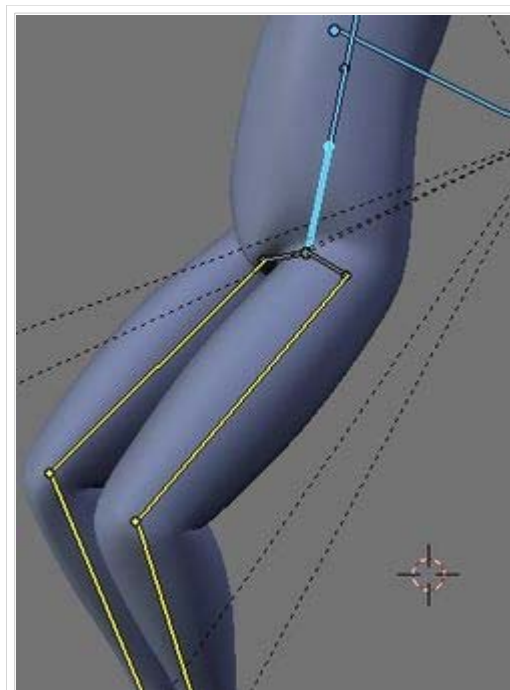


Turn off Auto IK and rotate individual bones for sharper angles.

- *master*: move and rotate this bone to manipulate the entire armature. For example, using the *master* bone is the only way to get the whole character off the ground in a jump.
- *spine1*: Move this bone to move the entire upper body while the feet stay put. Very useful for walkcycles.

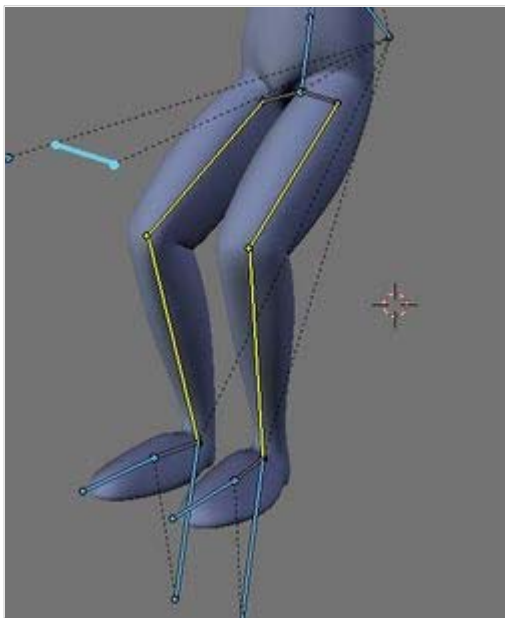


Move the *spine1* bone . . .

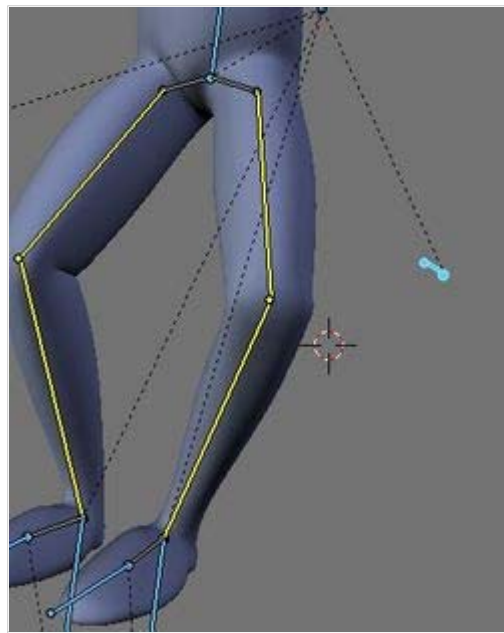


. . . to move the upper body while the feet stay put.

- *knee*: these bones are not used very often and are primarily used to keep the IK constraints on the legs under control. But when you need to point the knees in a direction, use these bones.



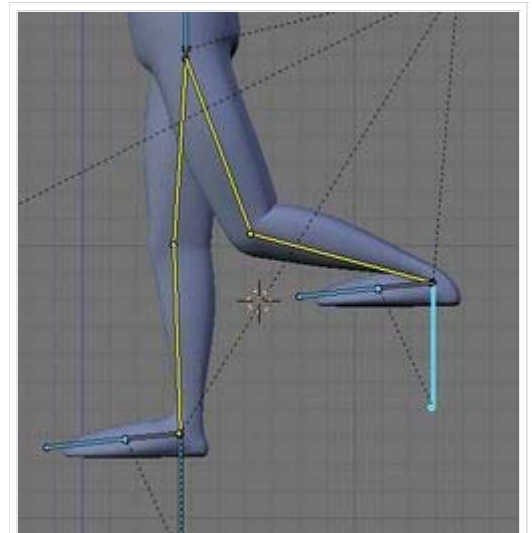
Move the *knee* bones . . .



. . . to point the knees.

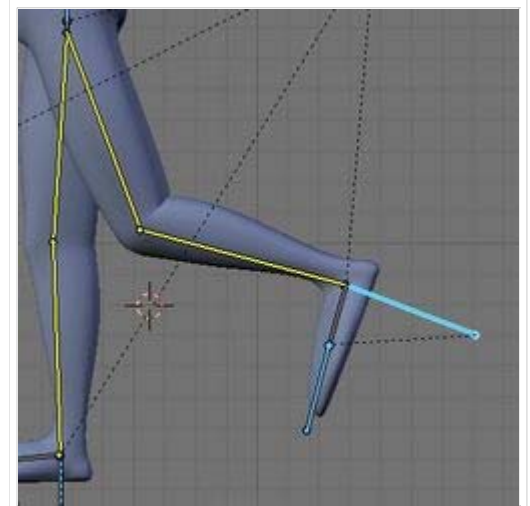
- *upper_leg* and *lower_leg*: leave these bones alone. The *leg* bones and IK constraints do all the work.

- *leg*: These are the most versatile bones. Move the *leg* bones to move the entire leg. **Be careful:** with this simple of a foot rig, you run the risk of stretching the feet away from the legs.



Move the *leg* bones to move the entire leg.

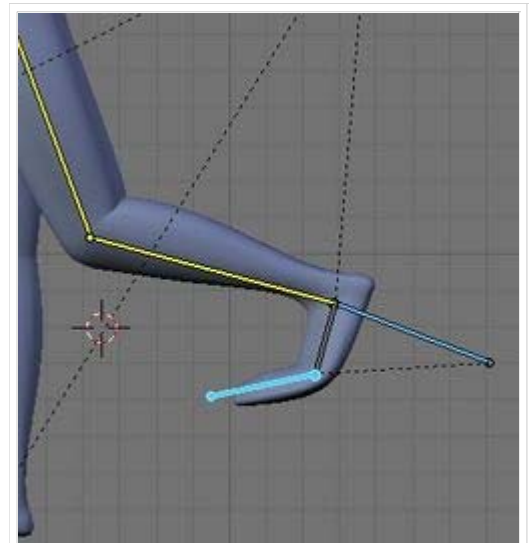
- Rotate the *leg* bone to rotate the entire foot.



Rotate the *leg* bones to rotate the foot.

- *foot*: leave this bone alone, it is rotated by rotating the *leg* bone.

- *toe*: Rotate these bones to bend at the toe.



Rotate the *toe* bones to bend the foot at the toe.

Next: Setting up for animation

Previous: Lower body: armature and weight painting

Back to Index

Retrieved from

["http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Final_rig_adjustments"](http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Final_rig_adjustments)

- This page was last modified 17:51, 27 July 2006.

BSoD/Introduction to Character Animation/Setting up

From BlenderWiki

< BSoD | Introduction to Character Animation

Now we'll use some of Blender's animation tools to create a simple action: a wave of the arm. This will be a very simple action that can be blended in with the walkcycle we'll create later.

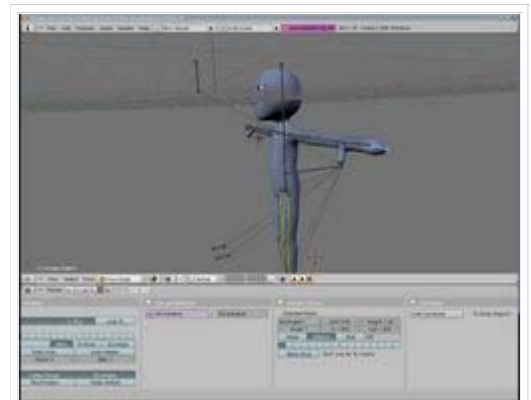
Contents

- 1 Setting up the workspace
- 2 Animating a "wave" with the Action Editor
 - 2.1 Moving to a different frame
 - 2.2 Finishing the wave


Setting up the workspace

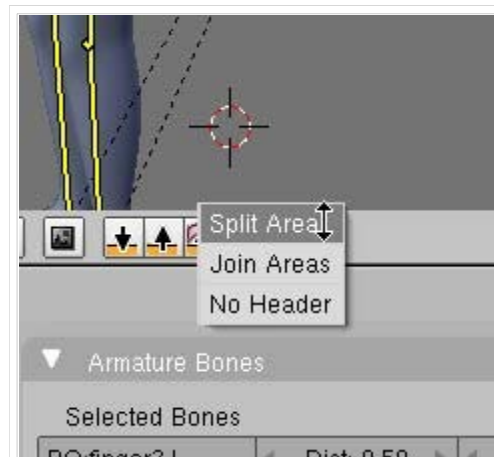
Up until now, we've been using primarily the 3D Window and the Buttons window. That's all we've really needed so far. Now, we're going to use some more window types while simultaneously viewing the 3D Window and Buttons window.

- The default workspace we've been using so far looks like this.



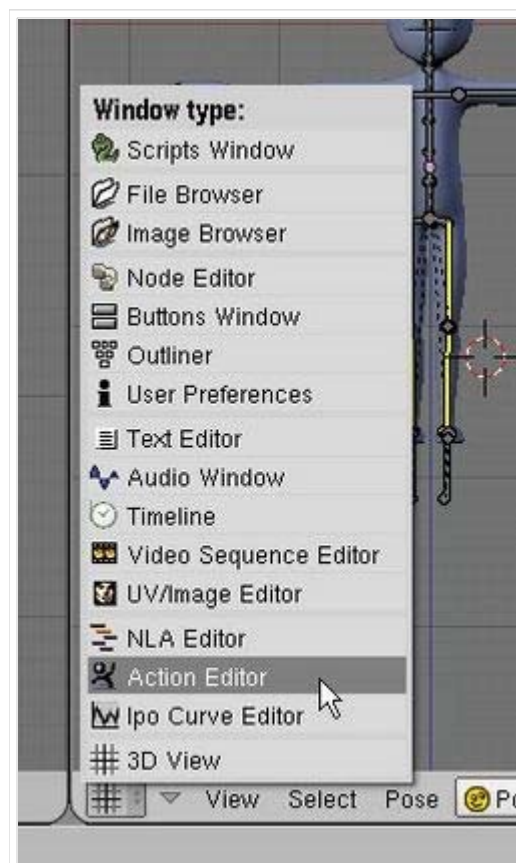
Default workspace.

- Click RMB  on the thin dark border between the 3D Window and the Buttons window. Choose *Split Area*.
- A vertical line will appear, this will be the dividing line for the new window. Move it where you'd like.



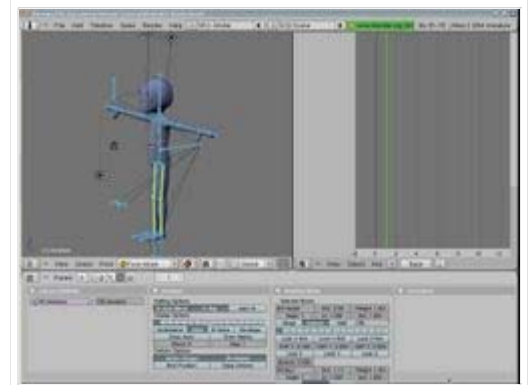
Splitting the 3D Window.

- A new 3D Window was created. You can grab the border to resize the window.
- Change this window to the Action Editor by choosing *Action Editor* in the Window Type menu in the header of the new window.




Change the new window to the Action Editor.

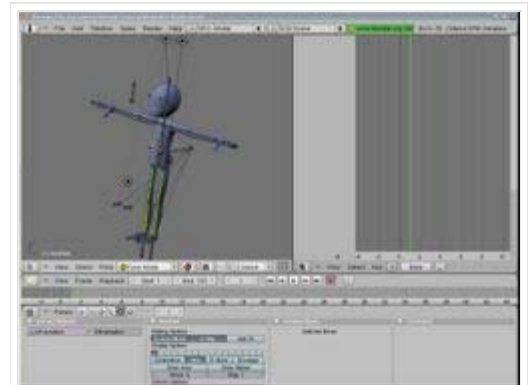
The Action Editor is where we will create individual actions: blinking the eyes, nodding the head, a walkcycle, and so on. Later we will mix the actions in another window.



Modified screen, with the Action Editor on the right.

Split the buttons window into a timeline:

- Use RMB  on the right or left edge of the Buttons window and Split Area.
- Split the Buttons window and change the new window to a Timeline window.

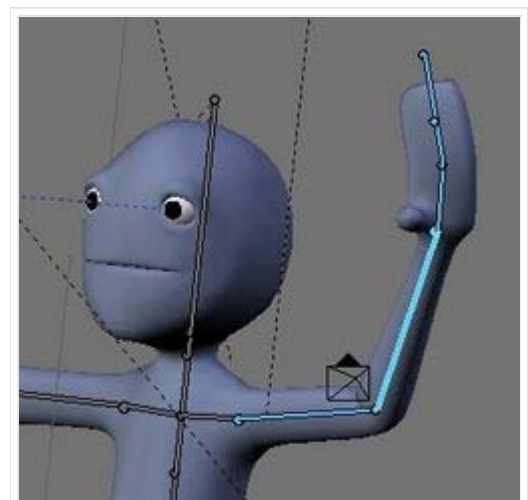


The new window setup.

Animating a "wave" with the Action Editor

We'll start out with a simple action to demonstrate the animation tools: a quick wave. Later, we'll mix this action with a walkcycle.

- Pose the arm. I started by enabling Auto IK to get the arm roughly in place, then turned off Auto IK and individually rotated the arm bones.

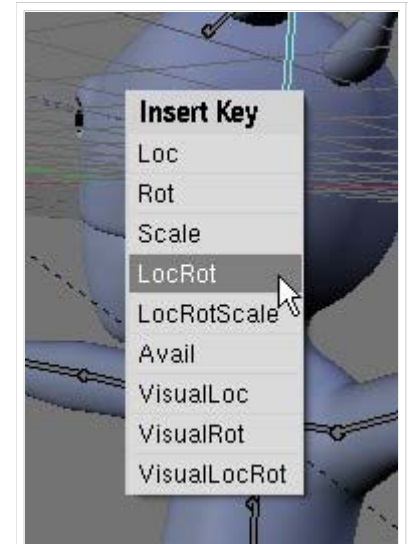


Position the arm.

With the bones in position, we can add keys. A **key** describes the orientation of a bone: it's location, rotation, and size.

Generally speaking, you only want to add keys for bones that actually did anything. In other words, we don't want to add any keys for the *master* bone or the *leg* bones because we didn't move them.

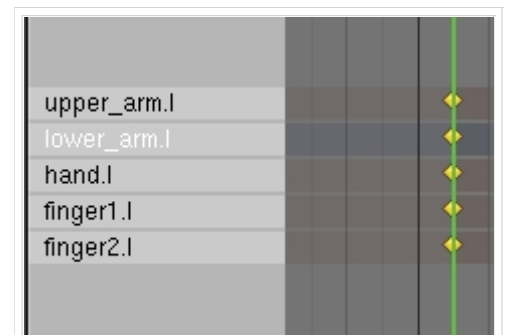
- Select the bones that you moved for this pose - in my case, it was all the arm bones (from *upper_arm.l* down to *finger2.l*).
- Insert a LocRot (location and rotation key) for the selected bones with **I** >> *LocRot*.



Add keys for Location and Rotation (choose LocRot from the menu).

Take a look at the Action Window. It has several rows, or **channels**, and each has a name that corresponds to a bone in the selected armature. Each channel is connected to the bone it is named after. When you select a bone, the corresponding channel is selected, and when you select a channel, the corresponding bone will be selected. In this image, the *lower_arm.l* channel is selected.


Also, note that some yellow diamonds popped up. These are the keys. They are lined up along the green line, which indicates what frame of the animation we're on. Currently we're on frame 1, and the keys were all added on frame 1.



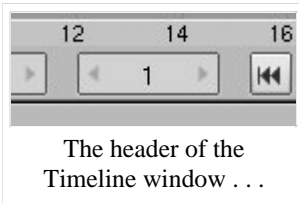
New keys for the bones in this pose.

Moving to a different frame

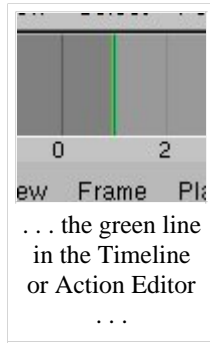
There are several ways to change which frame we're on in Blender, and which way you use depends on what window you happen to be closest to or if your hand happens to be on the keyboard. We can change frames by:

- using LMB  on the green line, either in the Action Editor or in the Timeline.
- using the arrow keys:
 - Right Arrow moves ahead one frame.
 - Left Arrow moves back one frame.
 - Up Arrow moves ahead 10 frames.
 - Down Arrow moves back 10 frames.
- Clicking on the frame counter in the header of the Timeline or Buttons window to move by the number by single frames, or drag the button to change the number quickly.

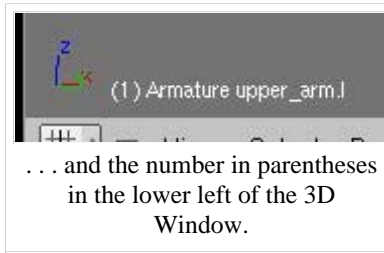
There are several indicators as to which frame we are currently on. The frame number is shown in:



The header of the Timeline window ...



... the green line in the Timeline or Action Editor ...



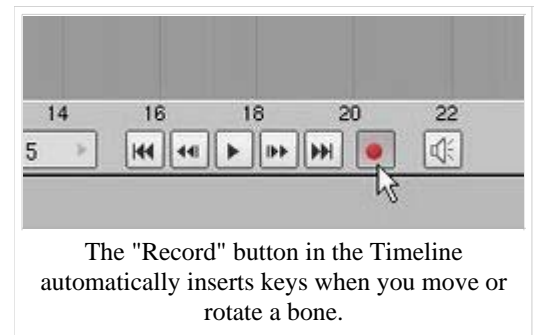
... and the number in parentheses in the lower left of the 3D Window.

Finishing the wave

- Using any of the above methods, move to frame 5.

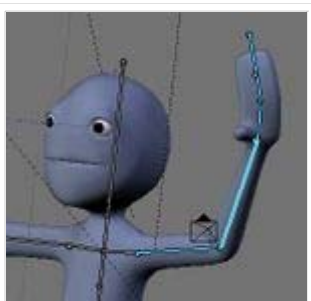
We've already inserted keys with **I**. When working on an animation, it can get tedious to select each bone and add a key. Instead, we can have Blender automatically insert a keyframe whenever we move, rotate, or scale a bone.

- To enable auto keyframing, click the "Record" button in the Timeline window. Blender will automatically insert a keyframe any time we move or rotate a bone. In other words, we won't have to remember which bones we moved and to insert keys for them.

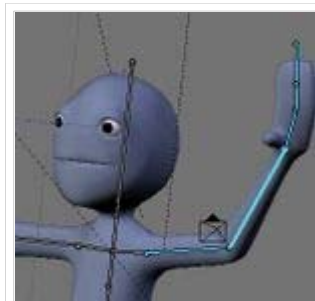


The "Record" button in the Timeline automatically inserts keys when you move or rotate a bone.

- To see how this works, rotate the *lower_arm.l* bone outward to wave the arm.



Pose at frame 1.



Pose at frame 5.

After rotating the bone, look back at the Action Editor, and note the following:



- The keys were automatically inserted for the arm bones that were moved, in this case, only the *lower_arm.l* bone.
- No key was inserted for the other arm bones, since they weren't moved in this frame.
- The old keys are deselected (white) and the new keys are now selected (yellow). Note that the keys from frame 1 are still selected since they didn't have any new keys added.





New keys for the 5th frame.

Important: *Bones remain where they were until you tell them otherwise.* Since we did not set another key for the *hand.l* bone in frame 5, it will stay in the same position as it was in frame 1.


- Move the frame slider (the vertical green line) to frame 1 and back again to frame 5 to view the motion.

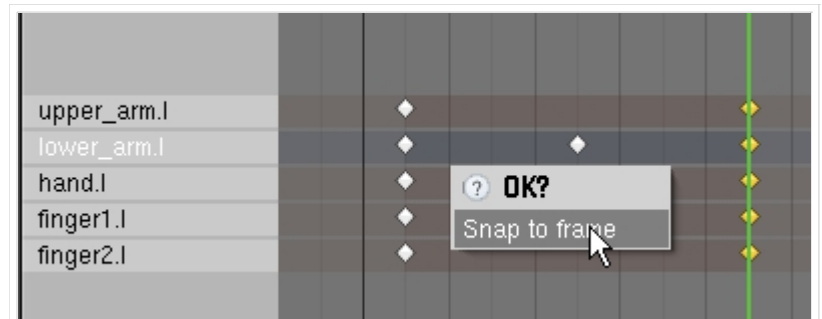
- Now move to frame 9.
- In the Action Editor, some familiar selection commands work:
 - **A** to select/deselect all keys.
 - **RMB**  to select a key
 - **Shift** **RMB**  to add a key to the selection.
 - **B** to bounding-box select.

We're going to duplicate the keys from frame 1 and move the duplicates to frame 10. **If you need to see the keys a little better**, you can zoom and pan in the Action Editor:

- Zoom in on the keys in the Action Editor with **MW** .
- Pan the Action Editor view with **MMB**  so the keys come into view.

Now let's duplicate the keys in frame 1 and move them to frame 9, so the Wave action starts and ends in the same position.

- Deselect all keys with **A**.
- Bounding-box select all the keys in frame 1 (**B**), draw a rectangle around the keys with **LMB** .
- Duplicate the selected keys with **Shift** **D**.
- This should seem familiar - we've automatically entered grab mode after duplicating! Move the duplicated keys to frame 9. Don't worry if it's not exact.
- Snap the selected keys to the closest frame with **Shift** **S**. Now the keys are exactly on frame 9.
- Move the frame slider back and forth from frame 1 to 9 to see the animation.



Snap the new duplicated keys to the closest frame.

- At the bottom of the Action Editor is a text box labeled "AC:". Name the action something meaningful here.



Name the action.

Summary: We split the interface into an Action Editor and a

Timeline window so we can access some animation tools. We created an action (a quick wave) by moving the bones and inserting keyframes in the Action Editor.

Next: Animating a walkcycle

Previous: Final rig adjustments

Back to Index

Retrieved from "http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Setting_up"

- This page was last modified 19:48, 13 August 2006.

BSoD/Introduction to Character Animation/2-second animation

From BlenderWiki

< BSoD | Introduction to Character Animation

Contents

- 1 Creating a walkcycle
- 2 Contact pose
- 3 Flipping the pose
- 4 Recoil pose
- 5 Passing pose
- 6 High point
- 7 Viewing the animation
- 8 Tweaking the walkcycle

Creating a walkcycle

- Click the small arrows right next to the action name, and choose *Add New*.
- Rename the new action "walkcycle" or something similar (can you guess what we're doing next?)
- Delete all keys with **A - A - X**. Creating a new action makes a copy of the previously selected action, so these keys are just copies.
- You can verify that the "Wave" action is still there by selecting it in the dropdown menu in at the bottom of the Action Editor.



Note: Two important notes on Actions:

- Each Object can have many different Actions, but only one *active* Action, which is displayed in the Action Editor.
- This active Action is the one which will receive any new keys you insert, and whose keys you can directly edit.

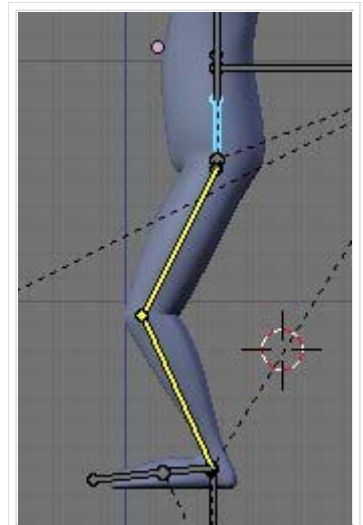
- Move the frame slider to frame 1.
- Clear the location and rotation of all bones in the armature with **A - A - Alt R - Alt G**. Notice that even though the "Record" button is still pressed, keys were NOT added when we cleared rotation and location. The automatic key insertion does not recognize clearing location or rotation as an actual movement (this is a good thing, but it's important to remember).

Contact pose

In a walkcycle, the contact pose is the point when the leading leg just touches the ground in front of the character. It's generally the first pose to animate in a walkcycle.

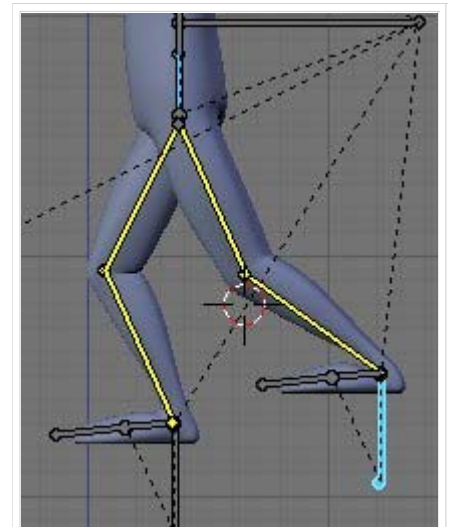
Note: It's impractical for me to take screenshots for each bone movement I make. I'll walk through the first pose (the contact pose), and from then on I'll provide screenshots of front, side, top, and oblique views of each major pose.

- In the 3D Window, switch to Side view in Ortho mode (NumPad **3** and NumPad **5**).
- **Make sure Auto IK is off**, otherwise the IK Solvers won't work for the legs.
- Move *spine1* down a little bit in the Z axis to let the legs bend.



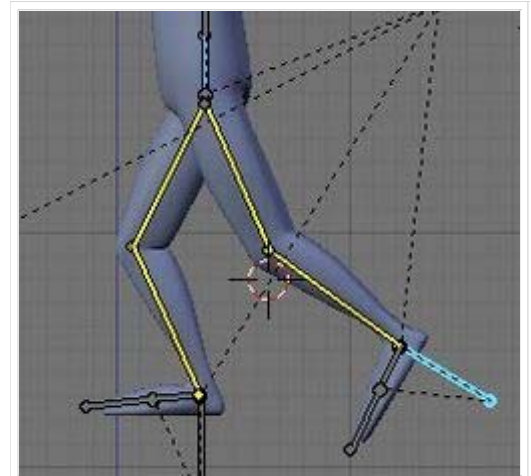
Move *spine1* down in Z.

- Select *leg.l*. You may have to move the view to see it before returning to side view. Alternatively, you can select the *leg.l* channel in the Action Editor to select the bone.
- Move *leg.l* back and up.



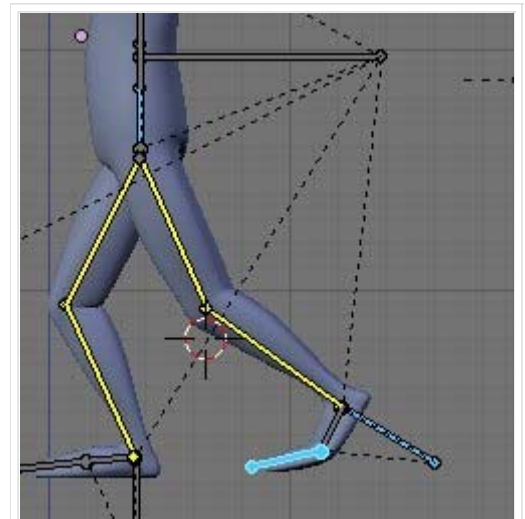
Move *leg.l* up and back.

- Rotate *leg.l* . . .



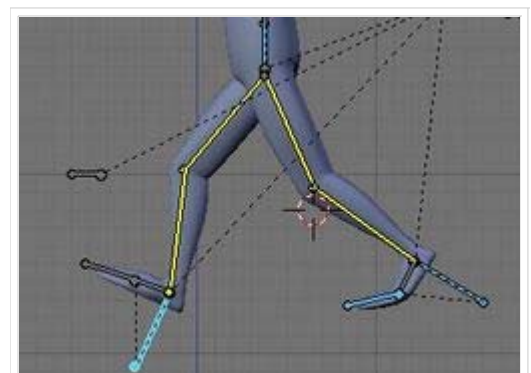
Rotate *leg.l*.

- . . . and rotate *toe.l* to bend the foot at the toe.



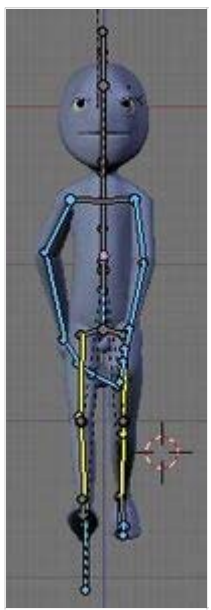
Rotate *toe.l*.

- Grab and rotate the *leg.r* bone in front of the body.

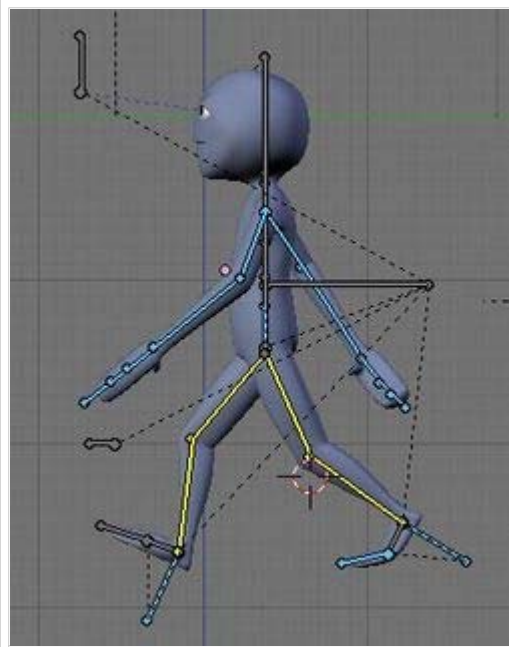


Move and rotate *leg.r*.

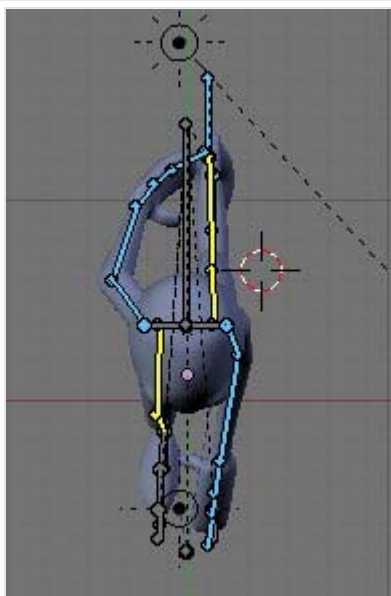
- Now pose the arms. I prefer to turn on Auto IK to get the general shape of the arms, then turn it off as soon as I'm done. You may have to move the view around to get the arms to move and bend in the direction you want. Keep in mind that the legs and arms move opposite: right leg forward means right arm back. Here are front, side, and top views after I posed the arms:



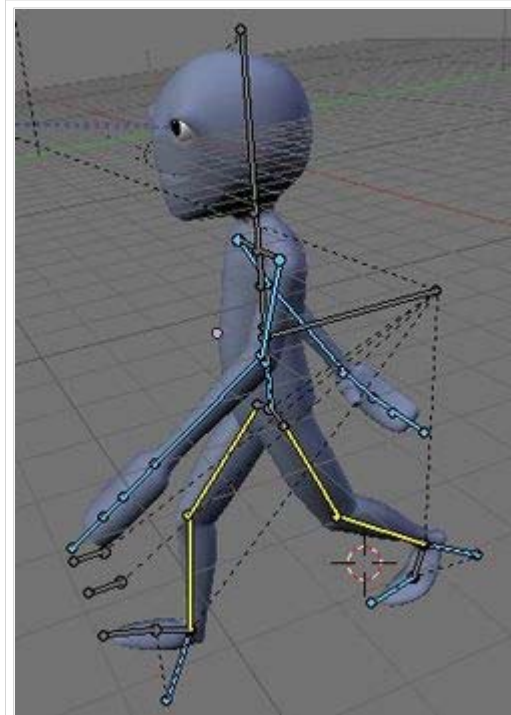
Front.



Side.

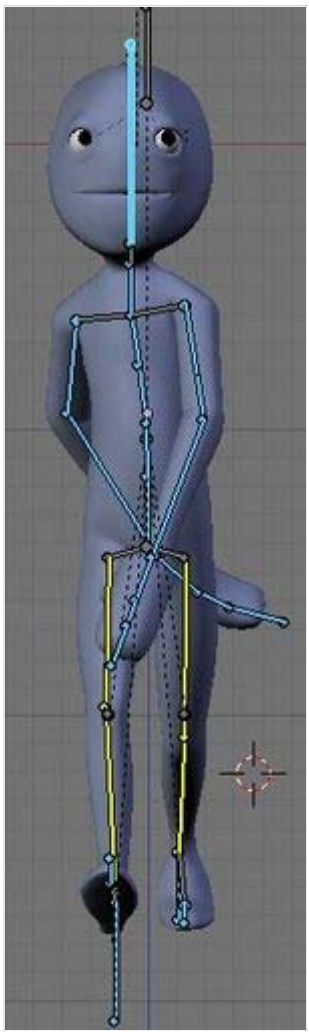


Top.

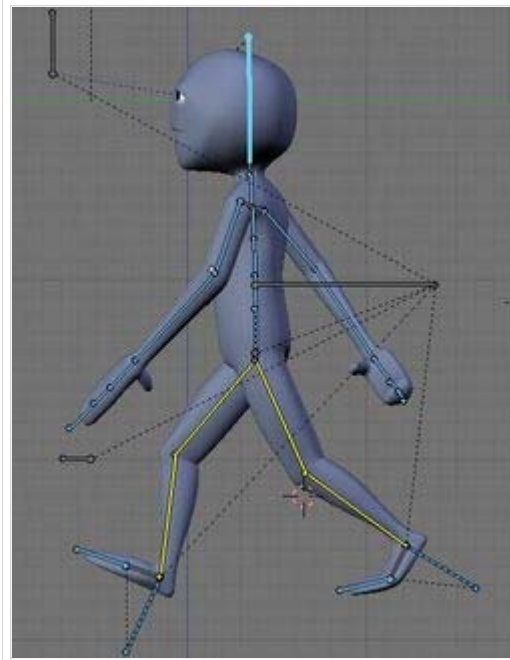


Oblique.

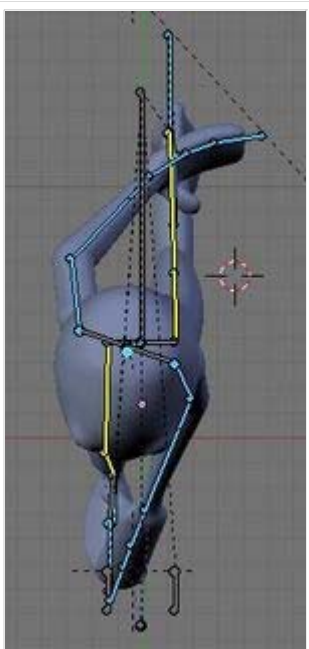
- Spend some time on the spine: when walking, the leading hand twists the top of the spine toward it, and the trailing arm twists the lower spine toward it. Be aware, though, that by rotating the lower spine bones, the upper spine, neck, and head will rotate as well. So you may have to compensate by rotating the upper spine bones back the opposite way.
- After rotating the spine, neck and head bones, I had a pose something like the images below. Notice the eyes continue to follow the *eyes* bone.



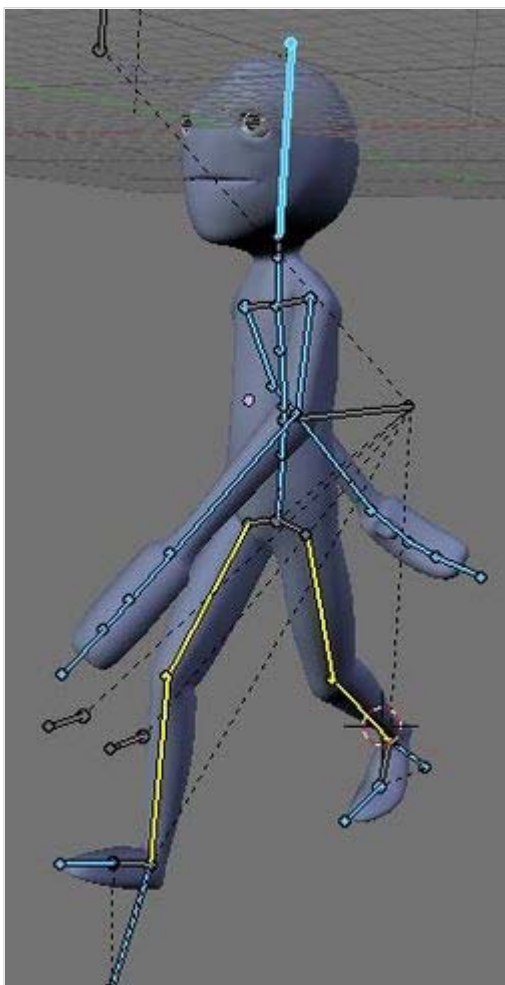
Front.



Side.




Top.

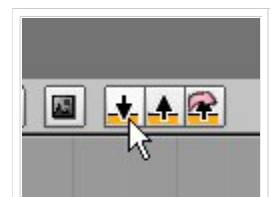


Oblique.

Flipping the pose

- Select all bones in the armature.
- Click the "Copy pose" button at the bottom of the 3D Window. This copies the keys (location, rotation, and scale) for all selected bones that already have keys.

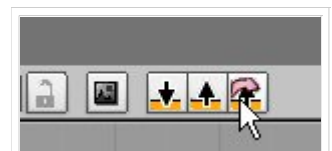
Note: You may have to use **MMB**  to pan the header of the 3D Window back and forth to find these buttons



Copy the pose from frame 1 . . .

- Move to frame 11 with **Up arrow**.
- In frame 11, click the "Paste flipped pose" button. This pastes the pose in a mirror-image of the pose we just copied, and is a very useful tool!

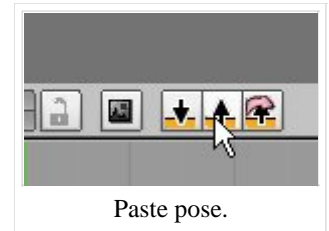
Note: This is one reason we went through all the trouble of naming the bones "l" and "r". Blender recognizes the bones are



. . . and paste the flipped pose in frame 11.

on the opposite side of the body and does the calculations to automatically insert the flipped pose for us.

- Move to frame 21 (remember you can zoom and pan in the Action Editor with **MW** and **MMB**, and **Up arrow** advances you by 10 frames).
- Click the "Paste pose" button. This pastes the pose we copied from frame 1 (WITHOUT flipping it) to frame 21, and is similar to using **Shift D** like we did for the "Wave" action.
- In the Action Editor or in the Timeline, move the frame slider back and forth between frames 1 and 21 so see the character walk. From the first contact pose to the opposite contact pose and back to the first pose takes 21 frames in this case. Everything else in the walkcycle will be filling in keys between frames 1 and 21.

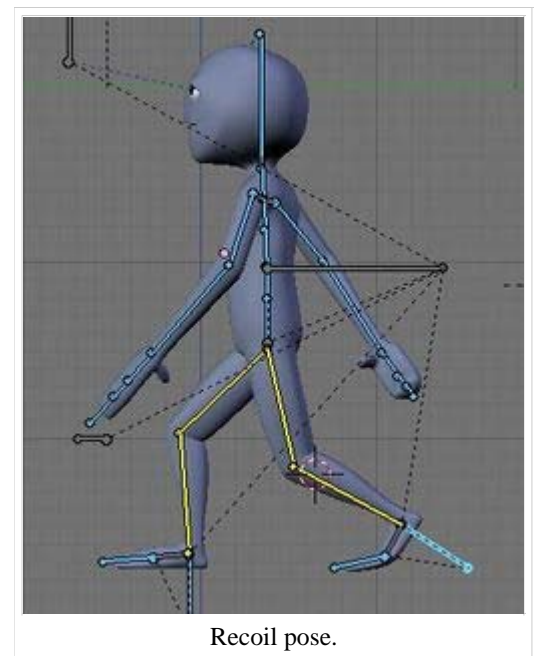


Recoil pose

We made a first cut at a walkcycle with just one pose and a couple mouseclicks! Next, we'll improve the walk by adding poses in between the ones we just created.

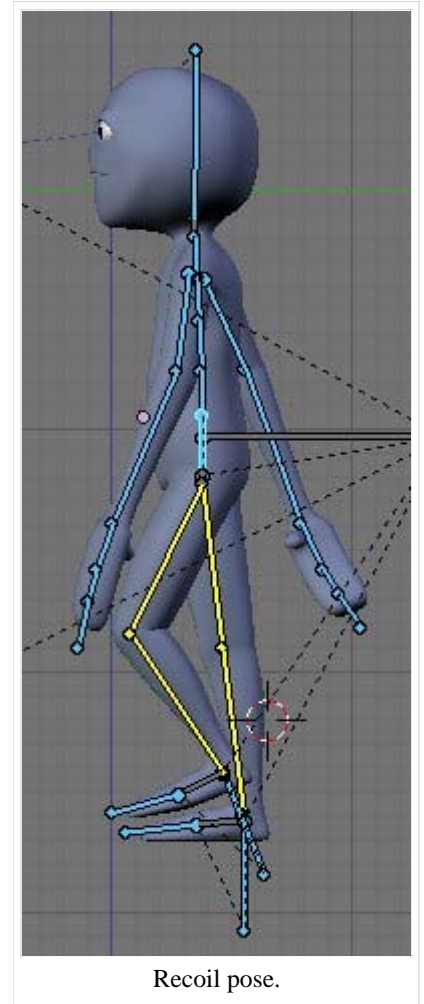
The recoil pose is when the front foot takes the weight of the body. The front foot flattens, and the body bounces down a little.

- Move to frame 2.
- Clear the rotation of *leg.r* with **Alt R**.
- Set a rotation key, but without rotating the bone, by pressing **R-Enter**.
- Grab the *spine1* bone and move it down a small amount. I ended up pressing **G**, then hitting the **Down arrow** a couple times to move it a small amount.
- Only Side view is shown, since it shows everything that changed.
- Copy this pose, and paste the flipped pose 10 frames later in frame 12.



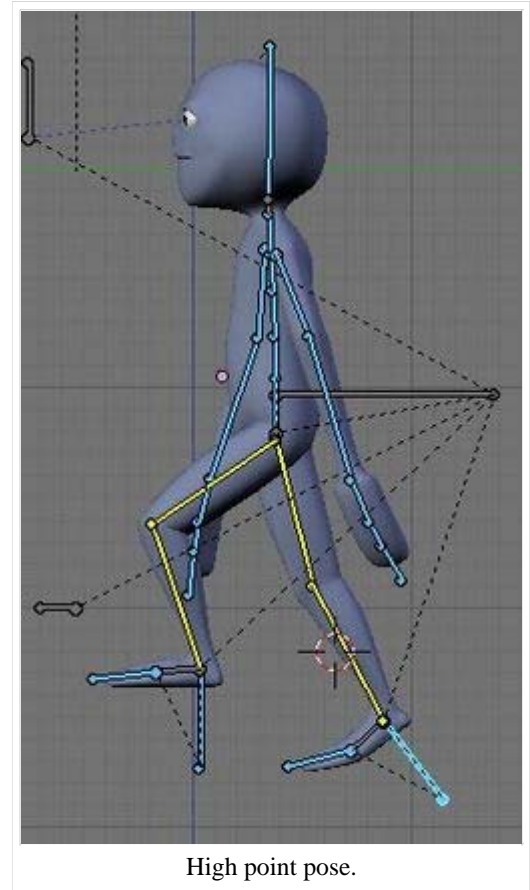
Passing pose

- Move to frame 6.
- Adjust *leg.r* and *toe.r* so that the foot is flat on the ground.
- Move *spine1* up so that the right leg is mostly straight.
- Once you're done, copy the pose and paste the flipped pose 10 frames later in frame 16



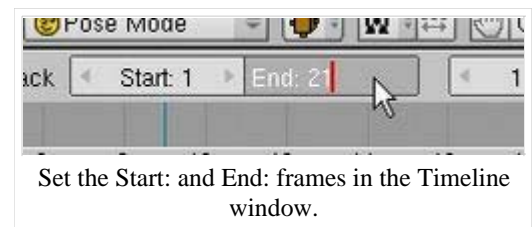
High point

- Move to frame 8.
- Rotate the *leg.r* so it points down more.
- Rotate *toe.r* so it hits the floor.
- Move *spine1* up a little bit. *spine1* should be at its highest point in this pose.
- Paste the flipped pose to frame 18.



Viewing the animation

- In the Timeline window, set the Start: frame to 1 and the End: frame to 21. This will only play the animation for these frames (they are the only frames where we have keyframes).
- In the 3D Window, press **Alt A** to start the animation. You can rotate the view around and even zoom while the animation plays (but you have to use **Ctrl MMB** instead of **MW**).



Tweaking the walkcycle

You could spend many hours tweaking a walkcycle. I added a little bit of a delayed head-bob and a little secondary movement on the hands and toes. For inspiration, check out the BioMotionLab Walker (<http://www.biomotionlab.ca/Demos/BMLwalker.html>) site.

You can download the .blend file so far here: `Media:WalkcycleTest.blend`

Summary: We created a new Action and animated a walkcycle in the Action Editor. Next, we'll mix the "Wave" action and the "Walkcycle" action together.

Next: Using the NLA Editor

Previous: Animating a simple action

Back to Index

Retrieved from

["http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/2-second_animation"](http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/2-second_animation)

- This page was last modified 17:43, 19 August 2006.

BSoD/Introduction to Character Animation/NLA

From BlenderWiki

< BSoD | Introduction to Character Animation

Contents

- 1 The NLA Editor
- 2 Convert an Action to an NLA strip
- 3 Mixing actions in the NLA Editor
- 4 Changing the stacking order of NLA strips

The NLA Editor

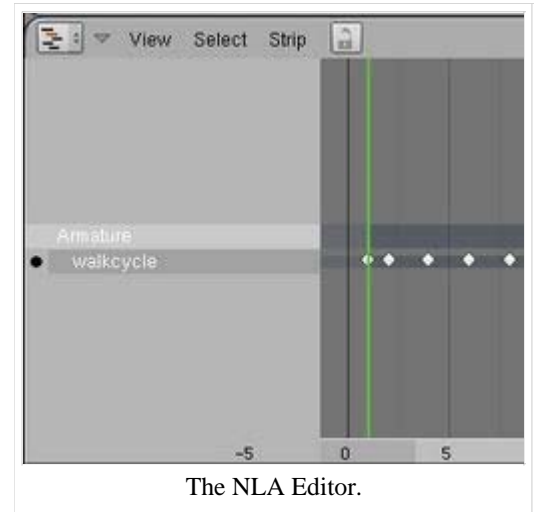
We now have two separate actions: "Wave" and "Walkcycle". There's another window in Blender called the **NLA Editor** (NLA stands for non-linear animation) where we can combine the two actions.

- Change the Buttons window to an NLA Editor window.

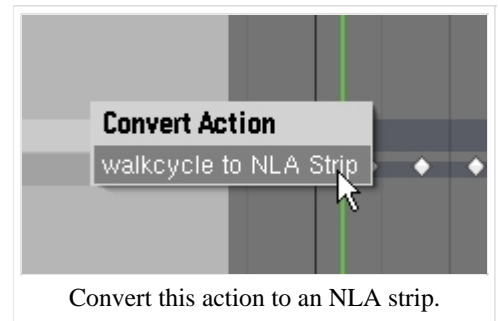


Change the window type to an NLA Editor.


This is the NLA Editor. It may not look like much now, but it's quite powerful. Right now there are only two rows. "Armature" refers to the object that has actions (our armature). "Walkcycle" refers to the current action selected in the Action Editor. The diamonds indicate where there is a keyframe on any bone in the walkcycle action.

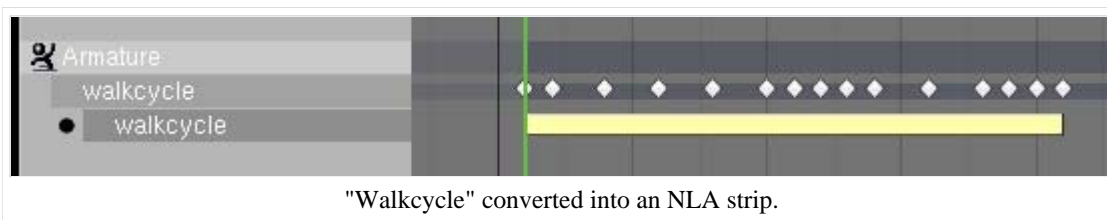


- In the Action Editor, select the "Wave" action from the dropdown box.
- In the NLA Editor, note how the row under Armature changed, and there are only three diamonds representing the three frames we have keys in the "Wave" action.



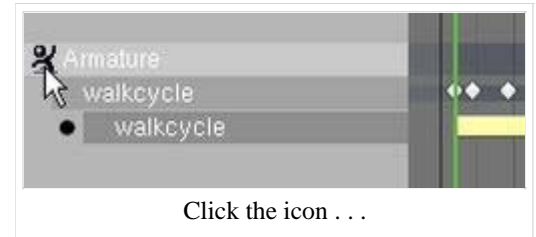
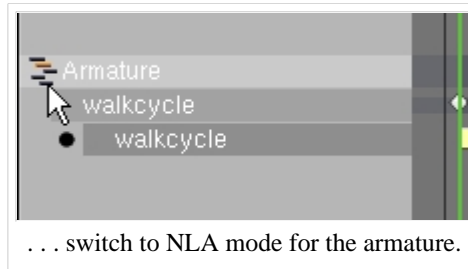
Convert an Action to an NLA strip

- Switch back to the "Walkcycle" action by selecting it in the Action Editor menu.
- In the NLA window, select the Walkcycle channel (the line with the diamonds, or keys) with RMB .
- Press **C** to convert this action into an NLA strip.

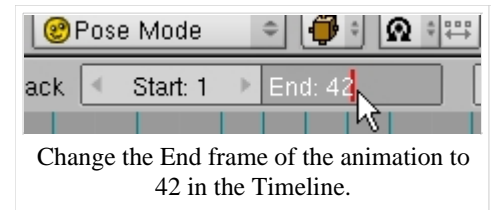


By converting the walkcycle into an **NLA strip**, we've "packaged" the walkcycle into a form that is easily manipulated for longer animations.

- **Important:** Click the "action" symbol next to the "Armature" row in the NLA Editor. (Also affectionately called the "shark-attack victim" icon). It doesn't look like a button, but it is. It will turn into an icon with strips, indicating that playback will use the NLA Editor and not the Action Editor. It doesn't matter right now with only one action in the NLA Editor, but it will matter later when more actions are added.

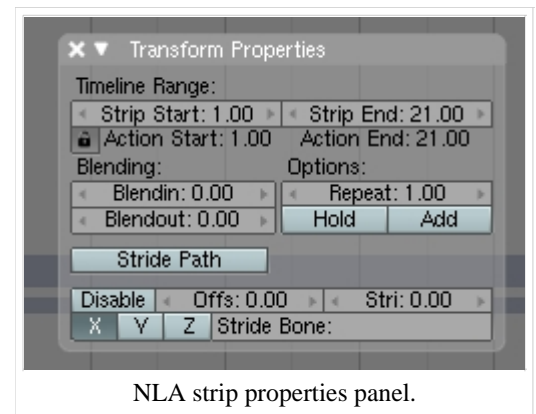


- Change the End: frame in the Timeline window to 42.

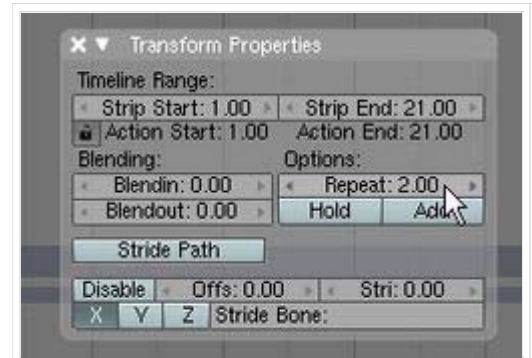


- Press **Alt A** in the 3D Window. Note that the character takes two steps and then stops at frame 21 because that's how long the walkcycle is. This is about to change ...
- In the NLA Editor, press **N** to bring up the strip properties panel.

Important: Most of the controls for the NLA Editor reside in this panel.

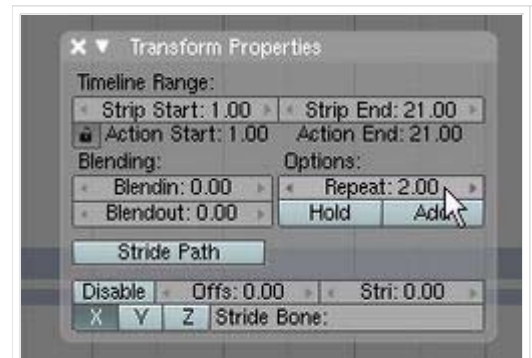


- In the Repeat number box of the strip properties panel, change the number of repeats to 2.
- Now play the animation with **Alt A** in the 3D Window. The character takes 4 steps now, because the walkcycle action of 2 steps was repeated 2 times. However, the walkcycle is very fast because those 4 steps are being taken in only 21 frames. Twice the number of steps in the same amount of frames = twice the speed.



Change the number of repeats to 2.

- In the strip properties panel, change Strip End to 42.
- Play the animation again. Now there are 4 steps, but they are happening over the course of 42 frames and so are at the same speed as the original action.



Change the Strip End to 42.

By entering numbers in the strip properties panel, you can adjust the speed and number of steps of a walkcycle. You can also adjust the speed by scaling the NLA strip with **S**.

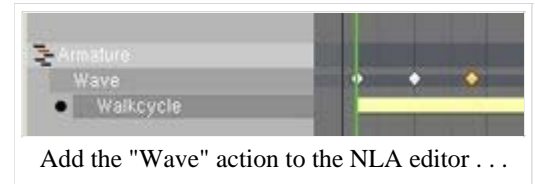
- Repeat the walkcycle action a total of 5 times.
- Make sure the frame slider is at frame 1.
- Instead of typing in numbers to keep the timing exactly the same (you'd type 105 into the Strip End number box to get the same speed as the original action), scale the strip with **S**. Tip: it's easiest if you have the mouse cursor above the range of the strip before scaling, otherwise the scaling directions get reversed. I scaled the strip up to about 120 frames; this will slow down the steps a little bit. Now we have 120 frames of walking! **Important:** To see all frames of the animation, you have to change the End: frame in the Timeline window to at least the last frame you have keys for, in this case, I would change the End: frame to 120 in the Timeline.

Note: When you use **S** in the NLA Editor and the Action Editor, the scaling occurs **relative to the frame slider**.

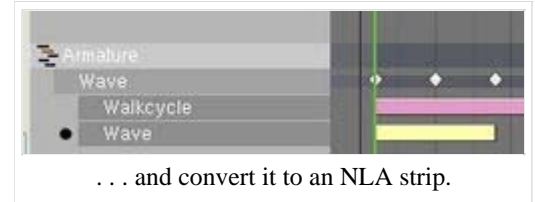
It also matters where the mouse cursor is when you start scaling. If the scaling doesn't act the way you think it should, try again either with the frame slider in a different position, or the mouse cursor in a different position.

Mixing actions in the NLA Editor

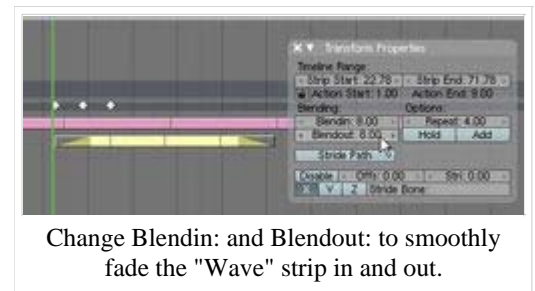
- In the Action Editor, select the "Wave" action. Note that once you select it, it appears in the NLA editor along with the three keyframes of the action.



- Convert this action into an NLA strip with **C**.



- Make the "Wave" action repeat 4 times within the NLA strip using the Repeat number box in the NLA strip Properties panel.
- Lengthen the strip either by scaling or by changing Strip End to something like 50.
- Move the strip later in the animation by selecting it and pressing **G** to move it. I moved it to around frame 23. This is why the NLA Editor is so powerful: once actions are defined in the Action Editor, you can add them one at a time to the NLA Editor. Once in the NLA Editor, you can move, scale, and mix the actions.
- Press **Alt A** in the 3D Window to view the animation. The wave and the walkcycle happen simultaneously, however, when the hand comes back down, it's a pretty abrupt motion. To fix this:
 - With the "Wave" action selected, change the Blendin: and Blendout: number box in the NLA strip properties panel to 8 frames or so. The NLA strip for the "Wave" action changes to reflect this fading in and fading out.
 - Play the animation again, the wave should be much smoother.



Changing the stacking order of NLA strips

The order of the strips in the NLA Editor matters. On top is the "Walkcycle" action, and under it is the "Wave" action. **Strips on the bottom override strips above them.**

In other words, the "Walkcycle" action has keys for all the arm bones. The "Wave" action ALSO has keys for the left arm bones. Since the "Wave" strip is below the "Walkcycle" strip, it overrides any conflicting keys.

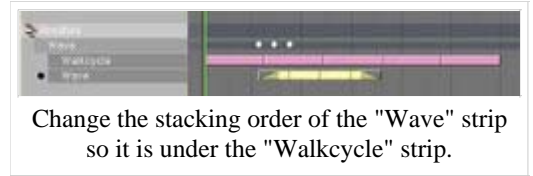
To change the stacking order of a selected strip, press **PgUP** and **PgDn**. Try this:

- Select the "Wave" strip.
- Move it up one row with **PgUp**.
- Play the animation.



The wave no longer happens. This is because the "Walkcycle" keys for the left arm bones override the "Wave" keys for the same bones.

- Select the "Wave" strip.
- Move it down one row with `PgDn`.
- Play the animation, and the wave should appear again.



Summary: We used the NLA Editor to convert the actions we made into NLA strips that can be easily manipulated in the NLA Editor.

Next: Creating facial expressions

Previous: Animating a walkcycle

Back to Index

Retrieved from "http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/NLA"

- This page was last modified 17:46, 19 August 2006.

BSoD/Introduction to Character Animation/Shapes

From BlenderWiki

< BSoD | Introduction to Character Animation

Contents

- 1 Shape keys
- 2 Creating the first shape key
- 3 The Open shape
- 4 The Wide shape
- 5 Mixing shapes
- 6 The Narrow shape
- 7 Shapes for the eyelids
- 8 The shape keys set

Shape keys

Eventually we'll add sound to this animation and lip sync the character to the sound. In order to do this, however, we need to make mouth shapes and facial expressions.

Just like we used keys to describe the position of a bone at one point in time, **shape keys** describe the position of vertices within a mesh.

Shape keys are most often used for facial expressions. We will create some basic shape keys like "open", "wide", "narrow", "brows up", "brows down", and "eyes closed". Then we can mix the facial expressions together in an animation, and lip sync to a sound file.

I should say up front that the process of creating a set of shape keys can be quite time consuming. Vertices have to be moved individually, and it's tough to get just the right expression. Just take your time and have fun giving the character life.

Creating the first shape key

First, we need to make the armature stay still so we can work on the facial expressions without the character walking around. To do this,

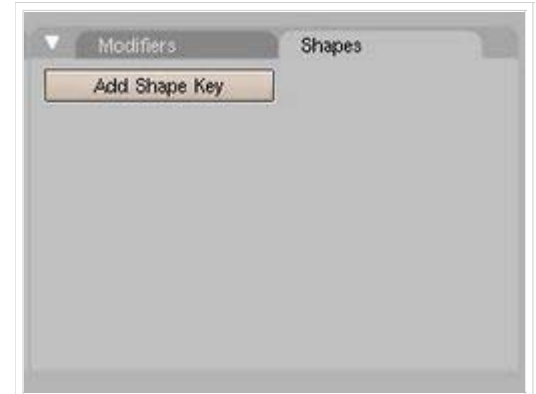
- Select the armature
- In the Armature panel under the Edit buttons, press the *Rest Position* button. Now the armature will stay still until we press the Rest Position button again.



Press the *Rest Position* button to make the armature stay still.

Now to create some facial expressions: that means more modeling in Edit mode.

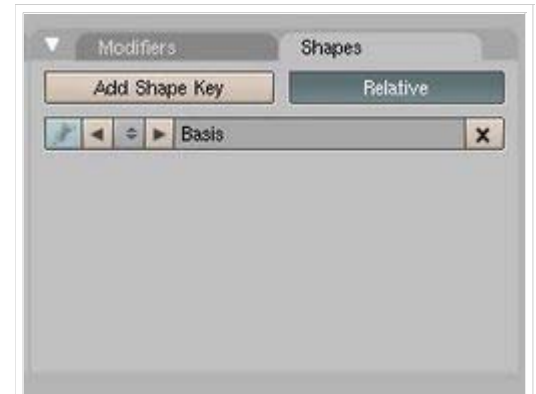
- Select the character mesh.
- In the Edit buttons, look for the Shapes panel. It is often seen as a tab next to the Modifier stack. Keep in mind that a mesh has to be selected before these controls appear.



The Shapes panel.

- Click the *Add Shape Key* button. This created a shape based on the current shape of the mesh, *ignoring the effects of the armature*. The only thing a new shape records is the shape of the vertices in Edit mode.

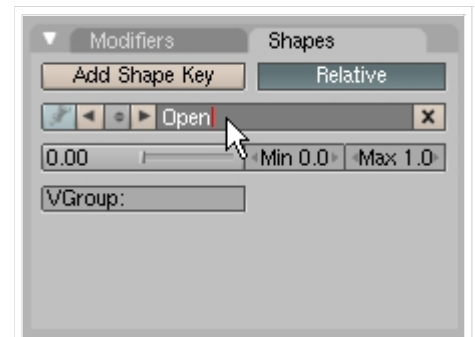
The **Basis** shape is the default shape that we modeled the character in, so we want to leave it alone and not make any changes. It is the reference shape that all other shapes will be based on.



The Basis shape.

- Add another Shape Key. A new shape key is a copy of the previously selected one. In this case, it's a copy of the Basis shape.
- Name the new shape "Open".

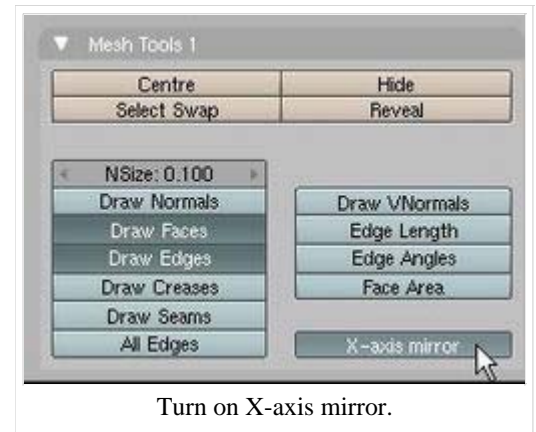
Note: From now on, we'll have to be careful when we switch into Edit mode. Whatever shape is selected in this panel is the shape that will be edited when you enter the mesh's Edit mode.



The new Open shape.

- In the Mesh Tools 1 panel (also under the Edit buttons), press the *X-axis mirror* button. This allows us to work symmetrically on the mesh.

OK, so why didn't we use a Mirror modifier? It turns out that if we used a Mirror modifier we would lose all shape keys and weight painting. OK, why didn't we use this X-axis mirror button when making the character? Because this tool only lets us move, rotate, and scale vertices -- it can't handle extrusions. And we sure did a lot of extrusions! The point is, each tool has its limitations. We are done with extrusions and will just be moving vertices around, so the X-axis mirror tool is perfect for us right now.

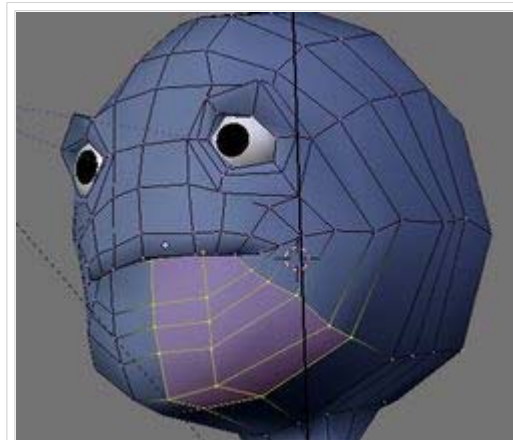


The Open shape

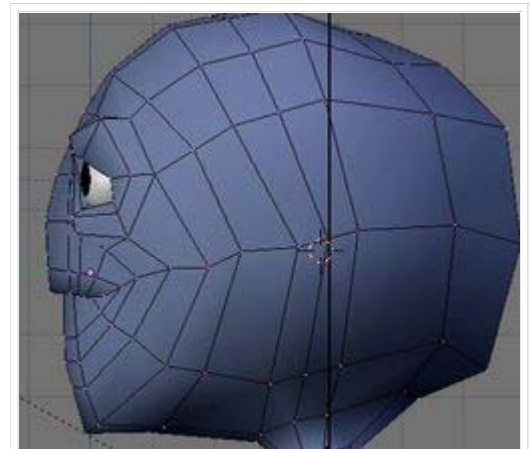
We're going to start modeling the "Open" shape. Remember, we saved the original state of the mesh by creating the "Basis" shape, so it's safe to move vertices around. We could move vertices one by one to open the mouth, but there's an easier way. First, keep in mind that when you open your mouth, your jaw doesn't just drop straight down, it *swings* in an arc, hinging at the back of your jaw. We can easily simulate this:

- Enter the mesh's Edit mode.
- Select a vertex on the centerline of the front of the face.
- Snap the 3D Cursor to the selection with **Shift S** >> Cursor to Selection. This gets the 3D Cursor on the centerline.
- Switch to Side view.
- Place the 3D Cursor in Side view, Ortho mode (NumPad 5) in the spot roughly as shown. The cursor was on the centerline from the previous step, and we can't move the cursor left or right of center in Side view (we can only move it forward/back and up/down), so the cursor will still be in the center of the head.

- Select the vertices that make up the lower jaw. Since we have X-Axis mirror enabled, we only have to select one side. Don't worry about the vertices on the inside of the mouth, we'll get them shortly. Do, however, try to get the lower lip vertices.

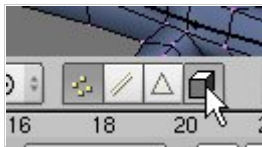


Jaw vertices selected.



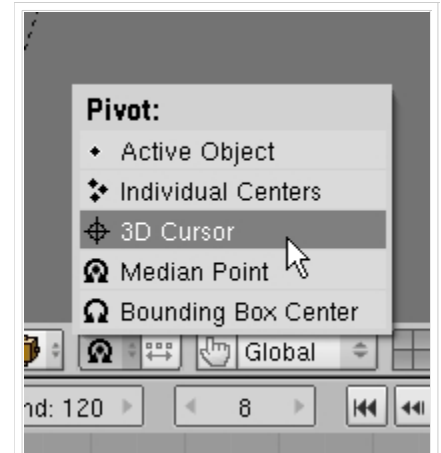
Setting the rotation center for opening the jaw.

Remember: If you can't seem to select a vertex or can't see it, use **Z** to switch to Wireframe mode, select the vertex, then switch back to shaded mode. Or, you can turn off the "Limit selection to visible" button at the bottom of the 3D Window.



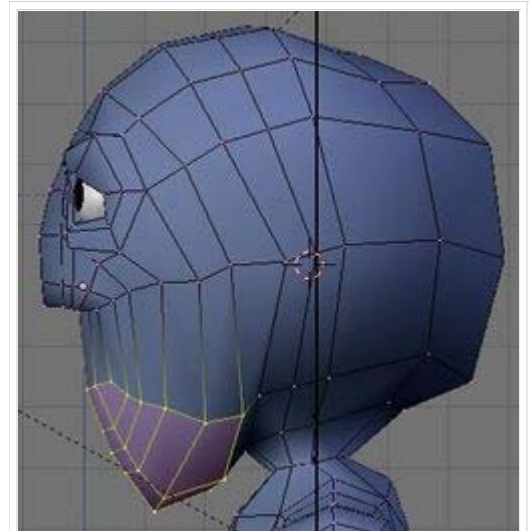
Limit selection to visible button.

- Switch to side view.
- Change the Pivot center to the 3D Cursor.



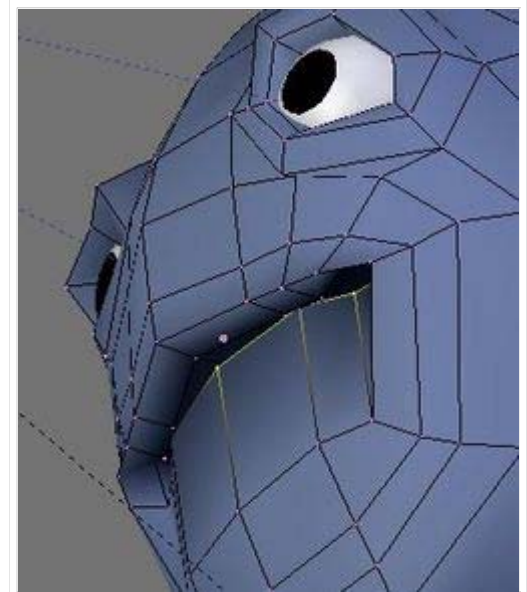
Select the 3D Cursor as the pivot point.

- Rotate the jaw around the 3D Cursor (**R**). That got us the general shape we wanted. Now we have to get the inside of the mouth to open, as well.



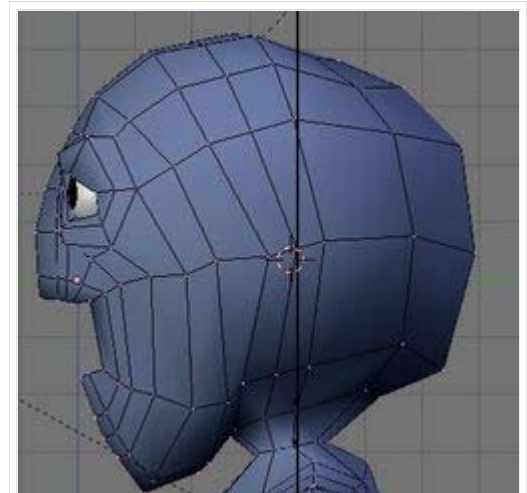
Jaw vertices rotated around 3D Cursor.

- Select the vertices on the inside of the mouth that you want to move with the lower jaw.



Inside of mouth selected.

- Rotate them around the 3D Cursor in Side view, as well.

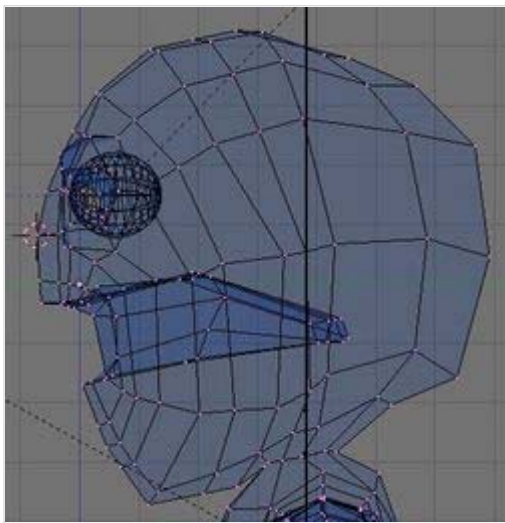


Inside of mouth rotated downward.

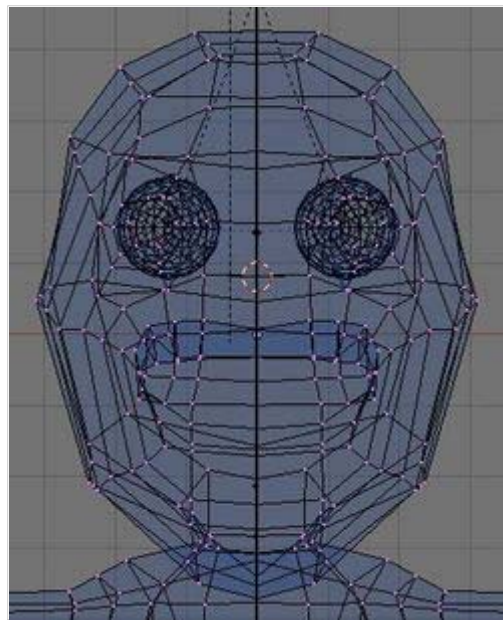
In the screenshots below, I spent some time to smooth out the vertices near the throat by moving them by hand. I also moved some vertices around in the inside of the mouth. Building shapes like this takes a lot of time, and is an iterative process. We'll have to see how this shape behaves when it is combined with a "wide" shape, and make some changes as necessary.

The bottom line is, it takes time and practice to build facial expressions. Don't get frustrated, and keep playing around. Sometimes it's useful just to create a new shape and start over rather than try to fix a shape you've been working on.

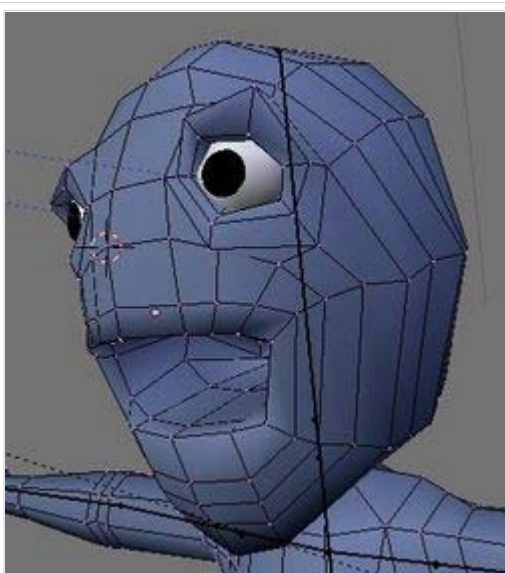
Here are three views of the "Open" shape after spending some time moving vertices around. The first two are shown in wireframe to better show the position of the vertices.



Open shape, side view.

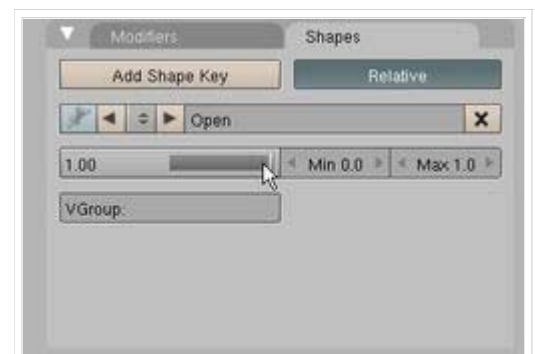


Open shape, front view.

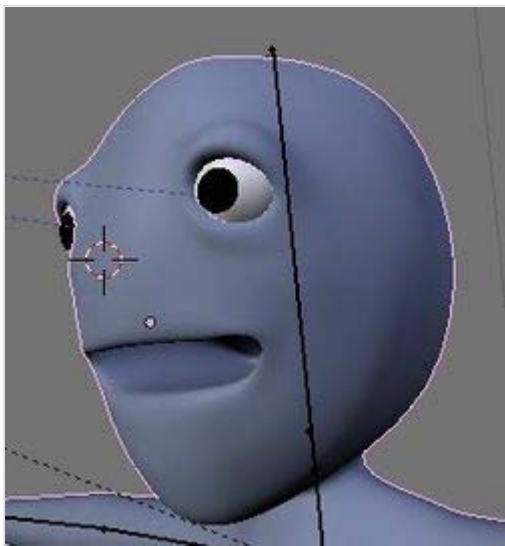


Open shape, oblique view.

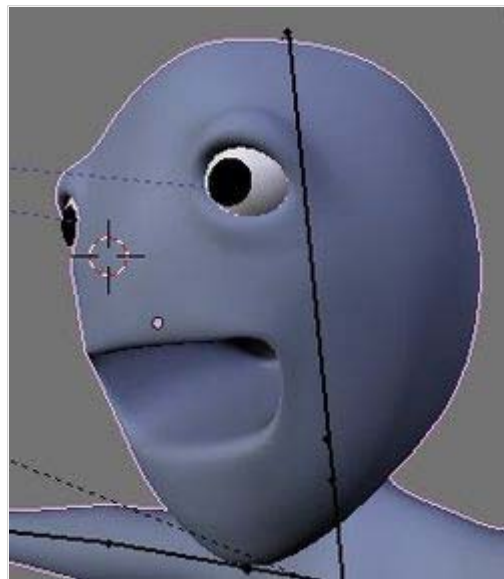
- Switch back to Object mode.
- Move the slider in the Shapes panel to determine how much of the selected shape (in this case, the "Open" shape) should be applied.



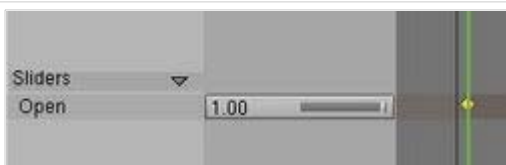
Set a key with the percentage of this shape.



Open shape, 50%.



Open shape, 100%.



New keys in Action Editor.

Note that when you move the slider, a key appears in the Action Editor, as well as a slider. I prefer to use the slider in the Action Editor instead of the one in the Shapes panel, especially for mixing multiple shapes (they both work the same way). Try this:

Set a key for the "Open" shape, say, 0%, at frame 1.

Move to frame 11 with Up arrow.

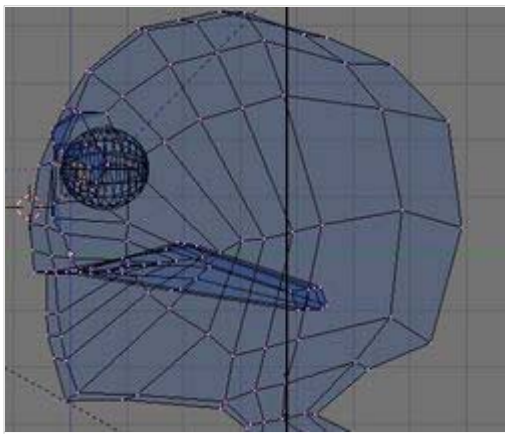
- Set a key for the "Open" shape, say, 100%.

- Move the frame slider back and forth to see the mouth open and close!
- When you are satisfied with your "Open" shape, delete the key in frame 11 (by selecting it and hitting X), move the frame slider to frame 1 and set the "Open" shape back to 0%.

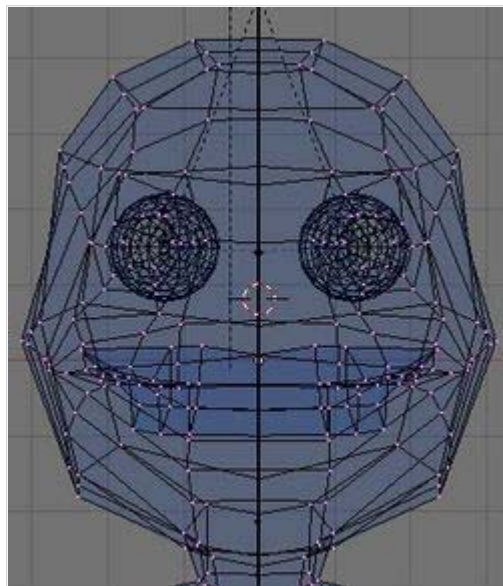
The Wide shape

Now we'll make a "Wide" shape.

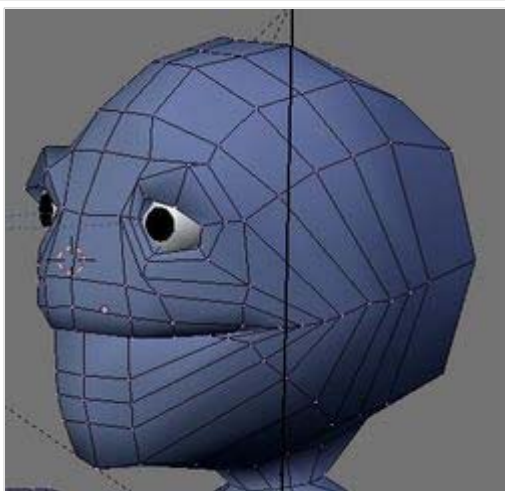
- Make sure you are in Object mode.
 - Select the Basis shape in the Shapes panel, by using the arrow buttons next to the shape name.
 - Press *Add Shape Key* to add a new shape. When you add a new shape, it turns into a copy of the previously selected shape. We want to start with the original shape of the mesh, not the Open shape, so that's why we had to select Basis first.
 - Name this shape "Wide".
 - If you want, you can set the strength of this shape to 100% now, so that when you switch back and forth between Object and Edit mode you will see the "Wide" shape in both modes. Otherwise, you will see the shape in Edit mode but in Object mode the shape will do whatever the Action Editor says it should be doing in that frame.
- Switch to Edit mode to start making the "Wide" shape. It's impractical to go step-by-step, so here are screenshots of the "Wide" shape I made. This shape initially looks like a smile - in fact, adding some eye squints would make it a smile. However, we also want to use it for making an "EEE" sound (when combined with the "Open" shape) so to serve this dual purpose, we don't want the eyes to squint (we can make an eye squint shape later, and mix it with this shape).



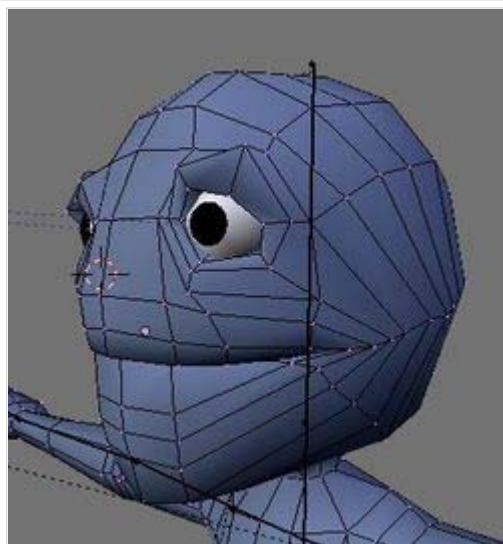
Wide shape, side view.



Wide shape, front view.



Wide shape, oblique view (Ortho).



Wide shape, oblique view (Perspective).

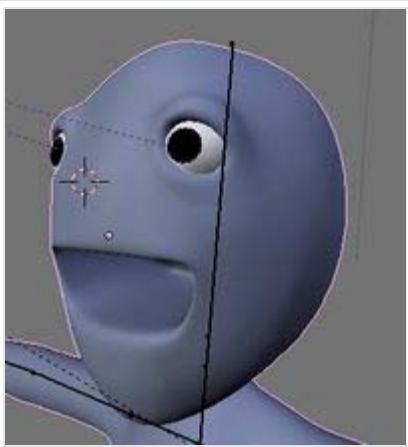
Some tips on the "Wide" shape:

- Take your time! There's no quick way to do this.
- Make sure you have Subsurf turned OFF in Edit mode
- Move one vertex at a time. It's tedious, but that's how it's done.
- Constantly change your view - often you will need to check the movement of a vertex in several views to make sure it's not making a dent or crease in the face where you don't want it to.
- The corners and inside of the mouth are tricky. Switch to wireframe mode, select one vertex, and rotate the view around to get a feel for what part of the mouth it is (inside upper lip? lower lip?). Then move it to where it should be.

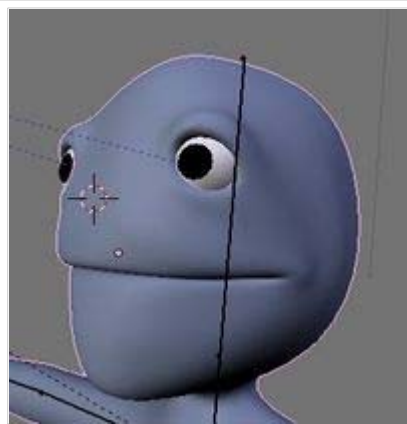
Mixing shapes

- When you're happy with the "Wide" shape, switch to Object mode.
- In the Action Editor, mix the shapes by moving the sliders. Here's what my shapes looked like individually and

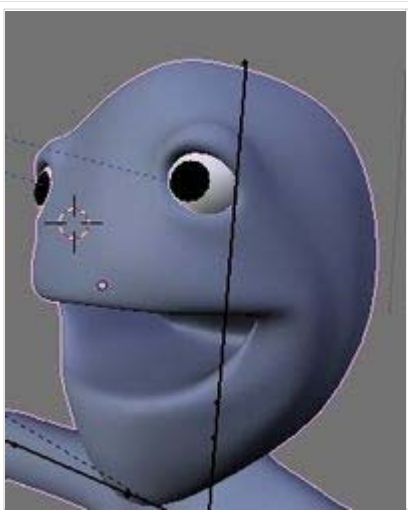
then mixed:



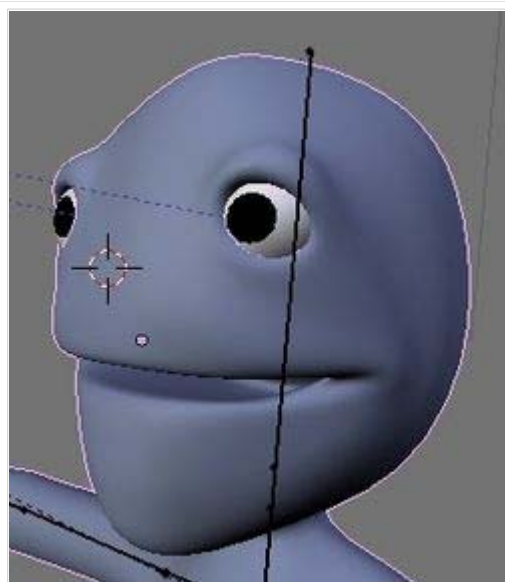
Open shape, 100%.



Wide shape, 100%.



Open 100% + Wide 100%.



Open 25% + Wide 100%.

If your shapes look strange while mixing, play with the sliders to try to figure out what's going wrong. This is one of those things that just takes some experience to figure out. If you'd like, you can download the .blend so far and take a look at the shapes.

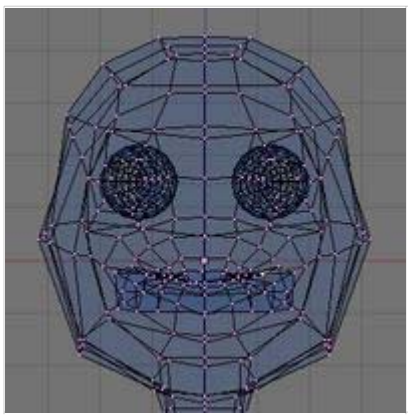
Media:BSoD-ItCA-shapes.blend

The Narrow shape

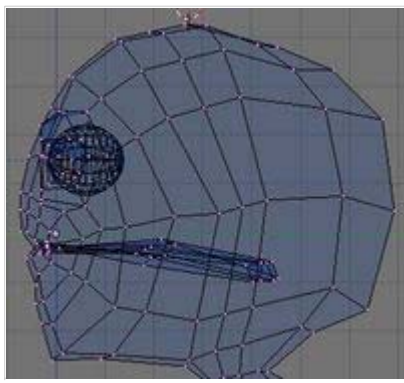
By now, you've got the basic idea:

- Switch to Object mode.
- Switch to the Basic shape.
- Add a new Shape Key.
- Name it (this one will be "Narrow").
- Switch to Edit mode to start shaping.

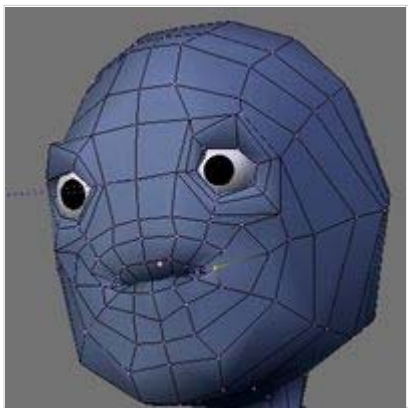
My "Narrow" shape looked something like this:



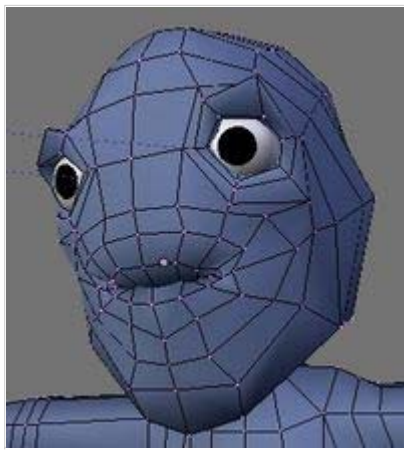
Narrow shape, Side view.



Narrow shape, Front view.

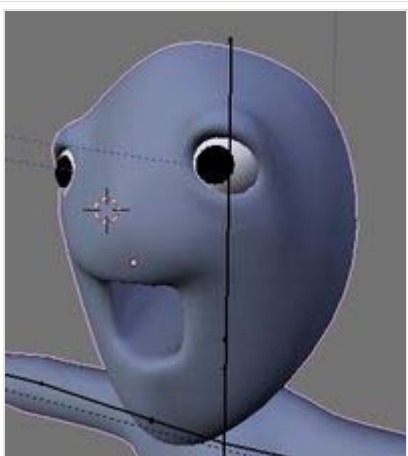


Narrow shape, Oblique (Ortho).

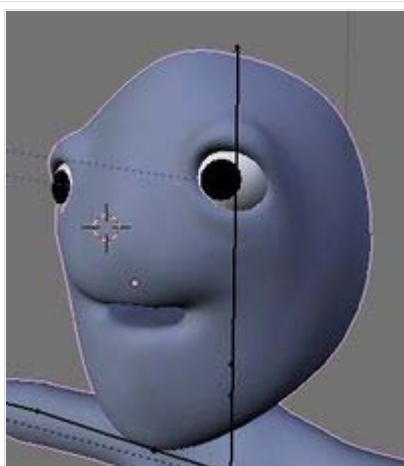


Narrow shape, Oblique (Perspective).

- Check to see that "Narrow" and "Open" work together. We don't need to check "Narrow" and "Wide" together, since they are opposites.



Narrow 100% + Open 100%.



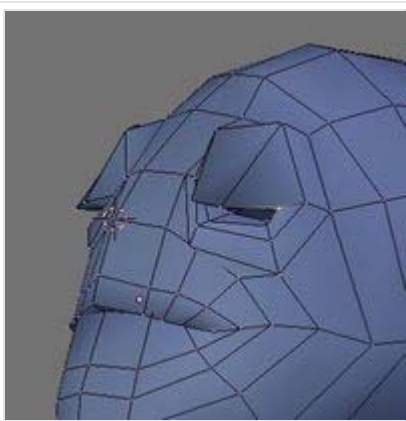
Narrow 100% + Open 25%.

Shapes for the eyelids

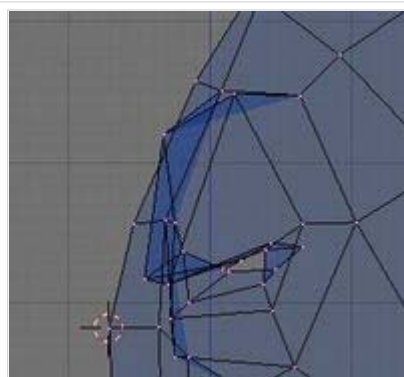
For the next shape, close the upper eyelids ("Upper lid closed"). To do this, I found it was best to turn on Subsurf in Edit mode, so that I knew how far the eyelid was moving.



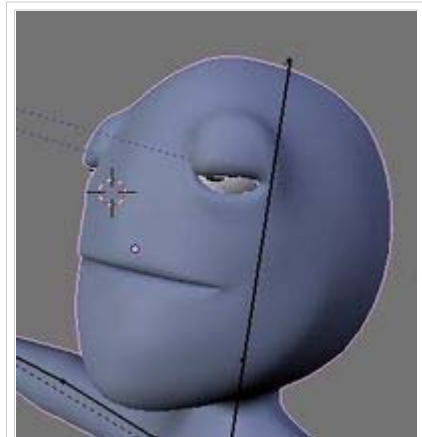
Turn Subsurf on in Edit mode.



Upper lid closed shape, Closeup oblique.

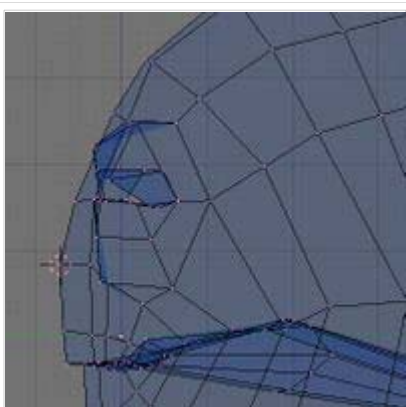


Upper lid closed shape, Closeup Side view.

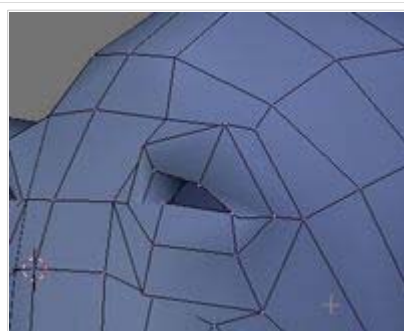


Upper lid closed shape, Ortho view in Object mode.

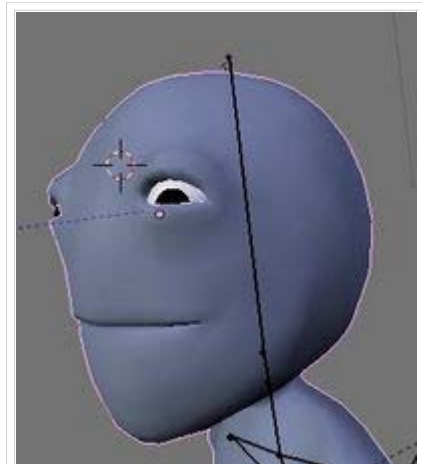
Next, make a "Lower lid closed" shape, something like this. Again, enable Subsurf in Edit mode.



Lower lid closed shape, Closeup oblique.

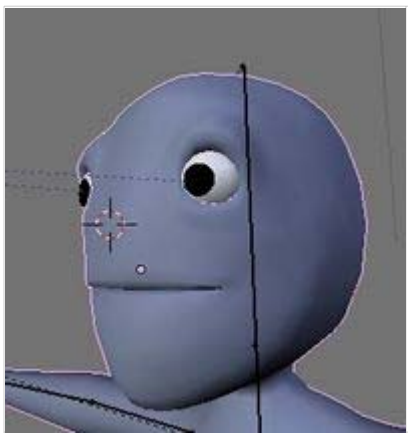


Lower lid closed shape, Closeup Side view.

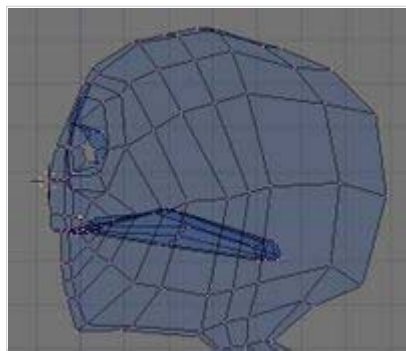


Lower lid closed shape, Perspective view in Object mode.

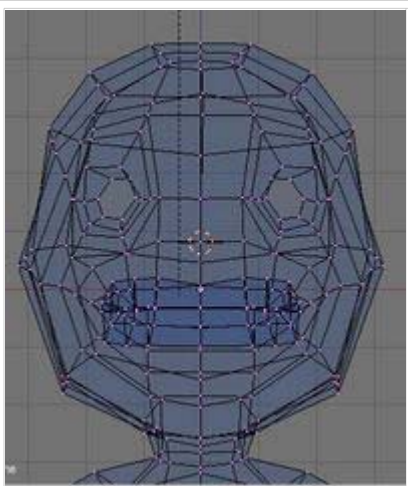
Let's make one more shape for now: a "Brows up" shape. I move the forehead vertices up and outwards.



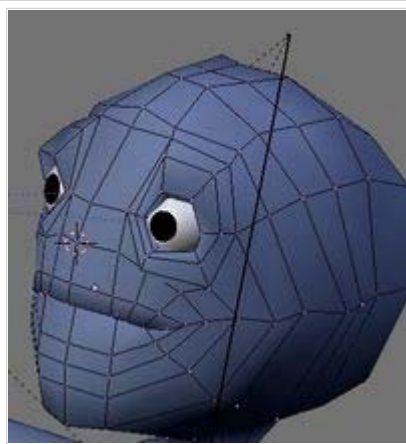
Brows up shape, Closeup oblique.



Brows up shape, Closeup Side view.



Brows up shape, Ortho view in Object mode.



Brows up shape, Ortho view in Object mode.

The shape keys set

Below are the shape keys we have so far. Now that you know how, you can build as many shape keys as you'd like. Good ones to build would be shapes to make "M", "F", and "B" sounds. For the purpose of this tutorial, this set is sufficient for now:



Basis.



Open.



Wide.



Narrow.



Upper lid close.



Lower lid close.



Brow up.

Summary: We created a basic set of facial expressions using Shape Keys. Next, we'll add sound to the animation and lip sync the character to the sound using these new facial expressions

Next: Adding sound and lip sync

Previous: Using the NLA Editor

Back to Index

Retrieved from "http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Shapes"

- This page was last modified 17:52, 19 August 2006.

BSoD/Introduction to Character Animation/Sound and lip sync

From BlenderWiki

< BSoD | Introduction to Character Animation

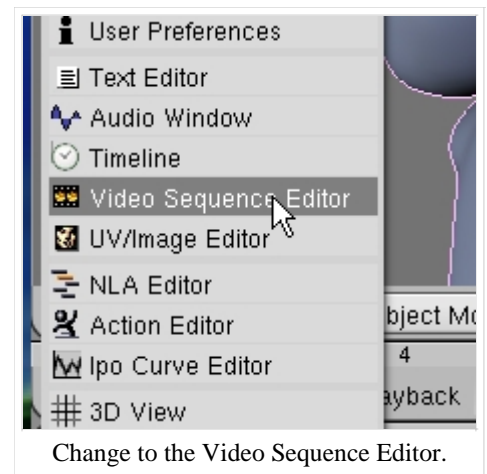
Adding sound

When learning computer programming, tradition holds that the first program you write displays "Hello, world!" on the screen. In keeping with that tradition, here is a sound file of someone saying "Hello, world!". It has been converted to a 16-bit sound file in the free audio software Audacity.

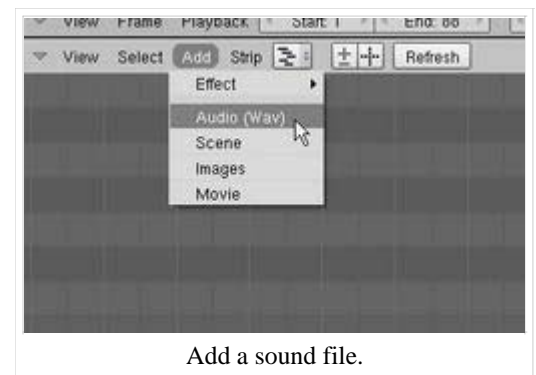
The sound file: The following file is actually a .wav file, I had to upload it as a .blend to get it onto the wiki. Save it to disk and rename as a .wav.

Media>HelloWorld-16bit.blend

- Change one of the windows to the Video Sequence Editor. I changed the NLA Editor into a Sequence Editor.



- In the Sequence Editor, choose *Add>>Audio(Wave)* from the menu.
- Load the HelloWorld-16bit.wav file in the file browser window.



The audio comes in to Blender as a strip, automatically entered into grab mode. Numbers at the beginning and end of the

strip show the frame numbers that it starts and ends. Move the mouse so the sound starts at frame 1. You can trim the ends of the sound strip by right clicking on either end, and you can move the sound strip by right-clicking on the center of the strip.



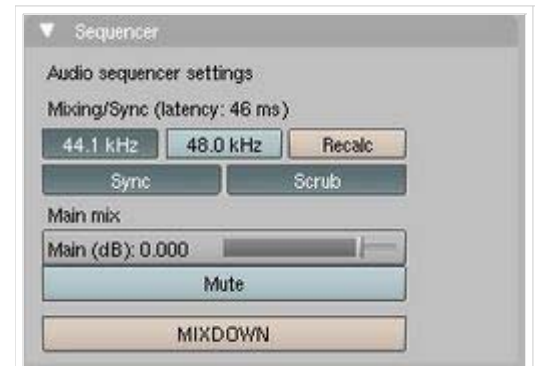
The new sound file. Note numbers indicating the start and end frames (move it to start at frame 1).

- Change the Sequence window into a Buttons window.
- Click the *Scene* context button, and then the *Sound block* button to access the sound settings.



The sound buttons.

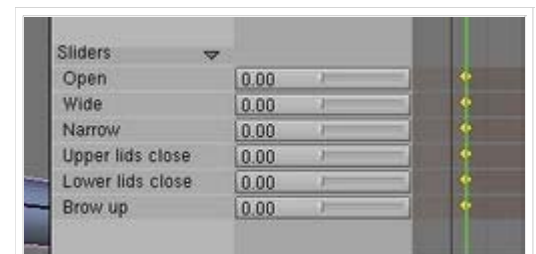
- In the Sequencer panel, select both *Sync* and *Scrub*. *Sync* will make the animation follow the audio, even if it has to not draw some frames to catch up. *Scrub* allows us to drag the frame slider and hear the audio while doing so.
- Move the frame slider back and forth in the Timeline window: you can hear the audio as you do so, no matter how slowly you drag the slider.



Enable both Sync and Scrub.

Here's the plan: we will animate the facial expressions to say "Hello, world!", then convert that into an action. Then we will be free to move the sound file, along with the lip sync action, anywhere we want in the animation.

- First, select the armature.
- Make sure the Rest Position button is still pressed in the Armature panel, to keep the armature from moving while we animate the lip sync.
- Now select the character mesh.
- In frame 1, add a key frame of 0.0 for all shape keys by grabbing the sliders and moving them up and then back again to 0. This gives us the beginning boundary of the lip sync.
- Advance the frame slider to find what frame the sound ends (at about frame 22). Add a 0.0 key for all shape keys here, to set an ending boundary. Now we can easily see that all lip sync animation has to occur between these two sets of keyframes.



To start, set all shape keys to zero.

Now comes the part where you start muttering to yourself and looking in the mirror to see how your mouth moves. Our first pass will be on setting the Open keys. When I say "Hello, world!", my mouth opens a little on

"Hel", it opens more on "lo", it closes all the way on "w", opens a little on "orld", and closes after the "d" is formed.

- Move the frame slider to listen to the audio and find where the "Hel" is. I set the Open shape to 0.39 at frame 2.

Note: Whenever you add a keyframe, it is added at the position of the frame slider, which is always exactly on a frame. However, once the keyframe is added, you can grab it and move it in between frames.

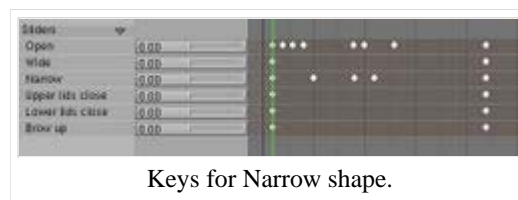
For the Open shape, I set keys as follows:

Frame 1 - 0.00
Frame 2 - 0.38
Frame 3 - 0.53
Frame 4 - 0.63
Frame 9 - 0.19
Frame 10 - 0.00
Frame 13 - 0.30
Frame 22 - 0.00



Next, I added keyframes for the Narrow shape. I had to tone down the "lo" part and make it less narrow so that there was enough emphasis on the "w".

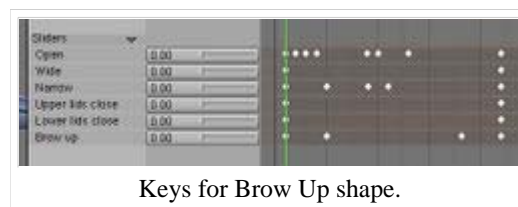
Frame 1 - 0.00
Frame 4 - 0.32
Frame 9 - 0.14
Frame 11 - 0.50
Frame 22 - 0.00



Finally, I added a Brow Up shape:

Frame 1 - 0.00
Frame 5 - 0.62
Frame 18 - 0.00
Frame 22 - 0.00

(next up: change head scratch to a wave, add blinks, work out the eyes, work out the lighting, esp on mouth)



Summary: We added sound to the file and lip synced the facial expressions to that sound.

Next: Final animation

Previous: Creating facial expressions

Back to Index

Retrieved from

"http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Sound_and_lip_sync"

- This page was last modified 10:40, 26 July 2006.

BSoD/Introduction to Character Animation/Final animation

From BlenderWiki

< BSoD | Introduction to Character Animation

Contents

- 1 Setting up for the final animation
- 2 Adding shape keys to the NLA Editor
- 3 Adding blinks
- 4 Package the sound file (optional)
- 5 Rendering the animation
- 6 Directions for adding sound with VirtualDub

Setting up for the final animation

It's time to mix together everything we've done!

Our goal is to have several strips in the NLA Editor that we can move around and adjust. The strips will be:

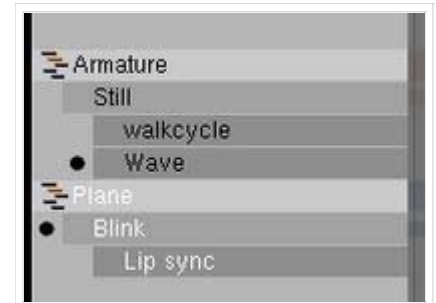
1. walkcycle
2. wave
3. lip sync
4. blink (several separate blinks)

We can check off #1 and #2, we have the walkcycle and wave actions already in the NLA Editor. We have the lip sync animated, but it's not in the NLA Editor yet.

Adding shape keys to the NLA Editor

The NLA Editor was originally designed only to work with armatures. In recent versions of Blender, we now have the option to add shape key actions (like the lip sync action) to the NLA Editor, but it's not as straightforward as with armatures.

Note that down in the NLA Editor, the Plane (the name of the character mesh, since that's what it started as and we never renamed it) has a row. I've kept the name "Plane" to remind us of how far we've come! (If you want to rename it, use the OB: text box in the Link and Materials panel, under the Edit buttons, or, the Object and Links panel under the Object buttons).



The "Plane", or the character mesh, now has a row in the NLA.

- Press **C** to create a strip from the lip sync action.

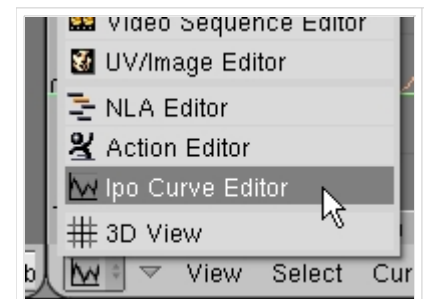
A strip is created, but it's only a placeholder for now. It's a single frame long and it's actually BEFORE frame 1. We have to change some settings to let the NLA Editor know we want this lip sync action to be a strip.

Note that there are no keys in the Lip sync channel right now.



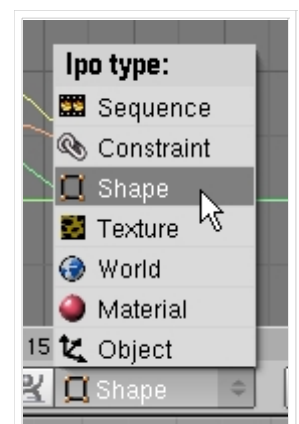
Converted the lip sync action to a strip . . .

- Change a window to the IPO Curve Editor. I changed the Action window into an IPO Curve editor.



Switch to the IPO Curve Editor.

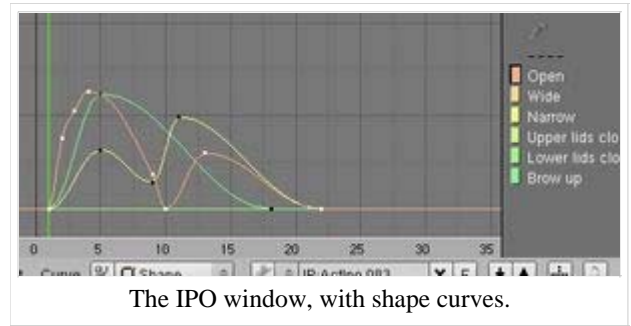
- In the IPO Curve Editor, change the IPO Type to shape.




In the IPO window, change the IPO type to *Shape*

Several curves appear. These curves are another way of visualizing the keys we added for the lip sync. The X-axis of the IPO window is time, or frames. The Y-axis of the IPO window reflects the slider setting, or how strong the shape is activated. Each curve is for a different shape, and they are color-coded. The dots on the curves are the actual keys.

If you compared the IPO window with the Action Editor, you'd find that the frame number of each dot in the IPO curves matches the frame number of each key in the Action Editor. Both views are useful.

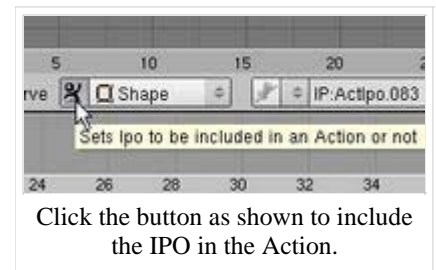


The IPO window, with shape curves.

You can fine-tune the animation through the IPO curves. RMB  to select a curve, and **TAB** to enter the curve's edit mode, where you can adjust the shape of the curves. This is where you have the finest control of the keyframes in the animation, and is often the place to go for really fine-tuning an animation.

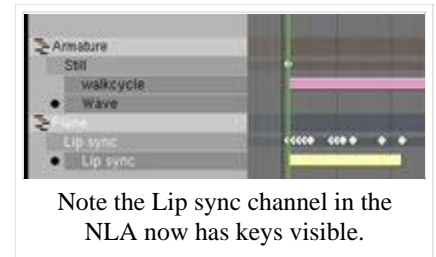
For now, we're going to leave the IPO curves alone, but before we leave the IPO window:

- Click the button next to the IPO Type menu (I'm actually not positive what the tool tip actually means, but I know this is what needs to be done for everything to work).



Click the button as shown to include the IPO in the Action.

Note that there are now keys visible in the NLA Editor for the Lip sync action, and the NLA strip returned to its true length.



Note the Lip sync channel in the NLA now has keys visible.

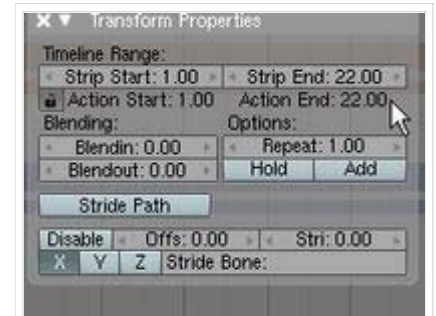
NLA Strips for shape keys: In summary, in order to add shape key actions to the NLA Editor, you have to:

1. Select the Action in the Action Editor
2. Open an IPO Window
3. Switch to the Shape IPO type from the menu
4. Click the button next to the IPO Type menu to allow the IPO curves to be used in the NLA

- **Important:** Go back to the NLA Window and make sure NLA mode is activated for both Armature and Plane.
- **Important:** Select the armature and turn off Rest Position
- Press **Alt A** in the 3D Window to view the animation. The character walks AND talks!

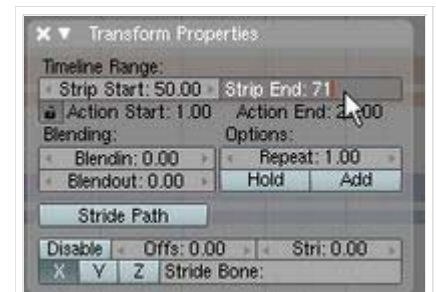
Now what happens if we want to adjust where in the walkcycle he says the words? We can easily slide the Lip sync strip where we want it, but the sound won't follow. The sound strip is not in the NLA, it's in the Sequence Editor. Slightly confusing, and future versions of Blender may change this. For now, we have to do things by the numbers.

- Select the Lip sync strip in the NLA Editor and press **N** to bring up the Strip Properties panel (if it's not already visible).
- Take note of how long the strip is: Action Start is 1, and Action End is 22, so the strip is 21 frames long. Strip Start and Strip End are currently the same, but we're about to change that.



Note that the action is 21 frames long

- We need the frame to start on an integer, because a limitation of the Sequence Editor is that we can only start the sound clip on an integer frame. One way to do this is to move the NLA strip for the Lip sync with **G**, and then round the Strip Start: and Strip End: number boxes down. I moved the strip to 50.42 (and it went to 71.42) then I typed in the Strip Properties the Strip Start: 50 and Strip End: 71.



Type in new strip start and end frames, making sure they are 21 frames apart.

- Switch to the Sequence Editor.
- Select the sound clip with **RMB** and move it with **G**.
- Move the sound clip so it starts on the same frame as you typed in for the Strip Start in the previous step (I moved it to frame 50).
- Play the animation - the sound and lip sync should now occur at frame 50.



Move the sound clip to the same frame you typed in as the Strip Start.

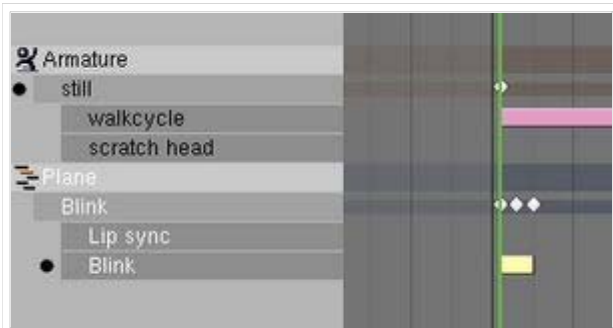
Adding blinks

Now we'll add some eye blinks to add a little realism. First, create a single blink action:

- Select the armature.
- Set the armature to Rest Position in the Armature panel under the Edit buttons.
- Select the character mesh.
- Select **ADD NEW** from the menu at the bottom of the Action Window.
- Rename the action to **Blink**.
- Add a 0.0 key for Upper lids close and Lower lids close at frame 1.
- At frame 3, move the sliders for these two shapes to get a good closed eye shape.
- At frame 5, add 0.0 keys again.

Now convert the "Blink" action to an NLA strip:

- With the action still selected, switch to the IPO window.
- Make sure Shape IPO Type is selected in the IPO window menu.
- Enable IPO curves for this action by pressing the button.
- In the NLA window, press **C** to convert this action to an NLA strip.



A Blink action added to the NLA Editor.

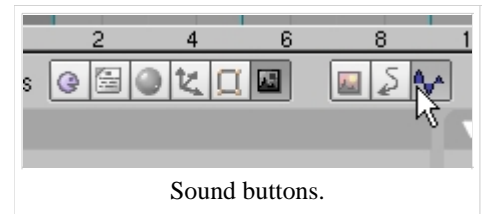
Now, duplicate the blink strips:

- With the Blink NLA strip selected, duplicate it with **Ctrl D**. A second Blink strip appears.
- Duplicate these strips as many times as you want blinks.
- Move the strips along the NLA Editor to where you want the character to blink.
- Enable NLA mode for the Armature, and **Alt A** to view the animation!

Package the sound file (optional)

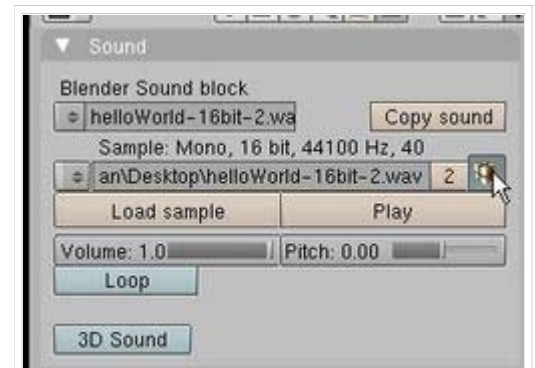
You can pack the sound file in with the .blend file so that you can send the file to another computer that doesn't have the sound file. To do this,

- Go to the Sound buttons



Sound buttons.

- Click the Pack button to package the sound file in the .blend. This will make the file size increase, but is very convenient when you're moving files around.



Click the Pack button to package the sound file in the .blend.

Rendering the animation

Animations with sound: Blender is not yet capable of rendering an animation with sound. What you have to do is save the

rendered animation (video only), then add the audio track in another program such as VirtualDub.

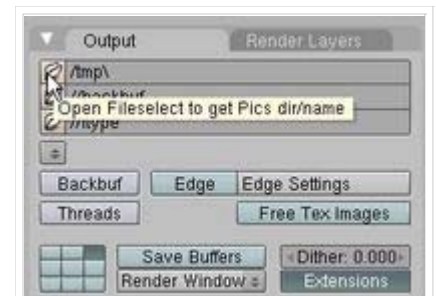
When you're ready to render an animation:

- Choose a camera angle and set it with **Ctrl** **Alt** NumPad **0**. Or, if you're feeling advanced, animate the camera! (Hint: insert keys with **I**. The keys will show up in the IPO window).
- In the Buttons window, select the Render buttons.



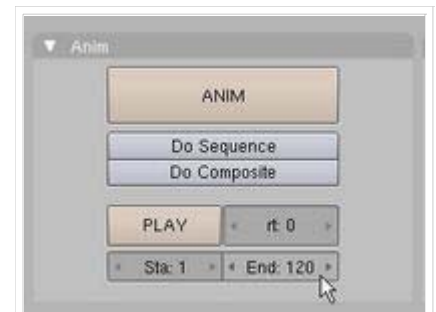
Switch to the Render buttons.

- In the Output panel, click the folder icon in the first row to choose an output directory. Once you've navigated there, click the SELECT OUTPUT PICTURES button in the file browser.



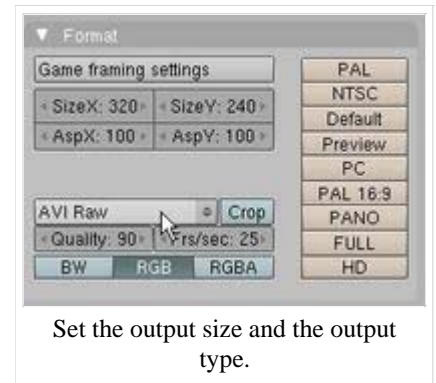
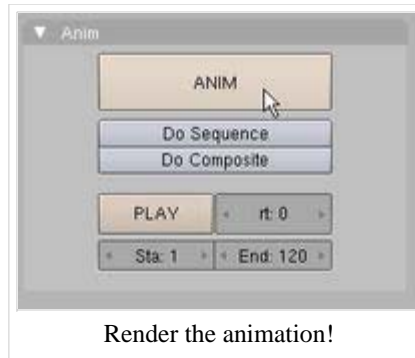
Set the output directory.

- Set the start and end frames to animate, either in the Timeline or in the Anim panel under the render buttons.



Set the starting and ending frames for the animation.

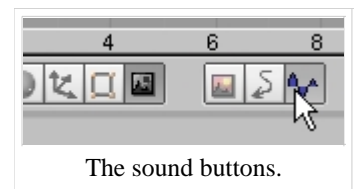
- Set the size of the output in the Format panel. A good size to start is 320x240.
- Set the output type to AVI Raw.
- Press the ANIM button. The animation will be rendered to the output directory you specified as an .avi file.



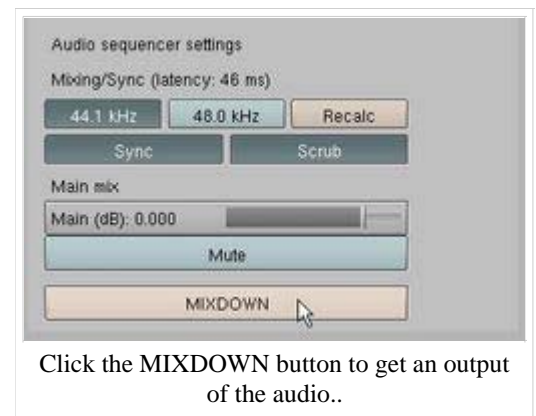
Directions for adding sound with VirtualDub

The .wav file that we animated to was short, and appears somewhere in the middle of the animation. We can have Blender add the appropriate amount of silence to the beginning and end of the sound clip and export it so it's the exact same length as the animation. To do this,

- Go to the Sound buttons



- Click the *MIXDOWN* button. It appears nothing happened . . . until you look in the Output directory (the directory you set in the Render buttons Output panel). There you will find a new .wav file, named after the Start: and End: frames you set in the Timeline window (something like "0001_0120.wav").



Note: Later builds of Blender will include the FFMPEG library, which will have a multiplexing option, allowing you to interleave audio with video in one step. For now, you can use the instructions below.

Here are instructions for adding audio with the excellent free program [VirtualDub (<http://www.virtualdub.org/>)].

- Download and install VirtualDub if you don't already have it.
- Load the *MyAnimationFileName.avi* file you created into VirtualDub
- Go to Audio>>Wav audio and load the .wav file you just exported from Blender.
 - *Optional: Go to Audio>>Interleaving.*
 - In the "Audio skew correction" section, enter a number of milliseconds into the animation you want the sound to start. This will take some playing around with to get right.
- Go to File>>Save as AVI and save the .avi file to disk. The .wav file will be added to the .avi.

Files: Final .blend file, with sound:
Media:BSoD-ItCA-final.blend

Final animation (rename to .avi after downloading):
Media:BSoD-ItCA-animation.blend

Next: Summary

Previous: Sound and lip sync

Back to Index

Retrieved from "http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Final_animation"

- This page was last modified 02:22, 28 July 2006.

BSoD/Introduction to Character Animation/Summary

From BlenderWiki

< BSoD | Introduction to Character Animation

Summary

Congratulations!

Think back to where we started: a simple plane with four vertices. From those original four vertices, we extruded the face and head, added eyes, modeled the body, added lights, and added materials and textures. Then we started rigging by building an armature and then weight painting, first for the upper body and then for the lower body. Next we started animating, first with a simple head scratch and then with a more complex walkcycle. Then we created facial expressions, added sound, and lip synced to a sound file. Finally, we mixed everything together into a final animation.

Now you have the knowledge to create your own character and animate it, completely from scratch. And you have the experience in Blender to be able to branch out and explore other parts of this incredibly deep program. For example, I didn't even talk about UV mapping, Stride bone, animating along curves, the nuances of the IPO window, the node editor for materials and compositing . . . the list goes on!

I hope you enjoyed this tutorial and learned a lot from it. I certainly spent a lot of time on it! You can email me at sendjunktoryan at gmail with any feedback.

Thanks for taking the time to do this tutorial.

-Ryan

Retrieved from "http://mediawiki.blender.org/index.php/BSoD/Introduction_to_Character_Animation/Summary"

- This page was last modified 17:36, 18 July 2006.