Blender 3D: Noob to Pro/Printable Version

From Wikibooks, the open-content textbooks collection
< Blender 3D: Noob to Pro

Contents

Beginner Tutorials

Next Page: Tutorial Syntax

So you've come to learn the Blender, eh? You've made a great choice. This is one of the most powerful 3D animation and 3D creation tools out there, especially if you're short on cash.

Learning how to use Blender can be a daunting task, so **don't give up!** But with the help of this wikibook, you can someday become a power-user and put those Maya folks to shame.

In addition to this wikibook, there are a number of other resources out there. The Blender 2.3 Guide (http://www.blender3d.org/e-shop/product_info.php?products_id=79) is a great book. You can get a lot of help at elysiun.com (http://elysiun.com) . If you have an Internet Relay Chat (IRC) client such as X-Chat (http://xchat.org) , you can connect to **irc.freenode.net** and talk to blender users in the following channels:

#blenderwiki (irc://irc.freenode.net/blenderwiki)
#blender (irc://irc.freenode.net/blender)
#blenderchat (irc://irc.freenode.net/blenderchat)
#blenderqa (irc://irc.freenode.net/blenderqa)
#gameblender (irc://irc.freenode.net/gameblender)

If you have any questions or problems with these tutorials, click the discussion tab on the page with which you're having trouble and explain the difficulty that you're having. Feel free to practice participating on the discussion on this page.

Note on Editing

If you find an error or an "opportunity for improvement," don't just *tell* somebody about it—DO something about it! At the top of each page is a link to "edit this page". Use it! You don't even have to create an account.

Warning

If you do anything damaging or malicious, or add something obviously useless to these pages, it will be quickly undone and your privilege to contribute to wiki projects in the future may be jeopardized. You have been warned.

Quick Installation Guide

Operating Systems (in alphabetical order):

- **FreeBSD:** Download the tarball from http://www.blender.org/ – Official homepage, unpack and run "./blender" in the Blender directory. Or install Blender from ports *../ports/graphics/blender-devel/*.
- **Linux: Arch:** execute this command: pacman -Sy blender
- **Linux: Debian:** execute this command: apt-get install blender (may not contain the latest version)
- **Linux: Fedora Core 3-5:** download the blender rpm from the Fedora Extras (execute: yum install blender)
- **Linux: Gentoo:** execute this command: emerge blender (may not contain the latest version)
- **Linux: SuSE:** Install Blender from YaST Package Manager
- **Linux: Ubuntu:** make sure that you have enabled access to 'universe' and perform: sudo apt-get install blender or open the Add Applications program, go to Graphics and find Blender, check it, then apply.
- **Linux: Mandriva:** Install through urpmi by "urpmi blender" as root.
- **Linux: PCLinuxOS:** Install Blender from Synaptic Software Manager.
- **Linux: Slackware:** Download the latest blender package from [1] (http://www.linuxpackages.net/search_view.php?by=name&name=blender&ver=10.2) and install it with installpkg
- **Linux: other distributions (x86):** Download the tarball (http://planetmirror.com/pub/blender/release/Blender2.37/blender-2.37-linux-glibc2.2.5-i3 , unpack it, then run "./blender" in the Blender directory
- **Linux: other distributions (PPC):** Download the tarball (http://planetmirror.com/pub/blender/release/Blender2.37/blender-2.37-linux-glibc2.3.2-p , unpack it, then run "./blender" in the Blender directory
- **Mac OS X: 10.2:** download the Mac OS X 10.2 installer of Blender 2.37a (http://public.planetmirror.com/pub/blender/release/Blender2.37a/blender-2.37a-OSX-10.

, unzip and run it
- **Mac OS X: 10.3/10.4:** Go to the official download page (http://www.blender.org/cms/Blender.31.0.html) and select from which site you want to download the Blender .dmg.
- **OpenBSD:** Install Blender from ports *../ports/graphics/blender/*.
- **Windows:** Go to the official download page (http://www.blender.org/cms/Blender.31.0.html) and select from which site you want to download the Blender MS-Windows installer. Run the installer once the download is finished. Blender is inherently portable. You can download and extract the zip version of the program to your USB Flash drive and run Blender from the flash drive wherever you go.
- Are we missing your operating system? Click 'edit this page' and add it, or ask about it in the discussion.

Please also refer to the official download page (http://www.blender3d.org/cms/Blender.31.0.html) .

Weblinks

- http://www.blender.org/ – Official homepage

Next Page: Tutorial Syntax

Tutorial Syntax

Next Page: Become Familiar with the Blender Interface
Previous Page: Beginner Tutorials

As you go through these tutorials, you will find yourself running into cryptic codes quite often. These codes refer to keys you need to press on the keyboard and buttons on the mouse you need to press. They are pretty standard throughout the Blender community at this point. You may wish to print this page for quick reference throughout this book.

Keyboard

Special/Function:

ALT	the Alt key on the keyboard
CTRL	the Ctrl (Control) key on the keyboard
CMD	the Command key on the keyboard (Macintosh)
F1 through **F12**	the F1 through F12 keys on the keyboard
SHIFT	the Shift key on the keyboard
SPACE	the Spacebar on the keyboard
TAB	the Tab key on the keyboard
ENTER	the Enter key on the keyboard
ESC	the Escape key on the keyboard

Alpha-numeric:

AKEY through **ZKEY**	the corresponding letter on the keyboard
0KEY through **9KEY**	the corresponding number on the keyboard (above the letters) on the keyboard—**not**

	on the numberpad

Numeric pad:

NUM0 through **NUM9**	the corresponding number on the numberpad—**not** on the keyboard above the letters *('Num Lock' usually needs to be enabled)*
NUM+ and **NUM−**	the corresponding key on the numberpad

- Note that in Blender, there is a big difference between the numbers on the number pad on your keyboard, and the numbers along the top of the keyboard.
- For laptop users: You might have to turn on the "Emulate Numpad" option if you don't have a number pad on your laptop. Drag down the from the bottom edge of the "File," "Add," "Timeline" menu bar, to pull out a new panel. One of the buttons on that panel is "System and OpenGL." Click the "Emulate Numpad" button, to make your regular keys on top of the keyboard behave like ones on a number pad.
- For Macintosh laptops, the F6 key without any modifiers turns on Keypad lock, wherein the right hand alpha keys emulate a numeric keypad. You must be sure to use F6 again in order to restore normal keyboard operation. You might find it a bit more convenient to instead hold the [fn] key on the bottom left of the keyboard to momentarily shift the keys to their number pad function.
- For Windows 2000/XP users, do not press right Shift 5 times in a row as it turns on the Windows Sticky Keys. Doing so will mess up the ability for your keyboard to recognize commands. If the box for sticky keys appears, press cancel (better yet, if you don't need accessibility features, go to Start → Settings → Control Panel; select Accessibility Options, and for each of the options, StickyKeys, FilterKeys, and ToggleKeys, (1) clear the "Use ..." checkbox, and (2) press the "Settings..." button and clear the "Use Shortcut" checkbox).

3-button Mouse

LMB	the left mouse button
RMB	the right mouse button
MMB	the middle mouse button

[*Note: If you don't have a MMB, you can use Alt-LMB to do the same.*]

- Gnome users, it is suggested not to use the "Find Pointer" function in Gnome's mouse settings. If your mouse pointer is being highlighted when you press and release CTRL go to "Mouse" in Gnome's "Desktop Settings" and uncheck the box under "Find Pointer". Otherwise it will impair your ability to use certain functions such as "snap to grid" or using the lasso tool.

Apple 1-button Mouse substitutions

LMB	the mouse button (default)
RMB	Apple (aka Command) key + the mouse button
MMB	Option (Alt) key pressed + the mouse button

[*Note: While Mac OS X natively uses the "Control" key to emulate the **RMB**, recent Blender versions for Mac OS X use the "Command" key for **RMB**, and the "Option" key for **MMB**. This behavior is also noted in the "OSX Tips" file that comes with the Mac version.*]

Path menu

SPACE → *Add* → *Mesh* → *UVsphere*

means :

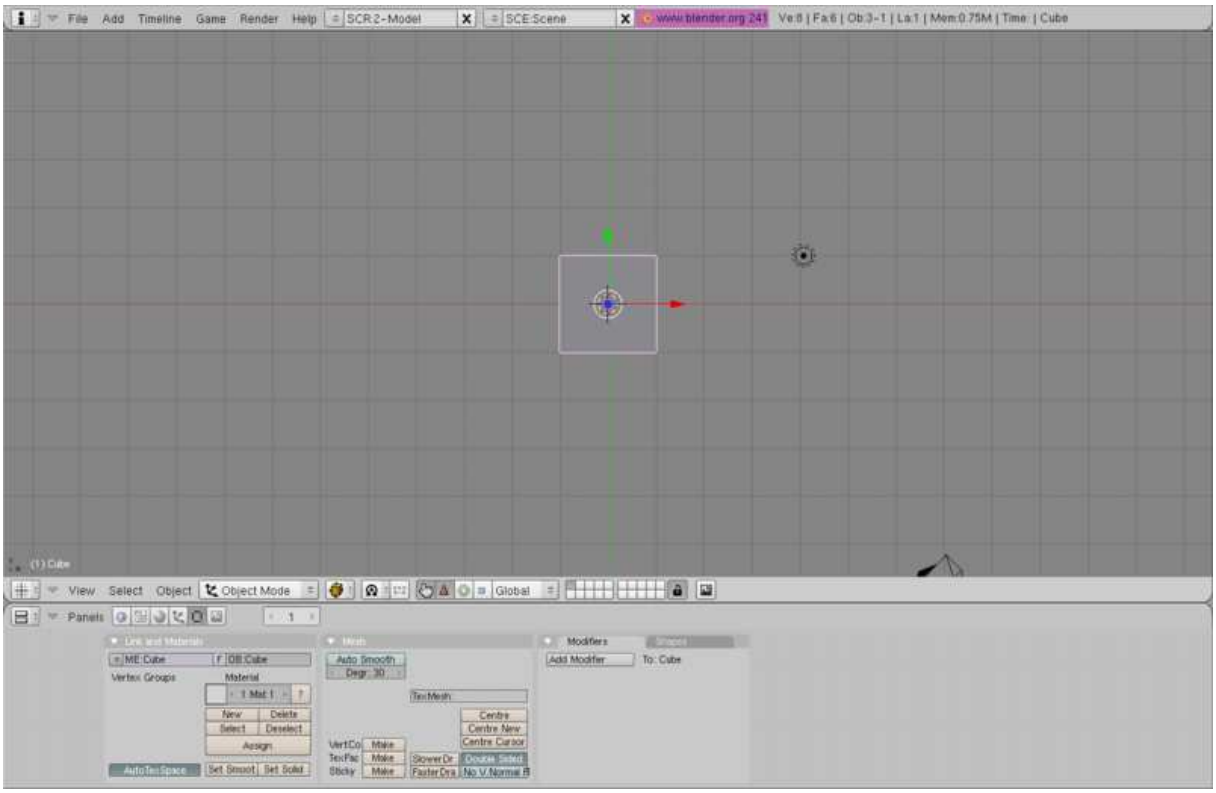
hit **SPACE** and, in the menu that comes up, choose *Add* then *Mesh* then *UVsphere*.

Next Page: Become Familiar with the Blender Interface
Previous Page: Beginner Tutorials

Become Familiar with the Blender Interface

Next Page: Learn the Blender Windowing System
Previous Page: Tutorial Syntax

The Blender Windowing System is a treat. I know, it looks like some sort of space-ship control panel and you have never seen anything like it. Once you learn it, however, you'll wish all your programs worked this way. Move on to the next page to learn more.



Next Page: Learn the Blender Windowing System
Previous Page: Tutorial Syntax

Learn the Blender Windowing System

Next Page: The Buttons Window
Previous Page: Become Familiar with the Blender Interface

Go ahead and open Blender if you haven't already. You'll see that it occupies the entire screen, and may obscure your taskbar. Users of Windows and of certain window managers can press ALT+TAB or ALT+ESC and Apple users may use CMD+TAB to get back to their web browser viewing this guide. Also, in Linux you can start it with the "-w" option so it will be confined to a window.

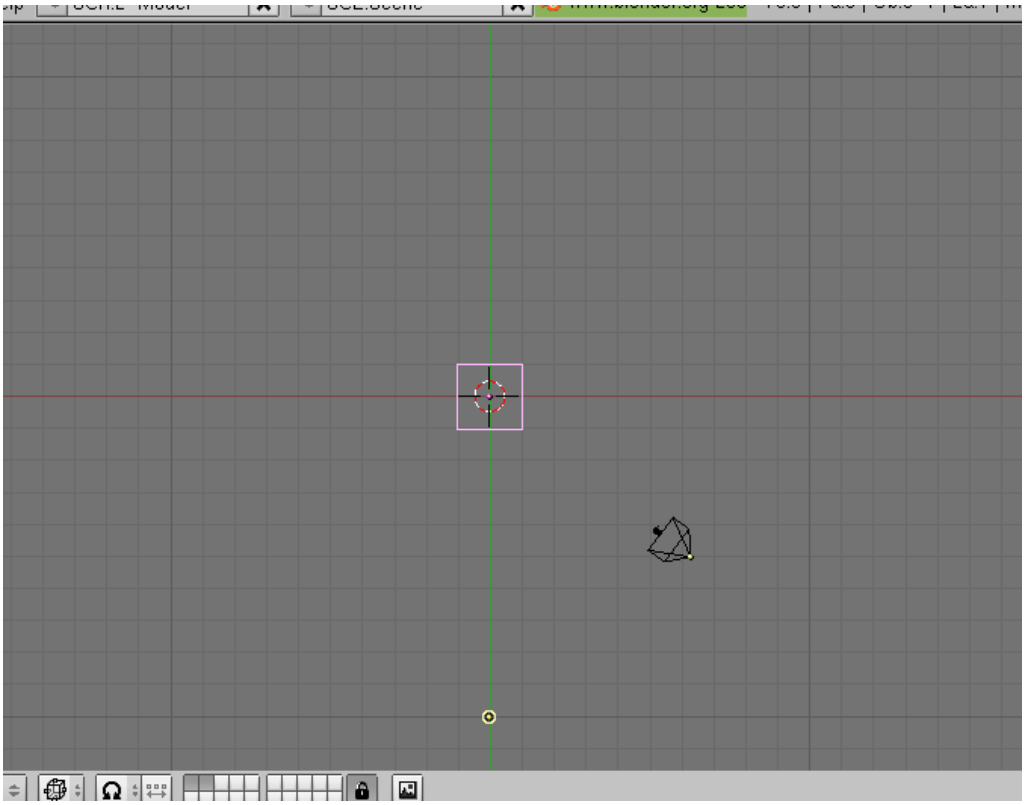
The Blender screen is divided into two sections (there are actually three but we'll get to that later). The largest section (in the middle, with the gridlines) is the "3D viewport". The section

at the bottom, showing a series of controls and buttons, is the "Buttons Window."

Unlike many programs, Blender draws and manages its own user interface **(UI)** elements (buttons, "windows", menubars, listboxes, dialogs, etc.) rather than relying on the "standard" ones provided by the windowing system. In fact, all of Blender's UI elements are rendered in OpenGL, just as the 3D scene objects you create are rendered. Blender's user interface thus appears and behaves a bit differently from the way a "normal" interface does, and can often be confusing to new users. However, you will soon discover that the interface has been designed intentionally to make workflow much faster and more consistent. (For example, you can zoom and pan around a window full of buttons and controls in just the same manner as you can zoom and pan around a 3D scene.) Another advantage to Blender's custom user interface is that it is much more portable, lightweight, and consistent across the various platforms supported by Blender.

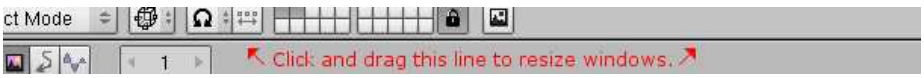
The 3D Viewport

The main division is the 3D Viewport window. It allows you to see and manipulate the 3D objects in your 3D scene. The grid lines represent one Blender Unit (BU). How big is a BU? It can be however large you would like it to be! A BU could be an inch, a centimeter, a mile, or a cubit. A BU lets you decide the scale.



Resizing the Windows

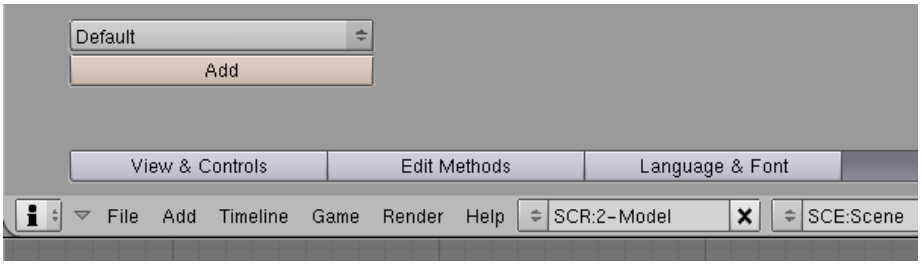
Hold your mouse over the border between the two windows (in this case the 3D view and the buttons window), and the mouse pointer will change to up/down arrows (or a hand on Mac OS X). Click the **LMB** and drag. You can resize the windows by doing this. Resizing the buttons window will cause the text size to be out of proportion with the window size, to fix this, click the **RMB** and select horizontal, which would fix the text size to the window size proportion.



Note, when modifying borders in Blender we are referring to the black subdividing lines within the Blender window; if you are clicking on the outside edge of the window, you are not doing what the tutorial intended. Also note that if you drag the border at the top of the 3D viewport (the black line separating the viewport from the "menubar," you will notice that what looked like a normal menubar is actually the bottom-most portion of another window called the "User Preferences" window.

Pressing CTRL+UP or CTRL+DOWN toggles the maximize state of the current window (in Blender3D the current window is the window that your mouse is over); repeat to restore the window to its original size.

User Preferences



The third division, User Preferences, is mostly hidden, because most of the time you won't need it. It is actually minimized to the point where only its bottom-most portion is visible and looks like a normal "menubar" such as you might expect to find at the top of an application like OpenOffice. To access the hidden portion, move your mouse to the top of the 3D Viewport, to the black border line under the file menu, until you see the mouse pointer change to the familiar up/down arrows. Then simply click **LMB** and drag down while holding. You will see many Blender configuration options hidden there. Each set of configuration options is grouped in an appropriate category, such as "View & Controls, "Edit Methods", or "Auto Save", etc., displayed as a horizontal row of buttons along the bottom. Clicking each of the category buttons displays its associated settings in the area immediately above.

For example, if you click the "Auto Save" category button, the settings associated with the automatic saving of files are displayed. I like to set "Save Versions" to "32," which preserves up to 32 older versions of the file I am working on. When a file is saved, rather than overwriting the previously written version, it is renamed "name.blend1". By preserving multiple versions of the file, I can see how my project has developed and even revert to certain older versions in case of a bad decision.

Unfortunately, in order to set this to default you must:

- 1. Open a new blender window from scratch.
- 2. Drag the menu into view.
- 3. Change any settings you may want to change.
- 4. Re-hide the bar.
- 5. Hit **CTRL + UKEY**.

SAVING YOUR PREFERENCES: It is a good idea to save a backup copy of .b.blend in the Blender home directory in case you ever want to restore your defaults.

Joining and Splitting Windows

Go to the border between the 3D Viewport and the Buttons Window (note that the mouse cursor changes as before). Click the **RMB** or **MMB**. A menu will come up with the options "Join" and "Split Area". Choose Split. A bar will appear in the current window. Move the mouse cursor to position the new border and **LMB** to accept the change, **MMB** to change splitting direction, or **RMB** to cancel. This is especially useful in the 3D viewport. You can split it into any number of smaller viewports. Each one can be configured to show a unique view of your scene.

Now try "Join Area". Note that this has to be done on the newly created dividing line. Select one side (the arrow shows which side is left over) Your partitions will become one window again.

Try splitting at the right (or left) edge of the Blender window—horizontal splits can be useful.

3 Button Mice for Mac OS X Users: While Mac OS X natively uses the "Control" key to emulate the **RMB**, recent Blender versions for Mac OS X use the "Command" key for **RMB**, and the "Option" key for **MMB**. This behavior is also noted in the "OSX Tips" file that comes with the Mac version. *If you're using a two key laptop pad, hold down both buttons at once to imitate the middle button.*

While a standard 3-Button USB mouse will work with OS X, Apple sells a product called Mighty Mouse which provides three buttons, a squeeze, and a built in trackball. Using Apple's System Preferences, the middle button (the trackball) can be mapped to 'Button 3' when depressed. If you are using a much earlier version of OS X, the shareware product SteerMouse will allow you to set the scroll wheel to the Middle Mouse Button (MMB).

Window Headers

Each of the windows we have discussed so far has a "header" that can optionally be displayed at the top or bottom of its window, or hidden altogether. The header area of the 3D Viewport, for example, shows the "View", "Select", and "Object" menus, as well as a variety of buttons and other controls. **RMB** in the header area, and a popup menu will appear, allowing you to change the location where the header is drawn, or whether it is drawn at all. As we have discussed, when Blender first starts up (using its default interface layout), the Blender screen is divided into 3 separate sections/windows: the 3D Viewport, User Preferences, and the Buttons Window. The topmost is the "User Preferences" window with its header (showing the "File" menu, among other controls) at the bottom but the bulk of its area "off screen" above the header. The 3D Viewport, on the other hand, has its header shown at the bottom. The Buttons Window has its header at the top of its area, so that it is adjacent to that of the 3D Viewport.

If you turn off a header and later wish to make it reappear, **RMB** on the edge of the window in question and select "Add Header".

Changing/Selecting Window Types

Each of the screen sections/windows within Blender (including those you introduce by "splitting") may be individually changed to display a desired view of your scene, or a particular set of controls. In this way, you can configure the layout of the interface in the way that is most appropriate and convenient to the task at hand. The "window type" is selectable using the icon in the leftmost corner of the window header. For example, select the icon (**LMB**) in the upper left corner of the Buttons Window (i.e., the leftmost end of Buttons Window header) and note the "Window Type" popup menu which appears. Select "3D View" from the menu and observe how the Buttons Window is replaced by another 3D Viewport. Reselect the Window Type button and change the window type back to "Buttons Window."

Next Page: The Buttons Window
Previous Page: Become Familiar with the Blender Interface

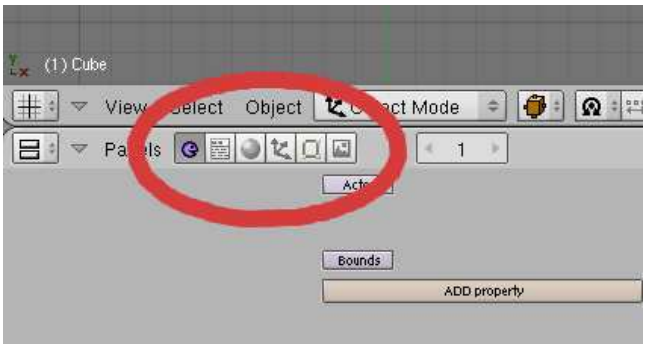
The Buttons Window

Next Page: The 3D Viewport Window
Previous Page: Learn the Blender Windowing System

The Buttons Window is one of the most powerful tools that Blender has. As with all of Blender's windows, it is rendered in OpenGL, and is very scalable and customizable.

When you have objects selected in the 3D viewport, there will be a number of operations you may wish to perform on the objects. For example, suppose you have modeled a person. People have different skin colors, eye colors, hair colors, and more, so you will create a material to make the person appear as you would like it to appear. The buttons window also handles sky color, render settings, animation, and a whole lot more.

There are many groups of buttons available to you in the buttons window. To switch between the buttons groups, select the buttons to the right of the word 'Panels.'



The buttons are (from left to right):

- Logic 
- Script 
- Shading 
- Object 
- Editing 
- Scene 

Click each one of them so that you can get an idea of just how many tools you have at your disposal.

Some of the buttons groups have sub-groups. For example, switch to the Shading  button. You will be able to see several new buttons appear to the right. Left to right, they are:

- Lamp Buttons 
- Material Buttons 
- Texture Buttons 
- Radiosity Buttons 
- World Buttons 

All of the above-mentioned button groups are broken down into smaller groups. For example, click the Editing Button  . You'll see five* smaller windows in the buttons window that you can now manipulate: Link and Materials, Mesh, Mesh Tools, Modifiers/Shapes (two tabs in the same window) and Mesh Tools 1. You can drag these windows into a different order, combine them, and minimize them. Try moving and manipulating these smaller windows for a minute. Also note that there may be more buttons than can fit in the window as it is currently sized. Click the **MMB** and drag (or alternatively, scroll the **MMB**) to pan around the buttons window.

** If you see fewer than four small windows, try setting the 3D Viewport Window to 'Edit Mode'. Press the **TAB** key to toggle between 'Edit Mode' and 'Object Mode'. In 'Edit Mode', there will be at least four smaller windows in the buttons window.*

Next Page: The 3D Viewport Window
Previous Page: Learn the Blender Windowing System

The 3D Viewport Window

Next Page: Other Windows
Previous Page: The Buttons Window

The Blender 3D Viewport Window is where you will be spending most of your time. Blender 3D gives you 100% control of how you're seeing your world. Here are a few things you can do to learn how to use the 3D Viewport.

Rotating the view

Position the mouse pointer over the pink square in the middle of the 3D window.

- Hold down the **MMB** and drag the mouse from side to side and up and down.
- Hold **ALT+LMB** for the same effect
- To rotate so that "upwards" stays "upwards", use **CTRL+ALT+SCROLL**

*NOTE: if you have own setting for your **MMB** in mouse configuration, you must reset this to using the **MMB** as a real Middle Mouse Button (no Doubleclick or something else). Otherwise you must use the alternate **ALT+LMB** for same effects.*

It's a cube! Holding down the MMB is the quickest and easiest way to rotate your view and get a new perspective on things. Right now you're looking at the cube in what's known as Wireframe Mode. Pressing **ZKEY** (yes, on your keyboard, the Z key) will toggle back and forth between Wireframe Mode and Solid Mode. Pressing **NUM5** will toggle between Orthographic and Perspective (perspective looks more natural). This does not affect how your final product will appear, only the way you see your scene while you're creating it.

As you move the view around, you will see that there are four objects in your 3D scene by default:

1) **The Camera**
The camera location and rotation will determine what you will see at render time. To see in your 3D viewport what the camera will see, activate that window by holding the mouse cursor over it and press **NUM0**(remember **0KEY** is different).

2) **A Lamp**
A lamp is simply a light source. It will not be rendered, but the light it provides to the scene will be rendered.

3) **A Cube**
This object will be rendered. The camera should be pointing at the cube so that you will see it at rendertime.

4) **The 3D Cursor**
This is not an object, but a tool for the artist to use to choose where to put new objects in the scene, much like the cursor when you're typing on a word processor (the 'insertion point').

Later you will learn more about how to use each of these.

[NOTE: If you are using a keyboard which doesn't have a numpad, e.g. a laptop, see laptop commands below.]

Numpad

Holding the mouse over your 3D Viewport and pressing the **NUM7**, **NUM1**, and **NUM3** buttons will bring you back to perfectly aligned top, front, and side views respectively. Pressing **CTRL+NUM7**, **CTRL+NUM1**, and **CTRL+NUM3** will result in displaying the bottom, back, and *other* side views, respectively. Try each of these views, and watch the camera and light move around with respect to your new viewpoint (make sure your **NUMLOCK** is on. Otherwise, this will **not** work).

Similarly, holding the mouse over a viewport and pressing **NUM2**, **NUM4**, **NUM6**, and **NUM8** will rotate the view down, left, right, and up respectively.

The object the viewport orbits around can be changed to a new object by first selecting it with the **RMB** and then pressing **NUM.** (the period key on the numpad) or **NUM,** (the comma key on the numpad) on some keyboard layouts.

Again, remember in Blender that there is a big difference between the number keys on your numberpad and the number keys along the top of the keyboard. For example, **NUM7** refers to the number 7 on the numberpad, while **7KEY** refers to the number 7 that's above the **YKEY** and **UKEY** on the standard US keyboard. If you accidentally pressed **1KEY**, **3KEY**, or **7KEY** during this step and it appears that everything disappeared, you can fix this by pressing the **`KEY** (that's a single back-quote key, to the left of the **1KEY** on a US or UK keyboard, usually on the same key as ~, not the single forward quote or apostrophe that is on the same key as the double quote). If you use a notebook laptop try pressing **1KEY** (it worked for me - **`KEY** didn't).

*[NOTE: the **1KEY** through **0KEY** and **alt-1KEY** through **alt-0KEY** switch layers. Hold shift to select more than one layer. **`KEY** selects all 20 layers. Layers will be covered later.]*

For laptop users: the num lock

As previously mentioned in this tutorial, recent laptops (some PC and all recent Mac) have a set of regular keys (from **M** in the lower left to **9** in the upper right) with additional markings corresponding to a regular numpad. This behavior can be toggled with **F6** (or the key labelled *num lock*, this may require pressing FN+numlock key) (**FN+F11** on some Dells). If nothing else works, or as an alternative, you can temporarily activate the numpad behavior by holding the **FN** key (lower left corner of the keyboard) and using the keys as a numpad until you release **FN**. This allows convenient use of the numpad camera controls without interfering with the normal use of that set of keys.

If you envision using your laptop for this kind of work, or indeed any work involving numeric data inputting, it may be worth investing in a USB Numeric Keypad, as Blender uses the numeric keypad quite a bit. Prices range from between $15 to $20 for a basic keypad.

Panning the View

To pan the view, you have your choice of alternatives:

- **SHIFT+MMB**
- **SHIFT+ALT+LMB**

—and move your mouse. Alternatively, if you have a scroll wheel you can use SHIFT+Scroll to pan up and down and CTRL-scroll to pan left and right.

Panning is an important skill to master; try it now.

Note that you must press **SHIFT** *before* **MMB**, otherwise your view will rotate instead of panning.

Also note, that in Windows XP the simultanious pressing of **SHIFT+ALT** is used to switch the keyboard layout (for example QWERTZ becomes AZERTY and vice versa). So when you find your keyboard layout all messed up, press **SHIFT+ALT** again, until it fits.

It's recommended using **SHIFT+MMB** instead.

Zooming the View

Zooming in and out the view is also important. Again, Blender offers you several ways to do what you need to do:

- If your mouse has a scroll wheel, scroll it.
- **CTRL+ALT+LMB** and scroll up and down (not left or right)
- **CTRL+MMB**
- **NUM+** and **NUM-**

Try these all out. Can you see this being useful?

Pro Tip

- If you can, find a mouse with side buttons. Anything like Microsoft's Intellimouse, or Logitech's Mediaplay, that have back/forth buttons, will do. Map those buttons to the MMB. Makes camera control feel a LOT more intuitive (plus it frees up a finger)

Placing the 3D cursor

Click the **LMB** to the right of the cube, half-way between the edge of the window and the cube. The red and white circle (the **3D cursor**) moves to where you clicked. Rotate again and notice that the 3D cursor marks a point in 3D space.

In any given rotational perspective, the set of possible 3D points where you can place the cursor is defined by the plane of your screen. If you're looking at the standard plane straight-on (meaning the standard plane is exactly parallel to your screen), you will place the cursor at the same height above or below the standard plane no matter where you click. Don't worry, you'll understand this point soon enough.

A more interesting experiment is to rotate the standard plane so the left end is farther away from you (and thus farther away from the plane of your screen) than the right. In this view, placing the cursor on the left will put it more toward the front of the plane, and placing it on the right will put it more toward the back.

Try the following exercise: put the 3D cursor inside the camera. Be sure to view the scene from different angles to make sure the cursor is in fact inside. Now put the cursor back inside the cube.

Adding and Deleting Objects

Make sure you are in Object Mode. If not, press **TAB**. The tab key switches between the edit and object modes. A status bar at the top-right of the window will indicate the current mode by displaying 'Ob' or 'Ed' depending on the currently toggled mode.

Click **RMB** (**Cmd+LMB** on Mac) on the cube to be sure it's selected. Press the **XKEY** or **DELKEY** to delete it. A window will prompt you to erase object. Click "Erase Selected."

To add an object, press **SPACE**. A menu comes up which is called the Toolbox. Select Add → Mesh → Monkey. (or just Add → Monkey if you are in Edit mode)

A new object will be added, and you will be in what's known as Edit Mode. Press **TAB** to get out of Edit Mode, then **CKEY** to center the screen on the cursor (where the monkey appeared). Press **ZKEY** a couple of times, which toggles the 3D Viewport between drawing the monkey solid and drawing it wireframe. Zoom in and out for a closer look (scroll the **MMB**, **+KEY**, or **ALT+CTRL+LMB**).

Next Page: Other Windows
Previous Page: The Buttons Window

Other Windows

Next Page: Learn to Model
Previous Page: The 3D Viewport Window

Just when you thought that you were getting the hang of the Buttons window and the 3D Viewport window, there are several more windows to learn about. Have no fear; we will gently guide you through this book and teach you about these windows as the need arises. For now, your only need is to know of them, to be aware of your many options.

In the 3D viewport window, you'll see a button on the header all the way to the left that has a grid on it (if not, click on a window separator with the **RMB** or **MMB** and choose "Add Header"). That button allows you to switch window types. Click on it with the **LMB** and you will see a number of different window types to which you can change. Try some of the different window types; you will learn about their relevance in time.

Change the window back to the 3-Dimensional Viewport before moving on to the next tutorial.

Next Page: Learn to Model
Previous Page: The 3D Viewport Window

Learn to Model

Next Page: Quickie Model
Previous Page: Other Windows

The most basic part of 3D development is modeling, because this is where you create content, or 'models.' Creating 3D models is fun and sometimes challenging.

On the next page, you will take the first step in learning how to model. If you're excited, great! But if you're scared, don't worry; it starts out very easily. Give yourself time and patience; Pixar and Dreamworks will still be in business when you're ready for them!

Next Page: Quickie Model
Previous Page: Other Windows

Beginners Tips

These are some basic tips that are often asked for in one form or another. Sometimes it is in reference to something completely different, but the basic methodology will work.

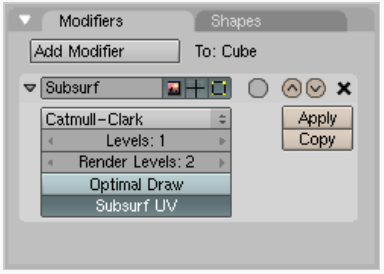
Starting with a box

Starting with the default cube as our example, we are going to try doing some things to it that can help illustrate what might happen down the road to our own models when applying the same effects...

Subdivision Surfaces

Subdivision surfaces, or subsurfing, uses a mathematical process of simulating a curved plane in space according to the placement of control points, or vertices. What this means is that you can create an object with a smooth surface that is easily controlled by relatively few vertices.

In Blender, the Subsurf operation is treated as a *modifier*, which means that the mesh you started with is left intact and editable even after you've Subsurfed it. You will be able to make changes to your pre-Subsurf mesh and see how those changes affect the result of the modifier.



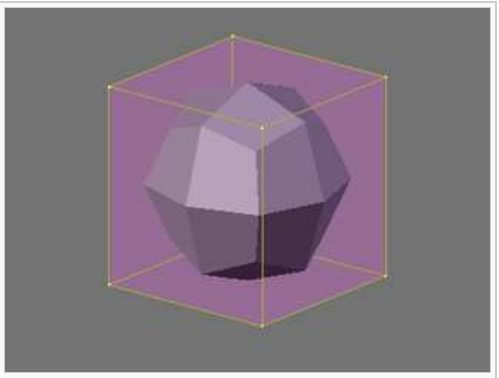
The Subsurf modifier

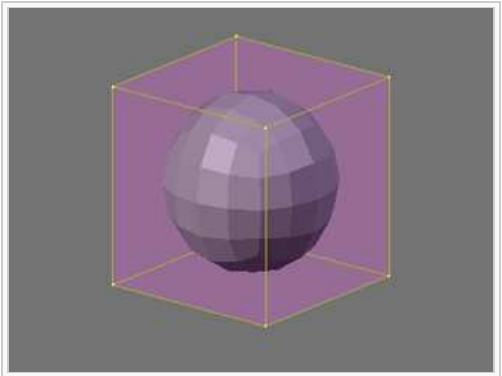
Choose the Editing panel set or press F9, and find the Add Modifier button in the Modifiers panel. Pressing this button will pop up a list of available modifiers, from which you should select *Subsurf*. Edit : if you don't find it, have a look in the Mesh part and click on Subsurf. Screenshot beside is not accurate.

You'll see a Subsurf panel appear inside the Modifiers panel, and the cube in the viewport will take on a rounded look as the modifier's default settings are applied.

Among the options in the Subsurf panel you will find two important options: *Levels* and *Render Levels*. The higher you set *Levels*, the more times the smoothing algorithm will be applied, and the smoother your mesh will look. *Levels* only affects the cube in the 3D view; you must use *Render Levels* to specify the number of levels used when rendering.

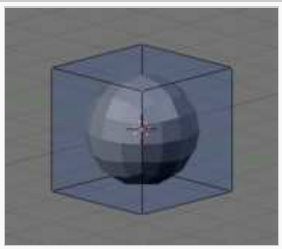
Try increasing *Levels* from 1 to 2 and see how the cube deforms.


Levels set to 1


Levels set to 2

But I want a box!

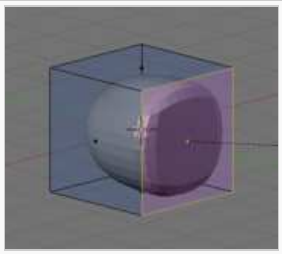
Often, you will want to render with your model having some sort of subsurf turned on. Face it, most things in real life just do not have super sharp edges. It is often the case that objects in the real world will have some sort of softer edge on it (unless it is a knife edge, or a block of material that has been machined in the shop!). It is just this fact that is often overlooked by people starting out in 3D: CG can sometimes look *too perfect*, resulting from impossibly sharp, clean, and well defined edges.


Subsurfed cube in Edit mode

This effect can be fixed by telling Blender that we want our cube to retain more of its original shape. We'll do this using a tool called Edge Creasing. Each edge in a Blender model has a crease value associated with it, which is used to tell the Subsurf modifier how sharp we want that edge to be. By default, all edges have a crease of 0, which is why our cube has lost all its sharp edges.

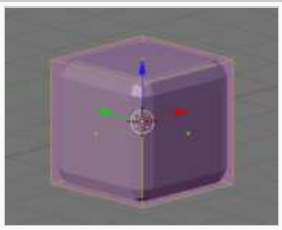
Before we fiddle with the creasing, set the Subsurf Levels up to 3 so you can see the effect more clearly.

Now, remember what we said about the Subsurf modifier remembering our original cube shape? Press '*TAB* to go into edit mode and you'll see that the original cube has come back to haunt us as a wire frame around the smoothed cube.


Editing edge creases

Now we'll experiment with edge creasing. Press **CTRL+TAB** and select 'Faces' to enter Face mode, and select one of the sides of our wire cube with **RMB**. Now press **SHIFT+E** and your mouse will be tied to the cube with a dotted line. Move it gently left and right to see the effect it has on the mesh.

Note that, although we are in Face mode, it is really the *edges* that we are creasing; selecting a face is just a quick way of selecting its four edges.


All edges creased for a bevelled effect

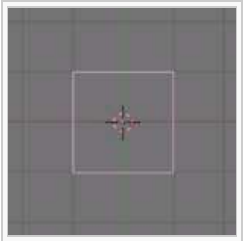
Click **RMB** to cancel out of crease editing mode, then press **AKEY** twice to select all faces. Crease them with **SHIFT+E** like before until your cube looks like the last image on the left. Click **LMB** to apply the changes, then **TAB** to cancel out of edit mode. Behold: your smooth cube.

Quickie Model

Next Page: Quickie Render
Previous Page: Learn to Model

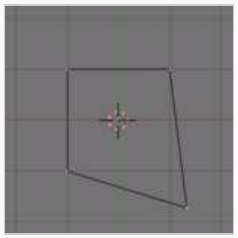
Your first model is easy.

Assuming that you haven't already changed the User Preferences, the default Blender file will come with your first model! If you've been following this tutorial straight through, just create a new model by typing **CTRL+XKEY** or selecting File->New and then confirm that you wish to erase your current model.


A cube in object mode.

You should see a square in the 3D viewport (if you rotate the view, you'll see it is actually a cube). Clicking the **RMB** (**Cmd+LMB** on Mac) over an object selects the object and the outline becomes pink. The cube should already be selected by default. You can use **AKEY** to select or deselect all objects.

Right now you're in what's known as **Object Mode**. In Object Mode you can move the cube around the 3D environment in relation to other objects. Note that with the **RMB** you can also select the lamp or the camera, in which cases you won't be able to go into Edit Mode (Cameras and Lamps are edited differently). With the cube selected, hit **TAB**. This puts you in what's known as **Edit Mode**. In Edit Mode, you can change the shape and size of the cube. You could turn the cube into a puppy… or at least some day you'll be able to.


The cube after editing in edit mode.

TAB *- toggles in and out of Edit Mode of the selected, active object.*

Now that you're in Edit Mode, you have access to the individual vertices. Vertices show up as pink dots when they're not selected, and yellow dots when they are selected. Vertices are control points that you can connect to create edges and faces. Edges connect two vertices, and faces connect three or more vertices. If all the vertices are yellow (selected), press **AKEY** to deselect all vertices (**AKEY** toggles between selecting all or selecting none). Go ahead and hit **RMB** (**Command+LMB** on Mac) over one of the vertices and you should see it change to yellow, which means that it is selected (if all you see is a big blue dot, change the pivot in the pivot menu (two boxes to the left from the one with the hand on it) to "Active Object" so that you now see just one dot instead of the arrow). Also try rotating the view to see what's actually going on.

If you cannot select a vertex, hit the ZKEY and make sure you are in transparent mode. If you then see a circle with some arrows pointing out of it, you've also had the 3D Transform Widgets turned on, and you're not ready for that yet! To turn them off, click the button on the header with a hand on it (in my version it is a closed hand with one finger pointing up and there are two such hands - one on the menu above and one on the menu below (click the hand on the menu below).

With the vertex selected, press **GKEY** and move your mouse around; you should see the selected vertex follow! Remember, **GKEY** lets you grab and move a selection. Choose a new spot for the vertex and hit **LMB**, **ENTER**, or **SPACE** to drop the vertex in that spot. Rotate the view around to see the incredible impact your small change has undoubtedly made.

GKEY *- "Grabs" the current selection and allows you to move it around with the mouse. Use* ***LMB****,* ***ENTER****, or* ***SPACE*** *to drop it in place. Use* ***RMB*** *or* ***ESC*** *to cancel the move.*

Next Page: Quickie Render
Previous Page: Learn to Model

Quickie Render

Next Page: Mesh Modeling
Previous Page: Quickie Model

If you haven't completed the previous tutorial, (the Quickie Model tutorial), do so now. Keep the same file open from that tutorial because we will be using it here.

Now that you've created your first model, undoubtedly you'll want to render it. Rendering is quick and easy. Make sure you're in object mode (press **TAB** if you're not) and simply press **F12**! On Macintosh OS X 10.4 and Gnome you can use **CTRL**-**F12** or **ALT**-**F12** to avoid the Dashboard and the Gnome Search Dialog, respectively.

Your first quick and dirty render should look something like this.

You can also click on '**Render → Render Current Frame**.'

- *A **render** is the creation of a picture from the camera's point of view, taking the environment's effects on your scene into account, and generating a realistic picture based on your settings. This first render will finish very quickly, but you'll find that as your 3D scenes become more complex, the rendering can take a very long time.*
- *If your cube looks black, you may not actually have a light source in the scene. Some versions of Blender don't create a lamp (source of light) by default, and you'll need to add one. To add a lamp, enter object mode (**TAB**) and then press the spacebar while your mouse is over the 3D window. Select **Add → Lamp** which will give you a choice to add several different types of lamps.*
- *You can interrupt the rendering at any time by pressing **ESC** while the rendering window has the focus.*

This is a relatively quick render. It can be cleaned up a bit but it will give you a good idea of what your model currently looks like. Feel free to use the **F12** key as often as you would like.

At some point you will probably want to save your renders. Above the 3D Viewport, select **File → Save image...** or just hit **F3**. A menu with a directory list will appear; the upper text line denotes the directory and in the lower one you type the name of the image, like "myfirstrendering.jpg". Note that earlier versions of Blender (before 2.41?) **will not** add the ".jpg" extension automatically if you leave it out.

Next Page: Mesh Modeling
Previous Page: Quickie Model

Mesh Modeling

Next Page: Modeling a Simple Person
Previous Page: Quickie Render

Mesh modeling is the most common type of modeling in all of Blender-dom. If you did the Quickie Model tutorial, then you've already participated in mesh modeling. A mesh is simply a collection of vertices that define a three dimensional object. This exercise will further help explain mesh modeling.

1. Get a piece of paper and a pencil.
2. Draw three dots that are no more than an inch (about 2.5 cm) apart roughly in the shape of a triangle in the center of the paper.
3. Each one of these dots is called a **vertex**. (The plural of vertex is either "vertices" or "vertexes.")
4. Now connect two of the dots with a line segment. The line segment is called an **edge**.

5. Draw two more edges so that the three vertices are all connected. You should now have a triangle drawn on the paper. Fill the triangle in. This is called a **face**.
6. Now draw another vertex (dot) on the paper. Connect it to two of the vertices (dots) you previously drew. You have another triangle. Fill it in to create another face.

Could you imagine doing this same sort of activity in 3D space? Essentially, mesh modelling is just that. The details are on subsequent pages in this tutorial.

You can keep filling up the paper with more vertices, edges, and faces if you want. You may want to try and create something interesting with your triangles.

Look closely at a 3D video game character some time. Believe it or not, every part of the character is created from little triangles joined together (of course, the triangles are much harder to see in newer games using more detailed technology).

When you're creating your models, remember that the whole point of having edges and vertices is so that you can have control points in 3D space for your faces. When the scene is rendered, only the faces will be seen. Any edges or vertices not connected to a face will not appear.

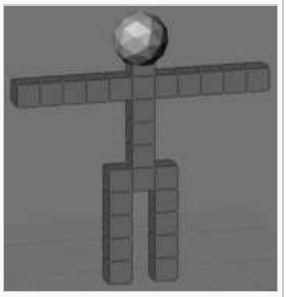
Next Page: Modeling a Simple Person
Previous Page: Quickie Render

Modeling a Simple Person

Next Page: Detailing Your Simple Person 1
Previous Page: Mesh Modeling

Creating a New Project

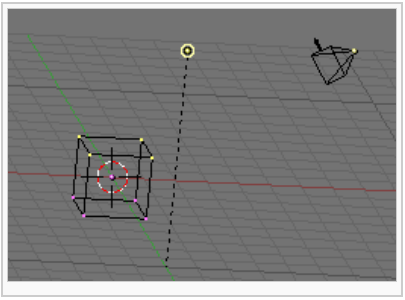
One of the first things you may want to do is to make a person. Oh, what fun!


Your simple person will look like this.

With Blender open, select *File → New*. A confirmation to "Erase All?" will appear. Click on it to accept. You should have your default beginning cube. Select the cube with **RMB** (**CMD + LMB** on Mac). Press **NUM1** to go into front view. Right now you're in Object Mode. The cube should be selected, so you can toggle between Edit Mode and Object mode with **TAB**. Leave it in Edit Mode.

*Reminder - The status bar at the top right of the screen will show '*Ob*' when in Object Mode and '*Ed*' when in Edit Mode.*

Learning about Selection



Once in the front view in Edit Mode, you need to select the top four vertices. The image to the right shows the view rotated a bit with the correct vertices selected. Try all five of these methods of doing this:

Note: Before we begin, you'll need to rotate your view a bit (**MMB** or for Mac Users **ALT + LMB**) so that you can see all of the vertices. Also, make sure the "Limit Selection to Visible" button is selected, the second furthest right icon of the 3D View header in EDIT Mode. This button is not visible in wireframe Mode, so toggle from it with **ZKEY** if you cannot find it.

1. Box Selecting

> [Note: this tool draws a square that you resize to select the top four (vertices\dots) you are not selecting the 3D (cube\box).]

Press the **BKEY** to activate what is known as the Border Select tool. Now, when you click and hold **LMB** and move the mouse cursor, a selection border will appear. When you release the mouse button, the vertices that are inside it will be selected. Select the top four vertices. If you made a mistake, you can start again after hitting **AKEY** to deselect the selected vertices. Make sure all the vertices are deselected (pink, not yellow) before trying the next method.

AKEY *- Toggles between selecting all or selecting none.*

BKEY *- Activates box-select tool.* (Note Pressing it twice will give you circle-select tool.)

2. Circle Selecting

Press the **BKEY** twice to activate the Circle Border Select tool. A circle appears around the mouse cursor; you can resize the circle by scrolling the **MMB**. Another way of resizing the circle is to press the **NUM+** and **NUM-** keys on the numeric pad (useful if you don't have **MMB** or if you're using a Macintosh). You can select vertices by moving the circle over them, holding **LMB**, and dragging the mouse. You can deselect vertices one by one by pressing **ALT + LMB** over the vertices. (This works with Circle Border Select Tool activated. Note that you normally deselect vertices by pressing **SHIFT + RMB**.)

Try those now. The Circle Border Select tool will be active until you press **RMB**, **ESC** or **SPACE**. Press **AKEY** to deselect the vertices before trying the next method.

wmii users: in this window manager **ALT + LMB** moves the current window so, to deselect a vertex use **CTRL + ALT + LMB** instead

3. Lasso Selecting

Like many graphics programs, Blender 3D has a Lasso Select tool. Hold **CTRL + LMB** and attempt to draw a circle around the four vertices you would like to select with the mouse cursor. Release the LMB when you're finished. Press **AKEY** to deselect the vertices before trying the next method.

4. One By One Selecting

You can also select the four vertices one by one. Select the first vertex at the top of the cube with **RMB** (**CMD + LMB** on Mac). To select additional vertices, hold **SHIFT** while pressing the **RMB** (**CMD + LMB** on Mac).

5. Face Selecting


Selection Modes

In addition to those selection methods, there is yet another option -- to the right in your viewport header you can see selection modes. Click on "Face select mode" (Looks like a triangle) and select the top face of the cube with the **RMB** (**CMD + LMB** on Mac). Switch now to "Vertex select mode" (looks like 4 dots in a diamond formation) before proceeding further. As you will see, all four vertices forming the top face are selected (this is also called "selection transformation").

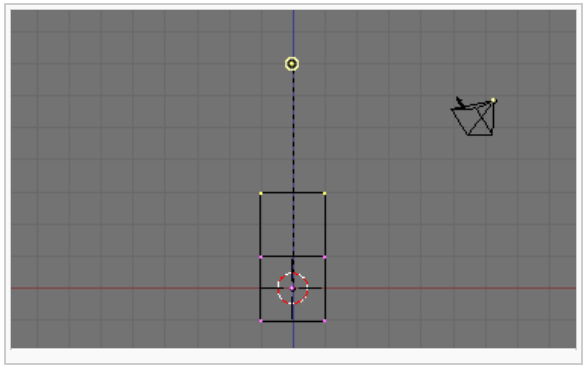
Alternatively, (with the mouse pointer in the 3D Viewer) you may select **CMD + TAB** (**CTRL + TAB** on Windows) and select "Vertices" from the floating drop down menu.

Learning Extrusion

[NOTE: The pictures below are done using the orthographic view. When Blender (2.42) opens, its default view is the perspective view. If you want the orthographic view, press

NUM5]

Restore the front view by pressing the **NUM1** key.



With the correct four vertices selected, hit the **EKEY**. A menu will come up; choose **Region** (if Blender gets into a state where you can only select "Only Edges" and "Only Vertices," you have not selected four vertices that make up a face). Then move the mouse to see four NEW vertices attached to the four that were selected before (you can move them around with the mouse and drop them in place with **LMB**, **SPACE**, or **ENTER**).

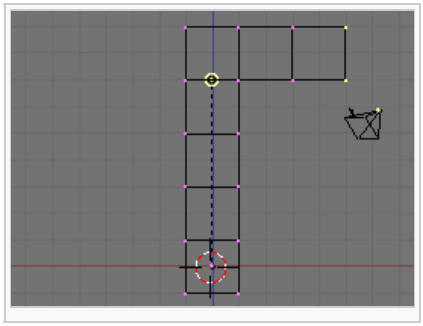
The **EKEY** "extrudes" the vertices. If you've never heard that word before, you'll want to remember it. This is one of the most widely used modeling tools available.

More than likely, wherever you extruded the vertices to is not the right spot for this tutorial. Hit the **UKEY** or **CTRL+ZKEY** to undo your last edit. You should see just your original cube with the top four vertices selected.

With the top four vertices selected, hit the **EKEY** again. Again, choose Region. This time, as you're moving the extruded vertices around, hold down **CTRL** and you'll see that they will only move to certain spots. This is called "snapping". The vertices "snap" to values in predetermined increments. We'll talk more about snapping later, but for now, set the vertices in the right spot so that it looks like two cubes of equal size stacked on top of each other.

The new extruded vertices should be selected still. Hit **EKEY** and choose Region yet again, and use **CTRL** again to control where the vertices are placed. Do it again. Keep doing it until you have five boxes of equal size stacked on top of each other. And that, my friend, is a very simple leg!

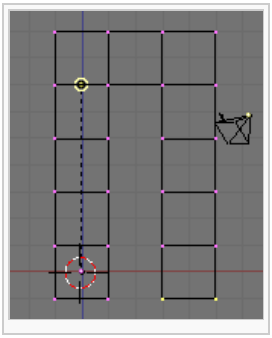
[NOTE: you *must* create all stacked boxes in sequence, or you won't get the nodes you require. Don't just stretch one box all the way.]



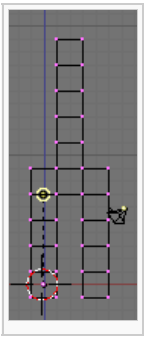
Hit **AKEY** to deselect the current vertices. Select the four vertices in the upper right hand corner of the leg. Again, you'll need to rotate your view a little with the **MMB** (**CMD+LMB** on Mac) to be able to see all the vertices. To select two or more vertices, press **RMB** for the first vertex. Then, press **SHIFT** and hold it. After selecting these two vertices, select the two vertices that are immediately below them. In total you should now have four vertices selected.

Hit **EKEY**, choose **Region**, and you guessed it, hold **CTRL** to set them in the spot so that the box is the same size as all the others. Do it one more time. Now you have the left leg and the connector between the legs.

To create the right leg, first press **AKEY** to deselect all vertices. Select the four vertices that are going to need to extrude downward in order to create the right leg. Extrude them the same way we've been doing. Do it several times until the right leg looks just like the left leg.



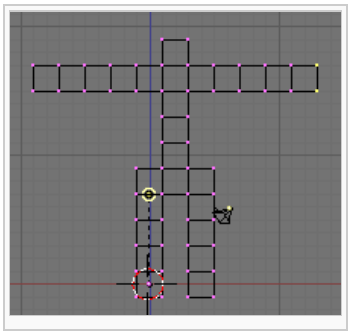
Now on to the torso! Hit **AKEY** to clear your selection, go into front view with **NUM1**, select the four vertices at the top of the box connecting the two legs.



Alternatively, it may be easier to change to "Face select mode" and select the top face of the box with the **RMB**. You will notice that in this case pressing the **EKEY** will automatically select "Region" as your extrusion method since you have selected the face itself rather than just its vertices.

Either way, extrude the connector box up five times the same way we've been doing it.

We're going to make the arms stick straight out to the left and right for now. Go into front view, clear the selection, select the four vertices on the left side of the torso, not the top two, but the four directly below the top two. Extrude out five times.



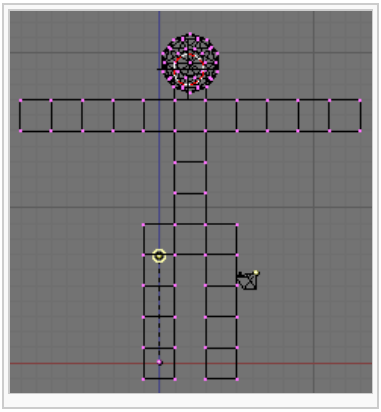
Now make the right arm after the same fashion.

Toggle solid mode on using **ZKEY** (see below), and check that all is well. It's easy to fix the model if some faces are missing. To create a face from four vertices, select the vertices and press **FKEY**. Alternatively, use Mesh menu in the viewport > Make Edge/Face.

Placing Geometry

Now we need to add the head. You may have noticed the crosshair-looking thing floating around in 3D space. It is called the 3D cursor. You place the 3D cursor with the **LMB**. The problem is, it's difficult to get it exactly where you want it. Click the **LMB** at the top of the neck to try to put the 3D cursor there now. Then rotate your view and see where the 3D cursor actually ended up. Is it where you intended it to be? Usually it's not. But sometimes if your camera angle is relatively correct you can come very close to actually getting it in the right spot!

In order to put the 3D cursor in the right spot, you'll need to put it in the right spot in two different views. Go to front view with **NUM1**. Use the **LMB** to put the 3D cursor at the top of the neck. Then hit **SHIFT+SKEY** for the Snap-menu and select **Cursor → Grid**. The cursor will snap to the grid. This is an invaluable tool when working with the 3D cursor. Next, go to side view with **NUM3**. Use the **LMB** to again put the 3D cursor at the top of the neck and snap it to grid again. Now rotate your view around and see that the 3D cursor is neatly at the top of the neck. Good job!



You're going to drop a head in at the top of that neck. With the mouse over the 3D Viewport, hit **SPACE**. In the menu that comes up, choose **Add →Icosphere**. Choose two subdivisions. You will now see a sphere of vertices added to the model at the spot where you placed the 3D cursor.

- When you're in Edit Mode, adding something will add vertices and faces to the object you're editing. When you're in Object Mode, adding something will create a new object with separate vertices and faces. (If you press **SPACE** in either mode, both eventually allow you to add icospheres)

You might feel that the head is a little low. Go to front view with **NUM1** and hit **GKEY** (**G**rab) and then press the **ZKEY** to move the head up off the shoulders a little bit (pressing the **ZKEY** before you drag toggles on Z-axis locking---so that your head moves only on the Z-axis). After selecting the new location for the icosphere (head), use the **LMB** to drop it. You can also change the size of the head with the **SKEY** (S for "scale").

- Note: recognize you may have built this man in a different plane (i.e. the y-axis may be your up/down axis). If so, substitute the **YKEY/XKEY** for the **ZKEY** after pressing the **GKEY** in the preceding paragraph.

Hit the **ZKEY** and the faces will appear with a grey material, letting you see more or less what your guy is going to look like! Using the **ZKEY** will toggle between different drawing modes.

Okay, so it's not all that great yet. Let's start fixin' it up now! (see the links to the next section, at the bottom of this page)

Summary: Keys & Commands

These are the keys and commands used on this page:

Key	Mode	Description
RMB or **CMD+LMB**???	Object	Select an object
NUM1		Go to front view
TAB		Toggle between Edit Mode and Object Mode
BKEY then **LMB** and drag		Box selection
AKEY		Toggle between Select All and Select None
BKEY BKEY (pressed twice) then **LMB** and drag		Circle selection
CTRL+LMB and drag		Lasso selection
RMB then **SHIFT+RMB**		One-by-one selection
(click the vertex/edge/face selection buttons)		Change the selection mode
CMD+TAB (**CTRL+TAB** in Windows)		Change the selection mode
select vertices then **EKEY**		Extrude
CTRL	while extruding	Enable snapping
MMB or **CMD+LMB**???		Rotate the 3D view
ZKEY		Toggle wireframe/solid view
FKEY		Make Edge/Face from selected vertices
NUM3		Side view
SHIFT+SKEY		Snap cursor or selection to the grid

GKEY		Grab the current selection and move it
ZKEY (or **XKEY** or **YKEY**)	grab mode (**GKEY**)	Constrain motion to the Z (or X or Y) axis
SKEY		Change the scale (size) of selection

"Hell, no! This many hotkeys?", you say? Yes! And it didn't even hurt, did it?

Next Page: Detailing Your Simple Person 1
Previous Page: Mesh Modeling

Detailing Your Simple Person I

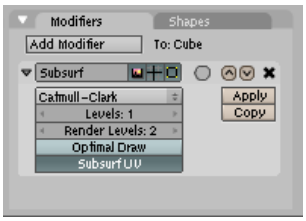
Next Page: Detailing Your Simple Person 2
Previous Page: Modeling a Simple Person

This tutorial uses the simple person model from the previous tutorial. If you didn't do it, go back and do it now---or find it pre-made just for you here (http://www.nusoy.com/blender) .

If your model does not appear to be solid, it is currently being drawn in wireframe mode. For this tutorial, you need it to be drawn solid. Press the **ZKEY** to see the model in solid mode.

Subsurfaces

You should already have the "Editing Panel" displayed in the "Buttons Window". If not, click on the "Editing" button (shown pressed in the image on the right) or press **F9** to have the Editing Panel displayed. On Macintosh OS X, use **CMD+F9** to avoid engaging the Exposé window effects. Note that the "Editing panel" is a different thing from "edit mode"; don't confuse them. Depending on whether you're in "edit mode" or "object mode" the "Editing Panel" will display different tabs. With the object (your man) selected (**RMB**) press **TAB** to view how the available buttons in the panel change.



First of all select the model. We're going to turn on subsurfaces, or Subsurf.

To enable Subsurf, you must go to the "Buttons Window" → "Editing Panel"(F9) → "Modifiers" subpanel → click "Add Modifier" → "Subsurf" from the list. You should immediately see your model change to look more round, less edgy. New options for "Subsurf" are now shown in the Modifiers subpanel. You may also perform this action by pressing **SHIFT+OKEY** while in object mode.

Note that the "Modifiers" subpanel will be displayed in both "edit mode" and "object mode".

To enter a value on a bar you can:

- click on the left or right arrows on either side of the bar to add or subtract a unit.
- click in the middle of the bar and enter a value with the keyboard.
- Hold down **LMB** and

What just happened? Each face was just divided into four smaller faces that are progressively angled, which has helped soften the sharp edges of the model where faces touch each other. Click the horizontal bar labeled "Levels" and change the value to '2'.

The model will change again because each of your original faces is now divided into 16. If you change the value to '3' each plane will be divided to sixty-four smaller planes, but don't do it unless you've got a computer that you're sure can handle it (newer computers should be able to handle it pretty easily). Note that subdivisions work with base 4, i.e., Level: 1 yields $4^1 = 4$ divisions; Level: 2 yields $4^2 = 16$ divisions; Level: n yields 4^n

move your mouse to the left or right while hovering the mouse cursor over the number.

divisions.

Notice the other bar labeled "Render Levels" below the "Levels" bar? That controls how many subdivisions to do at rendertime, while the value we've been changing handles the number of subdivisions while working in the Blender. Before moving on, set the first subdivisions value to 2 and the rendertime subdivisions to 3.

A Modifier is defined as the application of a "process or algorithm" upon Objects. They can be applied interactively and non-destructively in just about any order the user chooses. This kind of functionality is often referred to as a "modifier stack" and is found in several other 3D applications. The x in the upper right of the subsurf modifier will remove the modifier from the modifier stack. (The subsurf modifier can't be undone with the typical undo command.) The arrows at the left of the x will move the modifier (and its effects) higher or lower in the modifier stack.

The Optimal Draw button removes the extra wireframe lines which display as a result of having additional geometry. This button is especially useful to clarify and speed up the display of densely subdivided meshes. The blank roundish button towards the top of the Modifiers panel, just to the left of the up and down arrows, applies the modifier to the editing cage. Press this button now to remove the translucent, boxy cage, so you can edit the smooth mesh in the next parts of the tutorial.

Troubleshooting: If one or two of your sides don't subsurf, try selecting all vertices while in edit mode and typing **WKEY** to display the "specials" menu, select *Remove Doubles*.

Unless you have a good reason, don't press Apply on a Subsurf modifier. If you do, the modifier will be applied to the mesh. While this is useful for some modifiers, for Subsurf this will add many extra vertices and is generally not needed.

For a complete modifiers documentation go to
http://mediawiki.blender.org/index.php/Manual/PartII/Modifiers

For a complete subsurf modifier documentation go to
http://mediawiki.blender.org/index.php/Manual/PartII/Modelling/Modifier/SubSurf

For a complete subsurfaces documentation go to
http://mediawiki.blender.org/index.php/Manual/PartII/Subsurfaces

Smooth Surfaces

Subsurfaces do a good job of smoothing out objects and creating good curved surfaces. However, even with subsurfaces the model does not appear completely smooth; at this point it even appears scaly.


Your simple person after setting smooth.

In Edit Mode, hit the **AKEY** once or twice so that all the vertices are selected (if you're not in Edit Mode, remember to select the simple person and press **TAB**). Find the button that says "Set Smooth" (the center-right button inside the "Links and Materials" subpanel in the Editing [**F9**] panel) and click it. You will see the Blender smooth out the rough edges where faces were touching before. Next to it is the button labeled "Set Solid." Click it as well. You will see the simple person go back to the solid rendering. The simple person looks better smooth, so click the "Set Smooth" button again. (more information about this at [2] (http://en.wikipedia.org/wiki/Flat_shading) and [3] (http://en.wikipedia.org/wiki/Gouraud_shading)).

You need to keep this file open for the next several tutorials. Move on to the next page.

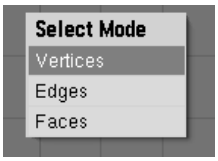
Next Page: Detailing Your Simple Person 2
Previous Page: Modeling a Simple Person

Detailing Your Simple Person II

Next Page: Creating a Simple Hat
Previous Page: Detailing Your Simple Person 1

This tutorial uses the simple person model from a previous tutorial. If you didn't do it, go back and do it now---or find it pre-made just for you here (http://www.nusoy.com/blender) .

Selection modes



Up to this point, you've been selecting vertices and manipulating them. In the first chapter, we touched on selecting faces. In fact there are three selection modes: vertices, edges and faces.

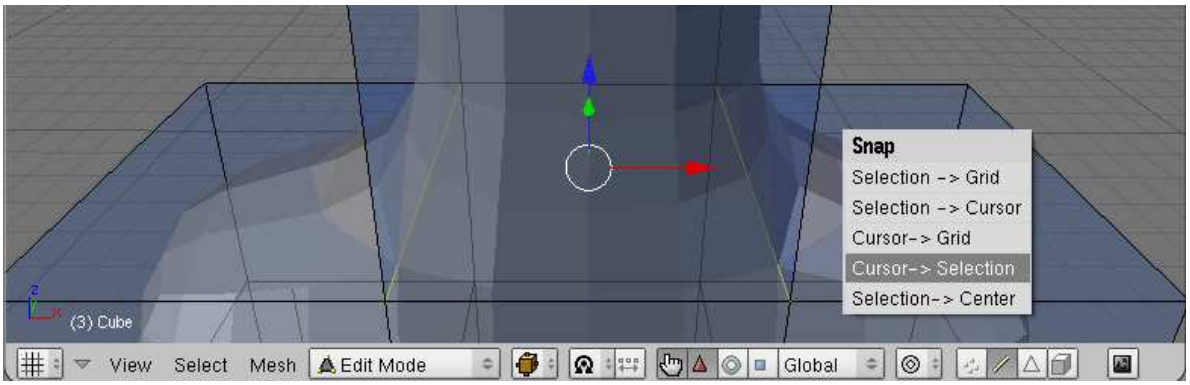
Make sure you're in Edit Mode (**TAB**) and in Solid Mode (**ZKEY**). Press **CTRL+TAB**. A menu will come up where you can choose Vertices, Edges, or Faces. Choose **Edges**. Under KDE, **CTRL+TAB** changes the desktop so you will have to use the statusline buttons instead. (Older versions of Blender do not have this feature. Instead, just select all vertices connected to the edge you want to select).

The three selection modes can also be selected with the statusline buttons shown to the right.

- It is important to remember that whether you're in vertices, edge, or face selection mode, moving or otherwise manipulating your selection will cause connected vertices, edges, and faces to be moved as well. This is because you cannot separate faces from edges or edges from vertices.

Scaling with axis constraint

We want to position the 3D cursor between the hips of the simple person, then use that cursor for scaling.



Use the 3D View's menu *Select → Deselect All* (**AKEY**) to make sure everything is deselected. Reset to *View->Front* (**NUM1**) and choose *View→Perspective* (**NUM5**). Adjust the Edit Mode settings as shown in the picture above.

Troubleshooting - If you do not see the cubes around your person, in the Modifiers tab try clicking the button just to the left of the "move modifier up in stack" button (^) for the Subsurf modifier. Otherwise, try deleting the subsurf modifier (the X right above apply in the Modifiers box) and redoing it (Add Modifier -> Subsurf).

By default, when editing in solid mode, the vertices, edges and faces that are on the back

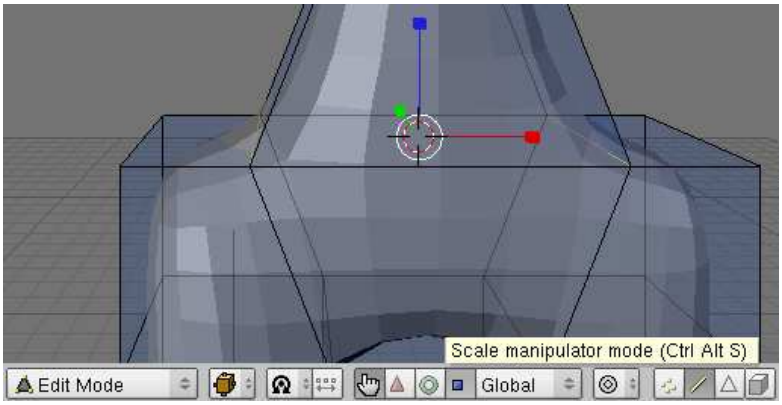
side of the model are not visible or selectable. This can be toggled by clicking the "Limit selection to visible" button located on the header of the 3D View window to the right of the Vertex, Edge and Face selection mode buttons (while in Edit and Solid mode).

We are now going to place the 3D cursor in the center of the pelvis. To do this we will place the 3D transform manipulator in the desired location, and then snap the cursor to that point. The 3D transform manipulator looks like a white circle with three colored (red, green and blue) arrows pointing in orthogonal directions. If you do not see the 3D transform manipulator, find the button in your 3D View's header that looks like a hand and click it.

Ensure that you've disabled "Limit selection to visible". Now, select one of the edges just above the "hips" of our person, where the legs connect to the torso. This would be an edge of one of the cubes to the left or right of the model's pelvis. Notice that the 3D transform manipulator jumps to the edge you selected. Now also select (**SHIFT+RMB**) the edge on the other side of the pelvis. The 3D transform manipulator should jump halfway between the two edges.

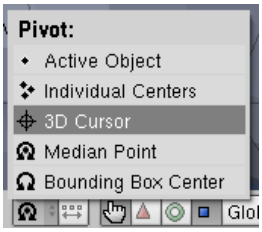
Once the 3D transform manipulator is in the center between those two vertices, bring up the *Transform → Snap* menu (**SHIFT + SKEY**) and select *Cursor → Selection* (**SKEY4**). This will move the 3D cursor to the location of the manipulator.

Choose *Scale Manipulator Mode* (**CTRL + ALT + SKEY**). Since the *Transform Orientation* is set to global, the manipulator's orientation is the same as the world's orientation shown in the lower left corner of the 3D View pane. The axes are colored RGB for XYZ, i.e. the X-axis is red, the Y-axis is green, the Z-axis is blue.

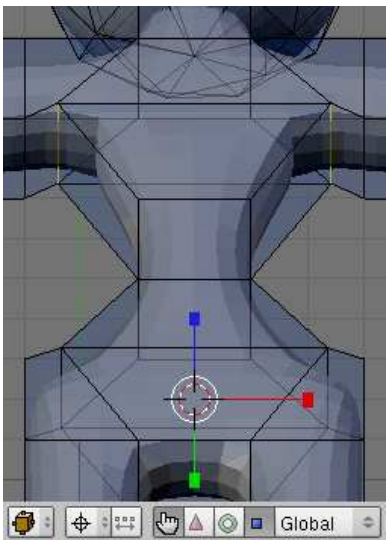


It's important to note that not only is there a global XYZ axis, but each of your individual objects has its own XYZ axes as well. We'll get into that in the next section.

Grab the cube-formed red handle and drag it with **LMB** to symetrically widen up the selection along the selected X-axis. While scaling, press **CTRL** to snap to the grid or **ESC** to abort the current manipulation. When it comes to scaling in the Blender, 1.000 means 100%, .6000 means 60%, and so on. Scale up to 2. Note, you cannot scale along the Z-axis, as the current selection's Z-dimension is zero — if you want to symmetrically lift the hips, switch back to Translate Manipulator Mode (**CTRL + ALT + GKEY**).

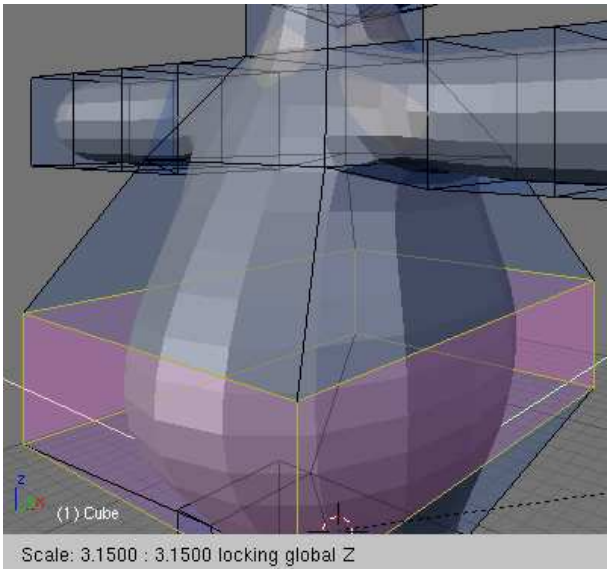


So far, you didn´t use the prepared 3D cursor's position, but the selection's center. Now, set the 3D Cursor as *Rotation/Scaling Pivot*. Since the 3D Cursor was positioned to the selection's center, the manipulators' behaviour stays the same. Finish forming the hips.



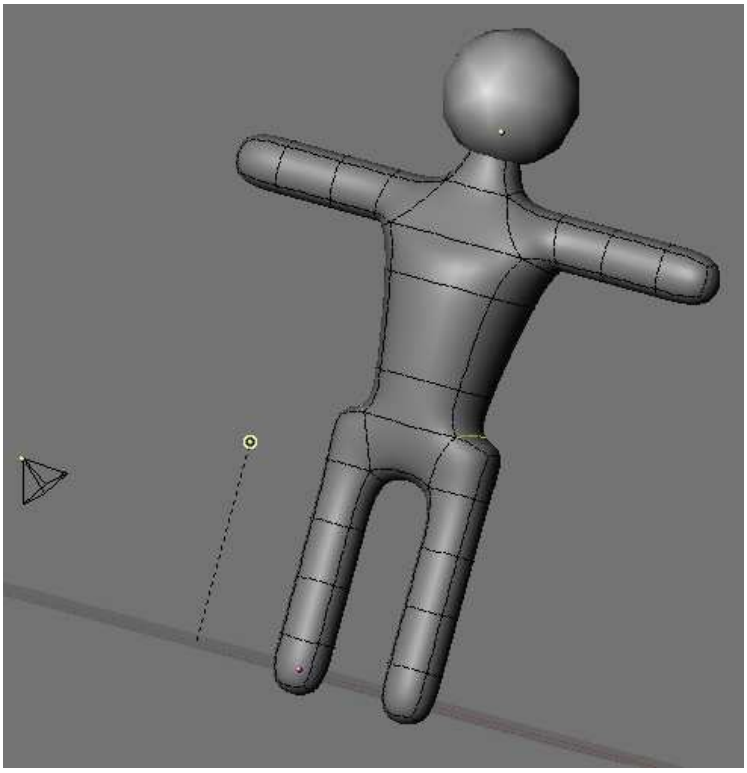
Select both the edges on the underside of each of the arms where they connect to the torso (the armpits, the edges run from front to back). This is easier being accomplished, after you rotated the view about the world's X-axis (*View Navigation → Orbit Down (**NUM2**)). See the manipulator not jump to the selection, but stay at the 3D cursor. Form the armpits. For better visual comparison to the width of the hips, switch to* View → Orthographic *(**NUM5**) before scaling*

along the red X-axis. Note, you can now scale along the blue Z-axis, as there is a distance along Z between the selection and the pivot.



Select the belly cube (if you use *View → Border* or **CTRL + LMB** lasso selection make sure to deselect the arm pits). This time, don't use the Scale Manipulator, but the Scale Tool. Press **SKEY** to scale, and then once **SHIFT + ZKEY** to lock the Z-axis. Now, the scale tool is constrained to the X- and Y-axis, the selection is not scaled along the Z-axis any longer, thus, scales equally along X and Y. The axes that the scale tool is constrained to are drawn through the pivot in brightened color. **SHIFT + XKEY** or **SHIFT + YKEY** work accordingly. Finish scaling by **LMB**.

Continue selecting different sections of the torso and scaling it to your liking. Here is an example of a man-figure:

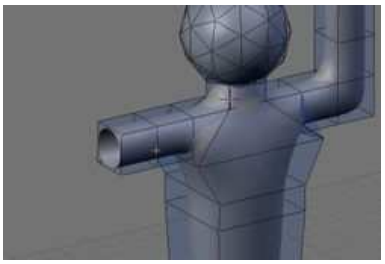


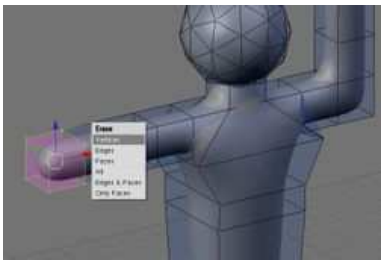
Modeling the arms

Screenshots and more specific instructions needed for this section.

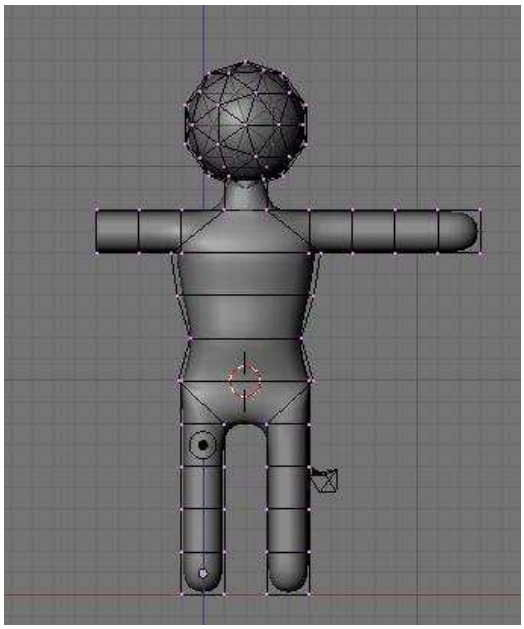
When you've got the basic shape of the torso, move on to the arms. Note that just like you can constrain scaling to the X, Y, or Z axis by pressing **XKEY**, **YKEY**, or **ZKEY**, you can constrain movement to an axis as well. Press the **GKEY** and then press the appropriate axis key. As you work on the arms, be sure to use the different viewing angles so everything is correct (**MMB** to rotate, **NUM1** for front view, **NUM3** for side view, **NUM7** for top view). Also, be sure to use **CTRL+ZKEY** to undo if you mess something up.

We'll start by making him put his hands up. First, make sure you're in Edit mode; if not, select the figure and press **TAB**. Also, make sure you are in Vertices Select Mode (**CTRL+TAB**). Before you begin, also uncheck the roundish button to make the arms into cubes again. You can change them back after you are done.

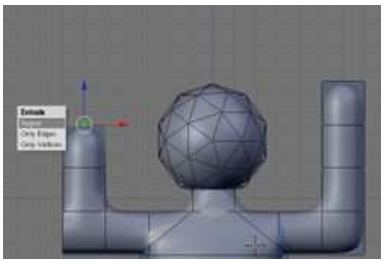




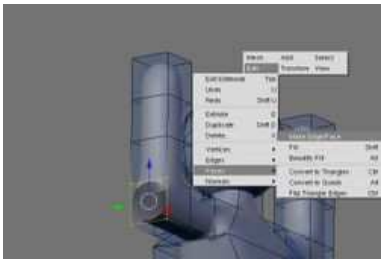
Now, select the 8 vertices on one arm that are furthest away from the torso (make sure you deselected the armpits or whatever other part you were previously modifying) and press **XKEY**, and press "*Vertices*" in the upcoming menu. Suddenly the boxes disappear, and at the end of the arm, there's a hole! Don't panic. :) We'll fix that in a moment.



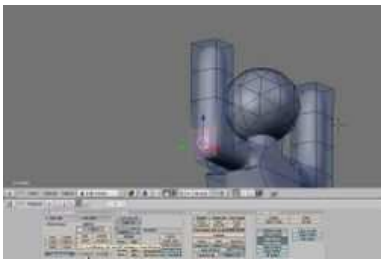
Select the top four vertices of the end-"box" (by pressing **BKEY** and dragging the box around the 4 vertices of the cube) and extrude up three times by pressing **EKEY** and **CTRL** to create three boxes the same shape.



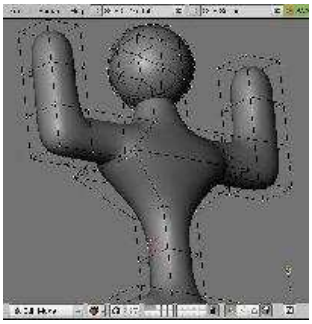
Now to fix the hollow elbow. Simply select the four vertices at the gaping hole, press **SPACE**-"*Edit*"-"*Faces*" and finally pick the option named "Make Edge/Face" (Making a face/edge can also be done by pressing **FKEY**, but the other way gives you a better impression of what you just did). Notice that the hole was covered by a face.



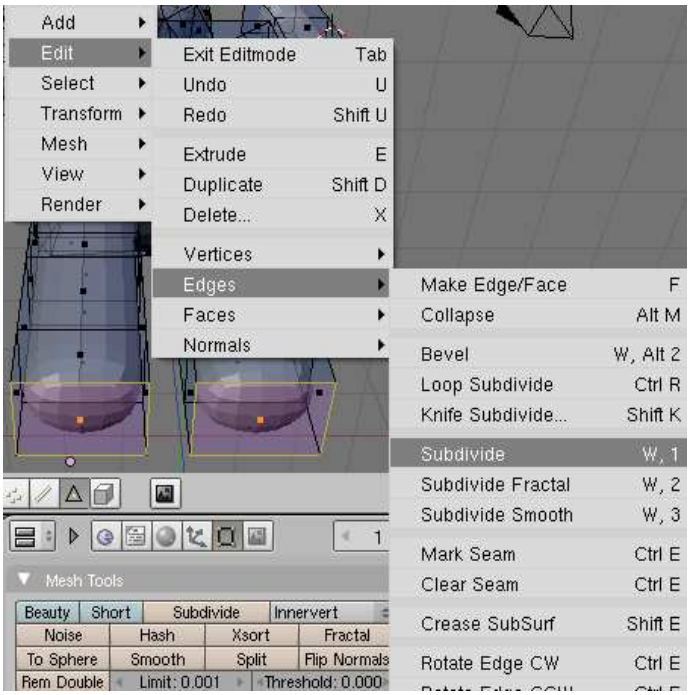
Now, click the "Set Smooth" button to make it a smooth face.



Do the same with the other side. Make sure to deselect all the selected vertices from the other arm by pressing **AKEY**. It is important to follow the steps in the same order; doing otherwise may result in arms of different lengths. Better undo your work and redo every step while selecting both sides symmetrically.



Modeling the legs



First we switch to the *Face Select Mode*. Press **CTRL+TAB** and select Faces. To select the two bottom faces of the feet (the soles of the feet), click **RMB** and hold down **SHIFT** when selecting the second one. Each face comes with a small square denoting the face center that turns yellow as well as the face outline when selected.

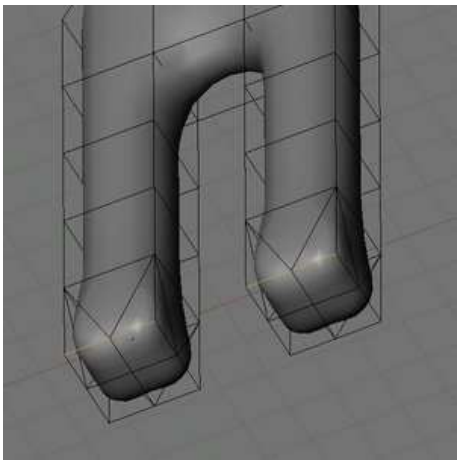
Select the 'Subdivide' Command (hit **SPACE** then, in the menu that comes up, *Edit → Edges → Subdivide*). This command subdivides the faces each into four smaller ones all of equal size. In the next step we need to switch to the *Edge Select Mode* by pressing **CTRL+TAB**.

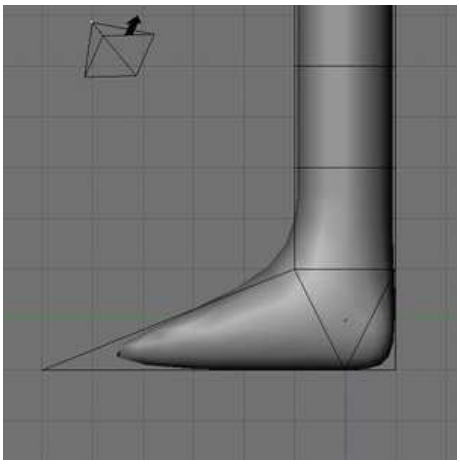
Press the **AKEY** to clear your selection.

Select (remember : click **RMB** and hold down the **SHIFT** when selecting the other one) the two bottom front edges on each leg that make up the tip of the feet (therefore, there are four selected edges in all).

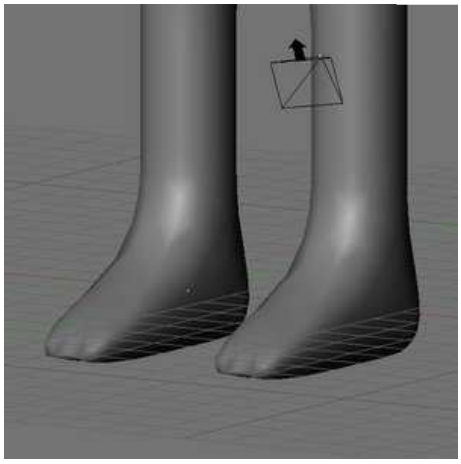
Switch to the side view with **NUM3** and press the **GKEY**. Now move the selected edges away from the legs as far as you like. Press **MMB** for orthogonal movement.

edit: Pressing the **YKEY** will also restrict movement along the Y-axis only, however orthogonal movement can be easier.



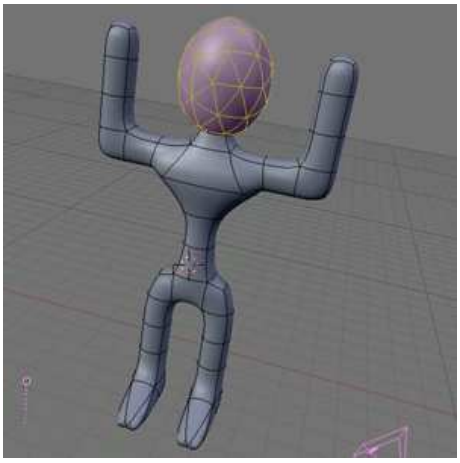


Congratulations! We now have feet!



Modeling the head

When you've got an acceptable shape for the legs, you should do something about that head. A little too spherical, don't you think ?



- Press the **AKEY** to clear your selection, move mouse cursor over the head.
- Place the mouse cursor over the head and press the **LKEY**. This selects edge, face, or vertex that the mouse cursor is closest to, as well as all edges, faces, or vertices that are linked to it. The faces for the head and the faces for the body pass through each other; however, none of the vertices in the head are linked to any of the vertices of the body via an edge or a face.
- With the whole head selected, press **SKEY** and scale it on the Z-axis in order to get a better shape. I think 1.5 is enough. Remember that you need to press **ZKEY** to restrict the scaling to the Z-axis only. It is helpful to restrict the axis you are scaling on in order to get a nice shape.
- After elongating the head, you may find that it is too low. To fix this, press the **GKEY** (to move the head) and **ZKEY** (to restrict movement to the Z-axis). Play around with it a little until you like the result.

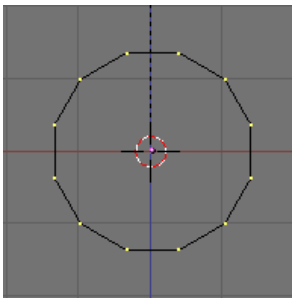
Next Page: Creating a Simple Hat
Previous Page: Detailing Your Simple Person 1

Creating a Simple Hat

Next Page: Putting Hat on Person
Previous Page: Detailing Your Simple Person 2

This tutorial uses the simple person model from a previous tutorial. If you didn't do it, go back and do it now---or find it pre-made just for you here (http://www.nusoy.com/blender) .

Adding an object

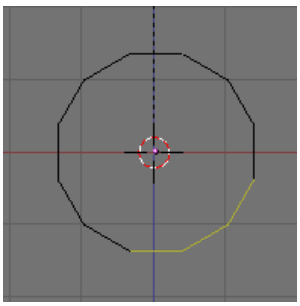


The first step to hat-making is editing a simple mesh circle.

Press **NUM3** to get a sideways view of the model. Pan the view with **SHIFT+MMB** until your view is a good distance above the simple person's head. We're going to add the hat as a separate object, so if you are in Edit Mode, press **TAB** to go to Object Mode. Click **LMB** on center of the view to place the 3D cursor (you can use **SHIFT+SKEY** then *Cursor* → *Grid* to snap the cursor to the grid after placement), then do *SPACE* → *Add* → *Mesh* → *Circle*, with 12 vertices. (In the latest version the default is 32 vertices but you may use the arrow to set it back to 12 [click and drag left or right to change number])

After the circle is added, you will automatically be placed in Edit Mode. The end result should be something like the picture (if you're in vertex mode).

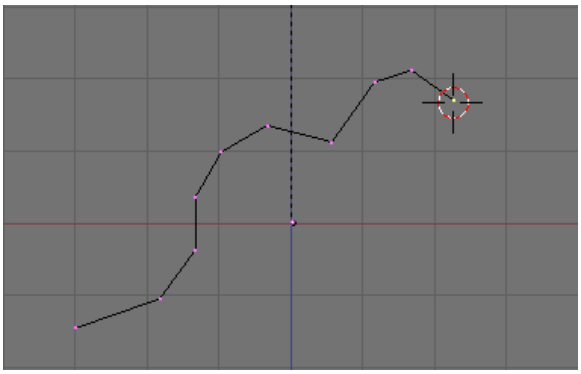
Deleting a selection



Switch to **Edge select mode** (**CTRL+TAB**) and have only the three edges selected as seen to the right (first click **RMB** and hold down the **SHIFT** when selecting the second one and the third one).

Delete these edges by pressing **XKEY** → *Edges*.

Creating the hat profile



Now switch back to *Vertex select mode* (**CTRL+TAB** → *Vertices*), and try to make the line to look something like what's shown to the right. To do that, you have to Use:

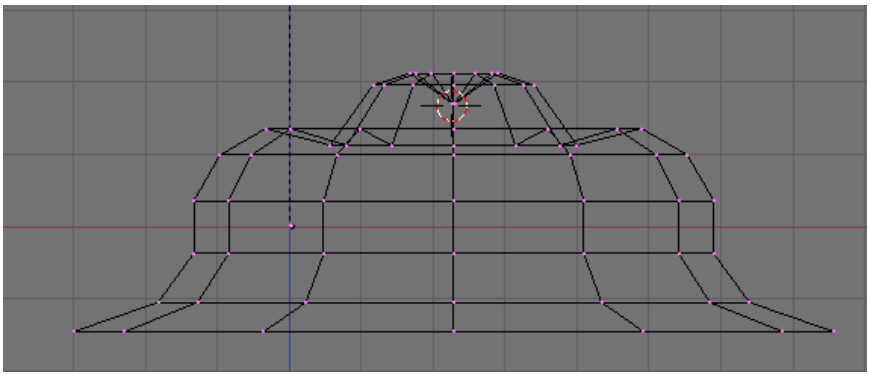
- **AKEY** to select/deselect all vertices
- **RMB** to select/deselect a vertex
- **SHIFT+RMB** to select/deselect multiple vertices, or **BKEY** for the border select tool
- **GKEY** to move a selection
- **CTRL** to move at regular intervals

Notice: In **edit mode** be sure that the **proportional edit** is off, an orange circle at the bottom of the 3D View indicates whether the proportional edit is on (orange) or off (gray), to have full control of each vertex seperately you should set proportional edit off (gray) by pressing **OKEY** (may be several times). More about proportional edit in a later tutorial, for this step switch the proportional edit off.

Spinning the hat

Make sure the 3D cursor is placed exactly at rightmost vertex (clear the selection first, select the rightmost vertex with **RMB** and then **SHIFT+SKEY** and **Cursor** → **Selection** and finally

clear the selection again thanks to **AKEY**).



Make sure "Limit Selection to Visible" is off. Change to the top view with **NUM7**, select all vertices of the hat by using the **BKEY**. Click on the Editing button (hotkey **F9**; **CMD+F9** for Mac users).

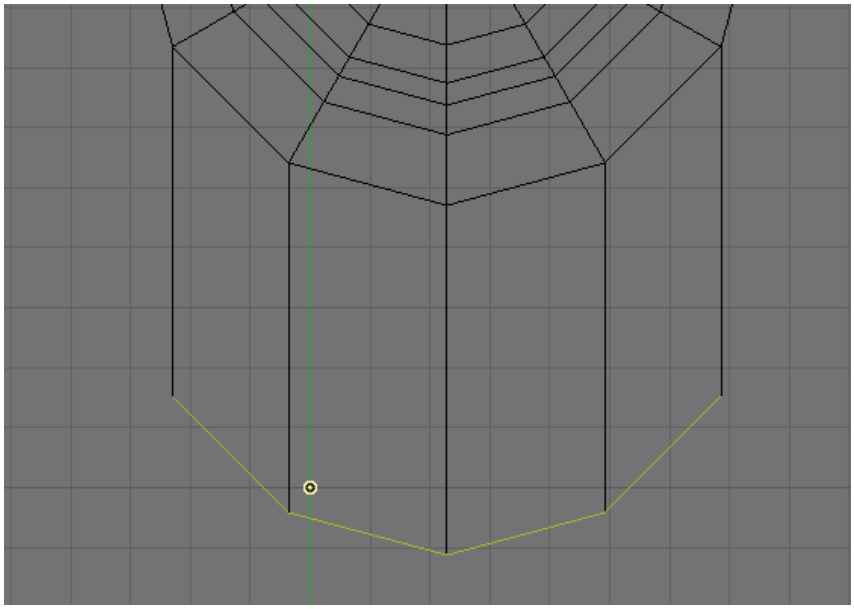
Under **Mesh Tools** tab locate **Screw, Spin** and **Spin Dup** buttons and fill the fields below:

Degr:360, Steps:12, Turns:1

Now hit **Spin** and when the cursor changes to a "?", click in the (top view) 3D window to have the polyline create a surface spun around the Z-axis.

Notice: Since Version (2.42) you have to be sure that all vertices are selected too, after following the steps above by pressing **AKEY** before you press **Spin** and no "?" will appear instead your hat is drawn instantly.

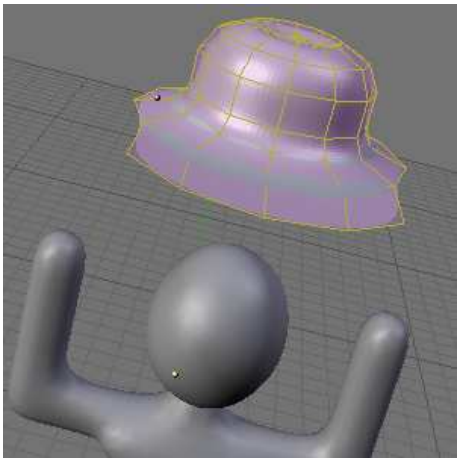
Final touches



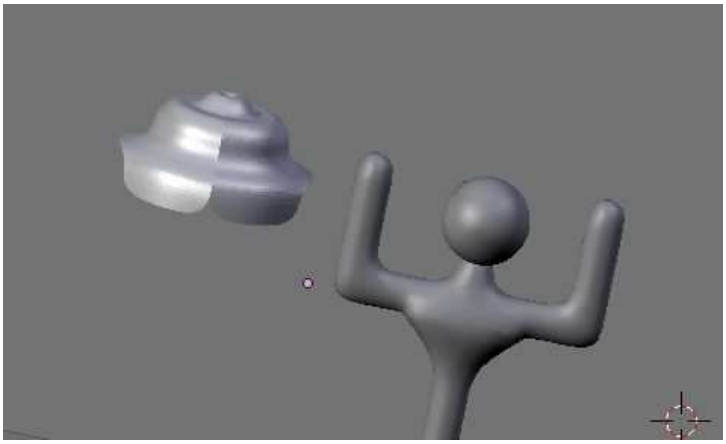
Deselect all with **AKEY**, go to the top view with **NUM7**. Use *Edge select mode* (**CTRL+TAB**), then click on the four frontmost edges of the hat with **RMB** (for the first one) and **Shift RMB** (for subsequent ones) to select them. Go into front view with **NUM1** (**NUM7**)*. Hit **EKEY** → *Region* to extrude the edges, and drag them down. You can press the **ZKEY**(**YKEY**)* to limit the extrusion to one direction.

- *You need to be in top view during extrusion so that the cap bill comes forward and not down,if it was made from the top view

Now it's time to subsurf. In the Editing buttons, under the Mesh panel, click on SubSurf and set Subdiv to 2 or 3. If you are using newer versions such as Blender 2.4, you may need to go to *Modifiers → Add Modifier → Subsurf*. Rotate the view around and you will notice that your hat has a "split at the seam". Because of the settings we chose for the spin, there are several pairs of vertices that share the exact same spot in 3D space. To correct this problem, in edit mode hit **AKEY** to select all vertices and then hit **WKEY** → *Remove Doubles*. Now all our seams will display correctly, and you have removed the unnecessary overlapping vertices in your mesh hat. Whew! You now have a lovely new hat! Pat yourself on the back, good work! You can neaten it up a little more by hitting **WKEY** → *Set Smooth* to give it a nice smooth finish.



Another view of the final hat looks like this.



Keep this simple person/simple hat file open because you will be using it still in the next tutorial.

Next Page: Putting Hat on Person
Previous Page: Detailing Your Simple Person 2

Putting Hat on Person

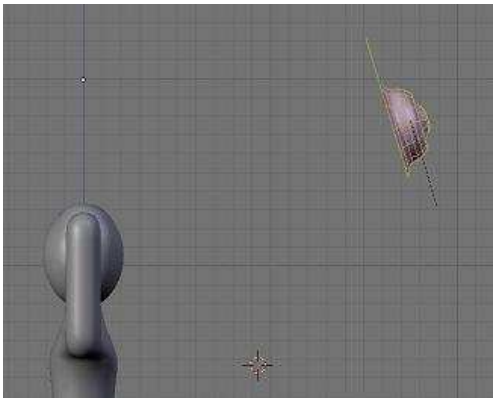
Next Page: Mountains Out Of Molehills
Previous Page: Creating a Simple Hat

This tutorial uses the simple person model and hat from a previous tutorial. If you didn't do it, go back and do it now---or find it pre-made just for you here (http://www.nusoy.com/blender) .

Once you have created the hat, and are satisfied with the 'form' of it, now it's time to change the rotation, location, and size of the whole object in 3D space. Switch to Object mode and select the hat.

Rotation

First, change the rotation of the object. To change the rotation of the hat, press **RKEY**. Now you can move your mouse around to change its rotation. It will rotate on a different axis depending on what viewpoint you are rotating it from. The rotation axis will always be perpendicular to your viewpoint, so it looks like you're rotating a 2 dimensional image. Press the **RMB**, or **ESC**, to bring you back to the original rotation.



When you press the **RKEY**, you are actually entering a rotation mode that can be altered by further key strokes. For instance, pressing the **YKEY** after the **RKEY** will rotate the hat about the Y-axis. Pressing the **ZKEY** will rotate it about the Z-axis, and the same goes for the **XKEY**. By pressing the **ZKEY**, **YKEY**, or **XKEY** just once, you will rotate the object in relation to the scene. If, however, you press the same key twice, it will rotate in relation to its origin, giving you an XYZ arrow pointer thing that displays the angle of the origin.

Important to note is that the shape will rotate around its origin, or center point, indicated by a small, blue dot that was created when you created the shape. It should be in the center of your vertices, but if it isn't, there are a couple of ways to get it back. One is to go into edit mode, select all vertices, and move them around the center point. Another is use the **LMB** to put the cursor where you would like the center point, go into object mode and press the "center cursor" button in editing panel (**F9**). Or you could hit **SHIFT+SKEY**, select *Selection → Center*.

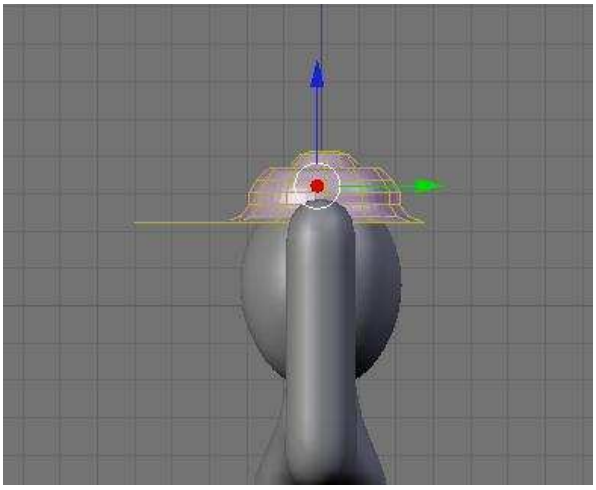
Hit the **NUM1** on the numberpad to get the front view. Hit the **RKEY**, followed by the **ZKEY** and move your mouse. This will rotate the hat perfectly around the Z-axis. Hold down the **CTRL** button so it only rotates in 5 degree increments and click the **LMB** when you come to the correct position. (Do this with the X- and Y-axis if needed).

Alternatively, you can click and drag the **LMB** in a circular motion around the object, to "draw" an arc. This is called a mouse gesture and has the same effect as pressing the **RKEY**.

Location

After you have the hat in proper rotation, you will want to move it to the proper position. You do this the same way you move an individual point. Press the **GKEY** (for "grab") and move the mouse. Pressing the X, Y, or Z key will have a similar effect as it did with rotation, restricting the movement to the X, Y, or Z axis. Pressing the **MMB** while moving will also restrict the movement. Pressing the **RMB** will reset the object to its original position, without making any changes.

Alternatively, you can click and drag the **LMB** in a straight line to activate moving the object. This is another mouse gesture and the same as pressing the **GKEY**.



Size

You may notice that the hat is a little big for the person we created. No problem, we'll just change the size. You do this by pressing the **SKEY**, for "scale". You can scale the object just along its X, Y, or Z axis, making it thinner, shorter, fatter, or wider.

Alternatively, you can click and drag the **LMB** back and forth from the object to scale it. Start at the object, move your mouse a little away from it, then drag back to the object to draw a line and go back over it. This is, you've guessed it, a mouse gesture as well and the same as pressing the **SKEY**.

So, just remember:

- **SKEY** is for Scale
- **RKEY** is for Rotation
- **GKEY** is for Grab (Move)

Putting it on

Once you have the hat in position, you will want to "put it on". To do this, we make the man the 'parent' of the hat. What this means is that, when we move the hat, we just move the hat. However, when we move the man, we move the man *AND* the hat.

> Save your work before doing the following because the program may crash and be unusable if you accidentally press **PKEY** instead of **CTRL + PKEY**. (Note: **PKEY** starts the Blender game engine. If you do accidentally press **PKEY**, **ESC** should stop it and bring you back.)

Make sure that you are in object mode and the hat is selected. Hold down shift and select the man by pressing the **RMB**. Both the man and the hat should now be selected. Hold down **CTRL** and press **PKEY** and select "make parent" in the confirmation box to make the man a parent to the hat. Now you will see a line from the hat to the man, indicating that the man is the hat's parent. If you move the man, the hat will move along with him. Otherwise if you only move the hat, the man will stay at its place. Don't forget to pay attention to the order of your selection. The first selected object becomes the child of the second one.

If you want to add color to the person, press **F5**. You will see a color preview (should be gray) at the far left of your color table at the bottom of the screen, You can make it any color you choose (purple, blue, flesh tone--you name it!). Since version 2.41 this is changed to three small bars on top of each other labeled R,G and B (red, green and blue)

You're probably thinking, "How does Blender make mountains out of molehills so well? It took me half an hour just to put the hat on!" If so, read on!!!!

Next Page: Mountains Out Of Molehills
Previous Page: Creating a Simple Hat

Mountains Out Of Molehills

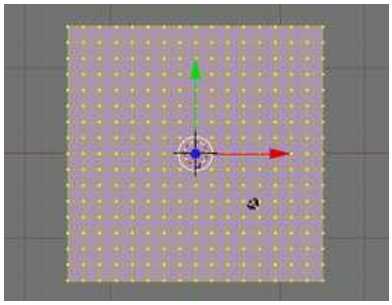
Next Page: Turning a Cube into a Puppy
Previous Page: Putting Hat on Person

Now that we've created our simple person, it's time to give him somewhere to go. In this tutorial we'll create a mountain range using a few simple, and handy tools.

Start off with a new project, using *File → New*, or hit **CTRL+XKEY**. If you have a default cube or plane just delete them now (select them with **RMB** and press **XKEY**).

Creating a simple plane

Our first step is to create a large plane that we'll use for the ground and grow our mountains out of.

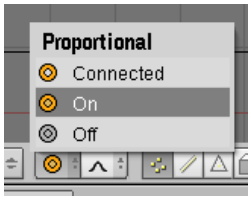


- First press on **NUM7** to enter top view. This way our plane will be lying flat when we create it.
- Now add the plane with **SPACE** → *Add* → *Mesh* → *Plane*. This will be our canvas.
- Scale the plane up by about 15. First put the mouse close to the center of the plane and press **SKEY** and drag the cursor away and watch the numbers in the bottom left of the 3D Window. Hold **CTRL** to increment by 0.1 for a more precise measurement. Alternatively, to enter the exact amount yourself press **SKEY**, then **NKEY** for a numerical entry, and type **15** and hit **ENTER**. (In later versions of blender, just press **SKEY** and begin typing a number for numerical entry.)
- Now we need to add some vertices to work with. In the buttons window, make sure we have the *Editing* buttons open (or hit **F9** in the buttons windows to switch there). Under *Mesh Tools* hit the *Subdivide* button 4 times. Alternatively, in the 3D View window you can press **WKEY** and select *Subdivide* (Or just hit **ENTER**).

First mountain

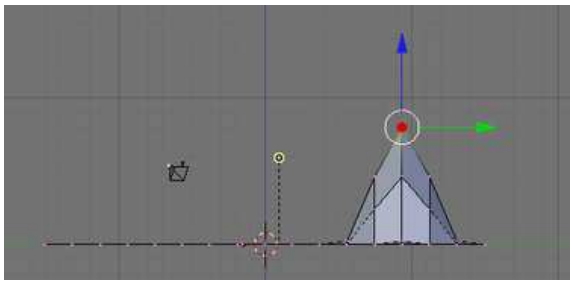
Now that we have the ground, it's time to start growing our mountains.

- Select a random vertex with **RMB**. I usually start at the one that is 3 down from the top, 3 in from the left (the 4th vertices if you count the edges).
- Change to the side view with **NUM3**.



- Press **OKEY** to change to proportional edit mode or use the button which shows a grey ring on the header of the 3D View. The button will change its color to orange. You can also use **SPACE** → *Transform→Proportional Edit*
- Once you've turned proportional edit mode on, another button appears to its right, the falloff button. Select Smooth Falloff here. Alternatively you can use the menu on the header of the 3D View (*Mesh → Proportional Falloff → Smooth*) or, using **SHIFT + OKEY** will switch between Sharp and Smooth Falloff (in versions prior to 2.37) or cycle through all 6 falloff types (in versions 2.37 and up) while using the Proportional editing tool.
- Press **GKEY** to grab the vertex. We should now have a circle surrounding the vertex, this is our *radius of influence*. Basically any vertices inside this circle will be affected by any changes to the vertex itself.
- Use the **Mouse Wheel** or **PAGE_UPKEY** and **PAGE_DOWNKEY** to adjust the radius of influence to include just over 2 vertices on each side of our selected vertex. (Depending on your version of Blender, you may need to use **ALT + NUM+** and **ALT + NUM-** and may need to hold down the **LMB** while using the **Mouse Wheel** to adjust the radius of the influence.) In 2.41 you must 'grab' the vertex first - only then can you alter the sphere of influence.

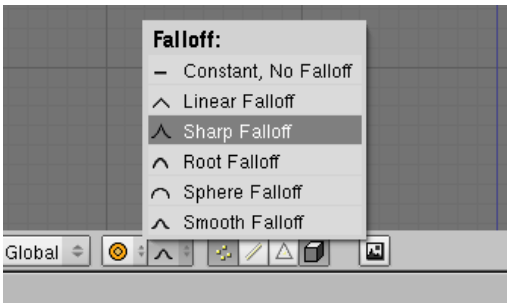
- Move the vertex up about 8 units on the Z-Axis. Do this by dragging the cursor up a little, and press the **MMB**; this should restrain the movements along the Z-axis. Now use **CTRL** to move it precisely. Alternatively you can use **ZKEY** to restrain movements to the Z-Axis and type **8** and hit **ENTER**. In older versions of Blender you may need to hit the **NKEY** before typing the number 8.



Congratulations, we just created our first mountain.
Now it's time to see what other things we can accomplish with the proportional editing tool.

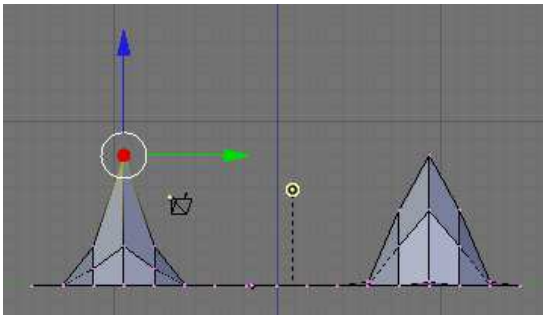
Peaks vs. hills

The 2.37 and onward releases offer 6 types and 2 modes of proportional editing. The previous release only has 2 of these types: Smooth and Sharp Falloff. We'll take a look at the difference between these two now.



- Change to top view again with **NUM7**.
- Select another vertex away from the first. Lets say 4 from the bottom 4 from the left (counting the vertices on the edges).
- Change back to the side view with **NUM3**
- Select Sharp Falloff from the menu on the bar of the 3D View. Alternatively, using **SHIFT+OKEY** will switch from one to the next of the 6 proportional editing modes while using the Proportional editing tool.
- As before, move the vertex up 8 units on the Z-Axis (*Note: The radius of influence will still be the same size as when we last used it*).
  - **GKEY**
  - **ZKEY**
  - Type **8** and hit **ENTER**

Now we can see the differences between the sharp and smooth falloff. The same number of vertices are affected in both cases, only the degree to which they are affected is altered.



The different proportional editing modes can be selected from the box immediately to the left of the proportional editing type box. The mode box contains three options: Off, On, and Connected. Off means that proportional editing will not be used. Connected means that only vertices linked to the selected vertices will be affected by the radius of influence. On means that all vertices will be affected.

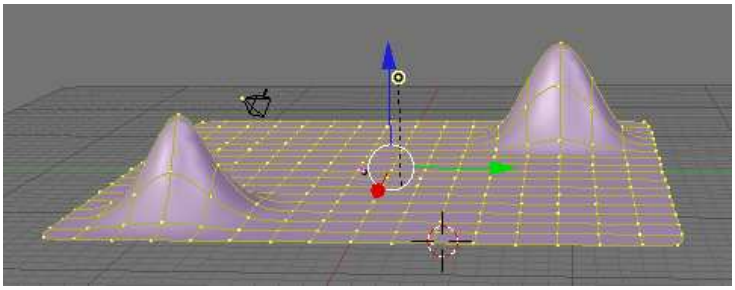
Shaping the world

Now that we've created a couple of Mountains, it's time to see how we can use proportional editing to shape them.

- First make sure we're in side view (**NUM3**).
- Then on the smooth falloff mountain, the first one we created, select the vertex that is immediately down and left from the topmost point.
- Press **RKEY** to rotate, and hold **CTRL** and rotate everything by -90. Alternatively, use **RKEY**, **NKEY**, and type **-90** and press **ENTER**

Feel free to play around with scaling or rotating from different view points.

Smoothing things out



Now that we have a couple of budding mountains, you probably think they look kind of choppy. Sure they would be good if we were making an 8-bit console game, but we're working with 3D here, we want things to look sharper than that. There are a couple of approaches to this. The first is to use more vertices when we create our plane. And I won't lie, it works. But it's also a HUGE resource hog. It would take your home computer hours of work just to keep things updated, let alone run it. So instead, we fake it. The easiest way to do this is to turn on *SubSurfaces* (we saw this in Detailing Your Simple Person 1. For our purposes, let's set the subdivision (*Subdiv*) to 2. Also, ensure our SubSurf algorithm is set to *Catmull-Clark* (this is the default setting).

Now, you'll notice that with SubSurf on, we lose a lot of hard edges that we had, essentialy we have no sharp corners anymore. I don't know about you, but to me that doesn't make for a very interesting mountain range. So to restore our corners, we are going to use *Weighted Creases for Subsurfs*.

- First turn off proportional editing with the **OKEY**, and ensure we're in side view with **NUM3**
- Next, while still in edit mode, change to *Edge Select* mode with **CTRL+TAB** and select *Edges*. Alternatively press *Edge Select Mode* button at the bottom of the object window.
- Under the *Edit buttons* under *Mesh Tools 1* ensure that *Draw Creases* is selected.
- On our Sharp Falloff mountain, the second one we did, select the first two edges on the right (starting from the point down).
- Press **SHIFT+EKEY** or **SPACE** → *Edit* → *Edges* → *Crease SubSurf*, then move the mouse away from the edge until the *Edge Sharpness* reads 1.000.

As you move the cursor away from the edge you will notice two things, the first is that the edge becomes thicker as we move from it; this is showing how much of a crease we have (with *Draw Creases* turned on). The second is that you will notice the subsurfed mesh moving closer to the edge as the sharpness increases.

Naturalness

Push **CTRL+TAB** and select vertices then go into front view **NUM1** and select the second vertex from the top in the center, go into side view **NUM3**. Push **GKEY** and drag the vertex inwards, not too far or your mountain will come out of itself on the other side. Just bring it in enough to make a small indent.

Then grab the top vertex and pull it down a small amount. You will notice that there is a small "crunch" in your mountain.

Don't forget to hit the '*set smooth* button to smooth everything out.

OK, so your mountains are starting to shape up. But they still look a bit too neat. You could spend time moving each individual vertex but the chances are your model will still lack the natural feel. What we need is some chaos. Thankfully this is quite easy to accomplish. Firstly select the vertexes that make up your mountain, all of them and a few around the base (box and circle select will make this easier). Select a few vertices between the mountains too. Next we use something called fractals. Fractals are chaotically (ie randomly) generated variables. In short you can use these variables to give your mountains a "wobbly" look. Fractals are located in the Mesh Tools section of your edit buttons (next to Noise, Hash and Xsort). Click it and you'll be asked for a value. This value is the strength of the fractal. 1 is very low and will barely change your model. 100 is very high and will twist you models into very odd shapes indeed. Have a play with different values until you find one that you like.

Around about 30 should do it. Hit OK and hey presto, your mountains have been transformed from clinical neatness, to lumpy chaos.

(Under Construction [TODO: finish me])

Next Page: Turning a Cube into a Puppy
Previous Page: Putting Hat on Person

Turning a Cube into a Puppy

Next Page: Modeling a Gingerbread Man
Previous Page: Mountains Out Of Molehills

Creating Models With Photo Assistance

This is a rewritten tutorial, the original is listed below.

My tutorial is about using guide images to place vertices in their proper places in 3D space. The original tutorial below mine is on how to take good reference pictures. My tutorial assumes that you have completed all previous tutorials.

Setting UP

Open a new blender project.

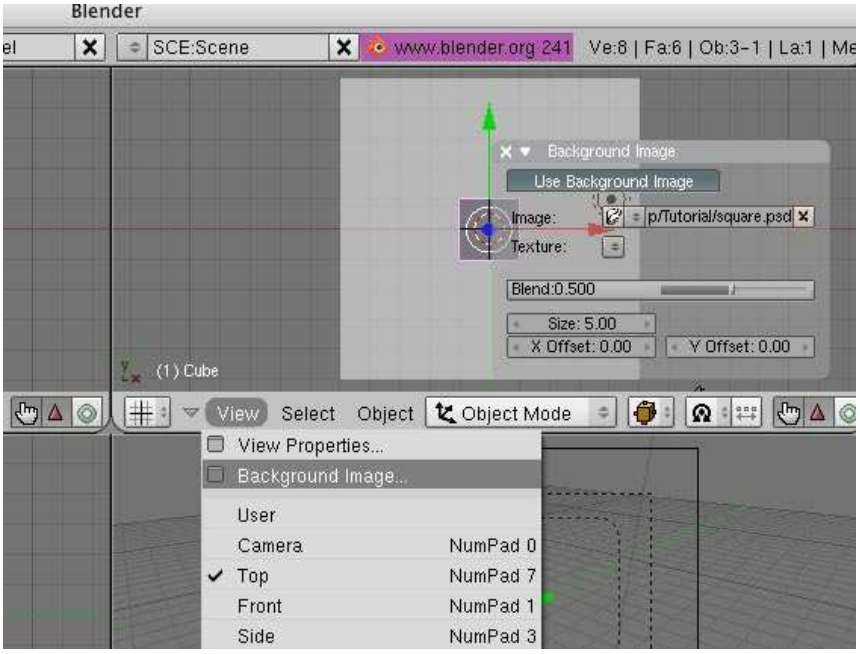
Split the Main 3D view window in to 4 windows. Explained in guide: Noob to Pro/Blender Windowing System.

Change the point of view in each window so that they end up like this:

Num 1 Num 7
Num 3 Num 0

Make a picture of a white square and of a white triangle in photoshop or other image maker. Save them to a place that is easy to access.

Load the white square into the **Num 7** window by going to the 3D view window and pressing View->Background image-> (click on the icon of a file) find your file and click “select image”



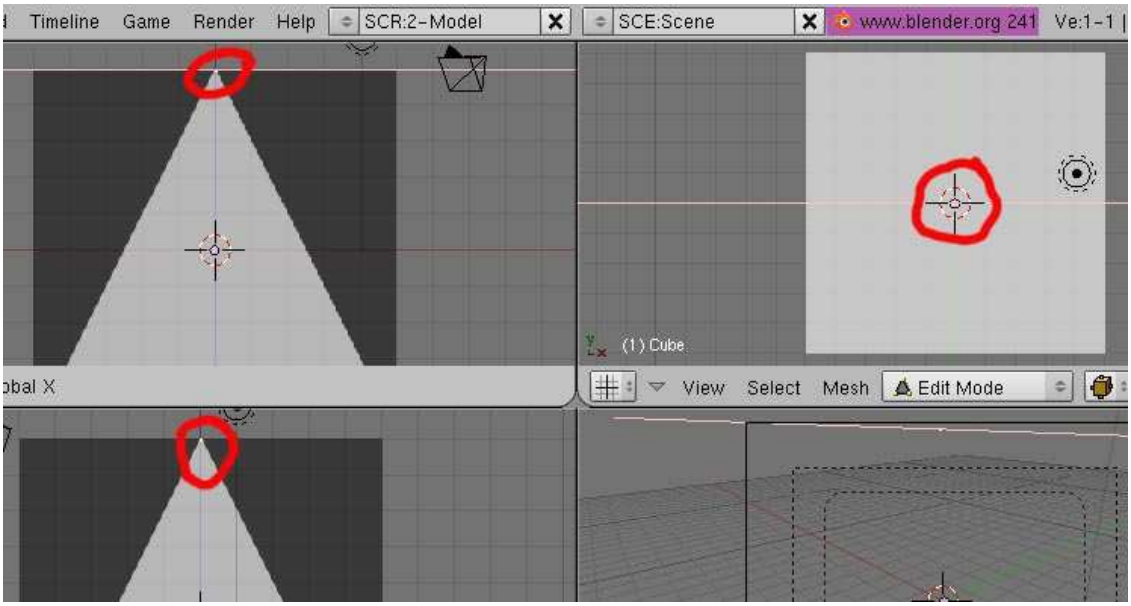
Load the white triangle into the **Num 1** and **Num 3** windows.

Now you have a guide for making a pyramid.

Making A Pyramid

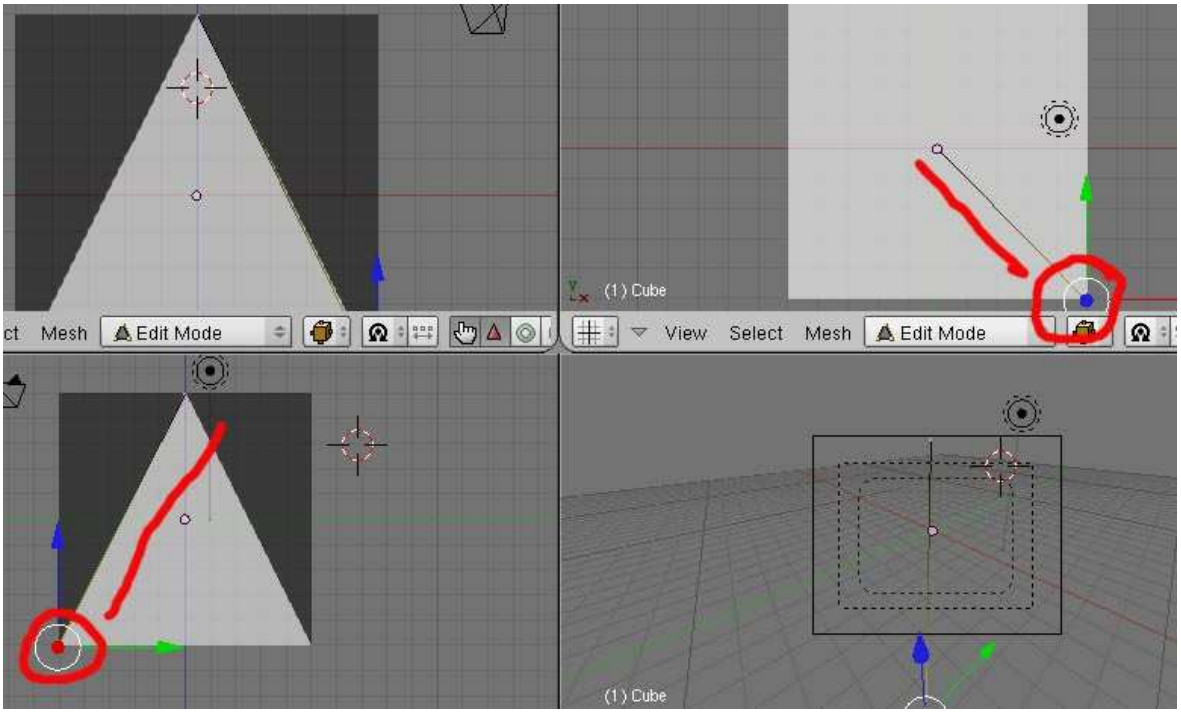
You should have a cube given to you when you open a new project. Control Tab to **vertices** and delete all but one vertex of your cube.

We will build the pyramid vertex by vertex. The first vertex will be the topmost point of the pyramid. Use the **GKEY** to move the vertex around. To get it in the right spot, line it up at the top most point in the **Num 1** and **Num 3** windows. If you look in the **Num 7** window the vertex should appear to be in the center. Make sure to keep the vertex highlighted for the next step.

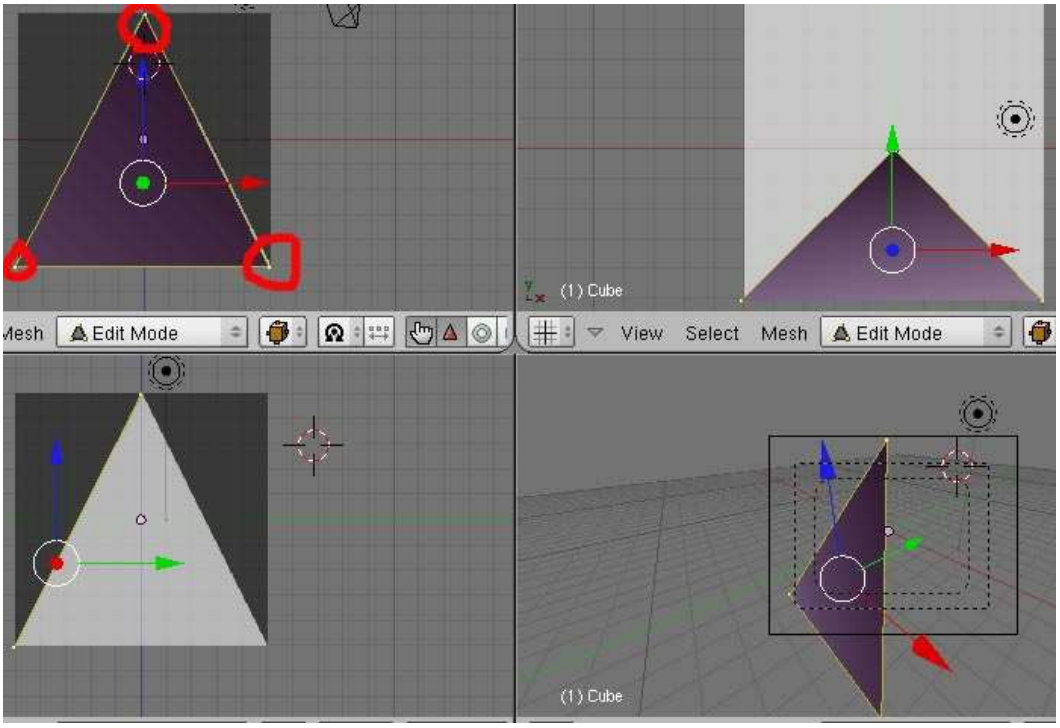


(note: it might be helpful at some point to zoom in and use the X,Y and Z movement restriction)

Next in the **Num 3** window, place a vertex on the lower left edge of the triangle by holding **control** and pressing **LMB** Line it up in that window and also at the lower right point in the **Num 7** window. This should create a line between your 2 points.



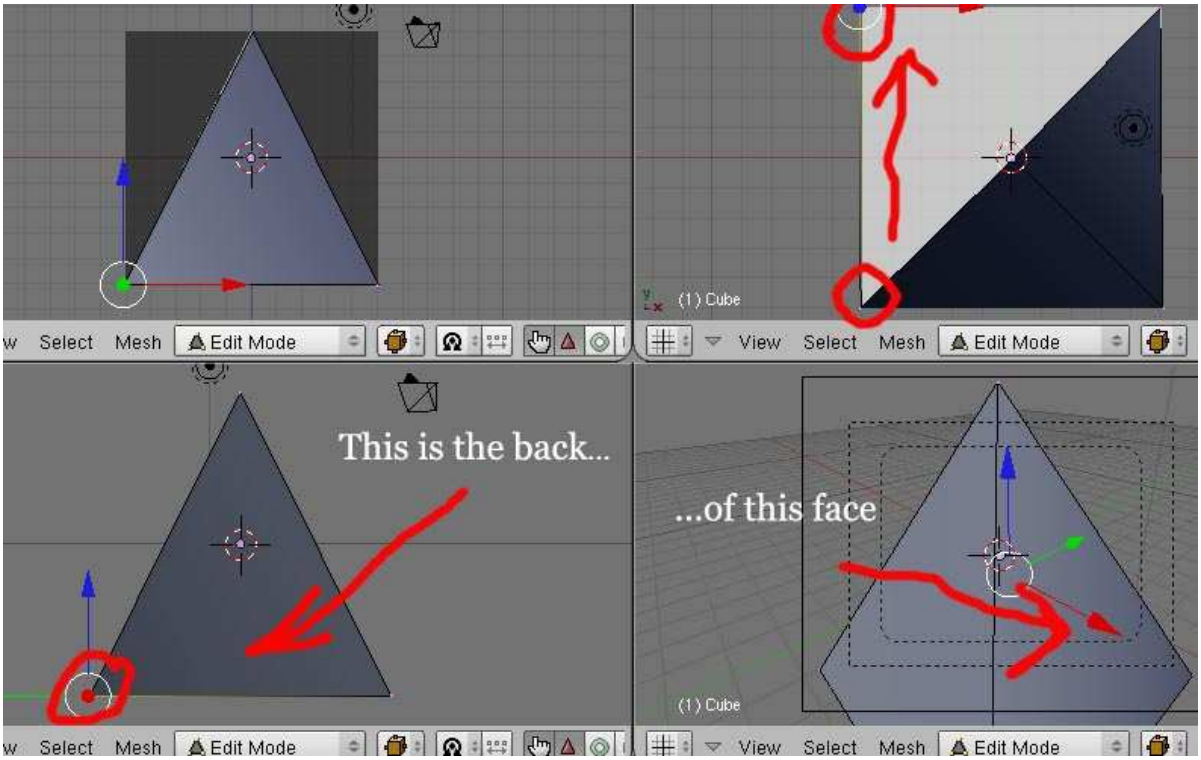
Next in the **Num 1** window, place a vertex on the lower left edge of the triangle. Line it up in that window and at the lower left point in the **Num 7** window.



Next highlight your 3 vertices in the **Num 1** window and press the **FKEY**. You should see a triangle appear. Press **AKEY** to deselect and then select the vertex at the top of the pyramid.

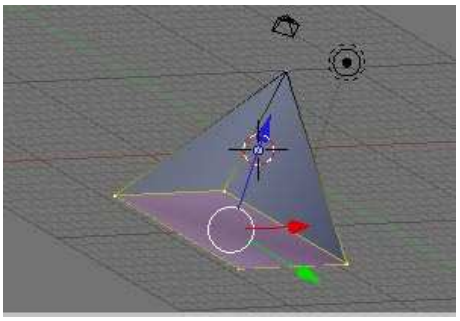
Make the next face by placing a vertex in the upper right point in the **Num 7** window (this should line up with the lower right point in the **Num 3** window.) Make a new face in the **Num 3** window with the **FKEY** and keep the face selected after you have made it.

In the **Num 3** window, rotate around 180Deg. (**control+alt+scroll** for MAC users), so that you are looking at the back of your new triangle. Deselect all vertices and in the **Num 7** window, select the lower left vertex. Then **Control LMB** place a new vertex on the upper left point in the **Num 7** window. It will line up with the lower left point in the **Num 3** window. Select the 3 vertices in **Num 3** window and make a face.



Control tab to **faces** and select the face in **Num 1** window and rotate around 180 Deg. **Control** tab back to **vertices**. Deselect all, and select the three vertices and make a face.

In the **Num 7** window select all four corners and make a face. You should have a solid pyramid!



This is a silly tutorial because you could make a pyramid by simply select the top four vertices of a cube and merging them together. I wanted to show how you could use different viewpoints and images to help to guide you to place vertices correctly in 3D space.

{[quote]You could make a pyramid by simply select the top four vertices of a cube and merging them together.[/quote] If I'm right, this wouldn't be the best to do: You always would have to select all four verticles, because they don't count as one vertex, but as four. You only wouldn't be able to size them any more, because every vertex is at the same point and how do you want to make a gap of lengh "0" wider than it is? So a third way to do is: delete the cube and create (In Top-view) a cone with as many verticles as the pyramide should have. (3 verticles for a 3-sided pyramide and 4 verticles for a 4-sided pyramide.) then rotate the view until you can see the bottom of the pyramide, mark the middle of it, press x and choose "verticles". Next, mark the rest of the bottom-verticles and press F to create faces. (if you only want ONE face and create a pyramide with more than 3 sides, then you need to follow the other two descriptions!) Choose that one, you like best: 1) Tricky, but a perfect pyramide 2) Very easy and very short, but the "apex" has 4 verticles instead of one 3) Easy and short, but pyramides with more than 3 sides will have more than one face at the bottom. (If you find any mistake in the part of the text marked with {}, please correct: I'm from Austria so my native language is german)}

[this is an interesting point but let me explain myself.

> The point of the exercise was not to create a pyramid, but to learn how to plot 3D space. I used a simple 3D shape to make things easy.
> Second, I should clarify about making a cube into a pyramid. If you select the top 4 vertices and hit **ALT + MKEY** you will get a prompt that says: Merge, and a choice to merge the points in the center or at the cursor. Blender deletes all of the doubles for you!Badalia]

The original tutorial is listed below with a discussion on how to take usable pictures.

Taking the Best Reference Photos

this is also the original tutorial

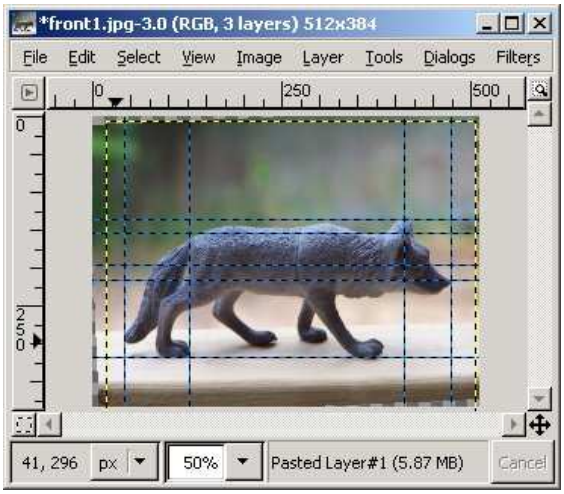
Step One: Get the pictures of the model

As of now, we have one picture. So, if you have a puppy and a digital camera, please take three pictures of the cute little rascal, upload them to Imageshack, or your favorite image repository, and I will check back occasionally and will make whatever modifications are needed to the images to make them work for this tutorial (which will also be documented here.) Ideally, the photos will be looking straight down at the top of the puppy, a side view, and a front view, and, most importantly, the puppy should be in the same pose in all three photos! Or at least close to the same pose...we all know puppies don't stand still very long.

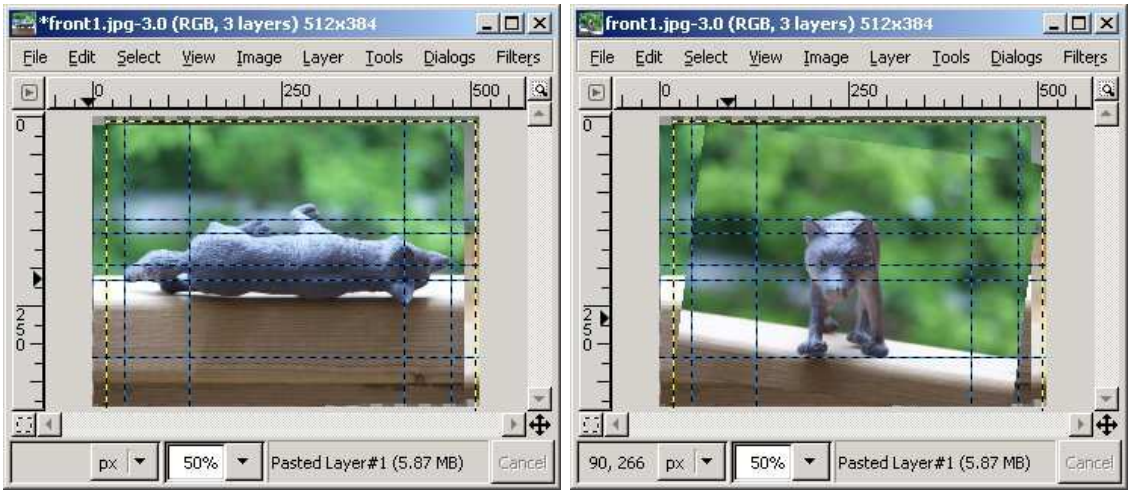
You could use two mirrors. One is placed next to the puppy at 45 degrees to the camera and 45 degrees to the puppy. Another is placed above the puppy, also at 45 degrees to the camera and 45 degrees to the puppy. This produces three images, one of the puppy (**NUM1**), one of its reflection seen 90 degrees to the right (**NUM3**), and one of its reflection seen from overhead (**NUM7**). Take the photo from a long distance away with a zoom lens to get close to an orthographic projection.

Or how about a toy wolf taken from 6 views: Left view
(http://img51.imageshack.us/my.php?image=10011325wu.jpg) Right view
(http://img162.imageshack.us/my.php?image=10011337aj.jpg) Front view
(http://img107.imageshack.us/my.php?image=10011347fw.jpg) Back view
(http://img107.imageshack.us/my.php?image=10011350gu.jpg) Bottom view
(http://img162.imageshack.us/my.php?image=10011363ql.jpg) Top view
(http://img119.imageshack.us/my.php?image=10011375bc.jpg)

Using your favorite image editor such as PhotoShop or the GIMP, the images need to be down-scaled to a reasonable size (I made mine 512x384), and then matched to each other. To match them, draw construction lines (pulled from the rulers above and to the left) on the left view for example to pick out key features. I picked the tail, front of back foot, eye level, tip of the ear, front of the nose:



I found when I picked out these features that this first image needed to be rotated slightly. That completed, I proceeded to scale, rotate and shift the other two views (top and front) until they matched fairly well as layers on top:



Once I had the proper results, I saved the resulting images, and these are the ones we will use in Blender.

The results are the files you'll need for Step Two:

Just right-click and save them some place where you can find them to load them into Blender for Step Two.

Step Two: Get the Picture into Blender

(user comment) so how do you make it 3D in blender? all i can see is user comments about how it didn't work? and then some pictures in blender but no explanation.

(user comment) To get an image into blender as background, simply go to each viewport's header (the menu at the bottom), click on 'View' and, from the menu provided, select 'Background Image ...'

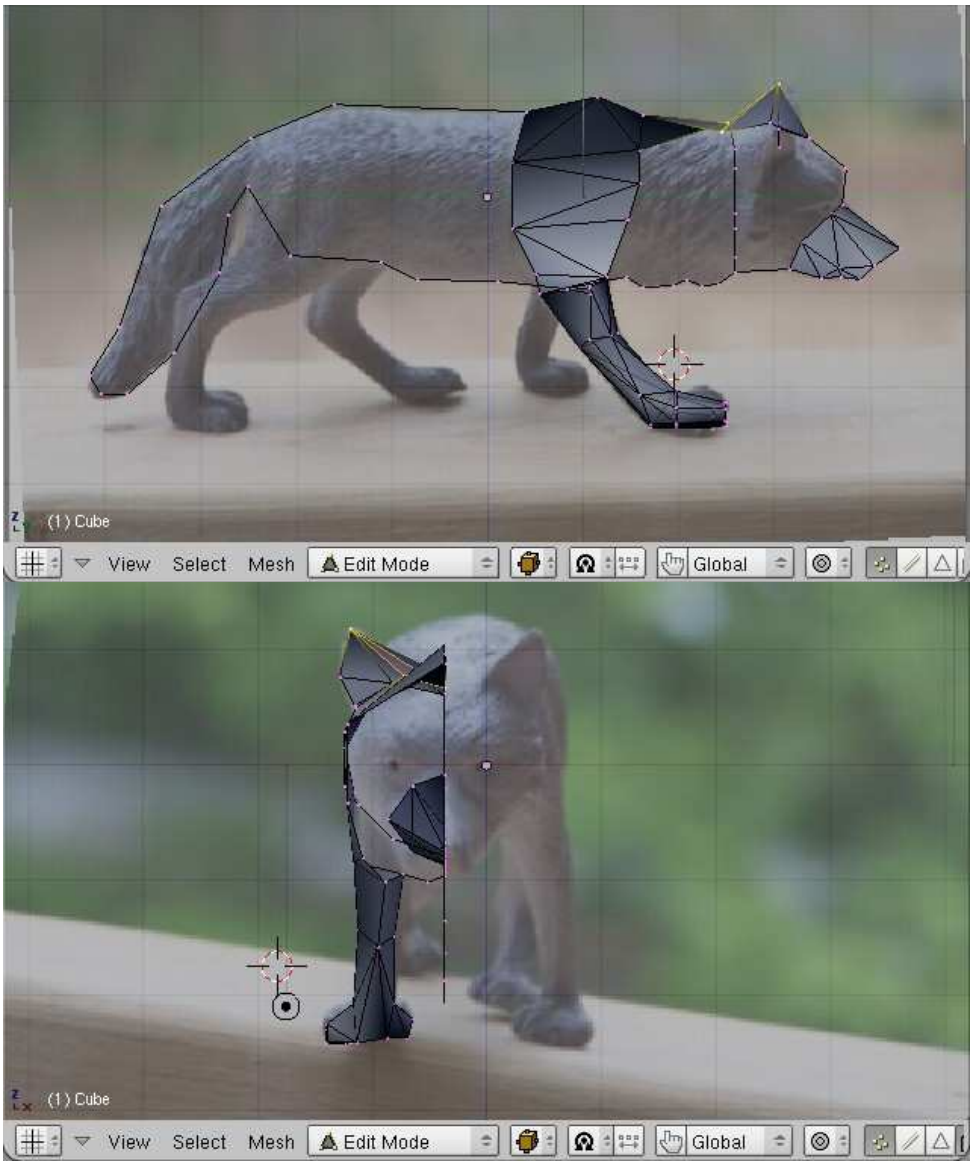
--86.59.104.195 05:42, 16 August 2006 (UTC)

(user comment) I am using v. 2.41 and it seems that the provided top view of the wolf toy is off by 90 degrees as a background image (that is, it has the wolf's nose to the right, but all my points in 3D according to the other two views would have his nose to the top). Am I out of my mind or doing something noobishly foolish? In the meantime I've manually rotated the image and continued on my merry way.

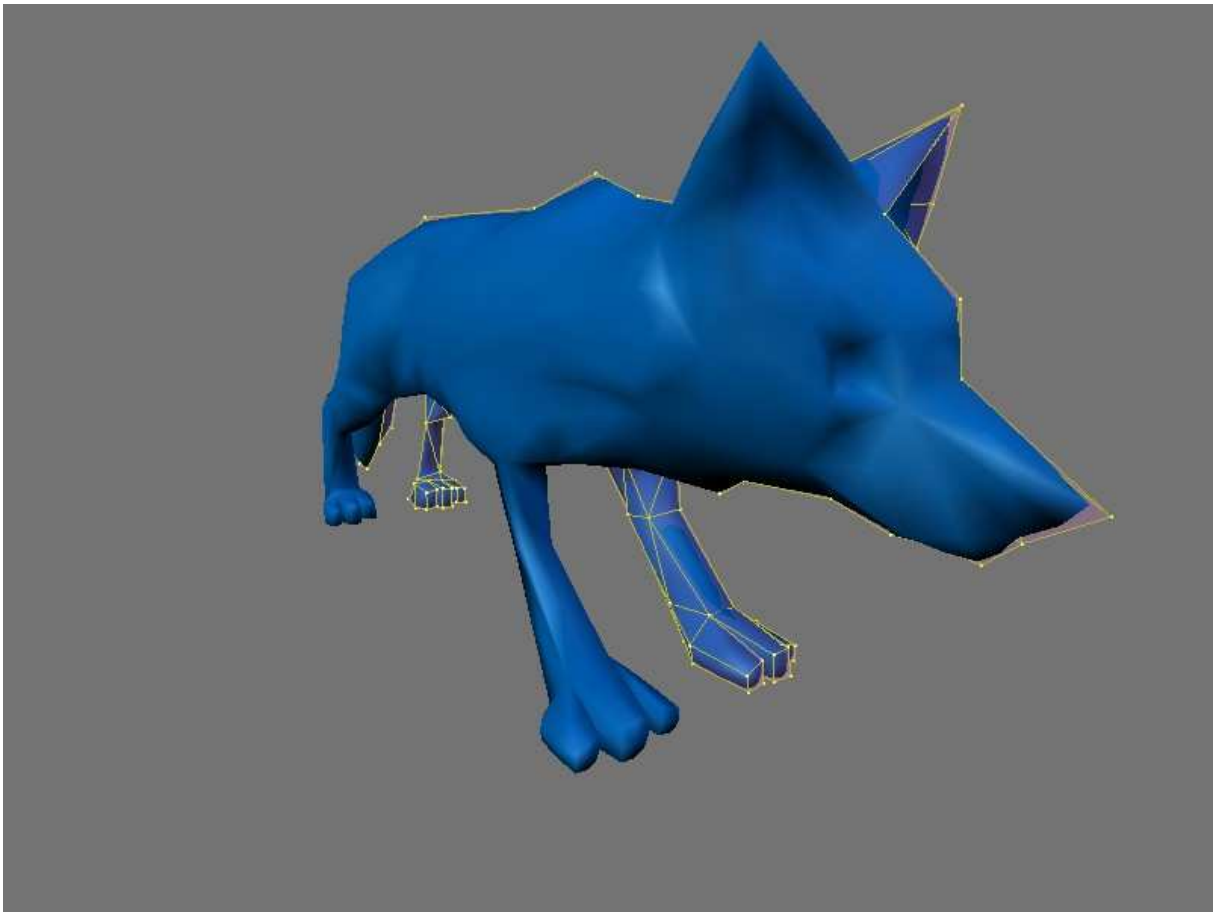
(user comment) I don't know too, why they go this way to dispay an ISO image, but if you use **NUM3** for front view, **NUM1** for side/left view and **NUM7** for top view it will work perfectly, the 3D view 'ISO' arrangement seems to be only in a little bit wired clock-wise view. I guess, this is probably the way 3d designer work out a drawing instead of many engineer disciplines will do ;)

(user comment) I was doing this tutorial and though I´m still a noob at this I thought I

could contribute a little to this. I decided only to do half a wolf and then mirror it to create a complete wolf. I started out with creating the silhuette of the wolf body in the side view, and then in front view I started to shape the wolf head, tail and legs...



And at any rate this is what I ended up with. Not the best wolf ever, but as I said before I´m still pretty noobish.



(user comment) I too was doing this tutorial when I realized the pictures were a bit off. So I decided to do my own photos of my own object. However this turned out to be more difficult then I thought at first. While taking your own photos be aware of several things. The camera you are using has to have its lens **parallel** to the object you are taking a picture of. Also when you take the 3 different photos (front, side, top) or even more you need to keep the camera the exact **same distance** from the object as well as the exact **same zoom** on the lens for all your pictures. Once you have your photos, rather then lining up key points in the photos you should center your object in the center of the photo.

(user comment) It's not necessary to center the object in the photo. In the Background Image window, you can tweak the center point with X Offset and Y Offset.]

From this point forward there are multiple ways to do the exact same thing, however for simplicities sake and so that I can be more detailed I will be using one method (the one which I use) and be using GIMP.

I found it best to size all the photos to a known width, with an easy to find center. (Mine happened to be 850x638 pixels, I don’t recommend that but you can choose any size you want really, as long as all of them are the same size). Then drag the construction lines to form a crosshair in the middle of the photo. To do this, click on the top ruler, and drag down to the middle (Exact middle) of the photo, then click on the side ruler and drag across to the middle (Again exact middle) of the photo.

If you are having troubles finding the exact middle of the photo, move the cursor to the very bottom left of your photo and the height of your photo will be listed at the bottom left of the GIMP interface. The numbers are listed in an (x,y) format so you want the first number to say 0 and the second to be the largest you can make it by dragging your cursor. The second number is the height, and half of that is the middle of your photo. You can do the same with the top ruler to find the vertical middle of your photo. Only this time the co-ordinates at the bottom left of the GIMP interface should list the second number (y) as 0, and the first number should be as large as you can make it by moving your cursor (to the upper right of the photo).Once you have your width again half of that will be the middle of your photo.

Then using construction lines put one at the top of your object, and the bottom of your object. Find the "height" of your object by the distance between them. Remove the construction lines from the top and the bottom, and place a new construction line above the horizontal center line by the half of the "height"(of your object). Now place a construction line on both sides of your object and find the “width” (distance between the new vertical lines), then remove those construction lines and place a new construction line vertically half of the "width"(of your object) to the right of the vertical center line. Now cut the object out, and drag it so that the point you used as the "top" is on the horizontal construction line that is above the middle. Then Drag the photo left or right until the right edge of the object is on the vertical construction line you put in right of the middle construction line.

Now the center of your object is at the center of your photo. This is a very important thing because when blender loads in the picture you will need this so that all of your pictures match up with each other 3d. You should repeat these steps with all 3 photos. I also dont recommend doing it in GIMP's "layered mode" as that caused more pandemonium for me. I recommend opening each photo in a new window .

Taking your pictures is the most important part, because if the pictures are not all in the same scale (object size to photo size) then your photos will not line up and you won't be able to place a dot on the same location from front view, side view, and top view.

As a recommendation I would recommend making your first model from a Lego man. That is what I did and it is very simplistic easy practice. To take my photos I took about 10 minutes to construct a photo platform for my object. It consisted of a cardboard box with two sides cut out. I covered the inside area with computer paper. I then used a 2”x4” and a ruler to make sure that the box stayed the same distance from the camera for all shots, as well as marking where the Lego man’s feet were positioned inside the box with a pencil. This will provide good pictures, providing you keep the camera at the same distance and zoom for all three photos.

Next Page: Modeling a Gingerbread Man
Previous Page: Mountains Out Of Molehills

This tutorial is incredibly vague. I can't follow it. Revise?

Modeling a Gingerbread Man

Next Page: Penguins from spheres
Previous Page: Turning a Cube into a Puppy

In this tutorial you will learn how to make a simple gingerbread man. In a later tutorial you will be able to make an animation with this gingerbread man.

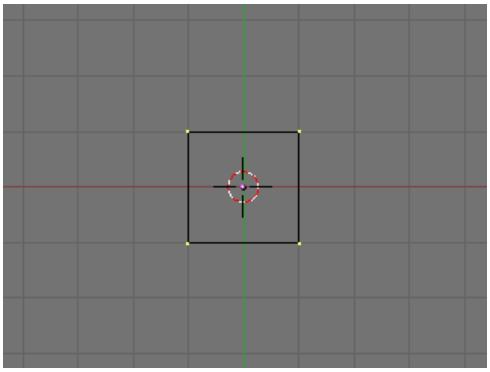
In this tutorial we will tie together everything we've talked about up to this point, including extruding, subdividing and rendering, and throw in basic lighting.

Modelling

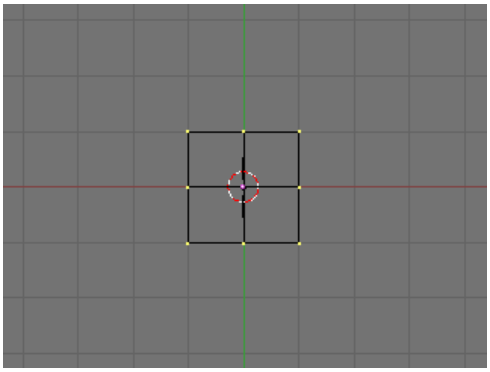
First, start Blender and zoom in until you can see everything (scroll with the **MMB** or press **CTRL+MMB**). Make sure you are in orthographic mode : press **NUM5** to go into orthographic mode.

- Select the cube by clicking **RMB** on it. To review, when an object is a whitish color, it is selected.
- Now press **TAB**. When you press **TAB** it will switch you between Object Mode and Edit Mode. If you pressed **TAB** you will see pinkish dots. The pink dots are called vertices. (You will know if you are in Edit Mode if you can see those dots.) When you select vertices with the **RMB**, they will turn yellow.
- Select all the vertices (**AKEY** once or twice) and then click on the editing tab  in the

header of the buttons window (or you can just press **F9**) to go to editing.

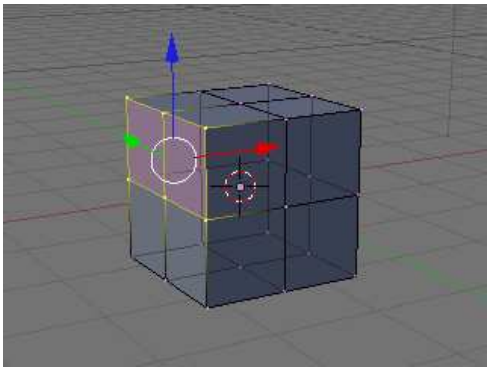


- Once you are there you will see a new menu at the bottom of the page, click on the subdivide button in the section called Mesh Tools (while all the vertices are selected). You will see that your cube now has more vertices. This tool is used for **dividing** an object so that you can do more complex models. [Nota] *In newer versions, you can also hit* **SPACE** *and, in the menu that comes up,* **Edit** → **Edges** → **Subdivide**.

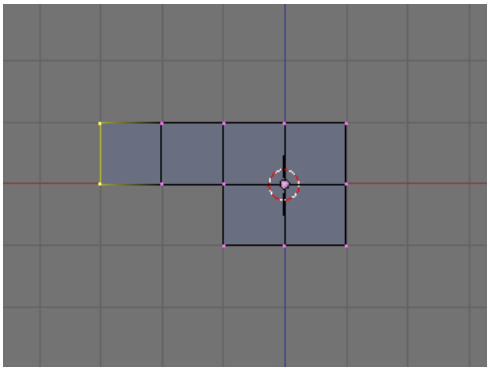


- Now press **AKEY** to unselect all the vertices, go to the front view (**NUM1**) and press **BKEY** and drag a square around the top left and middle left vertices or press **BKEY** twice and you will see a circle around your mouse - all the vertices in the circle will be selected by pushing **LMB**.

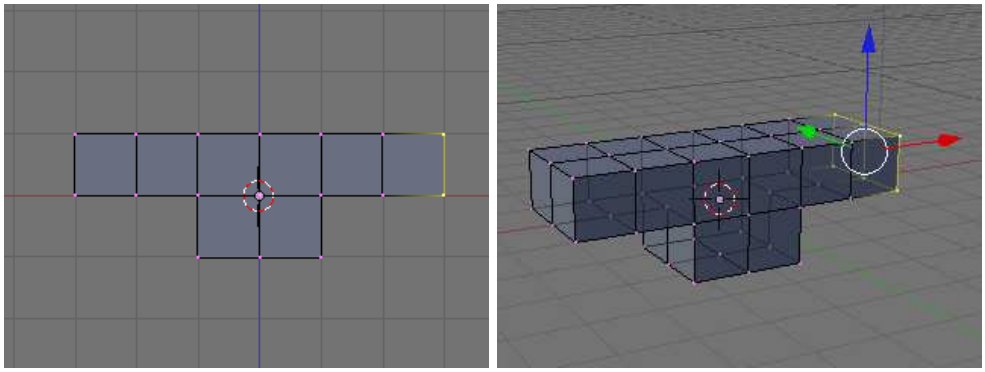
- Take a closer look on the selected vertices by viewing the model from a different angle (remember that you can use **MMB** to achieve this). If you find that you have only selected two vertices and not six, there are 2 ways of solving your problem. You could hit the **ZKEY** to toggle between wireframe mode and solid mode or you could hit (and deactivate) the Button in the selection mode buttons (*note that this button is shown only if you're in solid mode*). Repeat the previous step and see the difference.



- After selecting the 6 vertices press **EKEY** and select Region. This will extrude the selected vertices. Put the new vertices on the adjacent gray line of the grid one unit to the left (press **CTRL** to snap to grid). Do this two times so that it looks like below (the snapshot has been taken in a front view (**NUM1**)) :

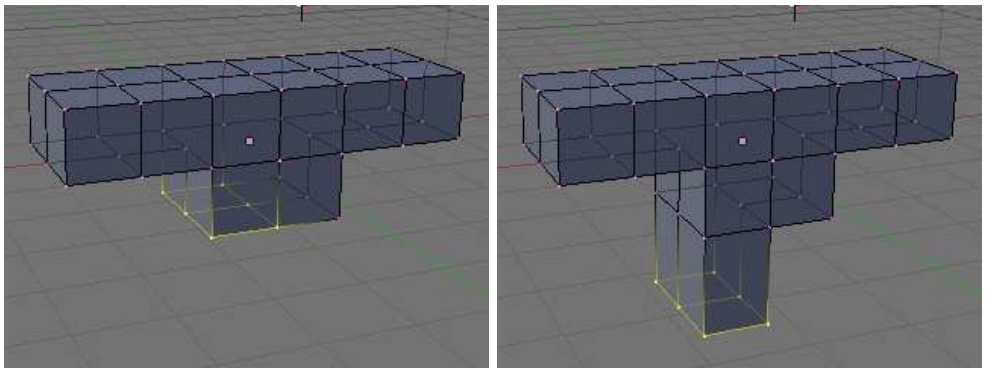


- Clear your selection (**AKEY**).
- Now select the other two vertices (six in 3D again) on the opposite side and do the same there as explained above. Now the arms are complete, as you can see in the illustration below.

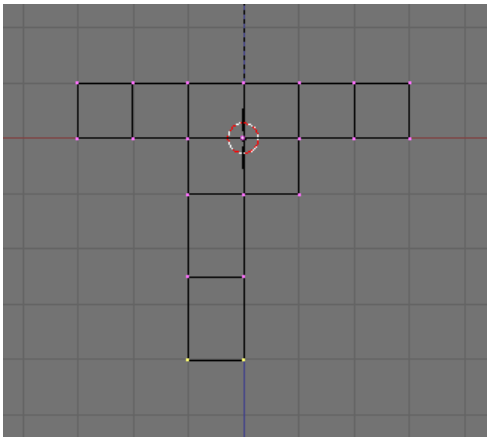


Now we will do the legs.

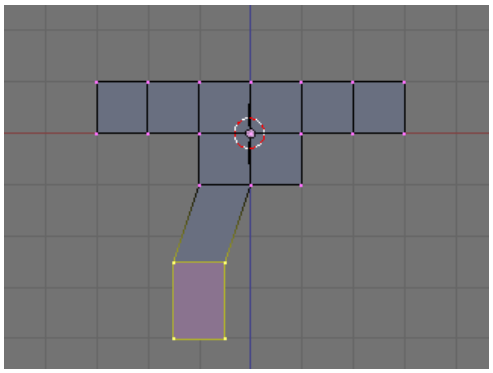
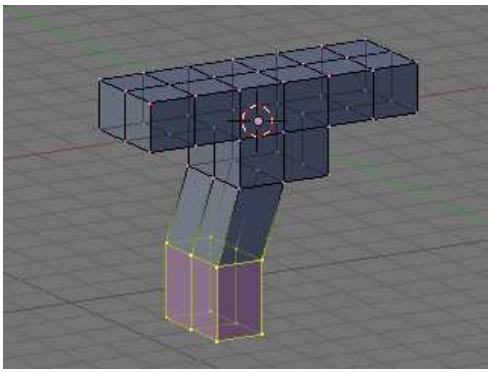
- First, unselect all the vertices : use **AKEY**.
- Select the bottom left two vertices, extrude it and put them in between the gray line (the gray lines in the grid representing the Blender units) and the second gray line below. If holding down **CTRL** you will notice that the two vertices snap to the grid in the background and you won't be able to select in between them, but jump between one and two of them. Press **SHIFT** as well and you'll be able to go in tenths of the units. (You can also just enter the number 1.5 to extrude it 1 1/2 units out.)



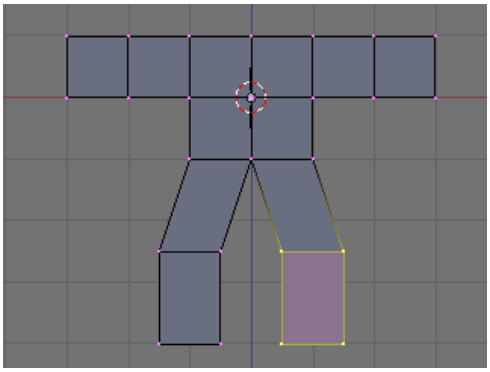
- Extrude it again and put it on the third gray line (or, once again, enter 1.5). It should now look like this:



- Use the **BKEY** to select the bottom 4 vertices (12 in 3D) of the leg, and use the **GKEY** to pull it out to the left by half a square so that it looks like this

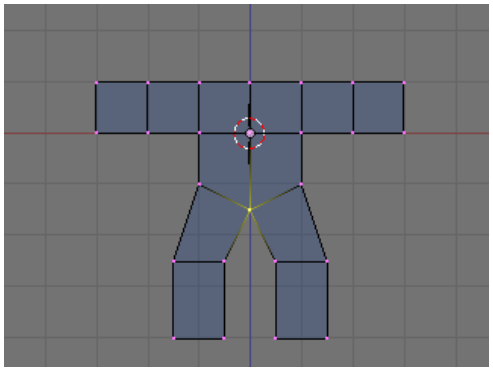
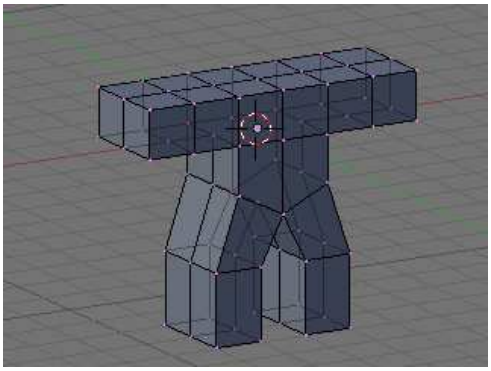



- Do this again for the right leg.

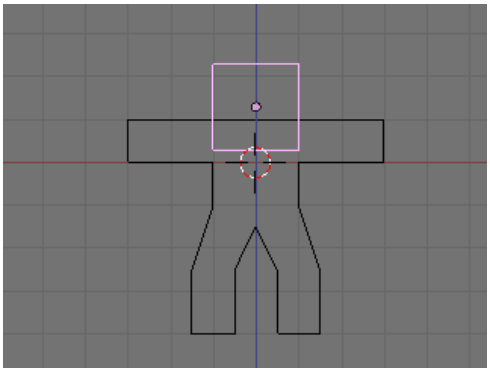


- Use **BKEY** to select the vertices at the groin (where the two legs join)
- Press **GKEY** and pull it down by 1/2 a square (type **GKEY**, **ZKEY** and write -0.5 - in older version you have to type type **GKEY**, **ZKEY** but also **NKEY** and write -0.5 then)

*(I had some problems here, trying to move the vertices. There were too many vertices in the same place, and that creates strange forms. To erase the duplicate vertices on top of each other, you can either select the entire model, or just the vertices you want to clean. Then press **WKEY** and choose Remove Doubles.)*

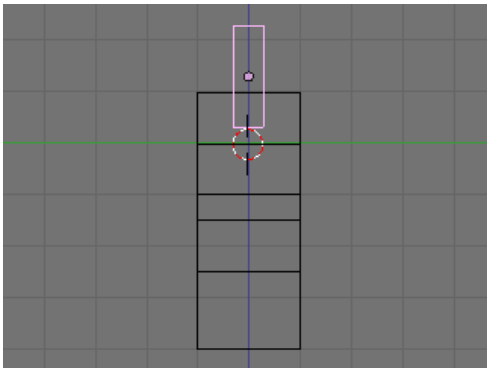



- Press **TAB** to go out of edit mode (you will know if you are out of edit mode if the vertices are not visible). You are now in object mode.
- Click **RMB** on the object to select it then press **SHIFT + SKEY** and select *Cursor → Selection*. This will make sure the cube you'll add next will be near where you want it.
- Press **SPACE** and put your mouse on the mesh option and select cube. In others versions, you can also hit **SPACE** and , in the menu that comes up, choose **Add → Mesh → Cube**.
- Press **GKEY** and put your new cube about 1/3 of the way down the neck (to achieve this, you can press **GKEY** and **ZKEY** : enter 1.33).

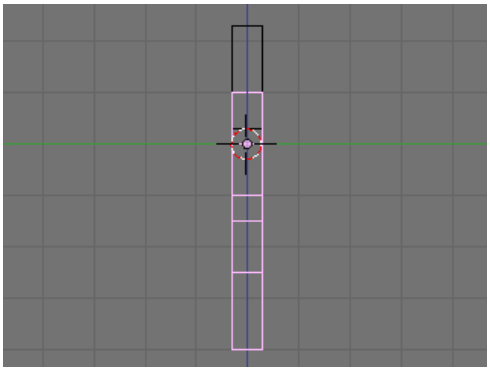


Now we will make it look more like a ginger bread man by making it thinner.

- Go to side view with **NUM3**.
- Press **SKEY** for scale and press **YKEY** for Y-axis and then move your mouse to the middle until the it is about 0.3 (use **CTRL** for fixed values).
- Remember X-axis is the Red arrow/line, Y-axis is the Green one, and Z-axis is Blue (like RGB video mode).



- Then use **RMB** on the body and press **SKEY** and then **YKEY** and make it as flat as the head.

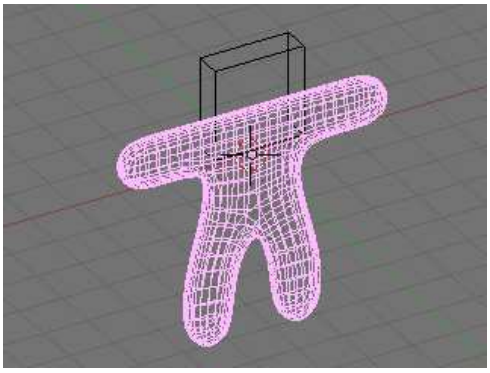


- Use the **MMB** to spin the view around and examine your handiwork.

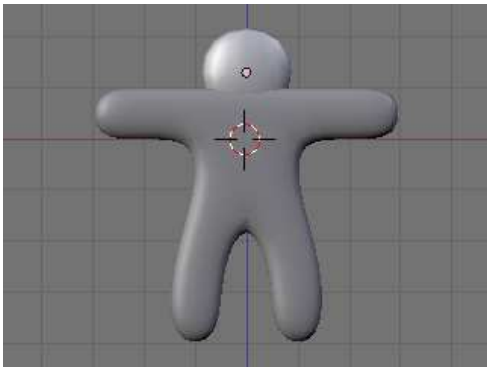
At this point, it doesn't look entirely like a gingerbread man, does it? It's a bit too ... chunky. For the last bit, we'll smooth it out.

- Make sure you've selected the body in object mode.
- Press **TAB** to switch to edit mode and press **AKEY** until you have selected all vertices.
- Select the editing panel in the buttons window (or hit **F9**).
- In the Modifiers tab, Add a "Subsurf" modifier.
- Set the level of the subdivisions to 2, and the number of render levels to 3.
- You can press the **ZKEY** to switch back and forth between wire-frame view and solid view.
- In the 'Link and materials' section, select 'Set Smooth'.

*(Note that here I had the same problem as before, with superposed vertices. press **WKEY** and select Remove Doubles to clean your model. You will see that it will look much better after removing the extra vertices with Remove Doubles)*



- Press **TAB** to return to object mode, and select the head.
- Press the **ZKEY** to return to wire-frame view.
- Now repeat the process above to smooth the head.



Looks a lot more like a gingerbread man, now, doesn't it?

Camera Positioning and Rendering

This guide will show you how to intuitively get the best shot of your 3D scene with no effort !

- Press **TAB** for Object view mode.
- Press **NUM0** to get the Camera View.
- Select the camera by clicking **RMB** on the outermost rectangle.
- Press **GKEY** and move your mouse to adjust the position of the camera (**XKEY**, **YKEY**, **ZKEY** and **CTRL** may be useful here).
- In addition, you can press **NUM7** to get the Top View and press **RKEY** to rotate the camera to the best angle.
- After you are happy with the position, press **F12** to render it.

Next Page: Penguins from spheres
Previous Page: Turning a Cube into a Puppy

This tutorial is incredibly vague. I can't follow it. Revise?

Die Another Way (dice modelling)

Next Page: Die Easy
Previous Page: Penguins from spheres

This tutorial needs some more explaining as to what is going on(and why) in section 4. Also needs updating to address Blender version 2.42. [colouring section in particular is not relevant to 2.42]

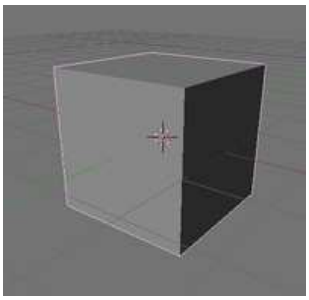
In the following tutorial you will be creating a dice. You will use:

- polygon mesh
- face loop cutting
- spin dup
- subdivision surfaces
- subdivision creases
- bevel
- set smooth

- multiple materials
- extrusion
- merge vertices
- remove doubles
- constraints

Step 1

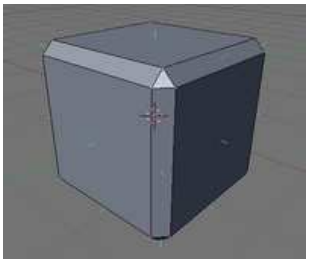
Press **NUM1**, **NUM7** or **NUM3** then **SPACE** → *Add* → *Mesh* → *Cube* to make a cube aligned to the axes.



> Note: You can, from here, directly divide the cube 2 times by clicking the *Subdivide* button in the *Mesh tools* twice. Then follow the instructions from Step 2 and you can go directly to Step 10.

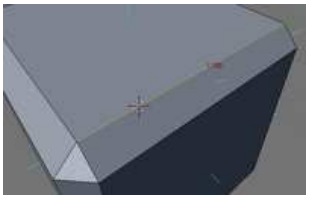
Step 2

Hit tab to go into edit-mode and select all faces to prevent bevel messing up normals. Hit **WKEY** → *Bevel*, *Recursion* → *1* (you'll see why later) then choose bevel size. Bevel of 0.15 is ok.



Step 3

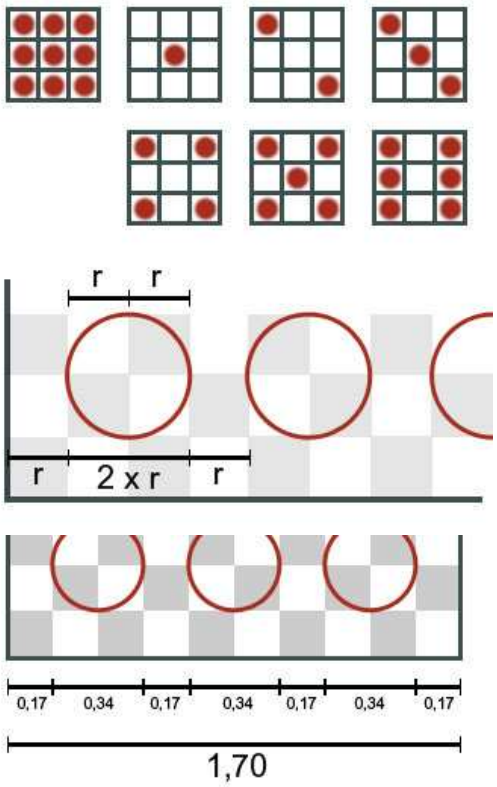
In editmode, go to the Editing tab (**F9**) and look at the mesh tools 1 panel. In Blender 2.37, there is a set of buttons for edge measuring. Turn on Edge Length and note the length of one of the sides of the square faces. This should be 1.7 if the above settings were used.



Step 4

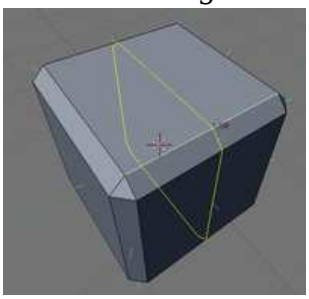
A typical die has a grid of 9 possible positions for the dents and the gap between the dents is

the dent radius (or half the diameter). So, there are conveniently 10 units on each edge of the square faces, where the gaps use 4 of the units and the 3 dents use two each. This means the gaps are of size 1.7/10 = 0.17 and the dents (1.7x2)/10 = 0.34.

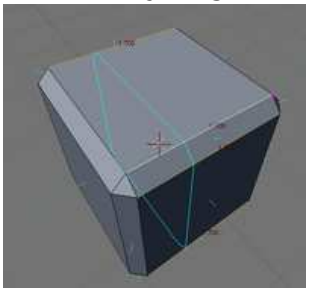


Step 5

To make these segments, use the face loop cut tool. Press **KKEY** →*face loop cut* and select one of the edges.



click and you get a blue line to position.



Unfortunately this tool (even in 2.42) seems only to input by percentages rather than actual values. Because it's very difficult to position using percentages, especially after the first segment is made because you then have to work out percentages of what's left, place the blue line to the far left edge and click once. Do not click again because this line is on top of

another one and it will be tricky to select it again. If you make a mistake, hit **UKEY** or **CTRL + ZKEY** to undo.

> Weasel note: in Blender 2.41 you can move the cut line roughly into position, then use **SHIFT** to precisely position at either 0.17 or 0.34 as appropriate.

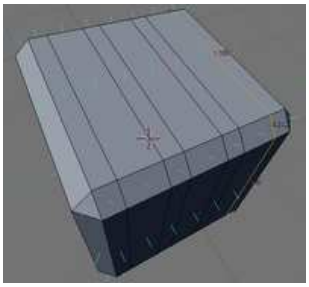
Now press **GKEY** then **YKEY** to move the line along the axis (this is why the cube needed to be aligned to the axes first). I'm using **YKEY** because of the way my cube is setup - the blue line moves along the green axis. If you've done it another way round, use the appropriate axis.

After hitting **GKEY**, **YKEY**, enter the exact value of the gap that we measured (0.17) and hit return - if you make a mistake before hitting return, hit delete and retype the number or press **UKEY** or **CTRL + ZKEY** after hitting return.

Select the remaining segment to the right of the line, cut this and move the blue line as far as you can, which will be to the line you just made. Move this line to 0.34 along y because it is where the dents go.

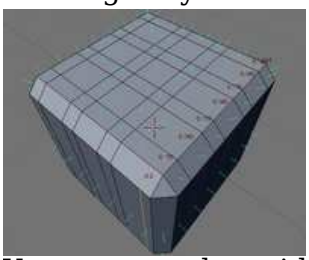
Step 6

Repeat step 5 with the remaining segment until you have made 6 cuts of alternating size (0.17, 0.34, 0.17, 0.34, 0.17, 0.34) and get something like this:

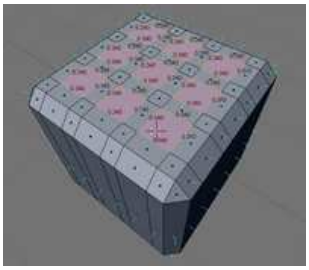


Step 7

Now turn the cube round and do step 5 for the adjacent edge until you get the grid forming. Note, you move the cuts along the x-axis now so do **GKEY**, **XKEY**, 0.17 etc. Don't worry if the last edge says 0.169 instead of 0.17.

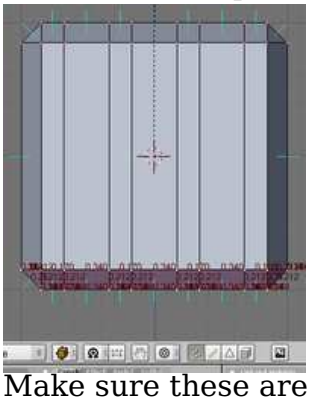


You can see the grid of 9 squares mentioned earlier:

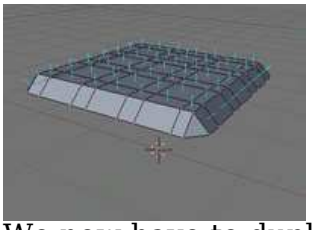


Step 8

The problem here is that it is very time consuming to do this for the remaining 5 sides of the die (Note: Actually, at this point there is already 4 remaining sides, an you need to repeat step 5 just once to finish them). Instead, we will just duplicate the one we have done already. [note: A time saving option is to complete Step 10 on one side of the face by merging the centre point of the 9 possible 'pip' locations before continuing from this point.] But first the rest needs to be deleted. So go into front view (**NUM1**) and orthographic mode (**NUM5**). Also turn off depth buffer clipping. Use vertex select mode and select the bottom vertices:



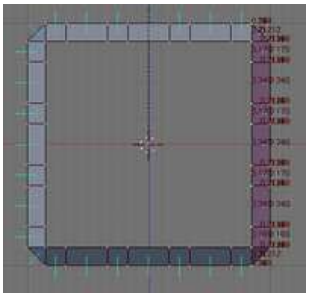
Make sure these are the right ones by rotating your view with the **MMB**. Now press **XKEY** and select 'vertices'. This should leave only the top side of the die:

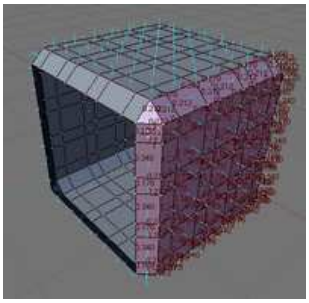


We now have to duplicate and rotate this side. There is a very handy tool to save doing this manually called Spin Dup located in the editing tab (**F9**) in the mesh tools panel. We want to duplicate the side 3 times in both the x and y axes to complete the cube - 3 times isn't necessary in the second axis because we will already have a bottom side but it's easier that way.

Spin Dup works relative to your camera and the cursor. So, put the camera into front view (**NUM1**) and orthographic mode (**NUM5**). If you haven't moved the cube, the 3D cursor should be at the center of the cube (at the pink dot). If not, go into object mode (**TAB**) and select what's left of the cube and press **SHIFT + SKEY** and select *cursor to selection*. You can also manually move the cursor by going to the View menu → *view properties* and editing the 3D cursor position. In this tutorial, it should be at (0,0,0).

The settings for Spin Dup are that we need 3 duplicates over a 270 degree rotation with 1 turn. Now press spin dup to get the following:

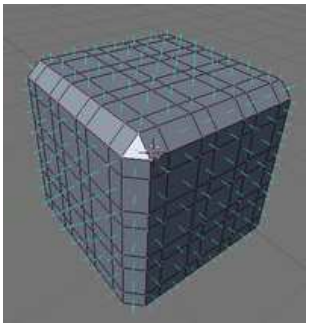




You will see the currently selected side is no longer the top side. This means we have to go into top view and duplicate around the z-axis so go into top view (**NUM7**) and hit spin dup again to fill the remaining sides.

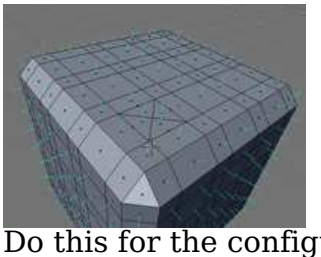
Step 9

There will now be a few overlapping points. To get rid of these, go to the editing tab again (**F9**) and in the mesh tools, there is a button called Rem Doubles. Select all the vertices by pressing **AKEY** and select remove doubles. This may leave some overlapping points because the threshold for removing doubles is set at 0.001 and some points might be just outside that value. Setting it to 0.003 should get rid of all the doubles to leave the die with 384 vertices (this information is at the top of the Blender window). You can check by counting that each side has 8x8 vertices and there are 6 sides on the cube.

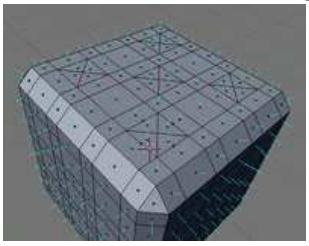


Step 10

The die needs the dents added. Select one of the faces where a dent would go and extrude the face by hitting **EKEY** and then **ESC**. Do not click after hitting **EKEY**. Collapse this face by using **ALT + MKEY** to merge the 4 corners of the extruded face into the centre where it will tell you you have removed 3 vertices. You should get the following:



Do this for the configuration of the dots on that side. So for example, 5 would look like this:

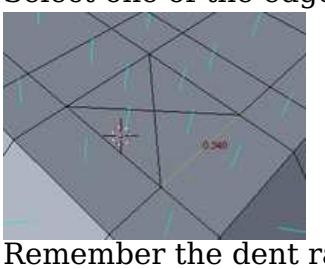


Notes :

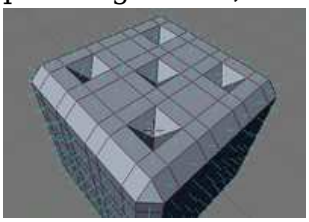
- It is also possible to directly use the **GKEY**, the **XKEY**, **YKEY** or **ZKEY** (depending on what xy or z direction you are working on) and typing 0.17 (perhaps add a minus '-' if it goes up instead of in the die). After SubSurf it will look nice.
- In previous note, if you grab the square surface, the dent is too large. If you grab the vertex at the center of the X, step 11 says exactly that.

Step 11

Select one of the edges of the dents to check the size is 0.34.



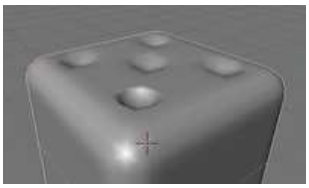
Remember the dent radius was 0.17. We need to use this value to lower the centre point of the dents down. Select all the 5 centre points at once to save time and move them inwards by 0.17. The side I put the 5 dents on here was the top so I move the vertices inwards by pressing **GKEY**, **ZKEY**, -0.17 and hitting **ENTER**. I then get this:



Step 12

TAB out of Edit Mode. If you haven't done this already, hit Set Smooth in the Editing panel and turn on subdivision surfaces

It should look something like this:

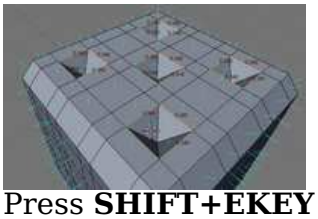


In 2.41, you will need to use "Add Modifier" on the Modifiers tab, to add a SubSurf modifier.

Step 13

On a die, the edges of the dents are usually sharp so we'll use subsurface creasing to do that.

Go back into editmode and with the edge select mode on, select all the perimeters of the dents like so (it may help to turn off subsurf for the moment):



Press **SHIFT+EKEY** to enable creasing and move the mouse until the display says crease is at 1. After pressing **SHIFT+EKEY**, you can then set crease values in the information box that you get by pressing **NKEY** when objects are selected. This can be useful to check if all the edges have the right crease because it gives you the average crease value and if it is less than 1, there is an edge wrong.

Step 14

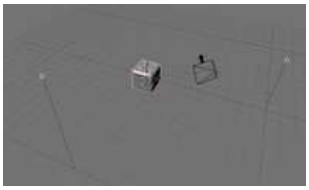
Repeat steps 10,11 and 13 for all the sides of the die. REMEMBER, a die is numbered so that opposite sides add up to 7. In my example, that means I put 2 on the bottom etc. Once you finish, if you turn on subdiv level 2, you will get something like this:

Step 15

You can make a test render now to see that the dents look the right size and the bevel is right. So, turn the subdivision level for the rendering up to 3. To help position the camera so that you centre the die, you can make the camera look at the die by adding a track-to constraint to it. I prefer to track an empty though, because it is more flexible.

Make an empty by going into top down view (**NUM7**) and hitting **SPACE** → *Add* → *Empty*. It's always best to go into one of the set orthographic views so as to align new objects to the axes. If you add something misaligned, just go to the object menu then clear/apply > clear rotation (or **ALT+RKEY**). Because the empty was created at the origin, you might not be able to see it as it is inside the die. Hit **ZKEY** to enable wireframe mode and select the empty. Just move it outside the cube until we get the constraint set up.

To add a track-to constraint, select the camera first then **SHIFT+RMB** the empty and press **CTRL+TKEY** and choose "TrackTo Constraint" from the list. Move the empty back inside the die. You can edit constraints in the object tab (**F7**). Add a couple of lamps (both intensity 1) to get the scene like this or feel free to experiment with a more advanced lighting setup:



Step 16

To render, set the size of the image you want. 800x600 is a decent size so put these settings in the format panel in the Scene tab (**F10**). In the render panel, make sure 100% is selected. If it's 50%, the render will come out as 400x300. For preview renders, don't turn on OSA, which is anti-aliasing because it slows your renders down significantly. Try to only use it for a final render.

Another way to position the camera is by selecting it and then looking through it as you move it. Look through the camera by pressing **NUM0**. Use the **GKEY** to pan across and rotate around the local axes of the camera by pressing say **RKEY**,**XKEY**,**XKEY** to rotate in X-axis. To zoom in and out press **GKEY**, **ZKEY**, **ZKEY** and then move your mouse forwards or backwards. The **mouse wheel** zoom moves your view towards and away from the camera, without actually moving the position of the camera.

Another important point is to set the image format. This is done in the format panel. The listbox has a number of image types. I find that png is generally the best because it is lossless and offers the highest compression among the lossless formats. It also supports an alpha channel for transparency. When rendering an animation, it is better to render as an image sequence than as a movie because it is easier to edit these and repair broken frames. Quicktime supports loading of image sequences and you can save as a movie using a wide range of compression formats.

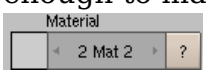
To save the render, go to the **file menu → save image** and type in the full name of the image including the extension e.g. die.png.

The output should now be looking something like this:

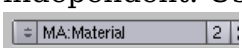
Step 17

To give it some colour, we will need to use multiple materials because a typical die has dents that are a different colour from the die itself.

Go to the editing section (**F9**) in the button panel again and make sure the die is selected. In the links and materials panel, there is a section for materials and the number in the box should be at zero. It may be at 1 if you have assigned a material to the object already. Add enough to make 2 materials in total.



Go back to the Shading panel (**F5**) and there is a box at the very top of the material panel with the number 2 beside it. Click this number to make the two materials you've just created independent. Use the arrows to the right of the ME button to switch materials.



Make material 1 bright red by just picking red in the colour picker or by setting the RGB sliders. Make material 2 white by doing the same. Or pick whatever colour you prefer and material settings.

Step 18

These colours need to be assigned to the right parts of the die.

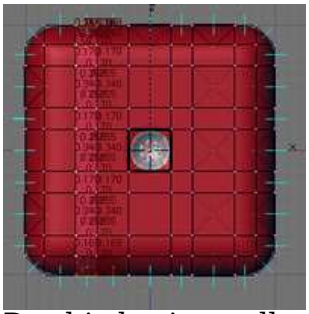
Go into editmode and turn off subsurf to help. Select the inner faces of the dents - if you accidentally don't press the **SHIFT**, press **CTRL+ZKEY** or **UKEY** to undo. Once they are all selected, go to the Editing tab again and the material panel. Select the colour for the dents. There is a button that says assign - press it and it will make the dents white but the die remains red.

Turn subsurf back on and render with OSA (only put it up as high as you need for the resolution of the image you are rendering):

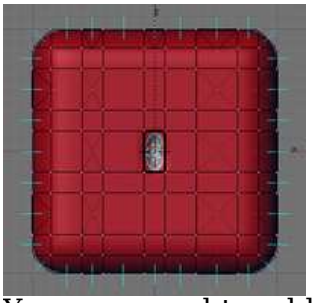
Extra

The reason I modelled the die this way is because it is also very easy to change the sizes of the components e.g. the bevel and the dent size. You do this by selecting the vertical or horizontal segments and just scaling them in one axis. Here we will reduce the dent size and the bevel by half.

Go into front view (**NUM1**), turn off clipping and select a line containing dents. Then just scale in one axis e.g. **SKEY**, **XKEY**, 0.5. Remember to have your pivot point set to median:



Do this horizontally and vertically around the die. You should need to scale 9 times for the dents and 6 times for the bevel:



You may need to add extra geometry once you are satisfied with the sizes of the dots and the bevel so that the edges of the die don't look warped due to the subdivision. You can use face loop cut again for that and add extra lines in the middle of the gap segments.

Next Page: Noob to Pro/Die Easy
Previous Page: Penguins from spheres

I found that there was some screwed edges of my die this way i found this was on section 12 however the fault may be further up than that

Edit Mode HotKeys Review

3D View by Mode: All HotKeys | Object Mode | Edit Mode | Pose Mode
Vertex Paint Mode | Texture Paint Mode | UV Face Select Mode
Other Windows: Scripts Window | File Browser | Image Browser | Buttons Window
Outliner | User Preferences | Text Editor | Audio Window
Timeline | Video Sequence Editor | UV Image Editor | NLA Editor
Action Editor | IPO Curve Editor
by Key: . , A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
*Relevant to **Blender 2.37a***

Edit Mode HotKeys

The Period Key

- **.KEY (on the number pad)** - centers the view around the current selection or active object.
- **.KEY (on the alphanumeric pad)** - changes the pivot point to the 3D cursor. The pivot point is the point where all things meet when scaled to 0, and the point of 0 translation during a rotation transformation. See the menu on the 3D view header, located immediately to the right of the **Viewport Shading menu**.

The Comma Key

- **,KEY** - changes the pivot point to the bounding box center

A

- **AKEY** - Toggles between selecting all or selecting none.
- **SHIFT+AKEY** - brings up the toolbox.

B

- **BKEY** - Activates box-select tool. Pressing it twice activates the Circle Border Select tool (2.41).

C

- **CKEY** - Centers the 3D View where the 3D cursor currently is.

D

- **DKEY** - Brings up a Draw Type menu.

E

- **EKEY** - Extrude selection

F

- **FKEY** - if two vertices are selected, create an edge connecting the two vertices. If three or four vertices are selected, or two edges are selected, create a face connecting the vertices or edges. If two co-planar faces are selected, join the faces to create an FGon, or dismantle a previously created FGon.

G

- **GKEY** - "Grabs" the current selection and allows you to move it around with the mouse. Use **LMB**, **ENTER**, or **SPACE** to drop it in place. Use **RMB** or **ESC** to cancel the move.
- **GKEY XKEY** - Grabs the selection and locks it's Z and Y position. In this mode it will only move along the global X axis.
- **GKEY XKEY XKEY** - Grabs the selection and locks it's Z and Y position on the local axis. In this mode the selection will only move along the local X axis.
- **GKEY YKEY** - Grabs the selection and locks it's Z and X position. In this mode it will only move along the global Y axis.
- **GKEY YKEY YKEY** - Grabs the selection and locks it's Z and X position on the local axis. In this mode the selection will only move along the local Y axis.
- **GKEY ZKEY** - Grabs the selection and locks it's X and Y position. In this mode it will only move along the global Z axis.
- **GKEY ZKEY ZKEY** - Grabs the selection and locks it's X and Y position on the local axis. In this mode the selection will only move along the local Z axis.

H

- **HKEY** - Hides the currently selected vertices, edges and faces. They will be hidden only while in Edit Mode.
- **ALT-HKEY** - Unhides vertices, edges, and faces that were previously hidden. Vertices, edges, and faces that are unhidden will be added to the current selection.

I

- **IKEY** - inserts a "key". Keys are used for animation.

J

- **ALT+JKEY** - converts triangular faces to quads.

K

- **SHIFT+KKEY** - knife tool. this doesn't seem to work with nurbs.

L

M

- **ALT+MKEY** - Merge selected points.

N

- **NKEY** - brings up a Transform Properties mini window.

O

P

Q

- **QKEY** - prompts if you would like to quit the Blender.

R

- **RKEY** - allows rotation of the selection. Move the mouse after pressing **RKEY** to rotate it. Press **LMB**, **SPACE**, or **ENTER** to confirm the rotation. Press **ESC** or **RMB** to cancel the rotation.

S

- **SKEY** - begins scaling (resizing) of the selection. Move the mouse to scale larger or smaller. Press **LMB**, **ENTER**, or **SPACE** to confirm the scaling. Press **RMB** or **ESC** to cancel the scaling.

T

U

- **UKEY** - undo last edit
- **SHIFT+UKEY** - redo previously undone edit

V

W

- **WKEY** - Boolean Tools menu *[in 2.41: Specials menu]*

X

- **XKEY** - delete the selection.

Y

Z

- **ZKEY** - Toggles between drawing the scene in wireframe and solid mode.
- **CTRL+ZKEY** - undo last edit
- **SHIFT+CTRL+ZKEY** - redo previously undone edit

TAB

- **TAB** - toggles in and out of Edit Mode of the selected, active object.

F1

F2

F3

F4

F5

F6

F7

F8

F9

F10

F11

- **F11** - Shows/hides the window with the last render.

F12

- **F12** - begins a single frame render based on the Scene settings in the Buttons Window.

LMB

- **LMB** - places 3D cursor where you click
- **CTRL+LMB** - places a new vertex at the place clicked. The new vertex will be joined to any previously selected vertices by an edge.

RMB

- **RMB** - selects vertex, edge or face, depending on select mode

Object Mode HotKeys Review

3D View by Mode: All HotKeys | Object Mode | Edit Mode | Pose Mode
Vertex Paint Mode | Texture Paint Mode | UV Face Select Mode
Other Windows: Scripts Window | File Browser | Image Browser | Buttons Window
Outliner | User Preferences | Text Editor | Audio Window
Timeline | Video Sequence Editor | UV Image Editor | NLA Editor
Action Editor | IPO Curve Editor
by Key: . , A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
*Relevant to **Blender 2.37a***

Object Mode HotKeys

The Period Key

- **.KEY (on the number pad)** - centers the view around the current selection or active object.
- **.KEY (on the alphanumeric pad)** - changes the pivot point to the 3D cursor. The pivot point is the point where all things meet when scaled to 0, and the point of 0 translation during a rotation transformation. See the menu on the 3D view header, located immediately to the right of the **Viewport Shading menu**.

The Comma Key

- **,KEY** - changes the pivot point to the bounding box center

A

- **AKEY** - Toggles between selecting all or selecting none.
- **SHIFT+AKEY** - brings up the toolbox.
- **CTRL+AKEY** - prompts to "Apply Changes." Size and rotation changes to the model object become permanent.
- **CTRL-SHIFT-AKEY** - prompts to convert dupliverted objects to real objects.

B

- **BKEY** - Activates box-select tool.

C

- **CKEY** - Centers the 3D View where the 3D cursor currently is.

D

- **DKEY** - Brings up a Draw Type menu.

E

F

- **FKEY** - In the 3D View, switches to UV Face Select Mode. Pressing FKEY again will bring you back to Object Mode.

G

- **GKEY** - "Grabs" the current selection and allows you to move it around with the mouse. Use **LMB**, **ENTER**, or **SPACE** to drop it in place. Use **RMB** or **ESC** to cancel the move.
- **GKEY XKEY** - Grabs the selection and locks it's Z and Y position. In this mode it will only move along the global X axis.
- **GKEY XKEY XKEY** - Grabs the selection and locks it's Z and Y position on the local axis. In this mode the selection will only move along the local X axis.
- **GKEY YKEY** - Grabs the selection and locks it's Z and X position. In this mode it will only move along the global Y axis.
- **GKEY YKEY YKEY** - Grabs the selection and locks it's Z and X position on the local axis. In this mode the selection will only move along the local Y axis.
- **GKEY ZKEY** - Grabs the selection and locks it's X and Y position. In this mode it will only move along the global Z axis.
- **GKEY ZKEY ZKEY** - Grabs the selection and locks it's X and Y position on the local axis. In this mode the selection will only move along the local Z axis.

H

I

- **IKEY** - inserts a "key". Keys are used for animation.

J

K

L

M

- **MKEY** - move selection to a different layer.

N

- **NKEY** - brings up a Transform Properties mini window.

O

P

- **PKEY** - starts the game engine.

Q

- **QKEY** - prompts if you would like to quit the Blender.

R

- **RKEY** - allows rotation of the selection. Move the mouse after pressing **RKEY** to rotate it. Press **LMB**, **SPACE**, or **ENTER** to confirm the rotation. Press **ESC** or **RMB** to cancel the rotation.

S

- **SKEY** - begins scaling (resizing) of the selection. Move the mouse to scale larger or smaller. Press **LMB**, **ENTER**, or **SPACE** to confirm the scaling. Press **RMB** or **ESC** to cancel the scaling.

T

- **TKEY** - brings up a Texture Space menu. Allows translation and scaling the Texture.

U

- **UKEY** - brings up Make Single User menu.
- **ALT+UKEY** - opens undo history menu.

V

- **VKEY** - enters Vertex Paint Mode. Pressing **VKEY** again will switch back to Object Mode.

W

- **WKEY** - Brings up Boolean menu. Choose Intersect, Union or Difference.

X

- **XKEY** - delete the selection.

Y

Z

- **ZKEY** - Toggles between drawing the scene in wireframe and solid mode.
- **CTRL+ZKEY** - the single best function of blender: UNDO!!! You can undo almost anything, and the program notifies you when doing a no-undo operation. Note: If Blender claims there are no more steps to undo, hit tab to switch to object mode and try again.

TAB

- **TAB** - toggles in and out of Edit Mode of the selected, active object.

F1

F2

F3

F4

F5

F6

F7

F8

F9

F10

F11

F12

- **F12** - begins a single frame render based on the Scene settings in the Buttons Window.

Curve and Path Modeling

Next Page: 2D Image (logo) to a 3D Model
Previous Page: Object Mode HotKeys Review

Frighteningly enough, we know what you're thinking. You're thinking that mesh modeling is cool and all, but it would be nice if Blender had a better way to create complex smooth 3D objects, right? Ok, so you weren't thinking that, but now you're curious about this better way. Good. Move on to the next page to learn more.

Next Page: 2D Image (logo) to a 3D Model
Previous Page: Object Mode HotKeys Review

2D Image (logo) to a 3D Model

Next Page: 2D Image (logo) to a 3D Model Part 2
Previous Page: Curve and Path Modeling

Using Bezier Curve to Model a 3D logo from a 2D logo

{Construction on hold, feel free to complete}


original 2d logo

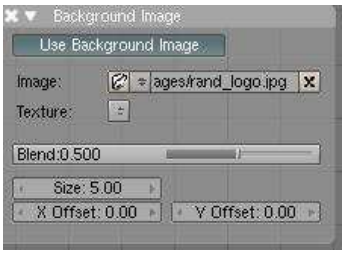
The image to the left is used in this tutorial. However, the tutorial is easier to follow using letter/numbers, or simple shapes/curves. Basically we will be using the graphic as a template for a 3d logo, tracing it, then discarding the 2d image.

Set up

You need a 2D logo similar to mine (preferably in JPEG format as Blender understands jpegs fairly well). If you haven't already done so, open blender and select one of the orthogonal view angles by pressing **NUM7**, **NUM3**, or **NUM1.**. At the bottom of the 3D viewport on the left, there are some menus, click **View-->Background Image**

A small window will appear containing just one button marked **use background image**; click this button. A few more buttons will appear. One of them says **image:** and has a small button with a picture of a folder on it; click this button. You are now presented with a file selection screen. Using the navigation techniques from the previous tutorials, find your 2D jpeg image on your computer, click the file in the list once then click the **Select Image** button at the top right of the screen.

Blender now displays this image in the background of the 3D view for you to trace its outline. The image is only displayed in orthogonal view. If perspective view is enabled, toggle to orthogonal view by pressing **NUM5**. The image will not be rendered as it is not part of your scene.



Once a background is selected you'll have a dialog like this one. (Note: I'm using Blender version 2.37 -older versions may differ.) The background dialog has 6 rows of buttons that function as follows. These are described from the top of the dialog and going down:

The top button **use background image** is a toggle button that turns display of the image on or off. Turning the button off will not clear the settings; it just hides the image. When you turn the button on again, your previous settings are back. Try it - click the button a few times.

Image selection is controlled on the 2nd row labeled **image**. There are 2 buttons, a text box, and a final button. The first button is for browsing for an image. The 2nd button is for selecting an image from a history list. (This will be empty for the first time. Selecting it now will display the image you currently have selected.) The text box allows typing in the file directly. The last button on the row, removes the current background image.

The third row is called Texture and will not be used for this tutorial.

The fourth line, labeled **blend** controls the transparency of the background image with a slider. A setting of 0 is completely solid and 1 is completely transparent. You can adjust it by clicking left or right of the knob for gradual changes, clicking and dragging on the slider for rough settings or clicking directly on the **blend** text for numeric entry.

The use of the blend function will become obvious once we start tracing our logo. For now, play around with it, see how it changes the image, and put it back to the 0.500 default.

The fifth line, **size**, controls the size of the image. This size setting is independent of the zoom for the 3D view window. To see how the size works move the default cube off to the side so that you can see both the cube, the background dialog and the background image. Now watch both the cube and image as you change the size. Notice how the image changes

size but the cube doesn't? Now press **NUM+** and **NUM-** to change the view's zoom. Now both the cube and image change size.

The final row controls the X and Y offset for the image. These controls move the image up and down (Y) or left and right (X). These settings can be useful if you need to reposition the image from the default position. Like the size, these offset values are independent of the view. As you change the offset values the cube you added earlier won't move. Now scroll the view using by clicking and dragging the **SHIFT MMB** and notice how the cube and image move together?

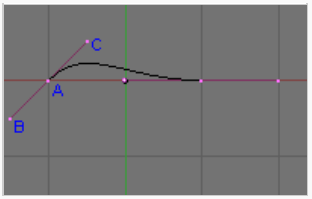
Once you start tracing the image you won't be using the size or offset setting. Delete the cube (select it, press **XKEY** and select **All** from the **Erase** menu), and set the size so that the entire image is viewable. Then set both the X and Y offsets to 0. Finally minimize the Background Image dialog. You'll only need it to adjust the blend setting until you finish tracing.

Introducing the Bezier Curve

The Bezier Curve (http://en.wikipedia.org/wiki/Bezier_curve) allows drawing graceful, complex curves and only requires a few control points. Specifically, it only requires 4 points for a curve. Two end points and two control points.

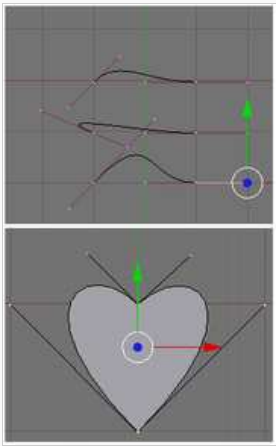
For the moment set the blend to 1 on the Background Image dialog. With the center of the 3D view still selected, press **SPACE -> Add -> Curve -> Bezier Curve**. Alternatively you can use the **Add** menu at the top of the screen or press **SHIFT -> AKEY** to jump directly to the add menu. You should now have something like this:

Unlike the traditional Bezier Curve each Bezier vertex has 3 points. I've labeled the 3 points for the left vector: A, B and C.



The Curve you added.

Point A is an end point. The curve will always go through this point. **Points B and C** are control points. These points influence the path of the curve as it leaves **Point A**. Because the path stops at **A**, **Point B** has no real effect on the path. Instead **B** is currently locked with **C**. (If you move either **B** or **C**, the other will move.) We will *fix* **Point B** to move independently a little later.


How the controls affect the path

The control points have 2 effects on the path exiting the end point: direction and distance (these are termed slope and magnitude in math circles) from **Point A**. The direction will provide the direction that the path will follow when it leaves **A** and the distance will determine how long the path follows that direction before it starts making its way to the next of the curve.

The example to the left shows how the control points influence the path of the curve. In the top picture, we see three curves. The top curve is the default curve. In the next curve down, **C** has been moved to give a drastically different direction. Notice how the path leaving **A** moves away from the other end point. The third curve, the distance was changed dramatically. Watch the path move much higher than the other two curves.

Image:Beziers (with *C* selected) Press EKEY to *extend* the Bezier, and drag it out; you will find it goes where you pointed *C*! *--Jawboot 01:30, 30 April 2006 (UTC)*

Image:Beziers Take *C* point and move it in the exact direction you want to turn

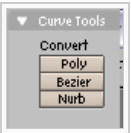
Image:Beziers N.B. To easily make sharp turns; take *B* point and move it really close to *A*.

In the bottom example, I've built a heart shape using just the points shown. *Dragging the bottom end point down will make the shape closer to a leaf.* You'll be able to do the same at the end of this tutorial. Go ahead move around the points for the curve and see how they all interact. Get a good feel working with the curve and when you're ready we'll move on to tracing.

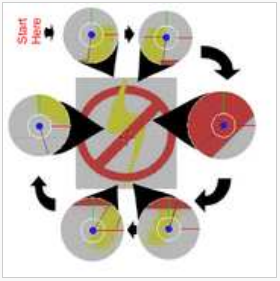
Now that you know how to work with a bezier curve set blend back to 0.5 on the Background Image dialog so we can start tracing.

Rough Tracing

The first step in tracing is to click the Polygon convert button on the curve tools panel. You'll find this in button on the **Buttons Window**. You may need to select the **Edit Panel**. Press **F9** if this panel isn't visible. If you don't already have a curve add one now. It will help to move the curve to the center of the yellow lightening bolt if you must add a new one.



Curve Tools for Blender 3D version 2.37



Move Vertices Here

Next, move the vertices of the curve to the points shown in the image to the left. This is called Rough Tracing because you don't need to exactly trace the image. You only need to approximate the image. Moving the vertices should be done using the instructions from the Creating a Simple Hat tutorial.

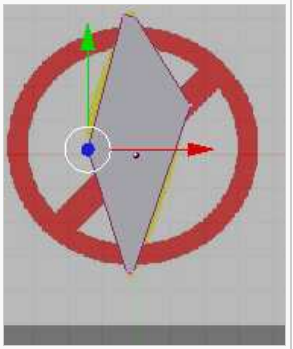
Note: Selecting the best place to put a vertex is a bit of an art that you'll acquire as you work with curves. For now follow the arrows along the cutouts and place each of the vertices as shown.

This tracing uses all the vertices of the polygon. Other cases, you'll need add or remove extra vertices. Adding and removing vertices as shown in Turning a Cube into a Puppy tutorial (Note: To add a vertex select the end point of your curve press **CTRL** and click **LMB**. At the place you clicked a new vertex will appear connected to your curve.).

After moving the last vertex, we finish the rough tracing by pressing the **CKEY** to close the polygon. You should see an image similar to the one on the right. (If you only have an outline switch your view port shading to solid by pressing the **ZKEY** for now.) Notice how the polygon doesn't cover all of the yellow of the bolt and how in some places the polygon fails to conform to the shape of the bolt. This is expected and should not be a cause for concern. We correct this in the next section.

Once you've finished several logos you should begin to get a feel for the required placement of vertices. Until then, here are some general guidelines to keep in mind:

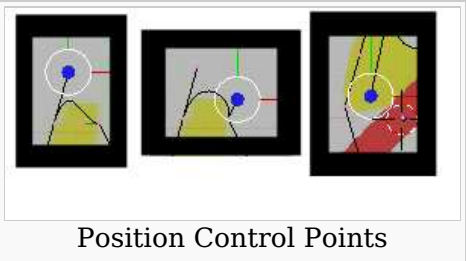
- A gradual curve may only require a single vertex.
- Tight curves will likely require two closely placed vertices.
- Curves may not require a vertex at all - you can define some curves using the control points of the adjacent vertices. We did this for both of the inside curves of the bolt above.
- Corners require a single vertex placed where the curve bends. A square, for instance, requires four vertices - one at each corner - to be modeled properly.
- The end point of a curve will always be on the curve. So should all of the vertices you place.


Final Rough Traced Polygon

We are now ready to move onto the next step modeling the logo. Press **ZKEY** to return to wire frame mode and prepare for the next step.

Polishing the Tracing

First, press the **Bezier** convert button to convert the polygon back to a curve. This will convert your polygon back into a curve. Nothing obvious will happen. If you look close, you should notice the number of points on the curve tripled. When you converted the curve back to a Bezier curve, Blender changed all of the polygon vertices to Bezier vertices. While the polygon vertex is a single point the Bezier vertex is made of an end point and 2 control points. So the extra points are the control points of the Bezier vertices. These control points are placed along the curve to produce the same shape as the converted polygon.



Position Control Points

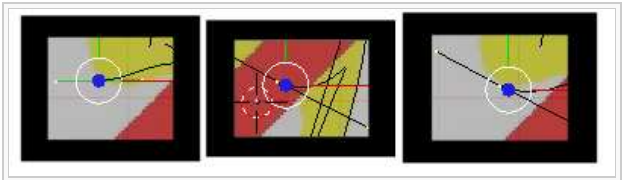
Our job is to move the control and end points so that the curve follows the edge of the bolt. The trick is to move the 2 control points between adjacent end points to bend the curve to the edge of bolt. First, move the right control point of the top-left vertex. This should pull the curve from its end point to more closely match the line of our bolt. After placing this point, we move to the next control point following a clockwise direction around the bolt. Use the **RMB** to select the point you want to change and move it with **GKEY** to place it.

As you move the second point notice how the curve exiting the first end point is drawn away from the edge of the bolt being traced. We now have to adjust the first control point again to get that line back on track. This quickly turns into a balancing act adjusting each set of control points. The trick is to make smaller movements for each iteration of adjustments. Make a game of it and move the control points all along the bolt. Always move along the clockwise direction. This practice is not just for consistency, it keeps your place and ensures that moving a control point doesn't change a portion of the curve that you've already completed. In time you learn to move the first point only part of the way. Then moving the second brings the curve for the first into correct alignment.

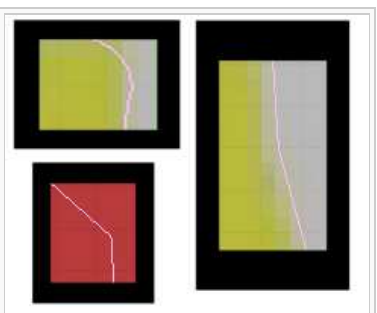
If you have some trouble aligning the curve to the edge of the bolt, consider adding a new point. There are two (at least) ways to accomplish this:

- Select 2 points that surround the problem spot where you want a new vertex and click the Subdivide button on the Curve Tools 1 tool panel.
- If near an end point, Select it, press the **CKEY** to open the curve, then **Control+LMB** click to add a new point beyond the end of the selected final vertex. Press the **CKEY** to reclose the curve.

The new end point should be positioned and then you have to adjust the curve on both sides of the end point you move. Any time you move an end point be sure that the curve going into both adjacent (clockwise and counter-clockwise) end points still aligns with the edge of the bolt.

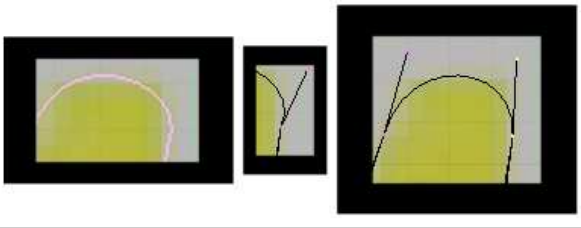


Adding a new end point

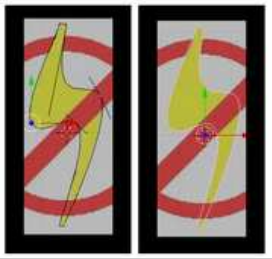


Finding trace defects

Once you've made the complete circuit around the bolt, you're ready for the final polishing of the edge of the curve. Press the **TAB** to switch to the object mode. This makes the polishing easier as Blender hides the points and lines for editing the curve. Now zoom in on the bolt's edge using the **NUM+** or **Control+LMB** drag. Use **Shift+MMB** drag the screen so that you closely observe the entire edge of the bolt while zoomed in closely. Look for places where the curve pulls away from the edge. Also look for sharp bends at each of the end points that should be smooth. You can see several defects that I found in my project after tracing the bolt. Switch back into edit mode to fix the curve and then go back into object mode to look for more defects.


Correcting poorly shaped curve

Sharp end points are adjusted by decreasing the angle between end point and the control points. For Blender specific case that I know of, sharp points have a tendency to show to the side of the end point. This typically requires adding a new vertex between the two end points to smooth out the curve. Places where the curve pulls away from the edge can be resolved by moving the control point closer to the edge. In the above image the curve was found to have been pulled away from the edge. This was fixed by moving the control point a little to the left.


Final Polished Curve

Here's the final polished curve for my project. It is shown in both edit mode and object mode so you can clearly see both the control and end points on the left and the curve to the right.

Note: If you have never worked with Bezier curves before, try it with a 2D paint program such as Inkscape or Paint Shop Pro. It might be quicker and easier to learn proper placement of control points in a program where drawing the curves is quick and simple.

*Helpful Tip: In blender 2.37 and later (not sure of earlier versions) pressing the **HKEY** toggles the control points between free and aligned (Edit Mode). Free Control Points are good for sharp angles, and aligned are good for smooth curves. This shortcut is in the **Space>Edit>Control Points** menu.*

This concludes the tracing of the bolt. All that remains is making the curve 3 dimensional, applying a material and positioning the final object. Before doing that, we will trace the circle in the next part of the tutorial. Save this project if you want to take a break before continuing. You'll need it on the next turorial.

Adding a Third Dimension

First, give the object some depth. Leave editmode, go to the editbuttons screen, and under the "Curve and Surface" tab, set the following values:

Ext1: 0.2 (the height of the extrusion on either side)

Ext2: 0.02 (the radius of the round bevel applied to the exruded edge)

BevResol: 4 (the number of subdivisions on the bevel curve)

(Note: In 2.4, Ext1 is Extrude and Ext2 is Bevel Depth.)

Also, if you have a simple logo go ahead and increase the DefResulU value to 25. If you have an extremely complicated image this is totally overboard but looks nice when you are just tracing text or numbers.

Now you can use your knowledge from earlier in this book to change the material and/or add texture to your logo. Feel free to rotate, add lighting, or whatever floats your boat. Don't forget to press **ZKEY** to toggle wireframe mode.

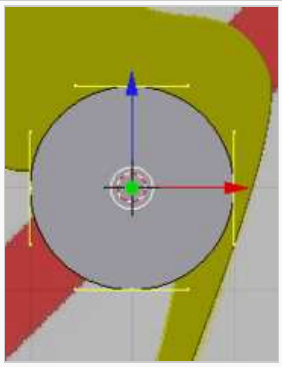
-=< Tutorial under Construction, ready soon, thanks for input spiderworm >=- A similar tutorial is available here (http://www.vrotvrot.com/xoom/tutorials/logoTut/logoTut.html)

Continue tutorial using bezier curves

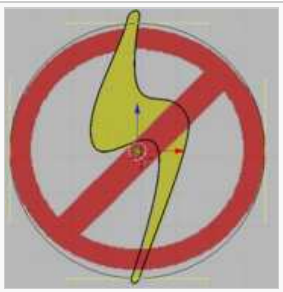
- This continues the tutorial, finishing up the sample logo from the image in the tutorial using bezier curves. These instructions make the assumption that you completed the first part in front view with **NUM1**. An alternative method for doing this using mesh circles has been presented below.

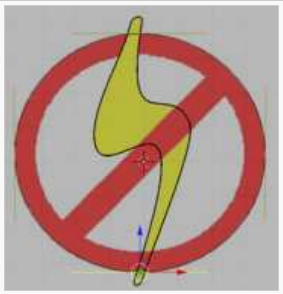
First, to make the lightning bolt distinct from the second part of the logo, it may help to apply a yellow material to it before getting started.

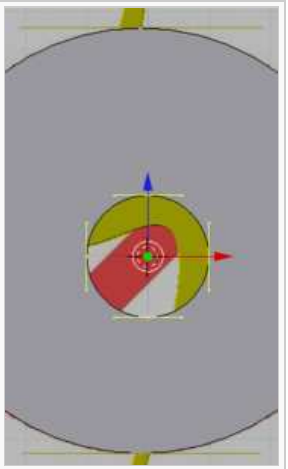
Adding the circle


Bezier Circle

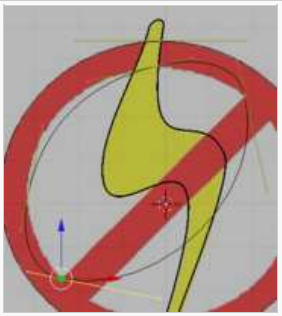
Switch to object mode by hitting **TAB** if you aren't already there. Press **space -> Add -> Curve -> Bezier Circle** to add a closed bezier curve with four points forming a circle. If you are in solid draw type, switch to wireframe with **ZKEY** so you can see the underlying image better. Hit **SKEY** to scale the bezier circle to fit over the circle in the image. You will probably find that the bezier circle is not dead center on the sample logo so you will need to move it with **GKEY** to center it. You may need to scale it and move it several times to get it right. You will also find that the circle in the sample image is actually a slight oval, so scale and position the bezier circle so that it touches the circle in the image on the left and right sides. Normally, you could then scale the circle and constrain it on the z axis by hitting **SKEY** then **ZKEY**, but it turns out that the oval isn't regular anyway, so just select the point on the top and hit GKEY then ZKEY to move it down until it touches the top of the oval in the image. Then do the same for the bottom point and you should have a pretty good fit.


Match up left and right sides

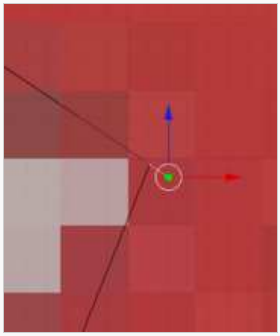

Move top and bottom points


Circle in circle

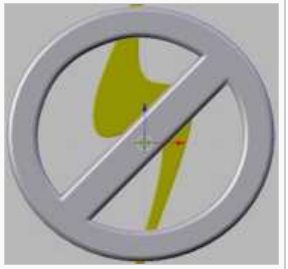
Just to understand what's happening in the next steps, switch to solid view with **ZKEY**. As you can see, you now have a circle, but it's filled in the middle where you want to be able to see through it. To cut a hole out of the circle, hit **space -> Add -> Bezier Circle** while you still have it selected in edit mode. A new circle will appear inside the larger circle. As you should be able to see in solid mode, the new circle actually cuts its shape out of the larger circle surrounding it. Switch to wireframe mode with **ZKEY** so that you can see the underlying image again. Scale up the smaller circle so that it approximately fits the inner part of the circle in the image. Don't worry about getting it exact since you'll be manually moving all four points anyway. Move the bottom point of the bezier circle to the top left corner of the bar that crosses the circle.


Place two points at either end of the bar

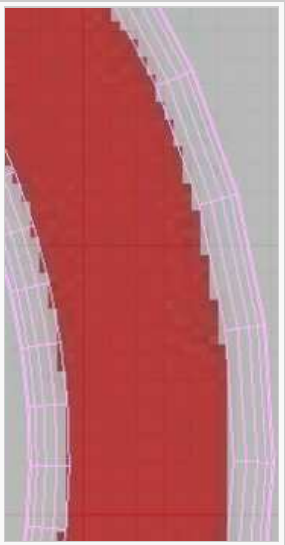
Move the right point of the circle to the other corner. To get a curve close to a 90 degree angle at those corners and an approximation of a straight line along the bar, you'll need to select the bottom control point for the point in the upper right corner and drag it very close to the point itself. You'll probably want to move the control point fairly close, then zoom the view in very close and adjust the control point further, otherwise it will be very difficult to control.


Place two points at either end of the bar

Then move the other two points and adjust their control points until you have a pretty good approximation of the rest of the inside curve. Next step is to press **space -> Add -> Bezier Circle** again and repeat the same steps, but for the lower opening in the logo. Once you've completed both openings, switch back to solid view with **ZKEY** and examine your work. Make any adjustments you need to swithing the draw type back and forth as needed.

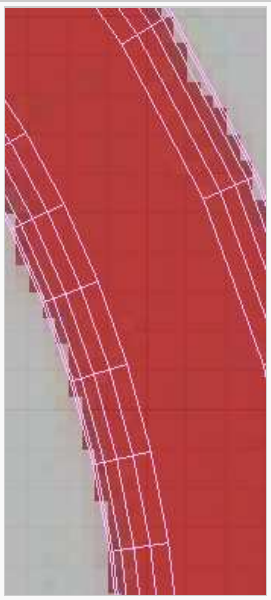

Wider bevel has a nice look

The next step is to make this part three dimensional like you did with the lightning bolt. Go to object mode with **TAB**, then select the editing buttons. Under curve and surface, set Extrude/Ext1 to 0.2, Bevel Depth/Ext2 to 0.02 and BevResol to 4. You can also set DefResolU to 25 as suggested for the lightning bolt. Looking at the results, the bevel effect may not be enough, so try increasing Bevel Depth/Ext2 to something like 0.15. That should look better, but there will be a problem. Switch back to wireframe mode with **ZKEY** and you'll see that the bevel has widened everything so that the circle no longer matches the original image. This can be fixed fairly easily by reducing the width parameter under curve and surface until it fits again.


Bevel has increased logo size

For final steps, select the lightning bolt again and switch into sideview with **NUM3** and hit **GKEY** and then **YKEY** to move the bolt backwards. Move the bolt back so it no longer intersects with the circle and bar. Apply a red material to the circle and bar portion. Finally, you can go to view, then to background image and hit the background image button to hid the image now that it is no longer needed. At this point, you can add any finishing touches for lighting and camera angles and render the logo.


Final render


Width adjustment restores to normal size

Johnos Addition (the tutorial on the next page does this too)

- This tutorial assumes that you were creating a logo from the one above, and that you are willing to listen to an idiot. :-)

I am new to Blender3D but I will try to finish this tutorial, and leave you with this:


The finished article

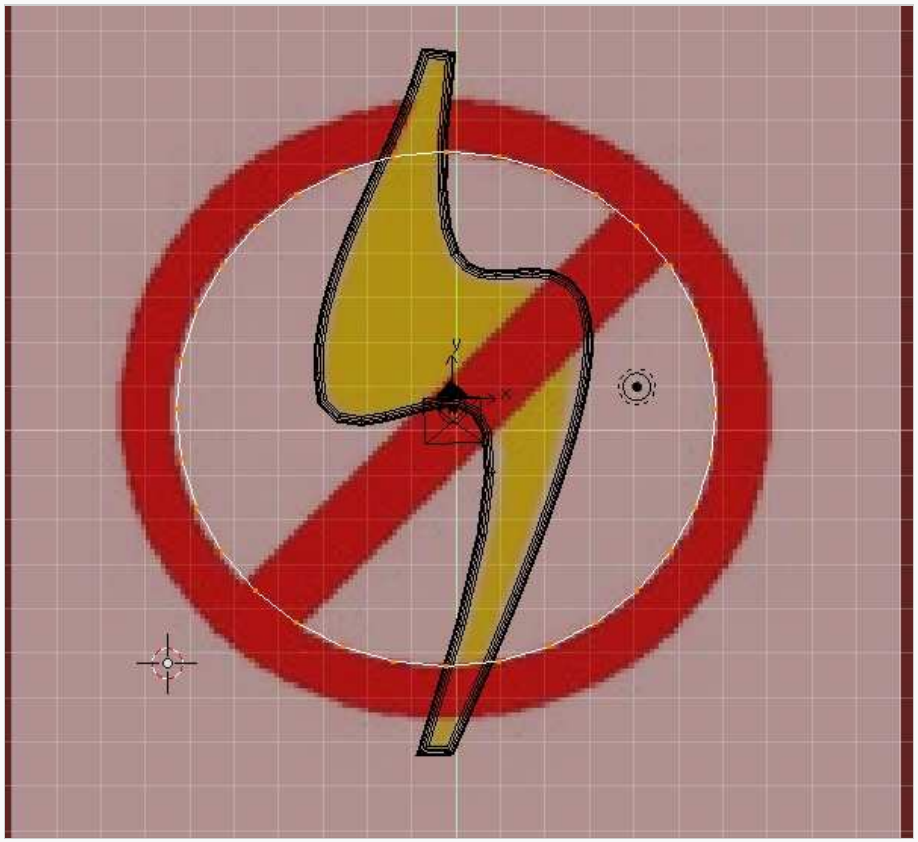
Adding The Circle

Ok, what you have so far is a lightening bolt, which is great. Its also nice and rounded which is even better. However, what we are missing at the moment is the outside circle. This is probably not the best way of doing it, but it is one way. Instead of using the Bezier Curve, I am using Circles.

Go to the top view (NUM7), and press SPACE -> Mesh -> Circle. Accept the 32 vertices, you can make it less but it wont look as good.

Move it into the center of the circle, if you don't then I advise you have wireframe on for the moment (press Z). Then press S, for scale, and make it the correct size for the inside of the

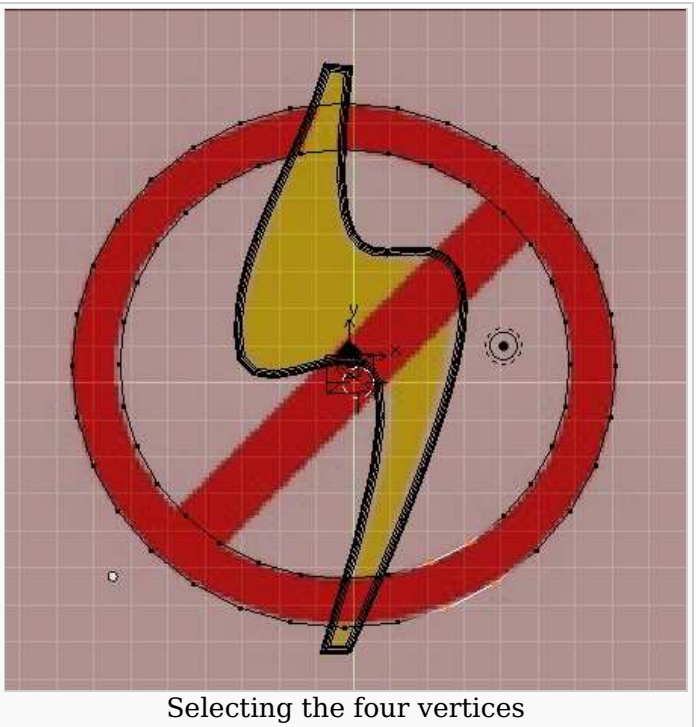
circle. Once you have that in the correct place, like so:



Adding the first circle

You may notice that I needed to stretch it sideways a little, you will too. Ok, deselect that, (press A, once or twice). Now make another circle, move it into the middle, and change the scale (S Key, remember) so that is reaches to the outside of the big red circle, bigger than the circle that you added before. Now here is where I am sure theres an easier way of doing it, but I don't know how so what I did to fill in the circle with faces was this.

If you dont read this carefully, then you may not get the wanted end result. Deselect the second circle, then select four vertices, that are in a square and will make up a face.

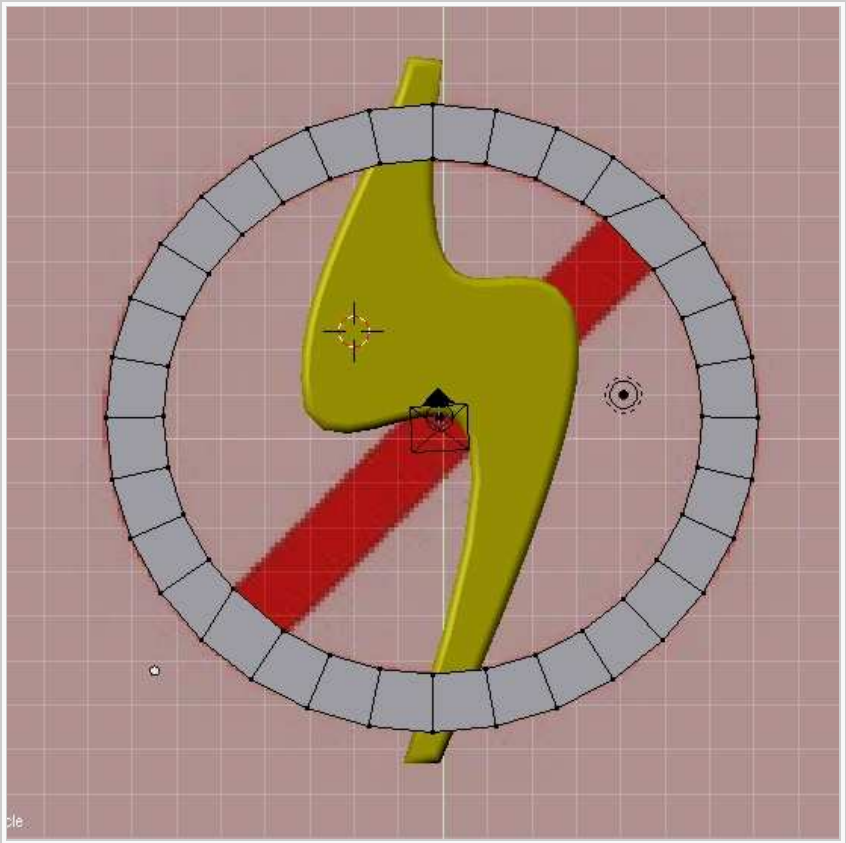


Selecting the four vertices

Once you have that, press F and a face will appear, I.E the box will be filled (turn off Wireframe, press z). Now do this right around the circle. To do this, hold down Shift and Ctrl, then draw a circle around the two vertices you wish to deselect with the Left Mouse Button. Then draw a circle around the next two vertices while holding down Ctrl and LMB, **NOT SHIFT**. Shift changes the control from selecting, to deselecting. This might take a while, sorry. Once you have got right around the circle its time to make the line through the middle... this is easy!

NOTE: A quicker way of making the circle object in this scene is to only create the small circle, select it, press 'E' to extrude (choose Only Edges), press 'S' to scale and another sized circle will appear, size this appropriately then click LMB. Then to create the cross, select 2 vertices at either side of the circle near where the cross overlaps, then press 'F' to create a face.

This way is a lot faster, thanks for your help :-) To make the crossing line however, you need to move 4 of the vertices slightly.. example below:



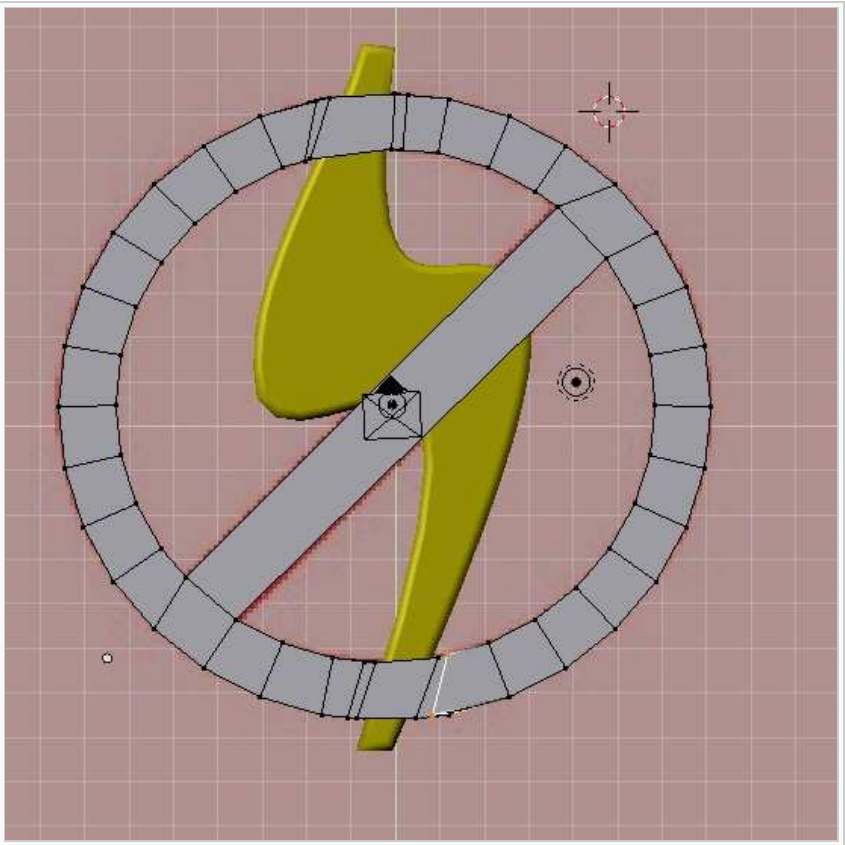
Moving the four vertices

Once you do this, highlight the 4 you moved, then press F.

Making the circle 3D

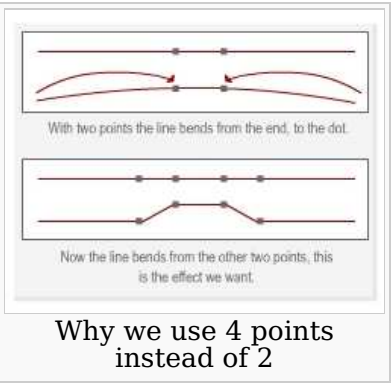
Highlight the full circle by pressing A either once or twice. Go to Side View, and press E for extrude and drag it down so that it is the same thickness as the lightning bolt (you'll see why). Now this bit is purely for arts sake, you won't probably learn anything here but its good practise.

Now look at what you have made... it looks nice enough but where the lightening bolt goes through the circle it looks a bit odd so we will make it look like the circle is lying on top of the bolt. Where the bolt goes through the circle, note the edges and the vertices. Move them so that the lines are just either side of where the bolt goes through. Then make new edges using CTRL+R on the outsides of these edges. Like so:



Creating the overlap

Now you have squares where the lightning bolt hits the circle. Change to Wireframe (z) if you are not already in it then highlight the 16 vertices of these boxes and raise them. Now do the same for where the center line crosses the bolt but create 4 lines instead of two. To explain why, I've done a diagram.



Why we use 4 points instead of 2

Now we can make the finishing touches, add subsurf and set it to 2 or 3 and add colour! Then you are done :-) I won't go over subsurf and adding colour because people have covered that better than I could in previous 'Noob to Pro' pages. I think thats everything, I did this and ended up with the first screenshot so I'm sure you will aswell. Sorry about any spelling mistakes, and I'm sure I didn't use the easiest methods but I've only been doing this kind of stuff for 3 days :-) Good luck

Next Page: 2D Image (logo) to a 3D Model Part 2
Previous Page: Curve and Path Modeling

Simple Vehicle

Next Page: Modeling a picture
Previous Page: 2D Image (logo) to a 3D Model Part 2

Design

The idea of this tutorial is to learn to face a complex project. A vehicle is a nice object to find new problems. First, we must understand that a project is a project. A project does not reproduce the real world, a project shows an idea or thought and will result in a final image or video. Whatever does not appear in the final result is unnecessary to include in the model.

We chose a vehicle for this tutorial because there are lots of different parts to work on.

Idea

What we will show with this vehicle? Let's use the model as an analogy. Take a human characteristic and reproduce it in the vehicle.

Well, I propose the following: Short persons who have bad manners and try always to show us they are capable by imposing themselves over other people.

General Characteristics

Great power Lot of machinery Small size Compact

Parts list

Wheels Engine General shape Color and textures Scapes Lights Doors Glasses motion Rocket Launcher

Wheels

IDEA:

Should be pretty big, too big for the body, outside of it. Just a small place between front and back wheel to open the door. No plain tires but thick textured ones with big rubber pieces to get a good grip. No straight profile for top speed but a rounded one for all kinds of terrain. Never mind what is in front of him just keep going over anything in fron of him.

MODELING:

We can model it from scratch by adding a cylinder and modifing the mesh with the tools used in previous tutorials or you can take a model made by someone else and modify it according to your needs.

To model from scratch I leave the space here to anyone who wants to show us exactly how to do it.

To take an existing model you can find the tires at :"http://www.katorlegaz.com/index.php?a=thumbs&c=Blender_3D_Model_Repository" released under the Blender Artistic License.

We should choose one, reduce the hole and enlarge the bump.

render of the tires that you can find in the Blender Model Repository

Modeling a picture

Next Page: Bones
Previous Page: Simple Vehicle

Modeling a picture

Ever seen an awesome looking picture you wanted to turn into a 3D model? Like a logo or a symbol? Well, it's actually pretty easy... it just takes some time to do.

- First off, you're going to need a picture to trace, I'm currently doing a project for a friend to do with devils and demons, so I chose a demonic looking face for this tutorial: Demonic Face (http://www.masterpiecepumpkins.com/Graphics/Devil2%20(1.5)_____PM.JPG)
- Now we open Blender and start a new project. Delete the default cube. Before you start tracing the face, you need to set the face as the background image. To do this, click 'view', then 'Background Image'. A box should pop up with only one button in it (Use Background Image), click it. Now some settings appear, we're only interested in one of them for this tutorial. Click the small button with a picture of a miniature folder on it (it looks kind of like a feather pen.) It's the first one under the 'use background image' button. From there, select the picture you want to trace. Like this: Background Selection (http://k.domaindlx.com/genis/images/bg_select.jpg)
- Ok, now for the long part. Zoom in to the new background image just a little bit. Now, add a bezier curve, and size it down a little. Hit **F9** and find and click the button that says 'Poly'. Now there should be a few more vertices to work with and the curve should be just a bunch of joined lines. Select one point at a time and using the **GKEY** move it to a point along the background image(or face in this case). Do the same for all of the rest of the vertices, making sure you only have one vertex selected at a time or you'll move more than just the vertex you want to. once this is done, select one of the end vertices of the curve (it doesn't matter which end) and use **SHIFT+DKEY** to copy that vertex. Move the newly copied vertex to a point along the edge of the face a small ways away from the vertex you copied it from. Continue doing this until you have a complete outline (of the whole face or just one part, like the ear). Here's what it should look like (I did the left ear):Tracing (http://k.domaindlx.com/genis/images/points.jpg) . You can't see it in the picture, but six of the points on the right side of the ear are connected, while the rest aren't. In order to get the effect we're looking for here, we need to connect all of the points around the edge to make an outline (make sure not to connect the points accross the picture or you'l have a messed up outline).
- To get the outline for the whole face, just do the exact same thing around all of the edges. We still have a problem though, most of the points aren't joined by a line, so all we have is a bunch of dots. This is easily solvable. Using the **BKEY** or the right click of the mouse, we select a bunch of vertices at a time (somewhere betwen 5 and 10), and hit the **FKEY** a few times. Every time you hit the **FKEY** it should connect two of the points. Do this until all of the selected points are connected, then unselect them and select another group and use **FKEY** to join them. Keep doing this until all of your points are connected. To connect the last two points, select all the points and press the "C" key, to close the polygon.

[edit: A better option would be to select a vertex on one of the ends of the whole line, hold down the **CTRL** and left-click on a certain point on the image. This will create a new vertex, immediately connected to the vertex you selected.]

- Now that we've got the entire face traced (or outlined if you want to call it that), we can make it 3D. Hit **F9** again and find the Ext1 and Ext2 properties, shown here: Ext1 & Ext2 (http://k.domaindlx.com/genis/images/make3D.jpg) . Change the values and see what happens. They correspond to the depth of the outline. Try changing them around until you find what looks good. Now, you'll notice that the lines just stick out straight. I'm still

investigating how to actually model a head from the outlined face ... so if anyone has any ideas, feel free to add them to this page.

- in order to make it have depth you should make the outline out of mesh points insted of a curve. add a primitive mesh and delete all the verts in edit mode, then ctrl click to all point outline. add depth to the surface in a side view(split vievs so you can see what your moveing). it helps to have 2 or more refrence images, but you can wing it. usuly the final result has to be subsurfed.

(USER EDIT: I accidentaly started it with mesh instead of curve. You can do the same thing with extrude, but I have no idea how to go on after that) (USER EDIT LATER: If you subsurf it, it creates a relatively 3D looking image. Its really cool)

(Another user, even later: IF you want to turn your curve into a mesh, hit Alt-C. Note that this is NOT reversible.)

Next Page: Bones
Previous Page: Simple Vehicle

Using Bones

Next Page: Materials and Textures
Previous Page: Modeling a picture

Bones

Bones are a modeling tool that is especially important if you're trying to model characters that move. Bones allow you to move characters' limbs in a way that is much simpler than trying to re-arrange the vertices every time.

Basically how it works is that a bone will be associated with certain vertices, which will move along with the bone when the position is changed in pose mode. Using bones is fairly simple once you get the hang of it, but, like many things in Blender, can be a little daunting at first sight. Never fear - that's what tutorials are for!

A model

Bones don't do much on their own - in fact, they turn invisible on render time! So, we'll need a model to use them with. If you haven't already, use an earlier tutorial to create a simple model, and we'll be on our way!

For this tutorial, we're going to use a model with human proportions, but bones can be used with just about any body type. The same idea can be applied to cats, spiders or whatever!,

Laying down bones

More to come soon!

Until more comes, here is a quick step by step to get you started

Thanks to Ilias for support. Written by Thomas Westin

Create a cylinder

- Select top view (**NUM7**)

- **TAB** to Object mode

- **SPACEBAR** in 3D view

Extrude the cylinder

- Select side view (**NUM3**)

- **TAB** to Edit mode

- **AKEY** to deselect all

- **BKEY** to box select top vertices

- drag the vertices up

Add armature

- **TAB** to object Mode

- **SPACEBAR>**Add>Armature

- Add some armatures...

Subdivide

- **TAB** to Edit

- **AKEY** to Select All

- **WKEY** to Subdivide. Do this 2 times

Parent it

- **TAB** to Object

- **SHIFT+RMB** to select Cylinder and then Armature. Armature selection MUST be lighter colored

- **CTRL+PKEY**, Armature, Create from closest bone

NB for total newbies; I started moving the armature around in pose mode and saw nothing happen, it took some time to work out that you need to move and scale the bones to sit inside your cylinder as bones would in an arm. Hope this helps A.A.

Try it

- Go to Pose mode

- Select and Drag the bones joints

Next Page: Materials and Textures
Previous Page: Modeling a picture

Materials and Textures

Next Page: Quickie Material
Previous Page: Bones

Materials and textures are the things giving your model colour and structure. Without materials your models would always stay the same dull grey blender uses by default. While materials give the model colour, textures give it a structure. They allow you to do things such as add a wood pattern to a desk and to make your rendered images look more realistic.

To find out more about materials and textures, go to the next page.

Next Page: Quickie Material
Previous Page: Bones

Quickie Material

Next page: Quickie Texture
Previous page: Materials and Textures

Your First Material

This page is under construction

You first need to go back to the default Blender scene. Hit **CTRL+XKEY** to start with a new Blender scene. Note this also switches you automatically to Object mode. Select the default cube with the **RMB** and open the Material buttons by pressing the shading button(**F5**).

The Blender should already have a material assigned to the cube. You know that because you see this:

Click the next to MA:Material in the Material panel to delete the link to the datablock. This removes the material from the object, removes several tabs from the Button window, and removes a lot of information from the Material panel and replaces it with an "Add New" button. You could click that to create a new material, but what we want to do is reapply the old material to it. Click the button that looks like this: . You'll see a drop-down list and you want to choose "Material". This nifty drop-down will list all of the materials you've created thus far and let you apply them to any object in the scene.

Naming Names

"Material" isn't a very creative name for a material (no offense, Ton!). There are a couple of different ways to rename the material.

1. Press the button that looks like a little car. It's right next to the [x] that deletes the material. This will "auto" rename the material. It will come up with a pretty boring name, "Grey". It's better than "Material", but we can think of something better.
2. Press **LMB** over the material name, and a cursor will appear for you to put your custom material name. To clear the entire name before entering your own, press **BACKSPACE**. There are many places in the Blender you can enter your own value using the same method. Rename the material to "Green Ooze".

Setting the Color

Obviously, just changing the name of the material doesn't make the material green. We have to do some work on it still. You'll notice this section of buttons: 

Set the **R** value to 0.149, the **G** value to 1.000, and the **B** value to 0.446. R of course stands for red, G for green, and B for blue. Mixing these values, we can achieve any color you want.

The easiest way to choose your color value is to click this box 

and use the window that pops up to select whatever color suits your fancy. You'll be using this one in later tutorials.

We've changed the color of the object, but we can also change the color of the specular. I'd tell you what the specular is, but I'll let you see for yourself when you change the specular color.

Click this button 

. Now, instead of the **R**, **G**, and **B** sliders showing the material color, they will show the specular color. Keep your eyes on the Preview of the material and start messing with the **R**, **G**, and **B** sliders. Do you see what the specular is now? [Note: You may have to hit Enter before your changes appear in the preview]

Set the specular color to **R** = 0.640, **G** = 0.990, and **B** = 0.566. With this value we should be able to get a good ooze down the road.

There's a billion other buttons here. We'll get to them all eventually. Keep this file open and go to the next tutorial, where we will perfect the ooze.

Next page: Quickie Texture
Previous page: Materials and Textures

Quickie Texture

Next page: Procedural Textures
Previous page: Quickie Material

This tutorial uses the file from the previous tutorial. If you didn't do it before, go back and do it now.

Intro

Textures are laid on top of materials to give them complicated colours and other effects. An object is covered with a material, which might contain several textures: an image texture of stone, a texture to make the stone look bumpy, and a texture to make the stone deform in different ways.

Adding a texture

Select the object to be textured. Add a Material to it if it hasn't got one already. Select the

Textures tab with the spotty square icon (or by pressing F6). Click Add New. Select the texture type from the drop-down list. You can also change the texture's name (It will be something like Tex.001) by clicking on the name box. Naming things is good!

Adding a cloud texture

Add a new Cloud texture (select Clouds from the texture typr drop-down list). A preview appears, as well as some parameters to experiment with. Go back to the Materials tab (Click the red sphere or press F5) and a coloured preview of the texture appears. It is purple! All new textures default to this colour. On the right hand side there are three tabs: Texture, Map Input, and Map To. Select the Map To tab. The RGB (Red, Green, Blue) sliders here adjust the colour of the texture. Some areas are transparent and show the material underneath. This allows you to layer textures. Make the colour black. Render the object (F12) to see the effect. Next we will add a stucci texture to make our clouds look bumpy.

Adding a stucci texture

Go back to the textures tab (F6) and select the next texture channel (one of the blank buttons under 'Tex'). Add a stucci texture. Back to the Material (F5), and click the "Map To" tab. Turn Col off and Nor on. Col means the texture affects the colour. Nor means it affects the rendered normal, or the angle the renderer treats the surface as - creating fake shadows on the surface. Play with the Nor slider, but leave it on about 4. Now click the Map Input tab - we are going to change the size of the texture. Set SizeX to 0.2 (stretching the texture five times on the x-axis) and SizeY to 10 (Squishing the texture 10 times in the Y-axis). Render! The object looks like it has grooves.

Adding an image texture

Any JPG will do for this bit, but if you can't decide, find a nice picture of wood you'd like to use as a texture on the web. Using the Image Search feature of a search engine is an easy way. Remove the two old textures (on the Textures tab, select the old ones and click the little X next to their name) and add a new Image texture. Still in Textures, select image from the texture type drop down menu, click Load Image, and choose your image. Hit F12 to render. Back to the Map Input tab (Materials, F5), try the effects of Flat, Cube, Tube and Sphere, and the XYZ buttons below. Also remember you can change the size of your image using the Size parameters (SizeX etc.)

Further Reading

Textures are a very powerful tool, and layering them can create all sorts of interesting effects. You can use Disp to actually displace the vertices of your object based on the texture (best used with a high vertex count or Subsurf) and do all sorts of other tricks. The Nor feature is very good for making objects look more realistic without increasing the rendertime overly much.

Next page: Procedural Textures
Previous page: Quickie Material

Procedural Textures

Next page: Creating Basic Seawater
Previous page: Quickie Texture

Procedual Textures

Texturing objects can be broken down into two categories: procedural and image texturing. Procedural texturing makes use of mathematical formulas to generate textures. These textures are generated within Blender. Image texturing uses images created or captured outside of the Blender, either in an image manipulation program such as the GIMP or Photoshop, or captured on a camera. It is important to know the advantages and disadvantages of each of these texturing methods. Move on to the next tutorial to learn about procedural texturing.

Current Procedual Textures

Blender currently supports many procedual textures, which are : Clouds, Marble, Stucci, Wood, Magic, Blend, Noise, Musgrave, Voronoi and DistortedNoise

Next page: Creating Basic Seawater
Previous page: Quickie Texture

Creating Basic Seawater

Next page: Texturing Basic Seawater
Previous page: Procedural Textures

75% of the Earth's surface is covered with water. In homage to this great fact, we will develop your materials skills first by creating basic seawater.

First we create a new file in Blender and delete the default cube by pressing **XKEY** and confirming the popup dialog. Now switch to top view with **NUM7** and enter **SPACE > Add > Mesh > Plane** to create a plane. Then scale it up to be many many times it's original size with the **SKEY** the way you've already learned in one of the earlier tutorials.

Now off to the actual texturing work. Press **F5** to bring up the Material Buttons in the Buttons Window. You will probably find two new small windows appearing here: one called *Material* [user comment: in my Blender 2.42a its called *Links and Pipeline*] and the other one *Preview*.

Click the 'Add New' button in the Material window to create a new material named `Material.001'. To make life easier we'll rename it to something meaningful like 'Seawater' by simply clicking it and typing in the letters, as shown here (**SHIFT+DELETE** in field to clear):

Now, on the same tab, give the seawater material a color of **RGB (0.0, 0.139, 0.400)**. Find the Alpha slider and move it until it reads approximately **Alpha 0.100**.

Next page: Texturing Basic Seawater
Previous page: Procedural Textures

Texturing Basic Seawater

Next page: Mountains Out Of Molehills 2
Previous page: Creating Basic Seawater

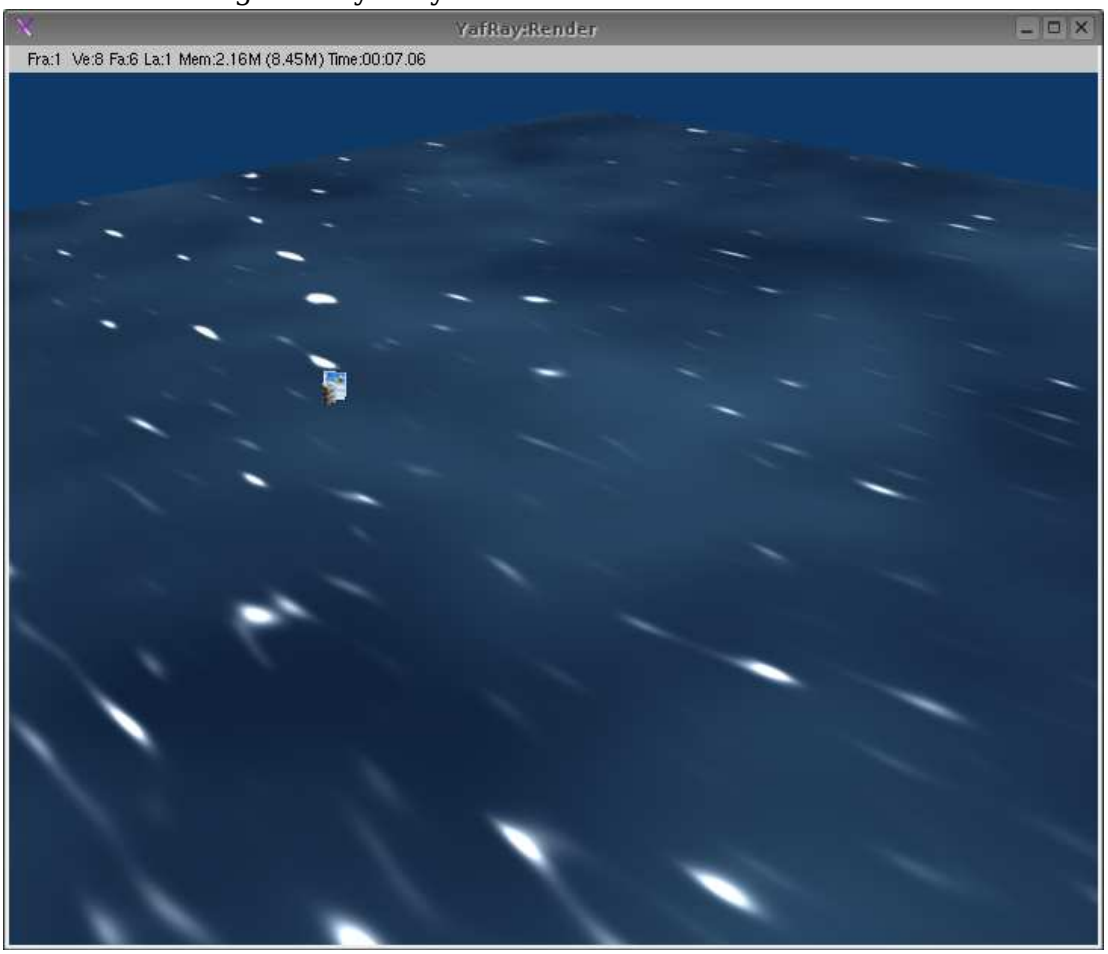
Disclaimer: This tutorial is for those of you who have installed the yafray

(http://www.blender3d.com/cms/Yafray.51.0.html) external renderer! The texture will NOT look like waves (or much of anything at all) with the internal renderer of Blender.

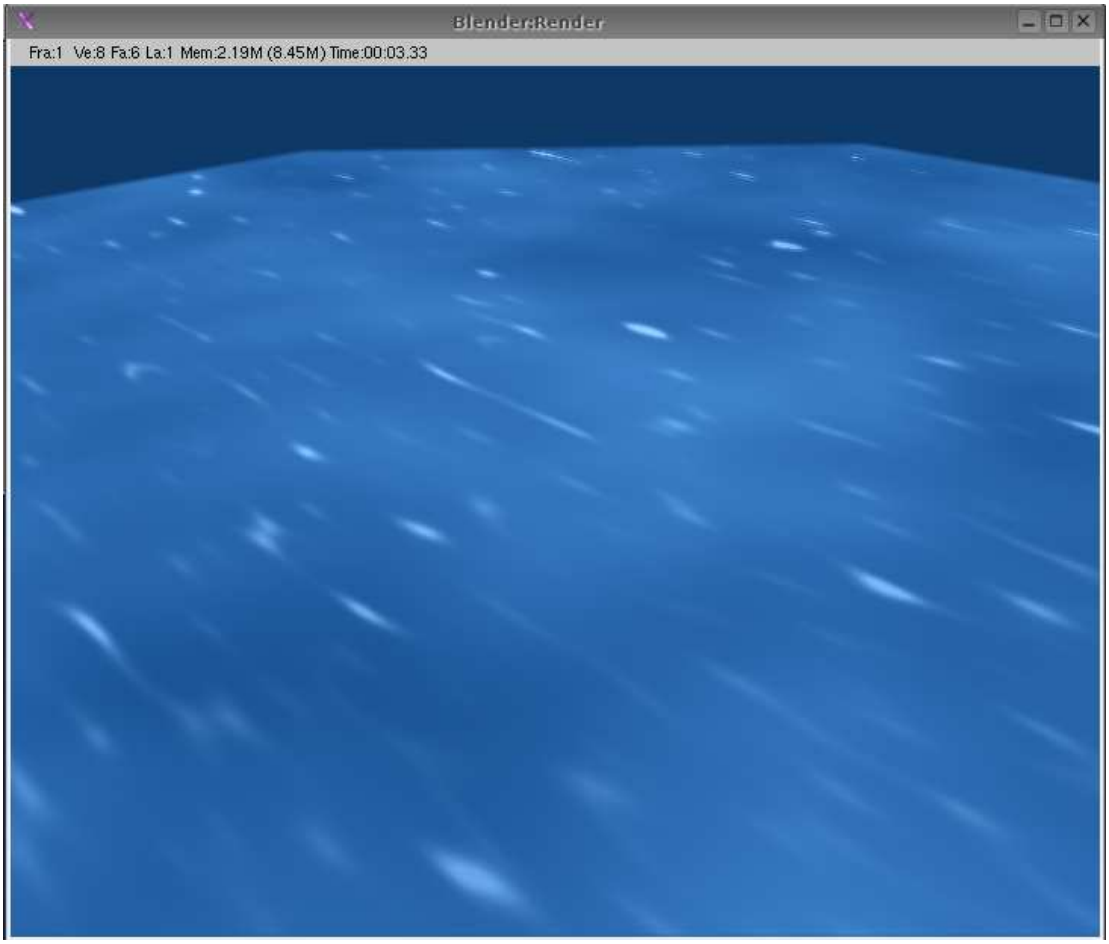
[Note: Since version 2.42 (or may earlier) you can use the internal blender render]

[*ed. note: Need a much more basic introduction to what materials, textures, maps, and all the accompanying terms are with illustrative examples before diving into a specific sea-water example. Much more effective learning when you know what you're changing.*]

A rendered image with yafray renderer



A rendered image with blender renderer



Now we'll add a procedural texture to our seawater, which will give it a "wavy" look. Click the Texture button (looks like bricks) or press F6 to view the texture buttons subcontext. Click on the knob to the left of the texture name and select the "Add New" button. This creates a new texture named "Tex.001". Click on the name and change it to "Waves".

Go to the Texture Type pull-down and select 'Stucci'. On the Stucci tab click 'Wall Out' and 'Soft noise', and change 'Noise Basis' to 'Voronoi F1'. [*ed. question: Need a little more detail on playing with what these parameters do*] Our Waves texture is ready; next, we will refine how it is applied to our Seawater material.

Left click on the Materials button (looks like a red sphere) to return to the material buttons subcontext. Look at the Texture panel, and you'll see that the "Waves" texture has been automatically associated with the Seawater material.

Select the 'Map To' tab. Click the 'Nor' and 'Spec' buttons so they're selected and have white text (the white text indicates a positive mapping). Click the 'Hard' button twice so it's selected and has yellow text (the yellow text indicates a negative mapping).

Select the 'Map Input' tab. There, look at the X, Y and Z scaling. The values to use here depend on the size of your water plane (you can see and edit the size of the plane by going to Object, Transform Properties or by pressing 'N'. With a plane size of X=3, Y=3, Z=1, a reasonable map input scaling is sizeX=5, sizeY=20, sizeZ=25. One important consideration, if you're mapping to a flat surface, is that the X and Y scalings shouldn't match (or Y and Z or X and Z for a vertical plane). This is part of what gives the Stucci/voronoi a good "wave" look.

[Note: for the subsequent step you may have to change the *Texture Blending Mode* to substract or multiply, instead of mix or add, to get a fine rendered image. Since version 2.42 (or may earlier) use **F5 -> MapTo -> drop down list** select **Substract or Multiply** and continue with this tutorial. This also works for the yafray renderer, later you can change it

back to Mix after you applied a second texture with 'Col' enabled.]

Now do a render (F12) and look at the result. If you just get a big blue-green looking plane, you are probably using the internal blender renderer. Install YafRay if you haven't already. To render with it, press F10 to view the Scene context. On the Render tab, beneath the huge 'Render' button, change the Render Engine from 'Blender internal' to 'YafRay'.

(Note: You have to have your light a good distance away from the plane, and have it set very powerful to see more than a black square.)

This looks okay, but a bit too jagged. To smooth things out, we'll add a second texture. Go to the Texture subcontext (F6) if you're not already there. In the Texture tab, click the first empty rectangle beneath the "Waves" texture. Click 'Add New'. Rename the new texture "Clouds", and change its texture type to 'Clouds'. In the Clouds tab, set NoiseSize to .2 and NoiseDepth to 2. Switch back to the material buttons subcontext, select the Texture tab, and verify that the new Clouds texture is checked and selected. Click on the Map To tab and select 'Nor' ('Col' should already be selected).

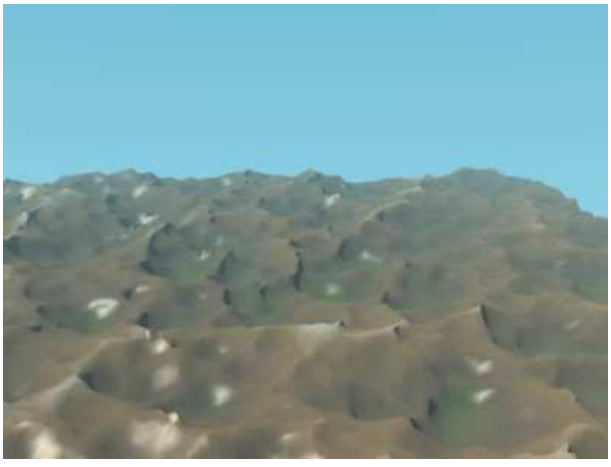
Render (F12), admire your water, and maybe drink a tall glass of something refreshing!

Next page: Mountains Out Of Molehills 2
Previous page: Creating Basic Seawater

Mountains Out Of Molehills 2

Next page: Basic Carpet Texture
Previous page: Texturing Basic Seawater
This tutorial shows you how to use displacement mapping to make a simple environment.

1. Make a grid. (Add/Mesh/Grid) 32x32 will do just fine.
2. Set it smooth. (Editing/Link and Materials/Set Smooth)
3. Make a new material for it. (Shading/Material/Add New)
4. Make a new texture for the material. (Shading/Texture/Add New)
5. Go to Shading/Texture Buttons. You can see your newly created texture there now.
6. Change Texture Type to Clouds.
7. Change the name of the texture to be more descriptive. For example GroundDisp or something similar.
8. Go back to Shading/Material buttons. You can see our cloud texture applied now but it's not applied correctly yet.
9. Go to Shading/Map To. This defines how the selected texture is mapped on our material.
10. Check out Col and check Disp on.
11. Set camera and a few lights to the scene.
12. Render.



You can tweak the environment easily by changing Nor value in the Shading/Map To. This defines how strongly the displacement texture affects the material.

You could also add subsurfing to the ground area to get smoother results. Also feel free to tweak the texture and try out different alternatives.
Next page: Basic Carpet Texture
Previous page: Texturing Basic Seawater

Basic Carpet Texture

Next page: Image Textures
Previous page: Mountains Out Of Molehills 2

I've got the basic tutorial, here on my comp, but i am working on maybe using the more recent aditions to blenders materials/textures to create a better effect. Once i have will post, or if it turns out just the same, ill put up the old tutorial.

-- Please post what you've got in the meantime? :)

Next page: Image Textures
Previous page: Mountains Out Of Molehills 2

Image Textures

Next page: The Rusty Ball
Previous page: Basic Carpet Texture

Procedural texturing is very powerful; however, sometimes it is difficult or impossible to generate the desired realism with them. Image texturing is there for you when you need it. To review, the basic idea is to take an outside image and wrap it around your model. Now move on to the next tutorial to learn how to do this.

Free Image Texture Editors

- Wood Workshop (http://www.spiralgraphics.biz/ww_overview.htm) A free utility (Requires Operating System: Windows 2000/XP) that generates surprisingly high quality tiling wood texture images. These textures can be exported as standard image files for use within Blender.

Next page: The Rusty Ball
Previous page: Basic Carpet Texture

The Rusty Ball

Next page: Creating Pixar-looking eyes
Previous page: Image Textures

Making objects with image textures is not really hard for simple objects like balls, cubes, and tubes. I'll show you how to do this:

- Make a new Scene in Blender and delete the default cube.
- Make an object you want to have the image on it (I recommend a Mesh plane, sphere or tube).
  - if you are making a Mesh Plane, change your view to above, by pressing **NUM7**
- Go to the materials (**F5**) and select the default material (of course you can also make a new material).
- Now go to the textures (**F6**) and choose "Image" as "Texture type" for the selected texture. There will appear many options, the most interesting one is "Load image". Click it and select an image. (Note: Bitmaps tend to get all screwy. JPG's are recommended)

- After this, you'll have to specify how your image should be applied to your object. To do this, go to the materials again, where you'll have to find the "Map input" tab (near the textures tab). If you have selected it, you'll see four buttons: *Flat, Cube, Tube and Sphere*. Select the option which meets your object best.
- Render your object. If you can't see your picture well, you can try to rotate your object or select another option in the "Map input" tab.

You can also render videos onto objects using this method. Just select a movie in the "Load image" dialog and enable the option "Movie" at the textures buttons. *NOTE:* Blender ONLY works with Full Resolution video, not video which has been compressed using a codec. Most video software will allow you to export video as "full frames" or "no compression". Experiment a bit!

Next page: Creating Pixar-looking eyes
Previous page: Image Textures

UV Mapping

Next page: Quickie UV Map
Previous page: Creating Pixar-looking eyes

Learning the basics of using UV mapping to map a texture accurately onto an object.

In case you're interested, UV mapping stands for the technique used to "wrap" a 2D image texture onto a 3D mesh. "U" and "V" are the name of the axes of a plane, since "X", "Y" and "Z" are used for the coordinates in the 3D space. For example: increasing your "U" on a sphere might move you along a longitude line (north or south), while increasing your "V" might move you along a line of latitude (east or west).

Next page: Quickie UV Map
Previous page: Creating Pixar-looking eyes

Quickie UV Map

Next page: UV Map Basics
Previous page: UV Mapping

See the next page for a relevant video tutorial.

Next page: UV Map Basics
Previous page: UV Mapping

UV Map Basics

Next page: Every Material Known to Man
Previous page: Quickie UV Map

Intro

Until someone types in the text for this chapter, you can watch a good video tutorial from the main Blender (http://www.blender3d.org) site. It is called **LSCM UV Mapping** and it is located on this page: http://blender3d.org/cms/Model_Material_Light.397.0.html

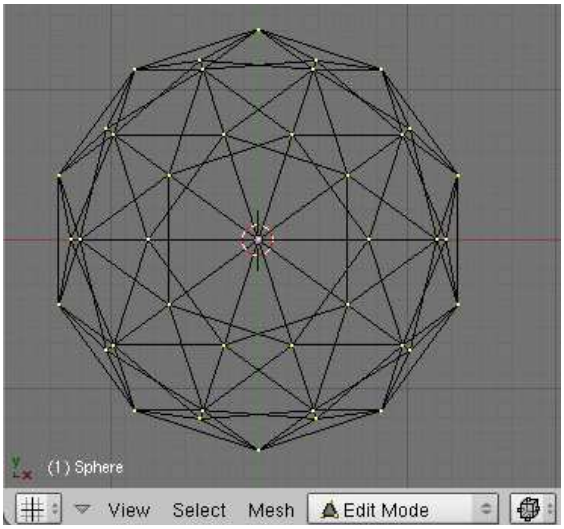
Someone now proceeds to write some beginning text to this page... (pretty much a summary of the *excellent* above video by GreyBeard.) Should use a better object (that requires pinning etc).

The Basics of UV Mapping

Add an icosphere

We'll use an icosphere (2 iterations) for this demonstration. So, create an icosphere with 2 iterations. (**SPACE**, Add->Mesh->Icosphere)

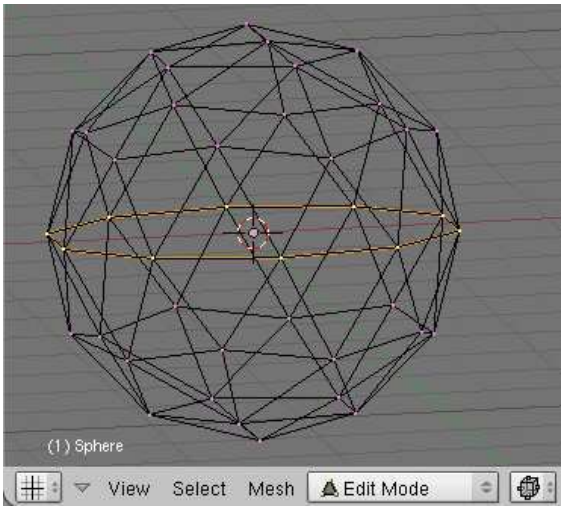
Make sure you are in top view like in the picture before you create it. Otherwise the equator of the sphere is probably not parallel to the x/y-plane and unwrapping will give strange results.



Mark a seam

Select a ring of vertices on the icosphere. (Like an equator).

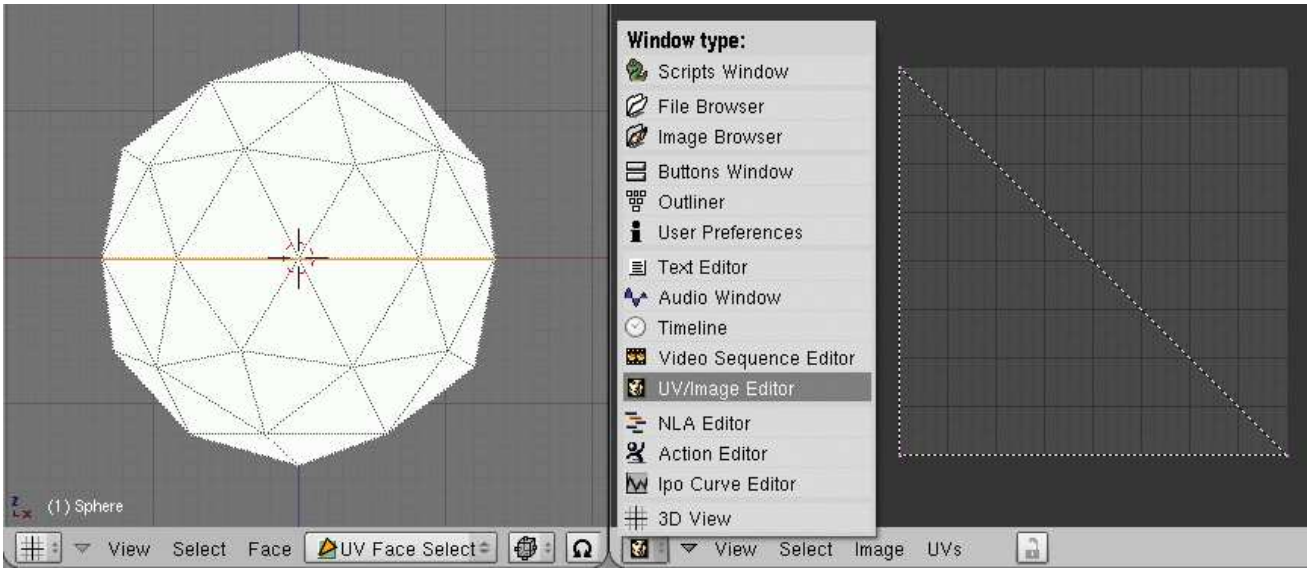
Press **CTRL+EKEY** and select "Mark Seam". This tells the UV unwrapper to cut the mesh along these edges.



Note: since Blender 2.42 you can alternatively go into UV Face Select-mode and mark single edges as seams with **CTRL+RMB** and you could also select the faces of the upper or lower half of the sphere and then press **CTRL+EKEY** and select "Mark Border Seam". This is quite nice because you don't have to switch between Edit-mode and UV-mode any more to mark seams.

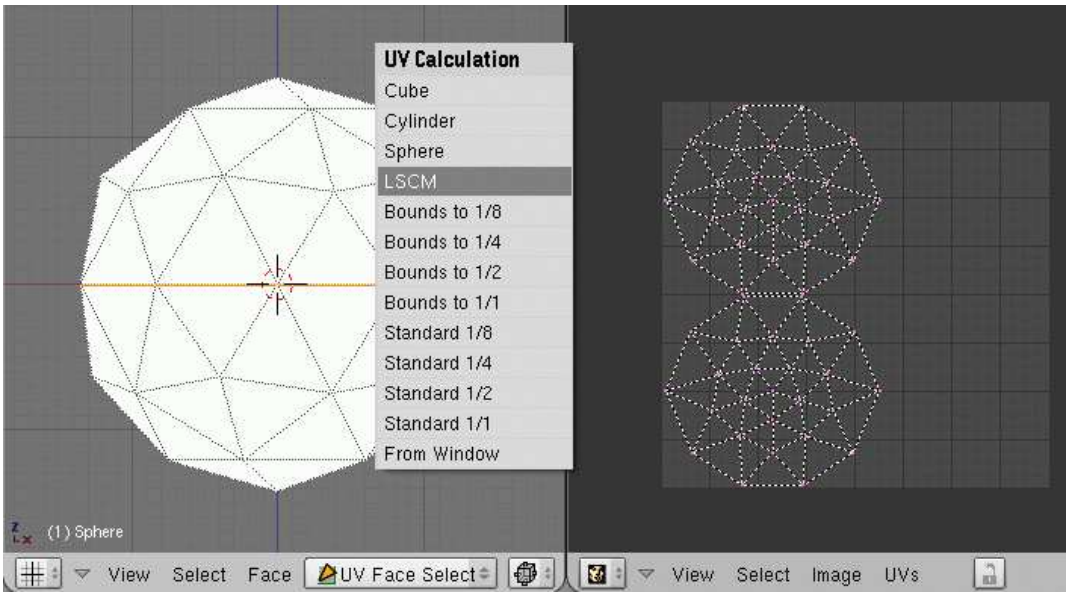
Unwrap the mesh

Set the window mode to "UV Face Select" (this mode is another mode like "Edit Mode" or "Object Mode", and can be set using a drop-down box at the bottom of the 3D viewport).



To make a second 3D viewport click the **RMB** near the border in the 3D viewport and select "Split Area". Set its window type to "UV/Image Editor" with the drop down box on the far left (looks like a grid for the 3D viewport) or with **SHIFT + F10**.

In the first window, select all your vertices, and hit **UKEY** and then *LSCM*. For Blender 2.42 select all the faces, and hit **UKEY** and then *unwrap* to use LSCM.



LSCM is one of the algorithms for unwrapping a mesh onto the 2-dimensional UV space, acronym for *Least Squares Conforming Map*. It is a very useful unwrapping method because it attempts to preserve the shape of each face, much like unwrapping the cloth of a garment.

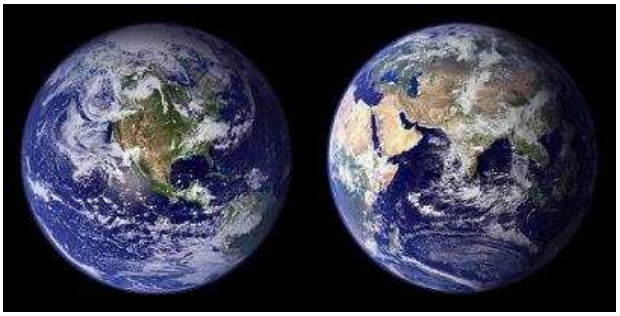
Make a template image

When you have tweaked a nice layout and intend to make the texture image yourself, you may ease the texture drawing by saving an image of the UV layout. This image can then be opened in your image editing program of choice to make a basis for the UV texture by showing where each surface goes. First run UVs->Save UV face layout: With *Wrap* selected, the layout will maintain its proportions which is best for general purpose. Not selecting it will scale your layout into square proportions which is mostly useful for Blender's game

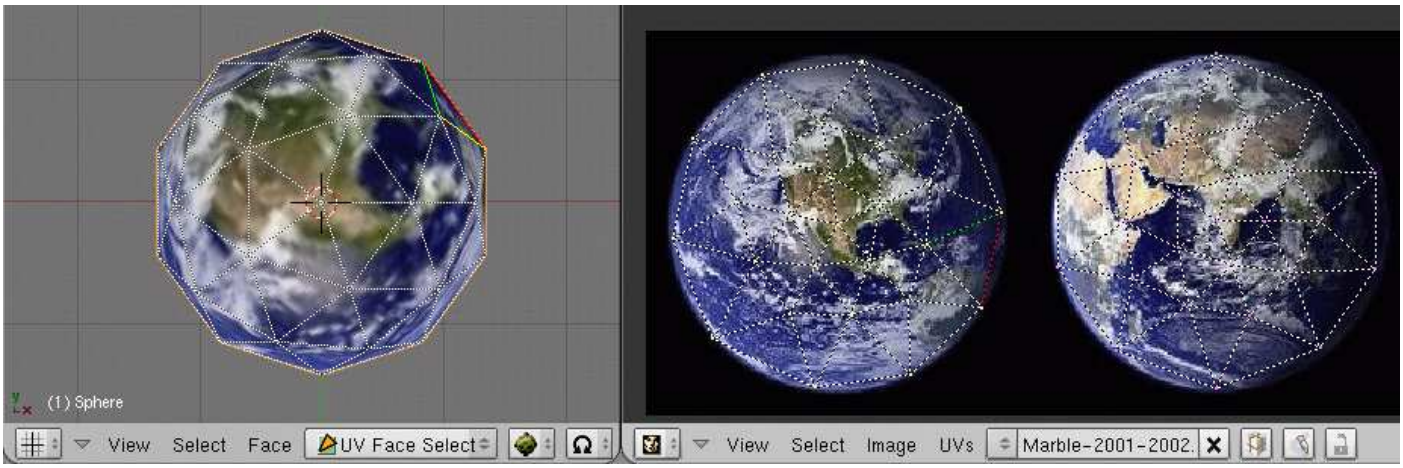
engine where textures should preferably be square. Choose location of the image file as desired (the default name is the name of the object to be textured) and press *Export*. *Tip:* if your image painting program supports layers, try putting the UV layout in a locked, transparent layer above the actual painting. If you do not alter the dimensions of the exported UV image in any way it will fit perfectly with your UV layout when the image is loaded back into Blender.

Apply an image

Save the following image:

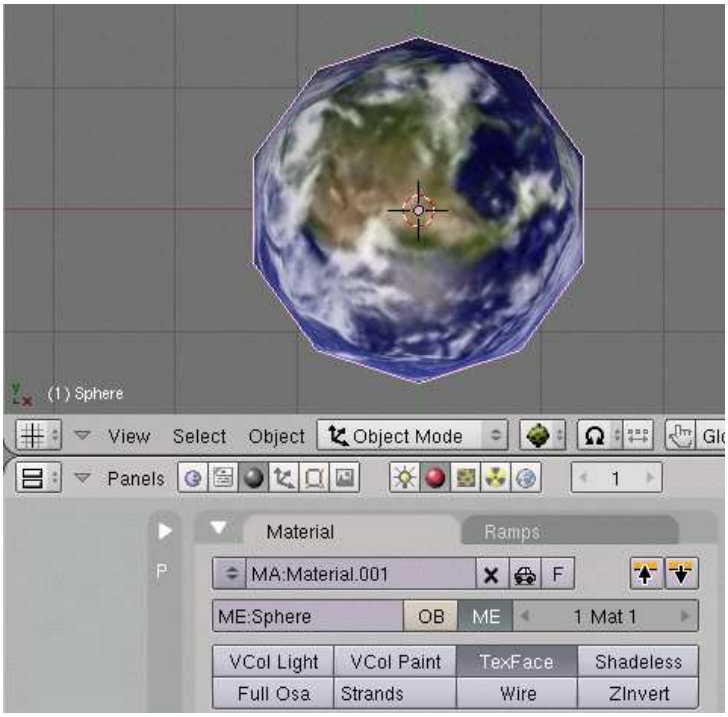


Load it in the UV/Image window by clicking Image->Load image (or Image->Open in version 2.4). Then with the very basic operations, grab, rotate and scale, adjust the unwrapped mesh so that it fits nicely on top of the image.

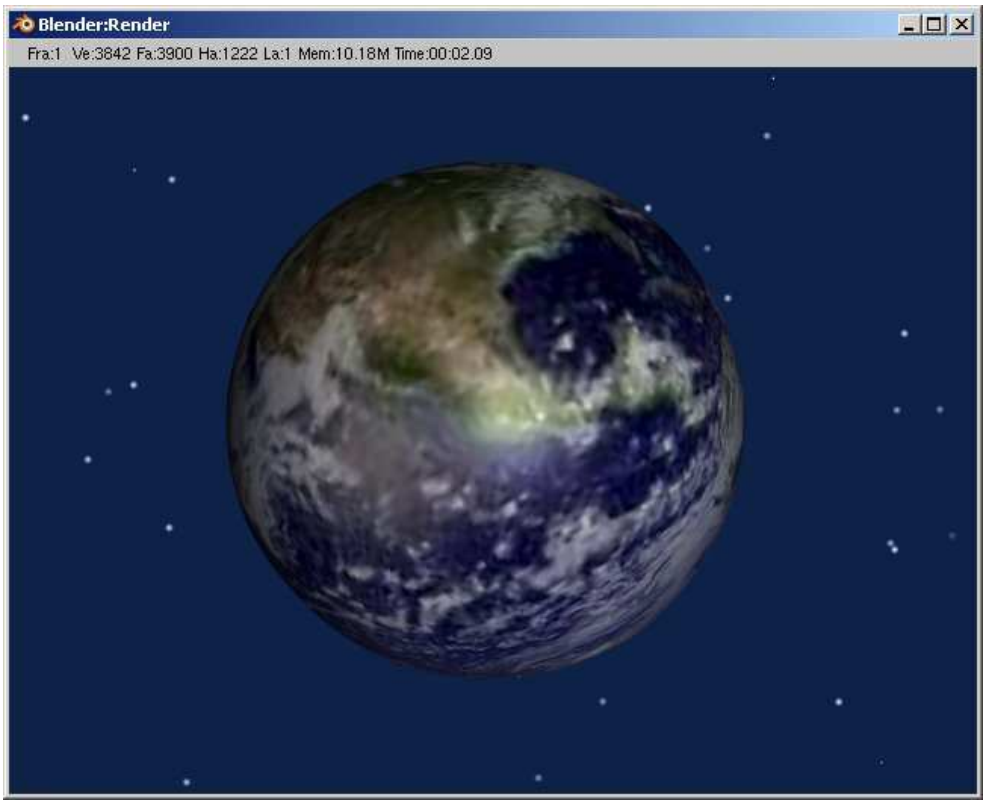


Admire your new creation

Back in the 3D viewport, put it in Object mode and set the Draw Type to Textured (it's normally Wire or Solid, another drop down box at the bottom of the window. Greybeard calls it potato mode). Hit TAB a couple of times to refresh your object, and admire your new picture mapped onto your object! To make the texture visible in renderings too, you also need to add a new material to the icosphere and then toggle on TexFace from the Material buttons:



To finish your work, switch to Edit Mode, select all vertices and from the Editing buttons (F9) click Set Smooth. Then next to the same panel reads Modifiers. Click Add Modifier -> Subsurf. Set Render Levels to three. Switch to Shading buttons (F5) and enable Stars. The scene is ready!



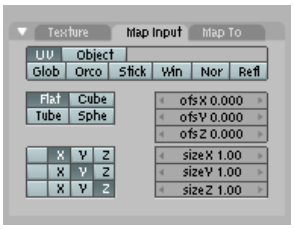
Some notes

All this relates to the UV/Image window.
If you are going to edit the layout of your unwrapping (so you can make a better picture), make sure Select->Stick Local UVs to Mesh Vertex is on. You can "pin" vertices when they have been unwrapped with **PKEY**. If you do, make sure you put at least one pin on every island. You can remove all pins with alt-p. LSCM works by trying to maintain the angles between each vertex. If an unwrapping should be symmetrical and it's not, try putting one

pin in the middle of the outside edge of the big side and one on the small side, and unwrapping again with **EKEY**.

IMPORTANT NOTES:

- Remember to set your map input with the UV and Flat buttons enabled.
- As said before, to make the texture visible in renderings, too, you also need to toggle on TexFace from the Material buttons.



Source

Greybeard's LSCM Mapping video tutorial:
http://blender3d.org/cms/Model_Material_Light.397.0.html

Next page: Every Material Known to Man
Previous page: Quickie UV Map

Every Material Known to Man

Next page: Modeling Keyboard Shortcuts
Previous page: UV Map Basics

Want to share your material settings with the world? This is the place!

Inorganic

Natural

Metals and Minerals

Glass
Gold

Other

Mirrors
Liquid
Marble

Abstract

Light Effects
Misty Globe

Organic

Plants

Woods

Human

Human Skin

Next page: Modeling Keyboard Shortcuts
Previous page: UV Map Basics

Modeling Keyboard Shortcuts

Next Page: Beginning Modeling Final Project
Previous Page: Every Material Known to Man

Blender HotKeys - *Relevant to Blender 2.36 - Compiled from Blender Online Guides*

Window HotKeys

Certain window managers also use the following hotkeys. So **ALT+CTRL** can be substituted for **CTRL** to perform the functions described below if a conflict arises.

CTRL+LEFTARROW. Go to the previous Screen.
CTRL+RIGHTARROW. Go to the next Screen.
CTRL+UPARROW or **CTRL+DOWNARROW**. Maximise the window or return to the previous window display size.
SHIFT+F4. Change the window to a Data View
SHIFT+F5. Change the window to a 3D Window
SHIFT+F6. Change the window to an IPO Window
SHIFT+F7. Change the window to a Buttons Window
SHIFT+F8. Change the window to a Sequence Window
SHIFT+F9. Change the window to an Outliner Window
SHIFT+F10. Change the window to an Image Window
SHIFT+F11. Change the window to a Text Window
SHIFT+F12. Change the window to an Action Window

Universal HotKeys

The following HotKeys work uniformly in all Blender Windows, if the Context allows:

CTRL+LMB. Lasso select: drag the mouse to form a freehand selection area.

- **ESC.**
  - This key always cancels Blender functions without changes.
  - *or*: FileWindow, DataView and ImageSelect: back to the previous window type.
  - *or*: the RenderWindow is pushed to the background (or closed, that depends on the operating system).

SPACE. Open the Toolbox.
TAB. Start or quit EditMode.
F1. Loads a Blender file. Changes the window to a FileWindow.
SHIFT+F1. Appends parts from other files, or loads as Library-data. Changes the window to a FileWindow, making Blender files accessible as a directory.
F2. Writes a Blender file. Change the window to a FileWindow.

SHIFT+F2. Exports the scene as a DXF file
CTRL+F2. Exports the scene as a VRML1 file
F3. Writes a picture (if a picture has been rendered). The fileformat is as indicated in the DisplayButtons. The window becomes a File Select Window.
CTRL+F3 (ALT+CTRL+F3 on MacOSX). Saves a screendump of the active window. The fileformat is as indicated in the DisplayButtons. The window becomes a FileWindow.
SHIFT+CTRL+F3. Saves a screendump of the whole Blender screen. The fileformat is as indicated in the DisplayButtons. The window becomes a FileWindow.
F4. Displays the Logic Context (if a ButtonsWindow is available).
F5. Displays the Shading Context (if a Buttons Window is available), Light, Material or World Sub-contexts depends on active object.
F6. Displays the Shading Context and Texture Sub-context (if a ButtonsWindow is available).
F7. Displays the Object Context (if a ButtonsWindow is available).
F8. Displays the Shading Context and World Sub-context (if a ButtonsWindow is available).
F9. Displays the Editing Context (if a ButtonsWindow is available).
F10. Displays the Scene Context (if a ButtonsWindow is available).
F11. Hides or shows the render window.
F12. Starts the rendering from the active camera.
LEFTARROW. Go to the previous frame.
SHIFT+LEFTARROW. Go to the first frame.
RIGHTARROW. Go to the next frame.
SHIFT+RIGHTARROW. Go to the last frame.
UPARROW. Go forward 10 frames.
DOWNARROW. Go back 10 frames.
ALT+A. Change the current Blender window to Animation Playback mode. The cursor changes to a counter.
ALT+SHIFT+A. The current window, plus all 3DWindows go into Animation Playback mode.
IKEY. Insert Key menu. This menu differs from window to window.
JKEY. Toggle the render buffers. Blender allows you to retain two different rendered pictures in memory.
CTRL+O. Opens the last saved file.
QKEY. OK? Quit Blender. This key closes Blender. Blender quit is displayed in the console if Blender is properly closed.
ALT+CTRL+T. TimerMenu. This menu offers access to information about drawing speed. The results are displayed in a pop-up.
CTRL+U. OK, Save User defaults. The current project (windows, objects, etc.), including UserMenu settings are written to the default file that will be loaded every time you start Blender or set it to defaults by pressing **CTRL+X.**
CTRL+W. Write file. This key combination allows you to write the Blender file without opening a FileWindow.
ALT+W. Write Videoscape file. Changes the window to a FileWindow.
CTRL+X. Erase All. Everything (except the render buffer) is erased and released. The default scene is reloaded.
CTRL+Y. Redo. Mac users may use CMD+Y.
CTRL+Z. Undo. Mac users may use CMD+Z.
SHIFT+CTRL+Z. Redo. Mac users may use SHIFT+CMD+Z

Object Mode HotKeys

These hotkeys are mainly bound to the 3D Viewport Window, but many work on Objects in most other windows, like IPOs and so on, hence they are summarized here.

HOME. All Objects in the visible layer are displayed completely, centered in the window.
PAGEUP. Select the next Object Key. If more than one Object Key is selected, the selection is shifted up cyclically. Only works if the AnimButtons->DrawKey is ON for the Object.
SHIFT+PAGEUP. Adds to selection the next Object Key.
PAGEDOWN. Select the previous Object Key. If more than one Object Key is selected, the selection is shifted up cyclically. Only works if the AnimButtons->DrawKey is ON for the Object.
SHIFT+PAGEDOWN. Adds to selection the previous Object Key.
ACCENT. (To the left of the **1KEY** in US keyboard) Select all layers.
SHIFT+ACCENT. Revert to the previous layer setting.
TAB. Start/stop EditMode. Alternative hotkey: **ALT+E.**
AKEY. Selects/deselects all.

CTRL+A. Apply size and rotation. The rotation and dimensions of the Object are assigned to the ObData (Mesh, Curve, etc.). At first glance, it appears as if nothing has changed, but this can have considerable consequences for animations or texture mapping. This is best illustrated by also having the axis of a Mesh Object be drawn (EditButtons->Axis). Rotate the Object and activate Apply. The rotation and dimensions of the Object are 'erased'.
SHIFT+CTRL+A. If the active Object is automatically duplicated (see AnimButtons->DupliFrames or AnimButtons- >Dupliverts), a menu asks Make duplis real?. This option actually creates the Objects. If the active Mesh Object is deformed by a Lattice, a menu asks Apply Lattice deform?. Now the deformation of the Lattice is assigned to the vertices of the Mesh.
SHIFT+A. This is the AddMenu. In fact, it is the ToolBox that starts with the `ADD' option. When Objects are added, Blender starts EditMode immediately if possible.
BKEY. Border Select. Draw a rectangle with the LeftMouse; all Objects within this area are selected, but not made active. Draw a rectangle with the RightMouse to deselect Objects. In orthonormal ViewMode, the dimensions of the rectangle are displayed, expressed as global coordinates, as an extra feature in the lower left corner. In Camera ViewMode, the dimensions that are to be rendered according to the DisplayButtons are displayed in pixel units.
SHIFT+B. Render Border. This only works in Camera ViewMode. Draw a rectangle to render a smaller cut-out of the standard window frame. If the option DisplayButtons->Border is ON, a box is drawn with red and black lines.
CKEY. Centre View. The position of the 3DCursor becomes the new centre of the 3DWindow.

- **ALT+C.** Convert Menu. Depending on the active Object, a PopupMenu is displayed. This enables you to convert certain types of ObData. It only converts in one direction, everything ultimately degrades to a Mesh! The options are:
  - Font -> Curve
  - MetaBall -> Mesh The original MetaBall remains unchanged.
  - Curve -> Mesh
  - Surface -> Mesh

- **CTRL+C.** Copy Menu. This menu copies information from the active Object to (other) selected Objects.
  - *Fixed components are:*
    - Copy Loc: the X,Y,Z location of the Object. If a Child is involved, this location is the relative position in relation to the Parent.
    - Copy Rot: the X,Y,Z rotation of the Object.
    - Copy Size: the X,Y,Z dimension of the Object.
    - DrawType: copies Object Drawtype.
    - TimeOffs: copies Object time offset.
    - Dupli: all Duplicator data (Dupliframes, Dupliverts and so on)
    - Mass: Real time stuff.
    - Damping: Real time stuff.
    - Properties: Real time stuff.
    - Logic Bricks: Real time stuff.
    - Constraints: copies Object constraints.
  - *If applicable:*
    - Copy TexSpace: The texture space.
    - Copy Particle Settings: the complete particle system from the AnimButtons.
  - *For Curve Objects:*
    - Copy Bevel Settings: all bevelling data from the EditButtons.
  - *Font Objects:*
    - Copy Font Settings: font type, dimensions, spacing.
    - Copy Bevel Settings: all bevelling data from the EditButtons.
  - *Camera Objects:*
    - Copy Lens: the lens value.

SHIFT+C. CentreZero View. The 3DCursor is set to zero (0,0,0) and the view is changed so that all Objects, including the 3Dcursor, can be displayed. This is an alternative for **HOME.**
DKEY. Draw mode menu. Allows to select draw modes exactly as the corresponding menu in the 3D viewport header does.
SHIFT+D. Add Duplicate. The selected Objects are duplicated. Grab mode starts immediately thereafter.
ALT+D. Add Linked Duplicate. Of the selected Objects linked duplicates are created. Grab mode starts immediately thereafter.
CTRL+D. Draw the (texture) Image as wire. This option has a limited function. It can

only be used for 2D compositing.
ALT+E. Start/stop EditMode. Alternative hotkey: **TAB.**
FKEY. If selected Object is a mesh Toggles Face selectMode on and off.
CTRL+F. Sort Faces. The faces of the active Mesh Object are sorted, based on the current view in the 3DWindow. The leftmost face first, the rightmost last. The sequence of faces is important for the Build Effect (AnimButtons).

- **GKEY**. Grab Mode. Or: the translation mode. This works on selected Objects and vertices. Blender calculates the quantity and direction of the translation, so that they correspond exactly with the mouse movements, regardless of the ViewMode or view direction of the 3DWindow. Alternatives for starting this mode:
  - **LMB** to draw a straight line.
- The following options are available in translation mode:
  - *Limiters:*
    - **CTRL:** in increments of 1 grid unit.
    - **SHIFT:** fine movements.
    - **SHIFT+CTRL:** in increments of 0.1 grid unit.
  - **MMB** toggles: A short click restricts the current translation to the X,Y or Z axis. Blender calculates which axis to use, depending on the already initiated mouse movement. Click MiddleMouse again to return to unlimited translation.
  - **XKEY, YKEY, ZKEY** constrains movement to X, Y or Z axis of the global reference.
  - a second **XKEY, YKEY, ZKEY** constrains movement to X, Y or Z axis of the local reference.
  - a third **XKEY, YKEY, ZKEY** removes constraints.
  - **NKEY** enters numerical input, as well as any numeric
  - key directly. **TAB** will switch between values, **ENTER** finalizes, **ESC** exits.
  - **ARROWS:** These keys can be used to move the mouse cursor exactly 1 pixel.
  - *Grabber can be terminated with:*
    - **LMB SPACE** or **ENTER:** move to a new position.
    - **RMB** or **ESC:** everything goes back to the old position.
  - *Switching mode:*
    - **GKEY:** starts Grab mode again.
    - **SKEY:** switches to Size (Scale) mode.
    - **RKEY:** switches to Rotate mode.

ALT+G. Clears translations, given in Grab mode. The X,Y,Z locations of selected Objects are set to zero.

- **SHIFT+G.** Group Selection
  - *Children:* Selects all selected Object's Children.
  - *Immediate Children:* Selects all selected Object's first level Children.
  - *Parent:* Selects selected Object's Parent.
  - *Shared Layers:* Selects all Object on the same Layer of active Object

- **IKEY.** Insert Object Key. A keyposition is inserted in the current frame of all selected Objects. A PopupMenu asks what key position(s) must be added to the IpoCurves.
  - *Loc:* The XYZ location of the Object.
  - *Rot:* The XYZ rotation of the Object.
  - *Size:* The XYZ dimensions of the Object
  - *LocRot:* The XYZ location and XYZ rotation of the Object.
  - *LocRotSize:* The XYZ location, XYZ rotation and XYZ dimensions of the Object.
  - *Layer:* The layer of the Object.
  - *Avail:* A position is only added to all the current IpoCurves, that is curves which already exists.
  - Mesh, Lattice, Curve or Surface: depending on the type of Object, a VertexKey can be added

CTRL+J. Join Objects. All selected Objects of the same type are added to the active Object. What actually happens here is that the ObData blocks are combined and all the selected Objects (except for the active one) are deleted. This is a rather complex operation, which can lead to confusing results, particularly when working with a lot of linked data, animation curves and hierarchies.
KKEY. Show Keys. The DrawKey option is turned ON for all selected Objects. If all of them were already ON, they are all turned OFF.
SHIFT+K. A PopupMenu asks: OK? Show and select all keys. The DrawKey option is turned ON for all selected Objects, and all Object-keys are selected. This function is used to enable transformation of the entire animation system.
LKEY. Makes selected Object local. Makes library linked objects local for the current

scene.

- **CTRL+L.** Link selected. Links some of the Active Object data to all selected Objects, the following menu entry appears only if applicable.
  - *To Scene:* Creates a link of the Object to a scene.
  - *Object IPOs:* Links Active Object IPOs to selected ones.
  - *Mesh data:* Links Active Object Mesh data selected ones.
  - *Lamp Data:* Links Active Object Lamp data to selected ones.
  - *Surf Data:* Links Active Object Surf data selected ones.
  - *Material:* Links Active Object Material to selected ones.

- **SHIFT+L.** Select Linked. Selects all Objects somehow linked to active Object.
  - *Object IPO:* Selects all Object(s) sharing active Object's IPOs.
  - *Object Data:* Selects all Object(s) sharing active Object's ObData.
  - *Current Material:* Selects all Object(s) sharing active Object's current Material.
  - *Current Texture:* Selects all Object(s) sharing active Object's current Texture.

MKEY. Moves selected Object(s) to another layer, a pop-up appears. Use **LMB** to move, use **SHIFT+LMB** to make the object belong to multiple layers. If the selected Objects have different layers, this is ORed in the menu display. Use ESC to exit the menu. Press the "OK" button or **ENTER** to change the layer setting. The hotkeys (**ALT**-)(**1KEY, 2KEY**, ... - **0KEY**) work here as well (see 3DHeader).
CTRL+M. Mirror Menu. It is possible to mirror an Object along the X, Y or Z axis.
NKEY. Number Panel. The location, rotation and scaling of the active Object are displayed and can be modified.
ALT+O. Clear Origin. The `Origin' is erased for all Child Objects, which causes the Child Objects to move to the exact location of the Parent Objects.
SHIFT+O. If the selected Object is a Mesh toggles SubSurf onn/ off. **CTRL+1** to **CTRL+4** switches to the relative SubSurf level for display purpouses. Rendering SUbSurf level has no HotKey.

- **CTRL+P.** Make selected Object(s) the child(ren) of the active Object. If the Parent is a Curve then a popup offers two choices:
  - *Normal Parent:* Make a normal parent, the curve can be made a path later on.
  - *Follow Path:* Automatically creates a Follow Path constraint with the curve as target. If the Parent is an Armature, a popup offers three options:
  - *Use Bone:* One of the Bones becomes the parent. The Object will not be deformed. A popup permits to select the bone. This is the option if you are modelling a robot or machinery
  - *Use Armature:* The whole armature is used as parent for deformations. This is the choice for organic beings.
  - *Use Object:* Standard parenting. In the second case further options asks if Vertex groups

should not be created, should be created empty or created and populated.

- **ALT+P.** Clears Parent relation, user is asked if he wishes to keep or clear parent-induced transforms.
  - *Clear Parent:* the selected Child Objects are unlinked from the Parent. since the transformation of the Parent disappears, this can appear as if the former Children themselves are transformed.
  - *... and keep transform:* the Child Objects are unlinked from the Parent, and an attempt is made to assign the current transformation, which was determined in part by the Parent, to the (former Child) Objects.
  - *Clear Parent inverse:* The inverse matrix of the Parent of the selected Objects is erased. The Child Objects remain linked to the Objects. This gives the user complete control over the hierarchy.

- **RKEY.** Rotate mode. Works on selected Object(s). In Blender, a rotation is by default a rotation perpendicular to the screen, regardless of the view direction or ViewMode. The degree of

rotation is exactly linked to the mouse movement. Try moving around the rotation midpoint with the mouse. The rotation pivot point is determined by the state of the 3DWiewport Header buttons. Alternatives for starting this mode:

- **LMB** to draw a C-shaped curve.
- The following options are available in rotation mode:
  - *Limiters:*

        - **CTRL**: in increments of 5 degrees.
        - **SHIFT**: fine movements.
        - **SHIFT+CTRL**: in increments of 1 degree.
      - **MMB** toggles: A short click restricts the current rotation to the horizontal or vertical view axis.
      - **XKEY, YKEY, ZKEY** constrains rotation to X, Y or Z axis of the global reference.
      - a second **XKEY, YKEY, ZKEY** constrains rotation to X, Y or Z axis of the local reference.
      - a third **XKEY, YKEY, ZKEY** removes constraints.
      - **NKEY** enters numerical input, as well as any numeric key directly. **ENTER** finalizes, **ESC** exits.
      - **ARROWS**: These keys can be used to move the mouse cursor exactly 1 pixel.
      - *Rotation can be terminated with:*
        - **LMB SPACE** or **ENTER**: move to a new position.
        - **RMB** or **ESC**: everything goes back to the old position.
      - *Switching mode:*
        - **GKEY**: switches to Grab.
        - **SKEY**: switches to Size (Scale) mode.
        - **RKEY**: starts Rotate mode again.

ALT+R. Clears Rotation. The X,Y,Z rotations of selected Objects are set to zero.

- - **SKEY.** Size mode or scaling mode. Works on selected Object(s). The degree of scaling is exactly linked to the mouse movement. Try to move from the (rotation) midpoint with the mouse. The pivot point is determined by the settings of the 3D Viewport header pivot Menu. Alternatives for starting scaling mode:
      - **LMB** to draw a V-shaped line.
    - The following options are available in scaling mode:
      - *Limiters:*
        - **CTRL**: in increments of 0.1.
        - **SHIFT+CTRL**: in increments of 0.01.
      - **MMB** toggles: A short click restricts the scaling to X, Y or Z axis. Blender calculates the appropriate axis based on the already initiated mouse movement. Click **MMB** again to return to free scaling.
      - **XKEY, YKEY, ZKEY** constrains scaling to X, Y or Z axis of the local reference.
      - a second **XKEY, YKEY, ZKEY** removes constraints.
      - **NKEY** enters numerical input, as well as any numeric key directly. **ENTER** finalizes, **ESC** exits.
      - **ARROWS**:These keys can be used to move the mouse cursor exactly 1 pixel.
      - Scaling can be terminated with:
        - **LMB SPACE** or **ENTER**: move to a new position.
        - **RMB** or **ESC**: everything goes back to the old dimension.
      - *Switching mode:*
        - **GKEY**: switches to Grab.
        - **SKEY**: starts Size mode again.
        - **RKEY**: switches to Rotation.

ALT+S. Clears Size. The X,Y,Z dimensions of selected Objects are set to 1.0.

- - **SHIFT+S.** SnapMenu:
      - Sel->Grid: Moves Object to nearest grid point.
      - Sel->Curs: Moves Object to cursor.
      - Curs->Grid: Moves cursor to nearest grid point.
      - Curs->Sel: Moves cursor to selected Object(s).
      - Sel->Center: Moves Objects to their barycentrum.

TKEY. Texture space mode. The position and dimensions of the texture space for the selected Objects can be changed in the same manner as described above for Grab and Size mode. To make this visible, the drawingflag EditButtons->TexSpace is set ON. A PopupMenu asks you to select: "Grabber" or "Size".
CTRL+T. Makes selected Object(s) track the Active Object. Old track method was Blender default tracking before version 2.30. The new method is the Constrain Track, this creates a fully editable constraint on the selected object targeting the active Object.
ALT+T. Clears old style Track. Constraint track is removed as all constrains are.
UKEY. Makes Object Single User, the inverse operation of Link

- - (**CTRL+L**) a pop-up appears with choices.
      - *Object:* if other Scenes also have a link to this Object, the link is deleted and the Object is copied. The Object now only exists in the current Scene. The links from

the Object remain unchanged.
- *Object & ObData:* Similar to the previous command, but now the ObData blocks with multiple links are copied as well. All selected Objects are now present in the current Scene only, and each has a unique ObData (Mesh, Curve, etc.).
- *Object & ObData & Materials+Tex:* Similar to the previous command, but now Materials and Textures with multiple links are also copied. All selected Objects are now unique. They have unique ObData and each has a unique Material and Texture block.
- *Materials+Tex:* Only the Materials and Textures with multiple links are copied.

VKEY. Switches in/out of Vertex Paint Mode.
ALT+V. Object-Image Aspect. This hotkey sets the X and Y dimensions of the selected Objects in relation to the dimensions of the Image Texture they have. Use this hotkey when making 2D Image compositions and multi-plane designs to quickly place the Objects in the appropriate relationship with one another.
WKEY. Opens Object Booleans Menu.
XKEY. Erase Selected? Deletes selected objects.
ZKEY. Toggles Solid Mode on/off.
SHIFT+Z. Toggles Shaded Mode on/off.
ALT+Z. Toggles Textured Mode on/off.

Edit Mode - General

Again, Most of these hotkeys are useful in the 3D Viewport when in Edit Mode, but many works on other Blender Object, so they are summarized here. Many Object Mode keys works in Edit mode too, but on the selected vertices or control points; among these Grab, Rotate, Scale and so on. These hotkeys are not repeated here.

TAB or **ALT+E.** This button starts and stops Edit Mode.
CTRL+TAB. Switches between Vertex Select, Edge Select, and Face Select modes. Holding SHIFT while clicking on a mode will allow you to combine modes.
AKEY. Select/Unselect all.
BKEY+BKEY. Circle Select. If you press BKEY a second time after starting Border Select, Circle Select is invoked. It works as described above. Use NUM+ or NUM- or MW to adjust the circle size. Leave Circle Select with **RMB** or **ESC**.
CTRL+H. With vertices selected, this creates a "Hook" object. Once a hook is selected, CTRL+H brings up an options menu for it.
NKEY. Number Panel. Simpler than the Object Mode one, in Edit Mode works for Mesh, Curve, Surface: The location of the active vertex is displayed.
OKEY. Switch in/out of Proportional Editing.
SHIFT+O. Toggles between Smooth and Sharp Proportional Editing.
PKEY. SeParate. You can choose to make a new object with all selected vertices, edges, faces and curves or create a new object from each separate group of interconnected vertices from a popup. Note that for curves you cannot separate connected control vertices. This operation is the opposite of Join (**CTRL+J**).
CTRL+P. Make Vertex Parent. If one object (or more than one) is/are selected and the active Object is in Edit Mode with 1 or 3 vertices selected then the Object in Edit Mode becomes the Vertex Parent of the selected Object(s). If only 1 vertex is selected, only the location of this vertex determines the Parent transformation; the rotation and dimensions of the Parent do not play a role here. If three vertices are selected, it is a `normal' Parent relationship in which the 3 vertices determine the rotation and location of the Child together. This method produces interesting effects with Vertex Keys. In EditMode, other Objects can be selected with CTRL+RMB.
CTRL+S. Shear. In EditMode this operation enables you to make selected forms `slant'. This always works via the horizontal screen axis.
UKEY. Undo. When starting Edit Mode, the original ObData block is saved and can be returned to via UKEY. Mesh Objects have better Undo, see next section.
WKEY. Specials PopupMenu. A number of tools are included in this PopupMenu as an alternative to the Edit Buttons. This makes the buttons accessible as shortcuts, e.g. EditButtons-> Subdivide is also `WKEY, 1KEY'.
SHIFT+W. Warp. Selected vertices can be bent into curves with this option. It can be used to convert a plane into a tube or even a sphere. The centre of the circle is the 3DCursor. The mid-line of the circle is determined by the horizontal dimensions of the selected vertices. When you start, everything is already bent 90 degrees. Moving the mouse up or down increases or decreases the extent to which warping is done. By zooming in/out of the 3Dwindow, you can specify the maximum degree of warping. The CTRL limiter increments warping in steps of 5 degrees.

EditMode - Mesh

This section and the following highlight peculiar EditMode Hotkeys.

CTRL+NUM+. Adds to selection all vertices connected by an edge to an already selected vertex.
CTRL+NUM-. Removes from selection all vertices of the outer ring of selected vertices.
ALT+CTRL+RMB. Faces loop select.
ALT+RMB. Edges loop select.
CKEY. If using curve deformations, this toggles the curve Cyclic mode on/off.
EKEY. Extrude Selected. "Extrude" in EditMode transforms all the selected edges to faces. If possible, the selected faces are also duplicated. Grab mode is started directly after this command is executed.
SHIFT+EKEY. Crease Subsurf edge. With "Draw Creases" enabled, pressing this key will allow you to set the crease weight. Black edges have no weight, edge-select color have full weight.
CTRL+EKEY. Mark LSCM Seam. Marks a selected edge as a "seam" for unwrapping using the LSCM mode.
FKEY. Make Edge/Face. If 2 vertices are selected, an edge is created. If 3 or 4 vertices are selected, a face is created.
SHIFT+F. Fill selected. All selected vertices that are bound by edges and form a closed polygon are filled with triangular faces. Holes are automatically taken into account. This operation is 2D; various layers of polygons must be filled in succession.
ALT+F. Beauty Fill. The edges of all the selected triangular faces are switched in such a way that equally sized faces are formed. This operation is 2D; various layers of polygons must be filled in succession. The Beauty Fill can be performed immediately after a Fill.
CTRL+F. Flip faces, selected triangular faces are paired and common edge of each pair swapped.
HKEY. Hide Selected. All selected vertices and faces are temporarily hidden.
SHIFT+H. Hide Not Selected: All non-selected vertices and faces are temporarily hidden.
ALT+H. Reveal. All temporarily hidden vertices and faces are drawn again.
ALT+J. Join faces, selected triangular faces are joined in pairs and transformed to quads

- **KKEY.** Knife tool Menu.
  - *Face Loop Select:* (**SHIFT+R**) Face loops are highlighted starting from edge under mouse pointer. **LMB** finalizes, **ESC** exits.
  - *Face Loop Cut:* (**CTRL+R**) Face loops are cut starting from edge under mouse pointer. **LMB** finalizes, **ESC** exits.
  - *Knife (exact):* (**SHIFT+K**) Mouse starts draw mode. Selected Edges are cut at intersections with mouse line. **ENTER** or **RMB** finalizes, **ESC** exits.
  - *Knife (midpoints):* (**SHIFT+K**) Mouse starts draw mode. Selected Edges intersecting with mouse line are cut in middle regardless of true intersection point. **ENTER** or **RMB** finalizes, **ESC** exits.

LKEY. Select Linked. If you start with an unselected vertex near the mouse cursor, this vertex is selected, together with all vertices that share an edge with it.
SHIFT+L. Deselect Linked. If you start with a selected vertex, this vertex is deselected, together with all vertices that share an edge with it.
CTRL+L. Select Linked Selected. Starting with all selected vertices, all vertices connected to them are selected too.
MKEY. Mirror. Opens a popup asking for the axis to mirror. 3 possible axis group are available, each of which contains three axes, for a total of nine choices. Axes can be Global (Blender Global Reference); Local (Current Object Local Reference) or View (Current View reference). Remember that mirroring, like scaling, happens with respect to the current pivot point.
ALT+M. Merges selected vertices at barycentrum or at cursor depending on selection made on pop-up.
CTRL+N. Calculate Normals Outside. All normals from selected faces are recalculated and consistently set in the same direction. An attempt is made to direct all normals `outward'.
SHIFT+CTRL+N. Calculate Normals Inside. All normals from selected faces are recalculated and consistently set in the same direction. An attempt is made to direct all normals `inward'.
ALT+S. Whereas **SHIFT+S** scales in Edit Mode as it does in Object Mode, for Edit Mode a further option exists, **ALT+S** moves each vertex in the direction of its local normal, hence effectively shrinking/fattening the mesh.
CTRL+T. Make Triangles. All selected faces are converted to triangles.
UKEY. Undo. When starting Edit Mode, the original ObData block is saved and all

subsequent changes are saved on a stack. This option enables you to restore the previous situation, one after the other.
SHIFT+U. Redo. This let you re-apply any undone changes up to the moment in which Edit Mode was entered
ALT+U. Undo Menu. This let you choose the exact point to which you want to undo changes.

- **WKEY.** Special Menu. A PopupMenu offers the following options:
  - *Subdivide:* all selected edges are split in two.
  - *Subdivide Fractal:* all selected edges are split in two and middle vertex displaced randomly.
  - *Subdivide Smooth:* all selected edges are split in two and middle vertex displaced along the normal.
  - *Merge:* as **ALT+M.**
  - *Remove Doubles:* All selected vertices closer to each other than a given threshold (See EditMode Button Window) are merged **ALT+M.**
  - *Hide:* as **HKEY.**
  - *Reveal:* as **ALT+H.**
  - *Select Swap:* Selected vertices become unselected and vice versa.
  - *Flip Normals:* Normals of selected faces are flipped.
  - *Smooth:* Vertices are moved closer one to each other, getting a smoother object.
  - *Bevel:* Faces are reduced in size and the space between edges is filled with a smoothly curving bevel of the desired order.

- **XKEY.** Erase Selected. A PopupMenu offers the following options:
  - *Vertices:* all vertices are deleted. This includes the edges and faces they form.
  - *Edges:* all edges with both vertices selected are deleted. If this `releases' certain vertices, they are deleted as well. Faces that can no longer exist as a result of this action are also deleted.
  - *Faces:* all faces with all their vertices selected are deleted. If any vertices are `released' as a result of this action, they are deleted.
  - *All:* everything is deleted.
  - *Edges and Faces:* all selected edges and faces are deleted, but the vertices remain.
  - *Only Faces:* all selected faces are deleted, but the edges and vertices remain.

YKEY. Split. This command splits the selected part of a Mesh without deleting faces. The split parts are no longer bound by edges. Use this command to control smoothing. Since the split parts have vertices at the same position, selection with **LKEY** is recommended.

EditMode - Curve

CKEY. Set the selected curves to cyclic or turn cyclic off. An individual curve is selected if at least one of the vertices is selected.
EKEY. Extrude Curve. A vertex is added to the selected end of the curves. Grab mode is started immediately after this command is executed.
FKEY. Add segment. A segment is added between two selected vertices at the end of two curves. These two curves are combined into one curve.
HKEY. Toggle Handle align/free. Toggles the selected Bezier handles between free or aligned.
SHIFT+H. Set Handle auto. The selected Bezier handles are converted to auto type.
CTRL+H. Calculate Handles. The selected Bezier curves are calculated and all handles are assigned a type.
LKEY. Select Linked. If you start with an non-selected vertex near the mouse cursor, this vertex is selected together with all the vertices of the same curve.
SHIFT+L. Deselect Linked. If you start with a selected vertex, it is deselected together with all the vertices of the same curve.
MKEY. Mirror. Mirror selected control points exactly as for vertices in a Mesh.
TKEY. Tilt mode. Specify an extra axis rotation, i.e. the tilt, for each vertex in a 3D curve.
ALT+T. Clear Tilt. Set all axis rotations of the selected vertices to zero.
VKEY. Vector Handle. The selected Bezier handles are converted to vector type.

- **WKEY.** The special menu for curves appears:
  - Subdivide. Subdivide the selected vertices.
  - Switch direction. The direction of the selected curves is reversed. This is mainly for Curves that are used as paths!

- **XKEY.** Erase Selected. A PopupMenu offers the following options:
  - *Selected:* all selected vertices are deleted.
  - *Segment:* a curve segment is deleted. This only works for single segments. Curves can be split in two using this option. Or use this option to specify the cyclic position within a cyclic curve.
  - *All:* delete everything.

EditMode - Metaball

MKEY. Mirror. Mirror selected control points exactly as for vertices in a Mesh.

EditMode - Surface

CKEY. Toggle Cyclic menu. A PopupMenu asks if selected surfaces in the `U' or the `V' direction must be cyclic. If they were already cyclic, this mode is turned off.
EKEY. Extrude Selected. This makes surfaces of all the selected curves, if possible. Only the edges of surfaces or loose curves are candidates for this operation. Grab mode is started immediately after this command is completed.
FKEY. Add segment. A segment is added between two selected vertices at the ends of two curves. These two curves are combined into 1 curve.
LKEY. Select Linked. If you start with an non-selected vertex near the mouse cursor, this vertex is selected together with all the vertices of the same curve or surface.
SHIFT+L. Deselect Linked. If you start with a selected vertex, this vertex is deselected together with all vertices of the same curve or surface.
MKEY. Mirror. Mirror selected control points exactly as for vertices in a Mesh.
SHIFT+R. Select Row. Starting with the last selected vertex, a complete row of vertices is selected in the `U' or `V' direction. Selecting *Select Row* a second time with the same vertex switches the `U' or `V' selection.

- **WKEY.** The special menu for surfaces appears:
  - Subdivide. Subdivide the selected vertices
  - Switch direction. This will switch the normals of the selected parts.
  - Mirror. Mirrors the selected vertices

- **XKEY.** Erase Selected. A PopupMenu offers the following choices:
  - *Selected:* all selected vertices are deleted.
  - *All:* delete everything.

VertexPaint Hotkeys

SHIFT+K. All vertex colours are erased; they are changed to the current drawing colour.
UKEY. Undo. This undo is `real'. Pressing Undo twice redoes the undone.
WKEY. Shared Vertexcol: The colours of all faces that share vertices are blended.

EditMode - Font

In Text Edit Mode most hotkeys are disabled, to allow text entering.

RIGHTARROW. Move text cursor 1 position forward
SHIFT+RIGHTARROW. Move text cursor to the end of the line.
LEFTARROW. Move text cursor 1 position backwards.
SHIFT+LEFTARROW. Move text cursor to the start of the line
DOWNARROW. Move text cursor 1 line forward
SHIFT+DOWNARROW. Move text cursor to the end of the text.
UPARROW. Move text cursor 1 line back.
SHIFT+UPARROW. Move text cursor to the beginning of the text
ALT+U. Reload Original Data (undo). When EditMode is started, the original text is saved. You can restore this original text with this option.
ALT+V. Paste text. The text file /tmp/.cutbuffer is inserted at the cursor location.

UV Editor Hotkeys

EKEY. LSCM Unwrapping. Launches LSCM unwrapping on the faces visible in the UV

editor.
PKEY. Pin selected vertices. Pinned vertices will stay in place on the UV editor when executing an LSCM unwrap.
ALT+PKEY. Un-Pin selected vertices. Pinned vertices will stay in place on the UV editor when executing an LSCM unwrap.

EdgeSelect Hotkeys

ALT+CLICK. Selects an Edge Loop.

FaceSelect Hotkeys

ALT+CLICK. Selects a Face Loop.
TAB. Switches to EditMode, selections made here will show up when switching back to FaceSelectMode with TAB.
FKEY. With multiple, co-planar faces selected, this key will merge them into one "FGon" so long as they remain co-planar (flat to each other).
LKEY. Select Linked UVs. To ease selection of face groups, Select Linked in UV Face Select Mode will now select all linked faces, if no seam divides them.
RKEY. Calls a menu allowing to rotate the UV coordinates or the VertexCol.

- **UKEY.** Calls the UV Calculation menu. The following modes can the applied to the selected faces:
  - *Cube:* Cubical mapping, a number button asks for the cubemap size
  - *Cylinder:* Cylindrical mapping, calculated from the center of the selected faces
  - *Sphere:* Spherical mapping, calculated from the center of the selected faces
  - *Bounds to x:* UV coordinates are calculated from the actual view, then scaled to a boundbox of 64 or 128 pixels in square
  - *Standard x:* Each face gets default square UV coordinates
  - *From Window:* The UV coordinates are calculated using the projection as displayed in the 3DWindow

Render Window Hotkeys (to be written)

To be written (if someone could it would be very useful! : there's no place about it on the internet (or I didn't search enough))

JKEY. Changes the image output. You have two slots in which to render. Very useful when you want to see what a specific change did to the image.
AKEY. Toggles display of alpha channel. The alpha channel of a picture determines it's transparency: Black areas are fully transparent, white areas are fully opaque and grey means semi-transparent. This can be useful when rendered images are arranged in layers, or used in applications which support alpha channels in images. To save images with alpha channel, make sure that the RGBA button in the render panel (F10) is enabled. Also, not all image formats support alpha channels, i.e. TGA and PNG do, but JPG does not. Note: Any background texturing which is done via the world panel (F8) will have an alpha value of 0, meaning it will be transparent. However, the world background will still be rendered correctly on (opaque) surfaces as reflections (i.e. mirrors) - this must be taken into account when later composing rendered images with a different background.
ZKEY. Toggle Zoom (2x). This will zoom the rendered image. The mouse can still be used to scroll around the zoomed image.

Next Page: Beginning Modeling Final Project
Previous Page: Every Material Known to Man

Beginning Modeling Final Project

Next Page: Beginning Lighting
Previous Page: Hot Keys

Now that you've gotten the hang of 3D modeling, it's important to get some community feedback on your progress. Don't be an ass and skip this part, or you'll regret it later. Basically this will help you track your progress and give you something that you'll be working on over a long term and something you'll be proud of.

- First, you need to come up with a project idea. You can choose your own modeling project, or choose one from the list below.
- Second, you need to create a model of your idea. Spend a couple of hours on it, and give it some details.
- Third, once you believe you've come far enough with the model, post it in the Works In Progress (http://www.elysiun.com/forum/viewforum.php?f=12) forum on elysiun.com (you will have to create an account if you haven't already). Post several screenshots of your model from within the Blender (note: creating screenshots is outside the scope of this wikibook, though see note lower down the page). You can post whatever subject and message with your posting that you would like, or you can use this suggested subject and message:

  Subject: Beginning Modeling Final Project - <project name>

  Message:

  > Hello. I'm new to the Blender and have completed the tutorials on Beginning Modeling found at the online wikibook http://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro. Before I continue with the wikibook, I need some feedback from the community about my skill level so that I need to determine my progress so far. Please evaluate my model and these screenshots for the following:
  >
  > 1) I demonstrate an ability to navigate in 3D space.
  >
  > 2) I demonstrate an understanding of the basic parts of a 3D model, such vertices, edges, and faces.
  >
  > 3) I demonstrate an ability to create form in three dimensions.
  >
  > Please assist me with any feedback on my model, keeping in mind that I am an absolute beginner still. I appreciate your help.

- Wait for feedback. It usually comes very quickly. If you have any questions about feedback that you are given, don't be afraid to ask your questions in the forum.
- When you and others that have viewed your work feel that you are ready, save your model in some place you can get back to easily. You will continue working on this project once you've learned some new skills.
- Move on to the next page.

(BTW. in Windows and/or maybe other OS, to take a screenshot press 'PrtScn' (PrintScreen). It will copy the screen to clipboard for you to paste in your fav graphic app. This may not work in other OSs but try anyway. You can also create Blender screenshot directly from Blender using menu **File>Dump 3DView...** or **File>Dump Screen...**)

(In linux under the KDE DE I use ksnapshot, check under the graphics tab and see if you have it. If not it should be just a google search away :) gl and happy blendering)

(On Mac OS X, press Command (Apple) + Shift + 3 to do a full screen capture)

- List of ideas:
- A Computer and keyboard
- A fishing rod
- A train engine
- A skyscraper
- A robot
- A Tank (real or made up)
- An airplane

- A truck or car
- Household appliances

Next Page: Beginning Lighting
Previous Page: Hot Keys

Beginning Lighting

Next Page: Adding Lamps
Previous Page: Beginning Modeling Final Project

Lighting, you say? Psshh. Just throw up one light source and let her run, right?

Wrong. Lighting is probably the most underestimated part of a scene by new 3D artists. By the following tutorials, you will gain knowledge of the technical use of lights in your scenes.

Next Page: Adding Lamps
Previous Page: Beginning Modeling Final Project

Adding Lamps

Next Page: Shadows
Previous Page: Beginning Lighting

You can quickly add several different types of lights to your blender scene

SPACE > Add > Lamp > Spot

A light will appear in the location of the 3D cursor. You can move a light just like any other object.

If you want to quickly light a scene just for illumination, not for a specific look, add four lamps around your subject. If you are interested in experimenting with a lighting arrangement, a nice quick way to experiment is to create a Monkey in the scene to test with.

SPACE > Add > Mesh > Monkey

The monkey is just as good of a test subject as a human face, so give it a try. You can throw various materials on the monkey and try different textures too. Don't bad mouth the monkey, she is really useful.

Explaining the Different Lights:

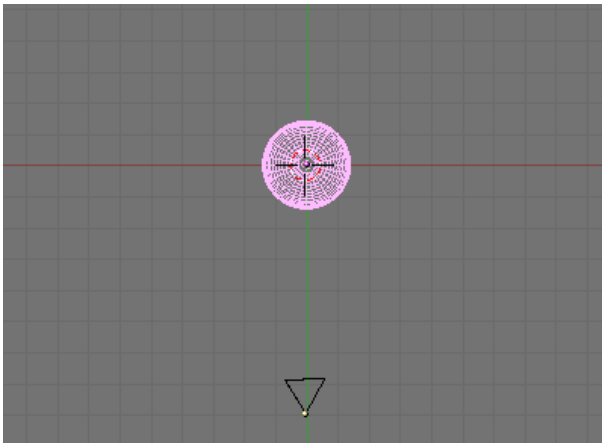
- **Lamp**: Simple light source, that shines in all directions. Ideal for using as a background light or simply if you don't want the Lamp to cast shadows
- **Sun**: a directional light source with parallel rays. As the name says: used to emulate sunlight
- **Spot**: another directional light source with ray going from one center, forming a cone. Also the only light that naturally casts shadows.
- **Hemi**: 180° constant light source. For special purposes.
- **Area**: Rectangular Area that casts directional light. Used for: ---neon tubes of course ;-)

Creating a basic scene with basic lighting

- This addition is simply a way to apply what you know about lights and to discover a few settings like colors or creating simple shadows. The purpose here is to create a basic scene with a sphere over a plane, nicely lighted. You should already know the basics of blender (creating a mesh, moving and rotating it, rendering).

Creating the scene

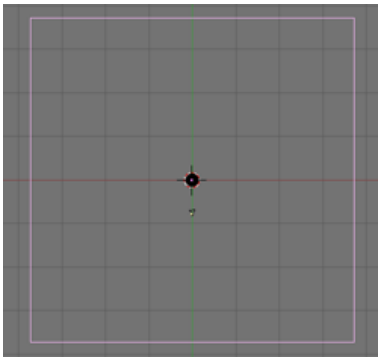
- Okay, let's start! Open a new file. Add a **UVsphere** of 32 rings and 32 segments. Exit **EditMode**. Leave it in the center of the scene.



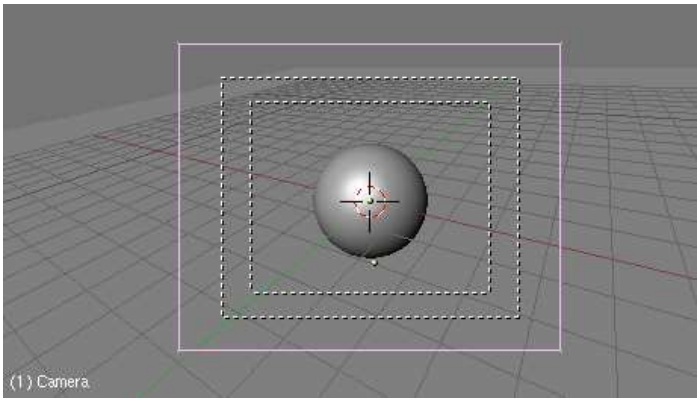
Trick : Go in the Editing buttons  , and push **Set Smooth** 

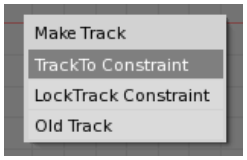
so the sphere will render as a nicely smoothed sphere.

- You should already have a plane in the basic scene, otherwise add a **Plane**. Have it in **ObjectMode** and move it just under the sphere. Scale it so it is very big. The ideal would that we can't see borders with the camera.

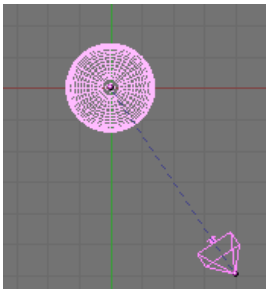


- Then we will move the **Camera**. Grab it and rotate it so it looks at the sphere from top and a bit from the right. You can have an idea of what it sees pressing **Num0** to have a **CameraView**.



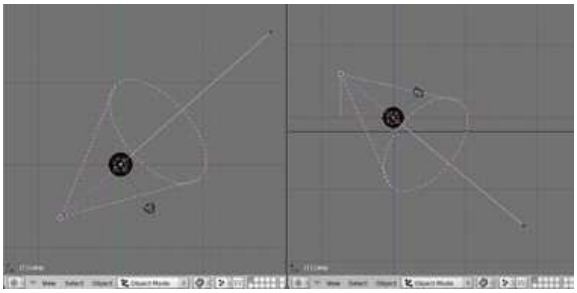
Trick : First click on the Camera, then on the Sphere holding **Shift**(the order is very important). Do a **Ctrl+T** and select **TrackTo Constraint** 

. The camera will be looking at the center of the sphere... You can then move either the camera or the sphere

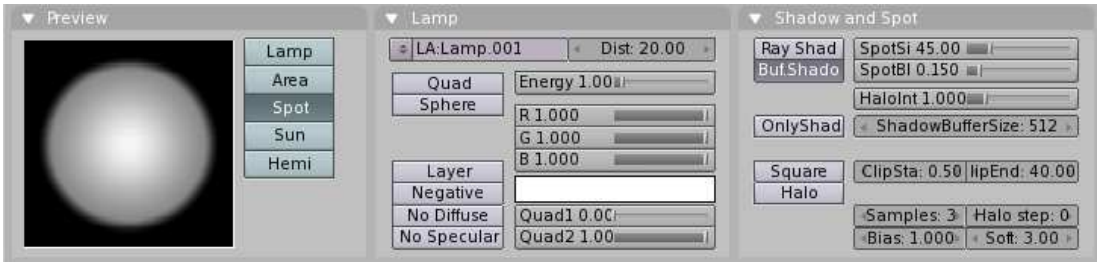
and the camera will still point the sphere. 

Adding the lights

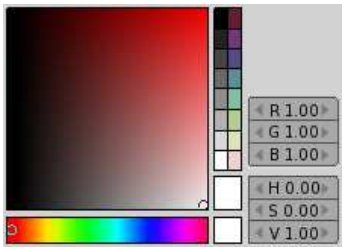
- Okay, we have a pretty beautiful scene, isn't it?... Well, it isn't! But it is enough to add some lights... Here I will describe a basic lighting scene I use as a default for fast renderings. I picked it out from another tutorial, you can find the link at the end of this page.
- So we will add our so awaited lights. **Add>Lamp>Spot**. Yes, we will first use the Spot light. We can see it as a projector. This is the only light casting shadows. Place it so it is upper and on the left of the sphere. Rotate it so it looks at the Sphere (you can use a **Trick** I gave you before to have the light looking at the sphere).

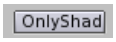


- Okay, let's see what we can tune with the spot light. Having the lamp selected, go in the **Lamp** settings  . You will see these buttons.

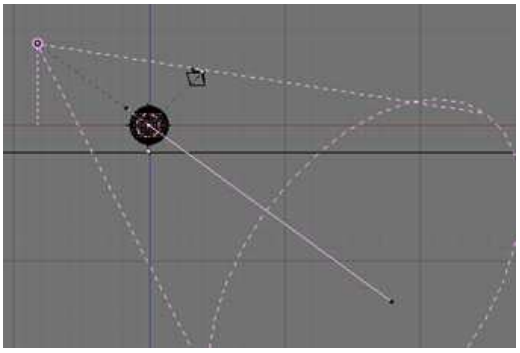


- Yeah, really lots of options. Don't worry, I'll explain the basic ones.
  - **Dist** : Sets the maximum distance the light can reach. Increase it so the lighting cone really goes behind the sphere. I set it to **Dist:40**.
  - **Energy** : This is the force of the light. You can leave it at **Energy:1**.
  - **RGB** : You can change the colour of the light. Click on the colour and a little window will appear to select the colour you want. Leave it **white**.

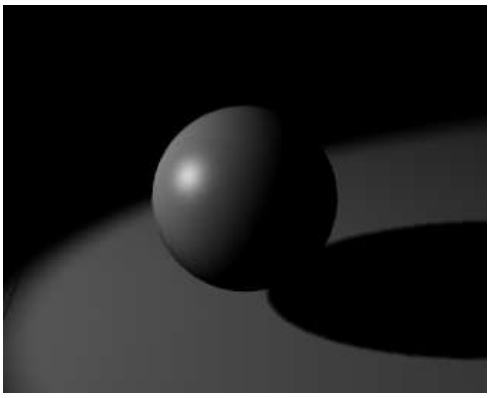


-
  - **Buf.Shadow** : Enables the light to cast shadows, leave it **pushed**. 
  - **OnlyShadows** : This light only creates shadows, without casting light (yeah quite unrealistic, but it can be useful). Leave it **unpushed**. 
  - **SpotSi** : This is the angle of your cone, in degrees. Leave to **SpotSi:45**.
  - **SpotBl** : This smoothes the circle cast by the light. We will smooth it so that it looks better. Set it to **SpotBl:0.4**.

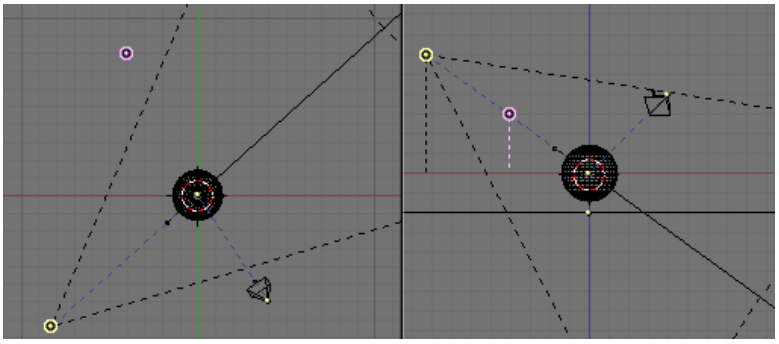
- **ClipSta** and **ClipEnd** : This is the distance from the light between which shadows will be cast. You can see the "line of effects" in the 3D windows when you change these. Set them so the line starts before the sphere and ends far (well, a bit !) behind the plane. You should obtain something like this.



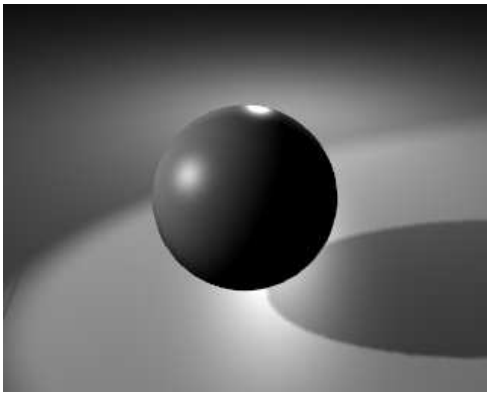
- Now, we have set our Spot light. This light will be our side light and shadowing light. You can make a fast preview pressing **F12**. You can see your so nice shadow. But there isn't enough light... Let's add some more!



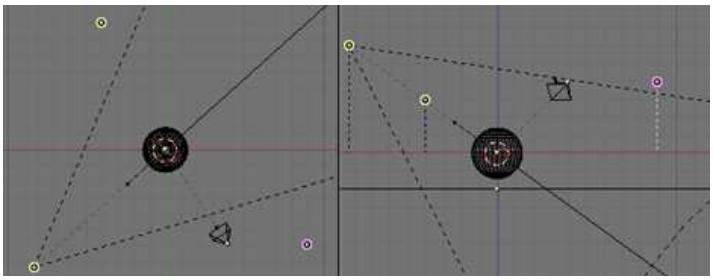
- Time to add a second light! **Add>Lamp>Lamp**. This time, we will create a basic lamp. This is like a point which emits light in every direction from that point. You should place it at the opposite of the camera, quite at the same height. This light will be used to better see the form of the sphere and to add a sort of general lighting of the scene.



- Go in the Lamp buttons, and set it to a higher energy like **Energy:1.25**. You can make a quick render to see how much this light is important to a scene.

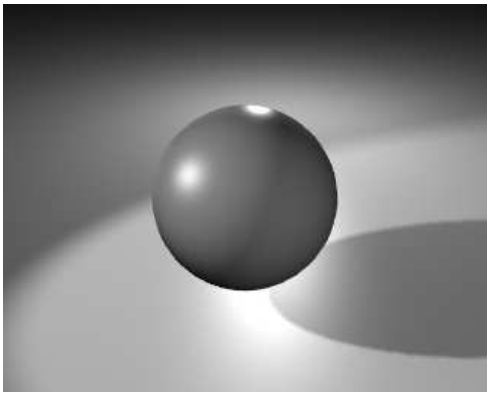


- Now, we will add a second basic lamp. **Add>Lamp>Lamp** and we will place it just behind the camera point of view, a bit moved at the opposite of the Spot light. This third light will slightly light the dark parts of the sphere.



- Decrease the energy of this light, as it is only supposed to fake the reflections of the environment. I set it to **Energy:0.8**. Another little trick, as this is not supposed to be a direct light, there shouldn't be a little white glow called **Specular** on the sphere coming from this light. Push the **No Specular** button. 

- Okay, it's time for the final rendering. Of course, this is a really basic lighting set you can use for rendering a simple mesh; but for more complicated scenes, lights can come from other places, with other colors, etc... Thus we didn't use the **Sun**, **Hemi** and **Area** lights, which are a bit more complicated I think.



For a more in-depth tutorial, here is a tutorial from the Blender Documentation (http://download.blender.org/documentation/html/x4029.html) , which has been a great source of help for me.

Outdoor lighting

- Here you will use a **Sun** in conjunction with a **Spot** light and some little **Lamp**s.

Next Page: Noob to Pro/Shadows
Previous Page: Beginning Lighting

Shadows

Next Page: Render Settings
Previous Page: Adding Lamps

Creating Soft Shadows

Next Page: Render Settings
Previous Page: Adding Lamps

Render Settings

Next Page: Output Format Options

Previous Page: Shadows

Render Settings

The render settings control various options related to the output of rendered, or full quality images. Rendering an image will calculate effects not displayed in the editing environment (due to their complexity), and therefore takes a larger amount of time to produce an image.

Next Page: Output Format Options
Previous Page: Shadows

Output Format Options

Next Page: Blender 3D: Noob to Pro/OSA
Previous Page: Render Settings

Size

When you render a scene in Blender, by default there will open a small window showing your image. If you want to publish your picture, you may want to render it a bit bigger. To do this, you have to open the Scene context (F10) and locate the Format sub-context (usually on the right).

You will see some settings there. Let's go through their meanings:

- SizeX: This parameter sets the width of the image in pixels.
- SizeY: This parameter sets the height of the image in pixels.
- AspX, AspY: These parameters specify the aspect ratio of the pixels. By default, this is 100:100, because a pixel on a computer screen has equal width and height. These settings can be used for screens whose pixels don't have equal width and height. For example, on a PAL system one pixel's width/height ratio is 54:51, which you can select there easily. Notice that these parameters don't change the size of the image.

There is one more thing changing the render size: In the Render sub-context (usually in the middle) there are 4 buttons: 100%, 75%, 50%, 25%. If you have chosen SizeX = 640, SizeY = 480 and you clicked 50%, your image will be 320 pixels wide and 240 pixels high.

WARNING: I have encountered some crashes when I changed those parameters. If you have problems too, locate the Output sub-context (usually on the left) and select the option DispView instead of DispWin. This renders the image "into" Blender instead of making a new window.

Output Formats

Images:

- BMP: uncompressed.
- Ftype:
- Iris:
- Iris + Zbuffer:
- PNG: open, lossles compression, alpha channel.
- Jpeg: default format. Lossy compression.
- HamX: extremely compact but only for the "Play" option
- Targa:
- Targa Raw: uncompressed Targa.

Video:

- AVI Codec: saves an AVI with a compression codec. Once selected a pop up menu will appear giving options as to what codec you want.
- AVI Jpeg: saves an AVI as Jpeg images. Compressed but lossy.
- AVI Raw: saves an AVI with uncompressed frames.
- QuickTime:

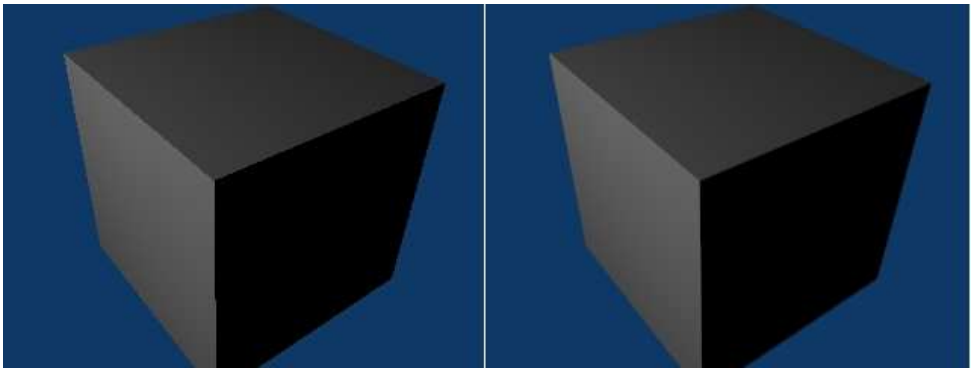
Next Page: Blender 3D: Noob to Pro/OSA
Previous Page: Render Settings

OSA

Next Page: Looking All Around - Panorama Settings
Previous Page: Output Format Options -- Back to Index

OSA stands for *oversampling*, also known as *anti-aliasing*. It prevents "jaggies" or *aliasing* as its called. This is when you have a diagonal change of colour which results in rough edges. Remember drawing a diagonal line in Paint? To overcome this hinderance of square pixels a technique called anti-aliasing or oversampling is used. What it does is it blends the colours around the rough edge to create a smooth, but defined edge. One way of doing this is to create the image twice as large, then scaling it down - oversampling. Blender can do this for you if you select an OSA rate. Remember this will take much longer, but results in better renders, so use this for the final product, not while testing. In some cases the scene can seem blurred due to oversampled textures, try changing the OSA setting, or oversampling yourself.

Here's a quick illustration of how OSA changes a render (look at the edges):



The image on the left has no OSA. The one on the right has 16x (the maximum amount allowed by Blender.

Next Page: Looking All Around - Panorama Settings
Previous Page: Output Format Options -- Back to Index

Looking All Around - Panorama Settings

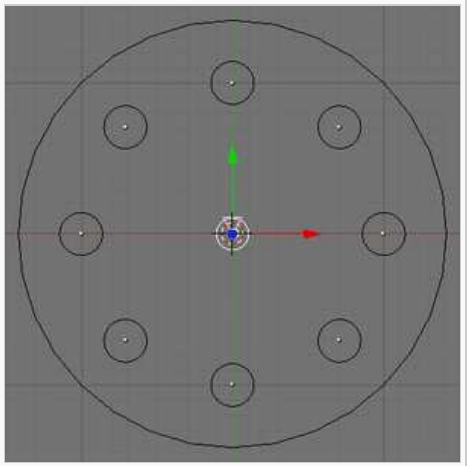
Next Page: Blender 3D: Noob to Pro/Other Important Render Options
Previous Page: Blender 3D: Noob to Pro/OSA

Panoramic Renderings

Building the Example Scene

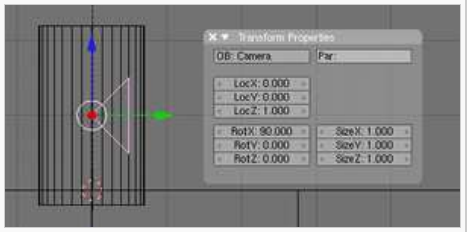
Ever wondered how that cool looking 360 degree panoramas that you see on some websites are made? Well, I don't know either, so someone else will have to tell about that. What I found out though is how to make Blender output a 360 degree panoramic image - quite probably this at least is the first step when making something like on those websites.

Now, let's try to progress like in a tutorial, as in the other pages of this Wikibook. So, fire up Blender, and look at the well known initial box scene. Or rather, change it somehow. Assuming you have read the book in order, it shouldn't be hard to get something like in the picture.



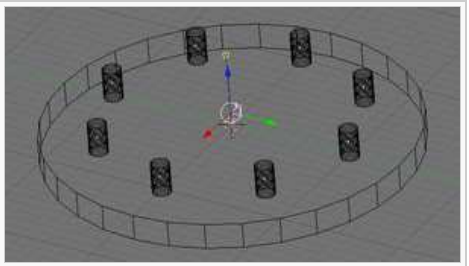
The example scene, see from above

I placed the camera and the light both above the origin, deleted the cube, and added 8 cylinders, all around the origin. One way to do this is to place one cylinder, then duplicate and rotate it around the origin. For this to work, set the rotation center to the 3D-cursor (with the Pivot button), and position the 3D cursor at the origin (use **SHIFT+S** to make the 3D cursor snap to the grid).Place the first cylinder using **GKEY** and then holding **CTRL** while moving the mouse. Change to object mode, then use **SHIFT+D** to duplicate it. Next, press **RKEY** to rotate it, then hold **CTRL** while rotating it by 45 degrees. Repeat the same for the remaining cylinders. Use **G** to move the light. Use **NKEY** to enter the camera values like shown, so it looks parallel to the ground.



The example camera, seen from the side

Speaking of ground, lets also add a ground. I added another cylinder, below the other ones, as shown. It's easy to add by switching to front view (**NUM1**), duplicating a cylinder (**SHIFT+DKEY**), and then moving it holding **CTRL** pressed again, so it snaps to the grid. To scale it, use the **SKEY** , but hit **SHIFT+ZKEY** to lock the z scaling, and hold **CTRL** pressed while scaling in x and y direction so you can snap to the exact size you want.



The complete scene with a ground cylinder

Now, to have a complete scene, we need some colors. I made the ground cylinder green, the 8 example cylinders gray, and also added a noise normal to the ground. This doesn't really matter here though, just needed some example scene for the 360° camera.

Something more interesting is the sky texture, because Blender's sky can be made to seamlessly wrap around with 360 degree. Belows is a screenshot of the sky settings I used. To get there, click on the Shading button, then select World.

To change the texture, click the textures button or press **F6**, then add a new texture with the small button with two arrows, and select the type (e.g. "Marble" or "Clouds") instead of "None" for the texture. Go back to the World pane, and enable the Real and Blend buttons. And don't forget to use nice colors, I used blue and white. Now it looks a bit like white clouds

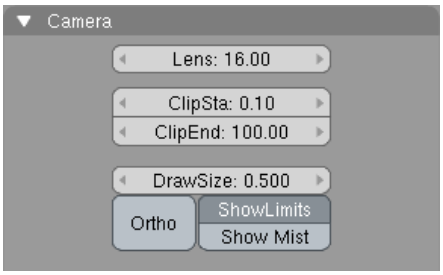
in a blue sky.

[Maybe should add here more details? What is important so you don't get a seam at the 0 - 360 degree point? How do you map pre-made sky textures?]

And we are already done, this is the example scene. All that is left to do is to render it as a panorama.

Panoramic Rendering

Go to the render settings (**F10**), and click the Pano button. Change the Xparts value to 4. The value in Xparts will tell how many times the camera will 'turn' horizontally when rendering. The Yparts value would do the same vertically. Each part (Xparts or Yparts) will render the size of the whole image you set. To make things easier, make the output image size quadratic and set both X and Y Aspect to 100. E.g. make your image size 600x600, but not 800x600, otherwise the following will not work.



If you want a seemless 360 degree view, it is important to know how many degrees one single image spans. For example, if you know one image is 90 degree, then you can set Xparts to 4, and the result will be a single panoramic picture, and with the right panorama viewer, you can spin around in it endlessly.

But, how do you make a single picture 90 degree? The angle a picture spans is called field of view (FOV). And it is a property of the camera. To change it, select your camera, then click the edit button (or hit **F9**). Make sure you haven't set your camera to orthographic (the "Ortho" button), since FOV only exists with a normal camera.

Unfortunately, you can't directly enter the FOV in Blender's camera settings (as of version 2.37) - instead there is just a parameter called **Lens**. Type in 16 to get a FOV of 90 degree. Now hit **F12** to render our test image.

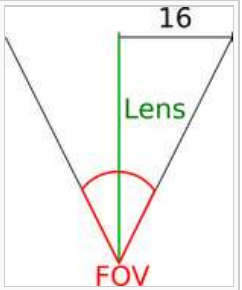
If you want, you can also try it out with any other scene. Place the camera somewhere in the middle with a good view all around, set the camera's **Lens** to 16, and go into the render settings (**F10**). Set **Xparts** to 4, enable **Pano**, and render. You should get an image 4 times as wide as a normal image, and the left and right edges should fit together seemlessly. Also note that Blender doesn't know if it should apply the FOV as horizontal or vertical viewing angle, but we always want it to be the horizontal angle. Therefore just make the single images of the panorama quadratic, as mentioned above - then horizontal and vertical FOV is the same and Blender gets it right.

This is the resulting panoramic image:

One problem remains, the image will be quite distorted. This is because a FOV of 90 degree is too high. Let's try with 45 degree and 8 XParts instead. As you will have noticed when playing around with the Lens parameter before, setting Lens to 8 instead of 16 will not achieve this, it will make the FOV even bigger. Setting it to 32 looks better, but definitely is not 45 degree. The picture to the right explains how FOV and Lens relate to each other:


Panorama with 4 parts.

The camera is at the bottom, the red angle is our desired FOV, and the length of the green line is what the Lens parameter represents in Blender. Therefore, the formula to calulcate Lens when we know FOV:



The **Lens** parameter of the Blender camera.

Lens = 16 / tan(**FOV** / 2)

For a FOV of 45 degree, we therefore get a Lens of 38.627. Now, write that into the Lens field in the camera settings, and render with 8 XParts. This time it will render an output image which is 8 times as wide as a normal output image, and again it will be a seamless 360 degree all around panorama of your scene. And this time with less distortion. With the above formula, you can now use any number of XParts you want - just divide 360 by the number of parts, and calulate the Lens parameter for the resulting FOV. Note that you also need to take the Aspect settings into account, unless you set it to 100 and create a quadratic output, as we did.

Some Lens settings:

- 4 parts: 16
- 8 parts: 38.627
- 16 parts: 80.437

The output with 8 parts can be seen below.


Panorama with 8 parts.

As always, feel free to modify this page in any way you want or add feedback to the Talk page.

Next Page: Blender 3D: Noob to Pro/Other Important Render Options
Previous Page: Blender 3D: Noob to Pro/OSA

Other Important Render Options

Next Page: Basic Animation
Previous Page: Looking All Around - Panorama Settings

Rendering by Parts (Bucket Rendering)

Images can be rendered in pieces or layers rather than all at one time. Your computer will only need to compute smaller bits of information thus using less memory. By changing the Xpart and Ypart values (up to 8 each since Blender can't support more than 64 [8x8] parts) in the Render panel of the Scene context(F10,) you can divide your image into an invisible grid. The pieces will layer one at a time until they are whole.

Edge Renderings

Blender has an option of adding a border-an edge to objects (like in cartoons). To do that you need to go to the render buttons (F10) and then, under "output" change the edge setting: set the "Eint" to a value of about 100 and check the "edge" icon to enable edge rendering. This would give you edge line at the edge of each polygon. To prevent it from applying on all polygons the same way change to "unified render" under the format tab. Then, in the edge settings change the "antishift" value, it will decrease itself from the fint value when the line is between identical materials. By also checking the "all" icon you tell the render to apply the edge rendering on transparent materials as well.

Multi Thread Rendering

Blender Has the option to render with 'Multiple threads' meaning it can render 2 parts at once at the same speed as rendering one. This is done by sending 2 Render parts to the cpu to process at the same time,if your CPU can handle it. The Option is found in render buttons (F10) as the button "Threads" under the Backbuf filepath and button

Next Page: Basic Animation
Previous Page: Looking All Around - Panorama Settings

Basic Animation

Next page: Blender 3D: Noob to Pro/Basic Animation/Bounce
Previous page: Other Important Render Options

Animating, in principle, isn't that hard. These steps show you how to make a simple keyframe animation:

1. It is a good idea to switch into the animation screen using the screen selection drop-down on the main menu (change it to **SCR:1-Animation**).
2. Go to the starting frame, using the arrow keys on your keyboard (Shift+Left Arrow goes to Start Frame, Shift+Right Arrow goes to End Frame, the Up and Down arrows Skip 10 Frames, the Right and Left arrows skip 1 Frame)
3. Go into the correct mode. To animate solid objects, go into Object Mode. To animate bones, go into Pose Mode.
4. Place the object or bone at the desired starting location and/or rotation.
5. Press the **IKEY** and you will see a menu of the different properties which you may animate. Choose the most appropriate one, based on the properties you wish to change in the next keyframe. For example, if you wish to change the location, rotation *and* size of the object, select LocRotSize. You have now made the first *keyframe*, which is a frame of animation which you have personally set up.
6. Now go to the frame of the next keyframe you wish to define, and put everything in the place you want for that frame. Remember, if you have a frame rate setting of 25 frames per second (fps) then if you want to make the next keyframe one second later in the animation, you need to go to frame 25 to make the keyframe. Then press the **IKEY** again, and select the correct option again, depending on the changes you've made. (All in-between frames are automatically made to *interpolate* between the two neighboring keyframes).
7. Repeat the previous step for each keyframe of the animation.
8. If you want more control over the transition between the keyframes, use the IPO-window (if you followed step 1, it should already be open on the right of screen). Select the object you have keyframed and the IPO-window will display its animation curves (one curve for each of the properties you selected). You can select and edit the curves using the normal Blender controls. Use the curve menu for more options, such as making the curves actually curve instead of being straight lines.

For more information, see the Blender userguide (http://download.blender.org/documentation/htmlI/)

Next page: Blender 3D: Noob to Pro/Basic Animation/Bounce
Previous page: Other Important Render Options

Particle Systems

Next page: Making Fire
Previous page: Blender 3D: Noob to Pro/Basic Animation/Bounce

Particle Systems are used to simulate large amounts of small moving objects, creating phenomena of higher order like fire, dust, clouds, smoke, or - using a trick - fur, grass and other strand based objects. Due to the nature of particle systems, that is, the large amount of objects, the user interaction is limited to controlling global parameters, like direction of particle movement, applied forces, obstacle definition, randomness, particle life and so on, but doesn't allow access to single particles. Though it is possible in other software packages (and useful in certain situations), Blender doesn't provide methods to control single particles. The amount of control is limited to statistic values, and the resulting effect can only be controlled by those.

For the most basic particle system in Blender you need one Mesh Object - the emitter. Emitters must be mesh objects and can emit particles from their vertices' position, and also from faces. Any Mesh Object is changed into an emitter via the Object panel (accessible with the F7 key), by applying the Particles Effect in the Effect Tab. Once activated, the object itself becomes invisible to the renderer, and only its particles are visible. By default, the Material renders as type Halo rather than as a solid object. To see the particles moving, crank up the "Norm" value and hit "ALT+A" in a 3D viewport.

The direction of emmitance is controlled by a couple of values:

- **Norm**: a value that takes into consideration the vertex/face normal and sends particles in this direction (if the value is greater than zero)
- **X**, **Y**, **Z**: Those values can give particles speed and direction along the objects local axes. again, values greater than zero are required. Can be mixed with Norm.
- **External Forces**: particles can be further affected by simulated forces like wind and gravity. Their values can be positive or negative.

Next page: Making Fire
Previous page: Basic Animation

Making Fire

Next Page < Blender 3D: Noob to Pro/Furry
Previous Page < Blender 3D: Noob to Pro/Particle Systems

Using Blender's Particle System to Create Simple Smoke and Fire

(2.41)

The simplest way to think about the particle system is that any mesh object can be made to emit particles. By default, the particles are emitted from the object's vertices in an invisible ray.

You can specify the colour and transparency of the particles, the speed and direction in which they travel, and their size and range of travel. These values are particle "attributes". By combining different attributes, you can make particles look like bullets, Jedi lightsabers, smoke plumes, or the flames of a fire.

The following tutorial is not the only way you can create such effects, and expert blender users can probably do it much more efficiently and economically. It is a simple effect, intended to let you learn some of the features of basic particle animation.

Some screenshots would be helpfull for this tutorial. Add if you can

What You Need to Know before You Start

In order to complete this tutorial, you must...

- Understand how the Blender 3D interface works so that you can select objects and manipulate them in 3d space.
- Know how to select options and change numeric values.
- Be able to create and illuminate a simple mesh with lights.
- Be able to adjust the camera.
- Know how to configure the render and animation settings.

Setting Up the Workspace

Start from the default blender configuration and set up your view as a 2-panel display of the main 3D view window, and a smaller panel underneath it for the buttons window.

1. Create a mesh of a 3-division icosphere in the center of the 3d window. This icosphere will be our first "particle emitter".
2. Add a couple of area lamps at very low energy (about 0.075). Arrange the lamps on either side of the icosphere.
3. Press the F10 key to display the render buttons window.
4. Set up the render options to give you a quick, low-resolution render as follows:
    1. set OSA, SHADOW, ENV MAP, and RAY off (not highlighted)
    2. set image size to 50%
5. Now press F5 and then click materials, now under the “Shaders” tab make sure halo is selected.(edit, should be tried if below does not work)
6. Press the F12 key to make a quick render, proving that you can see your icosphere centered in the view. If not, adjust the camera direction so that it points right at the icosphere as follows:
    1. In the view window, press KP5 to toggle to grid view (grid showing).
    2. In the view window, press KP1 to toggle to front view.
    3. In the view window, click on the camera to highlight and select it.
    4. In the buttons window, press the F9 key to display the editing buttons.
    5. Click on the Show Limits button. The camera now has a direction line, showing where it is pointed.
    6. Press the R (rotate) key to select the camera for rotation and rotate it with the mouse until the direction line runs through the centre of the icosphere.
    7. Press the Enter key to confirm the new direction and exit rotation mode.
    8. In the view window, press KP7 to toggle to top view.
    9. Repeat the alignment described in steps 6 and 7.
    10. Press KP0 to return to camera view mode and press F12 to repeat the quick render. The icosphere should now be dead centre in the image.

(Note - you can also use the rotate widget to point the camera, but I'll leave that for a different tutorial).

Why not do it now? See that depressed finger button right under the viewport? (don't worry it'll cheer up!) right next to it are three essential time saving buttons. Going from right to left are the Move, Rotate, and Resize widgets. These appear at the 3D pointer location and affect the currently highlighted objects. left click on a colored axis to perform the desired function solely along that axis. Usefulness = 11 out of 10

Making a Smoke Plume

Smoke is easiest to model, and learning it first will give you some insight into the particle system. Begin by making a simple particle emission that you can see in the view window as follows:

Simple Particle Emission

1. RMB on the icosphere to select it
2. In buttons window press F7 to display Obj. buttons
3. Click the "Physics" Tab and in the second pane labeled "particles" hit "NEW"
4. The **Particles** and **Particle Motion** tabs are displayed (** It is possible to *split* the panels so that you can see both at once. LMB on either the **Particle** or **Particle Motion** label and drag to the side. The panels will seperate. ** This is an optional step and not necessary to complete the tutorial)

Before going further, Press the F12 key to make a quick render of the current state of the object. What you should see at this point is nothing. Your icosphere appears to have vanished. This is because you have now declared all its vertices to be nothing more than particle emitters. Also, the particles have no visible attributes at this point, so there is nothing for the renderer to see.

LMB on the **Particles** tab (if you haven't split the panels)

Make the following two changes to the particle system:

1. Click on the Static button.
2. Click on the "Particle motion" tab.
3. Change the value of Norm from 0.000 to 0.100 (Hint: click once on the right arrow symbol in the box).

Notice that the view window now shows a stream of particles (You MUST be in OBJECT MODE to see particles) jetting out from every vertex in the icosphere, making it like like a star. Use MB3(MMB) to rotate the view so that you can appreciate the symmetry of the particle stream. Press KP 1 to return to front view.

Press F12 to make a quick render of the changed object. You should see something that lookes like a white explosion. Each particle is rendered as a glowing ball, but the particles are fairly large at this point, and you cannot see them as individual particles.

Use the following procedure to make the particles smaller. At this time, you will also give them a color and some transparency to make them look a little more like puffs of smoke.

Smoke Particle Material

1. In the buttons window, press F5 to display the shading window.
2. Click on the material buttons icon (red gobe) to display the material.
3. Click on the Add new Button
4. Change the name of the material to MA:smoke

The default colour is a brownish grey, which is fine for smoke, so do not adjust it at this point. However, you want to give it the following attributes:

Smoke Particle Attributes

1. Click on the Shaders tab to display the shaders panel.
2. Set the specularity (shininess) to 0.000.

3. Click on the Halo button. Notice that the preview sample changes to something resembling a cloud.
4. Set the value of Hardness to 15
5. Set the Halo size to 0.300
6. Set the value of A (Alpha) to 0.80.

The preview sample might now look as if it has vanished. However, press F12 for a quick render and you will see that the former white explosion has now become a star of smoke puffs. This pattern is still too organized and regular to look much like smoke, but you are getting there.

To model the smoke, press F7 to display the Object buttons in the buttons window again. When you first set up the smoke model, you set the particle emission as "Static". However, you want your smoke to build and to move in the final model, so the next step is to randomize the particles and make them move like drifting smoke.

First, let's look at the default animation of the smoke particles:

1. Click on the Static button. Notice that the star of particles vanishes, and the mesh is now drawn only as vertices - no polygons.
2. Ensure that the icosphere object is still selected (highlighted purple).
3. Pres KP0 to go to camera view.
4. Put the cursor In the 3D View window and press Alt-A. Notice that each vertex now emits particles in the same star pattern.

Before you proceed, take a quick look at a rendering of the animation as follows:

1. In the buttons window, Press F10 to display the Scene buttons.
2. In the Format panel, click on the PC button.
3. In the Format panel, choose AVI JPEG as the file format.
4. In the Render panel, set the image size to 25% (small, for a fast render).
5. In the Anim panel, set the value of End to 100, to create a 100-frame animation lasting about three seconds.
6. In the Output panel, set the filename to /tmp/smoketest, or any other save location that you prefer.
7. Press the Anim button to begin rendering.
8. When all 100 frames are rendered, press the Play button to review the animation.

What you should see is a cloud of smoke puffs starting from the location of the icosphere (which is invisible) and radiating outwards. it looks a little like an explosion of smoke, but is still too organized and regular to appear as smoke.

Now that you have invested some time in creating the animation, it might be worth saving the .blend file. After you have saved the file as smoke1, you can experiment with making the particles look more like smoke. Go back to the section titled:

Smoke Particle Attributes and experiment with the values of Halo Size, Hardness and Alpha. You can also go back to:

Smoke Particle Material and experiment with the color of the smoke. You do not need to create a new material, you only need to change the existing RGB values.

For example, a Halo Size of 0.900, an Alpha value of 0.100, and a hardness value of 1.0 will produce a denser, more realistic smoke. After you finish experimenting with the values, use these values and save the file again.

Randomizing the Smoke Particles

If you are modelling smoke from a fire, you want the particle system to move randomly, but to drift in a predictable direction as if moved by a breeze or simply billowing up from a source of fire.

Blender's particle interaction options provide some sophisticated ways to control particles and make them interact with the environment. However, for this tutorial you will simply

specify some variables that randomize the particles and move them in the desired direction. You can learn about particle interaction later.

To set up the model, first revert to a static view of the particles, as follows:

1. In the buttons window, press F7 to display the object buttons.
2. In the Effects panel, press the Static button. Notice that the 3D view changes to the particle star.
3. Press KP1 to display the front view.
4. In the Effects panel, find the group of variables labelled "Force" [Under "physics" button and "particle motion" tab in Blender 4.1] and change them as follows:
    1. Change the value of X to 0.200. Notice how the particles are now skewed to the right.
    2. Change the value of Z to 0.200. Notice how the particles are now skewed upward.
5. In the Effects panel, change the value of Randlife to 0.200. Notice that the pattern of the particles becomes more disorganized.
6. In the Effects panel, change the value of Rand to 0.100. The pattern becomes even more disorganized.
7. Press KP0 to return to camera view.

What was formerly an organized star of particles has now become a random mass. You can now preview the animation as follows:

1. In the Effects panel, click on the Static button.
2. In the 3D view window, press Alt-A.

The cloud of particles now drifts randomly to the right. Preview the animation as follows:

1. In the buttons window, press F10 to display the Scene buttons.
2. Press the Anim button to start the animation

You should now see a much more realistic drifting cloud of smoke. However, one of the problems with it is that you can still see a pattern of particle emission from the vertices of the object. One way to make this less visible is to reduce the scale of the object, squeezing the vertices together, although that will make the source of the smoke appear very dense. Using the mesh edit decimator option to reduce the number of vertices will make it less dense, as will ramping up the Alpha value of the smoke halo.

The *BEST* way to truly randomize particle emission is to select the emitter and go into edit mode (TAB key), select ALL vertices of the emitter ("A" key), go to the Editing menu (F9), and then to the Mesh Tool Subpanel. There, press the HASH button, then leave edit mode (TAB key). You will now see particles emit in a TRULY random fashion, something that has been overlooked by nearly every other tutorial available.

There are many other ways to make the smoke look more natural, such as by animating the value of Alpha or color to make the smoke seem as if it is getting thicker. You can use several emittors to send out particles of smoke of different color and density, with different rates of emission. By animating emittors, you can create a very realistic effect - or even a moving cloud system.

However, for this tutorial, we will keep it simple, and move on to the next step of animating simple flames and sparks to mix in with the smoke.

At this point, you can save the file, and then experiment with changing the values of the particle system to see how it affects the smoke.

Next Page < Blender 3D: Noob to Pro/Furry
Previous Page < Blender 3D: Noob to Pro/Particle Systems

Furry

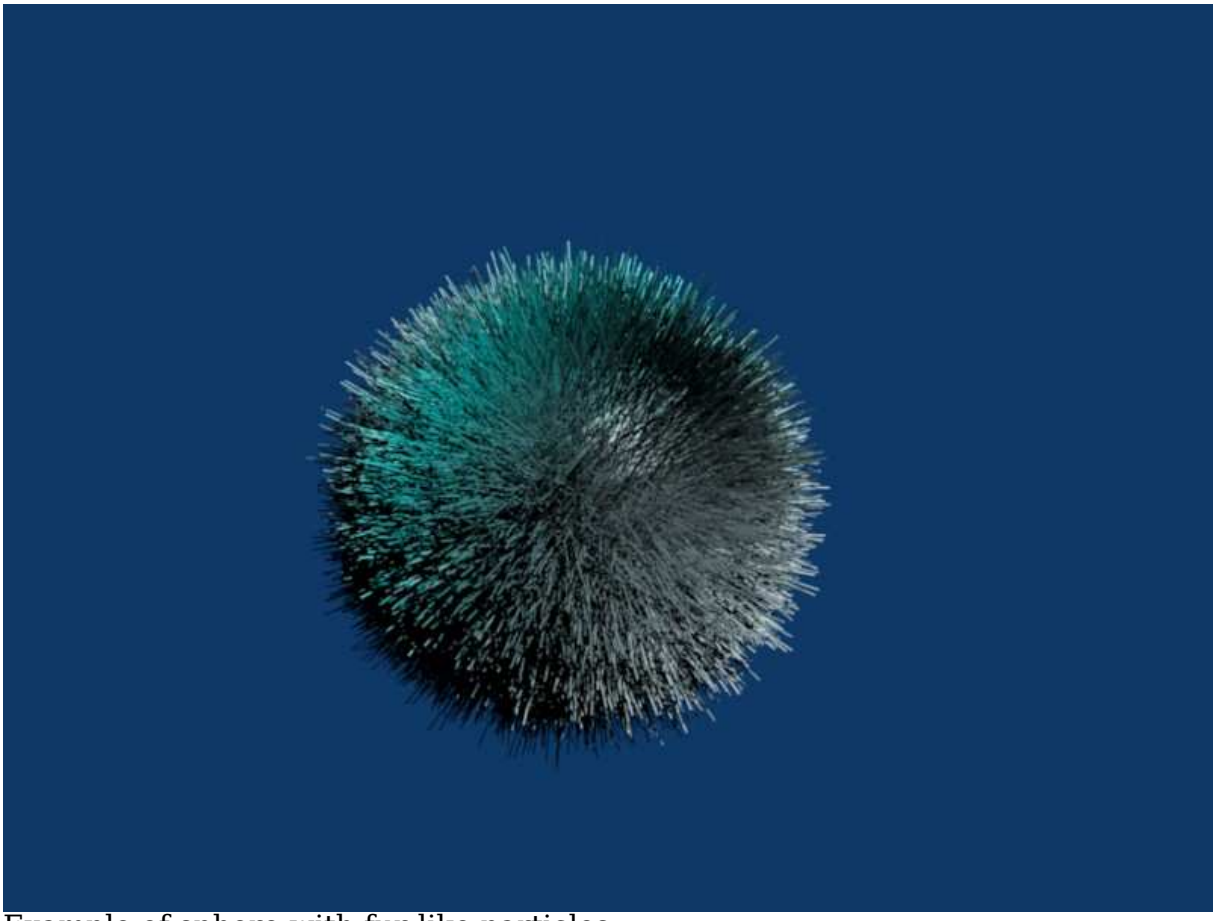
Next Page: Game Engine Basics

Previous Page: Making Fire

Let your hair hang down

To learn how to make **fur or hair**, click on the link below and scroll to the bottom. It is the last tutorial.

Or...

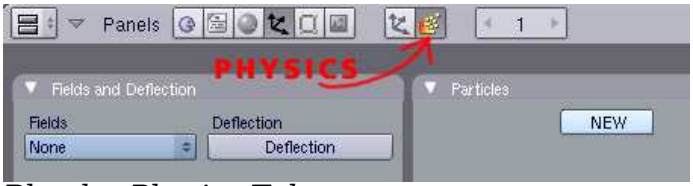


Example of sphere with fur-like particles

Creating Fur in Blender 2.40

Matt Liebrich

First off, select the object you want to be furry. (You don't have to duplicate the mesh anymore!) Then, go to the "Object" tab (**F7**), and click on the "**Physics**" tab *(to the right, on the button window menu bar)*. (Or Press **F7** repeatedly to "cycle" between the **Object** and **Physics** *SubPanels*)



Blender Physics Tab

Under the "Particles" palette, click on "**NEW**". *For best results you should be in object mode.*

A bunch of things will pop up in place of that "New" button. Several things are important for

the creation of hair/fur. The most important are:

- Static
- Vect

And under the "Particle Motion" palette:

- Normal
- Random
- And the "Z" (or possibly X or Y, depending on the result you want.)under "Force"

Anyway, back on the "Particles" palette, push "**Static**" and then "**Animated**" if you want it to be in an animation. Then push "**Vect**" for actual "strands" of hair.

Afterwords, in the "**Particle Motion**" palette, adjust the "**Normal**" value until the hair is as long as you like. Increase the **randomness** value for added realism. *(Not too much, though!)* Then, make the value of "Z" under "**Force**" somewhere around -0.02.

Further information shall be added later by any one interested in doing so. Possible topics to write about include:
Fur/Hair Material Settings
Further, more advanced set-up of hair/fur.

This is the link:
http://download.blender.org/documentation/NaN_docs/Manual2.0/Particles.html

---

Next Page: Game Engine Basics
Previous Page: Making Fire

Game Engine Basics

Next Page: Your First Test
Previous Page: Furry

Purpose: To demonstrate the object collision feature of Blender's Game Engine.

Object Collision Basics For The Blender Game Engine

With the mouse cursor over a 3D viewport, press NUMPAD-7 to switch to TOP view.

Press the spacebar and select Add >Mesh >Plane. Press TAB to leave edit mode and enter object mode. [4] (http://img192.echo.cx/my.php?image=17bf.jpg)

Enlarge the plane by pressing S and dragging the mouse cursor away from the center of the plane. Click to stop re-sizing. The plane will serve as the 'floor'.

Add A UVSphere using the same sequence for adding the plane object. Press TAB to enter object mode.

With the mouse cursor over a 3D viewport, press NUMPAD 1 for Front view, then press G and move the sphere above the plane. Click to exit Grab Mode. You may need to scale the sphere down. It will be the ball which collides with the floor.

Go to logic panel (move cursor over Button Viewport and press F4).

Click on actor button (in the upper left corner of the logic panel), then Dynamic, then Rigid Body. This activates physics properties for the sphere so it can bounce. A dotted circle

appears around the sphere; use the Size Slider to adjust the dotted line so it closely conforms to the perimeter of the sphere.

Next, you want to add some colour to the sphere and the plane (see Blender_3D:_Noob_to_Pro/Quickie_Material for details), so you can tell the two apart during simulation.

Move the cursor over a 3D viewport and press P. The sphere will drop down and collide with the plane. Press ESC to end simulation

For fun, try rotating the plane in front or side (NUMPAD 3) view and press P again. The ball will first hit the plane, then roll down. Add several more planes rotated to differing angles and position them in the falling path of the ball to keep it going.

Next Page: Your First Test
Previous Page: Furry

(Note: Where is the size slider in 2.42a?)

Your First Test

Next page: Blender 3D: Noob to Pro/Build a skybox
Previous page: Blender 3D: Noob to Pro/Game Engine Basics

The Game Engine is an interesting feature of Blender. The Game Engine is basically a 3D environment in which 3D objects move around and react to each other upon contact. The Game Engine is most easily applied to 3D architectural tours.

As a start, we will teach you to make a ball roll realistically down the hill using Blender's game engine.

First, make a plane. Subdivide it a few times and scale it up a lot. Now, make a sphere. It can be any kind. With the sphere selected, go to the game engine settings (the little purple Pacman-icon). You will see something that says "Actor". Press it. Now press *Dynamic*. You will see a bunch of settings available now. Change *Size* to 1.4. This changes the boundary that iniates reactions when made contact with by another object. You notice a dotted circle around the object; this is the boundary. Also enable *Rigid body*. This makes the ball roll, instead of staying completely upright the entire time. Edit the plane you made, pulling vertices up or down. For this example, make it look like a canyon. Put the ball at the start point and press the P-Key. You will see the ball fall in, and roll. If it doesn't go the way you wanted, Pull some vertices of the canyon downwards. If you just see white silhouettes, don't worry. Press escape to exit the game mode. Press Z key. Go back into game mode (P) and everything will be in wireframe while you play the game.

Now, how about adding a little color to this scene so it doesn't look all white. We will do this using a method called vertex paint. Press Z again for the normal, solid mode. From the object mode, locate a scroll-up menu on the 3d view task bar currently labelled "Object mode". Click it and select "Vertex Paint". Note that all the objects in the 3D view window are white silhouettes again and that your cursor also has a circle around it. Select the edit buttons menu below. Here, you will find a square for color selection. Pick a grayish brown color for the rock. Now go into the 3D view window and click on the rock. Every vertex in your cursor's circle should start to turn a shade of that color.

Next page: Blender 3D: Noob to Pro/Build a skybox
Previous page: Blender 3D: Noob to Pro/Game Engine Basics

Build a skybox

Next page: Blender 3D: Noob to Pro/Advanced Tutorials
Previous page: Blender 3D: Noob to Pro/Your First Test

One way to add a realistic feeling to your 3d environment in a game engine is to create a skybox. A skybox is a large cube which has on its inside a projection of a 360° environment. When the player (camera) is inside this environment, the scene is rendered with the illusion of being inside a gigantic world. This is a similar effect to Quicktime VR (see http://fullscreenqtvr.com for examples). And, by setting up the skybox as a simple cube shape, you place the least amount of strain on the graphics engine. It's a great advantage for your game with very little overhead.

This tutorial will show you how to create skyboxes relatively easily from panoramic photos. My favorite part is, you can do it easily using free tools such as Blender and the Gimp.

Using the Gimp to manipulate images is not really in the scope of this tutorial... check out some other page on using that software. You should have an understanding of how to edit images and apply alpha channels. (You could also use the Gimp to apply a polar coordinate texture to your rectangular image in order to create a fisheye image. Hint: it's not the sphereize filter.)

Gather your graphics

You can take panoramic images yourself using a regular digital camera and a tripod. A quick way to accomplish this is to draw marks on your tripod base at every 30 degrees (think of the hours on a clock face). Make a single mark on the swivel of your tripod to allow you to line up your shots -- twelve shots at 30 degrees each. Then, using a program such as the Gimp (http://www.gimp.org) or the incredibly cool Autostitch (http://www.autostitch.net) to merge the photos into one big panorama.

Or, if you're lazy like me, you can just grab photos online to use as templates to create original images. There are also many places you can download non-copyrighted photos for free as well. One resource for cloudy sky textures, as well as panoramic photography instructions, is Philippe Hurbain's site Philo's Home Page (http://www.philohome.com) . This tutorial will use a fisheye sky photo from his copyright-free Panoramic Skies images collection.

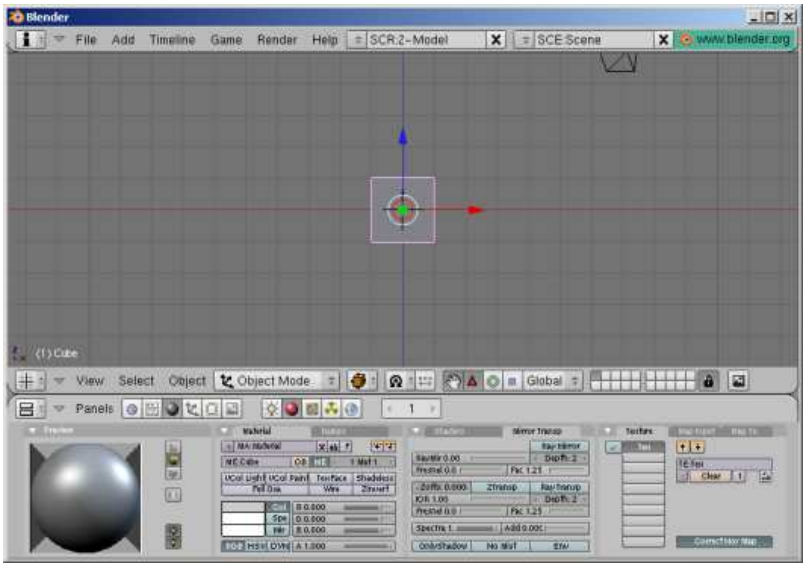
You'll also probably want a photo for your ground, unless you prefer to use real models such as buildings in your skybox. This earlier chapter (http://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Mountains_Out_Of_Molehills) on creating landscapes can be incorporated into setting up your skybox. However, this tutorial will use the sky photo for the top half of our world, and a panoramic landscape with an alpha channel for the bottom half. I've created a ground image using copyright-free textures obtained from Accustudio (http://www.accustudio.com) .

Here are the images I'll be using (you'll want to use images with higher resolution):

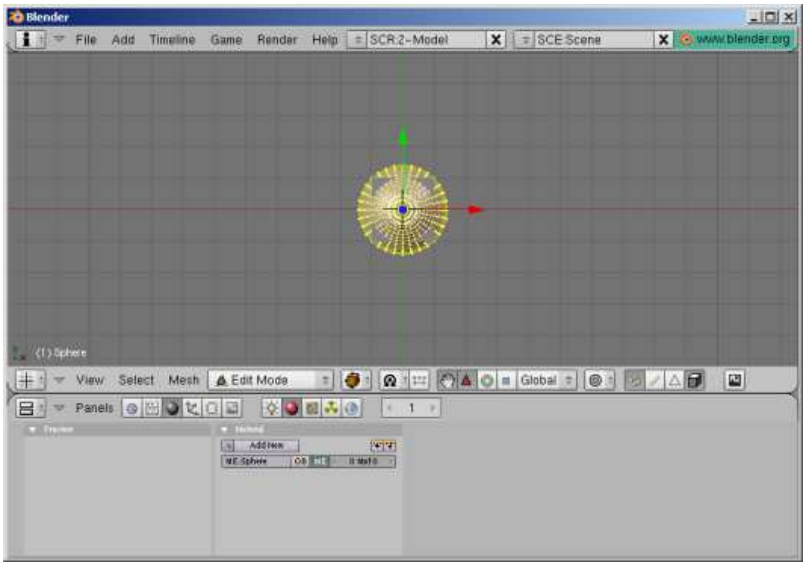
Note: I've outlined the horizon of the ground texture with an alpha channel which will allow me to place the ground mesh right against the sky mesh with a very natural feel.

Create a dome for the sky

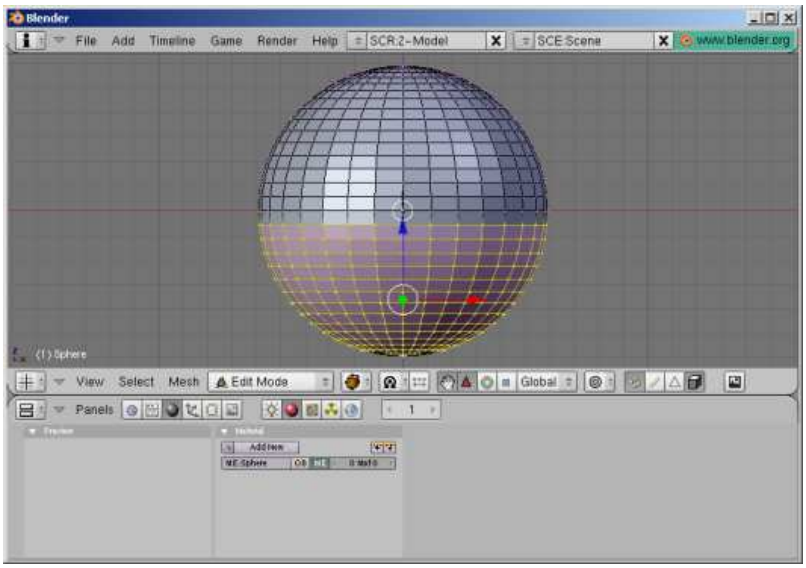
Open a new file in Blender. Your default new file will probably be a two-unit cube in the center of the screen, with a single light source and a camera. You can delete the light source because we won't be needing it. Leave the cube, because that is what will become our skybox.



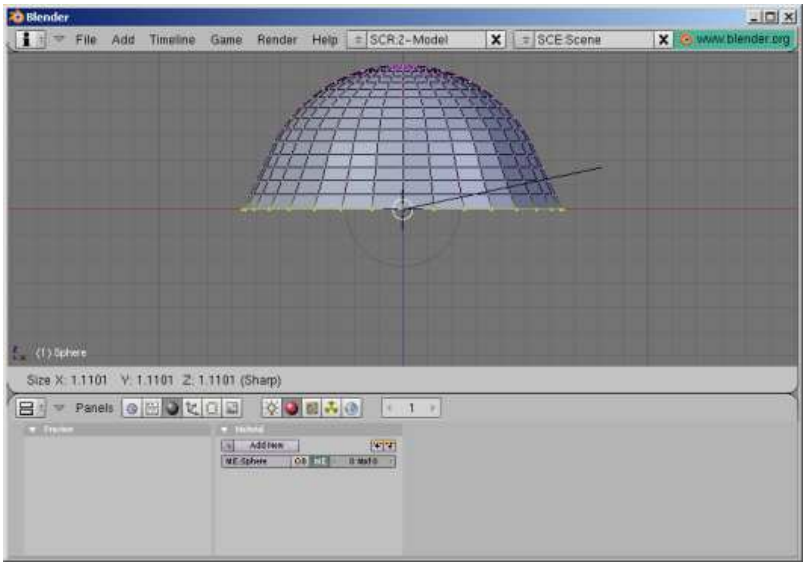
The cube will be the center of our environment, so use Object->Snap->Cursor To Selection if your cursor is not centered. Then, from the top view [KEYPAD-7], Use [KEY-SPACEBAR] to insert a new mesh; make it a UV sphere. I find a 32-segment, 32-ring sphere to be sufficient. We create the sphere from the top view because that is the projection from which we want to add the sky texture.



Scale up the sphere so it resembles a large "arena" in comparison to your cube, and select and delete the lower half of the vertices, using the front view [KEYPAD-1] and [KEY-B] to create a bounding box. It helps if "Select Visible" is turned off so you can select all of the vertices in one go.



Turn on proportional editing with [KEY-O], then select the bottom row of vertices and scale them up with [KEY-S] so that the bottom of the sphere gets a bell shape. Because the projection of the sky texture will be from the Y-axis (ceiling) we need the bottom faces of the sphere to be at an angle, to catch the texture. (Faces perpendicular to the projection will look like smears.) Alter the influence of proportional editing with [KEY-PAGEUP] and [KEY-PAGEDOWN]. Linear or Sharp falloff works best with the sphere shape.

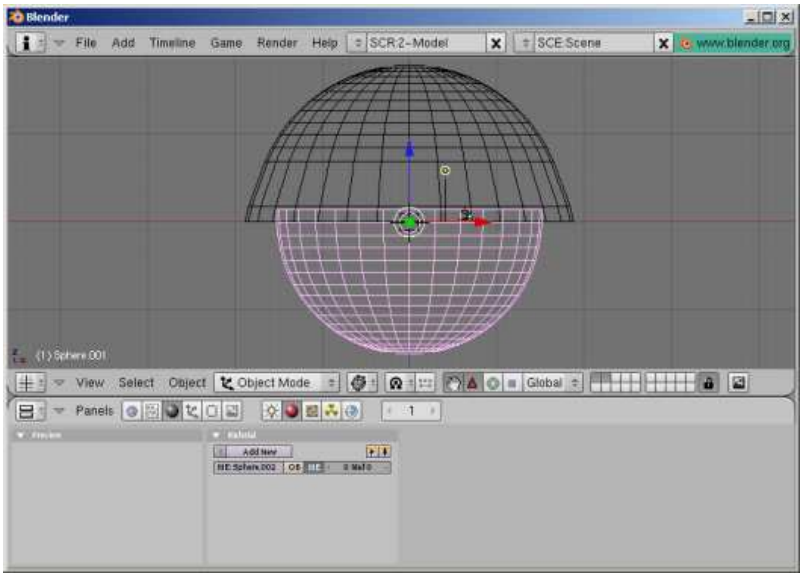


Now you're ready to add your sky texture to this mesh. In the Materials menu, create a new material and a new texture. Be sure to set your material not to recieve shadows by clicking the "Shadeless" button. Then, in the Texture menu, set the texture type to Image, and click the Load Image button to insert our sky texture. Back in the Materials->Texture->Map Input menu, you may need to scale your image to get rid of the distorted textures at the edges of the fisheye by setting the Size to, say, 0.950 for X, Y and Z.

At this point, if you wish, you can reposition the camera and render the scene to see how your sky mesh looks.

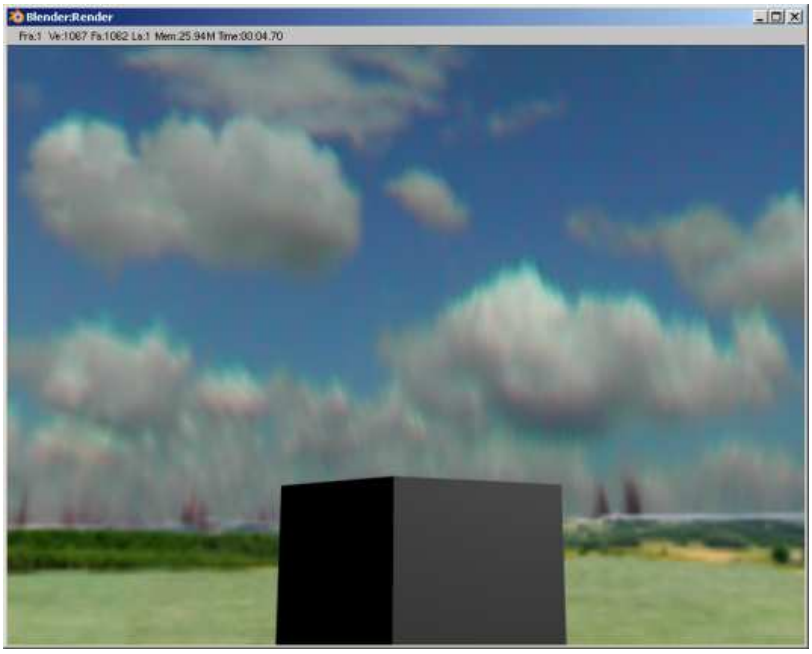
Create a dome for the ground

I found it easiest to move the sky dome to a new layer with the [KEY-M] move to layer command. Then you can select the cube, Object->snap cursor to selection if you need to, select the top view [KEYPAD-7] and insert another UV sphere just as before -- except this time, remove the top hemisphere of vertices. I left an extra row of vertices at the "equator", scaled up, to function as a "billboard" to display the the horizon of our ground texture with the alpha channel. This sphere should be slightly smaller than the sky hemisphere.



This time, I will apply the ground texture with a tube projection, so it is projected onto the mesh horizontally. Because I have an alpha channel on this texture, I click "Use Alpha" in the Texture menu and Map To -> both Col and Alpha in the Materials menu. You will also need to set ZTransp in the Mirror Transp menu so that your alpha channel shows up in the envmap (which will become your skybox), and Alpha to 0 to allow the masked areas to be transparent. (Alpha channels appear to require Z buffering to appear on procedural textures.) Also, you may need to adjust the offset of the ground texture (Y-axis), so that the horizon appears properly on the "billboard" area of your ground hemisphere.

Again, you can reposition the camera and render the scene to make sure everything is properly aligned. Be sure to activate the layer where you moved the sky mesh. Your results will look similar to the following image. Set OSA on in the render screen for best results. Also, use higher resolution images with cleaner alpha channels -- the image below is rather blurry and you can see a halo around the horizon.



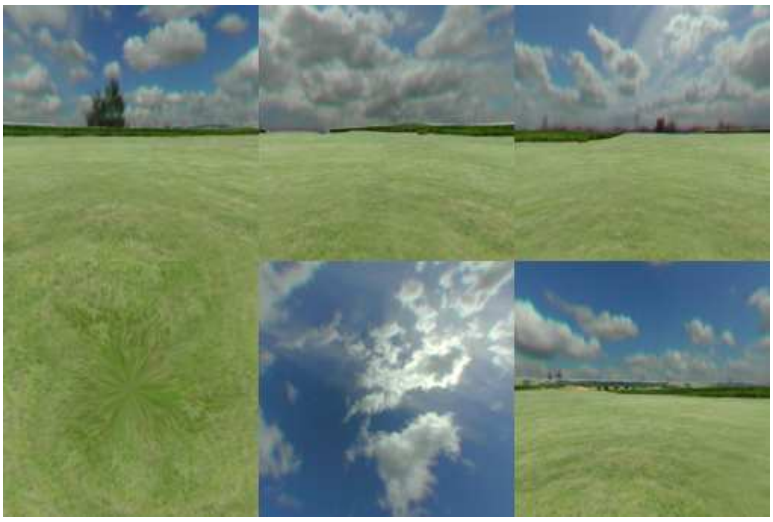
Render the environment map

The last step is to use the procedural Envmap texture to project the dome textures onto the cube, which will become our skybox. Select the cube and create a new material. Set the material to "Shadeless." Add a new texture and make its type Envmap. Set the CubeRes to whatever you want the resolution of your skybox to be (512 is a good resolution for a game; 1024 or 2048 are fairly high-res; I stuck with low-res for this tutorial). If your sky & ground hemispheres are very physically large, you may also need to increase the ClipEnd value to include all of the faces. You may want to set the Envmap calculation to Anim so you don't have to keep freeing envmap data if you're experimenting. (Anim automatically clears Envmap data with every render, otherwise you must click 'Free Data' to reset the Envmap.)

Once you've created the Envmap texture, you should be ready to render the Envmap for your skybox. If you want to set your file format such as JPG or PNG, you should do that first. Then, simply go to the render screen and click "Render." Again, make sure all layers are visible. The rendering window appears. First, Blender renders the environment map of the cube. Afterward, the camera view is rendered, at which point you can hit [KEY-ESCAPE] to stop rendering -- we are only interested in the environment map which is already complete.

Select the cube again, then get to its texture menu. You will see the newly-rendered Envmap on the sample texture. Click "Save EnvMap" in the texture menu to save the rendered Envmap.

Blender environment maps are saved as a 3x2 matrix of squares, as seen here:



You can now load this image as an envmap texture in a new cube, which you can incorporate into your game as a skybox. This file can also be edited in the Gimp to remove any unwelcome artifacts such as trees, buildings, jet trails, etc. Also, because I used a tube projection on the lower hemisphere, in the bottom face of the envmap you see a strange star shape at the "pole." You'll most likely have a floor in your game, so you probably won't see that face anyway, but sticklers can avoid it with clever use of the Filters->Distorts->Polar Coords filter in the Gimp or Filter->Distort->Polar Coordinates (Polar to Rect.) in Photoshop. Patching also works well.

To make the skybox appear as a static background in your game, vertex-parent it to your player object.

Next page: Blender 3D: Noob to Pro/Advanced Tutorials
Previous page: Blender 3D: Noob to Pro/Your First Test

Match Moving

Next page: Blender 3D: Noob to Pro/HDRi
Previous page: Blender_3D: Noob to Pro/Build a skybox

1. Take a look at the WikiPedia article on Match Moving
2. Download the free Match Moving solver, Voodoo (http://www.digilab.uni-hannover.de/docs/manual.html)

Next page: Blender 3D: Noob to Pro/HDRi
<large>*Previous page: Blender_3D: Noob to Pro/Build a skybox*</large>

The Blender 3D: Noob to Pro/Printable Version module is a **stub**. You can help Wikibooks by expanding it

(http://en.wikibooks.org/w/index.php?title=Blender_3D:_Noob_to_Pro/Printable_Version&acti
.
If possible, use a bookshelf-specific categorization template.

High Dynamic Range imaging (HDRi)

You may have heard various people talk about HDR images. (WETA, Lucas, even Tim Sweeny). HDR images are part of a technology called HDRi which stands for "High Dynamic Range (image)". So... what on earth does that mean?

Here's a link to Wikipedia's article on the HDR format (http://en.wikipedia.org/wiki/High_dynamic_range_imaging) which I personally give all my credit to Paul Debevec for putting it to use for computer graphics purposes. Anyway, before you start trying to understand the usefullness of HDRi, please read the wikipedia link.

Also, visit Paul Debevec's website (http://www.debevec.org) if you've got some more time to spare.

To sum up the excitement of HDR CG, think of it like the hype of the next-generation videogames that are about to come out, except set the stage for 1996 instead of 2006. Paul Debevec pioneered paralax mapping, HDR lighting, image-based modeling, his latest work includes some even more amazing technologies, and for the record he's my hero too.

To use HDRi images for 3D rendering, you need something called a light probe...

HDRI stands for High Dynamic Range Imaging, and is basically a technique to use a picture of the environment to light your scene. This will result in very realistic and convincing shadows, higlights and reflections. This is very important for realistic emulation of chrome for example.

First of all, you'll need an HDR image. There is a whole range at http://debevec.org/Probes/ that you can download for free. I will use the Uffizi Gallery probe, but any other HDR image will do just fine.

To apply the HDRI environment to your scene:

- Go to the shading settings (press F5) and click the World button.
- In the "Texture and Input" tab, click "Add New" and "Angmap".
- Then go to the "Map To" tab and deactivate "Blend" and activate "Hori".
- Now go to the Texture settings (press F6) and change the "Texture Type" to "Image".
- Click the "Load Image" button and locate your HDR image.
- To be able to render using this environment, you need to enable YafRay. Press F10 and change the "Blender Internal" to "YafRay".

The Blender 3D: Noob to Pro/Printable Version module is a **stub**. You can help Wikibooks by expanding it

(http://en.wikibooks.org/w/index.php?title=Blender_3D:_Noob_to_Pro/Printable_Version&acti
.
If possible, use a bookshelf-specific categorization template.

Creating a Light Probe

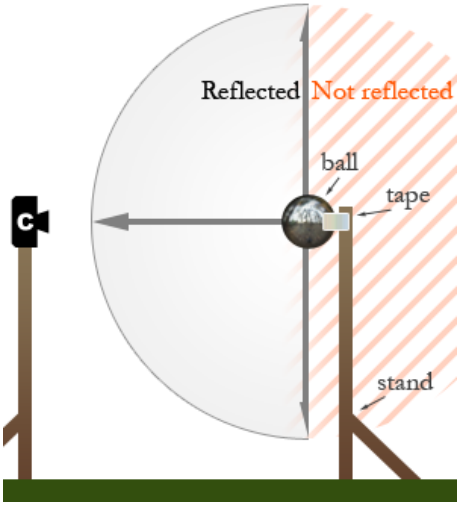
The light probe is, in the simplest terms, a photograph of your environment. they work in very much the same way that reflection maps do, and are made the same way.

Equipment:

1. A pure silver ball. Try a plastic christmas tree ball ornament.
2. A camera, prefferably digital. If you have a high-end digital camera, you'll have less work ahead.
3. A place you'd like to capture the lighting from. Try laying different things in your environment to get a good idea.

4. Something to fire the shutter without touching the camera. For digital cameras, Paul Debevec reccomends using a program that will take all the pictures for you.
5. A tripod.

Set it up like so. Remember that the height of your camera and the height of the ball relative to each other controls the angle at which the horizon will be shot. In other words, shoot the ball at the same angle that you plan to shoot your 3D scene in. If your scene is animated... consider making a similar rig except attaching your reflective ball to a videocamera as was done for wikipedia:Flight_of_the_Navigator.



1. Note that the diagram is incorrect. The camera will see items reflected from behind the sphere as well.

The process:

1. Set up your rig and take

The Blender 3D: Noob to Pro/Printable Version module is a **stub**. You can help Wikibooks by expanding it

(http://en.wikibooks.org/w/index.php?title=Blender_3D:_Noob_to_Pro/Printable_Version&acti
.
If possible, use a bookshelf-specific categorization template.

Blender FAQ

I haven't been able to find an unto date FAQ on the web so Please add in any FAQ or answers that you know. I will add a bunch in over the next week but the more people who get involved the better.

- Q) After installing I get an error message "A required .DLL file, MSVCR71.DLL, was not found"?

  A) The file is part of the built in Python scripts. It's freely distributable and can be found with a quick search of google. [[5] (http://www.dll-files.com/dllindex/dll-files.shtml?msvcr71)]

- Q)How do I print from Blender?

  A) You can't print from blender. You can export the image to a graphic file, e.g. PNG or jpg and print that from a graphic program, e.g. Gimp (http://www.gimp.org/) ,IrfanView (http://www.irfanview.com/) or msPaint. If you want you can also print from a word, e.g. Corel or Open Office, but the quality is better in a proper graphics program.

- Q) How do I export an image to a file?

A) Press F10 and in the far right hand corner their is a drop down menu which allows you to select a file type. jpeg, png, gif, etc. as well as different movie formats such as avi, and mpeg. Select the format you want the resultant image to be in.

Render the image.

Press F3 and the "save file" window comes up.

Type your file name and extension and press "save"

- Q) How do I accelerate rendering through a network?

A) Taken from elysiun (http://www.elysiun.com/forum/viewtopic.php?t=38702) and written by Pedro Doria Meunier (pdoria).
Requirements DrQueue (http://www.drqueue.org/)

Well... you're right... time to give back what I've learned from Jorge Daza. So here goes the 'quick & dirty' solution:

1st go through the install procedure explained at DrQueue's (http://www.drqueue.org/) site.
PAY EXTRA ATTENTION at those environment variables! You have to set them in your .bashrc file. do whatever procedure you like to activate them (relogin, etc..)

2nd make a NFS share (ex. /mnt/shared/DrqQueue). chmod it so that you don't have any permission issues. the envvars set in step 1 *must* match the nfs share! ex. if you've created a nfs share /mnt/shared/DrQueue then DRQUEUE_ROOT must be set to /mnt/shared/DrQueue.

3rd compile and install it. it'll get installed in that shared folder. hold on to your hats! we're halfway through!

Now for a quick test:
execute /mnt/shared/DrQueue/bin/master ---> this'll start the master
execute /mnt/shared/DrQueue/bin/slave ---> this'll start a client
execute /mnt/shared/DrQueue/bin/drqman ---> this'll call the GUI. check if there's a :computer (your own!) listed in that 'Computers' tab. if it's there your master computer is set up!

Now for the SLAVES:
all you have to do is mount that nfs share somewhere... say /mnt/DrQueue.
Something that helps is to mount the share at boot time... editing your /etc/fstab and adding the appropriate line will do the trick.

if you've mounted it there then exec /mnt/DrQueue/bin/slave.
go back to the master. you should see another computer listed there!

Something **important** about the scene files and textures:
You **must** place the scene files **inside** that shared folder. same thing with the textures that are referenced inside the scene file.
A neat trick is to pack the scene file and then unpack it inside the shared folder. this way all the textures are unpacked at /the/shared/folder/textures .remember to save the scene file after doing this!

Hope this helps. Happy Blending!

Advanced Tutorials

Next page: Blender 3D: Noob to Pro/Python Scripting

Previous page: Blender 3D: Noob to Pro/Build a skybox

Author: Anthony Gomez (Extensor) Date: March 5, 2005

Short Description: RVK tutorial

- Letters in brackets ie:(z) mean there is addition information at the bottom of the page.

Introduction:

This tutorial is meant to stop all the RVK questions. Orly?

Window Layout:

Set the left half of the screen as 3D View. The other half is divided in two. The top is Action and the bottom is IPO (set to vertex display).

Setting your Neutral Pose

Make sure you are on the first frame (a). With the cursor over the 3D View, select the mesh you want to animate. (mesh in object mode) and press the I key. Select Mesh from the pop up menu then Relative Keys from the next pop up menu. A line will appear in the IPO view. This line is your neutral pose.

Setting up your additional Pose Lines

Now, figure out how many key frames you will need. If you want to move both eyebrows up and down then you will need 4 additional IPO lines.

Left Brow Up Left Brow Down Right Brow Up Right Brow Down

Press the up arrow (cursor key) to move to forward 10 frames. Press the I key while over the 3D View and select Mesh. Repeat until you see a total of 5 lines in the IPO window.

Set your Poses

Right click on the Neutral pose line in the IPO window. This sets the mesh to the neutral pose. Now Right click on the next line up in the IPO window. Enter edit mode in the 3D View and move the vertices as desired (in this case you will be moving verts to get the left Brow up pose). Press Tab to exit edit mode. Now right click your Neutral pose line in the IPO window. You will see your object in its neutral state. Right click the next line up and you should see the changes you just made to your object. Set up all your mesh poses following the above instructions.

Name your Poses

RIght click on the Key names in the Action window. Change the name and click OK.

Time to Animate (b)

Click on the arrow next to the Sliders text. This will give you access to the pose sliders. Move to frame 20 to start your action. Move the pose slider but release the mouse when set to 0. Now move 10 frames forward and move the same slider to 1.00 (maximum). Use this method to set up all your actions(c). Remember to add a 0 value frame to end the pose.(d).

Adjust your Slow in & Out

In the IPO View select from the menu to find the IPO curves. You can get back to the Pose lines by selecting KeyIPO from the same menu. Right click the spline you want to edit and press TAB to enter edit mode. Move the handles to adjust slow in/out.(e)

(a) In this case moving to a frame has nothing to do with animation. It is done so that your pose lines are seperate from each other. (b) Select your key frame marker and use the usual commands to move <g> and duplicate <d> them. (c) Be subtle by not pushing the slider all the way to 1.00. (d) Try overlaping your poses. (e) When setting slider values they can sometimes go into the negative value. This will give you weird results. Although sometimes they can make your animation more interesting. To fix this edit the IPO, select the point where the line dips below zero and press the V key. Do the same at the othe end of the curve if needed.

Warning! Blender has a limit to the number of verts you can use.

Next page: Blender 3D: Noob to Pro/Python Scripting
Previous page: Blender 3D: Noob to Pro/Build a skybox Help. When you save runtime it doesn't carry across the materials that were mapped to the object. If you render then it is there but not in the .exe. Also how do I network the game once I have made it. ie how do I send messages across a port? Why does save dynamic runtime never work?

Please help. The reason why I am posting this here is because I have not been able to enter an email forum. So far. Please can you tell me how to do this. Please post your reply to ir_nyad@hotmail.com

Python Scripting

Next page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Introduction
Previous page: Blender 3D: Noob to Pro/Advanced Tutorials

One of Blender's powerful features is its Python API. This allows you to interface with Blender through the Python programming language. The Python interface allows you to control almost all aspects of Blender, for example you can write import or export scripts for meshes and materials of various formats or create procedurally generated textures. You can also create complete animations procedurally and write scripts to modify existing scenes in any way you can think of. On top of all, you can easily create a user interface for your script, transforming it into a generally useable tool.

http://www.blender3d.org/cms/Python_Api.380.0.html

http://www.blender.org/modules/documentation/240PythonDoc/index.html

http://www.blender3d.org/cms/Python_Scripts.3.0.html

http://www.elysiun.com/forum/viewforum.php?f=5

Next page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Introduction
Previous page: Blender 3D: Noob to Pro/Advanced Tutorials

Introduction

Next page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Export scripts
Previous page: Blender 3D: Noob to Pro/Python Scripting

For a general introduction to python programming, see:
http://en.wikibooks.org/wiki/Programming:Python

Introduction

Python is used in blender to write plugins as well as automate tasks.

Next page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Export scripts
Previous page: Blender 3D: Noob to Pro/Python Scripting

Export scripts

Next page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Import scripts
Previous page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Introduction

---

Introduction

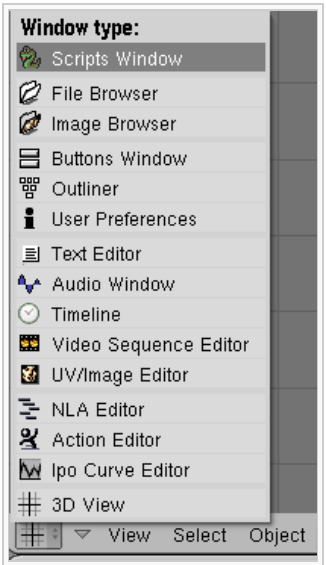
Blender is not just useful to create complete animations, but it's also a great modeller. You can build your complete 3D scene in Blender, and then export it to a useful format. In fact, you can use it for much more, for example I was using it as a level editor for a freeware 2D game someone else made. There was a short deadline for the game to be finished, and 2 weeks before that deadline, there still was no level editor for it. It had a custom ASCII level format, consisting of lists of materials, vertices, triangles and objects. So, remembering the Blender Python exporters, I volunteered to write an export script for Blender, so it could be used as level editor. And it worked out very well, Blender can be completely used as level editor for that game now.

In this tutorial we'll learn how to write a simple Python export script for Blender. Without requiring previous Python knowledge, it will explain how to query the objects in your scene, and how to write them to a file. It will also demonstrate the usefulness of export scripts, by showing how you can process the data while exporting, so you can achieve things that would not work by using any other existing format.

So, open Blender, make sure the default scene is loaded, and let's begin..

Finding out about things in a scene

Before we can export something, we must know what to export. One way to get this information is the Outliner window (**SHIFT-F9**). It will list all the things currently know to Blender. Now, we want the same information from a script. Open the scripts window (green snake symbol). Click on the menu titled **Scripts**, and go to **Scripts->System->Interactive Console**.



Changing the window type to the scripts window

Now, you are ready for the big moment, you are about to execute the first Blender scripting command. Type this and hit **RETURN**:

```
Blender.Object.Get()
```

As a result, you should see this:

```
[[Object "Camera"], [Object "Cube"], [Object "Lamp"]]
```

Now, what just happened? The line "Blender.Object.Get()" consists of three words, separated by two dots, and then a pair of parenthesis. The parenthesis at the end mean that a function is called. The dots just separate different things. The first, "Blender", means to use a function from the Blender module. Object is a sub-module of Blender. And finally Get is the function out of Blender.Object that we called. The Get() function is used to list all available objects in a scene. In our case, this is a Camera, a Cube, and a Lamp.

To get more information about an object, you can pass its name to the Get() function, and assign it to a variable, like this:

```
camera = Blender.Object.Get("Camera")
cube = Blender.Object.Get("Cube")
lamp = Blender.Object.Get("Lamp")
```

We just assigned the three objects to three variables, camera, cube and lamp. To see the contents of a variable, type just its name:

```
cube
[Object "Cube"]
```

```
camera
[Object "Camera"]
```

```
lamp
[Object "Lamp"]
```

Sometimes it's useful to use Python's dir() function to get more information about an object. For example

```
dir(cube)
```

will write the names of all functions and properties of the object. Quite a lot. But don't worry, soon you will know how to use all of them. You also may want to find out the type of something, which you can do like this:

```
type(cube)
```

In this case, just typing "cube" already displays the type, but from within an actual script,

you would use type(). Something else which can be useful is viewing the documentation of Python objects. To do so, use the help() function on a variable or object.

```
help(Blender.Object.Get)
```

This will print the documentation of the Get() function we used. Of course, an easier way to view the documentation is the online HTML help. Click on **Help->Python Scripting Reference**. Hopefully now your browser opens and displays the online documentation of the Blender Python API. If not, you should find it also here:

http://www.blender3d.org/documentation/237PythonDoc/index.html

(If you compiled Blender from source, it's also easy to create the documentation yourself.) In the documentation, click on Object, then on Get, and you should see the complete documentation for that function. Using the documentation will get absolutely vital whenever you need to do something in a script not covered in a tutorial. And you will need to do so, else you wouldn't want to learn scripting at all.

Another resource you will need, depending on how far you will go with scripting, is the Python reference:

http://docs.python.org/

For this tutorial, maybe read the "Tutorial" section in the python docs, but you will understand everything without doing so.

Now, let's try to find out more about our cube. Type:

```
cube.getType()
```

It will tell us that the cube really is a Mesh object in Blender. Look up getType() in the online docs. Since the variable cube holds an Object, and getType() is a function of that Object, click on Object. There you find getType().

Now that we know that the cube is a mesh, let's find out more about the mesh.

```
nmesh = cube.getData()
```

Every Blender object has data assigned to it, depending on the type. In the case of a mesh, the data are of type NMesh. In the documentation, go to the top again, and look for the NMesh module. It will contain documentation for the NMesh type. You can also try

```
dir(nmesh)
```

to get an idea about the available functions and properties. Try these:

```
nmesh.verts
nmesh.faces
```

The first line will list the 8 vertices of the cube's mesh. The second line will list its 6 faces.

To get a member out of a list, you specify the index in square brackets, starting with 0. So:

```
v = nmesh.verts[0]
```

This will assign the first vertex of the cube to the variable v. By now, you already know how

to use dir() to get a list of possibly interesting things in v, find out about its type with type(), and where to look for the API documentation. It is in the module Blender/NMesh, when you click one "NMVert" under "Classes".

```
v.co
```

This will display the 3D coordinates of the first vertex. Now, what if we want to know the coordinates of all vertices? We could of course assign them all to a variable, but the real way to do this is using a looping constructs. There are numerous ways to do this, but one simple way looks like this:

```
for v in nmesh.verts: print v.co
```

The **for variable in list:** construct assigns each element of the list to the variable in turn, and then executes the commands after the colon with the variable having the value of one particular list element. In a real script, you will have much more than a single command after the colon - so you would write them in the following lines.

By now, you should know enough to try yourself at a real script in the next section.

Creating a script

You can write scripts either in an external text editor, or in Blender's built in text editor. The builtin text editor can be hard to use if it doesn't have the standard shortcuts of your preferred text editor, or if you can't copy/paste between Blender and other applications - but else, is quite usable. You reach it over the window selector, or by pressing **SHIFT-F10** (**SHIFT-F11** for Blender 2.41). If you want, you can enable line numbers and syntax coloring with the buttons at the bottom. Create a new script with **File->New**, paste the below code into it, and save it. Or alternatively, paste the below code into a file, and open that file with **File->Open** in Blender. As name choose something with the extenstion .py, for example wikibooks.py. Put it into Blender's user scripts path. Under unix based systems, this is ~/.blender/scripts/.

Under Mac OSX the path is actually hidden in the blender.app so to know the path you would have to know that the script folder is actually hidden in the blender.app itself. Assuming that Blender is in the applications directory the path would be "/applications/blender-2.37a-OSX-10.3-powerpc/blender.app/contents/MacOS/.blender/scripts If you try to open the .app contents from the finder you will notice that .blender section of the path is not visible, while blender will still be able to navigate to this folder. To see this folder from the OSX terminal use the ls -a command (lists all folders/files even hidden) in the MacOS forlder of the listed path. It is probabyl a good idea to create an alias to the scripts folder in the "/applications/blender-2.37a-OSX-10.3-powerpc" folder so that scripts can be easily manipulated through the finder. I know that its confusing that Blender should have its script folder buried inside the app but it is necessary to keep the app portable and not require an install.

Under Windows the default installation it would be this: "C:\Program Files\Blender Foundation\Blender\.blender\scripts"

```
#!BPY

"""
Name: 'Wikibooks'
Blender: 237
Group: 'Export'
Tooltip: 'Wikibooks sample exporter'
"""

import Blender

def write(filename):
    out = file(filename, "w")
    scn= Blender.Scene.GetCurrent()
    for object in scn.getChildren():
        out.write(object.getType() + ": " + object.getName() + "\n")

Blender.Window.FileSelector(write, "Export")
```

Now, go back into the scripts window, and in its menu, click **Scripts->Update Menus**. If you saved it into the right path, from now on there should be an entry "Wikibooks" in the File->Export menu. Try exporting any scene with it. It should open the file chooser dialog, and after you select a file and press the "Export" button, write a list of all objects in the scene into it. There will be one object per line, with the type, followed by a colon and the name.

How does it work? If you look at the script, you probably already know. But just in case, let's look at the script line by line. The first line contains this:

```
#!BPY
```

It tells Blender that this is a Blender script, and therefore it will consider it when scanning for scripts. Next simply follows a string, enclosed in triple quotation marks, so it can span multiple lines.

```
"""
Name: 'Wikibooks'
Blender: 237
Group: 'Export'
Tooltip: 'Wikibooks sample exporter'
"""
```

It contains four items, which Blender uses to place the script into its menus. The name, group (menu location), and tooltip, all enclosed in single quotes. And the Blender version this is for.

```
import Blender
```

Remember how we said all functions from the Blender module start with "Blender."? In the interactive shell, we could simply use them, but in a python script, all used modules must be declared with an import statement. So the above simply allows us to use the functions from the Blender module in our script.

```
def write(filename):
```

This defines a function in Python. The syntax is **def name(parameters):**. In our case, the name is "write", and we have one parameter, called "filename".

```
    out = file(filename, "w")
```

Here we open a file for writing (the "w"), with the name passed to the function (filename). The python function "file" will open the file, and return a reference to it, which we store in the variable "out".

```
    scn= Blender.Scene.GetCurrent()
    for object in scn.getChildren():
        out.write(object.getType() + ": " + object.getName() + "\n")
```

These three lines are our real export script. You already know what the first line does - first we get the currrent scene, then get a list of all objects in that scene, the for loop is assigning each one in turn to the variable "object". The second line writes to the file - first the type of the object, then the string ": ", then the name of the object, and finally a newline.

```
Blender.Window.FileSelector(write, "Export")
```

This is where execution of the script starts. It is simply a call of a Blender function (look it up in the API docs), which opens the file selector. It will display an "Export" button, and when the user clicks it, our function "write" from above gets called and is passed the selected filename.

This script isn't really very useful yet, but it shows the basics. You should now be able to e.g. also list all the materials in the scene. (Hint: They are just like objects, try to find them in the API docs.)

In the next section, we will learn how to export additional information about objects to our text file.

Exporting a Mesh

Our export script lists the type and name of every object, but that's not very useful yet. If we want to load the exported data in another application, we need more. Let's try to export a mesh object in the OBJ format.

The example below is a cube in the OBJ file format.

```
v 1.000000 1.000000 -1.000000
v 1.000000 -1.000000 -1.000000
v -1.000000 -1.000000 -1.000000
v -1.000000 1.000000 -1.000000
v 1.000001 1.000000 1.000000
v 0.999999 -1.000000 1.000000
v -1.000000 -1.000000 1.000000
v -1.000000 1.000000 1.000000
f 1 2 3 4
f 5 8 7 6
f 1 5 6 2
f 2 6 7 3
f 3 7 8 4
f 5 1 4 8
```

Here is a simple obj export script that exports a selected mesh object, used to export the OBJ file above.

```
import Blender

def write_obj(filepath):
        out = file(filepath, 'w')
        scn= Blender.Scene.GetCurrent()
        object = scn.getActiveObject()
        mesh = object.getData()

        for vert in mesh.verts:
                out.write( 'v %f %f %f\n' % (vert.co.x, vert.co.y, vert.co.z) )

        for face in mesh.faces:
                out.write('f')

                for vert in face.v:
                        out.write( ' %i' % (vert.index + 1) )
                out.write('\n')
        out.close()
Blender.Window.FileSelector(write_obj, "Export")
```

This script will export an OBJ file that can be read by many applications. Let's look at whats going on.

```
scn= Blender.Scene.GetCurrent()
object = scn.getActiveObject()
```

Here we are getting the object you last selected in the current scene. this will raise an error if there are no selected objects, but its an easy way to test a new exporter.

```
mesh = object.getData()
```

This gets the objects linked datablock. At the moment we dont know its a mesh, another case where error checking would need to be added.

```
	for vert in mesh.verts:
		out.write( 'v %f %f %f\n' % (vert.co.x, vert.co.y, vert.co.z) )
```

Here we write a line for every vertice, using string formatting to replace the "%f" on the left, with the 3 values on the right.

```
	for face in mesh.faces:
		out.write('f')

		for vert in face.v:
			out.write( ' %i' % (vert.index + 1) )
		out.write('\n')
```

In the OBJ format each face references a number of vertex indices. For every face we have a line starting with "f", then loop through the vertices in the face. Just as mesh.verts are a list of all the the vertices in a mesh, face.v is a list of verts in the face limited to 4 vertices maximum. (where mesh and face are arbitrary variable names assigned to an NMesh amd NMFace objects) Every vertex writes its index on that same line with 1 added. This is because with the OBJ file format the first vertex is indexed at 1, whereas with Python and Blender the first item in a list is 0.

A new line is written so the next face will start on a new line. - in python '\n' represents a new line when written to a file.

Next page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Import scripts
Previous page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Introduction

Import scripts

Next page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Procedural object creation
Previous page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Export scripts

---

Introduction

Importing objects into Blender is not that different from exporting. However, there are a few additional things to take care of. Firstly, all references to "export" in the header should be changed to "import". Secondly, instead of simply writing out data that Blender provides to us, we are responsible for giving data to Blender and ensuring that it is properly formatted. Although Blender is flexible, allowing us to ignore things like vertex indices, we do need to be careful that we do things in a sensible order.

Additionally, there is a bit of housekeeping to deal with. We should be in edit mode while modifying the mesh data. We also need to link up our newly created data to the scene, after it has been properly constructed, so that Blender can see it and maintain it. This makes it visible to the user, as well as ensuring that it gets saved along with the scene.

Importing a Mesh

Here is a simple script that can import an OBJ file created by the export script.

```
import Blender
def import_obj(path):
        Blender.Window.WaitCursor(1)
        name = path.split('\\')[-1].split('/')[-1]
        mesh = Blender.NMesh.New( name ) # create a new mesh
        # parse the file
        file = open(path, 'r')
        for line in file.readlines():
                words = line.split()
                if len(words) == 0 or words[0].startswith('#'):
                        pass
                elif words[0] == 'v':
                        x, y, z = float(words[1]), float(words[2]), float(words[3])
                        mesh.verts.append(Blender.NMesh.Vert(x, y, z))
                elif words[0] == 'f':
                        faceVertList = []
                        for faceIdx in words[1:]:
                                faceVert = mesh.verts[int(faceIdx)-1]
                                faceVertList.append(faceVert)
                        newFace = Blender.NMesh.Face(faceVertList)
                        mesh.addFace(newFace)

        # link the mesh to a new object
        ob = Blender.Object.New('Mesh', name)
        ob.link(mesh) # tell the object to use the mesh we just made
        scn = Blender.Scene.GetCurrent()
        for o in scn.getChildren():
                o.sel = 0

        scn.link(ob) # link the object to the current scene
        ob.sel= 1
        ob.Layers = scn.Layers
        Blender.Window.WaitCursor(0)
        Blender.Window.RedrawAll()

Blender.Window.FileSelector(import_obj, 'Import')
```

This will load an OBJ file into Blender, creating a new mesh object. Let's take a look at the more interesting portions.

```
Blender.Window.WaitCursor(1)
```

Turn on the wait cursor so the user knows the computer is importing.

```
name = path.split('\\')[-1].split('/')[-1]
mesh = Blender.NMesh.New( name ) # create a new mesh
```

Here, we create a new mesh datablock. The name is made from the path only with the filename.

```
ob = Blender.Object.New('Mesh', name)
ob.link(mesh)
```

Next, we create a new object and link it to the mesh. This instantiates the mesh.

```
scn = Blender.Scene.GetCurrent()
scn.link(ob) # link the object to the current scene
ob.sel= 1
ob.Layers = scn.Layers
```

Finally, we attach the new object to the current scene, making it accessible to the user and ensuring that it will be saved along with the scene. We also select the new object so that the user can easily modify it after import. Copying the scenes layers ensures that the object will occupy the scenes current view layers.

```
Blender.Window.WaitCursor(0)
Blender.Window.RedrawAll()
```

Now the finishing touches. We turn off the wait cursor. We also redraw the 3D window to ensure that the new object is initially visible. If we didn't do this, the object might not appear until the user changes the viewpoint or forces a redraw in some other way.

Next page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Procedural object creation
Previous page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Export scripts

Procedural object creation

Next page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Scripts for modifying meshes
Previous page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Import scripts

---

blender tools:

- http://www.makehuman.org/

- http://www.geocities.com/blenderdungeon/lsystem/

python:

http:///www.alcyone.com/software/lsystem/

other:

- http://www.devx.com/Intel/Article/20333/2046 , "Procedural 3D Content Generation, Part 2"

- ModelingCloudsShape --antont, Sun, 20 Mar 2005 03:33:07 +0200 reply http://www-evasion.imag.fr/Publications/2004/BN04/

these here via:

- http://studio.kyperjokki.fi/FirstAndLast/ProceduralContentCreation

---

Next page: Blender_3D: Noob to Pro/Advanced Tutorials/Python Scripting/Scripts for modifying meshes
Previous page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Import scripts

Scripts for modifying meshes

Next page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Creating a GUI for your script
Previous page: Blender_3D:_Noob_to_Pro/Advanced_Tutorials/Python_Scripting/Procedural_object_creation

---

(to be written)

Also see saltshaker (http://saltshaker.sourceforge.net/) a basic but function python script for

blender, page includes details of how it was made.

Next page: Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Creating a GUI for your script
Previous page: Blender_3D:_Noob_to_Pro/Advanced_Tutorials/Python_Scripting/Procedural_object_creation

Creating a GUI for your script

Next page: Blender_3D:_Noob_to_Pro/Advanced_Tutorials/Advanced_Animation/index
Previous page: Blender_3D:_Noob_to_Pro/Advanced_Tutorials/Python_Scripting/Scripts_for_modifying_meshes

---

It is very easy to create a GUI for your script, and that way make it easy to change aspects of it for everyone.

(to be written)

Next page: Blender_3D:_Noob_to_Pro/Advanced_Tutorials/Advanced_Animation/index
Previous page: Blender_3D:_Noob_to_Pro/Advanced_Tutorials/Python_Scripting/Scripts_for_modifying_meshes

Advanced Animation

Next page: Introduction
Previous page: Creating a GUI for your script

This section will show you the Animation system as it is in Blender 2.4. Most of the features will be explained and some tutorials will follow. I assume the user has a good understanding of Blender here.

This text is based on a presentation I did at the Montreal Blender Conference. I hope you'll find it useful and instructive.

Gabriel Beloin aka --Gabio 23:59, 31 October 2005 (UTC)

If you wish to discuss it further: Visit us at Elysiun: Animation Workshop 2 (http://www.elysiun.com/forum/viewtopic.php?p=511798#511798)

Index

Next page: Introduction
Previous page: Creating a GUI for your script

Introduction

Next page: Guided tour
Previous page: Advanced Animation

Welcome to the wonderful yet complex world of computer animation! Through these pages I will try to show you everything old and new about the new animation system in Blender 2.4. But, before we get started, there are some basic notions about datablocks you should know. Animation in Blender is based on the fact that you have something moving in a Blender scene. For example, a ball bouncing on a floor plane:

-So you have a scene datablock, which holds some info about the scene itself, as you can see in the Render button window (F10KEY). -You populate this scene with various objects (which in this case refers to containers for data, not the actual mesh data that shapes the object itself). The only goal of an object is to hold the whereabouts of the data you want to see in your scene. It also holds the object *instance's* properties such as "does it have soft body or particle options, and do we draw its name?". Most of the info on an object can be seen in the Object Window (F7KEY).

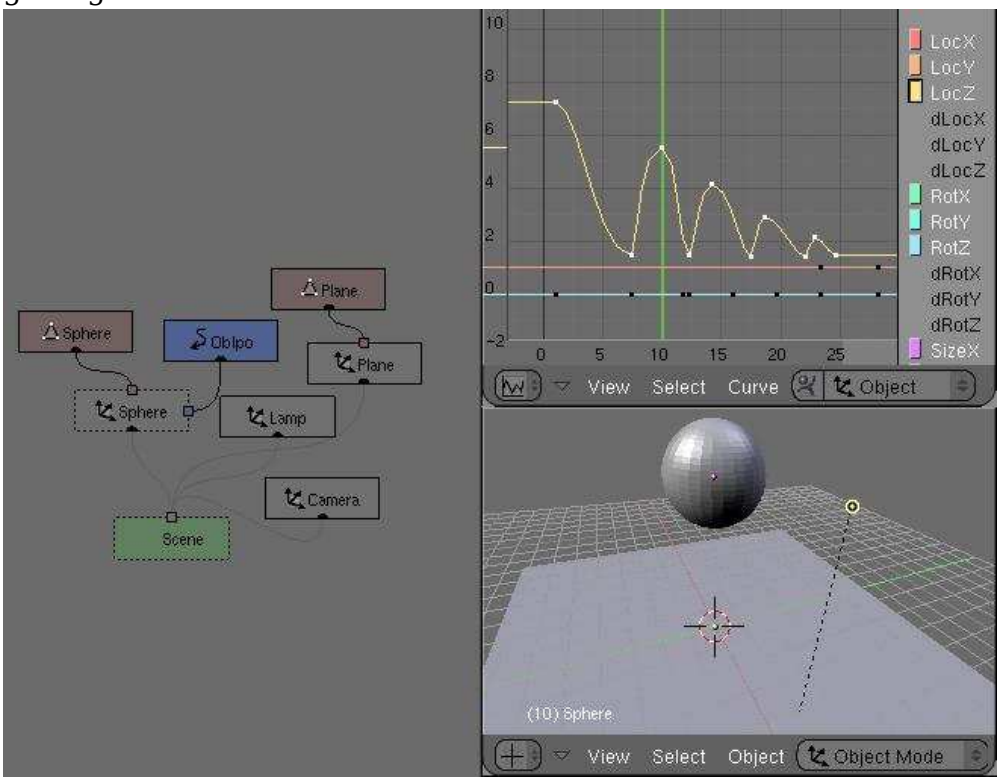
An object links to all of the data you can see in a 3D view such as **mesh**, **curves**, **nurbs**, **lattices**, **armatures**, **metadata**, the **empty** property, text, camera and lamps.

So the ball you just added to the scene is in fact a mesh, linked to an object that is in turn linked to the current scene.

Now there are also data blocks you can't see in 3D view, such as **material**, **texture**, **Ipo**, **action** and **image**. Instead, you have a special window in which to edit them. This is the idea behind the Blender interface, wherein each data block has a window for you to edit the data.

So back to this bouncing ball: It's also moving across the plane! So an ""Ipo"" data block is linked to the object, telling it where in space the object will be at each frame of the animation. This Ipo is editable in the Ipo window when selecting the ball in 3D view. In Blender, the work you are performing is always on the currently active (selected) object and data.

Looking at the OOPS (object oriented programming system) view (or SHIFT-F9KEY), we can get a good idea of the internal data structure:



Again, you are working in the scene "Scene", with an object "Sphere" linked to the mesh data block "Sphere" and the Ipo datablock "ObIpo". Why is that important? Because from there, you can start playing with the datablocks, linking them all around your projects to reuse old work. For example you can create more than one Ipo, and use the one you want, or tell more than one object to use the same Ipo, or to use the same object in more than one Scene.

Most of the linking job can be done in the Edit button window (F9KEY). Where you can tell an object to use another mesh's data block for Ipo, material, texture or image. There is always a little dropdown menu button for you to select an already-existing data block.

Now, when it comes to animation, you have to understand the way Blender handles data very well, because using Blender is always a matter of plugging data blocks together when working with Ipos, actions and NLA objects.

Next page: Guided tour
Previous page: Advanced Animation

Guided tour:

Next page: Armature Object
Previous page: Introduction

Here I'll show you all the stuff you need to know about the interface when animating. Where is it? How does it work? Why use it?

We are going to talk about:

- Armature Object
- Mesh Object
- Constraints
- Timeline Window
- IPO Window
- Action Window
- NLA Window

Next page: Armature Object
Previous page: Introduction

Armature Object

Next page: Armature Object in Object Mode
Previous page: Guided tour:

The Armature Object in itself is a tool for the animator to move an object or group of vertices in a reliable way. An armature is made of bones, which can be parented to each other, or connected to each other. It was built with the idea of a skeleton in mind.

You can add it using the SPACEKEY in 3Dview and selecting Armature. You'll then enter into editmode where you can add or move bones to build your default rig. An armature has 3 states. You can switch using the dropdown menu in the header of the 3Dview or use TABKEY to switch between Editmode <-> [Objectmode|Posemode] and CTRL-TABKEY to switch between Objectmode <-->Posemode:

- Object Mode: Your armature is like any other Object, you can move it around the scene, scale it, rotate it and edit options in the button window.
- Edit Mode: Your armature is in what we call rest position, you can modify the bones it contains.
- Pose Mode: Your armature is ready to be animated, each bone can be moved, scaled or rotated, constraints get applied, you can pose your character and animate the bones' behavior over time.
  Take note that Pose mode is now a state of the armature you can switch on/off using CTRL-TABKEY. So when in Pose, you are still in object mode (you can select another object, contrary to the editmode)

Note: The following 3 pages of this tutorial contain screenshots and discuss techniques that are only avialable in Blender 2.40a and later. Refer to the Blender 2.40a release notes (http://www.blender.org/cms/Blender_2_40_alpha.598.0.html) on Armature draw types and Armature envelopes.

Next page: Armature Object in Object Mode
Previous page: Guided tour:

Armature Object in Object Mode

Next page: Armature Object in Edit Mode
Previous page: Armature Object

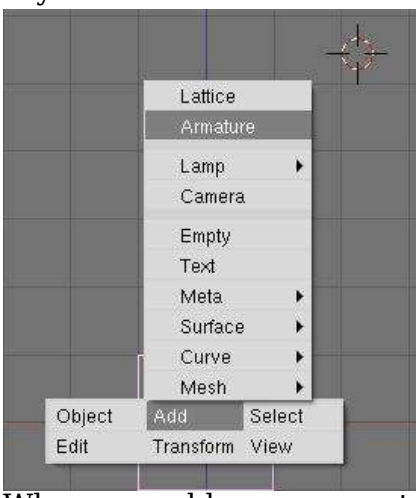
The Armature Object

Armature Object is like any other object type:

- It has a center, a position, a rotation and a scale factor.
- It can be edited.
- It can be linked to other scenes, and the same armature data can be reused on multiple objects.

- All animation you do in object mode is only working on the object, not the armature's contents like bones.

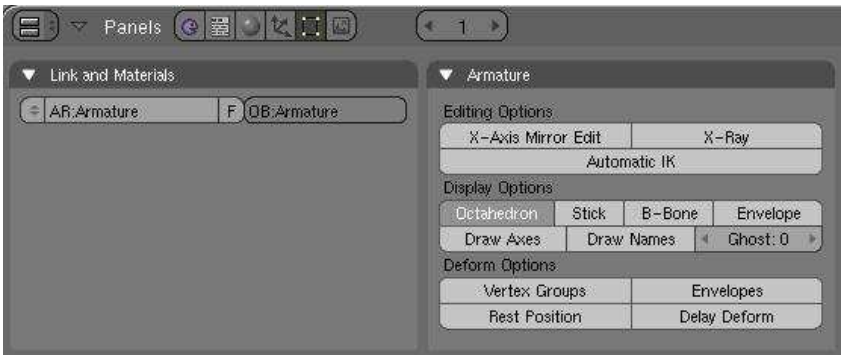
Try it now: add an armature to your scene: SPACEKEY --> Add --> Armature.



When you add a new armature, you'll enter editmode automatically. To switch between modes, use the TABKEY or the dropdown menu in the Header of the 3Dview window:

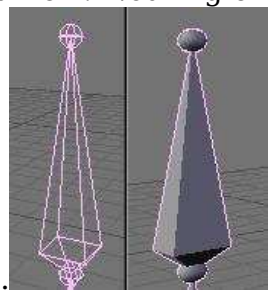
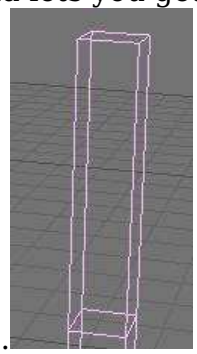
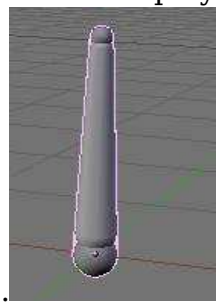
The Edit Panel When in Object Mode

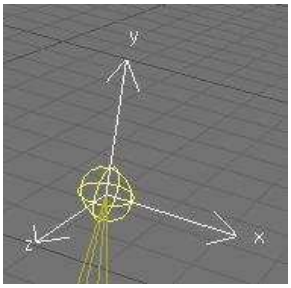
This is how the edit panel looks after you have added a new armature and switched to object mode (TABKEY):

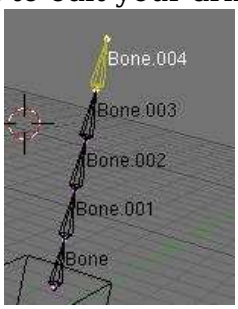


- Link and Materials panel:
  - The AR: field let you rename your armature Datablock. The dropdown is a quick way to select which Armature datablock you want to connect to this armature. You can keep more than one version for the same character. Useful when you have a special move to achieve in a shot, you can turn on an armature for a special purpose.
  - The F button is an option to assign a Fake user to the Armature. Again if you have more than one armature for your character, it's a good idea to turn the Fake on, because if your armature datablock is not used (linked) it's not going to be saved in your .blend files. You can always do batch Fake-assignment of armatures by opening the Datablock browser (SHIFT-F4KEY), go in Armature datablock, select all the armatures you want to keep, and Press the FKEY.
  - The OB: field is just to Rename your armature Object to something more cool and useful than Armature... Armature.001...

- Armature panel:
  - Editing Options:
    - X-Axis Mirror Edit: Not really useful now, it's more of an editmode option. This

feature tells Blender you want to replicate all of your bones on one part of the Armature to the other. It's a clean way to just do half the job ;). The axis of mirroring is X so left<-->right in frontview (NUMPAD_1KEY) and the center is the center of the armature object. We will see this feature in detail in the next page.
    - X-Ray: This option will let you see the armature through anything in the scene, solid or not. It's useful to see where your bones are in your character so you can select them.
    - Automatic IK is a Posemode option. It lets you pose a chain of bones as if the bone you were holding was an ik target. More info in Posemode page.
- Display Options: These options give you the chance to visualise your bones in various ways. Also note that there is some specific options and features regarding the display mode you're in.
    - Octahedron: This is the default view. Nothing exciting except you have a good idea of the rolling of the bones. 
    - Stick: This display mode is really useful when you have a lot of bones in your view. It lets you "unclutter" the screen a bit. It draws the bones as tiny sticks. 
    - B-Bones: It's more a feature than a display mode. This is only useful to visualise the effect you get when you activate the B-bones (Bezier-Bones). Each bone acts like a curve handle and lets you get extremely curvy poses. This will be exposed in the following pages. 
    - Envelope: Again it's more a feature than a display mode. But in this case the visualisation will be useful to tweak your rig later. Envelope lets you easily tell which part of you character this bone will animate and it's visually possible to change the zone of influence exclusively in this display mode. The zone is only visible in Editmode or Posemode though. 
    - Draw Axes: To draw the axes on each bone of the armature when you are in Editmode or Posemode. Useful when you want to know where you are and which axis to use in a constraint for example. Mental note: Y is up, Z is depth and X is side, contrary to object for which Z is up, Y is depth and X is side.



- Draw names: This lets you see names of bones whatever the mode you are in. It's useful again to edit your armature, create parent dependencies or add constraints. 

  - Ghost: This option lets you see a ghost of the armature in frames behind and over the current time. This is only working when you have an action linked to the armature, as we will see later.
  - Step (Armature_button_obj.jpg needs update): This option lets you choose the frames interval between ghost instances.
- Deform options:
  - Vertex Groups & Envelope: These two toggles let you choose if you want the armature to deform your character using the Vertex Groups and/or the Envelopes. We will see that later.
  - Rest position: This will bring the character back to factory default (item as Editmode), and no actions will be applied to the armature so you can easily edit it in the middle of an animation.
  - Delay Deform: This was useful before as the old system was *very* slow. What it does is when you do a manipulation to the rig, it waits until you finish to update the view. Can still be useful though.

Next page: Armature Object in Edit Mode
Previous page: Armature Object

Armature Object in Edit Mode

Next page: Armature Object in Pose mode
Previous page: Armature Object in Object Mode

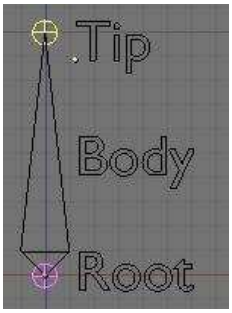
Now you've got your armature, but it's not much use until you add some more bones to it. Think about your body for a moment -- you've got this thing you call a 'skeleton', which for our purposes correspondings more or less to an armature object. Your skeleton consists of a number of bones (about 206, to be precise), but generally these are not independent from each other. If you move your femur (the bit of your leg between your pelvis and your knee) then conveniently the rest of your leg moves with it. In that example, the tibia/fibula would probably be counted as one bone, with the femur as their 'parent' bone. In this way, you build up a hierarchy of bones, making animation much simpler.

Editing an Armature Object gives you the chance to add, move or connect bones together. Whilst in edit mode, you will see all of the bones within the currently selected Armature.

When you create a new armature in Object mode a single bone will automatically be created for you, centered at the cursor position. Blender will then switch straight to Edit mode to allow you to add further bones. At this point we're just defining the default 'rest' position of the bones and specifying how they connect together -- you'll have to wait until the next

chapter to find out how to create specific poses.

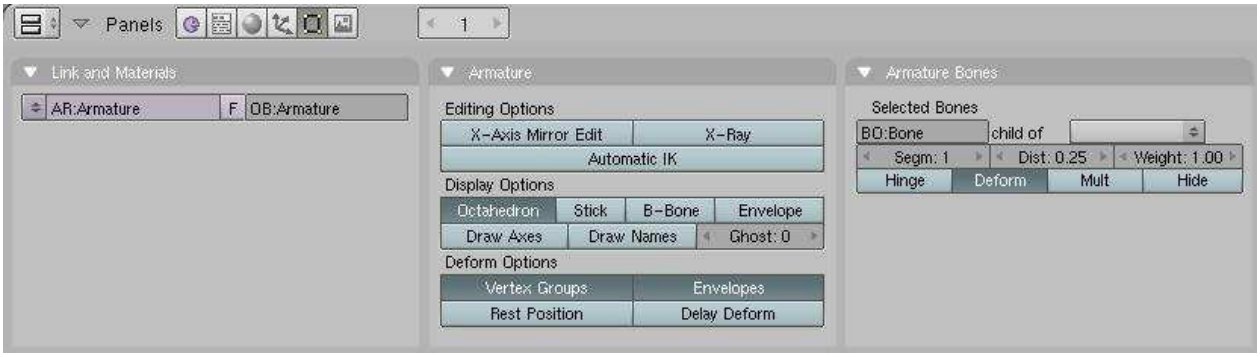
Now the basics about bones



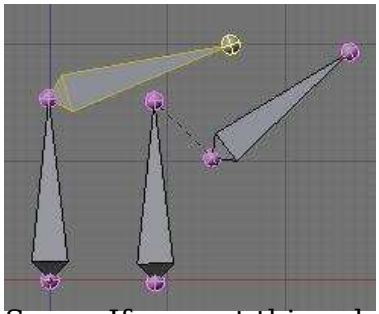
Having created and selected an armature in Object mode, you can add and modify the bones in this armature by switching to Edit mode.

- You can add a new bone at cursor position by pressing SPACEKEY in the 3DView --> Add --> Bone.
- A bone has two ends: a root (the lower part) and a tip (the upper part). You can select and move the tip or the root independently with RMB, or you can select the entire bone by clicking on its body.
- You can extrude a new bone from the selection using EKEY. This will create a bone connected to the original one, meaning the Root of the new bone will follow the Tip of the original one. You can also CTRL-LMB to extrude a new bone. It will extrude to where you clicked.
- Alternatively, you can connect two existing bones by selecting them one after the other and pressing CTRL-PKEY. You can then choose either 'Connected' (the child bone - the one you selected second - will automatically be moved so that it touches the parent) or 'Keep offset'.
- You can use SHIFT-DKEY to duplicate a bone
- Using the WKEY menu, You can subdivide your bone or flip the name of the bone between Left-Right (See Naming convention below).
- You can delete the bone with XKEY
- You can select a chain of bones (connected together) using LKEY, when you hover your mouse over a bone.

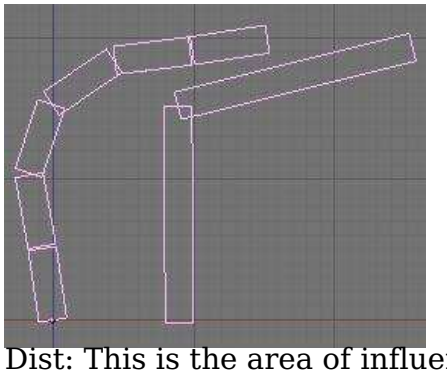
The edit panel



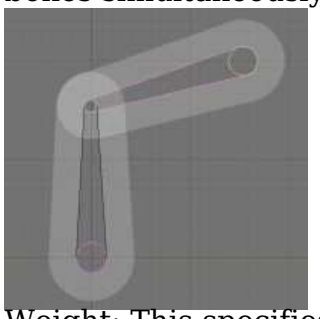
- Armature Bones Panel
  - BO: this field lets you rename your bone.
  - "Child of" Dropdown: lets you choose which bone will be the parent of this bone. If a parent is selected, there will be a small button labelled "con", meaning connected. Setting the parent defines the relationship between your bones. When one bone has another as its parent, it will do everything the parent does, such as rotate, move and scale. A dotted line between the parent and child will appear. If you select Connected, the Root of the Children will go stick to the tip of the parent, giving you a chain of bones like the 2 bones in your arm.



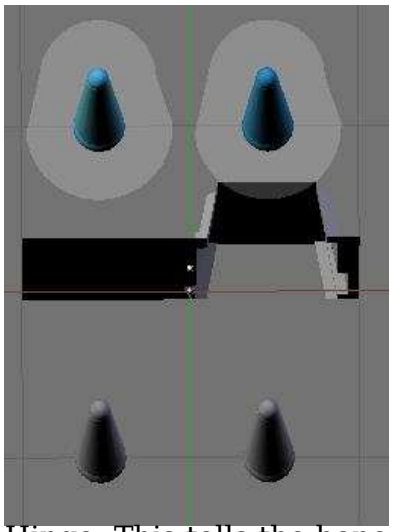
  - Segm: If you set this value to something greater than 1, it will cut your bone into several little segments and deform them on a bezier curve - referred to as a 'B-Bone'. You need to create a chain of bones to really show off this feature though. In the example below, the image on the right has 1 segment, and the one on the left has 3 segments each (these are shown in Object mode to show the effect more clearly):



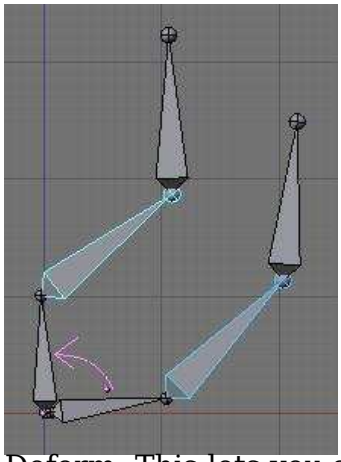
- Dist: This is the area of influence of the bone. It can be visualised using the Envelope display mode. We generally don't touch this field as there is an easier and faster way to change this option. Turn Envelope on and select a bone. Then using ALT-S, you can scale the zone of influence. This has the advantage that you can do it on multiple bones simultaneously, and it works in both editmode and posemode:



- Weight: This specifies how strongly this bone will influence the geometry around it, relative to the other bones. If two bones crossing each other, both with envelope influence, have the same weight (like 1:1) they will influence the surrounding geometry equally. But if you set one to 0.5, the geometry will be affected more significantly by the other one, with weight 1. For example, in this image, 2 bones using envelope influence try to move the same geometry. The 2 on the left have the same weight, you can see the geometry didn't move. On the right, one of the bones has 0.5 so the bone with weight 1 is winning the tug-of-war!:



- Hinge: This tells the bone to remain motionless in a chain. It doesn't copy the rotation and scale of the parent. Useful for mechanical rig I would say, as you can animate the rotation of the hinge bone without having to correct it because the parent rotated:



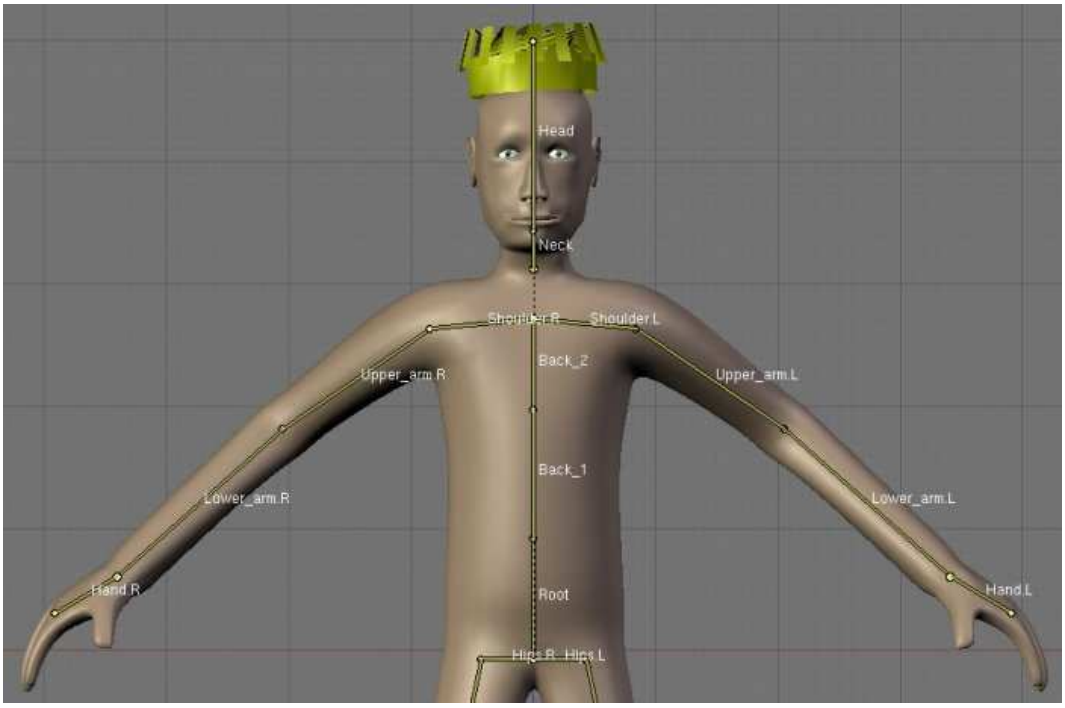
- Deform: This lets you say if you want the bone to deform the geometry at all. Switching it off is like setting the weight to 0, except it's faster this way. Useful when using a bone as a target or a controller, i.e. a bone you just want to use to control other bones, but not the geometry itself.
- Mult: to deform geometry you can use vertex group and/or Envelope. The ability to mix both of these methods is handy for using one to tweak the other. For example, you might use envelope everywhere but tweak difficult places manually with vertex group. We'll discuss this in more detail later on.
- Hide: This option lets you hide the bone. You can use it to hide the less important bones when you want to see what you're doing or for when you come to animate later on. For example, when you animate you don't need to see the entire chain of the leg, just the controllers. The values you select here apply to both Editmode and Posemode.

Naming convention

In many cases, rigs are symmetrical and can be mirrored in half. In these cases, it is helpful to use a left-right naming convention. This is not only useful for your own sake, but it gives Blender a hint that there is a pair of equivalent bones, and enables the use of some very cool tools that will save you some significant work.

- It's helpful to name your bones with something useful telling you what it's there for, such as leg, arm, finger, back, foot, etc.
- If you get a bone that has a copy on the other side, however, like the arm (you have 2 arms right?), then the convention is to call them arm.Left and arm.Right.
- Other alternatives are also possible, like _L, _LEFT, _left, .L, and .Left. Anyway, when you rig try to keep this left-right thing as accurate as possible; it will pay off later on.
- You can copy a bone named blah.L and flip it over using WKEY --> flip name. So the bone will be blah.L.001 after you copy it, and flipping the name will give you blah.R. Blender handily detects if the .001 version already exists, and increments the number for you.

This is an example of naming in a simple rig:



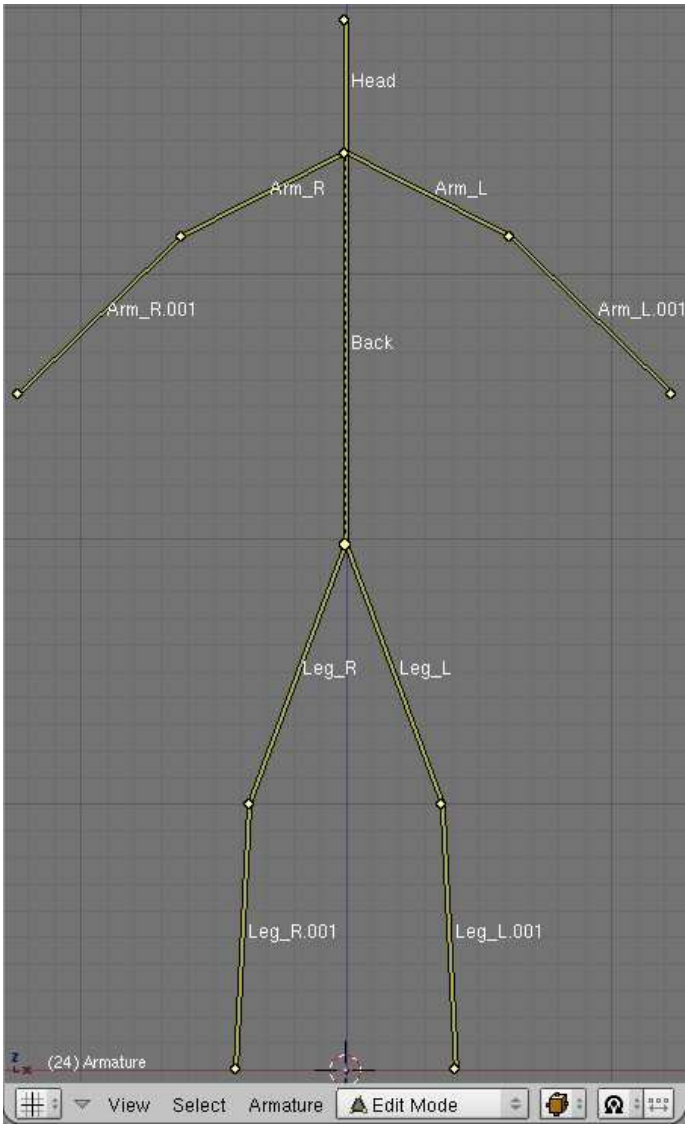
Mirror Editing

Now we come to the X-Axis Mirror Edit feature. This handy little number allows you to define only half of your character and tell Blender to automatically repeat the same actions on the other side. It's cool, it's simple and it saves a whole lot of time.

We will create a little guy out of sticks for the occasion -- don't worry about the geometry yet.

- Add a new armature to an empty scene. Enable 'Draw names' from the 'Display options' section of the Editbutton panel, so we can see what we're doing. You'll also need to enable the X Axis Mirror Edit mode that we're going to use, under 'Editing options'. Since, by definition, this feature mirrors along the X Axis, make sure you've got the front view selected (NUMPAD_1KEY) so that the X Axis runs from left to right. You'll also need to use the center of the armature (indicated by a purple dot) as the center of your rig, otherwise the symmetry will go wrong when we come to create the mirror image.
- Name the first bone you have "Back". You can scale it to make the entire back of the guy.
- Select the tip of this and extrude a new bone from it to do the Head. Name it Head.
- Select the tip of Back again and do SHIFT-EKEY to tell blender you're starting a mirrored chain of bones. Blender will automatically extrude another bone and will create an exact mirror of whatever you do. Take note that the name of both bones are Back_L and Back_R. Blender also tries to keep to the naming convention. Unfortunately, since we extruded from the Back bone, the names aren't quite right anymore.
- To change the names: Start by editing one of the names as Arm. Add the suffix to it (_L or _R). Then hover you mouse over the name field and do CTRL-CKEY. You just copied the name of the bone! Select the other bone, hover you mouse over the name field and do CTRL-VKEY. This will paste the name as-is. But as there is already a bone with the same name, Blender will add .001 after it. No problem; just go into 3Dview and do WKEY --> Flip name. There you have it -- a working mirror again.
- Mirror editing works using names. If you move a bone named blah_L and there is a bone named blah_R in the same armature, Blender will mirror the move you do to it, so make sure you follow name convention correctly.
- Then we can continue: extrude an other bone to make the lower part of the arm using EKEY or CTRL-LMB. The new set of bones should be arm_L.001 arm_R.001.
- Then we will add the legs. Up till now we have always worked from the tips of the bone. This is easy as blender understand you want to create children of the selected bone, but to make the legs you need to extrude from the root of "Back". So go ahead, select the root of "Back" and do SHIFT-EKEY to start a pair of chains. Rename them to "leg"+suffix.
- Now take note that doing so will not parent or connect the new bones to anything. We don't want it to be connected to the tip of "Back", it would look silly. But we want it to follow the body!
- The way to go is to parent the two leg we just created to the "Back" bone. The old way (pre 2.40) was to select all bone and select the parent manually in the drop down. There is an active bone and a selected bone now in editmode and posemode. The active bone is

the last you selected. Selected bone are all other selected bone. In this case we can't work with more than 2 bones selected. Select the child (a leg) then select the parent (Back) and Do CTRL-PKEY. A menu will popup asking connected or "keep offset". The difference is as simple as connected or not. For now use "keep offset" so you just parent it. Do it for each leg.
- it's also possible to remove parent easily. Select any bone you want to remove parent relation from and do ALT-PKEY. A menu will popup asking if you want to clear all or just to unconnect. Of course you don't need to select the parent and/or the child for this to work since any parent relationship will be cleared. So if you do that on a bone which is parent of 5 bones, then immediately all the children will be parentless.
- Extrude one more time to get a leg with 2 bones.
- Turn on the Stick display mode and enjoy your guy made of sticks!



- Now you can go into Posemode and pose your guy as you want.
- You can move the entire guy just by moving the "Back" bone, since this is how we built him. This bone is the highest in the bone hierarchy, "The daddy of all bones", you could say!

Next page: Armature Object in Pose mode
Previous page: Armature Object in Object Mode

Armature Object in Pose mode

Next page: Mesh Object
Previous page: Armature Object in Edit Mode

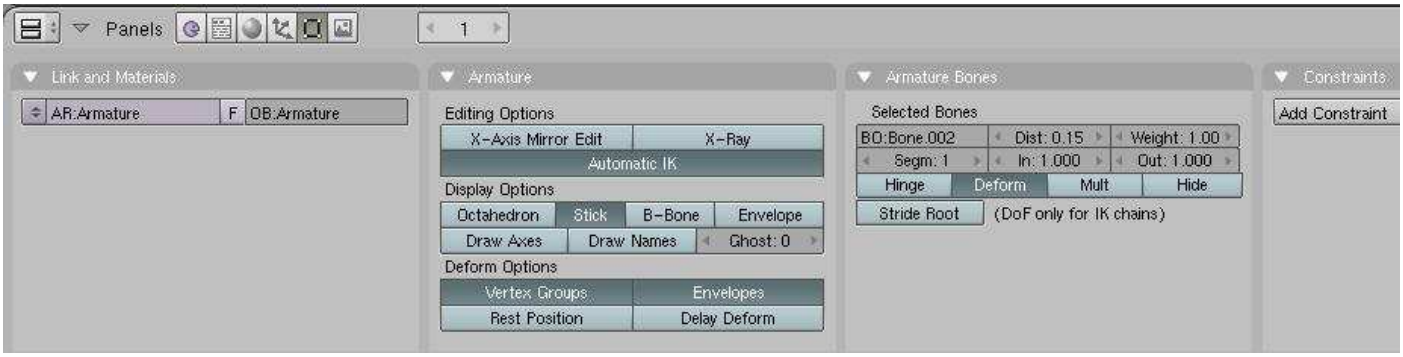
Posemode is a very versatile place where you Animate your character, create and manage constraints and apply your rig to your character.

Contrary to Editmode, Pose mode isn't an obligatory mode where you can't do anything else. It's now part of the UI like any other object. A good example of it is you can be in posemode and still select another object.

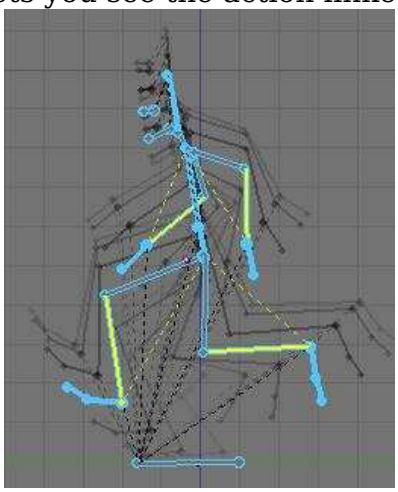
So What Can You Do?

When you are done building your armature, you can go into Posemode to add constraints and start creating actions. There are also some new tools accessible in Posemode that you may want to look at. You can easily get into "pose" mode by selecting the mode from IPO type list box in the left portion of the lower screen.

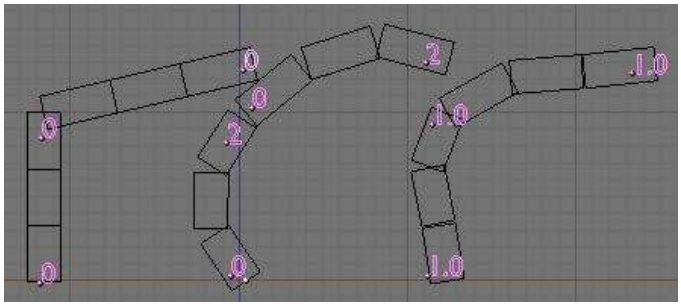
The panel has changed a bit too:



- What's new in the panels?:
  - You can use the Automatic IK feature in the Editbutton(F9) to pose a chain of bones like it was an ik chain. It's usefulness is very limited though. It works well only if there is no other ik solver in the chain, and if your chain is isolated from the rest of the rig.
  - Ghost: in the armature panel the ghost option lets you see the action linked to the armature over time. Also called onion skinning.

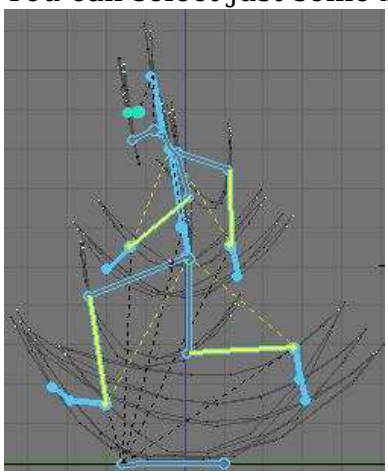


  - There are two number fields to better tweak the effect of B-Bones. The in/out is used to tell the scale of the virtual handle of the besier curve. In is the Root of the bone, and Out is the Tip. The bigger the value, the bigger the effect of rotation.



- There is now a Constraint panel where you can add a constraint to a bone, like any other object in the scene. This will be shown later.

- You can pose your rig using GKEY, SKEY and RKEY. Note that if the bone is part of a chain it can't be moved (except if it's the first of the chain, moving all the chain as they are all children), so you rotate the bone instead.
- You can do ALT-SKEY on one or more bones while in Envelope display mode to tweak the envelope size in real time while animating. Useful when for example you move the hand and some part of the character isn't in the influence zone; the result will be that some vertices will stay behind.
- You can do CTRL-CKEY to copy stuff from a bone to bones. The options are location, rotation, scale and constraint. Constraint is very handy when you wan to copy a constraint to other bone. The way it works is easy.
- The WKEY menu get some neat options too:
  - Select constraint target: Will select the target of the bone's constraint currently selected.
  - Flip name: Yep, you can flip name in Posemode too.
  - Calculate/Clear path: This is a visual way to see the action linked to your armature. You can select just some bones and ask Blender to show you the paths of the bones.



- You can pose your character and select all bones you want to see included in the action and press IKEY. You can insert a key just for loc, rot or size. Avail will add a key to all available channels in IPO window (all channels you previously added something).
- When you insert key for your armature, a new action is created and linked to the armature if there was no action before. You can also see the curves of each selected bone of the armature in the IPO window. We will see action window and IPO window later.
- You can parent a bone to an external object by selecting this object then selecting the bone in question so it's active (The armature is in Posemode so you can select a bone). Do CTRL-PKEY. Then when you move the bone the object will follow. This kind of Hard relationship doesn't include any bending at all. It's useful when doing robot rigs as you are just moving objects around.

Next page: Mesh Object
Previous page: Armature Object in Edit Mode

Mesh Object

Next page: Connection between Armature and Mesh

Previous page: Armature Object in Pose mode

This section will explain you how to deform your mesh using the armature.

There are two ways to tell Blender which vertex will go with which bone: Vertex group, and Envelope.

There is also a tool useful when animating which is part of the mesh object: the Shape key, to create a preset deformation. For example: deform the face to look like a smile.

- Connection between Armature and Mesh
- Envelope
- Vertex Groups & Weight Paint
- Shape Key

Next page: Connection between Armature and Mesh
Previous page: Armature Object in Pose mode

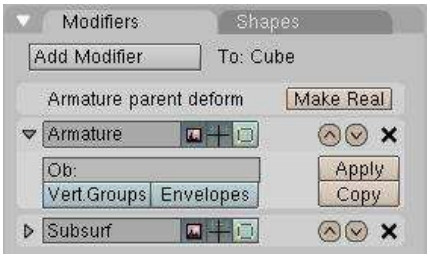
Connection between Armature and Mesh

Next page: Envelope
Previous page: Mesh Object

How to tell Blender: "use this armature to deform this mesh"?

The Armature Modifier

Blender now has a Modifier stack (Editbutton, F9KEY). As such, we should use it over existing methods to pair mesh and armature, as the modifier stack is optimised and simple to use. Note: You don't need to parent the mesh to the Armature anymore. The only case you could need to do this would be animating the Armature object itself. Then the mesh should also follow the armature. In this case select mesh, then armature, and do CTRL-PKEY --> Object.



The clean way to do so is to go in the Editbutton window (F9KEY) and press "Add modifier" in the Modifier panel, then select "armature" in the dropdown menu. Then you'll get a new modifier "Armature" like the previous picture. There you can change the name by clicking on the name field, enable/disable the modifier when rendering, enable/disable when working to only move the armature (could get handy with massive character), and when editing (that's very handy, you can edit the topology while it's deformed). There are also two toggles to tell Blender what it should use to deform: Vertex Groups and/or Envelopes. You may have noticed these options are repeated also in the Editbutton --> Armature panel, but as the tooltip says: these two are used when you use virtual modifier (the old way) to keep compatibility with old files.

Parenting the mesh to the "armature" will create an old-way link, still visible in the modifier stack, but not very useful. The first entry with the "make real" button is what appends if you

do a CTRL-PKEY to "armature". You should not use that kind of connection when you see that. Press "make real" to get a working modifier.

The Old Way

This way is not recommended but can still be useful. When doing CTRL-PKEY to "armature", you will get a menu like this:

- Don't Create Groups will just create a virtual modifier so you can deform the mesh (the "make real" button)
- Name Groups is almost useless now as blender will create a group for you when you do weight painting.
- Create From Closest Bones is a function to remember when you want to bake all your envelopes to vertex groups.

Tip: Bake envelope to vertex groups

The workflow is very simple. When you are done with the envelope's tweaking and you have gotten the best out of it, delete the Armature modifier and parent the mesh to the armature(CTRL-PKEY). Parent it to "armature" when asked and "Create From Closest Bones". Do ALT-PKEY and redo the Armature modifier. Now all the envelope influence are converted to Vertex Groups. This way you can further tweak influence zone using Weight paint. More info in the following pages.

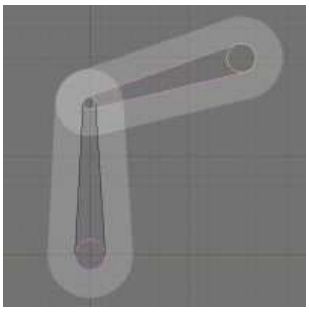
Next page: Envelope
Previous page: Mesh Object

Envelope

Next page: Vertex Groups & Weight Paint
Previous page: Connection between Armature and Mesh

What is Envelope

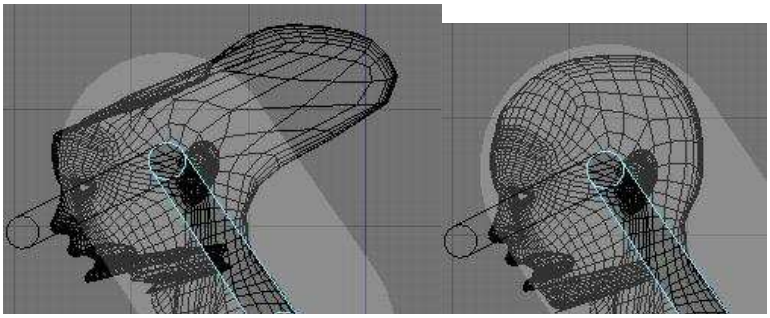
Envelope is a new visual tool to help you rig your characters faster and easier. It can often save you a lot of time. Each bone has a special area around it, allowing you to tell Blender what part of the geometry will follow each bone. This zone is customizable so you can move, scale and blend them together.



Edit Envelope

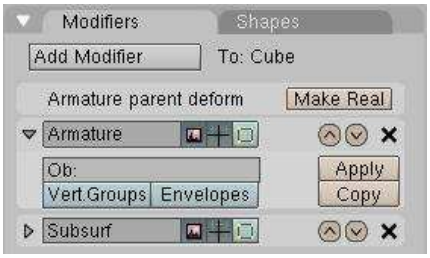
You can edit this white zone in Editmode or posemode by going in Envelope display mode, selecting bones and using SKEY or ALT-SKEY.

In Editmode: you can select the Tip, the Body or the Root and scale using SKEY. This area in the middle will assign a weight of 1 to all vertices contained in here. All vertices with a weight of 1 will completely follow this bone. The white tranparent area around the center is a zone of influence which loses power as you go away from the center. Example of influence (http://www.blender.org/cms/typo3temp/pics/e1c577807c.jpg) This area is scaled when selecting the body of a bone and doing ALT-SKEY. *In Posemode:* You can only scale the zone of influence with ALT-SKEY when in Envelope display mode. It's realtime, and lets you tweak the influence while you animate. So if you notice there is a vertex not following in the new pose you just did: Just select the bone it should follow, and scale the zone a bit until the vertex goes back with his friends. Example:

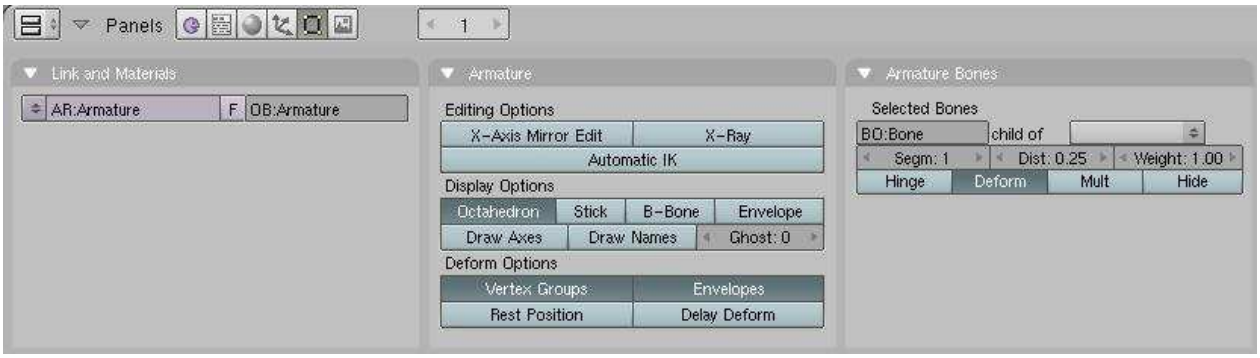


Envelope Options

It's possible to enable/disable the use of Envelope in the Modifier stack using the "Envelope" toggle.



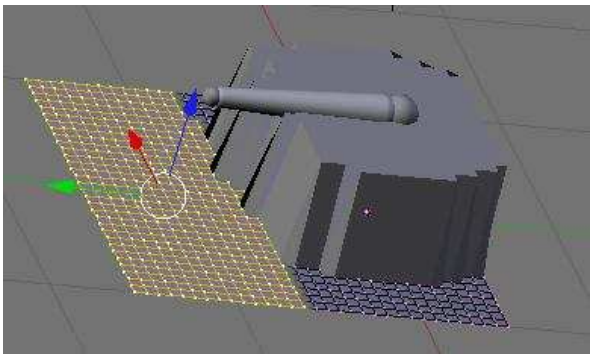
There are also two important buttons in the Armature Bones panel: Deform and Mult.



Enabling the Deform button will tell Blender to deform geometry with this bone. It's useful because in a more complex rig not all the bones are there to deform, some bones are just there to move other bones.

The Mult option will tell Blender to multiply the weight it get from envelope (let say 0.7) with the weight you painted in weight paint (let say 0.5). The result will be 0.5*0.7=0.35 so in fact you just tweaked the envelope influence to 0.3 when it was at 0.7. If you don't want vertices

to be part of the zone, you can always paint it with 0, as 0*(something) will always give 0. This way you can give custom shape to your envelope. More on weight paint on next page.



In this example you can see that all the selected vertices are not following the bone. This is because I painted a weight of 0 on them. In weight paint you'll see nothing. But just the fact that they are part of the group with a weight of 0 will make that possible. If Mult is off and you have both a vertex group and envelope, Blender will add value.

Next page: Vertex Groups & Weight Paint
Previous page: Connection between Armature and Mesh

Vertex Groups & Weight Paint

Next page: Shape Key
Previous page: Envelope

What Are Vertex Groups?

Vertex groups are very useful. You can use a vertex group to:

- Group vertices together while you model (keep a selection and come back to it later).
- Define which vertices softbody simulation affects.
- Define which vertices emit particles.
- Define which part of a mesh will follow a specific bone.

Vertex groups are specific to the Mesh object and can be modified in Editmode.

If you have vertices assigned to multiple groups (for example, in a character you may have some vertices in the "upper arm" vertex group that are also in the "lower arm" vertex group), you can assign weights to those vertices to specify how much relative influence the different groups have. A weight can range from 0 to 1 and is assigned when you create the group. Let's take a peek at the GUI of vertex groups in the Editbutton(F9KEY):

From top down:

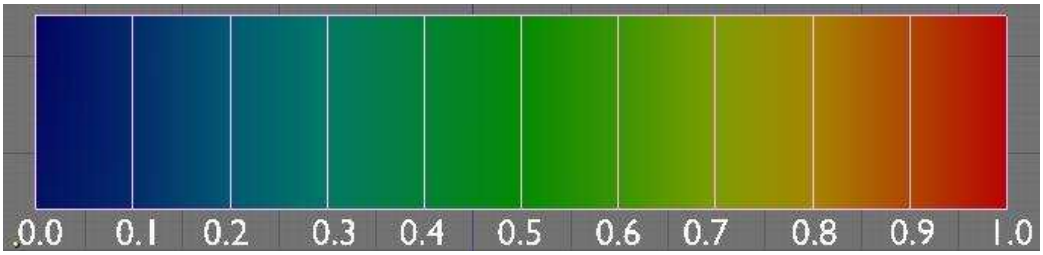
- The dropdown menu lets you select an existing vertex group or rename the current one.
- The weight numfield lets you choose the weight value assigned when you add vertices.
- You can add a new group or delete the current one.
- Assign or remove selected vertices to/from current group.
- Select/deselect all vertices in current group.

Weight Paint

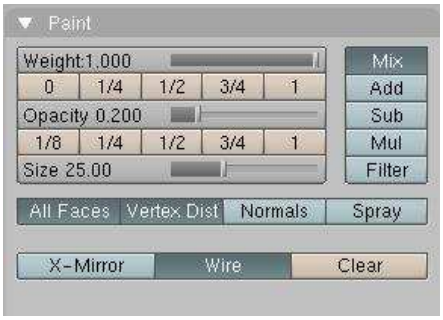
As mentioned above, you may often find that you have some vertices that are assigned to more than one vertex group. By assigning weights, you can specify the relative influence each of the vertex groups have. You have two options to assign weights: 1) manually selecting each vertex and typing in a weight value, or 2) use weight painting to - you guessed it - paint weights.

Weight painting lets you paint weight values on the mesh like you were painting on a wall with a can of spray paint. It is a Mode of the 3Dview and is accessible in the 3Dview's header in the dropdown menu with Objectmode, Editmode and such. A hotkey is also available: CTRL-TABKEY.

In Weightpaint Mode, the first thing you'll notice is the blue color of the mesh. Blender provides an easy way to quickly visualise the weight value of each vertex. This is the color spectrum used:



When you are in Weightpaint mode you can paint all over the mesh as if it was a solid object on your desk. The paint only works on vertices so don't try to paint in the middle of an edge or a face, it will never work ;). To help you in your task there is a new panel in Editbutton:



- The weight slider is just the same thing as the weight numfield we saw earlier in the vertex groups button. It's just easier to work with. It's simply the weight you want to apply to the vertices. In painting terms, think of this as the color.
- The buttons from 0 to 1 are shortcuts for weight value, to speed up the workflow.
- The opacity slider (and shortcuts) tell Blender what is the percent of the weight value you want to apply in one shot. If you set opacity and weight to 1 the vertex will turn red instantly. In painting terms, think of this as the pressure.
- "All faces" tells Blender if you want to paint on all faces in the mesh or just the visible one.
- "Vertex Dist" tell blender to use vertex distance instead of faces. When active, the painting will only check if the vertex is in the brush, then apply a weight value. If it's off, all vertice part of the faces in the brush will receive weight value. Turning on Vertex Dist can give good results when you have a lot of polys in you mesh.
- "Normals" will apply vertex normals before painting. This means Blender will take consideration of the direction the vertex is pointing when painting: the more it's facing away from view, the less it will receive value.
- "Spray" really makes it like spraying paint. Without it, a single click will only paint one value. With Spray on, each time you move the mouse a a paint shot will be added. To get good effect, use little oppacity value so the weight will top less faster.
- "X-mirror" will tell Blender to apply the weight paint on the other group if there is one. Like Hand.L --> Hand.R. If you paint the group hand.L and there is a hand.R the paint will be copied over. For this to work your groups must be created, the name of the groups have to follow name's convention (left right) and both side of the mesh need to be identical.

- "Wire toggle" toggles the visibility of wire while painting. Useful to find where the vertices are (activate the edit mode option "Draw all edges" to see even better).
- "Mix"/"Add"/"Sub"/"Mul"/"Filter" is how you want to apply the paint based on what is already there. Mixing will do a mean from brute weight value and current weight value, "Add"/"Sub" will directly add or subtract value, "Mul" will multiply (exponential painting) and "Filter" will paint based on alpha value.

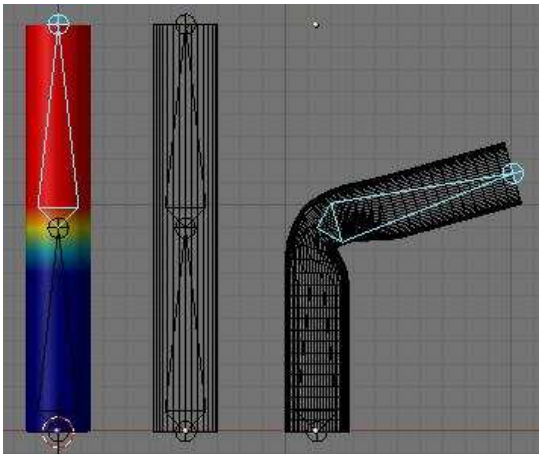
Vertex Groups and Armatures

So what use are vertex groups in rigging? You can specify what vertices will move when a bone moves. When you want to paint a mesh for an armature, do the following:

- Make sure the Mesh has an Armature modifier.
- Turn Armature into Posemode.
- Select the mesh and enter Weightpaint mode (CTRL-TABKEY).
- Select the bone you want to paint for with RMB.
- Paint what you want.

You'll notice that, if there is no group created when you first paint, Blender will create a group for you, and give it the same name as the selected bone. This is important, because when the "Vert. Groups" toggle is on in the Armature modifier, Blender will try to match bones with Vertex Groups based on the same names.

What happens when we try to blend groups together? See this simple example of 2 bones trying to bend a tube:

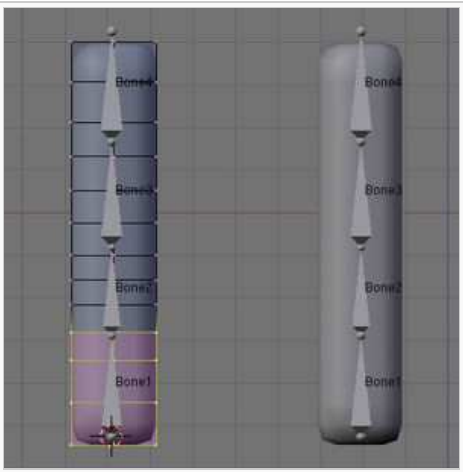


The Groups are painted so the body of each bone is in red and the zone between the two bones are gradually going from 1 to 0. This will bend nicely. If, for a special reason, you want a side to react differently, you can always move the bone while painting and try the new modification you just did. By the way, having Subsurf on while painting can be very cpu expensive. It's a good idea to turn it off.

Using Weight Painting with Armatures

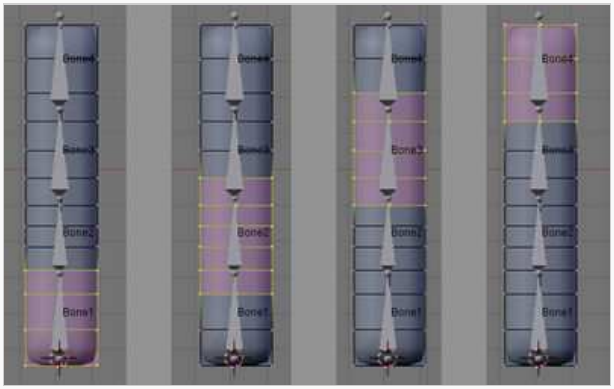
Armatures are used for many purposes, but one common use is to deform a mesh with an armature. This example will demonstrate how weight painting can improve armature-deformed meshes.

In this example, we have two objects; each has an armature modifier applied. The one on the left is going to be the "before" and the one on the right will be the "after".



The two objects in this example.

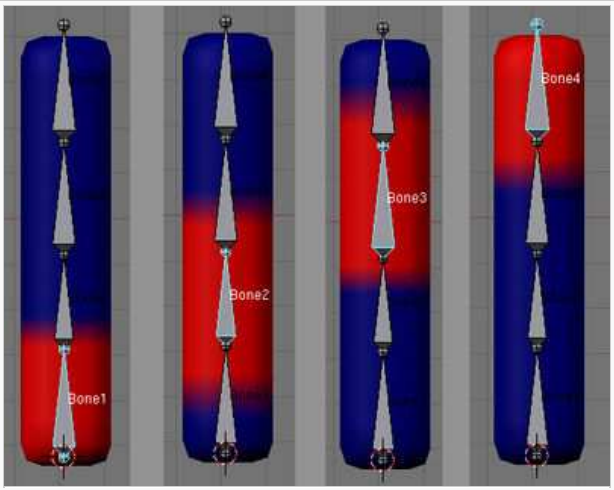
For the object on the left, take a look at the vertex groups as initially assigned (from left to right: Bone1, Bone2, Bone3, and Bone4). These same vertex groups were assigned for the object on the right:


Vertex group assignments for each of the two objects.

Important: A bone in an armature will only act upon those vertices that are in a vertex group with exactly the same name as the bone.

- In Blender 2.37 and previous, this was the ONLY way to get a bone to deform a mesh.
- In Blender 2.40 and on, selecting the "Envelope" button in the armature modifier will allow bones to deform even if you haven't assigned any vertex groups yet.

If you enter Weight Paint mode (CTRL-TAB with object selected) right after assigning the vertex groups, you can see that the vertex groups as assigned all have a weight of 1.0:



Initial weights for the vertex groups assigned above.

OK: both objects have vertex groups assigned and they have armature modifiers. Let's grab a bone (select the Armature, CTRL-TAB to enter Pose Mode, select Bone4, GKEY to grab, and move) to deform the mesh. We haven't made the objects different from each other, so after moving their armatures in the same way . . there's still no difference. That's good.


Armatures deforming objects: before weight painting

Next page: Shape Key
Previous page: Envelope

Shape Key

Blender Tutorials/Advanced Animation/Guided tour/Mesh/Shape

Constraints

Next page: Copy Location
Previous page: Lip-Sync with Shape Keys

The Constraint

A constraint is what makes everything easier, magic, automatic, customised (add more words here) in a rig. It tells a bone or an object to do something special based on the position of another object, and the position of the constrained object itself. There are many constraint types for you to play with. Most will work everywhere but, the IK solver will only be available in the Armature Editmode or Posemode.

There are no strict rules to follow for when to use constraints. As long as they save you time and make everything work by itself. A constraint should never be "time-consuming" or

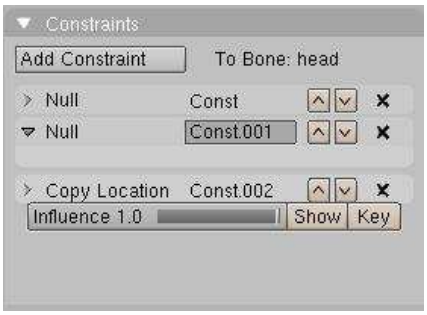
difficult to use. Think about the animator who is going to work with this rig (it could be you!). So, do everything in a smart way.

It's possible to copy constraints from one object/bone to a bunch of objects/bones. A useful thing to know when doing a repetitive task like rigging all the fingers of a hand. Just select all bones/objects that you want to give a copy of the constraint, and then select the bone/object containing the constraints. Press **CTRL-CKEY** in 3DView, and select Object Constraints from the popup menu. The idea behind this is to copy the constraints of the active object to the selection.

When working on an armature in Posemode, the bones will change color if they contain a constraint. Green for almost all, except for the IK constraint, which turns the bone Yellow.

The Constraint Panel

You can add a Constraint to an object or a bone by going in Object button window(**F7**) for objects and bones. Look for a Constraint panel like this (note, it's usually empty):



The panel also appears in Editbutton(**F9**) when you are in Armature Editmode or Posemode. So what you get:

- A button to add a new Constraint. The choice you have is listed down this page.
- When you add a new Constraint, A block get added in the stack. The UI is almost the same as the Modifier Stack. Each block represent an entry. You can delete it with "X", move it up or down in the stack, Close or open it.
- Constraints are calculated from first to last. So if you have two Constraints working on the same channel, let say Location, The last one will most probably win the chance to move the object. But...
- Most of the constraints have influence slider to tell how much it influence on the stack. If the last constraint have an influence of 0.5 it will mix the result with the one before.
- You can animate the influence of the Constraint by moving the time, changing the Influence and adding a key with the "key" button. The "show" button will bring the correct curve in the IPO window for you to edit it.
- You can change the name of the Constraint by clicking on the name when the constraint is open.
- By Clicking on the white jacket of the Constraint you select which one is active for edition, same as "show" button.
- If most of the Constraint you can enter the name of the Object you want to work with as a target or reference. For a bone, you need to enter in which Armature object it is, then an other field for the bone name will appear. When filling those fields, remember you can use autocompletion using **TAB**.

The Constraint Index

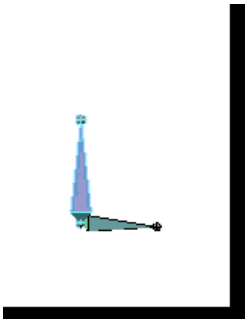
- Copy Location
- Copy Rotation
- Track-To
- Floor
- Locked Track
- Follow Path
- Stretch-To
- IK Solver
- Action

Next page: Copy Location
Previous page: Lip-Sync with Shape Keys

Copy Location

Next page: Copy Rotation
Previous page: Constraints

Copy Location



The Copy Location constraint does as the name states: it will copy the location of the target to the source (constrained object). Less Influence will drag the constrained object less and less to the target.

If it's an armature, a new field will appear to let you tell which bone will be the target. Don't forget TABKEY completion while writing the name of your object/bone!

You can tell Blender to work only on the selected axis. Many uses are possible :)

The Constraint Panel

- The Target field will let you select which Object the constraint holder will follow.

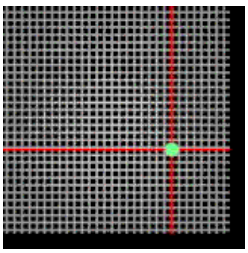
Where To Use It

Most of the time this little constraint is useful to stick objects to one another. By playing with the Influence you can tell when it will work, when it will remain motionless.

A good use of it is to ask a character to pick up something. By having a bone or empty for each side of the releationship (hand <-> glass), as the hand approaches the glass, you can align the two empties and fire the constraint up (1.00) to stick them together. You add another child-bone in the middle of the hand to tell where the glass will be. Thus moving the hand will move the glass. On the side of the glass just add an empty and make it parent of the glass. Add a copy location to the empty pointing to the bone in the hand we just did. There you go. Of course when the hand rotates the glass will not. For that you will need to add a Copy Rotation Constraint.

Before Blender 2.40, the above method was a good way of faking parent relationship without rotation. But now we have the hinge option which does the same.

Create this kind of tracking device using the X Y Z toggle button



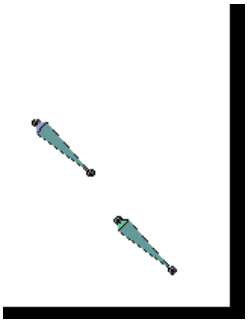
[Link to the Blend (http://satishgoda.com/blender/projects/TrackingDevice/feb0406_trackingDevice.blend)]

Next page: Copy Rotation
Previous page: Constraints

Copy Rotation

Next page: Track-To
Previous page:Copy Location

Copy Rotation



This constraint copies the rotation of the target. As simple as that. It can be an object or a bone. As you can see in the example, only the rotation gets copied.

The Constraint Panel

- You have 3 buttons to select which axis get copied over.

Where To Use It

Can be used with Copy Location to fake parent relationship. As you can key the influence you can make a character pickup something and holding it in his hands. Check the .blend for the hand-glass scene.

(?)link to the Blend(?)

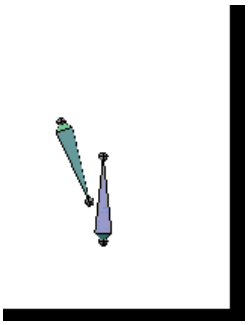
You can also use this to align a plane with a 2D effect on it to the camera at all times. This works better than pointing it at the camera in some cases, such as a ring of atmospheric halo around a planet, where you don't want it disappear behind the planet.

Next page: Track-To
Previous page: Copy Location

Track-To

Next page: Floor
Previous page: Copy Rotation

Track-To

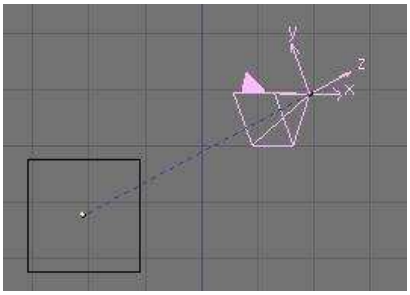


The Track-To constraint lets you influence the Rotation of the constrained object by making it track a target with one of the constrained object's axis.

The Constraint Panel

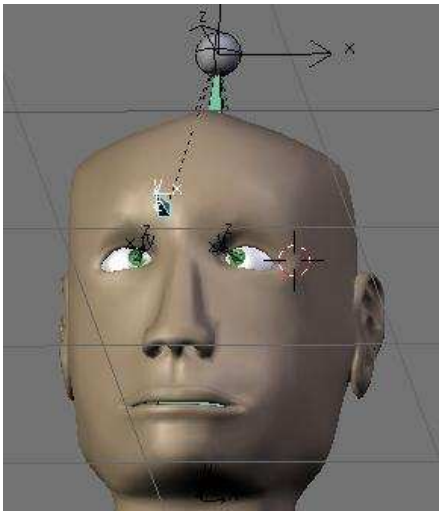
- You can enter the name of the target you want to track.
- The can select which axis is going to track the target.
- You can select which axis is going to stay up.

Where To Use It



A good example of use is the make a camera track an object. The setting to use on a camera is track: -Z and up: Y. You can turn Axis drawing in objectbutton window to help you choose the good axis.

Another example with armature would be the eyes of a character:



(?)link to the Blend(?)

Next page: Floor
Previous page: Copy Rotation

Floor

Next page: Locked Track

Locked Track

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/lt

Follow Path

Next page: Stretch-To
Locked Track

Follow path

Image:Ie cons

The Constraint Panel

Image:Panel

Where To Use It

Image:Example

(?)link to the Blend(?)

Next page: Stretch-To

Locked Track

Stretch-To

Next page: IK Solver
Follow Path

Stretch-To

Image:Ie cons

The Constraint Panel

Image:Panel

Where To Use It

Image:Example

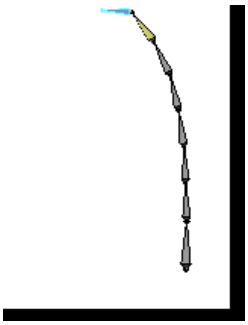
(?)link to the Blend(?)

Next page: IK Solver
Follow Path

IK Solver

Next page: Action
Previous page: Stretch-To

The IK solver



The IK solver constraint is a wonderful tool for all animators. IK stand for "Inverse Kinematic" and is the opposite of FK (Forward Kinematic, Duh!).

- FK: You have a dependancy to the root of the chain. In Blender, a FK chain is a chain of bones connected together. You have to animate the bones rotation one by one to get it animated. It takes longer, but gives you entire control over the rig.
- IK: Both ends are roots. All bones in the chain are trying to rotate to keep both ends on targets. Now this Constraint got most of the attention durring Animation refactoring,

hopefully we have a lot of toys to play with now.

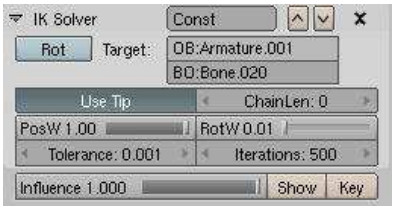
The IK solver has a special shortcut in Posemode to be added easily to a bone. If you select a bone and press '**CTRL-IKEY**', You get a little menu asking for more info on the new constraint, the target: to a new empty object or without target. It's now possible to work without target. Though you have less freedom (no rot feature, difficult parent relationship).

You can also select the target and then the IK constraint holder and press **CTRL-IKEY**. With this way of selecting ensure that your target is selected, but the bone you want to apply the constraint to is active (the last one selected). The menu will then let you add a constraint to the current bone with a target. If the target would itself be part of the IK chain, you get an error message - so make sure the target bone is not connected to the bone you want to add the constraint to.

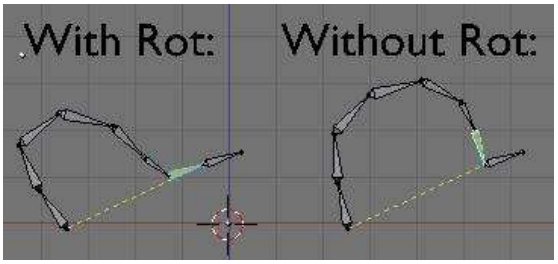
It's also possible to remove all IK constraints from selected objects or bones with '**ALT-IKEY**'.

USER: WHY my ctrl+i doesn't show menu ??? (ver. 2.41)

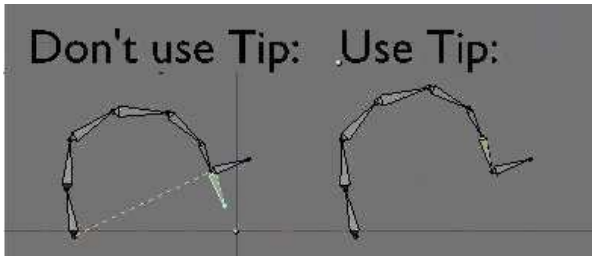
The Constraint Panel



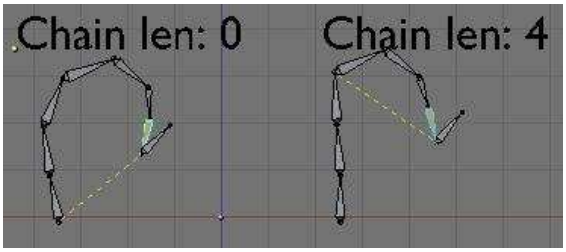
- You can rename the constraint.
- You can select which Object or bone will be the target. Don't forget Tab completion.
- The Rot button let you tell Blender to use the rotation of the target to influence the rest of the chain:



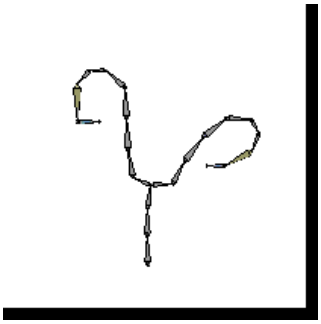
- The Tip button lets you tell Blender which part of the bone is the target, the Tip or the Root. It's interesting to use tip, because this way the Bone holding the IK constraint can be used to deform geometry.



- Len lets you tell Blender the length of the chain the IK solver will try to rotate. If set to 0, the entire chain will enter in the constraint. If for example the len is 4, only the 4 last bones of the chain will try to touch the target.

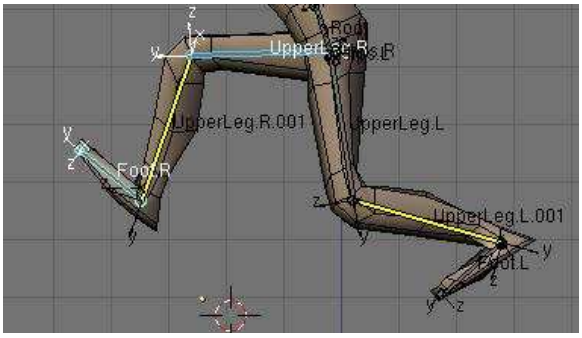


- Also If you set len to 0 and your chain's root is a child of another bone, The IK solver will reach and rotate all the bones until it gets to the end of the parent relationship. If all the bones are linked up to a master root, then all other sub-branchs will be affected. If there is another IK target in other sub-branchs of the rig, Blender will try to mix them. This concept of multiple IK targets in a rig is called Tree IK and can be used to get completely automated animations. For example like a doll: if you pull one hand, all the body will follow. In the 3D-view you'll see a yellow line from the IK solver to the root of the chain it covers. This line appears when you select the bone containing the IK solver.



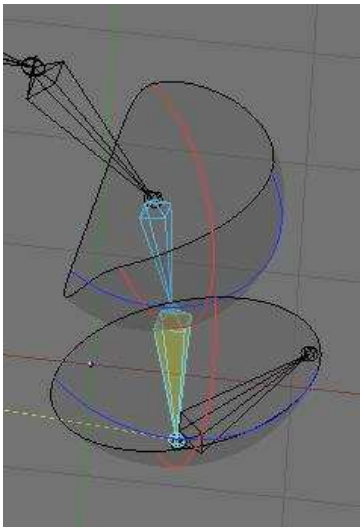
- PosW and RotW let you tell Blender if this IK solver will influence the final result more of less in the case of a Tree IK setup. With these options it's possible to use an IK solver just for location and an other one just for rotation.
- Tolerance and iterations are performance and precision options. IK solving is done in more than one pass, the more passes you calculate, the more accurate results you get. The tolerance is the distance from the IK solver to the target you can accept. If Blender manages to place the target near enough, it will stop doing iterations. The Iterations value is a hard limit you set to limit the time blender can reach on each IK solver per frame. Try to set it to a very low value to know why Blender needs more than one pass ;).
- You can set the general influence this constraint will have over bones, and it's animatable.

Where To Use It

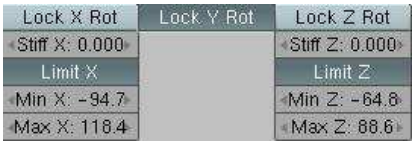


In any chain of bones you don't want to animate by hand but you want both end to be at precise location. The best example is a leg: The leg is connected to the body and to the foot. You don't need to animate the 2 bones in the legs, just place the body and the foot, the leg will follow automagically.

Degree Of Freedom



DOF are now possible to set for bones in a Ik chain. this way you can set what will block where. Very usefull when doing mechanical rig as you can limit the move or better, lock completely an axis.



- There you can set a limit on each axis, or completely Lock it.
- No limit gives it complete freedom (which is the same as [min:0 max:360] or [min:0 max:0]).
- The stiffness let you tell Blender if there is an axis more difficult to rotate than the rest. If all bone have a stiffness of 1 on X and you try to curve that chain in a way that all bones need to turn on X to follow the target, the Solving will find really weird poses to still touch the target without rotating on X.

Next page: Action
Previous page: Stretch-To

Action

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/act

Timeline Window

Next page: IPO Window
Previous page: IK Solver

Next page: IPO Window
Previous page: IK Solver

IPO Window

Next page: Data Type
Previous page: Timeline Window

Next page: Data Type
Previous page: Timeline Window

Data Type

Next page: Channel
Previous page: IPO Window

Next page: Channel
Previous page: IPO Window

Channel

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/ipo/channel

Curve Edition

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/ipo/curve

Driven IPO

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/ipo/drive

Action Window

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Action/index

Introduction To Action Data Block

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Action/intro

Key Edition

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Action/Key

NLA Window

(about this placeholder)

Since there has been nothing written on this page for a while, I asked for some basic information on another forum. I'll just quote what I learned from that forum, and this can be massaged into a real page of documentation over time. I won't attribute the individual various quotes, but thank you all for helping out. Remember, ANYONE can edit a Wiki, so create an account and make this thing better.

The NLA Window

Forum Notes

```
It's quite easy and maybe that's why there's no specific tutorial.
Let's say you want to make two actions, AC:Hit and AC:Kick.
Start with poseing Hit and an Action will automatically be created
in the Action Editor consisting of all the Bones that use Action IPO's.
That's done so return to Frame 1 which will be your default Stance of AC:Hit.

Now, in the Action Editor, click the X (delete) next to AC:Hit and the
datablock menu will disappear. (If you Add New instead of deleting then it
will copy selected bones to the new action and you don't always want that).
If you want a new default pose for AC:Kick, then Pose it or the same stance
will be used as was the default in AC:Hit. Pose and Keyframe your Kick
action and name it.

Over in the NLA Editor you can now use Shift-A to add NLA-Strips of your
Actions, Grab and Scale them and use the Transform Properties tab to input
how they Blend.
```

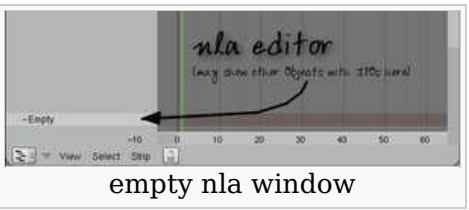
```
If you select an Action with the dropdown menu, at first its name will appear in the NLA window
along with its keys. To make this Action into an NLA strip, point at the Action's name in the
NLA and press CKEY.

Close any open Actions by clicking the scary X in the Action Editor. If
you don't do this, only this action will play. Now in the NLA editor,
play the animation.

If you see any keys (diamonds) in the NLA window, instead of strips
(rectangles), you're still editing an action. It's so much easier if you
have both the Action and NLA windows open so you can see whether an Action
is open or not. (Edited by CD38 23 Feb 2006)
```

Walkthrough

With no Actions selected, both the Action Editor and the NLA Editor appear empty. Here, the NLA Editor window does list one Object called **-Empty** because that object is not an armature but it has some IPO curves attached.



empty nla window

empty action window

Select an Action you've already made. Here, an Armature named **Yui** has bones involved in a one-frame action called **-AtEase**.

nla window with action

action window with action

Convert the listed Action to an NLA Strip in the NLA Editor by pressing the CKEY with the mouse hovering over the Action to be converted. No change in the Action Editor; it is still available as an Action.

nla window converting to a strip

nla window after conversion

Once converted, note the changes in the NLA Editor. The Action icon appears next to the Armature's name: . This is actually a button though it does not look like it, and you can toggle it between the Action and NLA Strips icon by clicking on it: .

More soon.

Introduction To NLA Editor

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/NLA/intro

Key Editor In the NLA

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/NLA/act

Strip Edition

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/NLA/strip

Strips Properties (NKEY)

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/NLA/nkey

The Stride feature

At the moment, the best documentation for Blender Stride features can be found here: Blender Stride Tutorial (http://www.telusplanet.net/public/kugyelka/blender/tutorials/stride/stride.html)

which takes on where the Official Blender Stride Page (http://www.blender3d.org/cms/Advanced_Stride_suppor.720.0.html) takes off. Good luck!

Working example: Bird

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/bird/index

Build The Rig

This page is a candidate for speedy deletion. The user who listed it for deletion gave the following explanation: "*no content, creates "deceptive" bluelink. Remove this tag is content is added*"

If you disagree that the page should be speedily deleted, please explain this on Wikibooks:Votes for deletion.

asdfasdf

Add Constraints

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/bird/const

Deform The Mesh

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/bird/connect2mesh

Create A Fly Cycle

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/bird/fly

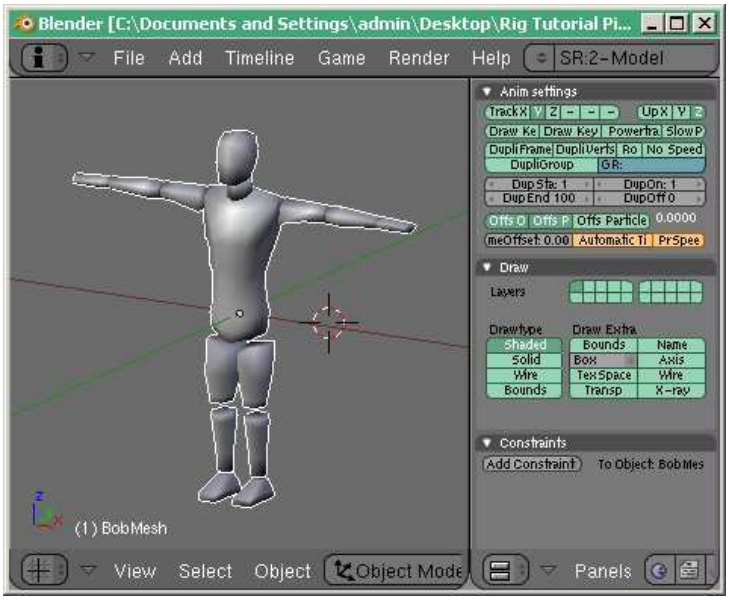
Working example: Bob

Next Page: Build The Rig
Previous Page: Fly

Because of my involvement with BSoD, this tutorial is not under development at this time. Instead, go here:
http://mediawiki.blender.org/index.php?title=BSoD/Introduction_to_Rigging

In this tutorial, we are going to be constructing a fully functional, humanoid character, with a complete rig, and we are going to animate him performing a walk cycle.

Getting Started



If you need a character model for this tutorial, you can download the model shown here (http://exenex.com/wavez/bob.blend) *(-- **404 File not found!!**)*

- Build the Rig
- Deform the Mesh
- Create a Walk Cycle

Build The Rig

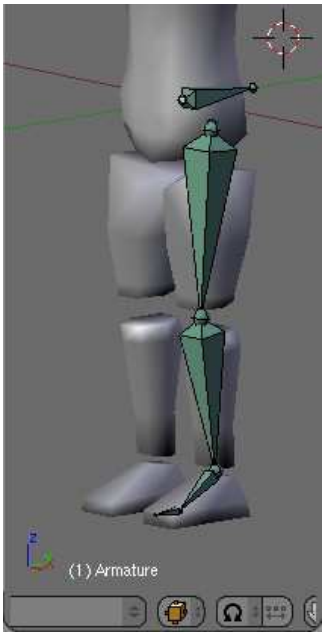
Next Page: Deform the Mesh
Previous Page: Working Example: Bob

This page is inactive due to my work for the BSoD

If you think you already know what a rig is, then you probably need to read the definition of "rig" and "rigging" before we get started. It's important to note that an armature is not a rig, but a rig can be an armature. Assigning a mesh to be deformed by an armature is not rigging.

A rig should always be designed for the types of animations your character is going to be performing. Only make your rig as complex as it needs to be to allow for the types of actions you need.

To make everything with the armature easy to deal with, we're going to make our character in the crucifix pose. If he's not, you will have headaches trying to deal with bone roll angles. Once Blender can easily allow the user to roll the bone to align with a roll target, then I'll edit this tutorial for that. But in the meantime, we will use vertical legs and horizontal arms. We will build the legs from the side view and arms from the top view.



Center the cursor (shift+c) and add an armature. In **Object Mode**, press alt+r to clear the rotation. You have to have a bone for the hip, and it needs to stick out of his front or his back, so take your pick, because they both look bad. Don't point the bone upward at some odd angle, we need to be able to roll the hips easily, and to that end, we will place the bone horizontally.

In front view, place the cursor and add a bone. **IMAGE**

In side view, move points and extrude them until your chain looks like this. Note the slight bend in the knee. This is very important! **IMAGE**

Snap your cursor to the root of this chain (shift+s) and add a bone. Now select the points at the hip joint and the ankle joint, and snap the cursor to the selection. **IMAGE**

Select the tip of the newest bone and snap it to the cursor. **IMAGE**

Now give these bones some names. It's a good idea to use the same names I do to avoid confusion, since I will refer to the bones by name. Select upperleg.l and then shift+select leg.l, and press ctrl+p to make upperleg.l the child of leg.l. Do this again, but make leg.l the child of hip. **IMAGE**

In front view, center your cursor and select

pivot point

Add Constraints

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/bob/const

Deform The Mesh

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/bob/connect2mesh

Create Shape Key

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/bob/shape

Create A Walk Cycle

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/bob/walk

Miscellaneous Tutorials

Blender 3D: Noob to Pro/Miscellaneous Tutorials

All Blender Tutorial Links

Back to Blender 3D

Here you can find useful **Blender Tutorial Links** in English language. Feel free to add some cool tutorial, but here only in *English*. For tutorials in other languages please read About. Tutorial about editing this pages is described here.

New in 3D world? Read To Those Learning 3D (http://glenmoyes.com/articles/to_those_learning_3d.shtml) .

Official Blender Documentation

Almost all **Manuals** you will find in latest official **Blender Manual Books**. For additional tutorials look below this topic.

- Blender MediaWiki (http://mediawiki.blender.org/index.php/Main_Page) * exposed
- Documentation at Blender3D.org (http://www.blender3d.org/cms/Documentation.628.0.html)

Interface

- User Interface (http://www.blender3d.org/_media/education/quickstart/Blender_User_Interface.html)
- Blender Interface Theme Repository (http://www.frontiernet.net/~krich/themes/)
- Basic Editing (http://www.ee.oulu.fi/~kapu/cg_course/bigfiles/blender/blendman.html)
- Blender Hotkeys
- Blender Hotkeys II (http://blender.excellentwhale.com/)
- Basic Hotkeys (http://viewhow2.qarbon.com/vf/vkxfjoo/4/hotkeys_viewlet.html) *(Java tutorial, broken link)*
- Navigating in 3D Space (http://www.blender3d.org/_media/education/quickstart/Navigating_3D_Space.html)
- Product Presentation (http://www.blender3d.org/Education/index_old.php?sub=TutorialPresentation)
- A short overview of functionality (http://www.blender3d.org/_media/education/quickstart/Blender_Windows.html)
- Basic Blender (http://www.planetannihilation.com/terragen/tut_basic.shtml)
- Rotation, Scaling and Layers Tutorial (http://www.blenderwars.com/tut-path.php?module=rotation)
- Appending Textures (http://www.b5-blender.com/tutorial4.html)
- Basic Blender (http://www.b5-blender.com/tutorial1.html)
- Basic Blender Interface (http://viewhow2.qarbon.com/vf/vkxfjoo/3/interface_viewlet.html) *(Java tutorial, broken link)*
- Basic Interface

(http://www.users.bigpond.net.au/blendage/pages/beginners_tutorials/tut1/basic_interfac
- Basic Loading and Saving (http://viewhow2.qarbon.com/vf/vkxfjoo/8/saveloadblend_viewlet.html) *(Java tutorial, broken link)*
- Basic Parenting (http://www.ingiebee.com/Blendermania/Basic%20Parenting.htm) *(broken pictures links: 7.3. 2005)*
- Basic Scene (http://viewhow2.qarbon.com/vf/vkxfjoo/5/basicscene_viewlet.html) *(Java tutorial, broken link)*
- Blender Multimedia (http://www.malefico3d.com.ar/tutor/audiovideo-en.html)
- Changing Views (http://blender3dfr.free.fr/anglais/tut2/tut2.htm)
- First Impression (http://blender3dfr.free.fr/anglais/premiere.htm)
- Getting Started (http://blender3dfr.free.fr/anglais/tut1/tut1.htm)
- Removing Orphan Vertices (http://membres.lycos.fr/bobois/Tuts/reperer_sommets_orphelins/sommets_orphilins.html)
- Seeing Textures with Alt-Z (http://membres.lycos.fr/bobois/Tuts/subdiviser_pour_conquerir_la_texture/subdiviser_pou
- The Camera (http://www.malefico3d.com.ar/tutor/camera.html)
- Stereoscopic camera for editing (http://brunetton.tuxfamily.org/index.php?n=En.BlenderStereo)

Mesh Modeling

- DupliVerts Tutorial (http://dev.newmediaworx.com/johnnyb/bplanet/tut_dupliverts.htm)
- Heightmaps (http://members.tripod.com/~funky_munky/tuts/blender/heightmaps.htm)
- Mesh Editing Techniques (http://vrotvrot.com/xoom/tutorials/Corridor/Corridor.html)
- Object Extrusion and Procedural Objects (http://vrotvrot.com/xoom/tutorials/mineRide/mineride.html)
- Loops Corkscrews Problem (http://membres.lycos.fr/bobois/Tuts/Courbes/tire_bouchon/tire-bouchon.html)
- Loopings Problem (http://membres.lycos.fr/bobois/Tuts/Courbes/Looping/looping.html)
- Knots in Curves Problem (http://membres.lycos.fr/bobois/Tuts/Courbes/Aplat/a_plat.html)
- ZeTool (http://membres.lycos.fr/bobois/Tuts/Courbes/ZeTool/zetool.html)
- DupliFrames Modeling I (http://membres.lycos.fr/bobois/Tuts/DupliFrames/Modeling/modeling_with_duplifames_e
- DupliFrames Modeling II (http://membres.lycos.fr/bobois/Tuts/DupliFrames/Modeling/pizza_boxes/stack_of_boxes_e
- Dupliframes (http://membres.lycos.fr/bobois/Tuts/DupliFrames/les_bases/les_bases_en_1.html)
- Spin and Spin Dup (http://membres.lycos.fr/bobois/Tuts/Spin_et_spin_dup/Spin_and_spin_dup_tutorial.html)
- Automatized BevOb (http://membres.lycos.fr/bobois/Tuts/Extrusion_par_bev_ob/Ext1_Ext2/Ext1_Ext2_en.html)
- Extrusion along a Path using BevOb (http://membres.lycos.fr/bobois/Tuts/Extrusion_par_bev_ob/Alignement_des_axes/Alignem
- Path Extrusion (http://membres.lycos.fr/bobois/Tuts/Extrusion_par_bev_ob/Les_bases/Les_bases_en.html)
- DupliVerts (http://membres.lycos.fr/bobois/Tuts/DupliVerts/Dupliverts/dupliverts_en.html)
- Limit DupliVerts (http://membres.lycos.fr/bobois/Tuts/DupliVerts/limiter_les_dupliverts_en.html)
- Types of Handles for Bezier Curves (http://membres.lycos.fr/bobois/Tuts/Courbes/Courbes_de_bezier/Types_de_poignees_bezi
- Blender Booleans (http://www.ingiebee.com/Blendermania/Booleans.htm) *(broken pictures links: 7.3. 2005)*
- Beveled Cube (http://w1.185.telia.com/~u18510119/tutorials/makeacube.pdf) *(pdf tutorial)*
- Joining/Separating Parts of a Mesh (http://membres.lycos.fr/bobois/Tuts/joindre_et_separer/joindre_et_separer_ang.html)
- Extrusion Controlled by IPO (http://jmsoler.free.fr/didacticiel/blender/tutor/en_modelextipo00.htm)
- Basic objects (http://blender3dfr.free.fr/anglais/tut3/tut3.htm)
- Modelling Techniques and Strategies (http://www.elysiun.com/forum/viewtopic.php?t=21382)
- Easily Remove Orphan Edges and Vertices (http://jmsoler.free.fr/didacticiel/blender/tutor/modesel_somparaz_en.htm)
- Non-destructive bevel effect (http://blendertips.blogspot.com/2006/03/bevel-modifier-workaround.html)
- "Cage" Technique Tutorial (http://www.cyphertxt.com/blendertechniquetut.php)

Nurbs and Subsurface Modeling

- IPOs, Lattices, Nurbs & Stuff (http://dev.newmediaworx.com/johnnyb/bplanet/tut_bottle.htm)
- Cross-Sections (http://vrotvrot.com/xoom/tutorials/Cave/Cave.html)
- Curve Resolution (http://membres.lycos.fr/bobois/Tuts/Extrusion_par_bev_ob/DefResolU/DefResolU_en.html
- Weight Parameter (http://membres.lycos.fr/bobois/Tuts/Courbes/Courbes_nurbs/Poids_des_voisins/Poids_des
- OrderU, UniformU and EndpointU (http://membres.lycos.fr/bobois/Tuts/Courbes/Courbes_nurbs/Influence_de_order/Influenc
- Curves -n- Bevels (http://www.ingiebee.com/Blendermania/curves%20and%20Bevels.htm)
- Subsurf Modeling (http://www.malefico3d.com.ar/tutor/subsurf-e.html)
- Making a Hole in Subsurf (http://membres.lycos.fr/bobois/Tuts/subsurfs/Trouer_une_surface/trouer_une_subsurf_an
- Subsurf Edge Rolling - Round (http://membres.lycos.fr/bobois/Tuts/subsurfs/rouler_un_bord/rouler_un_bord_rond_ang.h
- Subsurf Edge Rolling - Square (http://membres.lycos.fr/bobois/Tuts/subsurfs/rouler_un_bord/rouler_un_bord_carre_ang.h
- Modeling a fork (http://www.ingiebee.com/Blendermania/fork.htm) *(broken pictures links: 7.3. 2005)*
- Modeling with lattices - a fork (http://www.blender3d.org/documentation/htmlI/x10279.html)
- Subsurf modeling 2 (http://blender3dfr.free.fr/anglais/tut5/tut5.htm)
- Metaballs (http://blender3dfr.free.fr/anglais/tut6/tut6.htm)
- Abstract SubSurf modeling I (http://www.deviantart.com/view/26691152/)

Specific Object Modeling

- Double-Helix (http://www.deviantart.com/view/27132715/)
- Sword (http://alfisko.xhosting.cz/modellingsword.htm)
- Turtle (http://www.elysiun.com/forum/viewtopic.php?t=44686)
- Leaf Shader Tutorial (http://www.elysiun.com/forum/viewtopic.php?t=39810)
- Balcony and Fire Escape (http://www.project-newhorizon.net/bt_p5.htm) *(Broken picture links, 2/1 2006)*
- Oscilloscope (http://www.elysiun.com/forum/viewtopic.php?t=43289) *(MISSING?)*
- Sears-Roebuck Dairy Barn (http://www.harkyman.com/searsbarn01.html)
- Golf Ball (http://www.elysiun.com/forum/viewtopic.php?t=31781&postdays=0&postorder=asc&sta
- Celtic Knot (http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_celticknot_en.htm)
- Simple Tree (http://www.elysiun.com/forum/viewtopic.php?t=32306)
- Making a pedestal (http://web.telia.com/~u91121962/tut1_1.htm)
- Building a Castle (http://www.blender3d.org/_media/education/quickstart/Building_Castle.html)
- Ice Cube (http://caron.yann.free.fr/blender/IceCube.pdf) *(pdf tutorial)*
- Water I (http://www.selleri.org/Blender/tuts/Water.pdf) *(pdf tutorial)*
- Water II (http://www.selleri.org/Blender/tuts/Water2.pdf) *(pdf tutorial)*
- Space Pod (http://dev.newmediaworx.com/johnnyb/bplanet/tut_pod.htm)
- Grass (http://lib.hel.fi/~basse/blender/tutorials/grass/grass_tut.php)
- Bongo Creature (http://www.enricovalenza.com/makebongo.html)
- Spiral Stairs (http://www.blenderwars.com/tut.php?module=stairs)
- Waste Basket (http://www.blenderwars.com/tut.php?module=waste)
- Beveled Cube (http://www.blenderwars.com/tut.php?module=bevcube)
- Simple Box (http://www.blenderwars.com/tut-path.php?module=box1)
- Fountain with Moving Water (http://www.users.bigpond.net.au/blendage/pages/tutorials/tut5/fountain.html)
- Track Creation (http://www.geocities.com/swdoughty/blendertute1.html)
- Dolphin (http://vrotvrot.com/xoom/tutorials/Dolphin/UnderWater.html)
- Dice (http://vrotvrot.com/xoom/tutorials/Die/dice.html)
- Logo (http://vrotvrot.com/xoom/tutorials/logoTut/logoTut.html)
- Cutting through steel (http://vrotvrot.com/xoom/tutorials/Welder/Welder.html)
- Roller Coaster (http://membres.lycos.fr/bobois/Tuts/Courbes/completeRC/foreword_en.html)
- Roller Coster Cobra (http://membres.lycos.fr/bobois/Tuts/Courbes/Cobra/cobra.html)
- Modelling a Glass (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender_material/tp1-i
- Ocean View (http://www.ingiebee.com/Blendermania/create%20ocean%20view.htm)

- Solar Systems for beginners (http://www.ingiebee.com/Blendermania/Solar%20System.htm)
- Chair (http://membres.lycos.fr/bobois/objets_a_partager/a_simple_chair.html)
- Spiral Stair (http://w1.185.telia.com/~u18510119/tutorials/spiralstair.pdf) *(pdf tutorial)*
- Lighthouse (http://www.users.bigpond.net.au/blendage/pages/tutorials/tut1/lighthouse.html)
- Landscape (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/didacticiel-ang.html)
- Landscapes Easy (http://www.users.bigpond.net.au/blendage/pages/tutorials/tut2/landscape.html)
- Cloth Tutorial (http://www.free-webspace.biz/sonix/BlenderTutes/QuickClothTute.html)
- Cup (http://kahuna.clayton.edu/~jbrooks/blender/tutorials/cup/)
- Cola cane (http://free.hostdepartment.com/B/Blender3D/cantut/)
- Candle (http://download.blender.org/documentation/oldsite/oldsite.blender3d.org/93_Blender%2(
- Volcano (http://kahuna.clayton.edu/~jbrooks/blender/tutorials/volcano/)
- Orbital Logo (http://kahuna.clayton.edu/~jbrooks/blender/tutorials/universal/)
- Realistic Planet (http://www.enricovalenza.com/realplan.html)
- Landscape Cartoon (http://www.selleri.org/Blender/tuts/CartoonishLandscape.pdf) *(pdf tutorial)*
- Text (http://blender3dfr.free.fr/anglais/tut4/tut4.htm)

Human

- Head Modeling (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/head-subsurf/index-ang
- Head Modeling & Texturing Tutorial (http://kotinetti.suomi.net/fsware/hippie/tutorial1/index.php)
- Head (http://www.visiontovision.com/BlenderHead1.html) *(Part 1 of 12-part Flash Tutorial)*
- Get a Head (http://members.lycos.co.uk/duxbellorum/get_a_head.html) (link broken)
- Hand Palm (http://jlp.nerim.net/tutorials/hands-modelling/hands-tuto-01.html)
- LowPoly Leg Modeling (http://mywebpages.comcast.net/jmandmc/legtut/legtut.html)
- Make Human: Modeling a New Target (http://feeblemind.tuxfamily.org/dotclear/index.php/2005/04/11/23-didacticiel-modelisatio
- Eyes (http://web.pdx.edu/~wlf/tut.html)
- Hair Tutorial (http://www.elysiun.com/forum/viewtopic.php?t=16008)
- Real Hair with Blender 3D (http://www.prodigyweb.net.mx/nivel9/hair/realHair.html)
- Female Character (http://otothegardener.free.fr/tutorials/Femme/femme.htm)
- Clothes for humans (http://kokcito.tk/tut1) (link broken)
- Texturing Skin using Vertex Painting and Repeating Image Textures (http://pages.zoom.co.uk/nick.towers/tutorials/skin_tutorial/skin_tutorial.html)

- Wikipedia: Human Body Proportions (http://en.wikipedia.org/wiki/Body_proportions)

Cars

- Porsche 550 (http://perso.wanadoo.fr/speedtiti/tutoriels.htm) * exposed
- Car IV small (http://otothegardener.free.fr/tutorials/Voiture/voiture.htm)
- Car V F1 (http://otothegardener.free.fr/tutorials/F1/f1.htm)
- Creating a Toon Car (http://www.raivestudios.com/tutorials/blender/tooncar/)

Texture Mapping

- Making a Lightsaber with Halos (http://www.elysiun.com/forum/viewtopic.php?t=64227)
- The Unofficial Texturing Tutorial (http://www.elysiun.com/forum/viewtopic.php?t=11889)
- Material Indice Tutorial (http://feeblemind.tuxfamily.org/dotclear/index.php/2004/12/21/5-blender-indices-materia
- Raytraced Transparency and Refraction (http://feeblemind.tuxfamily.org/dotclear/index.php/2004/12/26/7-didacticiel-la-refraction-
- Chrome and Shiny Metal Surfaces (http://www.ualberta.ca/%7Enwy/blender/blender_tut1.htm)
- Texturing a castle (http://www.blender3d.org/_media/education/quickstart/Texturing_Castle.html)
- Texturing Tutorial (http://www.planetannihilation.com/terragen/tut_buildpics.shtml) (NOTE: For the game Total Annihilation, not really a general tut)

- Textures in OrCo mode (http://www.enricovalenza.com/textures.html)
- Textures for Allosaurus (http://www.enricovalenza.com/textall.html)
- Chrome Effect (http://www.elysiun.com/tutorials.php?id=8)
- Texturing Part I (http://www.geocities.com/woodsmith102000/Tutorials1.html?)
- Texturing Part II (http://www.geocities.com/woodsmith102000/Tutorial2.html?)
- Texturing a Ship (http://www.blenderwars.com/tut.php?module=texture)
- Material Indices (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender_material/tp2-i
- Textures Channels (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender_material/didac
- Textures Mapping (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender_material/didac
- Specular Color (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender_material/didac
- The Color (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender_material/didac
- Materials in Blender (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender_material/index
- Alpha, Bump, and Specular image textures (http://www.ingiebee.com/Blendermania/BumpSpecAlphaTex.htm)
- Image Textures (http://www.ingiebee.com/Blendermania/attach%20image%20file%20to%20object.htm)
- Decals (http://www.ingiebee.com/tutorials/Decal%20Mirror%20Modelling/theeth%20decal.htm)
- Chrome (http://www.ingiebee.com/tutorials/Digital-Mark.htm)
- Alpha Masks (http://membres.lycos.fr/bobois/Tuts/trouer_par_texture/trouer_par_texture.html)
- Skies I (http://www.malefico3d.com.ar/tutor/skies.html)
- Skies II (http://www.malefico3d.com.ar/tutor/skies2.html)
- Shockwave (http://www.elysiun.com/tutorials.php?id=5) *(broken link 1. 3. 2006)*
- Texture Types in Blender (http://www.elysiun.com/tutorials.php?id=4) *(broken link 2.23.2006)*
- Basic Texturing (http://www.b5-blender.com/tutorial3.html)
- Basic Texturing 2 (http://viewhow2.qarbon.com/vf/vkxfjoo/7/basictextures_viewlet.html) *(Java tutorial)*
- Textures with Alpha (http://www-users.cs.umn.edu/~mein/blender/tutorials/dust/alpha.html)
- Textures with Bumpmapping (http://www-users.cs.umn.edu/~mein/blender/tutorials/dust/bump.html)
- Using More Than One Color (http://www.users.bigpond.net.au/blendage/pages/tutorials/tut9/two_colours.html)
- Adding Color To Your Shapes (http://www.users.bigpond.net.au/blendage/pages/beginners_tutorials/tut3/adding_colour
- Using Textures and Bumpmapping Them (http://www.users.bigpond.net.au/blendage/pages/beginners_tutorials/tut4/adding_texture
- Texture Mapping Tutorial (http://dev.newmediaworx.com/johnnyb/bplanet/tut_mapping.htm) *(pictures ok now 18/03/2006)*
- Environment Mapping (http://dev.newmediaworx.com/johnnyb/bplanet/tut_envmap.htm) *(pictures ok now 18/03/2006)*
- Mapping Caracter (http://otothegardener.free.fr/tutorials/LittleOTO/anonce/anonce.htm)
- Creation and Mapping (http://membres.lycos.fr/bobois/Tuts/boite_de_projection/boite_de_projection005_en.html)
- Displacement Mapping (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/displacement/index-an
- Game Sprites (http://www.ingiebee.com/Blendermania/Game%20Sprites.htm)
- Stencilling Textures (http://www.ingiebee.com/Blendermania/Stencilling.htm)
- Using Texture Stencils (http://feeblemind.tuxfamily.org/dotclear/index.php/2005/02/27/16-didacticiel-usage-des-n

UV Mapping

- Intermediate and Advanced UV Mapping (http://www.elysiun.com/forum/viewtopic.php?t=25918) * exposed
- UV Mapping & Texturing (http://biorust.com/index.php?page=tutorial_detail&tutid=85)
- Sub-Surf and UV-Texturing problem (http://barney.gonzaga.edu/~amoore1/uv_mapping_project/)
- UV Texture Skinning (http://otothegardener.free.fr/tutorials/LittleOTO/littleoto3/littleoto3.htm)

- Character UV Textures (http://otothegardener.free.fr/tutorials/Armure/anonce/anonce.htm)
- UV Mapping a Head (http://mywebpages.comcast.net/jmandmc/uvtut/uvtut.html) (broken link 07.04.06)
- UV Texturing in Blender (http://www.planetannihilation.com/terragen/tut_texturing.shtml)
- Tuhopuu UV editor mapping (http://users.pandora.be/blendix/blender/uv/)
- UV Mapping (http://membres.lycos.fr/bobois/Tuts/uvmapping/uvmapping01.html)
- Complex material for a sword (http://67.15.36.49/team/Tutorials/swords&daggers/swords&daggers_01.asp)

Animated Textures

- Blender TexMesh Tutorial (http://www.telusplanet.net/public/kugyelka/blender/tutorials/texmesh/texmesh.htm) * exposed
- animating masks for simulating ice freeze (http://www.elysiun.com/forum/viewtopic.php?p=455683#455683)
- Animated Procedural Textures (http://www.cogfilms.com/tutorials.html) *(pdf tutorial)*

2D Texture Painting Techniques

This part isn't about using Blender, but about necessary **2D knowledge** for advanced Blender users.

- Steven Stahlberg's Tutorials (http://www.androidblues.com/howto.html) * exposed
- Dirty Metal Texture (http://div.dyndns.org/EK/tutorial/texture/)
- Gritty Pipelines (http://div.dyndns.org/EK/tutorial/gritty/)
- Digital Painting (http://www.planetquake.com/polycount/resources/general/tutorials/HitmanDaz_Tut01/Di
- Advance Painting and Weathering Techniques for Train Textures (http://www.cham-ministry.org/msts/tutorial1.html)
- NWN Texturing Tutorials (http://www.btinternet.com/~i.nation/tutorials/tutorial_00.htm)
- Fixing Lighting Irregulaties in Self-Tiling Maps (http://www.3drender.com/light/EqTutorial/tiling.htm)
- Texture Map for the Iris of the Eye (http://www.kandsdesign.com/kim/eyemap-tut.html)
- Airplane Texture Tutorial (http://www.geocities.com/SiliconValley/Haven/2470/txtrtut/txtrtut.html)
- Game Skinning Tips (http://www.planetquake.com/pandemonium/html/skintips.htm)
- Game Lighting Basics I (http://www.planetquake.com/pandemonium/html/skintut3.htm)
- Game Lighting Basics II (http://www.planetquake.com/pandemonium/html/skintut4.htm)
- Texturing with Gimp (http://otothegardener.free.fr/tutorials/GimpTexture/gimptexture.htm)

Lighting, Shadows and Rendering

- Light - a detailed tutorial (http://www.itchy-animation.co.uk/tutorials/light01.htm)
- Yafray as an Integrated External Renderer (http://wiki.yafray.org/bin/view.pl/UserDoc/GauravGuide)
- Basic 3 Point Lighting (http://www.andrew-whitehurst.net/3point.html)
- Radiosity I (http://blenderman.free.fr/tut/radiosity/uk/)
- Radiosity II (http://download.blender.org/documentation/html/chapter_rendering_radiosity.html)
- Ramp Shaders (http://www.blender3d.org/cms/Ramp_Shaders.348.0.html)
- Simulating Radiosity (http://jmsoler.free.fr/didacticiel/blender/tutor/lumiere_radios_en.htm)
- Caustic Sampler Tutorial (http://www.elysiun.com/forum/viewtopic.php?t=30265)
- Shadows Control (http://dev.newmediaworx.com/johnnyb/bplanet/tut_shadows.htm)
- Light Types (http://www.stormpages.com/eeshlo/otherBlender.html)
- The 'World' buttons within Blender (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/world/index-ang.html)
- Mastering Shadows (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/shadows/index-ang.htn
- Ambient Occlusion (http://feeblemind.tuxfamily.org/dotclear/index.php/2005/01/16/10-didacticiel-locclusion-a
- Area Lights (http://feeblemind.tuxfamily.org/dotclear/index.php/2005/01/02/9-didacticiel-les-aires-lum

- Batmanish Spot Logo (http://www.ingiebee.com/Blendermania/batman%20spotlight.htm) *(broken pictures links; 7.3. 2005)*
- Depth of Field (http://www.malefico3d.com.ar/tutor/dof-en.html)
- Radiosity (http://www.geocities.com/blengine/radiosity.html) *(Broken Link 11.19.2005)*
- Different Lighting Methods (http://www.geocities.com/blengine/lighting.html) *(temporarily(?) down 6.16.2006)*
- Soft Lights (http://www.selleri.org/Blender/tuts/SoftLight.pdf) *(pdf tutorial)*
- Blender's Mist (http://www-users.cs.umn.edu/~mein/blender/tutorials/dust/mist.html)
- Basic Lighting (http://www.katorlegaz.com/index.php?a=article&display=1075723200.php)
- Optimizing Renders (http://www.katorlegaz.com/index.php?a=article&display=1078185600.php)
- Flares (http://www.centralsource.com/blender/flares/)
- Toon Shading (http://free.hostdepartment.com/a/aner/TUTORIAL.html)
- Simple Gold Ring with Caustics (http://www.geocities.com/sound_man_dave/caustics.htm)

Armatures and IK

- Dancing Flor (http://otothecleaner.free.fr/tutorials/Flor/flor.html) * exposed
- Animation Workshop II (http://www.elysiun.com/forum/viewtopic.php?p=511798) * exposed
- Using Ipo driven shape keys to correct deformations in joints (http://kokcito.tk/rvk/ipo.html) * exposed
- Animation recode project (http://www.blender.org/cms/How_Armatures_work.634.0.html)
- Action Constraints tutorial made easy (http://www.elysiun.com/forum/viewtopic.php?t=49603)
- Rigged Character (http://www.elysiun.com/forum/viewtopic.php?t=31839)
- Driven Hand Rig (http://www.elysiun.com/forum/viewtopic.php?t=19347)
- Constrained Mechanics (http://argoslabs.com/%7Emalefico/tutor/mecano-en.html)
- Armatures Tutorial (http://www.anycities.com/turbog/Tutorial1.html)
- Animation using Armatures (http://www.users.bigpond.net.au/blendage/pages/tutorials/tut4/animatin_armatures.html
- Ikas Blender 3D - Introduction (http://mmaigrot.free.fr/didac-blender/ikas/ikas-eng/ik-intro.php)
- Constrained Mechanics (http://www.malefico3d.com.ar/tutor/mecano-en.html)
- Using Armatures (http://www.elysiun.com/tutorials/animation/)
- Making and Using Armatures (http://www.users.bigpond.net.au/blendage/pages/tutorials/tut3/armatures.html)
- Character Animation tools (http://www.elysiun.com/tutorials/animation/)
- IK Solver Constraint (http://www.elysiun.com/forum/viewtopic.php?t=42310)
- Softbody for Rigged Characters (http://www.enricovalenza.com/softb.html)
- Rigging limbs that can twist (http://kokcito.tk/tut/rig3.html)
- Rigging tricks (http://kokcito.tk/tut/rig1.html)
- IKA Tutorial (http://kahuna.clayton.edu/~jbrooks/blender/tutorials/IKA/)

Animation

- Simple Animation (http://feeblemind.tuxfamily.org/dotclear/index.php/2005/03/08/17-didacticiel-animations-
- Basic Keyframing (http://www.users.bigpond.net.au/blendage/pages/beginners_tutorials/tut2/Keyframeanim
- Effects (http://download.blender.org/documentation/html/c1585.html)
- Character Animation (http://dev.newmediaworx.com/johnnyb/bplanet/tut_charanim.htm)
- Run Cycle (http://rodri.aniguild.com/tuto_run/run_en.php) (Non Blender specific)
- Lip Sync I (http://www.meloware.com/blender/lipsync.htm)
- Lip Sync II (http://dev.newmediaworx.com/johnnyb/bplanet/tut_charanim2.htm)
- Camera Switching (http://www.elysiun.com/tutorials.php?id=7) link broken?
- Walking Blues (http://argoslabs.com/~malefico/tutor/walking.html)
- Non Linear Action Editor - NLA I (http://mmaigrot.free.fr/didac-blender/nla/eng/index.php)
- Non Linear Action Editor - NLA II (http://argoslabs.com/~malefico/tutor/nla-en.html)
- Wave Effect (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/animation_effects/dida
- Build Effect (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/animation_effects/dida

- Animation effects (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/animation_effects/inde
- Seascape (http://homepages.nildram.co.uk/~raytrace/tutorials.html) *(pdf tutorial)*
- Flag Creation in Blender (http://homepages.nildram.co.uk/~raytrace/tutorials.html) *(pdf tutorial)*
- Plane Following a Path (http://www.ingiebee.com/Blendermania/plane%20follow%20path.htm)
- Object Following a Path (http://www.ingiebee.com/tutorials/Path/theeth%20paths.htm)
- Cyclic Animation (http://membres.lycos.fr/bobois/Tuts/animation_cyclique/animation_cyclique_01_ang.html
- Blending into Wireframe (http://www.ingiebee.com/Blendermania/Blending%20to%20Wireframe.htm) *(broken pictures links; 7.3. 2005)*
- Animating Materials (http://www.users.bigpond.net.au/blendage/pages/tutorials/tut7/colourchanging.html)
- Walking Tutorial (http://www.fortunecity.com/skyscraper/true/947/id13.htm)
- Tracking Cameras to Paths (http://www.katorlegaz.com/index.php?filename=/articles/blender/Tracking_Cameras_to_
- Relative Vertex Keys (RVKs) (http://www.katorlegaz.com/index.php?filename=/articles/blender/RVKs.php)
- Change Cameras during an Animation (http://www.elysiun.com/forum/viewtopic.php?t=42840&highlight=change+camera)
- Camera Tracking With Additional Roll Constraint (http://www.elysiun.com/forum/viewtopic.php?t=46181)

Particles

- Particle Interaction (http://blender3d.com/cms/Particle_interaction.349.0.html)
- Static Particles (http://www.elysiun.com/forum/viewtopic.php?t=41356)
- Static Particle Effect (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/animation_effects/dida
- Particles Effect (http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/animation_effects/dida
- Making a Fireplace (http://www.geocities.com/blengine/fireplace1.html) (down 06.08.2006)
- Making a Fountain With Particles (http://www.users.bigpond.net.au/blendage/pages/tutorials/tut5/fountain.html)
- Dust Particles (http://www-users.cs.umn.edu/~mein/blender/tutorials/dust/particle.html)
- Flames (http://www.centralsource.com/blender/flames/)
- Liquid (http://www.centralsource.com/blender/liquid/)
- Explosions (http://www.centralsource.com/blender/explosions/)
- Smoke (http://www.centralsource.com/blender/smoke/)
- Making a Rain Effect (http://feeblemind.tuxfamily.org/dotclear/index.php/2004/12/24/4-blender-faire-pleuvoir---
- Particle and field basics (http://www.deviantart.com/view/27104504/)

Fluid Simulation

- Fluid Simulation Basics (http://www.deviantart.com/view/27479896/)
- Fluid Tutorial (http://www.penguinscore.com/fluidsimtut.htm)

Compositing

- Matching Real Lighting (http://www.andrew-whitehurst.net/fx_light.html)
- CGI/Live Action Interaction (http://www.blenderwars.com/tut.php?module=blendcg)
- Compositing CG and Live Action in Blender (http://www.weirdhat.com/blender/compositing2/)

Game Engine

- Game Engine Developing Team (http://wiki.blender.org/bin/view.pl/Blenderwiki/GameEngineTeam) broken link
- New fully integrated game engine? (http://www.blender.org/forum/viewtopic.php?t=6409&sid=868ed3b00fd68157b704e87f
- Game Blender Documentation

(http://web.archive.org/web/20011207173733/http://www.blender.nl/gameBlenderDoc/bo
(Last modified 13. 07. 2001)
- #GameBlender (http://www.antihc3.dyndns.org/gameblender/game.php) (Broken Link)
- Walkthrough Tutorial (http://www.blender3d.org/Education/index_old.php?sub=TutorialWalkthrough)
- Overlay Scenes (http://www.fortunecity.com/skyscraper/true/947/id16.htm)
- Walking Tutorial (http://www.fortunecity.com/skyscraper/true/947/id13.htm)
- Make a Menu (http://www.fortunecity.com/skyscraper/true/947/id17.htm)
- Flipper (http://otothegardener.free.fr/tutorials/Flipper/flipper.htm)
- Armatures in the Game Engine (http://www.users.bigpond.net.au/blendage/pages/tutorials/tut8/realtimearmatures.html)
- Multilevel Maze (http://www.users.bigpond.net.au/blendage/pages/tutorials/tut6/maze.html)
- Creating Boulders that make the map restart when hit (http://www.users.bigpond.net.au/blendage/pages/tutorials/tut10/adding_traps.html)

Python and Plugins

- Python Lessons (http://www.sutabi.tk/timmeh/index.php?option=com_content&task=category&sectionid=
* exposed
- Make Hair (http://www.dedalo-3d.com/index.php?filename=SXCOL/makehair/abstract.html) * exposed
- Povanim Export Script (http://jmsoler.free.fr/util/blenderfile/fr/povanim_en.htm)
- AI Path Importer (http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_import_ai_en.htm)
- 3D-No Plugins; Put your 3D Blender space on web! (http://www.thoro.de/portfolio/verschiedenes/3DNP.html)
- Randomizer Script (http://www.elysiun.com/forum/viewtopic.php?t=37400)
- Embedding the 3D web plugin in a web page (http://www.blender3d.org/Education/index_old.php?sub=TutorialEmbedplugin)
- Different Useful Scripts (http://www.selleri.org/Blender/scripts/text.html)
- AfterGlow, Polyline, Hitchcock ZoomEffect, ... (http://www.hgb-leipzig.de/~daniel/blender/)
- Dynamica (http://www.centralsource.com/blender/dynamica/)
- BlendSaber (http://www.blenderwars.com/tut.php?module=blendersaber)
- Lsystem tree maker (http://marief.soler.free.fr/Monsite/lsystem_en.htm)
- Python Scripting for Procedural Animation (http://www.ingiebee.com/Blendermania/Genos%20Spring.htm)
- Python Scripting Part I (http://www.blenderbuch.de/tutor/python1/python1_eng.html)
- Python Scripting Part II (http://www.blenderbuch.de/tutor/python2/Python2_eng.html)
- Python Scripting Part III (http://www.blenderbuch.de/tutor/python3/Python3_Eng.html)
- Focalblur - A Matter of Depth (http://www.elysiun.com/tutorials.php?id=2)
- Python API Introduction (http://jmsoler.free.fr/didacticiel/blender/tutor/english/python_script00.htm)
- Python API, Making a Square Mesh (http://jmsoler.free.fr/didacticiel/blender/tutor/english/python_script01.htm)
- Python API, Iterations (http://jmsoler.free.fr/didacticiel/blender/tutor/english/python_script02.htm)
- Python API, Automating Vertex Creation (http://jmsoler.free.fr/didacticiel/blender/tutor/english/python_script03.htm)
- Python API, Automating Face Creation (http://jmsoler.free.fr/didacticiel/blender/tutor/english/python_script04.htm)
- Python API, Making Potatoid (http://jmsoler.free.fr/didacticiel/blender/tutor/english/python_script05.htm)
- Python Script, To build an Empty for EnvMap (http://jmsoler.free.fr/didacticiel/blender/tutor/english/eng_scriptmirror.htm)
- Python Script, Bezier Curves Import (http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_curvesimport_en.htm)
- Python Script, Paths import (http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_paths_import_en.htm)
- Python Script, Importing Adobe Illustrator Format (http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_import_ai_en.htm)
- Python Script, Mesh Explosion (http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_meshexplosion_en.htm)
- Python Script, Level Of Detail (http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_LOD_en.htm)
- Python Script, Wire Shadows and extrusions

(http://jmsoler.free.fr/didacticiel/blender/tutor/python_wireshadows_en.htm)
- Python Script, Changing the active camera instantly (http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_changerdecamera_en.htm)
- Subsurface Scattering in Blender (http://www.dedalo-3d.com/index.php?filename=SXCOL/experiments/ss_scattering_pytho
- Using the Superficial Scattering Script (http://feeblemind.tuxfamily.org/dotclear/index.php/2005/04/25/39-mh-tutorial-using-the-s
- City Block Genarator (http://www.elysiun.com/forum/viewtopic.php?t=16217)
- Blender Camera Calibration with Live Camera (http://www.elysiun.com/forum/viewtopic.php?t=32715)
- Import-export different formats, Different Generators, etc... (http://www.redrival.com/scorpius/blender-plugins.htm)
- Horn Extrude (http://www3.sympatico.ca/emilio.aguirre/hornextrude.html)
- Mesh shaker and tutorial (http://saltshaker.sourceforge.net/)
- Batch STL (http://blender.formworks.co.nz/batch_stl.html)
- Vertices to a Curve Converter (http://www.elysiun.com/forum/viewtopic.php?t=45576)
- Zutils, Z-Buffer Utilities (http://www.elysiun.com/forum/viewtopic.php?t=35355)

Using other Programs with Blender

- YafRay (http://www.yafray.org)
- Verse (http://blender.org/modules/verse/)
- SPE - Python IDE for Blender (http://pythonide.stani.be)
- Wings 3D Subdivision Modeler (http://www.wings3d.com/)
- Verse Gimp-Blender Plugin (http://users.pandora.be/blendix/verse/old/demo.html)
- Kerkythea Renderer (http://www.softlab.ece.ntua.gr/~jpanta/Graphics/Kerkythea/)
- Equinox 3D (http://www.equinox3d.com)
- Wings3D, a Quick and Accurate UVmapping Tool for Blender (http://mywebpages.comcast.net/jmandmc/uvmap/uvmap.html)
- Povray export (http://jmsoler.free.fr/util/blenderfile/fr/povanim_en.htm)
- Adobe Illustrator Paths import (http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_import_ai_en.htm)
- Inkscape SVG import (http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_import_svg_en.htm)
- Using Blender content in PowerPoint 2000 (http://www.blender3d.org/Education/index_old.php?sub=TutorialPowerpoint)
- Ter2Blend (http://users.skynet.be/sky33676/ter2blend1.html)
- Batch Processing Images (http://membres.lycos.fr/bobois/Tuts/traitement_par_lot/traitement_par_lot_ang.html)
- BVH Info (http://www.centralsource.com/blender/bvh/)
- Stereogram (http://www.geocities.com/blenderlab/tutorial1.html)
- Importing VRML (http://www.blender3d.org/Education/index_old.php?sub=TutorialVrml)
- Creating Quicktime VRs (http://www.katorlegaz.com/index.php?filename=/articles/blender/Quicktime_VR.php)
- Blender, Python and Mac OS X (http://www.katorlegaz.com/index.php?filename=/articles/blender/Python_and_OS_X.php)
- Stylistic Rendering (http://www.flippyneck.com/temp/NPR.htm)
- Voodoo Camera Tracker (http://www.digilab.uni-hannover.de/docs/manual.html)
- Miniature UV mapped building for Lionhead's The Movies game (http://themovieseditor.com/tutorials/blender-mini-building.html)

Distributed Computing

- BURP - Big and Ugly Rendering Project (http://burp.boinc.dk/)
- Render Planet (https://renderplanet.com/)
- Global Rendering-Farm (http://renderworld.futureware.at/)
- OS X Distributed Blender Network Rendering with Xgrid (http://www.katorlegaz.com/index.php?filename=/articles/blender/Using_Blender_with_X
- Bfarm Distributed Rendering via Internet (http://geocities.com/tronovan3d/)

Maybe someday ...

- LightRay (http://www.tacc.utexas.edu/~cburns/lightray/lightray.php)
- Toxic (http://www.toxicengine.org/)

Video Tutorials

- Official Blender Video Tutorials (http://blender3d.org/cms/Video_Tutorials.396.0.html)
- Greybeard's Blender Video Tutorials (http://www.ibiblio.org/bvidtute/)
- Head Tutorial (http://www.visiontovision.com/BlenderHead1.html) *(Part 1 of 12-part Flash Tutorial)*
- Series of short VTuts that explain basic details (http://www.blendernation.com/2006/03/01/videotutorials-blender-3d-workshop/)

Blender WikiBooks

- Blender 3D: Blending Into Python * exposed
- Blender 3D: HotKeys
- Blender 3D: Import and Render a SolidWorks Model
- Blender 3D: MemoBook
- Blender 3D: Noob to Pro
- All Blender WikiBooks Modules

FAQ

- Blender FAQ (http://www.museum.state.il.us/ismdepts/library/linuxguides/blender/blender_faq_0.html) (Generated on September 24, 2001)
- Blender Tips (http://dev.newmediaworx.com/johnnyb/bplanet/tut_blendertips.htm)

Repository

This section is not about tutorials, but you can find here different useful stuff for your Blender.

Blueprints

- cgworld.ru (http://cgworld.ru/modules.php?name=Blueprints) * exposed
- blueprints.onnovanbraam.com (http://blueprints.onnovanbraam.com/)
- smcars.net (http://smcars.net/)
- boats (http://www.boatdesign.net/boat-plans-archive/index.htm)
- old cars (http://www.mgussin.freeuk.com/00Plans.htm)
- War Planes (http://www.airwar.ru/other/draw_1w.html)
- All sorts of cars (http://www.suurland.com)

Materials

- Sonix' Material Library(esp. Cars) (http://www.free-webspace.biz/sonix/Cars/Blender234CarMaterialLibraryR1.html)

Models

- blendermodels.katorlegaz.com (http://blendermodels.katorlegaz.com/)

Photos

- Human Photo References (http://www.3d.sk)
- Brain scan (http://www.med.harvard.edu/AANLIB/cases/caseNA/pb9.htm)

Textures

- Mayang's Free Textures Hi-Res (http://www.mayang.com/textures/index.htm)
- Jeremey Engleman's Public Textures Hi-Res (http://www.art.net/~jeremy/photo/public_texture_frameset.html)
- Image*After Free Images Hi-Res (http://www.imageafter.com/)
- Free Textures Mid-Res (http://digitalcraftsman.com/textureBin/textureBin.htm)
- Sky Maps (http://www.elysiun.com/forum/viewtopic.php?t=24738)

Miscellaneous

CG artists *must see or read:*

- How to succeed in Animation (http://genedeitch.awn.com/)
- Bitter History of 3D Business (http://home.nordwest.net/Adger/tips/tip077.html)
- Blender Art Gallery (http://centralsource.com/blenderart/index.php)
- Classical Film and Video Knowledge Base (http://www.rondexter.com/)
- Quick Tips in Design&GFX (http://www.atpm.com/9.07/design.shtml)

Open Movies

- www.blenderprojects.com (http://www.blenderprojects.com/)
- Orange (http://orange.blender.org/cms/The_Movie.555.0.html)
- NaNo - Blender Internet Virtual Movie Studio (http://nano.prods.free.fr/)

IRC

- irc://irc.freenode.net/blender
- irc://irc.freenode.net/blenderchat
- irc://irc.freenode.net/blenderclasses
- irc://irc.freenode.net/blenderqa
- irc://irc.freenode.net/blendercoders
- irc://irc.freenode.net/blenderwiki
- irc://irc.freenode.net/gameblenderdev
- irc://irc.freenode.net/gameblender
- irc://irc.freenode.net/verse

Tests

- Blender Benchmarks (http://www.eofw.org/index.php)

Other Lists

Please read more about on talk page.

Tutorials

- blenderartists.org (http://blenderartists.org/cms/index.php?id=38) * exposed
- A Monolingual List of Tutorials (http://www.zoo-logique.org/3D.Blender/index.php3?zoo=dif)
- A Multilingual List of Tutorials (http://membres.lycos.fr/bobois/Liens_Links/big_list.html)

Miscellaneous

- Blender 3D Links & Resources (http://www.katorlegaz.com/index.php?filename=links/blender_3D.php)
- Blender Heads around the Globe(Where live other Blenders?) (http://www.frappr.com/blenderheadsaroundtheglobe)
- Blenderart Magazine (http://s12.yousendit.com/d.aspx?id=14ZVYHLTFHOAK2V3YTCWU38RQ3) Mirror 1 (http://www.apollux-designs.com/BA-Magazine/blenderart_mag-nov-05.pdf.zip) ,M2 (http://blendertestbuilds.de/index.php?dir=Blenderart/20051115/) ,M3 (http://www.intellidesign.org/blenderart_mag/)
- Blender Classroom Tutorial (http://www.statikonline.com/Blender/)

About

This links list is language filtered and extended version of personal collection originally provided by **IamInnocent**. So if you looking for tutorials in other languages check this link at www.elYsiun.com (http://www.elysiun.com/forum/viewtopic.php?t=13380) .

- German Blenders should have a look at http://de.wikibooks.org/wiki/Blender_3D:_Tutorial_Linkliste

If you want to add Blender tutorial link in some other language you can add it temporary on

talk page. We will later made such WikiBooks in other languages too.

Thank you all who contributed to this nice and useful links collection! Feel free to add your name or link if you think you need to be mentioned here.

-- Popski, Mar 2005 – tutorials from all over the Web

Ways to create "fluffy" effect (materials and lights)

The ways to create "fluffy" (brightened) edges.

based on this Elysiun thread (http://www.elysiun.com/forum/viewtopic.php?t=42662&postdays=0&postorder=asc&start=(

- Spherical Blend texture method

Add a Blend texture set to "sphere" (circular pattern in preview). map it to empty|empty|Z coordinates and coordinates source to "Nor" instead of "Orco" , set method to Add and channel to Emit.

- Backlight (aka "@ndy's top secret material" :))

Just add a sufficiently, but not overly bright, colored Hemi light behind the object (relative to camera). You may need set it to be "layer only" light if it interferes with rest of a lightring rig. Like ramp, this method can make edge not just brighter but of different color.

- Color ramp (http://www.blender3d.org/cms/Ramp_Shaders.348.0.html) with input set to normal.

Pretty straighforward, but many advise against it.

- Minnaert shader

Available in 2.37, "Darkness"<1 actually brightens edge. A cool shader, but not very useful for this purpose.

Compiled by : Trident (under construction)

Troubleshooting

ATI Radeon Slowdown Problems

Go the the ATI site driver downloads section and select the appropriate OS/graphic card. This will (most likely) link you to the d/l for the current Catalyst 4.4 drivers.

At the bottom of that page is a link to "Previous driver versions".

D/L the Catalyst 3.7 driver package. Run the EXE and it will extract the driver install package to:

C:\ATI\SUPPORT\

In this directory will be a directory named (for XP/Win2K users: \wxp-w2k-7-93-030812a1-010735c-efg\

This is where the driver install package is located. DO NOT RUN THE SETUP. You don't need to install the old driver.

Navigate down thru the dir structure to the following dir: C:\ATI\SUPPORT\wxp-w2k-7-93-030812a1-010735c-efg\2KXP_INF\B_10679

In this dir you'll find a file called: atioglxx.dl_ This is a "packed" version of the ATI OGL driver.

The next step specifies how you can extract these files to your hard disk using the 'expand.exe' utility included on the cd.

The 'I386' folder on the Windows XP CD contains a utility 'expand.exe' that can be used to uncompress all compressed dll files. It is a commandline utility, so you have will have to run it from either the command prompt or the Run dialog. Some examples of its usage are:

```
expand X:\I386\ADROT.DL_ C:\ADROT.DLL
The above command decompresses the compressed DLL
```

ADROT.DL_ on the WinXP CD, copies it to C:, and changes the extension to .DLL .

Just extract that file (atioglxx.dll) to your Blender install directory. Usually C:\Program Files\Blender Foundation\Blender\

Launch Blender.... no more slowdown.

For people who the above solution leads to Blender crash at start up

Thanks to Xenobius at Blenderartists Forum

This solution is tested on ATI radeon express 200 (x200) integrated video card. However it should work on other ATI card.

Download the Nvidia 53.03 Driver HERE (http://www.megagames.com/news/redir.cgi?http://download.nvidia.com/Windows/53.03/53.0

Extract the driver to C:\NVIDIA

Copy the nvoglnt.dll from the C:\NVIDIA folder to your blender program folder (ie. C:\Programs Files\Blender 2.40\)

Rename nvoglnt.dll (the one in the blender program folder) to atioglxx.dll

Run blender...Viola! Blender runs like it should! assorted technical hang-ups and what to do about them

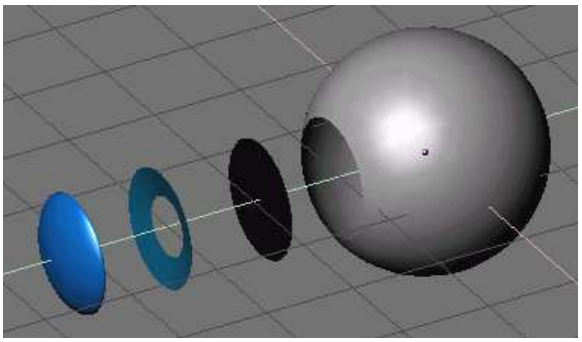
Creating Pixar-looking eyes in Blender

Next page: UV Mapping
Previous page: The Rusty Ball

Note: This tutorial uses the same modelling and texturing technique described in the well-known MAX tutorial by Adam Baroody (http://www.3dluvr.com/rogueldr/tutorials/eye/eyes.html). The sole purpose of my tutorial is to make this technique more popular among the Blender users by explaining how to achieve the same result with Blender.

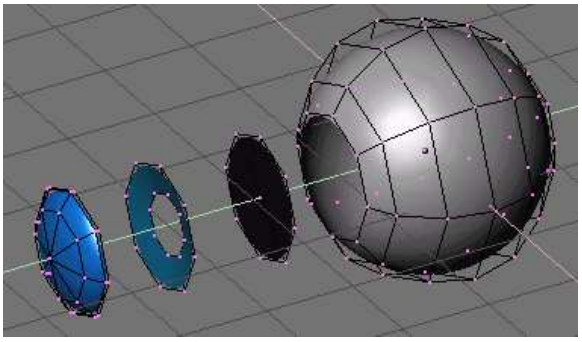
So here we go!

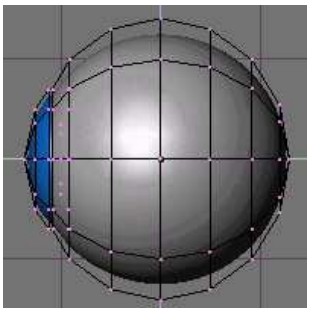
The goal of this tutorial is to make a Pixar-looking eye. One of the main reasons that Pixar's characters really convey life is in their eyes. They have depth, you can see how the eye not only shines but it "collects" light. You may think that you can't achieve this effect without raytracing but you're wrong. The secret of this depth is in the modelling of the eye. Lets see how it works!

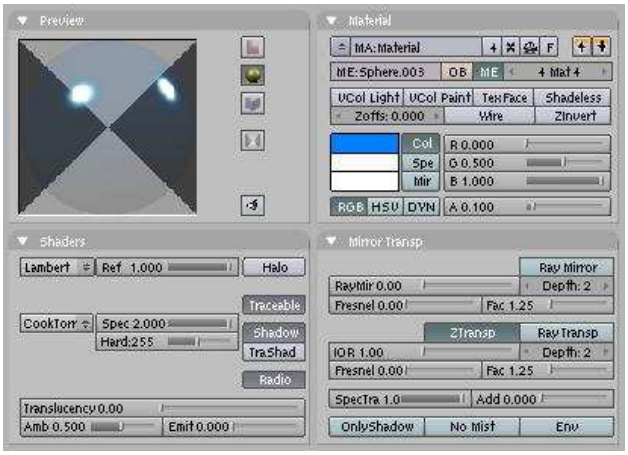


In this picture you can see the "ingredients" of the eye model. The blue mesh at the left is the cornea. It's shape allows for a small spot of specular light to appear on it even if the light is in a far side position. The mesh next to it is the iris. Now notice how it's a bit concave. That's the tricky part - the shape of the iris allows for a wide soft specular light to appear at the opposite side of the lamp direction. This fakes refracted light from the cornea and makes the illusion of "collecting" light and creates depth. The next mesh is the eye pupil - a simple circle. You can position it close to the inner side of the iris. And finally - the eyeball. It's a simple sphere with a hole in it.

I won't go deep into modelling of each element - it uses Blender's subdivision surfaces and it's quite simple as you can see.







Now let's look at the materials. The cornea uses a transparent material (**alpha = 0.1**) with **Spec = 2**, **Hard = 255** and **SpecTra = 1**. "**Ztransp**" should also be turned on.

[NOTE: I set my cornea up exactly how it is in the picture (and it looks transparent in the preview) but when I render the eye, the cornea/iris area just looks black]

[Timbot says: For dude above and whoever else reads this, just disable the "Traceable" button in the "Links and Pipeline" box, that did it for me.]

<---- The iris uses this image texture.

[NOTE: I can't seem to get the iris to show correctly. I need to know how to make the iris.] [Another Note: I can't get it working either; I need to know how to get the texture of the iris in the exact middle.] [3rd NOTE: If you go to the Map Input tab, you can offset the image anyway you want]-Badalia [Note: I can't get the pic on my iris, it is shown in the preview window, but nothing I did up to now could make the pic appear on my mesh. I don't know if anyone can help me, but i'd be very grateful for any idea what I might do wrong.(Khayne)]

[Ok try this: go to map input. and change the settings to

Badalia

[No, still doesn't work, but thanks for trying]

The iris is built from a circle of 8 vertices. Select opposite vertices of the circle and join them using FKEY. Subdivide each new edge using subdivide multi with 2 cuts (WKEY then subdivide multi. Check number of cuts = 2. Press OK).
Select the new vertices two at a time (adjacent ones) and join them using FKEY. This will form a ring which will become the pupil hole. Change to edge select (CTRL + TAB) and select the inner edges where the pupil will be. Delete these edges (XKEY then press Edges). Change back to vertex selection and select the inner vertices around the pupil hole. Change to a side view and pull the vertices back a little to formthe concave shape. Scale the iris to the same size as the hole in the eyeball and position it behind the cornea. Create a new material and add the iris texture (Choose Image as the texture type and then press Load Image button).
The texture should be flat texturing (top button to right of preview) and flat in the Map Input tab. Occo coordinates work. All that is required now is a bit of tweeking of the texture size using the Xsize and Ysize values in the Map Input tab, and scaling the pupil hole size in Edit mode.

[NOTE: I've found another way to create the iris: I simply added another eyeball (UVsphere with 8 segments and 9 rings). First I deleted the outmost vertex at the backend of it. Then I deleted the rest except the two rings I needed for the iris. Finally I selected the smaller ring and scaled it hitting s to give the pupil hole the right size. By the way I had created the cornea in a similar way. I had taken the frontpart of a UV for that.]

You can tweak the RGB values and brightness/contrast of the image to achieve the appearance you want. Use a smaller value for Hard (about 50) otherwise you'll have a too shiny look instead of soft specular that fakes refracted light. The Spec value depends on the energy and distance of the light that illuminates it. Generally you'll need to take care that the refracted ligth on the iris should be no more than half as bright as the small specular spot on the cornea - otherwise you'll achieve the bad effect of two specular spots. Oh, another important thing - join the four meshes before tweaking the texture coordinates. Otherwise you'll have to do the job twice after you join them, because the texture space is changed. And activate shadeless button.

The material for the pupil is a simple black color with the "shadeless" button on.

The eyelid has white color and high values for Spec and Hard (depends on the lighting). Optionally you can use a reflection map to make it look more wet but I usually don't do this.

The lighting is simple - move the eye to a new layer, create a new lamp and make the lamp affect only this layer. Position the lamp at a good angle so you have a small shiny spot of specular light on one side of the iris and a soft spot of "refracted" light on the other side. You can use a backlight to prevent the eyeball from being too dark at the non-illuminated part.

That's it! Now you're (almost) ready to start with character animation. You have a nice eye, now you only need a character for it!

Next page: UV Mapping
Previous page: The Rusty Ball