

# Blender 3D: Noob to Pro

# Contents

## Articles

<b>Intro</b>	<b>1</b>
Blender 3D: Noob to Pro	1
Blender 3D: Noob to Pro/Beginner Tutorials	10
<b>Unit One: Knowing Before Making</b>	<b>11</b>
Blender 3D: Noob to Pro/Unit 1: Knowing Before Making	11
Blender 3D: Noob to Pro/Avoiding Headaches Early On: Good 3D Modeling Practices	12
Blender 3D: Noob to Pro/Know What You're Doing	12
Blender 3D: Noob to Pro/Cheat the 3D	13
Blender 3D: Noob to Pro/Performance vs. Quality	14
Blender 3D: Noob to Pro/Introduction	15
Blender 3D: Noob to Pro/Understanding coordinates	15
Blender 3D: Noob to Pro/Understanding Coordinates	16
Blender 3D: Noob to Pro/Orthographic Projections	17
Blender 3D: Noob to Pro/Isometric Projections	19
Blender 3D: Noob to Pro/Perspective Views	20
Blender 3D: Noob to Pro/Coordinate Spaces in Blender	21
Blender 3D: Noob to Pro/Tutorial Syntax	24
Blender 3D: Noob to Pro/Non-standard equipment	26
Blender 3D: Noob to Pro/Operating System specific notes	27
Blender 3D: Noob to Pro/Blender Interface	29
Blender 3D: Noob to Pro/Blender Windowing System	30
Blender 3D: Noob to Pro/The User Preferences Window	37
Blender 3D: Noob to Pro/The Buttons Window	43
Blender 3D: Noob to Pro/The 3D Viewport Window	47
Blender 3D: Noob to Pro/Other Windows	53
<b>Unit Two: Creation and Development</b>	<b>54</b>
Blender 3D: Noob to Pro/Learn to Model	54
Blender 3D: Noob to Pro/Introducing Meshes	55
Blender 3D: Noob to Pro/Beginning Tips	59
Blender 3D: Noob to Pro/Quickie Model	63
Blender 3D: Noob to Pro/Quickie Render	65

Blender 3D: Noob to Pro/Modeling a Simple Person	67
Blender 3D: Noob to Pro/Detailing Your Simple Person 1	73
Blender 3D: Noob to Pro/Detailing Your Simple Person 2	76
Blender 3D: Noob to Pro/Creating a Simple Hat	83
Blender 3D: Noob to Pro/Putting Hat on Person	87
Blender 3D: Noob to Pro/Bones	89
Blender 3D: Noob to Pro/Mountains Out Of Molehills	95
Blender 3D: Noob to Pro/Modeling a Gingerbread Man	102
Blender 3D: Noob to Pro/Penguins from spheres	115
Blender 3D: Noob to Pro/Die Another Way	129
Blender 3D: Noob to Pro/Die Easy 2	140
Blender 3D: Noob to Pro/Model a Silver Goblet	154
Blender 3D: Noob to Pro/Model a Silver Goblet cylinder	160
Blender 3D: Noob to Pro/Simple Vehicle	162
Blender 3D: Noob to Pro/Simple Vehicle: Wheel	163
Blender 3D: Noob to Pro/Simple Vehicle: Seat	168
Blender 3D: Noob to Pro/Simple Vehicle: Rocket Launcher	171
Blender 3D: Noob to Pro/Simple Vehicle: Body	175
Blender 3D: Noob to Pro/Simple Vehicle: Some Assembly Required	184
Blender 3D: Noob to Pro/Modeling a 3D Parachute in Blender	188
Blender 3D: Noob to Pro/Model a Low Poly Head	193
Blender 3D: Noob to Pro/Building a House	211
Blender 3D: Noob to Pro/Pipe joints	221
Blender 3D: Noob to Pro/Creating Models With Photo Assistance	224
Blender 3D: Noob to Pro/2D Image (logo) to a 3D Model	240
Blender 3D: Noob to Pro/2D Image (logo) to a 3D Model Part 2	254
Blender 3D: Noob to Pro/Curve and Path Modeling	259
Blender 3D: Noob to Pro/Intro to Bezier Curves	259
Blender 3D: Noob to Pro/Curves In 3d	266
Blender 3D: Noob to Pro/Deforming Meshes using the Curve Modifier	268
Blender 3D: Noob to Pro/Making Your Creation Smoother	271
Blender 3D: Noob to Pro/Materials and Textures	272
Blender 3D: Noob to Pro/Quickie Material	273
Blender 3D: Noob to Pro/Quickie Texture	275
Blender 3D: Noob to Pro/Procedural Textures	277
Blender 3D: Noob to Pro/Halo Materials	277
Blender 3D: Noob to Pro/Image Textures	281
Blender 3D: Noob to Pro/Creating Basic Seawater	281

Blender 3D: Noob to Pro/Mountains Out Of Molehills 2	283
Blender 3D: Noob to Pro/Basic Carpet Texture	285
Blender 3D: Noob to Pro/The Rusty Ball	289
Blender 3D: Noob to Pro/Procedural Eyeball	290
Blender 3D: Noob to Pro/Creating Pixar-looking eyes in Blender	296
Blender 3D: Noob to Pro/UV Map Basics	301
Blender 3D: Noob to Pro/Realistic Eyes In Blender	309
Blender 3D: Noob to Pro/Light a Silver Goblet	319
Blender 3D: Noob to Pro/Every Material Known to Man	324
Blender 3D: Noob to Pro/Beginning Lighting	327
Blender 3D: Noob to Pro/Faked Gi with Blender internal	327
Blender 3D: Noob to Pro/Render Settings	330
Blender 3D: Noob to Pro/Output Format Options	334
Blender 3D: Noob to Pro/OSA	336
Blender 3D: Noob to Pro/Looking All Around - Panorama Settings	337
Blender 3D: Noob to Pro/Other Important Render Options	340
Blender 3D: Noob to Pro/Adding Lamps	341
Blender 3D: Noob to Pro/Yafra Render Options	349
Blender 3D: Noob to Pro/Stamping	350
Blender 3D: Noob to Pro/Basic Animation	353
Blender 3D: Noob to Pro/Basic Animation/ScreenLayout	354
Blender 3D: Noob to Pro/Basic Animation/Lattice	356
Blender 3D: Noob to Pro/Basic Animation/Bounce	364
Blender 3D: Noob to Pro/Creating Basic Water animation	369
Blender 3D: Noob to Pro/Flying Through A Canyon	371
Blender 3D: Noob to Pro/Using the sequencer to compile frames into an animation	376
Blender 3D: Noob to Pro/Particle Systems	379
Blender 3D: Noob to Pro/Making Fire	381
Blender 3D: Noob to Pro/Furry	384
Blender 3D: Noob to Pro/Fireworks	391
Blender 3D: Noob to Pro/Soft Body Animation	394
Blender 3D: Noob to Pro/Simple Cloth Animation	397
Blender 3D: Noob to Pro/Soft Body with wind	399
Blender 3D: Noob to Pro/Your First Test	402
Blender 3D: Noob to Pro/Platformer: Creation and Controls	403
Blender 3D: Noob to Pro/An aMAZEing game engine tutorial	405
Blender 3D: Noob to Pro/Platformer: Physics Fixes	414
Blender 3D: Noob to Pro/Making exe	415

Blender 3D: Noob to Pro/Build a skybox	418
Blender 3D: Noob to Pro/Basic mouse pointer	426
Blender 3D: Noob to Pro/Text in BGE	427
Blender 3D: Noob to Pro/Python Platformer: Creation	428
Blender 3D: Noob to Pro/Beginning Modeling Final Project	430
<b>Unit Three: Python In Blender</b>	<b>432</b>
Blender 3D: Noob to Pro/Advanced Tutorials	432
Blender 3D: Noob to Pro/Python Scripting	434
Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Introduction	434
Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Export scripts	437
Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Import scripts	444
Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Procedural object creation	446
Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Scripts for modifying meshes	449
Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Creating a GUI for your script	449
<b>Unit Four: Advanced Modeling</b>	<b>452</b>
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/index	452
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/Modeling	452
Blender 3D: Noob to Pro/HDRi	453
Blender 3D: Noob to Pro/Creating a Light Probe	456
Blender 3D: Noob to Pro/Making Landscapes with heightmaps	457
Blender 3D: Noob to Pro/How to Do Procedural Landscape Modeling	461
Blender 3D: Noob to Pro/Landscape Modeling I: Basic Terrain	464
Blender 3D: Noob to Pro/Landscape Modeling II: Texture Stenciling	471
Blender 3D: Noob to Pro/Landscape Modeling III: Exporting as a Heightmap	477
Blender 3D: Noob to Pro/Realistic Water using Fluid Sim and Yafray	482
Blender 3D: Noob to Pro/BBB's Particles -- New 2.46 Particle System	486
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/Textures	487
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/Bump Mapping	487
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/Normal Mapping	490
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/Texture Normal Mapping	491
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/Color Map Normal Mapping	494
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/Light Mapping	497
<b>Unit Five: Advanced Animation</b>	<b>498</b>
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/index	498
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Introduction	499

Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/index	501
Blender 3D: Noob to Pro/ Advanced Tutorials/ Advanced Animation/ Guided tour/ Armature/index	501
Blender 3D: Noob to Pro/ Advanced Tutorials/ Advanced Animation/ Guided tour/ Armature/object	502
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Armature/edit	506
Blender 3D: Noob to Pro/ Advanced Tutorials/ Advanced Animation/ Guided tour/ Armature/pose	512
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Mesh/index	514
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Mesh/Amodif	514
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Mesh/env	516
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Mesh/vg	518
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Mesh/Shape	523
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Mesh/Shape/Sync	526
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/index	533
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/cl	535
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/cr	536
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/tt	537
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/fl	538
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/fp	539
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/st	539
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/ik	540
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/timeline	544
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/ipo/index	544
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/ipo/type	545
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/NLA/index	545
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/NLA/intro	548
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/NLA/stride	552
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/bob/index	552
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/bob/Build rig	553
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/ Piston, Rod and Crank	554
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/ Cutting Through Steel	557

<b>Unit Six: Advanced Game Engine</b>	<b>566</b>
Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Game Engine/index	566
Blender 3D: Noob to Pro/ Advanced Tutorials/ Advanced Game Engine/ Game Creating Techniques(GUI)	567

Blender 3D: Noob to Pro/ Advanced Tutorials/ Advanced Game Engine/ Game Creating Techniques(GUI)/Creating Pop-Up Menus	<b>567</b>
Blender 3D: Noob to Pro/ Advanced Tutorials/ Advanced Game Engine/ Game Creating Techniques(GUI)/Creating Moving Menus	<b>572</b>
Blender 3D: Noob to Pro/ Advanced Tutorials/ Advanced Game Engine/ Game Creating Techniques(GUI)/The "5-Layer" Button	<b>575</b>
Blender 3D: Noob to Pro/ Advanced Tutorials/ Advanced Game Engine/ Game Creating Techniques(GUI)/Creating Object Outlines	<b>575</b>
Blender 3D: Noob to Pro/ Advanced Tutorials/ Advanced Game Engine/ Game Creating Techniques(Python)	<b>576</b>

## **Unit Seven: Miscellanea** **577**

Blender 3D: Noob to Pro/modelling tips	<b>577</b>
Blender 3D: Noob to Pro/Creating Weapons based on 2D Images	<b>578</b>
Blender 3D: Noob to Pro/Match Moving	<b>580</b>
Blender 3D: Noob to Pro/Motion Tracking with Icarus	<b>581</b>
Blender 3D: Noob to Pro/Create a Clayman	<b>583</b>
Blender 3D: Noob to Pro/Organic Modeling	<b>586</b>
Blender 3D: Noob to Pro/Free Blender Samples	<b>588</b>
Blender 3D: Noob to Pro/Understanding the Fluid Simulator	<b>588</b>
Blender 3D: Noob to Pro/Creating a jewel in Blender	<b>591</b>
Blender 3D: Noob to Pro/Modeling Realistically	<b>596</b>
Blender 3D: Noob to Pro/Modeling a picture	<b>599</b>
Blender 3D: Noob to Pro/Creating Ogg-Theora movies using Blender	<b>600</b>
Blender 3D: Noob to Pro/Creating animated GIFs using Blender and Gimp	<b>601</b>
Blender 3D: Noob to Pro/3D Tiling Backgrounds For The Web	<b>602</b>
Blender 3D: Noob to Pro/Cool Things	<b>609</b>
Blender 3D: Troubleshooting	<b>621</b>
Blender 3D: Noob to Pro/Creating Blender Libraries	<b>622</b>

## **Appendix: References** **625**

Blender 3D: Noob to Pro/Blender Glossary	<b>625</b>
Blender 3D: Noob to Pro/Object Mode HotKeys Review	<b>626</b>
Blender 3D: Noob to Pro/Edit Mode HotKeys Review	<b>629</b>
Blender 3D: Noob to Pro/Hot Keys	<b>632</b>
Blender 3D: Noob to Pro/FAQ	<b>646</b>
Blender 3D: Noob to Pro/Blueprint Links List	<b>647</b>
Blender 3D/Tutorial Links List	<b>647</b>

## **References**

Article Sources and Contributors	667
Image Sources, Licenses and Contributors	674

## Article Licenses

License	692
---------	-----

# Intro

## Blender 3D: Noob to Pro

### About this book

This book is brought to you by many authors and contributors as well as anonymous editors.

### Introduction

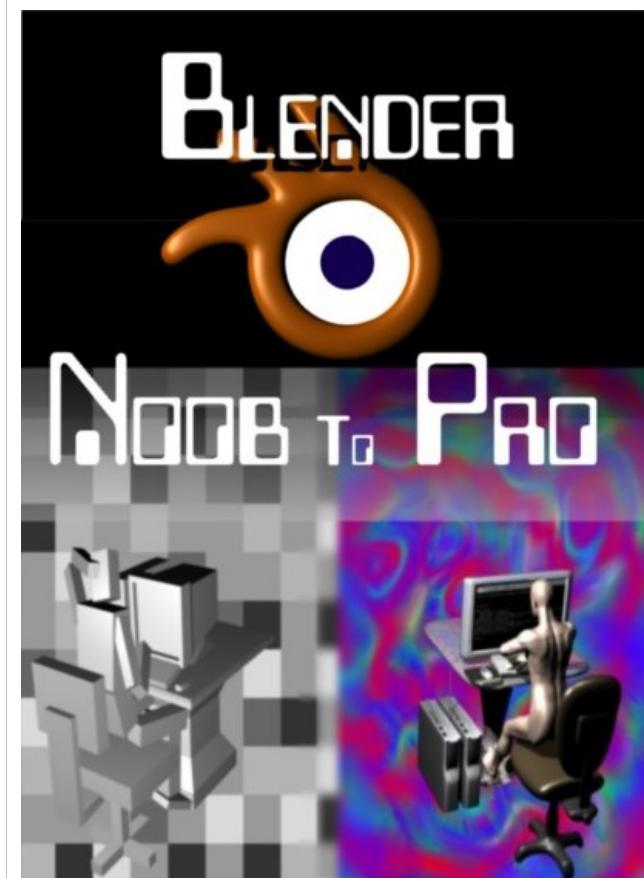
This book is a series of tutorials to help new users learn Blender [1]. The tutorials increase in difficulty, and later tutorials build on the lessons in previous ones. Therefore, **Blender beginners should follow the tutorials in sequence**. Intermediate users can skip to a tutorial of suitable difficulty. We have made efforts to accommodate everyone.

The latest version of Blender can be downloaded from here [2].

This book is one of the Blender-related Wikibooks. Have a look at the complementary ones. For a wiki with more technical information on Blender, please see The Blender Wiki [3].

### Contributing to this book

If you would like to contribute to this book, you are strongly encouraged (read as *begged*) to do so! There are a couple of things you may wish to do.



Front Cover

### Editing a page

Simply click on the **edit this page** link at the top of whichever page you would like to edit! Your contribution will be reviewed before it is officially published. You can also ask questions and make comments about each page by clicking the **discussion** tab at the top of the page.

### Adding a page

If you would like to add a page to the book, such as a new tutorial page or a reference page, create a link to it under the Contents section on this very page. Once the link is created, click it to be taken to the new page where you will be able to add content. If you're creating a tutorial and you don't know where it belongs, put it in the Miscellaneous section for now and we will put it in the right spot.

To maintain ease of navigation throughout the book, please be sure to create links to the pages before and after your new page, using the template shown here. Also be sure to update the page before and after your new page with links to your new page. You can skip this step if you think the page you are putting in will likely be moved from where you put it, but in this case you should also ask for an opinion via the **discussion** tab.

## Adding Images

In Blender tutorials, images are vital. They clarify your point, provide a point of reference, and improve the teaching process. However, **if the image's copyright is not attributed correctly, we must remove it.** This can make a once-great tutorial useless. Even if you believe that you know what copyright to use, please check the **proper Blender images copyright** page and see how you should add copyright and where. Please make sure that every image you use is not copyrighted or that you have permission to use it.

### Guidelines for including images

- Check Wikimedia Commons to see if the image you want to use already exists, as uploading the same image twice is wasteful.
- Make sure to check our image portfolio if you're about to upload a general image (such as icons or buttons) or if you want to use someone else's image as an example. Remember to attribute, if necessary. If you upload an image to the Commons and use it in this Wikibook, please create an entry in the portfolio log. This will help us check its copyright and we will file your image away for safe-keeping and later use.

### What you shouldn't add to pages

- Comments or questions regarding the text. (That's what the discussion pages are for!)
- External links that are not directly related to the article.

### Become a member of the team!

If you want to join and work with the team, read our team page for information and advice.

## Index

### Blender Help

- Blender FAQ
- Hotkey Reference
- /Output Formats/
- /Image Portfolio/
- Blender Wiki PDF Manual (1,426 pages) <sup>[4]</sup>

## Glossaries

- Glossary of Blender Terms
- Materials Directory: Every Material Known To Man

## Unit 1: Knowing Blender

### /Know What You're Doing/

- /Cheat the 3D/
- /Performance vs. Quality/

### /Thinking in 3D/

- /Orthographic Projections/
- /Isometric Projections/
- /Perspective Views/
- /Coordinate Spaces in Blender/

## Learning the Graphical User Interface

- /Tutorial Syntax/
- /Non-standard equipment/
- /Operating System specific notes/
- Become Familiar with the Blender Interface
  - Learn the Blender Windowing System
  - /The User Preferences Window/
  - /The Buttons Window/
  - /The 3D Viewport Window/
  - /Other Windows/
- /Navigation in 3D/

## Unit 2: Creation and Development

### /Learn to Model/

- /Introducing Meshes/
- /Beginning Tips/
- /Quickie Model/
- /Quickie Render/

### /Modeling a Simple Person/

- Detailing Your Simple Person I
- Detailing Your Simple Person II
- /Creating a Simple Hat/
- /Putting Hat on Person/
- Using Bones

**Beyond Basics**

- /Mountains Out Of Molehills/
- /Modeling a volcano/
- /Modeling a Gingerbread Man/
- Penguins from Spheres
- Die Another Way (dice modeling)
  - Die Easy revised for 2.44
- /Model a Silver Goblet/
  - Model a Silver Goblet Another Way
    - /Light a Silver Goblet/
- /Simple Vehicle/
  - /Simple Vehicle: Wheel/
  - /Simple Vehicle: Seat/
  - /Simple Vehicle: Rocket Launcher/
  - /Simple Vehicle: Body/
  - /Simple Vehicle: Some Assembly Required/
- /Modeling a 3D Parachute in Blender/
- /Model a Low Poly Head/
- /Building a House/
- /Pipe joints/

**/Creating Models With Photo Assistance/**

- /2D Image (logo) to a 3D Model/
  - /2D Image (logo) to a 3D Model Part 2/

**/Curve and Path Modeling/**

- /Intro to Bezier Curves/
- Curves in 3d
- /Deforming Meshes using the Curve Modifier/
- MetaObjects

**/Materials and Textures/**

- /Quickie Material/
- /Quickie Texture/
- /Halo Materials/
- /Procedural Textures/
- /Image Textures/

## Using Materials and Textures

- Using Textures
- /Using a texture to make a material partially transparent/
- /Creating Basic Seawater/
- /Mountains Out Of Molehills 2/
- /Basic Carpet Texture/
- /The Rusty Ball/
- /Procedural Eyeball/
- Creating Pixar-looking eyes

## UV Maps

- /UV Map Basics/
- /Realistic Eyes In Blender/

## Unit 3: Broadening Horizons

### Lighting

- /Beginning Lighting/
- /Understanding Real Lights/
- /Adding Lamps/
- Faked Global Illumination with Blender internal

### Rendering

- /Output Format Options/
  - OSA
- /Looking All Around - Panorama Settings/
- /Other Important Render Options/
- /Yafray Render Options/
- /Stamping/
- /2D, Toon, & CAD Rendering/

### /Basic Animation/

- Screen Layout
- Lattice Modifier
- Bouncing Ball with Lattice
- /Creating Basic Water animation/
- /Flying Through A Canyon/
- /Using the sequencer to compile frames into an animation/
  - External tutorial about regular 2D video editing, with lots of illustration. (keywords: sequencer, NLE) [5]

## /Particle Systems/

- Fire
- Fur
- /Fireworks/
- /Particles forming Shapes/
- /Billboard Animation/

## Soft Bodies

- /Soft Body Animation/
  - /Simple Cloth Animation/
- /Soft Body with wind/

## Blender Game Engine Basics

- Platformer: Controls and Movement
- Maze: Force and Multiple Levels
- Platformer: Improving the Physics
- How to make an executable
- /Build a skybox/
- /Basic mouse pointer/
- /Text in BGE/
- Platformer: Creating the Engine with Python

## Unit 4: Taking Off with Advanced Tutorials

### Python Scripting

- Introduction
- The Language
- Export scripts
- Import scripts
- Procedural object creation
- Scripts for modifying meshes
- Creating a GUI for your script

### Advanced Modeling

- High Dynamic Range imaging (HDRi)
  - /Creating a Light Probe/
- Landscape Modeling with Heightmaps
  - /Landscape Modeling I: Basic Terrain/
  - /Landscape Modeling II: Texture Stenciling/
  - /Landscape Modeling III: Exporting as a Heightmap/
- /Realistic Water using Fluid Sim and Yafray/

## Advanced Materials and Textures

- Bump Mapping
- Normal Mapping
  - Texture Normals
  - Color Map Normals
- Nodes: An Introduction

## Advanced Animation

- Introduction
- Guided tour:
  - Armature Object
    - Armature Object in Object Mode
    - Armature Object in Edit Mode
    - Armature Object in Pose mode
  - Mesh Object
    - Connection between Armature and Mesh
    - Envelope
    - Vertex Groups & Weight Paint
    - Shape Keys
      - Lip-Sync with Shape Keys
  - Constraints
    - Copy Location
    - Copy Rotation
    - Track-To
    - Floor
    - Locked Track
    - Follow Path
    - Stretch-To
    - IK Solver
  - Timeline Window
  - IPO Window
    - Data Type
  - NLA Window
    - Introduction To NLA Editor
    - The Stride feature
- Working example: Bob
- Working example: Piston, Rod and Crank
- Working example: Cutting Through Steel

## Advanced Game Engine

- Advanced Game Engine Techniques (GUI)
  - Creating Pop-Up Menus
  - Creating Dropping Menus
  - The "5-Layer" Button
  - Creating Object Outlines
- Advanced Game Engine Techniques (Python)

## Miscellaneous Tutorials

*This is our attic, mostly tutorials that could be useful to some extent if they would be revamped completely, but are of little use at the moment. If you can contribute to some of them, go ahead and rewrite them to your liking!*

- /Modeling a simple space-ship/
- /Modelling tips/
- /Modeling and Texturing a Bottle/
- /Creating Weapons based on 2D Images/
- /Match Moving/
- Match Moving/Motion Tracking with Icarus and Blender
- /Create a Clayman/
- /Organic Modeling/
- /Free Blender Samples/
- /Understanding the Fluid Simulator/
- /Creating a jewel in Blender/
- General Advice on Modeling Realistically



- /Modeling a picture/
- /Modeling with the Spin Tool/
  - /Spin Tool Introduction/

- /Illustrative Example: Model a Wine Glass/
- /Illustrative Example: Model a Mug/
- /Oops: Spin Tool Problems and Solutions/
- /Creating Ogg-Theora movies using Blender/ *Should this page be moved here?*
- /Creating animated GIFs using Blender and Gimp/ *Should this page be moved here?*
- /3D Tiling Backgrounds For The Web/
- Cool Things That Aren't That Obvious in Blender *Should this page be moved here?*
- /Troubleshooting/ Common Technical Issues and What to do About Them *Should this page be moved here?*
- /Creating Blender Libraries/
- All Blender Tutorial Links – tutorials from all over the Web
- /Add some depth with stereo/
- Ways to create a "fluffy" effect (materials and lights)
- /Human Body/
- /Rendering Informations/
- /Using Blender Libraries/
- /Beginning Modeling Final Project/
- /Using Inkscape to make advanced Bezier curves/
- Modeling Textures
- Light Mapping
- /Modeling techniques and Workflow/ (to be constructed)
  - /Polygonal Modeling/
  - /Box Modeling/
    - /Illustrative example: Model a Chair (Swan Chair)/
      - /Model a Chair-Preparations/
      - /Model a Chair-The Seat/
      - /Model a Chair-The Feet/
    - /Illustrative Example: Modeling a Simple Human Character/
      - /Modeling a Human Character - Preparations/
      - /Modeling a Human Character - Modeling/
    - /Illustrative Example: Model a Car (Box)/
    - /Illustrative Example: Model a Dragon/
  - /Polygon by Polygon modeling/
    - /Illustrative Example: Model a Woman Face/
    - /Illustrative Example: Model a Car (Polygon-By-Polygon)/
  - /Blocking with Primitives/
- /Animation Notes and FAQ/

## Resource Directories

- All Blueprints Links – blueprints from all over the Web

## References

- [1] <http://www.blender.org>
- [2] <http://www.blender.org/download/get-blender/>
- [3] [http://wiki.blender.org/index.php/Main\\_Page](http://wiki.blender.org/index.php/Main_Page)
- [4] <http://pdf.letworyinteractive.com/>
- [5] <http://www.straightedgelinux.com/videoBlender>

# Blender 3D: Noob to Pro/BEGINNER TUTORIALS

---

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.

If you ever get stuck for some reason in a tutorial, there are a number of places you can turn for help. The best way to get help is with an Internet Relay Chat (IRC) client such as X-Chat<sup>[1]</sup>. Connect to **irc.freenode.net** and talk to blender users in the following channels.

- #blenderwiki<sup>[2]</sup>
- #blender<sup>[3]</sup>
- #blenderchat<sup>[4]</sup>
- #blenderqa<sup>[5]</sup>
- #gameblender<sup>[6]</sup>

If you can't get help there, click the "discussion" tab at the top of the page that you're having trouble with, and explain your problem on that page. Wait at least 24 hours for some help.

If you're still not getting help, try asking for help in the BlenderArtists.org forums<sup>[7]</sup>.

## References

- [1] <http://xchat.org>
- [2] <irc://irc.freenode.net/blenderwiki>
- [3] <irc://irc.freenode.net/blender>
- [4] <irc://irc.freenode.net/blenderchat>
- [5] <irc://irc.freenode.net/blenderqa>
- [6] <irc://irc.freenode.net/gameblender>
- [7] <http://blenderartists.org/forum>

---

# Unit One: Knowing Before Making

---

## Blender 3D: Noob to Pro/Unit 1: Knowing Before Making

---

In this unit you will be learning how to use Blender, a powerful and complex 3D modeling program. You will learn how to **understand** 3D, how to **cheat** the 3D, and most importantly, how to **think** in 3D. You will learn all the ins and outs of 3D modeling, using Blender and using your brain. Unfortunately for you eager types, there are quite a few steps before you start modeling. Keep in mind that you need to learn how to *use* Blender! **Don't** jump into modeling straight away!

Here are some **things you will need** before you start:

- Blender (you can get the latest version [here](#)<sup>[2]</sup>)
- An image editor for post-processing (GIMP<sup>[1]</sup> is an open source 2-D image manipulation)
- Pencil & paper for sketches

Blender 3-D software has many features and functions. You may find it useful to take notes while you are doing these tutorials.

## References

[1] <http://www.gimp.org/>

# Blender 3D: Noob to Pro/Avoiding Headaches Early On: Good 3D Modeling Practices

---

## Coping with Designing Problems

Many problems with modeling in 3D come from poor practice early on in the creation of your project. These kinds of problems can be difficult to rectify later on without completely recreating the project. This chapter will help you deal with these problems before they occur. Navigate to the next page to start the chapter.

**This chapter deals with certain subjects or terms you may not have learned yet. While this chapter is useful knowledge, it may be difficult to comprehend certain subjects within the chapter. It will be more useful to simply take written notes and save them for later than try and absorb the knowledge.**

## Blender 3D: Noob to Pro/Know What You're Doing

---

Blender is a complex program with a slew of customizable features and applications. This means there is a large chance you may run into a slew of problems!

### If you are going through tutorials, keep the defaults at the defaults!

In Blender, a user can customize mouse clicks and Hot-Keys to perform functions as well as change nearly every default setting to suit their preference. As cool and convenient as this is, we recommend that you keep all the settings at default so you won't be confused while participating in Noob to Pro tutorials. Wait until you have learned the basics before you start fiddling with the settings.

### Organize early, organize thoroughly!

In Blender you will be using many items, such as the models you create, virtual lights, cameras, and more. In a Scene there can be hundreds of these items. Also, there are elements such as the appearance of your models and animation data; this can all become overwhelming if it is not properly organized.

It is wise to properly label each and keep organized as you go. A three dimensional environment is a vast haystack to search for a needle in. There will be ways to search for items by name, so have this sorted early on. ((Needs Link to Organization tutorial))

### Don't bite off more than you can chew!

One of the greatest problems overzealous Blender users run into is their own imagination. While it is encouraged to be creative and imaginative, remember that you need to pace yourself.

- Make a very simple version of everything first, before you start detailing things. That way, problems come out lightly and are much easier to fix, as well as there are fewer issues to sift through when working with process of elimination.
- For everything, be aware of your workload and how detailed your project is getting. Remember that your computer has its limits, and be aware of how much "lag" there is when you are detailing and testing your project. Also remember: learn to cheat the 3D!
- If making an animation or game, test it out frequently when implementing a new section or element, to make sure it works before moving onto a new element.

These tips should help keep your project running smoothly, and while you may be tempted to go wild, keeping a level head while swimming in your imagination will give you a much more satisfying result, with many fewer headaches as well.

# Blender 3D: Noob to Pro/Cheat the 3D

---

## Ways to Improve Performance

In 3D Graphic Design, there are many issues to consider. First, consider how the project moves, how the project performs. Next, consider how you create objects in your project, especially when working with meshes and lighting.

### Mesh

#### Modifying an object's properties

When you add a modifying property to an object, the computer has to calculate that property every time it moves in the animation. These modifying properties include soft bodies, particles, or mirror textures. They can significantly slow down the computer.

To make things move faster, pre-calculate and "bake" the modifier into the object. To bake a modifier, such as a soft body movement, save the modifier as a permanent animation. That way no matter how the object changes, it moves the same way. As a result, your computer does not calculate how the object moves. The move is already set.

There is a drawback. If the object moves and hits another section with different slopes, it will continue to move as if it was in the original baking site. You will have to calculate and bake the section again to get new results.

#### Making 2D backgrounds

Scenes will have a foreground, background, sky, and more.

Your computer will take more time to render a scene that is big, open, and filled with modeled objects. This wastes time and memory, especially when modeling an object far in the distance. I'm guessing however, you do not want to make a quick, plain model. So what do you do?

That is easy. You make a 2D background.

- \* Put the distant image or background on a plane .
- \* Verify that everything on the plane around the image is black.

If it is not black, set the Alpha to 0 if the image was saved in a format that supports Alpha. This will retain the detail but reduce the number of faces the computer must count in the scene.

- \* Set the image to Billboard to prevent the image from skewing by perspective.

## **Materials**

### **Using UV maps**

For those who do not know the term, a UV map is an image applied to a 2D plane. The 2D plane is then applied to the mesh.

Here is how to create a UV map:

- Spread out the image onto a 2D plane.
- Apply the 2D plane over the face of the mesh using the 3D view.

Many people never consider this alternative. Instead, they create memory-exhaustive scenes that need many computer calculations to work. The UV map method takes fewer resources. You can then apply those resources to more important areas of the scene.

# **Blender 3D: Noob to Pro/Performance vs. Quality**

---

## **Mesh**

**Polygon Count** - when you work with meshes in 3D, be aware that the memory usage in Blender with reference to meshes comes from the number of faces (or vertices) the mesh contains. That means that if you have many useless faces (such as 55 faces on the side of a mesh that are as flat as one face), the number of faces will directly affect the Render time of Images and Animations. Also remember that for smooth rounded meshes, you do not need an infinite number of faces to make it look really nice and smooth. You actually don't need that many faces to keep a smooth slope looking smooth when viewed from the side. The viewer's eye will not actually identify every flat face, but will trace it as a smooth curve.

**Face Structure - 4-side (quad) vs. 3-side (tri)** - Poly Count is very important when working in 3D graphics, and depending on the type of mesh you are making you could be wasting a vast number of faces because they are not the right structure. For example, if you want a smooth object, you can make it look mostly smooth when viewed from the side if you make it with well-structured 3-vertex faces. You will also yield the same results with fewer faces than if you use 4-vertex faces. Another example is that if you need to apply UVTexture, or have something to animate, a mesh being evenly deformed will yield a lower face-count if it is a 4-vertex versus a 3-vertex face-structure.

## **Lighting**

**Using RayShadows or Buffer Shadows** - There are a few options you have for using shadows. Ray Shadows use an advanced algorithm to trace the edge of any interfering objects and create a perfect shadow onto the receiving object(s); however, the Ray Shadow calculation is memory-intensive and can seriously slow down your Render-Time. Buffer Shadows, on the other hand, use a different algorithm for similar results. The difference is that buffer shadows use a bit-rate of shadow "pixels" that fill in the shadowing area. You can adjust the bit-rate to make the shadow higher or lower quality. This calculation is much more memory-friendly, and your Render-Time will not jump up as much as with Ray Shadows. An interesting topic because this will change with better hardware performance and eventually won't matter. Until we get into 3D.

# Blender 3D: Noob to Pro/Introduction

---

Most tutorials out there for Blender and other 3D applications assume some familiarity with three-dimensional modeling and visualization. Since not all of us have had the privilege of taking a high school technical drawing class, the fundamentals of 3D graphics may be elusive to grasp. This section of the book is intended to introduce newcomers to different techniques used to model the three-dimensional world. It is my hope that this section will help you visualize the world in a new way that you can translate into Blender as well as any CAD or 3D application.

As with any subject, there is a special vocabulary associated with 3D design. Do not fret, for all of these new terms will be highlighted in **bold** and summarily defined. These will come in handy later when reading other tutorials.

## Blender 3D: Noob to Pro/Understanding coordinates

---

### Understanding Coordinates

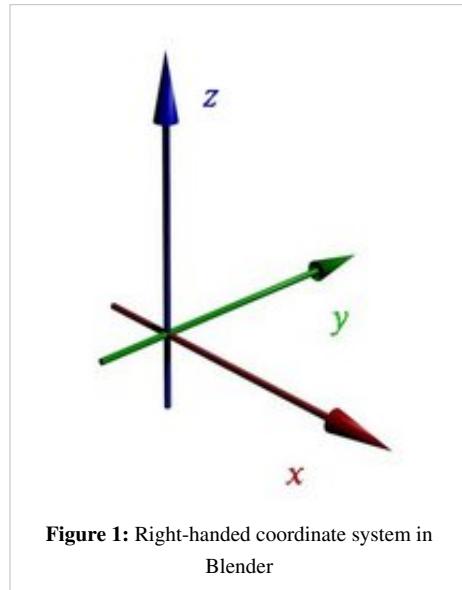
You may already be familiar with the X and Y axis from math class. Remember having to graph lines and equations of the form  $y = mx + b$ ? Well, 3D adds another wrinkle to this: the Z axis! Where X was "left" and "right" on graph paper and Y was "up" and "down", Z adds a third dimension: depth. That's right, the Z-axis pops right off the page! So how does one go about graphing it? Well, on paper, it usually involves drawing a diagonal line, putting tick marks on it and plotting points. This is not without its problems, so the preferred methods are **orthographic projections**, **isometric projections** and **perspective views**, all of which will be elaborated on in the following sections.

# Blender 3D: Noob to Pro/Understanding Coordinates

*This tutorial has been created with Blender v2.47*

You may already be familiar with the X and Y axes from math class, where you were asked to plot points on a set of axes, namely the X and Y axes. Since we will be dealing with three dimensions, you will now become acquainted with the Z axis.

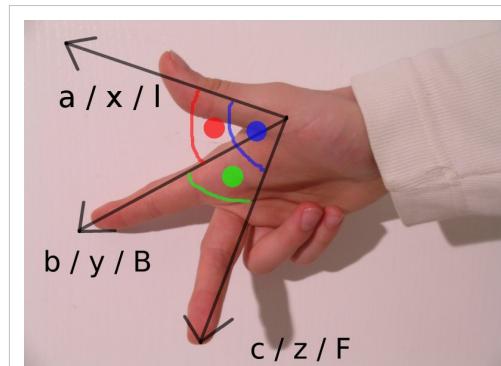
It feels quite natural to let the Z-axis run up and down, and keep the X-axis point to the right, but where shall the Y-axis go? Well, it points away from you into your monitor (**Fig. 1**). So how do you go about graphing it? Well, to sketch it on paper, you would draw a line that appears to go backward and draw tick marks to show its *depth*.



**Figure 1:** Right-handed coordinate system in Blender

Because of the way the axes are orientated, this system is called **right-handed**. If you let the thumb of your right hand point in the direction of the X-axis, your index finger points in the direction of the Y-axis, and the braced middle finger points in the direction of the Z-axis.

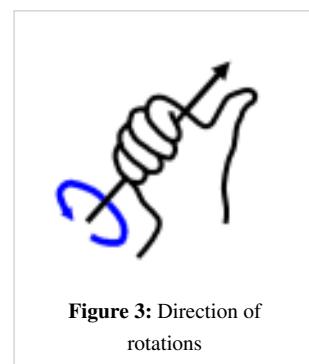
As you can see in **Fig. 2**, you can rotate the coordinate system, so it is not important where the axes point to, as long as their relative relations are kept, e.g. you will often find that the Y-axis points upwards and the Z-axis points towards you.



**Figure 2:** Right handed system

If you use your right hand again you can also determine the direction of a rotation. If you point your thumb into the direction of the positive rotation axis and close your fingers, they show the direction of the rotation, i.e. the angles get larger in that direction.

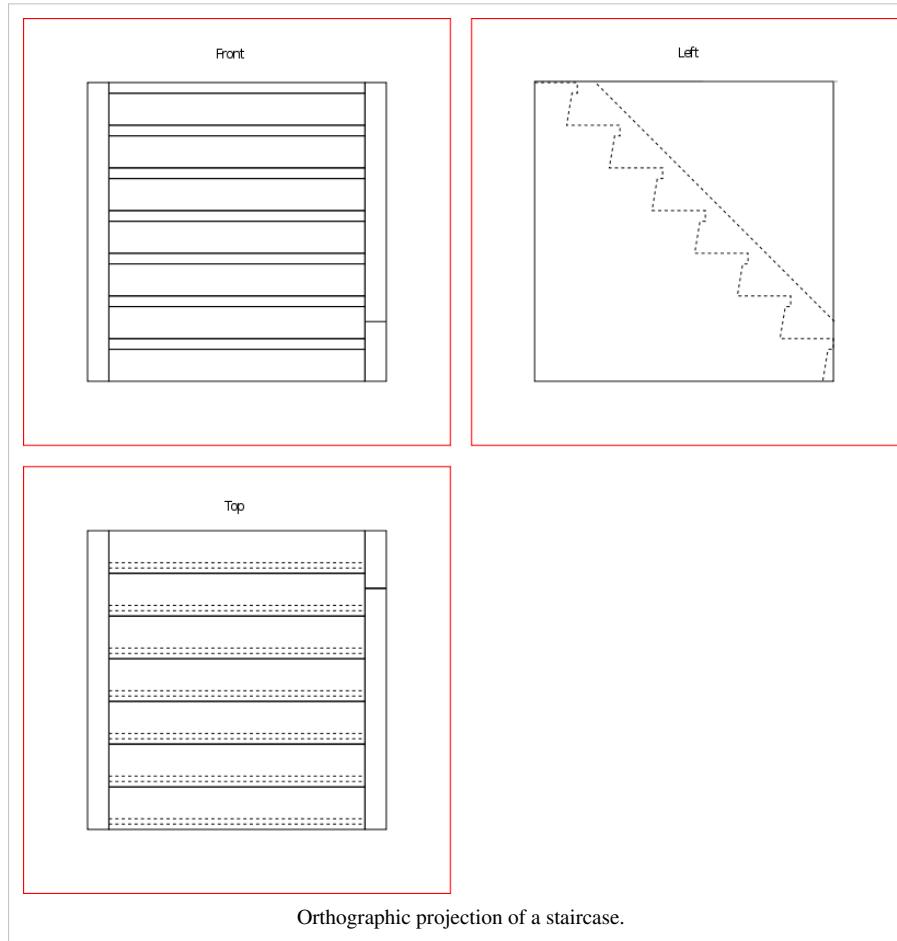
There are several ways to depict a 3D object on a 2D surface (paper, monitor, etc.). In this book we will cover the three most important types of projections/views: **orthographic**, **isometric** and **perspective**.



**Figure 3:** Direction of rotations

# Blender 3D: Noob to Pro/Orthographic Projections

Orthographic projections are the staple of 3D visualization. In short, they describe the shape of an object from at least two (usually three) different angles. From this information alone, one can come up with the full shape of a 3D object. The normal views used are front, side, and top, though back, and bottom may also be used. In orthographic projections, everything is to scale and proportional.



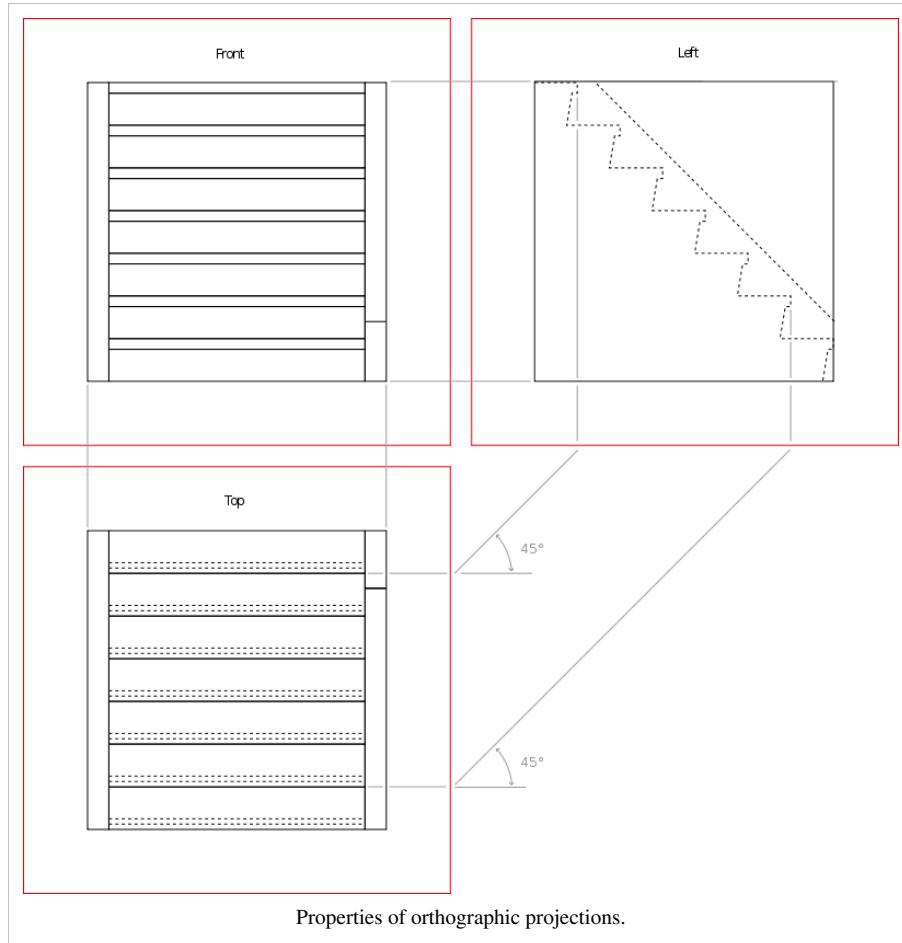
Notice in the above drawing there are three views (outlined in red). These represent the object from the front, top, and left. You may notice some dashed lines on the left view. These are there to indicate that these lines are behind something, namely the left wall of the staircase. Ordinarily, this projection would be made with the right view so that no lines would be hidden.

Earlier, I mentioned that at least two views are necessary. This is because of some of the properties of orthographic projections, namely:

1. The front view is constrained by the left (side) and top views. Any horizontal line in the side view will match a corresponding line in the front view. That is, if you draw lines from the top of each stair in the side projection, it'll be the top of that particular stair in the front view as well. Likewise, were you to extend a vertical line from the wall beside the staircase in the top view into the front view, it would also be the same wall. Using this property, it is fairly simple to "box" in every feature of the object in the front view.
2. The top view is constrained by the left and front views. This one is harder to see, as one cannot draw a vertical line from the "left" view and have it intersect anything. For this, we imagine that each vertical line in the left view extends to the bottom edge of the red box. We will then draw  $45^\circ$  diagonal lines from each of these points of

intersection down and left towards the top view. Where these intersect the bounds of the red box are *horizontal* lines in the top view, the length of which is constrained by vertical lines from the front view.

3. The left view follows the same logic as above. Simply draw the diagonal lines up and to the right to determine where each vertical line is. The height of each line is constrained by the front view.



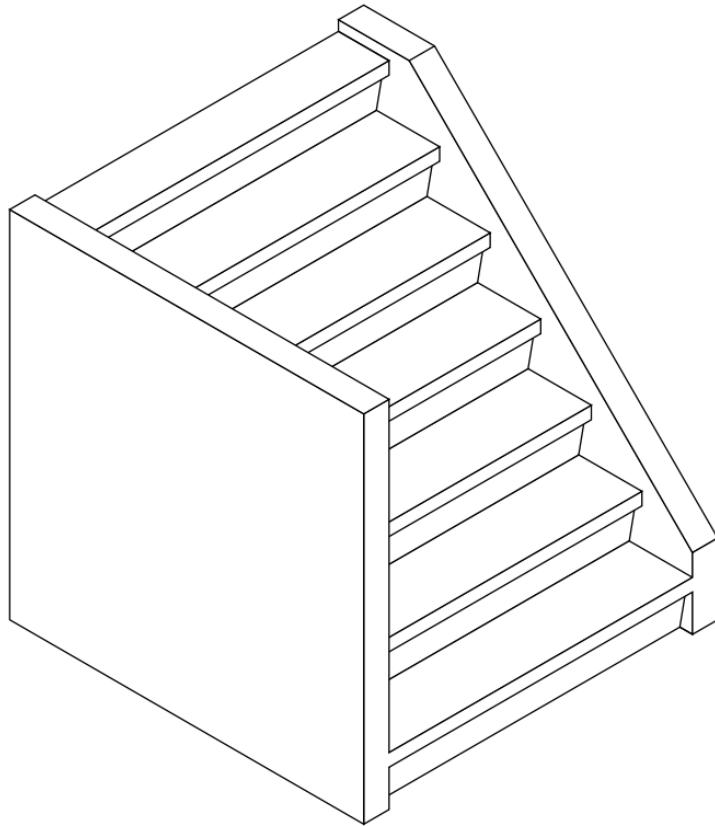
However, it should be noted that different projections, e.g. top-right-front have slightly different properties. For example, were I to have used the right projection, the layout would need to be almost a mirror-image with the right view on the upper left, the front view to the right of it, and the top view on the lower right, in order to keep these properties.

Some cases are ambiguous, such as when only the top and front view of the staircase are drawn. In this case, it would be impossible to know whether the riser of each step slopes in or out. Therefore, the side view is necessary. However, if instead only the side and front views were drawn, there would be no such ambiguity. So in most cases, two projections are all that are required.

# Blender 3D: Noob to Pro/Isometric Projections

Now that you have familiarized yourself with orthographic projections, you're wondering how this could possibly get any more difficult to visualize. Well, fret not, for another type of projection can do the job of orthographic projections, and is just as precise, though more ambiguous in some cases. I speak of isometric projections. It is likely that you have already seen an isometric view of something but may not have realized it. Indeed, many older games used isometric views in lieu of actual 3D.

The word **isometric** contains *iso* meaning *same* and *metric* meaning *measurements*. Simply put, it is a way of representing a three dimensional object in one view rather than three. To do this, one draws a set of axes such that each is at 120 degree angles with the Z axis vertical. Then, if one is drawing from orthographic projections, each view is represented in a pseudo-3D manner with the measurements intact, but at skewed angles.



An isometric view of the staircase in the previous section

In Blender, the isometric view is slightly different: it is not simply constrained to 30, 60 and 90 degree angles as conventional isometric drawings are. One may move the camera around in Blender's 3D space to view different facets of the object. Despite being able to zoom in and out, **every line remains to scale**.

# Blender 3D: Noob to Pro/Perspective Views

So now you know how to *simulate* 3D using isometric projections. By now, I'm sure you're hungering for a way to do 3D for real. There are several approaches to drawing in perspective. These are one-point perspective, two-point perspective, and three-point perspective. Each of these points refers to a **vanishing point**.

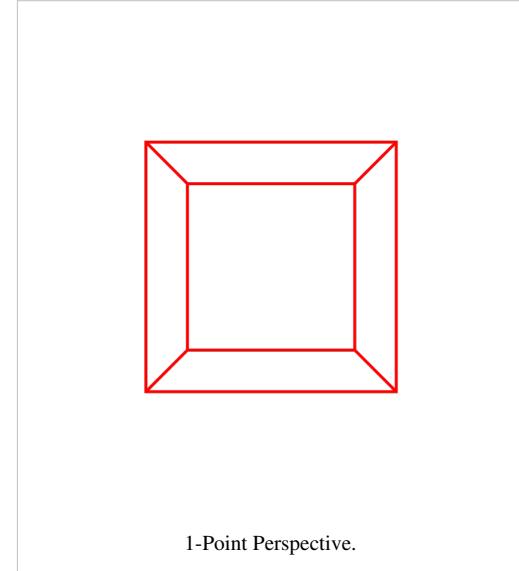
## One-point perspective

This is the kind of perspective that you use if you are facing an object head-on and staring right at the middle of it.

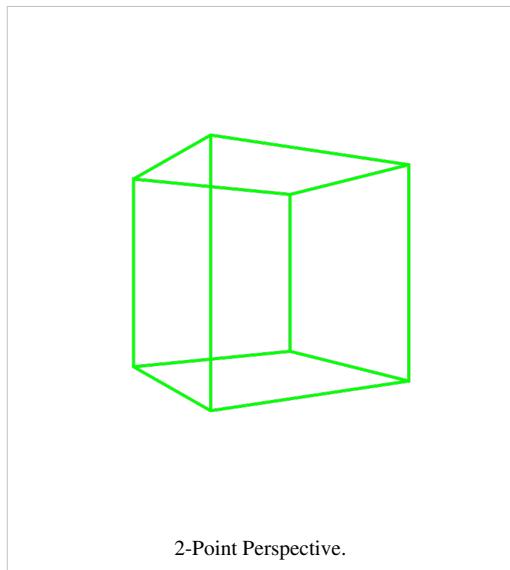
Imagine looking down a straight set of train tracks. The tracks will appear to converge at some point on the horizon. This is the vanishing point.

One-point perspective is the easiest to draw. Suppose you want to draw a cube. First put a single dot in the middle of the page: your vanishing point. Now extend lines out from the vanishing point, at each of the 4 diagonals. Draw 2 squares, one "inside" the other, so that all 8 corners lie on these 4 lines. Now connect all corresponding corners and you have a cube.

You can see that some of the lines in this cube are parallel, and some converge. Specifically, the left-right lines are parallel, the up-down lines are parallel, but the in-out lines converge. Since this is one-point perspective, there is only one point of convergence, **right in the middle**.



## Two-point perspective



But what if you're looking at the corner of something? Say the corner of a large department store that extends for a couple blocks. Now what? You can't use one-point perspective. Here, you must use two-point perspective. This requires two vanishing points, usually to the left and right. Draw the corner of the building as a vertical line, and then extend the lines toward the vanishing points.

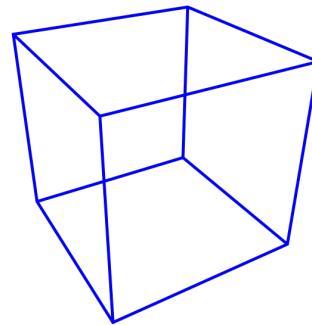
In the diagram, you can see that the up-down lines are still parallel, but now there are two sets of lines that converge: the four left-right lines converge off to the right, and the four in-out lines converge off to the left.

## 3-Point Perspective

Now here's the rub: you're looking down on that same building from a rooftop across the street. Now what? Three-point perspective, or the kind you actually see in reality. Here, you draw three vanishing points: left, right and *up*. Proceed drawing lines as with two-point perspective, but this time the sides of the building towards the distance will not be drawn straight up, but towards the vertical vanishing point. In this case, since you are looking down at the department store, the vertical vanishing point will be below the left and right points.

In this case, there are no longer any lines which appear parallel. The 4 up-down lines, the 4 left-right lines, and the 4 in-out lines all have their very own vanishing points that they converge to.

All of these are a pain to draw. Enter Blender, a program that will do it for you, provided you give it information about each side of an object (orthographic projections), and a shape that makes sense in perspective and isometric views. All these will become routine in getting to know Blender.



3-Point Perspective.

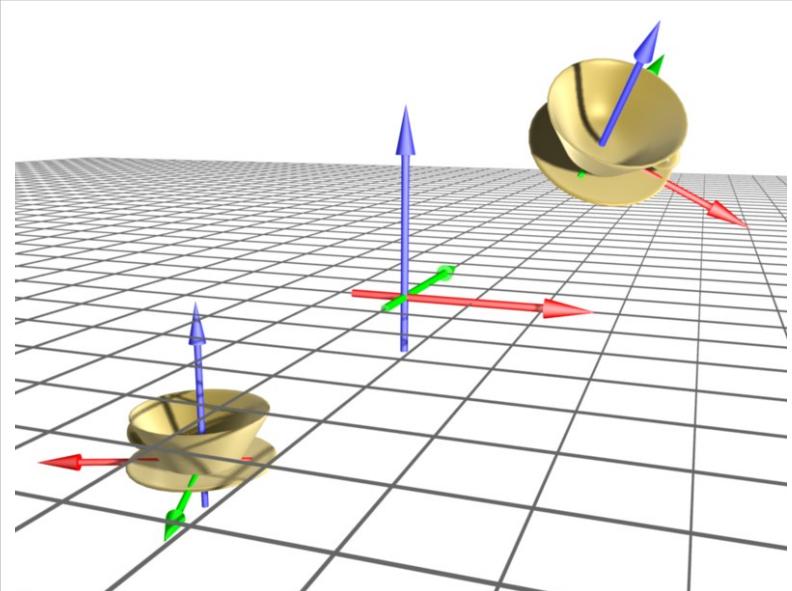
## Blender 3D: Noob to Pro/Coordinate Spaces in Blender

*This tutorial has been created with Blender v2.46*

Understanding three dimensions is one point - but how is this concept implemented in Blender?

As you have already learned, the location of an object in a three-dimensional space is defined by three descriptors (typically numbers). These three numbers make up what is called the coordinate of the location.

Blender uses the following convention: At the center of the coordinate space is its origin, the zero point. The distance from the zero point in three orthogonal directions is used to locate an object. The three directions are called the X, Y, and Z axes and are denoted by the colors red, green and blue. Viewed from the front, the X axis (red) points to the right, the Y axis (green) points to the rear, and the Z axis (blue) points upward.



**Figure 1:** Objects in a three dimensional space. In the center of the coordinate system is the origin of the world coordinates.

## Global and local coordinates

The coordinate system in the above described coordinate space is in Blender usually called the **global coordinate system**, though we probably should refer to it as the **world coordinate system**. The world coordinate system has a fixed origin and a fixed orientation, but we can view it from different angles when we rotate or pan around the world space.

The coordinates in the world coordinate system would entirely suffice, in principle, if we were working only with objects with a single vertex. But an object like the cup shown in **Fig. 1** may consist of any number of vertices (in this case, the cup has 171 vertices). Therefore it is much more practical to introduce a second coordinate system: the **local coordinate system**. The vertices from which the cup is built are defined in the local coordinate system. The origin of the local coordinate system is called the **center of the object** but is not necessarily located at the physical centre of the object. If you rotate or move the object the shape remains unaltered because the orientation of the object's local coordinate system is fixed relative to the object.

A three dimensional object is defined:

- by the vertices in its local coordinate system
- the location of the centre of the object
- the rotation of the local coordinates relative to the global coordinates

## Global coordinates for objects with parents

If an object has a so called *Parent*, the center of its global coordinate system is no longer the position 0/0/0 of the world system, but the object center of the parent. The orientation of the global coordinates is now the orientation of the parent.

So if you move the parent, its child is moved also (without changing the child's coordinates). If you rotate the parent the child is rotated also. In this way rotations along arbitrary axes are animated.

In **Fig. 1b** the cup is a child object of the right coordinate cross. The child itself has no local rotation. The right coordinate cross has itself an invisible parent. It is parent and child at the same time. The right coordinate cross is rotated along its global Z axis, the tilt against the world axis is created by the invisible *grandparent*.

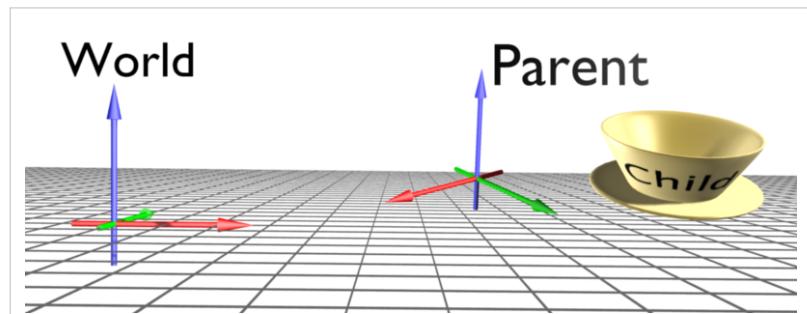
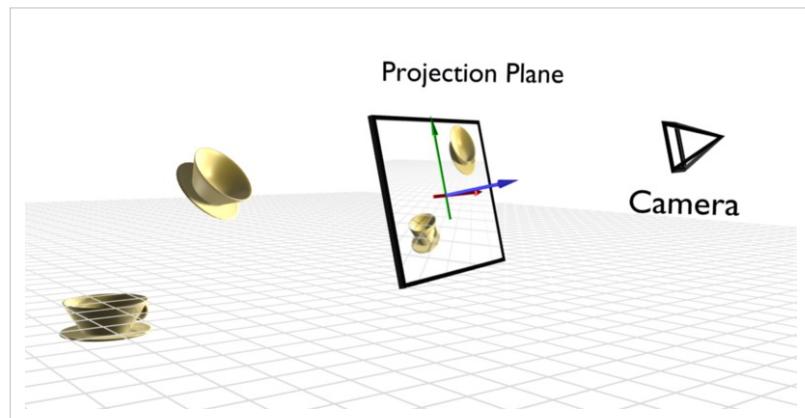


Figure 1b: A parent serves as the source of the global coordinates for its child object. The child is the cup, the parent's orientation is shown with the colored arrows. Animation of the above

## View coordinates

Taking the viewer of the scene in consideration, there is another coordinate space: the view coordinates. In **Fig. 2** the viewer is symbolized by the camera. The Z axis of the view coordinates always points directly to the viewer in orthographic projection. The X axis points to the right, the Y axis points upwards (**Fig. 3**).

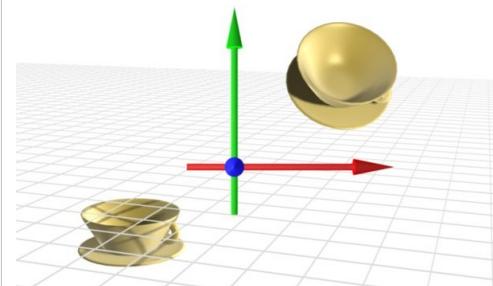


**Figure 2:** View coordinates and Projection Plane

In fact you always work in view coordinates if you don't set it any other way. This is particularly handy if you have aligned your view prior to modeling something, e.g. if an object has a slanted roof and you want to create a window to fit in this roof, it would be very complicated to build this window aligned to the local coordinate system of the object, but if you first align your view to the slanted roof you can easily work in this view coordinate system.

If you work in one of the three standard views (Front/Top/Side) the alignment of the view coordinates fits the world coordinates, therefore it is quite natural to model in one of the standard views and many people find this the best way to model.

## View Orientation

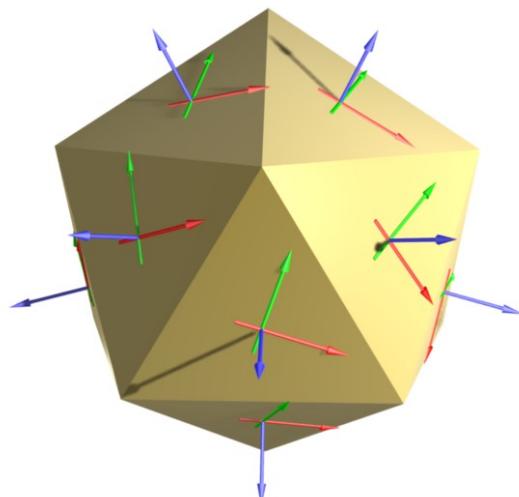


**Figure 3:** View coordinates in viewing direction

## Normal coordinates

Although Blender is a 3D program, only the faces are visible. The orientation of the faces is important for many reasons. For example, in our daily lives it seems quite obvious that a book lies flat on a table. This requires the surface of the table and that of the book to be parallel to each other. If we put a book on a table in a 3D program there is no mechanism that parallels the surfaces, we have to ensure that ourselves.

The orientation of a face can be described with the help of a so-called surface normal. It is always perpendicular to the surface. If several

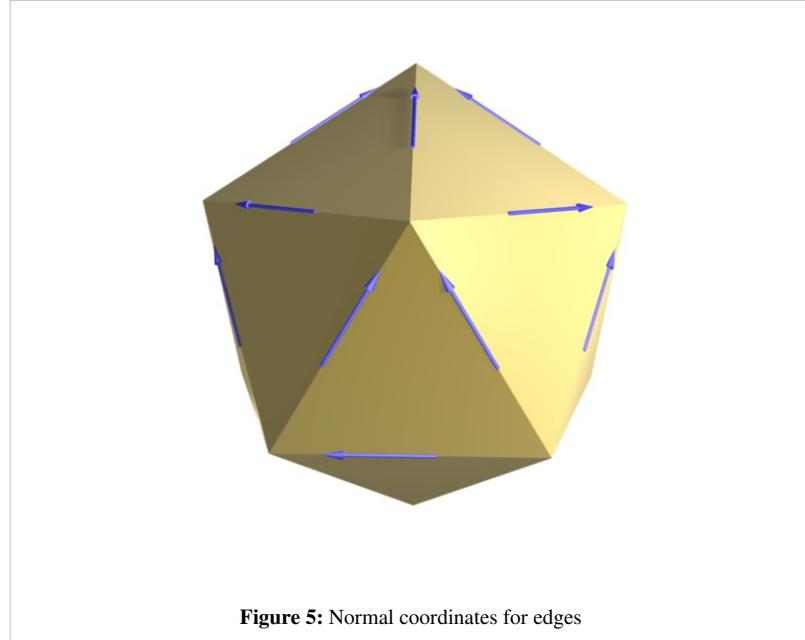


**Figure 4:** Normal coordinate spaces for faces. The normal is shown in blue.

faces are selected, the resulting normal is averaged from the normals of every single face. In **Fig. 4** the normal coordinates of the visible faces are drawn.

This concept can be applied to individual points on the object, even if the points themselves have no orientation. The normal of a point is the average of the adjacent faces.

The normal definition is more useful for edges: the normal runs along the edge (**Fig. 5**). For example, if you want to bend an object along an edge, you simply have to bend it along its normal.



**Figure 5:** Normal coordinates for edges

## Blender 3D: Noob to Pro/Tutorial Syntax

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

Most keyboards have number keys in two different places: in a row above the letters and in the numeric keypad on the right of the keyboard. While many computer applications use these two sets of keys interchangeably, Blender does not: It assigns a different set of functions to each set. If you don't have a numeric keypad ("numpad") see Blender 3D: Noob to Pro/Non-standard equipment for information on emulating the keypad's functionality.

(If you don't understand the manual, please e-mail your request to: [Blender-easy@live.co.za](mailto:Blender-easy@live.co.za))

## Special and Function

Abbreviation	Key
<b>ALT</b>	Alt key
<b>CTRL</b>	Ctrl (Control) key
<b>CMD</b>	Command key[1]
<b>F1</b> through <b>F12</b>	F1 through F12
<b>SHIFT</b>	Shift
<b>SPACE</b>	Spacebar
<b>TAB</b>	Tab
<b>ENTER</b>	Enter
<b>ESC</b>	Escape
<b>FN</b>	Function[2]

## Alpha-numeric and Numpad

Abbreviation	Key
<b>AKEY</b> through <b>ZKEY</b>	corresponding letter on the keyboard
<b>0KEY</b> through <b>9KEY</b>	corresponding number (above the letters) on the keyboard
<b>NUM0</b> through <b>NUM9</b>	corresponding number on the numpad
<b>NUM+</b> and <b>NUM-</b>	corresponding key on the numpad

## Mouse

Blender uses all three mouse buttons. If you don't have a three-button mouse (a one-button Apple mouse, for example), see Blender 3D: Noob to Pro/Non-standard equipment for information on emulating a three-button mouse.

### Mouse button definitions

Key	Description
<b>LMB</b>	the left mouse button (typically)
<b>RMB</b>	the right mouse button (typically)
<b>MMB</b>	the middle mouse button or mouse wheel (clicked, not spun), or you can use ALT + left click
<b>SCROLL</b>	refers to scrolling the wheel of the mouse

## Path menu

**SPACE** → *Add* → *Mesh* → *UVsphere*

means:

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

## Footnotes

1. ^ On an Apple keyboard, the key with the Apple logo, and on a Windows keyboard, the key with the Windows logo on it. These are also called the "Super" key in other manuals.
2. ^ **FN** is generally found only on laptops, and often in the lower left corner of the keyboard. (Some "ergonomic" keyboards use the laptop-style *pseudo-numpad* to reduce the distance between the normal keys and the mouse. On these, the **FN** key is often in the top row.)

## References

- [1] [http://en.wikipedia.org/wiki/Blender\\_3d%3A\\_noob\\_to\\_pro%2Ftutorial\\_syntax#endnote\\_command](http://en.wikipedia.org/wiki/Blender_3d%3A_noob_to_pro%2Ftutorial_syntax#endnote_command)  
[2] [http://en.wikipedia.org/wiki/Blender\\_3d%3A\\_noob\\_to\\_pro%2Ftutorial\\_syntax#endnote\\_FN](http://en.wikipedia.org/wiki/Blender_3d%3A_noob_to_pro%2Ftutorial_syntax#endnote_FN)

# Blender 3D: Noob to Pro/Non-standard equipment

---

Here you will find information applicable only to users with non-standard equipment. If you have a three-button mouse and a keyboard with a numeric keypad, you can skip this section.

## Keyboards lacking a numeric keypad

Most modern laptops have a set of regular keys with additional markings corresponding to a traditional numeric keypad ("numpad").<sup>[1]</sup> This *pseudo-numpad*'s behavior can often be toggled with **F11** or **Num Lock** on PCs, or **F6** on Macs. Alternatively, you can often temporarily activate the numpad behavior by holding the **FN** key 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 see the alternate labellings but don't know how they work, see your laptop's owner's manual.

As a last resort, you can use the *Emulate Numpad* functionality built in to Blender. This will allow you to use the normal numbers as if they were the numpad numbers. To enable this feature, go to the *User Preferences* window, click the *System and OpenGL* button, and click *Emulate Numpad*. **NUM0** will now be the same as **0KEY**.

Blender uses the numeric keypad quite a bit. If you envision using your laptop for this kind of work, it may be worth investing in a *USB Numeric Keypad*. On eBay, prices for simple external keypads start at around \$10 USD.

## Non three-button mice

For single-button mouse users, make sure that *View & Controls* (under "User Preferences" on the left-most drop-down menu) → *Emulate 3 Button Mouse* is enabled. The **MMB** can be emulated on most Windows and Linux machines by simultaneously clicking **LMB** and **RMB**. You'll need to set this up in the mouse settings in your Control Panel. On a Mac, you can accomplish this by opening the *Keyboard and Mouse* preference pane and enabling *Use two fingers to scroll*. Recent IBM Thinkpad laptops allow you to disable the 'UltraNav' features of the middle mouse button in order to use it as a 'normal' third button. Alternatively, some laptops allow areas (called gestures) on the movement pad to act as right button click or middle button clicks, and can be set up in the Control

Panel in the Mouse Pointer options, selecting gestures and editing features there.

### **Apple single-button mouse**

#### **Apple single-button mouse substitutions**

PC	Apple Analog	Description
<b>LMB</b>	<b>MB</b>	the mouse button (default)
<b>RMB</b>	<b>CMD+MB</b>	Apple key + the mouse button
<b>MMB</b>	<b>ALT+MB</b>	Option (Alt) key pressed + the mouse button

While Mac OS X natively uses both the *Control* key and the *Command* key to emulate the **RMB**, recent Blender versions for Mac OS X use only 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.

### **Footnotes**

1. ^ These keys typical cover the keys in the following locations {{7-0}, {U-P}, {J-;}, {M-/}}, or more simply, from **M** in the lower left to **0** in the upper right.

### **References**

[1] [http://en.wikipedia.org/wiki/Blender\\_3d%3A\\_noob\\_to\\_pro%2Fnon-standard\\_equipment#endnote\\_typical](http://en.wikipedia.org/wiki/Blender_3d%3A_noob_to_pro%2Fnon-standard_equipment#endnote_typical)

## **Blender 3D: Noob to Pro/Operating System specific notes**

---

Here we will put in specifics that pertain to single operating systems.

### **Linux**

If the **ALT+LMB** key combination moves the current window then there's a conflict with your window manager. You can resolve the conflict or use **CTRL+ALT+LMB** or **MMB** instead. If you are running KDE this can be resolved by: right-click on the title bar of the main Blender window → select *Configure Window Behavior* → go to *Actions* → *Window Actions* → in the *Inner Window, Titlebar and Frame* section → select the Modifier key to be **ALT** and set all the select boxes beneath it to *Nothing*.

In Gnome, Click *System* → *Preferences* → *Window Preferences*. Look for the last three options *Control*, *Alt* and *Super*. Select *Super*. Now you can hold super to drag windows around, and use **CTRL** and **ALT** as normal.

## Gnome

You're probably not going to want to use the *Find Pointer* functionality in Gnome, it will impair your ability to use certain functions such as *Snap to grid* or using the lasso tool. If your mouse pointer is being highlighted when you press and release the CTRL key, go to: *Mouse* in Gnome's *Desktop Settings* and uncheck the box *Find Pointer*.

## Ubuntu

There is a known incompatibility between Blender and the Compiz Fusion accelerated (OpenGL) window manager used in Ubuntu. By default, Compiz Fusion is enabled in Ubuntu, causing the problems to manifest themselves in Blender as flickering windows, completely disappearing windows, inconsistent window refreshes, and/or an inability to start Blender in windowed mode.

The fix for this is simple. Install compiz-switch (might be in universe). Go to Applications->Accessories->Compiz-Switch. This will disable compiz temporarily. Do the same to turn compiz back on when you're done using Blender.

## Mac OS X

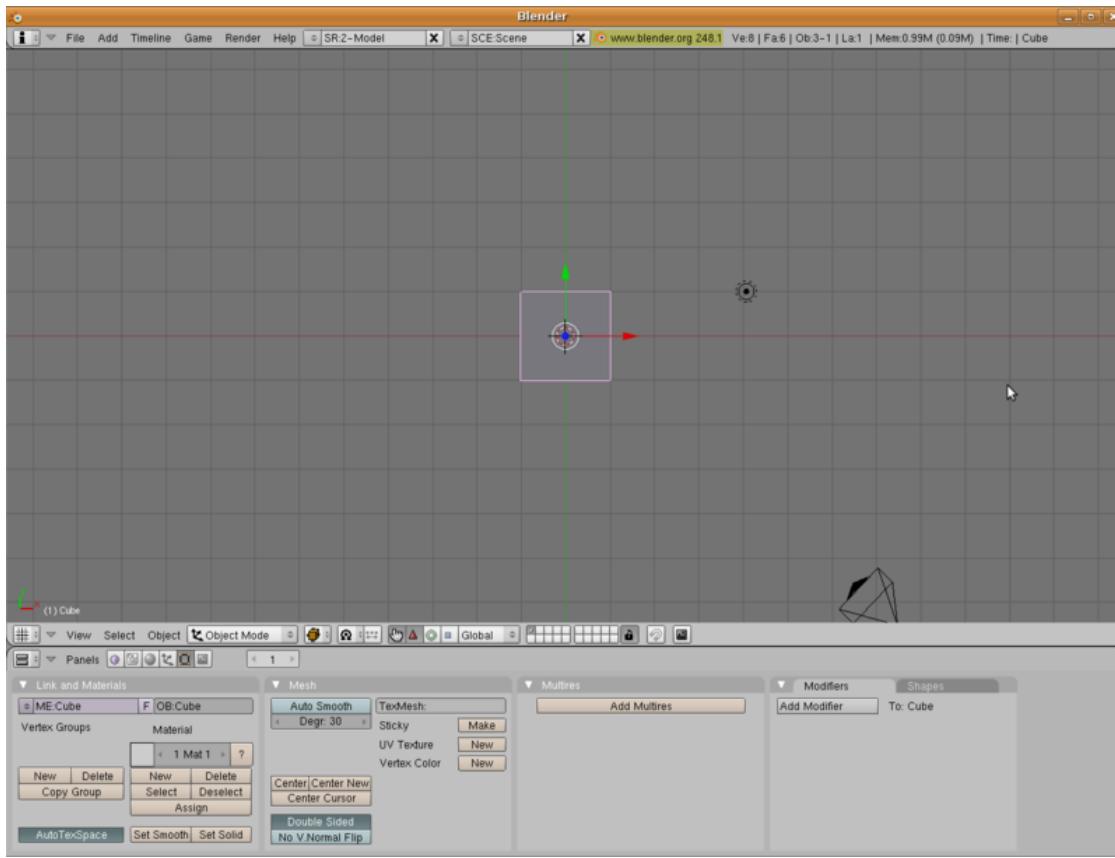
On Macs with the new thin Apple Keyboard, you may need to press "fn" in order to use the F1-F4 and F7-F12 keys. To expand a section in Blender, you should usually press Ctrl+Up, on mac if "Spaces" is enabled you may have to press Ctrl+Alt+Up.

## Windows

Pressing Shift five times in a row turns on the *StickyKeys*, but this is often not the user's intention. Doing so will change the way the computer recognizes commands. If the box for sticky keys appears, press cancel. If you don't need the accessibility features, you can disable sticky keys: 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.

The simultaneous pressing of **SHIFT** and **ALT** to achieve **SHIFT+ALT** will sometimes switch the keyboard layout, for example QWERTY becomes AZERTY and vice versa. For this reason, the tutorial does not teach the alternative **SHIFT+ALT** methods. If you find your keyboard layout altered you can press **SHIFT+ALT** again to toggle it back. You can disable this key sequence: *Windows Control Panel → Regional and Language Options → Languages (tab) → Details (button) → Key Settings (button) → Change Key Sequence (button)* and uncheck the *Switch Keyboard Layout* checkbox.

# Blender 3D: Noob to Pro/Blender Interface



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.

# Blender 3D: Noob to Pro/Blender Windowing System

---

## Before You Begin

Most readers will run Blender while reading this book, and the book is intended to be used this way. You may be surprised to learn that the first time you run Blender, it runs in full-screen by default. It's not strange that most experienced users prefer this mode, as having a large working area is a must. However, while becoming pros, we need an easy way to switch between Blender and browser windows. You can use OS keyboard shortcuts to switch between applications: **ALT+TAB** (Linux&Windows) **CMD+TAB** (Mac); **LAmiga+M** to switch between programs/screens (Amiga); Amiga window gadget far top right to switch between windows/screens - front/back; or to bring up your desktop: **CTRL+ALT+D** (Linux), **WIN+D** (Windows), **WIN+TAB** (Windows Vista & Windows 7), **F3/F11** (Mac), (Amiga - same as above), but keep reading if you find them non-productive. In Linux, running Blender in a separate workspace provides access using a single click of the mouse. Similarly, in Mac OS X you can switch on "spaces" and either use the mouse key to switch between them, or cmd and the directional arrows.

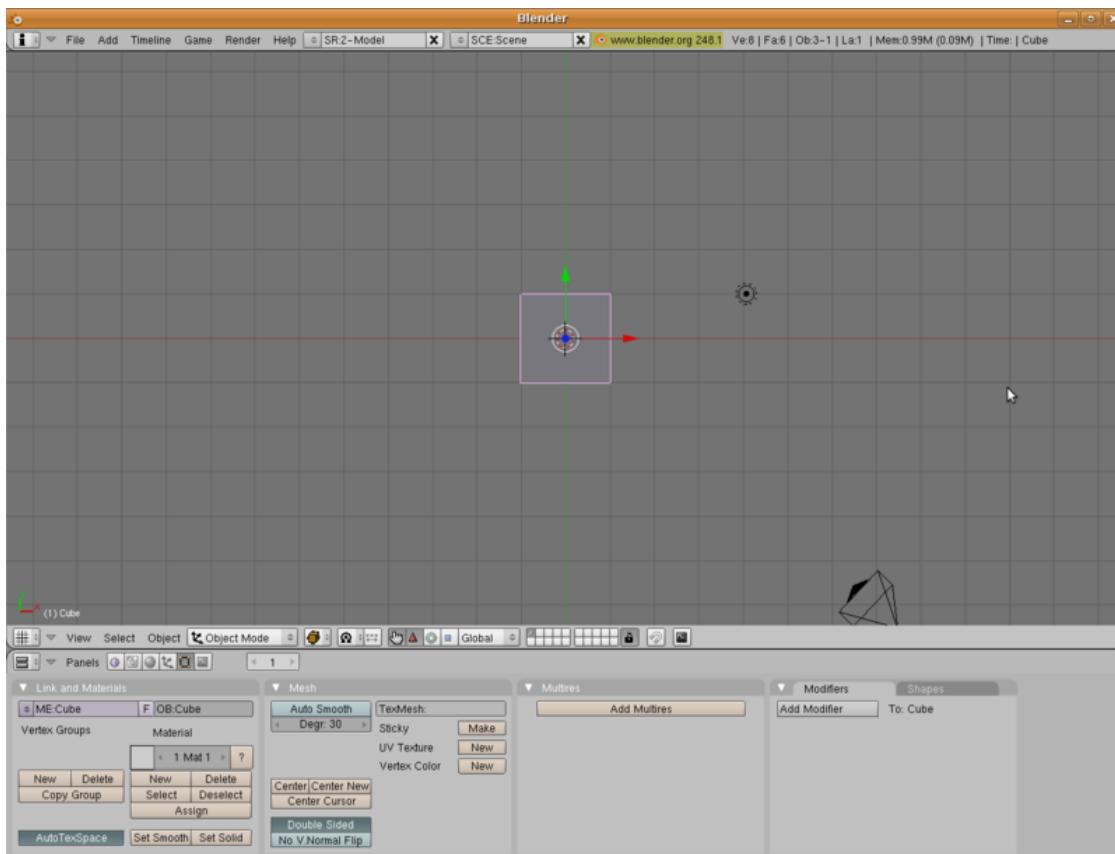
Fortunately, Blender is provided with some command line <sup>[1]</sup> arguments/parameters that will be helpful at this.

- **-w** results in Blender window opening non maximized, but this is not enough because the window is still at full size.
- **-p <sx> <sy> <w> <h>** where *sx* and *sy* are the values for the position the lower left corner will start at, *w* is width and *h* is height (all in pixels).

**For Example:** *blender -p 0 68 1255 956* results in Blender opening in a 1255x956 window aligned to the top left corner of the screen and leaving a 25 pixels margin at right and 68 pixels at bottom. This is ideal for a 1280x1024 screen resolution setup. Put your browser aligned to the right side of the screen, and switching between the two applications is as easy as clicking the lower right corner of the window that you want to bring to the front. Tweak the values until you find a comfortable working setup and be aware that the **-w** switch could be required or not depending on the OS you are using. Now working with Blender and reading the book are both a single click away.

## An Interface Divided Will Surely Stand

The Blender interface can be a bit intimidating at first, but don't despair. We will explore the power and flexibility of the Blender windowing system, and how to adapt it to suit your needs, one step at a time. First, we're going to talk about manipulating the **3D Viewport** and the **Buttons Window**.



Go ahead and open Blender if you haven't already. You'll hopefully be presented with something that looks a lot like this. You should be able to see two major divisions. There are actually three, but the third one is **hidden**. We'll talk more about the hidden one later.

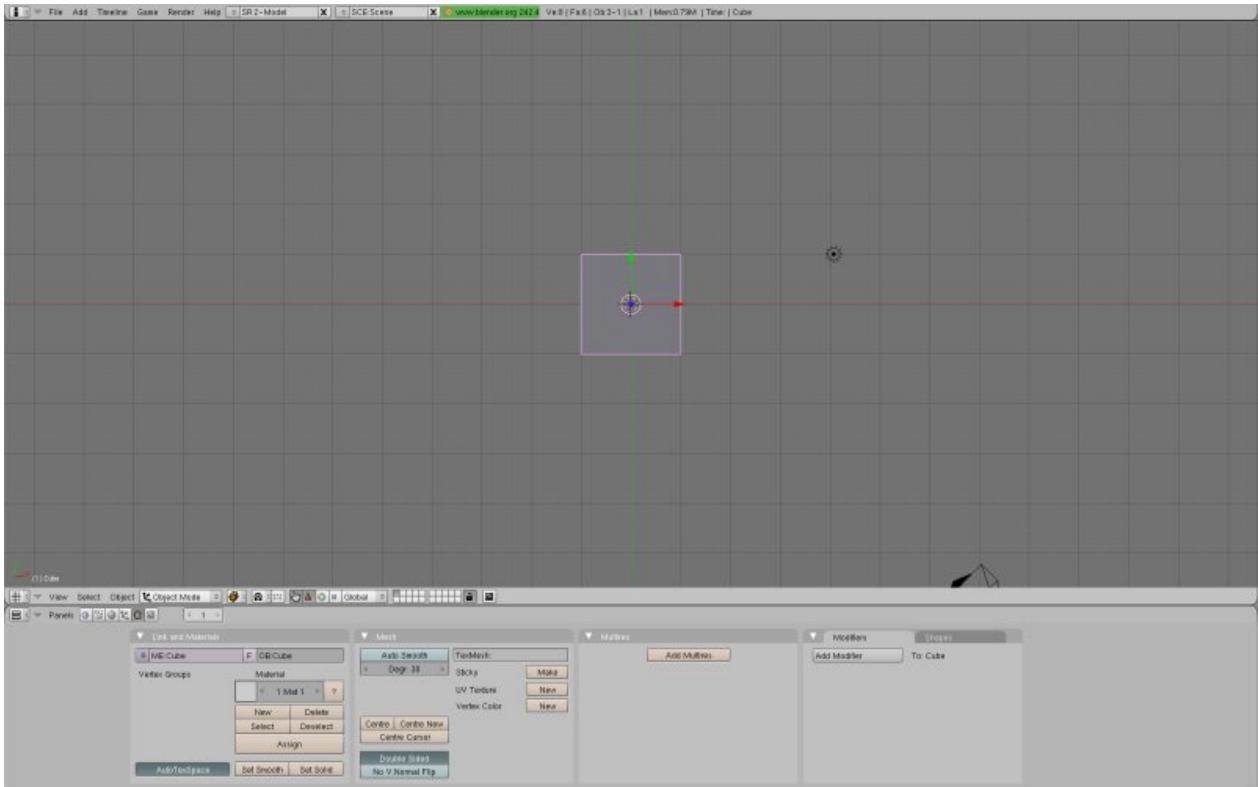
This top larger portion is the **3D Viewport** window. It allows you to see and manipulate the 3D objects in your 3D scene.

This section on the bottom is the **Buttons Window**. The buttons in the **Buttons Window** will allow you to manipulate the 3D objects you see in the 3D Viewport in many different ways.

The **3D Viewport's** grid represents **Blender Units (BU)**. A BU can be as large as you would like it to be: an inch, a centimeter, a mile, or a cubit. Blender lets you decide the scale.

## Window Headers

Every window has a window header. The window header can be at the top of a window, at the bottom of a window, or hidden. Let's take a look at the window headers for our **3D Viewport** and our **Buttons Window**.



The header for the **3D Viewport** is highlighted in **red**. Notice that it's actually at the bottom of the 3D Viewport, and not at the top.

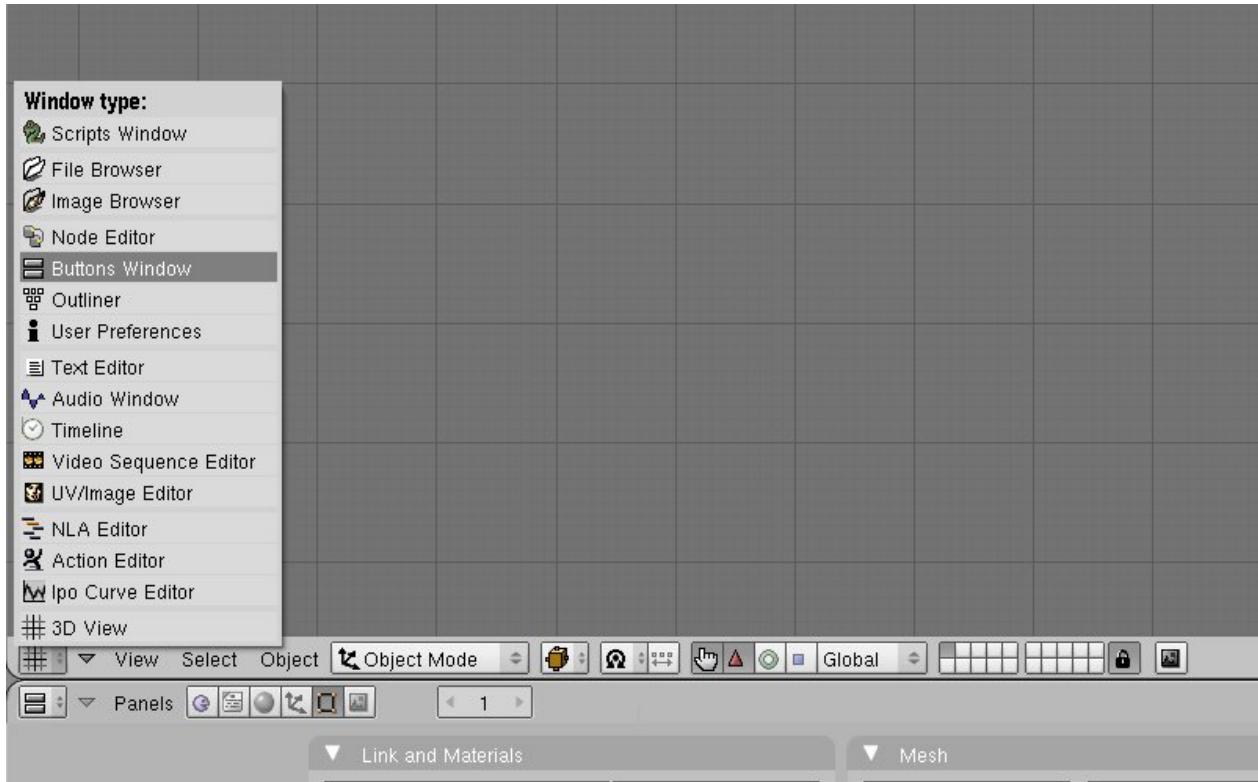
The header for the **Buttons Window** is highlighted in **blue**.

The **active window** is the window that will respond to what you type on the keyboard when you're using keyboard shortcuts. One, and only one, of the windows in Blender will be active at all times.

Making another window active is simple: simply move the mouse over one of the windows to make it active! Try changing the active window by moving your mouse rapidly between the **3D Viewport** and the **Buttons Window** now. You'll notice that the window's header lights up when it becomes active.

## Changing the Window Type

There are many window types other than the **3D View** and the **Buttons Window**, and you can easily switch any window to any other window type at any time.

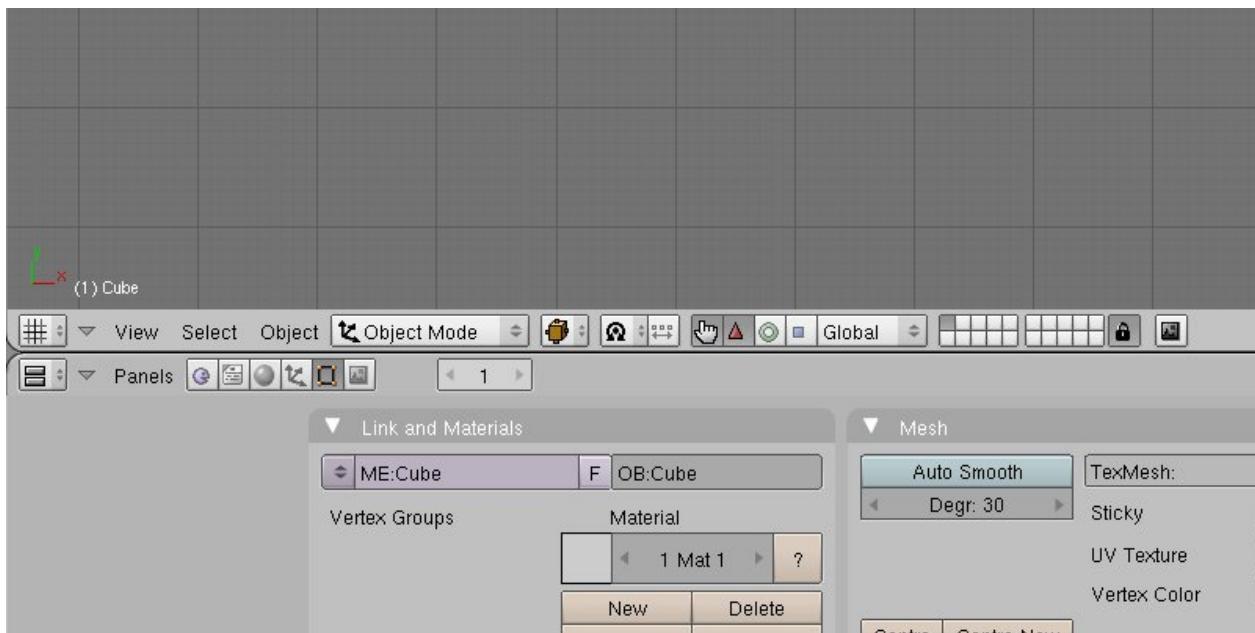


To change window types, simply click the leftmost button on the window header (highlighted here in red) and a menu will appear. Choose the **Buttons Window** from the list. You will now have two **Buttons Windows** on the screen!

At this point, having two **Buttons Windows** will not do us any good (but it will be useful later). Click the button again to change the window back to **3D View**.

## Resizing Windows

Resizing windows is easy<sup>[1]</sup>.



Hold your mouse over the border between the two windows that is indicated by the **red** box below, and the mouse pointer will change to up/down arrows (or a hand on Mac OS X).

Click (and hold) the border with the **LMB** (meaning **Left Mouse Button**) and drag up and down.

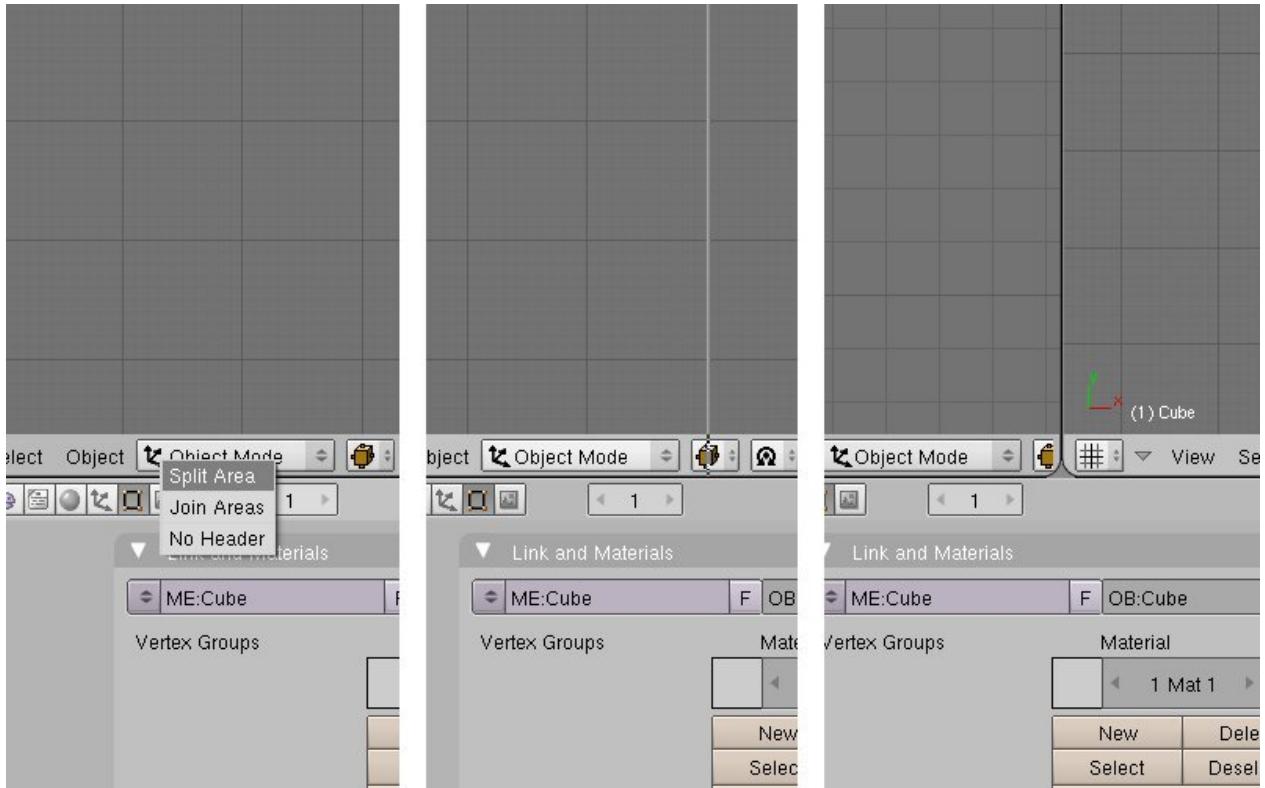
Once you've decided where to resize to, release the **LMB**.

You'll notice that as you increase the size of one window, you decrease the size of the other. Blender does not allow the windows to overlap, as they may in other programs. This is why Blender's interface is known as a **non-overlapping window interface....**

1. See related FAQ at bottom.

## Splitting Windows

Splitting windows is just as simple as resizing them, and will give you two windows of the same type.



Click the same border that you did last time, but this time with the **RMB** (**Right Mouse Button**), or if on a Mac, Option-click.

A menu will then appear. Choose **Split Area** from the menu. (Remember to hold down mouse button while selecting.)

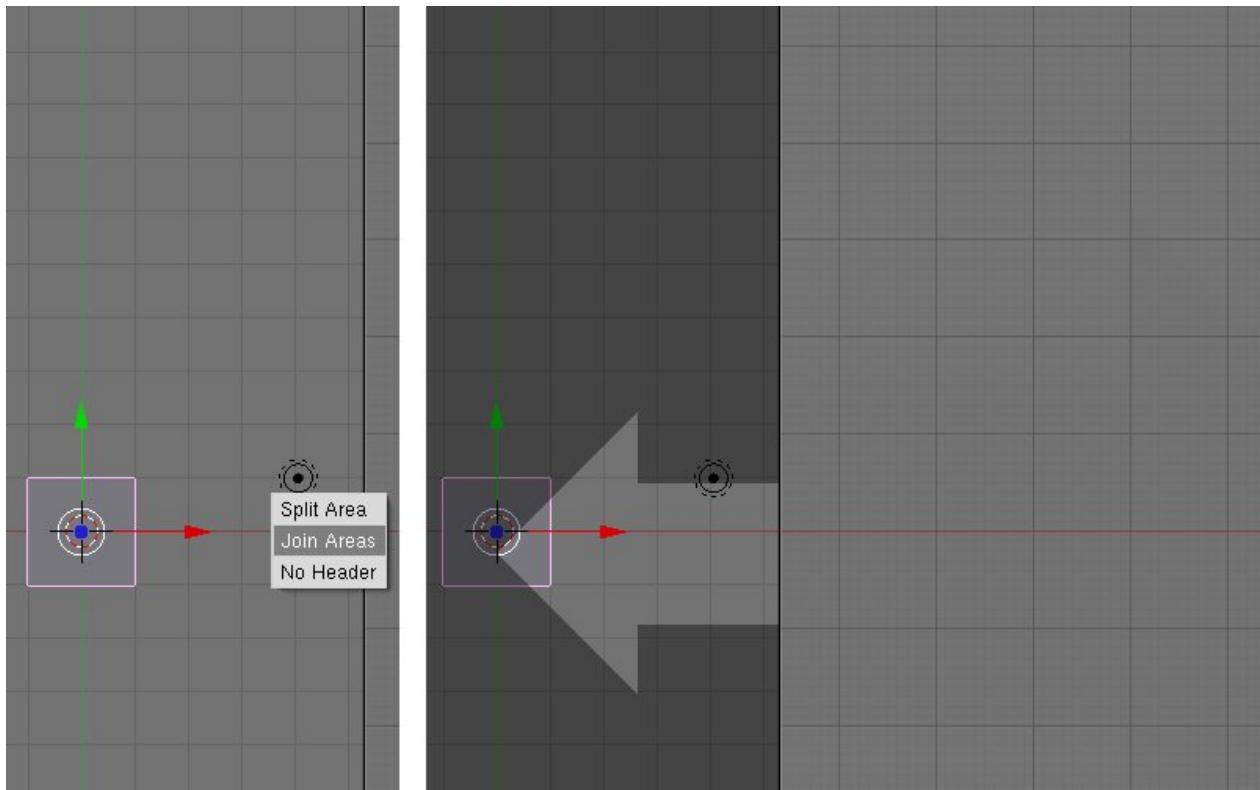
You'll see a preview line appear that will follow your cursor. Try dragging your mouse over both the 3D Viewport and the Buttons Window. You'll notice that this preview split line will follow you from window to window.

You can finalize where you want to split by simply pressing the **LMB**. You should then have two windows where before there was only one!

Splitting a window on a vertical division will give you two windows side-by-side vertically. Splitting on a horizontal division, as we have done, will give you two windows stacked horizontally. While in step 3, to switch between vertical and horizontal division, just use the **TAB** key. To exit without splitting a window, press the **ESC** key.

## Joining Windows

Rejoining two split windows is just as easy as splitting them. We'll rejoin the window we just split.



Click on the border you just created with the **RMB** (or Option-click for Mac) and choose **Join Areas** from the menu.

An arrow will then appear so that you can indicate in which direction you would like to join the windows. As you move your mouse from the left window to right window, the arrow will change directions.

Join the windows in either direction for now with the **LMB**.

Joining to the left means that the window on the left will be erased, while joining to the right means that the window on the right will be erased. Keep this in mind when joining different window types.

Note: If you right-click on a border and it doesn't give you the option to join, it is because that border touches more than 1 other window. You'll have to cover the window you want removed using a different handle.

## FAQ

1. ^ After loading a model in Blender, you can see only the edit window and the options window toolbar at top.

Trying to resize or split doesn't work. What happens?

— That occurs because Blender saves windows positions and preferences in the .blend model file. This one was saved while the edit window was maximized, although any other window would have had the same effect over user interface on loading if maximized at saving. To restore the window press **Ctrl+↑** or **Ctrl+↓** (toggle) and then, if necessary, press **Ctrl+←** or **Ctrl+→** to cycle between different window layouts.

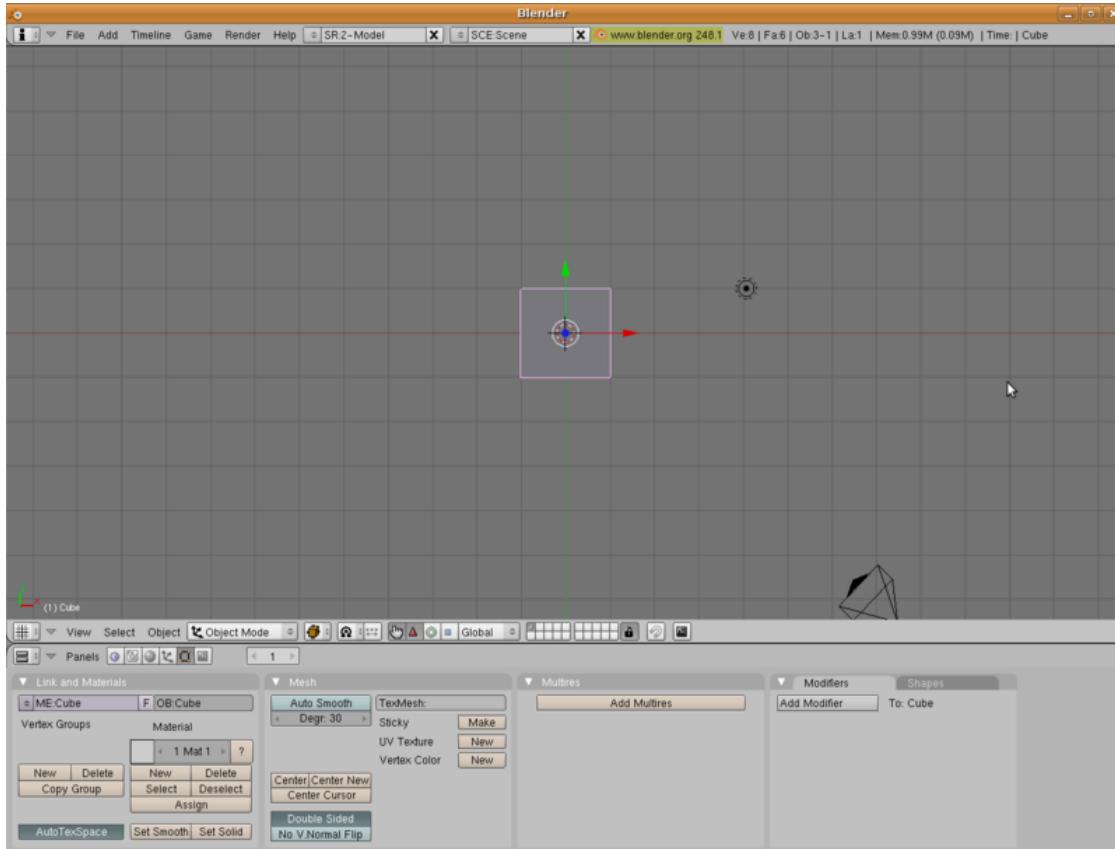
Splitting and Joining Windows Youtube Video [2]

## References

- [1] <http://wiki.blender.org/index.php/Reference/CL>
- [2] <http://www.youtube.com/watch?v=uYb1j8X-ulc>

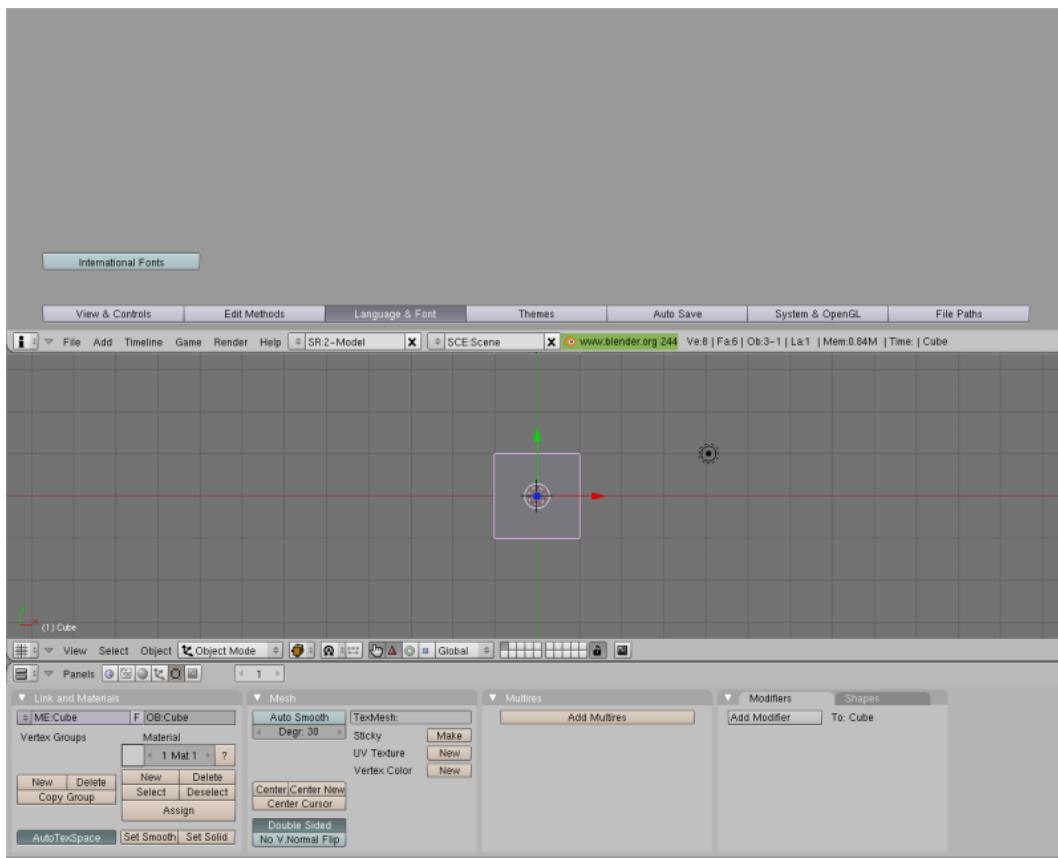
# Blender 3D: Noob to Pro/The User Preferences Window

## Showing the User Preferences Window



The first window we'll teach you in detail is the **User Preferences** window. This window is mostly hidden by default, because most of the time you won't need all of its contents. Only the **header** at the top of the main window (outlined in **red**) is visible here.

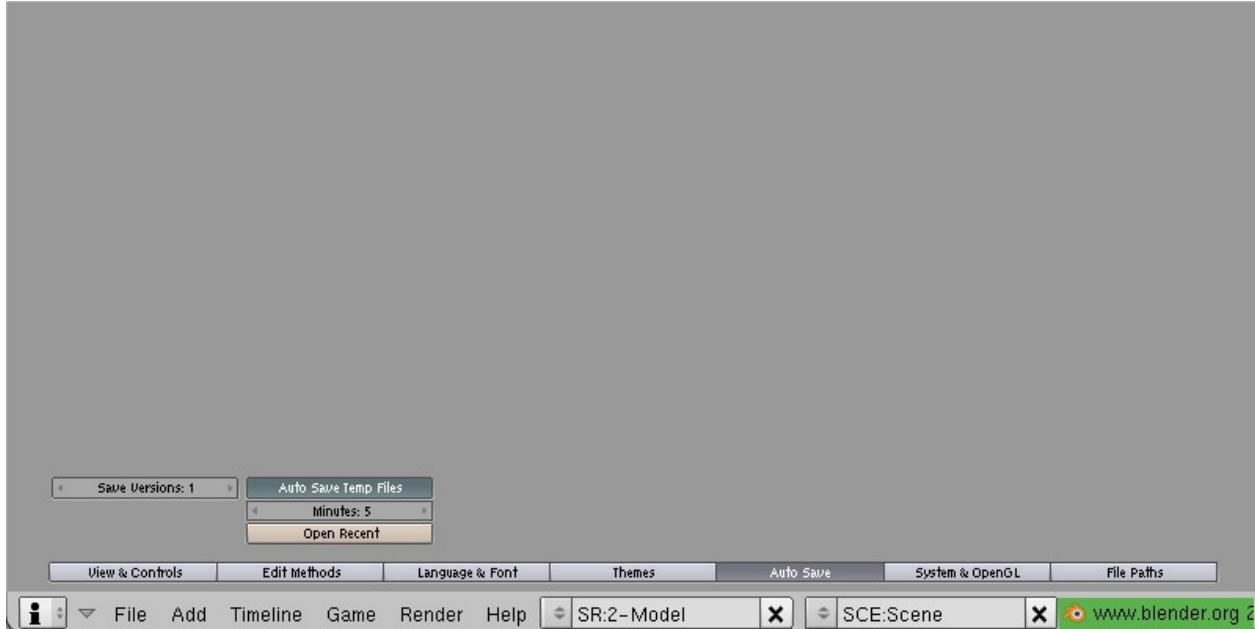
To make the **User Preferences** window visible, you need to resize it. You do that the same way that you resized the other windows in the previous tutorial. Click and hold with the **LMB** on the border along the top of the 3D Viewport, and drag.



Release the **LMB** to resize the windows. You should then have the entire **User Preferences** window visible!

## Configuring and Saving Your Preferences

### Setting Up Auto Save



The first thing you'll want to do is enable **Auto Save**. **Auto Save** will help you avoid the loss of important work in case Blender crashes, your power is cut off, etc.

To show the **Auto Save** options, click the **Auto Save** button (outlined here in **blue**).

Next, adjust the **Auto Save Settings**:

- **Auto Save Temp Files** - enabled by default, this turns the Auto Save system on and off. Leave it on.
- **Minutes** - how often should your work be auto-saved? (recommended 15 to 30)
- **Open Recent** - use this to recover your work if Blender crashes.
- **Save Versions** - how many versions of the file should Blender auto-save? (recommended 1 to 3 for space reasons)

### Adjusting The Theme

As with other programs with a graphical user interface, Blender has a way to customize or modify many aspects of the user interface itself. Each collection of customizations is called a *theme*. Clicking LMB on the "Themes" button will show a drop-down menu in the top left of the user preferences window which can be used to select different themes. Below this are buttons labeled "Add" and "Delete" that can be used to (surprise!) add and remove themes. Note that the delete button does not appear for the default theme. The other controls that appear in the user preferences window while the themes button is active are to adjust the parameters that are specific to the current theme.

This book presents all screen shots using the default theme. If you are new to Blender, you should continue to use the default theme as you progress through the book.

## More Undo Levels



The next thing that we want to do is take a look at the **Undo Settings**. By default, Blender will remember 32 of your previous actions in memory and allow you to undo all the way to that point. If you have a good computer with a lot of memory, you might want to increase that limit (up to 64), while if your computer is old and sickly, you may consider decreasing it to 10 or 20.

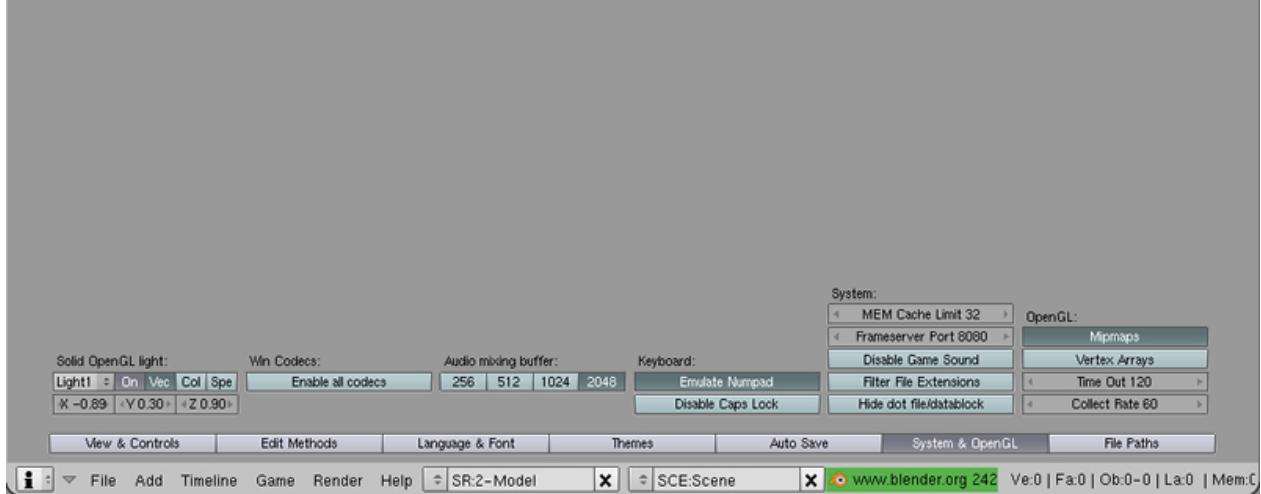
To show the **Undo Settings**, click the **Edit Methods** button.

The **Undo Settings** will then be visible. Use the **slider** to change the number of steps to any number (between 0 and 64). Keep in mind the amount of memory your system has, as we discussed above. Too many undo steps can slow your computer down, but the more you have, the easier it is to go back to a previous spot.

## Emulating the Numpad

By default the Numpad keys control the 3D viewport, while the normal numeric keys change the view layers. Emulating the Numpad, in effect, allows the user to control the 3D viewport using the normal numeric keys on the keyboard. This replaces the default function of the keys, but does not affect the Numpad.

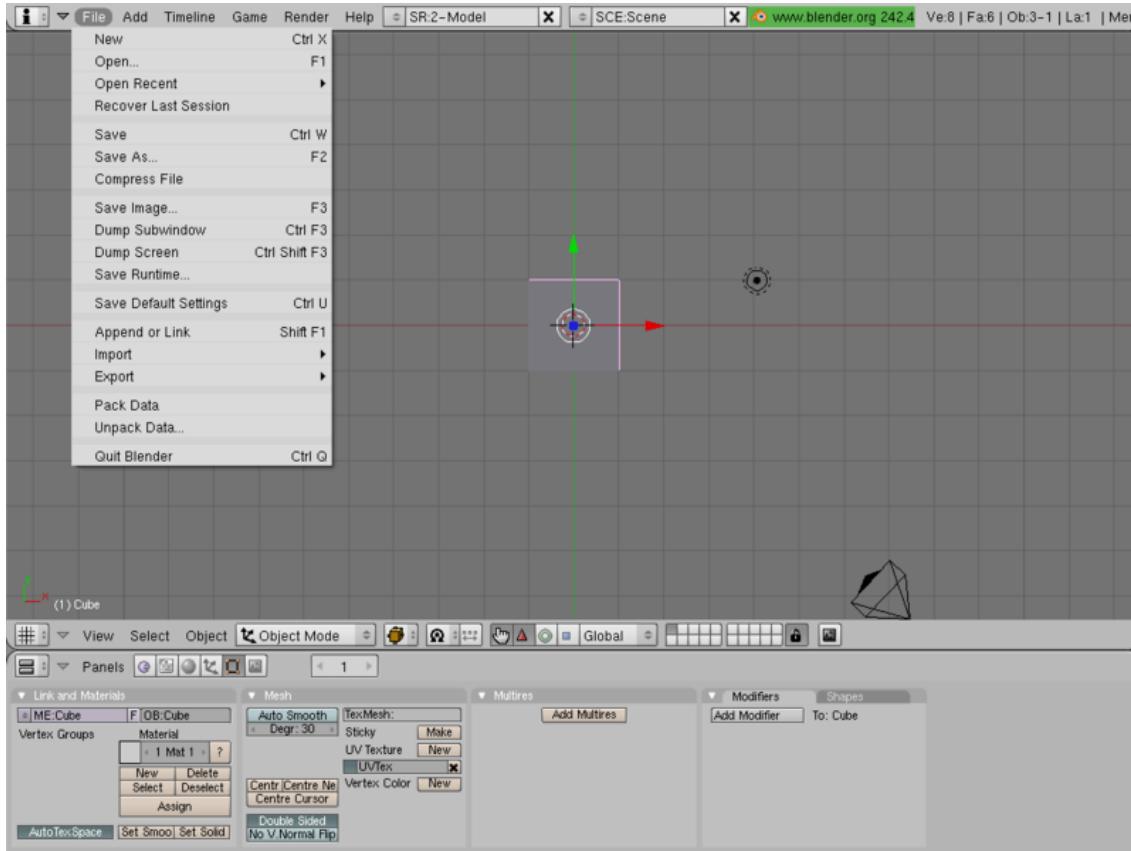
This is useful for computers/laptops where there is no physical Numpad available.



Navigate to the 'System & OpenGL' tab (outlined in blue) to open the menu.

Click on the 'Emulate Numpad' button (outlined in red) to enable it. If it is already in this darker shade of green, it is already enabled.

## Saving Your Preferences



Finally, you may want to keep your preferences as the default preferences whenever Blender starts up, unless you prefer to set them up everytime.

Expand the **File** menu. To do so, click on the **File** button on the User Preferences window.

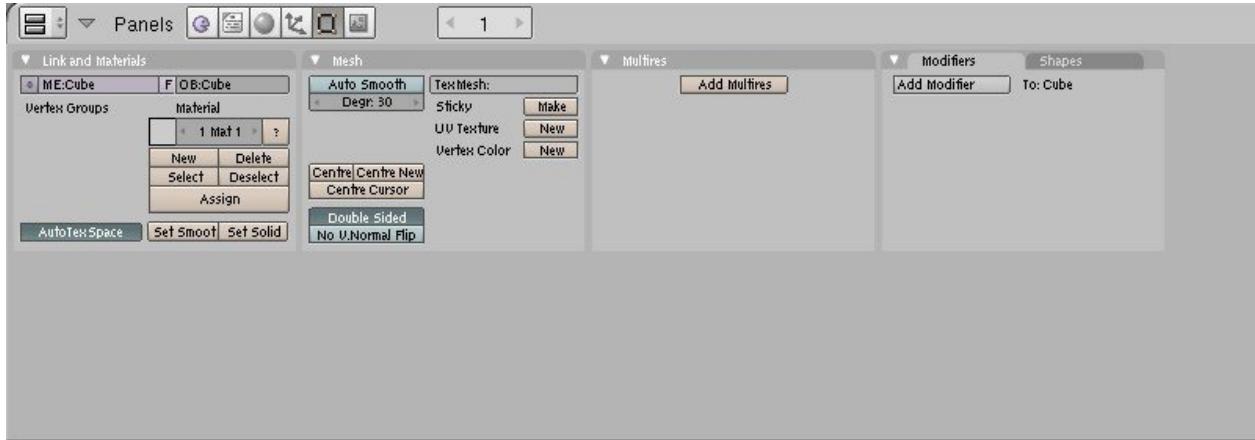
Click on the **Save Default Settings** button in the **File** menu to save your preferences.

Alternatively, you may press the **Ctrl+U** key combination to achieve the same purpose.

# Blender 3D: Noob to Pro/The Buttons Window

---

## What's with all the buttons?!



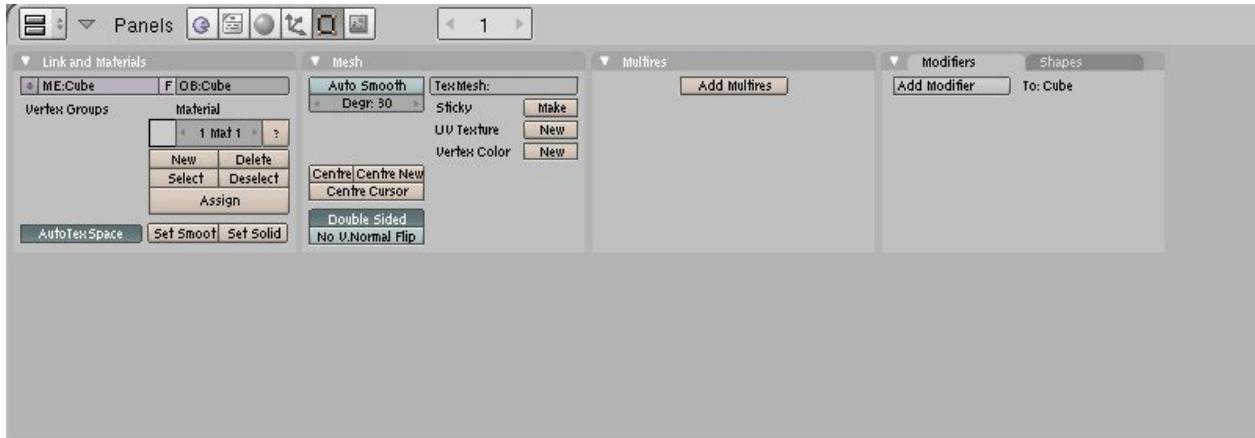
The Buttons Window is one of the most powerful tools that Blender has. When you have objects selected in the 3D viewport, there will be a number of operations you can perform on the objects by pressing the buttons in the Buttons Window. For example, suppose you have modeled a person. People have different skin colors, eye colors, hair colors, and more, so you will create a material to make the person appear as you would like him or her 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. The buttons outlined here in red allow you to change the type of the buttons (button group) that are currently being displayed. Click several buttons and notice that the entire Buttons Window changes when you choose a different type.

You can only display one type of "buttons" at a time in your Buttons Window. This may seem like a limiting factor, but keep in mind that you can create as many different Buttons Windows with your interface as you would like and they can all be displaying different buttons, giving you quick access to anything you need to do.

## Mini-Windows



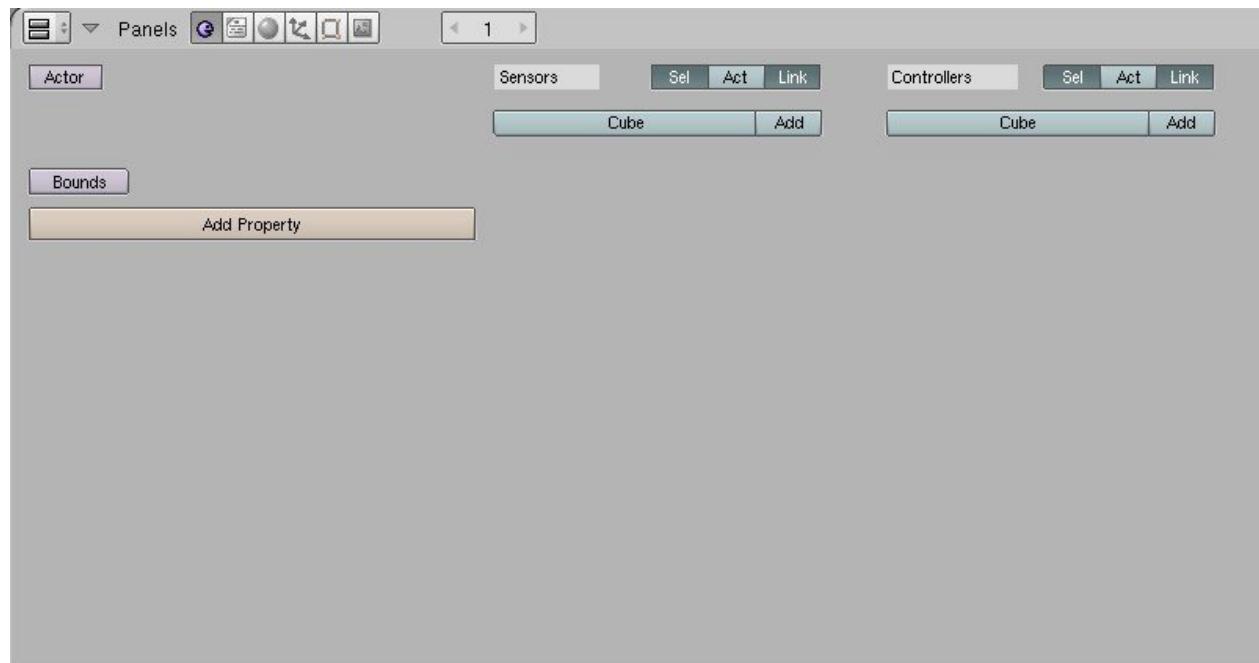
You'll notice that within the Buttons Window there are **Mini-Windows** (i.e., panels) such as those outlined here. Many manipulations can be performed with these windows, but for now we will only teach you the most important two.

The first thing you can do is click the top of the Mini-Windows with the **LMB** and drag them around. Try it! Also, try dragging and dropping them onto other Mini-Windows to make them join together (as tabs). You can separate the

tabs into individual windows again by dragging the active tab over to an available space. Also if there are too many mini windows and you can't see a certain one, use the **MMB** (or the mouse wheel, **SCROLL**) to pan across until the desired window is visible or use **ALT+LMB** and drag to move the mini windows within the Buttons Window.

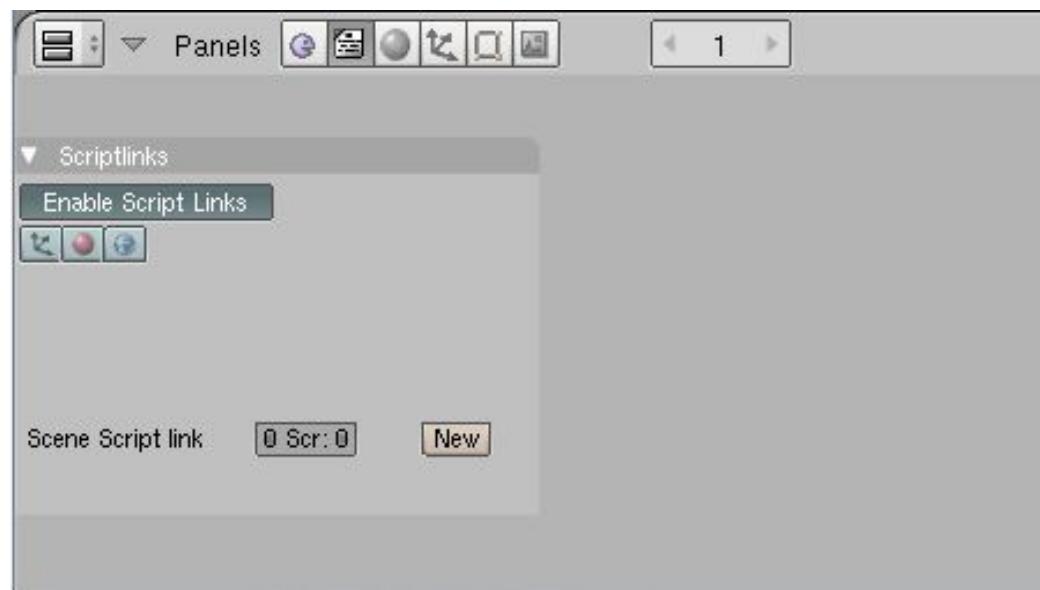
## Button Types

### Logic Buttons



Here we have selected the first "buttons" type, the Logic Buttons. We won't be using these buttons much for a long, long time. They are for use with the Game Engine. Pressing **F4** will activate this panel.

### Script Buttons



These permit connecting various events in Blender to scripts, thus extending Blender's functionality. Scripting itself is an advanced topic and the scripts are necessary for good-quality animations.

## Shading Buttons

The Shading button set allows you to apply and manipulate colors and textures on your objects, and control lights and world settings. When you press this button (or the **F5 key**) you will see five additional buttons appear. These are for lights, general material settings, textures, radiosity, and world settings (handy for giving your renders a quick background). Pressing the **F5 key** will cycle through these buttons.

## Object Buttons

You can press **F7** to cause the Object Buttons to appear. It should be noted that these are not the same buttons that appear when you choose Object Mode in the 3D Window. Some tutorials may refer to pressing the **F7 key** to change to **Object Mode**, and some will say you should press the **Tab key** to change to **Object mode**.

The **Tab key** changes from **Edit Mode** to **Object Mode** in the 3D Window, and **F7** changes the Buttons Window to show the **Object Buttons**.

## Edit Buttons

These are buttons used to edit objects in edit mode. You can press **F9** to get the edit buttons. To get to edit mode (in the 3D View window) press **TAB**.

## Scene Buttons

Basically these are for rendering (taking pictures) and animating (making movies). You can press **F10** to get the scene buttons. We'll get back to these later.

# Blender 3D: Noob to Pro/The 3D Viewport Window

Blender's **3D Viewport Window** (3d Viewport) gives you total control of how you visualize your world. You'll spend most of your time in this window, so here are a few things to know about the 3d Viewport.

## Rotating

Here you'll be able to fly around your 3D scene, rotating the planes as you see fit. You'll see that the default object is actually a cube, and half of it lies above the X-Y plane, and half below it.

Make the 3D Viewport active by placing the mouse pointer anywhere inside it.

- To free-form rotate (*any way*), while holding down the **MMB**, move the mouse
- To rotate around a vertical axis (*sideways*), leaving objects' vertical orientation unaltered, use **CTRL+ALT+SCROLL**, or using the keyboard, **NUM4** and **NUM6**
- To rotate around a horizontal axis (*upward*), leaving objects' horizontal orientation unaltered, use **SHIFT+ALT+SCROLL**, or using the keyboard, **NUM8** and **NUM2**

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 Solid Mode. Pressing **ZKEY** (yes, on your keyboard, the 'Z' key) will toggle back and forth between Solid Mode and Wireframe Mode. Pressing **NUM5** while NUM LOCK is on 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 the following three objects:

*Camera, Lamp, and Cube.*

We'll get into more in depth details about these later.

Object Icon	Name	Description
	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 pressing the <b>NUM0</b> key. (Remember the <b>0KEY</b> is different.) You may need to make sure NUM LOCK is on on your keyboard. To switch out of the camera view, drag the <b>MMB</b> . Or press <b>NUM0</b> and then <b>SHIFT+F</b> to enter "camera fly mode" to position the camera interactively from the viewport using the mouse. Press <b>LMB</b> to finish positioning the camera.
	Lamp	A lamp is simply a light source. It will not be rendered, but the light it provides to the scene will be rendered.
	Cube	This object will be rendered. The camera should be pointing at the cube so that you will see it at render time, if the camera is not pointing at the cube, or if it is somehow partially out of frame, the picture will reflect this.

Here is a table of some simple key combinations that will result in a *perfect view*.



### *Perfect View* key combinations

Key Combo	View	Key Combo	View
NUM7	top	CTRL+NUM7	bottom
NUM1	front	CTRL+NUM1	back
NUM3	right side	CTRL+NUM3	left side

The object the viewport orbits around (the object you see) 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.

[*NOTE: selecting an object with the RMB will only work if your viewport is set to 'Object Mode.' Press the TAB key to toggle between 'Edit Mode' and 'Object Mode.'*]

In Blender there is a big difference between the number keys on your numberpad (numpad) 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 have been changing the layer that you are viewing instead, press the **~key** (tilde key) to return to viewing all the layers, or press the **1KEY** to get back to viewing layer 1 which should have been originally active. The **1KEY** through **0KEY** and **ALT+1KEY** through **ALT+0KEY** switch layers.

## Panning

To **pan** is to move the camera on its X axis or Y axis. This results in the user being able to *view more*, or more aptly, to *view something else*. Think of a side-scrolling video game, such as any classic *Mario* or *Sonic*, the effect that your character's avatar always stays on the viewable screen while giving you the illusion it's running off the screen is because the character runs at the same pace the camera pans. This is evident in the background's continuous motion relative to the static avatar and camera which remain relatively synchronised.

To pan in Blender press **SHIFT+MMB**. Make sure to press and hold shift before the MMB, or your view will rotate instead. If you have a scroll wheel you can use **SHIFT+Scroll** to pan up and down, and **CTRL+Scroll** to pan left and right. If you do not have a scroll wheel or trackpad or a middle mouse button, press **SHIFT+ALT+LMB** to pan. You also have a choice of keyboard alternatives:

CTRL+NUM8	NUM:	Up
CTRL+NUM4	Left	Right
CTRL+NUM2	Down	

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

## Zooming

Blender offers you several ways to zoom in and out:

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

## Placing the 3D cursor

"I found that I would select the cube when left clicking on it in object mode, if the "**Use 3d transform manipulator**" button was enabled. To toggle this off, you click on the **gray pointing hand** in the 3d panel header, or (Ctrl Space)."

"When you want the cursor back into the cube, just select the camera with **RMB**, put the cursor into the cube following the steps above, and re-select the cube with **RMB**."

"I've discovered it helps a lot if you are in Object Mode and not in Edit Mode. I wrote the following before discovering this: *The problem with this exercise, for me, is that left clicking on the cube selects the cube instead of moving the 3d cursor. If I click on the cube outside of its central white circle I can get the cursor to move there, but only to outside of this white circle, and even then this only works sometimes.*"

"I failed at this until I had zoomed in close enough to the cube. When I was too far zoomed out I kept selecting the cube rather than creating an edit point."

"I had the same problem and found it was because the cube was selected. I made sure I was in object mode, right clicked on the camera to select the camera instead of the cube, and I could then position the edit point in the cube. However, doing this messed up the next part of the tutorial because you cannot switch into edit mode with the camera selected! Perhaps the suggestion of trying to put the 3D cursor in the cube should be dropped as it raises too many questions at this stage."

"You can deselect all by pressing the **AKEY** or the select button in the 3D View."

"Use wireframe mode works better to get the cursor in."

"To get it back in the cube: 1) Make sure you're in object mode. 2) Select the cube. 3) Object > Snap > Cursor to selection (cursor refers to the 3D cursor here) so it puts it right in the middle of the cube."

"I think it's an essential point to note that *in order to place the cursor inside the cube, the cube must NOT be selected*. AKEY was probably the best way to deselect the object."

"I would find an "undo" command of great use while learning and experimenting with the various keys. Sometime you do something you didn't intend to do. It would be nice to undo that undesired effect."

"The "undo" command exists, but it doesn't seem to be in any menu or key. On Macs you can undo using Command+ZKEY, on other systems I suppose that CTRL+ZKEY will do the thing."

"If I remember correctly, undo history gets cleared when you switch between object and edit mode."

"I wasted a lot of time here. *Thank you* to the reader who suggested (on the 3D view header) Object > Snap > Cursor to selection. It was the only thing that worked to get the cursor visible again and placed where clicked."

"You can set the number of steps the undo command can do as mentioned in earlier pages."

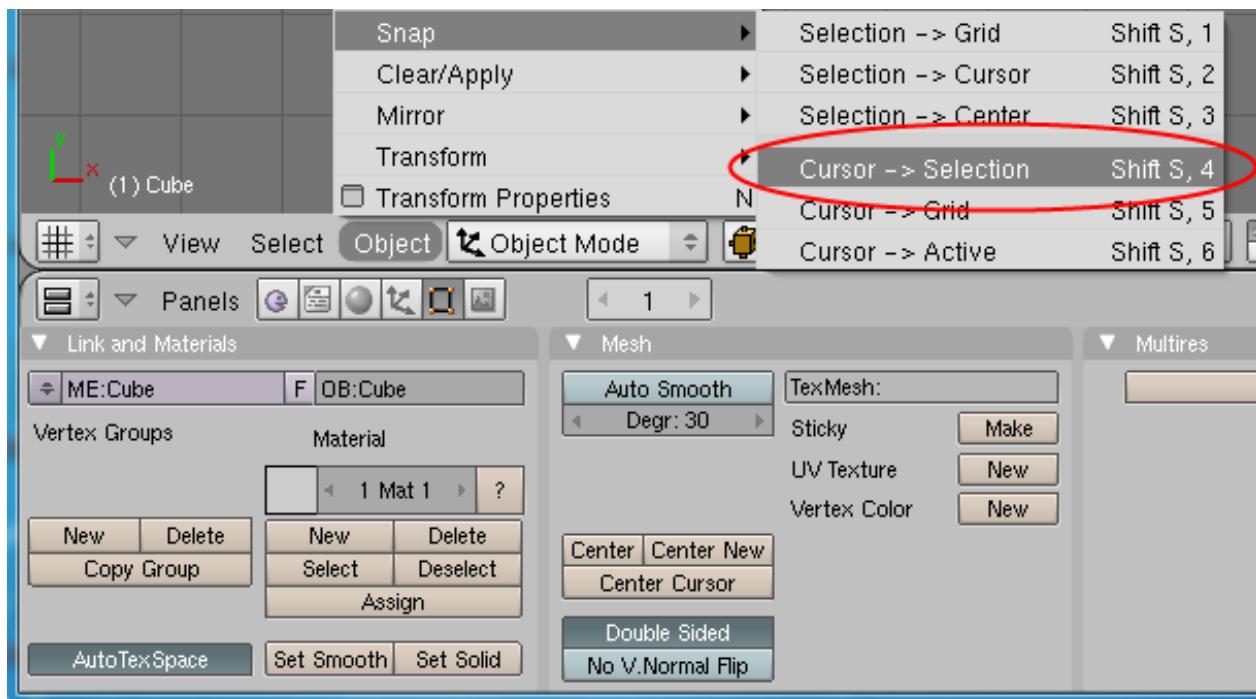
"I missed the point of the exercise first time around. You can't set a 3D point on a 2D screen without technique. Orthographic views are crucial. I am just learning, but take that, at least, away from it."

As with an ordinary text cursor (the vertical line that indicates where the text you type will appear), the 3D cursor is the insertion point for new objects. It is represented by a red and white circle which indicates the location of your editing point in the 3D environment.

Try clicking the **LMB** in empty space to the right of the cube. The red and white circle (the **3D cursor**) moves to where you clicked. Orbit the view by clicking the **MMB** and moving the mouse. Notice that the 3D cursor marks a point in 3D space. "So I can move the 3D cursor, but what if I want to put it back in its original spot?" you may be asking. To do that, just press **SHIFT+C** and the cursor will jump back in place.

In any given view of the 3D environment, the set of possible 3D points where you can place the cursor is determined through and limited by what is viewable through your screen. If you were to move the 3D cursor again while looking at your screen straight-on, the cursor would be placed at an unspecified distance beyond (or "behind") the screen, regardless of the view or where you clicked. This brings us to a problem common to all 3D design programs: "How do we work in a virtual 3D environment through a 2D screen?"

To illustrate, try to put the 3D cursor inside the camera (the pyramid-shaped object), then try to put the cursor back in the cube. Be sure to view the scene from different angles to make sure you have succeeded in placing the cursor inside each. If you try and put the cursor back in the cube, you select the cube instead unintentionally. So, while the cube is still selected, try going to the 3D view header and clicking on the menu options *Object > Snap > Cursor to selection*; this will snap the cursor to the cube you just accidentally selected. But try your best just to move it in the cube only using the mouse to place it, just for experience's sake.



Are you finding this difficult? That is because we need to clearly specify the 3 coordinates for the desired cursor location in the 3D environment. Try this: Make sure the 3D View is in "Orthographic" mode by clicking View > Orthographic (or pressing **NUM5**). Press **NUM7** to get to the front view and click again on the desired location to position the cursor. With these two clicks, Blender will have all 3 coordinates of the cursor position and you will have placed the cursor exactly.

## Layers

In the 3d viewport window, both in edit and object modes, everything you create is assigned to a visibility layer. This system has several uses:

1. Divides up different elements of a scene, so you can put scenery, characters, particles and lights all in different layers. They can then be viewed separately or in various combinations to simplify your screen.
2. When rendering, only the currently visible layers will be included. You can use this to render your scene in separate bits to review how they look.
3. Lights can be set to only illuminate objects that are in the same layer as they are, giving you more control over them. (This can also be done with grouping, but layers are faster to use at this stage).

To control layer visibility, the number keys on a standard keyboard will switch you to viewing the layers numbered 1-9 and 0 (0 being the rightmost layer). Holding **ALT** while using the keyboard numbers will give you access to the second row of layers.

Alternatively, there is a grid of buttons in the 3d View header that does the same thing.

Note to azerty users : standard number keys are &é"(-è\_çà keys (do not use **SHIFT** unless you want to toggle visibility as explained below).



Holding **SHIFT** while selecting a layer (by keyboard or mouse) will, instead of making only that layer visible, toggle the visibility. You can use this to select combinations, or to disable individual layers from your current view.

To select all layers at once, press the [ for UK keyboards, ` for US, ö for Swedish and German, æ for Danish, ù for AZERTY, ø for Norwegian, ò for Italian`] key on your keyboard. Holding **SHIFT** and pressing the key will return you to the last set-up you had before making them all visible.

An object you create will automatically be assigned to the layer you are currently viewing, if only one is selected, or the last layer you added to your selection. To move a selected object to a different layer, press the **MKEY** and select the new layer from the pop-up box.

## Exercise (3D space in 2D output)

Follow these simple steps to get a feel for a 3D representation of space in a 2D output device (your monitor):

1. Change to "Object mode" using the pull down option in the 3D viewport's window header bar. Or, hit **TAB** to toggle between "Object mode" and "Edit mode".
2. Disable the "Use 3D transform manipulator" option by using the icon located on the 3D viewport's window header (shaped like a pointing hand). Or, hit **CTRL+SPACE** to toggle.
3. Hit **NUM7** to change to top view. This can also be accomplished through the view menu.
4. Click a point somewhere between the cube and camera using the **LMB**.
5. Choose a different view by hitting **NUM1** (front view), or **NUM3** (side view). [1]
6. Click between the cube and camera with **LMB** again.
7. Rotate the view around to see how it turned out.

For the part where you are to get your cursor into the middle of the cube, just follow steps 3 through 6 again. Except this time, you'll of course be **LMB** clicking inside the cube, instead of between camera and cube, during step 4 and

step 6.

### Notes

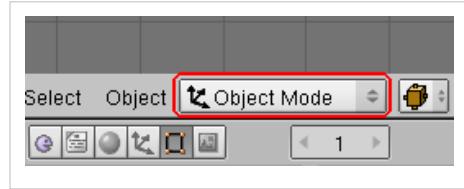
^ Because we are working in a 3D space you'll need to have two different views that intersect each other. For instance, viewing from top and then from bottom wouldn't be of much help in specifying the height or depth of the 3D cursor. These views can also be selected through the view menu.

## Adding and Deleting Objects

Make sure you are in Object Mode. If not, press **TAB**. (When an object is selected in edit mode, the **TAB** key switches between the edit and object modes. If you are in another mode, **TAB** toggles between that mode and the edit mode.) A status bar at the top-right of the user preferences window will indicate the current mode by displaying 'Ob' or 'Ed' depending on the currently toggled mode. Another way to check which view you are in is to check the bottom of the 3D view.

Also, remember to reactivate the '3D Transform Manipulator' if it's still toggled off from the previous step.

Make sure you have your cursor in the center of the cube. See the previous section (in the reader's notes) if you don't know how to do this.



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" (or "Erase All").

The reason for having your cursor in the center of the cube is that any object you add to the scene will be located where your cursor is.

To add an object, use the **Add** menu located in the menubar above your 3D View window, or press the **Spacebar** to access the same menu. Why not add a monkey? Choose **Add > Mesh > Monkey**. [If you prefer the monkey to be facing frontwards, make sure to be in FRONT view (**NUM1**) before adding the mesh - note: in Blender 2.48a, it's complicated. If Blender is in Object Mode, the monkey is always facing up. If it's in edit mode then the direction the monkey faces depends on the view (top, side or front)]

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 the **ZKEY** to toggle the 3D Viewport between solid and wireframe modes. Zoom in and out for a closer look (**SCROLL**, **NUM+**, **CTRL+MMB**, or **ALT+CTRL+LMB**).

## Non-standard equipment

*Further information: Blender 3D: Noob to Pro/Non-standard equipment*

### Mice lacking MMB

For simply rotating around the object, enable the "Emulate 3 Button Mouse" option in the View & Control Preferences, and press **Alt+LMB** and drag.

### Tablet PCs

In the Viewport, holding the **ALT** key while dragging your pen around will achieve the same effect as **MMB**.

## References

[1] [http://en.wikipedia.org/wiki/Blender\\_3d%3A\\_noob\\_to\\_pro%2Fthe\\_3d\\_viewport\\_window#endnote\\_pov](http://en.wikipedia.org/wiki/Blender_3d%3A_noob_to_pro%2Fthe_3d_viewport_window#endnote_pov)

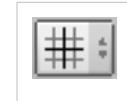
# Blender 3D: Noob to Pro/Other Windows

---

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, you only need to know one 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.



# Unit Two: Creation and Development

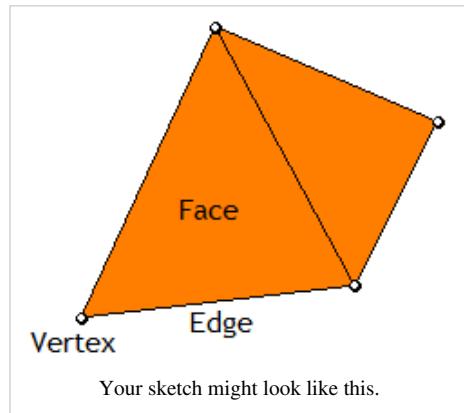
## Blender 3D: Noob to Pro/Learn to Model

The most fundamental part of 3D development is modeling, because this is where you create content, or 'models'. Creating 3D models is fun and sometimes challenging. To start with, we'll go over a concept called "mesh modeling".

### Mesh Modeling

Mesh modeling is the most common type of modeling in all of Blender-dom. A **mesh** is simply a collection of three core components; vertices, edges, and faces, that define a three dimensional object. This exercise will further help explain these components, and how they relate to mesh modeling.

1. Get a piece of paper and a pen or pencil.
2. Draw three dots that are no more than 2.5 cm (about an inch) apart from each other
3. Each one of these dots is called a **vertex**. (The plural of vertex is "vertices")
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 modeling 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. Blender also supports faces with four vertices (called quads), but faces with five or more (so-called N-gons) cannot be created.

*Try examining a 3D video game character for a while. 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.

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

# Blender 3D: Noob to Pro/Introducing Meshes

## Welcome!

First of all, we will learn what a model is. Make sure you first complete the "Thinking in 3-D" and "Beginner Tutorials" tutorial chapters before this one--you will need it here.

**This tutorial was made with Blender 2.48, at a later date than many beginner tutorials here, so it is more up-to-date but may refer to methods not existent in earlier versions of Blender. Keep this fact in mind as you learn.**

## The View

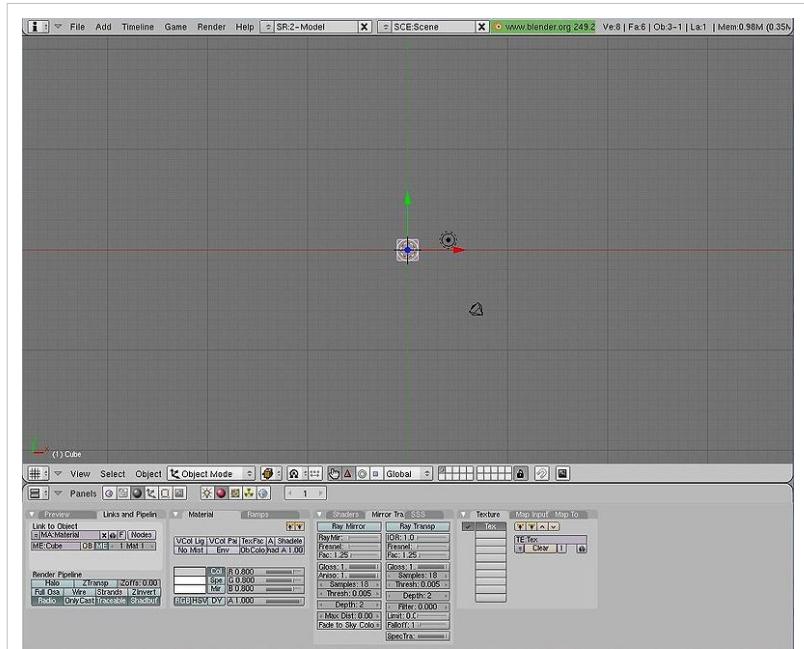
So, can you tell me what you're looking at? Of course not! How can you be expected to model if you have no clue how the software performs such a task!? So let's break everything down step-by-step. Identify each object as described.

### Identifying the 3D-View

- **Red-White 3D Cursor**

(*Crosshairs*): This is your first important artifact. This object is the Blender-user cursor. It is where objects are spawned when you "Add" an object. Also, it will be used with all references Blender uses for actions relating to "Cursor". If you ever see anything referring to "Cursor" this is the object being referenced.

- In the Blender-World, it is non-interactive/non-existent -- it will not interact with any objects, emit anything, or show up in Renders.
- This artifact can be placed (by default settings) by left-clicking anywhere on the 3D-view window. It will move from whichever (X,Y,Z) coordinates it had initially, to the (X,Y,Z) coordinates your mouse cursor is pointing to. Now when you click an area on the 3D view in an attempt to move the cursor, remember the screen is only 2D, so you will only be able to move the cursor on the 2D plane that you have in viewing the scene. This means that the cursor will retain its "Window Z"" coordinate--you will only move it in 2-coordinates. If you want to change that third coordinate, remember to change views (e.g. move from top-view to side-view).
- **Camera (displayed as a black, lopsided pyramid in above picture):** This is the eye that Blender uses to "View" the world. It is not discussed in this tutorial.
- In the Blender-World, it is limited-interactive/non-existent -- it will not initially interact with objects, emit anything, or show up in Renders.
- **Lamp (displayed as a yellow orb with a yellow circle around it):** This is the default light that Blender uses to illuminate mesh objects. It will be discussed in very little detail in this section.



Your Blender will look different as this is a custom color scheme.

- In the Blender-World, it is limited-interactive/non-existent -- it will not initially interact with objects or show up in Renders, but does emit "light" to make objects visible/"lit" within the distance set.
- **Mesh (displayed as a gray box with a red dot at the center):** This is the "default cube", the initial model Blender loads when you first load Blender. It will be discussed in this tutorial.
- In the Blender-World, it is interactive/existent -- it will interact with objects, show up in Renders, and be registered in the Blender universe in many different ways.

## The View's Options

So now you're familiar with the view, next let's look at your different options. In this view, the Camera, Lamp, and Mesh are selectable. Selecting them will cause an outline to appear around the object or change its color altogether. You just have to remember which objects do what when you select them.

Each object has its own set of options when you select it. To select an object, move the mouse cursor over the object in the 3D-View and **Right-Click**. It should then be outlined with a pink color.

The 3D-View has a lower border which houses its own set of options. It is visible directly below the 3D View



in the snapshot above as well as in the thumb to the right. Identify the given overview of each option (each option is listed from left-to-right):

- **3D-View (only described by a drop-down menu with a grid in it):** This option is in the border of every window. The drop-down menu changes the category in which this window is working. The Grid in the menu means the window will be displaying 3D View. The other selections will be discussed in later tutorials.

*Of the following options, the next 3 are only visible as long as the little gray arrow next to the "Current Window Type" is pointing down (if pointing right, click it to change)*

- **View:** This option will yield many options that change how you are viewing the scene in the 3D view. Do not mess with it much now or you may not be able to get out of an odd view and might even have to restart Blender to restore view.
- **Select:** This option will yield lots of options which you can use to change which objects are selected in the 3D view. Many options will be discussed in later tutorials.
- **Object:** This option will display several options which you can use to change certain properties of the object which is selected. It is visible if the next drop-down menu is in "Object Mode" and will change with respect to the Mode selected in that drop-down. Many options will be discussed in later tutorials.
- **Mode (Menu):** Depending on the selected type of object, this menu will yield anywhere from 1 to  $n$  options. When a mesh is selected, there are 5 options to start with in this menu. The two options you will be taught in this section are "Object Mode" and "Edit Mode".
- **Draw-Type Menu (shown only as a tan box next to the "Mode" Menu):** This option displays the objects in the 3D Viewport in different ways. For example, it can show them shaded or textured. (Note: called the "Viewport Shading" button in version 249.2)
- **Pivot Type (shown only as a menu directly right of the Draw-Type Menu, usually with assorted dot symbols):** Changes where the rotation/scale pivot point will be located for the selected object(s) when transformed.
- **3D Transform Manipulator (shown as a button with a hand pointing, which also has a drop-down menu connected to the right and, when depressed, will yield 3 more buttons): Very Important--**The triangular button will create 3 colored arrows at the pivot-point of the selections. These colored arrows can be selected to change the location of the the selected objects. The circle button creates 3 colored rings with an outer "window-facing" ring which can be selected to rotate selected objects at the pivot point. The block button will change the scale (size) of the selections at the pivot point along the 3 colored blocks presented in the 3D view. In each case, you can **LMB-drag** the guide arrows, circles or lines to affect the transformation. The menu next to that is used to

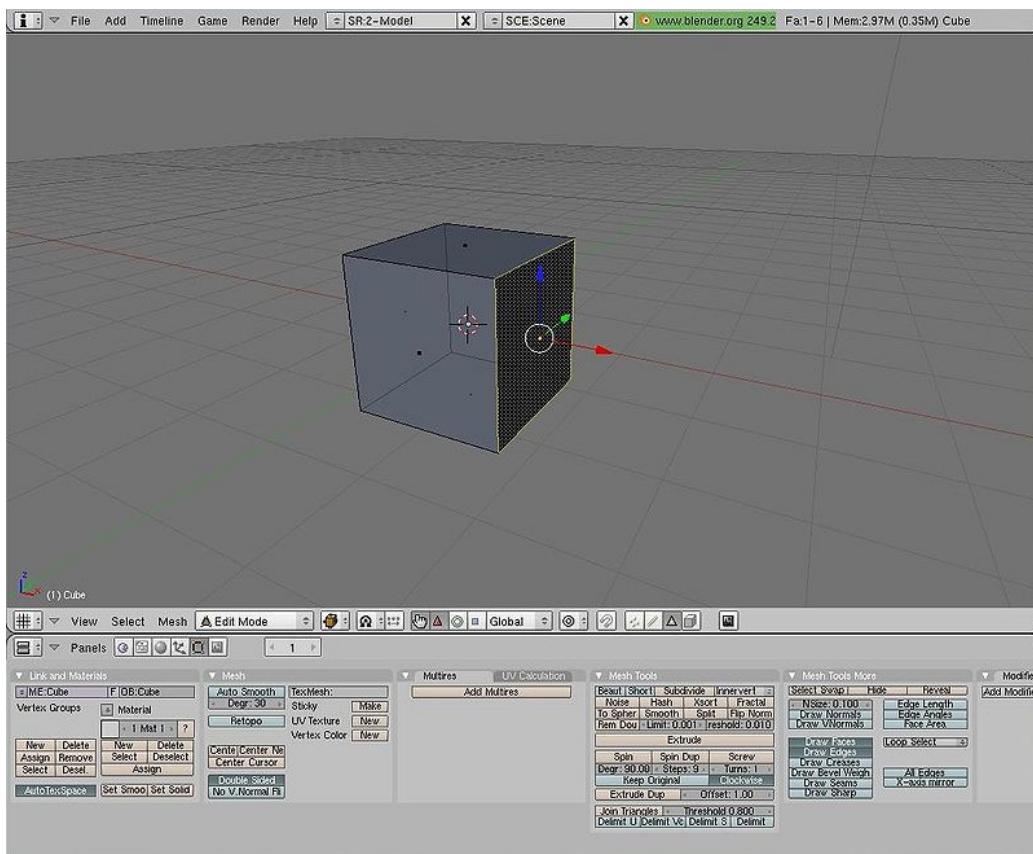
determine on which axes these transformations will take place.

- **Layer Buttons (shown as rows of buttons to the right of 3D Transform Manipulator):** Changes the layer you are viewing (when objects are created they are set on a layer).
- **Magnet Button (shown as a button with a magnet picture on it):** When selected will snap transformations of object(s) to selected integral values instead of "free" (multi-decimal) values.
- **Render Image Button (shown as a button with a scene picture on it):** Will take a picture of the 3D scene as you currently see it from your view.

Oh yes, it is a lot to review, but don't fret--all we are doing now is working with modeling an object. We will only really utilize the transforming aspects and, in the buttons window below, the Editing View.

## Edit View

So, first you will want to deselect everything, so you will not accidentally edit something you don't want to. Hit **A-key** to deselect everything if anything is selected. If nothing was selected, then everything will be selected, in which case simply hit **A-key** again. Next, zoom in and change the camera angle to your desire, preferably one in which the cube is in a 3D view and is almost encompassing the whole screen. Next, switch to the Edit Mode either by hitting **Tab** or by selecting it from the "----- Mode" Drop-down. Now the cube will have a black outline and new options will be available.



If you view the bottom border in this example, you will clearly see 4 new buttons to the right of the "magnet" button. As you saw in the previous section, a mesh has "Vertices", "Edges", and "Faces". Each button corresponds with the button's picture. In Edit Mode, you may manipulate and transform these 3 types as you wish using the transformation buttons. Give it a shot. That's right, just start out free-wheeling it for now, making sure to select with the mouse **R-Click** (right-click). Now **RMB-drag** to move the selected feature around. (You can let go of the **RMB** once you start the movement action. Finally, **LMB-click** to choose to keep the change, or **RMB-Click** to ignore the change. Change the selection types by pressing the "Vertices", "Edges", or "Faces" buttons in the bottom border area so you can choose which parts of the cube you want to select. Also you can select with these different ways, so practice with

them:

- **Shift + R-Click** -- Can select or deselect multiple objects, vertices, edges, or faces. Just hold and click what you want to.
- **A-key** -- Select/Deselect all objects, vertices, edges, or faces.
- **Ctrl + Z-Key** -- Undo your last action.

So now that you can transform a given face or edge, I bet you want to work with something more complex than a six-sided cube. If you want to add more detail (more faces, pretty much), then check out the window directly below the 3D View -- it is called the buttons window.

## Add More Detail

Using the controls in the Buttons Window, you can revise the mesh to make it more detailed by subdividing and extruding, somewhat like a sculptor would cut away or add clay to a starting, undefined lump.

Let's try subdividing first. Select the face(s) or edge(s) (**R-Click, Shift(Hold) + R-Clicking**, or just **A-Key**) that you want to work with. Next, hit the subdivide button in the "Mesh Tools" tab (located on the right side of the picture above in the Buttons Window). The selected face should be split into many new, smaller faces. These new faces can now be selected and subdivided and so on. Play around again by selecting another face and subdividing it. Remain calm as the giddiness of this feat consumes you, do not despair when the rapture wears off and you realize all you've got is a square with a bunch of lines in it. Hit **Ctrl-Z** to Undo back to the square we started out with.

Another method for adding detail is to use the "Extrude" function. Select the top face of the cube and either hit the Extrude button in Mesh Tools, or you can just hit the **E-Key**. Without pressing anything else move your mouse upward. You should see your selected segment pull out from the rest of the surrounding mesh, in this case turning a square into a rectangle. Hitting the **LMB** will finalize the extrusion, hitting the **RMB** will cancel the extrusion and send the face back to where it was.

Subdividing and extruding in this way allows you to transform shapes as you wish. Try it and see what happens!

## The Beginnings are Over?! Nope, you're just Beginning!

Done playing already? Well, have you at least an understanding of how modeling works now? If not, don't worry--this is a very basic overview, and more explanation will be given in the next sections. And also, while it may seem like it would be super time-consuming to model this way, you will learn how to model better soon, and more efficiently. Just master how to do these techniques for now.

# Blender 3D: Noob to Pro/Beginning 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.

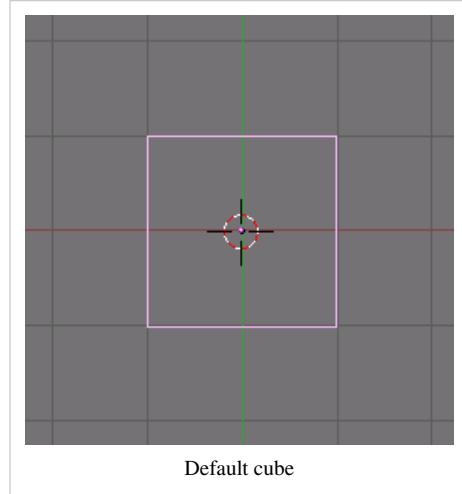
## Starting with a box

Tutorials will often start with the default cube you see right after opening Blender. Here are two ways to reset the scene without quitting the application:

- **Ctrl-X** (while holding the **Ctrl** key, press the **X** key);
- or select *File -> New* from the menu.

Then, you will see a prompt box asking **OK?** under your mouse pointer. You can confirm that you want to erase your current scene by clicking **Erase All** (or move the mouse around to dismiss it).

The cube is shown as a square in the 3D viewport. If you rotate the view while holding down **MMB** (middle mouse button), you'll see it is actually a cube. It is selected by default (pink outlines color). Also, you can hold down the **Alt** key and **Left Click** to simulate the Middle Mouse Button. Another way to do this for mice with only two buttons is by holding down both buttons at the same time. Navigating in 3D Space is assumed. Please see the excellent tutorials on Blender about User Interface Tutorial [1] , The Blender Windows[2], and Navigating in 3D Space[3] which are located on the Blender Quickstart page here: <http://www.blender.org/education-help/quickstart/>



Default cube

**i** You can change the default scene (and return to a personalized one when pressing **Ctrl-X**). Just modify the scene and arrange dialogs to suit your needs, then click *File > Save Default Settings*. Your current scene will now be used as the default when you click *File > New*. This is very handy indeed. To return to factory defaults, you can delete the file which contains those settings: **.blend** in your home directory. Starting from version 2.44, a new *Load Factory Settings* item is available from the *File* menu.

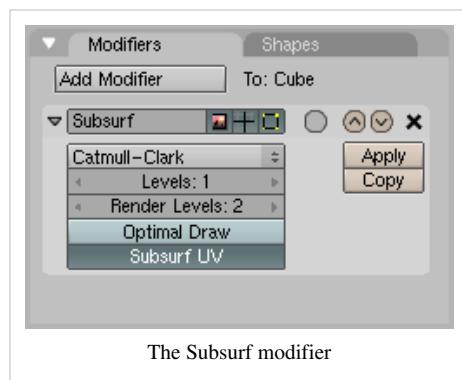
## Subdivision Surfaces

Subdivision surfaces, or subsurfing, is a common technique in 3D modeling. It 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.

### Adding a Subsurf modifier

First, select the cube by clicking **RMB** (right mouse button) on the cube in the 3D window (pink outlines indicate the selected objects). In other versions of Linux you accomplish this by clicking **LMB** (left mouse button). Now choose the *Editing* panel set in the "Buttons" (bottom) window:

- click on the icon in the panel list:
- or press **F9**.



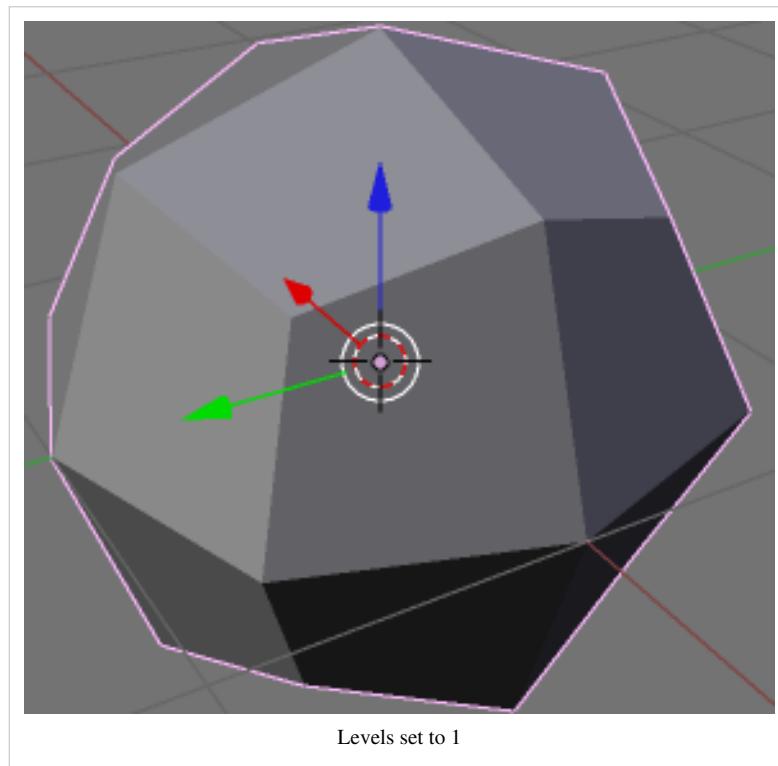
The Subsurf modifier

If you're not seeing a bunch of windows in the bottom view panel, such as *Link and Materials*, *Mesh*, *Multires*, *Modifiers*, *Shapes*, then there's a good chance you haven't got the cube itself selected or you are still in object mode.

In the *Modifiers* window, click on the *Add Modifier* button and select *Subsurf*. Click on the arrow to the right of *Levels:1* to increase the subsurf level. With each increment the cube becomes more smooth, and more planes are added. Don't do the following, but *if* you were to hit *apply*, the original form of the cube becomes lost. If you don't apply the changes they remain on the cube as if it was a filter, of sorts.

In order to complete the rest of this tutorial successfully, it is vital that you do ***NOT*** click *apply* within the *Subsurf* modifier tab.

Noobie asks: *I did this, and I cannot change the "Level". Why not?* Noobie2 responds: *Make sure you didn't hit multitires by mistake.* Noobie3 responds: *Maybe you accidentally clicked something else.* Noobie4 responds: *Maybe you overlooked the level "gauge"*

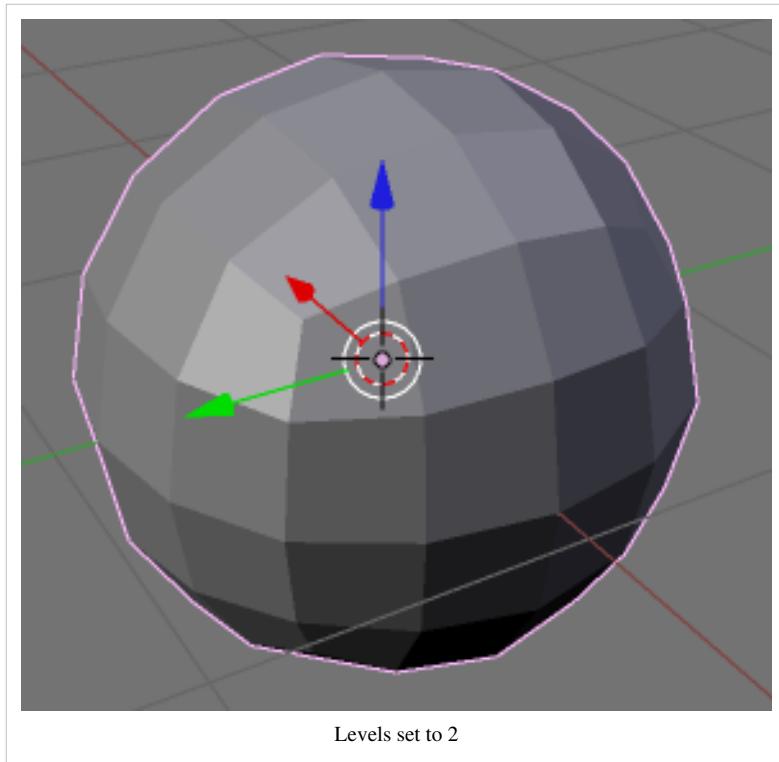


### But I want a box!

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. Unless the object is a knife edge, an object in the real world will have some sort of softer edge on it. 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.

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.



## Show subdivision surface's cage

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. (unless you're using version 2.45 of Blender)

Before we fiddle with the creasing, set the Subsurf *Levels* up to "4" so you can see the effect more clearly.

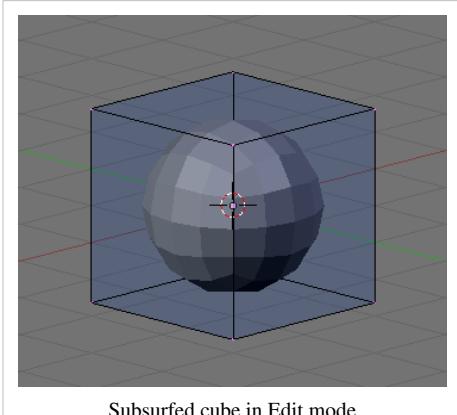
## Choose an edge to crease

Enter face mode by either:

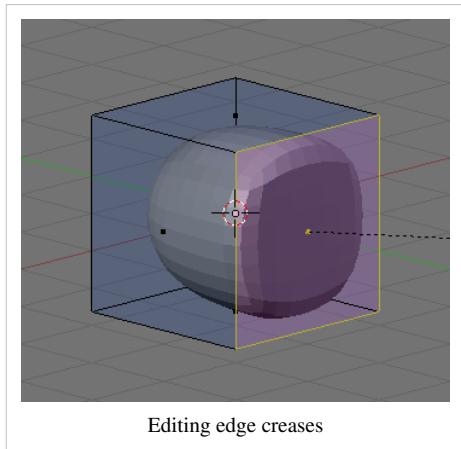
- Placing the cursor in the 3D View → changing to *Edit Mode* [4] → **CTRL+TAB** → choosing *Faces* (or pressing 3KEY)
- Changing to *Edit Mode* → clicking  (Face Mode Icon).

Select one of the sides of our wire cube with **RMB** by clicking near the dot in the centre of the face. You'll know when it's been selected because the other faces will change colour to grey, and the face you've selected will be highlighted.

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.



## Crease selected edges



Now crease the edges of the selection by either:

- Press **SHIFT+E** and slowly move the mouse observing the result of Crease operation on the selected face.
- In the 3D Viewport's header select Mesh → Edges → Crease SubSurf. 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. Mesh will change when pulling the dotted line on the opposite side of the creased face.

In the style of Blender, click **LMB** to apply the changes, or **RMB** to cancel creasing.

## Finally build a real box

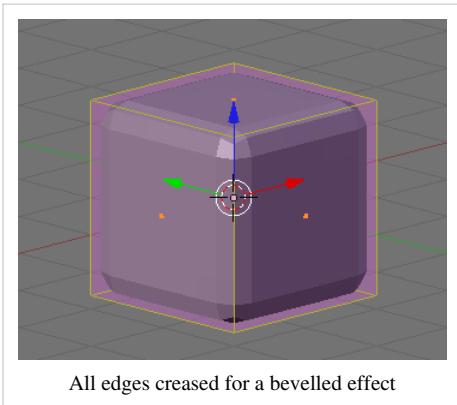
Either cancel the above edge crease, press **Ctrl-Z** to undo the last change, or start from scratch to get back to our simple subsurfed cube. (To get the same result as the picture on the right, set the subsurf to 3 or 4.) Then press the **A** key twice to select all faces. Crease them with **SHIFT+E** like before, until your cube looks like the image on the right.

*Note By a Nooby: If you are looking for a bevel effect that doesn't require so many polygons, try (in Edit Mode) pressing the **W** key then selecting Bevel.*

Click **LMB** to apply the changes then press **TAB** to cancel out of edit mode. Behold: your smooth cube.

When rendering:

- Make sure you have set the "Renders Level" in the Subsurf Modifiers tab.
- Also, double check that the "Enable modifier during rendering" toggle is on (dark) as it should be by default. It is the very small icon next to the Subsurf title button.



## Footnotes

1. ^ You can change to *Edit Mode* by selecting it from the drop down list on the 3D Viewport's header. Or, you can toggle between the current mode, and *Edit Mode* by using **TAB**.

## References

- [1] <http://www.blender.org/education-help/tutorials/tutorial-folder/blender-user-interface-tutorial/>
- [2] <http://www.blender.org/education-help/tutorials/tutorial-folder/the-blender-windows/>
- [3] <http://www.blender.org/education-help/tutorials/tutorial-folder/navigating-in-3d-space/>
- [4] [http://en.wikipedia.org/wiki/Blender\\_3d%3A\\_noob\\_to\\_pro%2Fbeginning\\_tips#endnote\\_editmode](http://en.wikipedia.org/wiki/Blender_3d%3A_noob_to_pro%2Fbeginning_tips#endnote_editmode)

# Blender 3D: Noob to Pro/Quickie Model

Your first model is easy.

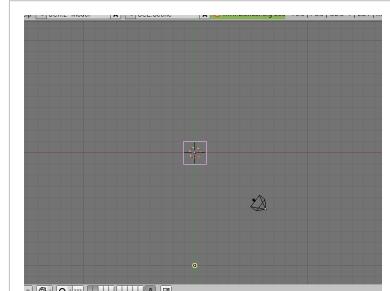
## Selecting objects

Start with the default scene. (**CTRL+X** or *File -> New*) It has three objects: a cube, a light source and a camera.

The cube is selected: pink outlines indicate the selected objects. With the mouse pointer in the viewport, you can select or deselect all objects by pressing the **A KEY** (the letter "A" on your keyboard). Select a single object by right-clicking on it (**RMB** or **CMD+LMB** on Mac).

**A KEY** - Toggles between selecting and deselecting all objects in a scene

**RMB** - Selects a single object



The default scene with the cube selected.

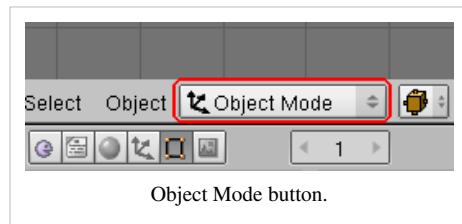
## Edit Mode

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. With the cube selected, hit **TAB**. This puts you in what's known as **Edit Mode**.

**Note:** If you've selected the lamp or the camera instead of the cube, you won't be able to go into Edit Mode (Cameras and Lamps are edited differently).

In *Edit Mode*, you can change the shape and size of the cube. You could turn the cube into a puppy... or at least soon you'll be able to.

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



Object Mode button.

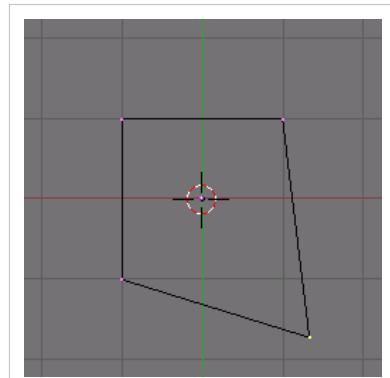
## Selecting vertices

Now that you're in *Edit Mode*, you have access to the individual vertices. Vertices are control points that you can connect to create edges and faces. Edges connect two vertices, and faces connect three or more vertices.

Vertices show up as *pink dots* when they're not selected, and *yellow dots* (*white* in version 2.49b) when they are selected. If you change the G.U.I. theme, these colors may change. For example, the Rounded theme uses orange and white for unselected and selected vertices respectively.

If all the vertices are yellow (selected), press **A KEY** to deselect all vertices (as seen above, this key toggles selection depending on the current mode). Go ahead and hit **RMB** over one of the vertices and you should see it change to yellow, which means that it is selected. (Mouse button actions can be changed under *View & Controls* in the User Preferences window.)

If all you see is a big blue dot:

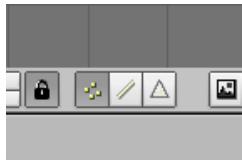


The cube after editing in edit mode.

- Make sure the *3D transform manipulator* is **off**; if not: depress the hand button  (in newer versions ), on the header; alternatively use the menu that appears when you press **CTRL+SPACE**. You'll know it's off when the icon showing the 3D axes disappears.

*If you cannot select a vertex:*

- Hit the **ZKEY** and make sure you are in transparent (wireframe) mode.
- If you can't get the cursor over the vertex, adjust your mouse/trackpad's tracking speed to minimum.
- Make sure you're in *vertex select mode*: if you can only select faces or edges, either press **CTRL+TAB** to select *Vertices* or click on the *Vertex select mode* icon as shown below.



Now try rotating the view to see what's actually going on. You can hold **ALT** key and drag (while holding the left mouse button, move the mouse) to rotate your view. If instead, it moves the Blender window, drag with the **MMB** (without holding the **ALT** key).

**ZKEY** - Toggles between drawing the scene in wireframe and solid mode.

**SHIFT+RMB** - extend selection (add or remove vertices from selection).

**CTRL+TAB** - Opens the selection mode menu.

**ALT+LMB** or **MMB** - Rotates the view.

**SHIFT+MMB** - Pans the view.

**CTRL+SPACE** - Opens menu for toggling the 3D transform manipulator.

## Moving vertices

With a vertex selected, use the *grab tool*:

- *Mesh > Transform > Grab/Move*,
- click and hold **LMB** on an empty space and draw a line,
- or just press **GKEY**.

Move your mouse around: you should see the selected vertex moving with the pointer! Click the **LMB** to drop the vertex at the current spot, or press **ENTER** or **SPACE** key. While moving, you can cancel the move and drop the vertex back where it came from by pressing **RMB** or **ESC** key.

You can also grab a selection using the mouse by holding **RMB** and dragging it around: release the button at the desired spot. Then, clicking on the same button cancels the move.

Now use the **MMB** to 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.

## Creating Vertices

While in mesh edit mode, simply hold the **CTRL** key while left clicking where you wish to create a vertex. Subsequent left clicks while holding the **CTRL** key will create a series of vertices with connected edges.

To select more vertex while clicking, hold the **Shift** key.

To create an edge, select two unconnected vertices and press **FKEY**.

To create a face, select three or four unconnected vertices (no more than four) and press **FKEY**.

To delete vertices, select one or more and press **DELETE**. A menu will pop up asking you what you would like to delete.

## Extra Practice

- A Video Tutorial on Edit Mode <sup>[1]</sup>

## References

[1] <http://www.youtube.com/watch?v=LL-cYoTQI4Q>

# Blender 3D: Noob to Pro/Quickie Render

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.

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.

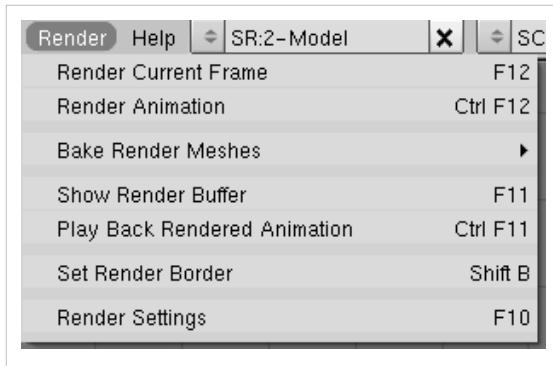
## Rendering the current scene

Now that you've created your first model, undoubtedly you'll want to render it. Make sure you're in object mode (press **TAB** if you're not), put the mouse pointer in the 3D view window and press **F12**! On Macintosh OS X 10.5 use **ALT+Fn+F12**. On Gnome you can use **ALT-F12** to avoid the Gnome Search Dialog. On the new Apple keyboard, use **Fn+F12** to avoid the Mac Dashboard.

If you have more than 1 processor, you can speed up rendering. (This is done automatically in Blender 2.46.) Hit **F10** to go to the render settings tab and in the bottom left corner, there is a threads button. Adjust the number of threads according to the number of cores in your processor (e.g. a dual core processor would be two threads, one for each core). Now, try re-rendering and you should get much faster results.



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



You can also use the menu in the User Preferences header:  
**Render > Render Current Frame.**

You can interrupt the rendering at any time by pressing **ESC** while the rendering window has the focus.

If you've put the render window behind the main window, you can get it back several ways: you can use the Windows taskbar or, under Windows and most other operating systems, you can use **ALT+TAB (CMD+TAB on Mac)**.

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.

**Note:** If your cube is completely 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. Remember to place the lamp in position where it is not inside the cube. This can be achieved using the **RMB** and pressing **G**.

**F12** - Starts the rendering from the active camera.

## Saving a render

At some point you will probably want to save your renders. In the User Preferences header, select *File > Save Rendered 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.

JPEG images, as opposed to PNG images, will contain unwanted artifacts (imperfections around edges)<sup>1</sup>. You can change the format by going to Render -> Render Settings or **F10**. Then under the "Format" panel, change the Type from Jpeg to PNG and hit **F3** again to update the file type in the file selector.

**F3** - Opens the *Save Image* dialog (if an image has been rendered).

- 1 - alternative route is to change the JPEG quality setting - just under the format selection list from the same "Render Settings" panel.

## Extra Practice

Tutorial on Using Multiple Cameras <sup>[1]</sup> <---- Pictures are missing from this tutorial

Basic Blender Camera Positioning (Rigging) <sup>[2]</sup>

## References

[1] <http://forum.nystic.com/viewtopic.php?f=34&t=2966#>

[2] <http://www.youtube.com/watch?v=NzEy2sCZcHo>

# Blender 3D: Noob to Pro/Modeling a Simple Person

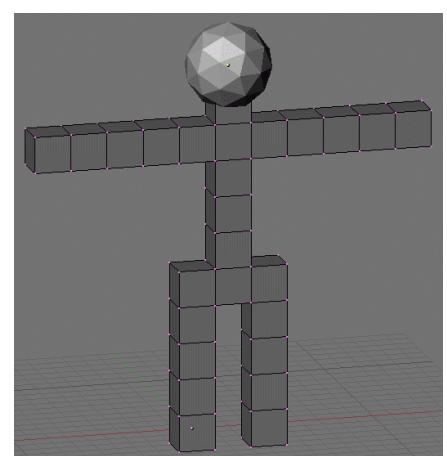
Now, we will create a simple character, learning about selection and extrusion along the way. Extrusion is one of the most widely used modeling tools available.

## Creating a New Project

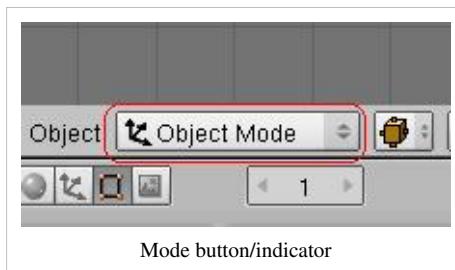
Start with the default scene (as explained here). You should have your default beginning cube.

### Reminder:

- Select the cube with **RMB** (**CMD+LMB** on one button Mac mouse).
- Drag with the **MMB** (**ALT+LMB** on one button Mac mouse) to have a look at the scene from different angles.
- Press **NUM7** to go back to the top view.
- Toggle between *Edit Mode* and *Object mode* with **TAB**; the option button shown on the image below tells what mode you are in at any given time:



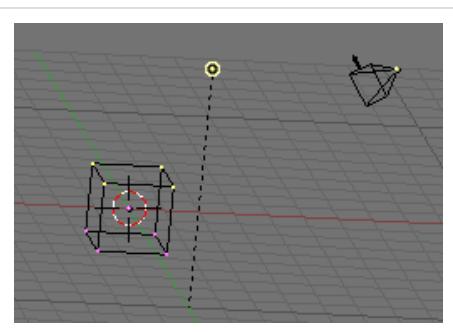
Your simple person will look like this.

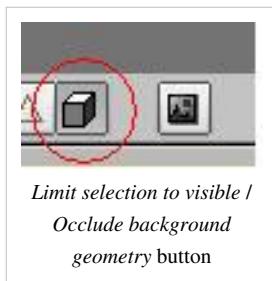


Mode button/indicator

## Selection Methods

This section proposes six methods for selecting the default cube's top four vertices. The image on the right shows the view rotated a bit with the correct vertices selected.





Before we start, make sure the *Limit selection to visible* button is on (in Blender 2.46 and above this button is called *Occlude background geometry*).

You'll only be able to find this button when you're in *Edit Mode*. Additionally, it is not available in *Wireframe Mode*: hit **ZKEY** if you cannot find it. It'll appear on the bottom of the 3D View window, far right, just left of the *Render* button.

**Note:** In Blender 2.42 for Mac OS X running on a MacBook, there is a *display problem* with the box and circle selections: the selection box and circle do not appear on screen (this is valid for both the Intel and the PowerPC versions).

The default is *Object Mode*. The cube should be selected; switch to *Edit Mode* then proceed.

## 1. Box Selection

This tool draws a square that you resize to frame the top four vertices (or dots).

1. Deselect all vertices by pressing the **AKEY**;
2. Press the **BKEY** to activate what is known as the *Box Border Select Tool* (it starts as two dotted lines).

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 unselect the selected vertices. Make sure all the vertices are deselected (pink, not yellow) before trying the next method.

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

## 2. Circle Selection

1. Deselect all vertices by pressing the **AKEY**;
2. Press the **BKEY** twice to activate the Circle Border Select tool.

A circle appears around the mouse cursor. You can resize the circle with **SCROLL** (the mouse wheel) or alternatively use the **NUM+NUM-** or **PgUp/PgDn** keys.

- Select vertices either by dragging with **LMB** or clicking at several places.
- Deselect vertices by clicking or dragging with the **MMB** (or **ALT+LMB**).

To adjust your selection, note that **SHIFT+RMB** toggles a single vertex selection.

**Note:** If **ALT + LMB** moves the current window, then to unselect a vertex use **CTRL+ALT+LMB** or **MMB** instead.

The Circle Border Select tool will be active until you press **RMB**, **ESC** or **SPACE**.

**BKEY+BKEY** - Circle Select. If you press **BKEY** a second time after starting Border Select, Circle Select is invoked. Use **NUM+** or **NUM-** or **MW** to adjust the circle size. Leave Circle Select with **RMB** or **ESC**.

### 3. Lasso Selection

Like many graphics programs, Blender 3D has a lasso selection tool.

1. Deselect all vertices by pressing the **AKEY**;
  2. Hold **CTRL+LMB** and draw a circle around the vertices you want to select. Release the LMB when you're done.
- To deselect with the lasso, use **CTRL+SHIFT+LMB**.

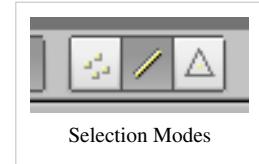
### 4. One By One Selection

You can also select the four vertices one by one.

1. Deselect all vertices by pressing the **AKEY**;
2. Select a single vertex with **RMB (CMD+LMB on one button Mac mouse)**;
3. Select additional vertices by holding **SHIFT** while pressing the **RMB**. Clicking again on a selected vertex deselects it.

### 5. Edge Selection

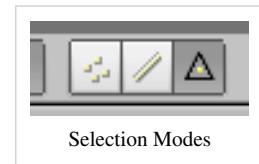
In addition to those vertex selection methods, there are two other options: on the right of your viewport header you can see selection modes. Choose the *Edge select mode* (the middle mode showing two parallel lines) and select the left edge of the cube with the **RMB (CMD+LMB on one button Mac mouse)**. Then **SHIFT+RMB** on the right edge to add it to the selection. Then switch back to *Vertex select mode* (the four dots in a diamond formation). As you will see, all four vertices forming the two top edges are selected (this is also called "selection transformation").



Selection Modes

### 6. Face Selection

The second alternate option to the vertex selection method is also available. On the right of your viewport header, choose the *Face select mode* (the right button with a triangle with a dot inside) and select the top face of the cube with the **RMB (CMD+LMB on one button Mac mouse)**. Then switch back to *Vertex select mode* (the four dots in a diamond formation). As you will see, all four vertices forming the top face are selected (this is also called "selection transformation"). To select additional faces, hold **SHIFT** while pressing the **RMB**. Alternatively, with the mouse pointer in the 3D viewport, you can hit **CTRL+TAB** and select *Vertices* or *Faces* mode from the popup menu.



Selection Modes

You can de-select a face by hitting **AKEY**

## Learning Extrusion

Note -latest version of Blender does not act completely as below, clarification please!

The pictures below are in orthographic view. Depending on Blender's version, the default view is either perspective or orthographic. If you need to switch to the orthographic view, press **NUM5** (or choose it from the **VIEW** menu, as shown in the picture).

Now press **NUM1** to switch to front view.

### Region extrusion

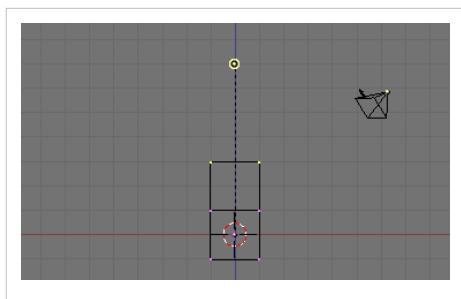
With the top four vertices selected (which will appear like the top two in your screen), hit the **EKEY**. Choose *Region* from the popup menu, then move the mouse upwards: four -new- vertices attached to the four that were previously selected are moving around with the mouse pointer. You can drop them in place with **LMB**, **SPACE**, or **ENTER**.

#### Notes:

- In Blender 2.42a and 2.43a, you may not have the *Region* option; so just ignore choosing region and continue.
- If the menu doesn't popup, you are probably in face selection mode. Move back to vertex selection mode by clicking the right icon.
- If the popup menu only presents the *Only Edges* and *Only Vertices* choices, you probably have not selected four vertices that make up a face. (It can also happen when some of the vertices are doubled: try selecting all vertices while in *Edit Mode* and hit the **WKEY** to display the *Specials* menu. Then choose *Remove Doubles*; it can also be accessed through the *Rem Doubles* under the *Mesh Tools* tab).
- In some versions of blender, you may find that, by default, the extrusion is performed along a different axis than the ones used here. You can set the axis along which the extrusion is performed. To do this, first press the **EKEY**, choose region, and then press **MMB** until the correct axis is selected.

**EKEY** - Extrude selection

### Starting with a simple leg



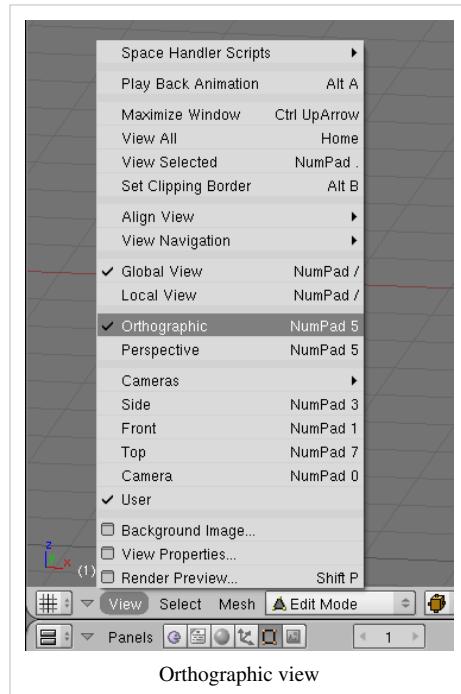
If you attempt to extrude the vertices but they do not end up at the right spot for this tutorial, use your **CTRL+ZKEY (CMD+ZKEY Mac)** to undo your last edit. You should see just your original cube with the top four vertices selected and then try what's next

Press the **EKEY** again. Again, choose Region. Now this time, as you're moving the extruded vertices around, hold the **CTRL** key down and you'll see that the extrusion will only move to certain spots. This is called *snapping*. The vertices *snap* to predetermined positions that

allow you to better work with the extrusion.

We'll talk more about snapping later. For now, set and release the vertices at the spot that makes it look like two cubes of equal size to the first one, stacked one on top of the other.

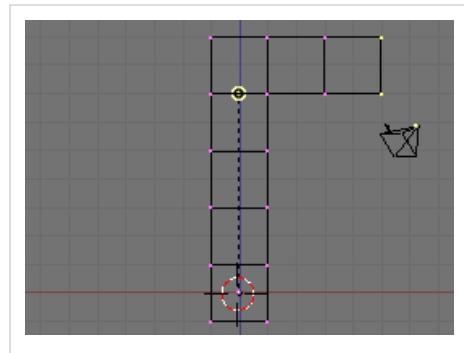
Then, repeat that same process until you have five boxes of equal size stacked one on top of each other. And that, my friend, is a very simple leg!



**Hint:** Don't stretch one box all the way to make the desired shape - You must create all stacked boxes in sequence, or you won't get the nodes (a more detailed mesh) that will be required to create the leg in this tutorial.

## And now, the pelvis

Hit **AKEY** to unselect the current vertices. Select the four vertices on the right face of the top cube. You may want to rotate your view a little with the **MMB** or with **ALT+LMB** to see them all. Alternatively, with *Limit Selection to Visible* off, a simple box selection (**BKEY**) over the two visible vertices will also select the ones behind them. Extrude twice to the right.



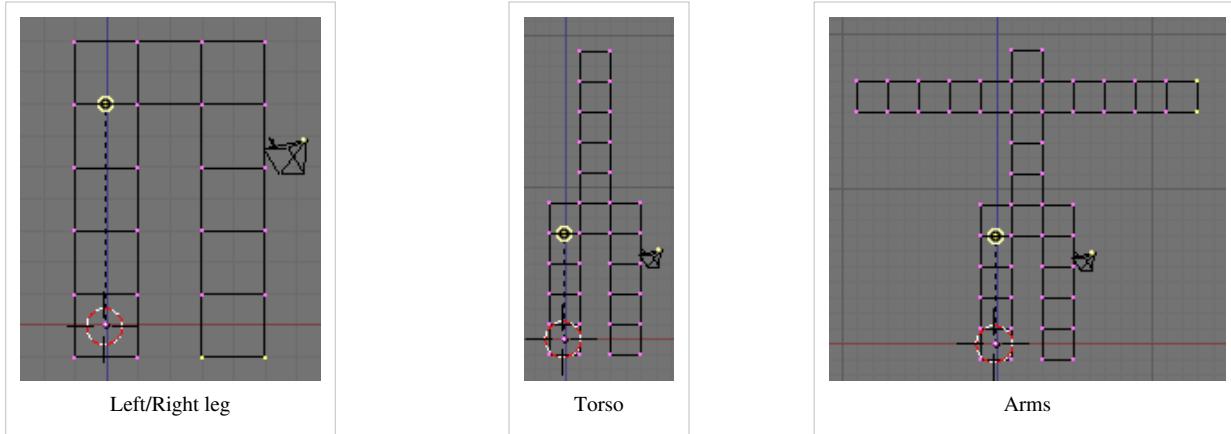
## Drawing the other body parts

The same trick is repeated over and over to build the rest of our simple body.

You may want to switch to *Face select mode* to select the four vertices of a face with a single click. This way the extrusion tools will automatically extrude a region, so you won't have to choose the *Region* option each time you extrude a face.

At this point your model might get too big to fit in your view. You can pan the view by:

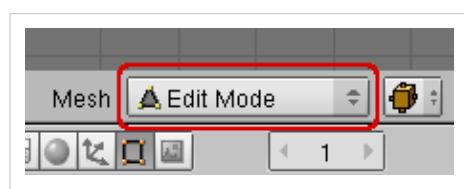
- either holding **CTRL** and pressing **NUM4**, **NUM8**, **NUM6**, **NUM2**,
- or holding **SHIFT** and dragging with **MMB**.



Now, check that all is well: toggle on solid mode by hitting the **ZKEY** and examine every body side. If some faces are missing, it's easy to fix. To create a face from four vertices, select them and press the **FKEY** (or choose the *Mesh → Make Edge/Face* menu from the viewport header). You need to remove any doubles by hitting the **WKEY** and select **Remove Doubles** from the menu.

## Adding the head

**Important note:** make sure you're still in Edit Mode (pictured) when adding the head. If you're not, the head and the body will not be part of the same object and changes on the body won't affect the head, which is required in the next section.



Select a point just above the top of the neck using the **LMB**: the red and white circle is the cursor. To adjust the cursor's position, switch between the top, front and side views (using the **NUM7**, **NUM1**, and **NUM3** key respectively). You can also use the snap tool: press **SHIFT+S** to bring up the snap menu and select *Cursor → Grid*.

Once you're happy with the position, press the **SPACE** key to bring up the popup menu. Select *Add → Icosphere*. In some Blender versions you may have to choose the subdivision number. Just click *OK*. You should now have a small sphere at the top of the body. To make it more proportional to the body, resize it with the scale tool:

- select *Mesh → Transform → Scale* from the viewport menu,
- while holding LMB, draw a triangle on the screen,
- or just press the **SKEY**, then hold LMB and drag the mouse.

If you unselect the head and then decide that you want to move it or resize it again, select one vertex of the head, then click *Select → Linked Vertices* (or use **CTRL+L**). All the head's vertices will then be selected again, and none of the body's. Then press **GKEY** to grab and move the head, or **SKEY** to resize it. Hold down **CTRL** as you move it around if you would like it to snap to the grid.

Don't forget that you are in 3D; use the **MMB** to move your point of view around to make sure that the head really sticks in the neck.

## Summary: Keys & Commands

These are the keys and commands used on this page:

Key	Mode	Description
<b>RMB</b> or <b>CMD+LMB</b> (Mac users with one mouse button)	Object	Select an object
<b>NUM1</b>		Go to front view
<b>TAB</b>		Toggle between Edit Mode and Object Mode
<b>BKEY</b> then <b>LMB</b> and drag ( <b>RMB</b> to unselect)		Box selection
<b>AKEY</b>		Toggle between Select All and Select None
<b>BKEY BKEY</b> (pressed twice) then <b>LMB</b> and drag		Circle selection
<b>CTRL+LMB</b> and drag		Lasso selection
<b>RMB</b> then <b>SHIFT+RMB</b>		One-by-one selection
(click the vertex/edge/face selection buttons)		Change the selection mode
<b>CMD+TAB</b> ( <b>CTRL+TAB</b> in Windows/UNIX)		Change the selection mode
select vertices then <b>EKEY</b>		Extrude
<b>CTRL</b>	while extruding	Enable snapping
	while moving	Enable snapping
	while rotating	Rotate in 5-degree intervals
<b>MMB</b> or <b>ALT+LMB</b>		Rotate the 3D view
<b>ZKEY</b>		Toggle wireframe/solid view
<b>FKEY</b>		Make Edge/Face from selected vertices
<b>NUM3</b>		Side view
<b>SHIFT+SKEY</b>		Snap cursor or selection to the grid
<b>GKEY</b>		Grab the current selection and move it

<b>ZKEY (or XKEY or YKEY)</b>	grab mode ( <b>GKEY</b> )	Constrain motion to the Z (or X or Y) axis
<b>SKEY</b>		Change the scale (size) of selection
<b>SPACE</b>		Brings up the menu
<b>WKEY</b>		To choose Remove Doubles; it can also be accessed through the Rem Doubles under the Mesh Tools tab

# Blender 3D: Noob to Pro/Detailing Your Simple Person 1

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 <sup>[1]</sup>.

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. Return to Object Mode (**TAB**) and press the **ZKEY** to see the model in solid mode.

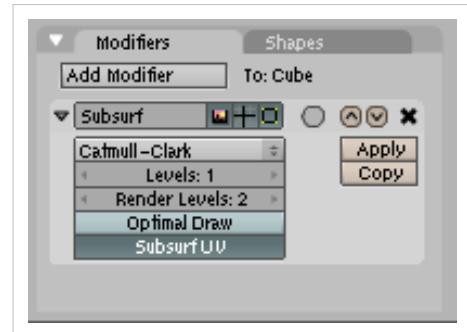
## Subsurfaces

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 (but make sure you are in Object Mode before continuing).



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 and less edgy. New options for *Subsurf* are now shown in the *Modifiers* subpanel. You may also perform this action by pressing **SHIFT+O** 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 move your mouse to the left or right while hovering the mouse cursor over the number.

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). Blender v2.48a is able to do this with much greater ease than previous versions, so if you are running this version go ahead and set the level to 3. 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  $4^n$  divisions.

[Troubleshooting: If one or two of your sides don't subsurf, press **AKEY** to select all vertices while in *edit mode*. Then, click *Rem Doubles* under the *Mesh Tools* tab in the *buttons window*. You can also change the *Limit* of how far Blender should look for vertices that are close together. If you prefer keyboard shortcuts, press **WKEY** to display the *specials* menu, and select *Remove Doubles*. ]

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



In Edit Mode, hit the **AKEY** once or twice so that all the vertices are selected (if you're not in Edit Mode, select the object and press **TAB**). Then press the blank roundish button towards the top of the Modifiers panel, just to the left of the up and down arrows. This button applies the modifier to the editing cage. Notice how it transforms the translucent, boxy cage to a more rounded one. Take note of this function for future reference, but for now press it again to return to the boxy version. You will need this boxy version for the next few lessons. The *Apply* button applies the modifier to the mesh. While this is useful for some modifiers, for Subsurf this will add many extra vertices and is generally not needed.

Remember, you can undo any accidental modifications by pressing **CTRL+Z** to go back a step. If you need to, you can go back several steps and then repeat the process correctly.

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

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

For a complete subsurfaces documentation go to [http://wiki.blender.org/index.php/Manual/Subsurf\\_Modifier](http://wiki.blender.org/index.php/Manual/Subsurf_Modifier)

If your Blender crashes when you attempt to subsurf a large object, you need to look into upgrading or possibly even downgrading your video card drivers. Blender doesn't work well with certain drivers of new videocards, but having the right one can save a lot of headache.

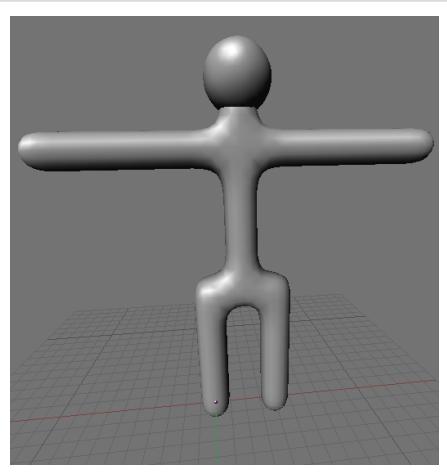
## Smooth Surfaces

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.

If your person is in wireframe view, hit **ZKEY** to change it to solid view. In Edit Mode, hit the **AKEY** once or twice so that all the vertices are selected (if you're not in Edit Mode, select the object and press **TAB**). Find the button that says "Set Smooth" and click it (the bottom-middle button inside the *Links and Materials* subpanel in the Editing [**F9**] panel). Note: If you can not find the button, try switching from edit mode to object mode.

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] and [3]).

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



Your simple person after setting smooth.

## References

- [1] <http://www.nusoy.com/blender>
- [2] [http://en.wikipedia.org/wiki/Flat\\_shading](http://en.wikipedia.org/wiki/Flat_shading)
- [3] [http://en.wikipedia.org/wiki/Gouraud\\_shading](http://en.wikipedia.org/wiki/Gouraud_shading)

# Blender 3D: Noob to Pro/Detailing Your Simple Person 2

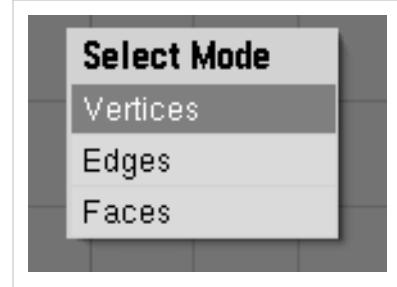
This tutorial uses the simple person model from the previous page. If you didn't do it, go back and do it now or find it pre-made just for you here <sup>[1]</sup>.

## Starting with the right 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)*,
- *Solid Mode (ZKEY)*,
- and *Edges select mode*: press **CTRL+TAB**, a menu will come up where you can choose Vertices, Edges, or Faces, then choose *Edges*. The three selection modes can also be selected with the statusline buttons shown on the right.



**Note for KDE users (not 3.5.6):** *CTRL+TAB changes the desktop so you will have to use the statusline buttons instead. But also you can configure KDE hotkeys for blender window. (Older versions of Blender do not have this feature. Instead, just select all vertices connected to the edge you want to select).*

It is important to remember that depending on the selection mode you're in (vertices, edges, or faces), 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.

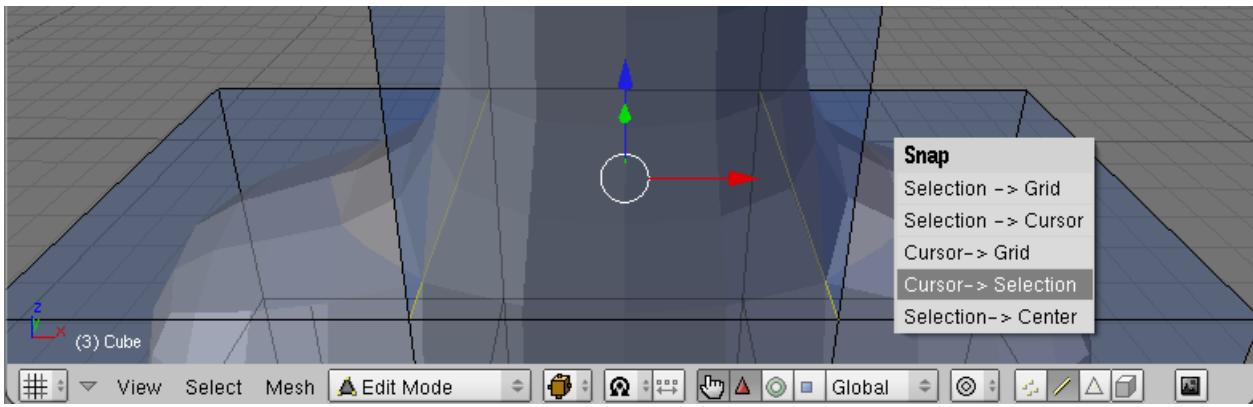
## Scaling with axis constraint

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

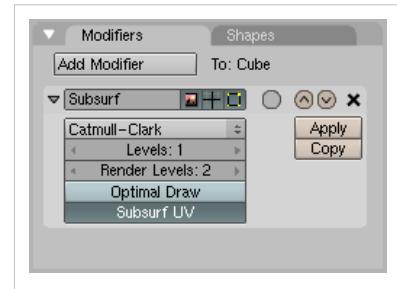
First, make sure:

- everything is deselected (**AKEY** or *Select → Select/Deselect All* from the viewport's menu),
- you're in perspective mode (**NUM5** or *View → Perspective* menu),
- the 3D transform manipulator is on (switch on the hand button ; this may appear as a hand with an index finger).

Our goal is to place the cursor as it is shown in the picture below, that is at the center of the two selected edges (instead of snapping it to the grid as we did when adding the head):

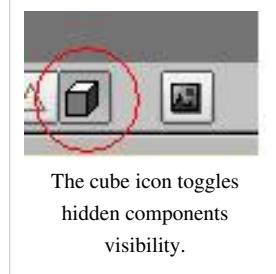


**Troubleshooting:** if you do not see the cubes around your person, make sure you are in Edit Mode as explained in the introduction. In case you played with the Subsurf modifier, you may have to uncheck the Apply modifier to editing cage during Editmode box, just on the left of the Move modifier up in stack button (^). If it still doesn't work, try deleting the Subsurf modifier (the X on the right, above the Apply button) and adding it again (Add Modifier -> Subsurf). Or you may have to redo it. (the entire model) or you can keep pressing **Ctrl+Z** until you undo the smooth command.



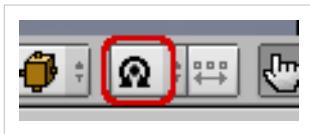
## Selecting two hip edges

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 (pictured on the right). Toggle it on and off a few times and observe how the edges of the wire cage appear and disappear as you click. **Disable it for now**, to show the hidden edges. (*Note that this button is called "Occlude background geometry" and behaves slightly differently in V2.46.*)



Now, select an edge of one of the cubes to the left or right of the model's pelvis just above the hips, where the legs connect to the torso: (**RMB**). Notice that the 3D transform manipulator jumps to the edge you selected. Now select the edge on the other side of the pelvis (**SHIFT+RMB**). The 3D transform manipulator should jump halfway between the two edges.

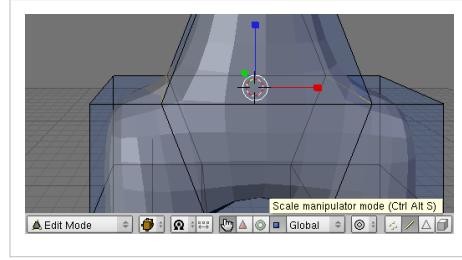
Once you have both edges selected as in the image, press **SHIFT+SKEY** to bring up the Snap Menu and select Cursor->Selection.



**Troubleshooting:** If, instead, it jumps to the second selected edge, change your Rotation/Scaling Pivot to Median Point.

## Scaling the hips

Choose the scaling manipulators: **CTRL-SPACE** and select *Scale* (or use **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. Make sure *Proportional Edit Falloff* is set to Off. The axes are colored R-G-B for X-Y-Z, i.e. the X-axis is red, the Y-axis is green, the Z-axis is blue.



It's important to note that in addition to the global XYZ axes, each individual object has its own XYZ axes. We'll get into that in the next section.

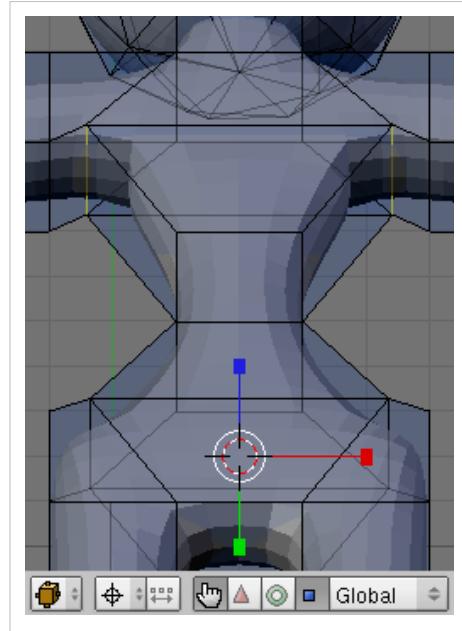
Grab the red cube-shaped handle and drag it with **LMB** to symmetrically widen up the selection along the selected X-axis.(For some people who get mixed up you might have to grab the green cube-shaped handle to widen up the person, remember you dont want to make the persons pelvis longer you want it wider.) While scaling, press **CTRL** to snap to the grid or **ESC** to abort the current manipulation. When it comes to scaling in Blender, 1.0000 means 100%, 0.6000 means 60%, and so on. Scale up to 2.

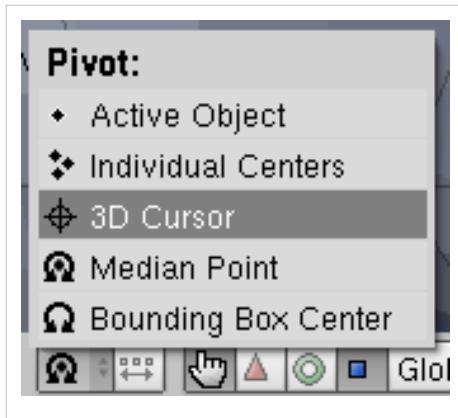
**Note:** you cannot scale along the Z-axis, as the current selection's Z-dimension is zero — if you want to symmetrically lift the hips, switch back to Translate Manipulator Mode (**CTRL+ALT+GKEY**).

## Drawing the armpits

We'll now use the 3D cursor instead of the selection's center: bring up the *Mesh → Snap* menu (**SHIFT-SKEY**) and select *Cursor → Selection* (**KEY4**). This will move the 3D cursor to the location of the manipulator.

**Troubleshoot:** If the joints seem to jump into the center, pulling the edges towards them, remember to select *Cursor → Selection*, not *Selection → Cursor*





Now, set the *Rotation/Scaling Pivot* to 3D Cursor. Since the 3D Cursor was positioned to the selection's center, the manipulator's behavior stays the same.

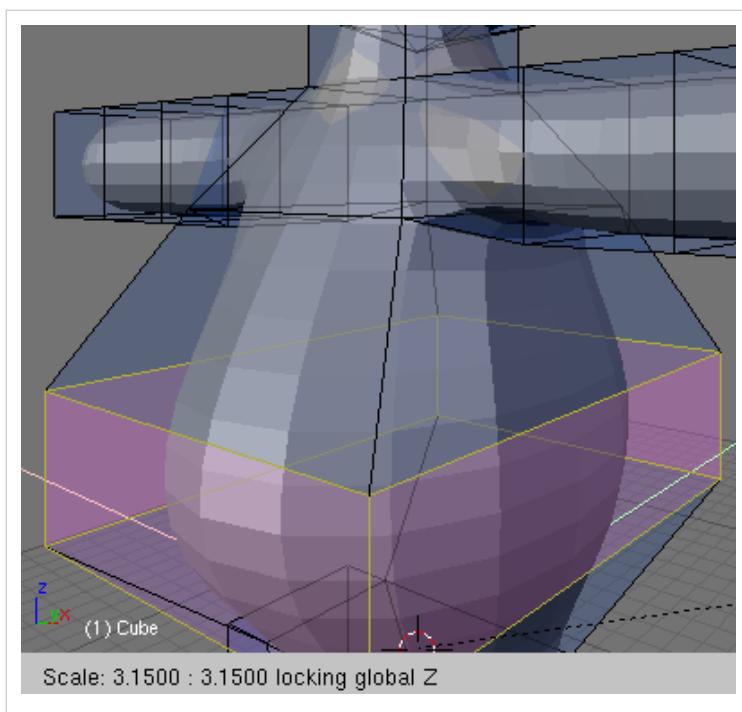
Select the two edges under the arms where they connect to the torso. This time, the manipulator does not jump to the selection but stays at the 3D cursor. Make sure you are in *Scale Manipulator Mode* (**CTRL+ALT+SKEY**) and form the armpits using the square handles on the manipulator: say 2.0 along X and 1.1 along Z.

#### Notes:

- It is easier to select the edges by rotating the view around the world's X-axis with View → View Navigation → Orbit Down (or **NUM2**).

- For better visual comparison to the width of the hips, switch to View → Orthographic (or **NUM5**) before scaling along the X-axis (the red one). You can now scale along the Z-axis (the blue one), as there is a distance along Z between the selection and the pivot.

## The belly and the chest



Now, deselect all and select the belly cube (use one of the methods described here). This time, use the *Scale Tool* instead of the scaling manipulator:

- press the **SKEY** to choose the scale tool;
- and then **SHIFT-ZKEY** to lock the Z-axis. Now, the scale tool is constrained to the X and Y axes (i.e. the selection is not scaled along the Z-axis) and those axes are drawn through the pivot in a bright color;
- scale the belly using **LMB**.

Continue with selecting different sections of the torso and scaling them to your liking, exercising above scaling methods.

*Note that just as 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. If you undo too many steps press **CTRL+YKEY** to redo what you just undid.*

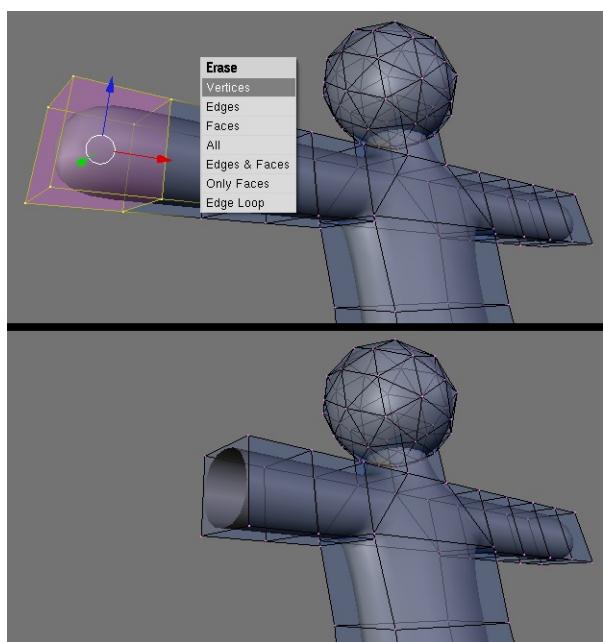
## Modeling the arms

When you've got the basic shape of the torso, move on to the arms. We'll start by making him holding his hands up.

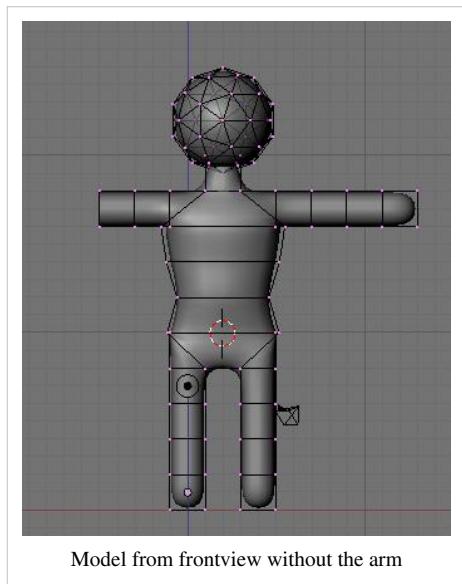
First, make sure you're in *Edit Mode*; if not, select the figure and press **TAB**. Also, make sure you are in *Vertex select mode* (**CTRL+TAB**). Now, select the 8 vertices at the end of one arm (the hand cube).

Press the **XKEY**, and choose *Vertices* in the popup menu. Suddenly the box disappears, and at the end of the arm, there's a hole! Don't panic. We'll fix that in a moment.

At this point, your person should look like the image below:



Removing the end of the arm

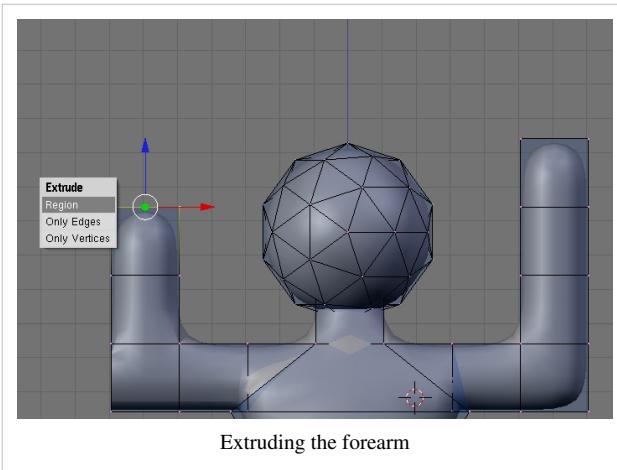


Model from frontview without the arm

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

If everytime you try to extrude it puts a weird angled section on the back right side of the extrusion then you may be still in Perspective mode. Select Orthographic mode *View → Orthographic* (or **NumPad5**)

Rotate manipulator mode is the double circle to the right of the 3D Manipulator hand (or **CTRL+ALT+R**) Scale manipulator mode is the small blue square to the right of the 3D Manipulator hand (or **CTRL+ALT+S**)

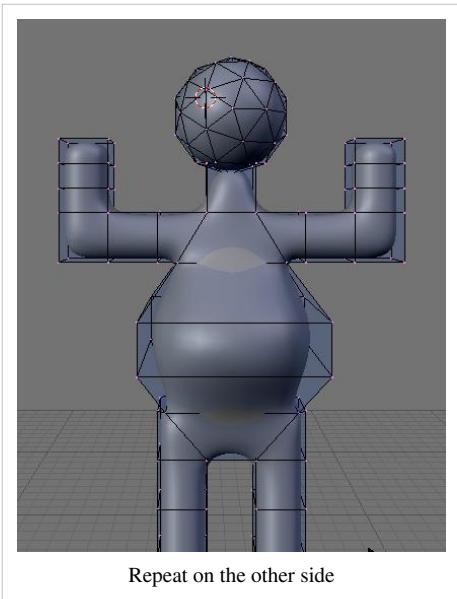
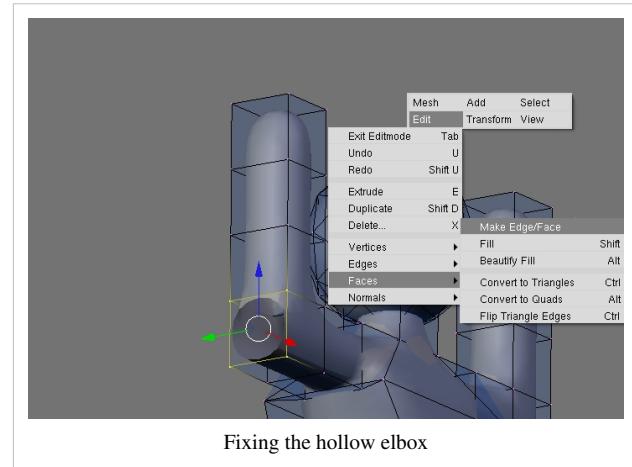


We'll now fix the hollow elbow. Simply select the four vertices at the gaping hole (turn on *Limit Selection to Visible* mode to make it easier), and press:

**SPACE → Edit → Faces → Make Edge/Face**  
(or **FKEY**)

Notice that the hole was covered by a face. Now to make it a smooth face, choose the option:

**SPACE → Edit → Faces → Set Smooth**



Do the same with the other arm. Make sure to deselect all the selected vertices from the first arm (**AKEY**). It is important to follow the steps in the same order to end up with identical arms. If you're having troubles with the other arm, you can undo all the arm work and redo every step simultaneously on both arms.

**Troubleshooting:** if the surface of the model swells out where you added the face to cover a hole, use **CTRL-Z** to undo the face. Try selecting all the hole vertices (or even select the whole figure) and choose: Mesh → Vertices → Remove Doubles from the viewport menu, and try to add the face again. If it still looks strange, then without undoing it, select the whole figure with the **AKEY** and use **CTRL-N** to recalculate the normals.

## Modeling the legs

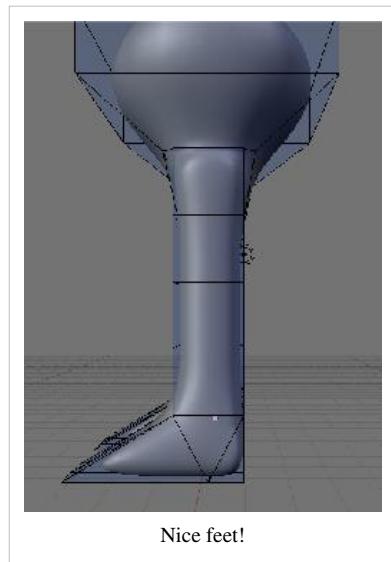
First switch to the *Face select mode* (choose from the **CTRL+TAB** menu, or click the triangle icon).

Select the two bottom faces of the feet (the soles): use **RMB** and hold down **SHIFT** when selecting the second one. Each face comes with a small square denoting the face center that turns orange when selected, while the outline is highlighted in yellow.

Then, subdivide them: **WKEY**, *Subdivide*, or, **SPACE** → *Edit* → *Edges* → *Subdivide* (or, from the viewport menu: *Mesh* → *Edges* → *Subdivide*).

Now, switch to the *Edge select mode* (**CTRL+TAB**) and clear the selection (**AKEY**).

Select the bottom front edges making the toes (**RMB**, then **SHIFT+RMB**). You should end up with four edges selected.



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: drag with **MMB** for orthogonal movement and drop them with **LMB**.

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

Congratulations! We now have feet!

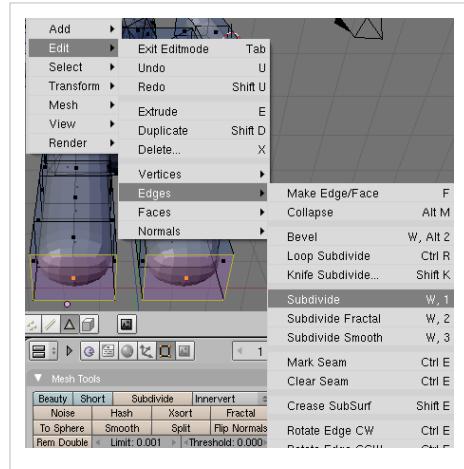
## Modeling the head

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

- Press the **AKEY** to clear your selection.
- Place the mouse cursor over the head and press the **LKEY**: this selects the closest edge, face, or vertex, 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.

- Place the 3D cursor in the middle of the head (**SHIFT-S**, *Cursor* → *Selection*) or just set *Median Point* as Pivot (**CTRL-,**). Then press **CTRL+ALT+S** and scale on the Z-axis (blue handle of the 3D manipulator) in order to get a better shape. I think 1.5 is enough. Without using the 3D manipulator, remember that you need to press the **ZKEY** to restrict the scaling to the Z-axis only (in both cases, **CTRL** snaps the values).
- After elongating the head, you may find that it is too low or too high. To fix this, press the **GKEY** (to move it) and the **ZKEY** (to restrict the movement to the Z-axis). Play around with it a little until you like the result.

**Note:** another course of action would be to put the cursor (and thus the 3D transform manipulator) at the underside of the head. That way the neck will keep the same length, while you can scale the head at will.



# Blender 3D: Noob to Pro/Creating a Simple Hat

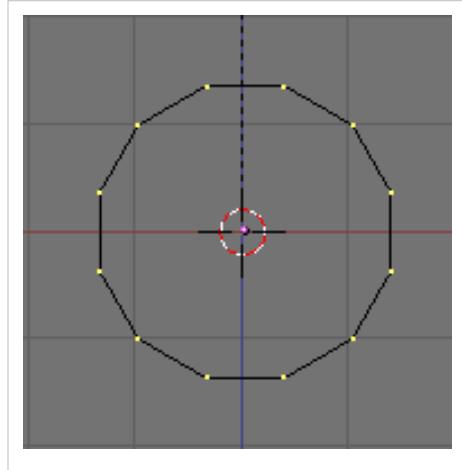
This tutorial uses the simple person model from a previous tutorial. If you didn't do it, go back and do it now---or find it pre-made just for you here <sup>[1]</sup>. Alternatively, if you just want to make the hat without putting it on a person at the end, you can go ahead and do that without having completed the other tutorial.

## Adding an object

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

### Setup

- use the orthographic view (**NUM5**);
- get a sideways view of the model (**NUM3**);
- if you're starting with the simple person model, pan the view (**SHIFT+MMB**) to place the scene center a good distance above the simple person's head;
- make sure you're in *Object Mode* (**TAB**) to add the hat as a separate object. (*Note: It will cause problems if you do not switch into Object Mode. Actually, you will rework that manual again later*)



Now, place the 3D cursor on center of the view (**LMB**) and snap it to the nearest grid node (**SHIFT+SKEY** then *Cursor → Grid*).

### Create a circle

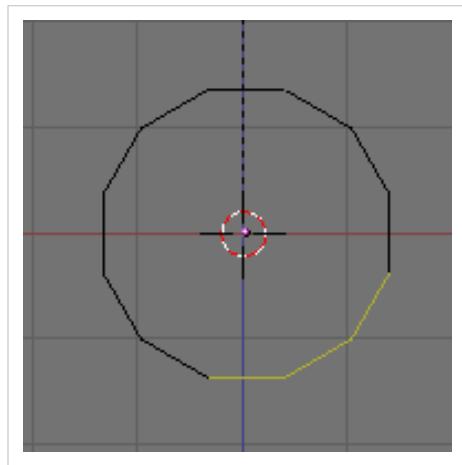
Do **SPACE** → **Add** → **Mesh** → **Circle**, with 12 vertices. In the latest Blender version the default is 32 vertices but you may use the arrows to set it back to 12: click the arrows to change, or click and drag left or right, or click on the number and type a new one. Change the radius to 1.41. Click on *OK*. We now have a selected circle. In the bottom left corner of the viewport, you should see the name of the selected object: *Circle*.

If all you see is a line, then you are looking at the side of the circle. Rotate it around to see it from a different angle, or switch to Top view(**NUM7**)

## Deleting a selection

Make sure you are in *Edit mode*, then switch to *Edge select mode* (**CTRL+TAB**) and have only the three edges selected as seen in the picture (**AKEY** to deselect all, then click **RMB**; hold down the **SHIFT** key when selecting the second and the third ones).

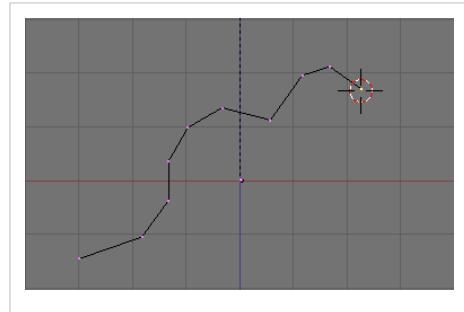
Delete these edges by pressing **XKEY** → *Edges*.



## Creating the hat profile

Now make sure you're in edit mode and 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:

- **AKEY** to select/deselect all vertices
- **RMB** to select/deselect a vertex
- **SHIFT+RMB** to select/deselect multiple vertices (or **BKEY** to use the *Border select tool*)
- **GKEY** to move a selection (hold down **CTRL** while moving to use snapping)
- **EKEY** to create another vertex attached to the one selected

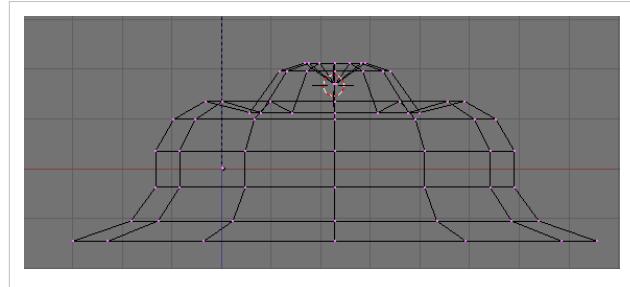


**Notice:** Be sure that the *Proportional Edit Falloff* button is off (orange is on; gray is off). You can toggle this control with the **OKEY**. We'll learn more about proportional edit in a later tutorial, but for this step, it should be off, meaning that you have full control on each vertex separately.

## Spinning the hat

To make a hat out of this curve, we'll use the *Spin* tool to create a surface of revolution.

**Note:** the *Spin* tool, like some other Blender operations, works differently based on the rotation of the 3D view you are on: it will rotate the polyline around the axis that is perpendicular to the plane of the active 3D view and that passes through the 3D cursor.



## Setup

This will make sure the 3D cursor is placed exactly at the rightmost vertex:

- select the rightmost vertex (**RMB**),
- then choose **SHIFT+SKEY** and *Cursor* → *Selection*.

Then select the curve to spin and choose the rotation axis:

- select all the curve vertices using your favorite method (**AKEY**, **BKEY**, etc...),
- choose the top view (**NUM7**).

## Spin the curve

Switch to the *Editing* panel (**F9**, or **CMD+F9** for Mac users). In the *Mesh Tools* tab, locate the *Spin*, *Spin Dup* and *Screw* buttons and fill the fields as indicated below:

Degr: 360

Steps: 12

Turns: 1

**Noob Note:** There are 3 similarly named tabs [Mesh], [Mesh Tools] and [Mesh Tools 1] (in Blender 2.46, [Mesh Tools More]). Look under [Mesh Tools], which can be found in the editing tab.

Now hit *Spin* to create a surface of revolution around the Z-axis.

**Troubleshooting:** in case you have more than one 3D window open, your mouse cursor may change to a "?" sign: additionally click on the window in which you want to perform your rotation (the top view window).

**Note:** the rotation axis is parallel to the Z-axis because we chose the top view.

## BIG UPDATE

After much struggling to follow many tutorials based on older versions of Blender, I have downloaded multiple versions to discover why the tutorials based on versions such as 2.43 don't work when attempted on updated versions such as Blender 2.48a and above.

Newer versions such as 2.48 have added a new option to **/not/** have added objects rotated to the current viewpoint. With older versions, being in top, front or side view would cause any newly added objects to face different directions on creation. Newer versions of Blender introduced the ability to force all objects into the same global orientation; even worse, they set it up that way **BY DEFAULT!** This means that unless the user deliberately changes the settings in the new versions, many older tutorials will act as if they are broken.

Here is how newer versions of Blender can be set to act in the same way as the older versions: resolved by simply setting the **(i): USER PREFERENCES** menu in the right way. Newer versions of Blender (such as version 2.48) can be made compatible with older tutorials by changing some important settings:

- 1) Find the "**(i) : User Preferences**" window
- 2) Select the **[EDIT METHODS]** tab (find it by dragging down the user preferences window)
- 3) Under the header "**Add new objects**", click the "**Aligned to View**" button.
- 4) *Optional:* For some tutorials it may also be helpful to click on the "**Switch to Edit Mode**" button.

Making these simple changes will "unbreak" tutorials written under Blender version 2.43, by allowing new objects to be automatically oriented to whatever viewscreen orientation is selected in the active viewscreen. Any time object rotations, lattices or whatever else end up completely out of alignment with what older tutorials say should happen, these steps are your first best fix for almost every such situation.

### User contributed edit:

What actually happens by default on newer versions of Blender is that the axis of rotation is perpendicular to the screen. It means that, instead of revolving around the vertical axis, the object will revolve in the plane of the screen. Another way to deal with this is to change the view just before performing the rotation (I used **NUM1** view) and come back to **NUM7** once done.

## Final touches

We'll now extrude the hat front to make an eyeshade:

- use the top view (**NUM7**),
- choose *Edge select mode* (**CTRL+TAB**),
- then select the four frontmost edges of the hat (**RMB** for the first one, and **SHIFT+RMB** for subsequent ones),
- finally extrude the selection (**EKEY** → *Region*): drag them down; you can press the **YKEY** to limit the extrusion to this very direction.

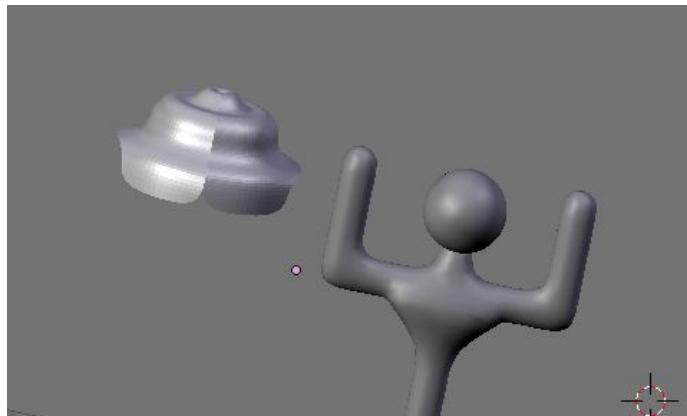
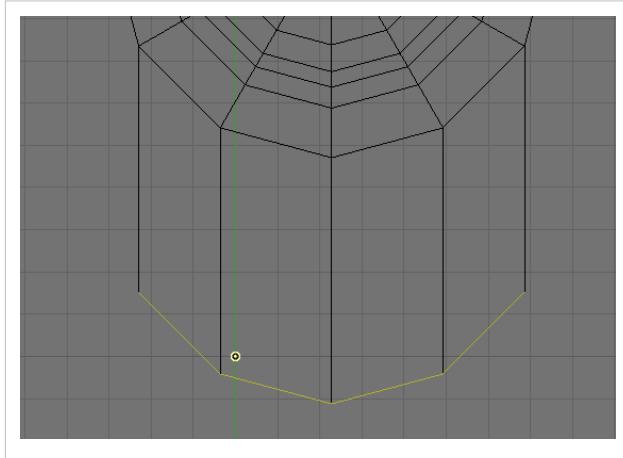
## Now it's time to subsurf

In the *Editing* buttons, from the *Modifiers* panel

choose: *Add Modifier* → *Subsurf*. Rotate the view around and you will notice that your hat has a "split at the seam". Because of the *Spin* tools options we chose, there are several pairs of vertices that share the exact same spot in 3D space which produce those subdivision seams. To solve this issue: in *Edit Mode* select all vertices (**AKEY**) then choose **WKEY** → *Remove Doubles*.

Now all our seams will display correctly, since you've removed the unnecessary overlapping vertices in the mesh. Whew! You now have a lovely new hat! Pat yourself on the back, good work! You canneaten it up a little more by hitting **WKEY** → *Set Smooth* to give it a nice smooth finish.

If you do not remove doubles, your hat will look like this:



Keep this simple person/simple hat file open because you'll need it in the next tutorial.

# Blender 3D: Noob to Pro/Putting Hat on Person

---

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 <sup>[1]</sup>.

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**.

If you press the **ZKEY**, **YKEY**, or **XKEY** just once, you will rotate the object around the global X, Y, or Z axis of the scene. If you press the same key a second time, it will rotate around its local axis instead. This will only make a difference if the object has been rotated before, because initially its local axis will be the same as the global one. So rotate an object a few times to see the difference between local and global rotation. This effect is most dramatic if you select multiple rotated objects and try to rotate them around their individual local axis together.

### User contributed edit:

*With Blender 2.49b (and possibly other versions?) repeatedly pressing an axis constraint key as described above will in fact toggle through 3 modes of constraint. **RKEY** will start rotate mode around the view axis. The first press of a constraint key places rotation in global mode and the second press places rotation in local mode. A third press returns you back to view axis rotation.*

Also keep in mind that you can select which pivot point to rotate around. If you did the previous exercises it is probably set to the 3D cursor. If so, set it back to Median Point.

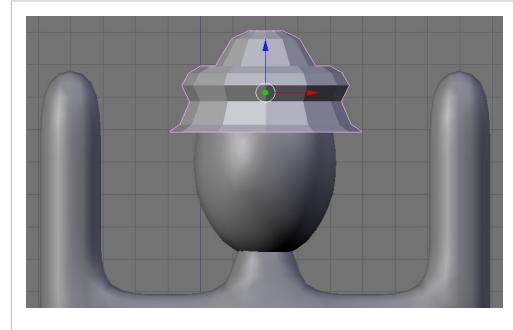
Important to note is that the shape will rotate around its origin, or center point, indicated by a small, pink dot that was created when you created the shape. It should be in the center of your vertices, but if it isn't, there are a couple of ways to get it back. One is to go into edit mode, select 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*. The final method is to click on the Editing button (**F9**) and click either the **Center** or **Center New** button in the Mesh panel. The **Center** button will automatically move the object's vertices to the dot, and the **Center New** button will move the dot to the center of your vertices.

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.

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 when you move the hat 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.

# Blender 3D: Noob to Pro/Bones

**Bones are used for shifting models and making them posable. If you are not ready for this yet and wish to continue simply modeling, please skip this tutorial to the next section.**

Bones are a modeling tool that are especially important for animating characters. 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 at 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!

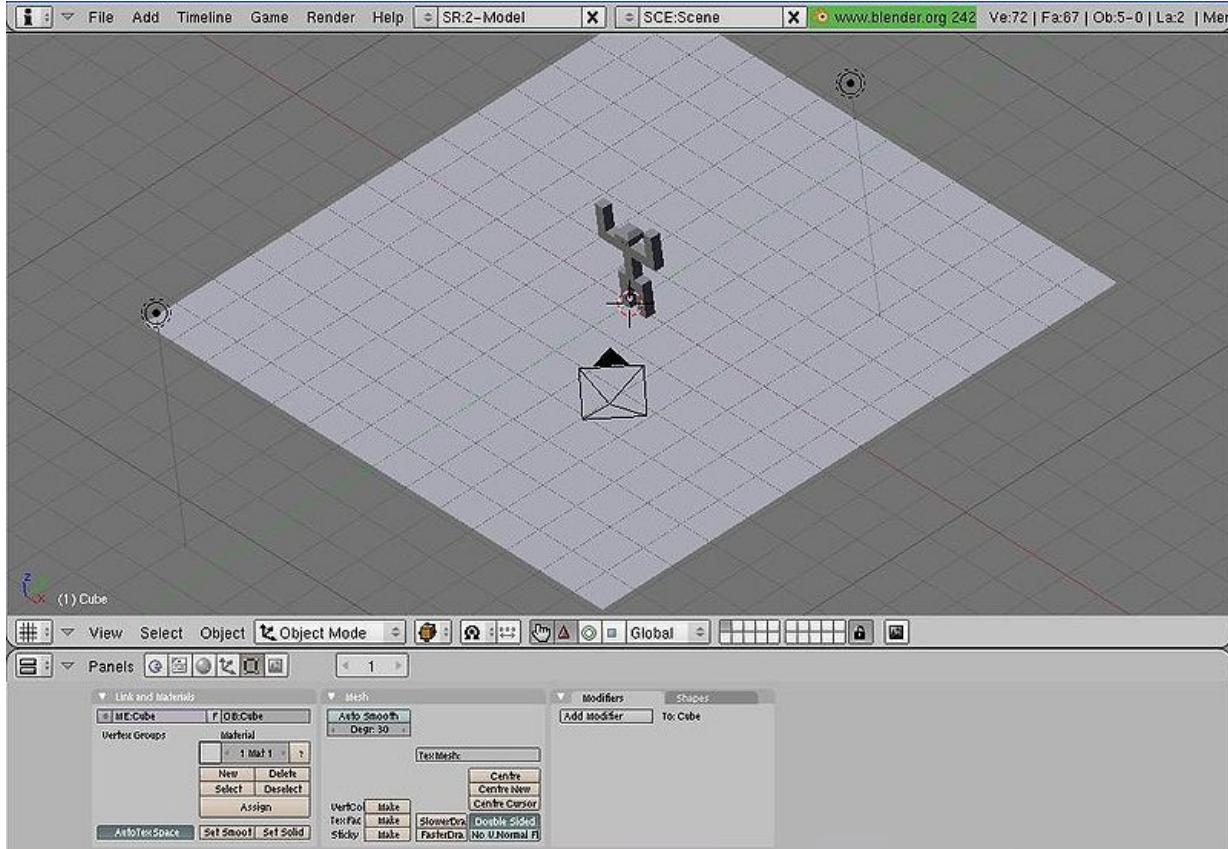
*For this tutorial, we're going to use a model with human proportions, but bones can be used with just about any body type. The same idea can be applied to cats, spiders or whatever!*

## Laying down bones

*Note: This just shows the basics of adding bones to an object. Go to the advanced animation page for a more comprehensive guide on this.*

First of all, we'll need a model to put some bones on! For this tutorial, we're going to use a humanoid model. I'm using a quick model that I made based off of the Blender 3D: Noob to Pro/Modeling a Simple Person first tutorial. It's rather blocky, but this isn't a detailed tutorial. However, you can find a model pre-made just for you here<sup>[1]</sup>.

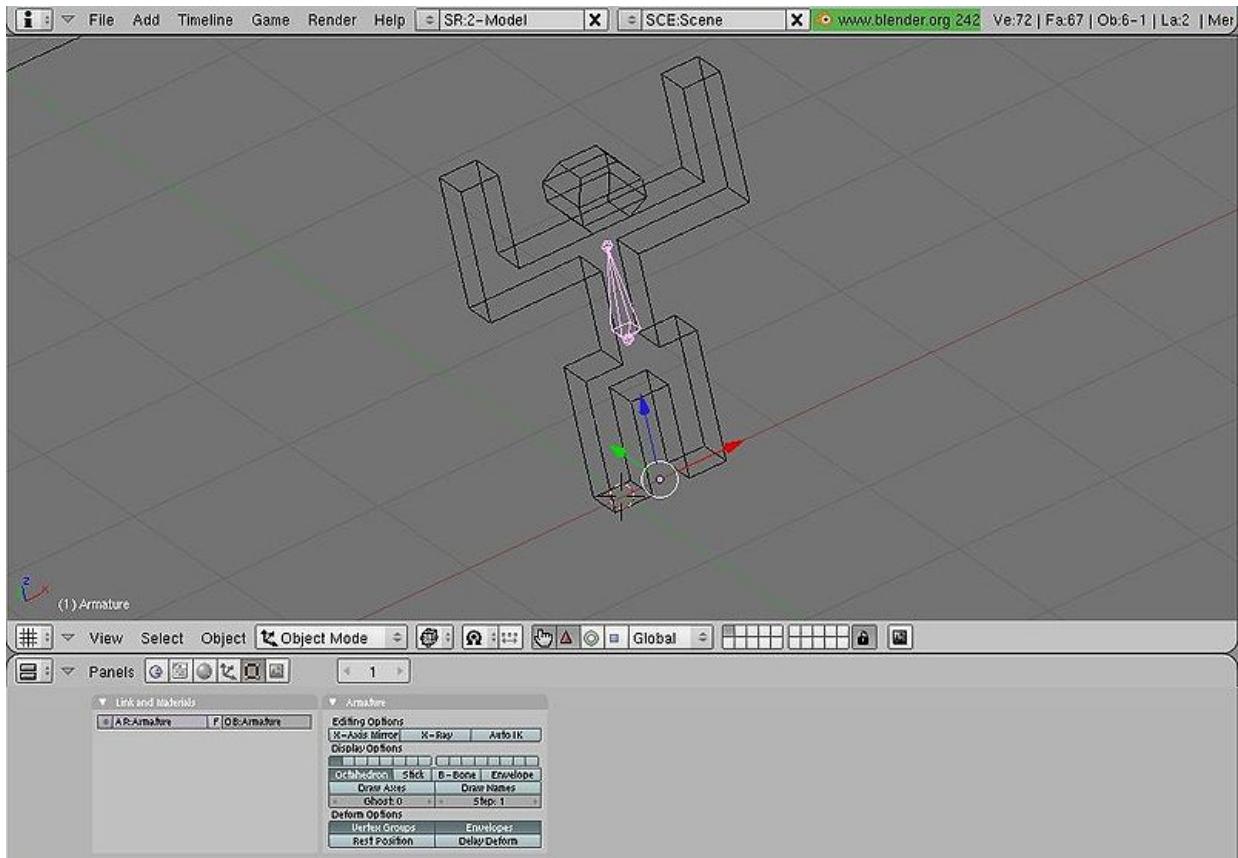
Okay, first of all, here's our setup, with Block Dude standing on the plane.



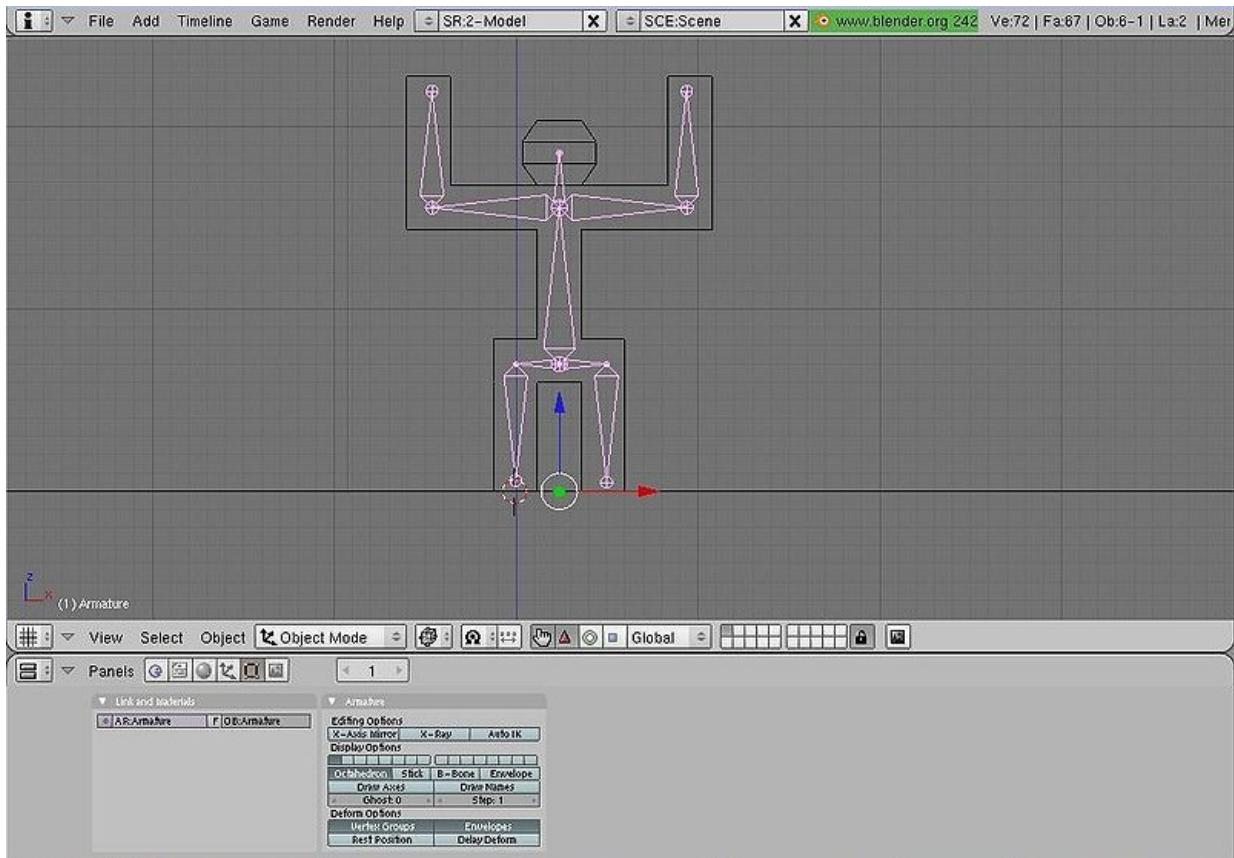
Now, let's put some bones on Block Dude! In Object Mode go to **Space Bar -> Add -> Armature**.



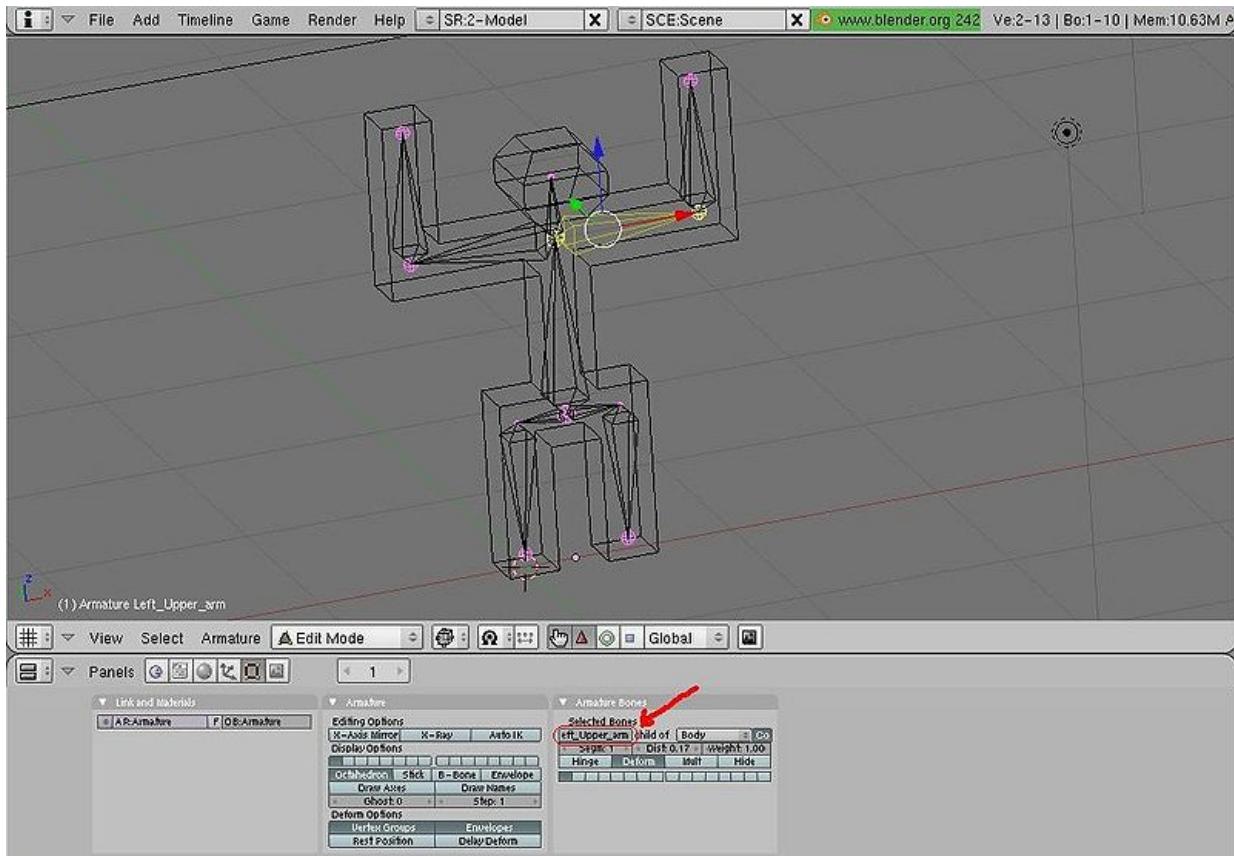
What we are looking at is an armature. This is a single bone. Now, we need to put the bone in Block Dude! Move and rotate the bone so that it's in the middle of Block Dude's chest. If your bone does not have the correct length, then change the size of the bone by moving one of the ends of the bone : switch to Edit Mode, select one of the ends of the bone, then move it using **GKEY** (You might be tempted to scale a bone, but this will mess things up). You might need to toggle wireframe mode if you have solid mode enabled - simply press **ZKEY** to toggle between them.



To create a second bone starting from one of the ends of the first bone, make sure you are in Edit Mode, select the end of the first bone, then press **EKEY**. The second bone appears, with its start point on the selected end of the first bone. Move the mouse to position the end point, then press **LMB**, **ENTER**, or **SPACE**. Now, extrude (**EKEY**) and scale the bone as needed to put the bones in his body! Make sure that you are in Edit mode, and click the pink dots at the end of the bone to do things like extrusion. These operate much the same way as vertices, you can extrude, rotate, move, and even subdivide. Your finished result should look something like this:



Now, just to make things easier, we're going to name the bones. For example, my bones are named RT\_Forearm, Left\_Forearm, RT\_Upper\_Arm, etc. Select the bone and press **F9** to display the Editing panel. In the "Armature Bones" sub-panel click the top left box to edit the name, indicated with a red arrow in the screenshot below. (noob note: when you are naming the bones remember that if you are looking at the person from the front, your left is the person's right. To make the naming easier switch to viewing the person from behind using **CTRL+NUM1**.)



Now, we need to parent the bones to the mesh. Go back into Object Mode and select Box Dude. Now, select the Armature as well, and press **CTRL+PKEY** (noob note: the selection order is important in defining which object is the parent, so you cannot select both objects at the same time). A menu will pop up, select **Armature**, then **Create From Bone Heat (Create From Closest Bones** before blender 2.46). Select the armature, and enter Pose Mode (**CTRL+TAB**) (you need to select the armature for this option to appear!). Try moving a bone around. If you've done everything correctly, your mesh should move when you move the bones! If this doesn't happen, scale the bones up so that they fit better in the mesh, and scale up the bones until they do what you want (read comment below on adjusting the bones envelopes if you do not get an effect while moving/rotating the bones). With the bones now, you can put Box Dude into a lot of different positions without moving individual vertices.

*Note: **Create From Bone Heat** creates vertex groups within the parented object that are associated with armature bones. This is done automatically, according to the bones' position and size. Alternatively, choose **Name Groups** instead of **Create From Bone Heat** if you don't want Blender to assign groups automatically, and do it manually. To manually change vertex groups go to Edit Mode (**Edit Mode, F9, Links and Materials tab**) and use the **Assign** and **Remove** buttons. 'Assign' adds vertices to the group (without removing any members that are not currently selected) and 'Remove' obviously removes them. Vertices can be assigned to multiple groups.*

*In response to the previous comment, if the vertices are properly assigned to the bones they will move regardless of whether the bones are inside the cylinder or not (HOW they deform WILL be affected however). I'm guessing that your mistake was creating and (more importantly) parenting the cylinder to the armature while it was outside the cylinder, which caused Blender not to assign vertices to any bones at all. You can check this by editing the cylinder, selecting a vertex group in the **Links and Materials tab**, and pressing **select**. This will highlight the vertices associated with the bone. If none are selected, it means none were assigned in the first place - in which case you need to assign them manually as explained above.*

*If there is no effect, select in Edit or Pose mode that bone (or bones) and choose Envelope display mode (F9 -> Armature -> Envelope), then press Alt+S and increase its area of influence to cover all faces that should be influenced by the bone.*

Finally, here's an example of how you can move Box Dude with the bones:

Noob: can someone fill in how to move individual bones?

I can move the complete set of bones in object mode, and the man follows.

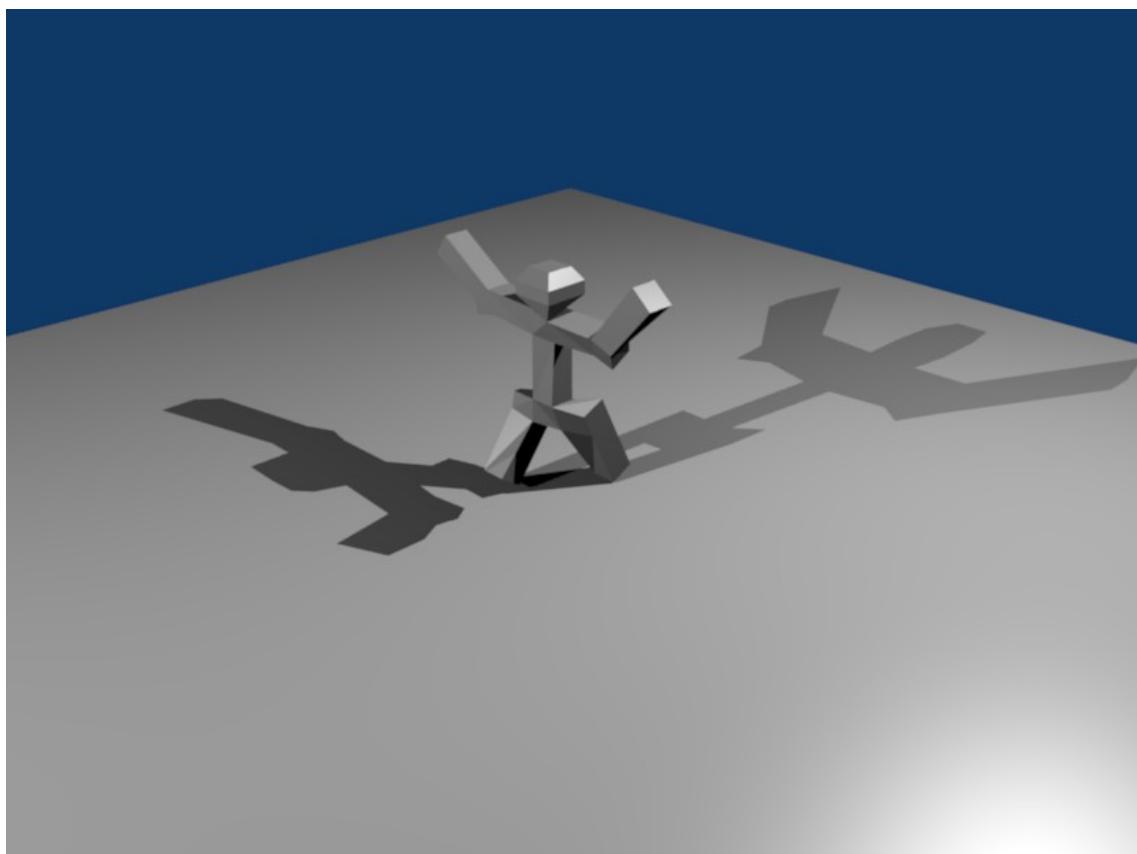
I can't select a single bone by itself in object mode.

I can select a single bone in edit mode, but then the man does not follow any changes.

Select armature in object mode then switch to pose mode mate.

Noob2: I was in pose mode and bones wouldn't rotate or move. After lots of hair pulling,

I found out the cause: the "Move Object Centers Only" button (just to the right of the Rotation/Scaling Pivot button).



# Blender 3D: Noob to Pro/Mountains Out Of Molehills

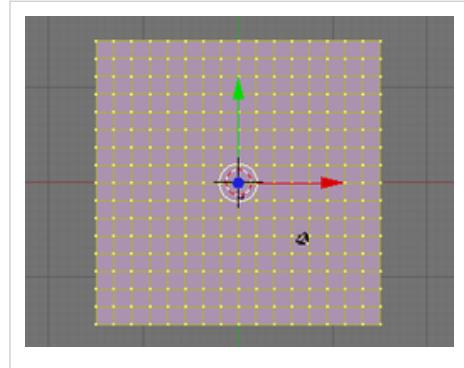
Now that we've created our simple person, it's time to give him somewhere to go. In this tutorial we'll create a mountain range using a few simple, and handy tools. First we need a clean area to work with. Either:

- 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**).
- or you may change to a new layer by pressing a number from 2 to 0 on your keyboard or by selecting one of the twenty little gray boxes grouped together in the header of the 3D window.
- Pro tips for layers: 1-0 selects layers 1-10. ALT1-ALT0 selects layers 11-20. Hold shift to de/select multiple layers.

## 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.

- Press on **NUM7** to enter top view. This way our plane will be lying flat when we create it.
- Press **Shift + C**. This will be the center of the plane we will add.
- 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 simply type **15** and hit **ENTER**.
- 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).



**[Noob Note:** You also seem to need to go into edit mode before 'Mesh Tools' shows up as a tab under Edit - at least in 2.4.6 and later]

- Under *Mesh Tools* hit the *Subdivide* button 4 times. Alternatively, in the 3D View window you can press **WKEY** and select **Subdivide**

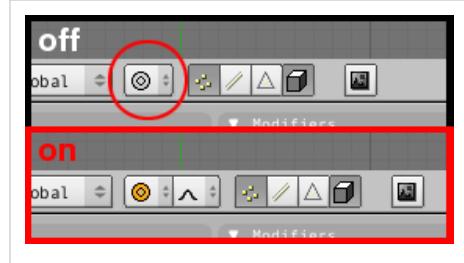
(Or just hit **ENTER**, because *Subdivide* is the first option under **WKEY**).

**Note:** As in many of the next tutorials, you may choose to add a grid instead of adding a plane and then subdividing it to have vertices throughout the plane. Adding a grid has the added advantage of allowing you to select the number of vertices along the x and y axes (the two edges) of the grid at the instant of creating the grid, so there is no need to make further subdivisions.

## First mountain

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

- Make sure you have nothing selected (**AKEY**).
- Select a random vertex with **RMB**. I usually start at the one that is 4 down from the top, 4 in from the left (the 4th vertex 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.

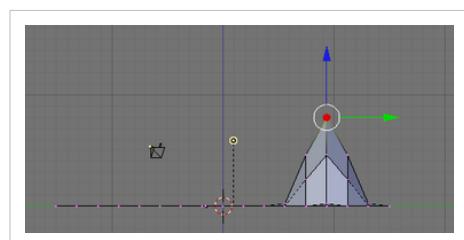


**Noob Note:** If you're having trouble seeing or changing the radius of influence, try saving your scene and restarting Blender.

- 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+ / PAGE\_UPKEY** and **ALT + NUM- / PAGE\_DOWNKEY** and may need to hold down the **LMB** while using the **Mouse Wheel** to adjust the radius of the influence. On Mac, hold the "fnKey" down and hit "page-up" or "page-down"). In 2.41 you must 'grab' the vertex first - only then can you alter the sphere of influence (in my version, 2.42a, the 3D cursor had to be snapped to the selection before the wheel appeared).

**Noob Note:** To change the radius on a Macbook press "fn"+ up/down. OSX uses function + arrow keys as replacements for windows "home", "end", "page up" and "page down" keys.

- 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.

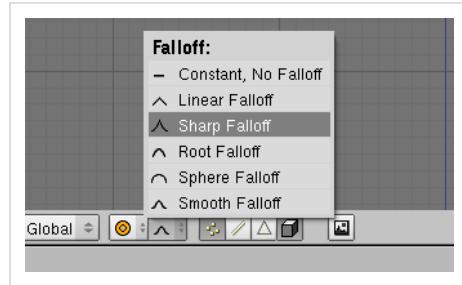


Congratulations, we just created our first mountain. Now it's time to see what other things we can accomplish with the proportional editing tool.

## Peaks vs. hills

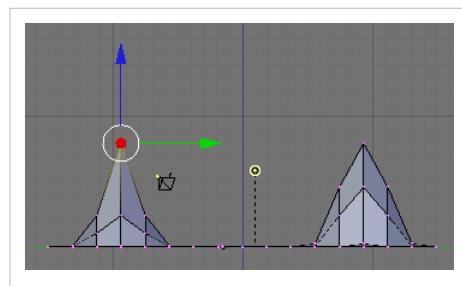
The 2.37 and onward releases offer at least 6 types and 2 modes of proportional editing. The previous release only has 2 of these types: Smooth and Sharp Falloff. We'll take a look at the difference between these two now.

- Change to top view again with **NUM7**. You'll notice that now your "mountain" looks like a few differently shaded squares in the grid; you're looking down on shaded tiles, but in the Z axis, they're all still perfectly aligned with the original grid.
- Select another vertex away from the first. Let's say 4 from the bottom 4 from the right (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 different.

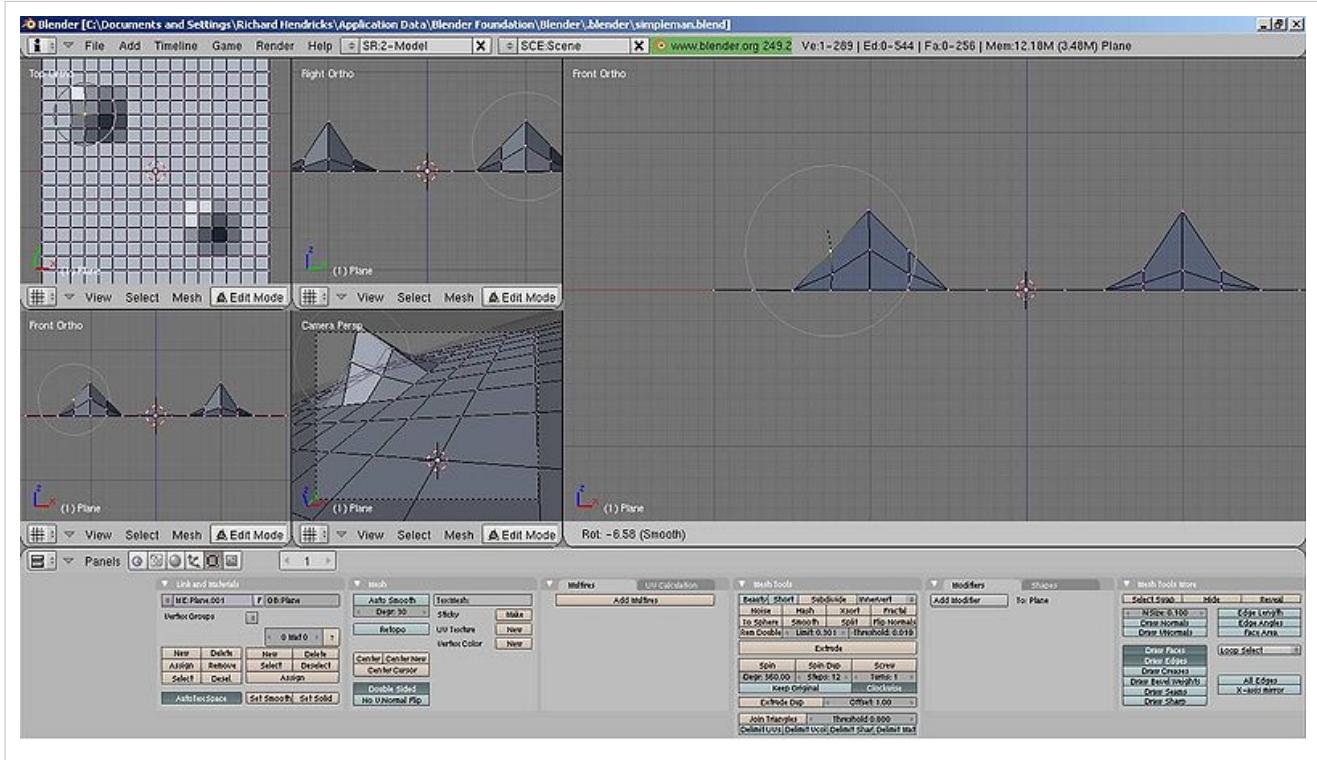
The different proportional editing modes can be selected from the box immediately to the left of the proportional editing type box. The mode box contains three options: Off, On, and Connected. "Off" means that proportional editing will not be used. "Connected" means that only vertices linked to the selected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



## Shaping the world

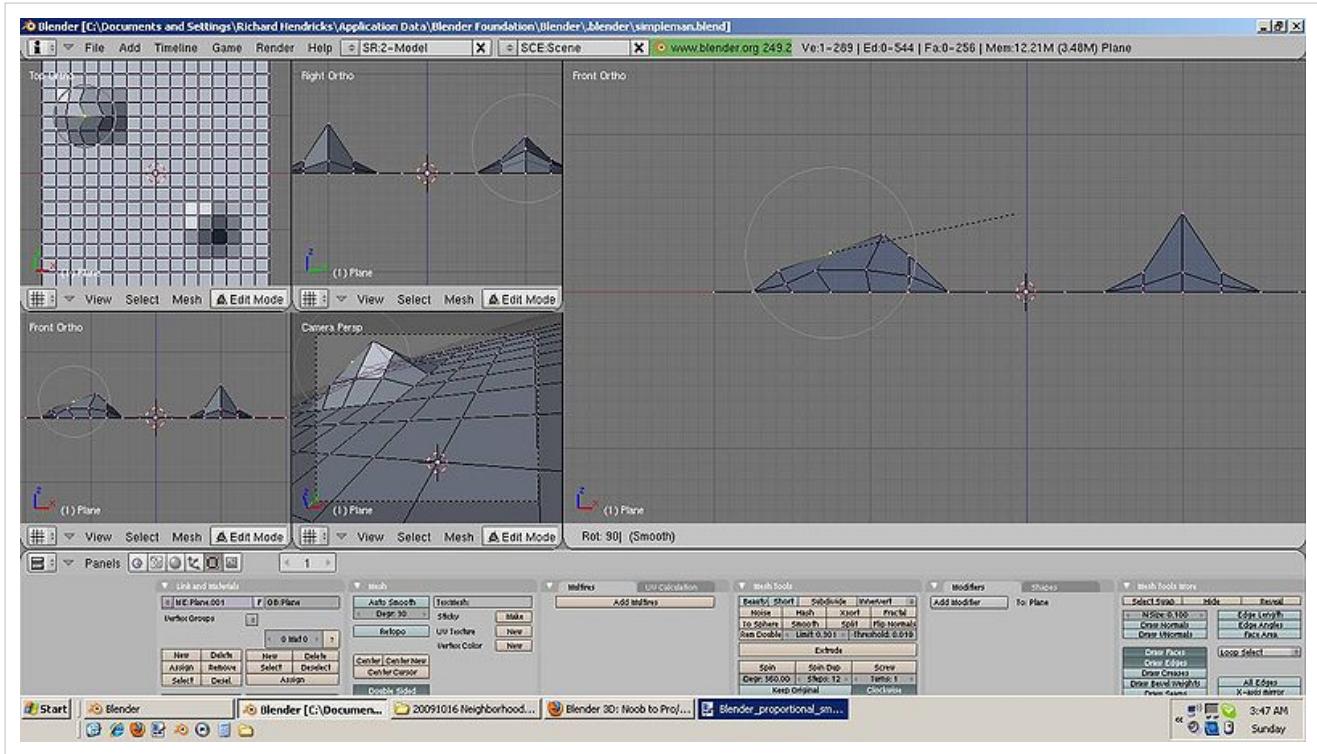
Now that we've created a couple of Mountains, it's time to see how we can use proportional editing to shape them.

- 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, scroll the **MMB** to change effective radius so it includes other points. Your screen should look like this:



You can see the size of the proportional editing circle, and that there is only one vertex on the mountainside selected.

- Next hold **CTRL** and rotate everything by -90. Alternatively, use **RKEY**, **NKEY**, and type **-90** and press **ENTER**. Your mountain should now look like this:



Notice that the vertex itself did not move; since it is at the center of the circle it had no effect. The adjoining vertecies within the edit circle were rotated around it in decreasing amounts the further from the center they are. Try doing it again with a larger proportional editing circle. Feel free to play around with scaling or rotating from different view points (don't forget that you can also use **GKEY** to move vertecies vertically or horizontally).

[Noobie Roy says: Huh??? I don't understand what I'm supposed to be achieving here. What is supposed to happen when I rotate? As far as I can see, nothing at all happens. Can you maybe show us a before and after screenshot? Or maybe explain in more detail? Thanks!] [Noobie Sean says: Roy, you made the mountains by using proportional editing with grab/move. Moving one vertex moved its neighbours, within the proportional circle. Rotate and scale also use proportional editing to rotate or scale about the selected vertex much like the grab/move operation.] [noobie larry says i tried rotating just that one point but nothing happens] [One thing to watch out for is the range of affected vertices. If the range is too small, then rotating will affect just the selected vertex. If the range is too large, it will rotate everything together. You can adjust the range by scrolling.]

[Dusty says: To answer Noobie Roy, the reason nothing may happen, is that the circle of influence may be only including the vertex that you have selected when you first hit R. After pressing R scroll the mouse wheel to increase the circle include the other vertices next to the one selected, then type 90 and you'll see the effect. Scroll the mouse wheel further more (to include more vertices) and you see the effect spread out.]

[Another newbie says: two things to keep in mind here: really weird things will happen here if you set the rotation/scaling pivot to "3d cursor" at this point (I assume the intention is to rotate around the single selected vertex, so the other pivot modes will all work, I think), and (as Dusty says) you must increase the effective radius quite a bit (use the scrollwheel after pressing **RKEY**, just keep going until you are affecting half the plane to see what's going on). What makes it a bit confusing is that some vertices seem to be inside the initial affected region (with the radius you used to create the previous mountain) in the two-dimensional side view, but that affected region is actually a three-dimensional **sphere**, so you cannot tell from just that side view what vertices are inside the affected region. I am not explaining this very well, but it helps to have more than one 3d view open so you can see that every vertex is outside the affected radius in at least one view (try splitting the main view into 4 equally sized parts so you can have a top, front, side and "free" 3d view open all at the same time).] [Yet another noob says: if you are following the tutorial exactly, your proportional edit falloff will be at sharp at this point. This makes the effect of the rotation hard to see unless your circle of influence is quite large.]

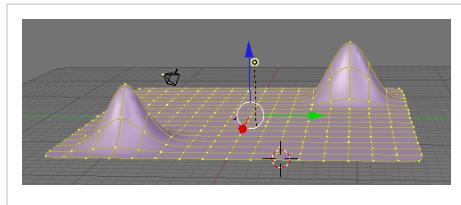
[Dbproguy mentions: From what I am seeing, this effects the texture of the mountain, so it's not so plain.]

[hendric: Newbie, but I think I got the tutorial demonstrating what it should be doing. I added screenshots of my full workspace and changed the wording to explain a little better. I am wiki-illiterate so any changes to prettify it are welcome.]

Try viewing your world from top view while rotating with a large effective radius. You will see the nearby vertexes move close to the full amount while vertexes further away move less.

## Smoothing things out

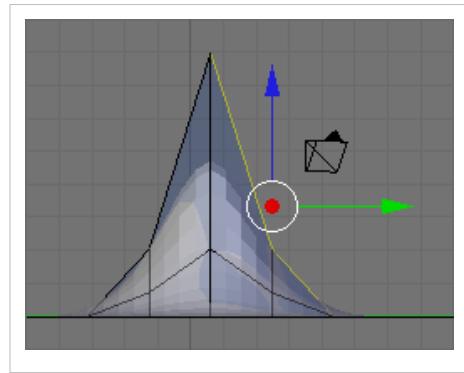
Now that we have a couple of budding mountains, you probably think they look kind of choppy. Sure they would be good if we were making an 8-bit console game, but we're working with 3D here, we want things to look sharper (or maybe smoother) than that. There are a couple of approaches to this. The first 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 (*Levels*) 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, essentially we have no sharp corners any more. 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**
  - *Noob note: That means, the letter O key, not the zero key. Also, NUM3 means "3 on the numeric keypad," not on the top row of the keyboard.*
- 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* (*Mesh Tools More* in versions 2.46 and later) ensure that *Draw Creases* is selected. (*Mesh Tools 1* may be off the screen, if so, use the scroll wheel when over the Edit Panel to reveal it. Alternatively, you can pan the buttons window by dragging with the **MMB**; or you can collapse some of the panels by clicking the arrow next to their names.)
  - *Noob note: "Mesh Tools 1" is a separate panel from "Mesh Tools" and from "Mesh". By moving the mouse cursor over the buttons window (remember this 'activates' windows) and using the mouse wheel, you will cause the tabs to scroll from side to side, revealing some you couldn't see before. If the one you need still isn't there you should check to make sure you're still in edit mode, or make sure it isn't 'minimized' which is done by clicking the little white arrow at the top left of the tab.*
- On our Sharp Falloff mountain, the second one we did, select the two edges on the right. (see image below)
- Press **SHIFT+EKEY** or **SPACE** → *Edit* → *Edges* → *Crease SubSurf*, then move the mouse away from the edge until the edge *Crease* reads 1.000 in the 3D viewport header. If moving the cursor there seems to be impossible, just hit 1 and enter.

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.



*Noob Henry: What I do is take the mountain, and then set a subsurf level of two. Then, I hit the 'set smooth' button, and use the individual vertices to make it look authentic.*

## Naturalness

Press **CTRL+TAB** and select vertices. then go into front view **NUM1**. Select the second vertex from the top in the centre of our Sharp Falloff mountain, then 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 select all, press **WKey** and 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 vertices that make up your mountains, 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 your models into very odd shapes indeed. Have a play with different values until you find one that you like. Around about 15-30 should do it. Hit OK and hey presto, your mountains have been transformed from clinical neatness, to lumpy chaos.

- If you make too many fractals, your computer will slow down. However, the more fractal you add, the more bumpy and realistic it looks!

*Noob Note: Repeatedly using the fractal tool seems to rapidly multiply the amount of vertices on your canvas. I suggest using the tool once, and if the result isn't satisfying, undo the result (**CTRL + Z**) and try it again with a different fractal strength. Even after undo, your selected vertices remain selected.*

*Noob Note #2: If you have a new enough version of Blender (2.49 and up definitely has it, not sure about lower versions), there is a setting called Random in the Proportional Editing settings. Using this with a high radius may give a desired result for you without adding more mesh.*

(Under Construction [TO DO: finish me])

## Adding your guy with the hat

If you changed to a new layer-- press (**SHIFT**) select the layer with your guy on it or (**SHIFT + 1**). This will display both layers. If you would like to move the landscape from the second layer to the first then with the landscape selected, press (**M**), and select the layer(s) you would like it on. note- make sure you're in object mode. If you started a new project-- To insert your guy with a hat, you can simply go to **file > append or link(shift+F1)**.

- Then select the file from the previous tutorial.
- You will then see a list of objects you can insert or 'append'.
- First go into the 'Object' folder. Unless you renamed him/her, your person will probably be named "cube." Select it.
- Make sure that 'Append' button is pressed; otherwise you will not be able to scale or translate, or edit your model at all. (The 'Link' button will link a copy of your object into the current scene, and will update any changes when you reload the file. Unfortunately, this includes location and size, so we will not use it right now.)
- Press 'Load Library' to place your guy into the mountain scene. Please note, the *Load Library* button is top right. Confusingly, there is some text bottom left that says *Load Library* as well, but isn't clickable.
- Repeat the process to get the hat (probably named "circle"). To reset the parental relationship, see the instructions at the end of the previous tutorial. To scale them to the appropriate size, go to object mode by pressing **TAB** (unless of course you are already there). Select the object you want to resize, then press the **SKEY** and use the cursor to scale as you see fit. To undo any mistakes, use **CTRL+Z**.

**Noob Note:** To import the file once you have the list select the item, i.e. cube, and in the bottom right select "Active Layer" that will append it to your current proj when you click "Load Library" your work should be visible now.

**Noob Note:** Multiple items may be appended simultaneously. For instance: both 'cube' and 'circle' objects can be appended at the same time by highlighting both object names within the file browser by means of a RMB Click. RMB on object again to deselect.

# Blender 3D: Noob to Pro/Modeling a Gingerbread Man

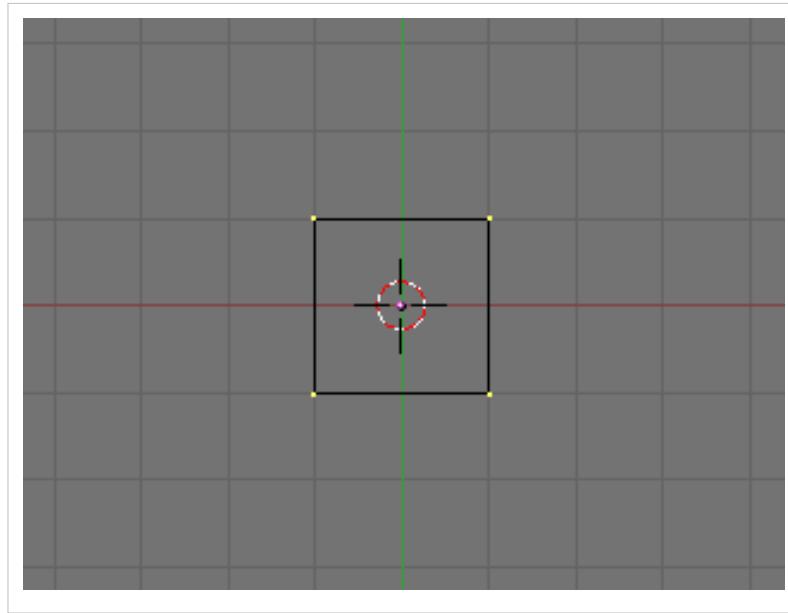
In this tutorial you will learn how to make a simple gingerbread man. In a later tutorial you will be able to make an animation with this gingerbread man.

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.

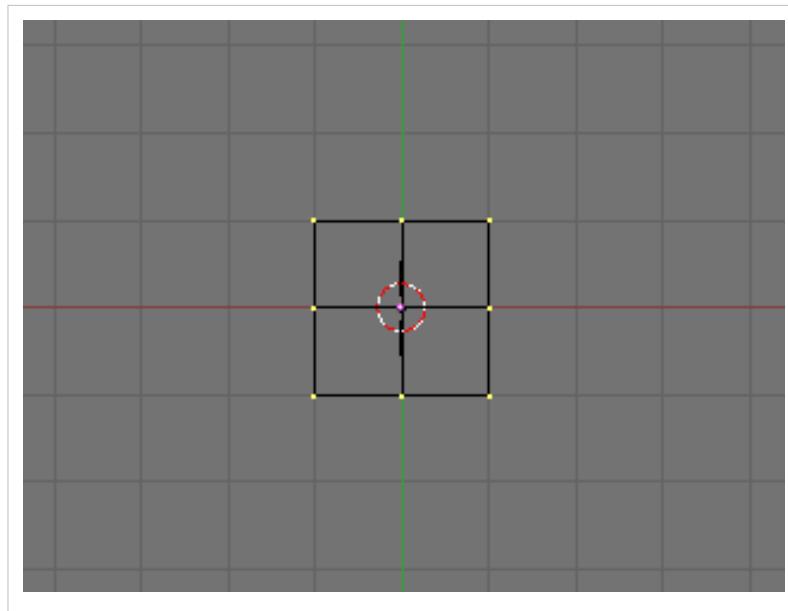
## Modelling

First, start Blender. You should see a cube in the 3D View. (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 pinkish 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.

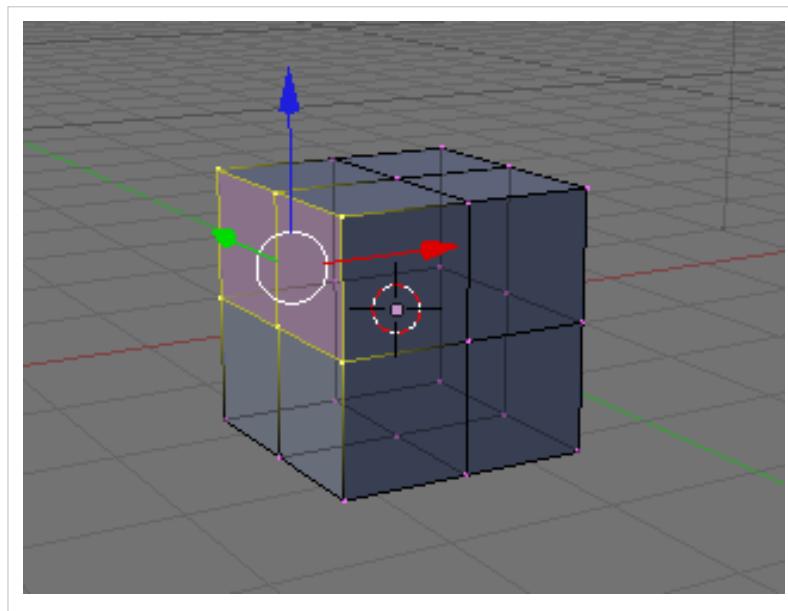


- Once you are there you will see a new menu at the bottom of the page, click on the subdivide button in the section called Mesh Tools (while all the vertices are selected). You will see that your cube now has more vertices. This tool is used for **dividing** an object so that you can do more complex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

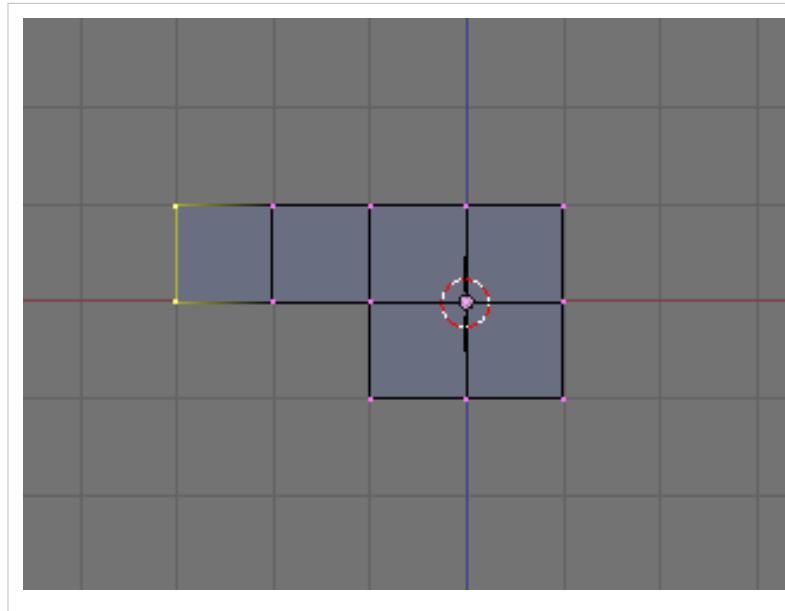


- Now press **AKEY** to unselect all the vertices, go to the front view (**NUM1**) and press **BKEY** and drag a square around the top left and middle left vertices or press **BKEY** twice and you will see a circle around your mouse - all the vertices in the circle will be selected by pushing **LMB**.
- 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 "Occlude Background Geometry" button ("Limit selection to visible" in Blender 2.45 and earlier) 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.

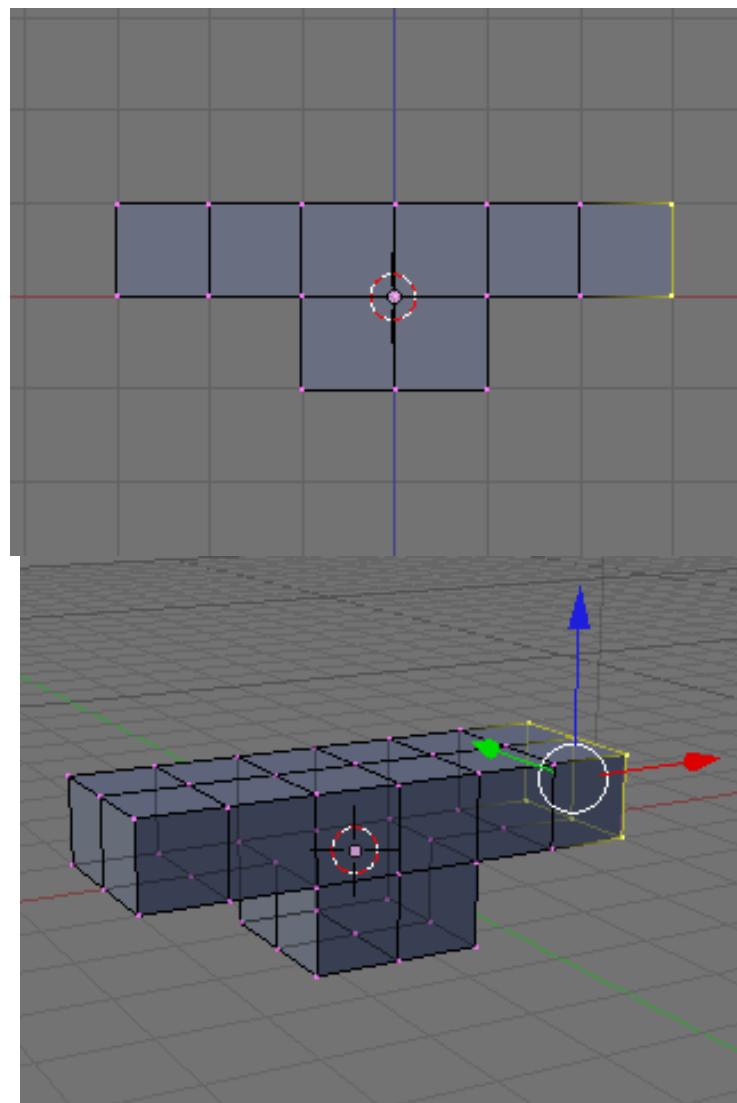


- After selecting the 6 vertices press **EKEY** and select Region. This will extrude the selected vertices. Put the new vertices on the adjacent gray line of the grid one unit to the left (press **CTRL** to snap to grid). Do this two times so that it looks like below (the snapshot has been taken in a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



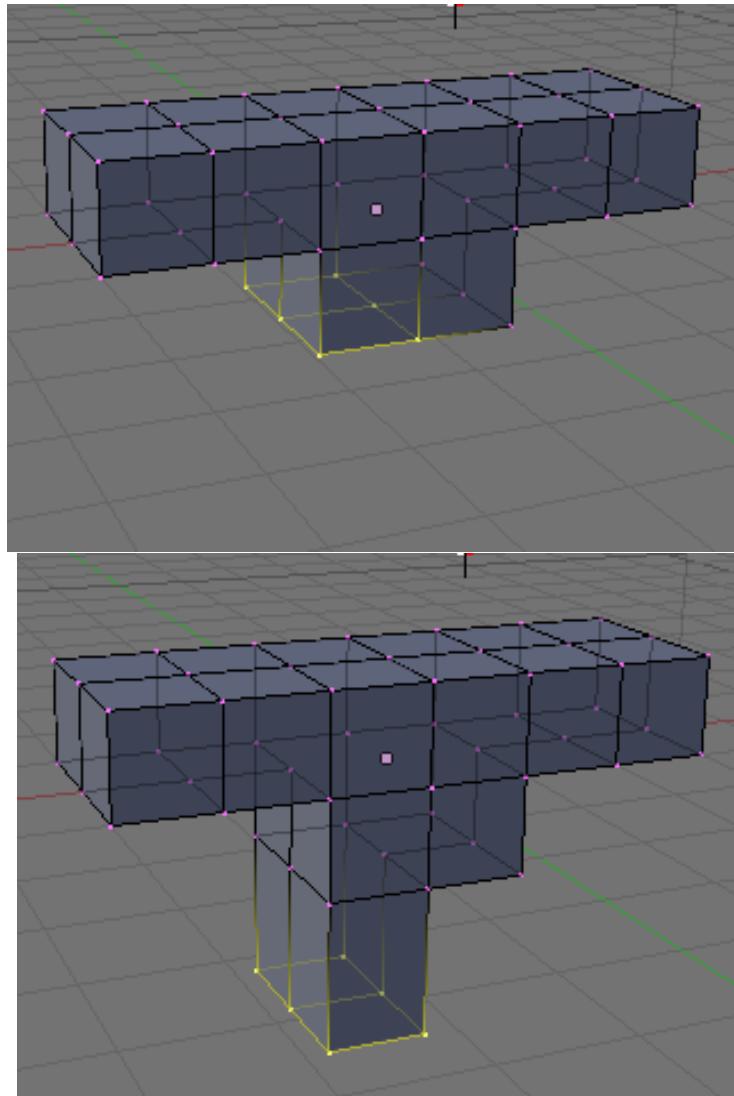
- Clear your selection (**AKEY**).
- Now select the other two vertices (six in 3D again) on the opposite side and do the same there as explained above.

Now the arms are complete, as you can see in the illustration below.

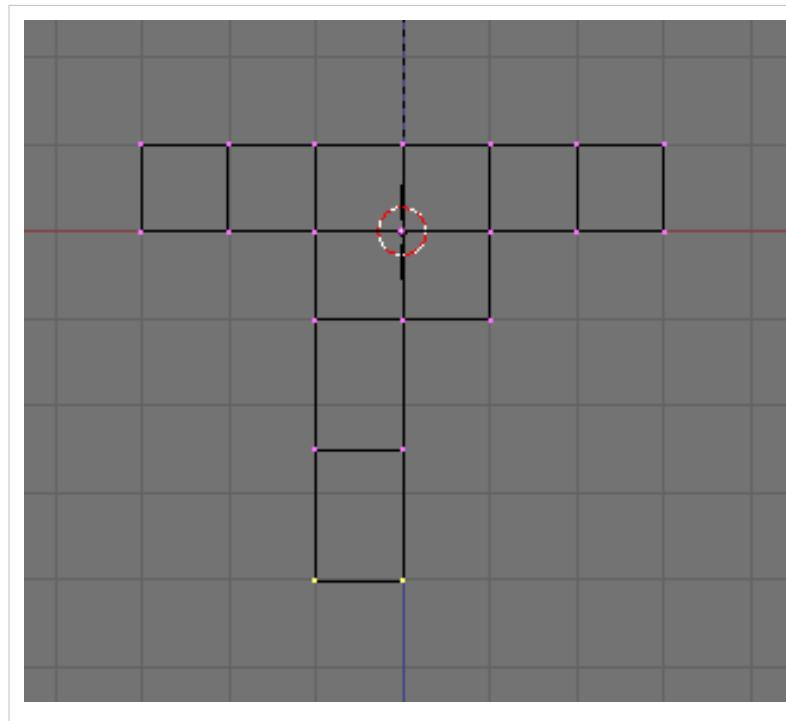


Now we will do the legs.

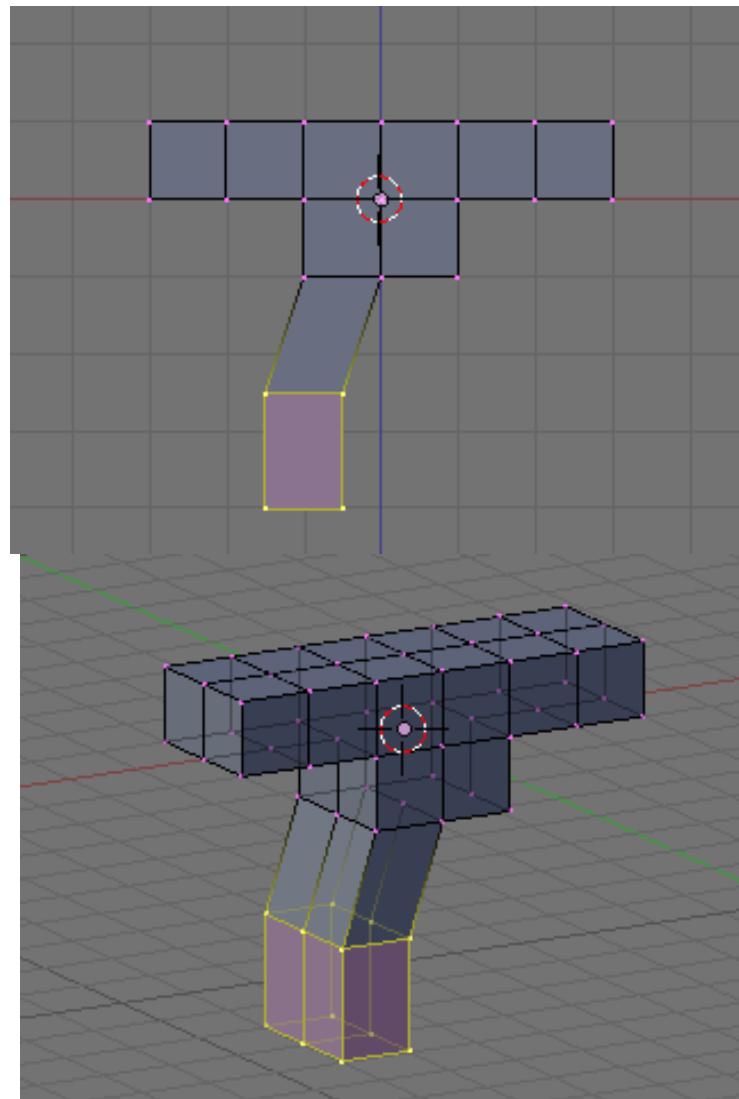
- First, unselect all the vertices : use **AKEY**.
- Select the bottom left two vertices, extrude them and put them in between the gray line (the gray lines in the grid representing the Blender units) and the second gray line below. If holding down **CTRL** you will notice that the two vertices snap 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. On Mac, enter the number 1, press fn with the key that is right under **LKEY** and **MKEY** on Azerty (the one with /;,), and press the number 5)



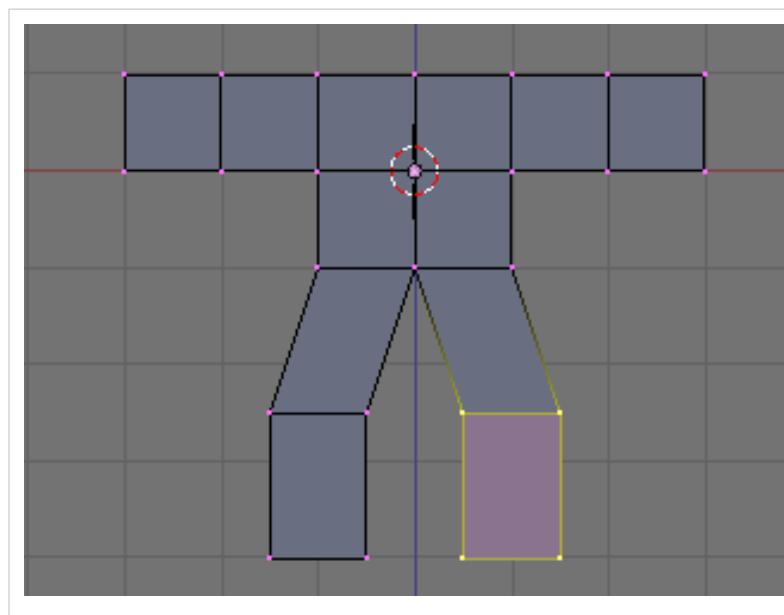
- Extrude it again and put it on the third gray line (or, once again, enter 1.5). It should now look like this:



- 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



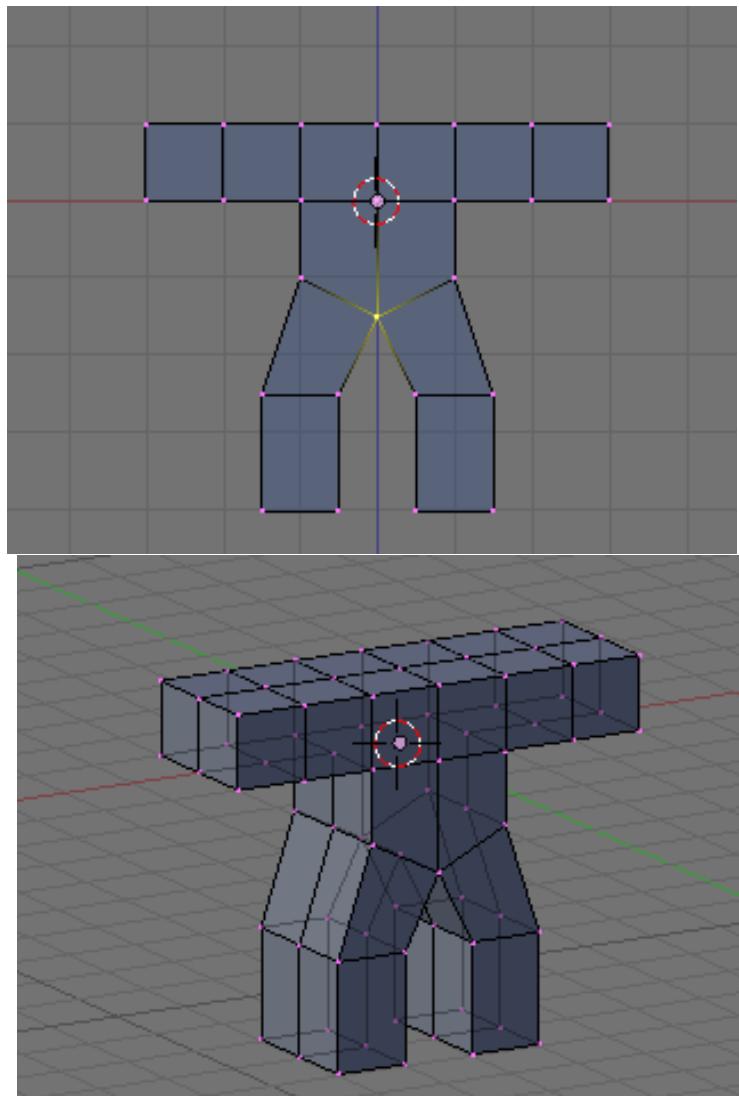
- Do this again for the right leg.



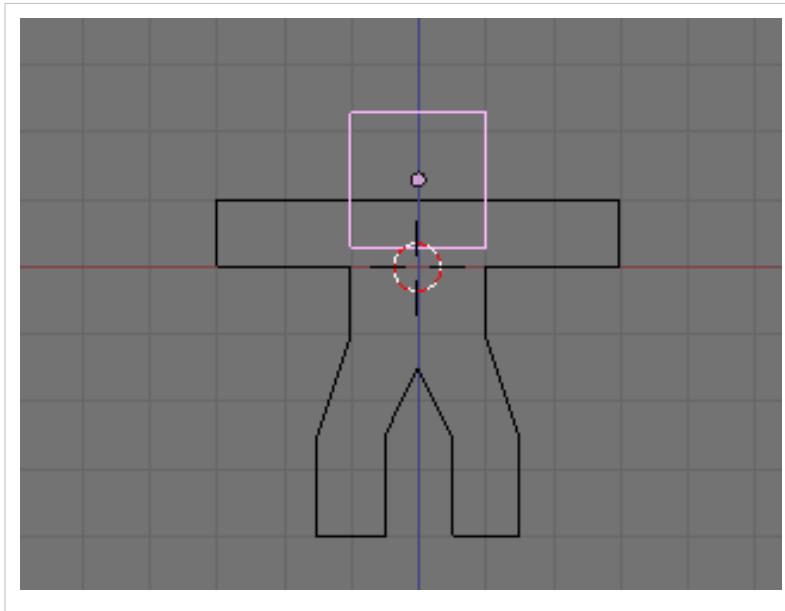
- Use **BKEY** to select the vertices at the groin (where the two legs join)

- 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 **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.)

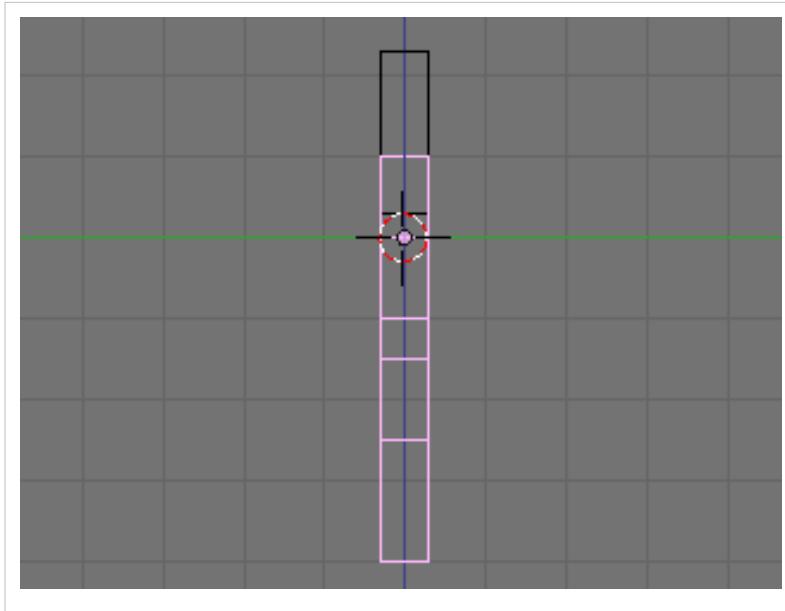


- Return to Object mode
- Click **RMB** on the object to select it then press **SHIFT + SKEY** and select *Cursor → Selection*. This will make sure the cube you'll add next will be near where you want it.
- Press **SPACE** and put your mouse on the mesh option and select cube. In others versions, you 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.

- Select all with **AKEY**.
- 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 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).



- 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.
- 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.

**Noob question:** When I add the cube for the head, it stops me from being able to edit the body - it will only select the head to apply subsurf to, even if the body looks like it's selected!

**Answer:** When you created the cube you made a second object. To select a different object, press tab to enter Object mode. Select the body. Then enter edit mode again if you want to edit the body.

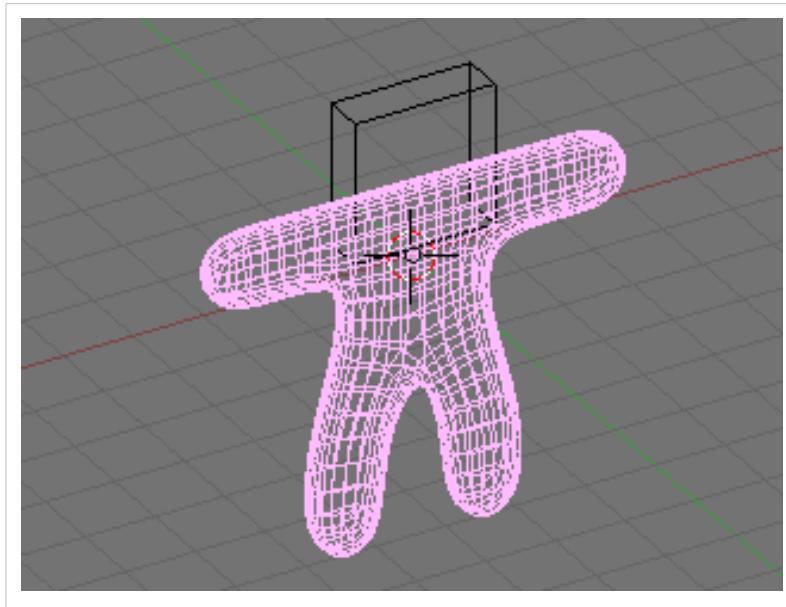
- You can press the **ZKEY** to switch back and forth between wire-frame view and solid view.
- (Noob Note: Easiest way to really get a feel for what is going on in the 3d world is to split into four screens and setting each one to **NUM7**, **NUM3**, **NUM1**, and **NUM0** to see all angles and what it will look like at render.)

**Noob Question: How?**

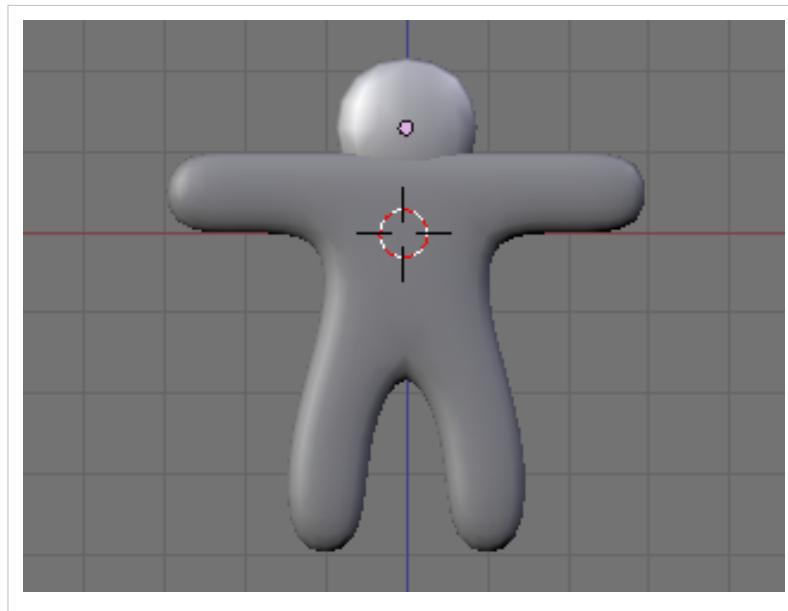
**Answer:** To split an area move the cursor to an area between two current areas (e.g. between the 3D view and the buttons), when you see the double ended arrow (used to move the divide) click RMB and select *Split Area*, you will then see a line appear dividing the area in two. Move this to where you want the divide and click LMB.

- In the 'Link and Materials' section, select 'Set Smooth'.

(Note that here I had the same problem as before, with superposed vertices. Select all vertices, then 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 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 frame 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.

If your render comes out a little dark, try moving the lamp closer to the gingerbread man.

Noob note: Another way to move around the camera is pressing **SHIFT + FKEY** after pressing **NUM0** to enter Fly mode. The keys for fly mode appear in the header of the 3D view pane.

Noob note: **Ctrl+Alt+NUM0** "teleports" the camera to your 3d view.



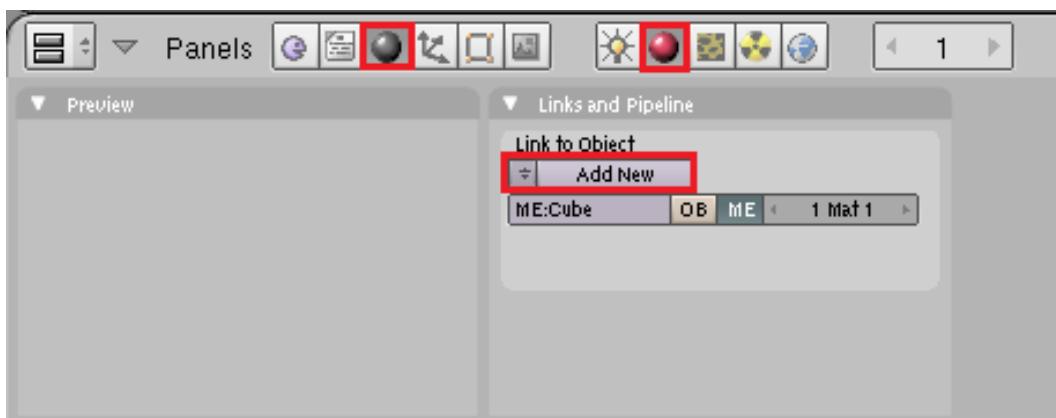
Noob Note: By pressing X, Y or Z twice you will use a local base of the space, with those it's much easier. For example if you are facing the Z axis from 45 degrees, and you want to go left 1 unit, using the global base, you will have to go  $1.72$  (around  $\sqrt{2}$ ) along X and the same along Y, instead moving by 1 in the local frame of reference.

## Applying Textures

This builds on the previous guide: Modeling a Volcano <sup>[1]</sup>.

Note: It seems that textures can only be applied to one object at a time, so this must be done twice (i.e. The head and body are two separate objects.) The settings that were chosen can successfully be applied to each object for a consistent result. Some settings can not be applied equally for consistent results.

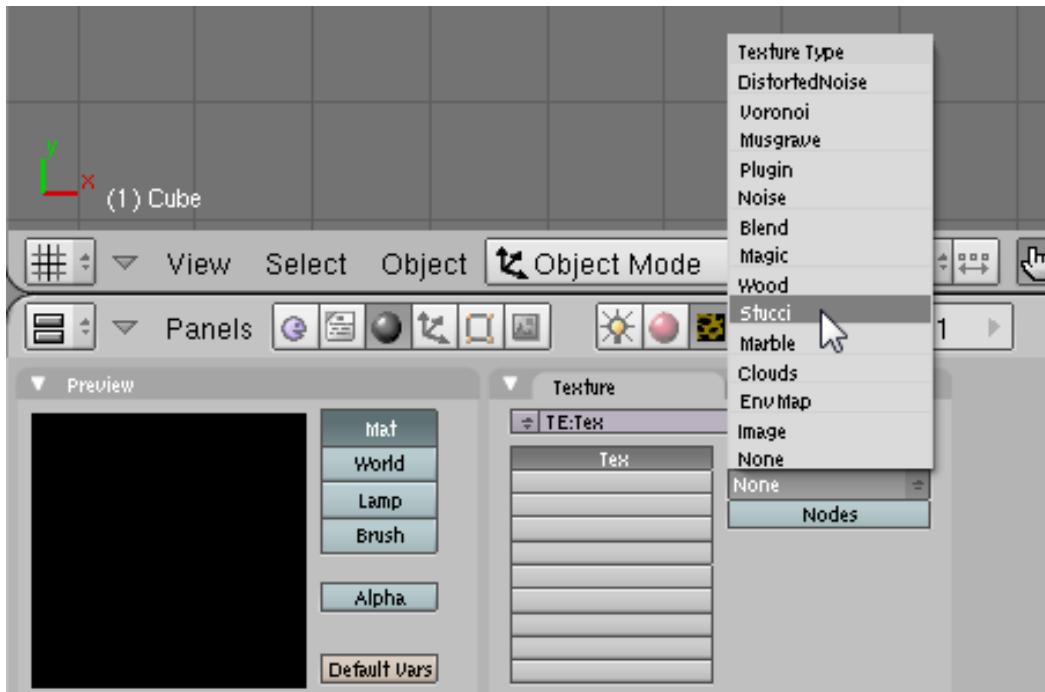
- In "Object Mode," select the body (or the head.)
- Press **F5** to open the shading panel or use the shading panel button.
- In the "Links and Pipeline" panel, under "Link to Object," click "Add New."



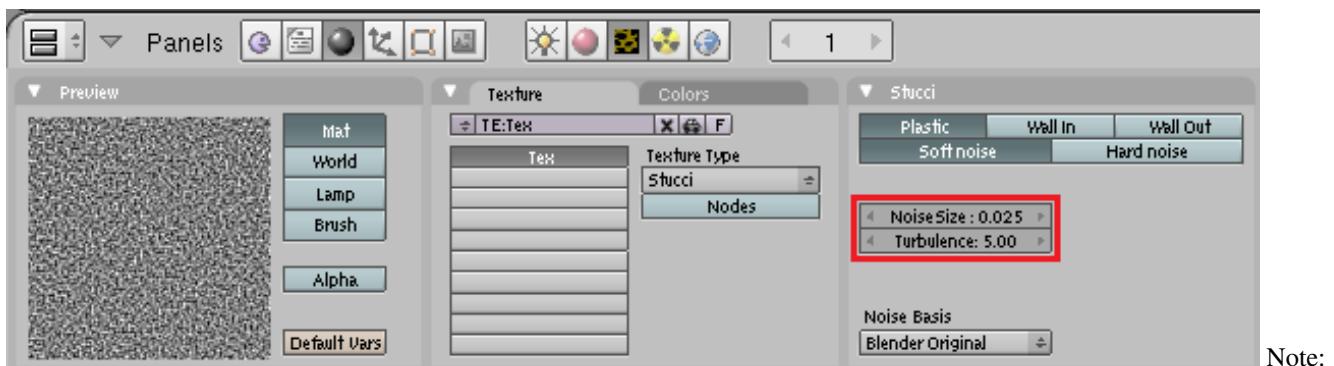
- Press **F6** to open the "Texture Buttons" panel or use the textures button.
- In the "Texture" panel, click "Add New."



- Change the "Texture Type" to "Stucci."



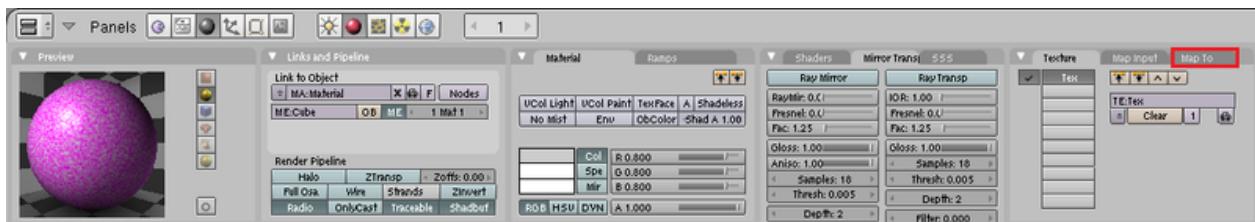
- In the new "Stucci" panel, change "Noise Size" to something near 0.025 and leave the "Turbulence" at 5.00.



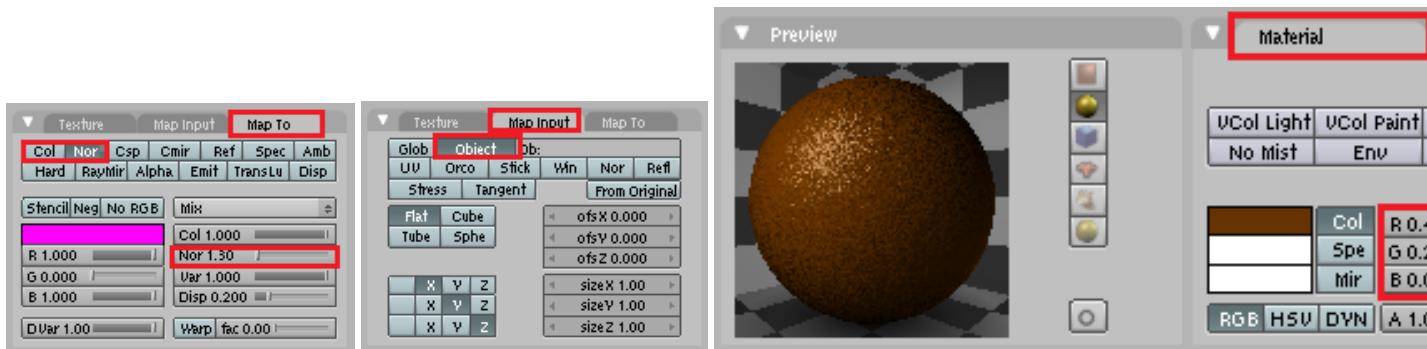
Note:

When finished with this section of the guide, come back to this panel and try different combinations of "Plastic," "Wall In," "Wall Out," and "Soft Noise" / "Hard Noise." Press **F12** to render after each change to see the effect.

- Press **F5** again or use the "Material" button directly on the left of the "Texture" button. Then look for the "Map To" panel.



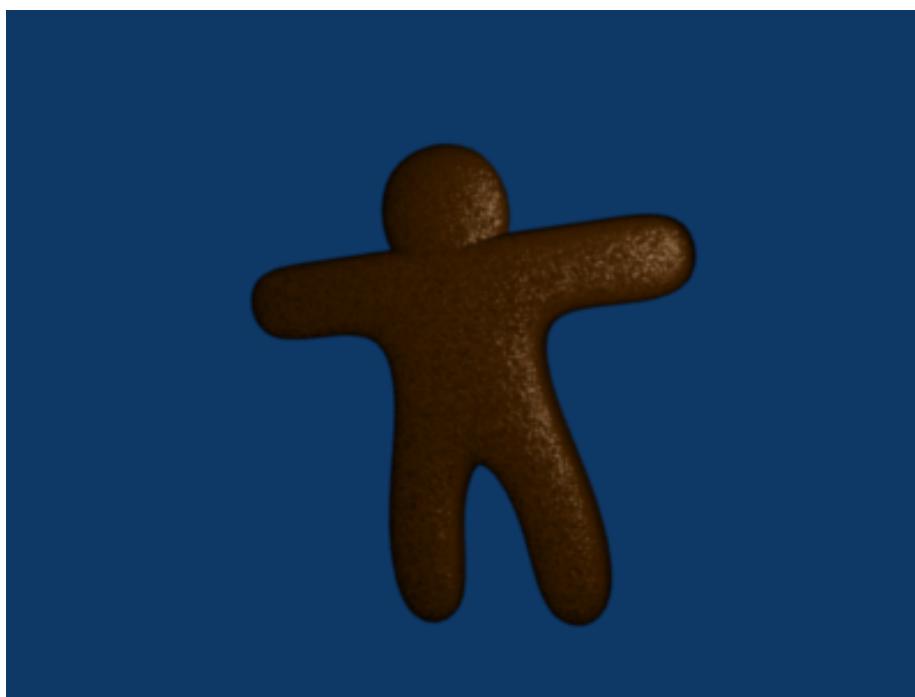
- In the "Map To" panel, deselect "Col[or]" and select "Nor[mal]." and change the "Nor" Value to approximately 1.30.
- In the "Map Input" panel, change the texture coordinates to "Object" by clicking the corresponding button.
- In the "Material" panel, change the "R[ed]" slider to approximately 0.400 and the "G[reen]" slider to approximately half that, about 0.200. Blue can be set at 0.00.



While it is true that textures can only be applied to one object at a time, textures as well as materials can be shared between objects. In this case it is best to let both the head and the body share the same material.

- To do this, simply select the object without the materials(head or body).
- Press **F5** to open the shading panel or use the shading panel button.
- In the "Links and Pipeline" panel, under "Link to Object," click the arrow next to(left of) the "Add New" button.
- Select the brown material.

The steps in this section give a nicely textured, brown surface to the "Gingerbread."



Now all you need to do is add eyes and gumdrop buttons!

## References

[1] [http://en.wikibooks.org/wiki/Blender\\_3D:\\_Noob\\_to\\_Pro/Modeling\\_a\\_volcano](http://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Modeling_a_volcano)

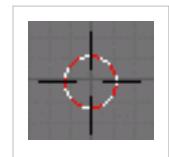
# Blender 3D: Noob to Pro/Penguins from spheres

## Setup

Start with the default scene: it should contain a selected cube. Delete this cube by pressing the **XKEY** and clicking on *Erase selected Object(s)*.

Put the 3D cursor at the scene center by pressing **Shift + C**.

**Note:** after deleting the cube you must be in Object Mode. If not, switch with **TAB**.



## Creating the body

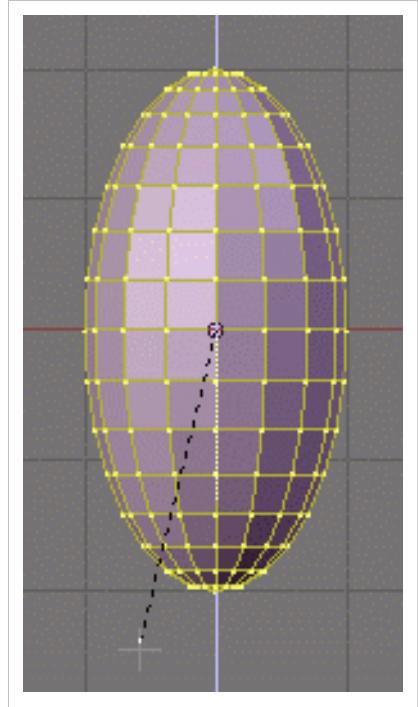
(Note: to ensure that you don't become confused, make sure that your viewport is set up in the same direction you see in these pictures. The colored arrows are red, green, and blue and they control the *x*, *y*, and *z* axes [that's pronounced \ak-seez\, not \ak-siz\], respectively.)

- We start by creating our main body from a sphere. Press **NUM7** to switch to top view, and then **SPACE** → *Add* → *Mesh* → *UVSphere*, then choose 16 segments and 16 rings from the pop-ups.
  - **Note:** You have to manually switch to the Edit Mode after creating the sphere; do this by pressing **TAB** (In older versions of Blender before 2.45, Blender automatically switches to the Edit Mode. This is OK, since we will start editing the mesh now.)

We're going to make it look like a penguin body:

- press **NUM1** to switch to the front view,
- with all the vertices selected (if not, **AKEY**), choose the scale tool (**SKEY**),
- restrict scaling to the Z-axis (**ZKEY**),
- and move the mouse away from the 3D cursor while holding down the **CTRL** key (this snaps the scale values to whole numbers),
  - **Note:** Make sure the mouse cursor is not too far away from the sphere when hitting the **SKEY** or else you may not be able to reach a 2.000 scale value. The scaling steps are proportional to the distance from the 3D cursor when calling the scale tool.
- the current scale shows in the lower left corner of the viewport, click when you've reached 2.000 (**LMB**).
  - **Note:** You can also do this by typing in 2 after starting the scale and restricting movement to the Z-axis

This is our main body!



## Shaping the head

We're going to shape the penguin head from the top of the sphere.

Start by selecting the top two smallest circle segments.

**Note:** selection has been explained in a previous tutorial. Here, the easiest methods are either box selection (**BKEY**) in the front view (**NUM1**) and Limit selection to visible off, or lasso selection (**CTRL+LMB**) in the top view (**NUM7**) and Limit selection to visible on. Don't forget to deselect all first (**AKEY**).

**Reminder:** make sure the Limit selection to visible button is in the right state each time you select vertices, edges or faces. When it's off, selection affects any item, visible or not.



## Building the neck with the 3D transform manipulators

To turn on the 3D transform manipulator, either push down its button or use **CTRL+SPACE** and choose *Enable/Disable*.

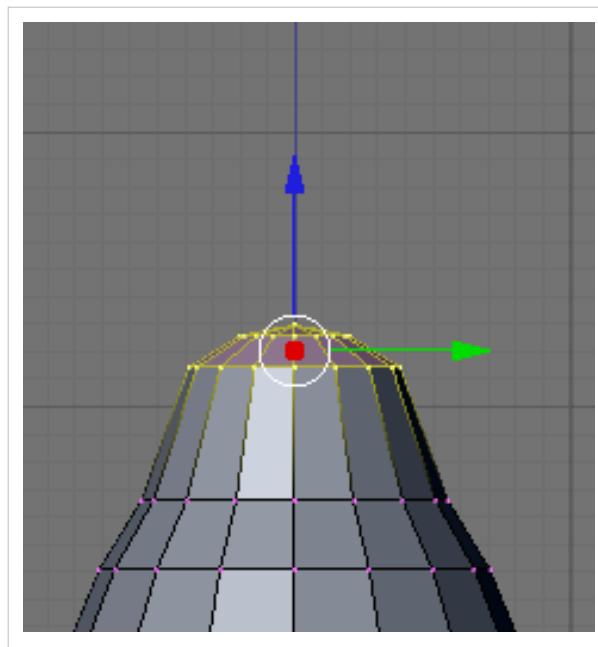
### Moving the two selected circles up

Choose the *Translate* manipulator (red triangle). Go to the front view (**NUM1**). Drag the blue arrow while holding the **CTRL** key down to move the selected vertices 0.3 units up.

**Note:** you may not be able to snap the extrusion lengths to tenths of units. **CTRL** snaps to the grid size by default: if you can only translate by one unit (1.0), zoom in until the grid divides itself into tenths (**SCROLL**). Some Blender versions allow to snap to one tenth of the current step by holding both the **SHIFT** and **CTRL** keys while moving the mouse.

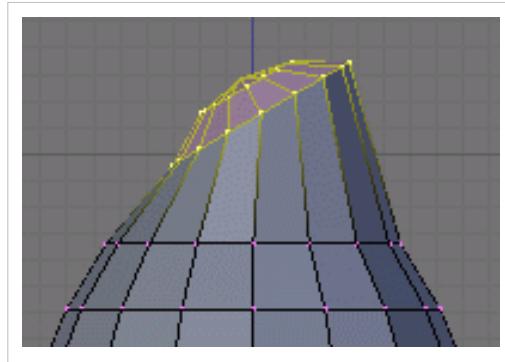
**Note:** instead of the Translate manipulator, you can use the **GKEY** and constraint the movement to the Z axis (**ZKEY**).

Your model should now look as below.

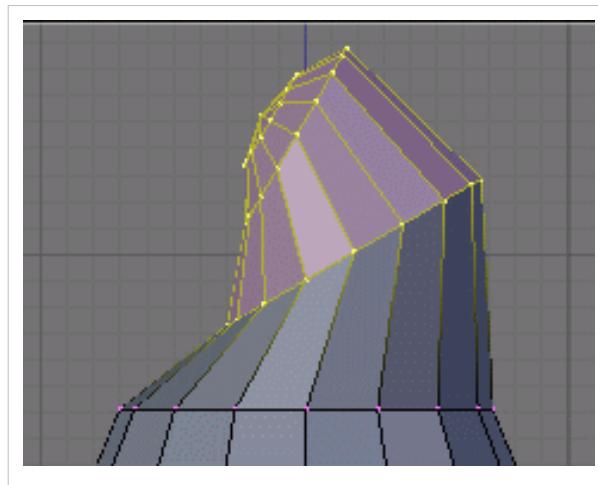


### Rotating the neck

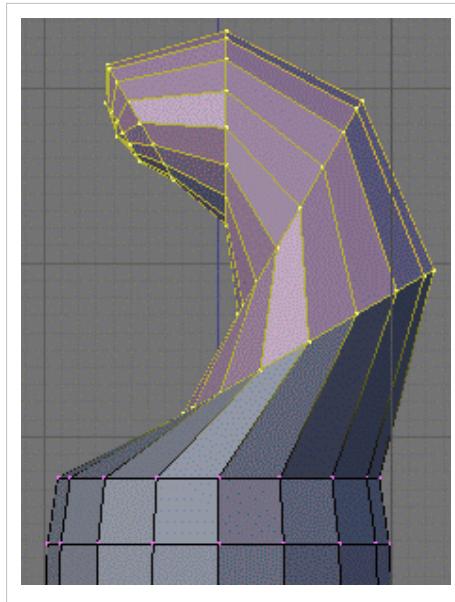
Now switch to the side view (**NUM3**) and make sure that the rotation/pivot point is set to "median point" either by selecting it from the third drop down menu right of the "Mesh" menu, or by pressing ";". Choose the *Rotate* tool (**RKEY**). Move the mouse with the **CTRL** key down to rotate the selection 30 degrees counter-clockwise. Use **LMB** to validate the rotation.



Select an additional ring of vertices by expanding the selection (**CTRL + NUM+** (*Note: NUM+ Refers to the addition symbol on the NUM Pad, KEY+ will not do.*)). You can contract the selection by pressing **CTRL + NUM-**. Move these vertices an additional 0.3 units up, then rotate them as previously 30 degrees counter-clockwise in the side view.

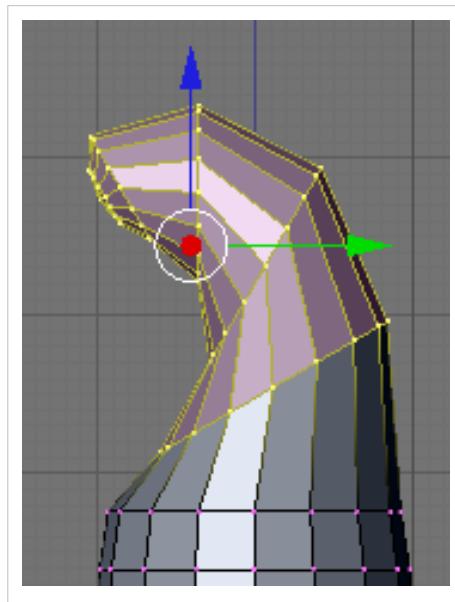


Repeat those steps (selection expansion, translation and rotation) two more times and you'll end up with the body seen in the below.



That doesn't really look like a penguin, yet!

Now move all of the selected vertices to the left 0.4 units by pulling the manipulator's green arrow (and of course holding the **CTRL** key down). This straightens out the neck as seen in the next picture.



[Comments: The red arrow is the arrow I had to pull, not the green one. Answer: You probably didn't switch to side view, and modeled the neck in front view. If you realize here that you have this done from another view then press **AKEY** twice, then **Space** -> **Transform** -> **Rotate** and type 90 or 270]

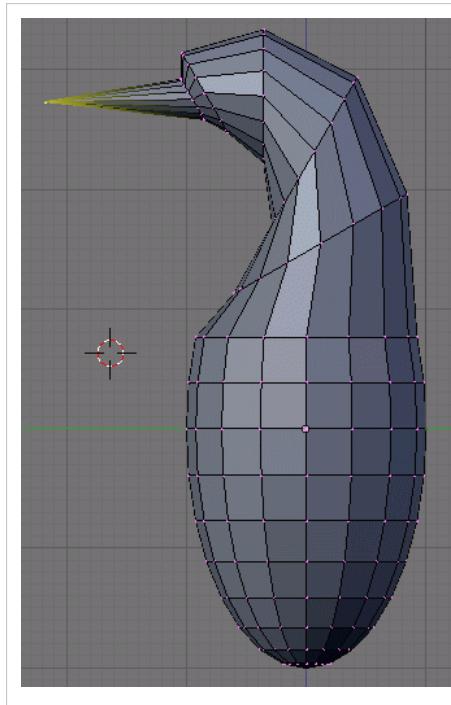
**Note:** you can also use the **GKEY** and translate the selection by -0.4 as displayed in the bottom left corner of the viewport. Still do this in the side view (**NUM3**).

### Creating the beak

Switch to the front view (**NUM1**), and select the frontmost vertex (the one that originally was the top vertex of the sphere) with the **RMB**. Then switch to the side view (**NUM3**) and translate this vertex to the left by 1.2 units using the manipulator's green arrow or the translate tool.

*Note: some Blender versions allow moving the vertices from the keyboard with the following sequence: **GKEY**, **YKEY**, **-1.2**, **ENTER**.*

(Noob note: I prefer to set the MMB to pan the view instead of rotate it, but until just recently I didn't know how to center the camera again, and I'm posting this to help other Blender newbies who prefer this. I found that if you press "C" you can easily move the view to the center of the scene. [For those who use versions earlier than 2.49b {the version I use as well as the only one I've ever used}, I don't know if this will work; still, try it and see if it works]. This made things *immensely* easier for me, since I now have better control over the camera. [In case this wasn't covered earlier, you don't need the MMB to rotate. With NUM lock on, NUM pad 4 rotates {the scene, not the view} left, NUM pad 2 rotates down, NUM pad 6 rotates right, and NUM pad 8 rotates up. These keys rotate the view right, up, left, and down, respectively.]



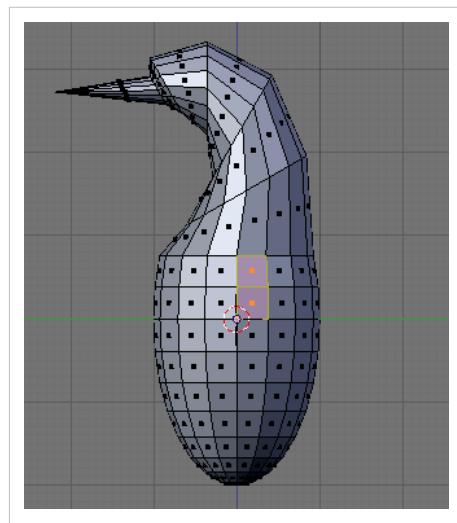
The main body of the penguin is now finished. The next step is to create some flappers for our poor little guy.

### Extruding the wings

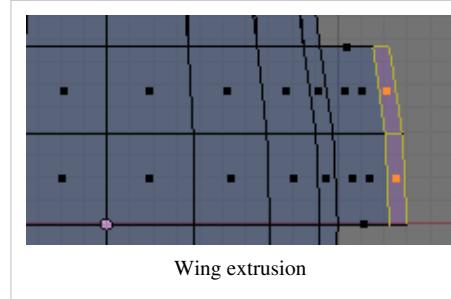
We are going to create the wings by extruding faces on each side of the penguin.

(You can rotate the whole object by pressing A to select all of its vertex and then rotate it pressing R or using the Rotate Manipulator until the axis on your screen matches the axis on the example image. That way, also, you can practice a little more)

Choose the side view (**NUM3**) and switch to the *Face select mode* (**CTRL+TAB** → *Faces*, or click on the triangle in the toolbar).



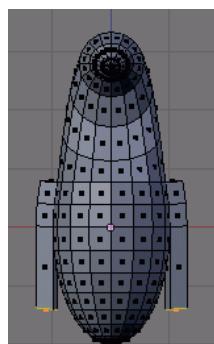
Now, select two faces that will make up the penguin's shoulder as shown on the left (either select the first one with **RMB** and the second one holding **SHIFT**, or use box selection (**BKEY**) to select them both in a single operation).

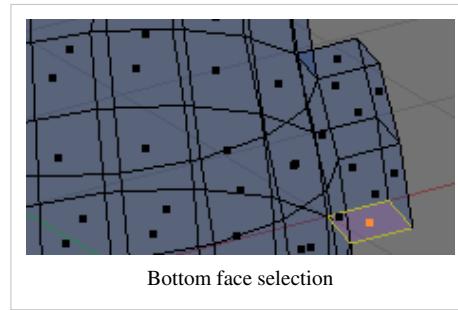


Then switch to front view (**NUM1**) and extrude the selection:

- choose **EKEY** → *Region*,
- constrain to the X axis (**XKEY**),
- hold **CTRL** to snap,
- and move the mouse to extrude the shoulder by 0.3 units.

(User Comment: I find the final product looks better if you only extrude it by 0.2)





We'll now extrude the bottom face of this new extrusion. Rotate the view to show it with:

- a **MMB** drag,
- or several presses on **NUM2**,
- or **CTRL+NUM7** (bottom view).

Press the **AKEY** to deselect all, and select the bottom face (**RMB**), switch to front view (**NUM1**), extrude by 1.4 units down (**EKEY** and **CTRL**).

Now do the same on the penguin's other side: use **CTRL+NUM3** to view the back side (you can also rotate with **MMB**, or press **NUM4** several times).

User comment: I really need to do this by hand? Is there no way to tell Blender to do the same on the other side?

reply: You can select **CTRL+SPACE->scale** and while in front view, use the red square. Just make sure the cursor is between the wings. After you select the faces to extrude, just **SHIFT+SKEY->Cursor->Selection**.

User reply: Modifiers-> Add Modifier -> Mirror. Use the X-axis. Click "apply" when done with the symmetrical stuff.

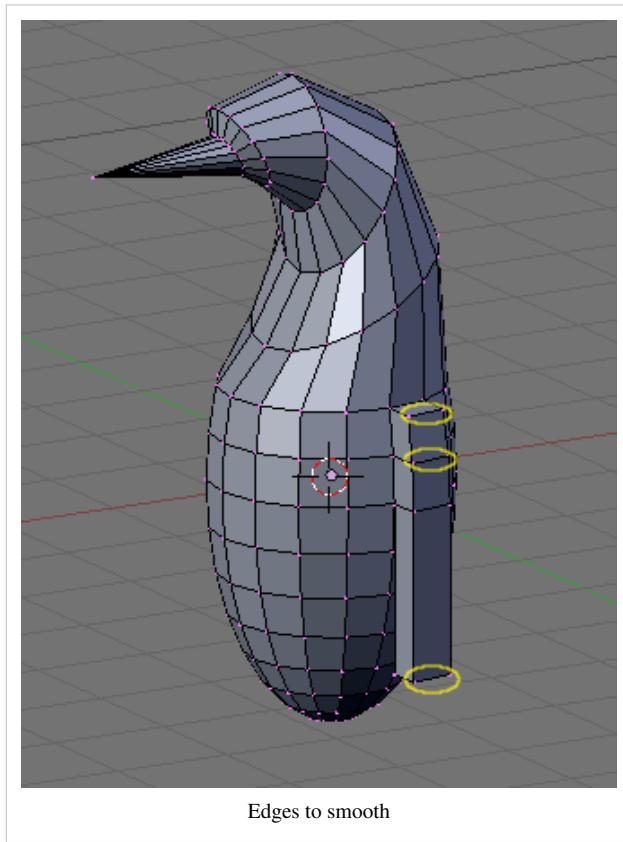
User response: adding a mirror modifier resulted in the surface of the penguin looking corrupt (response to self - because the penguin body was being mirrored on top of itself, but even when half of the penguin vertices are removed an unsightly seam is left).

Final user response: You can fix the 'corruption' by pressing **Ctrl+W** and choosing "Remove Doubles" after mirroring.

Appendix to Final user response: "Remove Doubles" will only get rid of the distortion if you select half of the faces in Front view (**NUM1**), hit (**XKEY**) and select "Faces" in the delete menu on the side to be mirrored, then the mirror modifier won't leave a seam, or distortion due to overlap.

Another User reply: Or you can select the bottom faces on both sides, and only then extrude.

## Smoothing the wings

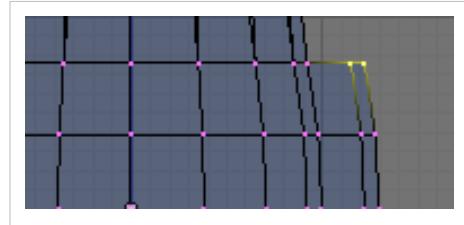


We're going to smooth out the shoulders and improve the wings. Though this can be done in many ways, we'll only use the merge tool.

Rotate your penguin so that you can see one shoulder from above. Then switch to *Vertex select mode* (**CTRL+TAB** → *Vertices*). Press **AKEY** to deselect all, then select the two shoulder vertices with **RMB** and **SHIFT**.

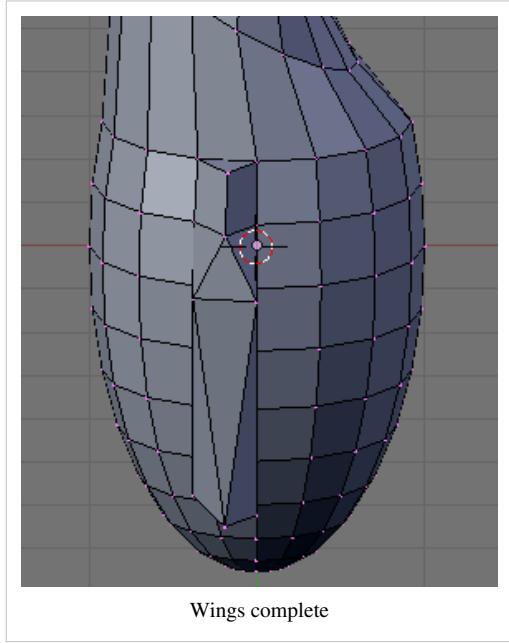
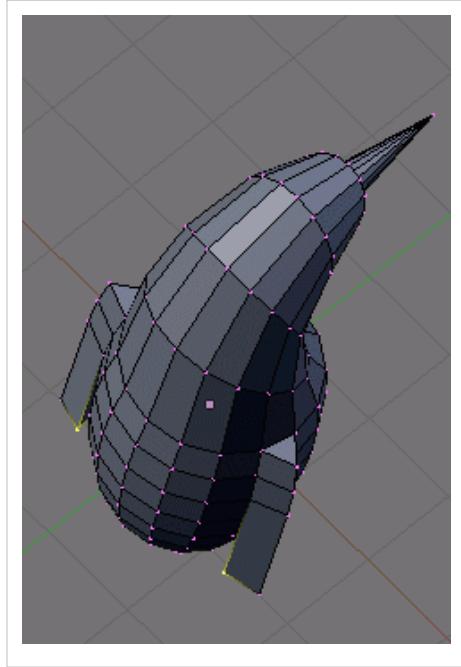
Press **ALT + MKEY**, choose *At Center* from the popup in order to merge the two vertices at their center. Finally dismiss the message saying *Removed 1 Vertices*.

Repeat the steps with the two other vertex pairs shown on the left picture, and smooth the other wing. I'm leaving the middle segments for now, else the wing tips will be too pointy.



**Note:** if you have troubles merging vertices, it comes from vertex duplicates in your mesh. You probably chose Individual Faces instead of Region when extruding the wings, which creates duplicate vertices and neighbouring faces. To clean up your model: select all vertices (**A**) and choose **WKEY** → Remove Doubles.

I finish off the wings by selecting the two backmost vertices of the wings, and moving them up using the blue arrow by 0.1 unit.



(User comment on grid snapping: If you hold **CTRL+SHIFT** while doing a manipulation, it allows you to use smaller fractions as grid snap than with **CTRL**. With translation this is 1 for just **CTRL**, and 0.1 for **CTRL+SHIFT**)

(User comment on using the arrows: I mentioned earlier that the grid snap seemed inconsistent. It now seems that when I've got the 3d manipulators enabled (the arrows), the grid size is 1.0 units. When I have them turned off, the grid size is 0.1 unit. I don't know why, and I haven't found where to change this. The previous picture shows where to turn the 3D manipulators on/off, and which arrow to drag (using **LMB**) to achieve Z-axis movement.)

(User comment: if you spin the view it becomes 1.0 snapping, go back by pressing **NUM1** and you should get 0.1 if not make sure **View → Orthographic** is checked and not Perspective this can also be done with **NUM5**)

(User comment: I've had trouble with the three axis arrows. When I click on the 'hand' icon, the three lines come up but with little blocks instead of arrows at the ends, which I believe means that they will scale instead of move. So, I have had to use **GKEY** for 'grab' and then **ZKEY** or **YKEY** or whatever to move along an axis.)

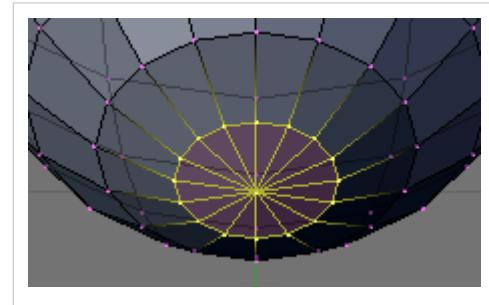
(Note from another user: Next to the hand click the triangle to go to Moving Mode).

(Reply to previous comment: You can change the size of the 3d manipulator arrows in the preferences area so that they are easier to click.)

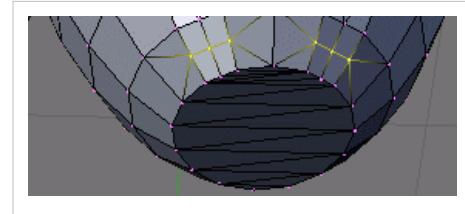
## Culling the underside

We're going to cut the penguin's lower end, for it to stand up! Select the bottom vertices (bottom vertex and the first ring above it) as shown in the picture. There are many ways, this is left as an exercise. Once they're selected, delete them (**XKEY** → *Vertices*). Now our penguin is hollow: select all the vertices around the hole, and fill it using **SHIFT+FKEY**.

**Note 1:** Instead of using **SHIFT+FKEY** to make a zigzag surface, you could extrude the circle of edges (**EKEY**), constrain the extrusion to the Z-axis (**ZKEY**) and enter 0 (**0KEY + ENTER**) as the distance. This effectively duplicates the circle. Then do a merge (**ALT+MKEY** → *At Center*) to create a nice surface.



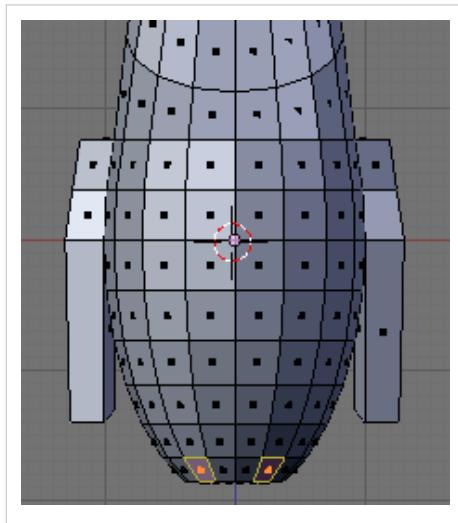
**Note 2:** Press **CONTROL+NUM7** to view the penguin from the bottom. Now, without deleting anything, select four vertices, top, bottom, left, and right in the second ring out from the center vertex. Then snap your cursor to the selection. **SHIFT+SKEY** → *Cursor -> Selection* This places the cursor at the center of those vertices. Finally, select the center vertex, expand the selection to include the first ring (**CONTROL+NUM+**) then merge as above (**ALT+MKEY**) but at cursor rather than at center. This achieves the same result but without having to delete or extrude anything.



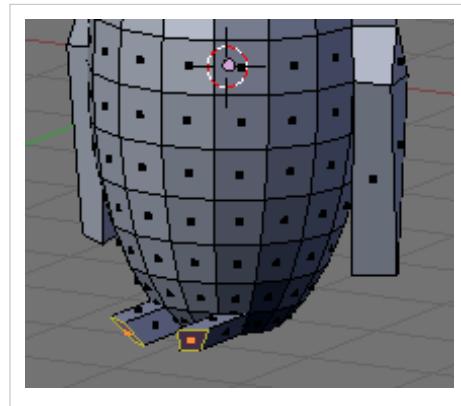
**Alternative Method - Extrude & Scale:** From a side view. Box select (**BKEY**) the bottom vertex & the 2 bottom most rings, then delete (**DELETEKEY** or **XKEY** → *Vertices*). Now box select (**BKEY**) the remaining bottom most ring. Initiate an Extrude (**EKEY** → *Region*) command, and escape terminate (**ESCAPEKEY**) said command. It will appear as if nothing has happened, but you have created a new (overlapping) bottom ring and is currently selected. Change the view to Orbit Down (**NUM2KEY**). Now Scale (**SKEY**) the selection, press the number **0** to close, then press the **Enter** key to commit. The bottom should have closed with a number of triangles, all pointing to a common center vertex. As a good practice, you should now remove doubles' (**WKEY** → *Remove Doubles*).

(User comment: To quickly select all the vertices around the hole, you can enter edge mode (**CTRL+TAB->Edges**) and then select one edge that goes around the hole. Now press **CTRL+E** and select *Edge Loop Select* which should select all edges around the hole. Now go back to Vertex select mode and continue with **SHIFT+FKEY**.)

## Adding the feet



The next step is to provide the little guy with feet. To do this, we're going to extrude two of the front faces:

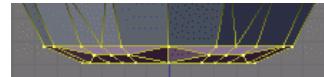
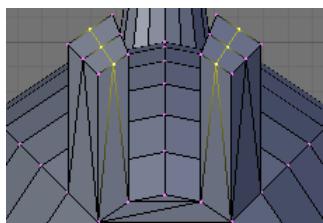


- choose the front view (**NUM1**),
- switch to *Face select mode*,
- turn on *Limit selection to visible*
- and select the face to the left and right of the middle two faces of the penguin.

Then switch to the side view (**NUM3**) and extrude the selection by -0.6 units (**EKEY** → *Region*, restrict to the Y axis: **YKEY**).



Keep the selection and look for the *Mesh Tools* in the *Buttons* panel. If you can't see it, press **F9** to switch to this panel. Then click on the *Subdivide* button (at the top, between *Short* and *Innervert*). Or press **WKEY** and choose Subdivide



Switch to "Vertex Select Mode", Now select the three middle vertices (or two edges) vertically at the tip of each foot, and drag them along the Y axis by 0.3 units towards the penguin *Note: if something goes wrong here, you may need to remove double first.*

*As always, to move the vertices, either use the manipulator or GKEY and YKEY sequence.*

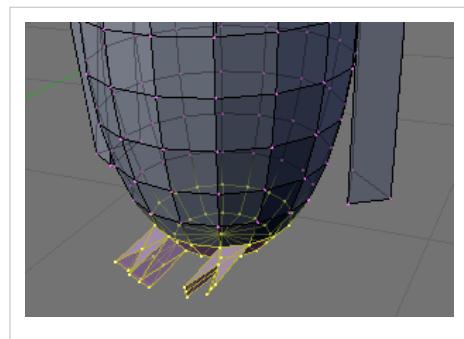
You should end up with what's shown in the right picture (minus the selection).

The feet look too thick, let's flatten them a bit. Switch to the front view (**NUM1**) and select the two bottom vertex rows (*Limit selection to visible* off, use either the lasso or box selection).

Then choose the scale tool (**SKEY**), limit its action to the Z axis (**ZKEY**) and scale down by a factor of 0.4.

The feet still look rather peculiar, so please go ahead and move the vertices around on your own as you like.

**'Reminder:** you can use the **GKEY** and restrict movements to the X or Y axis using the **XKEY** or **YKEY**. Try not to move vertices along the Z axis to keep the penguin's bottom flat.



## Extruding a tail



To complete the penguin, we have to add a tail (the end of the tuxedo):

- go to the back view (**CTRL+NUM1**),
- make sure you're still in *Vertex select mode*,
- and that *Limit selection to visible* is on.

Select the three middle vertices in the second row up from the bottom. Then, switch to the side view (**NUM3**) and extrude the edges 0.3 units away from the penguin and 0.08 units down (**EKEY** → *Edges*), so that the end of the tail is at the same level as the bottom of the penguin.

Press **CTRL+WKEY** to save your work!

- *Note 1:* I found it easier to select the center most, lowest vertex, and move it along the Y axis only. This creates a crude but very effective tail (that doesn't look all that bad when subsurfed).
- *Note 2:* to do this correctly (if I've properly understood the previous user) you can also select the two faces in the bottom row (**CTRL+TAB**→*Faces*), which are connected to the three middle vertices and their projection on the bottom line, then split them into triangles (**SPACE**→*Edit*→*Faces*→*Convert To Triangles* or using **CTRL+TKEY**), then select the right two of them (it was the right quad face before) and flip the triangles' orientation (**SPACE**→*Edit*→*Faces*→*Flip Triangles Edges* or using **CTRL+FKEY**). So, there will be the five edges connected to the middle point in the second row from the bottom and only three to the vertex, mentioned by previous user. Then just move this lowest, center most, vertex by Y on 0.4
- *Note 3:* don't do that extruding of the tail with only the edges or you will get an so called 'non-manifold mesh', cause you will have an edge with outgoing three faces, so blender can't recalculate the surface and you can't use the decimate modifier. this can be very nasty if you're modeling more complex things later. better select the whole two faces at the back end and extrude that and then shrink it along the z-axis.

## Subsurfing

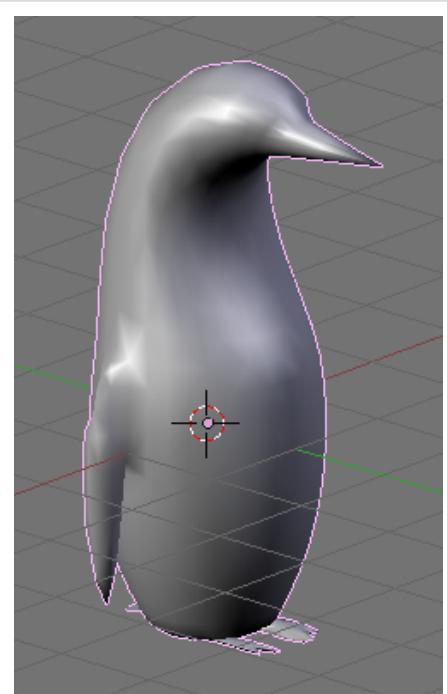
Go to *Object Mode* (**TAB**), and make sure the penguin is selected. Then check for the *Modifiers* toolkit in the *Buttons* panel. Press *Add Modifier* → *Subsurf* (or Press **SHIFT+OKEY**).

Look at the penguin now, he's much smoother. You can alter the levels of the subsurfing if you like, but I'll settle for level one. Under the *Links and Materials* toolkit, you can press the *Set Smooth* button as well, which makes the penguin really slick.

**Note:** *you may see some weird effects at the bottom and the tail after subsurfing the penguin. If so, there is an issue with normals: they have to be all pointing outwards. This can be achieved by selecting all vertices in Edit Mode and recalculating the normals outside (**CTRL+NKEY**). Click on the message to confirm. Note that **CTRL+SHIFT+NKEY** will turn the normals inwards and that **WKEY** → Flip Normals flips them.*

**Question:** *The finished penguin looks fine in Object Mode, but when I render it, it looks odd. Quite patchy.*

**Answer:** *You must click the apply button to the right for it to subsurf at render.*



Now that's what I call a penguin. But then again, there are no penguins where I live, so I might be wrong.

## Extra

The penguin can be colored or textured, but that will be part of later tutorials!



This is what the penguin (sans tail) looks like, textured and ready. Orbisonitrum



The eyes are there, just not easily visible in the thumb. At the top of the white part, two faces on the chest were subdivided to give the white more of a curve at the top. The faces were selected that were going to be white, and the I used separate (**P**) to make them a different mesh. I used a white material for the chest, black for the body, and grey uvspheres for the eyes. Apparently, an easier way to colour the chest can be found at Multiple Materials [1] *please feel free to replace this with your own image of the penguin you made, with comments on how you put your own style into it --Most recent is Bnty 11:35 PST, 19 May, 2006*



A pretty basic picture of a penguin. I subdivided the stomach and eyes, but then I also added some eyeballs by making a UVSphere, cutting the top of it off, and then placing it inside of my penguin's head. All the colors have specular colors, giving the penguin a slight blue glow under the black.

## References

[1] <http://www.blender.org/documentation/html/x4448.html>

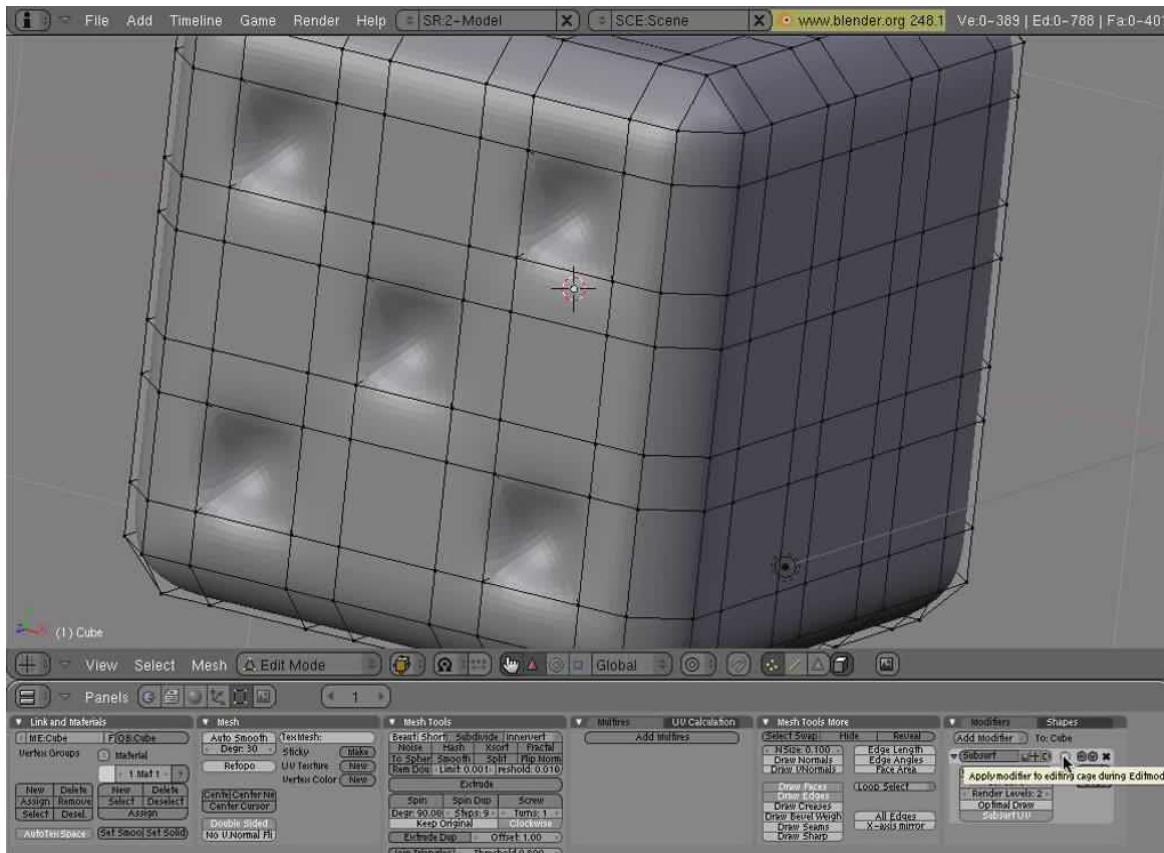
# **Blender 3D: Noob to Pro/Die Another Way**

## Video Tutorial

A video tutorial has been created for this chapter in Blender 2.48a.

It is compressed and packaged in the Theora (.ogg) video format and requires a player that is able to decode this codec in order to play it, such as the VLC [1] player which is available as a free download for Windows, Mac, and most Linux distributions. Firefox 3.5 is also able to stream Theora video.

For best results, it is recommended that you save this file to your computer for viewing, rather than streaming it inside a web browser, since it is 1020 x 746 pixels.



## Introduction

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

- polygon mesh
  - face loop cutting
  - subdivision surfaces
  - subdivision creases
  - bevel
  - set smooth
  - multiple materials
  - extrusion
  - merge vertices
  - remove doubles
  - constraints

There are two methods to create the circles for the die: subdivide first and manual sizing. In either case, start with the default Cube.

## Subdivide First

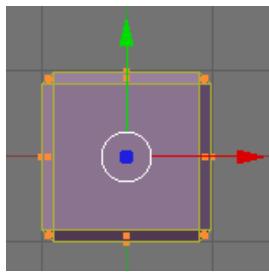
The die needs to have a 3x3 matrix for the coloured dots (pips). A quick way to do this is to simply *Subdivide* the cube twice before doing anything else. The disadvantage may be that the spaces for the pips may not be exactly the size that you want. If not, see the next section: Manual Sizing of Pips.

## Manual Sizing of Pips

### Step 1

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 (hit spacebar for manual input). Bevel of 0.150 is ok.

Note: If you have chosen to subdivide the die twice, jump to Section "Creating Pips" and put bevel of 0.17 in order to have pip's edges length 0.34



### Step 2

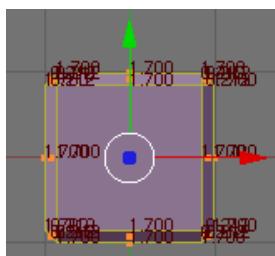
In editmode, go to the Editing tab (**F9**) and look at the **Mesh Tools 1** panel (*Mesh Tools More* in some versions). Turn on Edge Length and note the length of one of the sides of the square faces. This should be 1.7 if the above settings were used.

Button "Edge Length" may be outside the screen so you may need to close another set of buttons before you can get to it.

Or, you can use MMB to scroll over to see the Mesh Tools 1 panel (in later versions: Mesh Tools More) with "Edge Length" button on it.

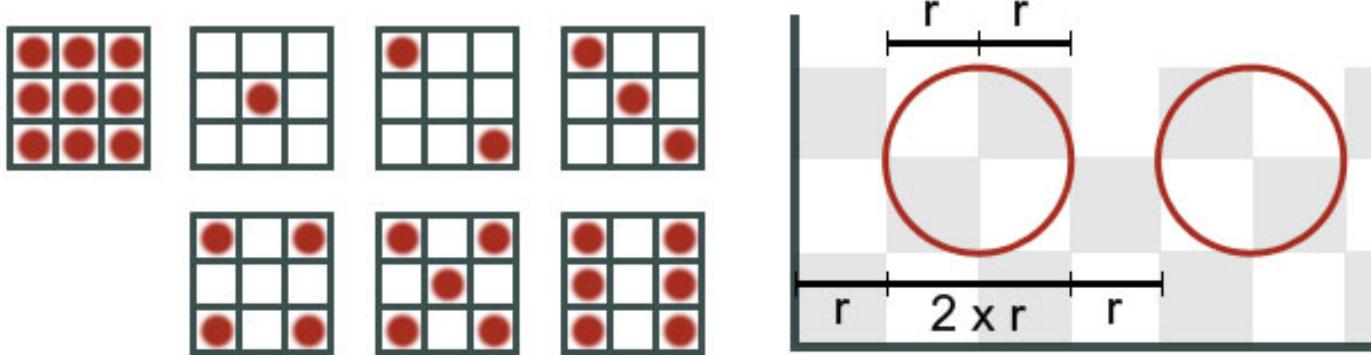
Or, you can zoom in and out in the menu window with CTRL and NUM+ or NUM-

Or, you can press HOME when button window is focused to see all buttons.



### Step 3

A typical die has a grid of 9 possible positions for the pips and the gap between the pips is the pip 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 pips use two each. This means the gaps are of size  $1.7/10 = 0.17$  and the pips ( $1.7 \times 2/10 = 0.34$ ).



### Step 4

Now it's time to subdivide the surfaces of the die according to the mathematics above. We'll do that using "edge loops" - additional edges you can add to existing objects.

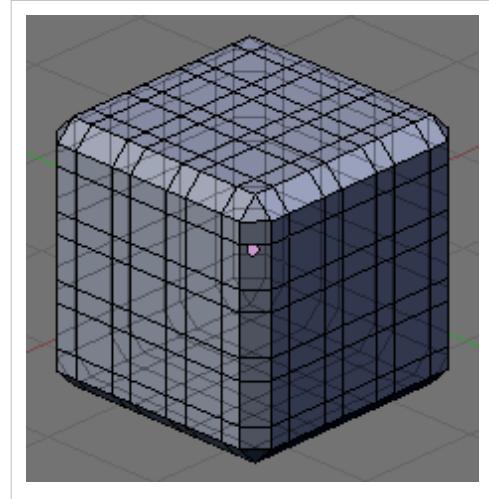
- Select axis aligned view: **NUM1**
- Enter loop cut mode: **CTRL+RKEY (OR KKEY→1KEY OR KKEY→Loop Cut OR CTRL+EKEY→Loopcut OR CTRL+EKEY→NUM5)**
- Select loops' placement: move the mouse around until you see a purple line going the right direction.
- Enter the number of loops: **9KEY (OR NUM9 OR SCROLL up 9 times OR NUM+ 8 times)**
- Add the loop: **LMB (OR ENTER etc.)** on one of the big faces

*Noob: What's "the right direction?" What should this thing look like after applying the loop/cut? HELP!*

*Another Noob: It sounds like you have version 2.44 or later. You should do the tutorial "Die Easy"(seriously, it is).*

Now we just have to get rid of the 2nd, 5th and 8th loops to make the undivided spaces for the marks.

- Select edge select or vertex select: **CTRL+TAB→NUM2 (OR CTRL+TAB→Edges OR CTRL+TAB→NUM1 etc.)**
- Deselect all edges with **AKEY**
- Choose a loop to remove (using the **BKEY** to enter box selection mode and drawing a box around the one you want; this will get the whole loop, all the way around the cube). Spoiler: you can also use Alt-RMB on an edge to select a loop (or select the edge and click **Select --> Edge loop**).



*Noob: It seems that I could not remove multi loops at the same time, since the error message kept bumping out. Another Noob: That happened to me when I tried to remove them all at once, but it worked fine when I removed one at a time.*

- Remove loop: **XKEY→7 (OR XKEY→Edge Loop)**

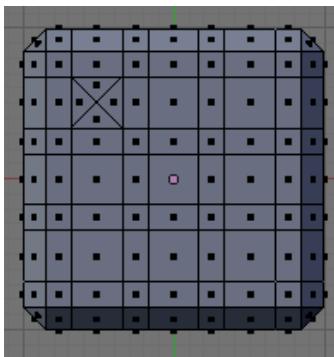
Change views with **NUM3** and **NUM7** and repeat steps as necessary. When you're done, your die should look like the one pictured to the right.

## Creating Pips

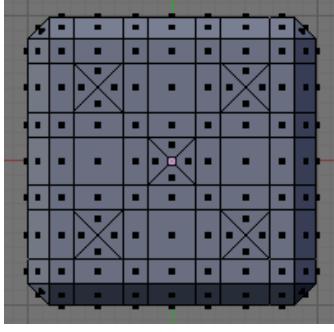
The die needs the pips added. Everyone knows how the pips on a die look, right?

### Extrude and Merge

(Note: This is Step 10) Select one of the faces where a pip would go and extrude the face by hitting **EKEY** and then **ESC**. Do not click after hitting **EKEY**. This actually replaces the first face with another one even though it looks like nothing has happened. Merge the second face by using **ALT + MKEY** to merge the 4 corners into the centre. It will tell you *Removed 3 vertices*. [User Note: To clarify, pressing ESC when extruding does in fact extrude the face as per usual, but by a distance of zero. This creates four new, infinitely narrow faces around the original square face. These four faces then get 'dragged' into the middle when the four vertices of the original square are merged into one. Test this by extruding by say, 0.01 (instead of pressing ESC), and you'll see the result is almost the same.][User Note: on my computer hitting ESC after the EKEY seemed to cancel the extrude and to get the correct result I had to hit the "ENTER" key after the EKEY] You should get the following:



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



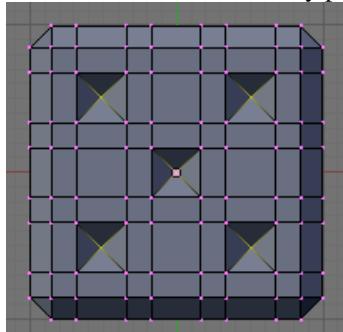
Notes :

- You could create this pip spot on all 9 spots and copy this side of the die to the other six. The amount of time spent doing all of that may be just as long as doing each side individually. You would need to delete the other 5 faces, copy the dented face 5 times, place each face precisely by rotating and moving, and remove doubles.
- (User comment) I accidentally selected some pip faces from the opposite side of the die (the side behind the side I was looking at). To prevent this, I selected *Limit selection to visible (Occlude background geometry in 2.47)* which should be the second button from the right in the header of the 3D View.
- **Noob, 19th Oct 2008:** You can save time by selecting all the 'pip faces' and extruding them simultaneously, **ESC-ing** immediately after you do so, like above. You'll still need to merge the four corners of each extrusion one-by-one, though, or you'll get some odd results.
- **Noob, 28th Dec 2008:** You can save more time by extruding all faces simultaneously, as above, and then selecting "Collapse" rather than "At Center" while merging.
- **Noob, 02/02/09:** I tried with the "Collapse" trick as well but all I got were just black squares.
- **Noob, 19/02/09:** I got the black squares as well, but it is just a display problem. Press **Tab** twice (going to *Object mode* and back to *Edit mode*) and they are gone.

## Create Pips

(Note: This is Step 11) Select one of the edges of the pips to check the size is 0.34.

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



(user comment) According to step 4 we are still in front-view (**NUM1**), but then the **ZKEY** modification gives undesirable result, changing view to top (**NUM7**) does the trick! This applies to blender version 2.44.

- (response) Actually, at the end of step 4 it states to change to side and top view (**NUM3** and **NUM7**) as necessary, so really there is no official view the tutorial left the user in. Also, in this step, the writer mentions they put 5 pips on the top.
- (user comment) using **YKEY** instead of **ZKEY** is also fine.
- (user comment) Depending on which axis it's supposed to be moved, use the **ZKEY,XKEY** and **YKEY** after the **GKEY** accordingly.
- (user comment) Pressing **ZKEY** twice should move them along the normal, which should work no matter which faces are selected.
- (user comment) I noticed a shortcut: When you extrude, extrude by -0.17 and then do the Merge -> Collapse. Then the point is already inside.
- (user comment) I just noticed something different. If you extrude and collapse, after applying the subsurf, you got sharp edges of the pips than the pips I got in this tutorial way.
- (user comment) In blender 2.49a there is an issue where when "Occlude background geometry" is on, sometimes you'll click a vertex and have a far away vertex clicked instead. Rotating the camera usually helps, and if you find your vertex isn't selected, but you know you clicked, clicking again will deselect whatever vertex you selected. Also a big issue with the extrusion - as with escape, right clicking to exit extrusion also creates an "unseen extrusion". The "unseen extrusion" can be noticed - there's black dots in the middle of your square edges. This is caused by the 0-width faces being selectable (and therefore having a dot in the middle). I had to completely do-over my model because Remove Doubles \_did not\_ remove the extra vertices. This became noticeable when I would extrude and it would only extrude vertices.
- (user comment): Don't collapse the vertices! In that case, the pits will be rendered badly. Collapsing removes four vertices, merging at the center removes three vertices --> so there is a difference!

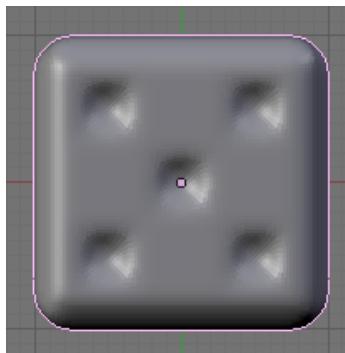
## Smooth Out

**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 present versions, you just need to use "Add Modifier" on the Modifiers tab (in Edit-mode), to add a SubSurf modifier. (Or press **SHIFT+OKEY**)

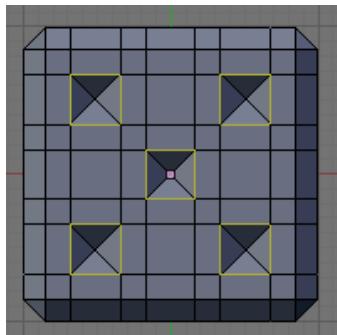
In the image below Levels is set to 3.



## Make Sharp Edges

On a die, the edges of the pips 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 pips 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. (to see the effect, you must have the subsurf modifier turned ON). 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.

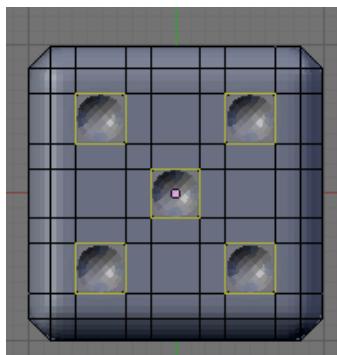
**Newbie Note:** Trying to crease all 6 sides of the die at once using **SHIFT+EKEY** and moving the mouse doesn'tcrease all sides of the die. Better to use **NKEY**, or do one side at a time.

## Repeat

Repeat steps 10, 11 and 13 (that is Extrude and Merge, Create Pips and Make Sharp Edges) 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:

[User Note: Do you mean subsurf instead of subdiv?]

- Noob note: On the side with 6 pips, I had a hard time getting the creases to work. It turned out to be because I selected in vertex mode when I selected the edges for creasing, and I tried to crease all 6 pips at once. This put crease values on the short vertical edges between the pips, which messed things up.



- User Shortcut: To do them all at once with no repeating. However, order of operation has to change to make it work:
  - In Face Select mode, select all pip faces on all sides.
  - **SHIFT+EKEY** to crease the edges. (needs to be done before extruding, since extruding changed the selection)
  - **EKEY -> Individual Faces** then type -0.17 to extrude all faces inward.
  - **ALT+MKEY -> Collapse** will merge all the extrusions to their respective centers.

(noobie) Some of my pips are square and I tried everything. What should I do?

## Camera Setup

You can make a test render now to see that the pips are the right size and that the bevel is right. So, turn the subsurf 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**. (**Noob note:** If you can't find "empty" in the list, make sure you are in Object mode.) 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:

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. Another useful keystroke (for pre-2.43 especially) to know is that when you are in camera view, pressing **Gkey** and then **MMB**, movement will be constrained to the way you are facing. The **mouse wheel** zoom moves your view towards and away from the camera, without actually moving the position of the camera.

You can also move the Camera in free "Fly" mode by going into the Camera view (press **NUM0**) and then **Shift+F**. Now you can "Fly" through the scene and use this setup the camera angle. Make sure that you keep the flying velocity very low by using the scroll wheels or -/+ buttons or the camera will be simply out of control.

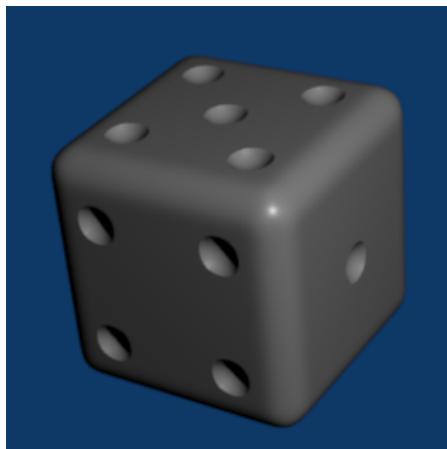
## Render

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 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** (Or press **F3**) 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:



[user note: to get such a nice render I had to bump the render level of

subsurf up to 5, otherwise I get artifacts around the pips]

[user note]If you set smooth on the whole shape, you'll get artifacts around the circles.

[user reply] If OSA is on for the final render, you'll get a nice render result as shown by the above figure with subsurf level 3.

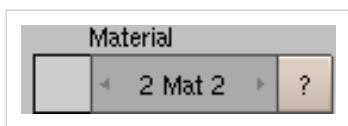
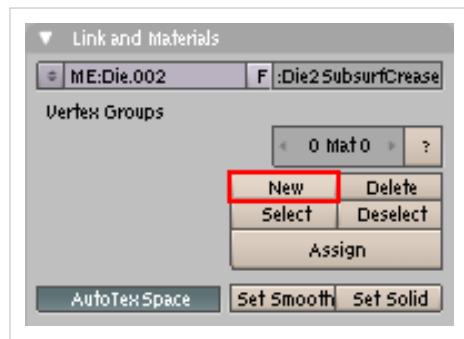
## Color

### Multiple Materials

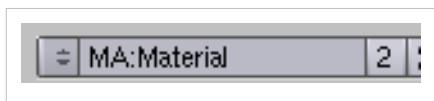
..

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

In the *Buttons Window* go to the *Editing* panel (**F9**) again and make sure the die is selected. In the *Links and Materials* subpanel there is a section for materials (the right; the left one is for vertex groups) and the box left of the question mark should read "0 Mat 0" (the first number is the number of material links for this object; the second number is the number of the currently selected material link).



The die may have more than zero materials if you had assigned materials to the object already. By pressing the *New* button add enough materials to make 2 in total.



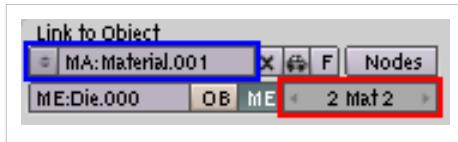
Go back to the *Shading* panel (**F5**) and there is a box at the very top of the *Links and Pipelines* subpanel with the number 2 beside it. If there is no such subpanel select *Material buttons* (cycle shading buttons using **F5** too). Click this number and select *Single user* in the dialog to make the two materials you've just created independent. Use the arrows on the left side of the box to switch materials.

**F5** too). Click this number and select *Single user* in the dialog to make the two materials you've just created independent. Use the arrows on the left side of the box to switch materials.

Note: If you do not see a 2 to the right of the material name, that means the material is already a single user material. To change it back you can click the button labelled *F*, but for this example, you do not want to do that.

Note: There should be at least two materials now. One has the materials initial name the other has a number appended to its name (e.g. *Material* and *Material.001*).

Use the *Material* subpanel to make material 1 bright red by just picking red in the colour picker (the rectangle to the left of the *Col* button) or by setting the RGB sliders (right of the *Col* button). Make material 2 white by doing the same. Or pick whatever colour you prefer and material settings.



--MSK61 (talk) 11:27, 17 March 2008 (UTC): The colour picker is to the left(not right) of the *Col* button, while the the RGB sliders are to the right(not left) of the *Col* button.

Note: It is possible that the two materials were not automatically linked to the material links of your die. If so use the *Links and Pipeline* subpanel to link the [materials](#) to the respective [material links](#). First select the link then the material.

*Note from noob:* Using white as a colour will not let you spot a change in colour, since the default colour is white. For test purposes I recommend you choose another colour (e.g. blue), so you can avoid getting confused.

*Note from another noob:* Actually (at least in blender 2.49), the default color is grey. To choose a white color, set all colors (R,G,B) up to 1.000.

## Assign Materials

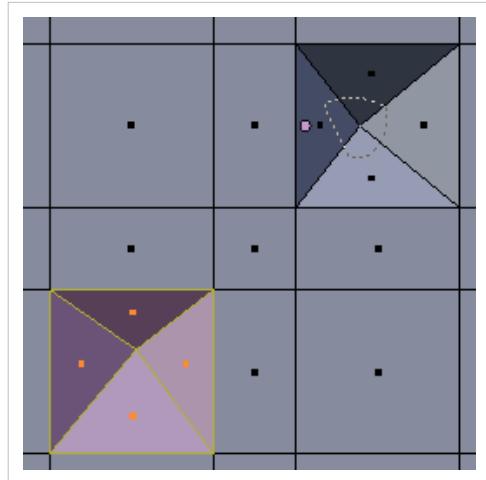
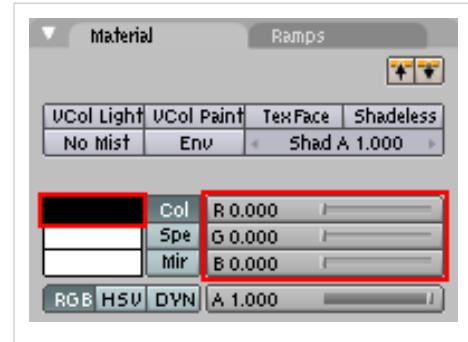
These colours need to be assigned to the right parts of the die.

Go into *Edit Mode* and turn off subsurf to make selecting easier. Do this in the *Editing* panels (**F9**) *Modifiers* subpanel. Right after the subsurf modifiers name there are three buttons (darkgrey). Press the rightmost to deactivate the modifier in the edit mode.

### To make the die red

- Select the entire die (AKEY).
- Click the Editing button (F9).
- In the Link and Materials panel, on the right-hand side you will see "2 Mat X" where X is either 1 or 2. Click the left/right arrows until you see the adjoining square to the left turn red. As you click on the arrows, notice that the label above "2 Mat X" changes from "Material" to "Material.001". These are the names of the materials that you created in the Shading (F5)->Links and Pipeline panel.
- Click the Assign button (in the Material column, not the Vertex Groups column). The Assign button associates the selected faces with the selected material.

The entire die should now be red.



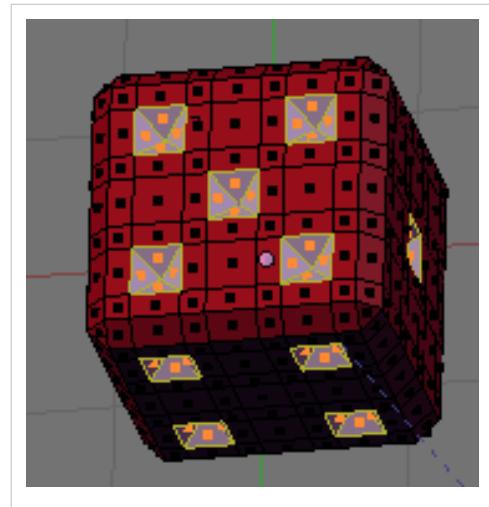
### To make the pips white

Use the same method as above, but with only the inner faces of the pips selected and with the colour white.

There are a variety of ways to select the inner faces of the pips:

#### Pip Selection Method 1

Note: Here you may find *Lasso Select* useful.



- Make sure you are in *Edit Mode* (TAB)
- Select *Face select* : CTRL+TAB→3KEY
- Select *Limit selection to visible* : should be the second button from the right in the header of the 3D View
- Go through the axis align views and select the faces:
  - Align view: NUM1 (CTRL+NUM1, NUM3, CTRL+NUM3, NUM7, CTRL+NUM7)
  - For each view, lasso select the pips' faces: hold **CTRL** and drag **LMB** around the pips' middle vertex (no need to press **SHIFT**, *Lasso Select* automatically adds the new faces to the previous selection).
- If all went well you should be able to read *Fa:84-449* in the *User Preferences* right after the Blender version number. ( $84 = 4*(1+2+3+4+5+6)$ )

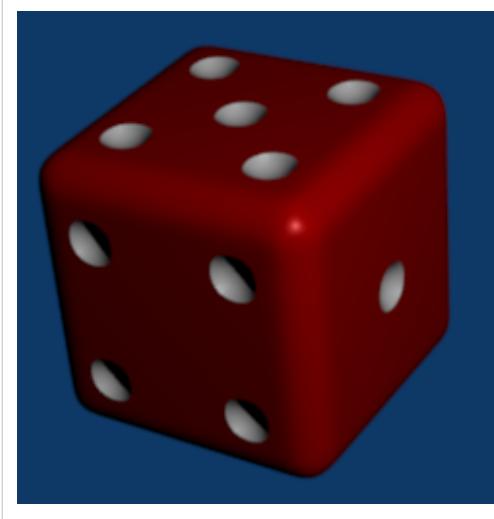
#### Pip Selection Method 2

Press **CTRL+ALT+SHIFT+3KEY** and every triangular face will be selected.(Every pip + The 8 corners of the dice. Just deselect the 8 corners and you're good to go!) ;-)

Similarly, you can select quad faces using the **4KEY**.

#### Pip Selection Method 3

You can also use circle select to easily select the desired faces. Enter circle select mode by hitting **BKEY** twice. The circle can be made bigger or smaller by using the **scroll wheel**. Drop out of the mode (**RMB**) to rotate the cube and drop back in as above. Selected vertices are added to those already selected like with lasso mode but without the need to keep holding the control key or draw an accurate lasso)



Turn subsurf back on (*Modifiers panel*→subsurf modifier→enable in edit mode) and render **F12** with OSA (only put it up as high as you need for the resolution of the image you are rendering).

(*Note: in Blender 2.44 you should use the Assign button in the editing panel (Link and Materials)*)

(*another noob says: don't be fooled (as I was!) by the fact that there are TWO Assign buttons in this tab, you want the big one on the right under materials, not the small one on the left under Vertex groups!!*)

#### Pip Selection Method 4

That's the easiest way: 1- select one of the triangles (the face) 2- **Shift+Gkey**, then choose perimeter or press **4NUM**.

### 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 pip size. You do this by selecting the vertical or horizontal segments and just scaling them in one axis. Here we will reduce the pip size and the bevel by half.

Go into front view (**NUM1**), turn off clipping (i.e., allow selection of invisible vertices) and select a line containing pips (i.e., Select vertices mode (**CTRL+TAB+NUM1**), and box select a thin vertical line of vertices to the left of a row of pips, then box select another thin vertical line of vertices to the right of that row of pips, making a total of 64 vertices). 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 pips and 6 times for the bevel:

(*Noob note: I find this confusing. what is a line containing pips? does this mean a loop line? Using alt-RMB no longer works to select a loop, but selects a single edge.*)

(*Noob response: he meant a pair of loops of vertices adjacent to a row containing pips. alt-RMB no longer seems to select complete loops - it works until it hits a pip and then it stops - but the box selection can be used.*)

(*Another Response: Any time you change the geometry of a shape you effect how the automated tools will work. Many dont work at all once you get to complex organic shapes, so its best to not rely on them too heavily.*)

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.

### References

[1] <http://www.videolan.org/vlc/>

# Blender 3D: Noob to Pro/Die Easy 2

---

The previous *Die Easy* tutorial needed so many changes to adapt to Blender 2.44 that it was easier to start from scratch. This *Die Easy 2* tutorial is heavily based on *Die Easy* and *Die Another Way*, and adds improvements.

In the following tutorial, you will use:

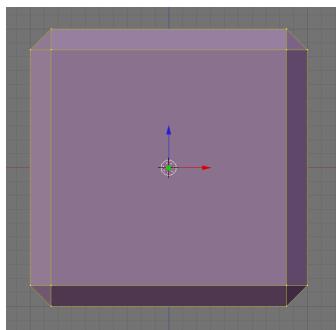
- polygon mesh
- face loop cutting
- subdivision surfaces
- subdivision creases
- bevel
- set smooth
- multiple materials
- extrusion
- merge vertices

## Start with a beveled cube

Start with the default scene: Blender should be in *Object Mode* and in *Top view*, with the default cube selected. Go to *Edit Mode* (**TAB**). All the vertices should still be selected. If you're not sure, press **AKEY** to deselect all (the cube goes dark), press **AKEY** again to select all (the whole cube is selected again).

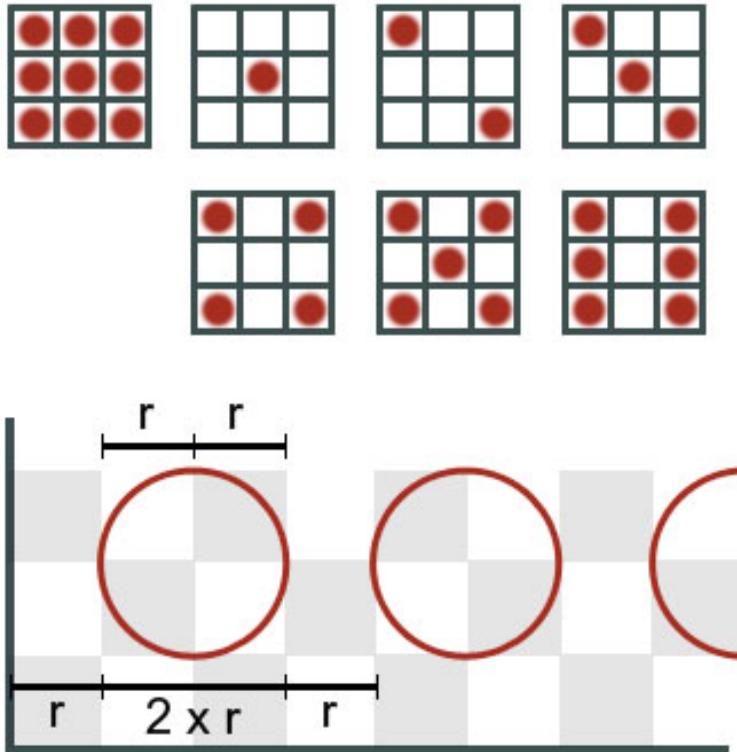
Let's bevel the cube. Press **WKEY**, select *Bevel* from the menu and click *OK*

- In the *Recursion: 1 OK* dialog, Press **ENTER** (it will revert to the black on grey) and click on the *OK* button. (For Blender version 2.48 and above, skip this step)
- Note : The recursion levels go from 1 to 4. At 1 it seems to be crude but the cube will be subdivided later, so a higher recursion level is not required.*
- Set the bevel to 0.1 by typing **.1** and pressing **ENTER**



## Divide the faces

A typical die face is made of a grid with 9 possible spot positions. The gaps between the spots usually measure half a spot (a spot radius) as unit, a die face will be conveniently divided into 10 equal parts through the length (that is, four gaps and three spots each two units long).



### First division

Make sure you're still in *Edit Mode* (else hit **TAB**), that you're in *Front view* (**NUM1**), and that the whole cube is still selected.

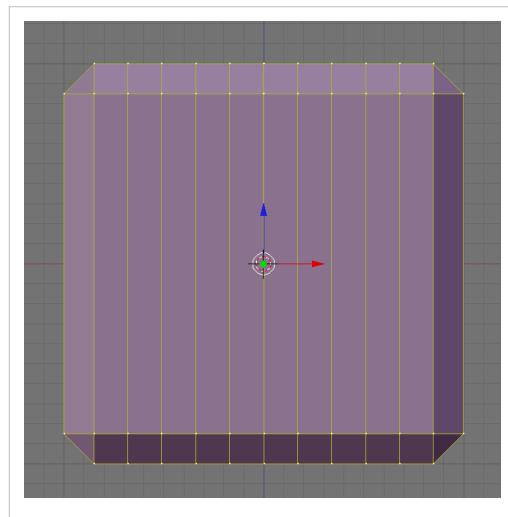
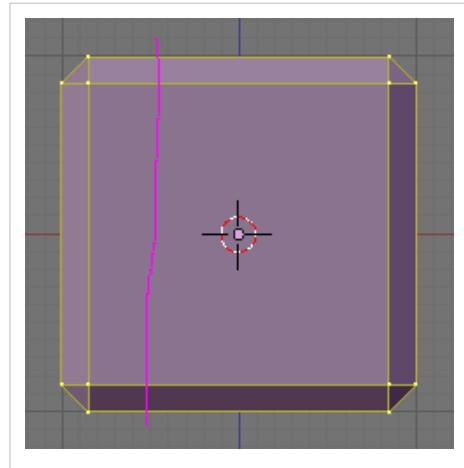
To divide the faces of the cube into 10 vertical strips, press the **KKEY** and choose *Knife (Multicut)* from the popup menu.

A new popup asks: *Number of Cuts:* 2. Change the value to 9:

- click on the *of* word and type 9,
- or drag from the input field (**LMB**) until you reach 9,
- or use the right arrow.

Then click on *OK*.

To actually use the *Multicut* tool, draw one vertical cut through the cube: the knife icon traces a pink line. It doesn't have to be exactly vertical nor does it have to go through the cube center.



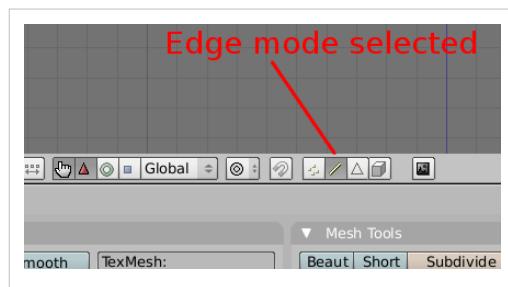
Finally press **ENTER** and all the faces of the cube crossing the knife will be divided into 10 vertical strips.

**'Note:** you can have a look at all the other faces by dragging with the **MMB**.

### Remove extra cuts

To get each spot position to be a single face rather than four contiguous faces, the second, fifth and eighth cuts must be removed.

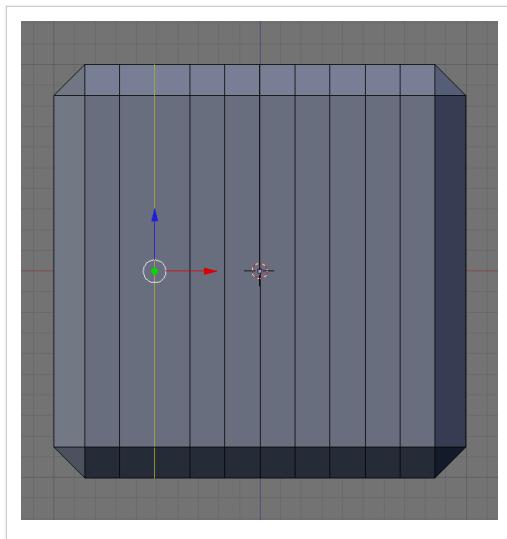
Press the **AKEY** to deselect all the vertices, switch to *Edge select mode* and make sure *Limit selection to visible* is off.



Select an edge from the second cut, and choose *Select → Edge Loop* from the viewport menu. This automatically select all the edges around the cube (a loop).

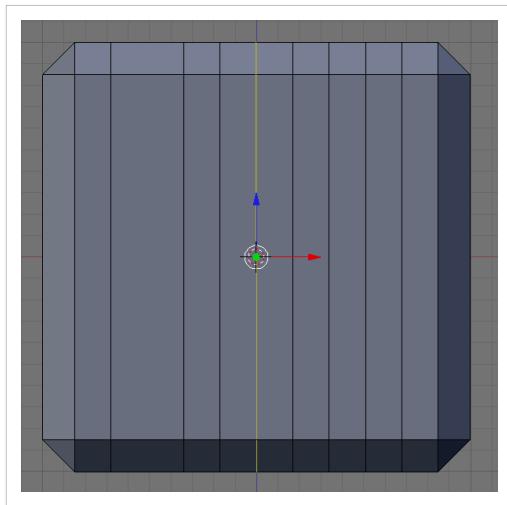
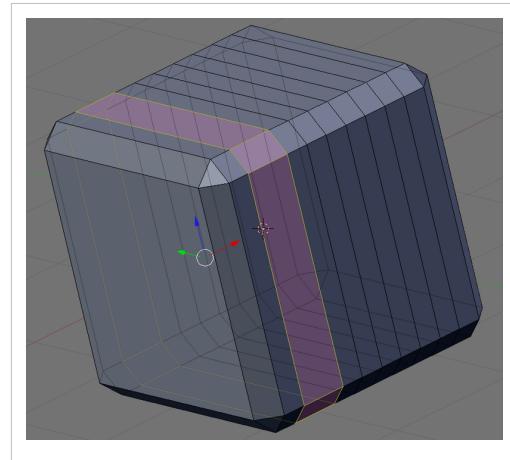
--195.33.69.68 (talk) 09:51, 23 April 2008 (UTC): You can alternatively press the **ALT** key while selecting an edge from the second cut to select all the edges belonging to this cut around the cube.





Press the **XKEY** and select *Edge Loop* from the popup menu. The cube will look like this:

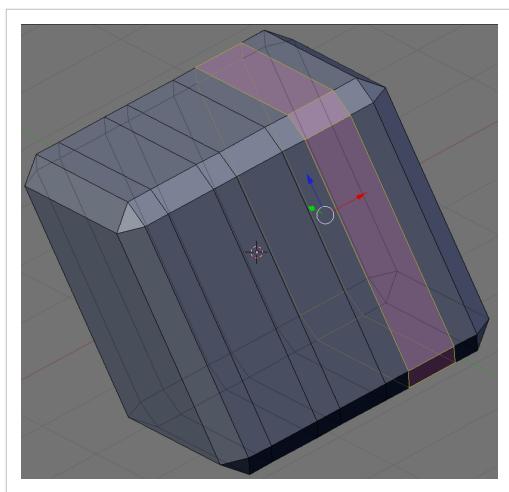
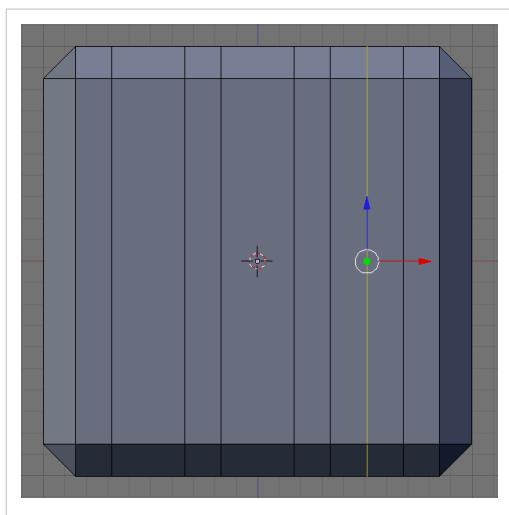
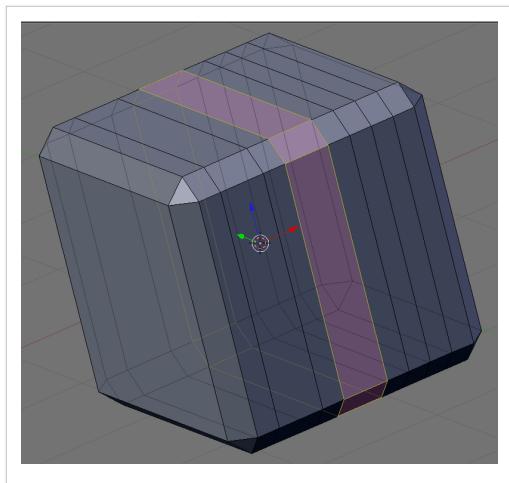
Deselect all (**AKEY**) and repeat the same actions to remove the fifth and eighth cuts:

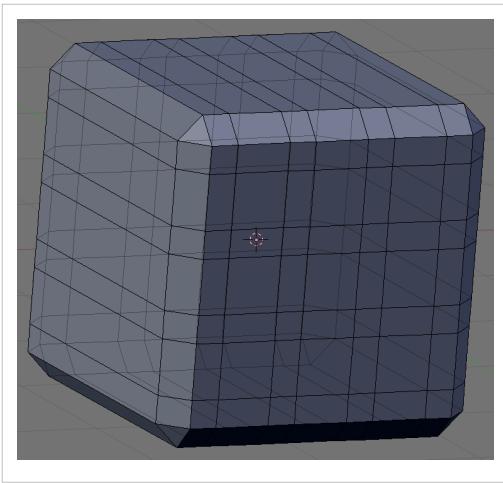
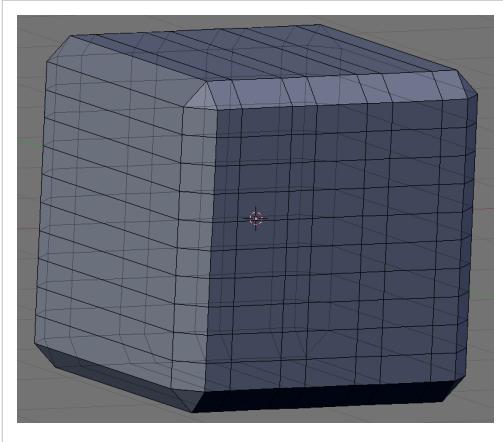


## Other divisions

For the horizontal cuts use the same steps as above, but horizontally!

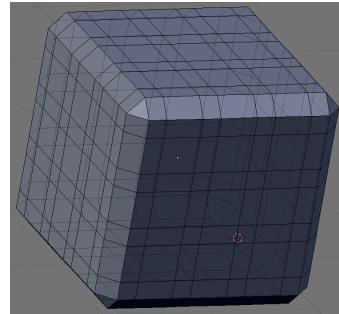
Noob Note: Make sure the whole cube is selected when cutting.





Now, if you rotate the cube around, you will see that the front and back faces (**CTRL+NUM1** to see the back face) are divided vertically and horizontally. But the four other ones were only divided once.

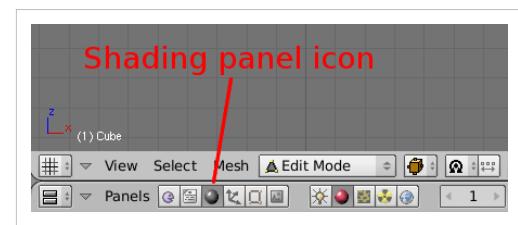
To complete the divisions, go to the *Side* view (**NUM3**), select the whole cube (**AKEY**). Make the vertical cuts and remove the second, fifth and eighth cuts again:

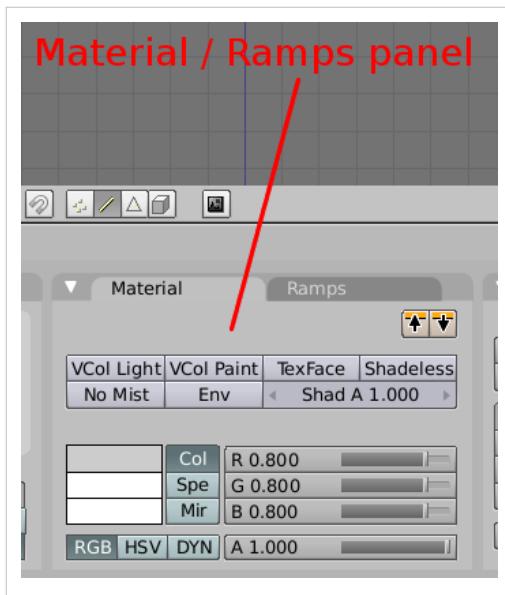


## Colouring the Cube

Select the *Shading* group from the header in the *Button Window* (**F5**).

Change the name of the material (which for example defines colours) currently linked to the cube: in the *Links and Pipeline* panel, under the title "Link to Object", click on "Material" in the "MA:Material" box. "Material" will now show over a red background: delete it and type "DieWhite" instead, then press **ENTER**.



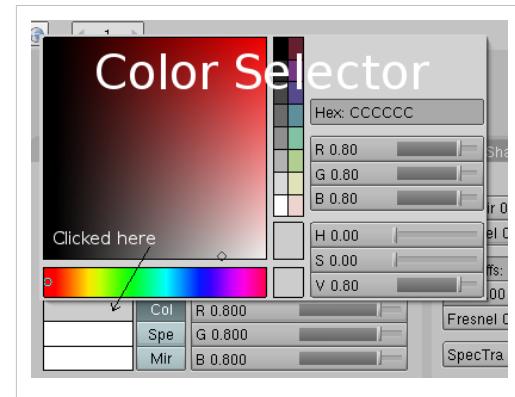


Now choose a color: go to the next panel (the one with two tabs, *Material* and *Ramps*), and make sure the *Material* tab is selected.

In the second group of buttons, click on the colour bar which is on the left of the *Col* button: it opens the colour selector.

Select the colour you want in the rainbow bar (bottom left of the popup), move the tiny circle in the colour window above it to adjust the shade. When done, press **ENTER**.

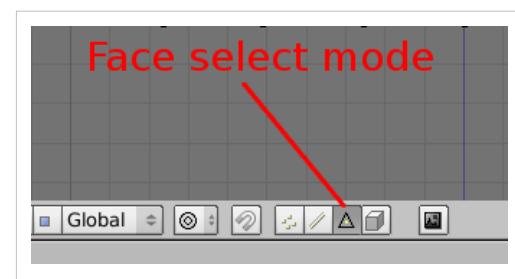
**Note:** you can choose any colour. In this tutorial we selected the white colour by clicking on the corresponding box available from the sixteen predefined ones (the two vertical rows in the middle).



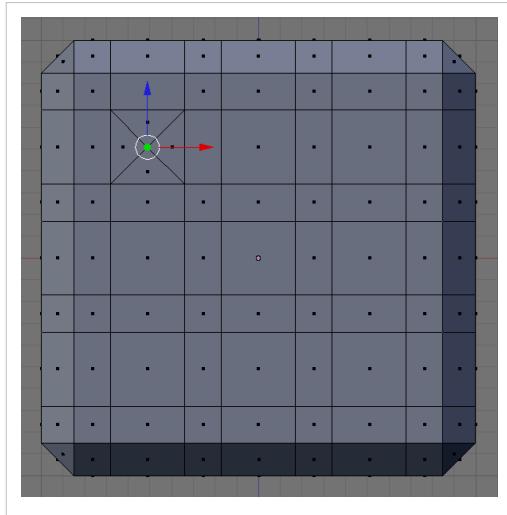
## Adding the spots

Go to the *Front* view (**NUM1**) and set *Face select mode*. Also make sure that *Limit selection to visible* is enabled.

We will create two different kinds of spots.



## First method



This one's from the *Die Another Way* tutorial.

We'll make the number five on the current face. Deselect all (**AKEY**) and select a first spot (the one in the top left corner).

Extrude the face by hitting **EKEY** and then **ESC** (do not click after hitting **EKEY**).

Collapse this face by using **ALT+M** and choosing *At Center* from the popup menu. This merges the four corners of the extruded face into its centre. Click on the message saying *Removed 3 Vertices*.

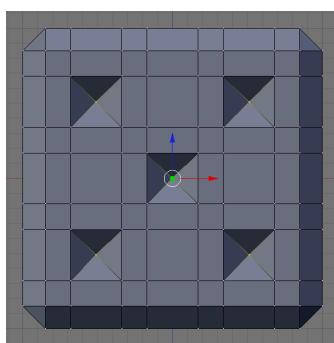
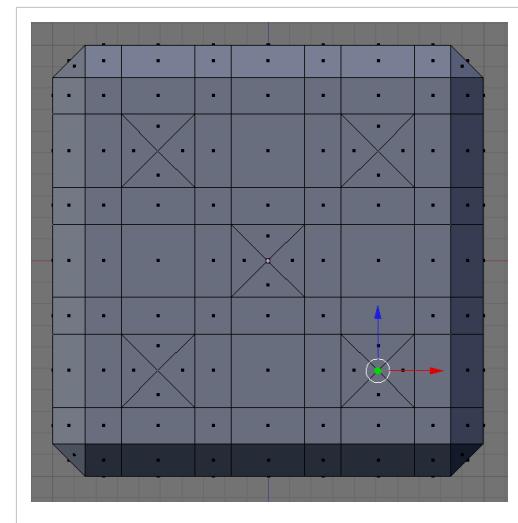
Repeat the same keystroke sequence on the four other spots:  
**EKEY, ESC, ALT+M, ENTER, ENTER.**

Now choose the *Vertex select mode* and select five vertices, one at each spot center (**RMB** for the first one, **SHIFT+RMB** for the subsequent ones). Then push the selection inwards using the **GKEY** and constraining the move to the Y axis (**YKEY**). Move by 0.17 units to get the same die as shown in the picture below.

**--MSK61 (talk) 12:44, 23 April 2008 (UTC):** I think what's meant here is to select five vertices, one at each spot center. The above description may be interpreted as selecting five vertices(the center plus 4 corners) at each spot of the five spots.

**--CamStone (talk) 13:19, 25 Jun 2009 (UTC):** Nope, you're supposed to be selecting faces: the five faces that will be the spots on this side of the die.

**Mast3rlinkx (talk) 23:31, 28 January 2010 (UTC):** CamStone, you're one step behind. MSK61, the original editor said the same thing you just said.



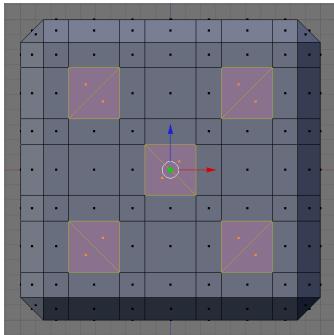
## Second method

This method was suggested by Natume in the *Die Easy* tutorial and modified by another user.

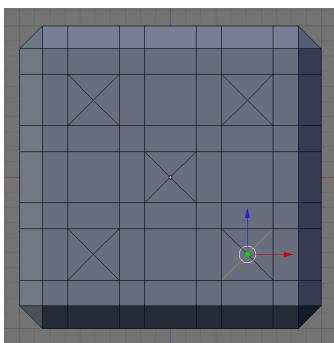
Deselect all (**AKEY**) and select the faces you want to create spots on (**RMB** for the first one, **SHIFT+RMB** for subsequent ones).

Press the **EKEY** to extrude them, if you selected more than one face a popup menu will give you two options: press **ENTER** to select *Region* (the choice doesn't matter), and then press **ESC**. Do not click after hitting **EKEY**.

Divide the selected faces into triangles with **CTRL+TKEY** (or **SPACE** → *Edit* → *Faces* → *Convert to Triangles*).



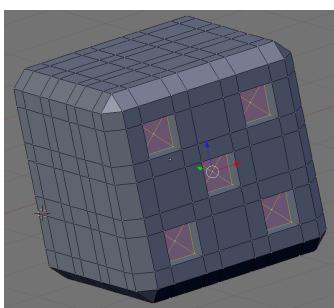
Deselect all (**AKEY**) and change to *Edge select mode*. Select all the diagonal edges resulting from the above divisions, and subdivide them: **WKEY+1KEY** (or **SPACE** → *Edit* → *Edges* → *Subdivide*).



Now deselect all (**AKEY**), go into *Vertex select mode* and select all five vertices of each spot face (for example, use circle select with **BKEY**, **BKEY**, and adjust the size of the circle with **SCROLL**, then click **LMB** to select; click **RMB** when the selection is complete).

(noob note) (Running V.2.49) When building the die with this method (method #2) I do not see the benefit of dividing and subdividing the faces for the spots. Learning the method of doing so was valuable for future work, however it is unnecessary to do so in order to create the desired effect for this exercise. Instead simply select the faces that will be used as spots and extrude them inward by the prescribed number of units (either .17 or -.17 depending on the side).

Push the faces inward by 0.17 units (**GKEY**, **YKEY**).



## The other die faces

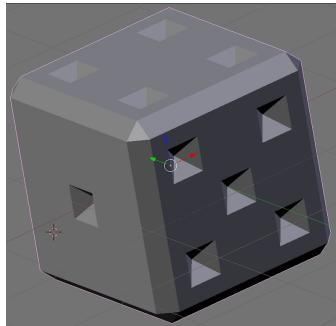
Now repeat the method to the other faces. Remember that adding two opposite face values (of the die) yields 7.

If you started with the *Front* view (**NUM1**) and set 5 spots, the *Back* view (**CTRL+NUM1**) will have 2 spots (the extrusion length will be -0.17 instead of 0.17).

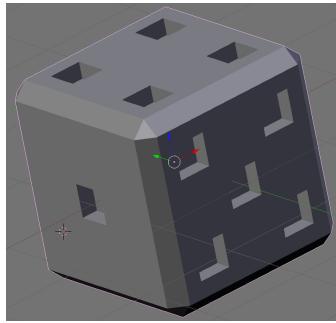
If you then go to Side view (**NUM3**) — the righthand side — and put 6 spots there (extrusion -0.17 along the X axis, **XKEY**), the lefthand side (**CTRL-NUM3**) will have 1 (extrusion 0.17).

If you put 4 on the top (Top view **NUM7**, extrusion -0.17 along the Z axis, **ZKEY**) then the bottom (**CTRL-NUM7**) will have 3 (extrusion 0.17).

When all the spots have been created, the cube should look like this — the top lefthand corner is nearest, using the first method (named *Alt-M*)



using the second method (named *Ctrl-T*)



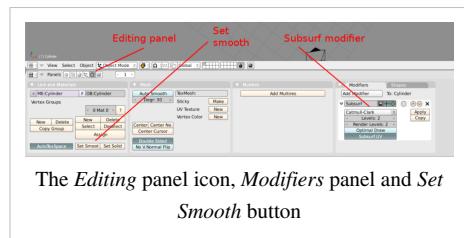
## Subsurfing and smoothing

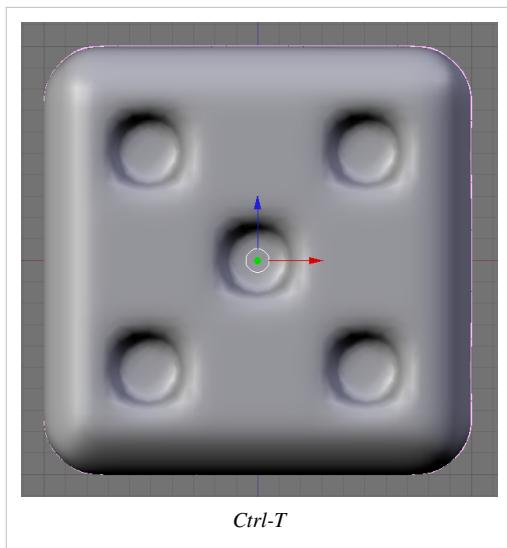
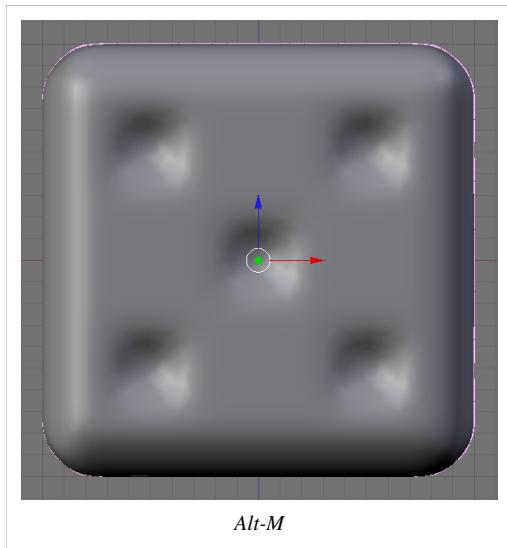
Leave the *Edit Mode* for the *Object Mode* (**TAB**) and select the *Editing* panel in the *Buttons* window (or press **F9**).

To turn on subsurfing, click on *Add Modifier* in the *Modifiers* panel (if you can't see the button, make sure the tab is selected) and select *Subsurf* from the popup menu.

The default value for *Levels* is 1 and for *Render Levels* is 2. For a nicer preview, change *Levels* from 1 to 2 by clicking once on the right arrow, and click on *Set Smooth* in the *Links and Materials* panel.

The cube will now look like this (*Front* view):

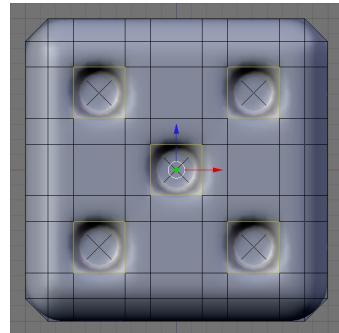




## Subsurface creasing

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

Go back to *Edit Mode* and *Edge select mode*, select every spot perimeter as shown below (**RMB** for the first one, **SHIFT-RMB** for the others):



Press **SHIFT+EKEY** to enable creasing, then hit **1KEY** (not the numpad key!) or set the value with the mouse. Hit **ENTER** to set the new value.

**User Note:** I use 2.49b, and numpad keys can also be used to set the value.

**Noob Note:** This is where you lose me. The step is much too vague. I followed the instructions exactly as laid out, and no matter how many times I try, nothing happens at all. Is this "Noob-to-Pro" or is this "Pro-to-Pro"? More accurate step-by-step instructions, please.

**'Noob-147 Notes:** Here is the step by step I use to do this action on Rev 2.48a: Header bar (just below the big work window) should say "Edit Mode". On the same bar most of the way to the

right I have "Edge select mode" on. This looks like a forward slash or perhaps two parallel lines and you can either click it or press Ctrl-Tab 2. A little further right I have the box like icon on to occlude background geometry so I don't select any edges on the opposite side of the cube. Then I press:

A-key or A,A-key to be sure everything is deselected.

B,B-key to TURN ON circle select mode.

[MMW] roll (don't press) middle mouse wheel to size the circle small enough to select single edges.

[LMB]x20 I have 4 edges around each of 5 "dots" so I click 20 times with the left mouse button (once on each edge) to select them all. If I select an extra unwanted edge anywhere Then I just click the wheel on it. (My wheel rolls or presses in.)

[RMB] to TURN OFF circle select mode.

Now move mouse pointer crosshairs (don't press any button here) near center of drawing.

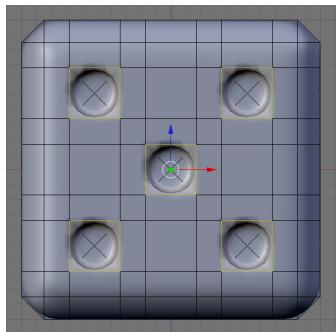
Shift-E-key to turn on Edge Crease mode. The bottom bar changes to a "Crease" number.

Now move mouse pointer crosshairs (don't press any button here) away from center of drawing. Watch the Crease number change as you pull out. It maxes out at 1.000.

[LMB] When you get the crease you want the left mouse button locks it in place.

A-key to deselect every thing.

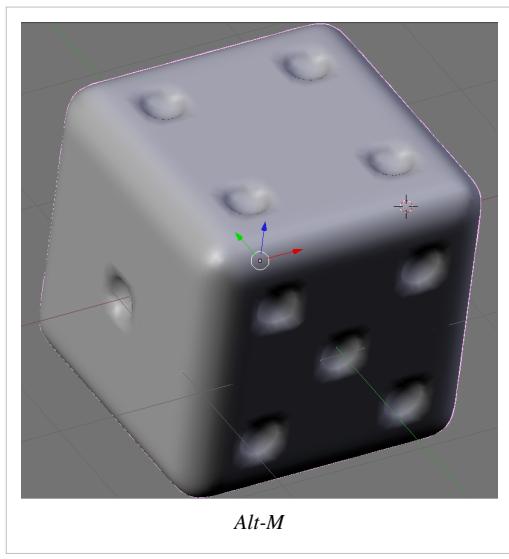
Finished on this side. View a different side and do it all again.



**Note:** the **NKEY** shows an information box (with the perimeter edges still selected). The last line says: Median Crease W: 1.000, if the value is less than 1.000 it means that an edge has been missed. You can update the crease values by clicking on the arrows. Click on the X in the top right corner to dismiss the dialog.

**Noob Note:** setting the value of the subsurf creasing affects the subsurf modifier. You will not actually see the creasing effect unless the object has the subsurf modifier active. Try adding the subsurf modifier to the object then set the creasing to one so that you can see the effect it actually has on the object.

Now crease the spots on the five other sides of the die. You should end up with:



## Colouring the spots

### Creating a new material

Go to the *Editing* panel (**F9**) in the *Buttons Window*.

The *Links and Materials* panel should be showing (in the right column) *DieWhite* and, under it, *1 Mat 1*:

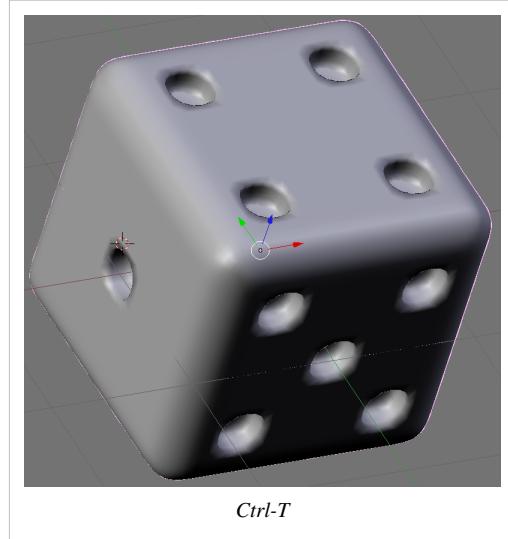


To add another material, click on the *New* button (right under *1 Mat 1*). This label changes to *2 Mat 2*, and *DieWhite* changes to *DieWhite.001*.

**Note:** if your Blender version is older than 2.44, there's an additional step. Go to the *Links and Pipeline* panel (Shading or **F5**), click on the *2* button right after the *MA:DieWhite* input and choose *Single user* in the *popup menu*.

*user request: please provide instructions for 2.45, 'Links and materials' panel has been replaced with 'Links and Pipeline' panel. 'ADD NEW' from dropdown initially containing 'MA: Material'*

*does not increment the '1 Mat 1' fields leaving no clear way to add more than one material to an object.* -- **Answer from another user:** You're looking at the Shading panel (**F5**), you need to switch to the Editing panel (**F9**), there you'll find the buttons just as described here.



Set the colour for this new material: click on the white square on the left of *2 Mat 2* (in the *Editing* panels) or on the left of *Col* (in the *Shading* panel). To get the same colours as in this tutorial, choose black. Press **ENTER** to confirm your choice.

In the *Links and Pipeline* panel (*Shading* or **F5**), change the material name from *DieWhite.001* to *SpotBlack*.

### Selecting the spot faces

Now go into *Edit Mode*, choose *Face select mode* and make sure *Limit selection to visible* is on.

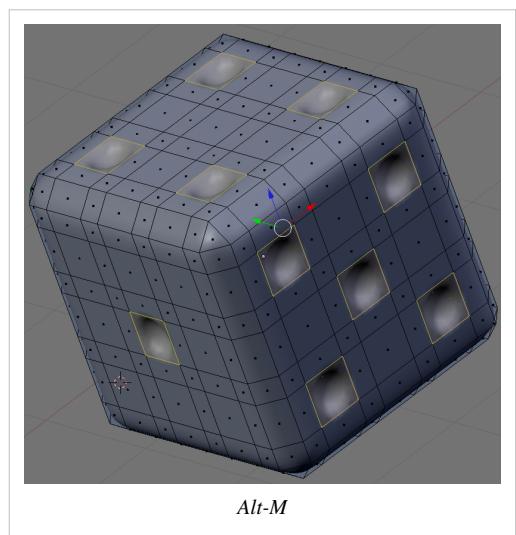
Version 2.46 The "Limit Selection To Visible", is now "Occlude Background Geometry". It remains the button to the right of "Face Select"

Starting again with Front view (**NUM1**) use circle select (**BKEY, BKEY** — use **SCROLL** to reduce the size of the circle to less than the spot size so you don't select the outside faces) to select all the inside faces of each spot.

Press **RMB** to finish, select the *Back* view (**CTRL+NUM1**), use circle select again and select the spot faces there, do the same with the other four die faces.

**'Note:** you can't change to a different face without stopping the circle select tool (**RMB**). You will have to click **RMB**, change to another face, and start circle select again (**BKEY,BKEY**).

This is what you should see when you have selected all the spots:



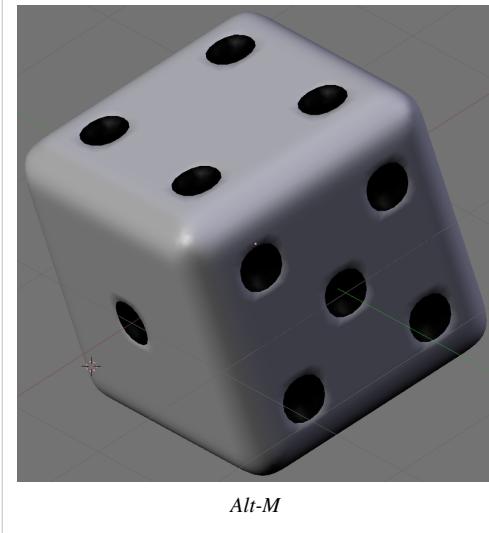
The selected faces internal to the spots are more visible in the *Alt-M* version as the spots created in *Ctrl-T* are more cylindrical; the *Alt-M* spots look more like hemispheres.

### Assigning the new material to the spots

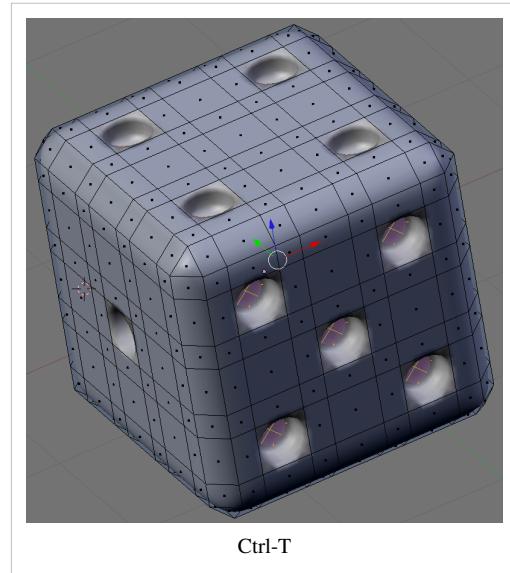
Now that all the spot faces are selected, you can assign the *SpotBlack* material to them.

Go back to the *Editing* panels (**F9**) and, in the *Links and Materials* panel make sure the material *SpotBlack* is selected. If it isn't, you can select it by clicking on the right arrow or the *I* on the right of *2 Mat 1*, which should update to *2 Mat 2*. Now click on the wide *Assign* button below and the spots will change to black (or the other colour you chose).

Here are the finished results:



*Alt-M*



*Ctrl-T*

## Afterword

Here's an external link to a tutorial showing a completely different way to create a die:

- [http://www.blendernation.com/tutorials/  
blender-3d-beginner-tutorial-a-die/](http://www.blendernation.com/tutorials/blender-3d-beginner-tutorial-a-die/)

The most efficient and easy tutorial can be viewed at:

- <http://www.youtube.com/watch?v=ZTwPvQbhHdY>



## Blender 3D: Noob to Pro/Model a Silver Goblet

---

### Basic Shape

**NOTE: If you find this section nightmarishly incomprehensible (as I did), feel free to zoom down to the "Alternative Method", which is explained much better.**

On first pass, the goblet looks like it is composed of a cylinder(s). While it is possible to model the goblet with a cylinder mesh, we can just use a cube. What is the benefit of using a cube? It is faster to make and there are fewer vertices to track.

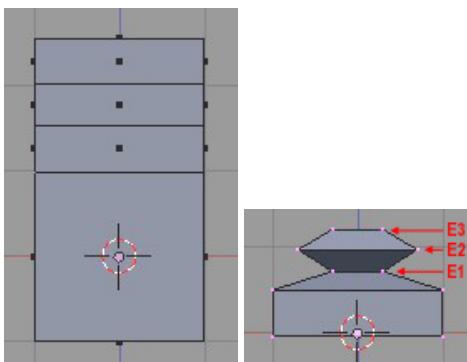
--Chuey 14:22, 6 June 2007 (UTC) (Noob) *Fastest way I use to make any cylindrical object is to first create a cross-section of half the goblet using vertices connected by edges. Then place the cursor on the center vertex and revolve around the center axis (Editing – Mesh Tools – Spin Button) 360 degrees. This allows you to create complex geometry very easily. The earlier Creating a Simple Hat tutorial explained how to do this.*

--note that if you use the above method, the goblet is made of 2d faces, so the actual "cup" part doesn't have a 3 dimensional feel to it. (I'm sure there is a way to correct this, but it seems beyond the scope of this tutorial). It is a great method for making completely solid objects with no hollow parts.

--Note that if you take the above notice to heart, remember that you can make an initial mesh that accounts for thickness and spin that.

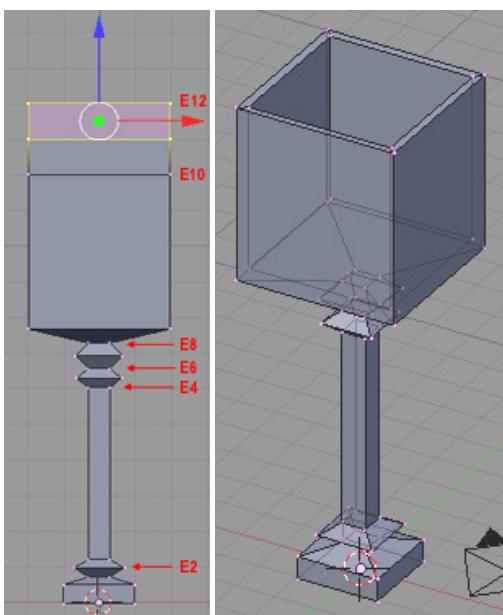
Starting with the default cube in **NUM 1** view and Edit Mode, extrude the top face three times. Scale each extrusion vertices to create the first knob. This example scaled E1 and E3 to 0.300 and E2 to 0.700.

Note that it would be extremely difficult to scale later extrusions in this tutorial if you scaled the first 3 extrusions(the ones you've just made) now because later extrusions will be based on the scaled areas of the first 3 extrusions(precisely the third extrusion). A better approach is to extrude 12 times subsequently, then scale every extrusion individually(after making all extrusions)"I dont believe that i did it your way and found it harder" . *Noobie Jon says: You can scale/extrude as the tutorial says; just scale the vertices that are supposed to be .700 after the initial 3 extrusions to 2.333. It's not difficult, just basic math. "Noobie Dave says: This pic shows how i think the extrusions should go* <http://i31.photobucket.com/albums/c366/1h23/Blender/Goblet.png> *This is before you re-size them, now follow the rest of the tutorial. I hope this helps*



Continue extruding up. This design will take 12 total extrusions - 3 to make the first knob + 1 for the stem + 4 to make two upper knobs + 4 for the cup.

After moving and scaling the first 10 extrusions, select the top 8 vertices (shown below). Scale these in along X and Y to 0.900. This will make them only slightly smaller than what will be the outside of the cup. Pull E11 down so that it is level with E10. This makes the rim. Finally, pull E12 down inside of the cube to form a container.



**Noob Note:** Because this section is really a bit vague I will try to explain it step by step: first select the top 4 vertices of the cup then extrude (EKEY) them by e.g. 1 in Z-direction -> then scale (SKEY) them to 0.9 -> hit GKEY and -1 in Z-direction (back to the top of the cup) -> now hit the EKEY and bring those 4 vertices back to the bottom of your cup (good to know how deep your cup is...) HTH

(Yoshi says Note: This section is vague so try experimenting, the main thing to do is to make knobs by making a small box, then a big one, then a big one, then a small one. (If you follow the tutorial: small = scale to 0.3, big = 2.333). Then at the end of the cup make a box the same size than hit G then Z and pull it down.

**Alternative Method:** The following is an outward scale & extrusion from the cube.

Looking blissfully at a 'new' cube, in Edit Mode. Move to the side view (**NUM3KEY**). Box select (**BKEY**) the top edges of the cube. Extrude (**EKEY** → *Regions*) upward about one grid square. Hold down **CTRL** while extruding, for incremental movement, and press the **ZKEY** to restrict movement to the Z-axis. Repeat 2 more times. Now Extrude a longer piece, for the Goblets stem, of approx. 10 grid squares. Next we will define the area for the 2 top knobs & the bottom of the glass, by Extruding upwards 5 more times at 1 grid square each. The glass itself will consist of another upward Extrusion of about 10 grid squares.... resulting in a tall rectangle, with several horizontal edges... minus the E11 & E12 extrusions as seen in the image provided below.

Now, lets begin to inflate the glass. First, clear all selections by hitting the **AKEY**, once or twice. Starting at the base & still in the side view (**NUM3KEY**). Box select (**BKEY**) the original cube, at the bottom. Then expand it outward

by scaling (**SKEY**), followed by '**SHIFT+Z**' to lock/prevent any scaling in height along the Z-axis. Repeat in a similar manner for the ornate knobs (what would be E2, E5, & E7), 1 below the stem & 2 above, just use your imagination. Then expand the top, Goblet (vessel) rectangle, until it is about 12 grids in width.

Last is the cups interior. Once again, you should still be in the side view (**NUM3KEY**). Box select (**BKEY**) the very top most edges of the Goblet. Once selected, get a better view by changing to 'Orbit Up' (**NUM8KEY**). With the top surface selected, initiate an Extrusion (**EKEY** → *Regions*), followed by a termination with the **ESCAPEKEY**. It will appear as if nothing has happened, but new (overlapping) edges have been created & are now selected. Next Scale (**SKEY**) the selected edges to create the inside lip of the cup. You can eyeball it, or just type **.8** and press the **ENTERKEY** as a thickness. Almost finished. Extrude (**EKEY**) the interior lip, along the Z-axis (**ZKEY**) downward to create the bottom interior of the Goblet. Finish off with a 'Remove Doubles' (**WKEY** → **Remove Doubles**), for good measure.

**Note:** If the thickness of the vessel sides or bottom is too thin. The Goblets interior faces may 'bleed through' affecting the exterior of the Goblet, when you apply a subsurf.

You many want to try modeling the inside/bottom of the Goblet (sim. to "Die Another Way" tutorial). Turning on "Occlude Background Geometry" will be helpful, as it turns off the ability to see & select through the model. Although, there is currently no Hotkey for it (should be **CTRL+TAB+4...** but it's not). The button icon is that of a "ISO Cube" located with the 3 select mode buttons (4 dots, 2 lines, & a triangle) of the 3D Toolbar, in Edit Mode.

## Smoothing and Defining

Time to take the mesh and turn it into a proper goblet. Add a subsurf level 2 to the mesh. Change to Object Mode (**TAB**) and select the Set Smooth button under *Links and Materials*. The cube-looking mesh will now look like an object that was created from a cylinder. This has removed all our crisp edges, and our goblet is looking very unstable! Let's rectify things by flattening the bottom.

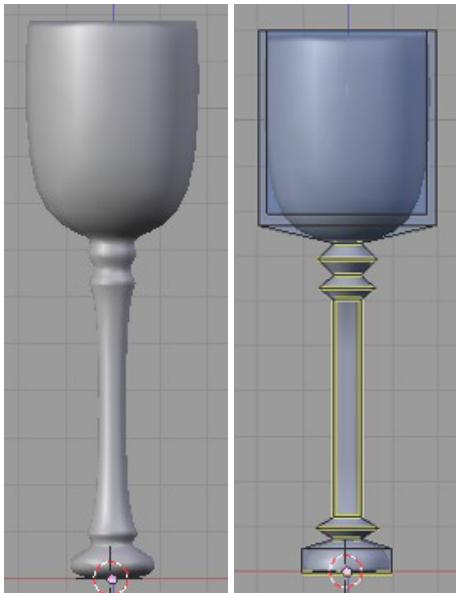
*Noobie Jon says: If your goblet has a bulge in it after applying the subsurf, go to edit mode (**Tab**), select all (**AKEY**), press **WKEY**, and select "remove doubles".*

[Noob Question: The Newer version (2.49b) of blender uses **WKEY** for boolean operations, what would I press then??] -answer- you must be sure to be in edit mode before using **WKEY**.

Select the four edges that surround the small circle at the very bottom (the very lowest set of edges) and press **SHIFT + EKEY** to enable creasing. Now drag the mouse up and down to select the level of the crease. When you're satisfied, hit **LMB**. Repeat the process for other edges you want to be sharp. I've turned on Draw Creases under *Mesh Tools 1* to illustrate which edges have been creased (highlighted yellow) in this example.

Note: Due to the vagueness of this tutorial I had problems trying to follow what was wanted. I got around it by beveling the goblet (for the cup-edge/thickness), and then merging necessary vertex. Here is how it turned-out:

Goblet <sup>[1]</sup> Render <sup>[2]</sup>



That concludes the creation of the goblet. Save the scene for use in the lighting tutorial. To jump to the relevant lighting section, go to Blender 3D: Noob to Pro/Light a Silver Goblet

## Transparency

Okay, this is pretty much just because I wanted to show off my work, but also because transparency was really hard to find and really easy to actually **do**. I don't know if there's a lesson on transparency in this, but I really wanted glass, and I'm impatient. This little bit is for anyone else who's interested in modeling the goblet as glass and is impatient like me. If you try this, first keep the non-transparent goblet saved somewhere, it is used in the "silver goblet and lighting" follow-up tutorial linked above.

Hit F5 to go the texture context and then click the red ball icon to open the material buttons.

The following images show the settings needed, and the final render of my green glass goblet.

In Materials is an Alpha slider- goes from 0.00 (invisible) to 1.00 (opaque). It needs to be pretty low, less than .25, I'd say. Also in the Edit buttons (F5) is a section called "Mirror Transp", for me it's a tab between "Shaders" and "SSS". In here the "Ray Transp" button must be on. The IOR value is refraction rate- the value of glass is 1.51714. Next to that is Depth, determines how deep to apply transparency. I recommend maxing it to 10, to play it safe.



- Tilescreen's addition

Ray transp stands for Ray Tracing Transparency, this basically involves shooting light out from the camera and bouncing it about your 3d scene. The depth value of the ray tracing settings determines how many times a ray of light can 'mirror' (reflect off a surface) or 'transp' (refract through a surface, just like at school with prisms and the light bends). It is possible to work out how much depth you need as each time it enters or exits a surface you will need +1 level of depth. So for this you should need about 4 or 5 depending on how complex you have made this goblet. If you don't have a sufficient depth it will just come out a solid colour with no transparency.

More on Ray tracing. Samples refers to how many beams of light will be fired out of the camera to calculate the refraction and reflection. higher numbers = better quality but also increase render time, but the real killer is the depth. Increasing it by one means it has to work out an (in my case) an additional 18 bounces. Also try adding a filter to slightly blur your reflections and refractions but only a tiny one.

To get a softer shadow then click on the lamp, press F5 and increase the samples (found in Shadow and Spot) this will shoot out more shadow rays and give you softer shadows.

- End of my addition
- Yoshi's addition:

Try experimenting with the colours. The easiest way I know to do this is:

Select the goblet in object mode. Go to the textures tab (next to the red ball), and under 'texture type' pick 'blend'. Then click 'colours' next to the textures subwindow, and hit 'colorband'. You can change the colours by click on the left of the bar, then double clicking the coloured bar under it. Once you pick the darkest colour (left), click the far right of the checkered bar and do the same as before.

- End of Yoshi's addition



If you want a shadow, you'll need to create a plane and go into the "Shader" settings for it. Turn on "TraShado", otherwise the shadow will be a flat black one and look all wrong.

*Noob says: Even when I turn on TraShado, the shadow is flat black... I can't get those lighter edge shadows. Solution: Make sure that TraShado is applied to the plane rather than the cube and that Shadow is also selected.*

Noob Note: Do not try and use this goblet for a fluid simulation it will not work.

**Reader note:** If the ray tracing transparency does not work for you, make sure that the ray tracing is enabled in the **Scene (F10)** menu. It's the **Ray** button in the *Render* panel.

*Noob says: No matter what I do, my goblet stays opaque and my goblet has no shadow when on a plane. Do I have to somehow link my goblet to my plane?*

noob answer:adjust the alpa press (f5) look in the material

and adjust the (A)slide under RGB slides (bottom right) to somthing like .250

Here's the final result:



Noob Note: If you have problems with a sandy texture showing up (like in the picture above) your cup is clipping through the plane, just move it up a little bit on the Z axis, and it should go away

## References

- [1] <http://i88.photobucket.com/albums/k185/captheathmalc/gobletinblender.jpg>
  - [2] <http://i88.photobucket.com/albums/k185/captheathmalc/goblet.jpg>
-

# Blender 3D: Noob to Pro/Model a Silver Goblet cylinder

## Preliminaries

This is a version of "Model a Silver Goblet" but starting from a cylinder rather than a cube.

Start Blender or start a new scene (**CTRL-XKEY**) and delete the default cube (in object mode, press **XKey**).

Change to Top view (**NUM7**), make sure you're in Object mode and add a cylinder (**SPACE Add > Mesh > Cylinder**): in the popup menu, change the number of vertices from 32 to 16, the Radius from 1.000 to 1.800, the Depth from 2.000 to .100 and leave "Cap Ends" as it is.

Change to Front view (**NUM1**).

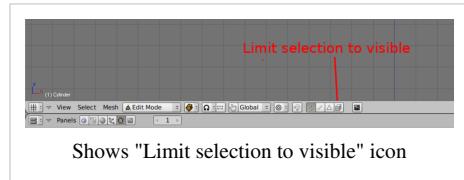
## Description of the modeling steps

The steps in this tutorial are almost all made up of "extrude" and "scale": so, to avoid repeating key sequences every time,

extrude

means

1. make sure "Limit selection to visible" is off
2. box select the top vertices of the cylinder: press **BKEY**, click and drag **LMB** to make a rectangle around the top vertices
3. press **EKEY**, select "Region" from the popup menu, press **ZKEY** and type in the amount of the extrusion and press **ENTER**; you can move the mouse instead but it is quicker and easier to type it in.



*Newbie question: Why select "Region" and not "Individual faces"?* If you use "Region", the common edge between two neighboring faces will remain a single edge after extrusion. If you use "Individual faces", the edge will become two edges sitting at the same location. This may create problems (black lines when rendering).

For example, the following keystroke sequence extrudes by 1.5:

**BKEY**, click and drag **LMB** to make a rectangle around the top vertices; **EKEY**, select "Region", **ZKEY**, **1KEY**, **.KEY**, **5KEY**, **ENTER**.

scale

scaling is by default restricted to the X-Y plane; although the numbers in the bottom left corner of the 3Dview show Z changing, in fact only X and Y change, and by equal amounts.

1. press **SKEY**, type in the value and press **ENTER** — you can use the mouse instead but it is quicker and easier to type in the number.

## Creating the Goblet

This diagram shows the connection between the E-numbers and the goblet construction.

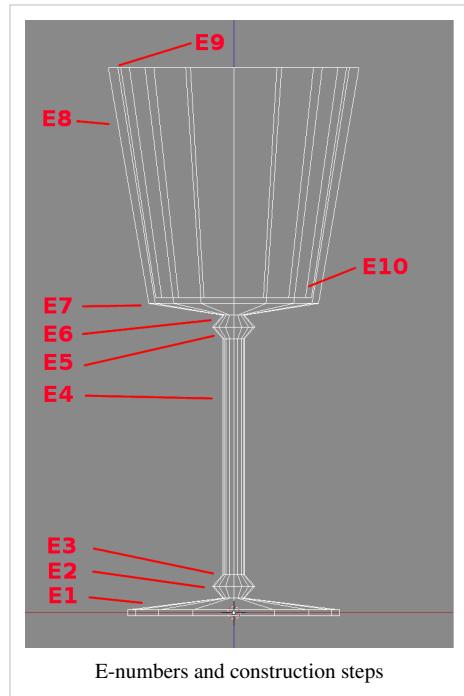
- **E1:** Deselect all vertices (**AKEY**) and extrude by 0.2: you may need to zoom in (**SCROLL-MMB**) to do this as it's quite thin. Scale to 0.1.
- **E2:** Extrude by 0.2, scale by 2.
- **E3:** Extrude by 0.2, scale by 0.5 to make the lower knob.
- **E4:** Extrude by 4 to make the stem.
- **E5:** Extrude by 0.2, scale by 2.
- **E6:** Extrude by 0.2, scale by 0.5 to make the upper knob.
- **E7:** Extrude by 0.2, scale by 8 to make the base of the cup.
- **E8:** Extrude by 4, if you wish to make a flared cup, you can scale by 1.5.
- **E9:** Extrude by 0.0, scale by 0.9 to make the rim of the cup. (This will create a new ring of vertices and then move them in towards the centre.)

Now go into Wireframe mode (**ZKEY**) so you can see inside to guide the next few steps.

- **E10:** Extrude by -3.9, that is, downwards. and scale by 0.69: you can do this last scaling with the mouse, if you like, to get the edges of the inside of the cup and the outside parallel.

You now have a goblet, the base of the inside of the cup is the face of the last extrusion, is circular and flat as it derives from a cylinder.

If you haven't already saved your work-in-progress, now would be a good time.



E-numbers and construction steps

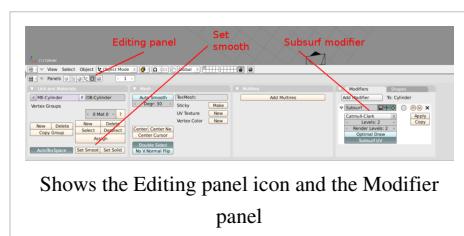
## Subsurfing and smoothing the goblet

The last step is to subsurf and smooth: go into Object mode (**TAB**) and enable Solid mode again (**ZKEY**).

Select the Editing panel from the Buttons window (**F9**) and, in the "Modifiers" panel, click on "Add Modifier", select "Subsurf" from the popup menu and, in the Subsurf display, increase "Levels" from 1 to 2.

At the bottom right of the "Links and Materials" panel, click on the "Set Smooth" button. At the bottom of the cup you will see fluting — this is an artifact caused by smoothing and subsurfing triangles on a curved surface. Here it adds to the appearance, don't you think?

Note: If your cup does not look smooth after adding subsurf and set smooth like mine, you may have to delete all but two of center vertices (Bottom vertex at base of cup and vertex at the bottom inside the cup).



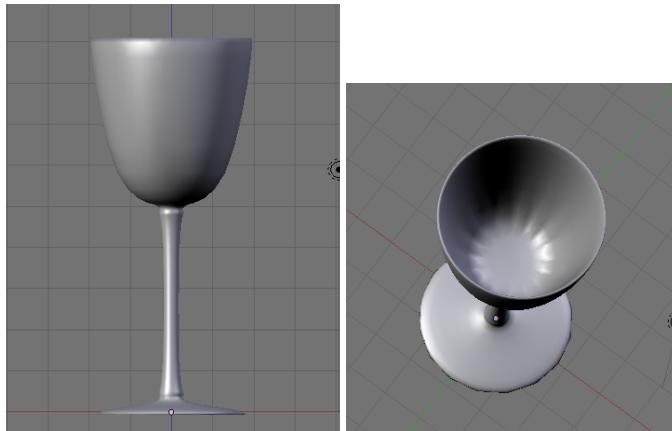
Shows the Editing panel icon and the Modifier panel

## Flattening the base of the goblet

One more thing, the base of the goblet is curved due to the subsurfing and needs to be flattened.

Go into Edit mode, press **AKEY** to deselect all and box select the lowest set of vertices. Now press **SHIFT-EKEY** and type in 1 followed by **ENTER**.

The final result should look something like this:



Save the scene for use in the lighting tutorial. To jump to the relevant lighting section, go to Blender 3D: Noob to Pro/Light a Silver Goblet

## Blender 3D: Noob to Pro/Simple Vehicle

The idea of this tutorial is to learn to face a complex project. A vehicle is a nice object to use to test yourself and find new problems.

First, we must understand that 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.

What vehicle should we make? Let's go with the classic jeep. This will allow for a lot of doodads.

Let's decide what objects of the jeep model will need to be made - body, wheels, seats, and a rocket launcher for good measure. Objects we can ignore include the engine, which remains hidden under the hood. There are many additional objects you can make such as seats and steering wheel to customize your jeep.



Let's make this jeep.

# Blender 3D: Noob to Pro/Simple Vehicle: Wheel

## Techniques

You should already know how to:

- Make a mesh
- Navigate the viewport
- Extrusion
- Create, edit materials

This section will recap and introduce:

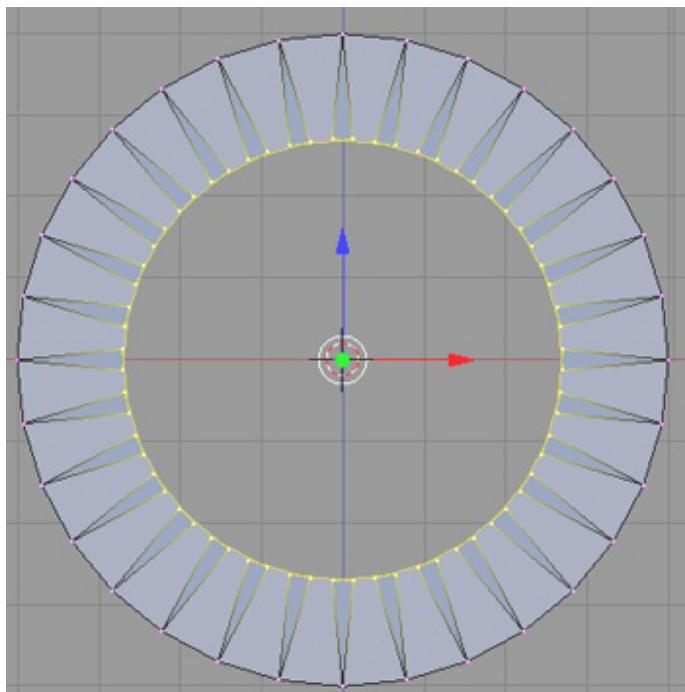
- Forming faces
- Subsurfing
- Merging vertices
- Object naming

For our premise, envision jeep tires. They're not too sleek but rather rugged for all kinds of terrain. We need a tire that can handle any obstacle in its way.

## Model from Scratch

Start with the viewport in XZ coordinates (**NUM1**). Delete the cube and add a cylinder (**SPACE > Add > Mesh > Cylinder**) use 32 vertices, set the radius to 4, depth to 3, and un-click the "Cap Ends" button.

Change to **NUM7** view and make sure you are in Edit Mode. With all the vertices selected, extrude **EKEY** the individual faces into the circle (Make sure to select INDIVIDUAL FACES not Region). Hold **SHIFT CTRL** while extruding the faces until the sides come in -1.200 units, or **just type** in -1.2 and hit enter (in older versions of blender hit **NKEY** first)



Rotate around the model and you'll notice the outside of the tire doesn't yet exist. Let's remedy this. Go back to **NUM1** view.

Invert selection (**SPACE > Select > Inverse**)

Press **FKEY** to create the missing faces

Select **auto**

(*A Noob: I'm using Blender 2.45, and using the F key and selecting auto doesn't do anything. A little help?*)

(another Noob: I'm also using Blender 2.45 and I don't even need to follow this step, as my tire already has an outside. I'm guessing you're encountering the same)

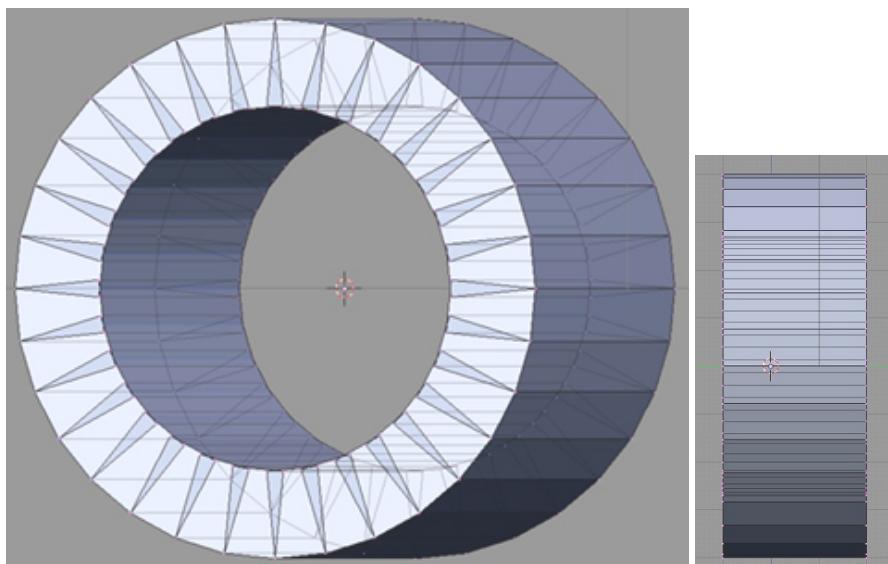
(*THE Noob: I'm using Blender 2.46, and i just skipped the SPACEBAR bit, and selected everything with AKEY and pressed F to fill in the missing parts, works fine.*)

**NOTE:** In Blender 2.42a pressing F gives a menu with Make and Clear options. Choosing Make says "No faces selected to make FGon". How do I make the faces? **Answer:** I'm using Blender 2.44 and pressing F gives a menu where you can choose Auto.

(*Yet another Noob: I'm using 2.37. Couldn't find any way to use Fkey globally. Deselected all, then selected outer vertices in sets of four, then pressed Fkey to get a face. Repeated many times to complete. Does someone have a better idea for the older versions? )*

(*Another Noob: In 2.42a the tire looks terrible if one manually selects the outside vertices and then makes faces with F and subsurfs. Better to make a tube, not a cylinder, a tube with 16 vertices, extrude outward, subsurf it, duplicated it with Shift-D and then rotate that a little along the Y axis to fill in the gaps. Looks better. And it's a much faster procedure too!*)

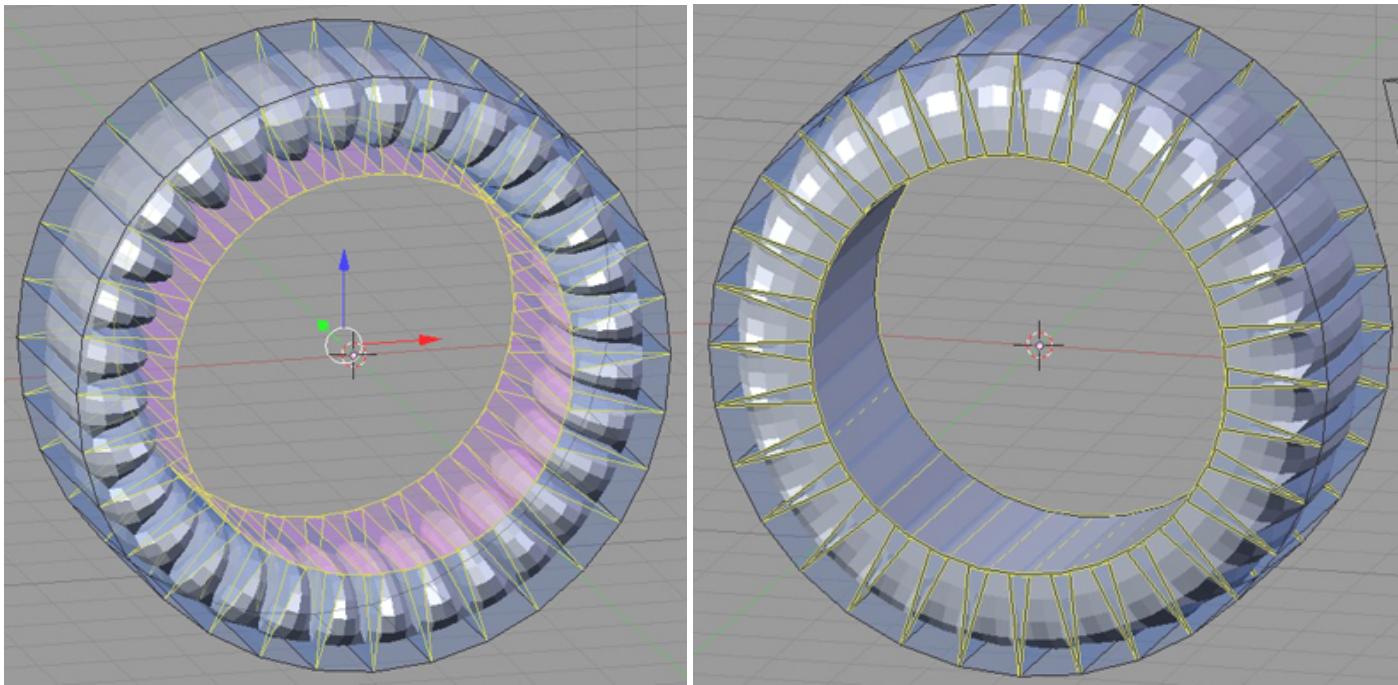
(*Noob Asking: Is there an easy way to eliminate the double vertexes and the hidden geometry or does it has to be one by one?? )*



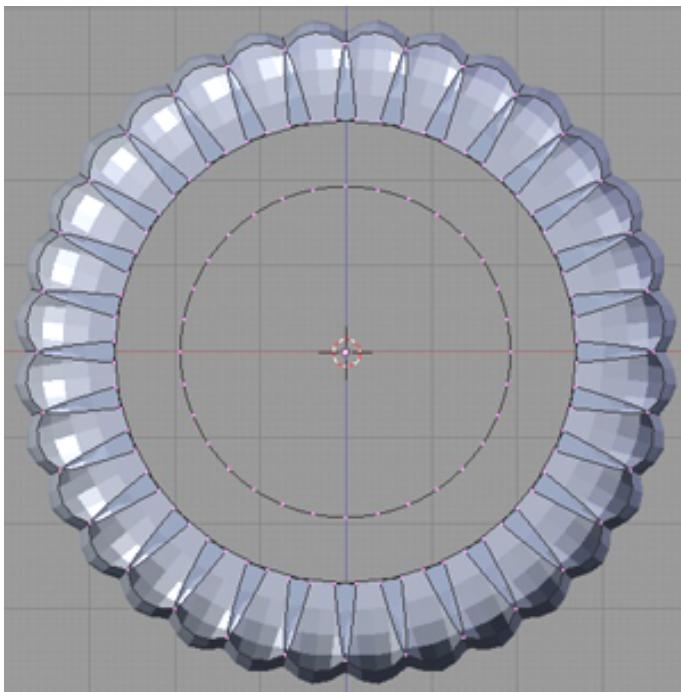
Now its time to make the tire look like a rugged tire. Apply a subsurf of level 1 or 2 to the mesh. The tire will now look like a bead necklace. [*Edit: If not, remove doubles.*] A little creative use of creases will restore our tire.

Turn off the Limit Selection to Visible. Enter select edges mode and bring up the circle selection tool (**AKEY** to unselect all, then **BKEY** twice). Use the scroll wheel to change the circle selection size to be in the center of the tire, between the inside and outside edges. This will select all of the inside edges, as well as the triangles on the side of the tire, as in the picture below. Crease these edges to 1.000 (**SHIFT EKEY**).

(noob note: DO NOT turn the 'limit-selection' tool on for the above selection.) Titlescreen says You could also consider selecting the outside faces on the tyre (the bit that will touch the road) and subdividing them once. Then you can selected a "left face" go down one and select a "right face" etc until you have gone round the wheel. You should have a checkerboard selection on the tyre. Then press E to extrude > Individual faces and pull them out a bit. now press SHIFT + E to crease the edges and you will have a thick and chunky tyre tread.



The tire is almost done. Let's add a simple hubcap to it. In **NUM1** view, add a tube mesh (32 vertices) and scale it up a bit to make it easier to work with. Again, if you are using Blender 2.43[edit:2.44], add a cylinder and make sure the "Cap Ends" button is turned off. (noob note: Scale it to about 1.7,I think) (noob note: Scale is closer to 1.9) (Note: If you are using Blender version 2.43 it IS NOT necessary to add a cylinder.) (noob note: Change back to vertices select mode **Ctrl+Tab>vertices**) (noob: How do you add a tube?)



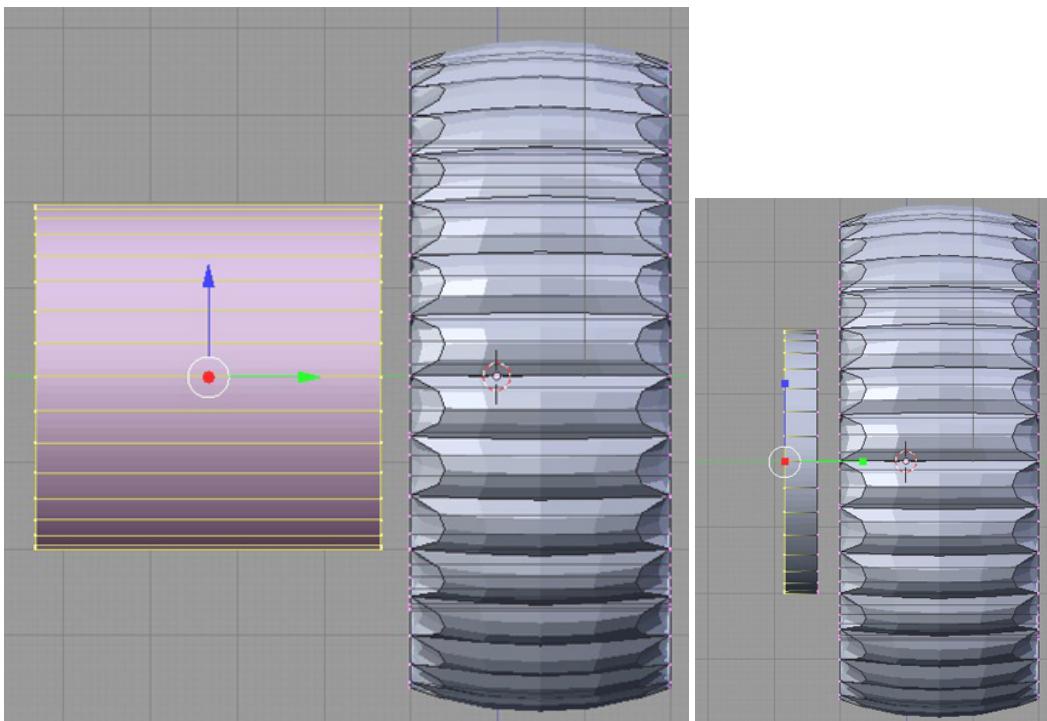
Set the view to **NUM3** and pull the tube out of the tire along the Y-axis. Reduce the length of the hubcap to about 0.5 unit. Select the left half of the hubcap vertices and return to the **NUM1** view.

Noob: How do I do this?

Another Noob: That depends. Are you talking about how to select the left half of the vertices, or how to reduce the length of the cylinder, or is it the whole thing?

Noob: If you select the whole hubcap-to-be and scale it along the axis you want to shorten it by, it will work. I scaled it to about .2. If you have lines where it used to be, select the edges you want to get rid of (circle select [B,B]), and

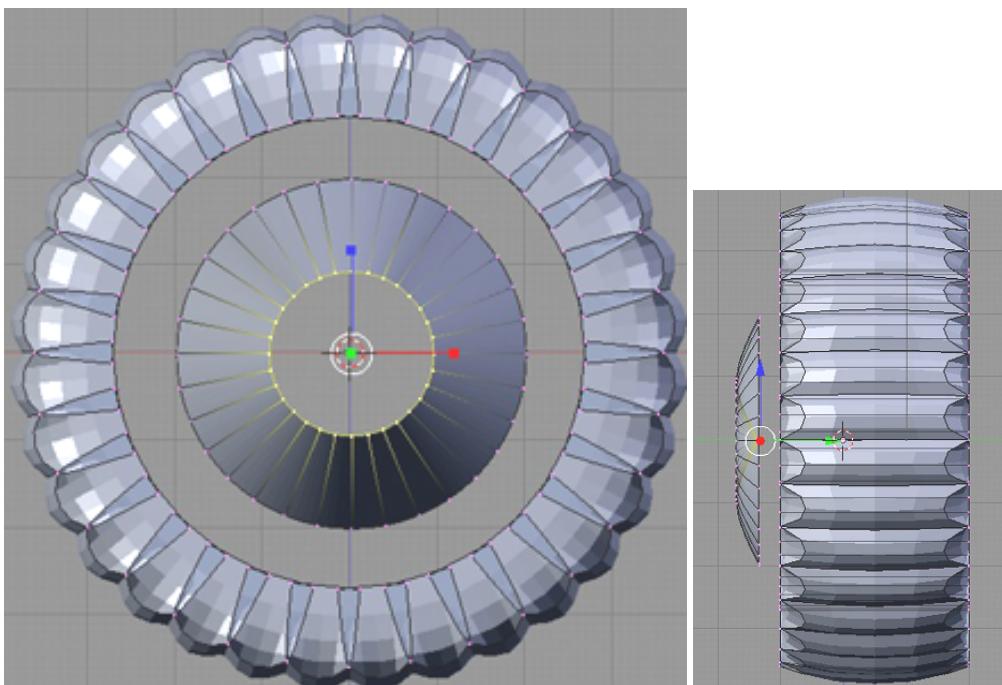
erase them (X, Vertices). That should work. If it doesn't, then it is my error.



Scale these vertices and pull them inward. Keep the vertices selected.

Use extrusion (Only Edges) and then press **ESC**. This will duplicate the vertices and place them at zero, which in this case happen not to be on top of each other.

We'll merge these vertices together to create a flat surface. **ALT MKEY** will bring up the merge window. Merge the vertices at center. Blender will reduce the 32 vertices to 1. Keep the 1 remaining vertex selected, enter **NUM3** view, and pull it along Y into the hubcap.



The final mesh editing is to select all of the hubcap vertices (easiest in the side view), scale up to slightly larger than the hole of the tire, and move back into the tire along Y. (Noob Note: Hold the cursor over the hubcap and press **LKEY** to select all vertices easily)

The last thing to do is to rename the wheel so we can find it easier later. Enter Object mode and select the wheel only. In the Link and Materials subwindow in Editing (**F9**) you'll see the Active Object name box. It should read OB:Tube in the grey box. This name was created because we started with a tube mesh. Click on the button and rename the object to something like 'wheel'. Save your file where you'll find it later and continue to the next step.

Noob: I cant seem to select just the wheel in object mode. Once I've moved the hubcab into the wheel they are both selected as one...

Noob Answer: Split objects in edit mode with **P** (seParate) or join objects with **CTRL+JKEY** in object mode (can cause side effects). Don't forget to apply the subsurf modifier before joining the hubcab and the wheel.

Resulting jeep tire:



## Optional

Change the materials to make it look like a tire. One object can have multiple colors/textures. Refer to the materials section for an explanation on how to do that, or to the Blender manual: Multiple Materials<sup>[1]</sup>

## References

[1] [http://wiki.blender.org/index.php/Doc:Manual/Materials/Multiple\\_Materials](http://wiki.blender.org/index.php/Doc:Manual/Materials/Multiple_Materials)

# Blender 3D: Noob to Pro/Simple Vehicle: Seat

---

## Techniques

You should already know how to:

- Make a mesh
- Navigate the viewport
- Extrusion
- Subsurf
- Crease edges

This section will recap and introduce:

- Loop subdivide
- Small, consistant vertex movement

## Basic Structure

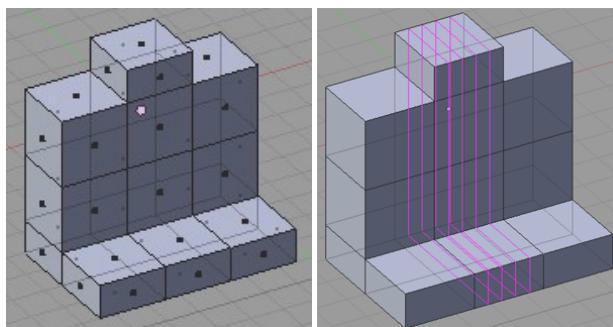
The design will be a allterrain bucket-type seat. Start in **NUM 1** view of the default cube and rename it. Extrude the cube multiple times to make your basic shape. In this example, a 3x3x1 block composes the body with one cube coming out the top for the headrest. The seat is the bottom cube's front faces extruded out.

To add a little texture to the mesh, we'll add some cushion seams. Use Loop Subdivide (**CTRL R** or **K** > Loop Cut) and you'll see the pink selection loop. You can use the **NUM+** key to increase the number of loops made at the same time. Use mouse wheel or press **NUM+ 3** times to form 4 loops and **LMB** the center column of blocks. You may also find it easier to add them one at a time in the correct place, than inserting them and then moving them.

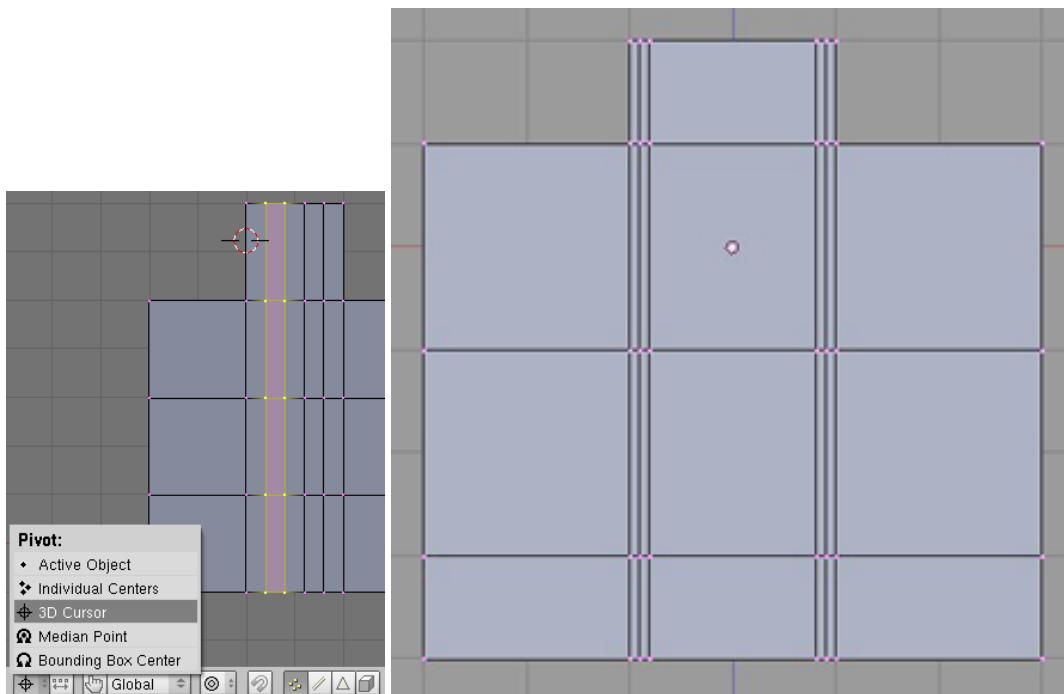
(To get multiple Loops instead of pressing + just press the number of loops you want in this case "4", this is a fast easy way to achieve this.) You can use a mouse-wheel as well.

Noob note: You should remove doubles at this point (**TAB** to edit mode > **WKEY** > Remove Doubles). When I removed doubles before adding the loop cuts, I found that they did not loop around the bottom of the seat correctly, and when I did not remove the doubles at all, I had trouble forming the seams in the next few steps. Noob note: I had to remove all faces inside my seat before loop cut works (and in object mode I join all faces **Ctrl+J**) Noob note: Also you could try deleting faces and edges which are unused inside the object (seat) this way there are no edges to cut up the loop into lots of different sections, so now you will have one clean loop :)

- Noob note: If the loop comes up with green lines rather than purple you have gone one step too far, just press **esc** (escape) and try again. When you see the purple lines use your **Mouse Wheel** or the **NUM+** button.

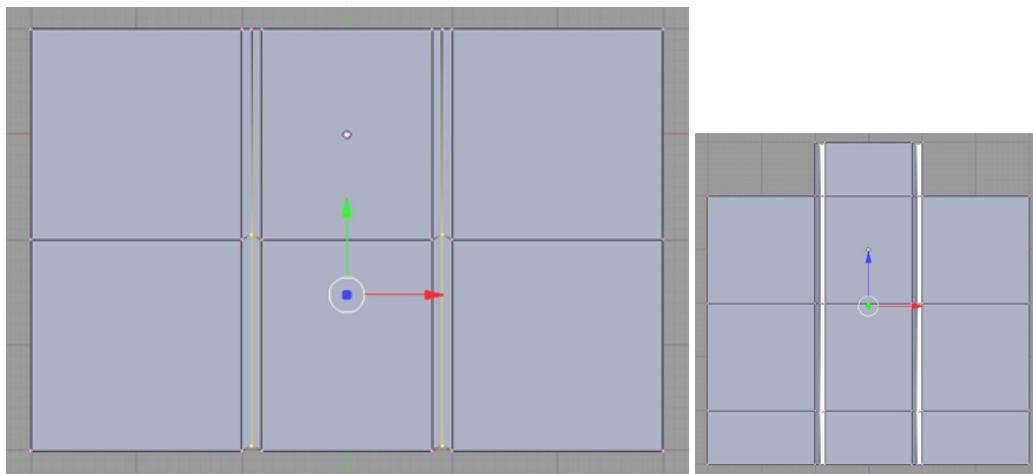


Go into **ZX** view (**Num 1**) and make sure the view is orthographic (**Num 5** to activate/deactivate orthographic view). Place the 3D cursor on one of the sides of the head rest (**SHIFT+S** to move cursor). Set the pivot to 3D cursor and select the two closest loop cuts. Scale it down on the X axis to 0.3. The goal will be to have the loop closer to the cursor to go into the cushion to become a seam. Now, do the same thing for the other side of the head rest.



In overhead view (**NUM 7**), select the vertices in vertical parts of the two seams, grab them (**GKEY**), move only along the Y axis (**YKEY**), and type in a small number using the **NUMPAD**. For example, type in **.05 ENTER**. This will truly create the seams embedded into the back.

Switch to **NUM 3** view and move the vertices in the horizontal parts of the same two seams, grab them and move them down by moving them -0.05 along the Z-axis.



Add a subsurf modifier to the object. Select the edges between the back and seat and crease them. Crease any edges you feel like to create your perfect jeep seat.

Noob note: a subsurf level 2 looks best

- Noob note: If your seat is noticeably misshapen after adding the subsurf modifier, you may just have to delete internal faces in your model. **TAB** into edit mode, and hit **ZKEY** to get into wireframe mode. Click the face select button and look for faces that are totally inside the model. There will probably be a couple vertical faces (in YZ-plane) under the seams in the seat. I found a few elsewhere, also. Deleting all these cleared everything up.

--MSK61 (talk) 16:20, 12 May 2008 (UTC): In blender 2.45, I tried getting into the wireframe mode, but didn't find any internal faces. Maybe they don't exist in all cases.

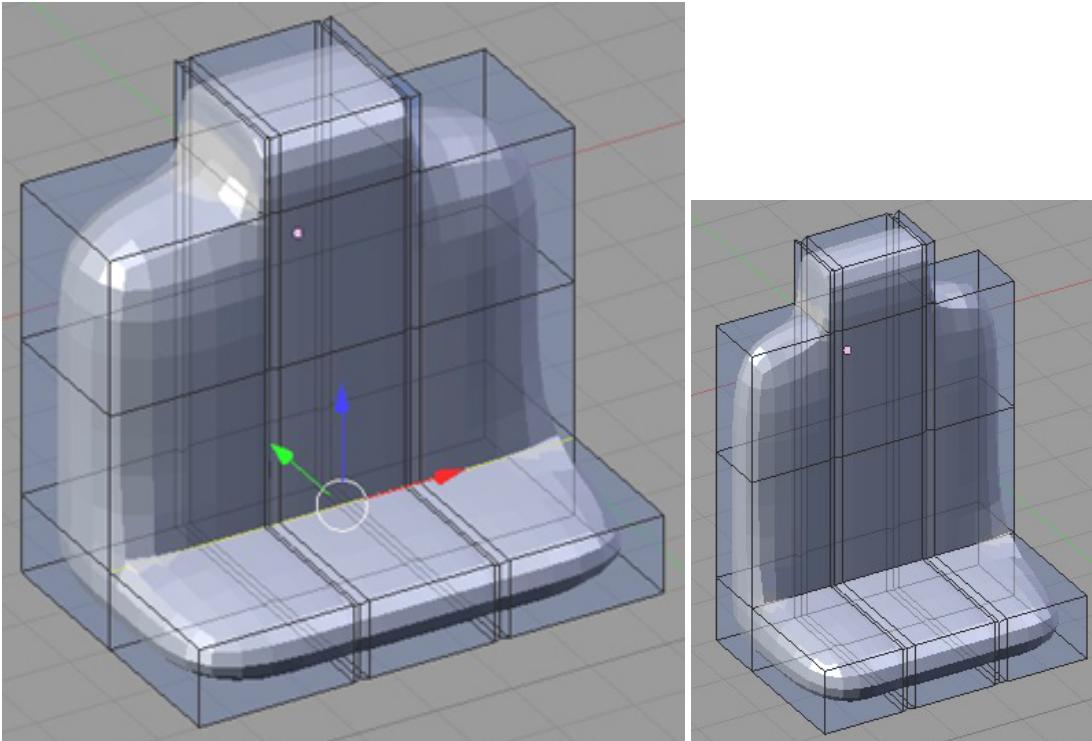
- I also use Blender 2.45 but I managed to find some internal faces (in the loops) and deleting these really did clear everything up. I was already wondering why my model looked so wrong.

- Viewer: Selecting all(AKEY)Pressing WKEY and selecting "Set Smooth" helped me a lot in the looks.

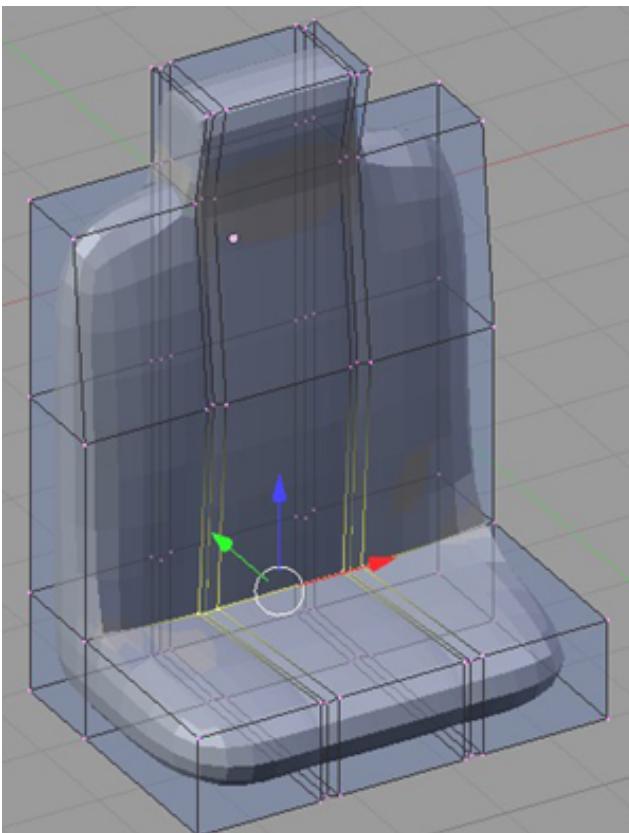
#### Yoshi's Addition-

After subsurfing to give the seams a leathery cord look. Hit A to select all then W and Subdivide Fractal and just keep the defaults and the seams will look like a bunch of vines until you render it and they look like leather seams .

Next, resize the seat height and widths.



As a final touch, the seat can be made slightly more concave to look like it would hold a person better.



This final seat renders to:



## Blender 3D: Noob to Pro/Simple Vehicle: Rocket Launcher

---

### Techniques

You should already know how to do:

- Previous Simple Vehicle techniques

This section will recap and introduce:

- UVspheres
- Changing object's center

### Overview

Two assumptions are going to be made here. One is that the rocket will not be launched in the future (use separate objects if you want to do that). The other is this is going to be a simple, simple design.

If you want to add options to your gun (think sight, trigger), go for it!

### One Tube, a Rocket & a Launcher

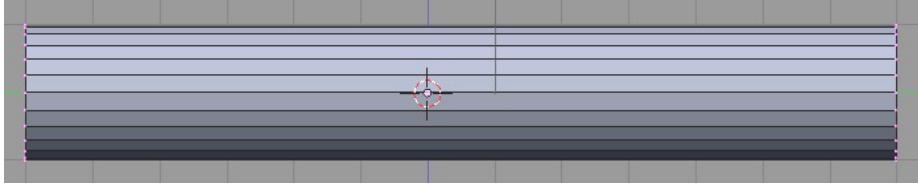
Noob note: I see no reason why this can't be done with cubes, subsurf, and creases. Wouldn't that make for a simpler model? Noob answer: It could certainly be done, but it would be much more difficult. This method is the most direct, simple way.

Start a new file and delete the default cube. In **NUM 1** view, add a cylinder mesh with 24 vertices with EndCap option on. We'll use 24 because the default 36 is overkill and will only increase rendering time. Rename and elongate

the cylinder along the Y axis. This will be the length of the launcher (minus the rocket).

Noob note/question: For some reason, even when I add the cylinder in front(NUM1) view, it is always placed so that the open ends are on the z axis. It is easily fixed, just rotate it 90 degrees, I know, but is there a way to make it appear the right way in the first place?

Another noob: I have the same problem. Maybe it's just one of those things you can't fix. Go to User Preferences, in the Edit Methods tab, and click the button that says Aligned to View. This will make things spawn facing you.

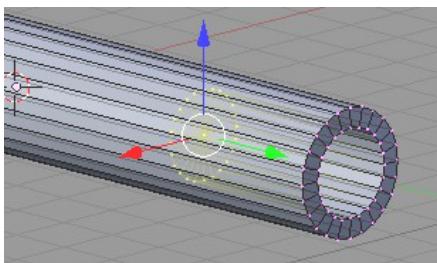


Select the vertices at the right end. Extrude them as region and press Escape to create a copy of the vertices.

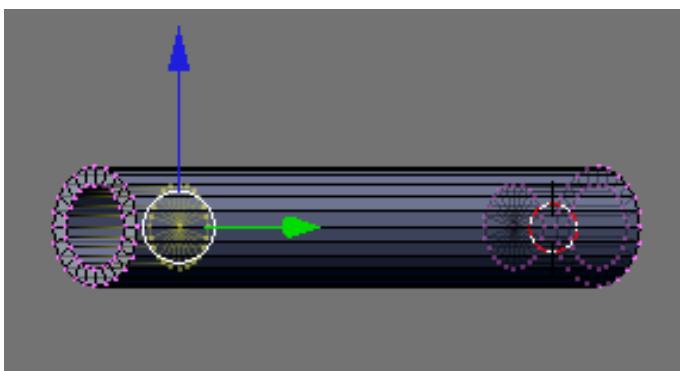
Noob note: I didn't have the option of extruding as a region, but I did it using edges and it worked out fine. I have version 2.46. Answer: The same problem exists in 2.45. Follow the example as shown, extruding edges instead of region. Then follow the instructions provided afterward. Simple explanation: You both had a cylinders with the End cap option turned off when creating it. You had no region to extrude. Which means there'd be no end cap at the end of the barrel (which you'd never notice anyway).

Then scale them by 0.7. Then extrude them as region toward the left. This will make this end of the rocket launcher look like a tube.

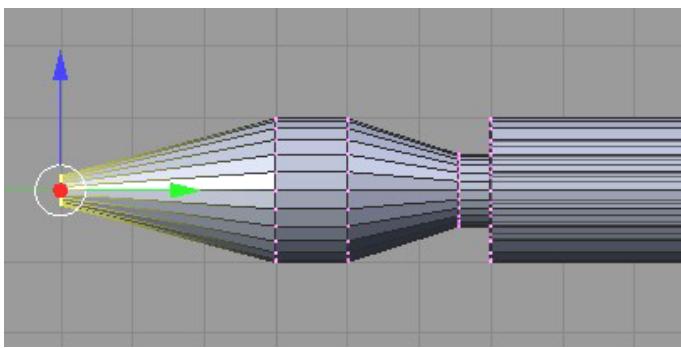
For newer versions, keep the previously extruded edges still highlighted. Then press extrude them, and then escape (**EKEY + ESC**). Then, merge the vertices at the center (**ALT + MKEY**). If you did this correctly, 23 vertices should be removed, and all come together to form a group of triangles inside of the tube.



Now work on the left end side. Make it look like a tube as before:



Extrude the face twice and scale the two new faces by 0.8. Extrude the face twice again and scale the two new faces by 1.5. Extrude it one more time and scale the new face by 0.2. Move the vertices along Y to create the shape of rocket you desire.

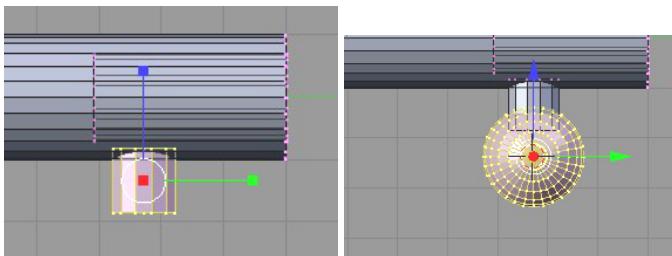


## The Mount

Having a launcher is nice, but we'll need to affix it to the jeep somehow. Let's add a mount to the tube. Think easy. Add a cylinder, scale, and move it to the bottom of the tube.

The easiest way to have a wide range of motion for the launcher is to use a ball joint. We can simulate one by just adding a UVsphere. The default 32 segments and 12 rings will be fine. This creates a smooth sphere. You can think of the number of segments as being the wedges visible when the sphere is viewed as you added it. The number of rings then could be described as the depth of the sphere from that same view.

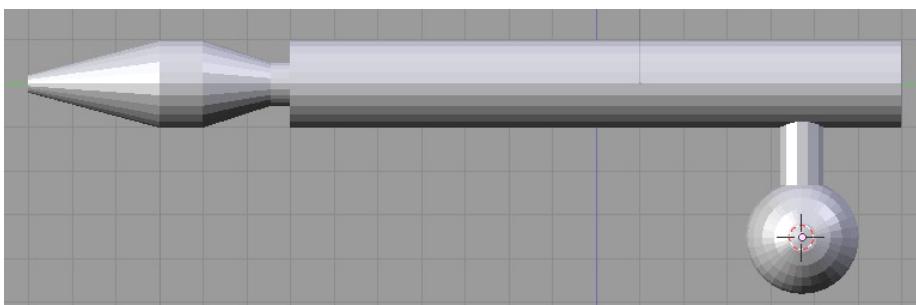
Resize and place the sphere at the new cylinder arm.



This next step will be important for continuing the tutorial. Get the 3D cursor to as close to the center of the sphere as possible. While the sphere is still selected after creation, you can press **SHIFT + SKEY** and snap cursor to selection, putting it in the exact center of the UV Sphere.

*Tip:* If you have already deselected the sphere, press **LKEY** while hovering your cursor over the sphere. Make sure not to include any vertices from the attached cylinder).

Switch to Object Mode. Under either the Object menu or **SPACE** go to Transform > Center Cursor. This should move the large pink dot where the cursor is located. This will give it a new center of gravity around the ball joint, making it easy to manipulate later.



Apply materials or additional items to the object and save for later use.

Do not forget to name the rocket for later use.

Noob note: This would be a perfect place to explain how to make that cylinder and that rocket smooth, but still have crisp edges on the ends of the pipe and on the creases on the rocket. A subsurf with creases just doesn't entirely do it, unless you use a large number of levels, and that slows everything down. What I did was to create a subsurf, set the object to smooth, and then I selected the edges I wanted straight, and used **YKEY** to split them. I did this on the ends

of the tubes and on the rocket. However, somehow I have the feeling that there might be a better way. I am sure a future tutorial will explain this, but I think this is really the place where it should be explained, or a reference should be put in. Noob note: By selecting the faces on the sides (but not end) of the pipe and pressing the smoothen button in the edit window (just like you would for an entire object, but for your selection) to achieve such an effect on the barrel. This does not work when selecting lines, however, so I don't know how to do the same thing for the rocket itself.

**Yoshi's Note:** The way I did this was to highlight it all then hit set smooth. Then go into Face Select and select the faces edges of the rocket tube and hit solid. And for the rocket leaving it solid and adding a subsurf (level 1) gave it a nice look.

Oh and I found the best way to make the rocket separate was to create a cone, rotate it and then extrude and scale to get the rocket shape.

A noob's way: I made the rocket separate by starting with a scaled-along-Y-axis cylinder, then extruding one side, scaling to about 1.3, then extruding two more faces, scaling the first about 0.3 and merging the vertices of the second at center.

Noob's note: What i did was set everything to smooth, then used different textures to get the launcher to look as crisp as i wanted. I used a brushed titanium look for the launcher (since i couldn't get an iron color to look nice), a green rocket with black rubber looking band around the center, and a white tip. Here is my Example launcher [1]

Noob's note: the easiest way to do this is simply know which edges you want to be crisp, subsurf the entire thing then go back to those edges. Select them then crease them with Shift+EKEY. Type in +1 or move the mouse until the number is positive one. The rocket will have a smooth look and still have sharp edges.

Noob note: to sharpen the end of the launcher after subsurf, double the verticies. Select the end of the launcher > EKEY > Region > EscKEY > AKEY.

## References

[1] [http://www.nsgounds.com/rocket\\_launcher.jpg](http://www.nsgounds.com/rocket_launcher.jpg)

---

# Blender 3D: Noob to Pro/Simple Vehicle: Body

---

## Techniques

You should already know how to:

- Make a mesh
- Navigate the viewport
- Extrusion
- Form faces
- Name objects

This section will recap and introduce:

- Deleting and creating edges
- Subdividing
- Merging vertices
- Loop subdivide
- Adding unconnected vertices in one object

## Planning

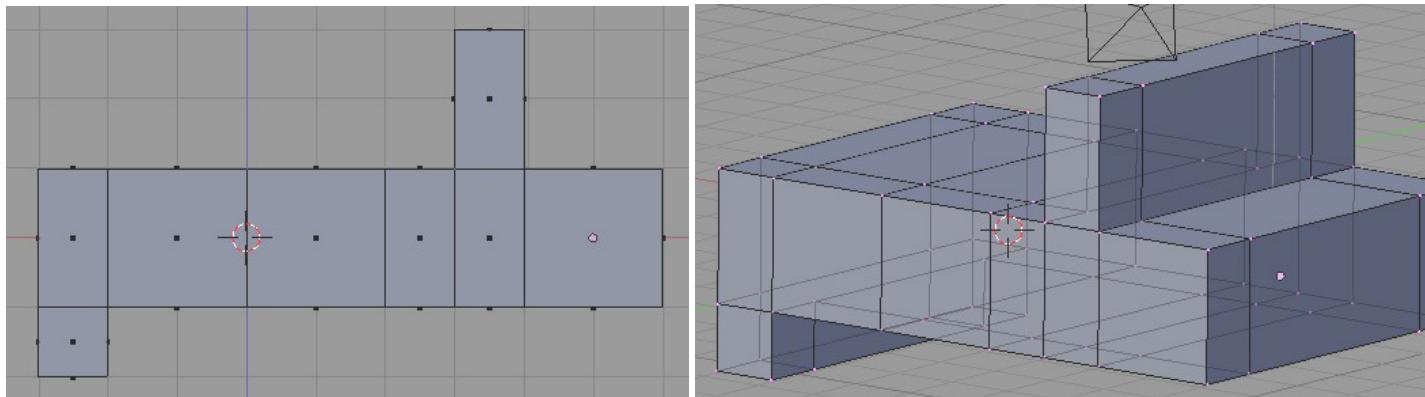
Think about what you want your jeep body to entail. As always, consider where the major sections will need to be made so that you don't have to backtrack later to accommodate your idea. Feel free to make your own body design; here is one I made up.

## The Basics

Start a new file. Take the default block and extrude it into your sections along the X axis (**NUM1** view). In this design, the sections are being designed to include the back, flatbed, doors, dashboard, window, and hood, left to right. The window is extruded straight up (older jeeps' windshields aren't slanted), and I decided to add a lower back to later hold the bumper/lights if you wanted to add them.

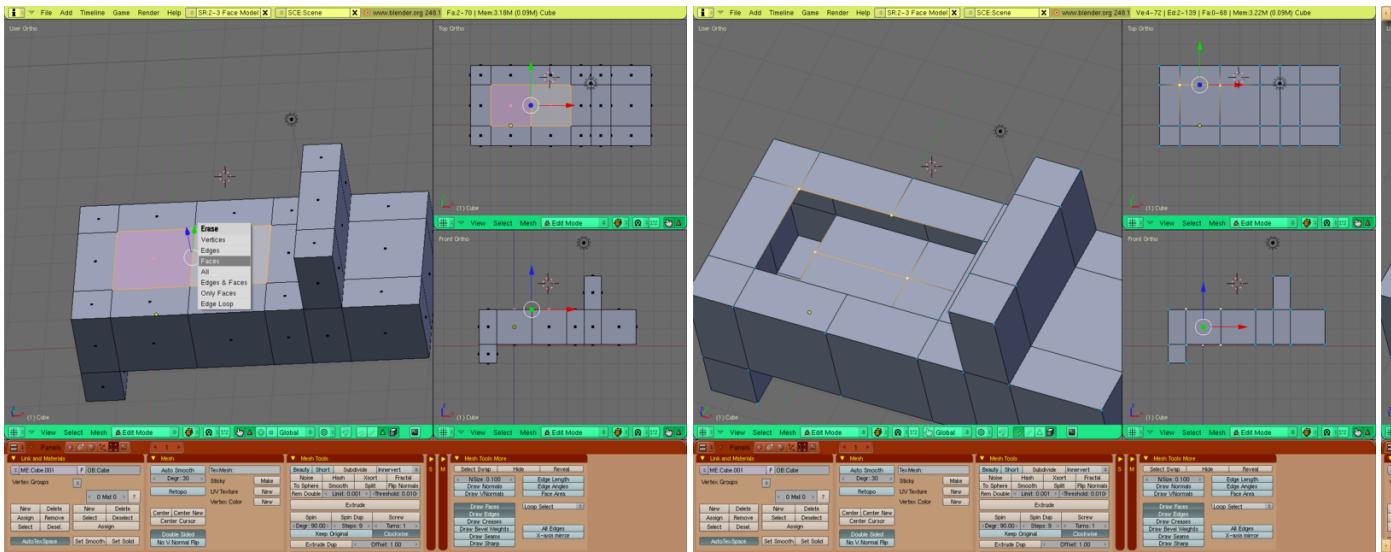
First extrude the right side of the block two units along the X axis. Continue with three more extrusions along the X axis of one unit, one unit, and two units. Now make an extrusion one unit in the negative X direction off of the left side of the default cube. Extrude the bottom of this new block one unit in the negative Z direction. Finally extrude the top of the second block from the right (in **NUM1** view still) up two units in the Z direction. You should now be looking at the side view shown below.

Now to widen the jeep body. Switch to **NUM7** view. Reduce the width of the existing cells to one square. Extrude the top (positive Y) side twice along the Y axis; once for four units, and then extrude again for one unit. This should give you the figure that is shown below.



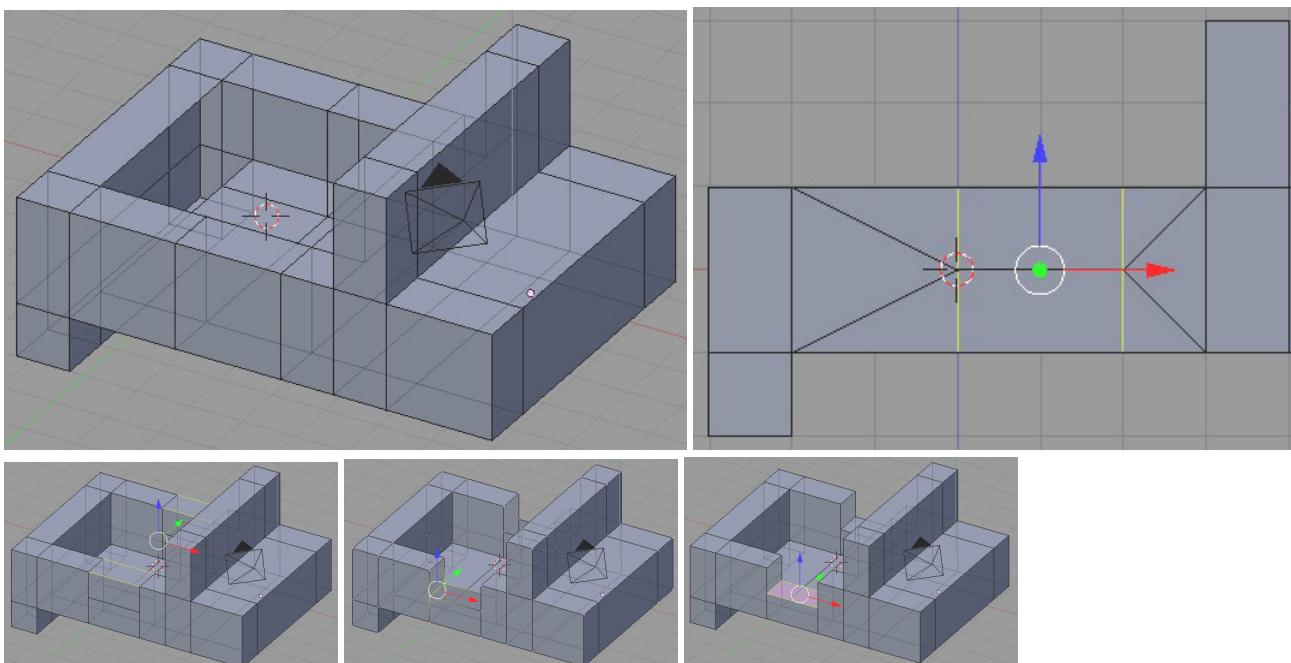
### Flatbed and Doors: Method One (method two is probably preferable, but it has no images)

- Go to **NUM 7** Top view, and enter Faces Mode. Select the two faces in between the bumpers and windshield (**SHIFT RMB**).
- Now delete both faces (**XKEY**).
- Switch to vertex mode, and select the the FOUR vertices that will make up the new faces inside the jeep body. You must select two topside vertices and two bottom vertices that are parallel to the top ones.
- See images of which vertices to select. Then press **FKEY** to create the face. Proceed around the jeep until all new faces are created.

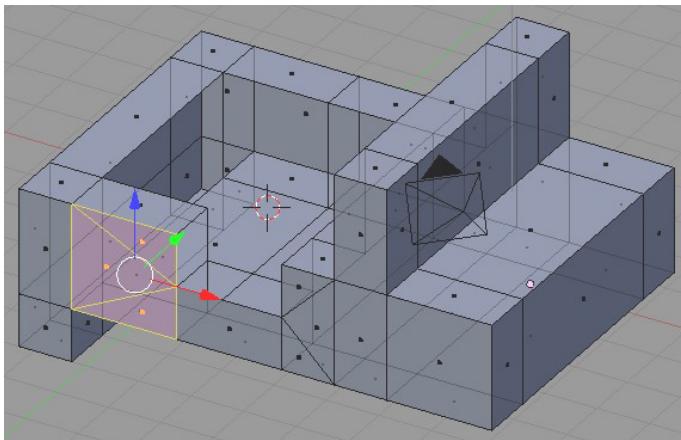


- Here is a video showing the whole face deletion, and creating the inside faces process I use Blender3D for Ubuntu Linux, therefor, during the capturing of this video I had the ComPIZ Desktop Effects Manager active, as such there are occasional "red-outs" (red dots against a black background)."

Adding the door: our jeep design will have somewhat of a cheat - no actual door. Before crying foul, many jeeps didn't have doors to facilitate quick entry and exit. To cut out one door, select the four vertical edges where the door will go. Use the subdivide command (**WKEY**) to cut the edges in half. You'll notice that the subdividing will also affect the adjoining faces. Do the same thing you did when the flatbed was cut out - removing/adding edges and creating new faces.



This is not a necessary step, but if you want to clean up the look of the model, we'll get started. In face select mode, select the faces that are going to be combined. Press **FKEY** and a little window will appear titled 'Make Faces'. Click on 'Make FGon' to merge the faces. As you may have guessed, if an FGon face is created and you want to later undo it, select the 'clear FGon' option in the Make Faces window. (Noob Question: what exactly is an Fgon? It doesn't seem to truly merge the faces, as now in face select mode the 'square' that designates a face is off-center.)



### Flatbed and Doors: Method Two

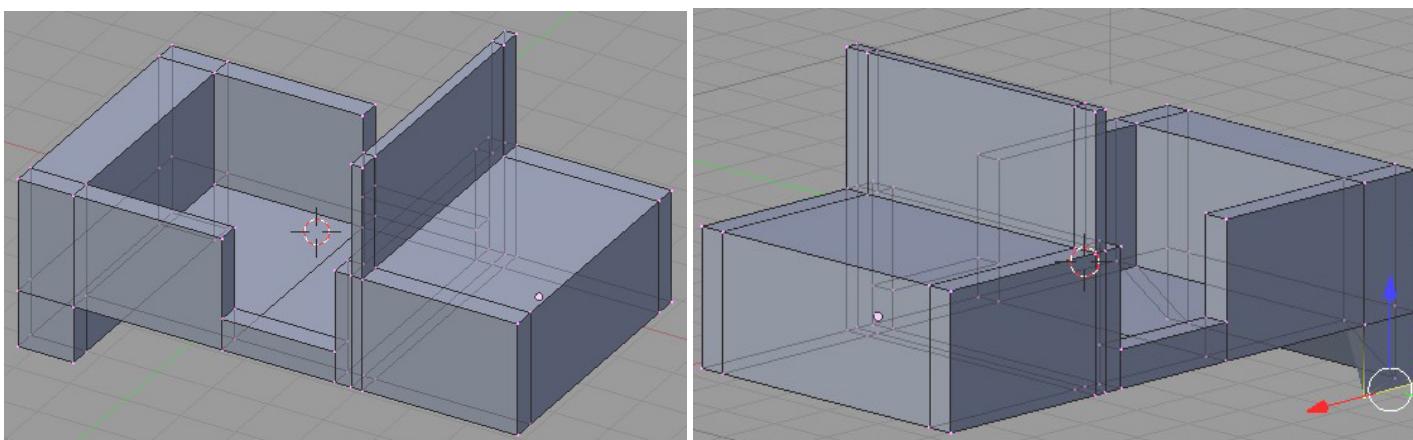
In **NUM7** view select the faces where the flatbed and doors should go. Only the top faces. There should be four of them, two large ones for the bed and two small ones for the doors. Extrude these faces -2 in the Z. Now in **NUM1** view select the side faces where the doors will go and delete them. Now select the faces in the bottom of the door wells and extrude them up 0.5 in the Z. If you want the bed to have some thickness extrude it up some as well (0.1 maybe). Finally, select the whole body and remove doubles (**WKEY**).

### Resizing

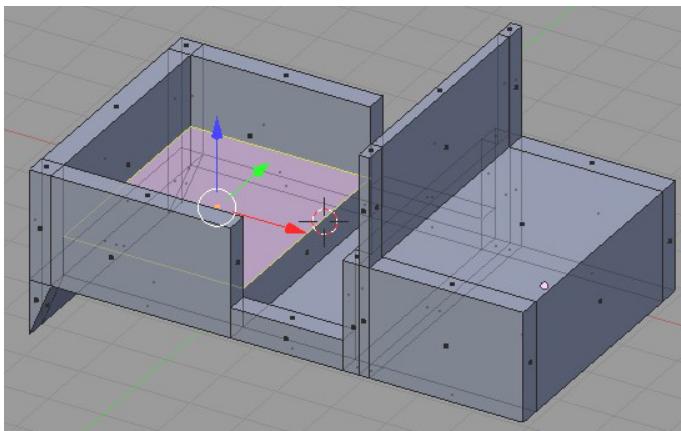
Go ahead and resize the widths of the bed and windshield. Always move as many vertices or edges at the same time to not only work faster but to make sure they are moved equally. The use of **SHIFT** while moving vertices is very helpful in fine movements.

If you want an object to come to a point such as a wedge from a cube, merge vertices. In this example, the lower back area will be modified. Select two vertices to join together and press **ALT MKEY**. Select the option for your merging. 'At First' or 'At Last' will probably be the option that will work here. Play around to see what each merge option does. After the merging, Blender will tell you how many vertices were removed.

**Noob, 20th Oct 2008:** I found *Mesh ==> Vertices ==> Remove doubles* is just as good when you bring one or more vertices into contact with one another, like he's done with the tailgate/wheelarch here.



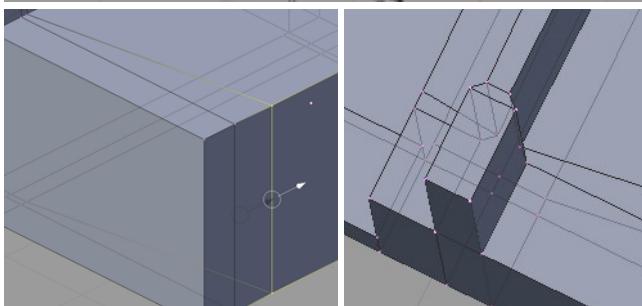
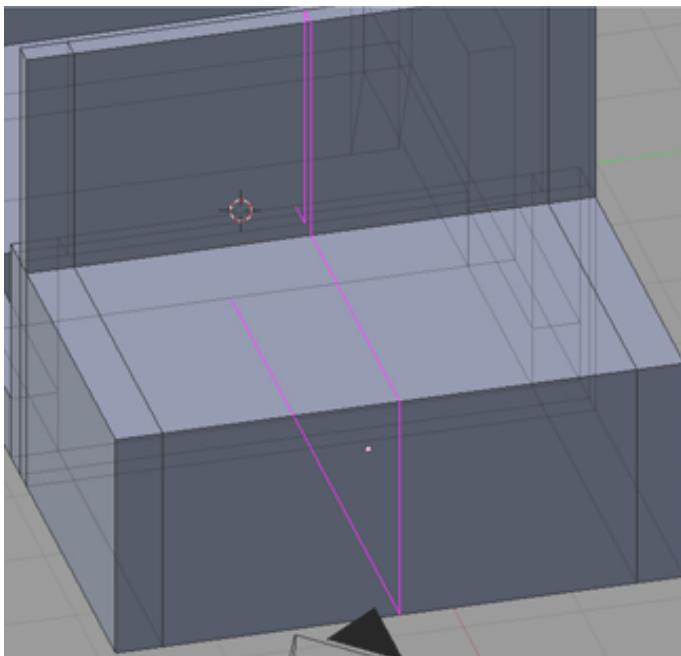
Extrude the bed surface upward. This is only useful in hiding the tires that we'll add later in the tutorial. Alternatively, you could make two boxes to hide them.



## A Touch of Detail

Let's add some detail to the model - how about the hood? First thing to do is add some edges to the front of the jeep. Press **CTRL RKEY** to enter Loop Subdivide. A pink loop will appear around the mesh. Put the cursor over the area to get the example picture to appear. When the loop is in the right place, **LMB** click. The place to put the actual cut can now be selected. Do this twice - once for each side. Move and align the resulting edges to form an angle to the front and bring the window vertices in.

*Newbie: I'm having trouble here. When I try to add a loop it gives me an error that reads "could not order loop" and then subdivides the face I tried to add the loop to.* Answer: I had the same problem. For me it was because I changed the jeep design too much so I had deleted and made several faces. Therefore the faces weren't attached to each other so a loop couldn't be made. If that happened then just subdivide the face twice to give you the lines you need and delete all the extra ones and remake the faces. It's more work but I don't know another way to do it.



Extrude the hood surface up a small amount. We don't want it too high, just high enough to catch the light.

Zoom in and select the top-front hood edge created from the extrusion. Drag it out along the X axis. Select the now diagonal face of the hood extrusion. Extrude from it. The result will come out of the surface at a diagonal angle. Take the resulting vertices and move them close to the front of the jeep.

**Readers note:** I have no idea what the heck this paragraph is talking about, and the pictures don't help, someone really needs to rewrite this that actually understands what is going on. I advise skipping the hood part because it is hard to understand and doesn't look like it changes much.

**Other Reader:** I hope he talks about shaping the hood with the lines from the Loop Subdivide, and then extruding that profile.

**User Note:** If you don't know, press e to extrude, then type the letter of the axis.

**Noob Note:** You're basically "drawing" a trapezoid on the hood and extruding it up. Loop divide works parallel to the axes so you'll have a square on the hood after the two loop-divides. Pulling the two vertices that are on the front edge of the hood inwards (along the edge) will create the desired trapezoid. Now just select the face (Ctrl+Tab Faces) and extrude it (EKEY).

### An Alternate method of adding the Jeep hood

Here is a technique that I used to raise the Jeep hood. It doesn't require loop cuts, so is a bit simpler. FYI, this is my first attempt at a wiki page edit, hopefully it will be clear. No photos yet, can't upload images until four days.

Step 1. Make Inner Square.

Switch to Top-view NUM 7, or slightly rotated off for easier viewing. Face-select the top square of the hood.

Do E-xtrude, then hit <esc>. NOTE: this WILL make a new surface, hitting <esc> doesn't cancel the extrude, just makes its location to be exactly on the old surface.

Do S-scale and type 0.9 and hit <return>. Now you will see the new surface as a smaller square (or really rectangle) on top of the jeep hood square.

Noob Note: What I had to do is S-scale on Z and type 0.9, otherwise the New Surface For the hood would end up shooting through the Windshield.

Step 2. Shape Hood.

Still in Top-view NUM 7, deselect all.

Box-select the right two vertices of the new square, towards the front of the jeep.

Do S-scale on Y, and type 0.8 and hit <return>. Leave the two vertices still selected.

Do G-rab on X, and type 0.2 and hit <return>.

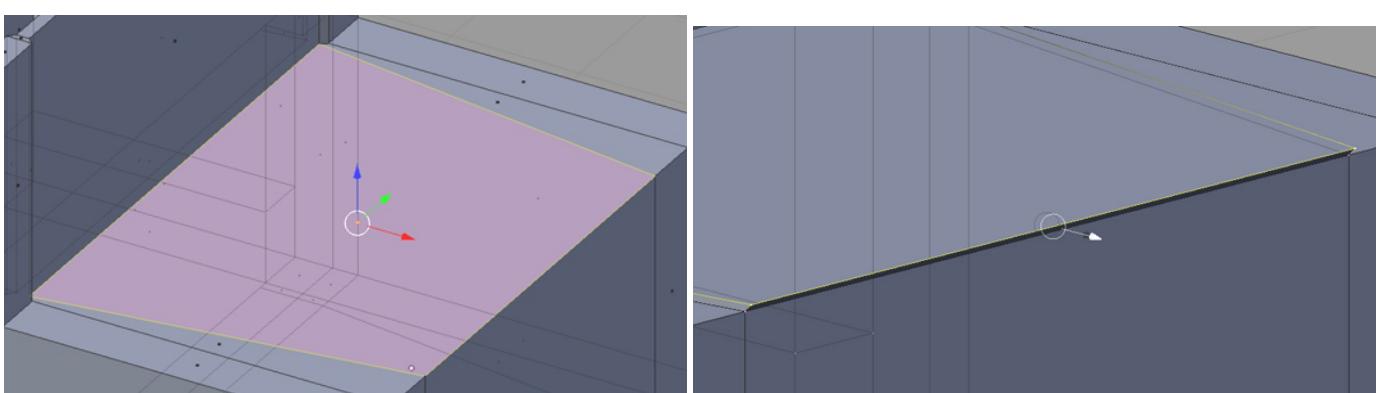
Face-select the resulting quadrilateral.

Switch to Side-view NUM 3.

Do E-xtrude on Z and type 0.2 and hit <return>. I used 0.2 to exaggerate the screenshot a bit, you probably want 0.1 instead.

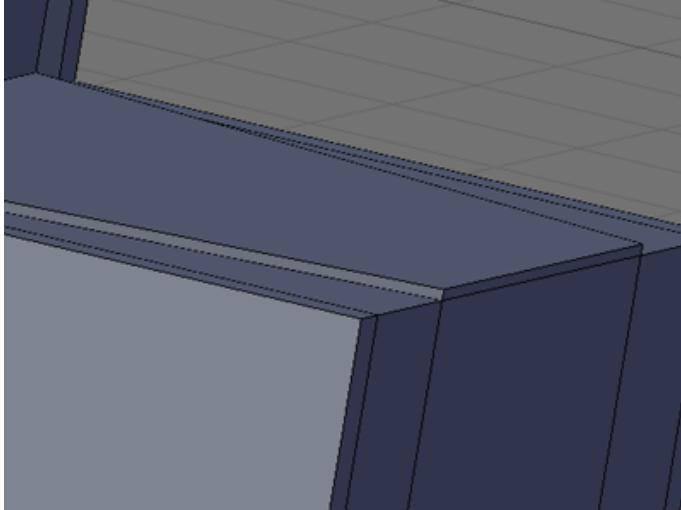
Now you should have a raised hood on the front of the jeep.

**End Alternate Hood method, we now return you to the original tutorial**

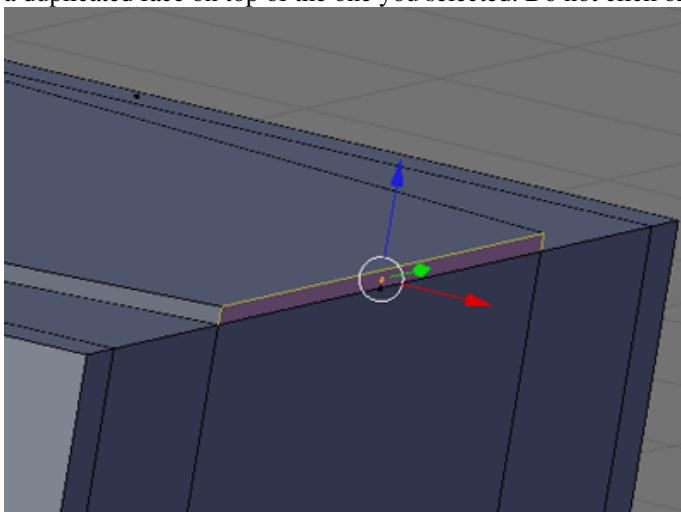


The above paragraph briefly explains extruding the Hood of the jeep including a lip that comes over the front. Below are step by step pictures on how I accomplished this;

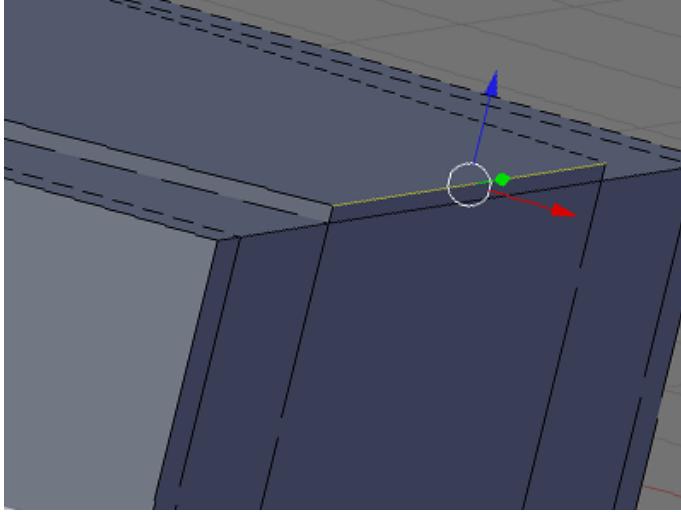
Step 1: In face select mode (**CTRL+TAB-faces**), select the top of the hood and press **EKEY** to extrude it. Only extrude it a small amount.



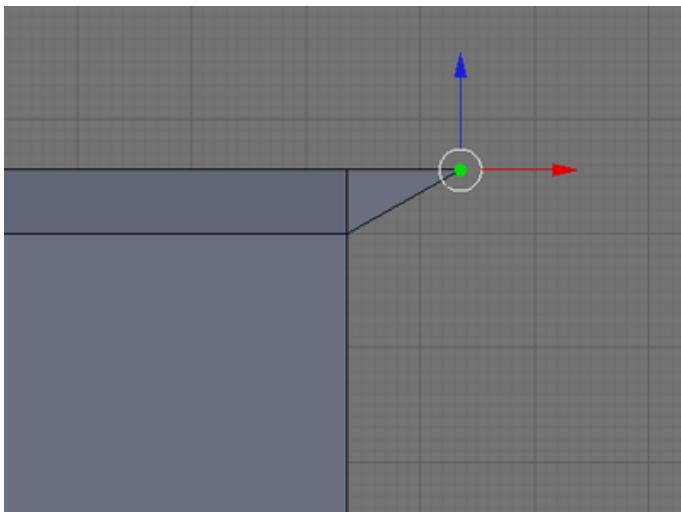
Step 2: Staying in Face select mode, here is the tricky part. Select the small face on the front of the hood you just extruded. Then, hit **EKEY** then **ESC**. This creates a duplicated face on top of the one you selected. Do not click or move the mouse between these two keystrokes.



Step 3: Now deselect the selected face by hitting **AKEY**. Then enter Edge select mode (**CTRL+TAB-edges**) and select the top edge of the face you just deselected.

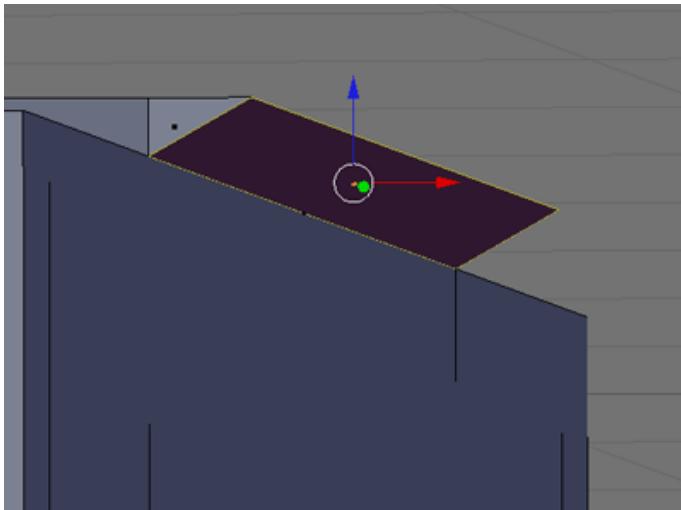


Step 4: Hit **NUM 1** to go to the side view. Now using the red X arrow pull the edge out a little further than you pulled up the hood itself then hit **LMB**.



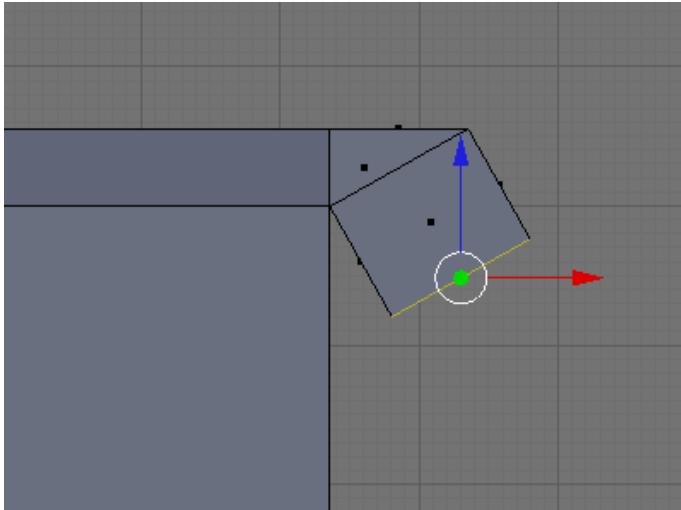
Step 5: Now rotate just enough so you can see under

the wedge you just made. Go back into Face select mode (**CTRL+TAB-faces**) and select the face on the underside of it.



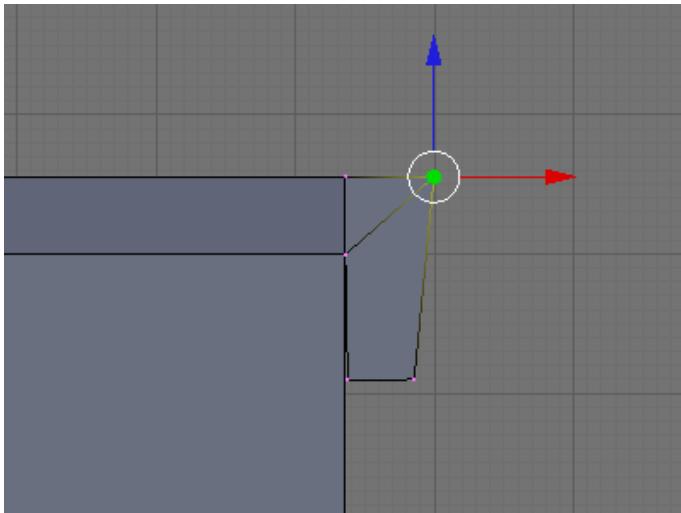
Step 6: Go back into side view with **NUM 1**. Now

hit **EKEY** to extrude the face a little with the mouse. When it's a good size hit **LMB**.



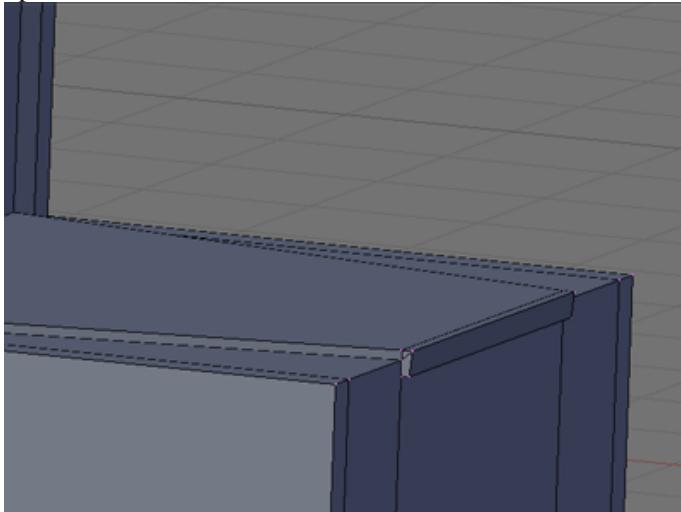
Step 7: Now make sure the "Select only visible"

button is turned off and go into Vertex select mode (**CTRL+TAB-vertices**). Play around with the vertices pulling them a little closer to the front of the jeep. It's best to select using the box select (**BKEY**) or the Lasso (**CTRL+LMB**).



The end result should be an extruded hood with a

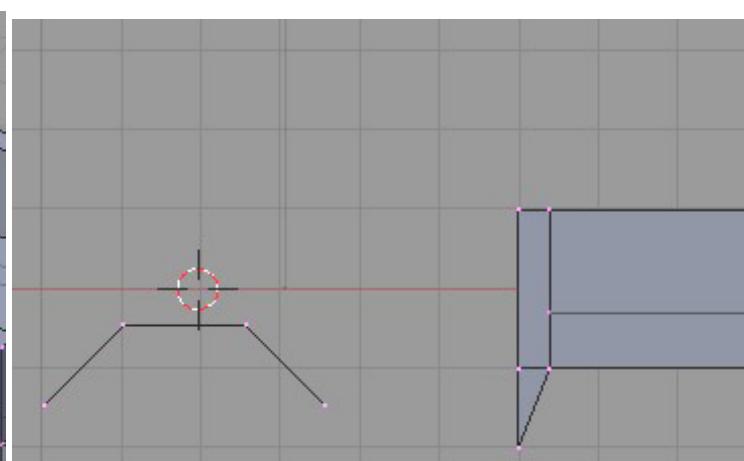
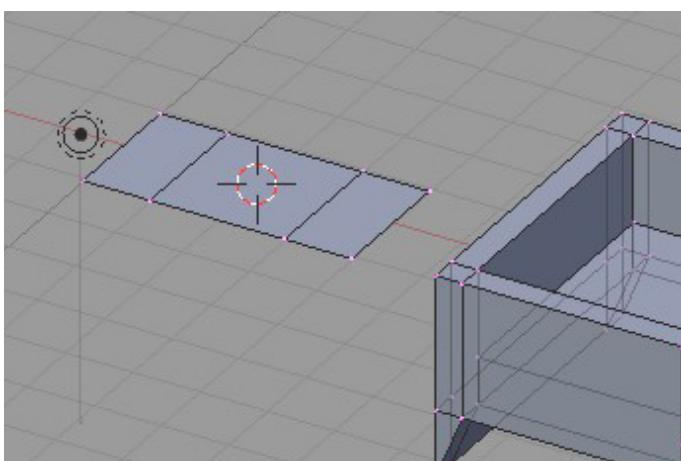
lip.



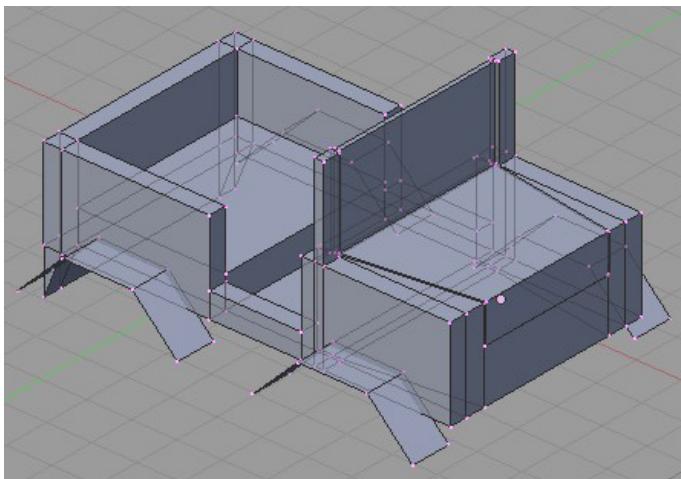
Alexwill84 08:23, 27 April 2007 (UTC)

In the topdown view (**NUM 7**), add a plane. Extrude an edge twice to result in three connected planes.

Pull the sides down to form a trapezoid shape and reduce the width. Once it is in the desired shape, select the three faces and duplicate it. Press **SHIFT DKEY** and all selected vertices, edges, and/or faces will be duplicated. The copy will automatically be grabbed for moving.

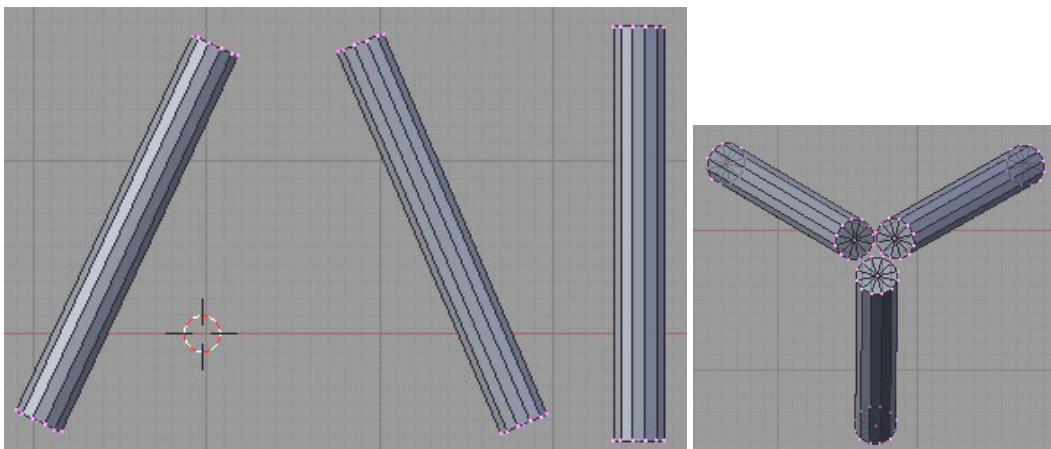


Move the duplicate to the jeep body and repeat the duplication two more times for a total of four fenders.

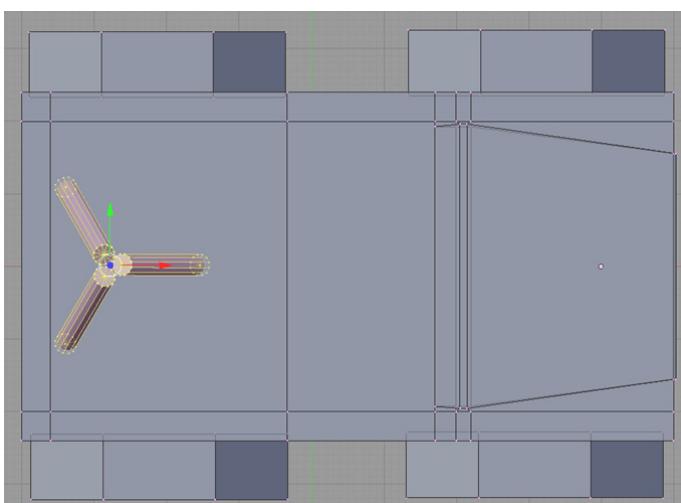


We'll move on to making a tripod support for the rocket launcher. Add a cylinder mesh with 12 vertices then scale and size it so that it looks like a tube. Once you have it to a size you like, duplicate it twice for a total of 3 cylinders. Rotate two of the cylinders in the **NUM 1** view by **LMB** clicking on the white circle that appears when the cylinder is selected in rotate mode. The picture on the left is an example of the end result.

Change to overhead view (**NUM 7**) and put together the three cylinders so the tops come close together. Now all three can be selected and moved or rotated accordingly.



Move the tripod onto the jeep flat bed. The final steps are to select your materials and rename the object (described in the wheel section). This will complete our simple jeep model.



## Optional Activities

Feel free to add anything you see fit such as bumpers, guard rails, doors, steering wheel, lights, etc. You can either have them on the same object or separate objects (useful if you want to move them around). Note: If you subsurf the result from this tutorial, you will get a bad looking result, please try not to subsurf the jeep. *Another Note: Don't subsurf unless you want to edge crease with Shift-EKEY*

# Blender 3D: Noob to Pro/Simple Vehicle: Some Assembly Required

---

## Techniques

You should know how to:

- Do everything discussed in previous tutorials

This section will recap and introduce:

- Append a file
- Duplicate an object

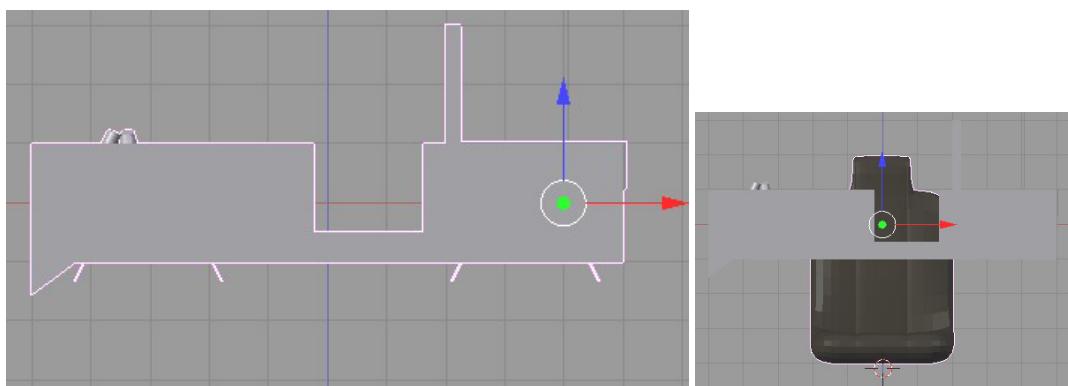
## Overview

The objects for the simple vehicle have been made if you have followed all the previous Simple Vehicle tutorials. Putting it all together will come very easy now.

## Appending the File

If you have the jeep body file open, keep it open. Otherwise, open the file for the jeep body. In Object Mode, go up to **File > Append** (about 3/4 of the way down the menu). The find file window will appear. Go to the location where the jeep seat was saved. When the .blend file is clicked, you'll go into it as if it is a directory.

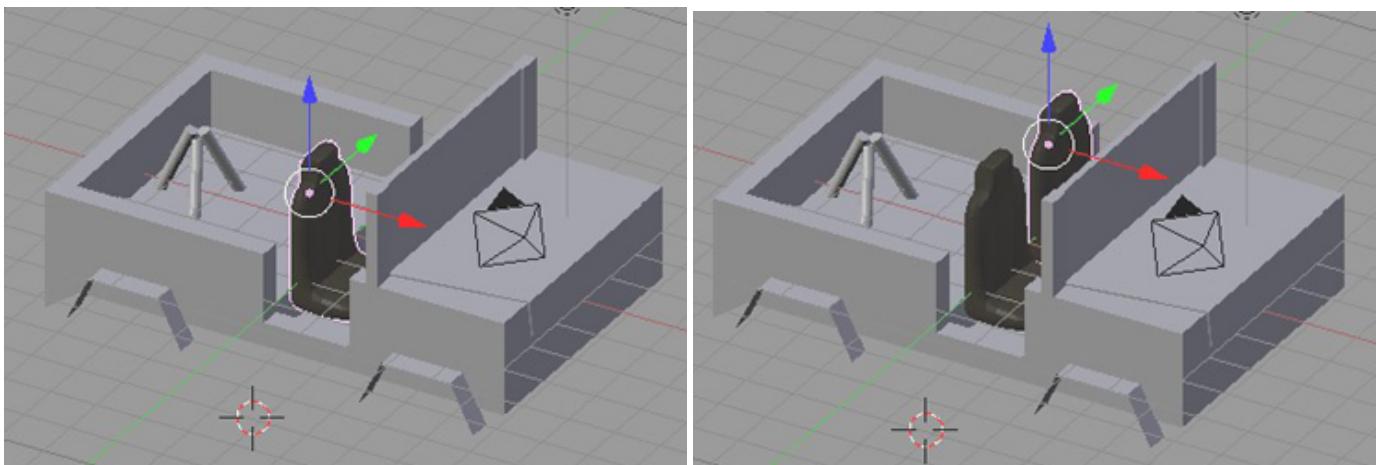
Here we have the categories of Camera, Lamp, Material, Mesh, Object, Scene, Text, and World. We are interested in the seat object, so click on Object. Now there are three items: Camera, Lamp, and Seat. That is, it will say Seat if you named your object Seat. This is why it is useful to rename your objects, materials, etc. If you forgot to rename the object, it will be called Cube (default for our starting mesh).



What should happen after selecting Seat and the button 'Load Library' is the seat will pop into our file where the 3D cursor was. It will definitely need to be scaled, rotated, and/or moved to the right position. In my example, I rotated the seat about the Z axis -90 degrees by pressing **RKEY ZKEY NUM9 NUM0 NUM-**.

(Noob Comment: When I tried this, I was unable to move my seat at all. Said that I could not edit the object.) (This noob either (A) linked the seat instead of appending it. when selecting the file, look at the bottom for the append/link options and make "Append" selected. (B) selected "Append or Link (Image Browser)" instead of the option above it in the File menu. (C) tried to enter edit mode without making a new instance of the mesh. In **F9(Editing Panel) -> Link and Materials** You will see at the top the mesh name "ME: seat" or whatever you called it. To the right there may be a button that makes the mesh data local. Click that if you want to make a copy of the mesh data.)

After placing the seat in the jeep body, let's make another so we have a seat for the driver and passenger. Still in Object Mode with the seat selected, duplicate it (**SHIFT DKEY**) and slide it over. After duplicating it, it will automatically go into grab mode. If you **RMB** or **ESC**, it will still be duplicated - just sitting on top of the original.



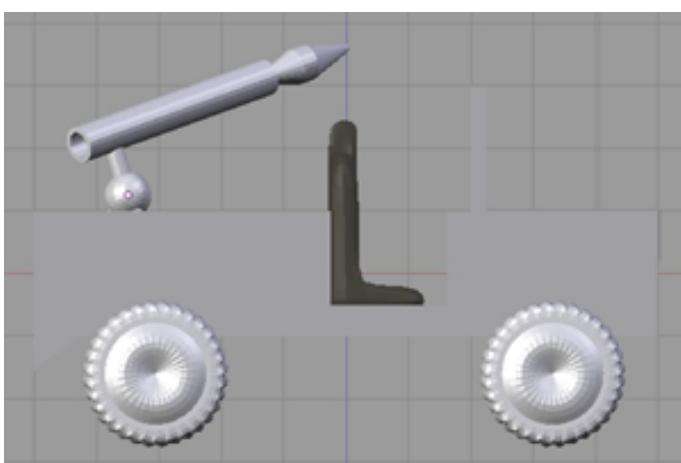
(user comment: When you move the body, the seat will stay in his place. To prevent this you must make the body parent of the other parts. For example the seat: first select the seat, then select the body.(so you have them both selected) Now press (**CTRL PKEY**) and select make parent. If you move the body, the seat will move with it. if you move the seat, the body won't move.)

## Rinse and Repeat

Append the file again to place the wheel object and rocket launcher in the file. Scale, rotate, move, and duplicate each object accordingly. Depending on the position of your camera, you may or may not have to make all four tires. Remember that the only important parts to draw are those that will be seen!

**Noob:**I may have done something wrong in the Rocket Launcher tutorial, but when I link to the .blend file each object (launcher, mount, and ball bearing) is listed separately and have to be reassembled in the new file. Any suggestions??

Set the ball over the tripod. The fun part is rotating the rocket launcher since the center of it has been moved to the ball joint.



The last thing is to apply materials to your objects if you haven't already. I'll point you to the materials section for information on that. I would suggest that the window be assigned a transparent material by making the Render Pipeline wire for example.

**User request:** explain in more depth how to make the windshield transparent

**User reply:** if you quit out of this tutorial while staying in that section of the book, there is a section called "every material known to man" (its near the bottom of the tutorial list) and there are several kinds of glass there.

**Noob:** how do you get just the windshield made out of glass?

**User Reply:** select the faces you want, press f7, (*note: it's f9 for me [2.47]*) then click add under links and materials. Press f5 then add your glass material

**Noob:** I have 2.4.6, and I don't know how to make just the windshield transparent, everytime I try the whole thing is transparent.

**Another Noob:** You have to create a new material and assign it to the appropriate faces. In edit mode, select the faces you want (make sure you select both sides of the windshield including: front face, back face, and top face). On the Editing section (F9 in 2.4.7), under Links and Materials, click the 'New' button under Materials then click the 'Assign' button. This assigns the new material to the selected faces. You can then go into the Shading section (F5 in 2.4.7) and adjust the material to be glass. Make sure you are editing the correct material (if you have just the one texture on the jeep, the default name for the new material should be something like "2 Mat 2").

**Yet Another Noob:** When I append the files, they lose their materials. When I link the files, I can't move or resize them.

**Yet Another Noob (the same one as just above):** I just figured out that my file is glitching. Append should make it possible to move and resize AND it should keep the material.

The final result of a simple jeep:



Another Possible rendering:



And yet another possibility:

Image:Green Jeep With Rocket Launcher.png

(File linked due to size)

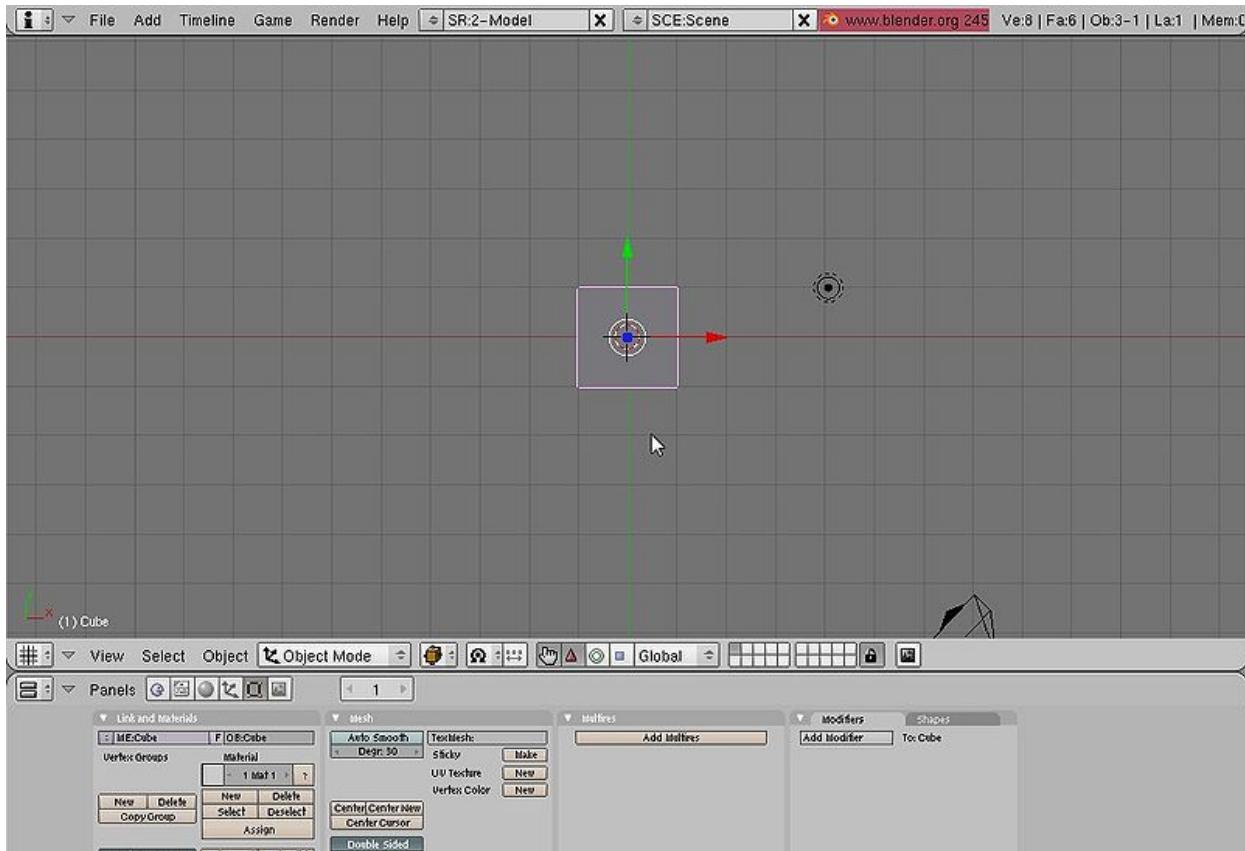
## Additional Tutorials

Append and Link Video Tutorial: <http://www.youtube.com/watch?v=69ZBIDrOIIY>

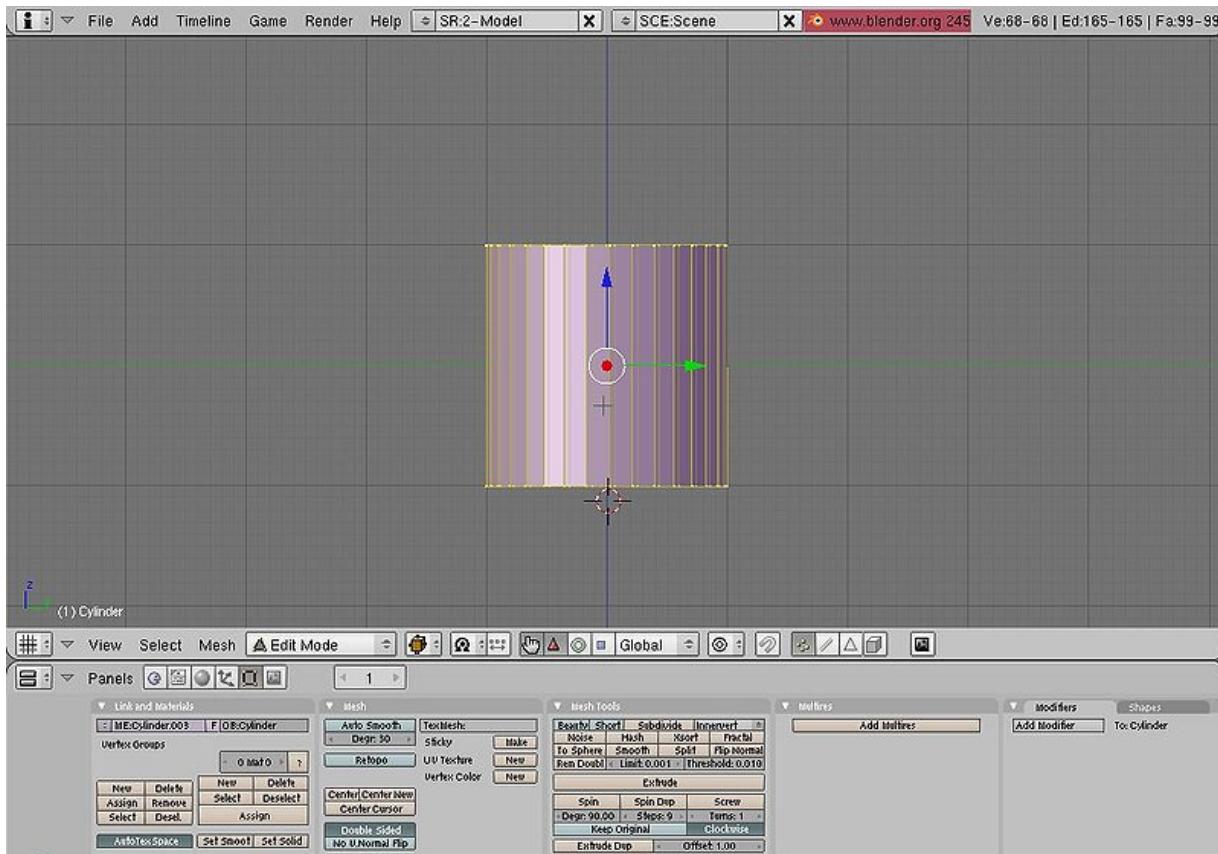
# Blender 3D: Noob to Pro/Modeling a 3D Parachute in Blender

## The Making of a 3D Parachute

When you first open blender you will see the default Blender user interface (UI) layout and the default cube.



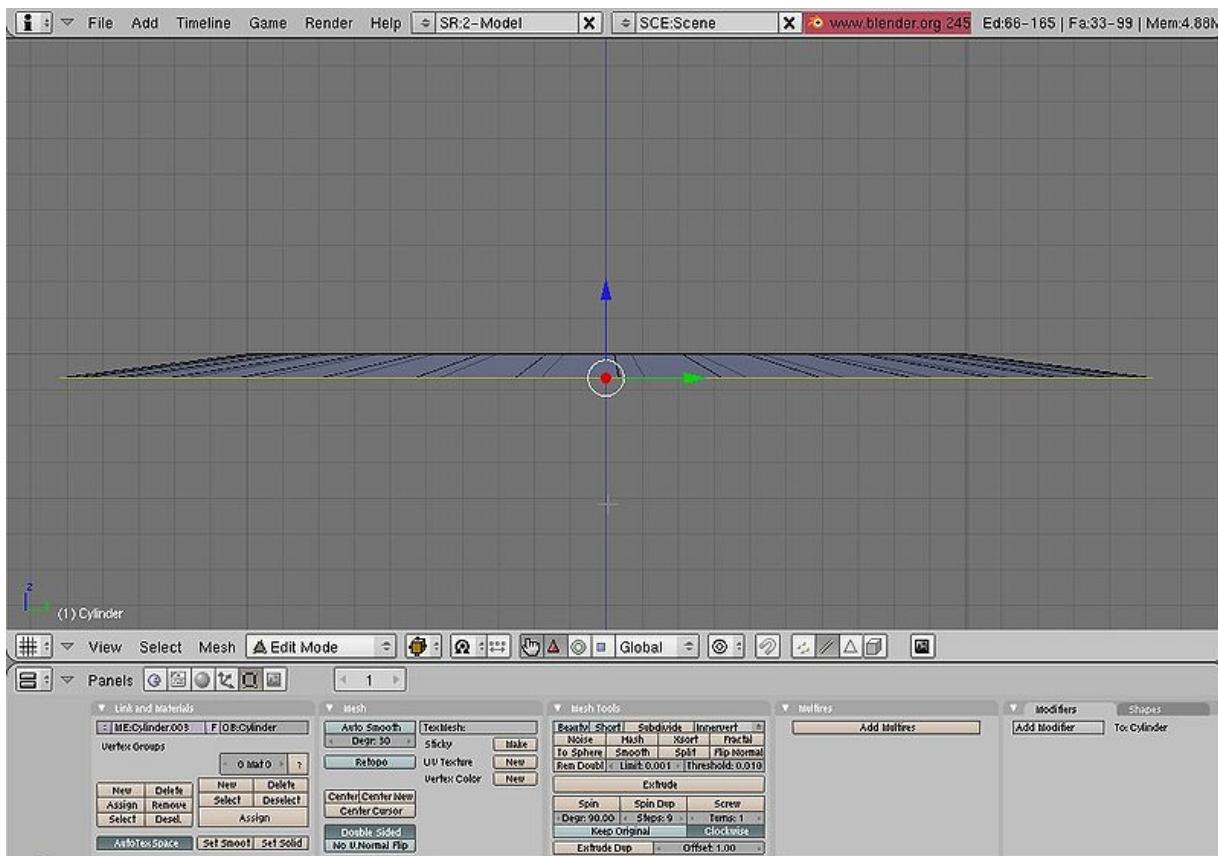
Delete the cube by selecting it with the Right-Mouse button and then clicking the X key. Next, add a Cylinder by clicking Space-bar> Add> Mesh> Cylinder. This will bring up a small window in which you can set several parameters including vertices, radius, Depth, and cap ends. We will leave the settings in their default state. Press OK to add the cylinder to the scene. Here Blender will automatically take you to edit-mode which is what we want. (You can switch between edit-mode and model-mode by pressing the tab key) Next you will want to delete the bottom row of vertices. To accomplish this go into side view by pressing "3" on the num-pad. Ensure that you are in vertex select mode by pressing Ctrl-Tab and selecting "vertices". The screen should look like the following:



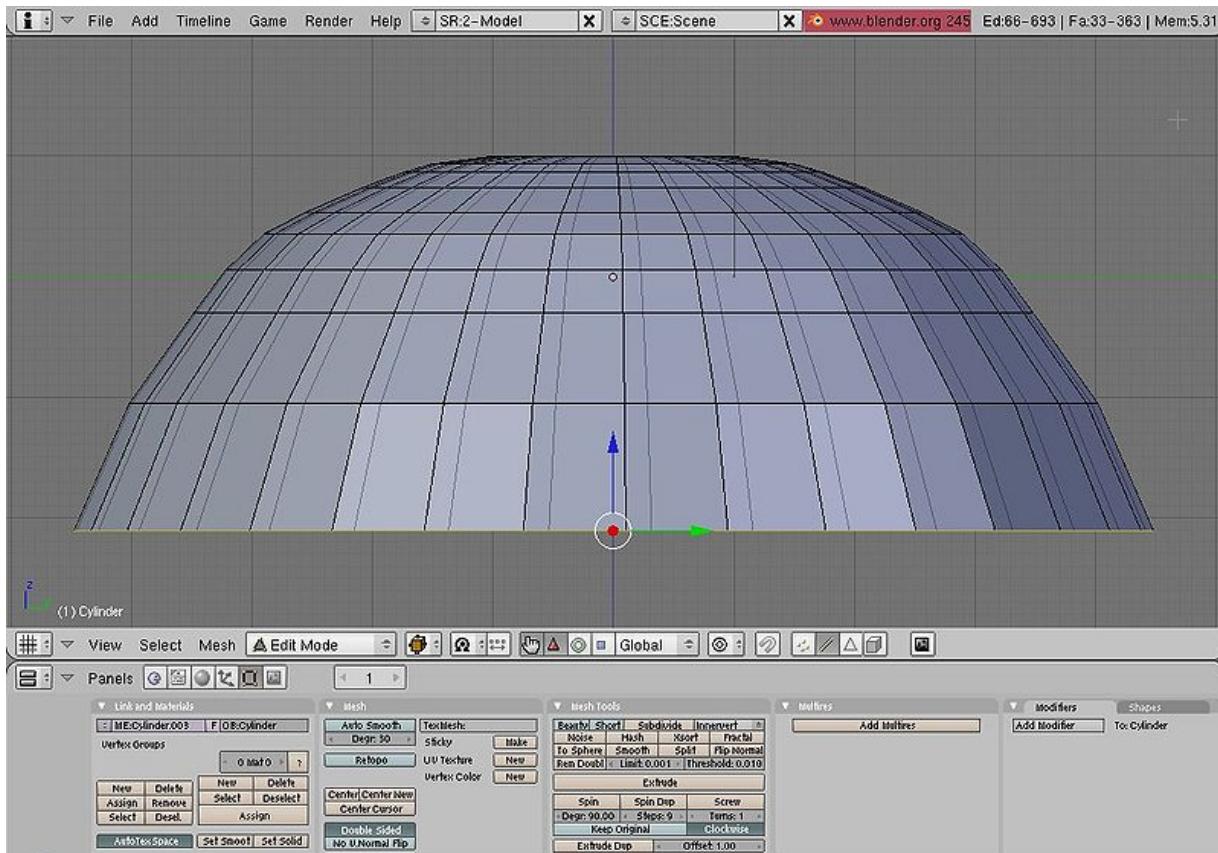
Press the B key, select the bottom row of vertices, and press the X key to delete and then confirm.

(Noob-Note: Instead of a cylinder you might want to add simply a circle, and jump directly to the next step, which is:)

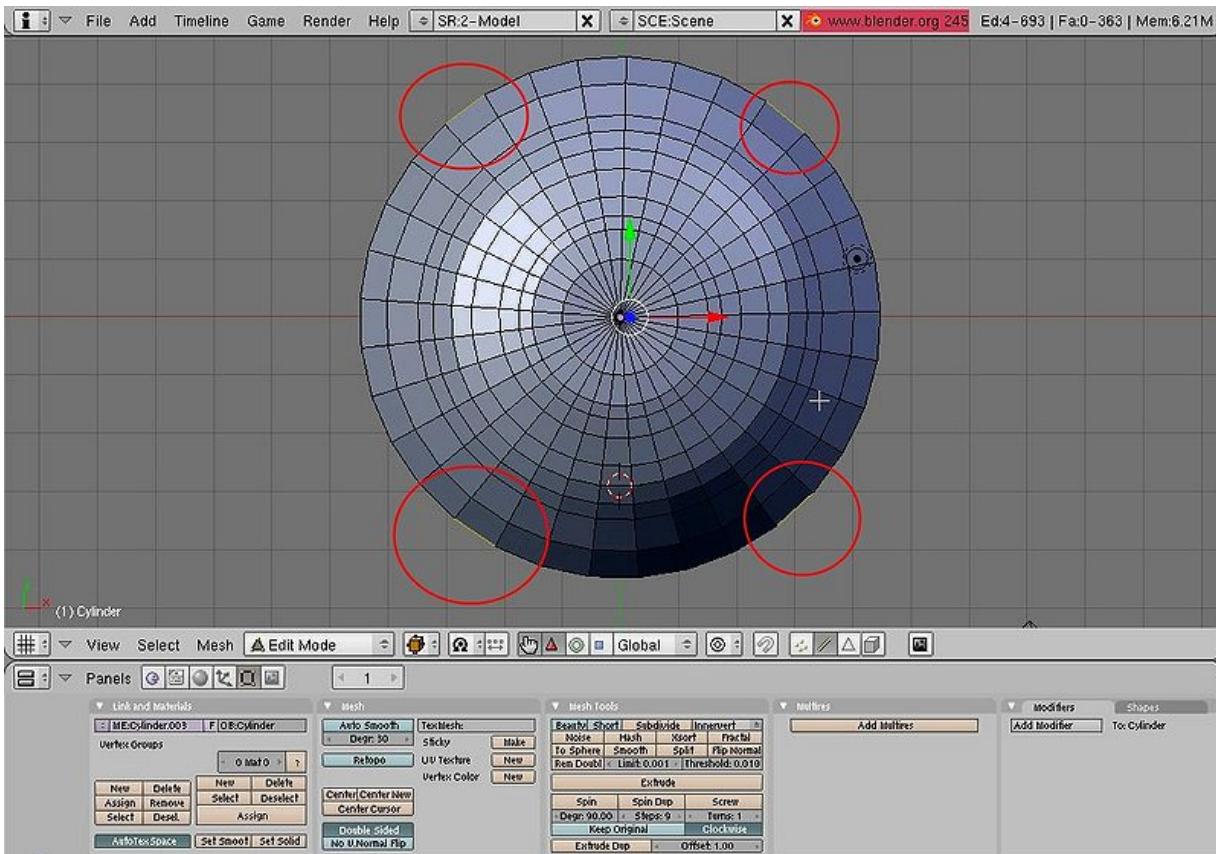
Now press the A key till all the vertices are selected and switch to edge select mode by pressing Ctrl-Tab and then Edges. Press the E key to extrude the selection and the Z key to constrain to the Z axis. Drag the edges a small ways down then click the left mouse button to release. As this makes a right-angle at the sides and parachutes to don't have straight edges, we need to scale the selection outward. Press the S key and move the mouse away from the model. You will see that the edges get smaller and bigger. Scale them out a small ways then left click to release. You will want to practice a little with the scale amounts till you can make a realistic parachute shape. The scene should now look like this:



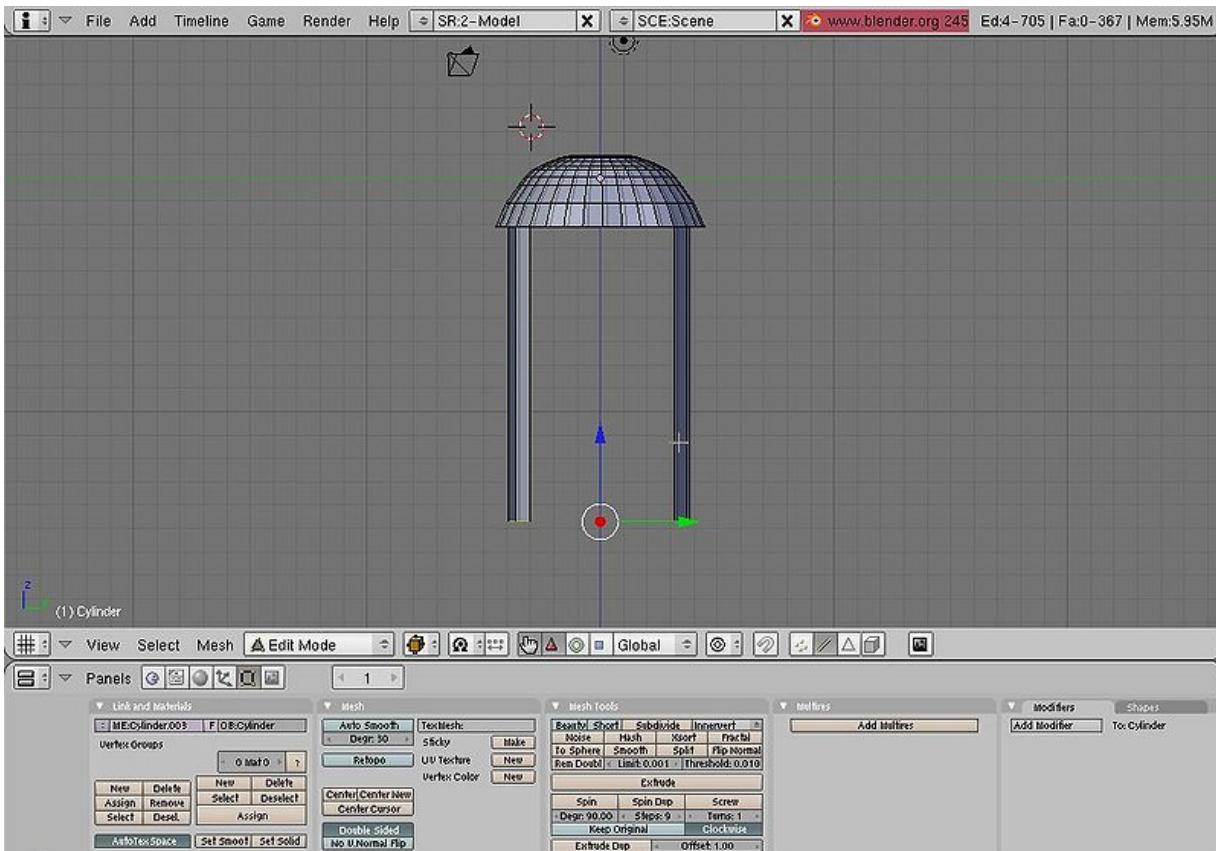
Continue extruding and scaling till you have a shape like the following:



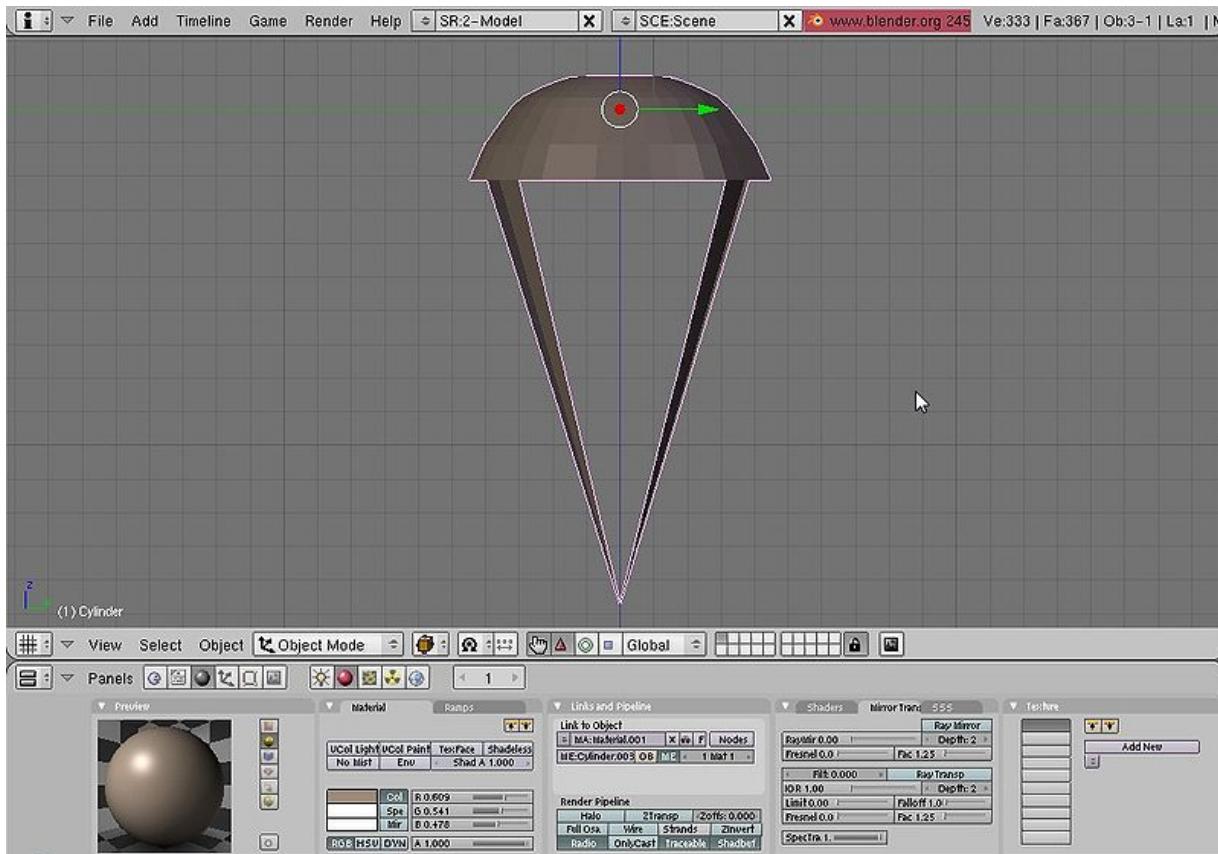
Now go into top-view by pressing the 7 key on the num-pad. Here you will select ~4 edges at opposite sides of each other as in the following picture:



Go back into side-view (num3) and extrude downward by pressing the E key and then the Z key, like in the following picture:



All that remains to finish your parachute is to press Alt-M then "At Center" which will merge all selected vertices. The final result should resemble this:



An alternative method to make a parachute. Start with a cylinder (*basic settings*), delete the center vertex on the bottom. Press **B-KEY** and select the bottom vertices, then press **SKEY** and scale to a value of 2. Extrude down to a value of 1. Subsurf the result and that will give you your basic shape.

Next, select the four corner vertices and extrude down as stated in the tutorial. Select all of the edges created by the extrusion and press **SHIFT + EKEY** to edge crease using a value of 1. You now have a completed parachute (that happens to look identical to the tutorial).

#### Alternative Method

**Noob note:** why working from a cylinder?? Just make a UV sphere (in the right orientation - "poles" aligned on Z axis). Then, looking on side orthographic view, use the box selection method (Key B) to select all the vertices "south to the Equator", then delete vertices (X -> Vertices). The ropes can be made as described above. **Another noob:** That would work, although note you'll not have as much control over the chute's shape that way; in fact, it'll produce a rather different shape that way.

**Yet another Noob:** I followed the instructions and the resulting parachute is solid on the bottom and not concave as it should be (am I the only one with this result?). Anyway, I fixed it by going to the bottom view (Ctrl, NUM7) and deleting the vertex in the middle of the bottom view. Now it is nice and concave!

# Blender 3D: Noob to Pro/Model a Low Poly Head

This tutorial is designed to teach users to make a low-poly animesque head in Blender.

All of the pictures were made by me as Blender3D screen shots. They are all free for any form of use. *NOTE: Pictures go from left to right*

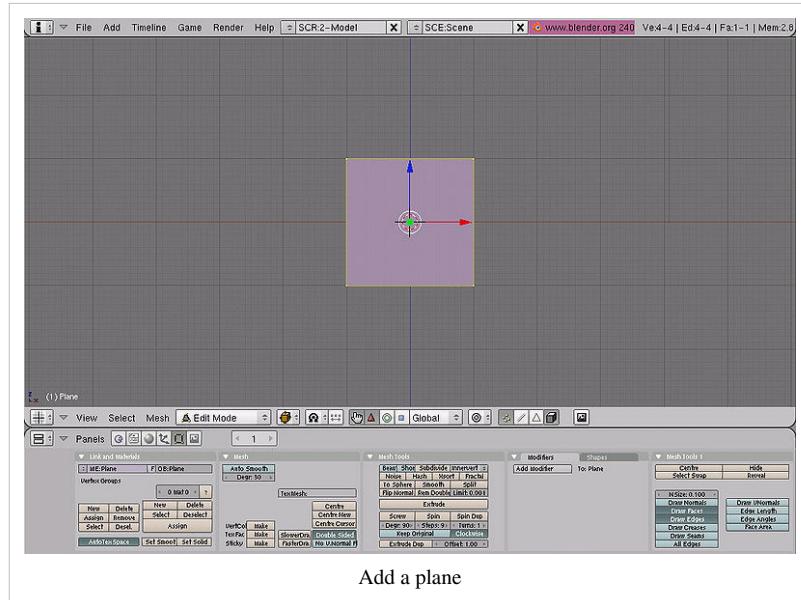
## Overview

What you need to know:

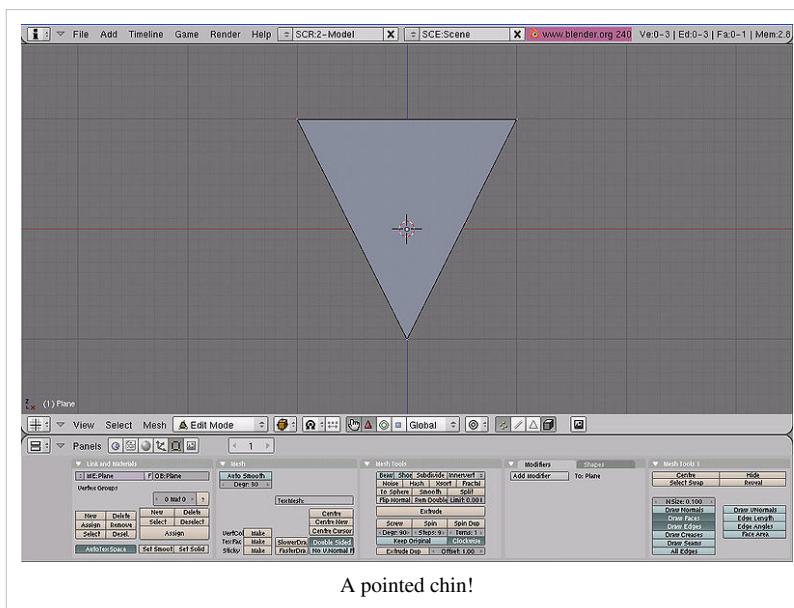
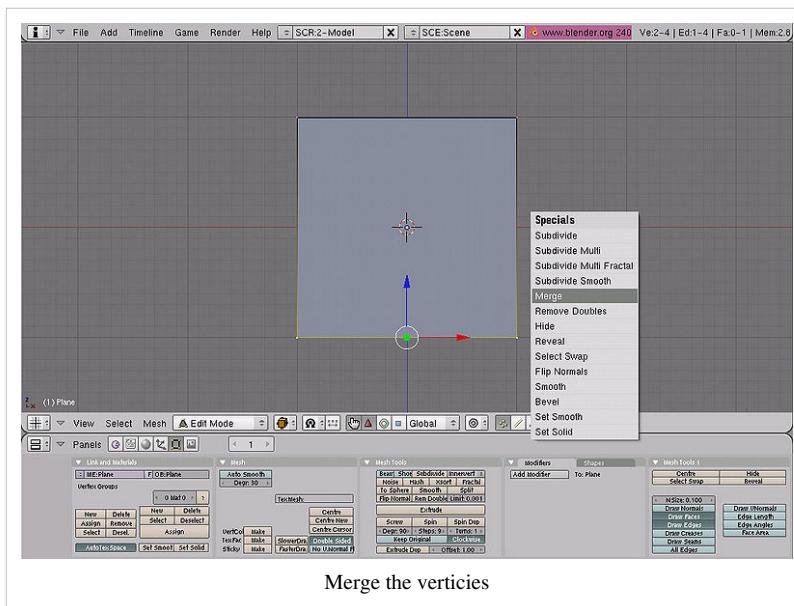
- Basic Blender controls

## Making the Model

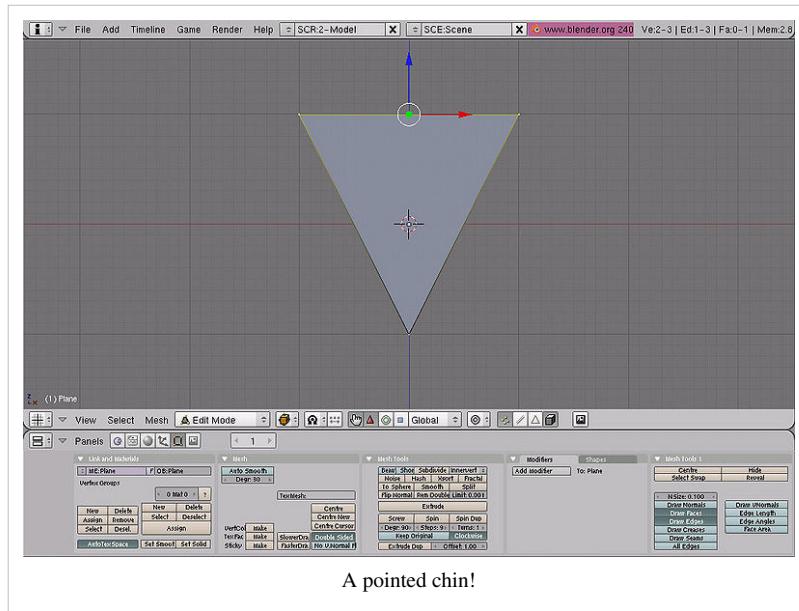
**i** Before doing anything, delete the default cube by selecting it in object mode, pressing **XKey**, and confirming. Go into the front view (keypad 1) and add a plane.



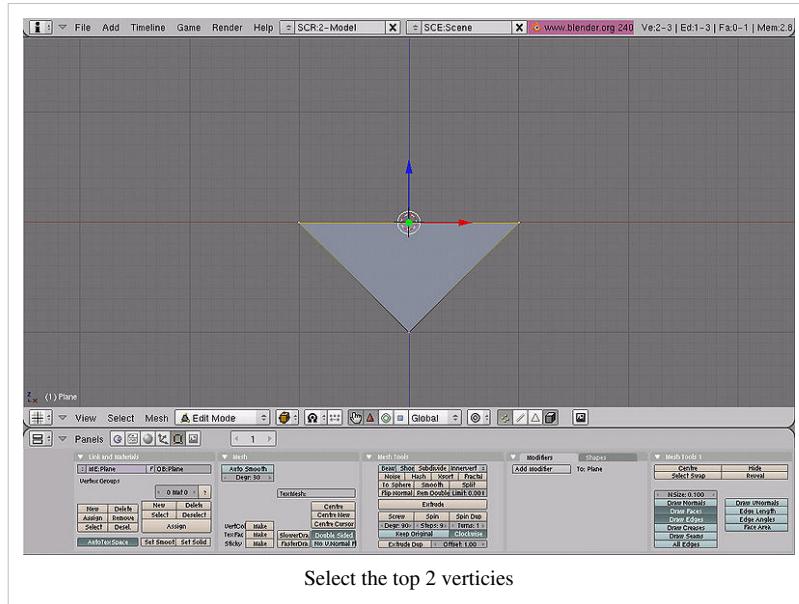
Select the bottom two vertices (Shift + Right-Click) and press W to bring up the vertex menu. "Select Merge", then "At Center".



Select the top two vertices (use the **BKEY**) and re-arrange them so they make more of a chin shape by pressing the **GKEY** to move and **ZKEY** to constrain the movement to the Z-Axis.

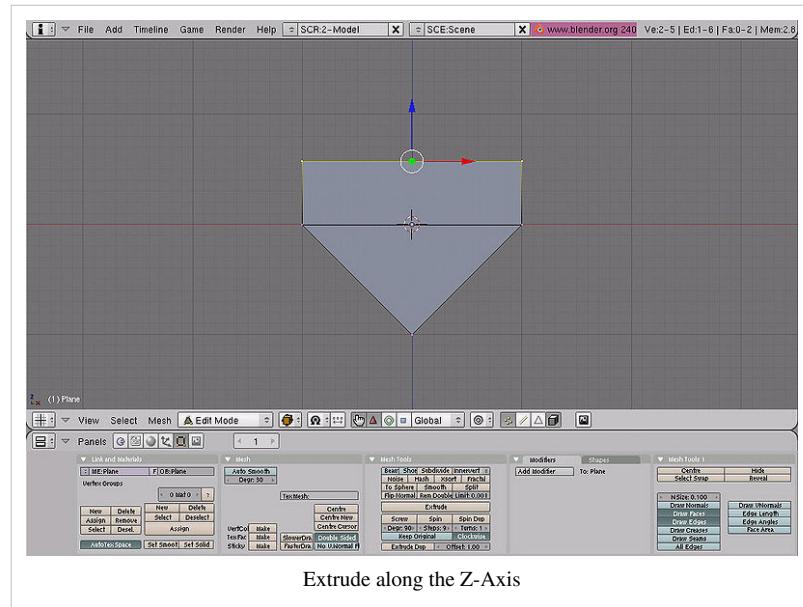


A pointed chin!

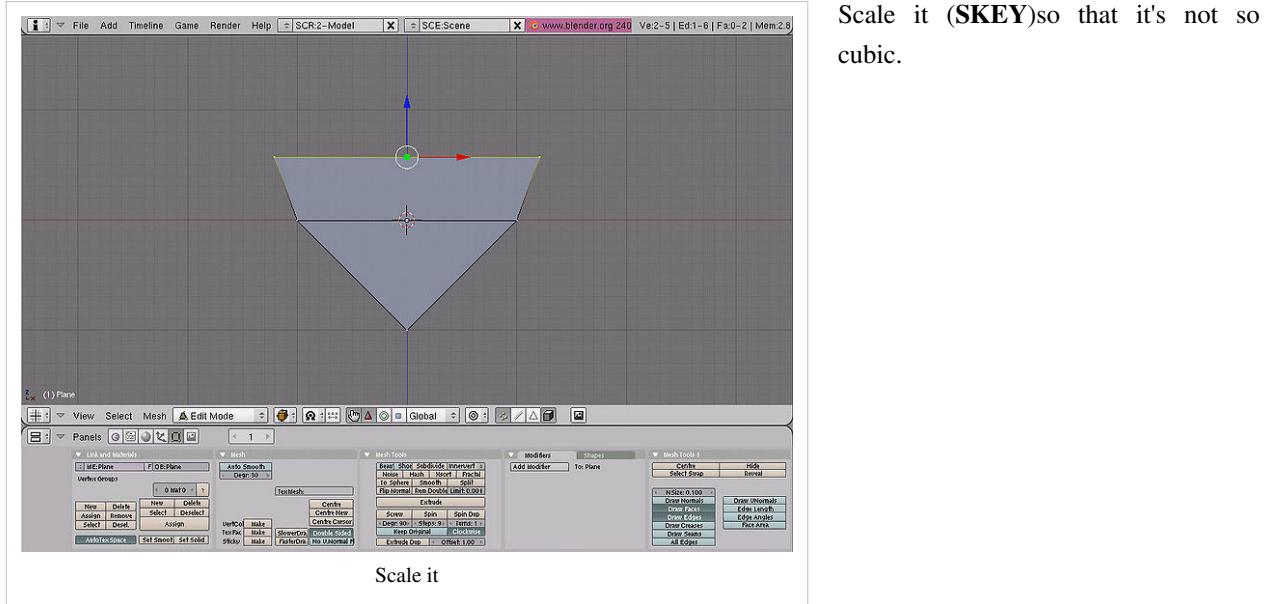


Select the top 2 verticies

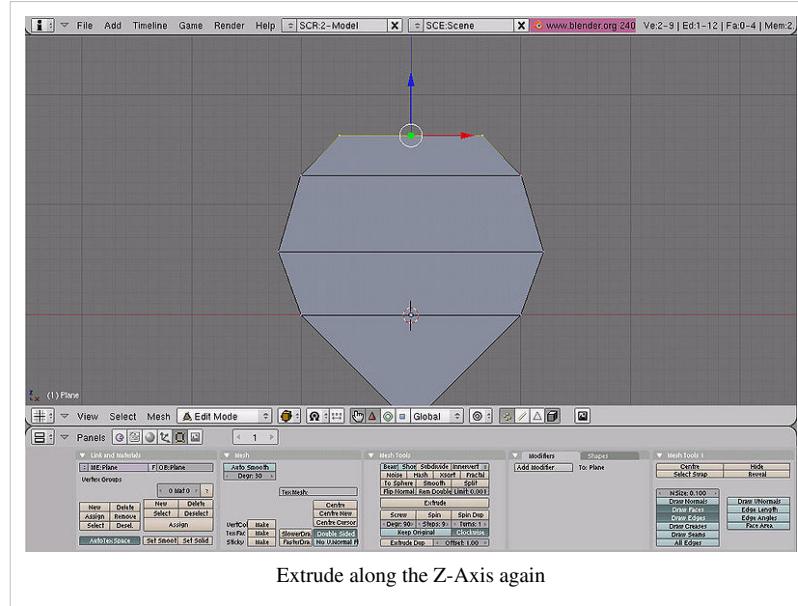
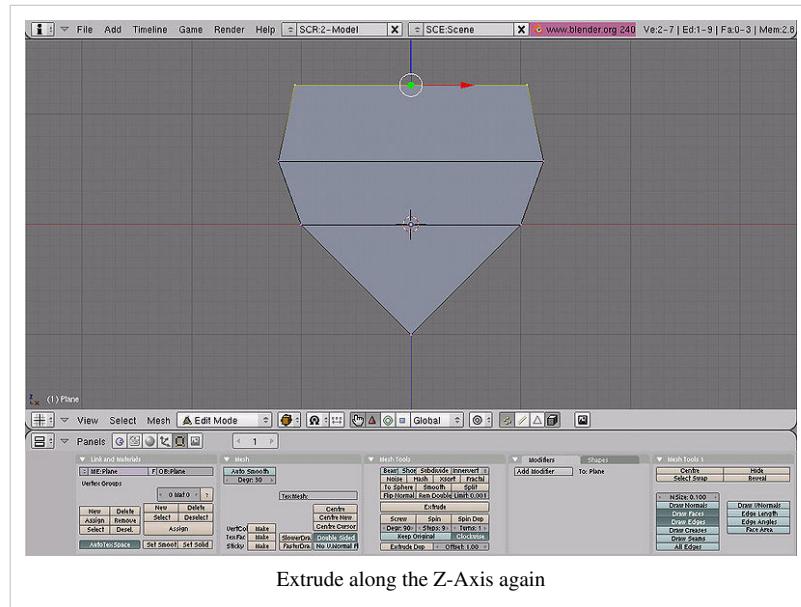
Now extrude (**EKEY**) along the Z-Axis (**ZKEY**) so that you have another area.



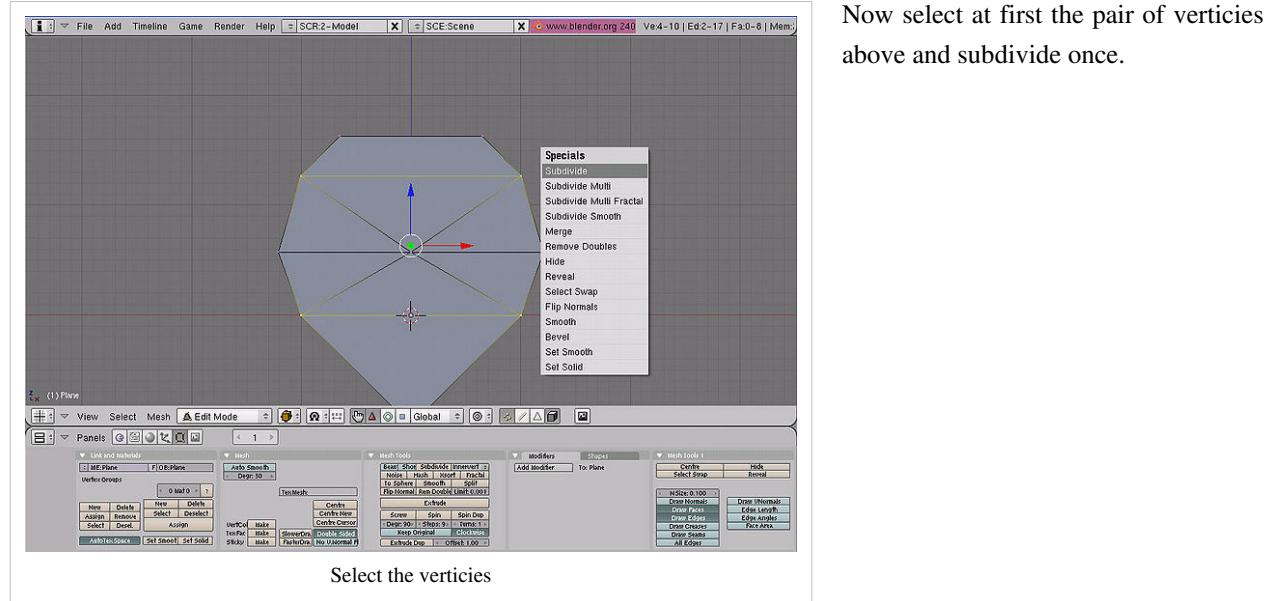
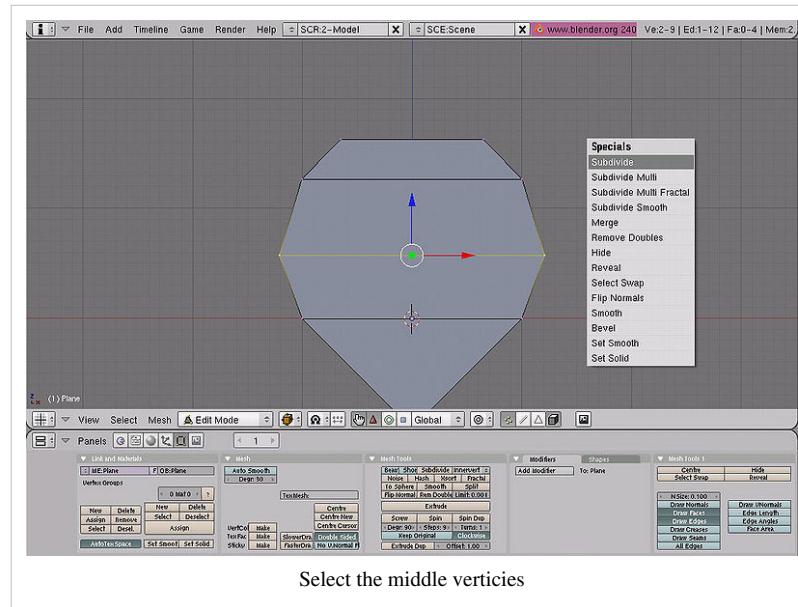
Extrude along the Z-Axis

Scale it (**SKEY**) so that it's not so cubic.

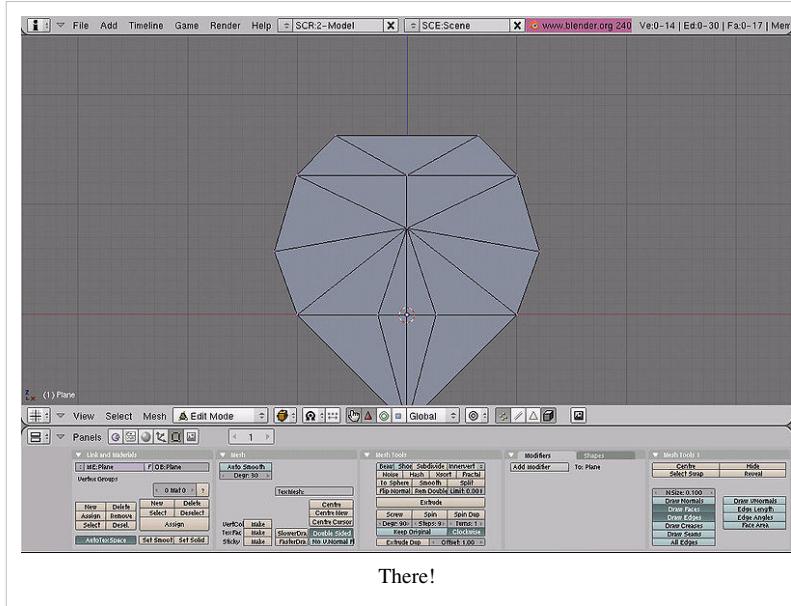
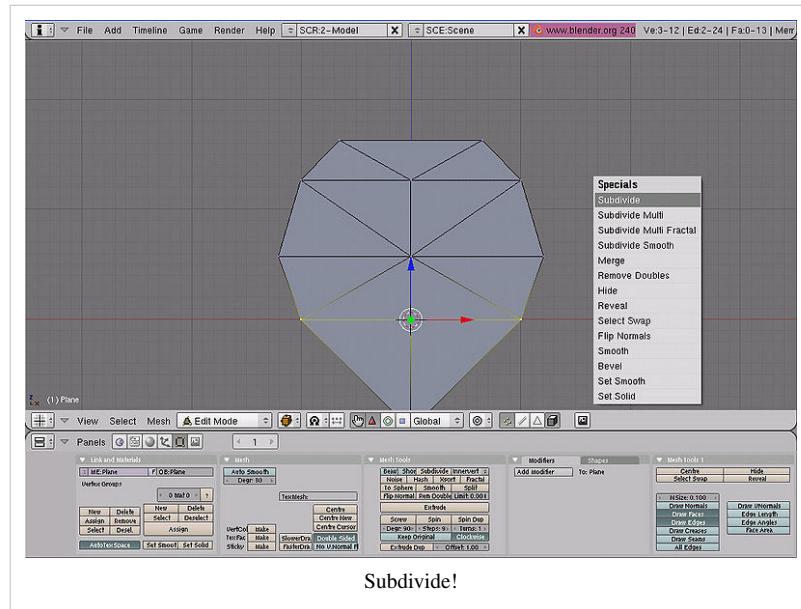
Now extrude along the Z-Axis again (**EKEY** then **ZKEY**) and scale (**SKEY**) it down a bit.



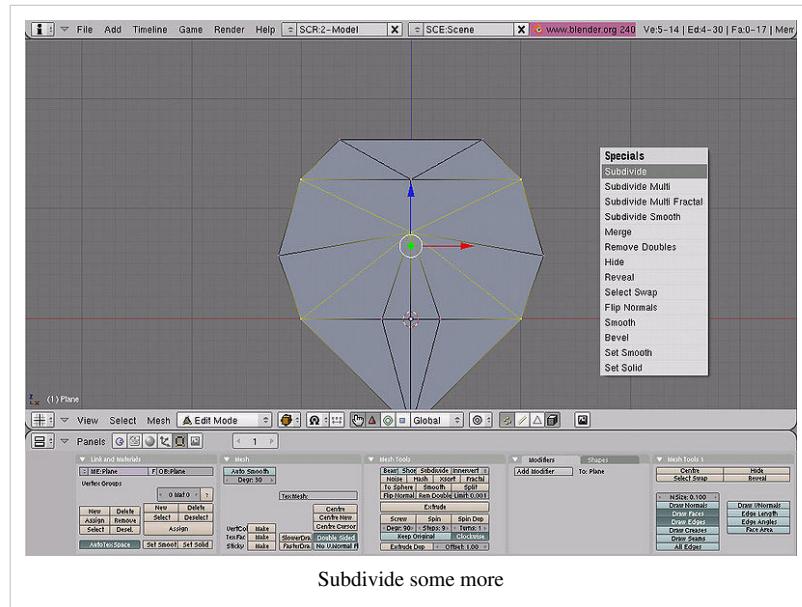
Select the middle vertices (**BKEY**) and press (**WKEY**) to bring up the edit menu again. Subdivide it once.



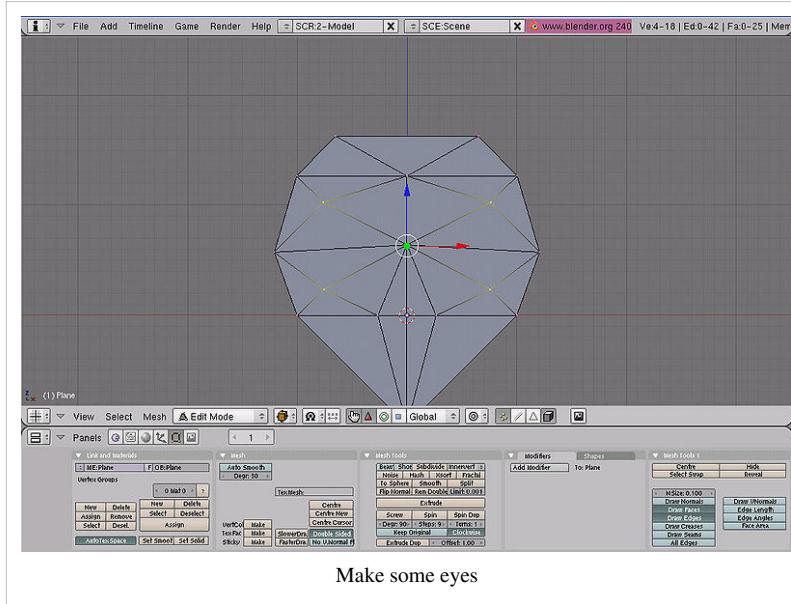
Do the same two times with the pair below.



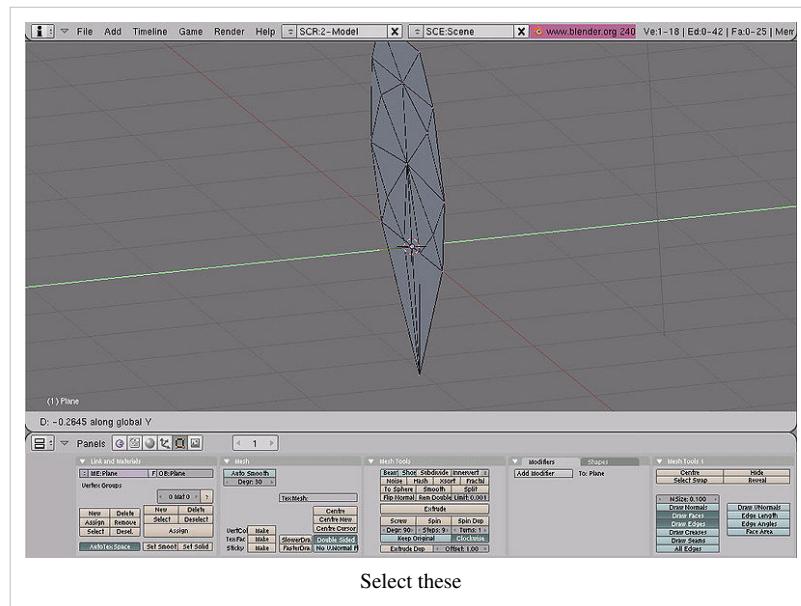
Select and subdivide here, so you can make the eyes.



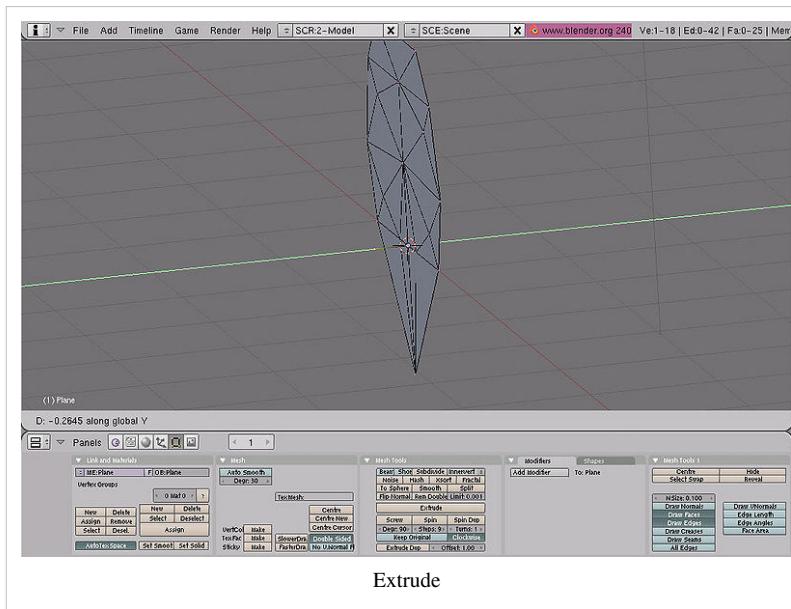
And rearrange the vertices so they make eye shapes.



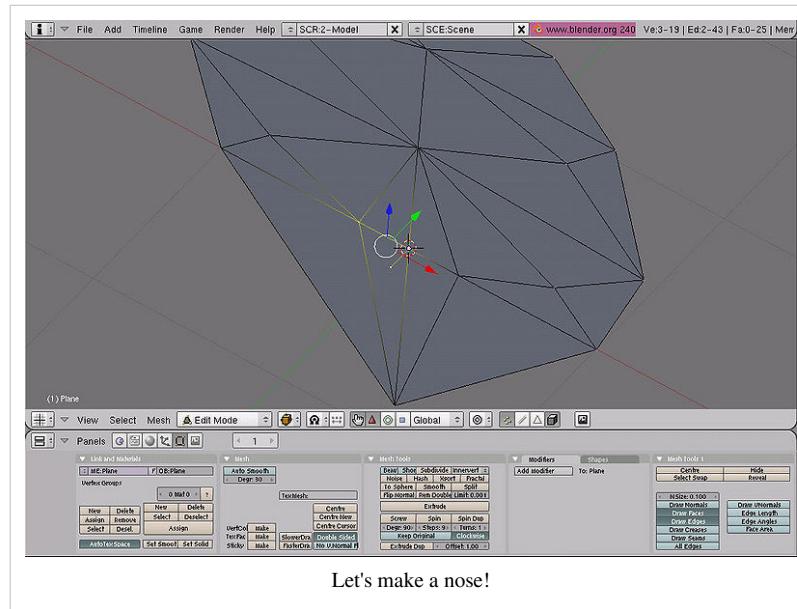
Now select the middle vertex here (right mouse button).



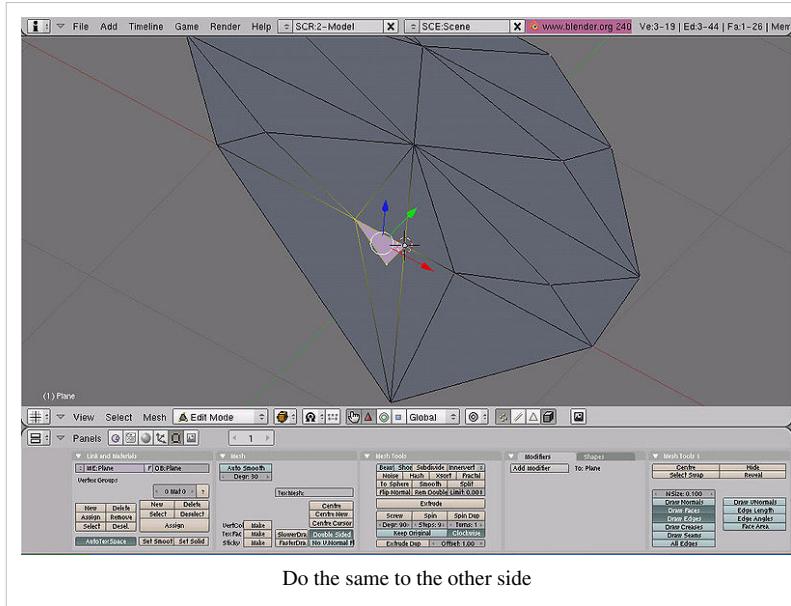
Extrude it along the Y-Axis (**EKEY**) and move your mouse around to change how far it moves.



Now we make the nose. Select these three vertices (Shift + Right mouse button) and press (F) to make a face.



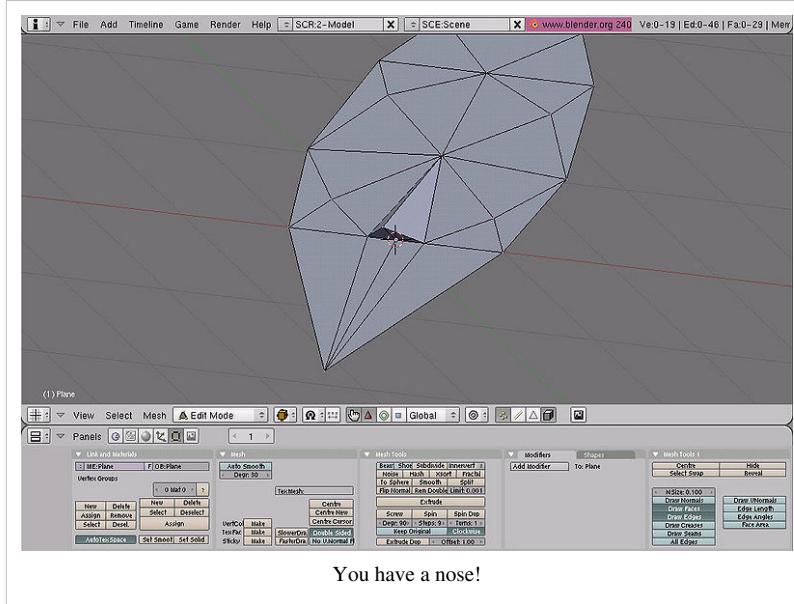
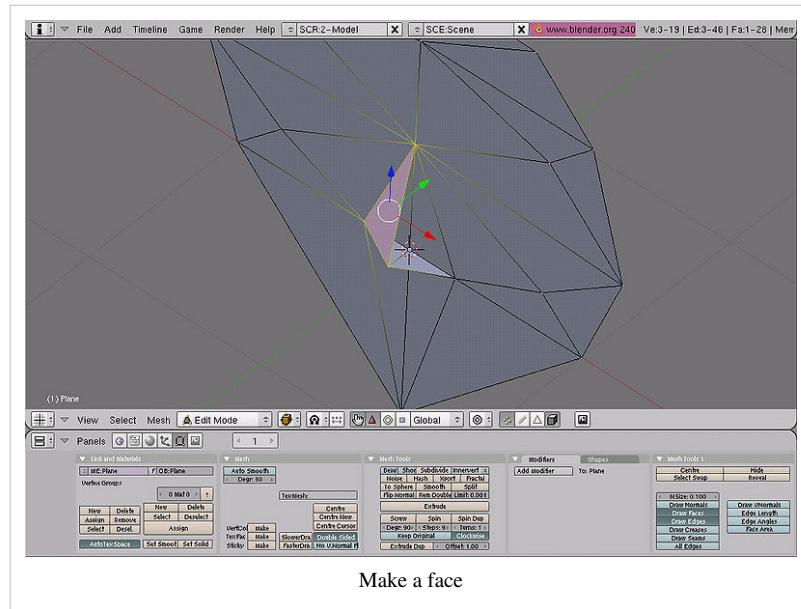
Let's make a nose!



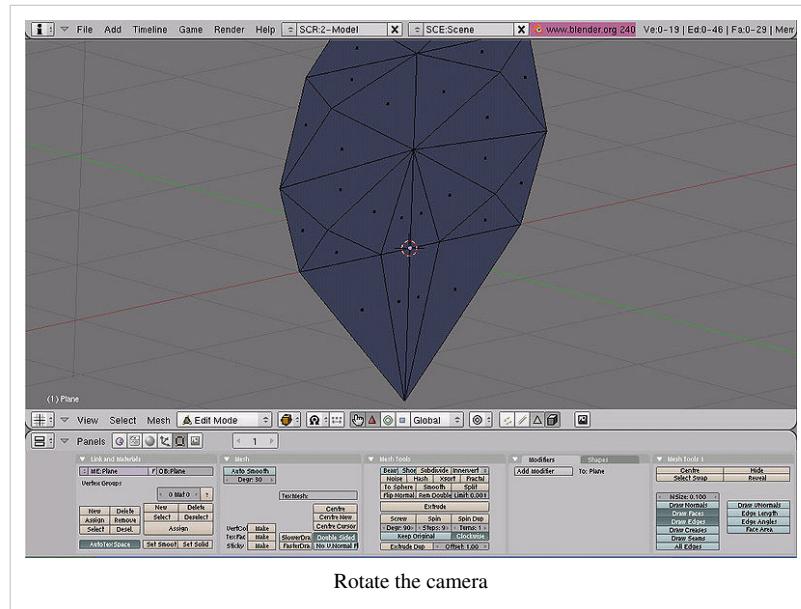
Do the same to the other side

Now do the same to the other side, and you will have the nose's base.

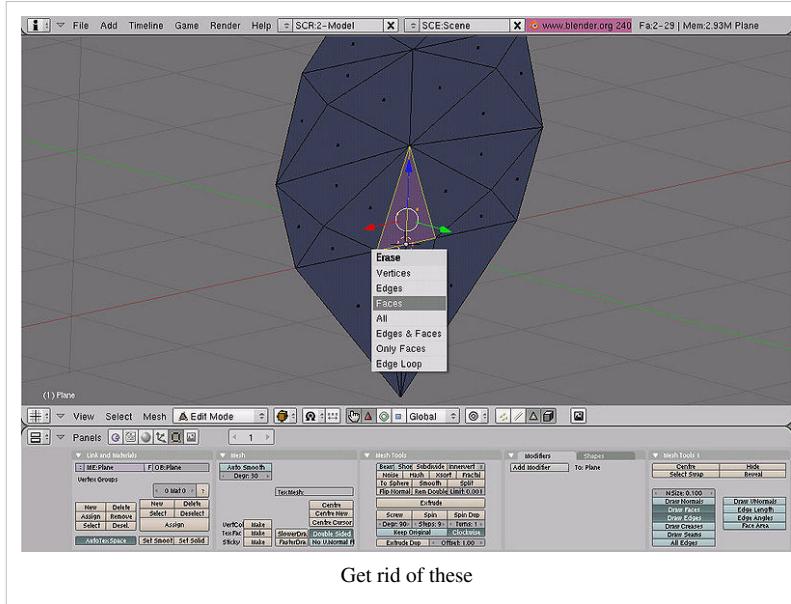
Select these vertices (Shift + Rightclick) and make a face (F). Do this to the other side.



Rotate the camera so you are seeing the back. Press **CTL+NUM1**. Also, change to face select mode.



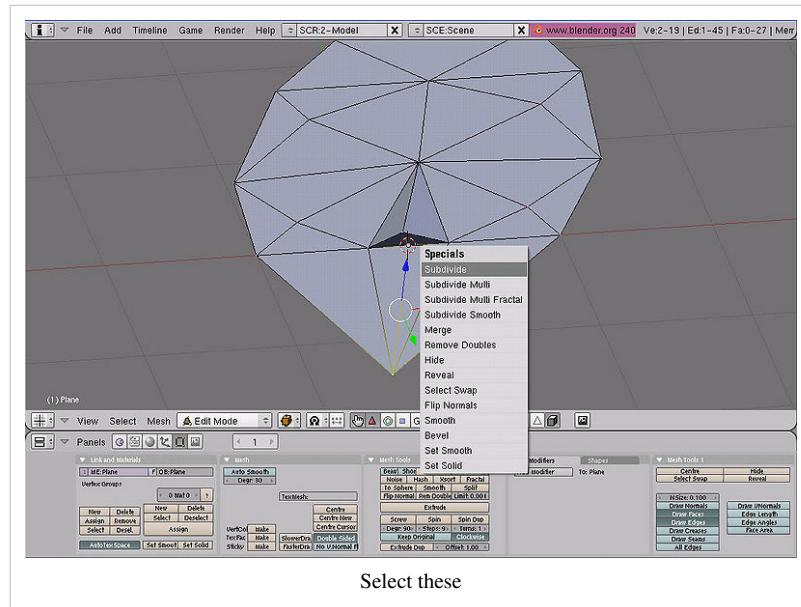
Rotate the camera



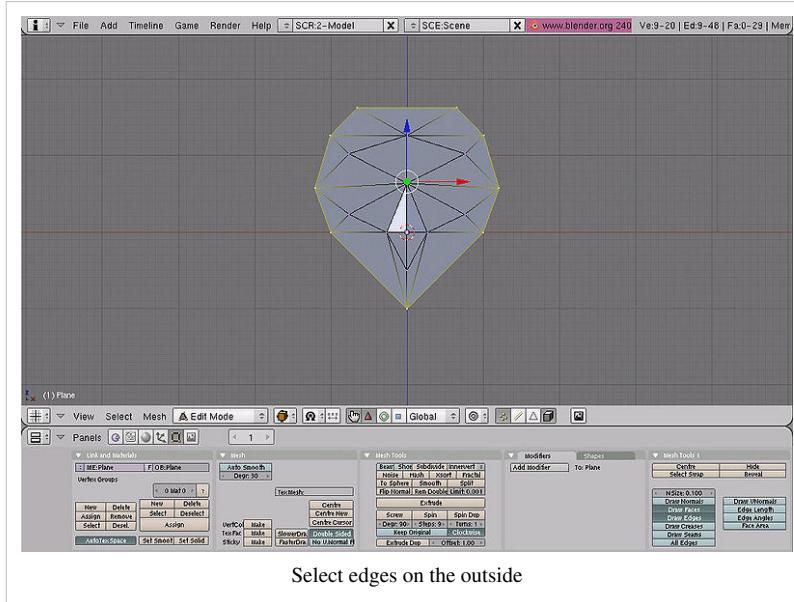
Get rid of these

Select the faces behind the nose. (Shift + Rightclick) and press (X). This will bring up the delete menu. Selected "Faces" to get rid of these.

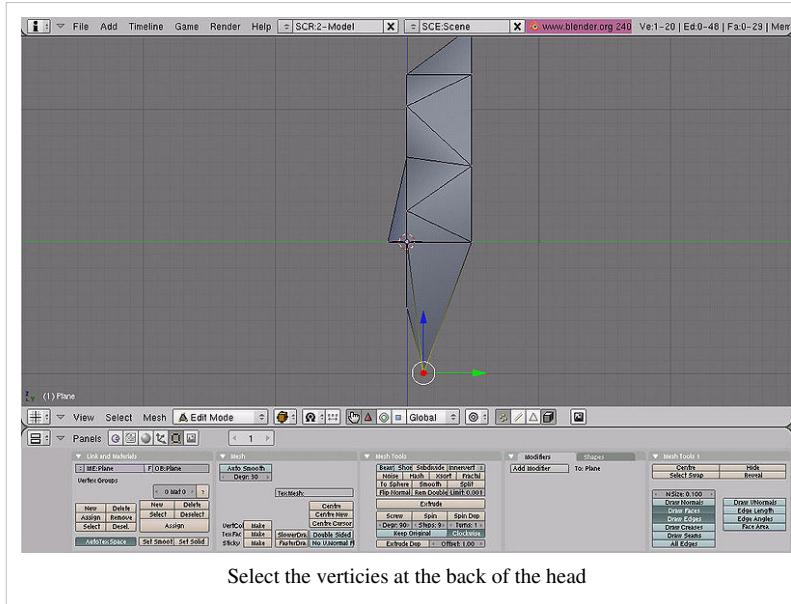
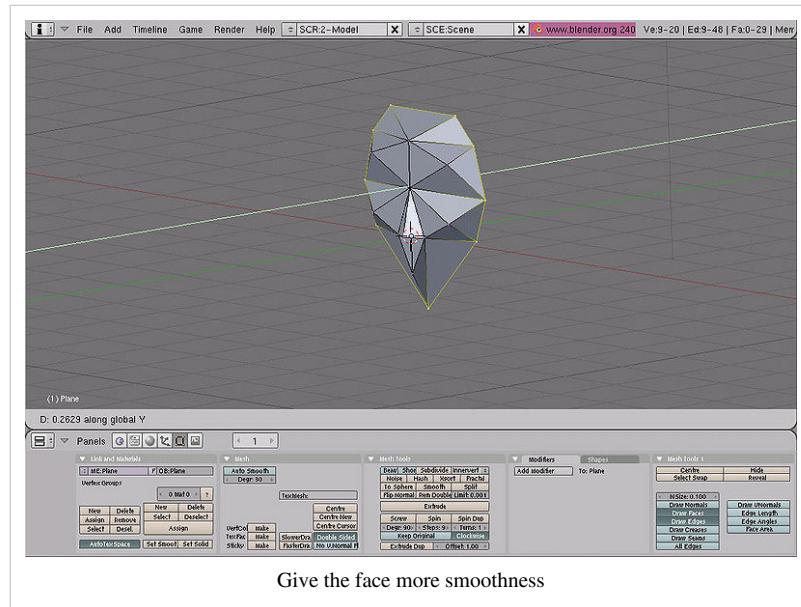
Now select these vertices (above and below(user comment: the one at the bottom of the chin, and the one at the bottom of the nose right above the first one), and subdivide them.



And select all the edges on the outside.

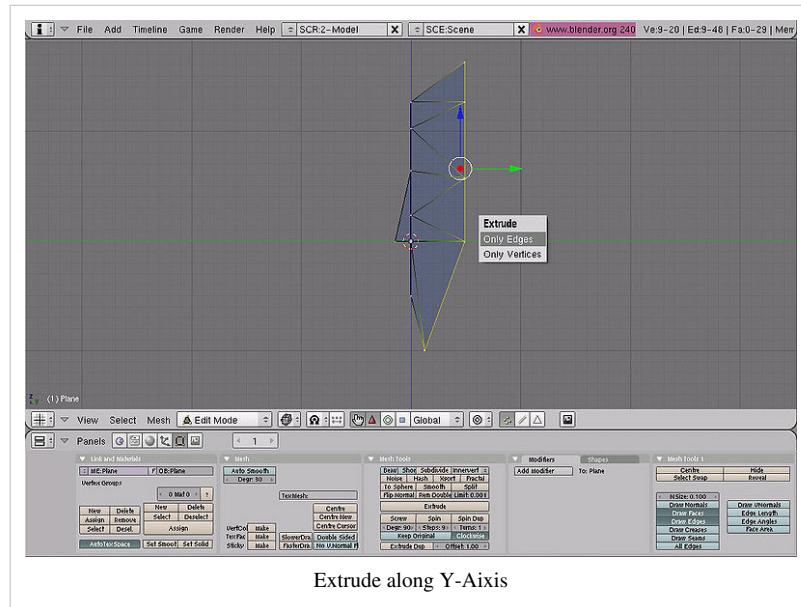


Move the vertices back a bit to give the face more smoothness (G) (Y)and rearrange the chin.

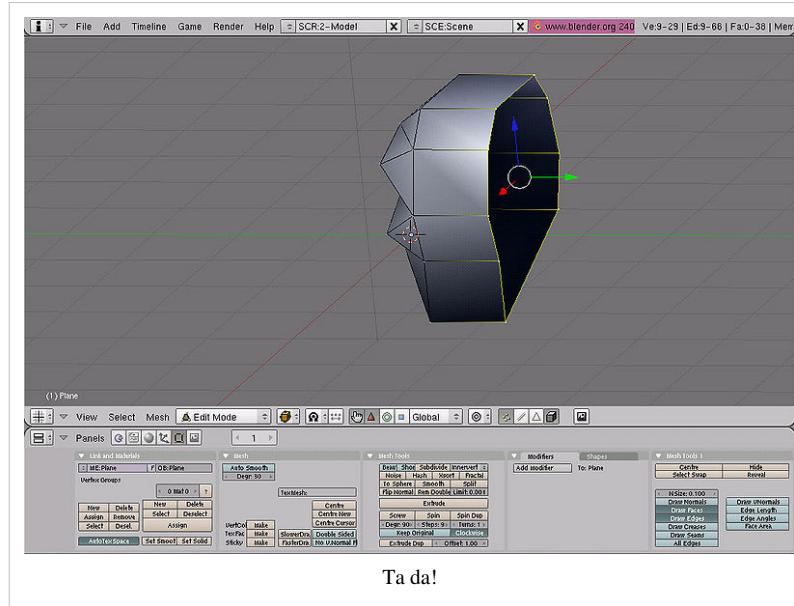


Now select the vertices at the back of the head, using (B).

Extrude these along the y axis (E) (Y)



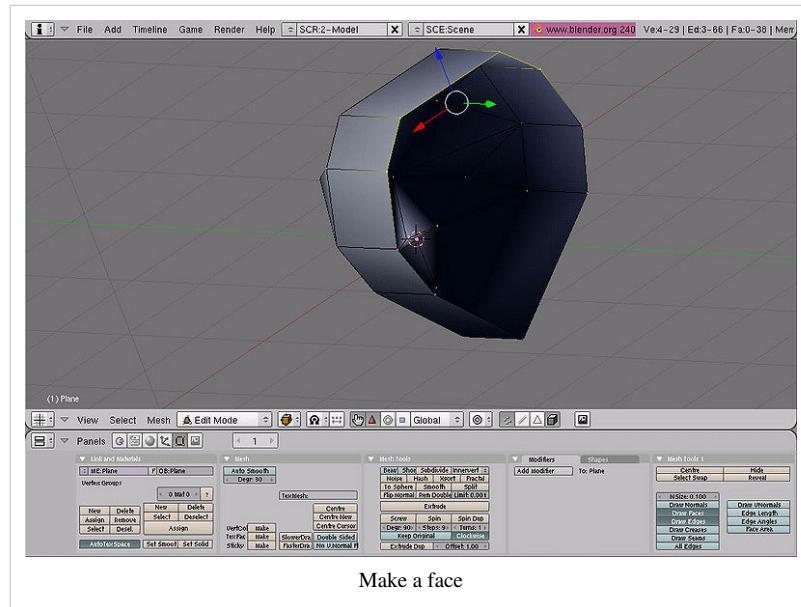
Extrude along Y-Axis



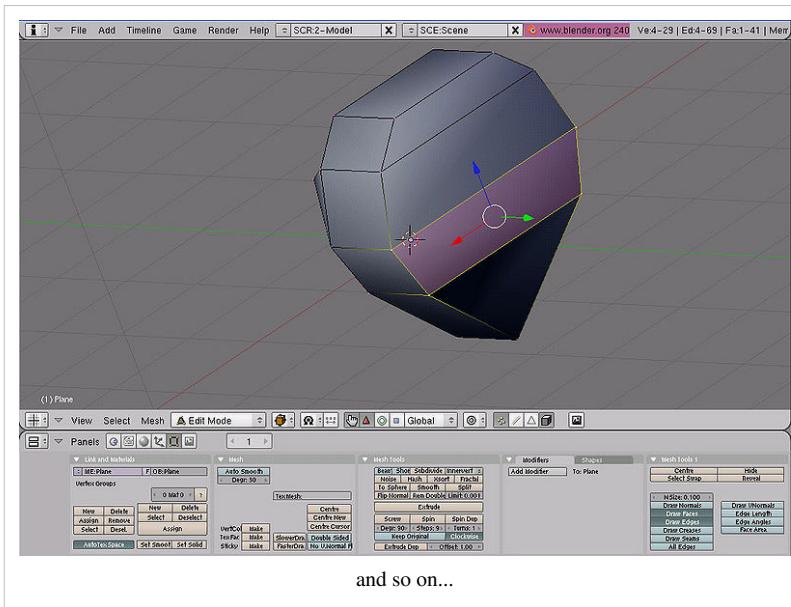
Ta da!

Ta da!

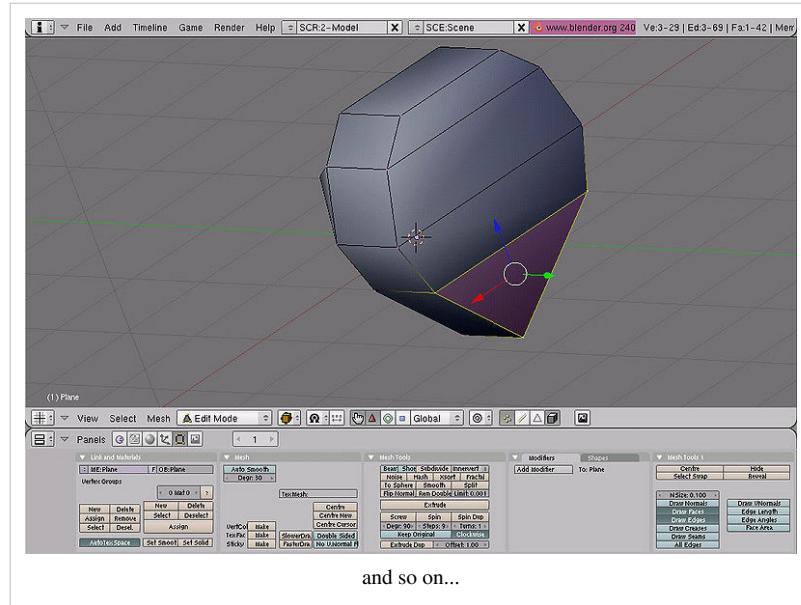
Now select the back vertices and press (F) to make a face.



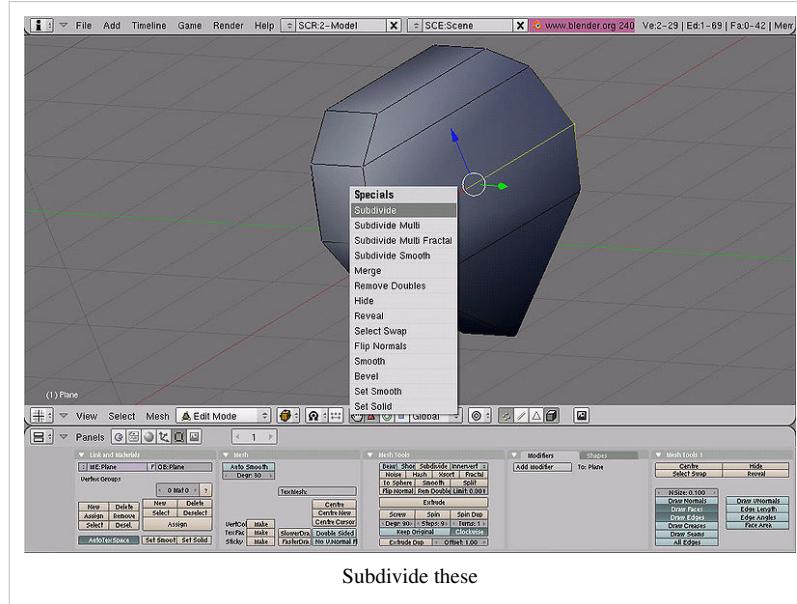
and so on....



and so on....



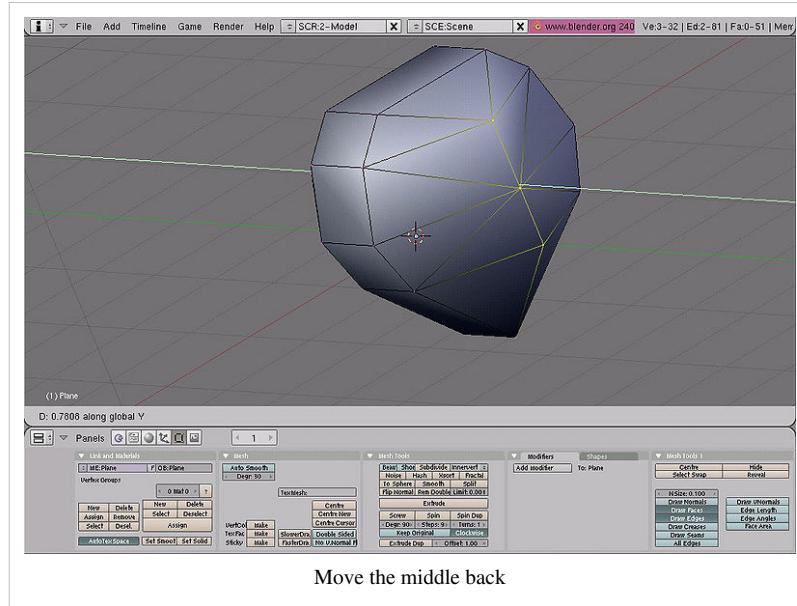
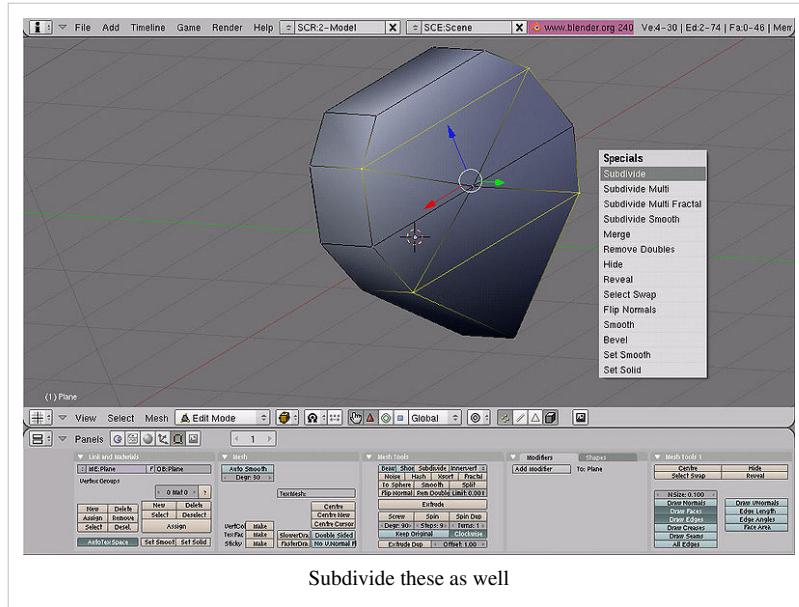
and so on...



Subdivide these

now select and subdivide these once  
(W)

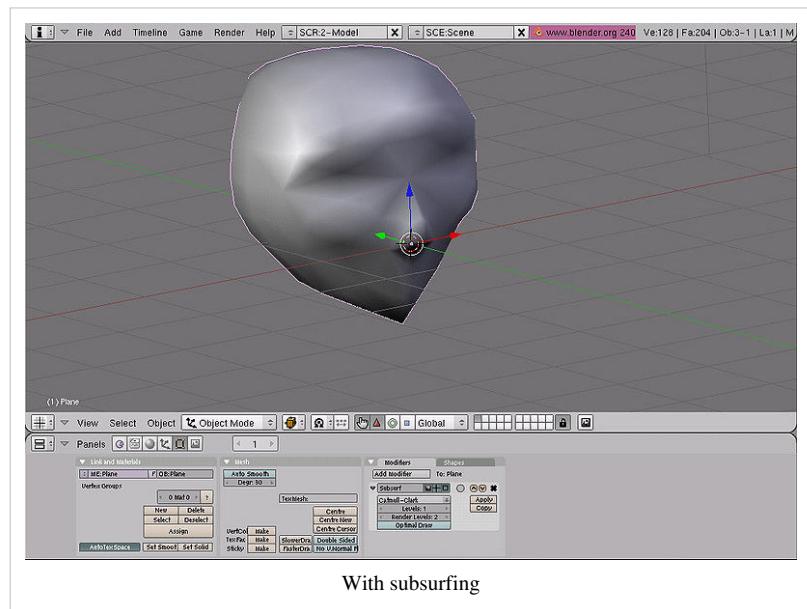
and these points too.



Now select the middle and move it back (G) (Y).

## Finishing it up

And here it is with sub-surfacing. You have finished. Hit (F12) to see the final render.



# Blender 3D: Noob to Pro/Building a House

This tutorial is based on Bart Veldhuizen's "Building a House" from Tutorial #01 published by NaN in 1999 which is also available in a PDF at <http://download.blender.org/documentation/BlenderTutorialGuide1.tar.gz> (a tar.gz containing BlenderTutorialGuide1.pdf) or <http://download.blender.org/documentation/BlenderTutorialGuide1.zip> (a ZIP file containing the same PDF).

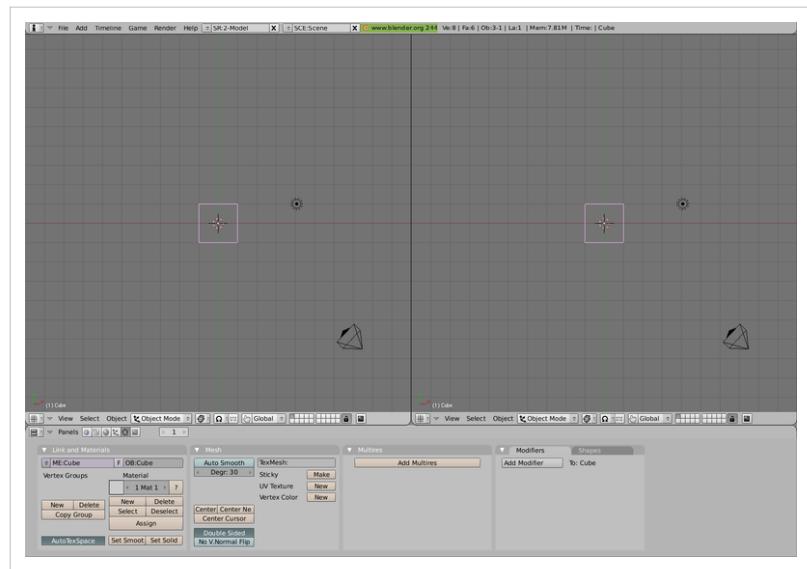
Permission was asked to use it and Ton Roosendaal said "Be assured that everything that was produced by NaN now is open and free content for everyone to reuse, including the tutorial "Building a House"."

## Setting the Scene

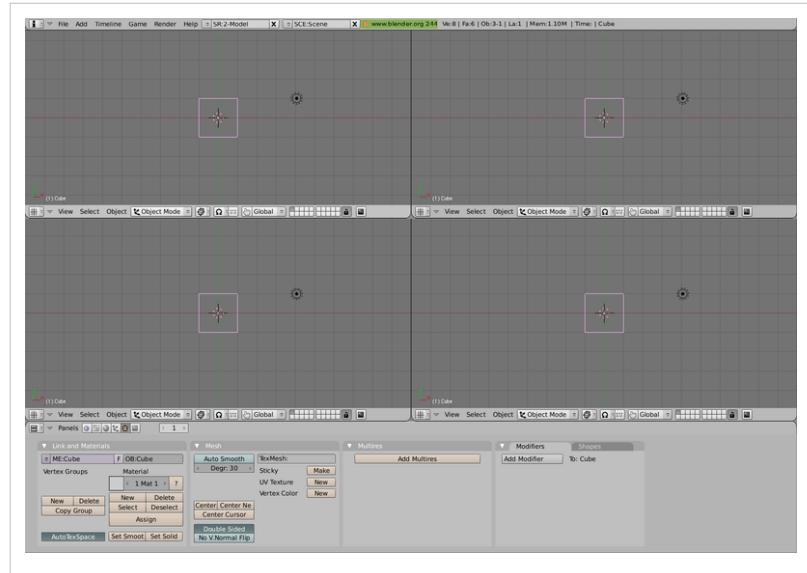
Start a new scene (CTRL-X) and leave the default cube which will become the body of the house.

Split the window into two side-by-side windows: move the mouse to the line between the 3D window header and the Button window where it becomes an up-down arrow.

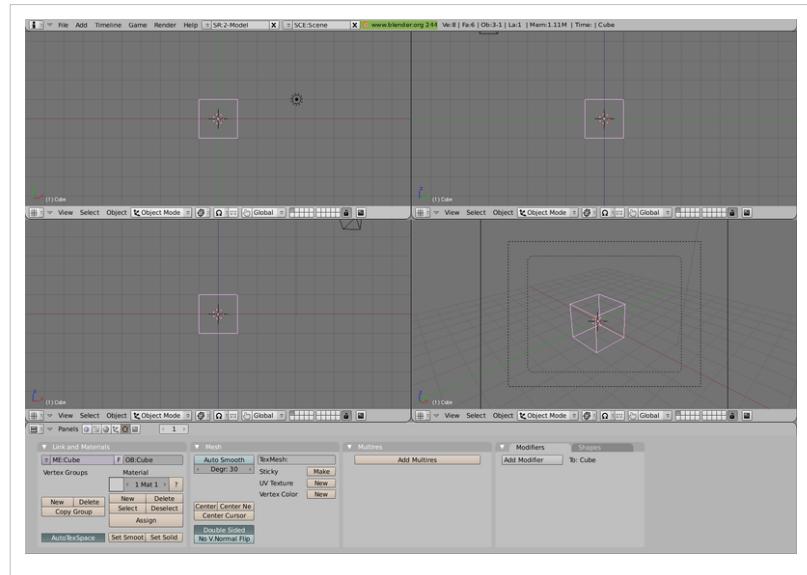
Press **MMB** and select "Split Areas" from the menu. Move the line which appears to the middle and click **LMB**.



Now move the mouse to the line separating these two windows, click **MMB** again and split the lefthand window into top and bottom windows. Do the same to the righthand window to get four windows.

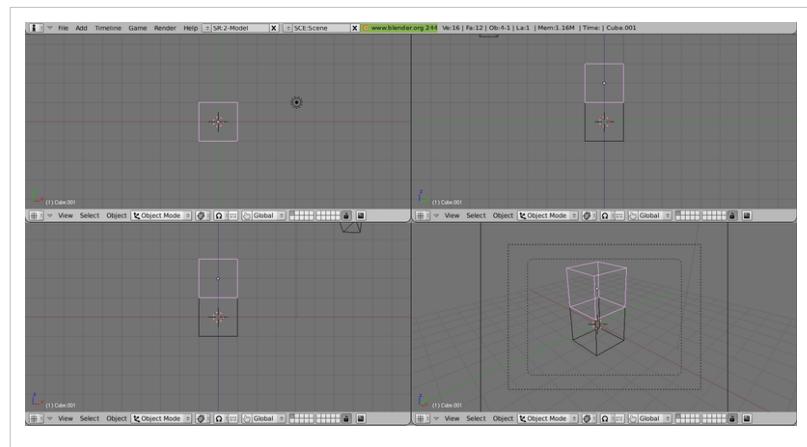


Move the mouse to the lower lefthand window and press **NUM1** to put it into Front view; move it to the upper lefthand window and press **NUM7** to put it into Top view. Move the mouse to the top righthand window and press **NUM3** to put it into Side view and then to the bottom righthand window, pressing **NUM0** to get the camera view.



## Making the roof

Now, in the Front view window (lower left) press **SHIFT-DKEY** to duplicate the cube: grab mode is automatically selected, press **ZKEY** to restrict the move to the up-down axis and move the cube to rest on top of the original one — pressing **CTRL** while moving will snap it to the grid and make it easier to position accurately. Press **ENTER** when it is in place. **Newbie Mistake--** Make sure you are in object

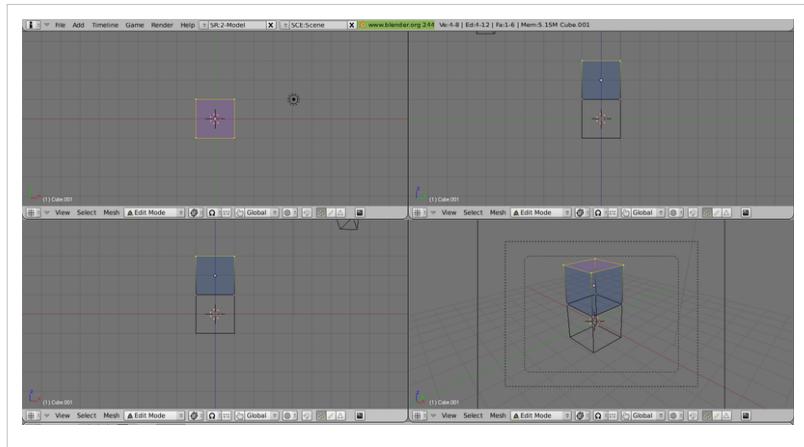


mode when you duplicate the cube to create your roof. It needs to be its own object so you can edit it separately from the house object. You can see how useful this four-paned window is, it shows you exactly what's going on.

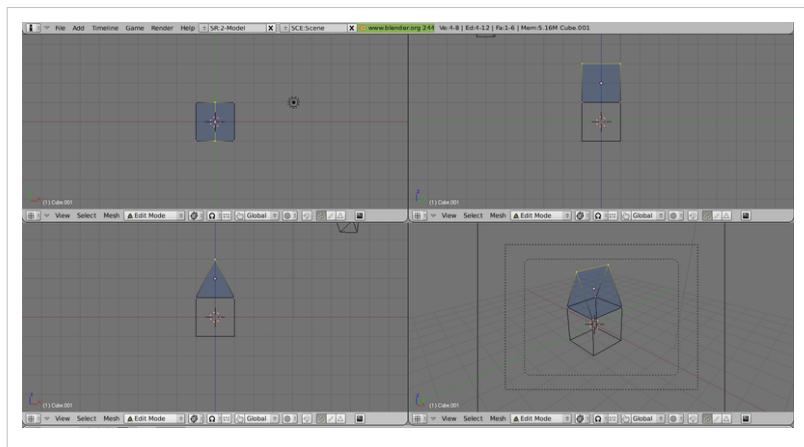
Wouldn't it be simpler to enter edit mode, then enter face select mode, select the top face of the cube and extrude it along the Z axis by 2 units? It'll also make manipulating the house much easier with it as one object. This method is not for people who don't know how to edit things separately from another part of the same object. It is possible to do the above in edit mode and still edit the roof separately, and is explained in previous tutorials. (You just select the parts of the object you wish to edit and edit them.)

--Mast3rlinkx (talk) 15:33, 11 February 2010 (UTC)

The top cube is going to become the roof and needs to be made into a triangle: select the top cube (if it is not already selected) by clicking **RMB** on it, press **TAB** to go into Edit Mode (check the box in the middle of the 3D window header), press **AKEY** to deselect all the vertices and box select (**BKEY**) the top four — you will be able to see that they have been selected in the camera view (lower righthand) window.



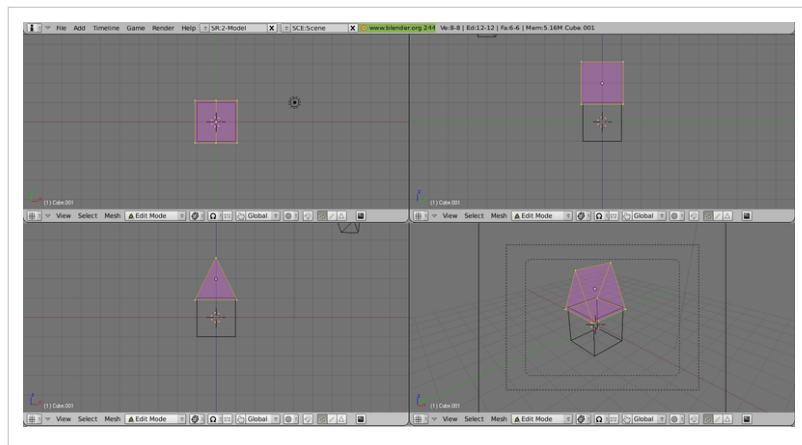
Now press **SKEY** for scaling and **XKEY** to limit the scaling to the X-axis (left to right) then either move the mouse to bring the top vertices together or just type in 0 followed by **ENTER** to set scaling to zero. (In order to activate the dot you must point to the object click left, then hold right click and then release, in this way you can change the shape of the object moving the dot up and down, left and right).



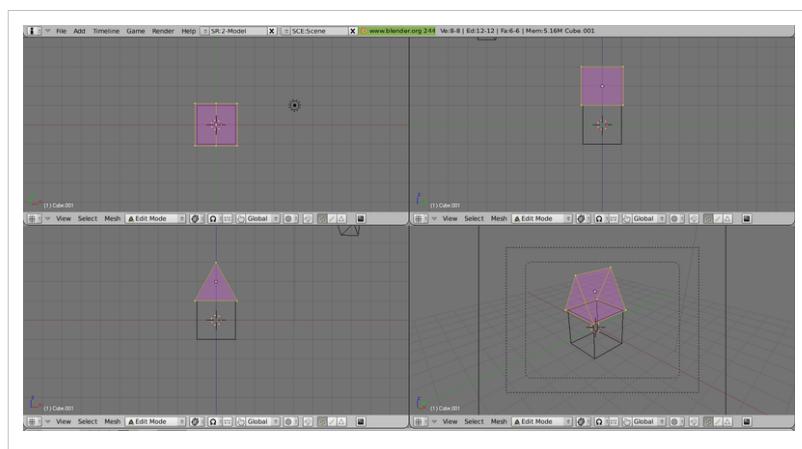
(**Note:** It's easier to choose 2 vertices and merge them at Center (**ALT+MKEY**, choose **Center** and click the Window). Repeat this with the other 2 vertices and you'll have the same result.)

The roof also needs to project over the walls of the house: this can be done in two ways: either scale the whole roof making it taller as well as wider and longer front-to-back or it can be left the same height but just scaled (in two separate actions) left-to-right and then front-to-back. Or you can scale it and press **SHIFT+ZKEY** to keep the height the same while scaling the rest.

The first way is simple: press **AKEY** to deselect everything and then box select the roof. Scaling the whole roof to 1.1 looks about right. Zoom in if needs be.



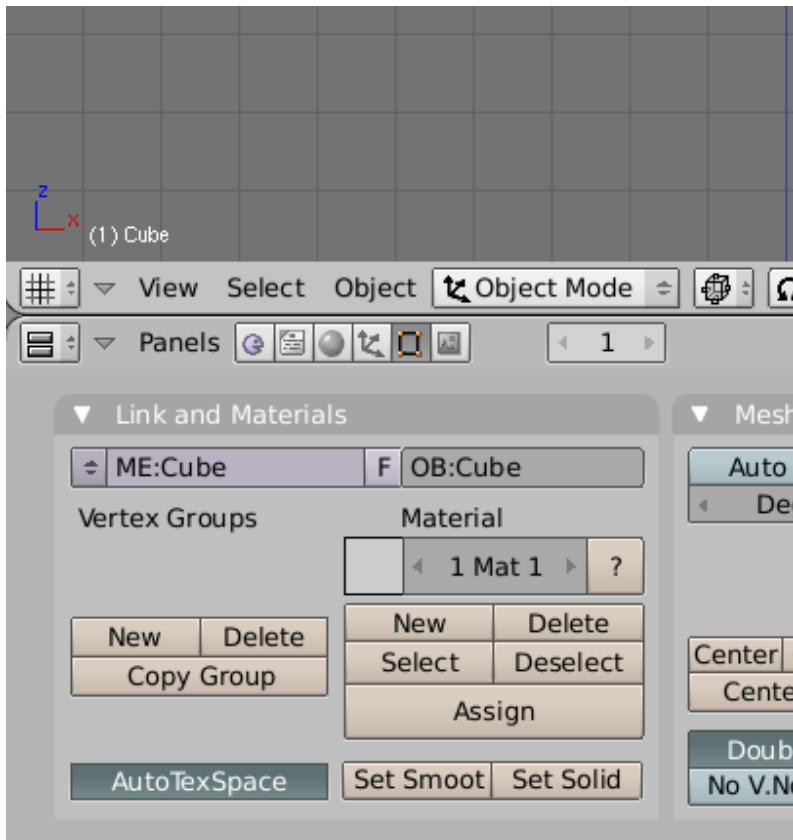
To change the roof leaving the height the same: in the Front view (lower left) window press **AKEY** to deselect everything and then box select the roof. In the upper left window (Top view) press **SKEY** and **XKEY** and scale to 1.1; press **SKEY** again but with **YKEY** this time and scale to 1.1 again. [User Comment: you can even press **SHIFT+Z** to restrict scaling to X and Y axis without affecting roof height, instead of scaling separately X and Y. **SHIFT+Axis** also works the same way with grabbing]



It is helpful to change the names from the default names to ones which describe the objects: for instance, you could change the name "Cube" to "House" instead and "Cube.001" to "Roof".

However, there is an object called "Cube" and a mesh called Cube which is linked to it so there are two names to change. There is also a material linked as well.

To change the name of the house object, go into object mode (**TAB** if not already there), select the original cube, select the Editing panel from the Button window header (**F9** or click on the icon) and, in the panel "Link and Materials", you will see



"ME:Cube" and "OB:Cube". Click on "Cube" and it will change to a red background, type in your new name, for instance "House" and press **ENTER**. Do this for both.

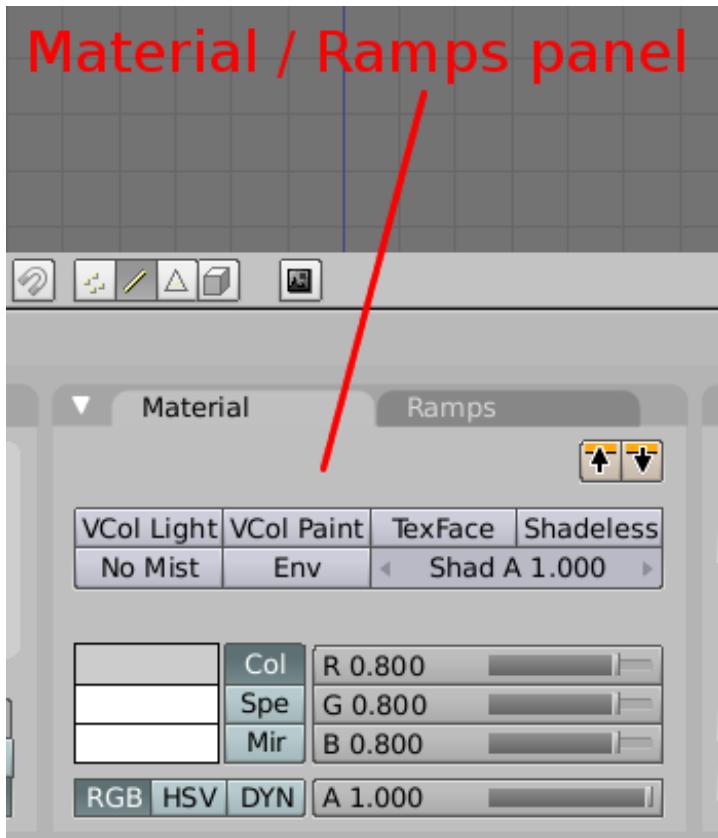
Now select the roof cube in the 3D window and go through the same procedure renaming it from "Cube.001" to "Roof".

The roof now needs to be given a different colour to the default grey. Select the Shading panel from the buttons



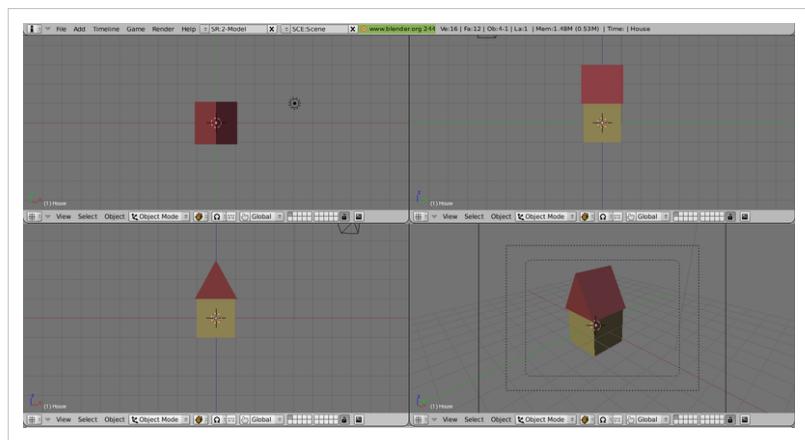
window (**F5** or click on the icon)

In the "Links and Pipeline" panel, click on "MA:Material" in the box under the "Link to Object" title. [*User Comment: In v2.45 you will not see any MA:Material box because the program does not assign any material to a new object. You will first have to click the Add New button to display the material info.*] "Material" will appear on a red background, rename it by typing "RoofRed" and pressing **ENTER**. You can now go to the second panel with two tabs "Material" and "Ramps" ("Material" should be showing, otherwise click on the Material tab to get it to show).



Now go to the second group of buttons and the line with "Col" in the second column and click on the grey bar to its left. This will bring up a colour selector, select a suitable red and press **ENTER**: lo and behold, the whole house turns red. If nothing happens you are probably in wireframe mode: press **ZKEY** in all four windows to change to solid mode.

To change the house colour to something more appropriate, go back to the lower left window and select the house cube: in the "Links and Pipeline" panel the object will change to "House". At present, both cubes share the same material, so, to colour the house cube differently, a new one has to be created. Click on the up-down arrow to the left of "MA:RoofRed" in the "Links and Pipeline" and select "ADD NEW". The text to the right of this arrow now changes to "MA:RoofRed.001". Click on "RoofRed.001" as before and change it to "HouseSandy". Now click on the bar coloured red next to the word "Col" in the "Materials" panel and you will get a colour selector, choose a suitable colour for the house, press **ENTER** and the house cube will now take the colour chosen.

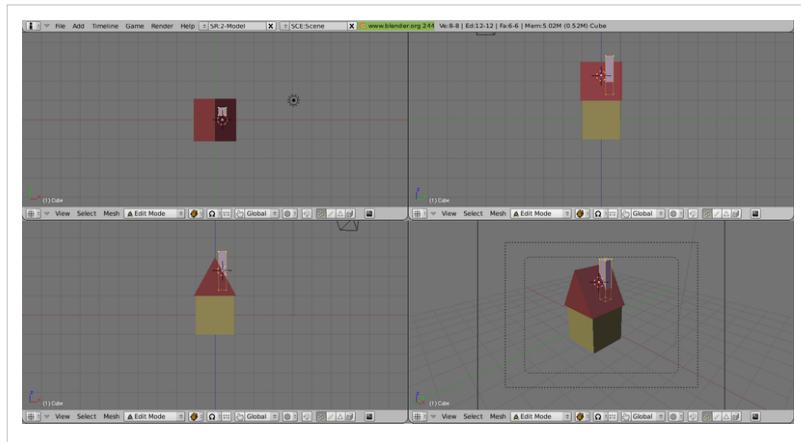


Note: I am no good with colours so I cheated: there is a file `rgb.txt` in the X Window System which contains about 750 colours represented as R, G, B triples. If you can access this file, you can convert the RGB triple into hexadecimal and enter it into the box called "Hex:" at the top right of the colour selector. I chose "eedd82" LightGoldenrod for the house and "cd5c5c" IndianRed for the roof.

## Adding a chimney

Go into Object mode (**TAB**) and in Front view (lower left window), click **LMB** on the righthand side of the roof, this will select the spot where the chimney will be created. Press **SPACE** and select Add > Mesh > Cube from the popup menu — a bit large, isn't it.

Scale it along the X-axis (**SKEY,XKEY**) to about 0.2 and along the Y-axis (**SKEY, YKEY**) to the same amount. Now grab it (**GKEY,XKEY**) and move it into position looking at the front view (lower left window).



When this is correct, move it into position along the Y axis (**GKEY, YKEY**) looking at the top view (upper left window). You can check the side view (upper right window) and the camera view (lower right window) to see that it looks OK.

You will see that the "Links and Pipeline" panel in the buttons window has no material selected and is showing "Add New". You can now choose to have the chimney a different colour to the roof or the house or reuse one of these colours.

If you want a new colour, click **LMB** on the "Add New" button and select it as you did for the house and the roof. Rename the material.

If you want to reuse the same colour, you can click **LMB** on the up-down arrow to the left of MA:Material and select from the choices "RoofRed" and "HouseSandy".

You can also change the name of the chimney object and the mesh linked to it to "Chimney".

## Adding a Window and a Door

To add a window: go into Object mode, move the mouse into the Front view window, click **LMB** to place the 3D cursor where you want the window to go and add a plane (**SPACE** Add > Mesh > Plane). Scale it down to 0.4 and then again by about 0.8 along the X-axis.

**Important:** After switching to Object mode, make sure nothing is selected (press the **AKEY** if necessary). If any object is selected before adding the new plane, the new plane will be part of that object causing problems later when you try to move the window plane.

Move it into position (**GKEY**) on the lefthand side of the house a little higher than midway. Move the mouse to the Side View window and move the plane along the Y-axis (**GKEY, YKEY**) until it is just in front of the house — you may want to zoom in (**SCROLL-MMB**) to see better.

Now, still in Side View, extrude the plane towards the house (**EKEY**, "Region", -0.07) until it is embedded in the wall.

To make this box into a window frame will require a larger view: move the mouse into the Side View window and press **NUM/** and then **CTRL-UPARROW**, you will now see only the window.

Deselect all (**AKEY**) and select the outer (leftmost) vertices using box select (**BKEY**), press **EKEY**, select "Region" and immediately (without moving the mouse) press **ESC**.

Scale the newly created vertices (it will be along the Z axis) by 0.9 (**SKEY**, 0.9, **ENTER**). Extrude again inwards to most of the depth of the box (**EKEY**, "Region", -.06, **ENTER**).

Now press **CTRL-UPARROW** and **NUM/** to get back to your original four views.

To change the name of the object and the mesh to "WindowFrame": go into Object mode, select the Editing panel (**F9**) and, in "Links and Materials" change "ME:Plane" and "OB:Plane" to "ME:WindowFrame" and "OB:WindowFrame". Under "0 Mat 0" click on the "New" button to assign a material to the window frame: when this changes to "1 Mat 1", click on the grey bar to the left of it and select white from the Colour Selector which appears.

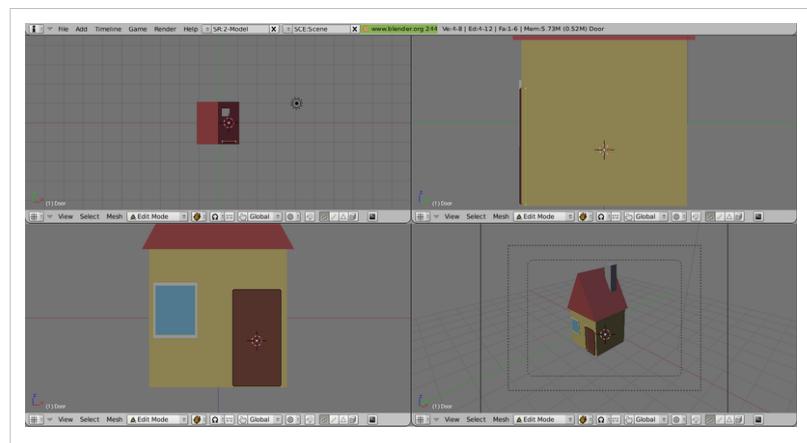
Select the shading panel (**F5**) and change the name of the material ("MA:Material") to "WindowFrame".

To make the window glass, deselect all (**AKEY**), create another plane, scale it to the dimensions of the window frame and move it in front of the frame. In Side view, move it to the front of the window frame.

Go to front view again, go to full screen (**CTRL-UPARROW**) and adjust the position and size so it fits within the frame.

Go back into Object mode (creating a new object automatically puts you into Edit mode) and change the mesh and object names to WindowGlass. Add a new material for the glass and colour it light blue (I chose "87ceeb" SkyBlue).

Add the door using the same method: in Object mode, deselect all, create a plane, scale the plane down by .7 and then again by .5 along the X axis. Move it into position using Front view and then move it against the wall of the house in Side view. Change the Object name and the mesh name to "Door", create a new material for it, change the material's name to "Door" and chose an appropriate colour for it (I chose "a0522d" sienna).



## Building the Fence

The way to build the fence is to create a paling and duplicate this as often as necessary.

The upright is made from a plane, scaled to 0.4 and then 0.1 along the X axis in Front view. To give it some thickness, go into Side view and extrude it to 0.05. The pointed top is made by going back into Front view, selecting the top vertices and extruding them by .07 (along the Z axis is selected automatically), scaling them to 0 along the X axis and then removing the doubled vertices (**WKEY**, "Remove Doubles": it should say "Removed 2 vertices").

It would be much simpler to enter face select mode, then extrude the top face by .07 on the Z axis. Then enter vertex select mode and select the top two vertices of the front face and merge them at the center. Repeat for the back face's top two vertices.

Also, if you want a more advanced method for building this model (like I did), go to Advanced Build of a House.

--Mast3rlinx (talk) 14:55, 12 February 2010 (UTC)

(If the pointed end extrusion is done before the paling is given thickness, the top doesn't move with the duplicated paling).

Change the name of the object and the mesh to "Paling", create a new material for it called "Paling" and make the colour white.

In Top view move the paling to the intersection of two grid lines a suitable distance from the house.

In Front view again, check that the bottom of the paling is level with the bottom of the house.

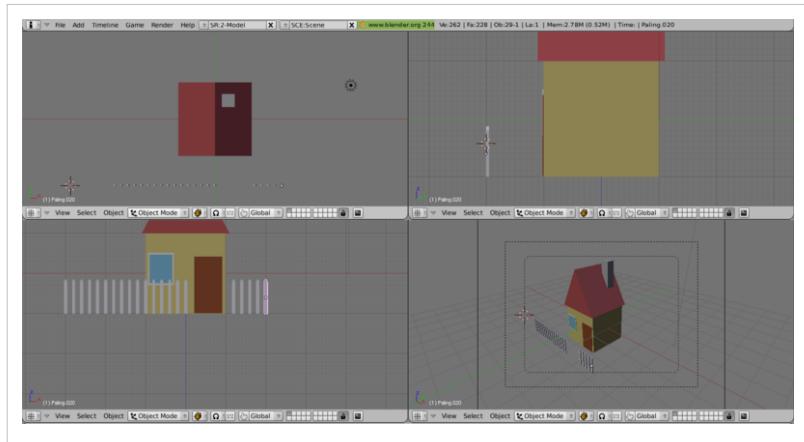
Duplicate the paling: Bart suggests using **ALT-DKEY** instead of **SHIFT-DKEY** as it creates a new instance of the paling — if you wish to make a change to the paling, you just need to change one and all the other instances will change too.

Starting in Front view, the flat face of the paling is along the X axis and it is easier to create the front fence first.

Change to Object mode, select the paling and press **ALT-DKEY** and "0.2", **ENTER** to move the duplicate into position along the X axis 0.2 units away from the first one.

Continue until you have the lefthand half of the front fence complete, there should be a gap between the lefthand and righthand halves for a path up to the front door. You can make this by increasing the gap at this point to 1.4 — this number will vary according to where the palings either side of the front door are placed.

**Noob note:** You will go bananas doing this. Better, scale the first pale so that it is big enough, then duplicate it in turn, pressing **CTRL** to snap the successive pales to grid (first pale has to be big enough so that grid distance = pale distance). When you are done you scale the fence down to fit the size of the house.



The fence needs horizontal bars to hold the palings in place: in Front view click below the middle of the lefthand fence, create a plane and scale it to the length of the fence. Scale it again to 0.05 along the Z axis and move it into position — if you keep **CTRL** pressed while you move it, it will snap to the grid. In Side view move it so it is against the back of the fence and extrude it by -0.04 along the Y axis. Go into Object mode, duplicate it and move it (keeping **CTRL** pressed) along the Z axis (**ZKEY**) to make the bottom bar.

To make the bars on the righthand part of the fence: in Front view duplicate the lefthand top bar and move it along the X axis (**XKEY**) to the top of the righthand fence. Scale it to the width of the righthand fence. Duplicate this top bar and move it down (**ZKEY**) to become the bottom bar of the righthand fence.

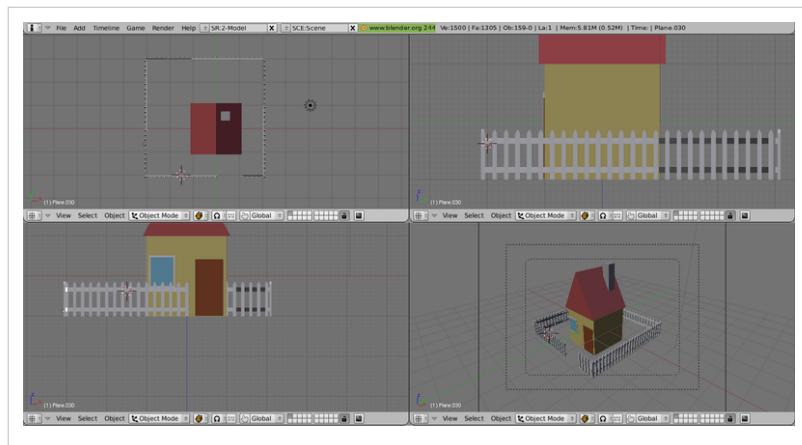
It is a bit tedious creating each paling individually so, now that you have a few ready made, you can just copy them to make the fence at the back of the house.

In Top view, deselect all (**AKEY**), select the whole front fence, including the gap, duplicate it (**ALT-DKEY** again) and move it to the back.

Go into Back view (**CTRL-NUM1**), select one of the palings on one or other side of the gap and duplicate that to close it. Select one or other of the horizontal bars (in Top view, it is going to be the upper one of the pair), duplicate it (**ALT-DKEY**), move it to the new palings and scale it to fit. Duplicate this bar and move it into position as the lower bar.

The back fence has the horizontal bars on the outside so must be rotated: in Top view select the whole of the back fence, press **RKEY** and type in 180 followed by **ENTER**. Move the fence back into position with **CTRL** pressed to snap it to the grid.

The back fence is already selected from the rotation, duplicate it (**ALT-DKEY**), press **ENTER** as it is not going to be moved and press **RKEY** to rotate it and type in 90 followed by **ENTER** press **GKEY** to move it into position as the righthand fence and **ENTER** when it is there. Duplicate it, rotate it and move the copy to the other side of the house.



Note:I found it easier to make the upright by subdividing the plane(after it was scaled down) and then moving the middle three verticies near the top. Then i merged the top three verticies and merged them at the center. I found this way much less confusing and easier to do.

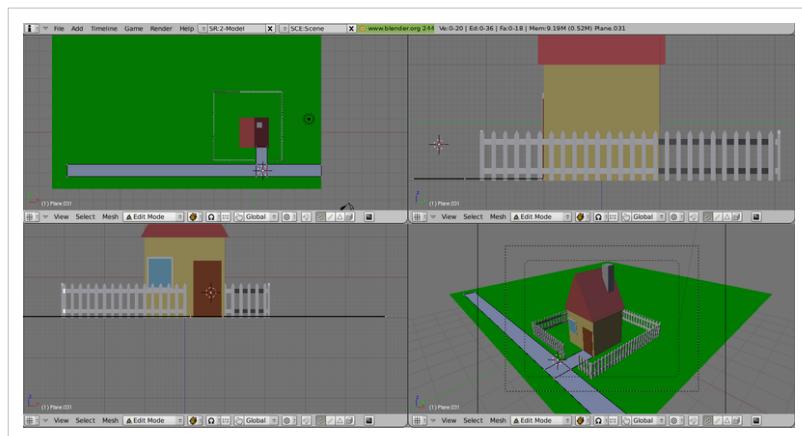
Note: I made the fence out of a cube, extrude up once and merge the top vertices.

If you just stop where this part of the tutorial ends, then the fence doesn't look right. What I did was enter edit mode for the fence and deleted one of the intersecting palings at each corner, make the remaining corner post a square when viewed from the top, then merged the vertices that make the peak at the center to make a square post with a pyramidal top at each corner. This makes for a much more realistic fence without any advanced methods used.

--Mast3rlinx (talk) 16:52, 12 February 2010 (UTC)

## The Ground plane and a Path

In Top view put the 3D cursor somewhere near the middle of the plot, (click **LMB**) create a plane and, keeping an eye on the camera view (bottom righthand window) scale it to extend just past the boundaries of what the camera sees — 10.0 seems a good number. Go back into Object mode and change its (object, mesh) name to "Ground", create a new material for it, rename it "Grass" and select a suitable green from the colour selector ("00cd00" green3). Check in Front view that it is level with the bottom of the house and fence and move it (along the Z axis) if necessary.



For the path, go into Top view, create a plane, scale it to fit the gap in the front fence with some space on either side and position it just outside the fence. Check, in Front view, that it lines up with the door and is level with the bottom of the house. Extrude it along the Z axis by 0.01 to give it a little bit of thickness. Scale it along the X axis to the same width as the door.

In Top view, extrude the top edge of the plane to the door and then the left and right edges to a suitable distance to the left and right to make the path passing the house.

## Blender 3D: Noob to Pro/Pipe joints

*This tutorial was created with Blender v2.46*

How to model pipe joints.

### T joint

- Add a cylinder object (*Add->Mesh->Cylinder*)
  - 16 Vertices
  - No Caps
  - Depth 4
- Switch to front view (*View->Front*).

The X axis now points to the right. We will work along the X axis from now on.

- Change to Edit mode.
- Create a Loop Cut in the middle of the cylinder (**Ctrl-R**).

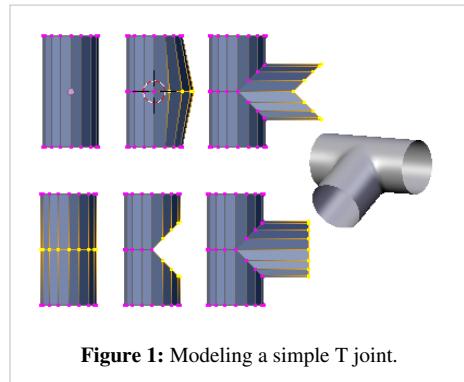


Figure 1: Modeling a simple T joint.

Make sure that the 3D cursor is placed in the middle of the cylinder.

- *Pivot-Point->Cursor*
- Select the seven vertices at the front.
- Scale with constraint Y axis (**S->Shift-Y**), about 1.4142 (square root of 2)
- Rip the mesh with **V**. Rotate by 45 degrees.
- Rotate the other side by 45 degrees also.
- Select the vertices of the gap and extrude them.
- *Pivot Point->Median*.
- To flatten the front: scale at the X axis by factor 0. (**S-X, 0**).
- Use the *Subsurf* modifier in combination with *Set Smooth* and use *LoopCuts* to get a nice transition.

### 6 cylinder joint

- Add a cube.
- *Subdivide* the cube once (1).
- Select the 12 vertices in the middle of the cube's edges.
- Scale by factor 1.4142 (press **S** key and type in the factor with the keypad) (2).
- Select all vertices. *Extrude->Individual Faces* (3).
- Hide the now selected vertices (**H**) (4).
- Select the original vertices in the middle of the faces and delete them (5).
- Unhide the vertices (**Alt-H**) (6).
- Select all vertices.
- *Remove Doubles*.

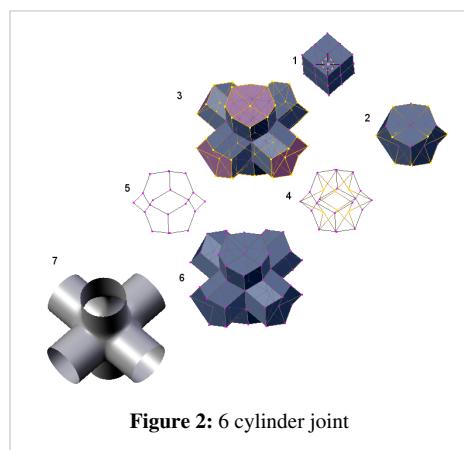


Figure 2: 6 cylinder joint

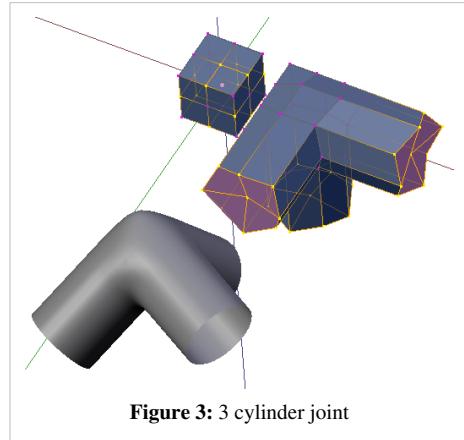
Clean your mesh as described above.

### 3 cylinder joint

This is very similar to above, only the selection for scaling is different.

- Select the middle vertices at the edges where the cylinders shall come from. Additionally select the middle vertices at the faces that shall be kept free. You have to select 12 vertices.
- Scale your selection by 1.4142.
- Select the faces that shall be extruded.
- *Extrude->Individual Faces*
- *Remove Doubles*.
- Delete the inner vertices.

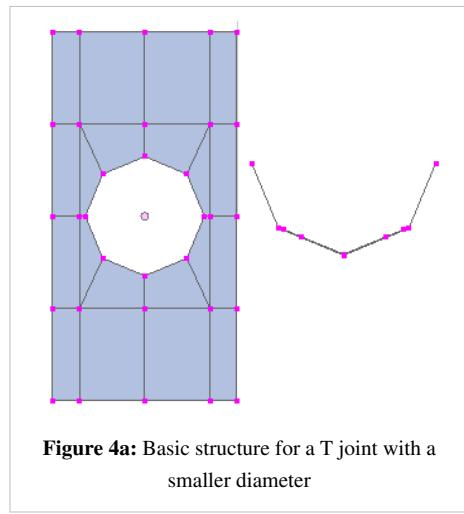
Clean your mesh as described above.



### T joint with smaller diameter

We're going to create a structure like in **Fig. 4a** and extrude the inner circle. To do that we place a circle with *Retopo* (Manual<sup>[1]</sup>) on the cylinder and join both objects afterwards.

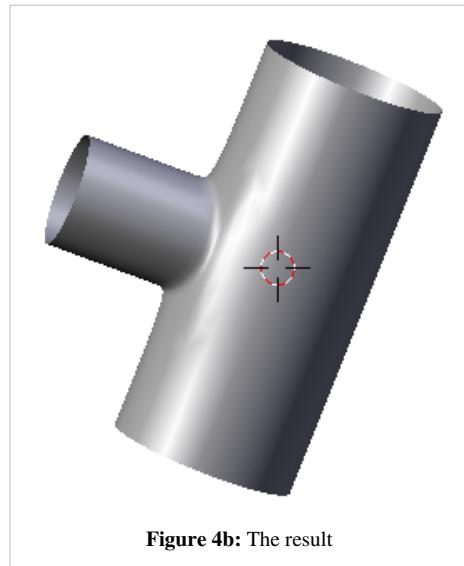
- *Add->Mesh->Cylinder*
  - 8 vertices (of course you can use more vertices if you like)
  - **(Noob note:** More vertices in your cylinder will mean more intuitive work and figuring [more loop cuts, face making, etc.] later on when retopo-ing. I suggest following the tutorial first, then try to do a more advanced model with more faces / vertices when you understand the tool. If you have trouble [as I did] getting really nice geometry, do a search for some tutorial videos<sup>[2]</sup>. There are many really helpful ones.)



- *No Caps*
- *Depth 4*
- *Add->Mesh->Circle* (**Noob note:** Make sure you create these as separate objects.)
  - 8 vertices
  - *No fill*
- Rotate the circle by 90° at the Y axis (**R->Y->90**).
- Move the circle in front of the cylinder(**G-->X->2**).
- Switch to side view (**Num-3**).
- Scale the circle to the diameter of the smaller cylinder.

(**Noob note:** I felt this was a little unclear. What the author means is, scale the circle down until it is the size of the diameter of the smaller cylinder you are going to create from it later by extrusion. You would

be wise to scale it so it fits (straight backwards from view) inside the two closest faces of the cylinder as per **Fig. 4a** if you wish to follow the tutorial. )



- Switch to edit mode, select all vertices of the circle.

(**Noob note:** To recap: you should be in edit mode for the circle, with the all of it selected, with the cylinder object directly behind it in from your viewpoint in orthographic mode.) Make sure that you don't work in Wireframe view.

- Click on *Retopo->Retopo All* in the *Mesh* panel of the *Editing* buttons.

(**Noob note:** What *Retopo* does [in this case]; is project the selected topography of an object in edit mode directly backwards from your selected view point onto the topography of another [seperate] object. So you must be lined up view-wise for this to work. I've also found sometimes I must press the **Enter** key for the **Retopo All** to take effect even if I've selected it with the **LMB** .)

If the vertices of the circle are not adjacent to the cylinder but have jumped to wrong places undo the last step (**Ctrl-Z**), deselect and reselect all vertices and try again. This does help (strangely). *Retopo* works with limited accuracy, if the vertices don't fit perfectly you have to move them by hand.

- Change to object mode, select additionally the cylinder and join both objects (**Ctrl-J**).

Now you work best in *Wireframe* view.

- Add three loop cuts (**Ctrl-R->3**)
- Delete the middle vertex.
- Connect the free vertices (**F**).
- Select the circle.
- Extrude and clean up.

## References

- [1] [http://wiki.blender.org/index.php/Doc:Manual/Modelling/Meshes/Remaking\\_the\\_Tolopogy](http://wiki.blender.org/index.php/Doc:Manual/Modelling/Meshes/Remaking_the_Tolopogy)
- [2] [http://video.google.com/videosearch?sourceid=navclient&rlz=1T4GGLR\\_enUS262US264&q=blender%20retopo%20tutorial&um=1&ie=UTF-8&sa=N&hl=en&tab=wv#](http://video.google.com/videosearch?sourceid=navclient&rlz=1T4GGLR_enUS262US264&q=blender%20retopo%20tutorial&um=1&ie=UTF-8&sa=N&hl=en&tab=wv#)

# Blender 3D: Noob to Pro/Creating Models With Photo Assistance

---

The first tutorial is about using guide images to place vertices in their proper places in 3D space. The second tutorial is on how to take good reference pictures. These tutorials assume that you have completed all previous tutorials.

## Making A Pyramid

*Note: This section may be incredibly frustrating to new Blender users. If you are starting out it is recommended you go over the earlier modeling tutorials first before going through this section.*

First we are going to create a pyramid the easy way. Then we are going to show how to use different viewpoints and images as a guide to place vertices correctly in 3D space.

You should have the default cube, if not, press **CTRL+XKEY** to start a new project.

- Go into Edit Mode (if in Object Mode, press **TAB**).
- Select side (right) view (**NUM3**).
- Make sure no vertices are selected by pressing **AKEY**.
- Ensure the "Limit selection to visible" ("Occlude background geometry" in Blender 2.46 and later) icon is disabled so we can select all four of the top vertices, even the back two which aren't visible.
- Go to Box Select mode (**BKEY**) draw a box around the top vertices of the box
- Merge vertices by pressing **ALT+MKEY** → *At center* and the top of the box is merged to a single point
- Unselect all vertices (**AKEY**).
- Box select (**BKEY**) the bottom 4 vertices
- Hollow out the bottom of the pyramid by deleting the bottom face **XKEY** → *faces* (remember that if you want to change to face select mode press **CTRL + TAB** → *faces*)

**note:** with versions 2.46 of Blender and later, you just have to select merge at center after the merge vertices step to create a pyramid, the last 3 steps aren't needed.

**note:** you can do this easier by making the top vertices closer to each other (**SKEY**)

and then use Remove Double (in editing buttons/Mesh tools tab)

Now that we have the pyramid the easy way, let's learn how to use photos as references to build models. First, unselect all the vertices by pressing **AKEY**. Next, select the bottom four vertices of the pyramid and delete them with **DEL** or **XKEY**. The only vertex left will be the vertex which makes the tip of the pyramid. This will be used later.

note: in Blender 2.48 you can go to edit mode face mode select top face and size it down.

## Window Layout

To make things easier, open the user preferences window and in the "View & Controls" sub-menu, turn on the "View Name" option. The window names will be used for reference inside the rest of this tutorial.

Split the Main 3D view window in to 4 windows.

**Reminder:** to split windows, move the mouse to the border of the view, when the cursor transforms into arrow, right-click and choose "Split Area". (Explained in the guide: Noob to Pro/Blender Windowing System.)

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

NUM1	NUM7
NUM3	NUM0

And if you click on View, you can see that these windows are respectively:

<i>Front</i>	<i>Top</i>
<i>Side (Right)</i>	<i>Camera</i>

By knowing which view you are looking at you may find that you quickly get the idea of what you are doing and can proceed somewhat intuitively in this section on your own without following all of the step by step instructions.

## Reference Pictures

Make a picture of a white square and of a white triangle in the GIMP, Paint :) , or some other image editor.

**Read the following section carefully:** Make sure that the *drawing* of the square is *square* and not just rectangular. Make the triangle the same width and height as the square. Make sure the apex of the triangle is directly above the midpoint of its baseline.

*Suggested method of construction#1:* Make the square. Save it, but keep it open. Delete everything except the bottom line. Make the triangle from the lower corners (ends of the baseline) and make sure the two new lines meet exactly above the midpoint of the baseline. Make sure the triangle is visible against a black background as shown in the image. Save as a new file.

*Suggested method of construction#2:* Download the black and white triangle image on the right of the screen and use that, make a square by filling it white and resaving it.

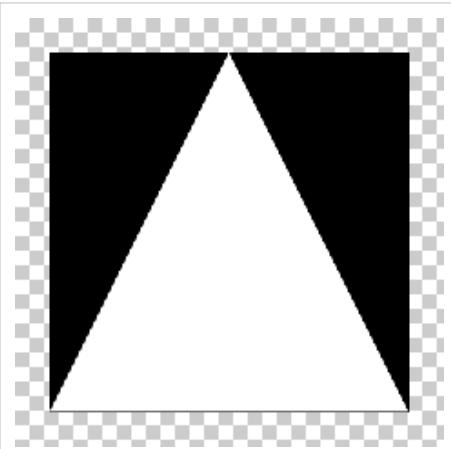
*Suggested method of construction#3:* (if you are using Photoshop): make a square selection of "n by n" size, remember the value of "n". Fill it with white color and save. To create a triangle of needed properties make a rectangular selection of same (n by n) size, on a new layer, click **RMB** on your document, choose "Transform selection" option in the pop-up menu. Once you are in "Transform selection" mode, right-click the blank image again. This time the pop-up menu would be different. Choose "Perspective" from it, and with **LMB** drag one of the two top vertices toward the other. Once the vertices meet (in the top-center of the image), exit the transformation mode, and fill the resulting triangular selection with white.

Note for Gimp users: turn on the grid (View->Show Grid, View->Snap to Grid), use the rectangle select with a fixed aspect ratio of 1:1 (in the tool options panel) to select a square that you can flood fill. For the triangle, use the node tool to draw a triangular path, convert to selection (Select->From Path) and fill it. Or you could just use Inkscape...

Save the files to a place that is easy to access. Blender only supports the TGA, PNG, and JPG image formats.



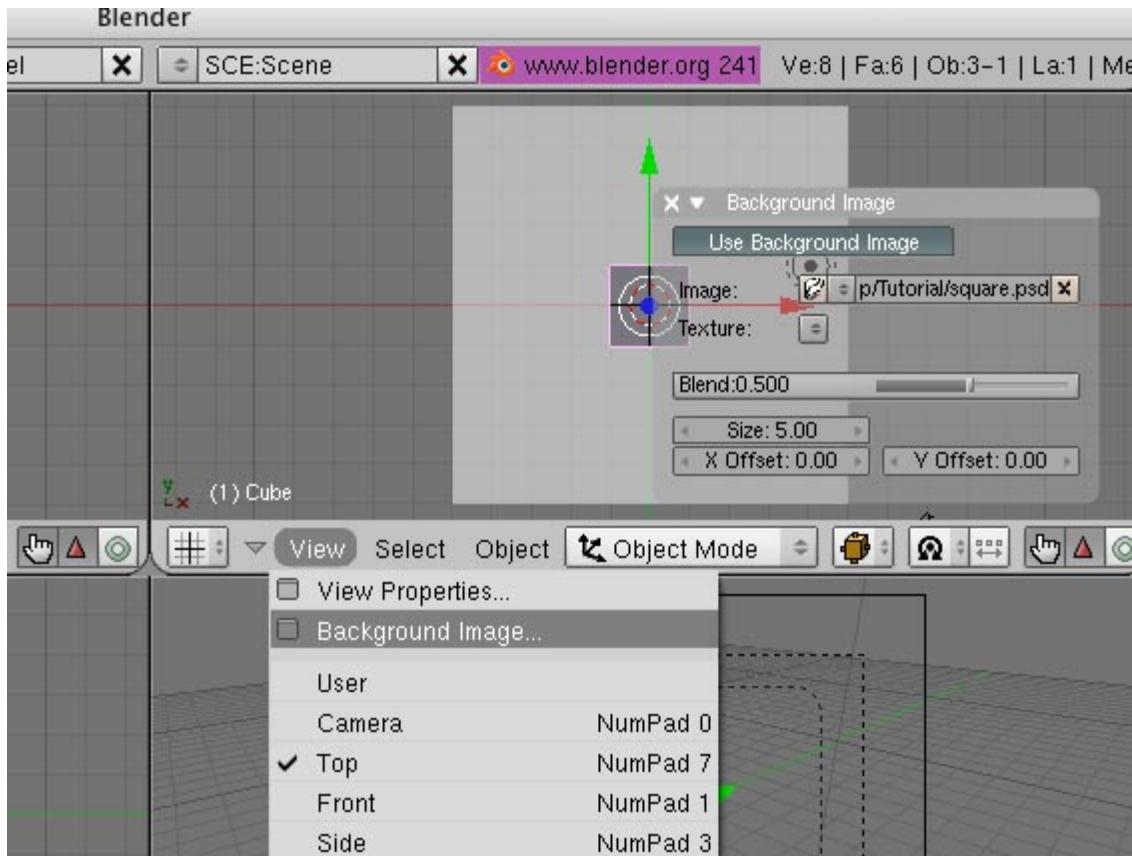
The square and triangle relationship



Save the triangle with a black background

## Background Images

Load the white square into the **top (NUM7)** 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 **front (NUM1)** and **side (right or NUM3)** windows.

If necessary, zoom out so that you can see the whole picture.

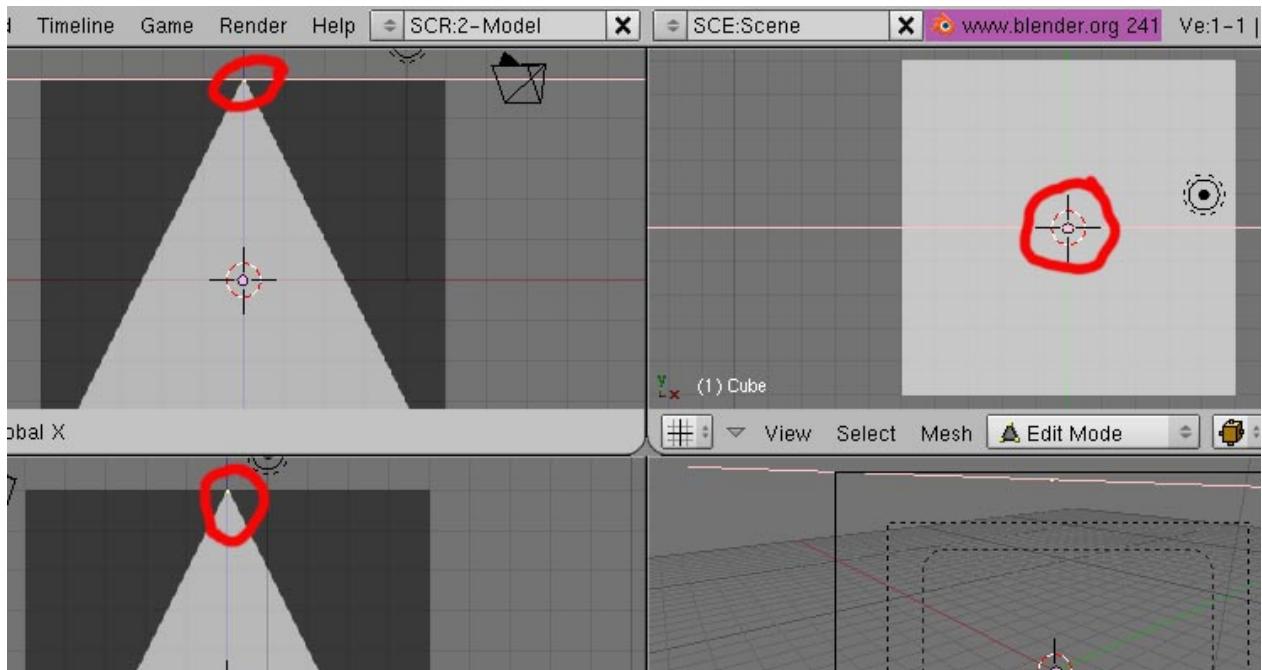
Now you have a guide for making a pyramid.

If you can not see the picture, switch to Ortho view by hitting **NUM5**

## Side One

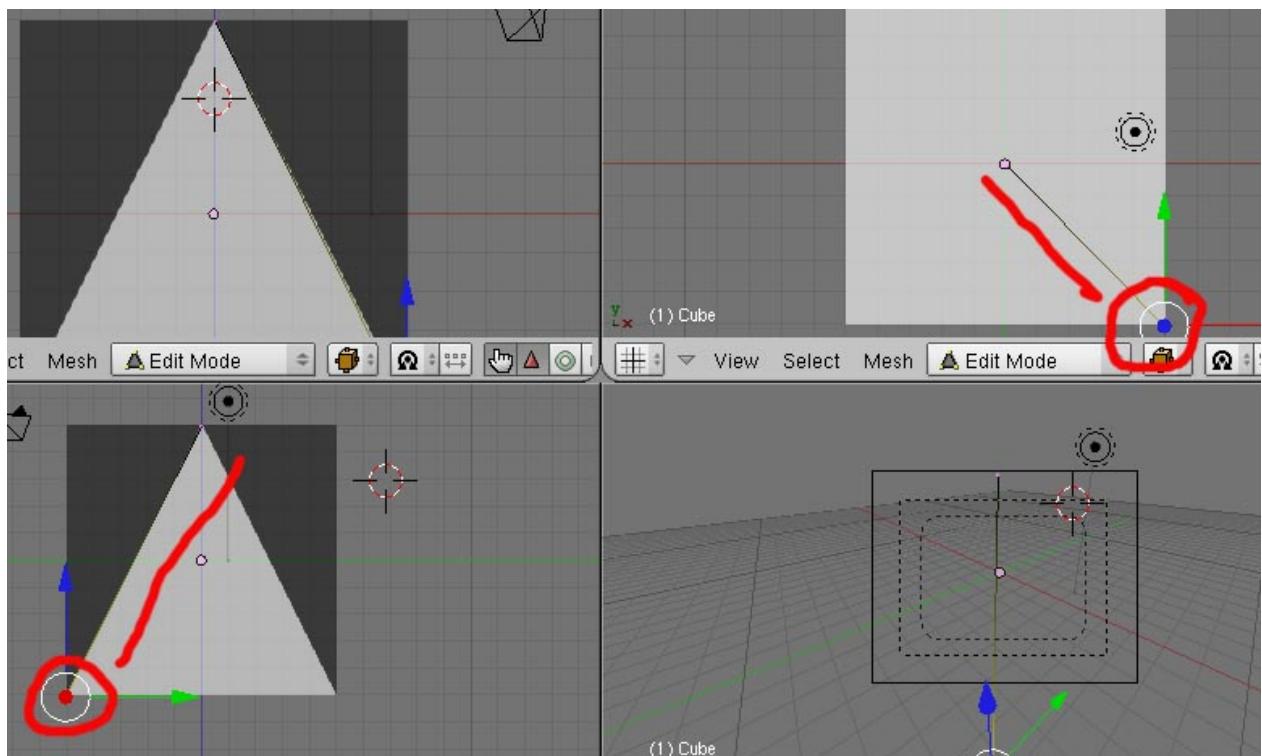
The vertex that is left will be the topmost point of the pyramid. Use the **GKEY** to move the vertex around. To get it in the right spot, line it up at the top most point in the **front (NUM1)** and **side (right or NUM3)** windows. If you look in the **top (NUM7)** window the vertex should appear to be in the center. Make sure to keep the vertex highlighted for the next step.

**Note:** If you delete all of your mesh objects in Object Mode (and remaining vertice), then you'll be stuck in Object Mode. You can however, add a new mesh object as a base, or an empty mesh via a python script (*Add* → *Mesh* → *Empty Mesh*) [Blender 2.49]. Once an empty mesh object has been created, select it (right click on it or **AKEY**), and then change to the Edit Mode. Now you can begin to create individual vertices with **CTRL + LMB**.



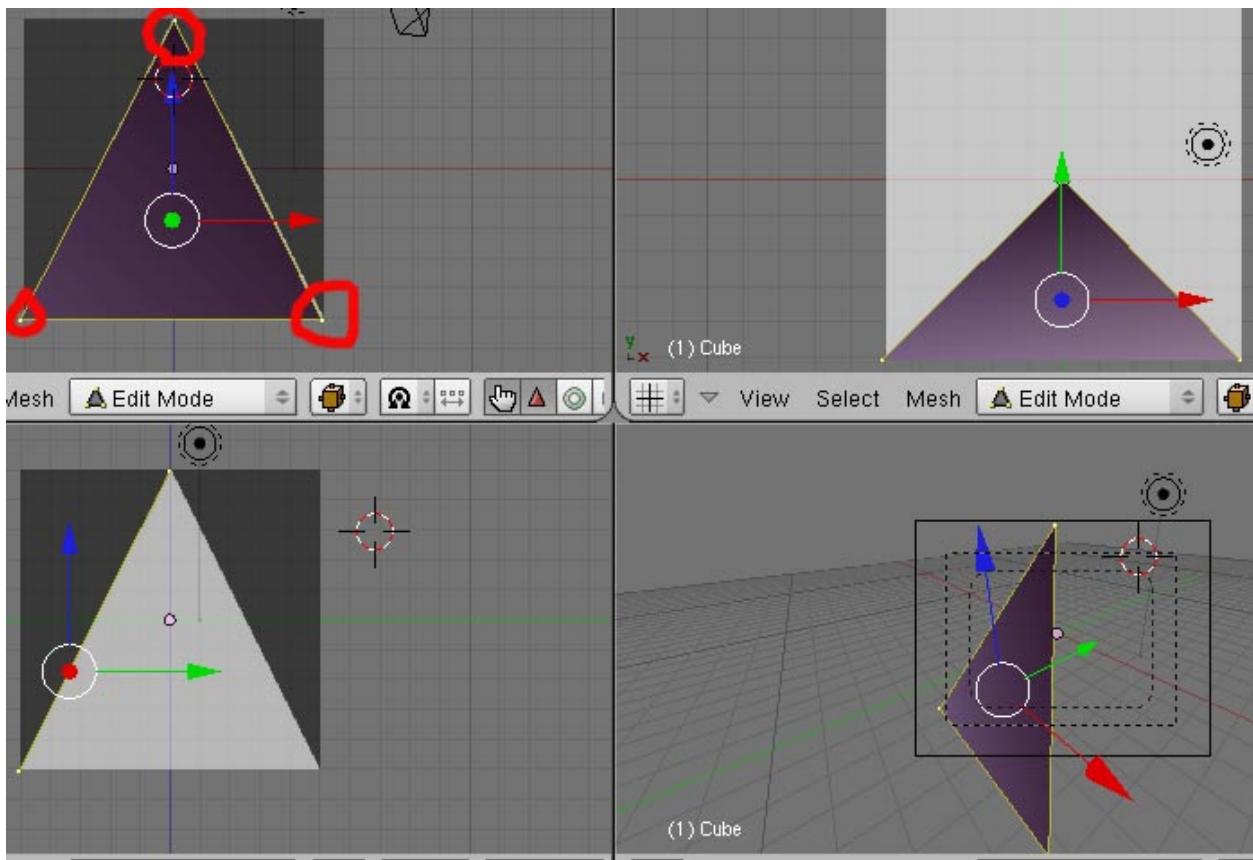
(note: it might be helpful at some point to zoom in and use the X,Y and Z movement restriction)

Now, we are going to place more vertices in the scene. Since we want to have the new vertices connected to the first one, we will make sure the first vertex is selected and hold down **CTRL** and click **LMB** to create a new one. In the **side (right or NUM3)** window, place a vertex on the lower left edge of the triangle by holding **CTRL** and clicking there. This should create a line between your 2 points. If needed, use **GKEY** to line it up in that window and also at the lower right point in the **top (NUM7)** window.



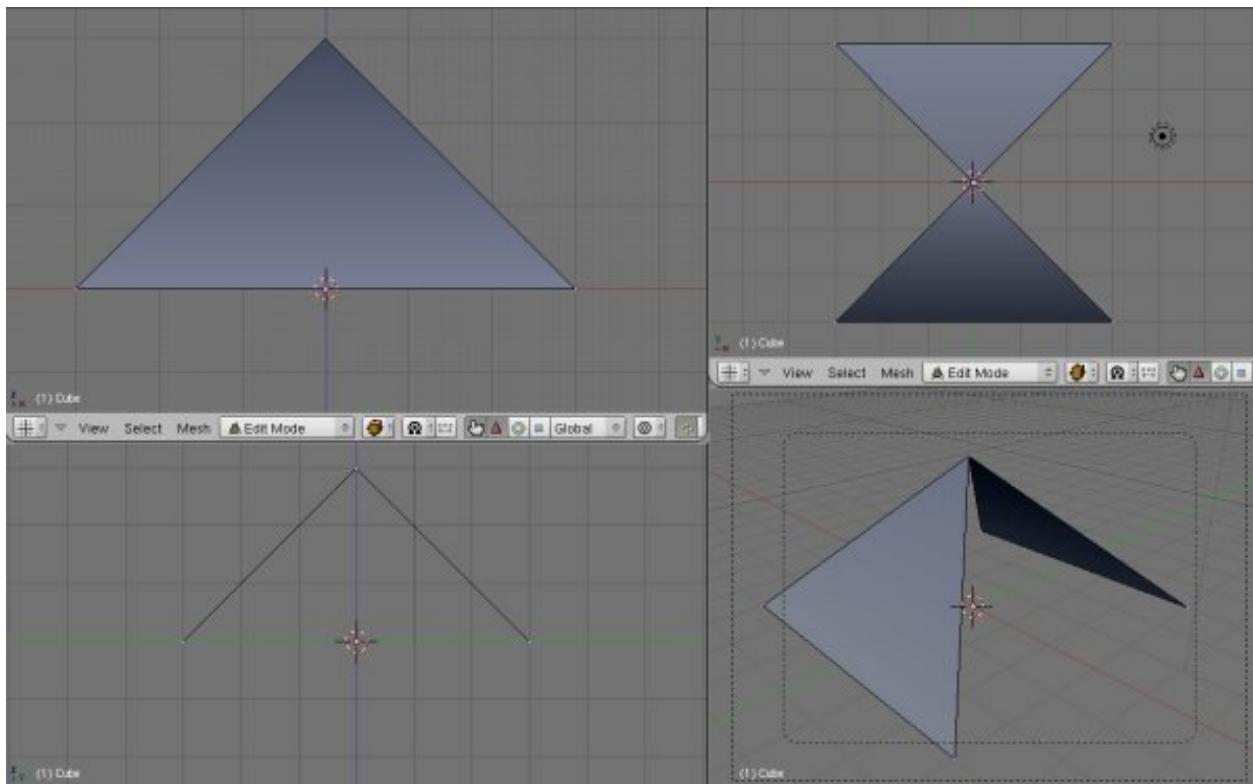
Next, with only one of the vertices selected, in the **front (NUM1)** window place a vertex on the lower left edge of the triangle using the same method. Line it up in that window and at the lower left point in the **top (NUM7)** window.

Now, select your 3 vertices (use **AKEY**) in the **front (NUM1)** window and press the **FKEY**. You should see a triangle appear.



## Side Two

Press **AKEY** to deselect all vertices and select the vertex at the top of the pyramid again. Repeat the process from Side One to make the next side. Place a vertex in each of the other corners of the square using **CTRL** and **LMB**. Line them up in two different views with the corners of the triangle and square. Make sure only the top vertex and the two new vertices are selected and press **FKEY**. This will fill in a face opposite of the first face in the pyramid.



## Sides Three and Four

To fill in the other two sides, select the top vertex and the two corner vertices on a side where there is no face yet. Again, use **FKEY** to fill in a face. Repeat this for the last side to have all four sides created.

## Bottom

In the **top** (NUM7) window, select all four corners and make a face. You should have a solid pyramid! Now, select all 4 faces, hit **XKEY**, and choose "faces".

## Taking the Best Reference Photos

Remember what I said about turning a cube into a puppy? We'll do that now.

### Step One: Get the pictures of the model

*Tip: The images here do not line up. Some need to be rotated and others do not match in size. They will be kept so that you can get the "real feel" for this project.*

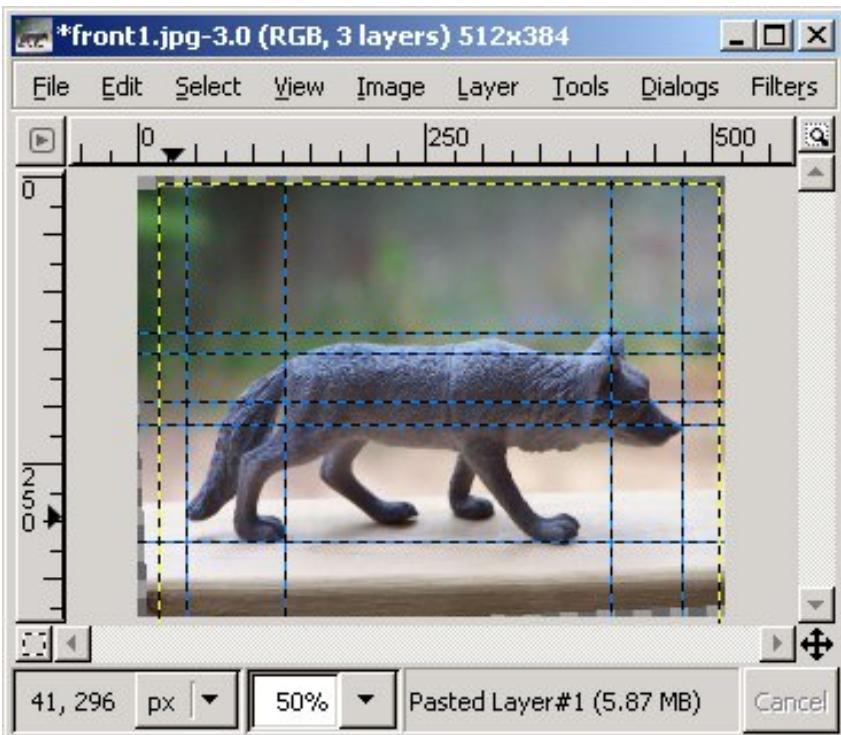
If you have a puppy and a digital camera, take three pictures of the cute little rascal and upload them. If you don't have a puppy, any object or small animal will do. Ideally, the photos will be looking straight down at the top of the puppy, a side view, and a front view. It's important that the puppy 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 (**front** \ NUM1), one of its reflection seen 90 degrees to the right (**side** \ **right** \ NUM3)), and one of its reflection seen from overhead (**top** \ NUM7). Take the photo from a long distance away with a zoom lens to get close to an orthographic projection <sup>[1]</sup>.

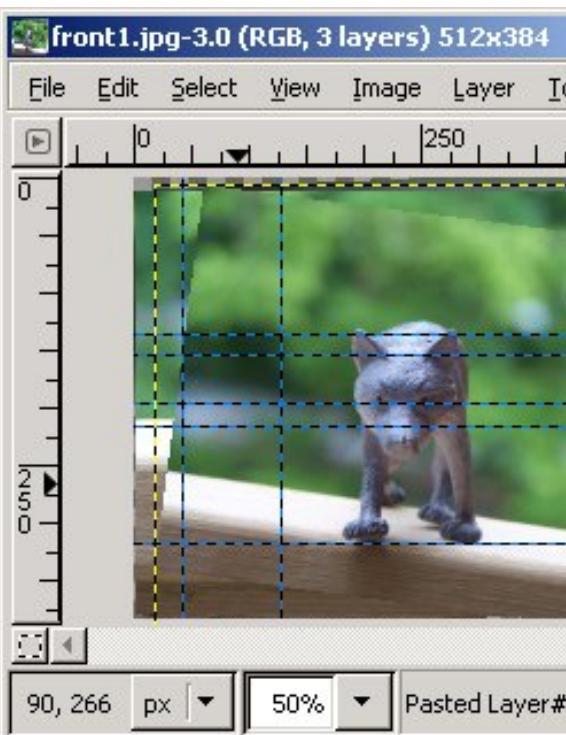
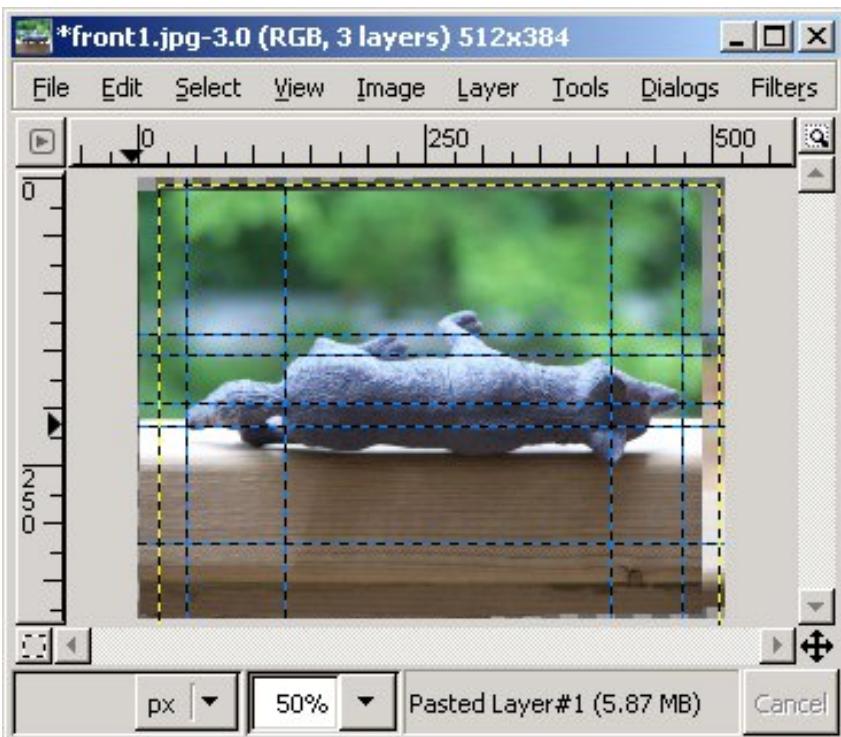
Or how about pictures of a toy wolf taken from 6 views points?:

- Left view <sup>[2]</sup>
- Right view <sup>[3]</sup>
- Front view <sup>[4]</sup>
- Back view <sup>[5]</sup>
- Bottom view <sup>[6]</sup>
- Top view <sup>[7]</sup>

Using your favorite image editor, such as PhotoShop or the GIMP, down-scale the images need to a reasonable size (I made mine 512x384), and then match them 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, the front of the back foot, eye level, tip of the ear, and the 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. You may notice the photos aren't perfect, but we'll use them just to show how you should deal with your real photos. When you are creating your own pictures to import, note parallax<sup>[8]</sup>. In this example, parallax is present, and we'll attempt to compensate.

*Reader note: I used Paint Shop Pro 7. I used layers and played with transparencies to be able to see where the views fitted best with each others and align them better.*

## Step Two: Get the Picture into Blender

Getting the image into blender is the easy part. The more difficult part will be creating the mesh, but first things first. Create a new file (*File → New*) to see the familiar default objects. Don't bother deleting the cube, we'll end up using it in the tutorial. Just as was done in the "Making A Pyramid" section, split the 3D Viewer into four views and assign them the following view ports. Note that the arrangement is different than that used in the previous tutorial.

NUM7	NUM3
NUM1	NUM0

Each window will show you different XYZ coordinates with the **NUM0** view being what your object will look like to the camera object.

Now that the screen is set up, let's load the images. In the top left viewport (XY), click *View* on the viewport's header (the menu at the bottom). From the menu provided, select *Background Image* .... A small window will appear.

Click the *Use Background Image* button and several more buttons will appear. Now click on the folder icon located to the right of *Image*:. A new full-viewport window will appear. Explore this window a bit and end up selecting the image file of the wolf from the top view. What you should get is the picture of the toy wolf from above with the default cube on top of it. Rotate your view of the cube. The picture disappeared! But don't worry - the picture is still there. Press **NUM7** to realign the viewport and see the picture again.

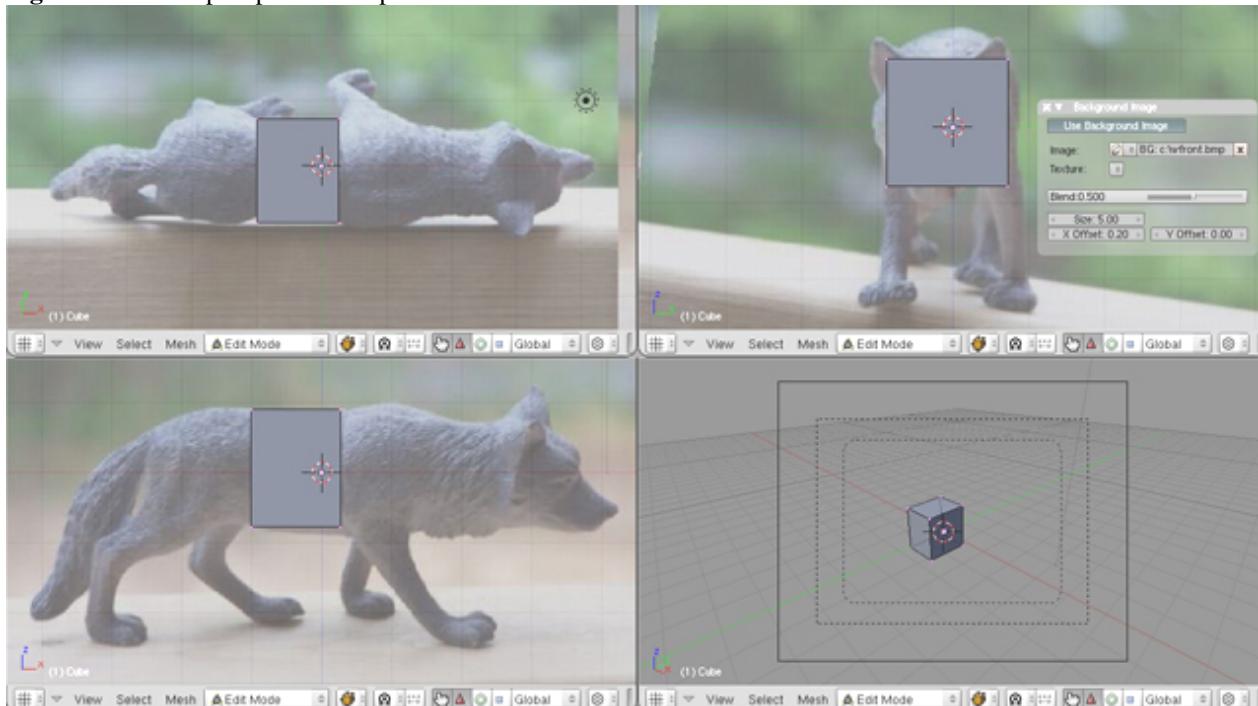
**Tip:** Be sure to be in orthographic view **Num 5** or else you won't see the picture.

Now load the front view of the wolf into the **NUM3** viewport (YZ) as you did the top view. Repeat the procedure and load the side view of the wolf into the **NUM1** viewport (XZ). As a note, each picture is specific to the viewport it was loaded into. When you switch between axes views, the picture will not change. Try this out by changing the top left viewport to **NUM3** and **NUM1**. Return the viewport to **NUM7** before continuing.

The pictures are now loaded into the Blender viewports. If you look at the pictures on the grid, you may notice that the front view of the wolf isn't quite center. That is okay, Blender has a way to fix it.

Move your cursor over the **NUM3** viewport and press **Shift + SpaceBar**, this expands the current viewport (NUM3/front view) to a fullscreen view, then open the Background Image window for the front view again. Notice that there are picture manipulation options available. One of these includes picture offsets. Click on right side of the 'X Offset: 0.00' to increase the offset to 0.20. The picture will be shifted over slightly so now the wolf is more centered. Press **Shift + SpaceBar** again to return to the four viewport view you created earlier.

**Figure 2.2.1** Viewport picture setup



The setup work is now done! Let's start on actually making the wolf model.

*Reader Note:* If you, like me, weren't paying that much care to how your "wolf front view pic" was aligned from left to right earlier on during the gimp editing phases, then you may need to use a different 'X Offset: ' value than '0.20' that the author recommends. If that is the case, then just use some appropriate alignment value to center it visually as best you can. Hopefully this helps with avoiding some potential confusion for other readers.

### Step Three: Rough Model Fitting

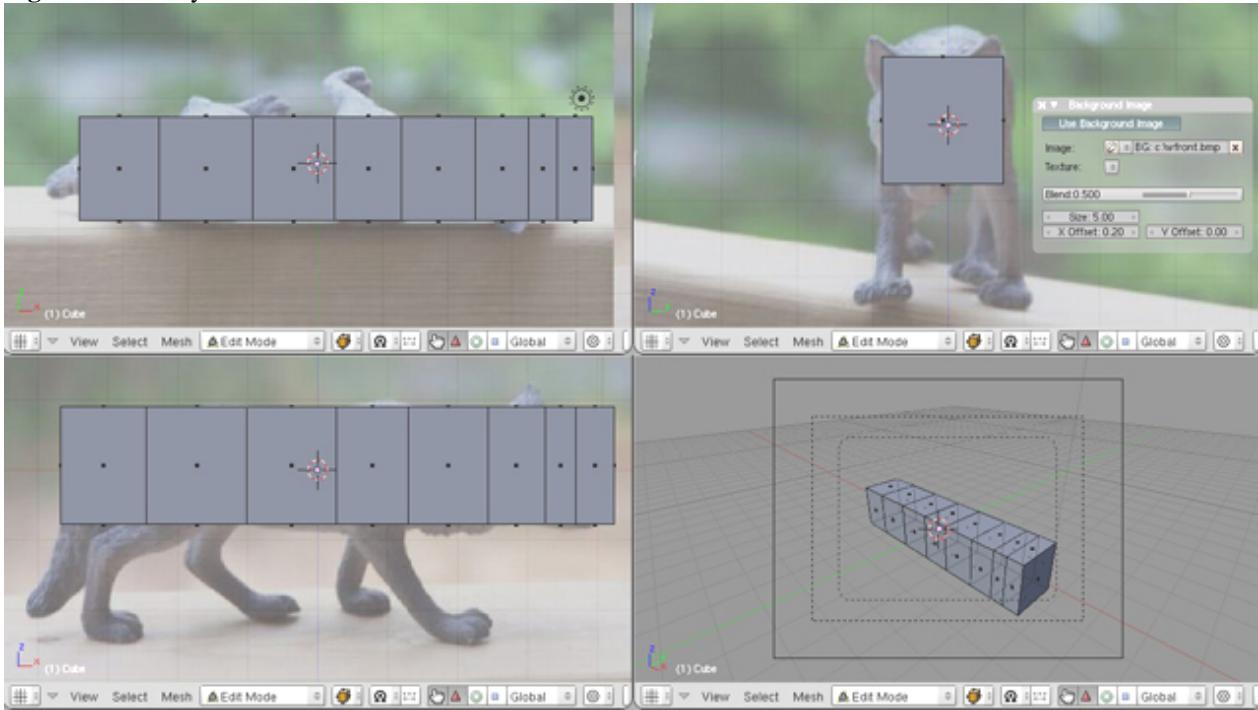
This is a brute force model creation using techniques discussed previously in this book. This section is meant to help you explore and become more comfortable with them. Do *not* try to follow the example to the tee. Your wolf and my wolf will probably not look the same since you may want to add more or have less detail.

[*Note:* If you are following this book all the way through, and are just getting started with blender, the following step (step 3 as a whole that is) may likely take several hours to complete. The best strategy is to take breaks, be patient, and with time you'll figure out the best way to go about this step.]

The rough fit stage requires either some planning or on-the-spot decisions. Think about where the wolf will have parts of its body flex or require parts jutting out.

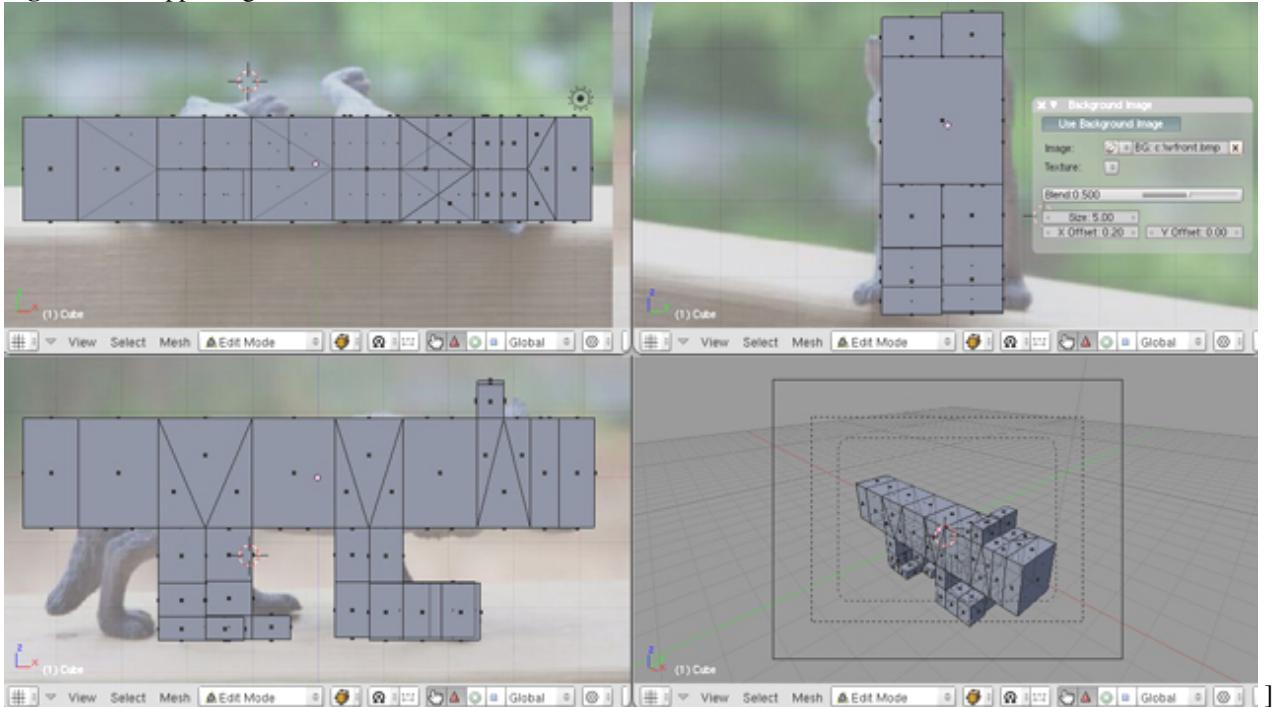
The first step is to create a blocky wolf. Start out with a column of blocks using the extrude face command (select face, **EKEY**). Don't worry about snapping the vertices to the grid since we are working with an organic figure.

**Figure 2.3.1** Body column formation



The next step is to split the ears and legs off of the body. Do this by subdividing the appropriate faces. Save often, and if you make a mistake, go ahead and use the undo option (**CTRL ZKEY**). Also, if you find yourself looking at redundant faces, combine them (**FKEY**).

**Figure 2.3.2** Appendage formation



If you are having trouble with this, try mousing over the perspective window (the one you designated with **NUM0**) and using the **MMB** to rotate the view so that you are looking at the underside of the wolf. Click on the face underneath the wolf that is alongside his front legs (use the side view to check this). We are going to subdivide this face in order to grow legs off the new faces. To subdivide, press the **WKEY** and choose subdivide. You will see that

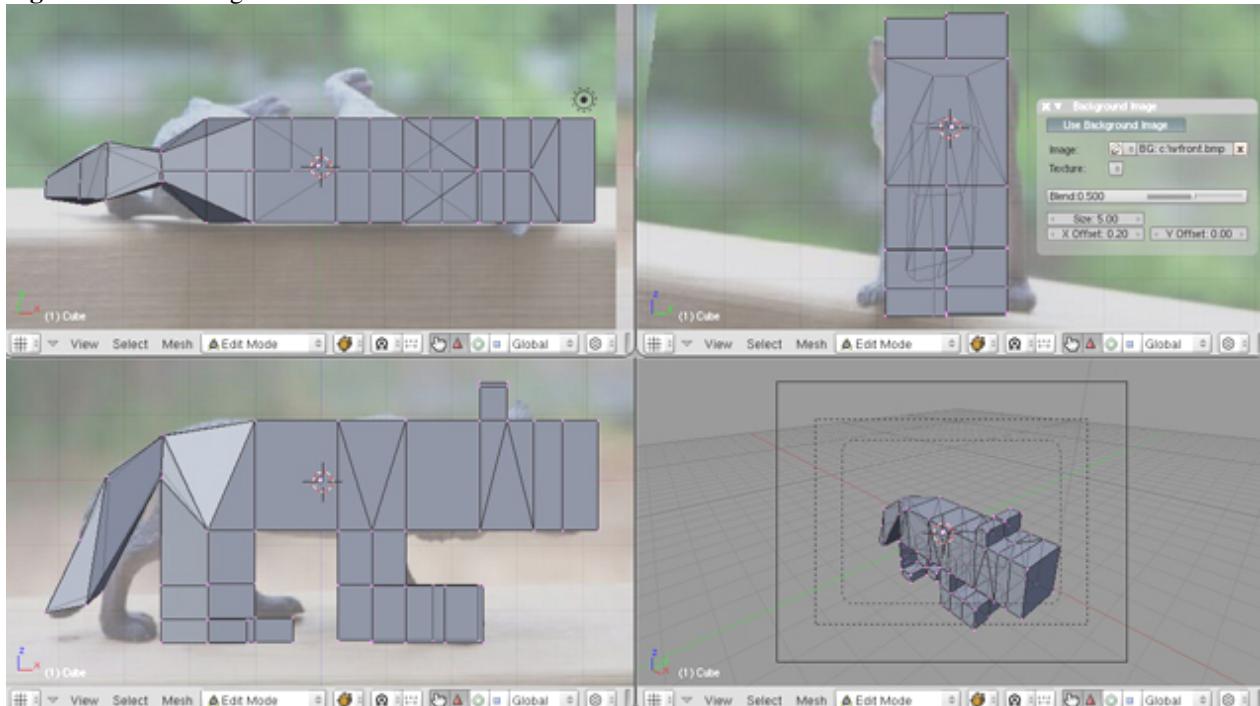
the face has been divided into four. Take one of these faces and extrude it as many times as is necessary to make the right leg. Then do the same again for his left leg. (NOOB NOTE: Be sure to extrude each leg and ear separately exactly as it says here. I multiple-selected both faces to extrude the rear legs from simultaneously and they ended up sharing common vertices at the corners where they meet, making the legs difficult to separate. 2nd Note: Actually it's fine to extrude both, just make sure to select "Individual faces" in the extrude popup menu, rather than region) Use the pictures as a guide.

Doing the ears is similar, except instead of working underneath you will start with the face on top of the wolf which is directly over the ears. Select this face and subdivide it once. Deselect everything using AKEY, then select one of these four faces and extrude it upwards once to make an ear. Do the same for the face alongside it to make the other ear.

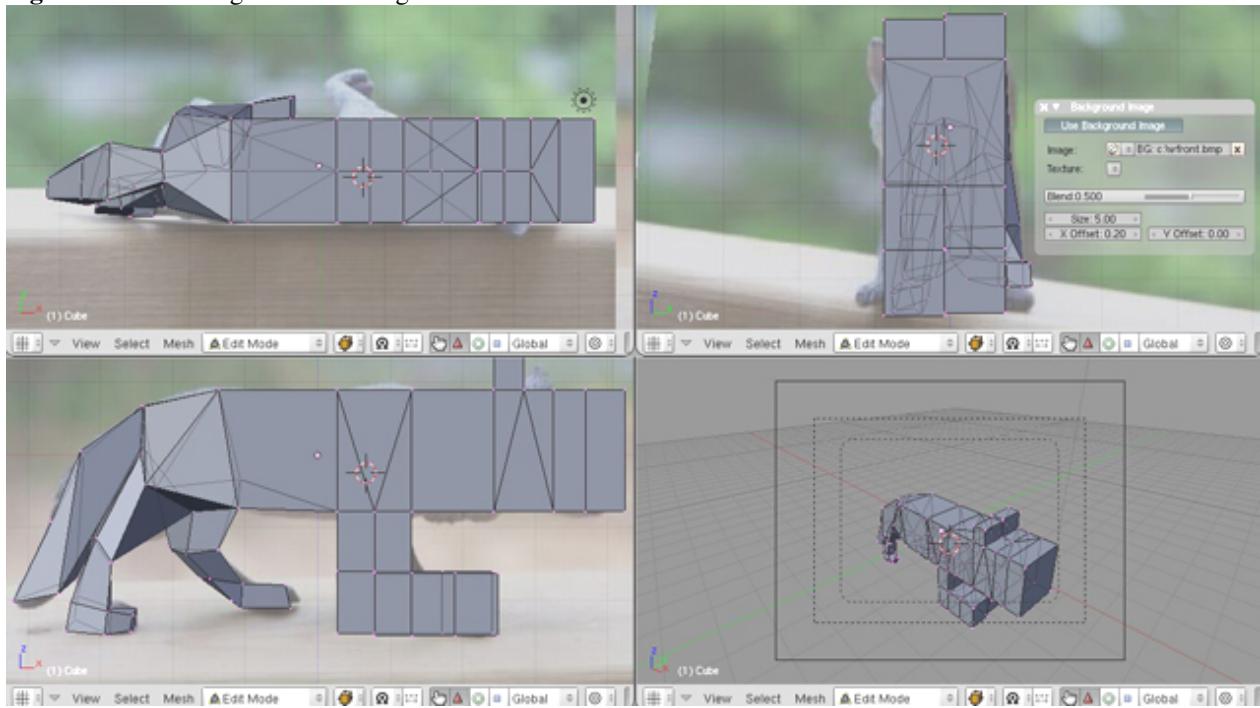
Finally, extrude the tail end of the wolf one more time, so that your wolf has as many divisions as the picture above.

Let's start refining the model starting with the tail. Try putting your viewports in wireframe mode by pushing Z, it may make things much easier. Line up the vertices over the wolf in each viewport by lasso selecting multiple vertices (**CTRL LMB, Drag**). then move to the right location with grab (**GKEY**).

**Figure 2.3.3** Working on the tail

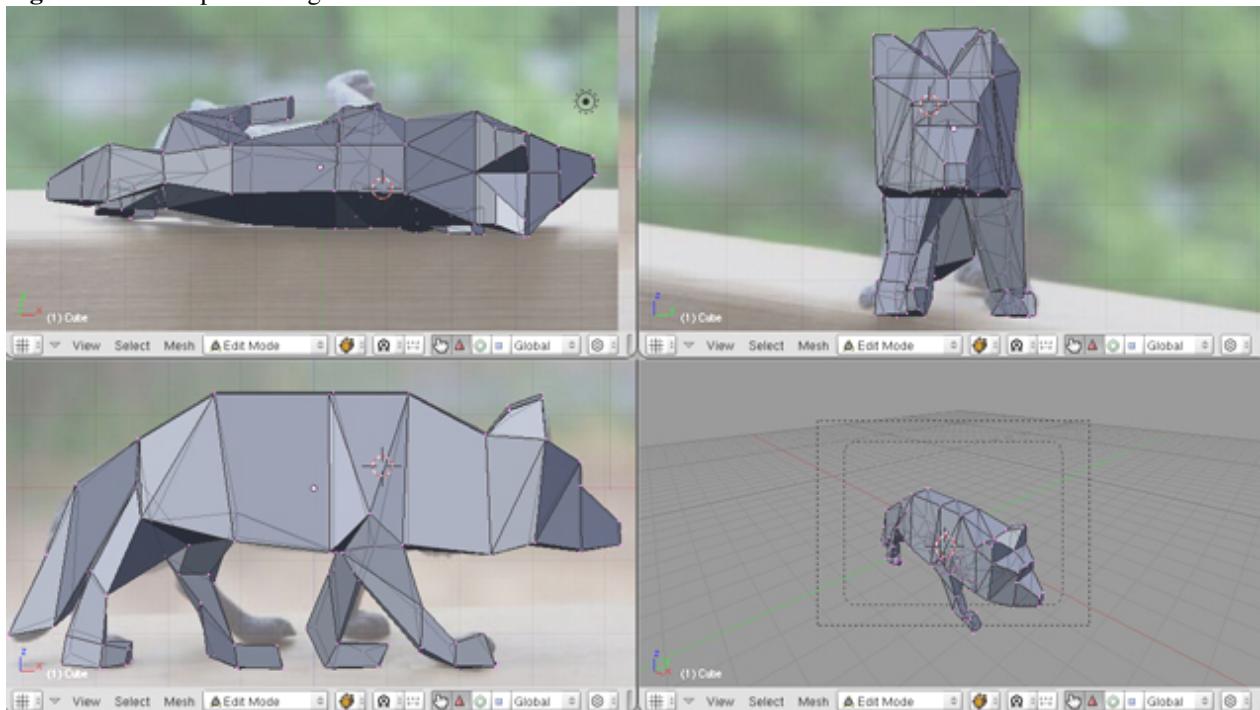


Continue onto the hind legs of the wolf. It is trickier to manipulate the legs so keep rotating a viewport to look at the model from multiple perspectives. Remember that we are working in three dimensions.

**Figure 2.3.4** Working on the hind legs

Continue working up along the wolf fitting the blocks to the pictures. If you have problems seeing the picture because the model is in the way, let's hide the model. In Edit Mode, select the entire model by **AKEY** or by pressing **LKEY** when you have the cursor over the model. Simply pressing **HKEY** will hide the selected items. To unhide the view, use **ALT+HKEY**. By hiding and unhiding the model, or parts of the model, you should be able to keep using the picture as a guide. *Note: It is much easier to just switch layers by, let's say, pressing **2KEY** (not the Num-Block one) to hide the entire model (thus getting a look at the picture) and **IKEY** to reveal the mesh again.*

Once you have the first pass done, you'll notice that the model just won't fit all three pictures correctly. This is due to parallax. The most obvious example is the side view. The four feet should be level, as they are all standing on a flat surface. Since they are not, we'll just ignore some of the aspects of each picture and continue with the model. (This is a helpful example to show what you need to consider when taking your own pictures.)

**Figure 2.3.5** Completed rough fit

Attention: The top right Viewport is said to be "NUM3 - Side" but the Picture shows the wolf from the front! So care and change the picture with the left window at the bottom or change the Viewportsettings

Reader Note: this is because for the pictures to line up correctly the front has to be in the side viewport and the side has to be in the front viewport.

Reader note: The author realizes that he can't line up the images this way in Blender but forgot to explain why after changing the position of the wolves. If you put the TOP wolf in a window set up for a TOP VIEW (NUM7) you'll have a VERTICAL looking down wolf. In the same way if you put a SIDE VIEW of the wolf (looking right) you'll have an error in the FRONT VIEW because SIDE VIEW (NUM3) is planned to receive a LEFT SIDE VIEW of your object, the result is that in FRONT VIEW You'll have the ass o the wolf.

Put in your SIDE VIEW (NUM3) an left looking wolf and in TOP VIEW a looking down wolf and all will be in order, or use the side view to represent front and front to show the side as he does at last, which is much more complicated and difficult to explain.

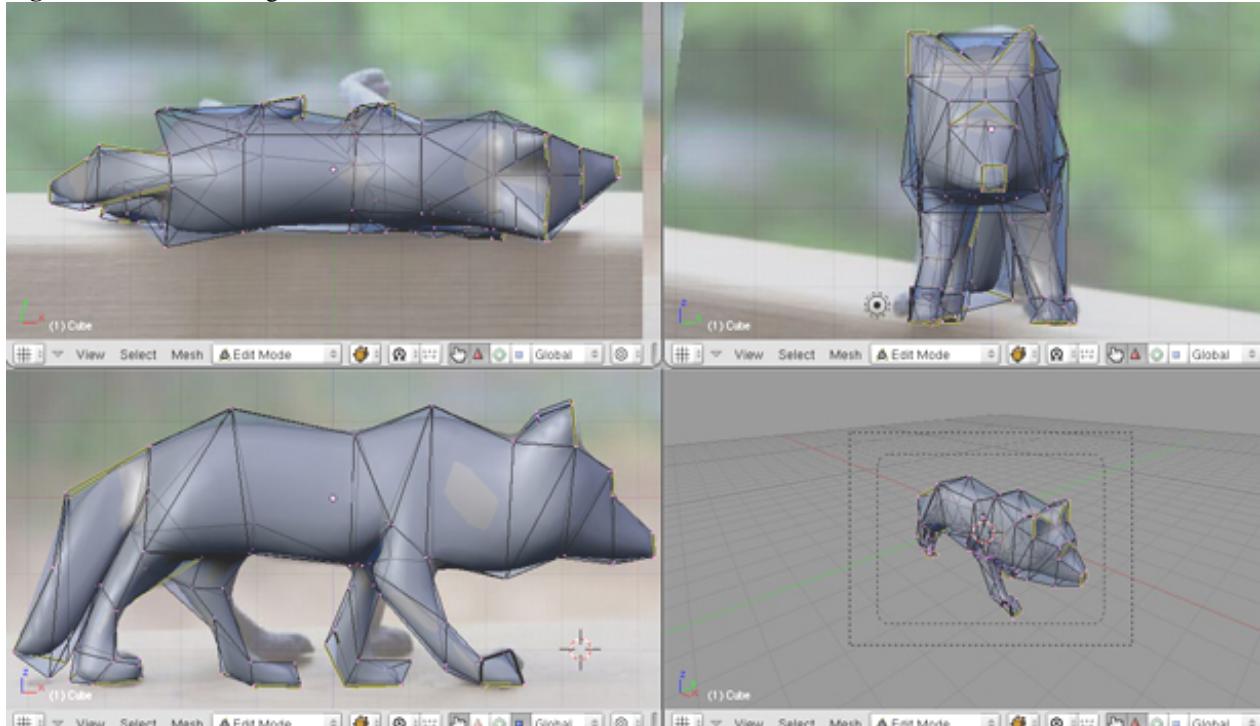
## Step Four: Refining the Wolf Model

Now that the rough fit is done, let's smooth out the wolf. Add a Subsurf modifier and set the Levels to 2. The wolf will now be smoothed, but we want to add some of the hard lines back into the model. This may be accomplished with creased edges.

First, turn on the view creased edges by toggling the *Draw Creases* button in the Mesh Tools 1 window. Enter Select Edges or Select Faces mode (**CTRL TAB**). Highlight the edge or face you want to crease and press **SHIFT + EKEY**. Use the mouse and pull away from the center until the Crease value is close to what you want. A value of +1.000 will give you the sharpest look and is useful for places such as the bottoms of the paws. When an edge has been creased, the edge will be highlighted in yellow (positive crease) or black (negative crease). These highlights are shown due to the 'Draw Creases' button being turned on.

In this example, I creased edges along the paws, tail, ears, and nose to give them some sharpness.

**Figure 2.4.1** Creased edges

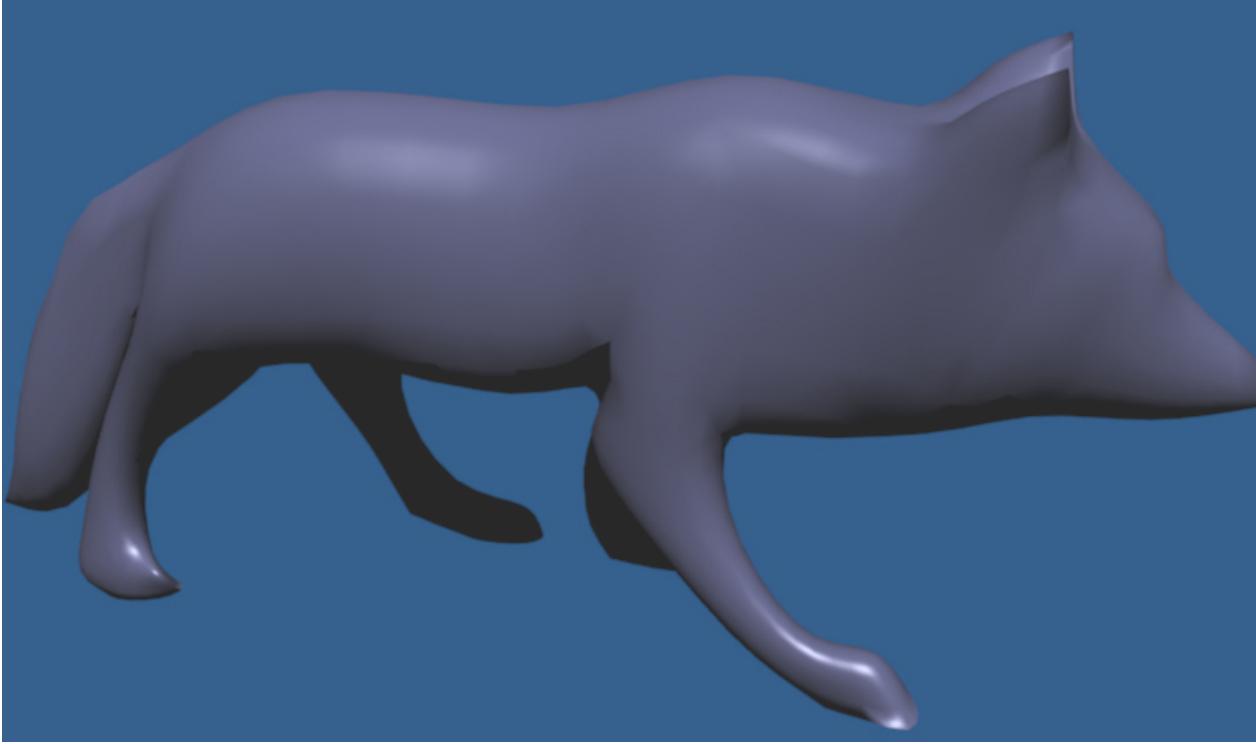


The last step is to refit the model to the pictures. You may have noticed that when the model was smoothed, the result didn't quite fit to the pictures. Now is a great time to tweak the vertices to fit to the pictures or add to/modify

the model.

And here is my basic wolf based on three pictures!

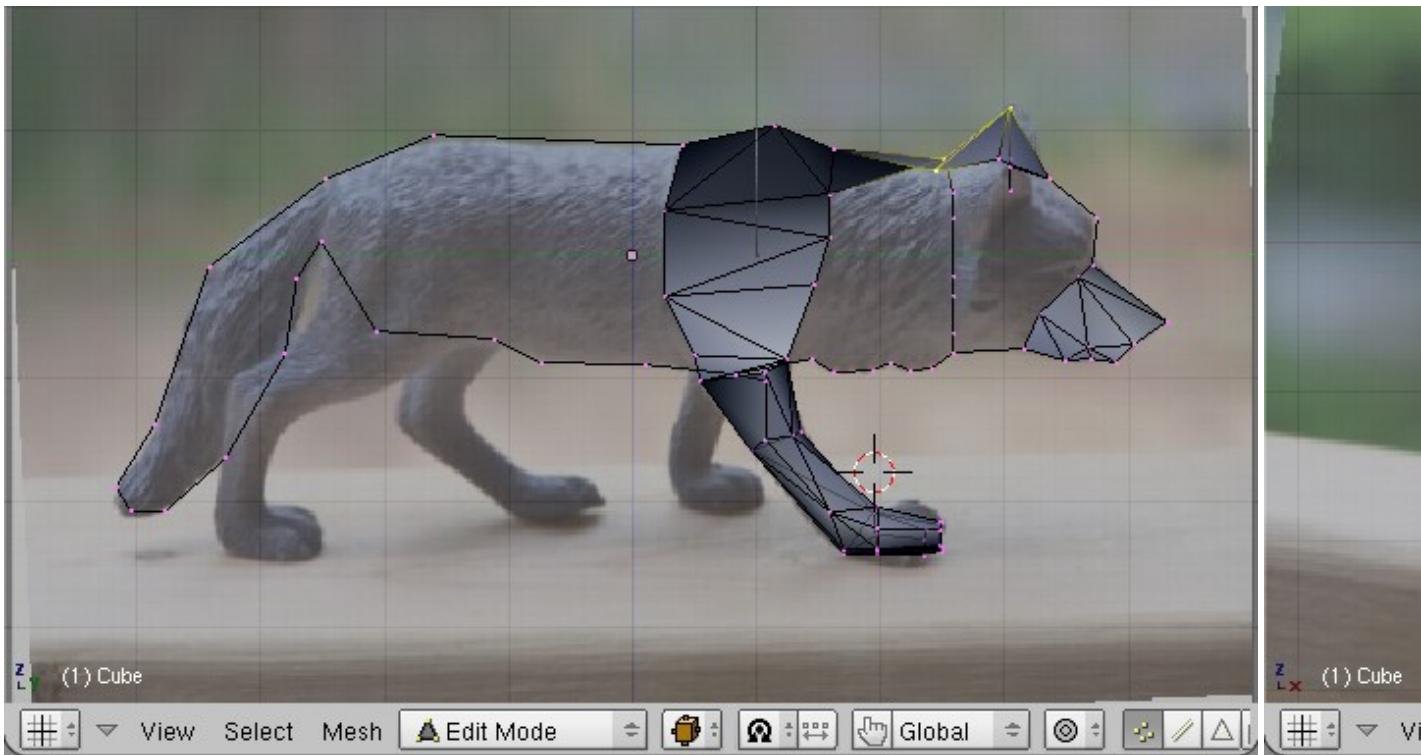
**Figure 2.4.2** Final toy wolf model



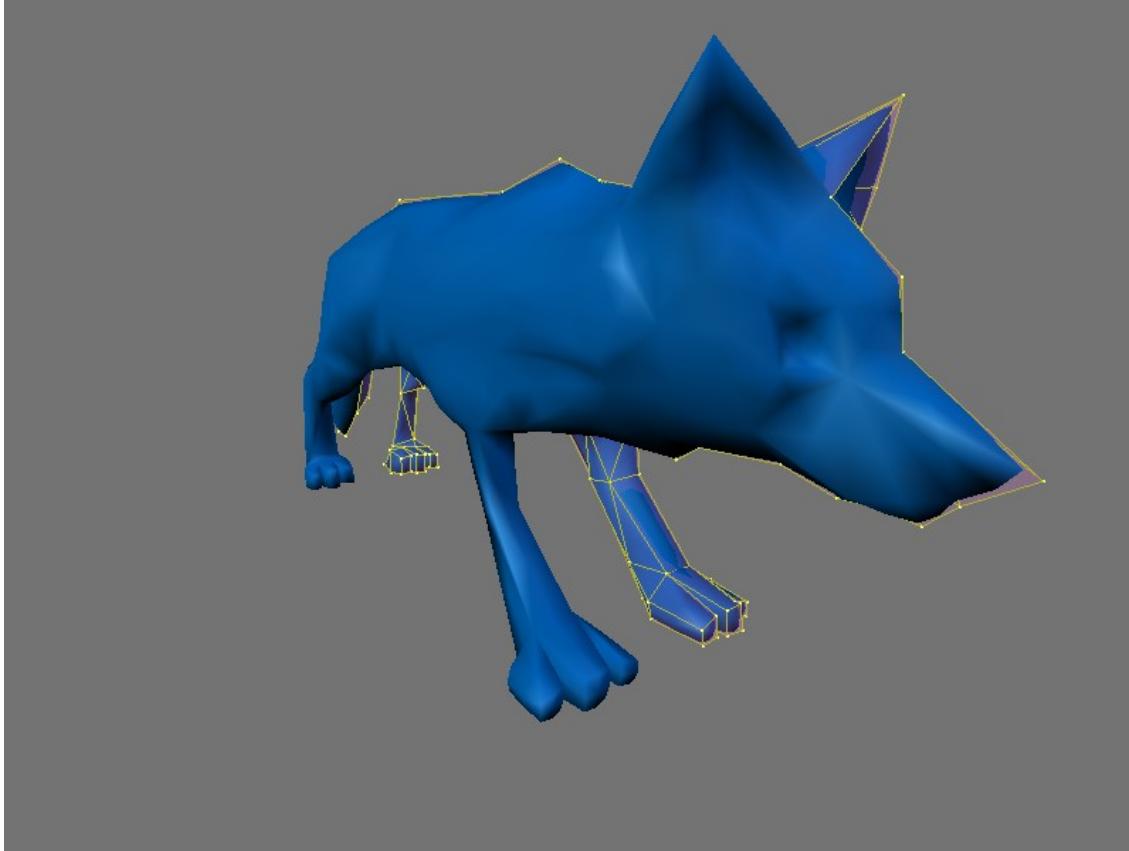
### Alternative Wolf Modelling

(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 silhouette of the wolf body in the side view, and then in front view I started to shape the wolf head, tail and legs...

User Note: If you are attempting this method, I read in a much later tutorial that it is best to avoid using triangles in rounded areas of a Mesh, because they tend to cause artifacts in your model (artifacts are protruding edges or other things that don't look realistic). Apparently, using quads is a much better alternative. Triangles should only be used on flat areas of your mesh if they cannot be avoided.



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.



From this point forward there are multiple ways to do the exact same thing, however for simplicity's 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.

## References

- [1] <http://en.wikipedia.org/wiki/Orthographic>
- [2] <http://img51.imageshack.us/my.php?image=10011325wu.jpg>
- [3] <http://img162.imageshack.us/my.php?image=10011337aj.jpg>
- [4] <http://img107.imageshack.us/my.php?image=10011347fw.jpg>
- [5] <http://img107.imageshack.us/my.php?image=10011350gu.jpg>
- [6] <http://img162.imageshack.us/my.php?image=10011363ql.jpg>
- [7] <http://img119.imageshack.us/my.php?image=10011375bc.jpg>
- [8] <http://en.wikipedia.org/wiki/Parallax>

# Blender 3D: Noob to Pro/2D Image (logo) to a 3D Model

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

{Construction on hold, feel free to complete}

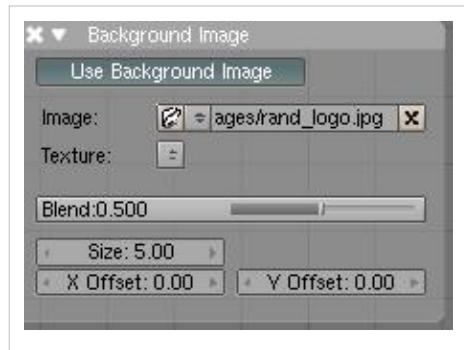
The image to the left is used in this tutorial. However, the tutorial is easier to follow using letter/numbers, or simple shapes/curves. Basically we will be using the graphic as a template for a 3d logo, tracing it, then discarding the 2d image.

### 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.

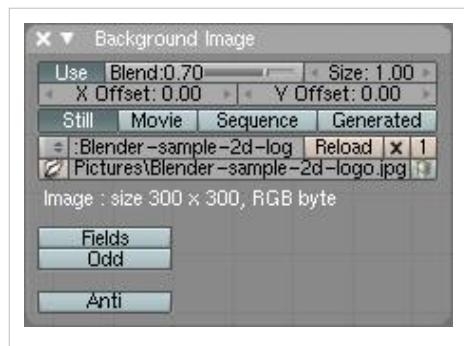
Blender now displays this image in the background of the 3D view for you to trace its outline. The image is only displayed in orthogonal view. If perspective view is enabled, toggle to orthogonal view by pressing **NUM5**. The image will not be rendered as it is not part of your scene.



Once a background is selected you'll have a dialog like this one. (Note: This tutorial was originally generated from Blender v2.37. v2.43 has been added - older versions may differ.) The background dialog buttons are described below:

The **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. In v2.43 the equivalent button is the **Use** button.

Image selection is controlled on the row labeled **Image:**. There are 2 buttons, a text box, and a final button. The first button  is used 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 `ages/rand_logo.jpg` allows typing in the file directly. The  button removes the current background image. Version 2.43 is the same with the addition of the **Reload** button that refreshes the image or movie, and the **1** button which shows the number of users of the image block.



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.

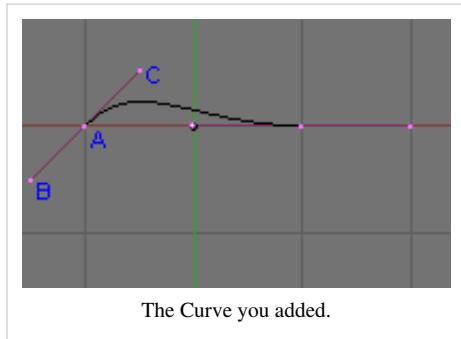
## Introducing the Bezier Curve

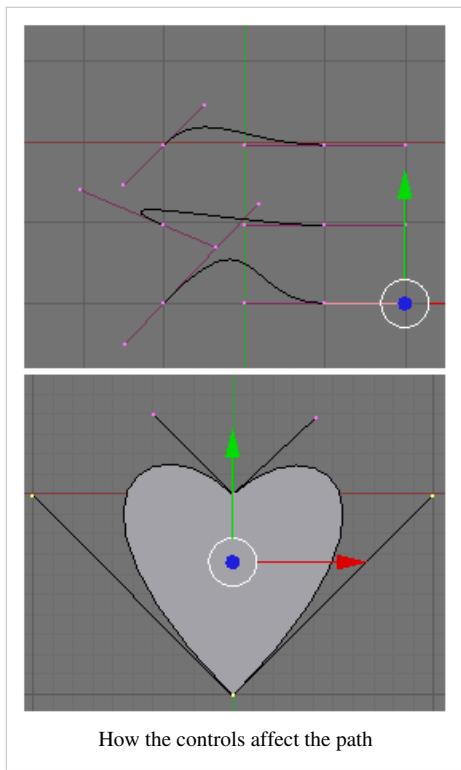
The Bezier Curve allows drawing graceful, complex curves and only requires a few control points. Specifically, it only requires 4 points for a curve. Two end points and two control points.

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. Be sure to be in EDIT MODE not OBJECT MODE. 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**.

**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.





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 curve's next point.

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.

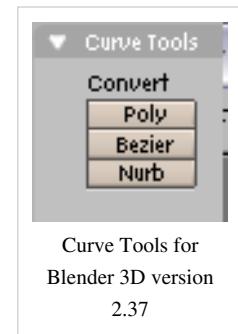
In the bottom example, I've built a heart shape using just the points shown. *Dragging the bottom end point down will make the shape closer to a leaf.* You'll be able to do the same at the end of this tutorial. Go ahead move around the points for the curve and see how they all interact. Get a good feel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

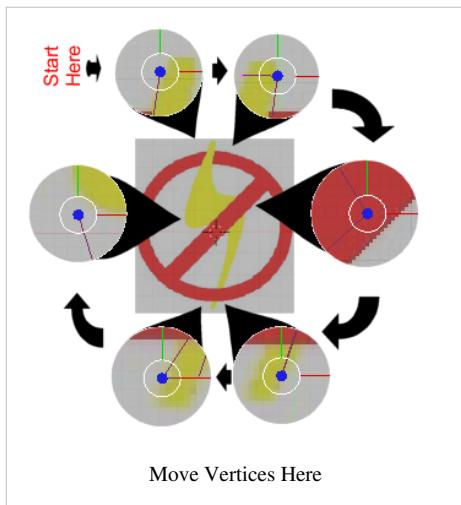
Now that you know how to work with a bezier curve set blend back to 0.5 on the Background Image dialog so we can start tracing.

## 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.

User's Note: The 'Curve Tool' panel will not appear if you don't have a bezier curve already placed. I learned this the hard way.





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.*

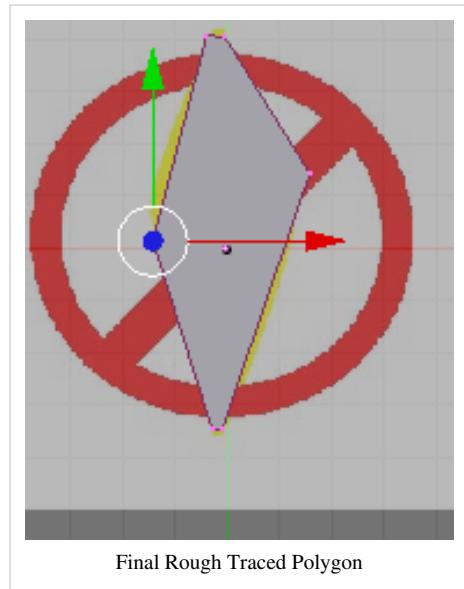
This tracing uses all the vertices of the polygon. Other cases, you'll need add or remove extra vertices. Adding and removing vertices as shown in Turning a Cube into a Puppy tutorial (Note: To add a vertex select the end point of your curve press **CTRL** and click **LMB**. At the place you clicked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

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.

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.

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.

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.

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 tutorial.

---

## 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:

(\*Make sure the Front and Back buttons under the "Curve and Surface" tab are selected, otherwise the logo will be a solid outline lacking a face on the front and back.\*)

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

Bevel Depth: 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 previous versions, Extrude and Bevel Depth were Ext1 and Ext2.)

Also, if you have a simple logo go ahead and increase the DefResolU 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.

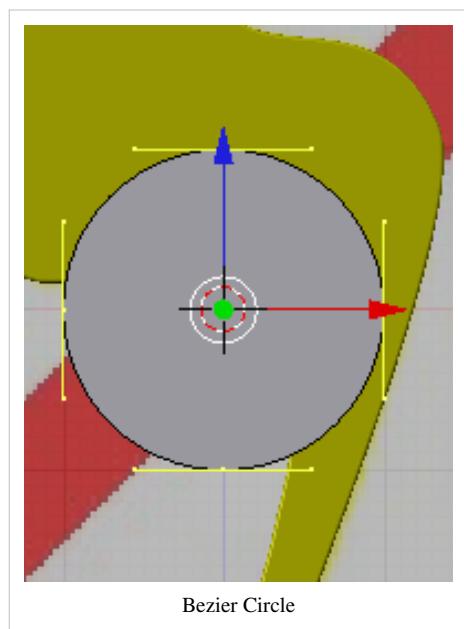
-=< Tutorial under Construction, ready soon, thanks for input spiderworm >=

## Continue tutorial using bezier curves

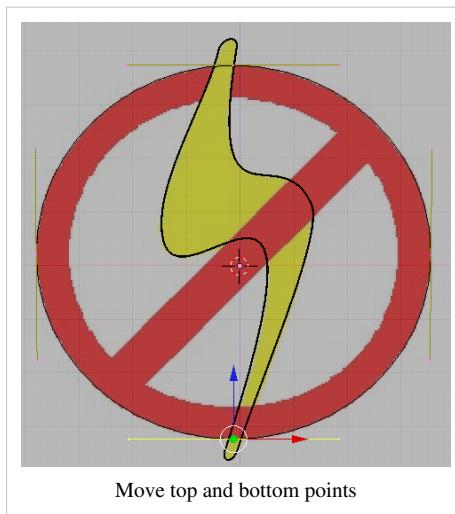
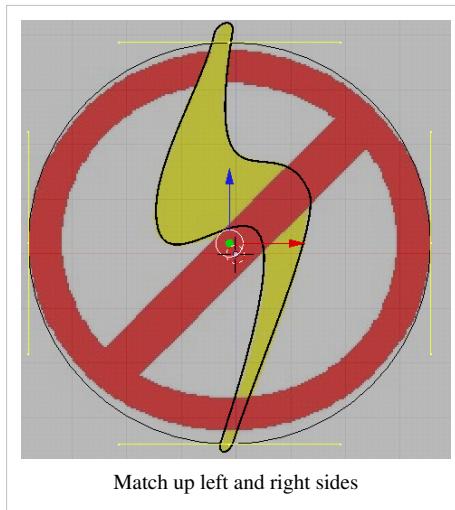
- This continues the tutorial, finishing up the sample logo from the image in the tutorial using bezier curves. These instructions make the assumption that you completed the first part in front view with **NUM1**. An alternative method for doing this using mesh circles has been presented below.

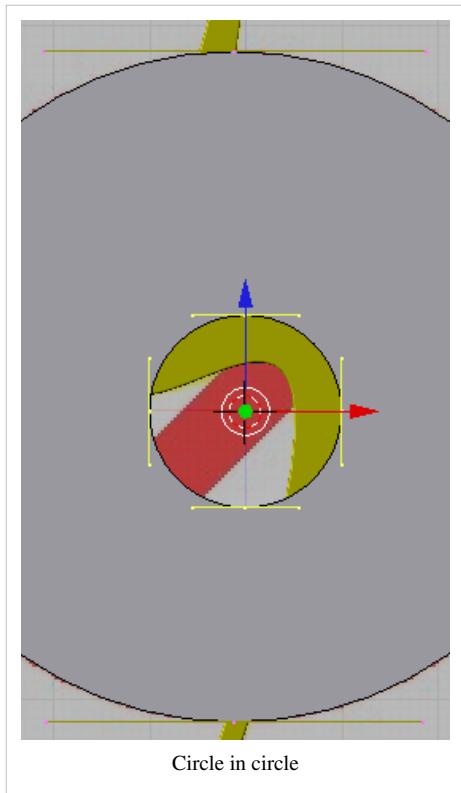
First, to make the lightning bolt distinct from the second part of the logo, it may help to apply a yellow material to it before getting started.

## Adding the circle



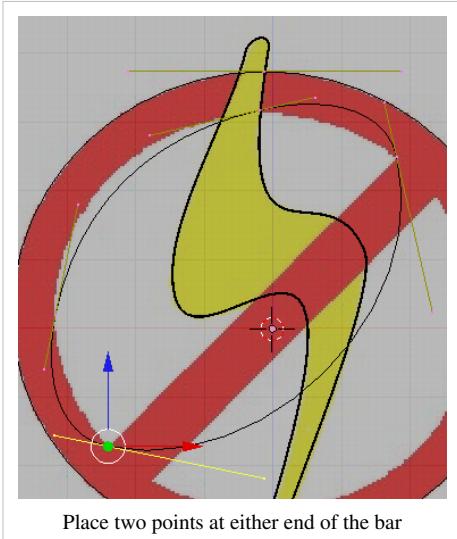
Switch to object mode by hitting **TAB** if you aren't already there. Press **space** -> **Add** -> **Curve** -> **Bezier Circle** to add a closed bezier curve with four points forming a circle. If you are in solid draw type, switch to wireframe with **ZKEY** so you can see the underlying image better. 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.

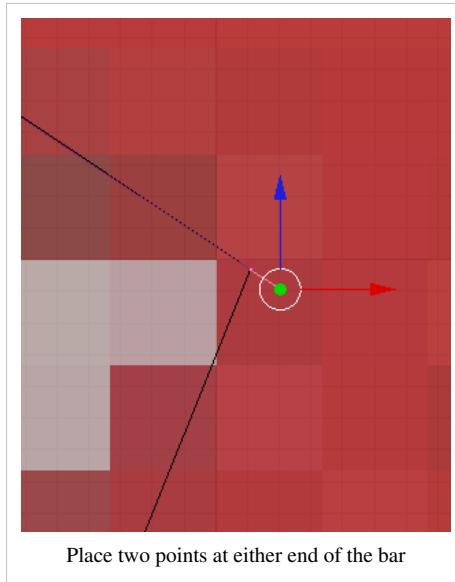




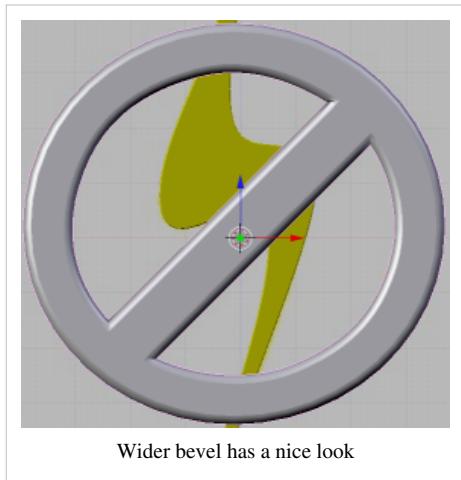
Just to understand what's happening in the next steps, switch to solid view with **ZKEY**. As you can see, you now have a circle, but it's filled in the middle where you want to be able to see through it. To cut a hole out of the circle, hit **space -> Add -> Bezier Circle** while you still have it selected 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.

Move the right point of the circle to the other corner. When you create a Bezier Circle, Blender by default sets the alignment of all the control points to aligned. To make the diagonal bottom edge you need to break the alignment on the two lower sets of control points. Hitting the **HKEY** will toggle between free control points and aligned. Once you've selected the two lower bezier points and hit the **HKEY** to make them free you can move each of the inner control points to create a nice straight edge.

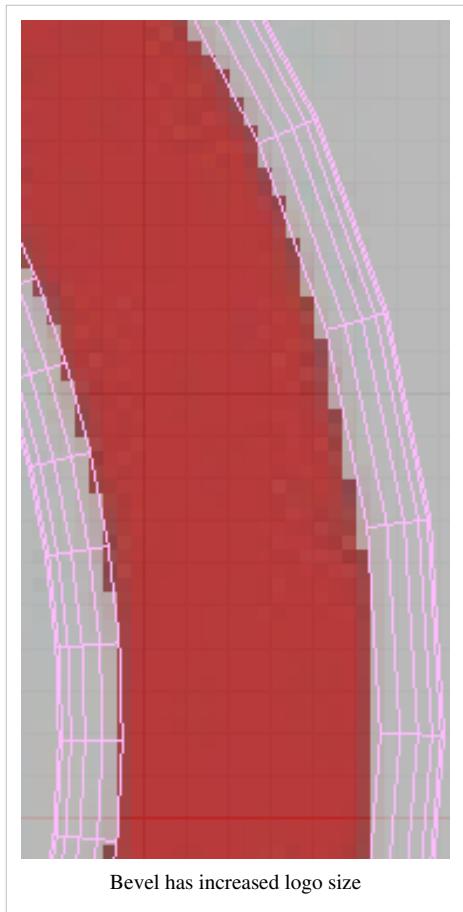




Then move the other two points and adjust their control points until you have a pretty good approximation of the rest of the inside curve. Next step is to press **space -> Add -> Bezier Circle** again and repeat the same steps, but for the lower opening in the logo. Once you've completed both openings,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



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.



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.



Width adjustment restores to normal size



Final render

**[Noob Note:** I'm new to Blender but have been dealing with bezier's for a long time, and this method of creating the slashed circle seems excessively complicated (whereas the other alternative presented bypasses the point of using curves altogether). The inside of the circle is just 2 half circles cut out from the original larger circle and would only need 2 points to draw out (for each half). Here's what I did.

Another noob note: I found that instead of doing all of this, just insert the inner circle, rotate by 45 degrees, and scale and grab along y and x axis and you will have a much better looking circle

After placing the large circle as instructed above, add a Bezier curve **space -> Add -> Curve -> Bezier Curve**.

Select the individual points and use to move them to the ends of the half circle. grab the control points and pull it up and away from the actual vertex points until the shape of the curve fits the image.

Select the vertex points again and hit **HKey** to change the control points to **Free** mode.

Now adjust the remaining two control points so they are pointed right at each other (so it will draw a straight line when we close the shape). Then **CKey** to close the shape and there is a Bezier half circle.]

note added 12/08..... all these methods produce something that is "LIKE" the logo jpg but all end up different. Look at the picture..THE TOP PART OF THE LIGHTNING BOLD IS ON TOP OF THE RED CIRCLE... only the body and bottom tip of the lightning bolt are underneath it. The lightning bolt must be rotated on the x axis to correctly finish the logo.

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

- The following 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 lightning bolt, which is great. It's 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.

Here are two ways of doing this:

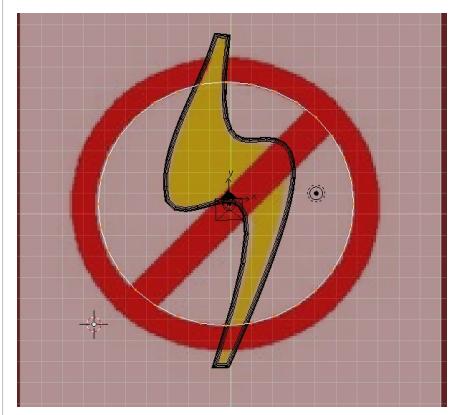
### **Easy way:**

Place two vertices (CTRL+LMB), one on the inner circle and one on the outer circle so that they form a line. This line has to be perpendicular to the circle you are tracing. Then place the cursor on the middle of the circle. Then just use the spin tool under the Mesh tools tab (360 degrees and 32 steps). This creates a circle made of 32 adjacent squares. Give it more steps to increase the quality of the circle.

### **Detailed way:**

Go to the top view (NUM7), and press SPACE -> Mesh -> Circle. Accept the 32 vertices, you can make it less but it won't look as good.

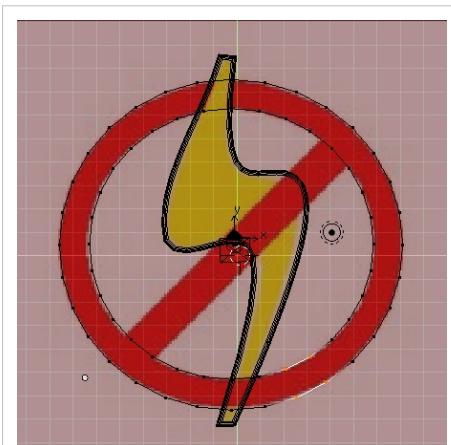
Move it into the center of the circle, if you don't then I advise you have wireframe on for the moment (press Z). Then press S, for scale, and make it the correct size for the inside of the circle. Once you have that in the correct place, like so:



Adding the first circle

You may notice that I needed to stretch it sideways a little; you will too. OK, press 'E' to extrude (choose Only Edges), press 'S' to scale and another sized circle will appear, size this appropriately then click LMB.

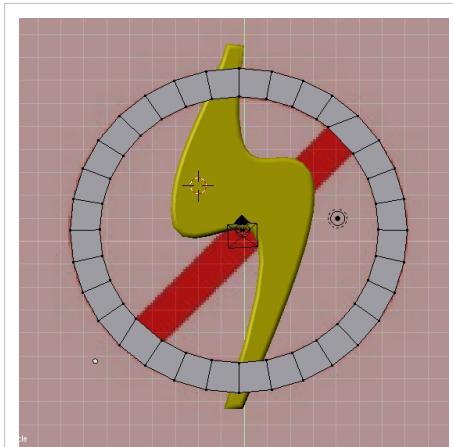
If you don't read this carefully, then you may not get the wanted end result. Deselect the second circle, then select four vertices that are near each other and that form a square.



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 it's time to make the line through the middle... this is easy!

To make the crossing line however, you need to move 4 of the vertices slightly; example below:



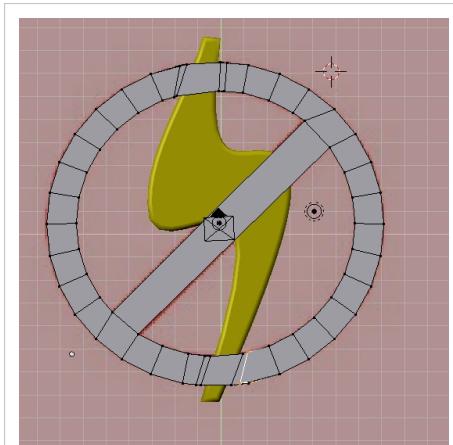
Moving the four vertices

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

### Making the circle 3D

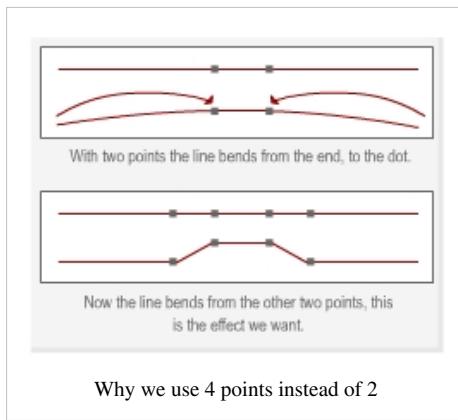
Highlight the full circle by pressing A either once or twice. Go to Side View, and press E for extrude and drag it down so that it is the same thickness as the lightning bolt (you'll see why). Now this bit is purely for art's sake, you won't probably learn anything here but it's good practice.

Now look at what you have made... it looks nice enough but where the lightning 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.



Now we can make the finishing touches, add subsurf and set it to 2 or 3 and add color! Then you are done :-) I won't go over subsurf and adding color because people have covered that better than I could in previous 'Noob to Pro' pages. I think that's everything. I did this and ended up with the first screenshot so I'm sure you will as well. 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

## Extra Practice

An Alternate More Difficult Tutorial <sup>[1]</sup>

## References

[1] <http://forum.nystic.com/viewtopic.php?f=34&t=3025>

# Blender 3D: Noob to Pro/2D Image (logo) to a 3D Model Part 2

## Tracing an Image With an Interior Cutout

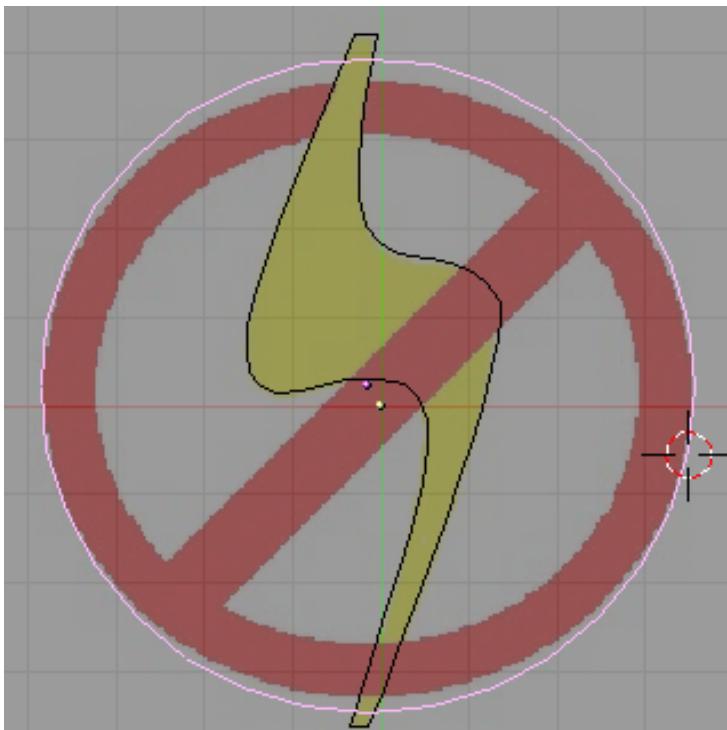
**n00b note.** This is a other way (faster) add a open circle, make it right size, extrude + esc, scale, mark them both and press F for fill, and rotate it so 4 vertexes are the corners of the line over the lightning, make edges with 2 vertex and F, I think u know the rest so ur done now :D

Tracing the **No symbol** is somewhat more complex than the lightning bolt. The reason is this symbol is hollow and requires additional planning to trace than just following an outline of an object.

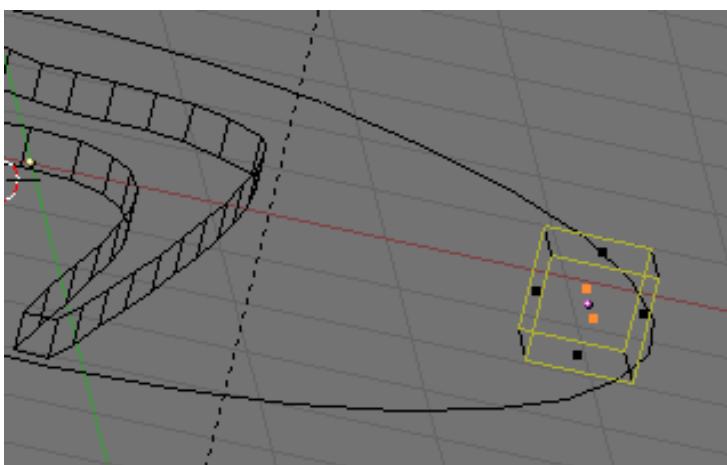
I'll be back to explain the difficulties in the near future, so stay tuned.

One way to do this is outlined below:

- The first thing to do is add a mesh circle. Hit **SPACE** and select 'Add -> Mesh -> Circle'. (This circle is mostly for measuring purposes, and will ultimately be removed.)
- Scale (**SKEY**) and move (**GKEY**) the circle so that its sides are even with the sides of the logo circle, and the top and bottom are an even distance from the top and bottom of the logo circle. (The logo circle is not perfectly round.)



- Make sure you're in object mode (**TAB**), add a cube (**SPACE**, 'Add -> Mesh -> Cube') and place it (**GKEY**) on the right edge of the circle so that it is centered on the right edge.
- Scale (**SKEY**) the cube so that it is the same width as the wall of the circle and touching each side of the wall.
- Now, we want the 3Dcursor at the center of the logo circle. To do this, go back to object mode (**TAB**), and select the circle that we added earlier (**RMB**), then press **SHIFT+SKEY** and choose 'Cursor -> Selection'. (This is the main reason why we have this circle here.)
- Reselect the cube (**RMB**) and **TAB** to edit mode.
- Choose **NUM7** for the top view, and use **RMB** to select only the front face of the cube. (Before this you have to make sure that you are in 'Face select mode'.) In the image below, I've used **MMB** to rotate the scene slightly so that you can see that only **one** face is selected (Note: In this picture there are two faces selected. The one you should select is front one in front view **NUM1** or the bottom one in top view **NUM7** - it's the same face).



- Press **XKEY** and choose Erase Vertices. This will reduce the cube to a square

(User note: Erase Vertices will erase all the vertices connected to the two faces selected. In this case this is all the vertices in the cube and will completely remove the cube. I think "Erase Faces" would be correct.) (Another User note: Only one face shall be selected, not two as shown in the picture.)

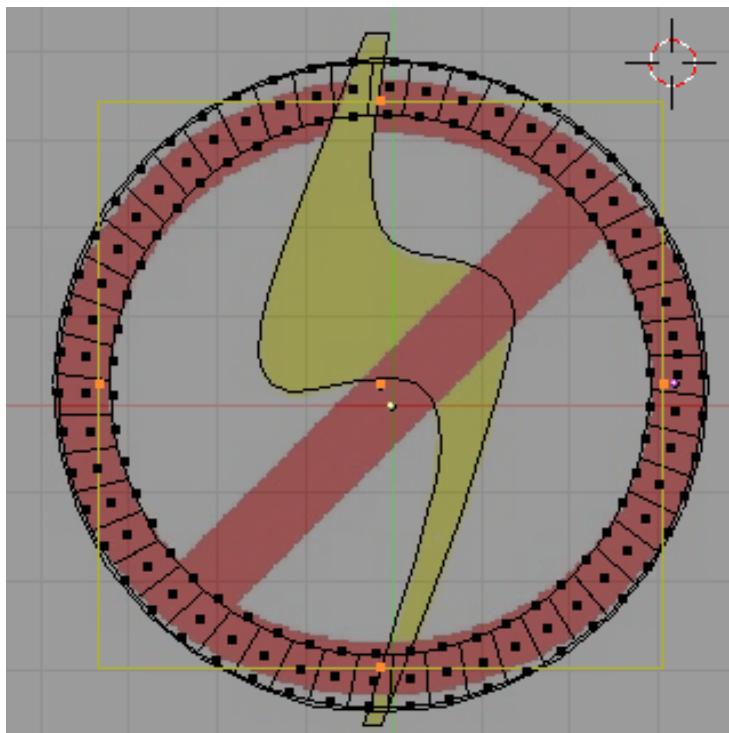
twofingers: @1st - READ carefully, @2nd - before editing Wikibooks DISCUSS it first in the DISCUSSION tab! and @3rd - SIGN YOURSELF!

- Press **NUM7** to go to the top view then select the Editing panel (**F9**).
- Press **AKEY** to select all, and then make sure that the rotate settings are set to degr:360 , Steps:50 , Turns: 1.  
(Actually, Steps can be whatever number suits your fancy. You may want to play with various values.) This is just like we did for creating the man's hat.
- Click on 'Spin' to extrude the square into a ring.

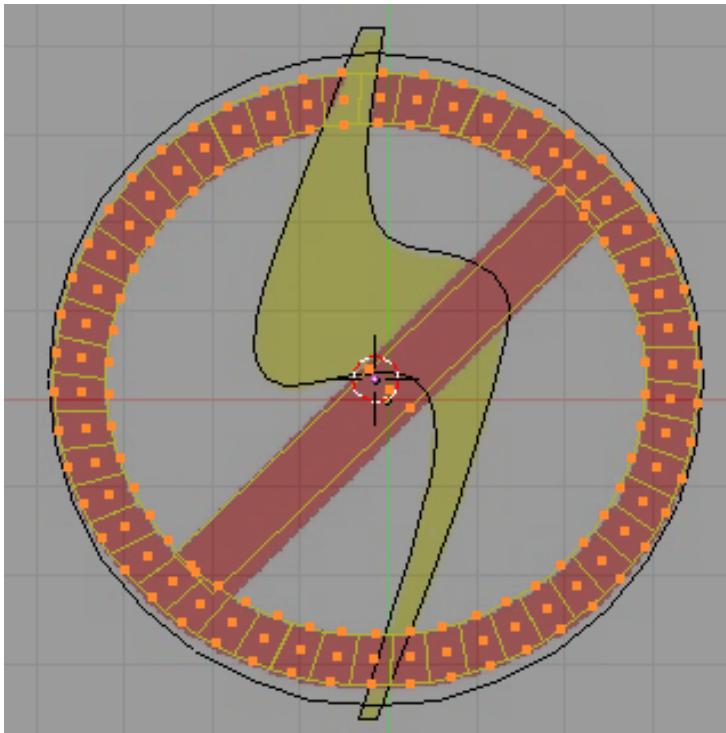
(Another way to do it is to create a ring with 2 Tubes (**SPACE** -> Add -> Mesh -> tube). Scale one to the size of the outer border of the ring and one to the size of the inner one. Then in edit mode select 2 Vertices next to each other on the inner and outer tube and made a face (**FKEY**) which covered one segment of the ring. Then you have to spin it like described above (the number depends on the number of vertices you choose for your tube tube = 32 Vertices -> 632 rotationsteps). Same on the other side. If your faces are not in the right place go to object mode select the tubes an press **SHIFT+SKEY** and choose 'Cursor -> Selection' to center the 3D cursor.)

Next we will create the bar.

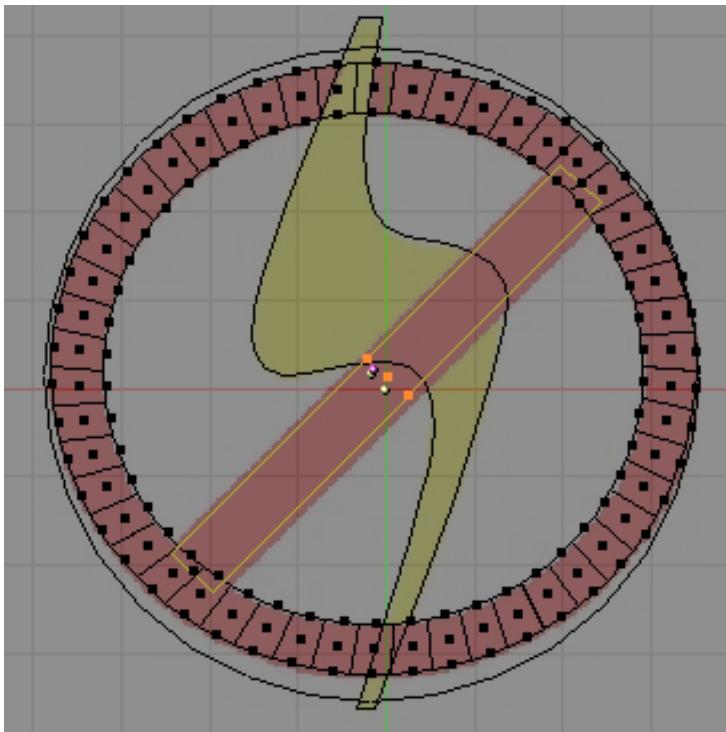
- Staying in Edit mode, hit **SPACE** and select 'Add -> Cube'. If you haven't moved the cursor since the ring extrusion, it should appear in the middle of the ring.
- Expand (**SKEY**) the cube so that it just encompasses the inside circle of the ring



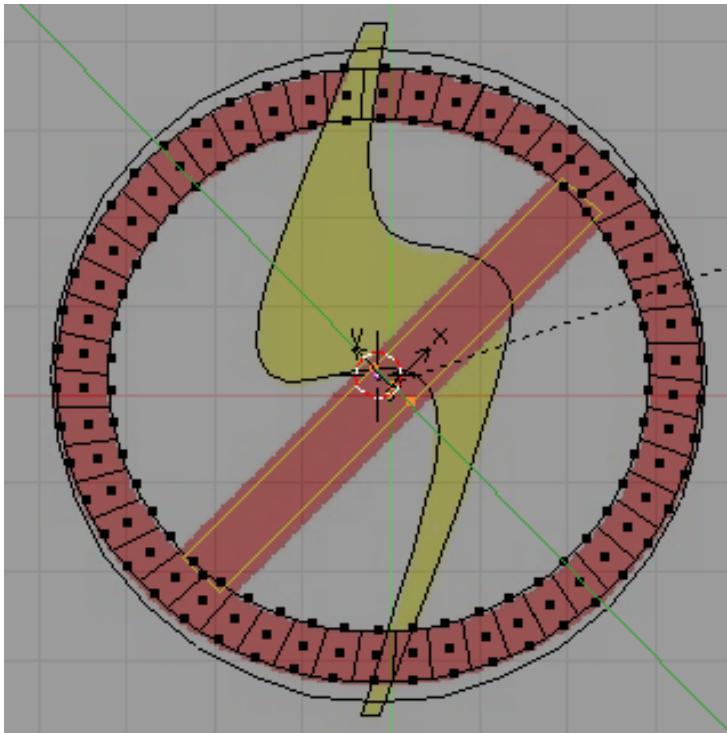
- Scale (**SKEY**) the square in the Y axis (**YKEY**) so that it is approximately the thickness of the logo bar.
- Return to object mode (**TAB**), and rotate (**RKEY**) the ring together with the just designed bar so that it is parallel to the logo bar.
- In Edit Mode (**TAB**), squash the ring in the Y direction (**SKEY**, **YKEY**) so that it fits the logo ring.



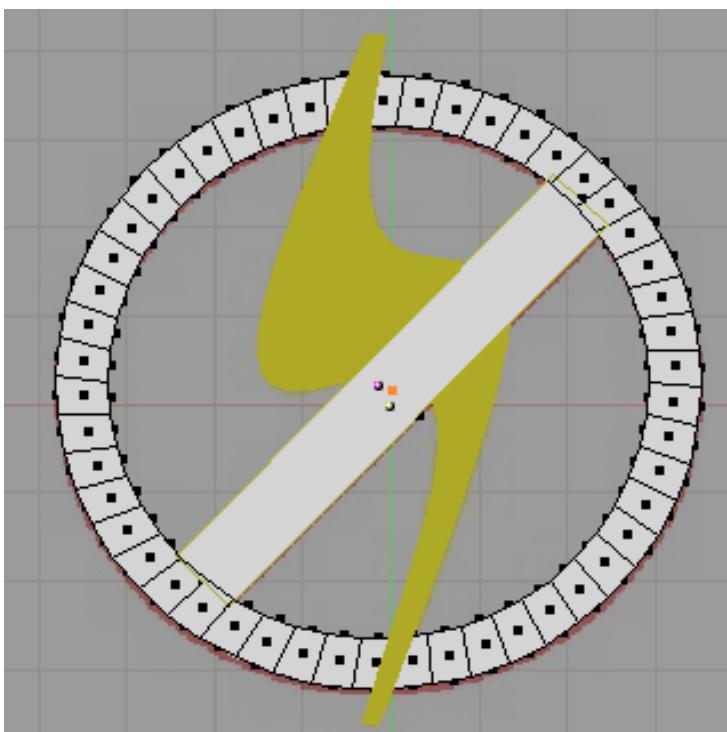
- In edit mode, reselect just the bar. To do this, choose 'Face select mode', and do an area select of the face selectors near the center of the bar (use the **BKEY**, or **SHIFT+RMB** methods).
- Re-rotate (**RKEY**) the bar (in Edit Mode) so that it is parallel to, and in the middle of, the logo bar.



- The next thing we need to do is scale the bar so that it is precisely the same width as the logo bar.
- To do this, choose scale (**SKEY**), and press **YKEY** twice. This will go to local scale mode (local to the object). This is why we were rotating the entire ring, since the ring and bar are part of the same object.



- We no longer need the original circle now, (it should be sticking out from the top and bottom of the ring), so select it (**RMB**), and delete it (**XKEY**).
- If you hit **NUM3** You'll see that the bar and the ring are at completely different heights than the lightning bolt. Select and scale them so that they are all roughly the same height (the bar should be slightly higher than the bolt, and the bolt slightly higher than the ring so that all the proper parts are covered.).
- You can use the trick above to select (**RMB**) only the bar, scale it in the Z axis (**SKEY**, **ZKEY**) and then hit **SPACE** and choose 'Select -> Inverse' to select the ring (all but the bar).
- When done rescaling, press **NUM7** to look at your handiwork.



- There is one last bit, which is to slightly rotate the whole ring around the X axis so that it is below the top of the bolt and above the bottom.

- Once that's done, all that's left to do, is color in the ring.

## Blender 3D: Noob to Pro/Curve and Path Modeling

Frighteningly enough, we know what you're thinking. You're thinking that mesh modeling is cool and all, but what about if you want to make an object that has smooth curves in it? Ok, so maybe you weren't thinking that, but in case you're curious move on to the next page to learn more.

## Blender 3D: Noob to Pro/Intro to Bezier Curves

Per request, this page is currently being edited to be more "tutorial like" and user friendly.

The work can be found in its Original Version (which is currently being displayed) and the Edited Version (which could be an option to replace this page).

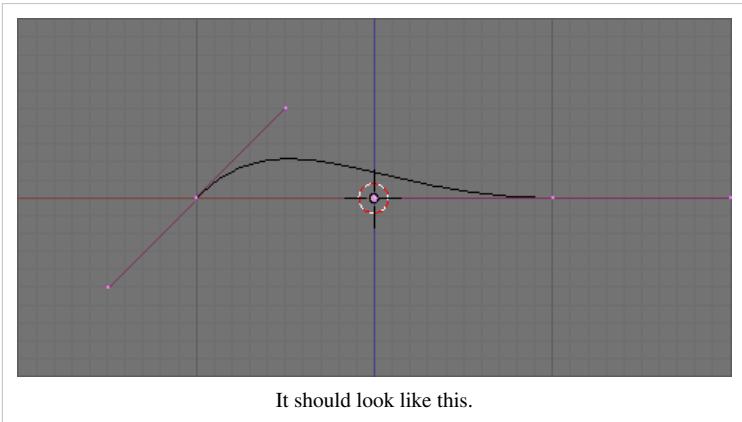
### Bezier curves

You wanted curved lines? Well here is how to get them, using Bezier curves.

- First start a new Blender project, and delete the default cube.
- Press: 'SPACE - Add - Curve - Bezier Curve' to create a new curve.

As you can see, there is a black, curvy curve on your screen, inside what looks like a normal mesh (purple). Well you wouldn't be far wrong with that interpretation, because that is effectively what a Bezier curve is!

- Try selecting the vertices and moving them as we have done previously. You should notice they behave much like a normal mesh does, but with a few important differences. Firstly, there are two types of point here, a 'control' point, and a 'handle' point. The control points define where each section of the curve has to start and finish, while the two pink handles help guide the curve between them.



By now you should have discovered you can drag the points around using the **GKEY**, and that the handles are fixed in position relative to their control point. This is not the only manoeuvre however, so here is a list of things you can do to the curve to modify it:

- Grabbing:** You can drag the points around separately or in groups.
- Rotating:** Using the **RKEY**, you can rotate a control point around with your mouse. This is different from a mesh, as rotating the control point will move its handles too.
- Scaling:** Using the **SKEY**, you can scale either an individual handle, or a whole control point at once. This changes how strongly a handle influences the curve's shape, not its direction.

Whatever you do to the defining points, Blender calculates the curve so it flows smoothly from one end to the other. When you leave Edit mode, only the black curve will remain visible, so don't worry about your control points getting cluttered!

Noob note: At any time you can convert a Bezier curve into a mesh by entering object mode (**tab** if in edit mode), hitting **ALT+CKEY** and selecting 'mesh' from the menu. You may then operate on the mesh like you would any other. This is great for creating a smooth Bezier curve, converting it to a mesh, and spinning it around an axis.

## Extending your curve

So far your curve has only two control points, but in order to build up more complex shapes you will need to add many more! There are quite a few ways to do this, but let's start with the basics:

- **Extruding:** Much like a mesh, you can use **EKEY** to extrude one end of your curve with a new control point. Select an end control point (not a handle, this will cause you a few problems) and press **EKEY**, then **LMB** when it is in the correct location. The new point will have an exact copy of the handles of your old point, so you will have to rotate and adjust it accordingly.
- Note you can use **CTRL+LMB** to achieve the same effect.
- **Subdividing:** Instead of adding onto the end of your curve, you can also split up sections of it into more control points, and here's how:
  1. Select two adjacent control points using **RMB**.
  2. Press the **WKEY**, and select **Subdivide** from the popup menu.
- The segment should now have a new control point mid-way between the first two, that sits exactly on the curve that already existed. Now you can add more shape to this section using the new point.

## Filling your curve

So far we have been working with a continuous curve, with a definite beginning and end. But what you can do is turn this wiggly line into a wiggly shape.

- Select any part of the curve in Edit mode, and press the **CKEY**. This will connect your last control point back to the first, and fill in the shape.
- If you cannot see any fill, you are probably in wire-frame mode. Press the **ZKEY** to recover.
- If you have weird results, say fill leaking out from the curve or empty sections, you have probably got a couple of parts of the curve crossing themselves. **Curve sections should never cross if you are making them into a filled shape!**

## Handle types

Thus far with curves, whenever you move a handle around its partner will move too to make the curve flow smoothly through the control point. This is not the only way to do things however, so here is a list of the different things you can set them to do:

1. **Linked rotation** (Pink handles): Both handles always point in opposite directions. This leads to a smooth curve.
  2. **Free rotation** (Black handles): The handles can be moved independently of each other, you can form corners and sharp bends in the curve. Switch between these two modes using **HKEY**.
  3. **Automatic** (Yellow handles): The handles are linked in rotation, and automatically rotate and scale to form the nicest (mathematically speaking) smooth curve through your points. Use **SHIFT+HKEY** to switch to this mode. If you try to move a handle, both will default back to Linked rotation.
  4. **Vector** (Green handles): The handles act as vectors, with each pointing directly towards the next control point. This leads to completely straight edges, so you can construct polygons. The hotkey is **VKEY**. If you move a vector handle, it will default to Free rotation, but the other will stay as a vector.
- Note, each handle from a control point can be assigned a separate type independently (excluding Auto). Just select only the handle vertex instead of the control vertex.

## Combining curves

As well as having one single filling curve in your object, you can combine several unconnected curves to create more complex shapes.

1. Firstly create a filled shape, as described earlier.
2. Now, stay in Edit mode and 'SPACE - Add - Curve - Bezier Curve' to create a new curve on top of your existing one.
3. Manipulate the new curve so it fits inside the larger shape somehow, this will not work if the lines intersect.
4. Making sure only the new curve is selected, fill it with **CKEY**.

You should see that the space enclosed by the inner curve is no longer filled. You can continue by adding more curves inside a larger one, or even a larger one around them all.

## Extruding

[Noob: I find that this is much easier to do using Bezier circles instead.] You may have noticed you can only modify the curve in two dimensions, and now its time to explore the third dimension! Extruding is where you define a two dimensional 'profile' shape, and it is 'swept' through space to create a volume.

Edit: Bezier circles are not true circles, they are an approximations. For artistic purposes, this may not really matter, but for precision modelling only NURBS circles should be used. This is due to the math used to describe the Bezier and NURBS curves.

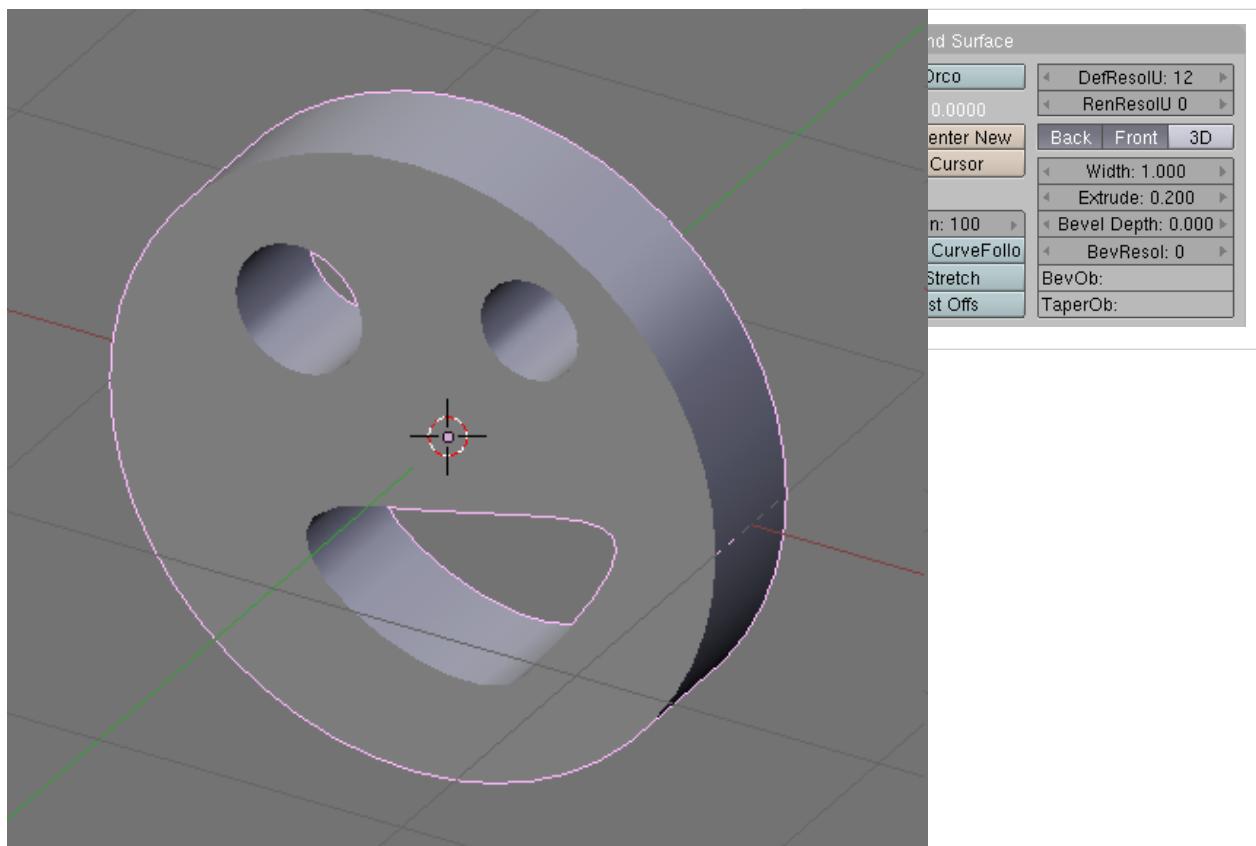
### Simple Extrude

If you want to make your curve 3d, the fastest way is to use the extrude slider. This simply extends your curve backwards by the specified distance to make a 3d object.

- **Make a curve and fill it:** Use the steps above to create something simple, and fill it in using **CKEY**.



- **Set the extrude depth:** Find the 'Curves and Surfaces' box in the Editing tab of the Buttons window. There is a slider on the right hand side called Extrude. Set the Extrude depth to something other than 0, and probably less than 1.

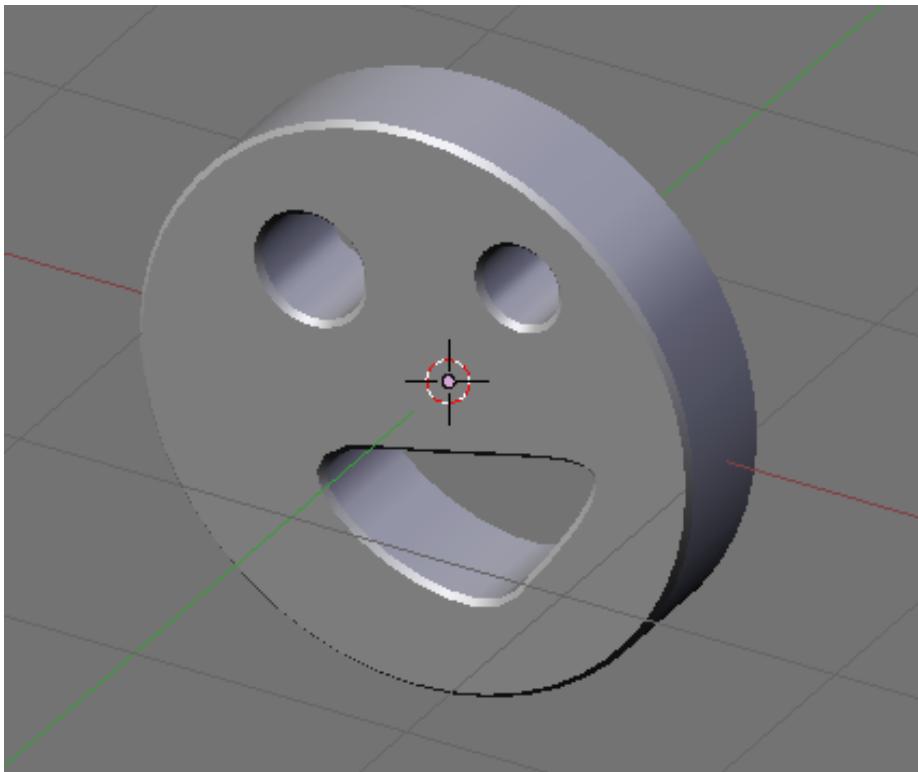


Behold, your 2d curve has transformed into a neat 3d structure. The great thing is, you can still edit the curve as if it were just 2d, and the changes will update in real time.

## Bevel

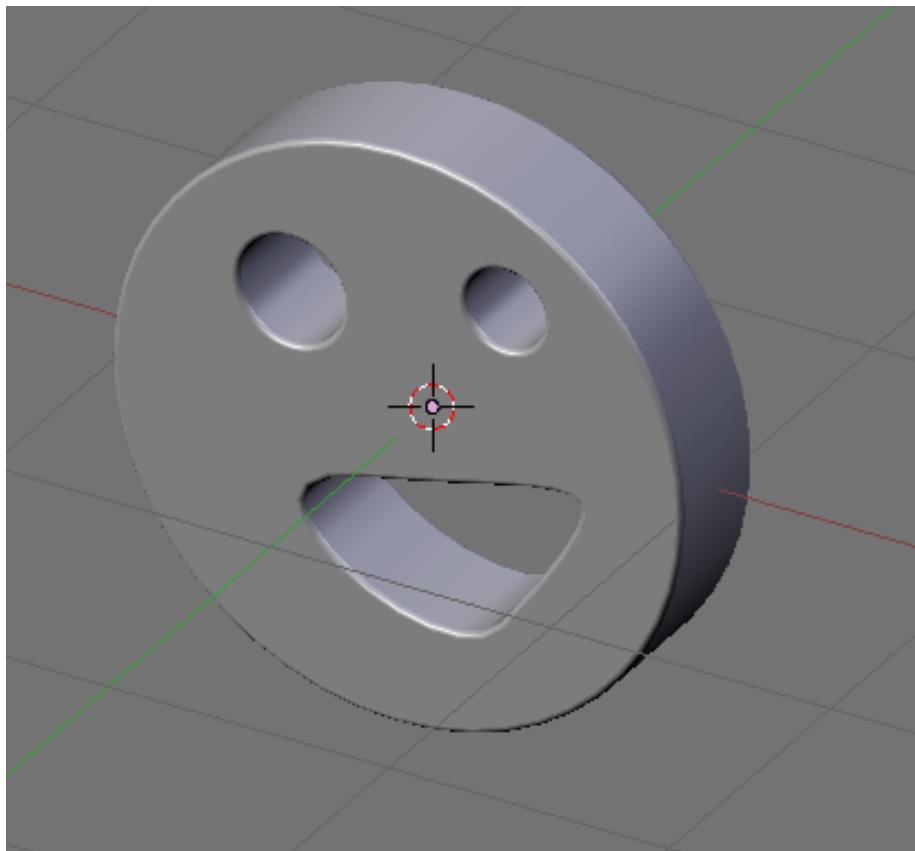
Now you have an extruded shape, you should start playing around with some of the other curve settings on offer, so here is a description of how the Bevel depth and Bevel resolution sliders work.

Try setting the Bevel depth to a small value, say 0.02. This will cut off all of the sharp edges, and give a bevelled effect all around the shape.



As you may guess, Bevel resolution decides how many times the algorithm divides up an edge. Higher values than 0 result in smooth curves rather than sharp edges, but dramatically increase the number of vertices in the shape.

Try setting the resolution to 3 or 4, you should see an effect like this:



## Bevel Objects

*[Noob: I couldn't understand what this topic was supposed to do. Could someone explain better and include images? Thanks...]/[2ND NOOB: I couldn't understand this either. Heck, I don't even have a typable field in 'Curve and Surface'... Help on this section would be extremely useful...]*

*[Noob answering: After you finish making the filled shape exit to object mode. Then add a curve and rename it to bevel or anything under the Link and Materials tap or just use the given name of the new curve. Then deselect the new curve and select your filled shape. With the filled shape selected type in the name of the new curve that you have just created in the BevOb: box. It is under the Curve and Surface tab at the lower right corner. Hope that helps :)]*

This technique allows for much more sophisticated extending of our curves into 3d. Basically, we can create a new curve which will be extruded all the way around our main shape.

1. **Create a curve:** Once again, make a filled shape like our friend the smiley. **Noob Translation:** In Object Mode, Add a bezier curve, go to edit mode and make a smiley.
2. **Create a bevel object:** Add a new curve and return to Edit mode, leaving it as the default shape. **Noob Translation:** Go back to object mode and ad a new bezier curve, then go to edit mode.
3. **Assign the bevel to the curve:** Type the name of the second curve (Default is Curve.001, or just Curve if you used a circle before) into the relevant box in the 'Curve and Surface' menu. **Noob Translation:** In edit mode (editing the second bezier curve) under the links and materials tab, on the OB: text field, change the name of the curve to bevel (the name doesn't matter though, it just makes it easy to remember). Go back to object mode and select the smiley and go to edit mode. Under the Curve and surface tab on the BevOb: text field write bevel (or the name you chose for the bevel line).

**DON'T PANIC** if you end up with a useless mess! This is expected, as we have yet to understand how the bevel object works.

Basically, the curve we created as a Bevel object is used to define the shape of the edge around our filled shape.

- The height of each part of the Bevel object above or below the local X axis is how far in front or behind the filled shape to extrude from the flat curve.
- The distance to the left or right on the local Y axis is how far towards or away from the center of the shape to pull each bit of the curve.

For our purposes, we want very little variation in the width of the bevel. This will hopefully solve the problem of all sorts of bits of the shape overlapping each other.

The default curve also starts and stops on the x-axis, we want to move the start and end points so one is higher than the other. (Preferably either side of the x-axis too, we don't want the extruded shape too far away from its curve).

Assuming you have kept the default new curve for your bevel object:

1. Move the left-most point down a whole large grid-square, and right until it is two or three small squares from the main axes.
2. Move the other point up a whole grid square, and left until it is roughly on the blue center-line.
3. Modify the control handles if you feel it's necessary.

This should leave your shape looking nice and neat, extruded quite a long way with a smooth curving edge. Experiment with the shape, see what works and what doesn't!

*[Noob: I had trouble cleaning up the 'useless mess' after assigning the bevel object to the extruded object. What I found was that I needed to keep the bevel object close to its central point (not sure what this is actually called, but it is the pink dot that you see when you select the curve). The following steps helped me resolve the 'useless mess'.*

1. If you are in object mode, change to edit mode, then type A to select the entire curve (you may need to type A twice to first deselect, and then select the curve).
  2. Now type G to grab and move the curve, move the selected curve over the pink dot that you see, and you should start seeing your extruded object look more like it was before it became the 'useless mess'.
  3. It may still need some work, so rotate (R) and scale (S) the bevel curve until the edges of your extruded curve match the shape of the bevel curve.
- NB: When rotating the bevel curve, you might notice that you can easily invert the bevel by rotating it by 180 degrees.
  - NB: The sliders for Extrude, Bevel depth and Bevel resolution are not used once a bevel object is in place.

(To come: Closed shape Bevel Objects)

## Front and Back

I would use wood working analogies to communicate what I expected to see and what I saw instead. Using a bevel object I assumed would be very much like using a Router with an Edge molding bit attached. When you run your wood piece thru your router its edge gets shaped/hollowed out. All those nice decorative edge effects, we see on furniture, and joinery aids, we may not even know exist, are generated by routers.

With that hope in mind I created a bevel curve object making sure it was smaller than the Extrusion depth (about 1/5th of it), it was an L looking curve, hoping it will hollow out my Smiley's edges. What I am seeing instead is quite surprising: The bottom 4/5th of my first filled extruded curve (my Smiley equivalent) is removed, top 1/5th remains. As if only the top surface has remained after all that work.

---

Below is some tip from one other user. I was afraid to remove it from my note in case, instead of creating a new note, I ended up modifying an existing one

---

1. **Create a filled, extruded shape:** Follow the process above, for speed just use an extrude depth.
2. **Disable Front and Back:** Click both 'Front' and 'Back' in the 'Curve and Surface' menu to disable them.

Now you should be able to see right through the curve! Cool isn't it.

- NB: You can disable just one of the faces and leave the other intact if you like.

## Blender 3D: Noob to Pro/Curves In 3d

### Bezier Curves in 3d

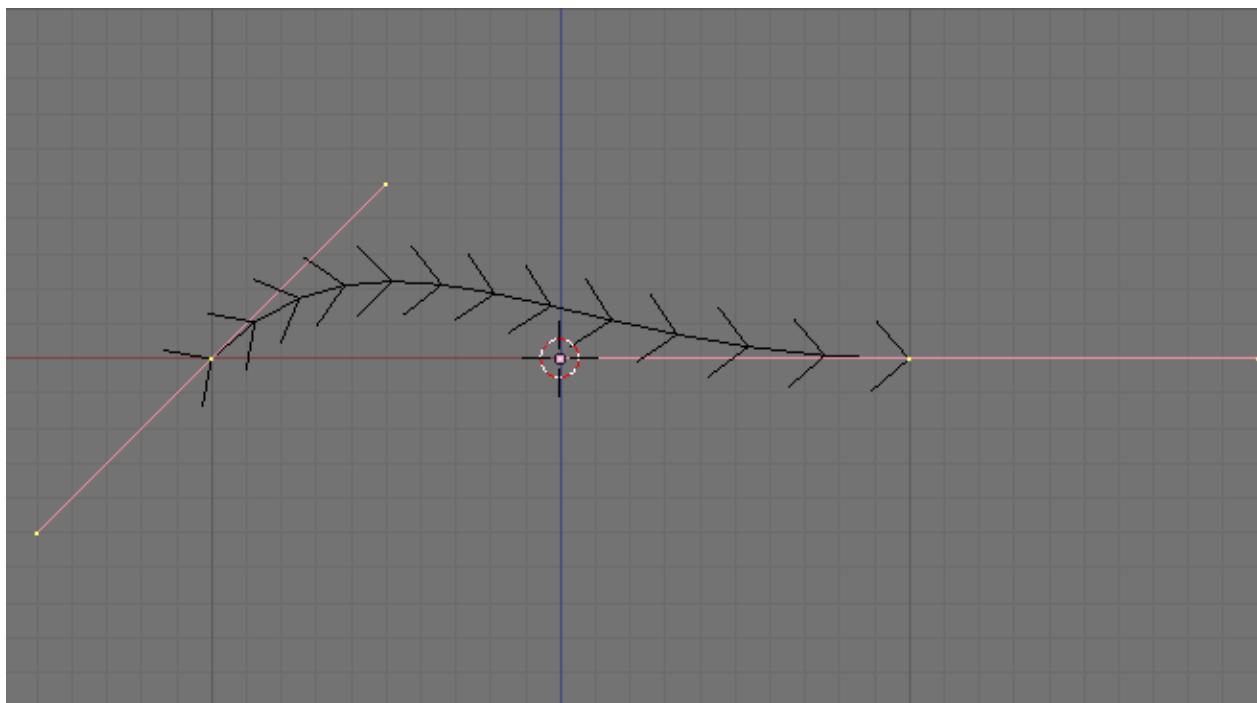
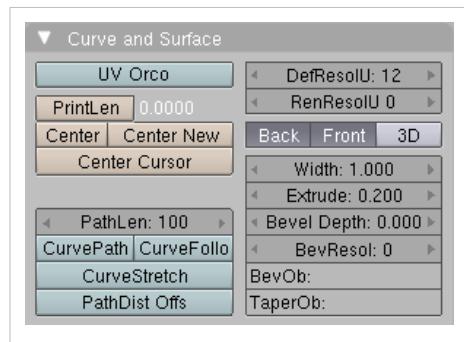
In the previous page, we learned how to create and basically manipulate Bezier curves in two dimensions, as well as extruding them into the third. Here we will be exploring the '3D' button you may have been wondering about in the 'Curve and Surface' menu and some of its implications.

1. **Create a basic curve:** Press 'SPACE - Add - Curve - Bezier Curve' in case you have forgotten.

If you have been playing around with the control points on any curves so far, you will have noticed that they cannot be moved outside the flat plane made up by the X and Y axes. This is because it hasn't been enabled yet, so go ahead and enable it...

1. **Enabling 3D editing:** Locate and press the button labelled '3D' in the 'Curve and Surface' menu.

Your curve should now look like this in Edit Mode:



Our previously simple curve has now been covered with arrows! These indicate the direction of the curve, and also the normal size and tilt. But more of that later...

## Getting started in 3D

Editing the curve should now be child's play, assuming you have already completed the tutorials on 3d meshes.

1. **Rotate the camera view:** Use the **MMB** or some Numpad keys to rotate the view so you can see the 3d axes.
2. **Move a point:** Select one of the control points (NB: A point, not a handle) and move it with **GKEY**. You may want to constrain its motion to the new Z axis using **ZKEY** twice.

If you rotate around the curve, you can see how the interpolation works in three dimensions, and that the arrows still follow its direction correctly. It is worth noting that they are all mostly vertically (for the moment at least), this will be discussed in the next section.

## Extruding revisited

I'm sure you will be delighted to learn that we can still use the Extrude option on 3d curves! Try cranking up the value, and you will see a surface created through your curve, at right angles to the 'normal' arrows.

The Bevel settings will also work as before, but you probably don't need to use them unless you are editing a 'filled' curve (**CKEY**).

Also, The Subdivided option(**W, 1**) still applies while working with curves. Just select two(or more) handles of the curve that are right next to each other, and subdivide. You should have a newly created handle(s) in between the handles you selected to subdivide.

## Bevel Objects

(Information on bevel objects to come...)

## Curve Tilting and Normal-scaling

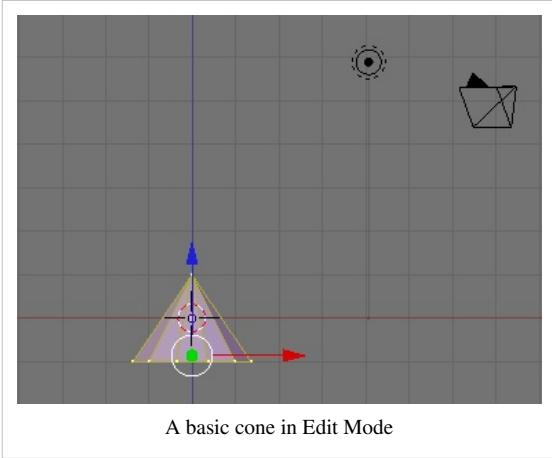
(To come...)

---

# Blender 3D: Noob to Pro/Deforming Meshes using the Curve Modifier

## Setting up your Mesh

While this example will be done with a cone primitive, you can use the default cube or another shape of your choosing so long as it can be loop subdivided along the axis you wish to curve. Delete the default cube if you're not going to use it as your base mesh, and add your chosen primitive. In this example, we'll be using a Cone.



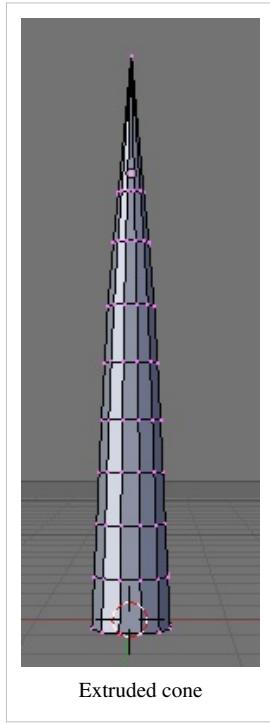
- Set your view to the X-Y Axis (Top View) by pressing **7KEY**, the 7 on the keypad.
- Press **SPACE > Add > Mesh > Cone**
- A smaller number of vertices are needed, since this cone will become just the tip of the finished shape.
- Right Click and drag to reduce the number of vertices to 12, then click the Ok button.
- Press **TAB** to enter Edit Mode, or choose Edit Mode from the bottom of the 3D viewport.
- Switch views to the Z-X Axis by pressing **1KEY** (1 on the keypad). Your screen should now look something like the picture at left.

- If your cone is selected (one or more faces are pink, one or more vertices are yellow) press **AKEY** (Select/Deselect All) until they are deselected.
- Select the point of your Cone with the **RMB** and drag it upwards with the blue arrow. You can hold down the **CONTROL** key if you want to constrain the scaling to set units. For this example, 5 squares of height were added to the cone.

## Extruding the Cone

- Still in Edit Mode, press the **NUM8** repeatedly to rotate your view to the underside of your cone. Select the center vertex of the circle, and hold down **CONTROL**, then press the **+KEY** on the keypad. This will select all of the adjacent vertices of the base. You could also select the whole cone **AKEY** and deselect the top point by holding down **SHIFT** and selecting it with the **RMB** to achieve the same effect.
- Extrude the base of the cone by pressing the **EKEY** and choosing *Region*.
- Pull the new verticies away from the cone a distance and click the **LMB** to set them in place.
- Press the **SKEY** and widen the new base of the cone a bit.
- Continue extruding the cone, widening the base after each extrusion until you have at least 5 segments. The more segments you have, the smoother your curved cone will be.
- Another way of adding segments is with the Loop Subdivide tool. **CONTROL + RKEY** to bring up the purple placement ring, then **LMB** twice to add a subdivision. ( reader note: I can't loop cut a cone (in Blender 2.48 - don't actually know why), as an alternative you can select all the edges that start from the tip of the cone (be sure to be in edge select mode) and subdivide/subdivide multi )

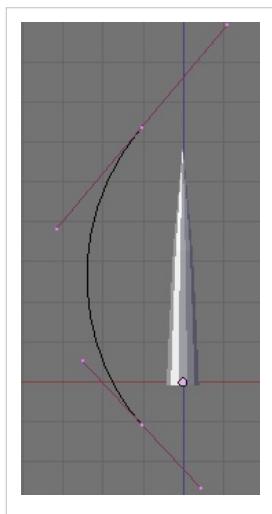
( another reader: I found that it works to make a cylinder, **Ctrl+R** it into 10-ish segments, then select the circle of vertices on the top (use the **B** key, preferably). Press **S**, use the wheel to increase the selection radius to include almost the entire cylinder, and scale them to 0 (make sure the pivot is set to "Individual Centers" - that's close to the middle of the 3D View window header). This should squeeze the top of the cylinder into a cone and compress the segments below it by smaller degrees. The segments won't be of the same height, but the idea's the same. At least it's easier than extruding and scaling ten times. )



Extruded cone

- You may find that your cone is quite large indeed now, so switch back to Object Mode (**TAB**) and resize it with the scale tool. While you're in Object Mode, try stretching the cone out a bit to make a nice long needle.

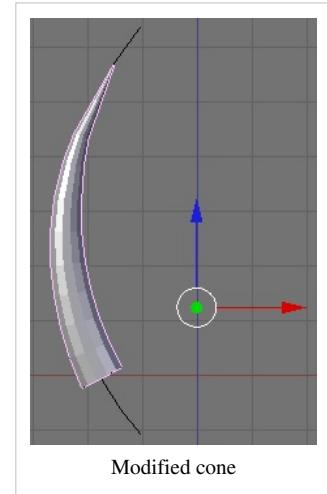
## Making a Curve



- In Object Mode, Z-X orientation, make sure you have nothing selected and press **SPACE > Add > Curve > Bezier Curve**
- Rotate your curve 90° (holding down the **CONTROL** key to make it rotate in 5° intervals) so that it lines up with your cone. Move it off to one side so that its not hidden by your cone.
- Grab the center vertex of each end and adjust its rotation (**RKEY**) until you have a nice shallow dome shape.
- Scale the curve in size (**SKEY**) until it is larger than your cone.
- Look at the panel below the 3D space, you should see the Editing panel is selected (**F9KEY**) Just below the panel buttons will be a window called Link and Materials. If your curve is selected, you will see a box with OB:Curve in it. This means the name of the curve is "Curve". This is important because that exact name is needed to deform the cone.

## Applying the Curve to the Mesh

- Select the cone in Object Mode. In the Editing panel window (**F9KEY**), the rightmost panel has two tabs, Modifiers and Shapes. Choose the Modifiers tab if it is not already selected.
- Click the *Add Modifier* button and choose *Curve* from the popup list.
- In the OB: box, type the exact name of your Bezier Curve (it will be "Curve" without the quotations if you left it as the default.)
- Notice the six buttons underneath the OB: and VGroup: boxes (in the Modifier Panel). They are **X, Y, Z, -X, -Y, -Z**. They affect which plane the curve deforms. For this example, you'll need to select the **Y** button.
- Move your Cone so that it overlays the curve in the 3D space screen, and notice how it follows the Bezier Curve.
- You can modify the Bezier Curve as well as the cone repeatedly until you are happy with the design.
- If you switch to Edit Mode, notice that the cone returns to its straight orientation.
- To apply the deformation to the mesh permanently, in the Modifiers panel, click the *Apply* button next to your Curve modifier.



Noob note: Whole tutorial worked fine and easy to follow, right up to the last step. I can't get it to deform to curve at all. :(

Noob2 note: Just the same with me. After adding the Curve modifier nothing happens.

Noob3 note: After I added a Curve modifier and pressed the "Y", the cone MOVED ALONG the Curve, not bent to it.

Noob4 note: This happened for me too, it just means you're aligning the cone on the wrong axis. Try using one of the other five (it was the x button for me, not y). also make sure that the curve is aligned to an axis, don't goof off too much in 3d view when making the curve, stay in top/side/front view.

**(Wi\_Tr Note:** In order to obtain a deformation like the one shown above, be sure you've pressed the **Y** button located at the bottom row of *Modifiers* tab. This seems to inhibit deformations through the selected axis).

Noob5 (6 months training anyway) I found the X axis button on the modifier must be pushed to display the cone like shown.

Noob6: Make sure the surface of the cone has many faces as well - if you just extruded the cone tip like I did the cone will not twist at all but only follow the path. I ended up extruding the tip out some ways and then subdividing the cone a few times in addition to setting x as the axis of deformation. It worked then.

Noob7: If you can not get it to work select your curve and look in the lower left under links and materials. You must find the *real* name of your curve, mine said "Curve.001" the .001 is important it will not work without the entire name.

Thanks for the fix... somehow that got left out! \*smacks forehead into desk\* Apologies... chalk it up to a newbie tutorial writer. I think it's fixed now... Embri (talk)

# Blender 3D: Noob to Pro/Making Your Creation Smoother

---

## Getting started

To begin, open a new scene in Blender. Let's clean up the scene a little by selecting the lamp and camera in the scene, press M to move them to a different layer and click on the fifth layer from the left to place the objects. Go back to layer 1 and delete anything else. You may also want to get rid of the grid by opening the View Properties and turning off the Grid Floor and X & Y axes.

To block out our character, we're going to use an object type that is probably the least used and useful of any known to mankind. Let's hear it for... Metaballs!

## Sculpting With "Lumps of Clay"

Go into the top view (important), press Space and add a **metaball**. Metaballs are a nifty, ancient piece of 3D technology that is useful for creating blobs. (Similar to lumps of clay, eh?) You create simple primitive shapes and scale and rotate them to block out your character's shape. When the primitives come close to one another, they "bleed into one another" in much the same way that water droplets merge when they touch. Cool.

Remember that Blender jumps out of Object mode into Edit mode when you create an object, so press Tab to switch back to Object mode. You won't be able to scale your Meta primitive non-proportionally in Edit mode. In Object mode, you can change these options for the entire Meta object, while tabbing into Edit mode gives you more options for the selected meta primitive, such as changing the type from Ball to Tube, Plane, Cube, ;etc.; (You can also make "Negative metaballs.")

Press Shift-D to duplicate the Metaball, and place it where you like. Continue blocking out your character, building enough blobs to represent the limbs or forms you will need to sculpt your masterpiece. Don't get carried away and put in too much detail at this stage: use as *few* shapes as you need. (This is supposed to be quick and fun, after all...)

## Meta-mess!

Now that you've got something that resembles what you're after, select all the Metaballs and (object mode) type Alt-C to convert it to polygons so you can actually do something with your blob. Delete any of those black rings left over from the metas and select your new polygon mesh. If you Tab into Edit mode you will see terrifying ugliness instead of nicely gridded mesh. "Surely we can't be expected to create anything useful out of this!" you shriek. Take it easy, my friend. It's time to add a *Decimate* modifier (make sure you're in object mode when doing this to see changes).

The **Decimate** modifier will do two things for us. Its primary job is to *reduce* the poly count of a mesh. A pleasant side-effect for our purposes is that it will begin to *rearrange* the topology into a more manageable heap of triangles and quads. Keep reducing the Ratio slider below 0.5 until it becomes as coarse as you can stand. You want the lowest polygon base you can have that still maintains enough detail in the limbs and shapes you made with the Metaball phase.

One thing to watch for is that this process sometimes creates holes as it does its best to simplify the mesh. I find that usually you can slightly change the Ratio to fix the problem, but if you're still finding holes, check out the "Tips & Tricks" section at the end. So as you can see, this step greatly reduces the Face Count, which will be good later. Next we need to get rid of as many of those triangles as possible. Click Apply (in the modifier panel) and Tab into Edit mode. Press A until all of the vertices are selected (turn yellow) and hit Alt-J to convert the faces from triangles to quads, which will subdivide better. Now you should have something you can work with.

## Beginning to sculpt

Right next the Modifiers panel is the **Multires** panel. A multires object has the options to add numerous levels of smooth subdivision to a mesh. While you can use the sculpt tools on any polygon or nurbs mesh, the great strength of sculpting with Multires is the ability to jump back and forth to different levels, quickly sketching out the gross form at lower levels, and adding finer detail at the higher levels. Add the Multires and Add a level; we're ready to sculpt!

In case you haven't noticed, Blender's different modes offer different tool sets and options. To make the sculpting tools available you need to be in **Sculpt** mode, accessible through the Mode drop-down in any 3D window header. You'll see two new tabs next to the multires panel, *Sculpt* and *Brush*. Sculpt has most of the options you'll need to begin shaping your mesh. Note that most of the different brushes in this panel have hot-keys which will save a lot of time(**G**=Grab, **D**=Draw, **S**=Smooth, etc.). Most important here is changing the brush size and intensity. Pressing F and dragging the mouse will resize the brush, while Shift-F will allow you to adjust the Brush Intensity. You can also turn on **Symmetry** to paint, for instance, both sides of a face at one time. This can speed up tasks tremendously, as long as your mesh is aligned to the axis properly.

*Noob Note: If nothing seems to be happening while you're trying to sculpt your mesh, it's because you haven't applied your Decimate and/or Multires. Those both have to be locked down and applied before you're actually allowed to do anything to your mesh.*

# Blender 3D: Noob to Pro/Materials and Textures

---

## From Red to Blue and Everything In Between!

Materials and textures are nearly as important to any great 3D Design as the model itself. Without the coloured design, or the vibrant texture, to whom will an animation targeted towards children appeal? How will that "Grassy-Hill Scene" look without the vibrant greens, gentle blues, and dazzling colors reflecting off a clear, shining pond? It would be a fairly boring scene if all your objects were.....gray.

This is why it is very important that you learn to use the Blender 3D Material-Windows as much as possible. The vast scope of functionality and customization that the material system in Blender offers will allow you to create the most dynamic scene possible. The next few tutorials will have you go over the entire spectrum of options in Blender's material system. Once you finish them, the tutorials after that will help you develop your skills and imprint the system into your mind until it is almost second nature.

# Blender 3D: Noob to Pro/Quickie Material

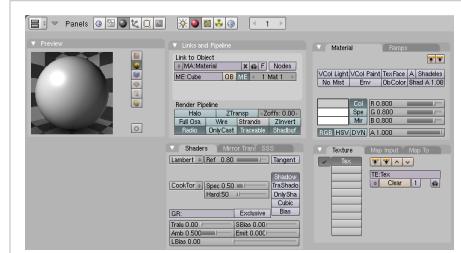
*This tutorial was created using Blender v2.49*

## Your First Material

If you open the default scene in Blender, the cube has a material already. All material settings are made in the *Shading* buttons, the basic material settings are done in the *Material* buttons.

The default material has a simple grey color. It is linked to the *Mesh* (*ME* button), not to the object directly. You can see that the mesh has only one material in use, and that you are editing this material (*1 Mat*).

Delete a material



**Image 1:** The *Material* buttons in the *Shading* context

Click the next to MA:Material in the Links and Pipelines panel to delete the link to the datablock. This removes the material from the mesh, 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.

Apply an existing material

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 mesh or object in the scene. Choose the one called "O Material" (which is the same material that was originally on the cube). The "0" in front shows that zero users are using this material.

**Note:** Unused materials in the Material List (those with a 0 in front of them) are deleted once a Blend project is saved and reopened. Hence, if you wish to delete a material from the Material List... save and re-open the project.

## Meaningful Names

"Material" isn't a very creative name for a material. What is worse than the lack of creativity is the difficulty of finding a specific material in a large scene using dozens of materials with names like "Material.001, 002, 003 ...". There are a couple of different ways to rename the link to a material.

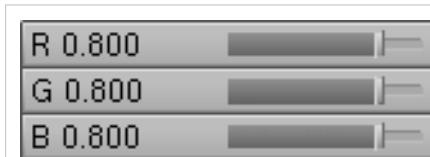
1. Press the automatic name button that looks like a little car . This will automatically give your material the name of the color currently assigned to it. For example: if the color of the material is currently grey, which is the default color, the word grey will appear in the material name space. If you have changed the color, the name of your chosen color will appear. This is a quick option for when you don't have time to give the material a unique name.
2. Press **LMB** over the material name, and the existing name will be highlighted. You can now type a new name for the material, or Press **LMB** again to place the text cursor and add to the existing name. Rename the material to "Green Ooze".

Note: Your materials will be much easier to find, and manage later when you give them brief, descriptive names you can recognize at a glance.

## 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 set a color with the *RGB* color sliders (**Img. 2a**).
- You can also **LMB** click on the number and type in the value directly.



**Image 2a:** RGB color sliders

- The most comfortable way to change the color is to **LMB** click on the color itself (**Img. 2b**), where you get a full fledged color selection panel including a sample *Pipette* to choose colors from any Blender window, including the rendering window (**Img. 2c, 5**).
- Set the **R** value to 0.149, the **G** value to 1.000, and the **B** value to 0.446.

**R**, **G** and **B** of course stand for red, green and blue respectively. By mixing these values, any color can be achieved. The *Col* value is the basic color of the material. *Spec(ularity)* is the faked reflection of a light source (like a lamp bulb) on an object. *Mir* is the mirror color for true reflections.

Physically most materials don't change the color of reflected light, so *Spec* and *Mir* may normally be left at their default values (white). A notable and important exception is metals. They do change the color of reflected light. We're not doing a physics course here, so we're going to set the *Spec* color to what we like.

- Click the button marked *Spec*.

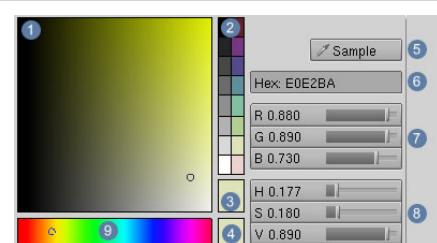
The color dialog can now be used to adjust the specular color. Keep your eyes on the Preview of the material and start messing with the **R**, **G**, and **B** sliders.

- Set the specular color to **R** = 0.640, **G** = 0.990, and **B** = 0.566.

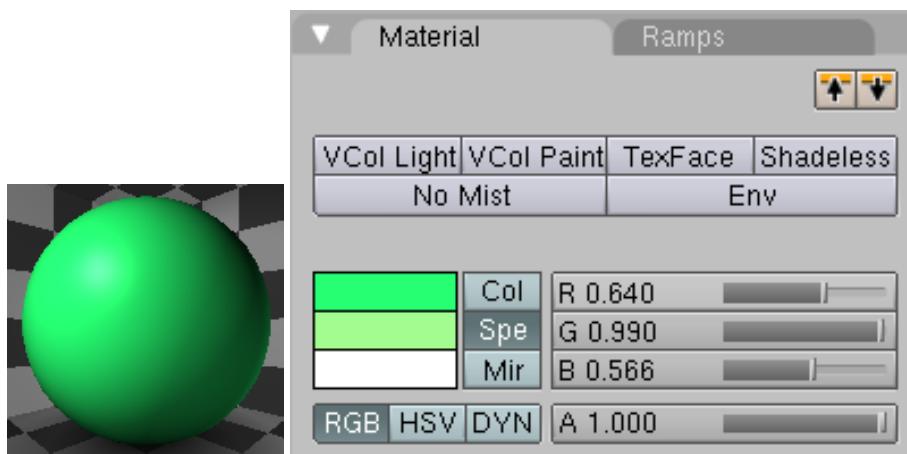
With this value we should be able to get a good ooze down the road. The colour preview and settings should now look similar to this:



**Image 2b:** Color field



**Image 2c:** Color selection panel



There are a lot of other material buttons. We will discuss some of them eventually. For examples on how to achieve a certain effect see Every Material Known to Man.

- Keep this file open and go to the next tutorial, where we will perfect the ooze.

## Multiple Materials

How to Assign A Different Material To Each Face Of A Cube: <http://www.youtube.com/watch?v=hCYViRJFf5w>

# Blender 3D: Noob to Pro/Quickie Texture

*This tutorial was created using Blender v2.49*

**Textures** are laid on top of materials to give them complicated colors 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. This tutorial uses the file from the previous tutorial. If you didn't do it before, go back and do it now.

A texture may be an image or a computed function. What the texture does and how it is mapped onto your object is set in the material buttons. Some commonly used texture types are shown on the page [Using Textures](#).

## Adding a texture

The file we're working with has a texture already. However, if you **add a new material** to an object, it has no texture by default.

To **add a texture**:

- In the *Texture* panel of the *Material Buttons* click on *Add New* to add a new texture, or select an existing one with the Drop-Down button.
- Select the *Texture Buttons* with the spotty square icon  (or by pressing F6).
- Select the texture type *Clouds* from the drop-down list. You can also change the texture's name, as we have done for the material.

A preview appears, as well as some parameters to experiment with. A *Clouds* texture provides some irregularities.

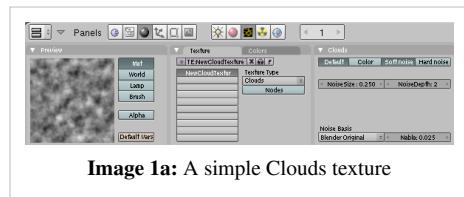
- Head back to the *Material Buttons* (Click the red sphere or press F5) and a colored preview of the texture appears. It is purple! All new textures default to this colour.

On the right hand side of the material buttons window there are three tabs: *Texture*, *Map Input*, and *Map To*. Since we want to change a material property that is affected by the texture we have to look at the *Map To* tab. This means: map the value that a texture provides to a material property. A texture may provide different values:

1. RGB color (all images, *Magic*, every texture with a colorband)
2. Intensity, either as grey scale or/and an alpha value (most of the procedural textures, image textures with alpha, textures with a colorband)
3. Normal values (*Stucci*, normal maps)

If you want to use textures you always have to be aware of the value a texture provides and the *Map To* settings for the material.

- Select the *Map To* panel.



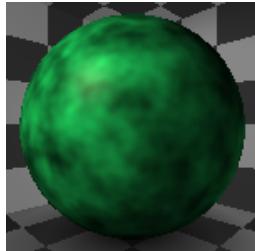
The activated *Col* button in the top row shows that the texture affects the color of the material. A *Clouds* texture provides an intensity value, ranging from 0 (where the texture is black) to 1 (where the texture is white).

The RGB (Red, Green, Blue) sliders here adjust the target color of the texture. This is mixed (*Mix*) with the material color, where the intensity of the *Clouds* texture is 1 only the target color is used, where the intensity of the *Clouds* texture is 0, only the material color is used.

- Set the target color to black.

Procedural textures are not shown in the 3D window (only if you would use excessive amounts of vertices), even if you use GLSL materials. This means that we have to render the image to see the texture properly.

- Render the image (**F12**), or use the preview to judge the result of the texture.



Next we will add a *Stucci* texture to make our clouds look bumpy.

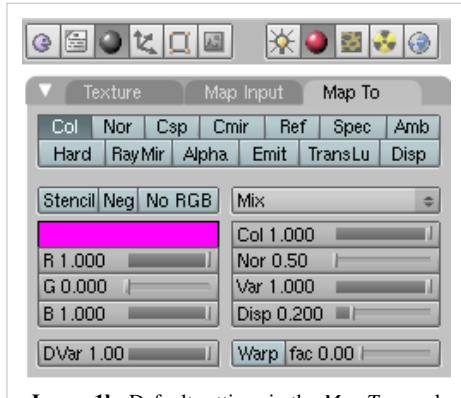
## Adding a Stucci texture

- Go back to the *Texture Buttons* and select the next texture channel (one of the blank buttons under "NewCloudTexture").
- Add a new texture here and set the type to *Stucci*.
- Back to the *Material Buttons*, 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, i.e. 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.

Render to see the effect. A texture changing the surface normal is called a "Bump" map or "Normal" map. Since that is fake 3D, you don't see it under every circumstance, you will get a greater effect on smoothly curved surfaces with high specularity, only a little or no effect on flat surfaces with low specularity.



**Image 1b:** Default settings in the *Map To* panel

# Blender 3D: Noob to Pro/Procedural Textures

---

## Procedural Textures

Texturing objects can be broken down into two categories: procedural and image texturing. Procedural texturing makes use of mathematical formulas to generate textures. This is nice because it can be used to make relatively nice looking textures without external images which are very temperamental where you put them. Procedural Textures are all stored in the .blend file. These textures are obviously generated within Blender itself. Image texturing uses images created or captured outside of Blender, either from an image manipulation program such as the Paint.NET GIMP or Photoshop, or captured on a camera. We have already learned about procedural texturing, so move on to the next tutorial to learn about image texturing.

## Current Procedural Textures

Blender currently supports many procedural textures, including: Clouds, Marble, Stucci, Wood, Magic, Blend, Noise, Musgrave, Voronoi and DistortedNoise.

# Blender 3D: Noob to Pro/Halo Materials

---

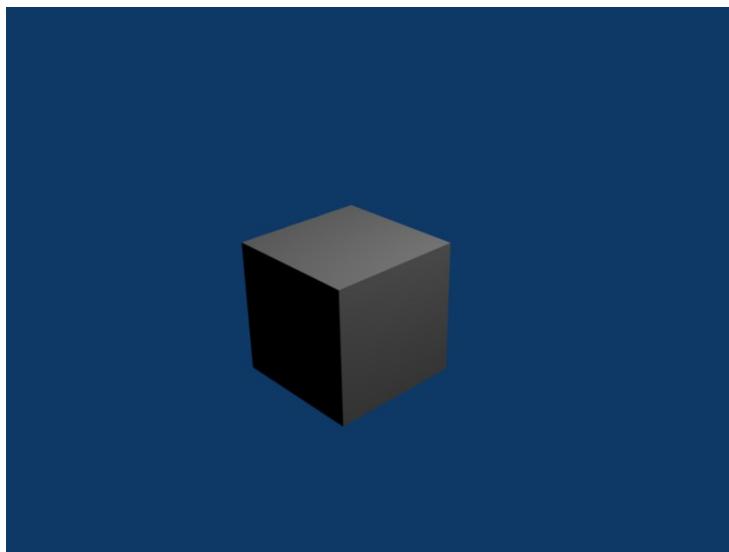
*This tutorial was designed for use with Blender v2.48.*

## Introduction

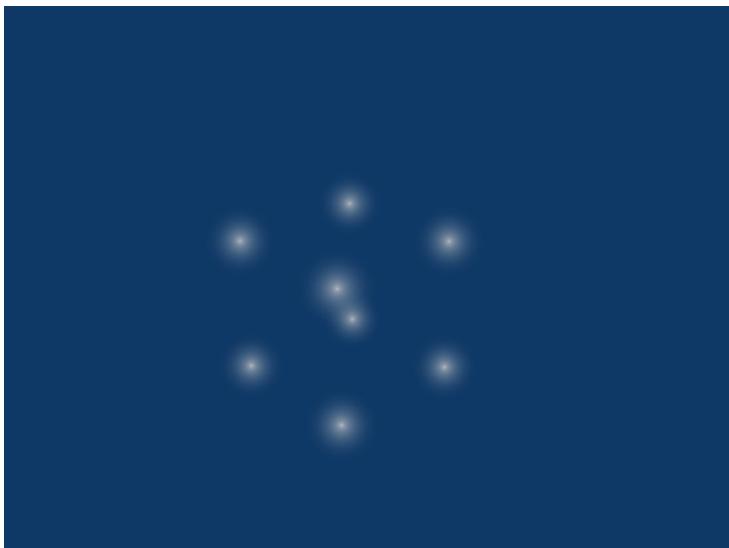
Halos are a neat effect that you can use in Blender. Instead of affecting the color/texture of a *face*, like normal materials do, when you use a halo material, it affects the *Vertices*. The particles with *Point* Visualization are rendered as Halos also.

Here is an example.

This is the standard Blender start file rendered as per normal, with a regular material:



This is the same scene, rendered with a basic Halo Material:

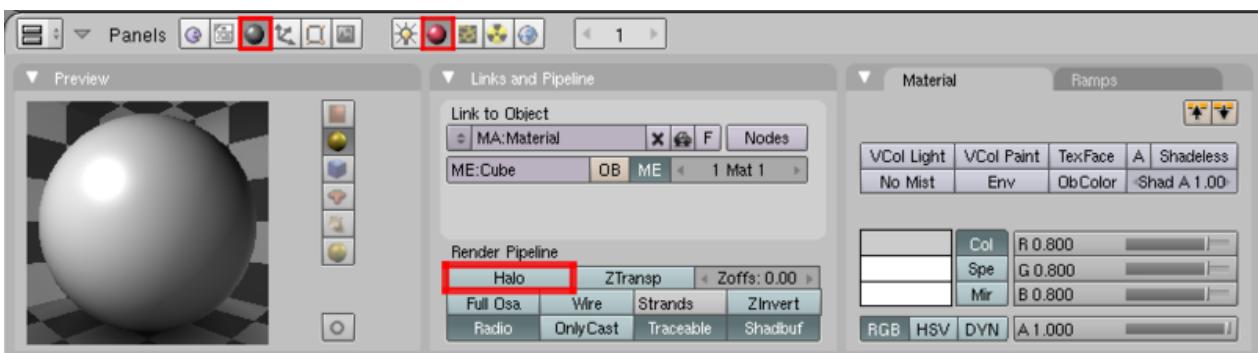


As you can see, it gives the vertices a 'star'-like appearance, and can be used to make some neat effects.

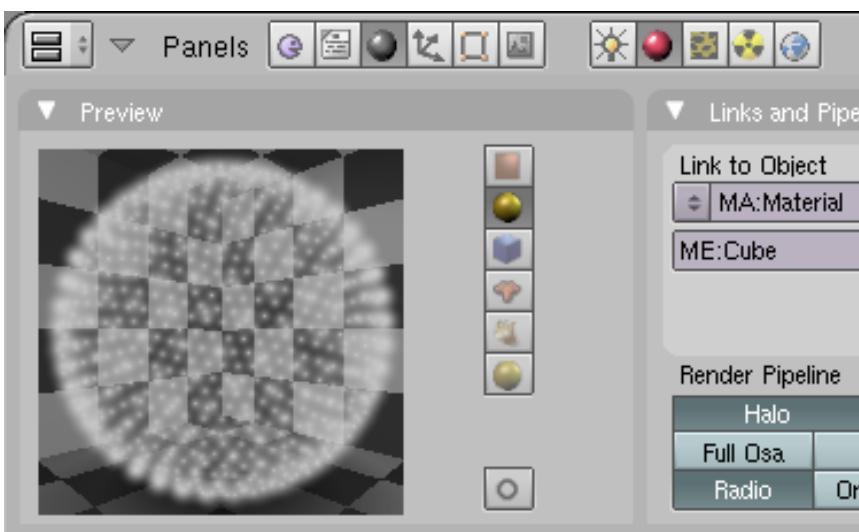
## Applying a Halo Material

By now you should know how to basically use the materials panel. If not, please complete the **Quickie Material** tutorial.

To apply a halo effect, simply click the Halo button in the Links and Pipeline panel. The buttons highlighted below are what you need to click to get to this panel (plus the Halo button)



The Preview panel will now show the halo effect on the sphere:

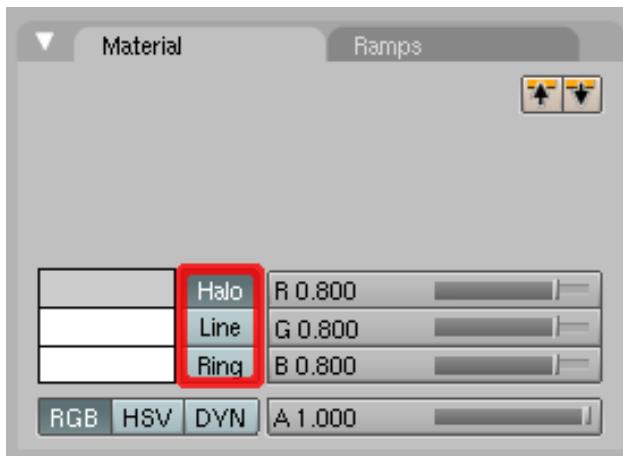


It's as simple as that!

## Adding/Modifying extra effects

Now you know how to apply halos, let's look at some of the settings for them.

Note that when you clicked the Halo button, the color panel to the right changed slightly

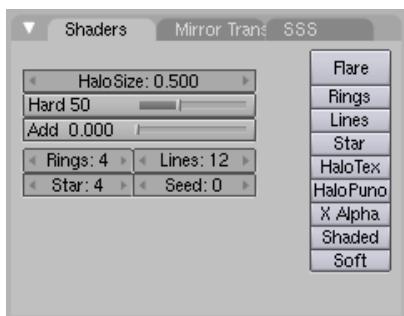


These settings will now allow you to change the colors of various aspects of the halos. Have a mess around with the Halo color and you'll see that it affects what you currently see in the preview panel. The other 2 buttons, **Line** and **Ring**, will affect other elements of the Halo we'll look at now.

Look over to the right of the bottom panels, and locate the 'Shaders' tab. By default, it should be next to the 'Mirror Transp' tab.



You can now see all the various options for the halos:



You can easily modify the size of the halos with the **HaloSize** box and the 'Hard' slider modifies the 'fuzziness' of the halos. The 'Add' slider controls how much the halo is 'added' to whatever is behind it - this is more used for textures to control how much of the texture you want to appear on the objects. An easy way to think of the 'Add' slider is as a transparency for all of the halos.

Here is a run-down on the buttons to the right:

### Flare

Adds lens flare - a technique which can add a sense of realism to an animation. By default it seems to just make the halos seem brighter, but if you modify some of the settings that appear down the bottom you can get some pretty wacky effects, especially if you have the verticies moving around in an animation.

### Rings

Adds rings around the halos. Number of rings can be modified in the 'Rings' number panel.

### Lines

Adds lines coming off the center. Number of lines can be modified in the 'Lines' number panel.

### Star

Gives the halos a star shape. The amount of points on the star can be changed to the right in the 'Star' number panel.

### HaloTex

Allows the Halos to be rendered with a texture. This can create some very cool effects

### HaloPuno

Lets the verticies' normals specify the size of the halos. Only becomes realy obvious on complicated meshes.

For a good example, have a look at what it does to Suzanne the monkey

### X Alpha

intensifies the transparency effect of the halos.

### Shaded

Allows the Halos to be shaded and their color affected by light sources.

### Soft

Softens the Halos

You can now see where the other color selectors from before work. They let you modify the color of the lines and circles as well.



# Blender 3D: Noob to Pro/Image Textures

---

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 <sup>[1]</sup> 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.
- The GIMP <sup>[2]</sup> GIMP, a free image editor that has many of the same functions as Photoshop.
- MapZone <sup>[3]</sup> A free utility for Windows (works perfectly in Wine <sup>[4]</sup>) that generates node based procedural texture maps. Mapzone can export diffuse, normal and alpha texture maps as standard image files. It can also import SVG regions created with Blender's UV mapping tools.

## References

- [1] [http://www.spiralgraphics.biz/ww\\_overview.htm](http://www.spiralgraphics.biz/ww_overview.htm)
- [2] <http://www.gimp.org>
- [3] <http://www.mapzoneeditor.com/>
- [4] <http://winehq.org/>

# Blender 3D: Noob to Pro/Creating Basic Seawater

---

*[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.]*

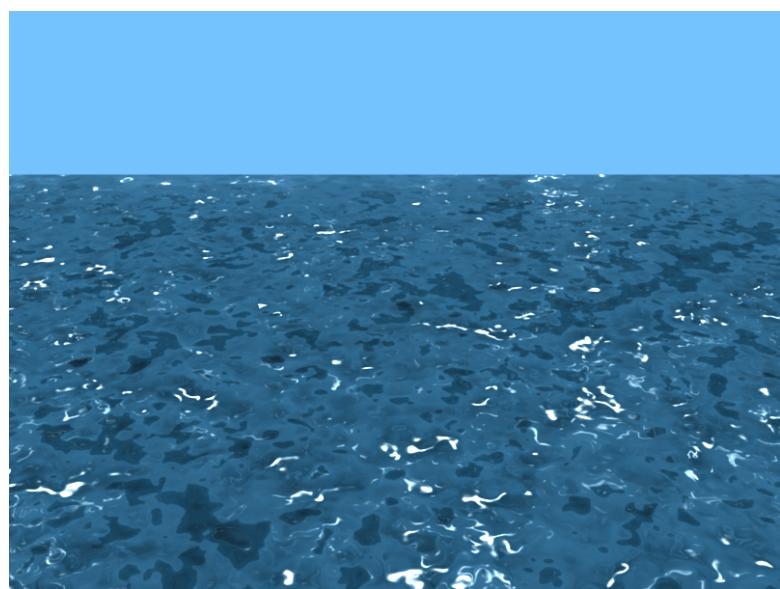
75% of the Earth's surface is covered with water. In homage to this great fact, we will develop your materials skills first by creating basic seawater.

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 20 its original size with the SKEY the way you've already learned in one of the earlier tutorials. Go to the side view (NumPad 3) and duplicate this plane two more times using Alt-D (not Shift-D), moving the plane down on the Z axis by two grid spaces each duplicate. This will make the transparency of the water more realistic once we set it.

Now off to the actual texturing work. Select any one of the planes and press F5 to bring up the Material Buttons in the Buttons Window. You will probably find two new small windows appearing here: one called Links and Pipelines 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 (0.100, 0.310, 0.435). Find the tab that reads 'Mirror Transp' and click it. Click on 'Ray Mirror' and 'Ray Transp'. For the "Ray Mirror" box, move the 'RayMir' slider to 0.3, the 'Fresnel' slider to 2.5 and the Depth to 5. For the 'Ray Transp' box, move the 'IOR' slider to 1.33, the 'Fresnel' slider to 2.0 and the Depth to 5. This will give the water realistic transparency and reflection. Also click the 'Shaders' tab, change 'CookTorr' to 'Blinn', move 'Spec' to 2.000, 'Hard' to 180 and 'Refr' to 10.000. This will make the water look more glossy.



The final rendered image

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 (F6) and select 'Clouds'. On the Clouds tab change 'NoiseSize' to 0.050. Our Waves texture is ready; next, we will refine how it is applied to our Seawater material.

Noise Size increases the size of the noise, in this case, the clouds. Soft Noise blends the intensities and reduces the contrast. Makes a mellow effect, like soft waves. Hard noise creates a high contrast, and brings out individual 'shapes'.

If you want to add more detail to your water, add another texture and rename it to "LargeWaves". Make it a cloud texture like the previous one, but make it's 'NoiseSize' 0.300 and use 'Hard noise'.

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). Click the 'Col' button so it is not selected, this button will show any color in the texture which we do not want. Find the 'Nor' Slider and move its value to about 5.00.

If you created the "LargeWaves" texture, select the "LargeWaves" texture under 'Texture and Input', go to the 'MapTo' tab, deselect 'Col', select 'Nor' and move the 'Nor' slider to 7.00. Do not select 'Hard' or 'Spec' this time.

For lighting press Space > Add > Lamp > Sun. You shouldn't need to move the sun or change any of it's settings. Finally move the camera to the edge of the plain and move it up towards the sky a bit.

Go to the Scene tab (F10), and look for the six buttons next to the big render button. Deselect all these, leaving only the 'Ray' button selected. This will tell Blender not to render some features in our scene that we really don't need. Go and press F12 to render the water, it may take a while depending on your system.

Admire your water, and maybe drink a tall glass of something refreshing!

## Extra Practice

This tutorial might also help you make even more realistic water: Link <sup>[1]</sup>

## References

[1] <http://forum.nystic.com/viewtopic.php?f=34&t=1445>

# Blender 3D: Noob to Pro/Mountains Out Of Molehills 2

---

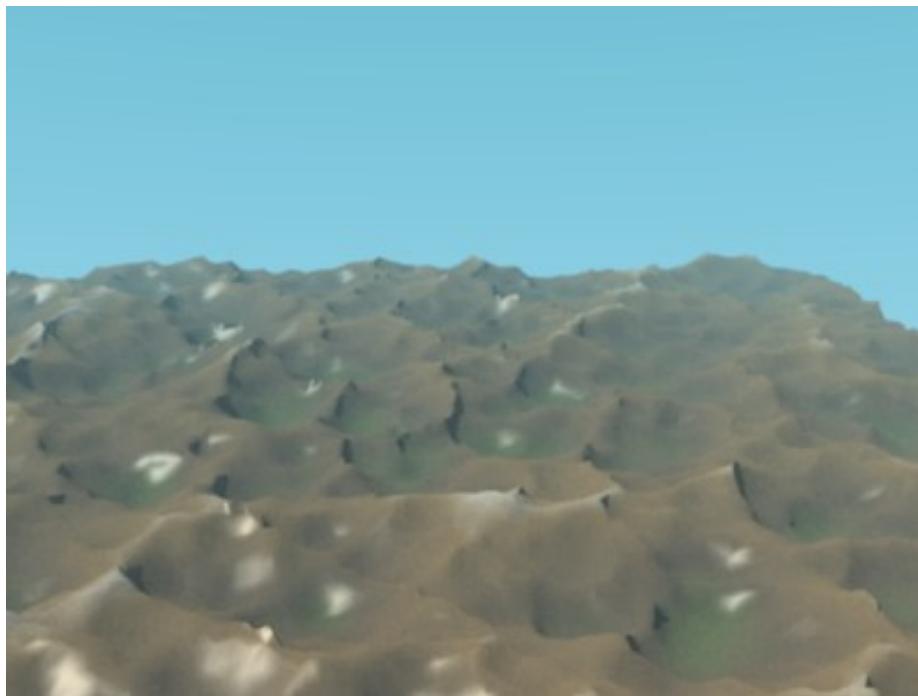
This tutorial shows you how to use displacement mapping to make a simple environment.

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. Uncheck Col and check Disp on and set the Disp slider to a value like 0.200
11. Set camera and a few lights to the scene.
12. Render.

[noob note: I just got a flat gray plane when i did this, I believe you have to press Nor on and increase the Nor value to something relatively high, I used 11.37.]

(another noob's note: In my current version (2.46) I don't think Nor is what you want. Nor will affect the bump-mapping effect (normal values). You want to have Nor unchecked, Disp checked and the slider for Disp set to something like 0.200. Though I'm not clear on the difference between Nor and Disp.)

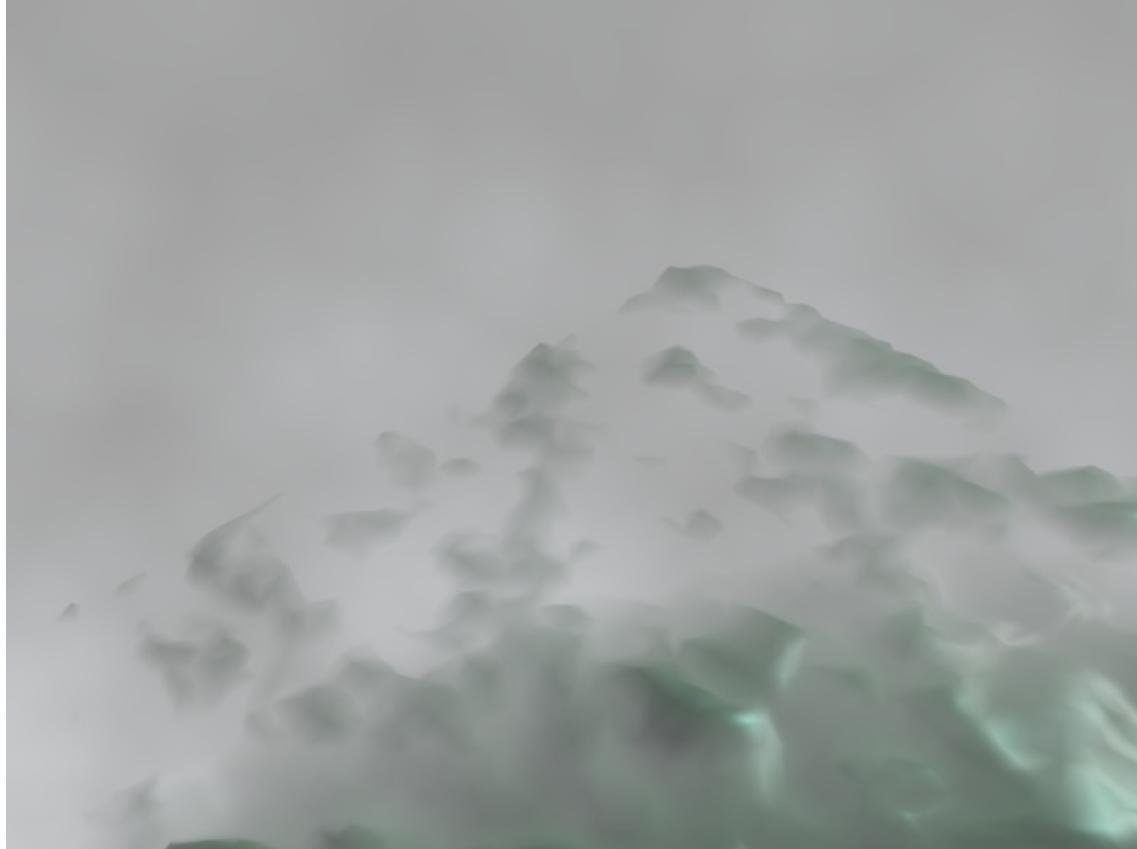
Nor (normal) only changes what direction the point on the plane looks like it's facing. Disp actually moves the points on the plane up and down



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.

By adding some Mist in the World buttons, adding a Cloud Texture to the World and make it blend from white to gray we could get something like this:



# Blender 3D: Noob to Pro/Basic Carpet Texture

## Goal

I am using a basic scene that I quickly set up before I started to create the carpet material. It shows a monkey (suzanne), a plane, camera, 3 area lamps and 1 spot lamp.



The purpose of this tutorial, is to highlight the power of blenders built in shaders and procedural textures to create a carpet material to use in your scenes.

**NOTE:** For those of you needing help getting a similar scene to the one above, here are some axis positions, etc to help out:

**(Spot-Lamp)** **X=1.62, Y=0.86, Z=6.74; rotation-X=37.26, Y=3.16, Z=181.34;**

**(Area-Lamp-1)** **X=4, Y=3.27, Z=4.12; rotation-X=54.67, Y=-18.59, Z=-109.47;**

**(Area-Lamp-2)** **X=-2.07, Y=-2.08, Z=4.85; rotation-X=29.37, Y=-28.98, Z=355;**

**(Area-Lamp-3)** **X=0.315, Y=-2.89, Z=4.29; rotation-X=49.23, Y=-10.63, Z=6.68;**

**(Monkey)** **X=0.05, Y=0, Z=0.42; rotation-X=58.61, Y=-16.07, Z=23.245; DIM(dimensions) X=2.734, Y=1.9**

**(Plane)** **x=0, y=0, z=0; (no rotation); Dimensions: X&Y=14.30, z=0.**

NewbyNote: With these settings, I get a big white-out.

## The Basic Material and Shader Settings

For the basic material for the carpet set the colour settings of your material as follows -

Col (R 0.714) (G 0.134) (B .134) Dark Red

Spe (R 0.590) (G 0.210) (B 0.084) Redish Brown

Mir (R 1.000) (G 1.000) (B 1.000) White

Change the specular shader to ('phong') and leave the default lambert diffuse shader as it is. Change the ('spec') to [0.13] and the ('hard') to [12]. Lastly click the ('Full Osa') button to enable it. If you render now you will notice that the plane looks like a ugly pastel colour (if not, you have a different lighting setup to mine and the shader will not look the same as the images in this tutorial.) Dont worry about this ugly looking plane it will soon be a beautiful carpet.

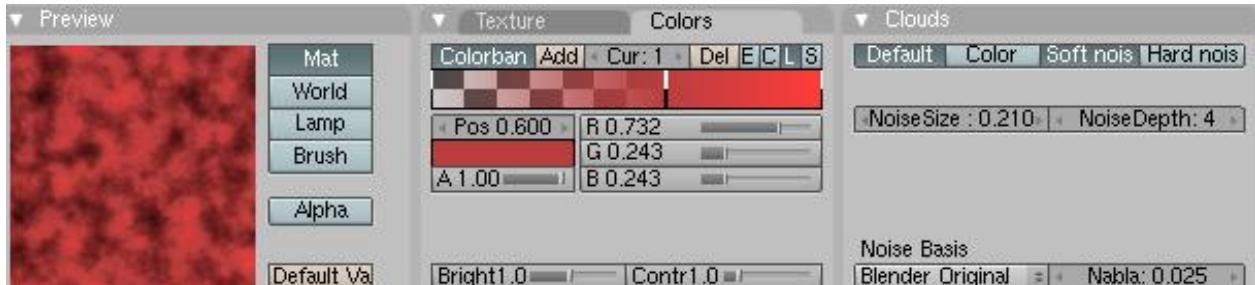




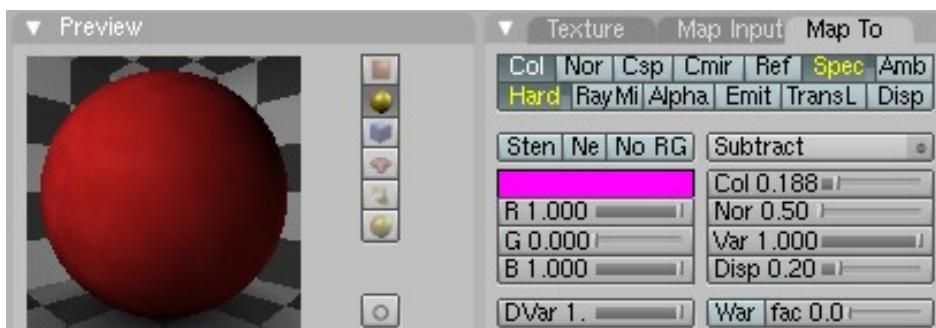
## Cloud Texture 1

Press 'F6' on the keyboard to bring up the texture panel. Click the bottommost of the long boxes to create a texture in the bottom channel. Create a new texture and rename it something like 'Red Clouds 1'. From the ('Texture Type') pull down select clouds. In the 'Clouds' settings panel change ('NoiseSize') to [0.210] and ('NoiseDepth') to [4].

Now select the Colors tab which will bring up the ColorBand for the texture. Press the orange ('Add') button to add a cursor on the colourband. Next make sure the 'Cur : 0' is showing next to the add button and change the colours as follows - (R 0.770, G 0.168, 0.168). Now click on right side of the 'Cur : 0' so it shows 'Cur : 1'. Change 'Pos' to 0.6. Set Alpha to 1 and change the colour to (R 0.732 G 0.243 B 0.243).



Now go back into the material settings and change the settings in the Map To tab as follows. Click ('Spec') twice so the text becomes yellow do the same for ('Hard'). Now select 'Subtract' for the texture blending mode. Change ('Col') to [0.188].



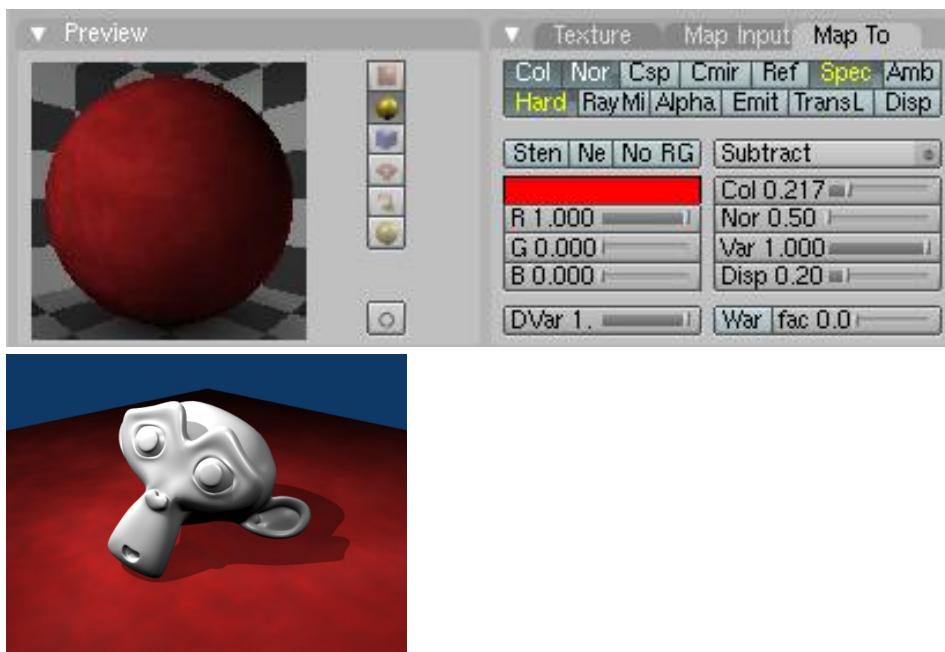
At this stage if you render you might find it hard to notice the difference, it is apparent only where there is low light levels on the plane, don't worry about this at the moment. It means if you render with Ambiant Occlusion you get a nice carpet effect of slightly varying colour.

## Cloud Texture 2

Go back to the texture panel and create another texture, call it 'Clouds' and put it in the channel above 'Red Clouds'. Change ('NoiseSize') to [0.054] and ('NoiseDepth') to 4. Select 'Improved Perlin' from the Noise Basis pull down. Finally change ('Nabla') to [0.031]. Do not change any more settings here.



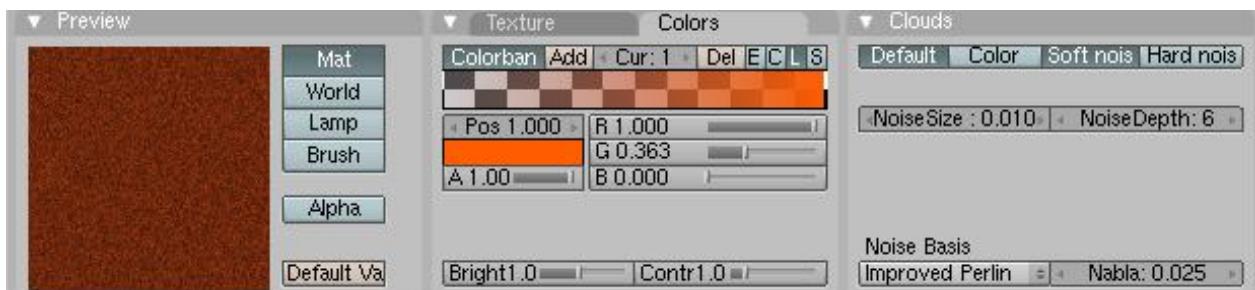
Now in the material panel, under the Map To tab change the following - Click ('Nor'). Click ('Spec') and ('Hard') twice so as they are yellow.



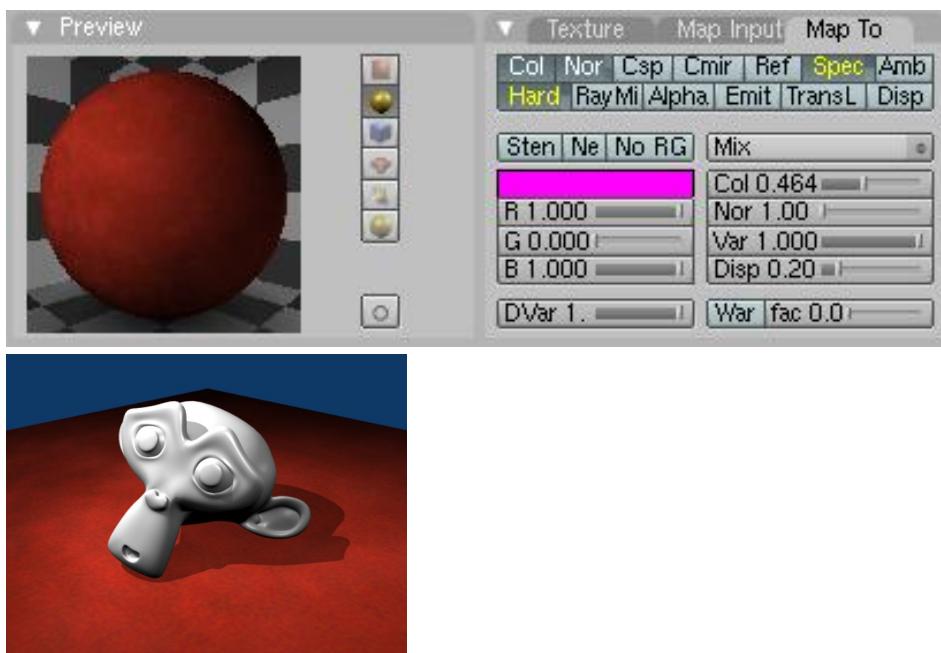
As you can see the material is starting to look a bit better, only 2 more textures to go.

## The Final Cloud Texture

Switch to the texture panel once again and create a new Clouds texture in the next channel up. Change ('NoiseSize') to [0.010] and ('NoiseDepth') to [6]. Now click on the 'Colors' tab and change the colour of the left cursor ('Cur : 0') to (R 0.713 G 0.262 B 0.223) and Alpha to 0. Switch to Cursor 1 ('Cur : 1') and its colour settings (R 1.000 G 0.363 B 0.000) and Alpha to 1.

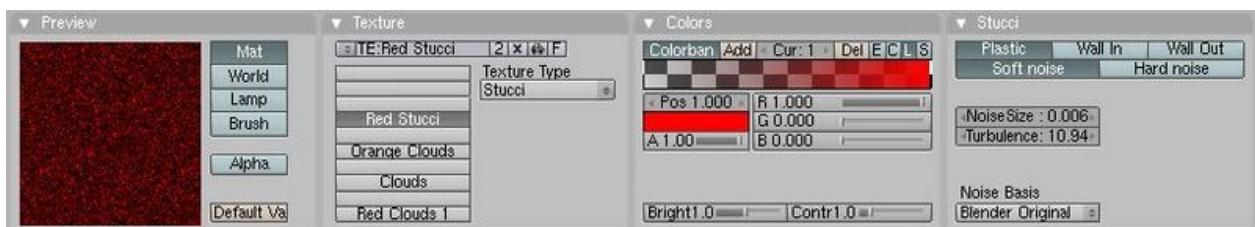


Now in the materials panel under the 'Map To' tab Click ('Nor'). Click ('Spec') and ('Hard') twice so as they are yellow. Leave the blending mode as 'Mix'. Change ('Col') to [0.464] and ('Nor') to [1.00].

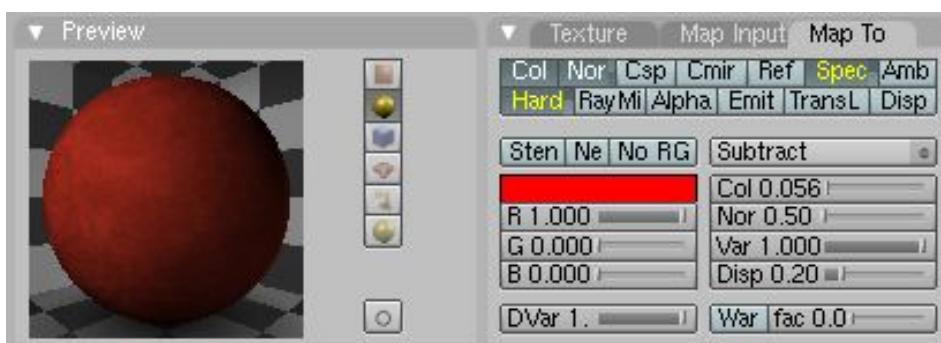


## The Last Texture

Finally, go to the textures panel one last time. Create a new Stucci texture in the next channel up. Change ('NoiseSize') to [0.006] and ('Turbulence') to [10.94]. Click on the 'Colors' tab. The first cursor 'Cur : 0' should be black with alpha 0. 'Cur : 1' should be red (R 1.000 G 0.000 B 0.000) with alpha 1.



Now go to the materials panel. Under the 'Map To' tab Click ('Nor'). Click ('Spec') and ('Hard') twice. Change the Blending type to 'Subtract'. Change ('Col') to [0.056] and ('Nor') to [0.50]. And that's it. When you render now, you should have a nice-looking carpet material. By tweaking with the colours you can create any colour of carpet.





## Blender 3D: Noob to Pro/The Rusty Ball

---

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 (**XKEY**).
- Make an object you want to have the image on (I recommend a Mesh plane, sphere or tube).

(If you are making a Mesh Plane, change your view to above, by pressing **NUM7**)

To make a rusty ball I suggest an icosphere with a size of 2 which will fit nicely in the camera view.

- Sequence is: spacebar, add, mesh, icosphere.
- Go to the materials (**F5**) and select "O Material" from the popupmenu by the "Add new" button. (the little arrows)
- Now go to the textures (**F6**) and choose "Texture Type" A drop down (or up) menu appears and you will get many options, the one we need is "Load image". Click it, then select "Load" and navigate to an image you want to use. (Note: JPGs, PNGs or TGAs are recommended for Blender. Bitmaps tend to get all screwy.)

(**Noob Note:** Do a search for **Rust Textures** <sup>[1]</sup> if you do not have one handy.)

- 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. You can see a simple preview of the different options in the "Preview" window and try the different modes.
- 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!

## References

[1] [http://images.google.com/images?sourceid=navclient&rlz=1T4GGLR\\_enUS262US264&q=rust%20texture&um=1&ie=UTF-8&sa=N&hl=en&tab=wi](http://images.google.com/images?sourceid=navclient&rlz=1T4GGLR_enUS262US264&q=rust%20texture&um=1&ie=UTF-8&sa=N&hl=en&tab=wi)

# Blender 3D: Noob to Pro/Procedural Eyeball

---

Level: intermediate

Version: 2.43

### Building a better (procedural) eyeball!

Originally created by Jon McKay (Ammusionist) and posted on [blenderartists.org/forum](http://blenderartists.org/forum)

[1] The end result of this tutorial is an eyeball that fulfills the following requirements:

1. Single mesh

- I wanted to be able to append a single object into any project.

2. Procedural Textures

- I didn't want to have to rely on image maps that could get lost.

3. Versatile

- One single model to be used for any type of character, be it human, alien or whatever.

4. Easy

- Any effects needed to be quickly accessible. I don't want to be wasting time faffing around with colour-band settings at the texture level

5. Impressive

- This sucker needs to look good any way it goes.

6. One other thing I really wanted was the iris musculature to follow the pupil dilation!

[Total Noob] I found this tutorial both informative and useful. It's best to download the PDF version with the screen shots if you want to get as much info out of it as I did, and in fact, understand all that's going on in the Tute

New Link (26th May '09): Here <sup>[2]</sup> is the full tutorial as a pdf file. 9 pages of screenshots and extremely detailed instructions. (2.7mb)

(Noob note: In my blender 2.49, I had to unclick the 'Traceable' setting on the cornea of my eye when following the PDF tutorial, else I ended up with a giant black 'olive', this does not appear to affect ray mirroring, and corrects my problem.)

## Creating the Mesh

[noob note1] Maybe worth to look at this link instead... <http://web.pdx.edu/~wlf/tut.html>

Note: You might want to check out this video first: <http://www.youtube.com/watch?v=KKmHUZroaBA>

The objective here is to make a great looking eyeball. That means it may not be anatomically perfect, but there are some things we need to take into account: The eye should consist of a white (white bit with veins); an iris (coloured bit); a pupil (black but in the centre of the iris) and a cornea (clear bit surrounding the whole eye). These may not be absolutely correct, but these are how I'm referring to the pieces of the eye.

- Start with a ball: Go to a front view (Keypad-1) and create a UV sphere. 24 Rings and segments should be sufficient. It'll give us plenty of smoothness and enough verts to work with when we create shapes. NOOB

NOTE: If you want the eye to point toward you, you may have to rotate it 90 degrees. Go to side view [num3], click Rkey and rotate while pressing the control key. Return to front view.

- This sphere will become the coloured parts of the eye, but we also need a second sphere for the cornea. Use the Z key to switch to a wireframe view (if you're not already there). Hit the A key a couple of times to ensure all points are selected then Shift-D to duplicate the sphere.
- Press S-Key to scale and scale the second sphere to slightly larger than the first. (Hint: Hold shift while scaling for fine adjustment) or you could also enter a value, 1.01.
- Now that we have a nice confusing mesh, we're going to make our life easier. We're going to hide the outer bit for the moment. Hit the "H" key.
- Don't panic like I did the first time I accidentally hit H, you can bring it all back with Alt-H. (Note: In blender Alt + some key often reverses the effect of pressing the key in the first place!)
- OK Now to create the basic mesh for the inner part. Basically, we need to make a concave iris/pupil section.
- Switch to a side view and use the Border Select (B-Key) to select the first four rings and the tip. Now we're going to flatten these. Press "S" for scale "Y" for Y axis only and "number 0" for scale-to zero.
- Now grab the flattened verts with the "G" key and press Y so that it goes the right way, and drag it a little into the eyeball.
- Now for the cornea. It's not perfectly spherical; it bulges slightly at the front. We're going to use the Proportional Editing tool.
- First, we need to un-hide the cornea. Hit Alt-H
- Usually you would use the O-Key, but this little menu item allows you to activate proportional editing for Connected vertices.
- Once activated, you can select the type of falloff from the 3D header bar. Select "Root Falloff"
- Press A-Key a couple of times again to make sure no points are selected, switch to a side view (Keypad-3) and right click just to the left of the front-most point. We just want to select that one vertex.
- Now comes some fun. We want to drag that point away from the eye in the y axis, so press "G" for grab and "Y" for Y axis. As you move the vertex, you'll see a circle that defines the influence of the proportional editing. Adjust it so it is about  $\frac{1}{4}$  the height of the whole eye and drag it out until it looks right.
- To clean the model up, we're going to apply a subsurface and apply some smoothing. Go to the editing buttons and find the modifier panel.
- Add a subsurface modifier and set both level and render level to 2.
- Now hit Tab to switch to Object mode and hit the "Set Smooth" button in the Link and Materials panel.
- Now would be a good time to save your work before we go on to textures.

## Adding the Textures

You're going to need multiple materials on this mesh. It's possible to apply materials to selected faces.

- First, switch to editing mode and change the select mode to Face select. (Oh, and switch off proportional editing if it's still on from the last section)
- Press A a couple of times to ensure nothing is selected.
- Press Z to switch off wireframe; this will be easier without it.
- Place the mouse pointer in the centre of one of the faces on the outside layer of the eye and press "L" to select linked faces. If you get a message saying "Nothing Indicated" try positioning the pointer on a different face and try again.
- Once the cornea section is selected, in the "Links and Materials" panel of the editing buttons, click "New". The counter above it will change to "1 Mat 1" best read as material 1 of 1. Click "Assign" to assign the cornea faces to material 1.

- Hide the cornea faces (H-Key). We'll assign the rest from the outside in. If a face is already assigned to a material and it is assigned to another, it just switches place. Therefore, it's easiest next to assign all the visible faces.
- Press A to select all and, as above, click "New" then click "Assign".
- Now for the iris and pupil. Hit A to deselect all.
- Switch to a front view and zoom in. Hit "B" twice to enable paint select and select the central faces plus one ring out. You can control the size of the selector by rotating your mouse wheel.
- Again, create a "New" material index and "Assign" the pupil faces to it.
- Finally, click A to deselect all and use paint select again to select the remaining faces on the iris. Assign them to a new material.
- Now we're ready to make the materials. Switch to object mode (Tab). Change to the object editing buttons and in the Draw panel, locate the "Transp" control. Set this on. This means as we go we'll be able to see through the cornea.

## Cornea Material

We'll start with the simplest texture first. The cornea is basically just colourless and transparent. Note that we're using Z-Transparency instead of Ray Trans. This is because some lighting conditions make the eye under the cornea difficult to see. (NOTE: Until I get the images uploaded, you should refer to the pdf version for the settings. Meanwhile, here is how I see them:

Go to the shading buttons (F5)

Under the Links and Pipeline tab, ZTransp is clicked along with Radio and Tracable.

Under the Material tab, col is white and alpha is 0.1 .

Under the Shaders tab, CookTor Spec is 1.674 and Hardness is 333.

Under the Mirror Transp tab, Ray Mirror is clicked, RayMir is 0.4, Depth is 3 --Bonapon (talk) 14:31, 6 October 2009 (UTC))

(Noob note: In my blender 2.49, I had to unclick the 'Traceable' setting on the cornea of my eye when following the PDF tutorial, else I ended up with a giant black 'olive', this does not appear to affect ray mirroring, and corrects my problem.)

## White Material

The white of the eye's a little trickier. The material here includes some red veins that can be seen at the side of the eye, but not from the front. This can be done with two textures, but there's a cheat we can use.

First, the veins:

Switch to Material Index 2 (In the Links and Materials panel, click the right arrow so it reads "4 Mat 2" or Material 2 of 4).

Click the cross or X next to the material name and add a new material.

Go to the texture buttons:

Using a marble texture, copy these settings.

(Here are the settings that I see

Under the Colors tab:

Cur: 0: Pos=0; R=1, G=0.617, B=0.629; A=1;

Cur: 1: Pos=0.340; R=1, G=1, B=1; A=0;

Under the Marble tab:

Sharper, Soft noise and Sin are clicked;

NoiseSize=0.634; NoiseDepth=5; Turbulence=26.04;

--Bonapon (talk) 15:25, 6 October 2009 (UTC)

The two colour band items are shown here separately. This makes a nice marble texture that will evenly cover the eyeball. So how to hide it from the front of the ball? Another colour band trick, but this time in the material itself.

Here they are in all their glory.

(As I see them:

Under the material tab, Col is white and alpha is 1;

Under the Ramps tab:

Cur: 0: Pos=0.690; R=1, G=1, B=1; A=0; Input is Shader; (These settings are not shown in the pdf so I have extrapolated. Please correct if wrong.)

Cur: 1: Pos=0..690; R=1, G=1, B=1; A=1; Input is Normal;

--Bonapon (talk) 15:43, 6 October 2009 (UTC)

The colour band items are both white with one alpha 0.0 and one alpha 1.0. As you can see they are very close together to give a fairly clear line that the veins will stop at. The input setting for the band is set to "Normal" this means the left hand side of the band refers to faces that are parallel to the camera view and the right is faces that are facing directly towards the camera.

I've also added a bit of mirroring so that we get a little scenery reflected!

There's also some colour band fun to be had with the pupil material.

## Pupil Material

I wanted a nice black texture, but I also wanted to get a red-eye effect if a light is shone directly at it from behind the camera. Why? I guess just because I can, and now so can you!

As before move to the next material index and create a new material.

There's no textures in this one. It's basically black and a little reflective with a nice hard specular.

(In the materials button (F5):

Under the Links and Pipeline tab, Ztransp, Radio, Tracable and Shadbuff are clicked;

Under the Material tab, Col is Black, presumably with an alpha of 1, while Spe is white with an alpha of 0.6;

Under the Shaders tab, CookTor spec is 0.594, Hard is 294;

Under the Mirror Transp tab, Ray Mirror is clicked, RayMir is 0.3;

--Bonapon (talk) 18:14, 6 October 2009 (UTC))

Have a good look at the colour band settings though!

Position 1 is black with an alpha of 0 and positioned at 0.97. Trust me on this setting, it took a while to get it right!

(Under the Ramps tab:

Cur=0; Pos=0.970; R=0, G=0, B=0, A=0; --Bonapon (talk) 18:22, 6 October 2009 (UTC))

The second is all the way across at position 1 and is fully red (with no blue or green) and alpha 1. We want the red-eye to respond to light not angle like the white of the eye, so set the colour band input to "Shader".

(Cur=1; Pos=1; R=1, G=0, B=0, A=1 (The alpha of the second position is not shown in the pdf) ; --Bonapon (talk) 18:22, 6 October 2009 (UTC))

Now the bit you've all been waiting for – the iris!

## Iris UV Map

Save your work and grab a cup of coffee for this part. It's a little involved.

We need the texture to stay glued to the mesh irrespective of the shape of the mesh. This gives the effect of the muscles in the iris expanding and contracting. In a nutshell, we're going to UV map a procedural texture to a single part of the mesh.

OK, here we go. Switch to a material editing window. You'll need to be able to see both a 3D window and an image window. It wouldn't hurt to see a buttons panel as well. My material screen kind of looks like this.

**(Noob Note:** To clarify: an "Image Editor" window option is in the **Select Window Type** drop down button situated at the very left of the header next to the **View** drop down. Select '**UV/Image Editor**'. To achieve the screen the author describes, first split your view. Then select a front ortho view with **NUM1** in the right window. In the left select **UV/Image Editor** from the '**Select Window Type** button as described.)--WithAlligators (talk) 05:59, 24 February 2010 (UTC) (Blender 2.49b)

Make sure the window is displaying a front view (Keypad-1) this will be important when we unwrap the bit we're going to use.

Change the 3D window to mesh editing / face select and use the brush select (B-Key twice) to select the iris faces.

Now change to UV Face Select mode. The faces should remain selected. (As of Blender 2.46 the UV Face Select mode no longer exists. It's been merged with Edit mode. All you have to do is select all the vertices/faces/edges in Edit mode and press the U-key.)

With the cursor in the 3D window, press "U-Key" for unwrap and select "Project from View". The image window should now contain a nice even circle of dots. Put the cursor in the image window and press "A" to select all the points. Scale and move them until they fit nicely in the outer square of the image grid. Now select each ring in turn using brush select, starting from the innermost, and scale them until they cover from the centre to the outside of the image grid as shown.

Now we've got a map, lets get a texture. Switch back to a modelling screen and change to object mode for the next part.

## Iris Material

Firstly, a texture to control the colour blend across the iris. Generally it starts on the outer edge dark and gets lighter as it moves to the centre. Sounds like a job for the blend texture.

The texture itself is fairly straight forward, but again, the colour band is where the magic happens.

I've pictured all three stops in the colour band. Note the alpha values. This blend texture is meant to tint the iris texture so it's lighter in the centre and darker in the outside. (**Note:** I will have to create my own screen prints as I don't know if the original author has uploaded images or if they are copyrighted. For this to happen, I will first have to complete the tutorial myself.--Bonapon (talk) 20:34, 8 October 2009 (UTC))

Now for that iris texture:

I stumbled over this texture almost by accident when I was working on a different project. Copy the settings as they appear here.

Note: the colour band points are black with alpha 0 and white with alpha 1. This allows the material colour to come through the texture, so we can set the main eye colour in the material buttons.

The main colour is blue which sets the colour of the iris. The centre colour comes from the texture settings:

The blend texture is applied to the UV map input. The green colour is used as a lighten mix to tint the iris near the centre. Using the Stencil option allows the other texture to show through.

The iris texture is applied to UV again and some normal as well as colour is applied. This makes some small shadows in the iris when the light is from the side. Add some lights and have a look at your eyeball so far.

## Pupil Dialation

We're going to use shape keys to create the pupil effects.

We're going to create a normal eye and a cat's eye. Fortunately, the fully dilated eye is the same in both cases so that's going to be our shape key basis.

Go to the edit buttons and select the Shape Key panel.

Make sure "Relative" is selected and click "Add Shape Key" to create the base shape.

Switch to edit mode, point selection. Select just the first loop in of the iris. To select an edge loop in point mode, place the cursor on a line between two points that's on the loop and Alt-Right Click. Shift + Alt-RClick will add the selected loop to what's already selected.

Now scale this to just inside the width of the loop that the iris shares with the white..

Now do the same with each of the other two loops. This should produce a nice dilated pupil.

Hint: If you scale the inner loop out, but the edge of the colour doesn't follow it exactly, drop it and hit Shift-E, then move the pointer to the right. This causes the subsurfaced mesh to closer follow the selected edge.

Now to create the normal (human) pupil shape. Switch to object mode (Tab) and add a new shape key. Name it "Normal Pupil Contract"

Now grab the edge loop between the pupil and the iris. Scale is down so it's just a little bigger than the loop in the pupil. Select all the points in the pupil as well now and scale them all very small.

Now select the next loop out and scale it down, followed by the final edge loop. It should look like this.

Now switch back to object mode and try the shape key slider out. Pretty neat huh?

Let's make a cat's eye now. Switch shape keys back to "Basis" and add a new shape key called "Cat Pupil".

Use the same method as before to scale the edge loops. As you scale, restrict your scaling to the X axis (That is "S" then "X"). You should end up with a shape key like this:

That's pretty much it. Try some other things like rotating iris edge loops in the shape keys for spiral eyes, or weird shapes for the contracted pupil.

Don't forget to save – and enjoy!

## References

[1] <http://blenderartists.org/forum/showthread.php?t=85722&highlight=eyeball>

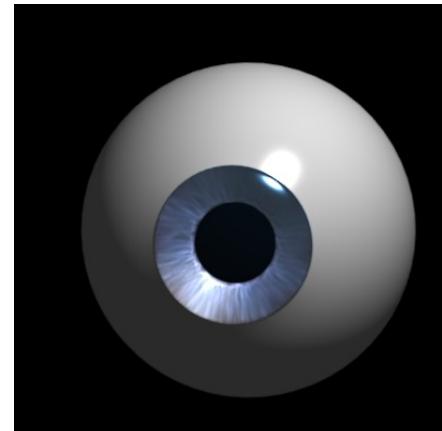
[2] [http://uploader.polorix.net//files/147/Helping/Eyeball\\_Tutorial.pdf](http://uploader.polorix.net//files/147/Helping/Eyeball_Tutorial.pdf)

---

# Blender 3D: Noob to Pro/Creating Pixar-looking eyes in Blender

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.

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. Let's see how it works!

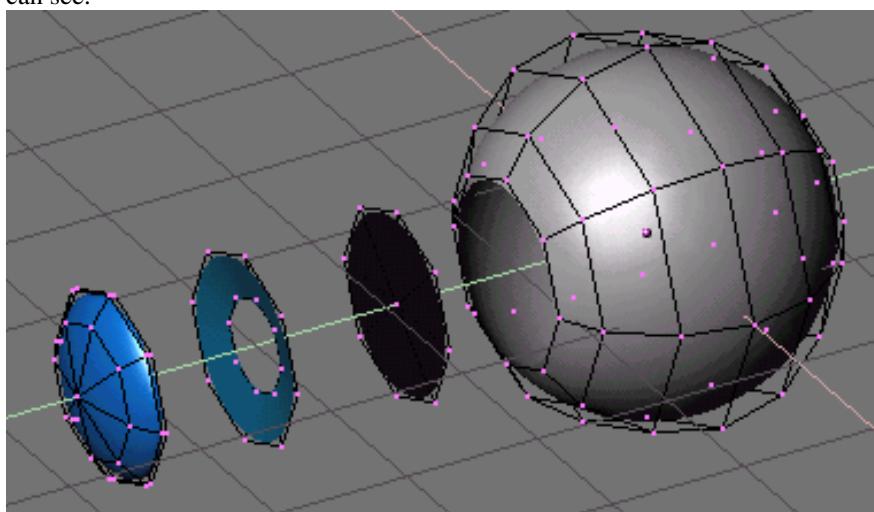


The final result, with a blue iris.

## Parts of the Eye

In this picture you can see the "ingredients" of the eye model. The blue mesh at the left is the cornea. Its 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. The pupil size is the same size as the iris hole. 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.



(**Noob Note:** *Link below is no longer active.*)--WithAlligators (talk) 01:42, 26 February 2010 (UTC)

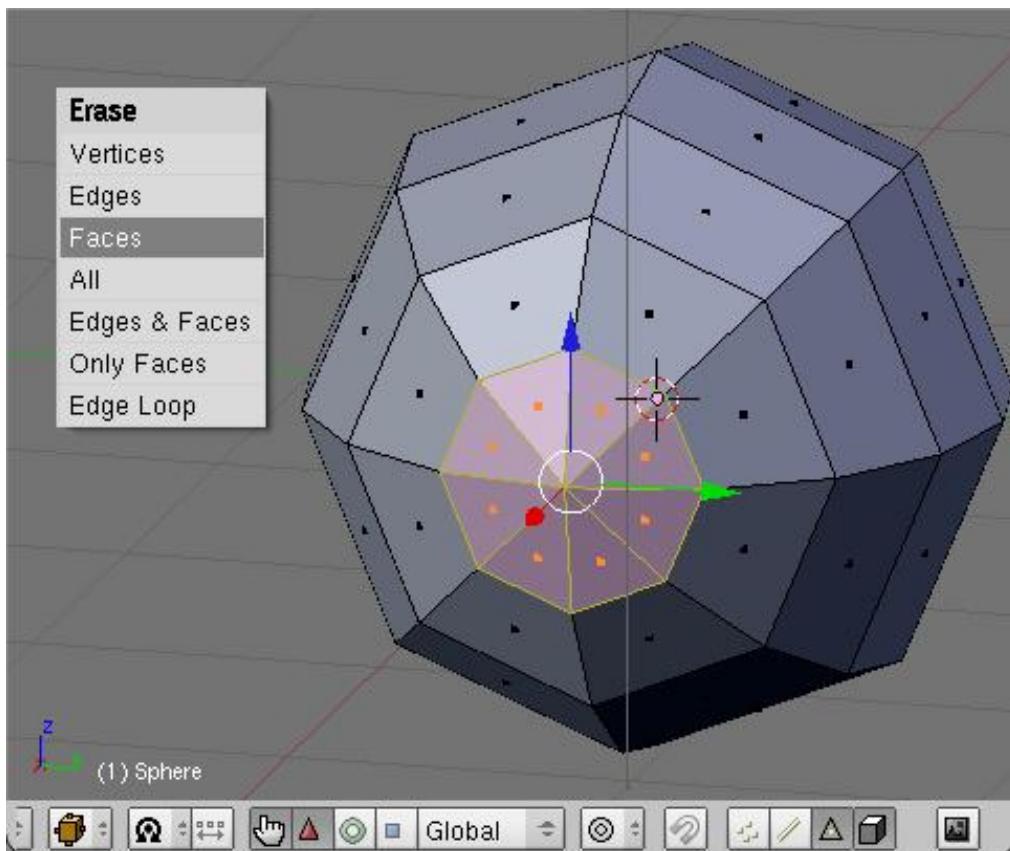
Below is a link to the diagram of the eye [1]

## Materials

Now let's look at the materials.

### Eye White

To make the eye white, press NUM3 to go into side view and add a UVsphere with 8 segments and 8 rings. To create the hole at the front of the eyeball, delete the 8 triangular faces that make up one end of the eyeball. Ctrl+Tab+3 to go into face select mode, B for border mode and select the middle 8 triangles as shown below, press Delete or the X button and LMB on Faces.



The resulting hole will be ringed by 8 vertices (which you could subsequently extrude and use to help you model the sections below - just a suggestion for those who feel capable)

The eye white<sup>[2]</sup> 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.

(User: Make sure shadow buttons are off or it will black out the iris. Spec and Hard are located in the Shaders tab.)

### Iris

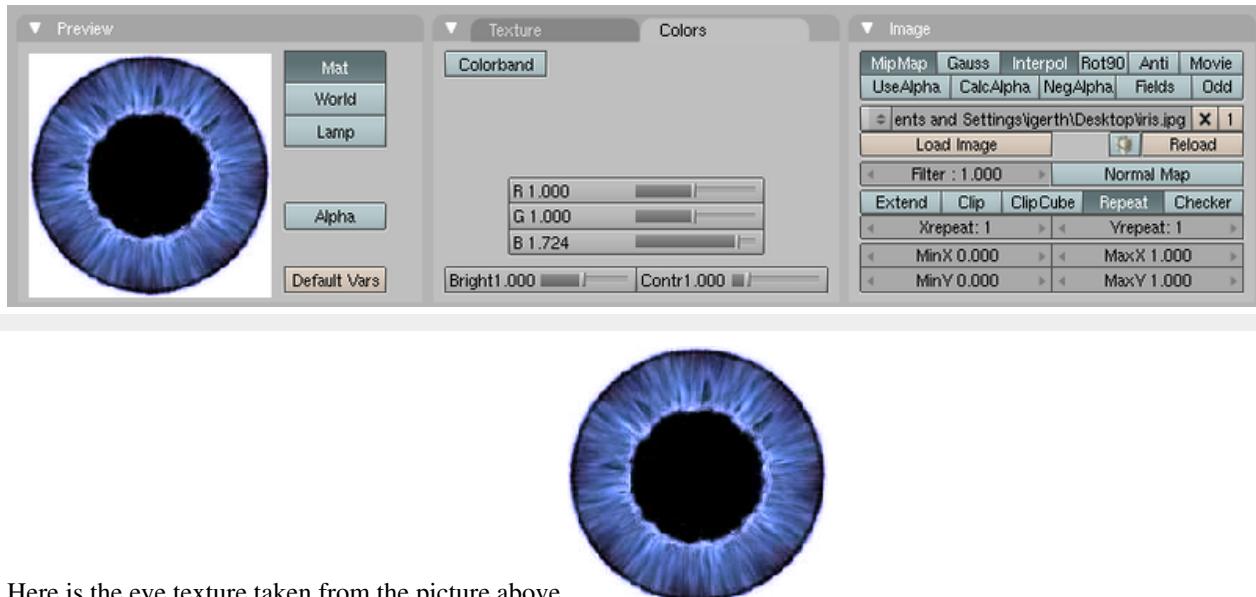
~~ Note ~~ I did this tutorial, first the way they said here, and then my way. I inserted a circle, extruded (edges only), but I DIDN'T move anything. I then scaled what I had extruded towards the origin of the circle (middle). It had the same effect, and was much faster, but you had to get it roughly 1/4 of the way for the same proportion. ~~ End of Note ~~

The iris is built from a circle of 8 vertices.

- Select opposite vertices of the circle and join them using FKEY. While creating these lines, 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 front view and pull the vertices back a little to form the concave shape. (Select vertices, then press G to "grab", and drag LMB away from cornea opening.)
- Scale the iris to the same size as the hole in the eyeball and position it behind the cornea. (Press S, then drag)
- Create a new material and add the iris texture (F6 or click the spotted wall tiny button, choose Image as the texture type and then press Load Image button). To create a material, see the material tutorial at Quickie Material.

The texture should be flat texturing (top button to right of preview) and flat in the Map Input tab. Orco coordinates work. All that is required now is a bit of tweaking of the texture size using the Xsize and Ysize values in the Map Input tab, and scaling the pupil hole size in Edit mode.



Here is the eye texture taken from the picture above

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 light 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.

## Pupil

The material for the pupil is a simple black color with the "shadeless" button on.

## Cornea

The cornea uses a transparent material (**alpha = 0.1**) with **Spec = .6**, **Hard = 255** and **SpecTra = 1**. "Ztransp" should also be turned on (found under Links and Pipelines tab). The cornea is simply a piece that fits exactly in the middle of the hole in the eye white. Make sure that the 'Traceable' button, under Render Pipeline, in the Links and Pipelines tab, is switched off. Noob note: I found it helps to turn off Shadebuff since, as far as I know, you don't generally want the cornea to cast a shadow.

Noob question: I don't know where is the "SpecTra" Value. I'm working on 2.46. Thanks a lot. You can find "SpecTra" under "Mirror Transp"

## Lighting

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.

To move the newly created lamp to a new layer, press SHIFT+M and select the second blue button and press OK. P

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!

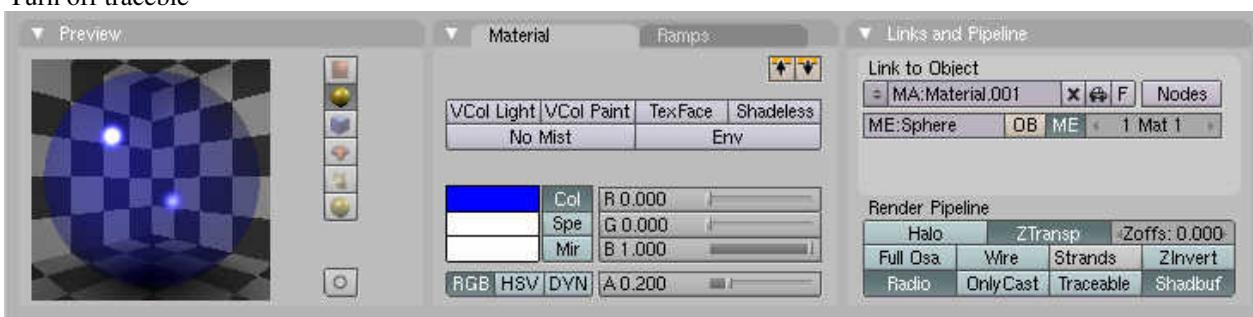
## Changing the Eye Color

You can change the color of the eye by either changing the cornea color, or by changing the iris color itself. Changing the cornea color might be like putting on colored contacts.

### Changing the Cornea Color

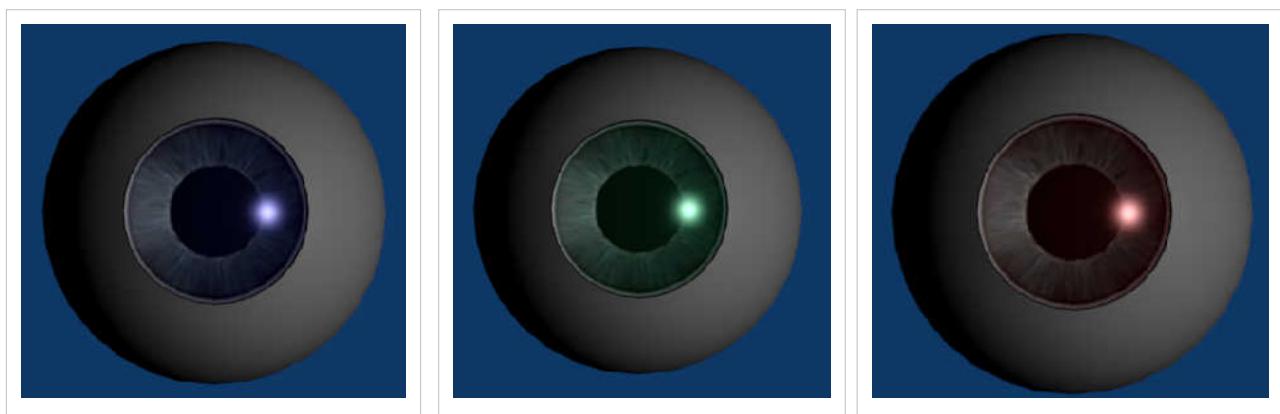
To change the cornea color, do the following:

1. In the mesh which is used to create Cornea set the value of the alpha slider to 0.2 (or more if you desire)
2. Change the color of the mesh to whatever color you like
3. Turn off traceable



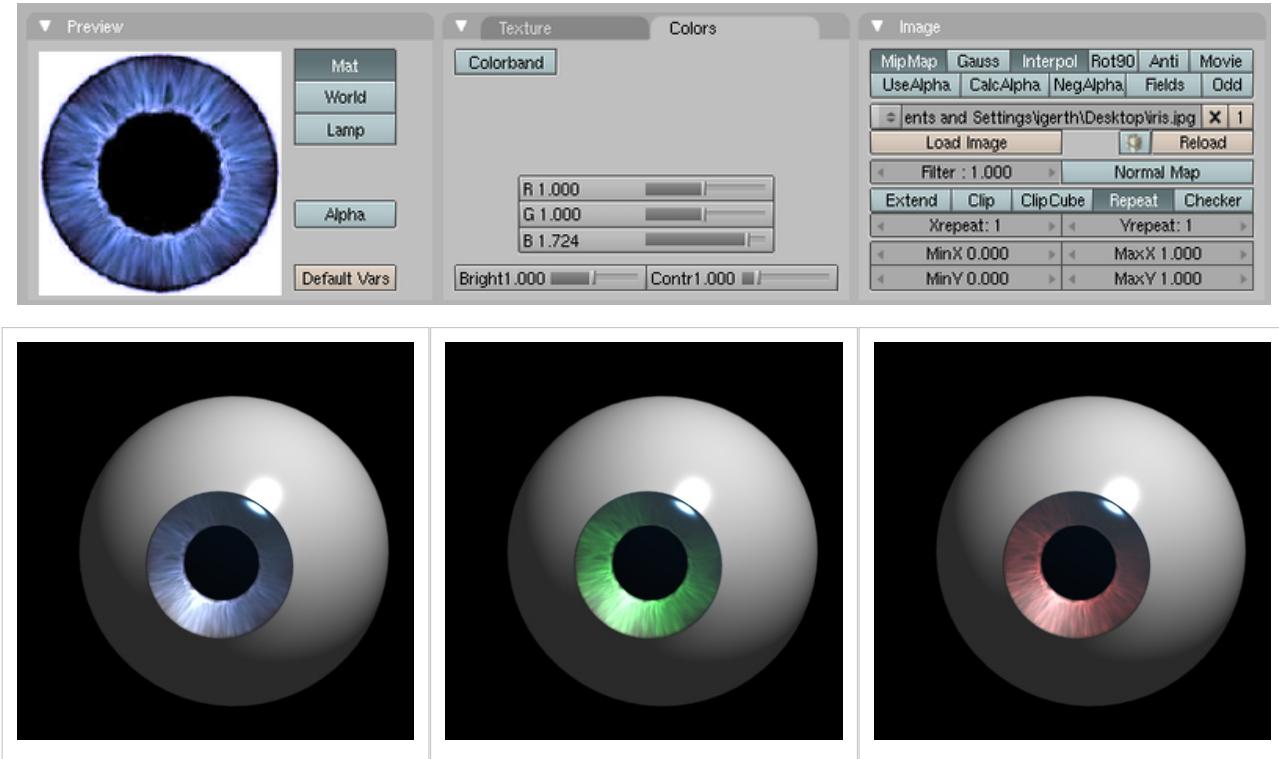
4. Render!

(and remember to smooth out everything by selecting the particular mesh going to edit buttons and clicking on set smooth)



## Changing the Iris Color

To change the iris color you can either edit the iris image external from blender or you can modify the RGB values of the image when imported into Blender.



## Pixar Eye Video Tutorial

- Ira Krakow's Blender 2.49 Pixar Eye Tutorial [3]

## Useful Links

For rigging an eyeball, (making it stay in one place while turning to look at something) and other eyeball stuff, you can visit this site. This tutorial assumes you already have the eyeball made above or one of your own. Click. [4] 5jun2009, the PDF shows no rigging, only creating an eyeball

## References

- [1] <http://www.basiclenses.com/images/uploads/Eye%20Diagram%20National%20Eye%20Institute%20Refno%20NEA08.jpg>
- [2] <http://en.wikipedia.org/wiki/Sclera>
- [3] <http://forum.irakrakow.com/viewtopic.php?f=19&t=250>
- [4] <http://rjnlk94.googlepages.com/eyeballtutorial>

# Blender 3D: Noob to Pro/UV Map Basics

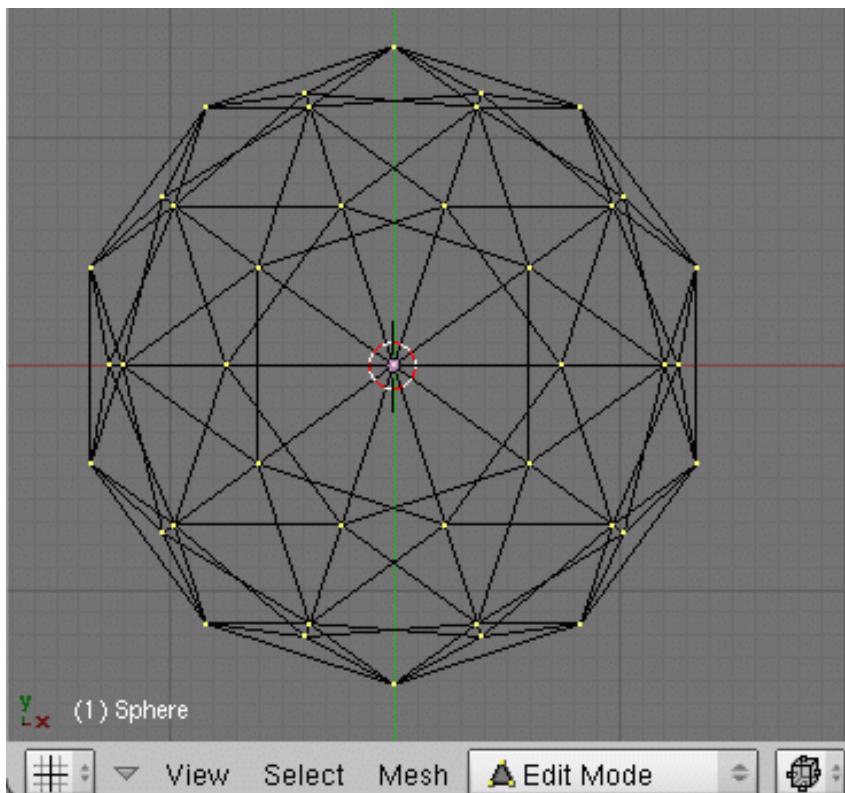
In case you're wondering, UV mapping stands for the technique used to "wrap" a 2D image texture onto a 3D mesh. "U" and "V" are the name of the axes of a plane, since "X", "Y" and "Z" are used for the coordinates in the 3D space. For example: increasing your "U" on a sphere might move you along a longitude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

You can watch a good video tutorial from the main Blender [1] site. It is called **LSCM UV Mapping** or (Least Squares Conformal Map UV Mapping) and it is located on this page: <http://www.blender.org/education-help/video-tutorials/modelmaterial/light/>

## The Basics of UV Mapping

### Add an icosphere

We'll use a sphere for this demonstration. Make sure you are in top view so the equator of the Iconosphere we are creating will be parallel to the x/y-plane and unwrapping will give good results (NumPad7 or View → Top). Delete the initial cube (x). Then, create a new model, an icosphere (**SPACE** → Add → Mesh → Icosphere). Leave the settings at default: subdivisions 2, radius 1.0

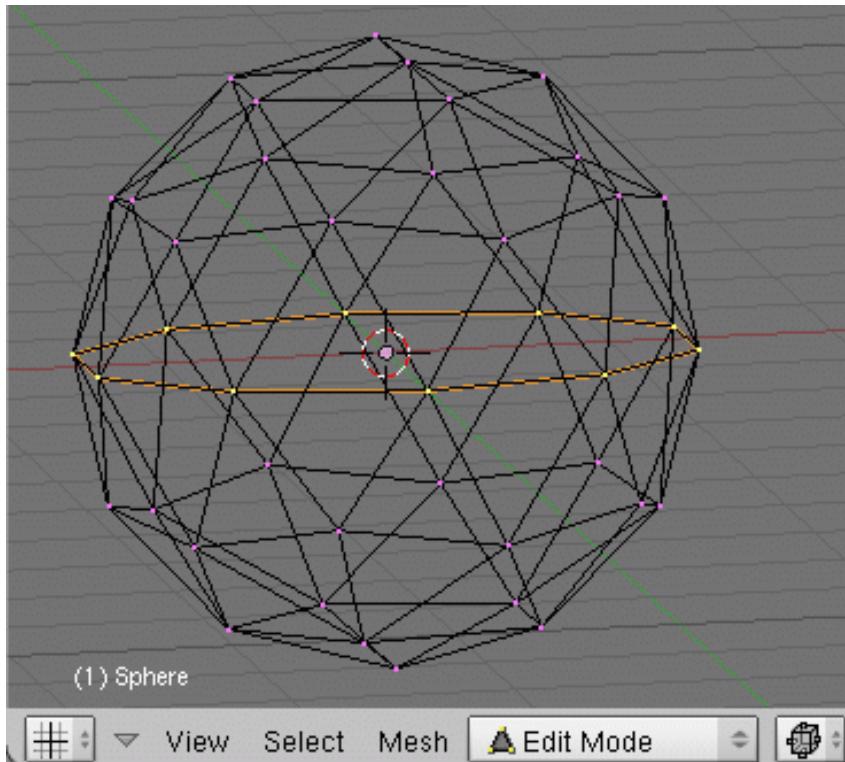


### Mark a seam

In side view, select a ring of vertices on the icosphere. (Like an equator). This can be done easily by going to a side on view (NUM1 or NUM3) and drawing a selection box around the middle row of vertices (BKEY + click&drag). Make sure Limit Selection to Visible is NOT enabled first.

(Limit Selection to Visible appears to be labelled "Occlude Background Geometry" in 2.48. It is the cube button that appears toward the right of the 3D View header [note on edit: this is also not available in the depicted 'Draw Type -> wireframe' mode for later versions])

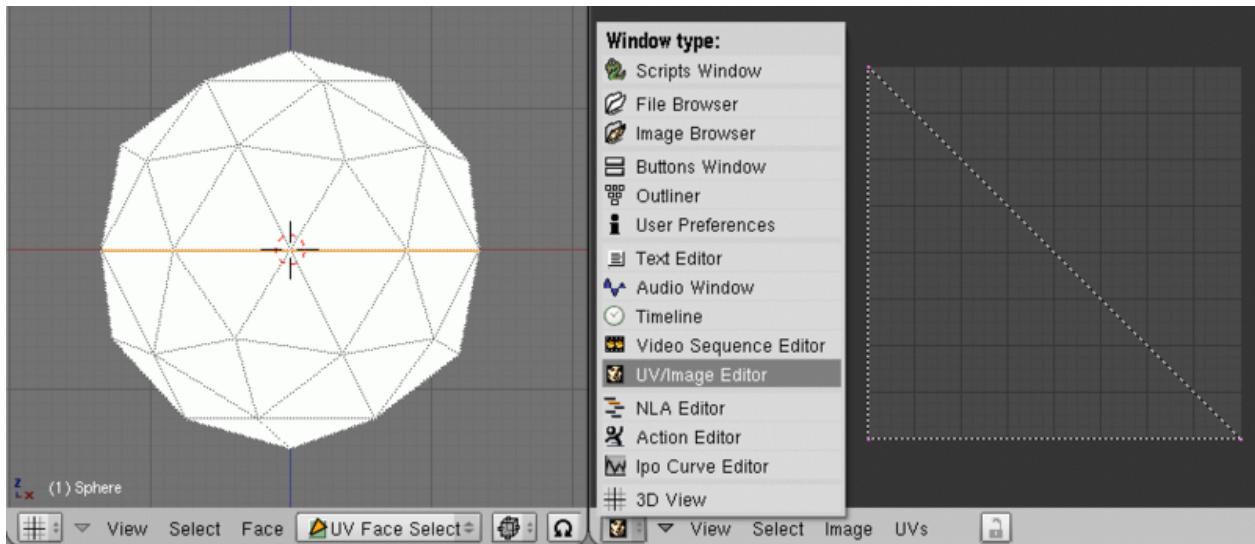
Press **CTRL+EKEY** and select *Mark Seam*, or select *Mesh* menu → *Edges* → *Mark Seam*. This tells the UV unwrapper to cut the mesh along these edges.



### Unwrap the mesh

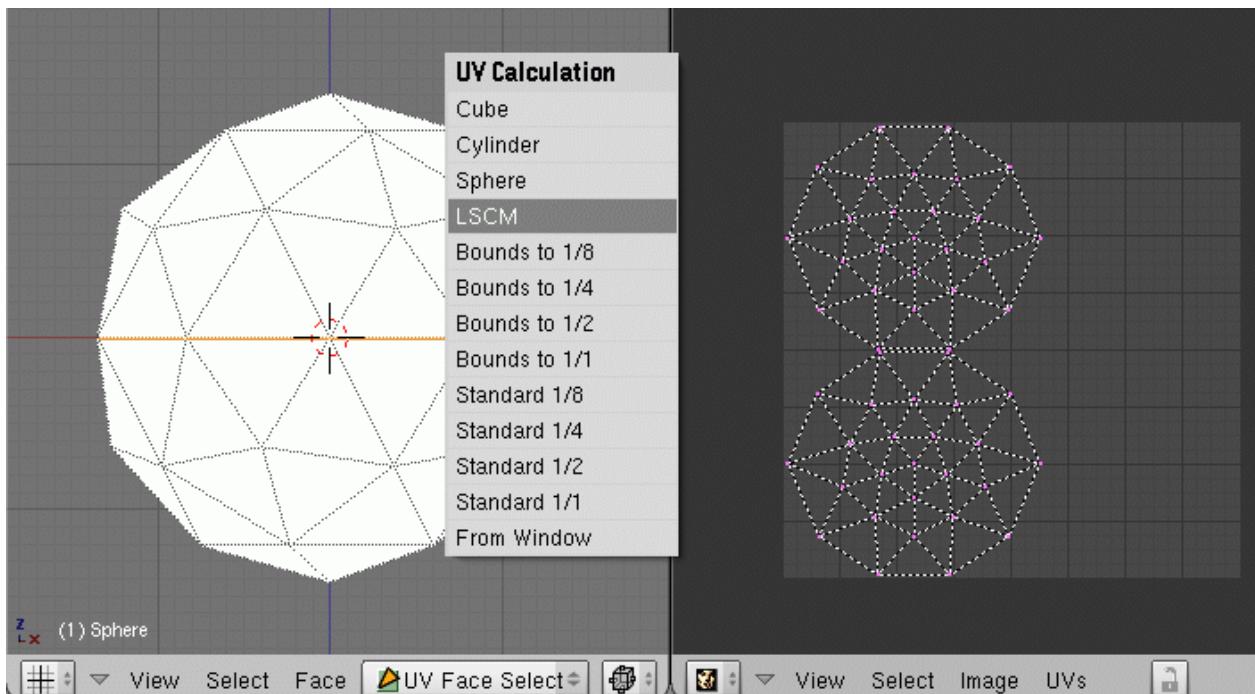
[2.49 is a complete revision, this method wont work but contains useful information and notes by users. For a finishing method that that is current see \*Don't Unwrap]

Next, create a window for the UV mapping: click the **MMB** (**RMB** works as well) near the top border of the 3D View window and select "Split Area". Set its window type to "UV/Image Editor" with the drop down box at the bottom left corner of the new window or with **SHIFT + F10**.



In the 3D View window, select all your vertices, and hit **UKEY** and then *LSCM*. For Blender 2.42 or later select all the faces, and use **UKEY** → *unwrap* or Face menu → *Unwrap UVs* → *Unwrap* to use LSCM.

**Noob Note:** I use Blender 2.45 and **UKEY** doesn't work for this. It seems to be an undo key. Is there another way to bring up the Unwrap menu? **Same Noob:** The tutorial doesn't say, but you have to be in UV Face Select mode for **UKEY** to work for this. **Yoshi:** For Blender 2.48a there is no LSCM but you can use Unwrap.



NOTE: Make sure you switch to the UV Face Select Mode in 3D window. Then "UKEY" does not work as a shortcut for Undo, rather it serves the purpose.

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.

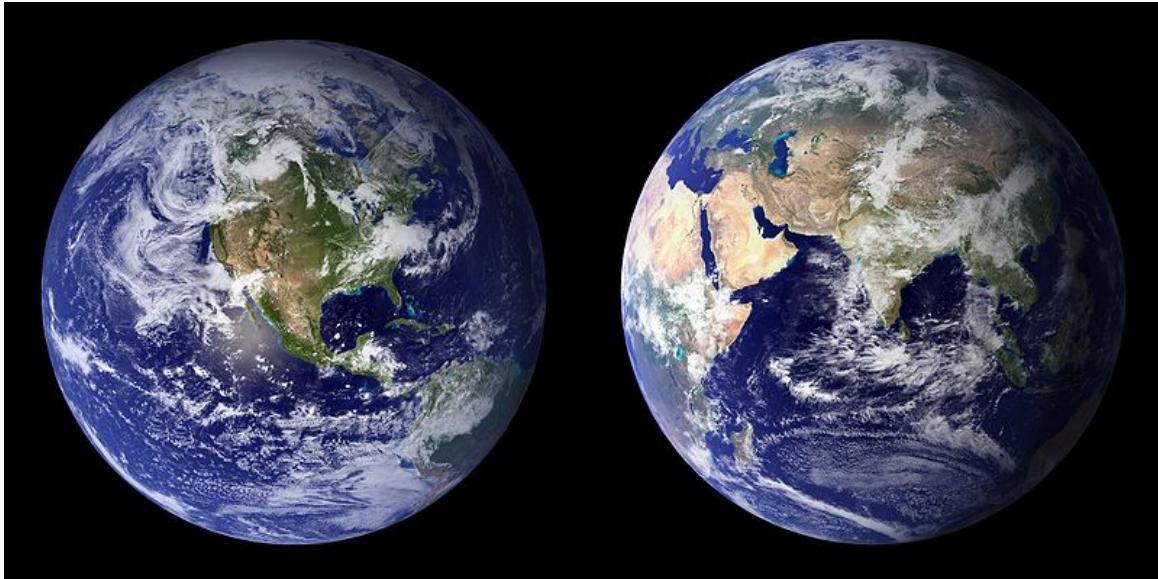
### Make a template image

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. In the UV/Image Editor window, select **UVs** menu → **Scripts** → **Save UV Face Layout...** (User note: for pre 2.43 versions, use **UVs** menu → **Save UV Face Layout**). <<Noob note:" I use Blender 2.48. "UVs" menu → "Scripts" is empty. There's no "Save UV Face Layout..." option... What to do?">>**'Yoshi:** I also use 2.48, make sure all the vertices are selected then hit **UVs -> Scripts -> Save UV Face Layout**. **New Blender User:** This is because you don't have the proper Python software installed. It is this software that provides all the scripts for Blender, so it is important to have the proper software. Note: Blender 2.48 requires Python 2.5.2, which is NOT the newest version. Python 2.6, the newest version, doesn't work with Blender, so make sure to get Python 2.5.2 (2.5.4 that is available at the Python site<sup>[2]</sup> works too). There is another way to export the images, however, that does not require scripts. Go to *Image -> New*, and this will create an image of the net. Then go to *Image -> Save As*, and save the image to where you want it.

With *Wrap* selected, the layout will maintain its proportions, which is best for general purposes. 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 the 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(Click to view in full high resolution (4,096 × 2,048 pixels)):

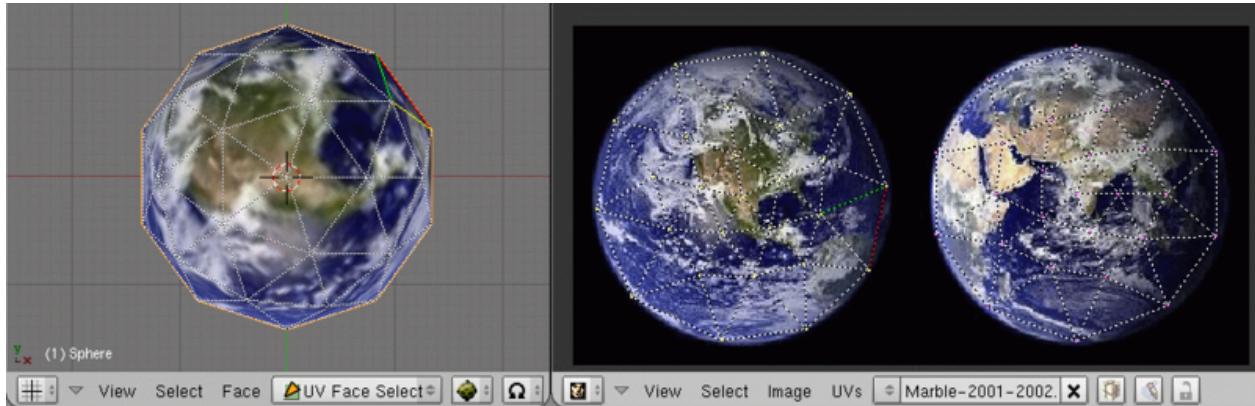


Load it in the UV/Image Editor window by clicking *Image* menu → *Open...* (or *Image* menu → *Load image* in older Blender versions). Then with the very basic operations, grab, rotate and scale, adjust the unwrapped mesh so that it fits nicely on top of the image.

[noob note: when i downloaded this image it did not load correctly, try copying it into paint if you have this problem -> or better click in the link and it's send you to the correct link] [another noob note: when I used the image (in 2.49a/OSX) , it deformed the mesh, I found using a resized image (405 px. square - the same image as above placed on a square black background) sorted this out]

[2.49b Don't unwrap. After **3D View -> 1. (CTRL+E) Mark Seam -- 2. (AKEY) Select All >> UV/Image Editor - 3. Load: BlueMarble-2001-2002.jpg. -- 4. Click the 'belly button' to select the picture. (You should see the picture.) Now unwrap. This will load the mesh horizontally on unwrap. Then use the regular manipulation keys (RKEY,**

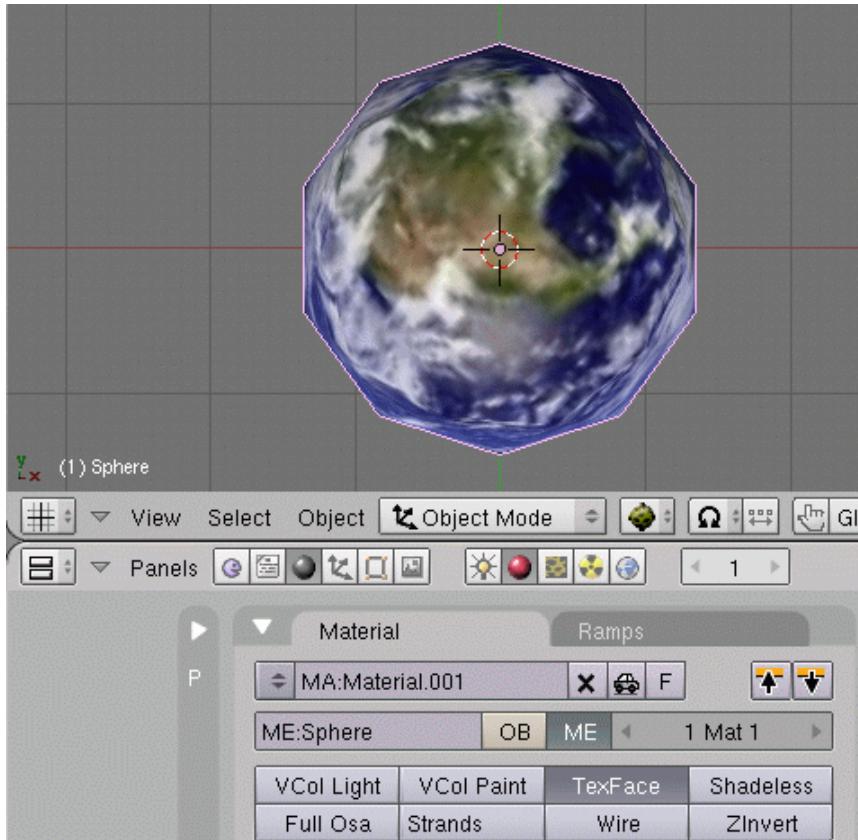
SKEY, GKEY) and buttons in **UV/Image Editor** to fit the uv meshes. *Observed: The uv mesh loaded alone is one vertical island while the uv mesh loaded on an image was two. Weird. In your Edit Panel buttons around now you can see the sub-context master panel called UV Calculation. This is like an important thing so don't worry if it is brain overload. Experimentation with that pack margin will separate the mesh islands in either case. ]*



### Admire your new creation

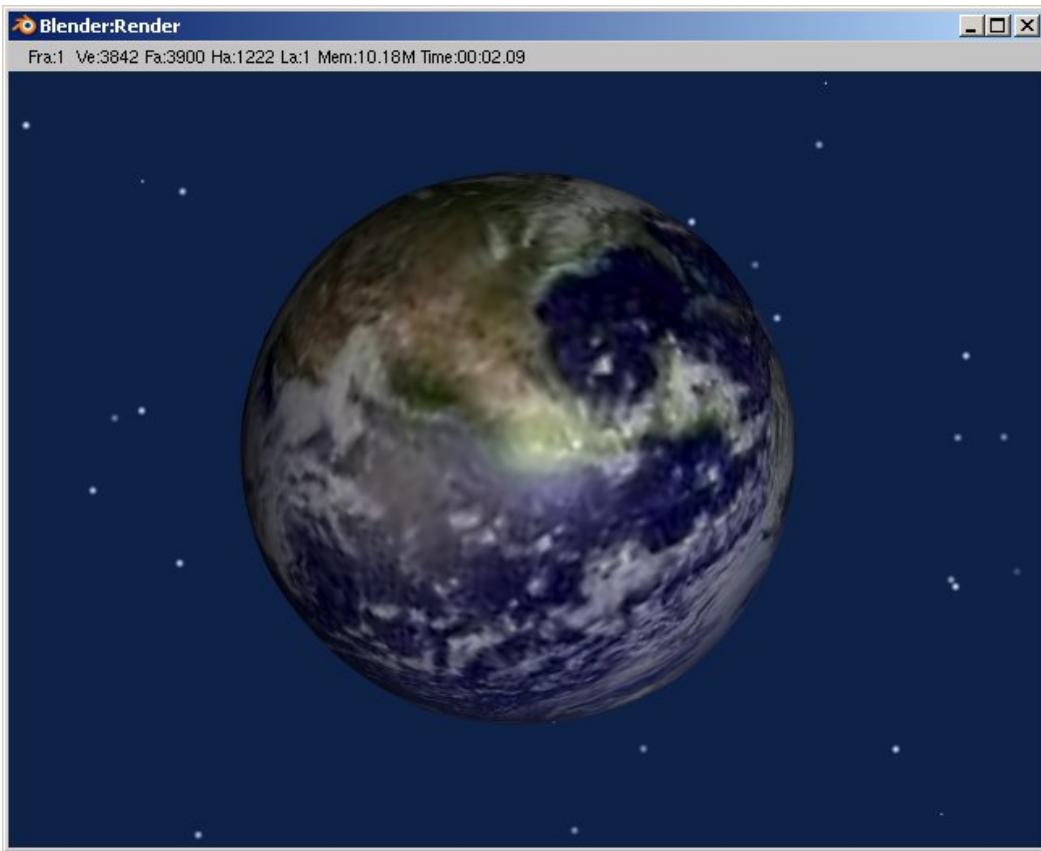
Back in the 3D View window, select Object mode. The next drop-down menu to the right is the Draw Type menu; use it to set the Draw Type to Textured. (Greybeard calls it "potato mode" because of the icon that the menu shows.) Hit TAB a couple of times to refresh your object, and admire your new picture mapped onto your object!

To make the texture visible in renderings, you also need to add the texture to the icosphere as a new material. In the Buttons window, switch to the Shading buttons by clicking the small shaded-sphere button or by pressing **F5**. Create a new material by pressing *Add New* in the *Links and Pipelines* mini-window, then turn on TexFace in the Material mini-window:



To finish your work, switch to Edit Mode and select all vertices. In the Buttons window, switch to the Editing buttons by clicking the small four-vertices-in-a-square button or by pressing **F9**, and then click *Set Smooth* in the

*Links and Materials* mini-window. In the *Modifiers* mini-window, click *Add Modifier* → *Subsurf* and set *Render Levels* to 3. Switch back to the Shading buttons (**F5**) and activate the World buttons by clicking the small "Earth" button in the second button group. Enable *Stars* in the *Mist/Stars/Physics* mini-window. The scene is ready! Render it by selecting *Render* menu → *Render Current Frame*.



### 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:

- 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.



## Questions

1. **Question:** I can't get the image to load in blender v.2.44, thought I did everything you said. What to do?
  - **Answer:** In v.2.44: click 'Image' button on the bottom of the UV/Image Editor view port. There is the 'Open' option for loading an image file.
  - **Answer 2:** There appears to be a bug in Blender. If you still don't see an image, try cropping your image to dimension that are power-of-2, like 256x256, or 512x512, or 1024x1024.
  - **Answer 3:** Make sure to select the image in the file browser with both the right AND the left mouse button
  - **Answer 4:** Undo (Ctrl + Z) one step also helps to refresh the window and makes the image visible again.
2. **Question:** Whenever I press U in the step above everything is deselected and nothing appears in the UV window. What to do?
  - **Answer:** First you select the circle on the icosphere (the 'equator'), ctrl-E, mark seam. Then where it usually says Edit Mode or Object Mode, click that and select UV Face Select, THEN Press U.
  - **Elaboration:** I'm experiencing the same thing even if I hit ctrl+E and mark seam. And AFAIK on 2.48a there's no UV Face Select mode. Well after hitting **UKEY->Unwrap** there's no projection of the sphere on the UV window. *Update:* I found the problem. In the 3d window you have to select all vertices in edit mode. Then the projection will show up.
3. **Question:** My image won't load and is there a way to load images other than typing in its entire address? (I.E. is there a 'browse' button?)
  - **Answer:** While there is no visual list of common shortcuts, if you type in the crude location of your file and double click on directories to get to the fine one containing your file.
4. **Comment:** I selected "Save target as.." instead of "Save picture as.." in my browser, so tried loading a saved HTML file instead of the jpeg. Also it seems you need to reload the directory listing in the "Open" dialog box in Blender else it won't find the newly saved file. You can do this by going to the parent directory and back into the child directory.
  - **Response:** Normally the image file used by the html file will be downloaded along with the html file. Loading an html file as your texture is not recommended.
5. **Comment** There does not seem to be a "UV Face Select" mode in blender 2.46 - There is a face select mode in 2.46+, but, first go to edit mode, than find the small triangle icon in your window's toolbar, hovering over it will give you the tooltip: "Face Select Mode". you can also use the shortcut CTRL+TAB+3KEY in edit mode. (The default mode is vertex mode, the four dots icon)
6. **Question:** Because 2.46 has no UV face select mode, how do I apply something like "two-sided" to a whole group of faces instead of just one at a time? - See the tip above on how to get in face-select mode
  - **Response:** Changing a material property for any user (even a single selected vertex) will change that for all the users of the material. You can assign faces their own material (those 3 vertex if the face is a polygon) even though they are normally selected with a mesh. With using Similar to Selection -> Material you can then perform other actions on those faces. Beyond that you can create groups that do not use materials as their commonality. I am a Noob, so I have never seen UV Select as a mode, but you can work out details before mapping, and there are some buttons like Sync which seems relevant.
7. **Question:** when i press u it does not show the menu you have on your screen shot it has unwrap and some other things but does not say LCSM
  - **Response:** "LSCM" has been renamed to just "Unwrap"
8. **Question:** When I finished the mapping, I had the normal earth sphere, but with this black ring where the seam was made. It's not a flat line.
  - **Response:** Make sure that all the vertices are inside the circumference of the Earth image. Also notice that each half of the Earth image is shadowed on the right side. Try rotating one half of the vertices 180 degrees

around so that the shadows on the Earth match up on both halves.

- **Question:** My earth won't show up in renders. I've mapped it, I've tried all sorts of lighting, but it won't show up. What should I do?
9. **Question:** In the UV Image Editor panel, I can click on Image > Open and select the edited TGA file, but when I click Open Image, nothing happens. Is there a problem between Blender 2.49 and Python 2.6?
- **Response:** Nothing actually changes in the UV Image Editor window. You might need to change your Draw Type in a 3D window to Textured.
10. **Question:** UV export is not working: I have python 2.5.4 with blender 2.49b. I can generate the UV in Blender but I cannot see any of the script options (menu empty) even when all the meshes are selected. Furthermore when I try to export an image with save or save as, it just creates an empty tga with only the background color and no edges whatsoever.
- **Response:** Make sure blender is properly installed, in my case the installer forgot the .blender and plugin directory. Try the .zip file from the blender website instead of the installer. uv\_export.py should be in the .blender/scripts folder. In the uv image editor use UVs -> scripts -> save UV face layout to export to tga.

## Source

Greybeard's LSCM Mapping video tutorial: <http://www.ibiblio.org/bvidtute/mytut/uvtut.avi>

## Video Tutorial

Ira Krakow's Blender 2.49 UV Map Basics Video tutorial: [http://www.youtube.com/watch?v=I\\_8OV92HLPY](http://www.youtube.com/watch?v=I_8OV92HLPY)

## References

- [1] <http://www.blender3d.org>
- [2] <http://www.python.org/download/releases/2.5.4/>

# Blender 3D: Noob to Pro/Realistic Eyes In Blender

---

## Overview

This tutorial will teach you how to effectively utilize textures and materials in Blender to create realistic eyes for characters. This tutorial was inspired by a great Maya tutorial in the Gnomon Workshop series by Alex Alvarez. My goal with this tutorial is to teach you how to get those same stunning results using Blender. Please check out Alex's tutorial as well, as it is very informative and covers more detail than I will be covering here. This tutorial assumes you know your way around the Blender interface and so I will not explain each key command as I go, but will instruct you in what steps to take and point out details such as key commands only when it seems relevant to do so. In addition, you will need some sort of image manipulation software, as this tutorial relies heavily on image-based textures, which we will be preparing outside of Blender. I will be using Photoshop, but feel free to use GIMP or whatever you're comfortable with.

*(Note: image was removed by moderators because there is no WikiBooks license that allows me to post an original piece of artwork without giving up the rights to it. Images hosted on external sites cannot be embedded either.)*

Above is an example of what can be achieved using the techniques described in this tutorial.

OK, let's dig in...

---

## Reference Material

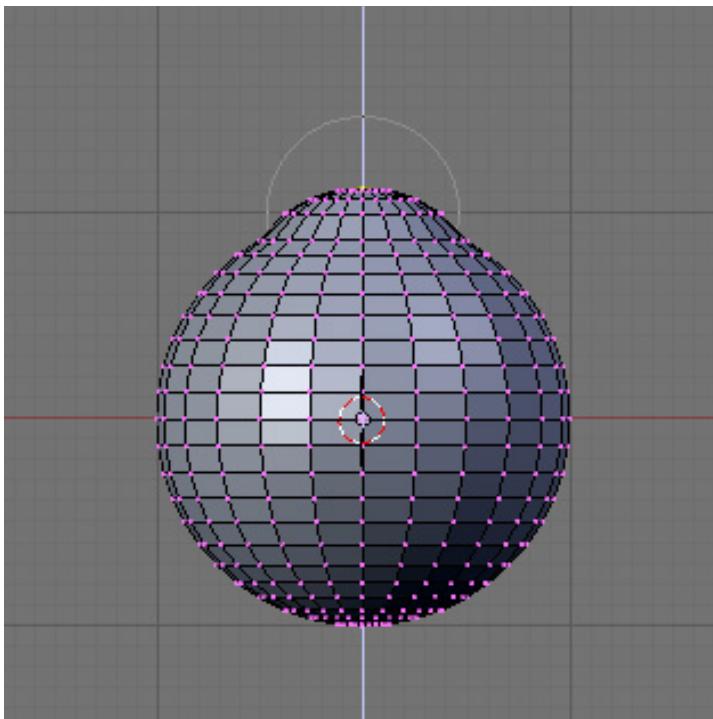
Let's Take a moment to familiarize ourselves with the anatomy of the human eye. Do a Google image search and examine whichever diagram is to your liking. Now, you don't have to pay attention to every single feature, but do take note of the **cornea**, the **iris**, the **pupil**, and the **sclera**. These are the only parts that are visible to us when we look at a person's eyes, and therefore the only parts we will actually need to create. Many of the diagrams you find out there will be disproportionate, and so in order to know how large to make our features in relation to one another, it is best to view a large number of diagrams, a large number of eye photos, and also photos of plastic models of the eye, as plastic models tend to be truer to scale than 2D drawings. I also found a photo of a guy popping his eyes half-way out of his head. All of these together gave me a good feel for what the appropriate proportions are for a human eyeball. To some extent though, cornea size and sclera shape will vary on a person-to-person basis, so there is no precisely correct size, per se.

---

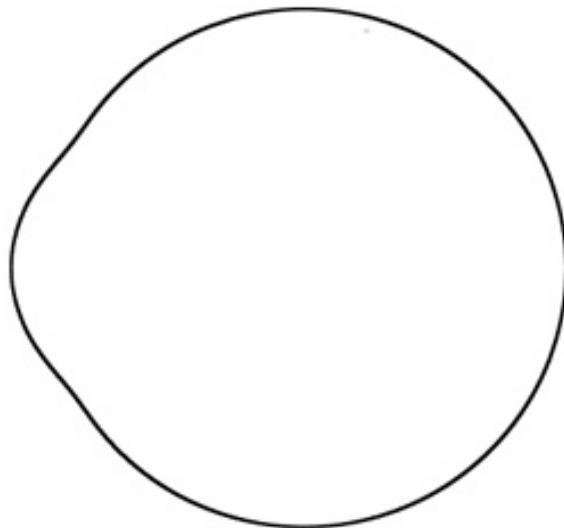
## Building The Eye

Create a UV sphere. 24 segments, 24 rings, and a radius of 1 is recommended. The top of the UV sphere will become the cornea of the eye. So now let's go into **side view**, in the **orthographic** view, and with the topmost vertex selected and **proportional falloff** enabled, **grab** and raise this vertex along the **Z axis**. Don't click to finalize the edit just yet - first we will use the **mouse wheel** to control how much falloff is going to be applied. This process can require some finesse and multiple adjustments until the bump of the cornea is anatomically correct. You will want to set the falloff shape to "**sphere**".

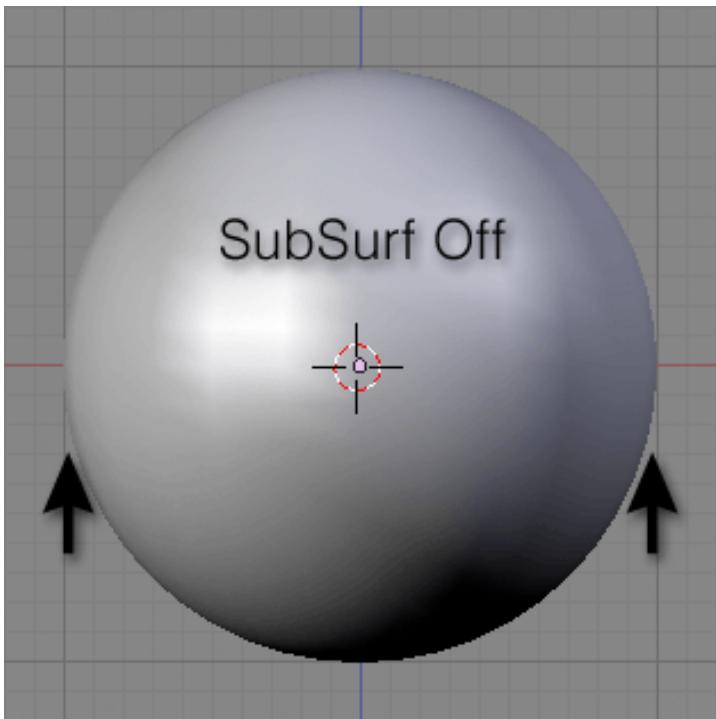
---



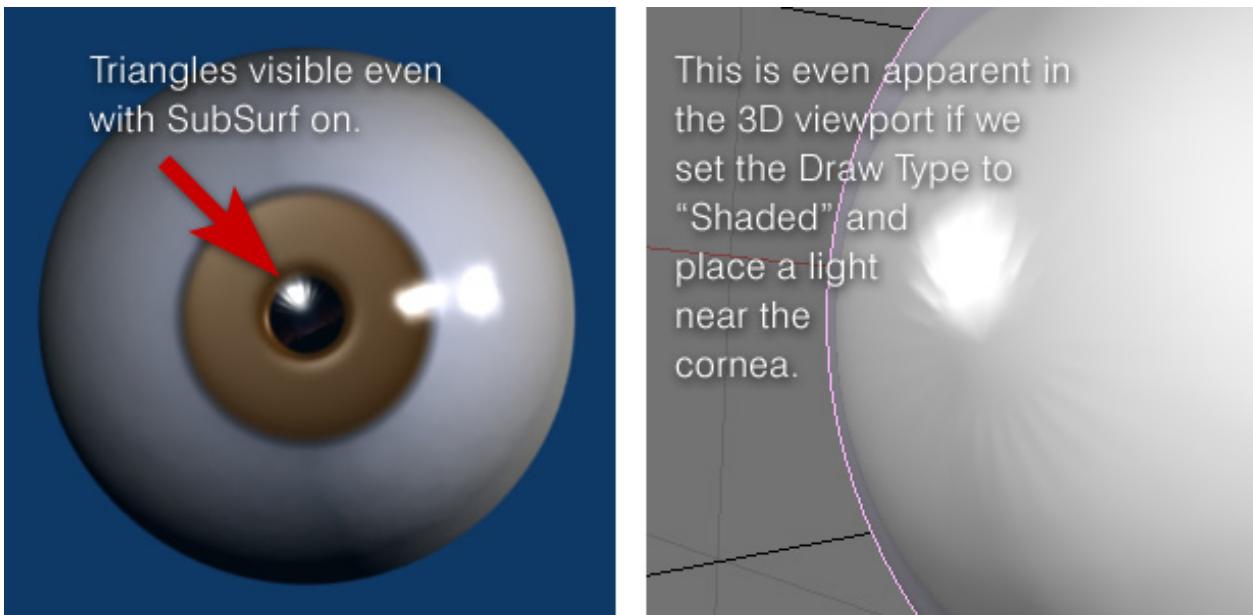
Below is a render of the curve I ended up going with. You can use it as a guide when modeling if it helps you. *This is with a SubSurf of 3 applied*, so keep that in mind.



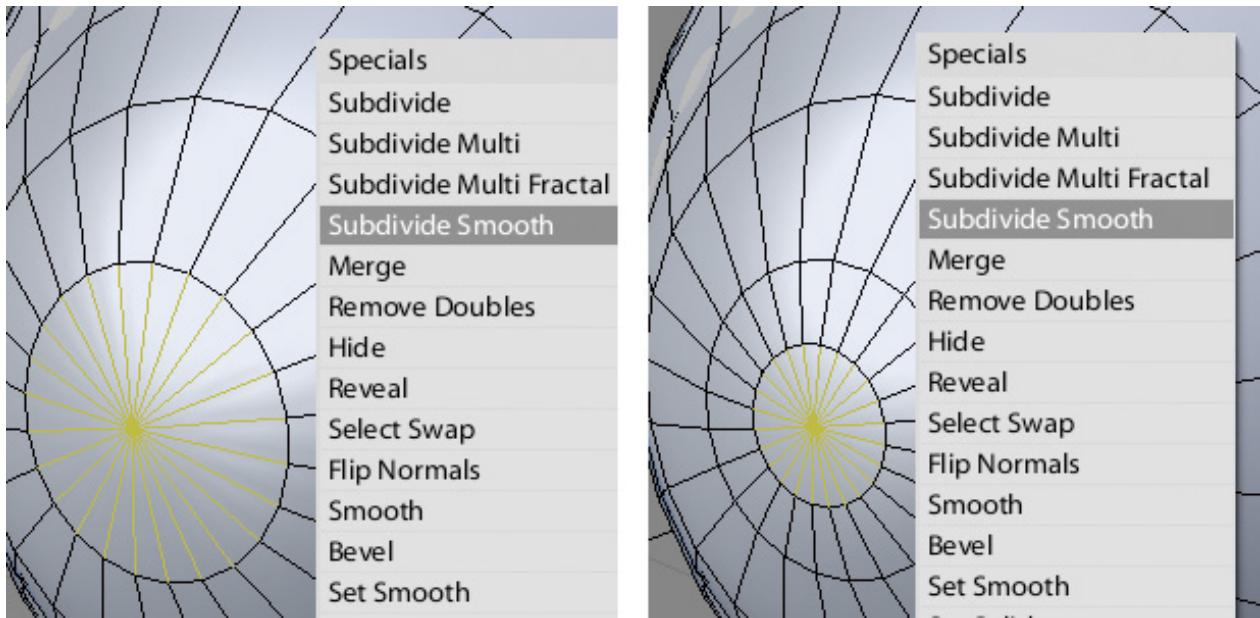
Now, some people may prefer to use a NURBS sphere for the geometry of their eyeball. If you are comfortable with NURBS modeling, and texturing NURBS models, go right ahead. It will give you a smoother and more rounded eyeball. I prefer not to complicate things with NURBS when polys will do just fine, and am guessing most of you feel the same, so I'm sticking with polys. This does however bring us to the topic of the SubSurface modifier and how it will affect your eyeball. When SubSurf is applied to a UV sphere such as our own, it causes the previously spherical shape to become slightly oblong. In other words, the eye will become nearsighted. This is barely noticeable, but it does happen, as a result of the way SubSurfing affects a UV sphere. You could create a UV sphere with more rings to ease this if it bothers you. It looks fine for our purposes though, to be honest. See the comparison in the animated GIF below.



There is one last issue to address with regard to using a polygon-based eyeball. That is the small triangles that persist on the poles of the eyeball. These stubborn tris will remain and you will find that they result in an ugly and unnatural pinched effect when a specular highlight appears over the center of the cornea. Adding a SubSurface modifier will not make them go away. This didn't become apparent until after I did some texturing work and test renders, but I want to point it out to you now, before we get into all that. (See examples below)



This can be remedied by adding more rings where those edges come together. Select only the edges in the very center of the cornea and **Subdivide Smooth**. Now **deselect** and again **select** the new innermost edges. **Subdivide Smooth** again. You can even do this a third time if you find it necessary. Rotate the model around in the viewport and do a couple of test renders to make sure the specular highlight doesn't reveal any artifacts.

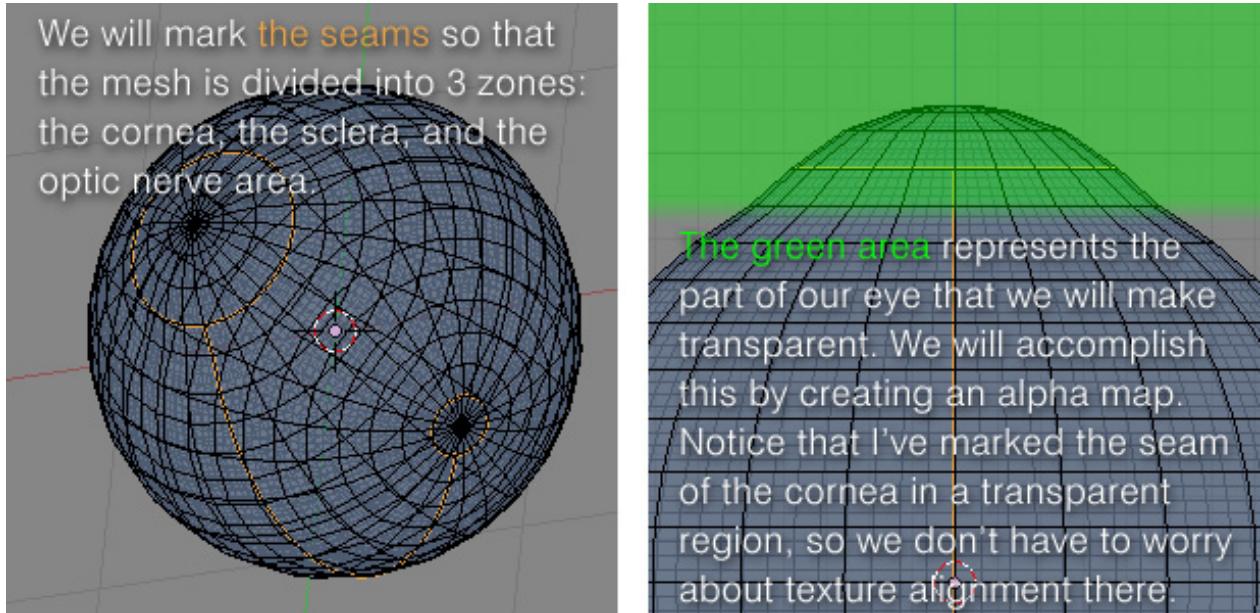


Now repeat this process at the opposite pole of the eye (the optic nerve area).

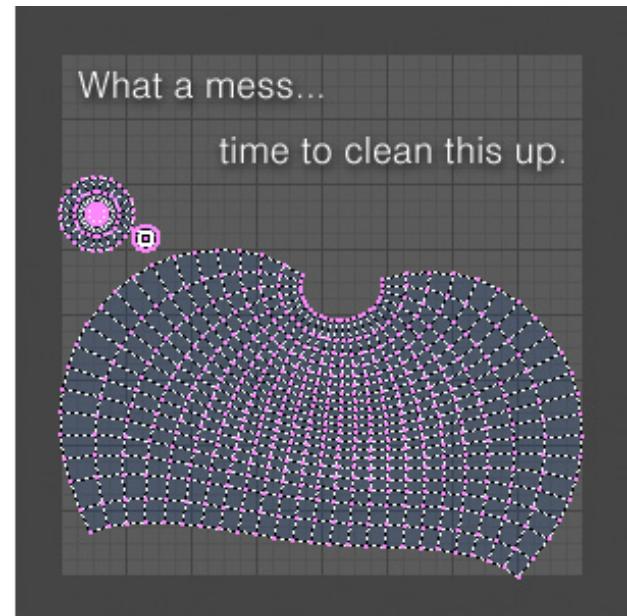
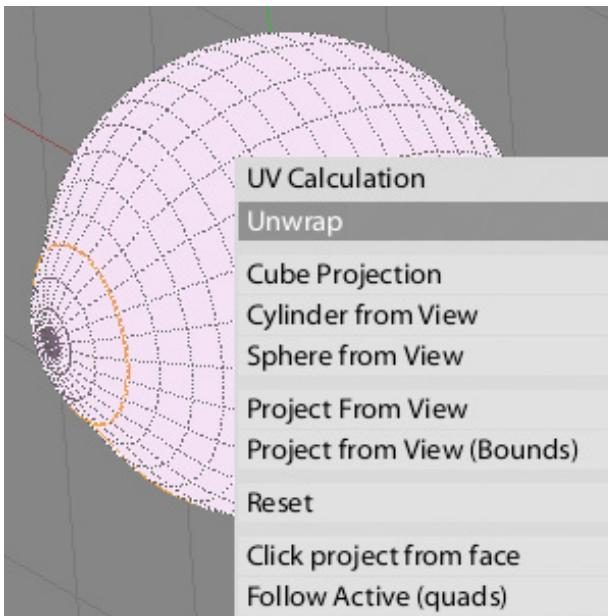
We are now ready to begin texturing the eyeball.

## UV Mapping

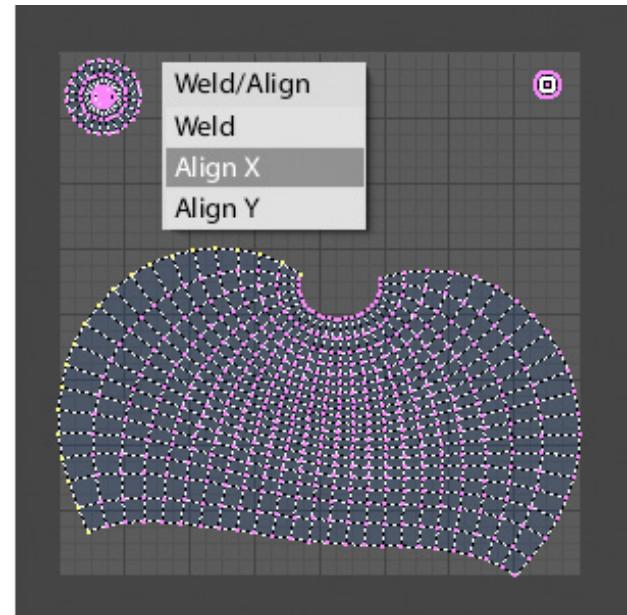
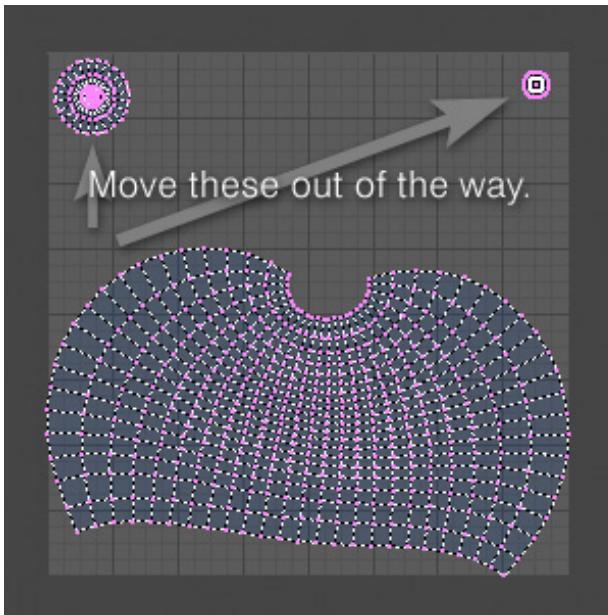
In **Edit Mode**, we will define the seams for our eye. Select the edges you want to mark as seams and type **Ctrl E**, then choose **Mark Seam** from the menu.



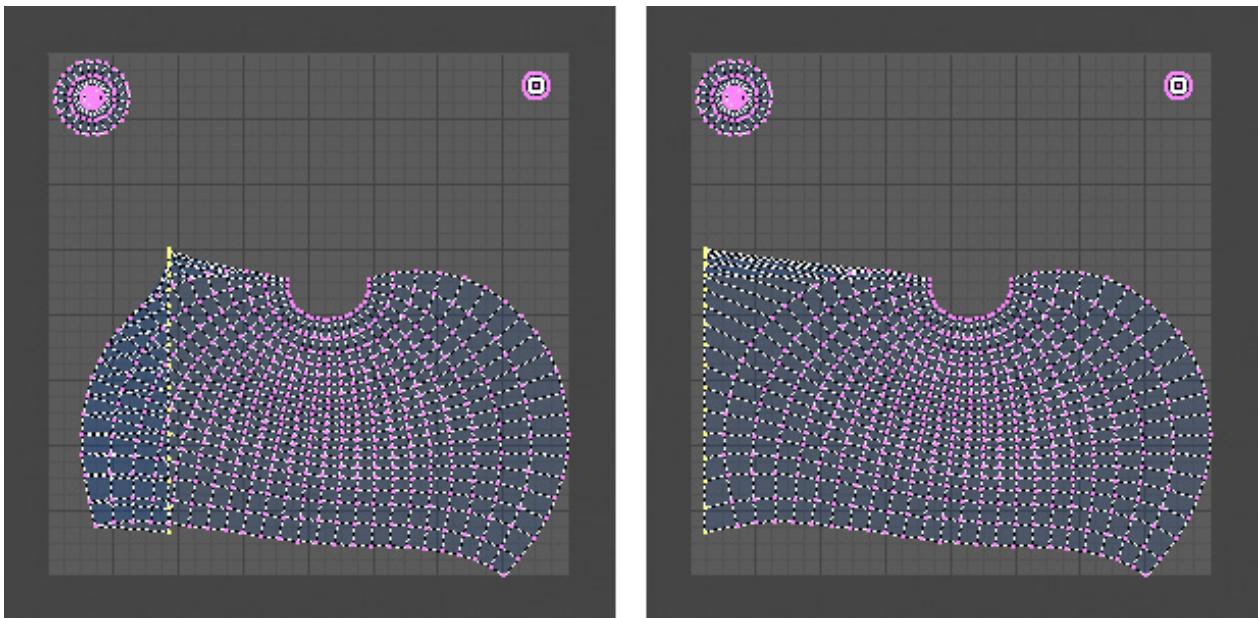
With all of your mesh selected, type **U** to unwrap. We can now see our unwrapped UV layout in the **UV/Image Editor** window. (For older versions of Blender, you might need to select **UV Face Select** mode to unwrap, but it's better if you download the latest.)



Those two islands we see are the cornea and optic nerve area. We can move them out of our way by selecting any vertex on them and typing **L** to select all linked vertices. Then **G** to grab as usual.

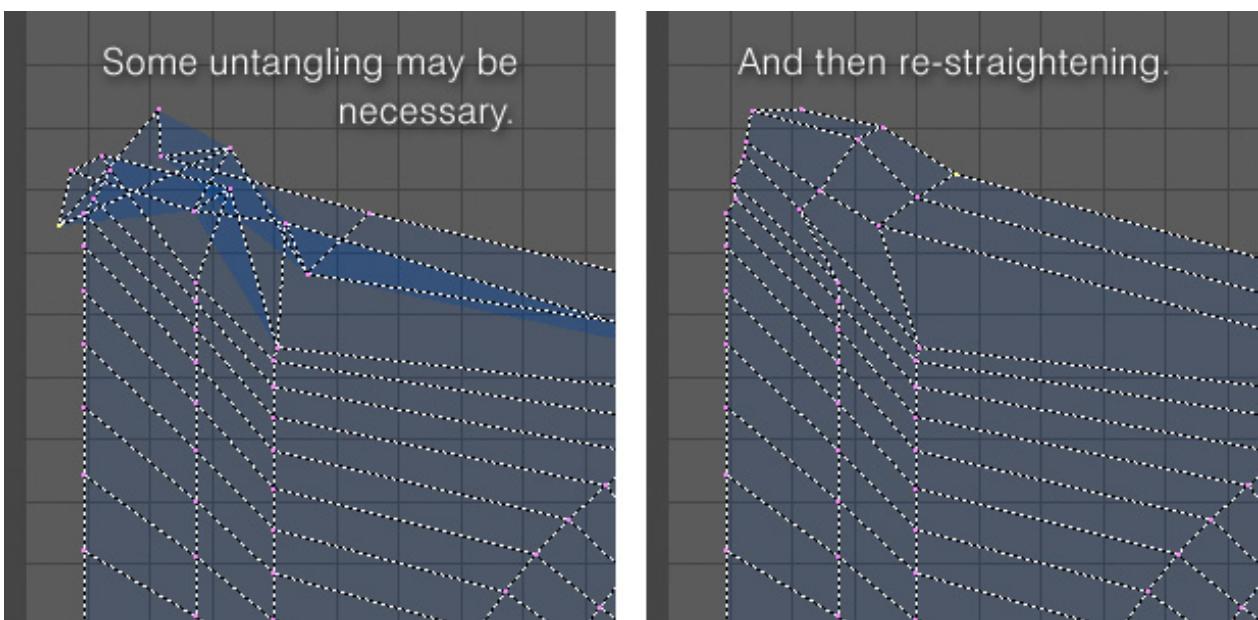


Select a column of verts on the sclera region of our UV layout. Type **W** to bring up the **Weld/Align** menu. Choose **Align X**. This is how we will go about straightening this contorted shape into a usable guide to map our textures.



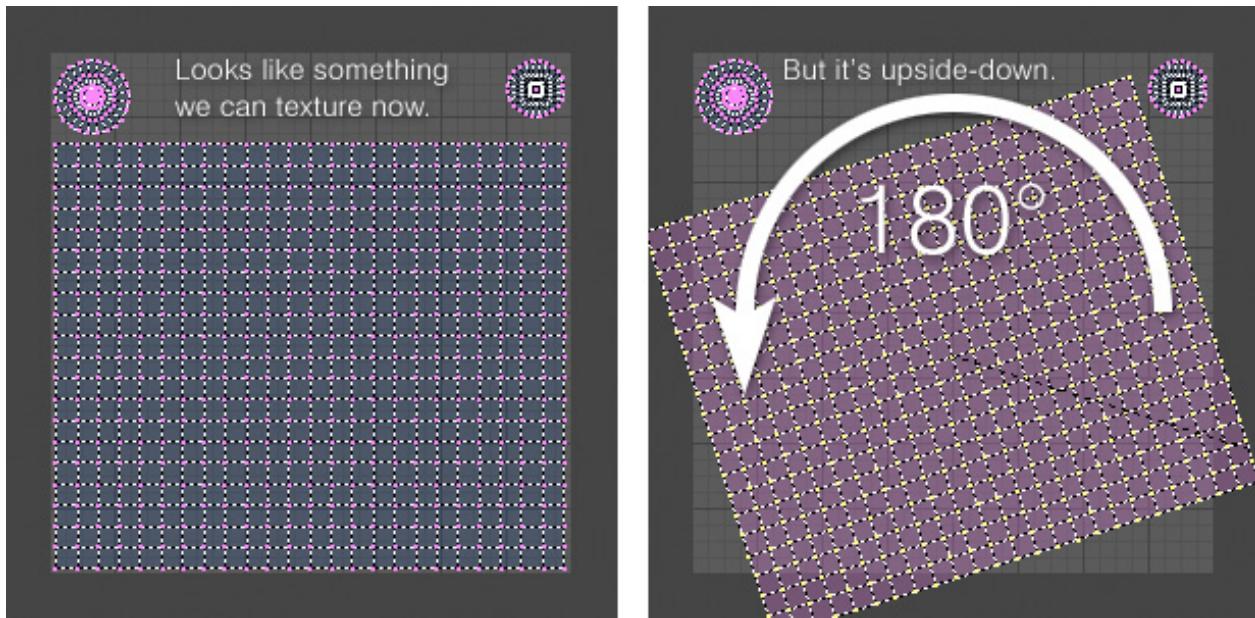
**Grab** and move your straightened column of verts over to the left.

Constraining to the X axis, by typing the "X" key after you grab, is a good habit to get into while straightening these out.



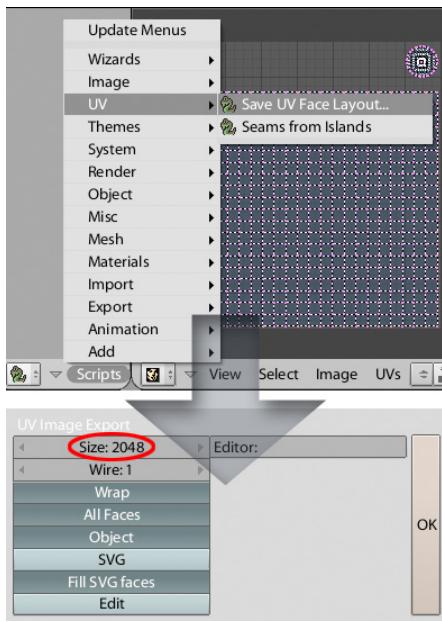
Straightening out all of these vertices into a usable grid shape will take you some time. It's one of the more tedious steps of this tutorial, but it's worth it in the end because having a rectangular grid will prove very helpful when we are texturing our eyeball. You'll want to use Blender's grid as a reference and just align the rows and columns to it exactly. Well, you can't get it *exactly* aligned, since there is no snap-to-grid feature, but you can eyeball it and everything will work out fine. The grid lines must be perfectly straight though, so always constrain to either the X or Y axis when moving anything. You can scale the whole thing up later, once each row and column is evenly spaced. Be sure to leave some space between your UV layout and the border of the layout area - don't "bleed" it to the edge or you will have texturing problems later.

[EDIT by nick\_chang207]: You can align them perfectly; all you have to do is select all the rows and columns one by one, (you can use Alt + R) and then select the axis along which you want them to be aligned. For example, select the top column and then press 'S' to scale, then 'Y' to align it to the Y axis, and then 'O' to align them. For the rows, do the same thing, only select the 'X' axis instead of the 'Y' axis this time. Its pretty easy this way.



It's important to pay attention to is which way the UV layout is facing. Chances are the top of the sclera region is actually what you will want to be the bottom. It doesn't technically matter which way it's facing, but you will probably want it to look intuitive when you are painting the textures. I had to rotate that piece 180 degrees to get the cornea pole on top and the optic nerve pole on the bottom.

Once you're done, **save your .blend file** and then **save your UV Face Layout**. This is done in the **Scripts Window: Scripts > UV > Save UV Face Layout**. The "UV Image Export" dialog box will appear. I recommend choosing a size of **2048** pixels.



## Preparing A Template For Texturing

Now launch your image editor and open the UV face layout you just saved. (You probably saved it as a .tga file, since that is Blender's default format for UV layouts) So we are now looking at a black and white UV grid. We want to get rid of the white regions so we can see through them to the layers below. The quickest way to do this is to set the blending mode to "multiply". If that works for you, go for it. Personally, I like my transparency to be represented by the checkered pattern, and for that we will need true alpha transparency. In Photoshop this can be done by going to the menu and choosing **Select > Color Range**. Using the eyedropper, **click any white part of the image, set**

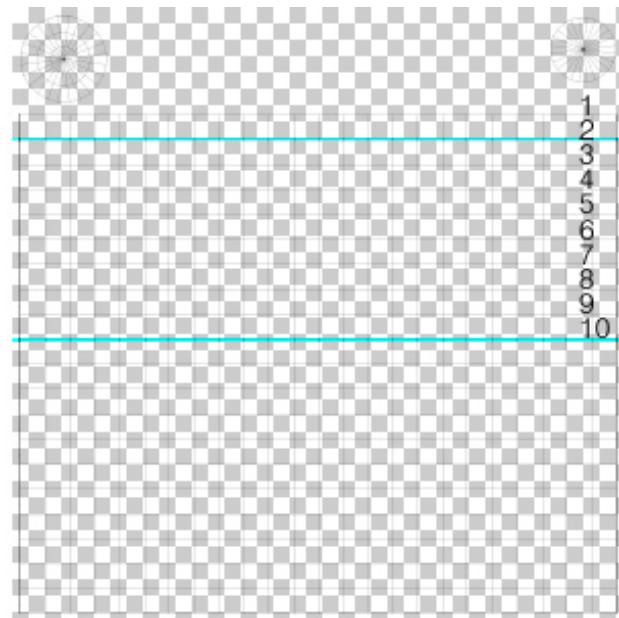
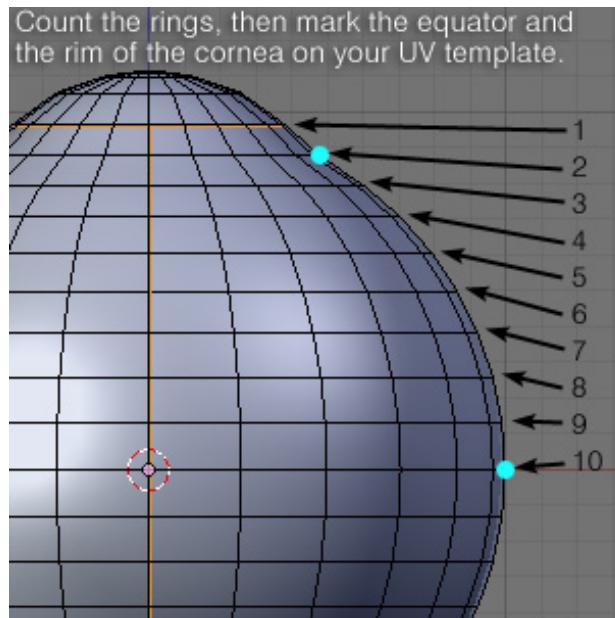
**Fuzziness to zero, and click "OK".**

Now **delete the selected white regions** by choosing **Edit > Clear** from the menu.

*nooby note: The same actions can be done in the GIMP editor by going to menu, choose Select > By Color and click any white part to select the white area. Then go to menu again and choose Colors > Color to Alpha and click "OK".*

*another nooby note: In PhotoShop you can also modify the layer's blending options so that areas of white or black will become transparent: When you first open the image in photoshop it will be a single layer image, and the layer name will be 'background' - Double click it and give it a name like "UV-Grid". THEN select the "UV-Grid" layer, Right-Click -> "Blending Options", in the bottom area of the dialog set the top value for the slider on "Blend If -> This Layer" setting on grey to be a little lower (most likely 254). This means any values on "this layer" that are greater than 254 will be blended with the underlying layers. - Make sure you create your other layers BELOW this layer so the grid will always be on top.*

One other element that will be useful to have in our template is a layer of markers that denote where the equator of the eye is, because what looks like the middle on the UV layout is not actually the middle. To find the middle, go back into Blender and count the rings of your eye mesh, starting at the seam of the cornea. Once you get to the equator, take note of how many rings it took to get there. Go back to your image editor and mark that area. You can mark it with a guide, or mark it with a coloring of some sort - as long as it's clear to you that it indicates the equator of the eye mesh. It may also be useful to mark the point where the transparency of the cornea should begin.

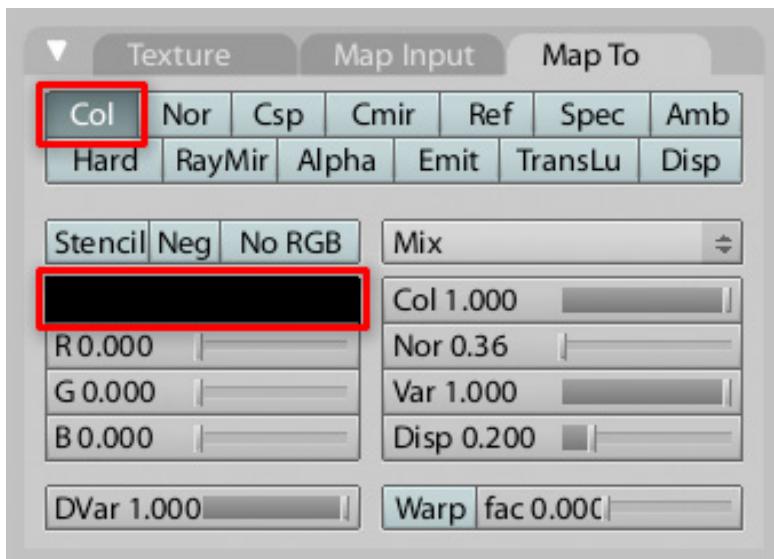


## First Bump Map - The Sclera

Create a material for your eyeball. Then within that material **create a texture**. Name this texture something like "*BumpMap(Procedural)*". Set the texture type to **clouds**. A noise size of **0.25** with a noise depth of **2 or 3** is fine, and you can experiment with other values if you like. The other texture settings can be left at their defaults.

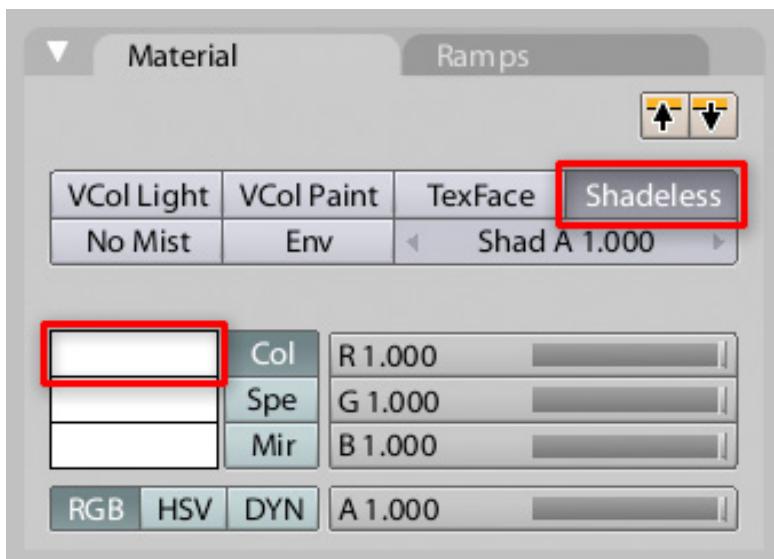
Now we are going to bake this procedural texture to get an image-based UV texture that we can edit in our paint program. Why? Because there is not an even level of bumpiness around the eyeball. The sclera has bumps, while the cornea doesn't. If your cornea was bumpy, your vision would be distorted. We want our 3D eyeball to accurately convey this subtle difference when light is reflected from the surface.

Before baking the texture, there are some parameters we must specify. In the materials panel click the **Map To** tab. The Color button is depressed by default, so you can leave it as is. Make sure this is the only depressed button. Then **change the color to black**.

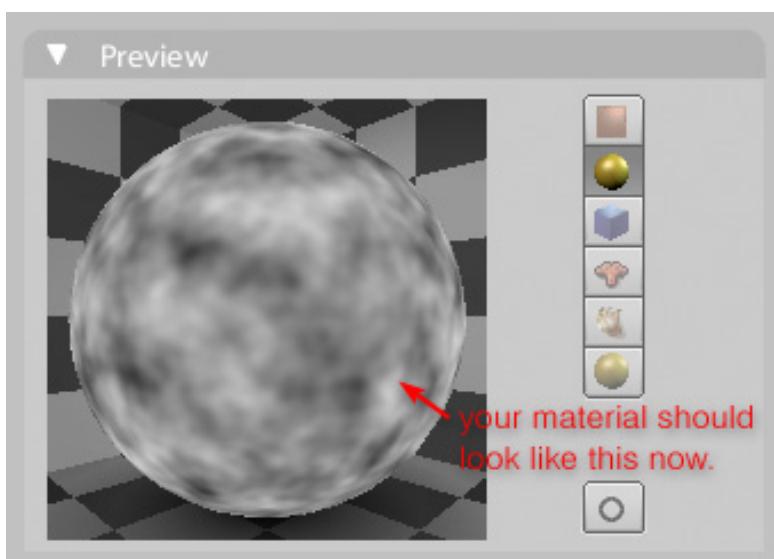


Now go to the **Material** tab. Change the diffuse color to **white** and click the **Shadeless** button.

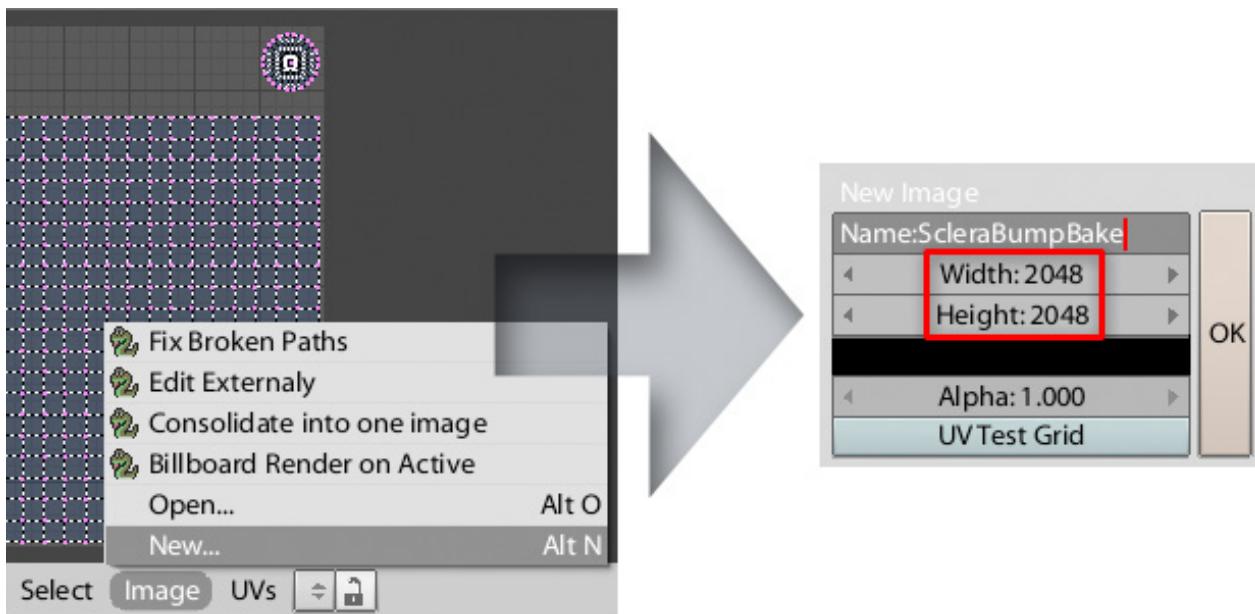
Note: In this particular case it doesn't actually matter which of these colors is set to white and which is set to black. The end result is basically the same, and you can always invert the image later.



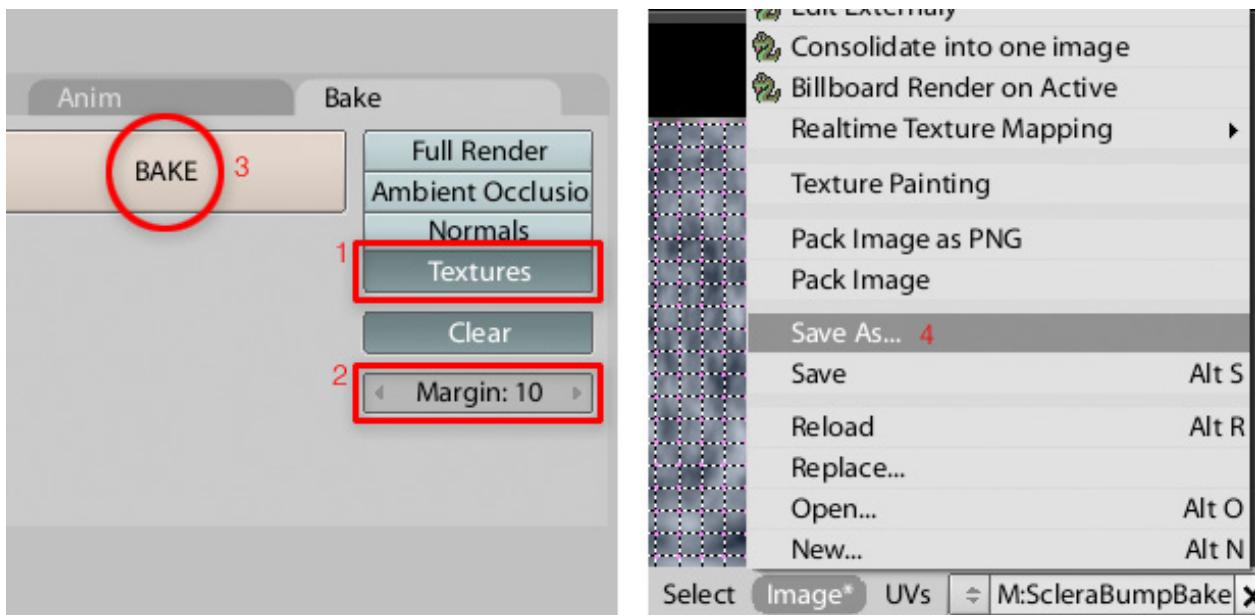
Check the Material Preview window to make sure it looks right.



Time to bake your eye. Make sure your eyeball object is selected and go into **UV Face Select Mode**. Then **split your screen** and set one viewport to the **UV/Image Editor**. From the menu at the bottom of the UV/Image Editor, choose **Image > New**. Enter a name like "ScleraBumpBake" and set both the Width and Height to **2048** (to match your UV guide). Click "OK" and a new all-black image will appear behind the UV grid.



Go to **Scene panel > Render > Bake tab** and click the **Textures** button. Then set the margin to **10** and click the **BAKE** button. You will see the black image in the **UV/Image Editor** fill in with your bump texture. From the menu choose **Image > Save As**.



Now would also be a good time to **save your .blend file**.

# Blender 3D: Noob to Pro/Light a Silver Goblet

## Techniques

You should know how to:

- Perform actions discussed previously in the tutorial.

This section will recap or introduce:

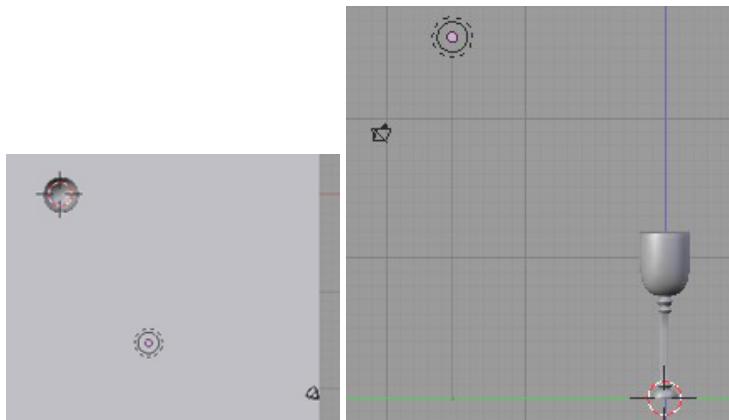
- Reflective material
- Positioning camera and light
- Editing the World colors

## Objects in the Scene

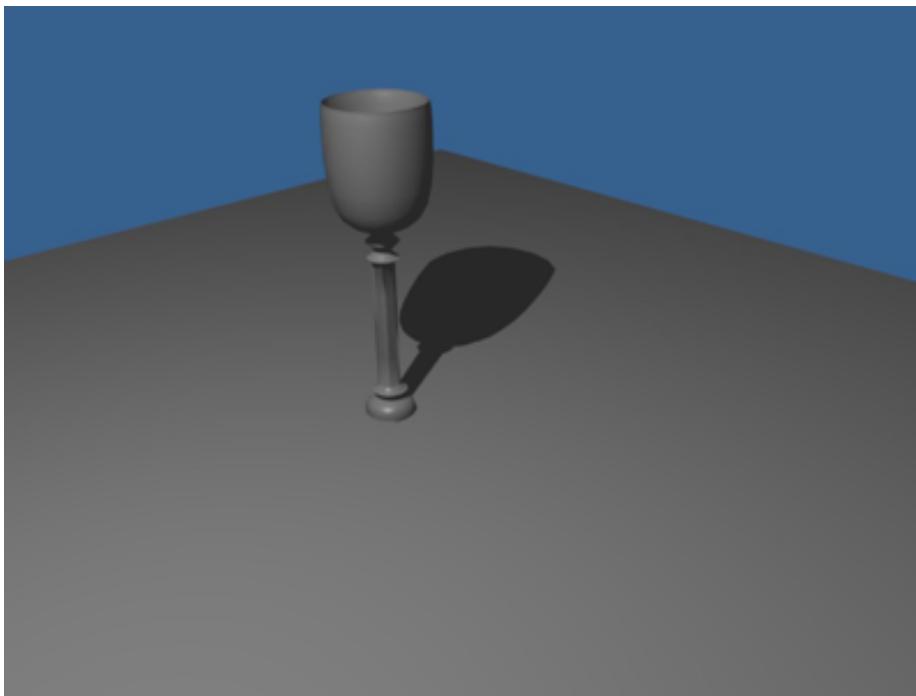
Create the goblet discussed in Model a Silver Goblet or load it if previously made. If you haven't already made the goblet, feel free to try the tutorial using a sphere or something else instead and you will still get a good outcome. In Object Mode with **NUM 7** view, add a plane mesh. Scale the plane to a very large size and make sure the goblet is sitting comfortably on top of it.

Select the camera and move it so that the goblet, its shadow, and its reflection in the plane will be seen. You can see the numerical location of the camera by bringing up the *Transform Properties* window by pressing **NKEY** in the viewport. In my example where 0,0,0 is the bottom center of the goblet, the camera is located at 27, -21, 19 XYZ with a rotation of 63.5, 0.62, 46.7.

Next, move the light so that it will create a shadow on the plane. Again, the location of this example is 9, -15.2, 25.8 for XYZ (rotation does not matter for this type of light source).



The rendering of this scene yields:



Note: You must turn on the Ray button in the Render (F10) buttons. Otherwise no shadows will appear!

## Adding the Atmosphere

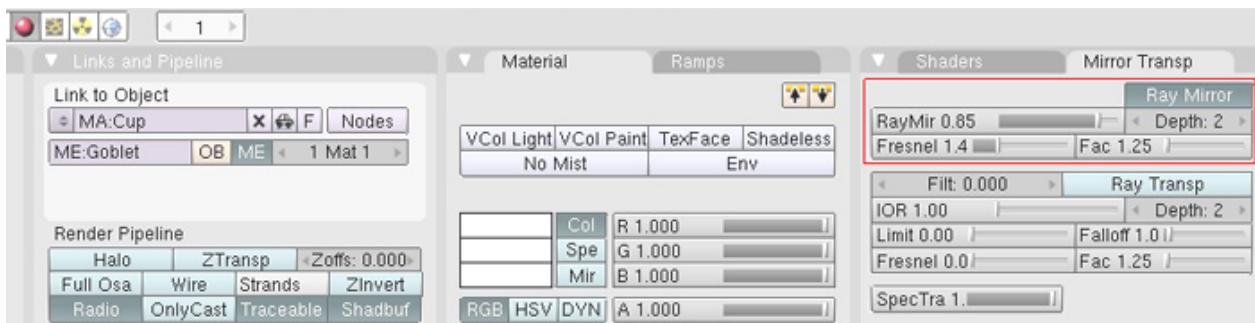
[Note: Many of the options here will be different if you have Yafray selected. In order to change to Blender's internal renderer, go to the Render tab (F10). Under the Render panel, click the drop down menu underneath the Render button. Select Blender Internal from the list.] [Note: On my version of Blender (2.49), the yafray button has been removed, so if you don't find it, don't worry.]

In Object Mode, select the goblet and go to the Shading panel (**F5**). If no material is linked to the goblet, add new material. Rename the material 'cup' or something similar. The area of interest is the *Mirror Transp* window. Highlighted below are the mirror options we'll be playing with. Press the Ray Mirror button to make the material act like a mirror and reflect light.

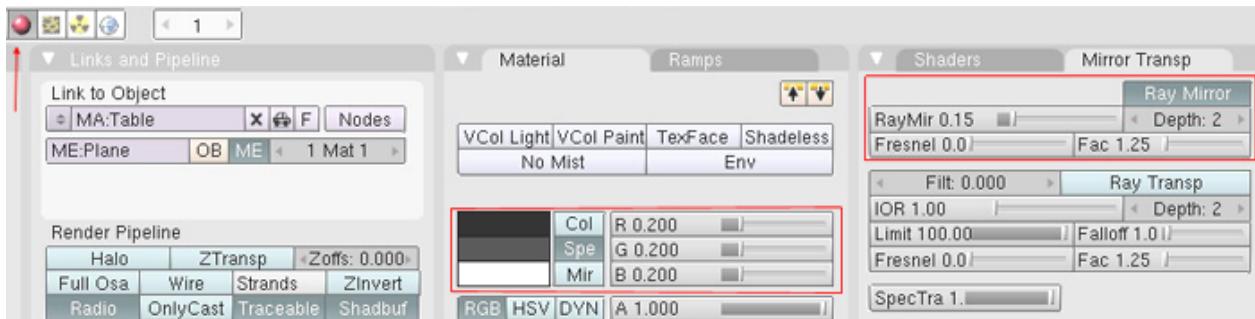
Move the RayMir slide to 0.85 or type it in after **LMB** on the number. This is how reflective the surface will be. A low number of 0.00 means that it reflects little while a high number of 1.00 reflects everything.

Also change the Fresnel slide to 1.4 from 0.0. This will increase the power of the Fresnel function. What this means is the color of the material will be strong because the light source is taken into consideration. If the Fresnel wasn't used, the object would appear dark because the light source isn't directly calculated in the mirror. Also, change the color of the goblet to white. Using a light color will give your goblet an interesting patina if you so choose.



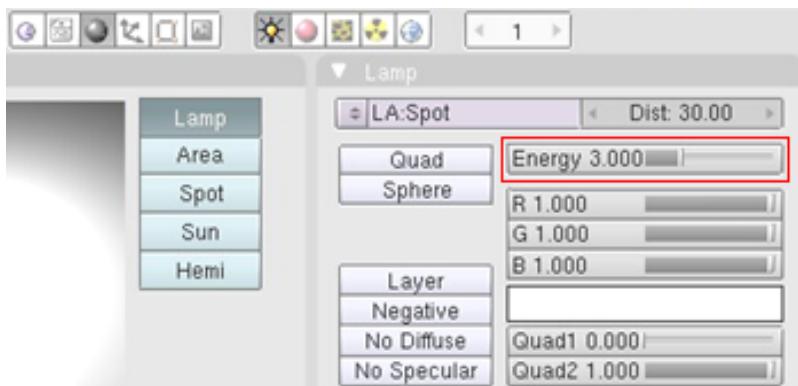


Next, select the plane and modify the material. We want the plane to be dark and shiny. Set *Col* and *Spe* to near black for the color. For reflectivity, turn on *Ray Mirror* to about 0.15 RayMir and ignore Fresnel this time.



The scene will still be quite dark so let's turn up the light. Sometimes you want to add additional light sources and other times you just want what you have to be brighter. Here we'll stick to the one lamp.

Select the lamp object and increase the Energy to 3.000 making it 3 times brighter than before. As always, try changing it to several levels to see how bright or dark you can make the rendering.

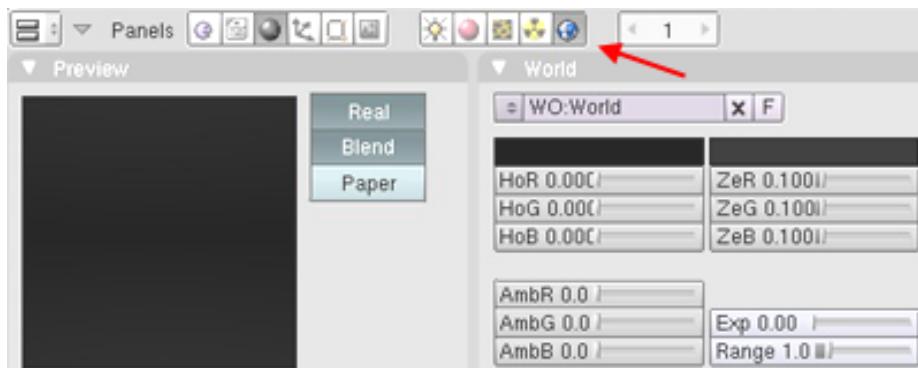


There is only one more thing to do before rendering the scene: change the world. Under the Shading panel there is a globe icon that will get you to the world buttons. Here you have *Real*, *Blend*, and *Paper* buttons under the Preview window. And under the *World* window are options for changing the color of the horizon (Ho RGB), zenith (Ze RGB), and ambient (Am RGB). We're interested in these two windows at the moment.

Using *Real* and *Blend* will affect the way the horizon and zenith interact. Experiment with them to see what they do in the preview. In this example, *Real* and *Blend* are turned on.

The *Paper* button works a little differently in that what you see in the preview will essentially be the background of your render. This effect is most noticeable when your camera is rotated. Despite the camera rotation, the preview would still be 'wallpapered' on the render.

For our world, set the RBG close to 0.00 for the horizon, zenith, and ambient.



That was the last step! Make sure the camera is in the right spot and render the scene. Here is the output of this example





Problem: When I render I see brown where I have used "Shift-E" and "G" and I followed everything.

Please remember, the author did say play with the settings a bit, this tutorial should be used as a guide. We are learning how to use all the tools that Blender has to offer, that is the important thing. It is up to us to experiment more with the settings set forth in these tutorials. If the settings in these tutorials do not give us the same results, that is OK, we should be changing them anyway to express ourselves! Also remember, NEVER strive for PERFECTION, but ALWAYS strive for EXCELLENCE. Perfection only leads to frustration, and it is frustrating enough, at times, to learn something new. Have fun learning, I know I am.

(:Noob note)-that was touching and true

(Noob note 2) you just changed my life :D

# Blender 3D: Noob to Pro/Every Material Known to Man

---

## "Every Material Known To Man" Project

Want to share your material settings with the world? This is the place! This page is going to be turning from a page to a chapter, but to do that we need your support!! If you have an idea for a material, even if you don't know how to make it, just put it in the list to help! Someone out there knows how to make it, and if they come across this page with it listed, they might be willing to share.

### Etiquette

~~Explain the basics of every material (such as Color, Spec, and shading, but not necessarily specifics such as Emit or Trans), even if it is with standard "Add New" properties.

~~IF you provide a picture, explain in the method all details to get us to make ours look the exact same way, do not omit any detail (including anything done in out tabs, or in other categories like Rendering. Also, say if you use something like RayMir or if you have additional lighting or objects to enhance the look!).) ~~Make sure you explain the "Tab" and the "Category" for a material alterations, because everyone might not know what you mean. Be as formal as possible, categorize and organize, and try using the blender-used words instead of your own lingo, if possible.

- EX

- Bad explanation: Set the material's texture's normal to 0.4
- Good explanation: Under the "Map To" Tab, Select the "Nor" Slider and set it to 0.4
- Best explanation:

-Map To:

--Output-Nor

--NorSlider-0.4

~~Do NOT replace or delete other people's method on your own. If you believe you have a better way, simply label their explanation "Method A" and create another sub-headline with your method as "Method B" (and so on). If you believe a method is incorrect, insufficiently detailed or plagiarism, appeal to the original author (if able) or on the Discussion page and get support for a removal.

~~Do NOT say your material is better than anyone else-- everyone has a right to their own opinion and preferences --you may say that your method is more "detailed" or "memory-efficient" or "adaptable", but DO NOT SLANDER other people's methods.

## Metals and Minerals

- /Alloy/
- /Asphalt/
- /Brass/
- /Bronze/
- /Cast Iron/
- /Ceramic/
- /Chrome/
- /Glass/
- /Gold/
- /Iron/

- /Concrete/
- /Copper/
- /Hard Plastic/
- /Linoleum/
- /Marble/
- Reflectives
- /Rubber/
- /Rust/
- /Steel/
- /Tile/
- /Trampoline material/

## Cloth and Fabrics

- /Cotton/
- /Silk/
- /Leather/
- /Wool/
- /Carpet/
- /Denim/
- /Mothball/

## Particle-Based

- /Fire/
- /Smoke/
- /Fog/
- Fur
- /Rain/
- /Mist/
- Grass (Patch)

## Earth and Home

- /Moss/
- /Water/
- /Soil/
- Grass(Strand)
- /Woods/
- Textured Ceiling
- Textured Wall
- Paint (Matte)
- Paint (Gloss)
- /Sand/

## SSS-Used

- /Crayon/
- /Wax/
- /Fruit Skin/
- /Milk/
- /Juice/

## Other

/Lunar Surface/

## Abstract

- Light Effects
- Misty Globe

## Human

- African Skin
- Latino Skin
- South Asian Skin
- Caucasian Skin
- /Head Hair/
- Other Hair
- /Teeth/
- /Bone/
- /Blood/
- General Organ Skin

## External material libraries

Additional materials can be found at:

- Blender Open Material Repository <sup>[1]</sup>
- Copper, Brass, Platinum, Silver <sup>[2]</sup>
- Fire, Snow, Carbon Fibre <sup>[3]</sup>
- Glass, Chrome, Gold, Brushed Alloy <sup>[4]</sup>

## References

- [1] <http://matrep.parastudios.de/index.php>
  - [2] <http://forum.nystic.com/viewtopic.php?f=34&t=1494>
  - [3] <http://forum.nystic.com/viewtopic.php?f=34&t=1538>
  - [4] <http://forum.nystic.com/viewtopic.php?f=34&t=1456>
-

# Blender 3D: Noob to Pro/Beginning Lighting

Lighting, you say? Pssh. 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.

## Blender 3D: Noob to Pro/Faked Gi with Blender internal

**Getting uniform studio lighting with the blender internal renderer**

### Preamble

This tutorial will teach you how to use blender to create faked Global Illumination (here forth called "GI"). The reason it is faked is that blender currently doesn't support true Global Illumination but it's not a difficult task to fake it. The results of this tutorial should look like this:



This tutorial assumes you know the basics of the blender interface and how to add objects, toggle editmode, and scale objects.

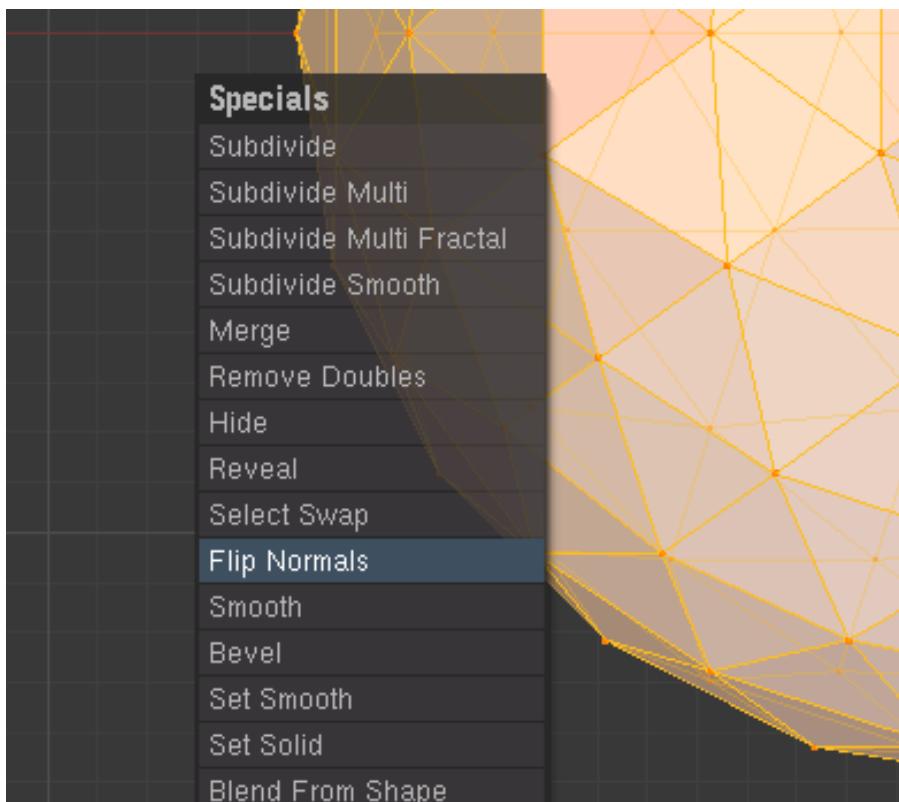
Now lets start blendering!

## Blender Faked GI Tutorial

**Add an ICO-sphere.** Fire up blender and add an icosphere(**SPACE** → **add** → **mesh** → **icosphere**). Set the subdivisions to 3 and accept.

**Scale the ICO-sphere** by 15 times

**Flip the normals of the icosphere so that they point inwards.** Go into edit-mode and press the A key till all faces/verts/edges are selected and press **WKEY** and *Flip Normals*.



**Add a sun light source** (**SPACE** → **Lights** → **Sun**)

**Set the energy value of the light.** This requires some special attention. If you keep the value at the default 1, you get a pure white, washed out scene because we will duplicate the light 162 times. A good way to calculate the light intensity required is to mess with the one sunlight and do test renders while tweaking the energy value of the sun as required till you get the brightness you want. THEN you divide the energy value of your sun by the amount of vertices in your icosphere and then finally set the sun energy to the number you got.

Noob Note: I found I had to set the energy of the sun lamp extremely low, at .044. Otherwise, you end up with a completely washed-out image.

**Parent the sun to the sphere.** Select the Sun THEN shift select the sphere and hit **CTRL+PKEY**

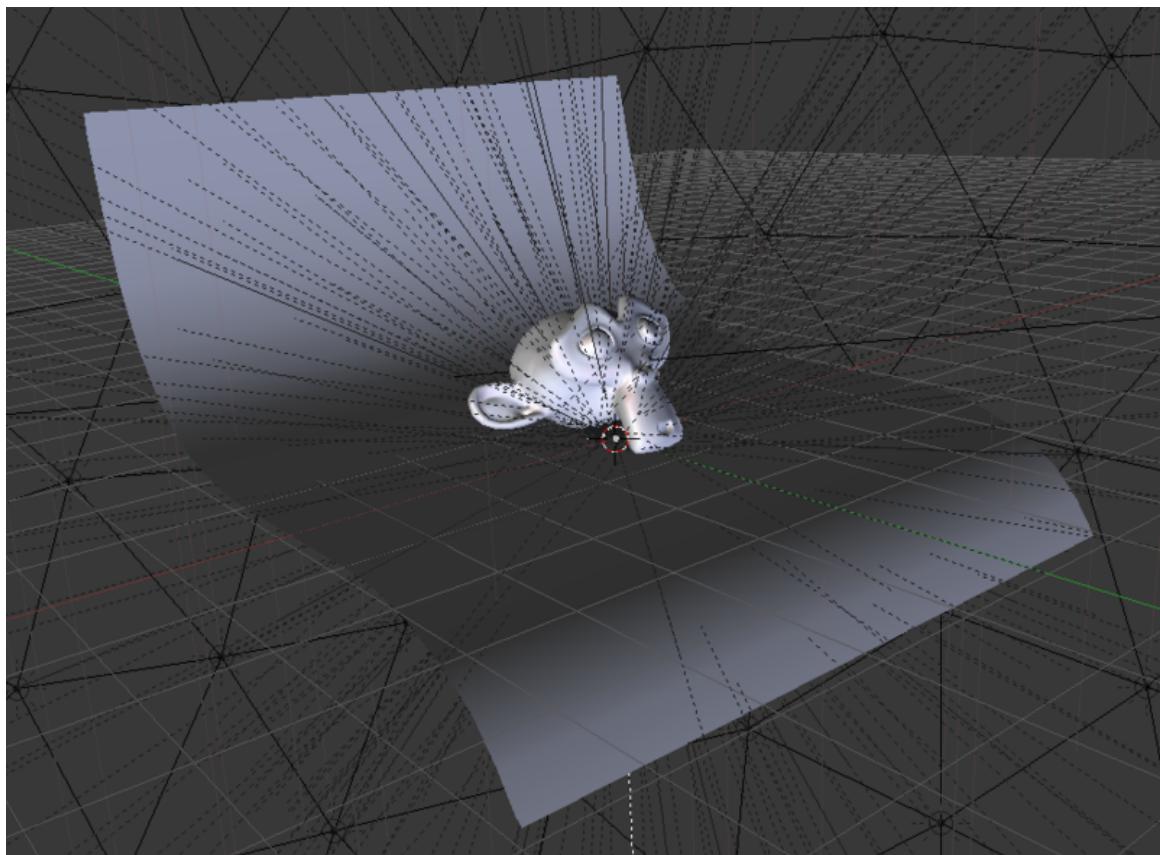
**Enable dupliverts for the sphere.** depress the DupliVerts and the ROT button.



**Enable AAO (Approximate Ambient Occlusion)** Set the AAO settings in the world buttons as in the following image:



**Set up the scene to render** This can be anything you want but I used a simple curved plane with Suzanne on top of it.



One more note: Often it is hard to see your scene what you have this huge sphere encompassing your scene so what you can do is to set the draw of the sphere to wireframe.



just one last note: if you want more directional lighting, select the sphere and go into editmode and press **CTRL+TAB+3KEY** for face select and select one face on a spot of the sphere and press **WKEY** and subdivided a few times to get more lights in one spot making that side a bit brighter and a stronger shadow on the far side of that cluster.

I hope you found this tutorial useful. If there are any errors in the tutorial, *Please* let me know, and I will fix them as soon as possible.

Blend happy and save often, --Radialronnie (talk) 22:47, 3 November 2008 (UTC)

---

Noob question: I followed this tutorial word-for-word but at the end the lights (on the vertices) are pointed strange ways. Is there something I should do to the sun before I click the Icosphere and hit **CTRL+PKEY** (i.e. positioning or rotation)? I'm going to try to use a lamp and see if that works at all. Update: Facepalm. I went through the tutorial again and found that I messed up last time. Let this be a warning to everyone: **DON'T** rotate the sun.

Noob comment: Perhaps you could clarify the phrase "Global Illumination" at the start of this page. I was expecting an explanation on how to fake true global illumination (ie radiosity), not just a constant light source. -The Author, well, would you like me to call it genuine faked GI? If you want the features you're talking about, you would need to use <http://yafaray.org/>

## Blender 3D: Noob to Pro/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. If you notice an error partway through a render, then you can press escape to abort.

### The Scene Tab

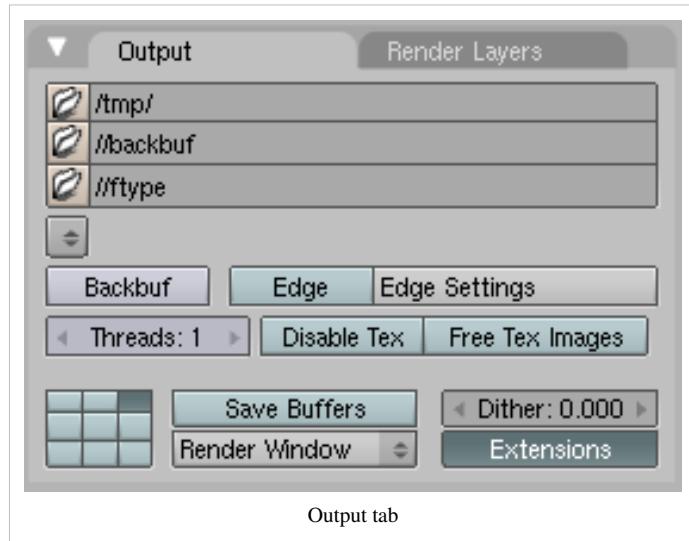
Let's take a look at all of the rendering options that Blender has. Open up a new scene (**CTRL + X**) and press **F10**. You should see something that looks like this:



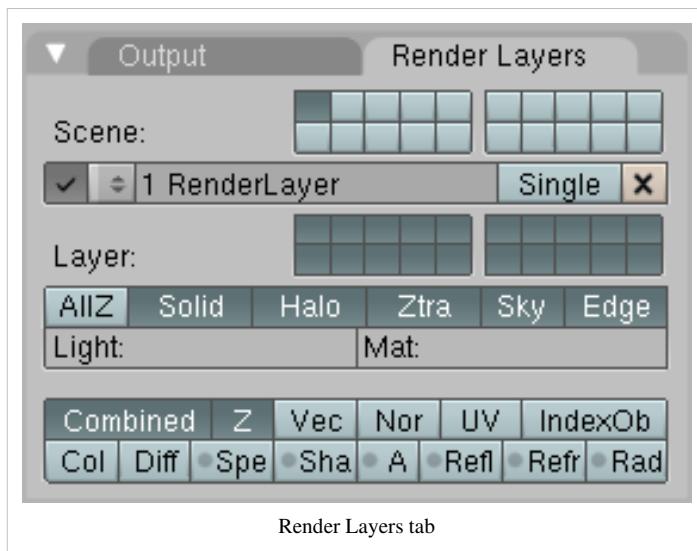
These are the default settings for Blender. Let's take a look at all of these tabs and see what they do.

## Output / Render Layers

This is the first tab on the left. First of all, we see a file locator icon (  ) to the left of `/tmp/`. This is Blender's temporary directory. (Normally `C:/tmp` on Windows) Below that, we see another file locator icon, labeled `//backbuf`. This is the Back Buffered image, one way to make an image the background on your render. Simply use that file locator to find a file, and Blender will use it as the background image (the `backbuf` button must be clicked to activate it). Finally, the last one is `//ftype`. The Blender description reads "image to use with FTYPING image type."



Output tab



Render Layers tab

the size of the edge that is rendered, as well as the color. The next button we are going to talk about is the 'Threads' button. Clicking this enables threads, which allows the renderer to do threaded processes. Basically, if you have a dual-core processor (Athlon X2, Intel Core2Duo, etc.) it will have one processor render part of the image at the same time that the other processor is rendering another part. This (generally) helps reduce rendering time. Finally, the last button on this row is the 'Free Tex Images' button, which "Frees all images used by textures after each render".

There is one last row of buttons below.  The bunch of little squares enable you to tell Blender where to put the render window on your screen. Try pushing one and then rendering to see for yourself. The "Save Buffers" button saves all the tiles from every render layer and SceneNodes to files, saving memory. Experiment to try it out. The dither button will put dither noise in your image. The render window button allows you to render your image in a new window, render it full screen in the UV Image Editor, or render the result in the UV Image Editor (but not in full screen), a good idea when working with nodes. The extensions button adds extensions to your animations.

The next row down, we see this:



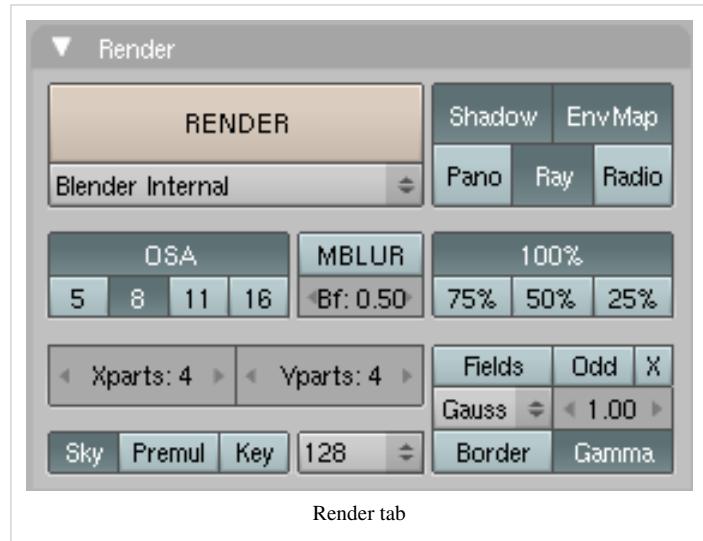
"What

do these do?!" you ask. Well, it's actually quite simple. First of all, the first button is the `backbuf` button that I just talked about. If you have selected a background image to render using the `"backbuf"` option, clicking this button will make it show up. Next, on the right, are the 'Edge' and 'Edge Settings' buttons. Clicking the 'Edge' button will render your image with edges. Edges are rendered where the camera sees an edge. For an experiment, turn it on and render to see what happens. Notice how the default cube now has a black edge around it. If you want it bigger, click the 'Edge Settings' button. Here, you can change

## Render

Now, let's move on to the render tab! The first thing you should notice is the VERY BIG button that says 'Render' on it. This button performs the same action as the **F12** button. Below this is a drop-down box, allowing you to change between the Blender Internal Render engine and YAFRAY. (For more on YAFRAY, see YAFRAY Render Options). Next to the 'Render' button are 5 buttons: Shadow, Env Map, Pano, Ray, Radio. These are explained in detail below:

- **Shadow:** Lights will cast shadows. If this is turned off, nothing will cast a shadow. To see this, add a plane below your default cube and render. Now turn this button off and see how there are no shadows.
- **EnvMap:** Use the environment map.
- **Pano:** This enables panoramic rendering. For more on panoramic rendering, see Panorama Settings
- **Ray:** This enables Ray Tracing. Turn it off, and render the same images as I talked about in the description of the Shadow button. "What!?" you say. "That did the same thing as turning off the shadows! What's the deal?" Well, ray tracing is used to calculate how light goes through objects like glass and stuff. It also has something to do with shadows.
- **Radio:** This stands for Radiosity.



Render tab

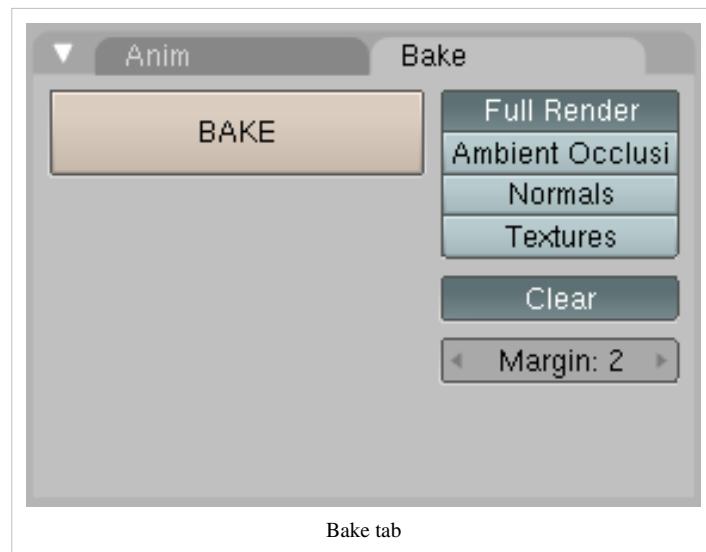
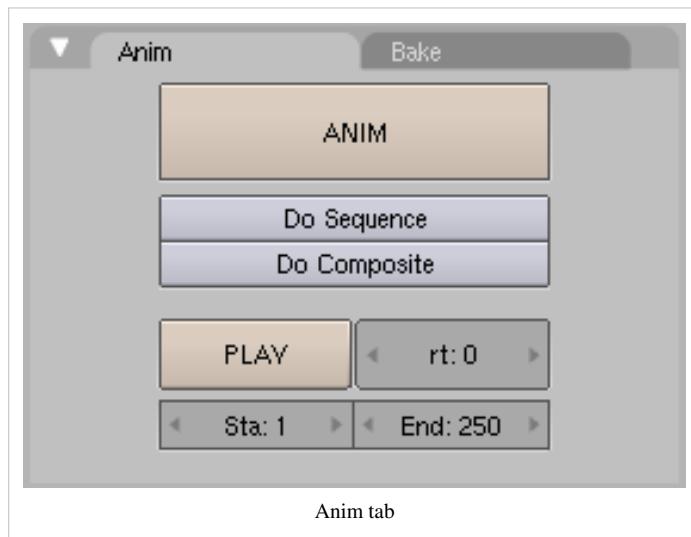
Moving on, the next button we come to is the 'OSA' button. For the use of this button, see OSA

To the right of the OSA button, are some percentages. These will set the renderer to render at the specified percent of the resolution that you have set. (We'll come to resolution later) For an experiment, press the '50%' button. As you can see, Blender renders the image at 50% of the set resolution. This is an easy way to make a quick render.

Next on our tour, we come to two boxes labeled 'Xparts' and 'Yparts'. These buttons change the number of boxes that Blender will render in total. Play around with these to see how they change the number of small boxes in the render window when you render.

## Anim / Bake

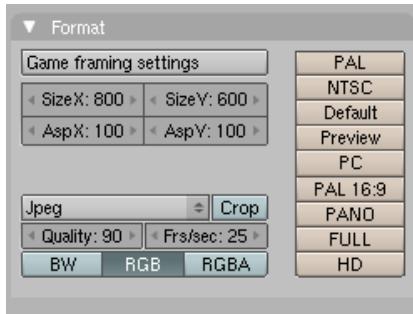
The final tab we'll (briefly) cover is the Anim(Animation) tab. If you have made an animation, pressing the 'ANIM' button will render your animation. Right below it are two buttons, Do Sequence and Do Composite. The Do Sequence button will render your animation with any effects you have done in the Sequence Editor and the Do Composite button will render an animation with your composite node work included. The Sta: and End: are the start and end frame numbers. The default setting is that your animation will start at frame 1, and will end at frame 250.



# Blender 3D: Noob to Pro/Output Format Options

---

## Output Format Basics



When you render a scene in Blender, by default a small window will open 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 tab (usually on the right).

You will see some settings there. Let's go through their meanings:

- **SizeX: 800**: This parameter sets the width of the image in pixels.
- **SizeY: 600**: This parameter sets the height of the image in pixels.
- **AspX: 100**, **AspY: 100**: 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.

Don't forget, setting the number of pixels changes the resolution. Resolutions for your computer desktop are normally 1024x768, but can be larger. This means that it is 1,024 pixels wide and 768 pixels tall. To make an image this size in Blender, change the **SizeX** to 1024 and the **SizeY** to 768.

If this does not come out correctly, then you should right-click your desktop and click Properties, then go to Appearance. The highlighted numbers are X by Y.

## Output formats

Blender has the ability to save in a wide variety of formats. To change the format, click the drop-down box (**PNG**) halfway down the format tab. Currently, Blender can save files in the following formats:

- TIFF
- OpenEXR
- DPX
- Cineon
- Radiance HDR
- Iris:
- Iris + Zbuffer:
- HamX: extremely compact but only for the "Play" option
- Jpeg: default format. Lossy compression.
- BMP: uncompressed.
- PNG: open, lossless compression, alpha channel.
- Targa Raw: uncompressed Targa.

- Targa: Compressed Targa.

**Single rendered images are not automatically saved to disk. Once Blender finishes rendering a scene (this can take some time), press F3 and the save dialogue will be opened where you can specify a filename and location for the output. You can retrieve the last picture rendered by pressing F11.**

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: saves a QuickTime MOV file. Once selected, a pop-up menu will appear giving various video codecs to choose from. Can also remain uncompressed.
- FFmpeg: Implemented in a Google Summer of Code event, this codec is composed of multiple software libraries, so you can choose which codec to use under the video tab. The audio tab lets you choose from several different audio codecs to use and also has a bitrate setting. The Multiplex Audio button will render your animation with sound (if you have any), a nice feature.

For more information, see

- Blender 3D: Noob to Pro/Creating animated GIFs using Blender and Gimp
- Blender 3D: Noob to Pro/Creating Ogg-Theora movies using Blender

## Cropping

You can see  button next to drop-down box. If you activate it, image automatically gets cropped after rendering.

## Palette



You can select three different palettes in Blender:

- **BW** - images are saved with BW (greyscale) data
- **RGB** - images are saved with RGB (color) data
- **RGBA** - images are saved with RGB and Alpha data (if supported)

## Pre-sets Just For You

PAL	NTSC	Default	Preview	PC
PAL 16:9	PANO	FULL	HD	

The render settings in blender are customizable for unique results, but blender includes 9 pre-sets for common render outputs. One can use the pre-set as is or can use it as a starting point at which the settings are then modified.

**PAL** and **NTSC** are systems used in analogue Television. If someone were making an animation for TV or DVD, that individual may use one of these depending on which is used in their country. Both the PAL and NTSC pre-sets in Blender only affect the dimensions and pixel aspect ratio of the render and doesn't turn on anti-aliasing (OSA) or affect any other variable in the render. **Pal 16:9** is a newer an improved version of PAL which is also known as PALplus.

The Default pre-set uses the dimensions and an pixel aspect ratio as PAL but also includes OSA and shadows.

The **Preview** pre-set generates a low quality setting for a fast render that usually is not used as a final output.

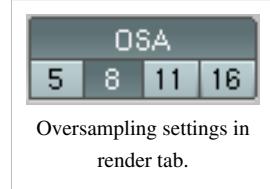
- **PANO** is short for Panorama. This setting allows for a full 360 degree panoramic render.

- **FULL** produces a render that fits your monitors full screen.
- **HD** is for High Definition.

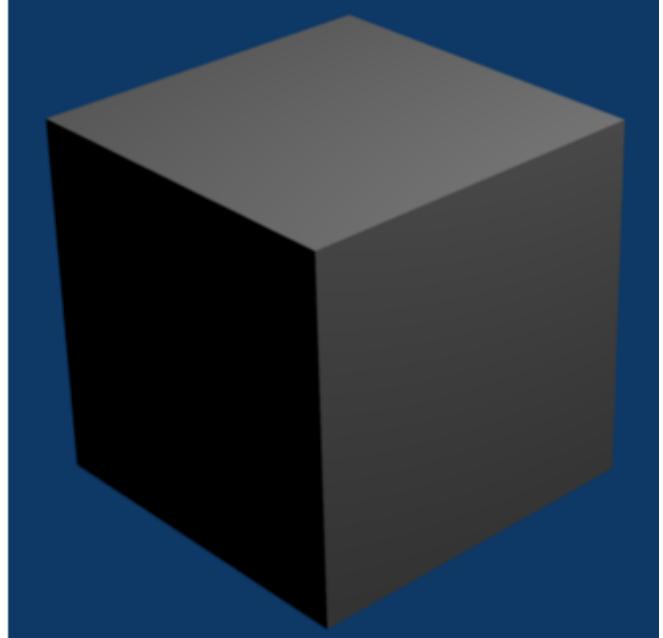
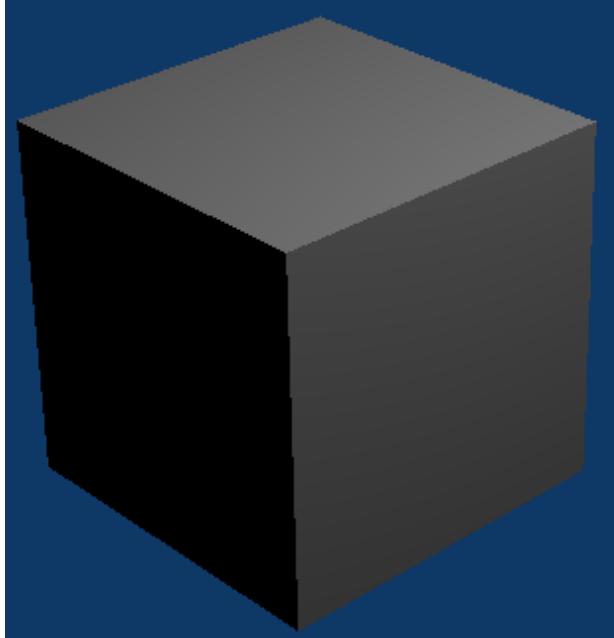
## Blender 3D: Noob to Pro/OSA

OSA stands for *oversampling*, also known as *anti-aliasing*. It prevents "jaggies," or "aliasing" as it's called.

This occurs when you have a diagonal change of colour, which results in rough edges. Remember drawing a diagonal line in Paint? To overcome this hindrance of square pixels, a technique called anti-aliasing or "oversampling" is used. It blends the colours around the rough edge to create a smooth, but defined edge. One way of doing this is to by creating 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).

OSA settings can be manipulated in from the Render Settings panel.

# Blender 3D: Noob to Pro/Looking All Around - Panorama Settings

## Panoramic Renderings

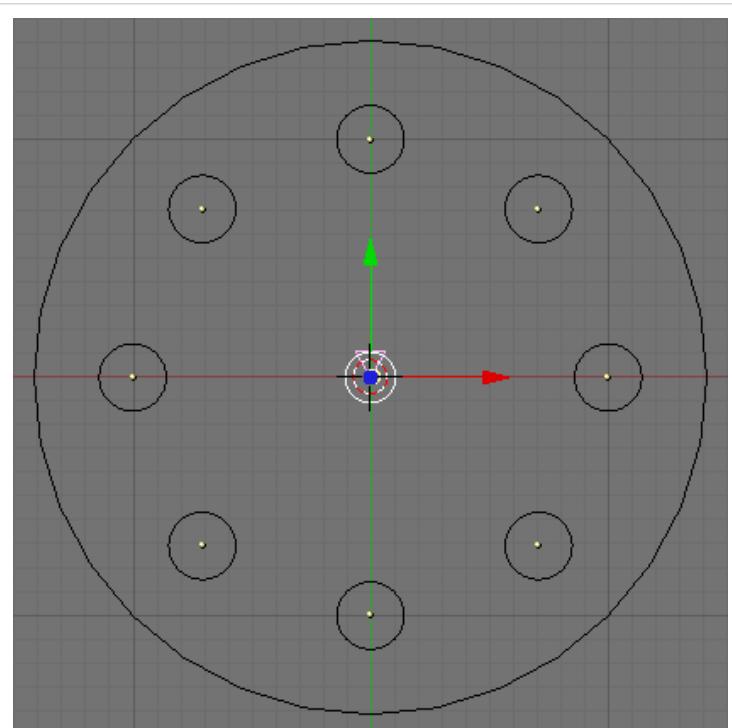
### This Article is Out Of Date

**This article is out of date**, and the panoramic features mentioned **do not work this way** in recent Blender versions. Panoramic features were changed somewhere between 2.37a and 2.42. More information, including how to actually use panorama, can be found in this thread in the forums: <http://blenderartists.org/forum/showthread.php?p=797886>

### Building the Example Scene

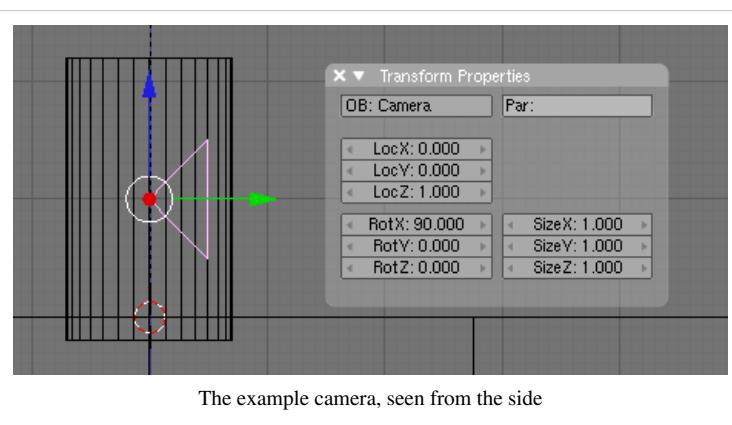
Ever wondered how that cool looking 360 degree panoramas that you see on some websites are made? Well, I don't know either, so someone else will have to tell about that. What I found out though is how to make Blender output a 360 degree panoramic image - quite probably this at least is the first step 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

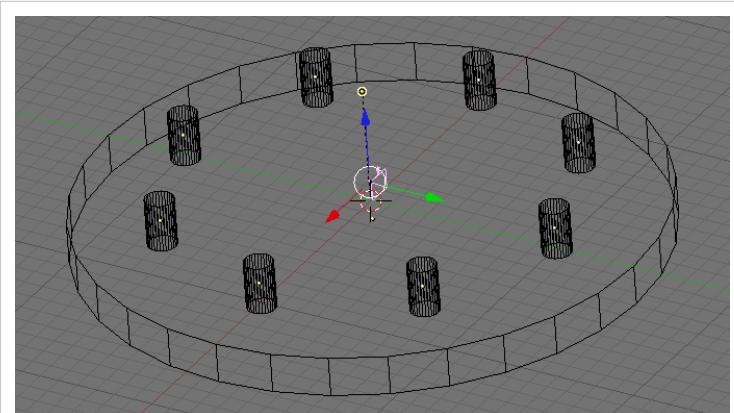
I placed the camera and the light both above the origin, deleted the cube, and added 8 cylinders, all around the origin. One way to do this is to place one cylinder, then duplicate and rotate it around the origin. For this to work, set the rotation center to the 3D-cursor (with the Pivot button), and 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**



The example camera, seen from the side

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.

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. Below 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 seamless 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**).

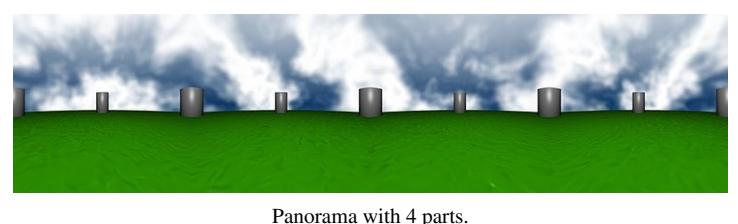
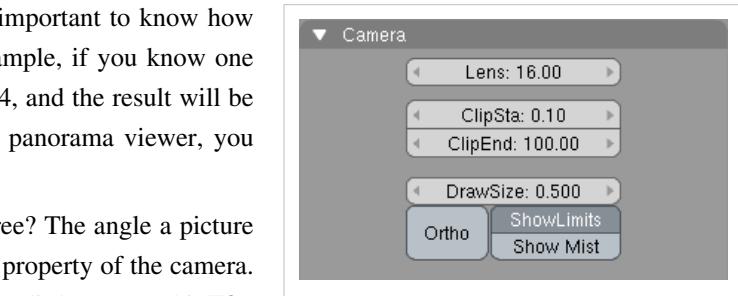
Make sure you haven't set your camera to orthographic (the "Ortho" button), since FOV only exists with a normal camera.

You can directly enter the FOV in Blender's camera settings (as of version 2.42) - Toggle the 'D' button on, next to the parameter called **Lens**. Type in 90 to get a 90 degree FOV. 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 90, 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 seamlessly. 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:

**Warning: Didn't convert following lens settings to the new possibility with the 'D'-button for passing the FOV value directly.** 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



Panorama with 4 parts.

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:

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 calculate Lens when we know FOV:

$$\text{Lens} = 16 / \tan(\text{FOV} / 2)$$

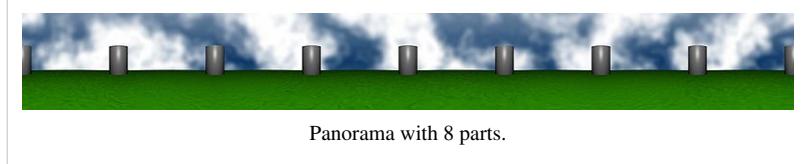
For a FOV of 45 degree, we therefore get a Lens of 38.627. Now, write that into the Lens field in the camera settings, and render with 8 XParts. This time it will render an output image which is 8 times as wide as a normal output image, and again it will be a seamless 360 degree all around panorama of 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 calculate 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.

As always, feel free to modify this page in any way you want or add feedback to the Talk page.



## Blender 3D: Noob to Pro/Other Important Render Options

---

### 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** - Tutorial Here <sup>[1]</sup>

Blender has an option of adding a border – a defined 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.

Note: User Terrywallwork - The edge bordering feature in Blender 2.43 only seems to work when Blender Internal renderer is used, when Yafray is used, no edge is show even if the options is clicked. I'm not sure if there is a way round this or not.

**HDRI Rendering** - Tutorial Here <sup>[2]</sup>

### Multi Thread Rendering

Blender has the option to render with 'Multiple threads' meaning it can render more than 1 part at once. This is done by sending more than 1 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). Change the number of threads used by changing its number.

**Note in Blender 2.46+:** Blender 2.46 automatically selects the number of threads.

## References

[1] <http://forum.nystic.com/viewtopic.php?f=34&t=3102>

[2] <http://forum.nystic.com/viewtopic.php?f=9&t=5355>

# Blender 3D: Noob to Pro/Adding Lamps

---

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:** Single point light source. This simple light source can cast sharp shadows, but is best for filling in dark areas without shadows turned on.
- **Sun:** A light with parallel rays that will illuminate the scene with a bright even light. Often a good starting point for outdoor scenes.
- **Spot:** Spot lights produce light in a cone shape, and have some special features. They are the only light source that can be made visible with the 'halo' option, to simulate light in a fog. They are also the only light source that cast fake shadows, called buffer shadows. Unlike raytraced shadows , these buffer shadows are smooth, but not as accurate. They are FAST though, the only real choice for animation.
- **Hemi:** 180° constant light source. Great for use as a fill light, or as a back light.
- **Area:** These are similar to lamps, except that they are rectangular instead of an infinitesimal point. As a result they can cast accurate raytraced soft shadows. These are my lamp of choice for still shots, but they take a lot of time to render.

**Standard****lamp****Area****lamp**



**Spotlight**

**lamp**



**Sun lamp**



lamp

## Creating a basic scene with basic lighting

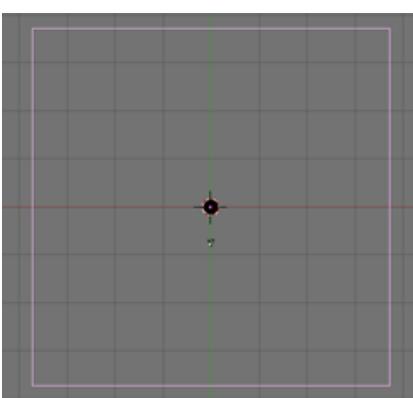
- This addition is simply a way to apply what you know about lights and to discover a few settings like colors or creating simple shadows. The purpose here is to create a basic scene with a sphere over a plane, nicely lighted. You should already know the basics of Blender (creating a mesh, moving and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

## Creating the scene

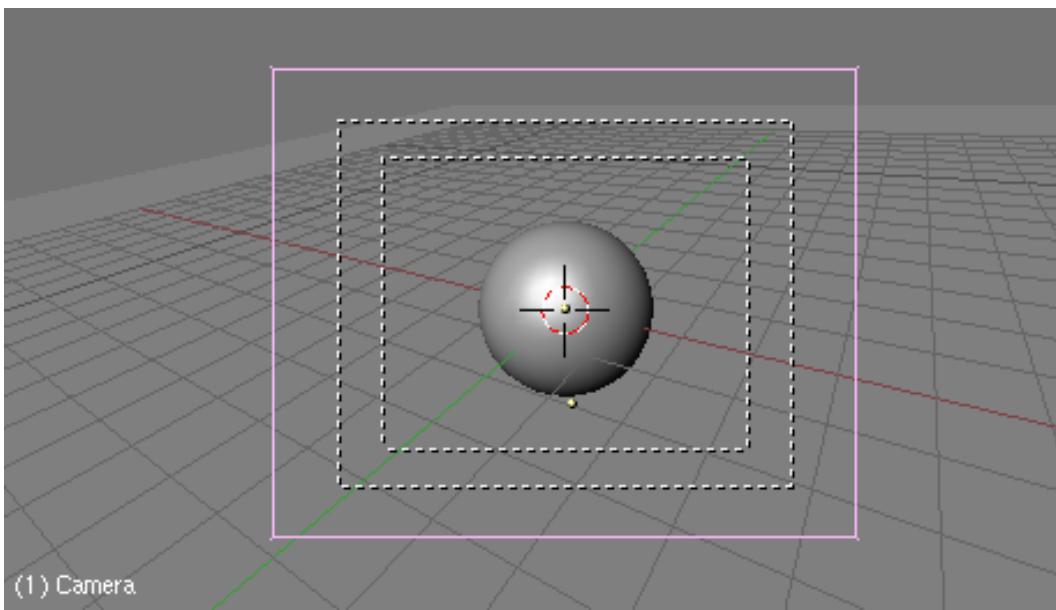
- Okay, let's start! Open a new file. Add a **UVsphere** of 32 rings and 32 segments. Exit **EditMode**. Leave it in the center of the scene.

**Tip :** Go in the Editing buttons  , and push **Set Smooth** so the sphere will render as a nicely smoothed sphere.

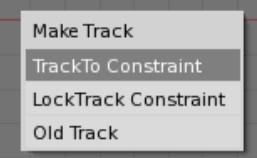
- You should already have a plane in the basic scene, otherwise add a **Plane**. Have it in **ObjectMode** and move it just under the sphere. Scale it so it is very big. The ideal would be that we can't see borders with the camera.



- 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**.

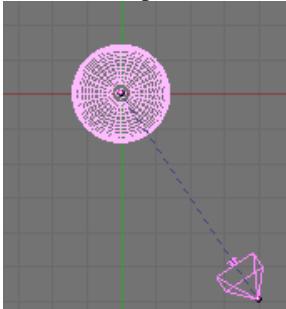


**Trick :** First click on the Camera, then on the Sphere holding **Shift**( the order is very important ). Do a **Ctrl+T** and



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



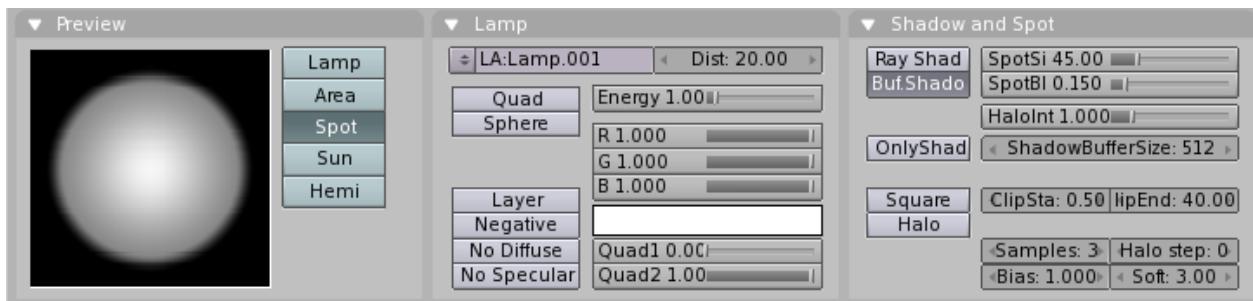
camera will still point at the sphere.

## Adding the lights

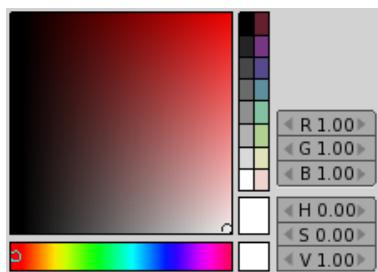
- Okay, what we have is a pretty beautiful scene, isn't it?... Well, it isn't! But it is enough to add some lights... Here I will describe a basic lighting scene I use as a default for fast renderings. I picked it out from another tutorial, you can find the link at the end of this page.
- Now we will 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:



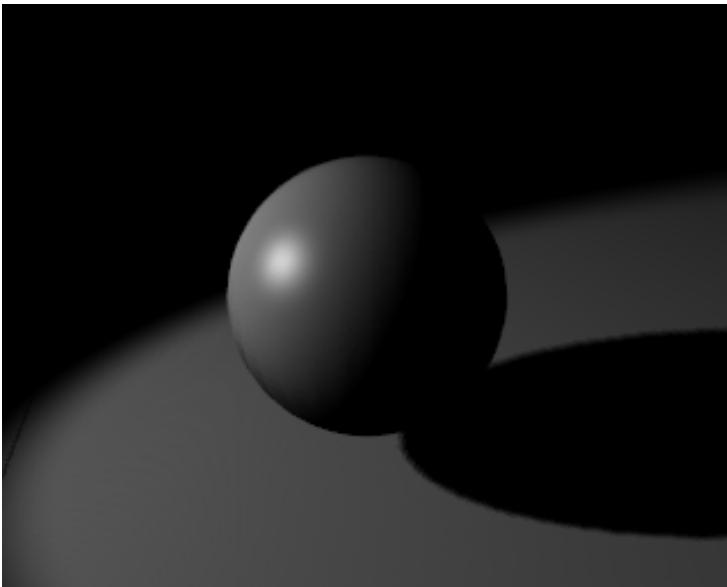
You will see these buttons:



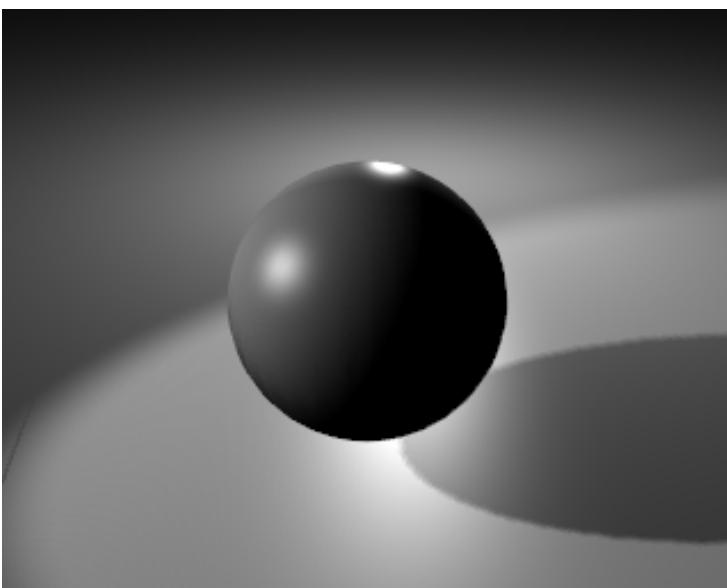
- Yeah, really lots of options. Don't worry, I'll explain the basic ones.
  - **Dist** : Sets the maximum distance the light can reach. Increase it so the lighting cone really goes behind the sphere. I set it to **Dist:40**.
  - **Energy** : This is the force of the light. You can leave it at **Energy:1**.
  - **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**.



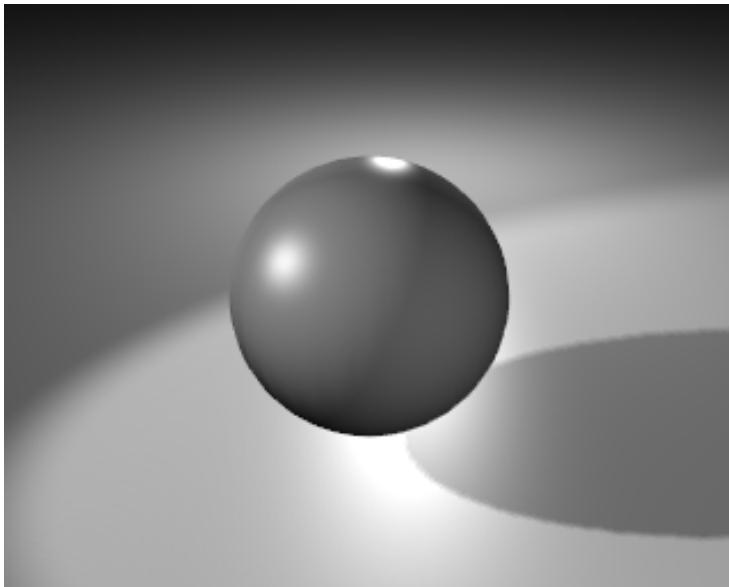
- **Buf.Shadow** : Enables the light to cast shadows, leave it **pushed**.
- **OnlyShadows** : This light only creates shadows, without casting light ( yeah quite unrealistic, but it can be useful ). Leave it **unpushed**.
- **SpotSi** : This is the angle of your cone, in degrees. Leave to **SpotSi:45**.
- **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. (this is very important... If you do this badly, it can result in completely dark scene!)
- 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.



Note that the following site contains nude figures: For a more in-depth tutorial, here is a tutorial from the Blender Documentation <sup>[1]</sup>, which has been a great source of help for me.

**User note:** you might want to check this tutorial on lighting: <http://www.youtube.com/watch?gl=US&v=YJiR2Q7uvbQ>

## Outdoor lighting

- Here you will use a **Sun** in conjunction with a **Spot** light and some little **Lamps**.

## Lighting Without Lamps

- It is possible to light a scene without lamps.
- Ambient Occlusion Video Tutorial: <http://www.youtube.com/watch?v=rKUAemD7oo4>
- Radiosity Lighting Video Tutorial: [http://www.youtube.com/watch?v=\\_M4ffj2w\\_WY](http://www.youtube.com/watch?v=_M4ffj2w_WY)

## References

[1] <http://download.blender.org/documentation/html/x4029.html>

# Blender 3D: Noob to Pro/Yafray Render Options

**YafRay** (Yet Another Free Raytracer) is an open-source alternative rendering engine supported by Blender.

**Reader note:** in the new version of Blender (2.46) internal renderer of Blender got a lot of upgrades and is now about two times faster than YafRay, so this article may be considered obsolete.

The YafRay render settings are normally hidden. To show them, switch from the Blender Internal rendering engine to the Yafray rendering engine.

- Go to the Scene Panel (Press F10)
- Click the Render Buttons button
- In the render tab, under the big Render button, click the Blender Internal button and select YafRay.

Now the YafRay and YafRay GI tabs appear.

## Getting Started with Yafray

First, I would like to state that Yafray renders 2-3 times faster and nicer (includes Global Illumination, Caustics and...) than the normal internal built-in renderer in BlenderWB:OR. So it would just be worth it to download and install it.

On Windows, just download the .zip file and unzip it. You get the .exe file. Run it and follow default instructions. Latest version is 0.1.0 as of October 2008. Now change the Environment variable by adding C:\Program Files\YafRay to your PATH or whichever folder you installed yafray.exe, users Windows 95/98/ME will need to restart where as WinXP users won't. Remember Yafray doesn't have a GUI and is a command line tool. Anyway you never need a GUI to run yafray as it runs nicely inside Blender as it is integrated into it. Go to the Command line and type yafray and press return. Now you have two paths

*Reader: Can someone make this a little bit clearer? What is the Command line?*

*Answer: The command line interface (CLI) is an interpreter for the operating system that provides a way of getting around by accepting commands from the keyboard instead of using the mouse to click on icons. It was the primary method of using the computer in MS-DOS, the predecessor to Windows. There are no buttons like in programs such as Blender. [1]*

You can access the command line in Windows by pressing Win-R then typing CMD and pressing the enter button. The command line can also be accessed by clicking on the Start menu, going down to "All Programs", over to the "Accessories" tab, and then clicking on the Command Prompt option. To change the environment variable, enter "path C:\Program Files\YafRay;%PATH%", without the quotes.

- 1) If there was an error regarding MSVCR71.dll, then download this file from the Web and place it in the c:\windows\system32 directory.

*Reader: I have been using Blender for a while now, but due to my school's administrative limitations I am only able to use it on certain computers in school. For example, the computers that have already had 3d programs installed on them (like the ones in the Drafting and Design room) allow me to use Blender. Other computers give me an error message concerning MSVCR71.dll when I try to use it (specifically the ones in my Study Hall). Could you give me an alternate method of installing the .dll (maybe I can put it on a flash drive or in the My Documents folder)? Answer: Unfortunately, without administrative rights you will not be able to install a DLL. Answer 2: My school has the same type of administrative blocks, however I placed everything from my blender folder at home onto a flash drive, and now it works perfectly on all computers. Maybe try that?*

- 2) Switch from the Blender Internal rendering engine to the Yafray rendering engine as explained and press F12 to Render. Wow, You should have noticed the lightning speed of Yafray.

## YafRay tab

- **XML button:** If pressed blender will export an xml file that describes the scene and will pass it to YafRay. If not pressed blender will render with the YafRay plugin. When using the plugin interface, the render in progress will update continuously in Blender's render window. This gives much faster feedback. It even shows the distribution of the samples taken by the path light when rendering with cached pathlight. YafRay divides the image into blocks and renders them in a random sequence.

## References

[1] [http://en.wikipedia.org/wiki/Command\\_line](http://en.wikipedia.org/wiki/Command_line)

# Blender 3D: Noob to Pro/Stamping

## TicToon Tutorial: Stamping

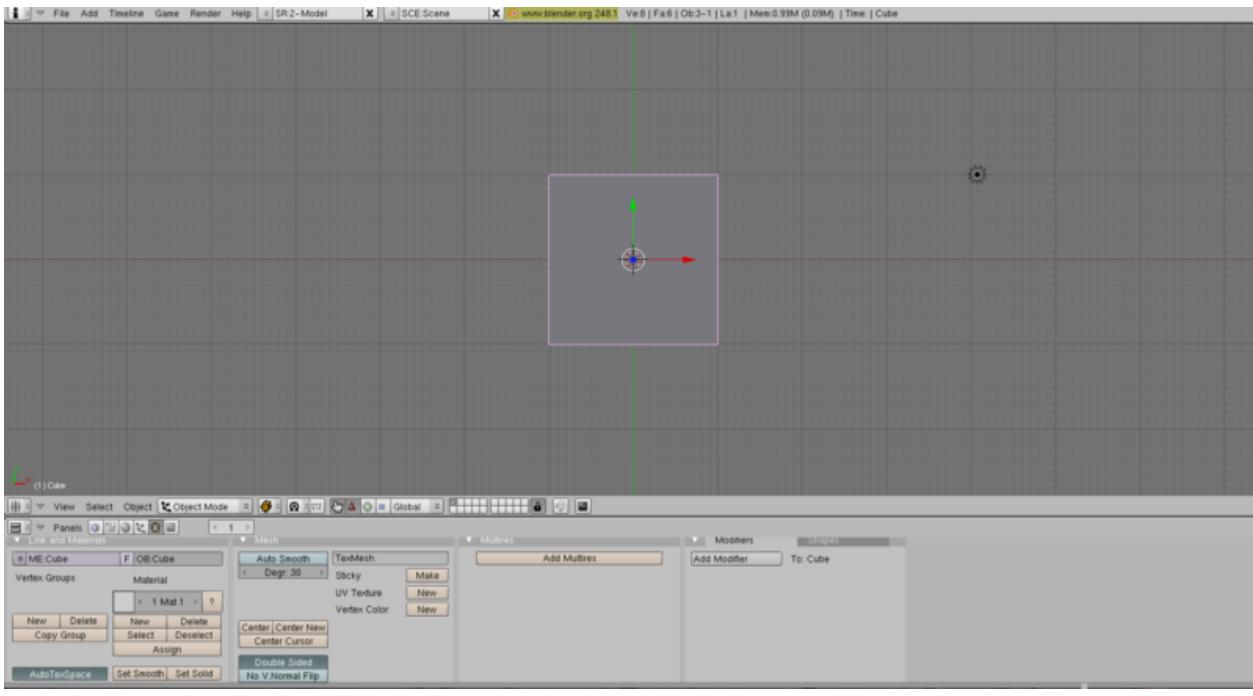
Stamping in blender is a rendering option. A stamp on a render can prevent that render from being lost, keeping your project organized, and keeping your files organized.

An example of a stamped Image:



Now my render is associated with a file, I know the date it was rendered at, and I've stated that the image is property of the OWF project.

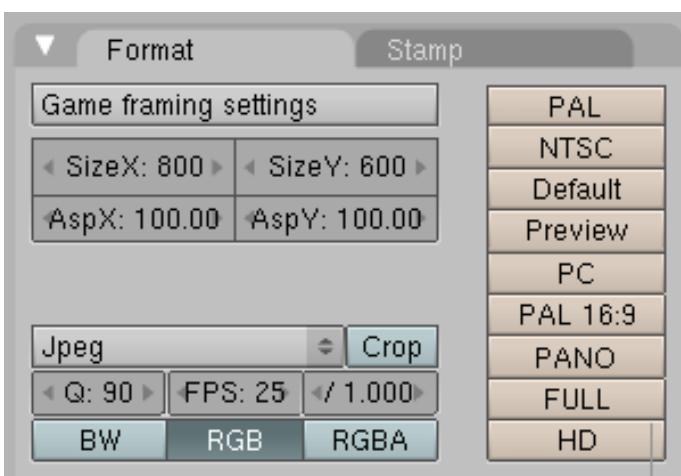
Let's start off by opening up Blender. We are presented with the default scene.



Go to "Scene". Hit F10 or click the Scene button.



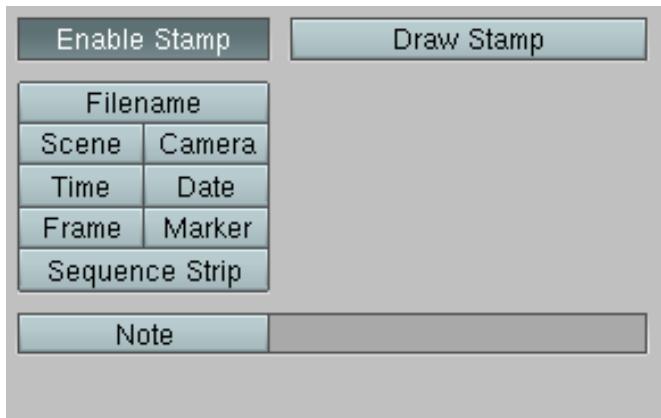
On the far right side you can spot a panel for the format of your renders.



Pick the "Stamp" tab.

You should be presented with a button: **Enable Stamp**

Click on the button and you should see a variety of options.



Enable Draw Stamp to see your stamp in your render, more on Draw Stamp later on.

Each setting does something different. Here's a breakdown:

**Filename:** If enabled, this displays the path to your file. If you haven't saved your file yet, it will display "*File <untitled>*" Filename is displayed in the top left hand corner. When more than one stamps are in the top left corner of your render.

**Scene:** If enabled, this will display the name of the Scene your render is on. It will be displayed in the bottom right hand corner of your render.

**Camera:** If enabled, it displays the name of the camera from which the scene is rendered. It is displayed on the bottom of your render in the middle.

**Time:** If enabled, this displays the time stamp of your animation. It is displayed in the bottom left hand corner of your render.

**Date:** If enabled, the date of the render is displayed. It appears in the top left corner of your render.

**Frame:** If enabled, the frame of your animation will be displayed in the bottom left hand corner.

**Marker:** If enabled, the marker will be stamped onto the render. It is in the bottom left hand corner.

**Sequence Strip:** If enabled, it will display the sequence strip. Displayed in the top right corner.

**Note:** This is a custom stamp. you can type anything you want in here and it will be stamped in the top left hand corner of your render.

**Draw Stamp:** This needs to be enabled for the stamps to be seen in your render. The drop down menu has various sizes for the text of your stamp. The Text Color is for the colour your text appears to be. The default is grey. "Background" is the background for the text. Default is black. "A" stands for Alpha. It indicates the alpha values for the background.

Now enable some buttons and hit F12 and you should see stamps on your render!

**Efficiency tip:** using this with Evernote can be very useful because Evernote can find text within images so you can quickly find a render if it has a stamp. I use the filename and date taken to find my images in Evernote. Keep blending :D

# Blender 3D: Noob to Pro/Basic Animation

---

## Note About Blender 3D: Noob to Pro/Basic Animation

I created this little note section to both warn learning users, and to give information to those who would make this wikibook better. The animation section in this book leaves a lot to be desired. It needs to go slower, and be more thorough. It is not finished (hence the accurately drawn "status images"). Here are a few things that need to be mentioned:

- What a key is
- How to manipulate keys after they are drawn
- How to delete keys (both IPO and Lattice)

I recommend that new users read the animation section in the blender documentation first. It is written for an older blender version, but it is more informative than this tutorial, and most of the information in there is still accurate. Here is a link: <http://www.blender.org/documentation/htmlII/p6390.html>

## Basic Animation Introduction

Animating, in principle, isn't that hard. Good animation is a very different story. 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 or by pressing **CTRL + Left Arrow**.
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 (in this case, you don't have any right now).
4. Place the object or bone at the desired starting location and/or rotation.
5. Press the **i-key** on your keyboard while the mouse is in the 3D view 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 (now called LocRotScale). 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 **i-key** 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 the 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 this YouTube Tutorial <sup>[1]</sup> or The Blender Userguide <sup>[2]</sup>

## References

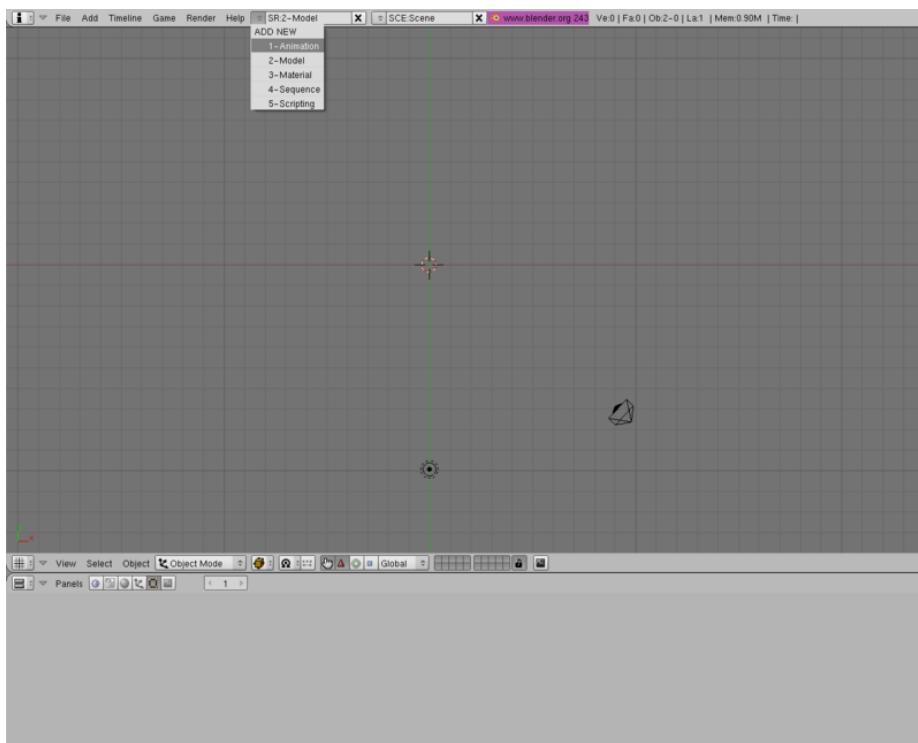
- [1] <http://www.youtube.com/watch?v=FAZFfLmbD2k>
- [2] <http://download.blender.org/documentation/html/>

# Blender 3D: Noob to Pro/Basic Animation/ ScreenLayout

## Intro

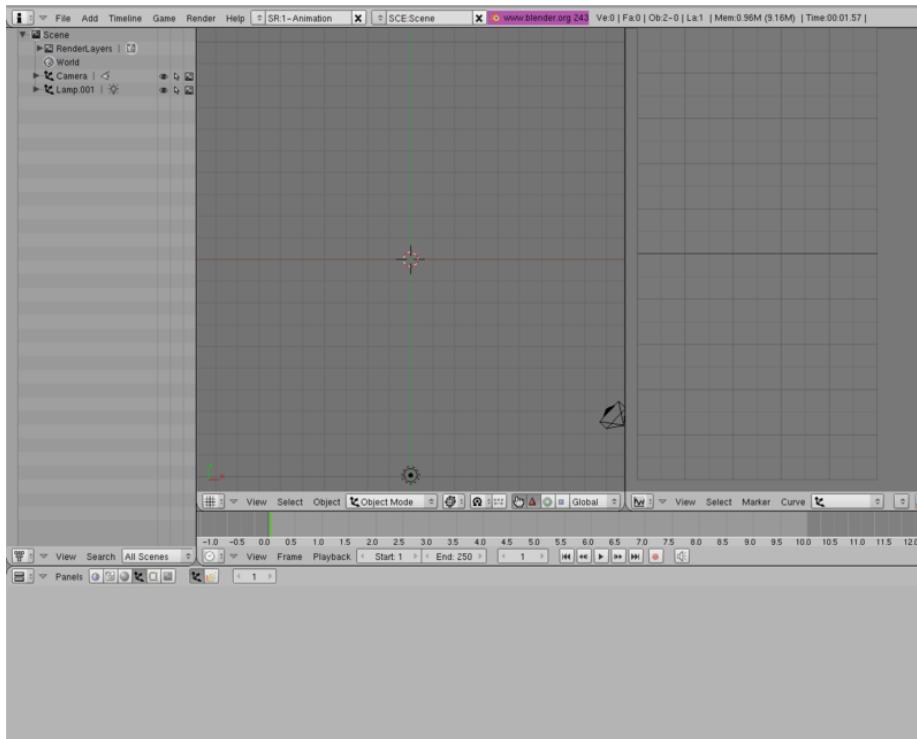
If you haven't done anything to the default blender startup, starting blender will produce the basic 3D window with a light, a camera, and cube, the buttons window, and the ever present basic drop down tabs at the top of the screen known as the user preferences window. This is the default setup for modeling, but now that we are starting animation, we are soon going to be looking at a new(ish) set up. You'll see what I mean.

## Setup



Now the first thing we need to do is enter 'animation mode.' To do so, look at the top of the window, the 'user preferences window.' You should see file, add, timeline, game, render and help. What we want is the little drop down menu next to 'help' that you've probably never touched much less noticed. It should read something like: 'SR:2-model' with little buttons on both sides of the text. Click the button on the left side that looks like two little arrows to open a drop down menu. Click on 1-animation, and DON'T PANIC. this is very hard but you must continue.

## About this setup



No blender didn't just go crazy, but as you can see, there are some changes that need to be explained. Blender has just opened a set of new windows each with their own functions and commands. You can probably recognize three of them. The 3D view is still there, but it's been resized smaller to fit the new stuff. Our old friend The buttons window is still there at the bottom, and of course the user preferences window is still at the top. But enough of what's the same - you're wondering what the heck all the new stuff is for.

There are now three new windows in all. From top left going clockwise, the first is the Scene window. This window is used for finding objects and such located throughout your 3D environment. By clicking on any one of the objects listed here, its counterpart in the 3D viewscreen will be selected. You can also select and tinker with your world settings and render layers from this window. The next window is the IPO (InterPOLation) curve window; it is the window that displays data concerning how your objects will actually move, and can begin to look very technical as we progress, don't worry it's normal. The final window is easier to miss. It is essentially a timeline, and is rather thin located between the 3D view/IPO windows and the buttons window. This is the Timeline window. For what we'll be doing for now, it will basically function as a kind of VCR. It is essentially the time codex associated with your animation. Clicking anywhere on the timeline will show you the animation as it is at that point, like fast-forwarding/rewinding then pausing. Like any good VCR the timeline window is equipped with the Play, Stop, Fast-Forward, and Rewind button set.

# Blender 3D: Noob to Pro/Basic Animation/ Lattice

---

## What is a Lattice?

A Lattice is essentially a simple container that can be used to deform and manipulate a more complex mesh in a non-destructive manner (ie. A lattice can be used to seriously deform a mesh then, if the lattice is later removed, the mesh can automatically return to its original shape).

## How to add and use a Lattice

A Lattice is added to the scene in the same way other objects are added. Either:

1. Shift-A over the 3D window and choose **Lattice** from the pop-up menu, or
2. Press **Spacebar** over the 3D window and choose **Lattice** from the pop-up menu, or
3. LMB ADD from the 3D window menu and choose **Lattice** from the drop-down menu

The default Lattice looks just like a cube when first added except that it is just one Blender Unit (BU) wide whereas a mesh cube is 2 BU wide. When the Lattice is added, the window remains in **Object Mode** and the Lattice can be moved, resized and rotated like any other Blender Object.

On its own, a Lattice serves no purpose whatsoever since it can't be seen in a rendered image. Its only use is to manipulate another object and so, to be useful, we need to associate another object with it. The current method for doing this is by applying a "**Lattice Modifier**" to a Blender object.

The basic workflow is:

- Add a mesh (cube, cylinder, cone, sphere, etc)
- Add a Lattice
- Select the mesh
- Press F9 and go to Modifiers Panel(buttons window). In the modifiers tab Press **Add Modifier > Lattice**
- In the Ob: Panel, enter the name of the Lattice (The default name is Lattice, Lattice.001, Lattice.002, etc - or you can give it a useful name that you'll remember later)

**NOTE:** *In older versions of Blender, a lattice was applied using a parent method - select the mesh, then select the Lattice, then press Ctrl-P and choose "Lattice Deform". While this method is still available, it does not offer the full functionality of the newer Lattice Modifier. If you use the parent method, you will need to press the "Make Real" button in the modifiers palette to enable the modifier functionalities. Using the method, the Lattice becomes a parent and a modifier. As a parent, it will act on the mesh in Object Mode whereas a true modifier lattice only influences the mesh when altered in Edit Mode.*



Adding a Lattice Modifier to an object

Now, we can change the Lattice in **Edit Mode** and any changes we make to it will affect the mesh.

### Note: Applying the Modifier Correctly

When adding the modifier to your object, you must enter the exact same name as your lattice. The default is **Lattice** with a capital "L" but if you've changed it or have more than one Lattice, then you will need to enter the new name. When you enter a valid Lattice name, the name stays visible in the Ob: box. If your entry disappears then you've entered the name incorrectly.

One simple way to get the right name is to select the Lattice, go to F9, Link and Materials panel and where it says Ob:Lattice or Ob:Lattice.001 etc, move the mouse over this field and press **Ctrl-C** (*don't click on it, just hover over it*). This copies the Lattice name. Then select your object, go to the Lattice Modifier panel, hover the mouse over the Ob: field and press **Ctrl-V** to paste the name in. Now it should stay there and your Lattice should work in Edit Mode.

### Basic Exercise:

1. Start with a new Blender scene, delete the default cube and add a UVsphere (Spacebar>Add>Mesh>UVsphere). Accept the default 32 Segments and Rings. You could use any of the mesh shapes but because the sphere is heavily subdivided (*made up of lots of edges and faces*); you will get a better idea of what the Lattice is doing.
2. Tab back into Object Mode then add a Lattice. It is generally wise to resize the Lattice so that it surrounds the mesh it will be deforming. So, press SKEY and enlarge the lattice to 2 BU wide.
3. Select (RMB) the Sphere and add a Lattice Modifier to it. Make sure that you type the exact Lattice name correctly into the box. Nothing appears to happen but that's fine.
4. Now select your Lattice, TAB into Edit Mode, select just one control point (a control point looks like a vertex) on the Lattice and move it around. You will see the sphere mesh stretching and squashing relative to that control point. Move each Lattice control point one at a time and see just how far you can deform the mesh. Zoom in your 3D window if you need a closer look.
5. The control points of the Lattice can be moved, scaled and rotated in the usual ways. Try selecting a few of all the control points and scaling them (SKEY) or Rotating them (RKEY) and watch the mesh follow along.
6. Now, exit Edit Mode (TAB) and select the sphere in Object Mode. Go to the modifiers panel and press the big "X" button to remove the modifier. Despite all that deforming, the sphere immediately returns to its original size and shape. Nothing was really changed in the sphere's mesh data. Immediately re-apply the modifier as before and see the Lattice immediately apply its own deformation to the sphere again.

### Getting more involved with Lattice

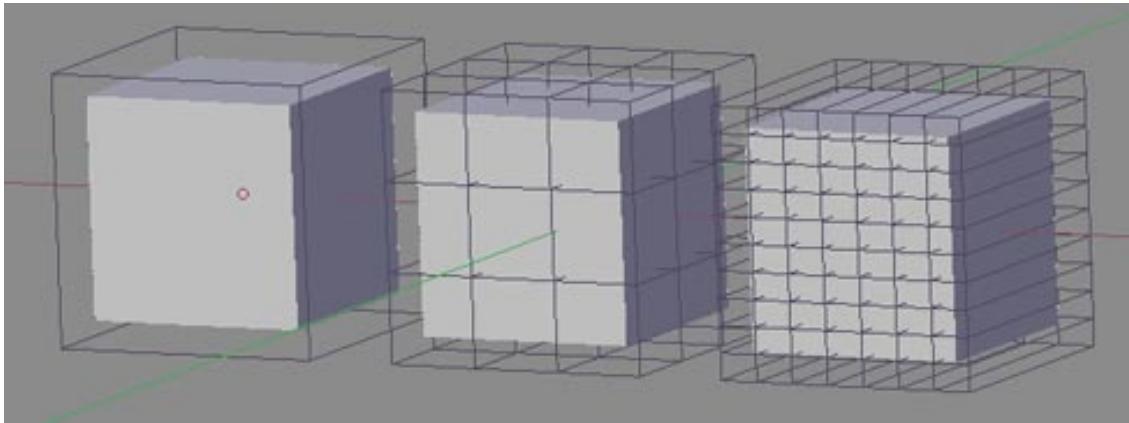
The default Lattice is two control points high, two wide and two deep, ie, it is two control points wide in each direction (these are referred to as the U, V and W directions). However, we can change the number of control points in one, two or all directions. This is done by selecting the Lattice, going to the Lattice Panel (F9) and changing the values in the U, V & W buttons. If you decrease one figure to a value of 1, the Lattice will become two-dimensional (planar). Decreasing two values to 1 will change the Lattice to a line (one dimension). This can be useful, especially if the remaining value is increased, but is not the most common usage of Lattice.



UVW settings for Lattice

**NOTE:** The "Make Regular" button will set the UVW control points of an unscaled Lattice to be exactly one Blender Unit apart in each direction. The "Outside" button effectively removes all the internal control points which are added when the UVW settings are higher than 2.

In most situations you will want to increase some or all the values as this gives you more control over complex deformations, much like a subdivided mesh. How much you increase the UVW directions depends entirely on how much detailed control you need over deformations. As you've already seen basic squash, stretch, shear and simple deforms can easily be achieved with the default 2,2,2 Lattice but by increasing the UVW values a little, a whole new range of deform possibilities becomes available.



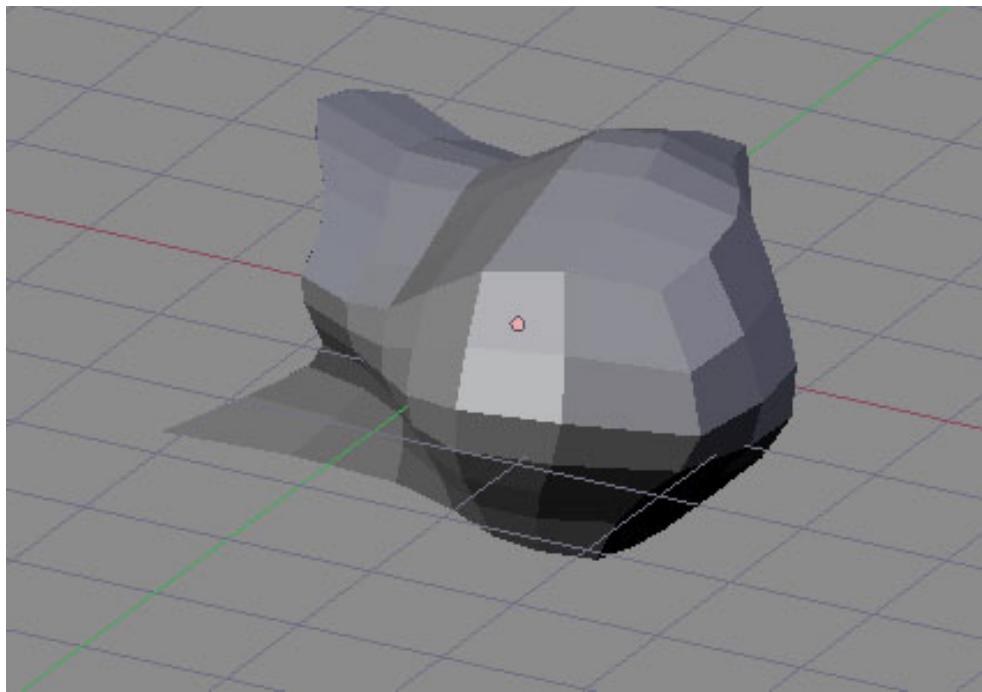
*Three Lattices with different UVW settings*

One thing to understand, however, is that a Lattice cannot bend the individual edges of a mesh (the lines connecting any two vertices) so the mesh must contain enough edges in order to apply complex lattice deformations to it. These edges must be genuine edges and not the virtual edges created by a subsurf modifier. Edges can be added to a mesh using a variety of tools including subdivide, knife and loop cuts.

N00b note here...um, I'm doing the tutorial cover to cover style, and haven't ever heard of 'knife'...or loop cuts, please elaborate.

New Note: Use the K key to open a menu for the knife, loop cuts, etc. Knives are used to add a vertex to each line you left click and drag the cursor over, and loop cuts allow you to make multiple cuts into your active object.

About loop cuts: Ctrl + R also is a direct shortcut to a loop cut. Furthermore, loop cuts' main uses are for subsurf modeling. In order to accomplish a hard edge in a subsurfed object (for example the edges and corners of a wooden table or the bumpers of a car) you must apply loop cuts to the faces surrounding the edge and place them very close to them (and parallel). Experiment with a simple subsurfed cube and you'll see. Keep in mind this is why when modeling it is highly recommended to use quads. A loop cut will only cut a quad face, or every face connected with it in one direction, but only quads. Another main reason why it is recommended to model with quad faces is regarding animation but I suppose other tutorials will deal with that later. More information: [1]



A simple subdivided cube after a complex lattice deformation

## Intermediate Exercise

For this exercise, start over with a new scene and repeat Steps 1 - 3 in the Basic Exercise above. Don't go into Edit Mode yet.

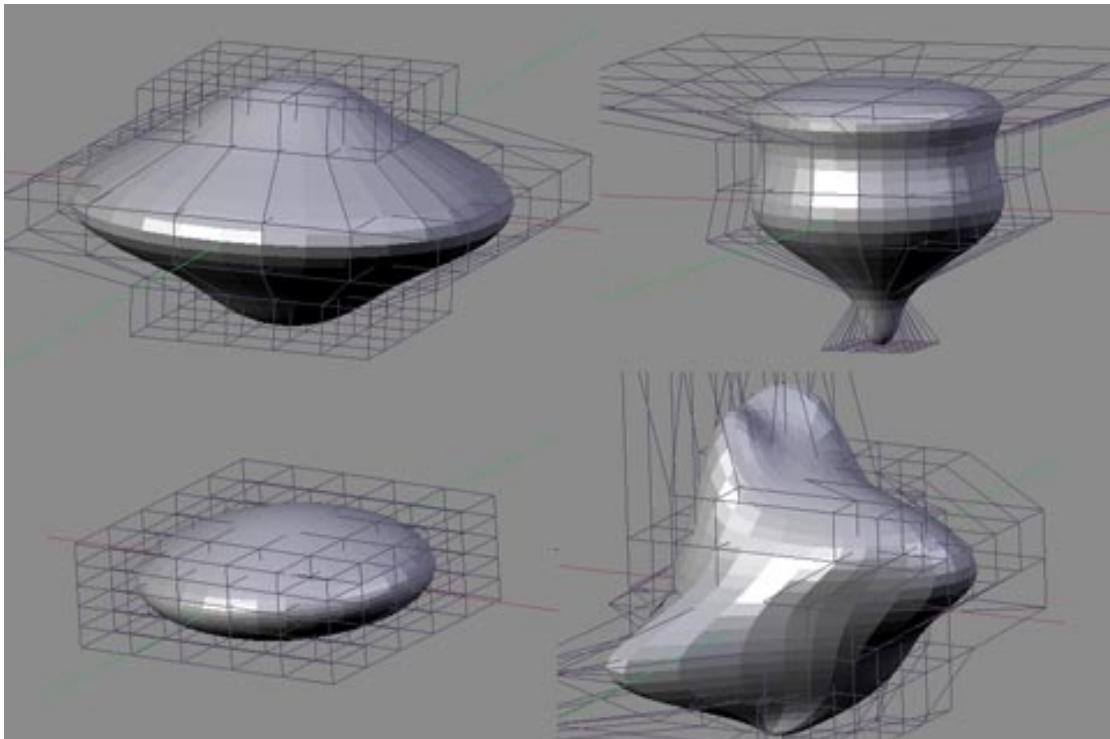
### Basic Exercise:

1. Start with a new Blender scene, delete the default cube and add a UVsphere (Spacebar>Add>Mesh>UVsphere). Accept the default 32 Segments and Rings. You could use any of the mesh shapes but because the sphere is heavily subdivided (made up of lots of edges and faces); you will get a better idea of what the Lattice is doing.
2. Tab back into Object Mode then add a Lattice. It is generally wise to resize the Lattice so that it surrounds the mesh it will be deforming. So, press SKEY and enlarge the lattice to 2 BU wide.
3. Select (RMB) the Sphere and add a Lattice Modifier to it. Make sure that you type the exact Lattice name correctly into the box. Nothing appears to happen but that's fine.
  - In Object Mode, select the Lattice then press F9 and go to the Lattice Panel. Increase all the UVW values to 4. You will immediately see the change displayed in the 3D window.
  - Tab into Edit Mode and play with the control points again. The first thing you might notice is that the corner control points have less influence now than before. This is because the effect is proportional to the distance of each point from the mesh. Move some points in the middle of a side face and you'll see the effect is significantly greater.
  - Try selecting groups of control points and pulling them away from the Lattice to stretch lumps and bumps out from the mesh. If you are familiar with the proportional edit tool, try that too - it works on Lattices just like it does on meshes.

Try doing the same exercise but increasing values in only the W direction. (*U divides in the X direction, V divides the Y direction and W divides the Z direction*). So, set the UVW to 2,2,4 and play with the control points again in Edit Mode. Select the two rows of control points around the centre of the Lattice and scale them up (SKEY) then Scale them down along Z-axis only (Skey>ZKey). Try other transforms constrained on different axes for interesting, controlled results.

There are three different options for how the Lattice Modifier affects each UVW direction. These are Linear, Cardinal and B-Spline. All I can say is that B-Spline is the default and to find out what the difference is, press the buttons and see.

## Examples



*Some examples of a Lattice deforming a standard UV Sphere mesh (not smoothed)*

## Making it stick

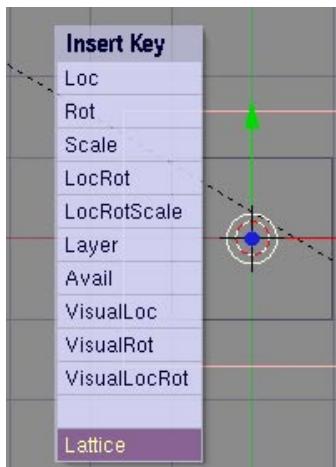
To keep the mesh deformed permanently, you can select it and press the "APPLY" button in the Lattice Modifier panel. This "bakes" the mesh in the deformed position and disconnects it from the Lattice. The Lattice can then be deleted without the mesh returning to its original shape. This is useful if you are using the Lattice as a modelling tool rather than an animating tool.

Also, instead of deleting the Lattice, you can use it to immediately modify another mesh. Simply move the Lattice over the new mesh in Object Mode then apply the Lattice Modifier to the new mesh. If the Lattice is already in a deformed state, the mesh will immediately be deformed too. Press Apply again to "bake" that mesh and keep re-using the already deformed Lattice on as many other meshes as you wish for matching results.

## Animating a Lattice Modifier

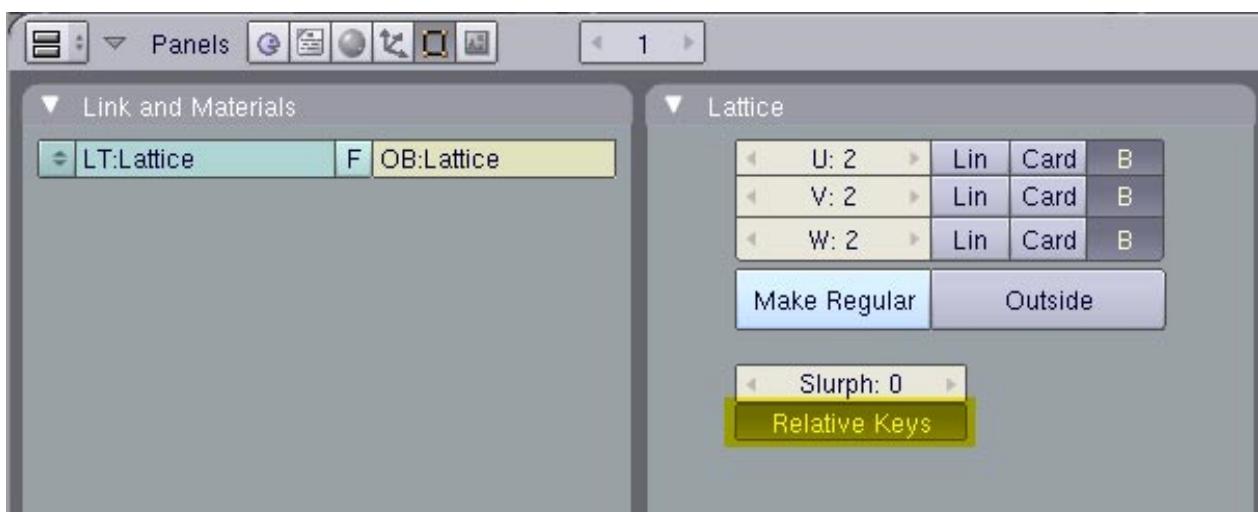
Lattice animation uses a workflow almost identical to the RVK (Relative Vertex Key) workflow from Blender 2.37 and earlier. *You don't need to know this but some people might find the information useful.*

With your Lattice Modifer already added to your object, select the Lattice and press **I-Key** to tell Blender you wish to animate the Lattice. Choose "**Lattice**" from the pop-up list. This sets the basis key (undeformed state) for the Lattice.



*Adding the basis key*

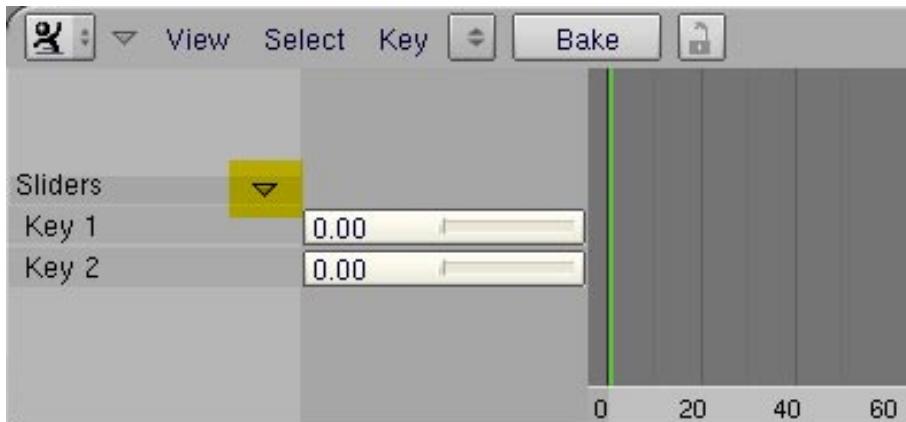
In the Editing panel (F9), press the "**Relative**" button.



*Set the keys to "Relative"*

**NOTE:** The Slurph setting determines the delay, in frames, for how long it will take to morph from their former state to the one applied by the new lattice.

To set your first deform key, press **I-Key** again then enter **Edit Mode**. Deform the Lattice by scaling or moving control points then Tab back into **Object Mode**. If you open an **Action Window** now, you will see your first key, "Key 1", added to the list. If you press the small arrow at the top of the list, it will display the **slider** for that key.



*Lattice key sliders in the Action Window [Noob note for blender 2.48a] In the action window, select "ShapeKey editor" in the dropdown "editing modes for this editor" (in the panel header)*

To add more Lattice keys, for a variety of deform shapes, repeat the process: I-Key, Tab to Edit Mode, move control points, Tab back to Object Mode. Each time you exit edit mode, a new key will be added to the list in the Action Window.

Animating the Lattice uses the same process as animating Shape Keys for meshes.

If you want the object to begin undeformed, then set each key slider to zero on the first frame of the animation. Move through the frames, setting the sliders as you go to deform the Lattice as desired. You will often find you'll need to set a key before and after each deform key in order to control the rate at which deformations take place in the animation.



#### *Setting keys in an animation*

Noob note: I'm doing everything as described and get no keys in action window when exiting Edit mode... Any suggestions?

Noob note; Is it possible to create a keyframe using the object's mesh rather than applying a lattice?

Noobie note: What to press after I-Key? I'm not getting any keys in the action window, help please.

Another Noob: No keys here. I got something changing to IPO Curve Editor and choosing Shape in IPO Type

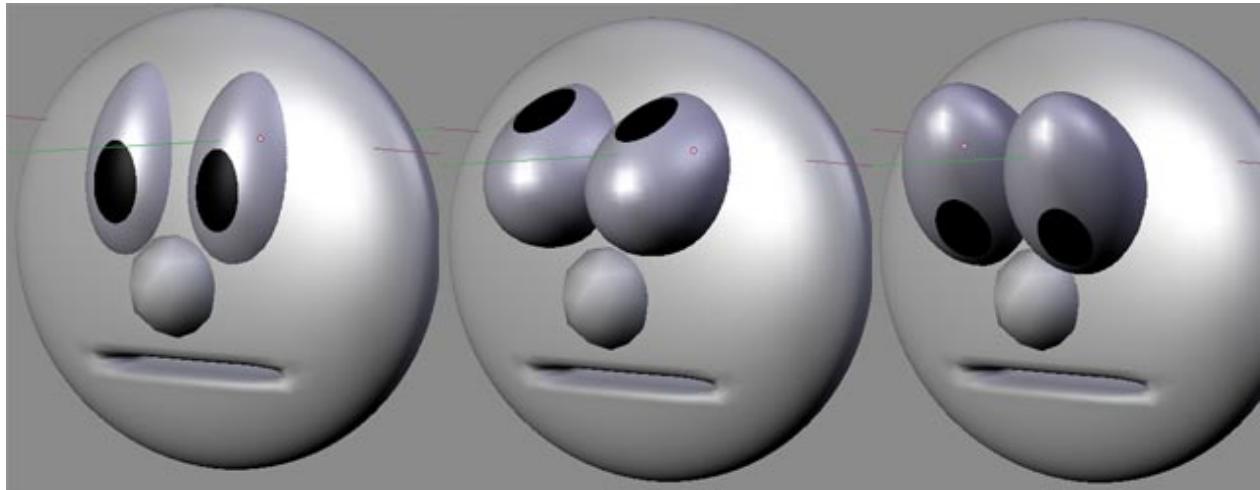
Noob note; Select Action Editor header->Shape Key Editor. Make sure you also selected Buttons Window->Editing (F9)->Relative Keys

## Let the fun begin!

Now you have your object safely locked away inside your Lattice, you can still animate the object itself, inside the Lattice!

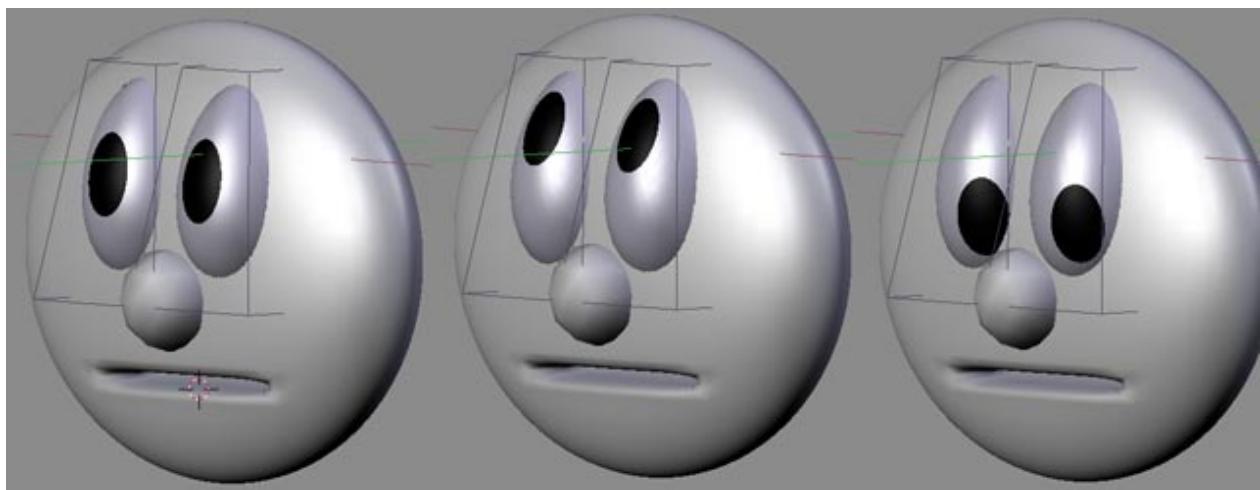
### Why would I animate the object too?

Take the case of the classic cartoon eyeball which is taller than it is wide. If you just stretched the sphere object itself, instead of using a Lattice to deform it, then you couldn't properly rotate the eyeballs to look up and down because the whole "egg-shaped" eyeball would rotate and end up lying on its front or back. The eyes would literally pop-out of the head.



*Cartoon eyes: The stretched spheres don't rotate properly.*

If, however, you use a Lattice to deform the basic sphere to make it into a tall "egg" shape, you can still select the eyeball itself and animate it within the Lattice. Now when you rotate the eyeball, it will look up and down but still maintain its deformed shape within the head.



*Cartoon eyes: Spheres deformed with Lattice can still be rotated with good results.*

## References

[1] [http://wiki.blender.org/index.php/Manual/Edge\\_and\\_Face\\_Tools](http://wiki.blender.org/index.php/Manual/Edge_and_Face_Tools)

# Blender 3D: Noob to Pro/Basic Animation/ Bounce

---

## How to make a ball bounce convincingly!

*This tutorial assumes basic Blender awareness and some knowledge of using Blender Lattices as well as keyframes basics regarding the Blender Panels. Knowledge of IPO curves will prove useful later as we progress to more advanced techniques.*

## How hard can it be to move a ball?

It would be easy to just put a sphere on the screen and animate it to move up and down but in all honesty, it would not look like a bouncing ball. It would not be convincing in any way.

To be believable, the ball must use some of the most fundamental principles of good animation. In particular, the ball must **squash** and **stretch** and change **speed** as it falls, hits the floor, bounces and rises ready to fall again. With a little effort we can make that boring sphere look alive!

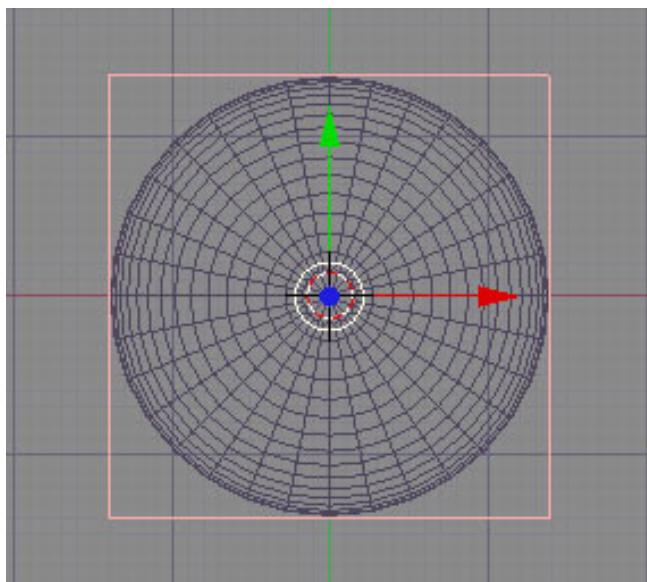
## Why use a lattice?

It is possible to make a simple bouncing ball animation without using a Lattice object but with the Lattice we can do more than just bounce the ball. For example, once we have a ball that bounces how we'd like it, we can later add rotation to the ball so it spins through the air and bounces then, as the bounces decrease, the ball can roll to a halt. Doing this without lattice would be a far more complex exercise as a rotating squashed and stretched sphere would look like it was wobbling in space (*like having the flat part of a flat tire spin around the tire instead of staying on the bottom, even while the wheel is turning*).

## Let's begin then

To begin, add a UVSphere in **Top View** (NUM7). It is important to add the objects in top view to keep their **Z-Axis** aligned upwards. Set Smooth and Sub-surf on the sphere if desired. Tab back into **Object Mode** then immediately add a Lattice. Resize (SKEY) it so it closely surrounds the sphere. Make the edges of the lattice align with the circumference of the sphere as this will make animation easier.

*(By default the Lattice Object Name is "Lattice". It is good to give all your objects unique and sensible names, especially if you are planning a complex scene. So you might call the Sphere "Red\_Ball" and the Lattice "Red\_Ball\_Latt" or something similar for easy reference and recognition later.)*



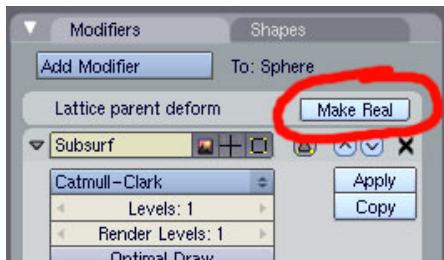
*Top view of UV-Sphere inside selected Lattice*

### Make the lattice deform the sphere

Prior to Blender 2.4, objects were parented to Lattices with a Lattice Deform option. In version 2.4+ you can select the object (the sphere) and apply a Lattice Modifier to it (Ob: Lattice). However, if you use this new method then the ball will not move where the Lattice moves and will not squash when the Lattice squashes and we want it to do both of these things.

Luckily you can still use the good old **Parent option** then use the Modifier palette to make the Lattice Deform "Real". So select the sphere and THEN SHIFT SELECT the Lattice then **CTRL-P** to make parent and choose the "**Lattice Deform**" option. If you now select just the sphere, press **F9** and look in the Modifiers palette, you'll see a listing for "**Lattice parent deform**" with a button for "**Make Real**". You can either ignore this button or press it. It really won't matter for now. If we wanted to do some weird things to our ball, like kicking it or deforming it in other complicated ways, then we would have to use the "real" Lattice modifier option.

So, this may not be the purist approach to using a Lattice but it is a convenient way to achieve the results we're after. Basically, it works and is reasonably intuitive.



*Lattice Modifier screen using parenting option.*



*If you press "Make Real" you enable the full Lattice Modifier*

You can easily check if the Lattice is working the way we want it to by selecting the Lattice and resizing, moving or rotating it. **The ball should do everything the Lattice does.** I don't wish this to be a full lesson in Lattices so if you

don't understand any of the above, or you're certain your Lattice is not working as it should, then please learn a bit about lattices before proceeding.

## Time to animate!

For our basic animation, we'll use just 23 frames starting with the ball high then falling, squashing, bouncing and ending up back where it started. When this is played back in the 3D window, it will loop and we'll see the ball bounce forever in one place.

This part is easier done in "SCR:1 - Animation" Viewport configuration. It is also useful to have multiple orthographic views open in the viewport by right clicking on the resize viewport arrow and choosing "split area" then making a top view, side, etc...

Go to **Front View** (Num1) Press F10 and set your start (Sta) frame to Frame 1 and your end frame to Frame 23. Then make sure you're on **Frame 1** (Shift+Left Arrow) to start animating. Note that we are **animating the Lattice**, not the ball. Everything we do to the lattice will directly affect the ball too.

Note: all the modifications to the lattice must be done in 'Object' mode. If not, they will affect all the frames.

Select the Lattice (**NOT** the ball) then do the following on the corresponding frames:

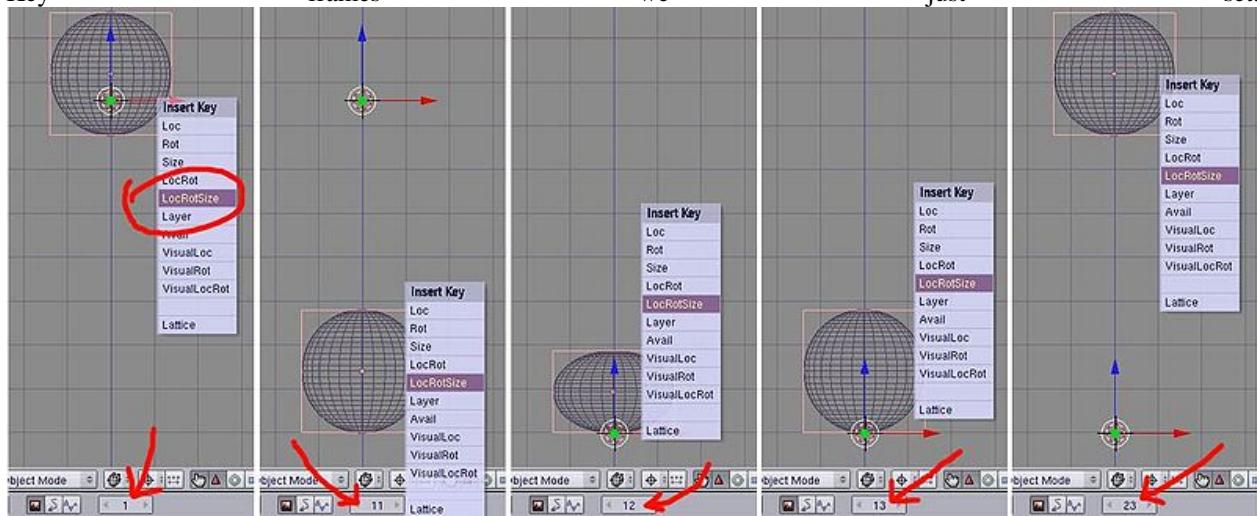
- **FRAME 1:** Press IKEY and set a **key** for LocRotScale - the ball's start position (high).
- **FRAME 11:** Move the Lattice to ground level and **key** LocRotScale again (*keep the fall distance small for now*).
- **FRAME 13:** Leave Lattice in same position as frame 11 and **key** LocRotScale again.
- **FRAME 23:** Move Lattice back to start position (use Num (N) Panel if you need to) and **key** LocRotScale again.

(Noob note- This may seem obvious, but move the lattice BEFORE you insert the keyframe.)

These four frames give a basic up and down animation with a pause at the bottom. We can add squash on Frame 12 to finish off.

- **FRAME 12:** Place the cursor at the base of the Lattice then set the Rotation/Scaling Pivot to "3D Cursor"  Scale the Lattice down along the Z-Axis **SKEY-ZKEY** and see the ball squash. **IKey** LocRotScale again.

When you have set all five keys, you can cycle through the frames and you should see the following result on the Key



*Screenshots of the 3D window, front view, on each of our five key frames.*

If you press **Alt-A** now in the **3D window**, you should see the ball bounce. Even though the squash happens on just one frame, it still reads properly in our minds and is a vast improvement over an animation with no squash at all. (*Press RMB or ESC to stop the animation playback*).

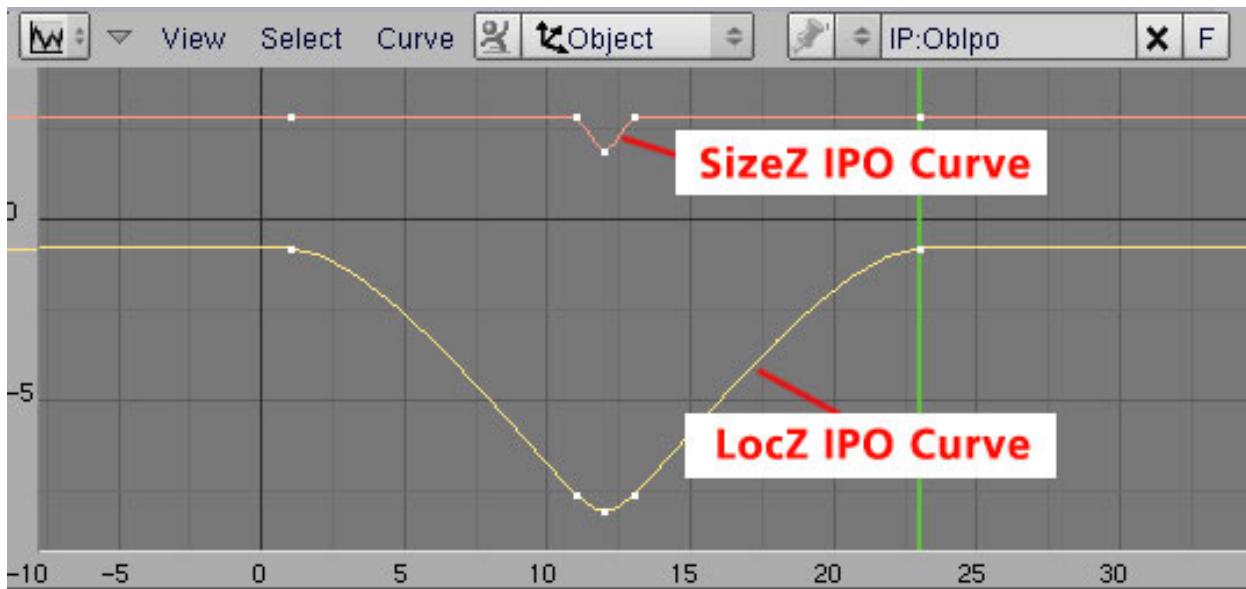
If you have a camera and lighting suitably set up you could render the animation out and use Blender's "PLAY" button to loop the rendered animation. See the bottom of the page for an OpenGL render made from the steps above. Add material to the ball and a ground plane for a more interesting result.

## Looking at the IPO Curves

If you want the ball to bounce three times then you could repeat the above instructions three times over - if you have the patience - but Blender has a better, easier way.

Select the Lattice then open up an IPO (Interpolation) window and zoom into the group of keys. If you select LocZ and SizeZ from the channel list on the right, you should see something like this (I've added labels to the screenshot):

*Note the the IPO Type here is "Object". Purists may be disturbed or perplexed by this but if you remember, we're not really using our Lattice in the purist, Lattice Modifier form. Don't worry about it though. It works.*

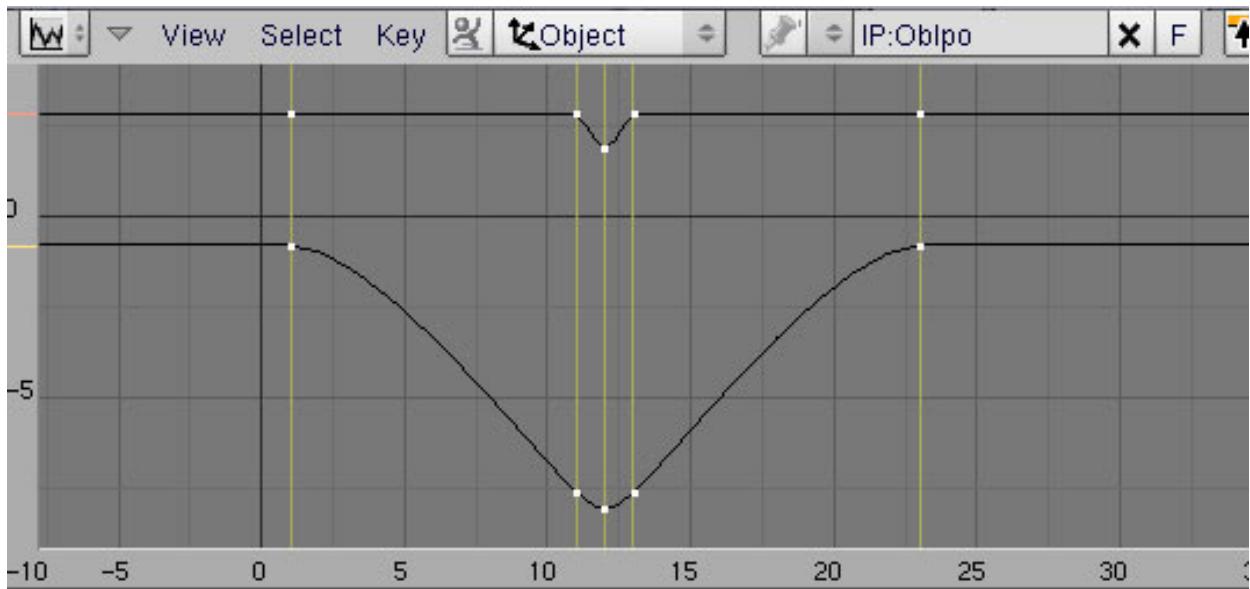


IPO curve for Lattice, showing LocZ and SizeZ channels.

This shows the relative position on the Z-Axis and size in the Z-Axis of the Lattice. You can see that the Size remains constant until frame 11, then reduces, then goes back and remains constant until frame 23. Similarly, the LocZ curve shows how the ball starts high, falls then stops before rising again. The rounded bottom of the LocZ curve occurs because the IPO curve maps the centre point of the Lattice and when we squash it, the centre point lowers slightly.

Of special interest is that by default, Blender uses Bezier curves for IPOs and this gives us rounded tops on the curve. This means the ball starts moving slowly downward then moves faster until it reaches the bottom. Also, as it nears the top, it slows down again. This is good because this is exactly what happens in reality when a ball rises and falls (*It could be better - but it's a good start*).

While in the IPO window, select both curves then press **KKEY** and you'll see each key frame indicated by a vertical line. (Note: You can also press *View->Show Keys* if you can't remember this shortcut - this has helped me in the past.)



*IPO window set to show key frames*

Press **AKEY** to select all key frames then duplicate (Shift-D) them. The curves will seem to deform but when you drag the frames to the right, they will sort themselves out. Position them so the "first" frame is at Frame 24 and the new end frame is now Frame 46. Reset the End Frame in the Scene (F10) window then watch the animation again. It's not bad. You can repeat this process for more bounces. (To have it bounce forever throughout a long animation, ignore the above procedure and just set the "Extend Mode" to "Cyclic" in the Curve menu. You'll have to adjust the curve a little on the first and last frame but you only do it once and Blender repeats the change for all other cycles.)

It's not bad for a start but there's a lot more to do to this ball before applying for that animation job.

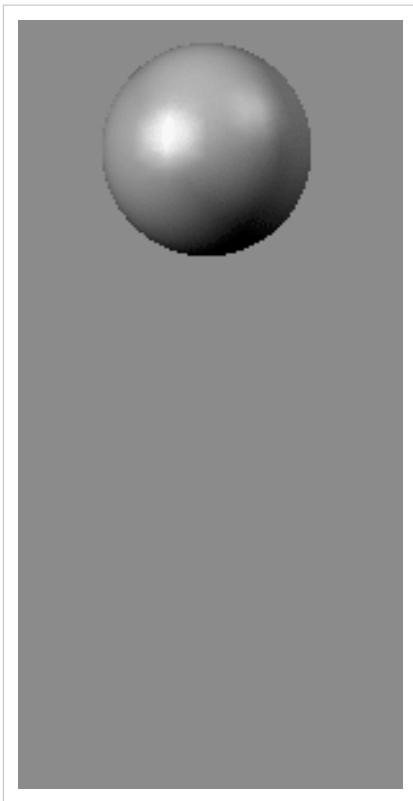
## So what next?

In real life and in a typical cartoon, balls don't just bounce forever in one spot. They move along and slowly lose their bounce before rolling to a halt. And they have more character than our bouncing ball.

### Things to consider next:

- Perfecting the squash - maintaining volume
- Adding stretch
- Look at acceleration and deceleration
- Rotate the ball as it bounces
- Reduce the bounce height over time
- Move the ball around as it bounces

See this ANIMATION LINK <sup>[1]</sup> (580kb, AVI-DivX) for an idea of how you can achieve interesting results with just a little more effort. but its k? k



## Links

Principles of bouncing ball animation: <http://www.idleworm.com/how/anm/01b/bball.shtml>

Blender 2.49B bouncing ball animation: [http://www.youtube.com/watch?v=BEY\\_6dFdUJY](http://www.youtube.com/watch?v=BEY_6dFdUJY)

Blender 2.49B vector blur animation, to add realism to your animations: <http://www.youtube.com/watch?v=qY4WcNgEXv8>

## References

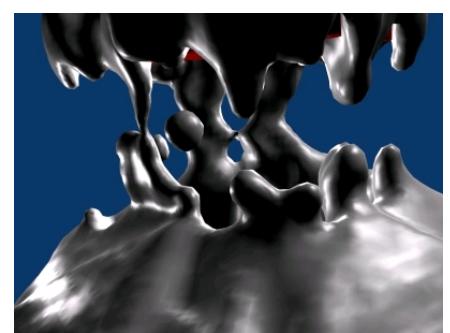
[1] <http://members.westnet.com.au/andydolphin/blender/bounce.avi>

# Blender 3D: Noob to Pro/Creating Basic Water animation

**Important:** Always make the domain after you have already made the fluid and everything else. Subsurf the object that you are currently working on before you create any other object except the domain.

## Water and Other Fluids

Water is without a doubt one of the most important compounds in our lives (second to air), and it covers about 70% of the earth and is therefore incredibly important in quite a few blender animations. Wouldn't it be great if we could get an accurate physical representation of this liquid in blender? We can, using a tool called Fluid Simulation. This tool looks unnervingly complex at first glance, but this tutorial should clear it up for you. At least to a basic level. (*Sorry, but I'm not the best with this feature. If anybody sees any missing information that you can help with, please don't be shy to contribute!*)



A low resolution, high viscosity fluid test

## The Domain

One can imagine how much time it would take Blender3D to think about everything in the infinite 3d world in terms of fluid objects and deflection, so we obviously need to cut down on that size. Create a fairly large cube. This will eventually be set as the volume in which all of the fluid simulation occurs. Don't make it too large, but not too small either. With this cube selected, press "F7" to get to the physics screen. Find the tab that says "Fluid Simulation", and click "Enable". Of the six new buttons that appear, click "Domain". All the fluid physics will be calculated inside this cube. Also in that tab, you'll see three buttons, "Std" (which will be pressed by default), "Ad", and "Bn". Each one opens a different set of values. These will be explained as they become important.

## The Fluid

As a basic start up for your experiments with fluid simulation, I am going to take you through a small demo in which we drop an object into a pool of water, creating a splash. To do this, we need a fluid object and an obstacle object. For the fluid, create another cube, scaled down to cover the bottom of your domain. Enable this object in fluid simulation also and select "fluid". When we start to bake the fluids, this will be set up as your liquid. Make sure you bake this animation COMPLETELY before you add the obstacle.

Let us go over what you have now. You have your starting cube, with the "Fluid Simulation" and "Domain" buttons depressed, and will be referred to as "Domain". Inside this is the cube that covers the bottom of your Domain , which is the liquid. The liquid will be a different object with the "Fluid Simulation" and the "Fluid" button depressed. Create a third object which we will call "Obstacle". This will be the object dropped into the water, and will be described further in the explanation.

## The Obstacle

Now, create the object you want to drop into the water. Give it an IPO to drop into the water, and enable it as--you guessed it--an obstacle.

**newbie:** Kamerad...well IPO stands for "Interpolation" or Animation

Newbie: Actually, I didn't guess it. What is an obstacle, and how do I enable the object as one?

Response to Newbie: Go to the Physics panel and enable the fluid simulation then set it as an obstacle. It should make perfect sense if you're playing along in blender as you read.

In my Blender 2.49 b the tab in the Physics panel is **Fluid** and the button is also **Fluid**

## Baking Fluids

Now we get to see what it all looks like. Before we do anything too fancy, however, go to the render buttons (F10) and specify the total number of frames to simulate the fluid in. Most projects don't need the whole 250 frames. Select the domain object, and find the big button that says "BAKE" under its fluid simulation panel. Press it, click the "continue" part of what comes up, and watch your masterpiece evolve. Be patient, blender has to render this simulation just like it would render the final product. The frame count of this rendering replaces the bar at the top of the screen where it says, "www.blender.org" with the blender logo. If at any time you feel that you want to abort this process, press "Esc". Once the bake is finished, open a timeline window and press play. There, you should be able to view the full simulation.

(note: Make sure the bake data is being sent somewhere or the time you spent waiting will have been for nothing!!)

(User edit: Specify a folder in the space at the bottom for blender to keep the temporal data. Else it will not work)

(Yet another User: I tried to Bake my animation, but all it says is something about no images to bake to. I know I need to set a folder or something, but have no idea how!) (You'll get that is you're still in the "Render Options", make sure to go back to the Domain object's "Physics" buttons.)

## Other Fluid Objects

There are two other incredibly useful types of fluid objects--"inflow" and "outflow". The both of these do exactly what they sound like. Inflow objects pour more fluid into the scene, and outflow objects drain it away.

## Keep This In Mind

Blender will make your water its default opaque grey unless you set its color. A good way to make realistic, clear water (as well as glass for that matter) is to edit the color, then apply a simple mirror effect plus transparency effect. After your water has rendered, select the domain in object mode. This selects the liquid in the frame you have just added. From shading [F5] then the 'Materials buttons' (the button that resembles a ball, if when you click on this the colors settings are not up, just press add new under the only panel there), then the panel called "shaders/Mirror transplants/SSS". Go to the Mirror transplants panel, play around with the settings until you get what you fancy. Suitable settings include [the 'Mirror trans' panel of Material:] click 'ray mirror', set rayMir to 0.1 for water, 0.15 for glass, set depth to 7; click 'ray Transp', set IOR to 1.3 for water, 1.5 for glass; then set the Alpha value to 0 while adding an appropriate color from the "Material" panel.

## Notes

This tutorial was just a simple example that leaves much to be desired. Below are some helpful links that will further your understanding of the fluid simulator.

A bunch of Youtube videos demonstrating parameters in the fluid panel (if using Adblock Plus, disable it for this page to work) <sup>[1]</sup>

Some very snazzy fluid examples made by Mpan3 (lots of videos) <sup>[2]</sup>

## Site Under Construction

More to come, so check back in a few days!

## References

[1] <http://www.pkblender.it/>

[2] <http://mpan3.homeip.net/?animations>

# Blender 3D: Noob to Pro/Flying Through A Canyon

---

## Creating the Canyon

### Forming from Textures

- Open up a new Blender file and erase the default cube (x)
- Keeping in the same perspective, create a plane (Space -> Add -> Mesh -> Plane)
- Subdivide the plane into 100 vertices (\*) (w -> Subdivide Multi -> Number Of Cuts: 100)
  - Just double click and enter the value 100
- Scale the canyon up to 5 times its current size
  - Do this by either hitting s and typing in 5.0 or hitting s, while holding Ctrl to snap and positioning for 5.0
- Go to the Shading Tab (F5) and to the Material buttons and hit "Add New"
- Go to the Texture buttons (F6) and hit add new
  - This texture will be used to create our canyon, so give it an appropriate name. In this tutorial, "canyon" was used
- Select the Texture Type to be Wood
- Go back into the Material buttons, go to the Texture tab, and uncheck the canyon texture
  - We will now displace the vertices based on the texture
- Go into the Editing Tab (F9), and under Modifiers, hit Add Modifier
- Choose Displace, and under the box labeled "Texture", type in "canyon"
  - Notice that the vertices have been displaced! However, the material is very jagged. This is because Blender is preserving the original faces on the underlying material, and the original faces don't run in the same direction as the grooves the wood material has created

Note by Noob: *You have to be in Object Mode to see the Modifier effect.*

Note by Dark: *Hit TAB if the Boolean menu pops.*

## Improving the Look

Now to make the canyon more interesting

- Go back to the textures tab, and hit Wood option from Bands to RingNoise and keep the default options here
  - To update the Displace modifier, go back to the Editing Tab, click on the "canyon" label and hit Enter. Doing this seems to refresh the texture.
- You will now see Rippling Texture. But it isn't a Canyon. For this we need to pack the Ripples closer together to get more detail per ripple.
- Go into Edit mode (Tab), and make sure all vertices are selected (select one and hit L). Now scale them to 0.2 of the original size
- Leave Edit Mode into Object Mode (Tab), and scale the Object (as a whole) up to 10.0 of the size
  - What just occurred is that you packed the vertices closer to each other, but not the texture. Then, you took the object as a whole (including the Texture) and up-scaled it. This is the difference between Scale in Edit Mode and Scale in Object Mode
- You should now have a semi-boring canyon, but it does have circuits with which the camera can float around in
- To make the Canyon more interesting, play around with the Texture values for Wood. Do not forget to update the vertices by going into the Editing Tab, and selecting the texture "canyon" under the Displace Modifier in order to refresh
  - I chose the values Noise Size: 1.0 and Turbulence: 15.00 as they produced an somewhat interesting outer ring
    - I found no difference whatsoever between different Noise and Turbulence values on Blender 2.48a, even though I updated the Displacement Modifier in edit mode.
- When we have the values correct, we will Apply the Modifier into the vertices
  - You may desire to sub-divide your vertices even further in order to get better details around the tops and on steep mountain edges. I subdivided mine with another 5 cuts
- Go into the Editing Tab, under the Displace modifier, hit apply.
  - The displacements are now permanently baked into the Mesh

## Eliminating the Unneeded

- We will now reduce the number of vertices. Go into Modifiers, and add Decimate. The mesh will appear to change slightly, but it hasn't, except that 4-vertices faces have been triangulated
- Add another Modifer, Sub Surf, and make sure that the Levels is just 1
- Under the Decimate, slowly reduce the ratio and watch the Mesh. Very tiny mesh details will be changed but the Decimate modifier will keep the shape while reducing vertices.
- I got to 0.07 on my mesh before I decided the detail was getting too low. Remember you will be flying through the canyon, not over it, so the detail on the walls of the canyon is more important.
- Hit apply to bake the reduced vertice count into the Mesh. Smaller vertice counts will make rendering faster.
- Increase the number of levels on your Sub Surf modifer and take a good look, you have just created a canyon! You may reduce the levels or remove the Sub Surf entirely afterwards.

## Coloring the Canyon

### A Rocky Material

This is one method to create a rocky wall. The result isn't particularly realistic but it works for our purposes. Creating materials is a difficult practice, and the first step is to know what you're trying to recreate, perhaps by using actual photographs.

- Create a new material for the cliff
- Under Shaders, set the Spec value to 0.132 (our cliff isn't particularly shiny) and Hard to 20
- Under the Textures Panel, add a new texture in the first slot.
  - Give this texture the Marbel Texture Type, active "Hard Noise", set the Noise Depth to 1, and Turbulence to 1.7.
  - Under the Colors tab, activate Colorband. Change the first color to black with full alpha and the second color to white with full alpha
  - This is meant to give our primitive rock a textured appearance
- In the second slot, add another texture.
  - Set this type to Clouds, give it a Texture Size of 0.025, a NoiseDepth of 2, and active Hard Noise
  - This one will recreate some bumpiness and small inconsistencies in the rock
- In the third slot, add another texture
  - Set the type to Clouds
  - In the Colors tab, modify the colors to go from Black, 0 Alpha, to a Brown of your choosing, 0.750 Alpha. Reposition the second color to Pos 0.820
  - This Texture will simulate dirt on the Canyon
- You will definitely want to do a render now. Reposition the camera and lights, and adjust render options to make for a fast render
- Go into the Materials tab and select your first slot under the Texture tab
- With that selected, under the Map To tab, uncheck the Col, and check the Nor buttons. Set the Nor value to 2.0
  - What we have just done is set the second texture not to affect color, but to affect the angle of the normal in a render. This can be used to reproduce jagged edges on flat surfaces, or waves in water
- Now, select the first texture again. Under the Map Input, adjust the sizeX, sizeY, and sizeZ values to 5.0. Do the same for the second texture.
  - You will want to render here, adjust the values as desired. For more layers vertically, increase the sizeZ value
- You should now have an unrealistic, but sort of rocky Canyon wall.

### Lighting the cliff

- Remove all existing lights
- Creating a new light with the type of "Hemi". Center it over our cliffs, and use Scale (s) to scale it down.
- Create another light, with the type of "Sun". Position it where you want (position doesn't matter with Sun), and use Rotate (r) to position its Ray at the Angle you desire.
  - In this example, the Sun was more of an evening Sun. Its light is at a 45 degree angle on one side. This creates some Canyon walls light, some dark.
- Position your camera and adjust the Energy values of your Hemi and your Sun. The Hemi will light up everything, it's an easy way to give light to canyon walls that would otherwise be completely black. As a starting point, set your Hemi value at 0.4 and your Sun value at 1.5 and observe how that looks from a few angles. Make adjustments if you desire.

## Flying through

### Creating and working a Path

- You will now create a path with which your camera will follow as it flies through the canyon
- From a top down view (NumPad 7), create a new path (Space -> Add -> Curve -> Path)
- Staying in edit mode, move the path over your cliff
- One by one, move all of the starting vertices to form the start of your path
  - Notice that Blender interpolates in between the vertices on the path to form a very smooth path
- When you have the initial vertices roughly where you want them, click on the end vertex, and Extend it (e). Continue adding new vertices like this until you come back around near where you started.
  - The less vertices you use, the better, but do not sacrifice accuracy of the path, often times more complex paths seem more real.
  - Also note that its easy to position the vertices in such a way that the interpolated path goes through the canyon wall. Watch out for this, and adjust if needed
  - If you want to add an extra vertex in between two existing ones, select those two existing ones (using Shift for one), and hit W -> Sub Divide
- When you are satisfied with your path from a top down view, hide the Canyon. Do this by going into the Animation screen (screen choice is at the top, beside the Help button).
  - On the left hand side in a list, you will see a "Plane", or "Canyon" if you chose to name it. Beside that will be an eye. Select that eye, and you will hide the canyon from view. Click it again to return the canyon to view.

*noob note: Alternatively, you can click on the canyon and then press the H button to hide it. To unhide it, press Alt-H*

- Go back to the Model screen at the top. Find and select your camera, and position it just above the first vertex on your path.
  - Don't put it directly on the first vertex as you will not be able to see that portion of the path when looking through the camera
- Make sure the camera is selected and go into the Camera View (NumPad 0).
  - Now, position the camera to be looking down your path. To do this, you can use Trackball. With the camera selected, hit R twice. The movement is a bit tricky but try and position the camera correctly. Left click with the mouse, and hit R once to rotate the camera so that it is facing up.
- When the camera is positioned correctly, select both it and the path (using shift for the path) and hit Ctrl-P. Select the "Follow Path" option.
- Go back to the animation screen. Change the End Frame to 250. Now, on the right hand side, with the path selected, select the IPO type Path on the screen (noob note: in the IPO Editor Window change the dropdown box on the bottom right from "Object" to "Path").
  - We need to make the speed last all 250 frames. While in the IPO Editor view, hit K, or go to View -> Show Keys. Select the last key. It should be at frame 100 already. Simply move it over 150 frames with Grab (g) and Ctrl
- Playback your animation to enjoy it

## Adding details

- Your path is likely pretty boring. Its straight and the camera pans abnormally with it.
- In the Model screen, go into Edit mode on the path.
- Select each vertice, hit Grab (g), and then hit Z. This will move the vertices just up and down. Reposition the vertices as you desire. The up and down motions makes the travel more interesting
- From a Top Down view (NumPad 7), make sure the interpolated path hasn't gone into any canyon walls. Reposition vertices as necessary.
- Go into a camera view (NumPad 0) and start at Frame 1
- Select each vertex as you go. Change the Tilt of each vertex using T. Remember that if you go around multiple times, it will record that as multiple times, it will now reset at 360 degrees.
  - The Tilt will make it seem as if you are banking in the canyon. Planes usually bank into their turns, as do helicopters.
  - This is largely a trial and error process. The Curve will interpolate your tilts to give the appearance of a smooth ride, however it can make some tight corners seem too soft, and some soft corners go in in the wrong direction. You may tilt each vertex many times to get it the interpolated curve to fit just right
- With the banking and the up and down motions, the Path is now quite a bit more interesting. With a larger canyon, you may have done many details. These could have included jerking up and down motions and close calls against the canyon wall if you were animating a high-tension high-speed flying canyon duel.
- Using the IPO editor, you could also make the camera appear to stop, then using the cameras own IPO curves, have it look around. This would give the impression of a scenic flight through the canyon

Warning / Noob Note: If you try to render the entire animation, unless you have a supercomputer or created a very, very short path, expect to be in for a very, very long wait. If you try to render the animation, it's recommended you save the entire thing beforehand before doing so, because you won't be able to do anything while it's rendering.

## Detailing the Canyon with Water

- One last thing is covered in this tutorial, and that is to add some water to the bottom of your Canyon. Water is a very tricky substance to simulate. It comes in a great variety of colors, transparencies, and surface-shapes
- The water we will create isn't the most realistic water there is, but it does look vaguely like water

# Blender 3D: Noob to Pro/Using the sequencer to compile frames into an animation

## Preamble

When you render an animation, often it's a good idea to render the individual frames and then compile them into the animation. This has several benefits over rendering directly to a movie file. Some of them are these:

- If you discover a mistake in the render, you can modify sections of the animation and rerender those frames without having to rerender the whole animation.
- If the render crashes because the power went out or, (psst), blender crashed, you can pick up the render at the last rendered frame without having to rerender the whole animation

## Tutorial

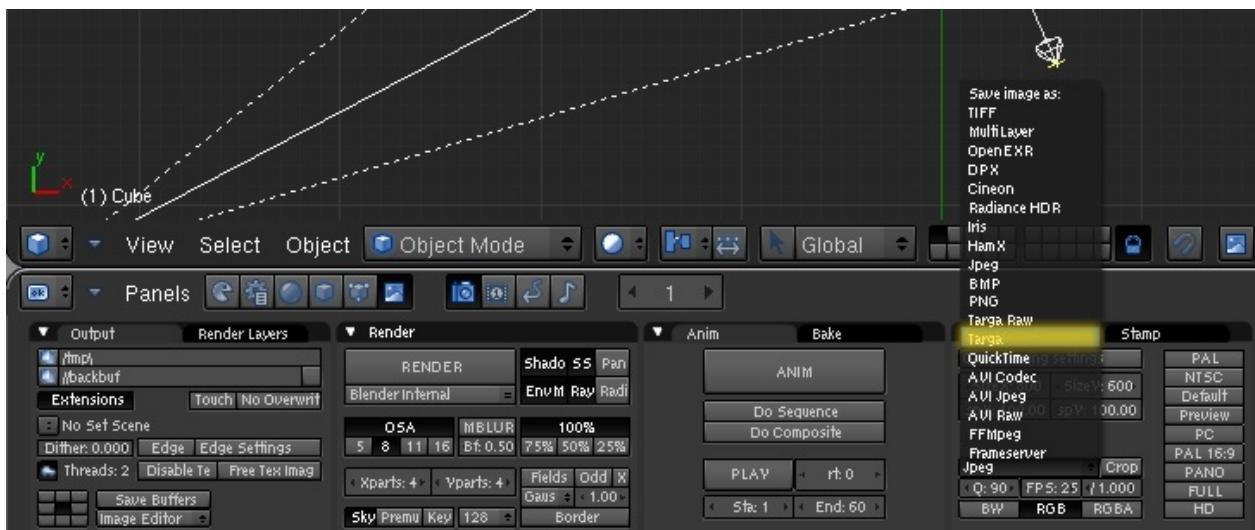
- Open your animation in blender

Begin by opening blender to an animation that is ready to be rendered. Here is a sample file of a simple cube dance if you don't want to make your own.

Download simple animation .blend [1]

- Set the render output to targa format.

press F10 to go the scene buttons and set the output format to Targa. Though you can use any image format you want, I use Targa because it has no compression and is at 100% quality.



- Set the file output location.

Set the location on your file system where you want all the frames to be saved to.

**IMPORTANT: Make sure that there is a slash at the end or else you are setting a filename not a filepath**



- Set Endframe to match the animation

If you animated 500 frames, set the end frame to that same number. the default is 250.



- Render the animation!

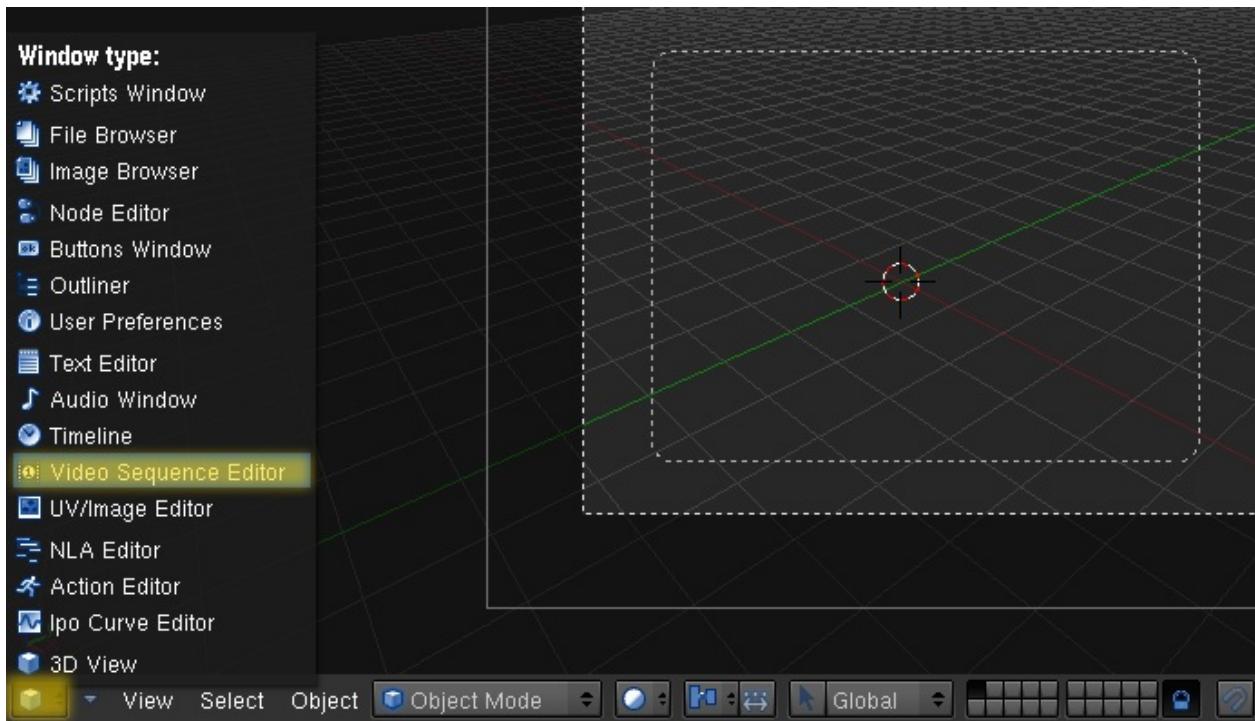
Simply press the "Anim" button.



## Use the Sequencer to compile the images into a movie file

- Open a new blender and delete everything except the camera.
- Change the 3d window into the sequencer window.

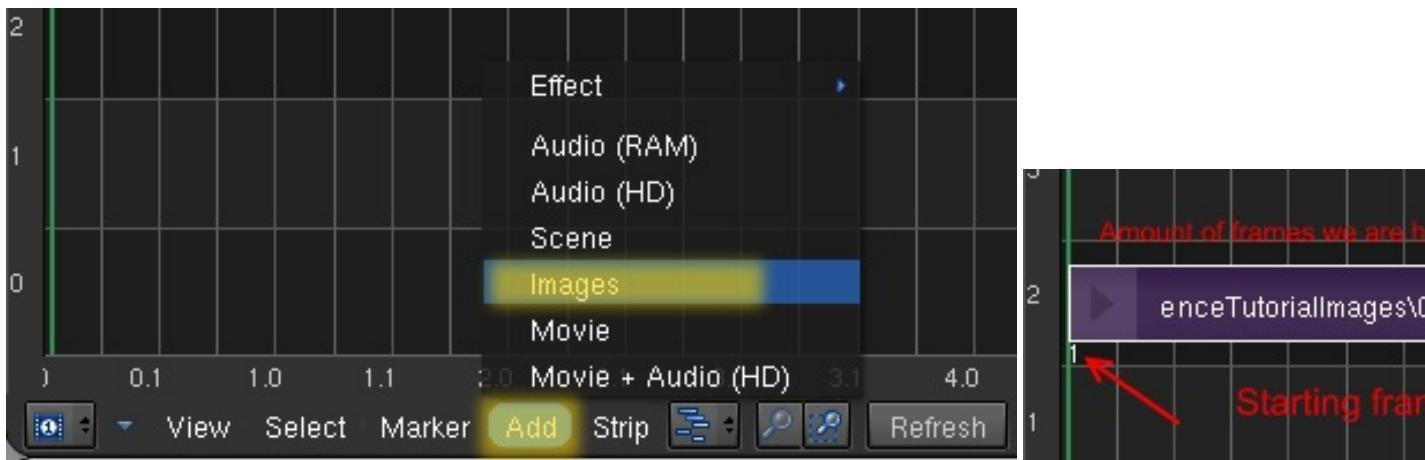
(See image for explanation)



- Add the images to the sequencer.

On the bottom of the sequencer window click Add>>Images then browse to the location where you saved the images. Press Cntrl-A (on linux enter "\*.tga" or "\*.jpg" in the field for filename) to select all the images and finally press "Select Image" on the top-right. Attached to your mouse curser will be a long horizontal bar which represents

all the images. Move it all the way to the left till the little number underneath it shows "1" and then Left-mouseclick to place it.



- Set final movie format.

Note that you can also set the location that the movie is saved to by changing the path we set earlier.



- Enable Sequence render and animate!

Depress the Do Sequence button and press the Animate button and wait a few seconds till it's done. Browse to the location you saved your movie and enjoy the animation.




---

I hope you found this tutorial useful. If there are any errors in the tutorial, *Please let me know, and I will fix them as soon as possible.*

Blend happy and save often, --Radialronnie (talk) 22:47, 3 November 2008 (UTC)

## References

[1] <http://dm-photographie.com/FileCellar/Animation.zip>

# Blender 3D: Noob to Pro/Particle Systems

*This tutorial was written for Blender v2.49*

Particle systems are used to simulate large amounts of small moving objects, creating phenomena of higher order like fire, dust, clouds, smoke, or fur, grass and other strand based objects. You may also use other objects as a visualization of particles.

Before you start with the tutorials, you should at least take a brief overview about the very extensive manual pages<sup>[1]</sup> of the particle system. You will find every single parameter explained in the manual if you have the desire to delve deeper ...

Don't forget: particles alone don't do any magic. They are only a placeholder for something nice to view. You have to take care of the visualization also, and that is usually the harder part than to create the particle system.

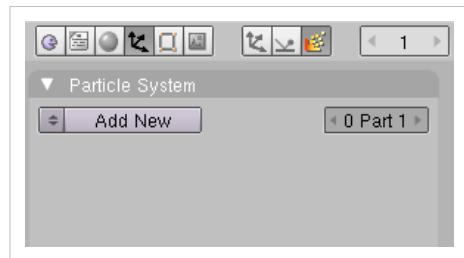
## The very first particle system

### Creating a particle system

To create a particle system:

- insert a plane
- select the object (only mesh objects can emit particles)
- change to the object buttons and to the particle subcontext
- click on **Add new** (**Img. 1a**)

Voila, your first particle system! It doesn't do anything useful now, but we're going to change that on the following pages.



**Image 1a:** Create a new particle system

- To see any particles press **Alt-A**. The frames change and you see particles appear. Press **Esc** to stop the animation.

You return to frame 1. The particle movement is cached (stored), so if you play your animation again it will go faster (well, you won't notice any difference in this simple case).

- If you want to stop in a certain frame, press **Space**.
- To see the particles even better change to wireframe mode (**Z**),

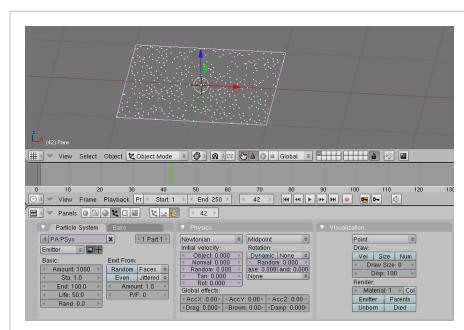
If you change anything in your particle system you always have to return to frame 1, to recalculate the system from start.

Use a *Timeline* window to change easily between frames (**Img. 1b**).

### Ultra Physics Coolness:

- Activate *Continue Physics* in the *Timeline* window. Menu: *Playback->Continue Physics*
- Hit the *Play Timeline* button in the *Timeline* window.

Now you can move the objects in the 3D window and the particle system is updated in realtime. The system is not cached then.



**Image 1b:** The very first particle system



## Changing properties of the system

The most important properties are probably:

- *Amount*: the total number of particles
- *Start/End*: the start and end frame of the emission
- *Life*: the lifetime of the particles
- The initial velocity. This is often set with *Normal* causing the particles to get a speed in the direction of the face normal (if emitted from faces).
- A constant global force. *AccX/Y/Z* set's a force in the respective direction.

For now change:

- the *Amount* to 5000
- the *Lifetime* to 100 frames
- the *Normal* value to 1.0.

The Particles are emitted directly upwards (if your plane is orientated that way) and live for 100 frames.

If you render now, you will see a white "Blob". This is the default *Halo* rendering of the particles.

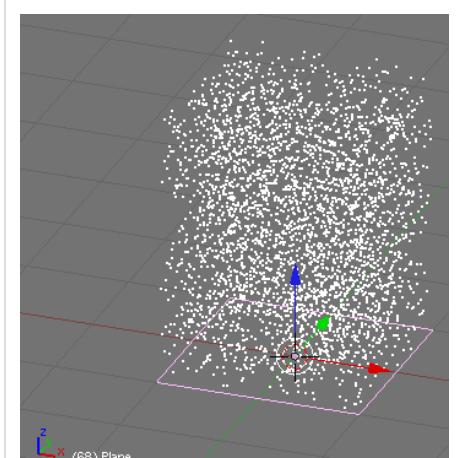
## Changing the material of the particles

- Change to the *Shading* buttons and assign a material to the plane.
- Activate *Halo* in the *Links and Pipeline* panel.

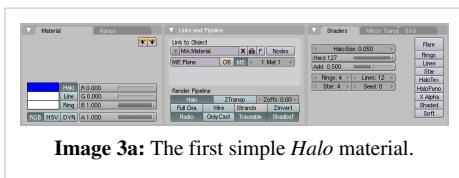
The *Halo* settings are in the *Shaders* panel. The color of the *Halo* is set in the *Material* panel (see also the Manual on Halos<sup>[2]</sup>). Halos are a post rendering effect, that is applied after the scene is finished. So halos can't shed any lights on other objects, they are not rendered behind *RayTransp* materials (like glass).

- Set the *Halo* color to deep blue (RGB: 0/0/1)
- *Halo Size*: 0.05 so each halo is quite small.
- *Hard*: 127 so that each halo has the maximum sharp edge
- *Add*: 0.5 so that the brightness increases where several halos overlap

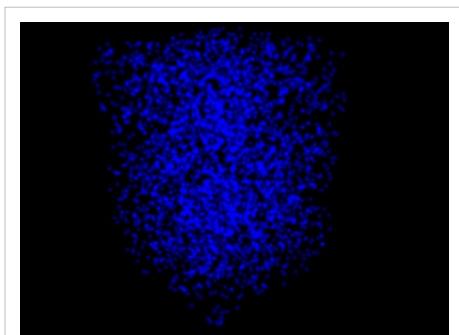
Set the world color to black and render (**Img. 3b**). Nothing special till now, but that will change soon. So proceed to the next page, where we're going to make some fire.



**Image 2a:** Particles emitted in the direction of the face normal of the plane



**Image 3a:** The first simple *Halo* material.



**Image 3b:** Our first particle system rendered in frame 68

## Links

- Blender 2.49b Basic Particles Video Tutorial: <http://www.youtube.com/watch?v=dwsR8v5VE4M>

## References

- [1] <http://wiki.blender.org/index.php/Doc:Manual/Physics/Particles>  
[2] <http://wiki.blender.org/index.php/Doc:Manual/Materials/Halos>

# Blender 3D: Noob to Pro/Making Fire

*This tutorial was made with Blender v2.48a*

We're going to create a camp fire with a simple particle system. This tutorial is based on the method described in the Blender Manual [1]. The result of this tutorial is shown in **Fig. 1**, the Blend-File is included at the bottom of this page.

If you need more realistic looking fire, you should use the method described in BlenderArt Magazine No. 16 [1], though that method is more advanced and uses Compositing Nodes heavily.

The starting point of the tutorial is how fire behaves physically. The flames are made of hot gases. These accelerate upwards due to their lower density in contrast to the cooler air in the environment. Flames are in the middle hot and bright, to the outside they are darker.



**Figure 1:** My idea of particle fire.

## The particle system

I've created the usual scene with some stones and a few pieces of wood (the wood is by courtesy of Teeth). (**Fig. 2a**).

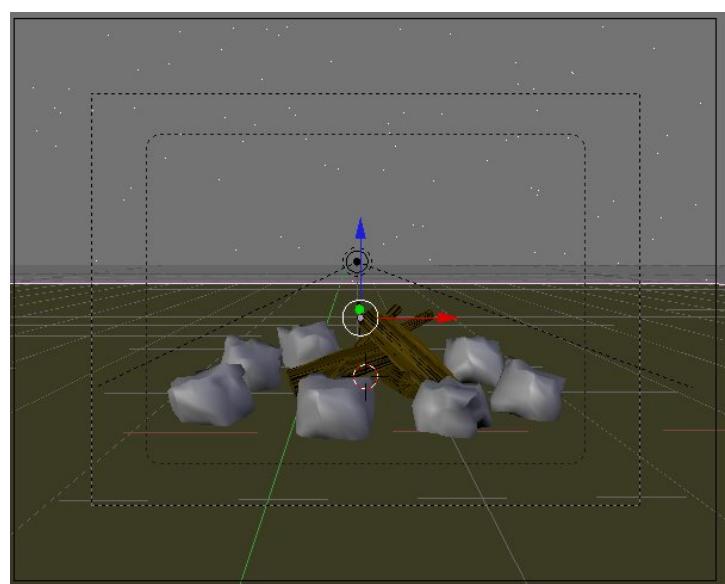
- Add a *Plane* in the middle of the stones.

This will become the particle emitter.

- Rename the *Plane* object to "Emitter".

If you use good names you will find it much easier to orientate yourself in your scene later. Having 100 objects named "Cube.something" will make it very difficult to quickly select a desired object.

- Subdivide the plane once in *Edit mode*.
- Change the shape of the plane, so that its shape equals the base of the fire.
- Change to *object mode*.



**Figure 2a:** A simple scene

- Change to the *Particle* buttons of the *Object* buttons.
- Create a particle systems (*Add New* in the *Particle System* panel - and make sure the plane is selected!).
- Type: Emitter* The plane emits the particles.
  - Amount:* 9000 The total amount of particles.
  - Sta:* -45.0 The simulation shall start before the rendering, to have a fully developed flame in the first frame.
  - End:* 200 The simulation shall last 200 frames - here: the particles are emitted till frame 200.
  - Life:* 16.6 I've adjusted the lifetime of the particles to their speed. Both parameters together regulate the height of the flame.
  - Rand:* 0.5 The lifetime is changed randomly.
- Emit from:*
  - Random*
  - Faces*
  - Even*
  - Random*

This creates particles with a random distribution on the faces of the emitter object.



Please note, that the particle simulation is only fully calculated if the bake parameters in the *Bake* panel match the positive lifetime of the particles. If you want to run the simulation longer than 250 frames, you have to increase the *End* frame in the *Bake* panel as well. This is independent whether you want to bake or not.

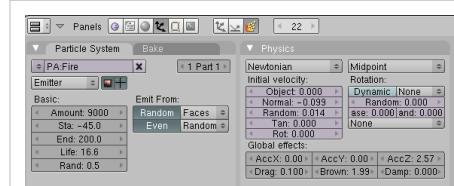
The movement of the particles is controlled with particle physics. You set the *Initial Velocity* and let the physics do the rest.

- Normal:* -0.099 The particles are emitted slightly against the direction of the face normal. This leads to a bit wider fire at the base.
- Random:* 0.014 This creates a random start velocity as well in speed as in direction (you could use a texture to randomize only the speed, see the discussion page for that).

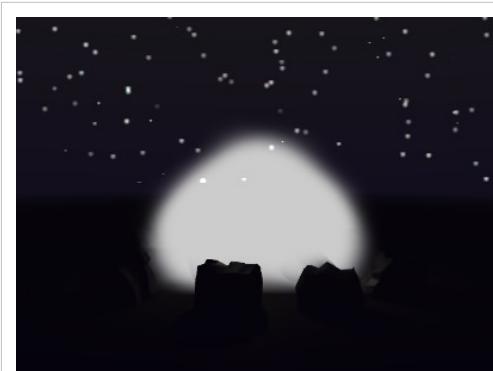
After you have given the particles an initial velocity they are moved by forces.

- AccZ:* 2.57 A force in positive Z direction (upwards).
- Drag:* 0.1 Air drag decelerates the particles.
- Brown:* 1.99 Random movement simulates agitated air movement.

The particle system is finished. Until now it doesn't look like much (see the white Blob in **Fig. 2c**). Therefore the emitter will get a material, this material will be animated.



**Figure 2b:** The particle system



**Figure 2c:** Particles without material

## Material

- Create a new material for the emitter, call the material *Flames*.
- Activate the *Halo* button in the *Links and Pipeline* panel. Else we couldn't set the particle parameters. The particles would be rendered with the default Halo values.
- *Halo color*: 1/0/0 (red)
- *Alpha*: 0.8 The particles shall always be a bit transparent.
- *HaloSize*: 0.132 I wanted many, but fairly small particles.
- *Hard*: 45 The transition from fully transparent to fully opaque.
- *Add*: 0.875 Several *Halo*s over each other combine their power. This makes the fire in the center really bright.
- *HaloTex*: A *Halo* can bear an individual texture, but only the texture in the first texture slot is evaluated.

To give the *Halo* a bit more structure, give it a texture:

- Add a new texture in the first texture slot.
- *Map To*:

- *Col*
- Color: Bright yellow (1/1/0.664)
- Use an adjusted *Clouds* texture with a *NoiseSize* of 0.6.

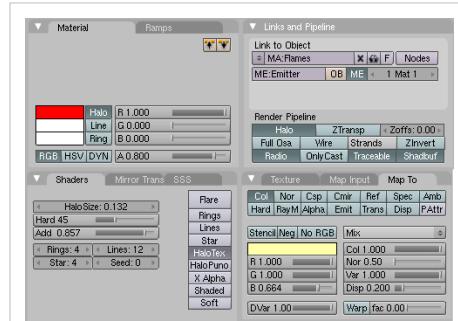


Figure 3a: Material for the emitter object.

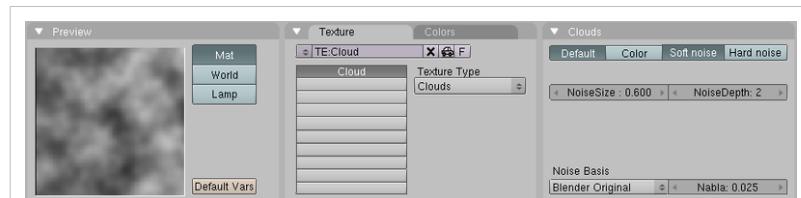


Figure 3b: Adjusted texture

## Animation of the particle material

The particles "pop" into life and vanish suddenly. We should change that. Therefore we're going to animate the *Alpha* value of the particles.

- Make sure the material buttons are visible in the buttons window.
- Change to frame 21, move the mouse cursor over the button window and press **I->Alpha**. This is going to be the maximum visibility of the particles.
- Change to frame 1. Change the *Alpha* value to 0 and insert the next key.
- Change to frame 100. Change the *Alpha* value to 0 again and insert the third key.

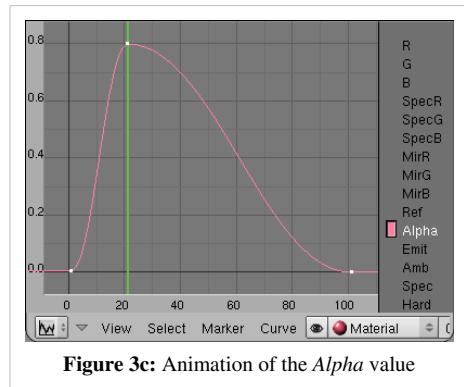


Figure 3c: Animation of the *Alpha* value

If you want to see the IPO curve in the *IPO Editor* window you must change the *IPO Type* selector in the window header from *Object* to *Material*.



*Note:* An animation of particle material is mapped from the first 100 frames to the lifetime of the particles. I.e. if the material is faded out during the first 100 frames (the ipo curve is 100 frames long) the particle will be faded out during its own lifetime, no matter how long that is. This holds true however only for *Point* visualization of particles, not for object visualization.

The *Alpha* value therefore changes during the individual lifetime of each particle from 0 to 0.8 and back to 0 (**Fig. 3c**).

## Rendering

Our particle animation is finished.

- Change the end frame in the *Anim* panel of the rendering Buttons to 200 and click on *Anim*.

**Note:** If after rendering your particles are too small, such that the fire doesn't look realistic, try increasing the *Halo size* slightly. I used **0.300** instead of **0.132**

To actually let the fire glow you have to use one or more lamps and animate them as well. But that would be part of another tutorial ...

## Links

- The resulting blend file: [2].

## References

[1] <http://www.blenderart.org/issues/>

[2] [http://www.snapdrive.net/files/180816/Blender3D\\_SimpleParticleFire.zip](http://www.snapdrive.net/files/180816/Blender3D_SimpleParticleFire.zip)

# Blender 3D: Noob to Pro/Furry

This tutorial was created using Blender v2.48a. There is an older version of this page created with Blender v2.40.

This tutorial deals with fur, i.e. lots of relatively short hairs covering a body. We will use particles to create the fur, and discuss a few aspects here:

- How to determine the length and the thickness of the hair.
- How to determine the place to grow the hair.
- How to color hair.
- How to render efficiently.

The particle system is far too complex to show more than one method in this tutorial. You can achieve many of the same effects shown here in different ways.



**Figure 1:** The result of this tutorial: some furry thing

## The emitter

- Remove the cube.
- Add an *UVSphere*. This will become our emitter.
- Change to the *Particle* buttons in the *Object* buttons (**Fig. 2a**).
- Click on *Add New*.
- Rename the particle system to "Fur".
- Change the particle system type to *Hair*.

A *Hair* particle system has a lot of specialties, the most important thing is that we can edit the particle "motion" by hand if we want to. Apart from that normal particle physics apply, so everything a particle does hair can do also and vice versa. A particle hair shows the way of the particle during its lifetime at once. To do that efficiently not every single frame is rendered as a point, but a certain number of control points are calculated. Between these control points there will be drawn an interpolated path. The number of control points is the number of segments + 1.

For fur you need lots of particles, like 1.000.000 upwards. This will hurt us badly if we have to deal with so many particles in the 3D window and want to render it. Therefore we will create the great amount of particles with so called "children", that mimic the behavior of their parents. The amount of particle parents should be as low as possible, but you need a certain amount to control the distribution of the hair. We will also use as little control points as possible, three segments should be enough for short fur.

- Set the *Amount* to 1000.
- Set the number of *Segments* to three.
- *Emit from:*
  - *Random:Faces*
  - *Even :Random*

This will create a nice, uniform distribution.

Let the hair grow - the hair shows the path of the particle:

- Set *Normal* to 0.05.
- *Random* 0.005

Nothing special here: the hair grows in the direction of the face normals. Length and direction are a bit randomized.

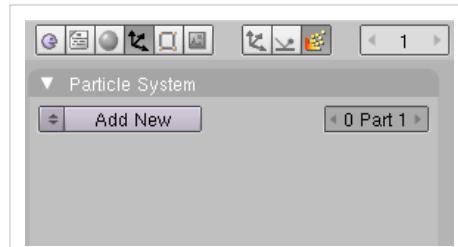
The *Visualization* type changes automatically to type *Path* if you select a hair particle system. If you would render now, you couldn't see the emitter object any more.

- Activate *Emitter* in the *Visualization* panel.
- Activate *Strand Render*.

The *Strand Render* (which I have baptized *keypoint strands* to differentiate from the "normal" *polygon strands*) renders the hair strands extremely efficiently and magnitudes faster than the normal strand. It is the only way to handle many hairs in terms of memory consumption. But it has a few disadvantages:

- They are not seen by raytracing, so you don't get raytracing reflections and no raytracing shadows. You can use environment mapping to compute the reflections and *Spot Lamps* with buffer shadows for the shadow.
- If the hair is very thick (like 1 BU) sometimes the shape is not correct.

- Activate *Children from Faces*.
- *Amount: 5* This is the amount of particles visible in the 3D window. Keep this low.



**Figure 2a:** Adding a particle system.



**Figure 2b:** Particle system settings

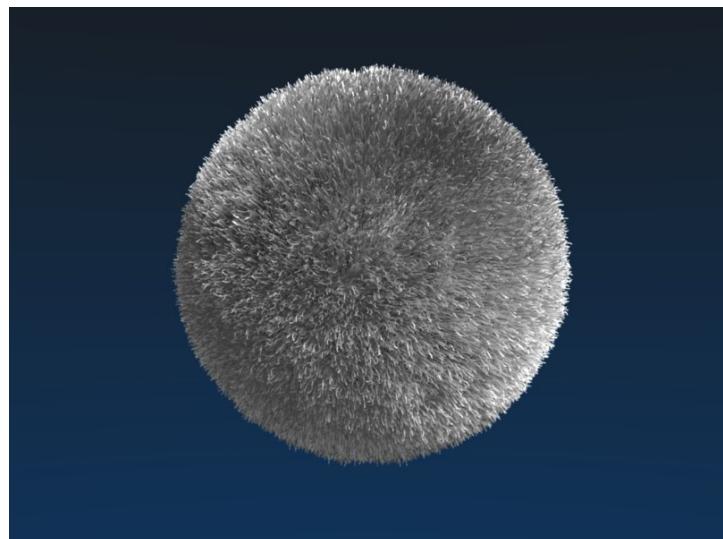
- *Render Amount:* 50 This is the amount of particles during the render.
- *Rough 2:* 0.1 Random variation of the shape of the particles. So the hairs will not stand plain upright and appear a bit curly.

The parent particles are not rendered by default, so now we have 50.000 Particles that render on my old machine in 6 seconds. If we use 1.000 children we have 1.000.000 particles, that need approx. 1 GB of RAM and render in 1:42 minutes. If you render *keypoint strands* with *Children from Faces* you can also use *Child simplification*, which will reduce the amount of particles on objects far away from the camera automatically.

Now we should change the lighting to get a preview.

- Select the lamp.
- Change to the *Lamp* buttons.
- Change the lamp type to *Spot*.
- Change the shadow type to *Buf. Shadow*
- Change the buffered shadow type to *Classic-Halfway*.

This is a great shadow type that renders keypoint strands very well and creates fewer artifacts than *Classical* (In my opinion). I have inserted two other lamps and used a the classical three point lighting for the first rendering (**Fig. 2c**).



**Figure 2c:** The first render without material.

## Material

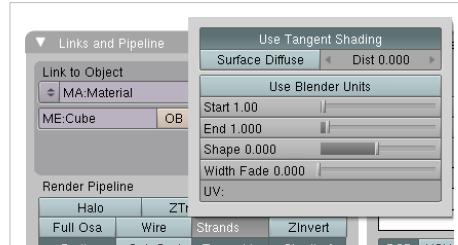
In the material buttons you can set different aspects for the strands:

- their width and form
  - the used shader
  - the base color
  - a texture along the strand
  - different particle attributes like length, density or roughness
- Add a material to the emitter.
  - Name the material "Fur".

## Strands Shader

The default strands settings for *Keypoint strands* are shown in **Fig. 3a**. Take a look in the Manual about Strands [1] for an explanation of all settings.

- Change the *End* value to 0.25, this will make the hair more spiky (not shown in **Fig. 3a**).



**Figure 3a:** Strands settings in the material buttons.



*Surface Diffuse:* It might be easier to achieve a good lighting environment if you check *Surface Diffuse*. With *Surface Diffuse* turned on, hair does not show as many highlights, which may be too bright without it. I've seen many great examples of fur that use this parameter. However, you have to decide right at the beginning whether you want to use it because it effects all the other material settings. This is something you have to decide for yourself.

## Giving the hair its base color

Strands are rendered with the material of the underlying face/vertex, including shading with an UV-Texture. Since you can assign more than one material to each face, each particlesystem may have its own material and the material of the underlying face can be different from the material of the strands. We will use an UV texture and use it for the surface of the emitter as well as for the color of the hair.

- Change to *Front* view in the 3D window (*View->Front*).
- Make sure you are in *Orthographic* view mode (also in the *View* menu).
- Change to *Edit* mode of the sphere.
- Press **U** to unwrap, select *Sphere from View*. This is a quick and well working method to correctly unwrap a sphere the easy way.

You don't need to assign a texture in the *UV/Image Editor*, we only need the coordinates now.



**Figure 3b:** Fur color texture

- Add a texture to the material, name it "FurColor".
- Set *Map Input* to *UV*.
- Go to the *Texture Buttons* and set *Texture-Type: Image*.
- Load an image texture. I have used the image in **Fig. 3b**.

Normally I would just stop here, I think the material is good enough. But if you want to make the fur more fluffy and soft, you should add a second texture along the strand, which changes the alpha value.

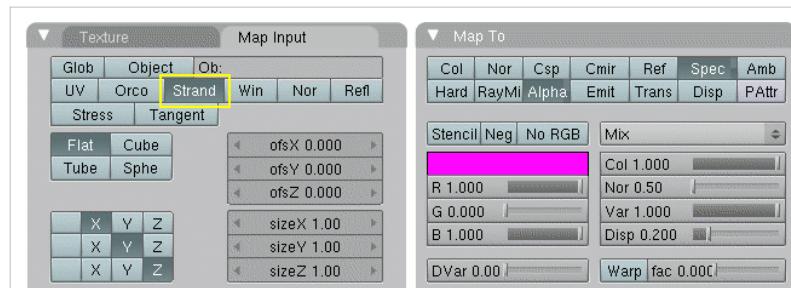


**Figure 3c:** Emitter with color texture.

If you want to do that:

- Activate *ZTransp*.
- Add a second texture.
- *Map Input: Strand*
- *Map To: Alpha and Spec, DVar=0*
- Use a blend texture (*Linear* or *Quad*)

You can change all other properties this way, for example the color along the strand (bleached tips).



**Figure 3d:** Settings for a texture along the strand.

## Changing Hair length with a texture

At first I will show you how to render the emitter mesh with a different material than the strands. Then I will show how to change the length of the hair with a texture semi-interactively.

- Change to the *Editing Buttons*.
- Change to *Edit mode*.
- In the *Links and Materials* panel click on *New* in the material section (right hand side).
- Make sure all vertices are selected.
- Click on *Assign*.
- Change back to *Object mode*.

Now the emitter bears a second material.

- Return to the material buttons.
- In the *Links and Pipeline* panel click on the **X** next to the material name (*Deletes link to this Datablock*).
- Add a new material.
- Name it *Emitter*.

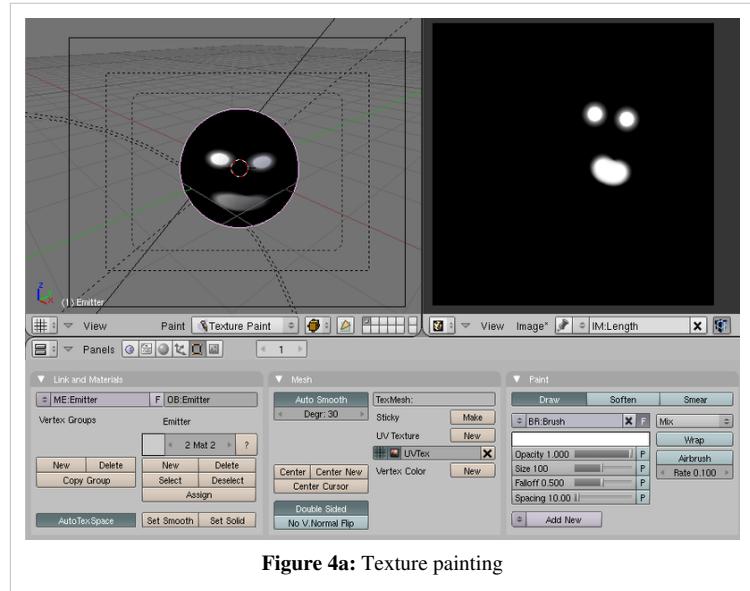
Now you have a new material on your emitter object. Since the particle system uses material no. 1 you can use different settings for the emitter.

We have already unwrapped the emitter, this is something that will probably be the case also for any real models. Now we will use an UV-Image and texture painting to determine the hair length.

- Split the 3D window.
- Change the right hand side to an *UV/Image-Editor* window.
- Change to edit mode of the emitter object.
- In the *UV/Image-Editor* use *Image->New...* and confirm the default settings. This will create a new image, that we will paint on.
- Click on the package icon in the windows header of the *UV/Image-Editor* window. Confirm.
- Change the object to *Texture Paint* mode.

Now you see the texture on the object.

- Paint a structure on the object.
- Change back to object mode.
- Go to the *Material* buttons.
- Change the active material to *Fur* (click on the arrows in the *Links and Pipeline* buttons where it reads *2 Mat 2*).
- Create another texture. Name it *FurLength*.
- Set *Map Input* to *UV*.
- Set *Map To*
  - Turn *Col* off.
  - *PAttr*



**Figure 4a:** Texture painting



**Figure 4b:** Material settings for setting the particle length with a texture.

- *Length*
  - *DVar=0* All the white areas on the texture will produce a particle length of 0.
- Load the image texture that we have painted.

The result is shown in **Fig. 4c**, you can also see the particles change in the 3D window.

There would have been other ways to achieve this result, e.g. with vertex groups or with particle editing. But I like to work with textures, because you have very fine control and may change the strength of the effect at any time. Vertex groups don't allow for such fine control or you need very many vertices in the emitter. Particle editing (what we will do in the next step) is lost if you change the base particle settings late on, and you can't change it's effect so easily.



**Figure 4c:** Controlling particles with a texture: result

## Comb it!

An effect that is often underestimated is the importance to comb fur in the natural directions. Fur doesn't simply stand upright, and it also doesn't follow gravity (or only to a small amount). So back to the particle system!

- Change back to the *Particle buttons*.
- Click on *Set Editable* in the *Particle System* panel.
- Change to *Particle Mode*.

*Particle Mode* only appears if you have made the particle system editable, and only hair systems can be made editable. There are a few lifesavers to know when working in *Particle Mode*.

- you can edit only the control points (remember the setting *Segments* from the beginning of this tutorial)
- you can only edit parent particles, so you need enough parents for good control
- Activate *Limit selection to visible*
- Activate *Point select mode* for even finer control.

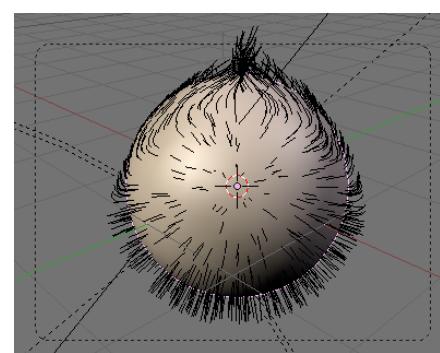
Both settings are in the window header of the 3D window.

- Open the *Particle Edit Properties* panel with **N** key in the 3D window.
- Select *Comb*.

You have quite a few different tools at hand in *Particle Mode*, see the manual on *Particle Mode*<sup>[2]</sup>.

- Comb the hair following the natural flow.

On the example sphere I have used here it is a bit difficult to tell what the natural flow should be ;-. So I have just very carefully combed and changed the length at a few places a bit. You find the rendered result in **Fig. 1**, the Blend file is linked below.



**Figure 5a:** Combing in *Particle Mode*

## Links

- The result of this tutorial: [3]
- A tutorial in German that shows how to create Grass with a very similar method.
- Awesome fur shown here: Tiger [4]

## References

- [1] <http://wiki.blender.org/index.php/Doc:Manual/Materials/Properties/Strands>
- [2] [http://wiki.blender.org/index.php/Doc:Manual/Physics/Particle\\_Mode](http://wiki.blender.org/index.php/Doc:Manual/Physics/Particle_Mode)
- [3] [http://www.snapdrive.net/files/180816/Blender3D\\_FurTutorial.zip](http://www.snapdrive.net/files/180816/Blender3D_FurTutorial.zip)
- [4] <http://blenderartists.org/forum/showthread.php?t=124839>

# Blender 3D: Noob to Pro/Fireworks

*This tutorial was made with Blender v2.48a*

How to create a firework from particles? We will use cascaded particle systems especially of the type *Reactor*.

Abstract:

- We create a emitter object in the appropriate size.
- Then we use three successive particle systems:
  - the first of the type *Emitter*
  - the second of the type *Reactor*. This system reacts to the death of the emitter particles.
  - the third again of the type *Reactor*. This systems reacts to the proximity of the second systems and thus creates a "drag".
- We create three different materials (one of them animated) and assign them to the three particle systems.



**Figure 1:** Some particle fireworks

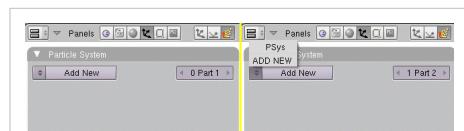
## The Emitter

We use a *Plane* as emitter object. Scale it to your liking. I have used a relatively large particle system, so my plane has a length of 15 BU (Blender units).

- Add a particle system (**Fig 2a**, left hand side).

- *Amount:* 25
- *Sta:* 1
- *End:* 250
- *Emit from:*
  - *Random*

- *Faces*
- *Even*
- *Random*
- *Initial Velocity*
  - *Normal:* 22
  - *Random:* 7
- *AccZ:* -9.8



**Figure 2a:** Adding particle systems. First particle system on the left hand side, a second particle system on the right hand side

- *Visualisation: Line*
- *Back: 1.1*

A few particles (25) are created in the first 250 frames of the animation and emitted upwards. Gravitation, velocity and lifetime are adjusted so, that the particles reach the end of their lifetime at the topmost point of their trajectory.

The *Line* visualization lets the particles appear as long drawn-out line.

- *Bake* the particle system. Use 500 as the *End* frame for bake. 300 would be sufficient here, but 500 do no harm.

The plane will get a *Halo* material.

- Activate *Halo* in the *Links and Pipeline* panel.
- *Halo*: color red
- *Halo Size*: 0.421
- *Hard*: 35

I've adjusted the size and hardness of the halo so long until I liked it, so there is no specific reason to use these values.

## Reactor 1

- Click on the arrow next to *1 Part 1* in the *Particle System* panel.
- The settings for the first particle system are no longer shown, now it reads *1 Part 2* (**Fig. 2a**, right hand side).
- Click on *Add New*, now it reads *2 Part 2*.

You just have created a second particle system. Rename this system, a good naming convention will help you a lot to keep the overview.

- Activate *React on: Death*.
- *Life: 104*
- *Emit from: Particles*

You don't have to set anything for the *Target*. It is not necessary to set the target object if it is the same object as the reactor. You just have to set the number of the target particle system eventually.

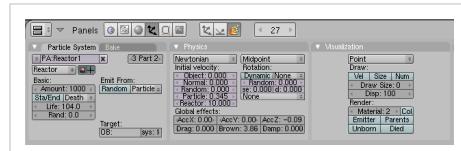
The *Reactor* particles react on the *Death* of the particles of the target system. They will be emitted from the point that the particles occupy at their death.

To control the movement of the system, I have made following setting in the *Physics* panel:

- *Particle: 0.345*
- *Reactor: 10*
- *AccZ: -0.09*
- *Brown: 3.86*

Because of the *Reactor* setting the star is moving away from the emitting particles. The *Brown* movement lets the trajectory appear trembling like affected by wind.

- Set the material number to "2" in the *Visualization* panel.
- Bake the system, again use 500 as end frame.



**Figure 3a:** The first reactor system

## Material 2

- At first you have to assign a new material in the *Link and Materials* panel of the *Editing Buttons*.
- Use the material settings from **Fig. 3b**,
- The material also gets a *Clouds* texture with *Noise* depth "3", simply leave all other settings unchanged.



**Figure 3b:** Material settings for the 1. *Reactor* system

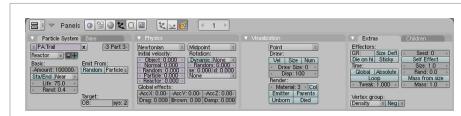
The texture is actually superfluous here, but the next particle system is given the same texture, and colors should be adjusted a bit.

- Animate the *Alpha* value of the material.
  - Set the first Ipo key in frame 1 (*Alpha=1*). To do that move your mouse cursor over the *Buttons* window. Press **I->Alpha** (and change ipo type to material in the ipo curve editor if you are using that).
  - Set the second Ipo key in frame 86 (*Alpha=0.75*).
  - Set the third key in Frame frame 101 (*Alpha=0.0*).

The second particle system is faded off relatively quickly, but the particles don't disappear suddenly.

## Smoke Trail

The third (and last) particle system is again a *Reactor* system and is reacting to the second system. It's going to react to the *Nearness* of the particles. Without moving the variation of the particles is created with a texture and a random variation of their lifespan. What Blender considers as close can be changed with the particle *Size*.



**Figure 4a:** Particle system for the trail

- Create a third particle system (like you created the second).
- Use the settings from **Fig. 4a**. Important are the settings:
  - Amount:* 100000
  - React on:* Near
  - Emit from:* Particles
  - Life:* 75
  - Rand:* 0.4
  - Target Sys:* 2
  - Material:* 3
- Bake the system, again use 500 as end frame.

## Material 3

- At first you have to assign a new material (the third) to the object in the *Link and Materials* panel of the object buttons.
- Assign the existing *Clouds* texture from material 2 also to material 3, but select a dark blue as target color (in the *Map To* panel).



**Figure 4b:** Material for the *Trail* particle system

## Render

- Render your animation, use end frame 500.

That's all. I hope you were able to follow the tutorial. Criticism and suggestions, write best on the talk page or improve the article directly.

# Blender 3D: Noob to Pro/Soft Body Animation

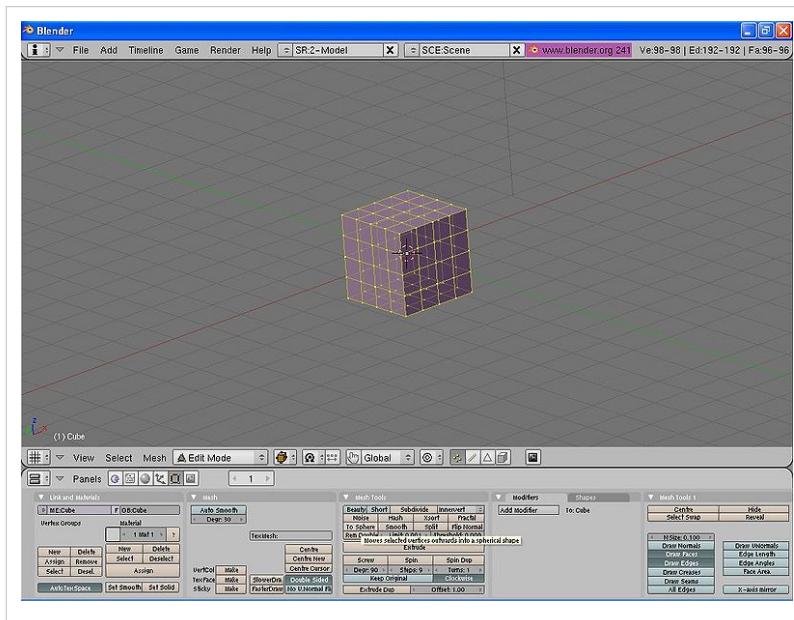
Due to rapid updates in Blender, some menu items may be different from those mentioned here. If the menus don't agree then just do what seems logical.

Softbody makes each individual vertex its own object that falls according to gravity and reacts to outside forces like fields. By adjusting the settings, you change the behaviour of the edges that connect the vertices. For example, you can make it so edges can stretch really far (aka elasticity), allowing the vertices to become distant, or you can make the edge stiff, so the vertices will always stay the same distance apart.

To put this in perspective, picture two cloth's, one elastic and one cotton. The elastic one has edges that can extend, so if you view them in wire-frame (with vertices and edges visible) you would see the edges are more extended than an equal distance. The cotton one would only stretch a little bit, so the vertices would stay essentially the same distance apart.

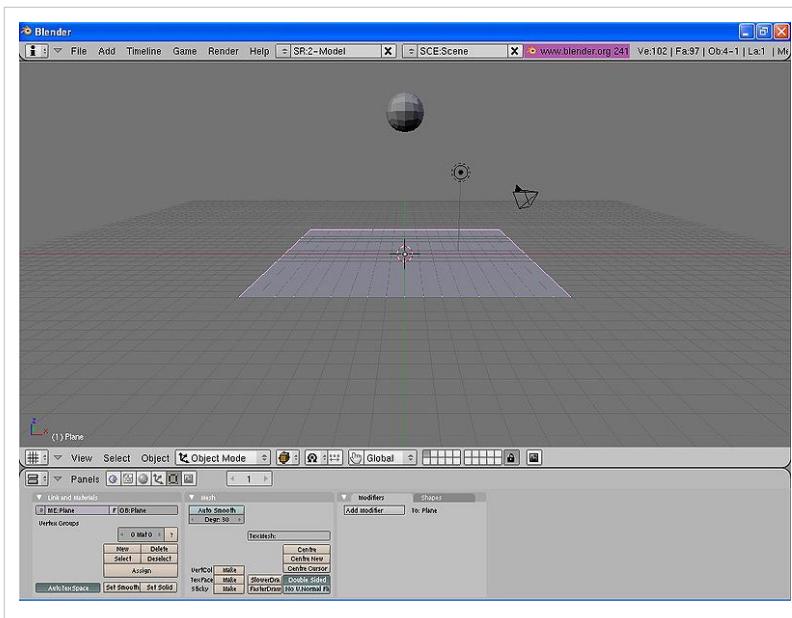
We are going to make a big rubber ball, but not a big bouncy one, a flat (and somewhat lifeless) one. Start with a sphere. I would use a cube sphere or an icosphere, UV spheres don't deform well as they have too few vertices. A cube sphere is made by subdividing a cube and doing a "To Sphere" in the Edit window, under Mesh Tools.

Move the sphere up and place a plane below it. Make sure to do this in the right views so that it is aligned properly. Gravity acts on the z-axis (sphere should be above the plane relative to the z axis).

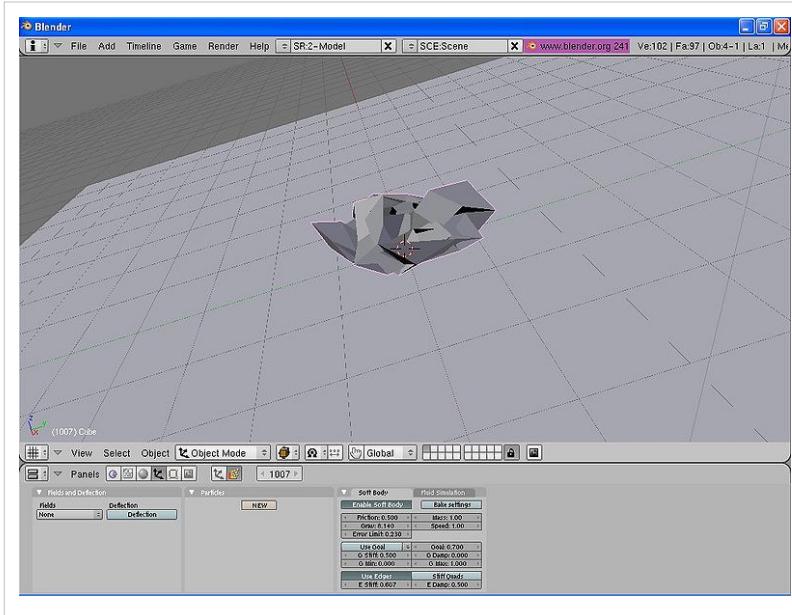


Now for the soft body select the ball and go to the object tab then the physics subtab or whatever your version has. Click "enable softbody" and then turn up 'Grav' to 9.8. Click off use goal. Press the > arrow key (a few times) and you should see the ball fall. The center will remain in place but this is not a problem. If you are on a slow machine you will notice lag. This is because blender moves it vert by vert, not efficient.

Note: In Blender 2.44, click F7 on keyboard, select the "Physics Buttons" button, select "Soft Body".



When it reaches the bottom it will go through the plane. To fix this we must make the plane effect the softbody. To do this make the plane deflect in its physics buttons.

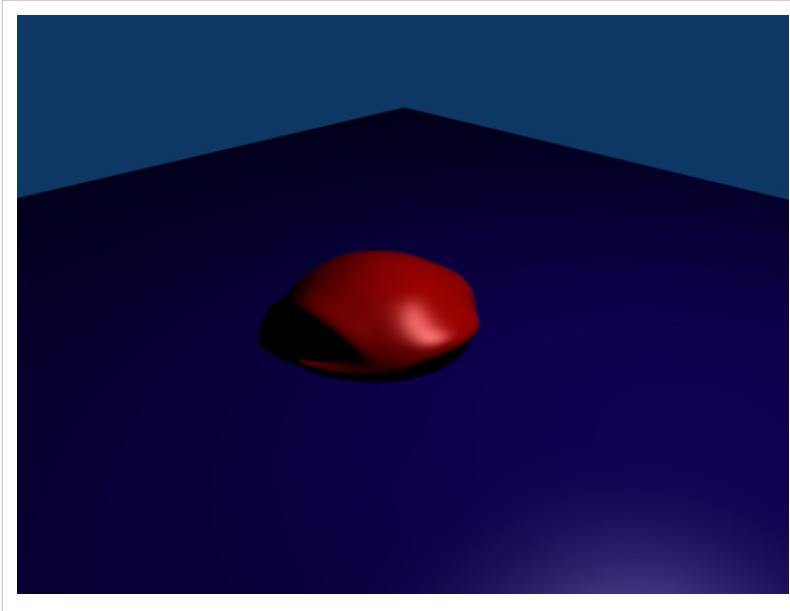


Now the ball collapses into a strange quivering wreck after impact. To fix this, you need to turn on the stiff quad button, but set the edge stiffness down a bit, so its more bouncy. You can use the bake function to solidify the settings see below

*Note from Noob: You might have to turn up the **Rigidity Level** to 0.100 (in the Soft Body Tab) as well in order to prevent the object from collapsing. (I used a subdivided cube as Object)*

*What I recommend to do before rendering as animation: In the **Bake Settings** (Soft Body Tab) Set **Interval** to 2 or 1, so the object will not start deforming too early before impact. This will slow down the bake process - just slightly - but make the object bounce more dynamically. Then bake again.*

*Noob note: In blender 2.46 you have to adjust the value of Be in the soft body tab, I changed it to 0.4*



another thing you could do is animate the scene. Just put up animation video and set frames at about 50. Go to object mode, select sphere and press i key, and select LocRocScale. =)

## Explanation of Settings

I invite you to correct and expand these definitions:

### Softbody

- **Friction:** creates a resistance to movement of the whole object, like being submerged in a viscous fluid
- **Grav:** the rate of velocity change

due to gravity. Results in a constant -z force.

- **Mass:** (Force = mass × acceleration) affects everything by making the object heavier.
- **Speed:** tweaks the simulation to *run* faster or slower.
- **ErrorLimit:** raise it and the simulation will solve faster but strange things might happen. Save frequently, as blender might go nuts with this or any physic simulation (but less so after 2.4)
- **Goal:** makes the object try to return to its original position, useful at times, in the tutorial you could turn gravity off and key the ball falling and use this to keep it a ball.
- **use Edges:** uses the edges as a means of resistance to movement for the object. Helps to keep it looking possible

# Blender 3D: Noob to Pro/Simple Cloth Animation

In this tutorial, we will be making a simple skirt. This tutorial does not assume that you have any Blender or other 3D program experience. **Note:** *In the new version of Blender, there is specific cloth functionality which works better than this.*

## Making the Skirt Mesh

1. Open Blender and delete the default cube, if you aren't looking down on the scene, press **NUM7**.
2. **SPACE** → **Add** → **Mesh** → **Circle**
3. You should get a prompt for the number of vertices. Use about 20 and hit **OK**.
4. Once the circle is added, you should be in 'Edit Mode' (You can tell that you're in edit mode if you can see the individual points of the circle and they are all highlighted in yellow). If you're not in 'Edit Mode', press **TAB**.
5. Press **SKEY** and resize (in 3D terms, scale) the circle to however big you want your skirt to be.
6. Tap the **AKEY** key twice to select all vertices.
7. Press **NUM1** to look at the skirt from the front. If you can't see it, that's because all the vertices are in the same plane, this is fine.
8. Press the **EKEY** then select **Only Edges**, wherever you move your mouse, you'll see a circle follow it. Pull the mouse downward and left click.

You should now have a hollow cylinder

1. In the bottom pane under 'Mesh Tools', click 'subdivide'. Then click fractal, set to '10' and press ok. Finally, click 'Subdivide' again. The subdivide function adds vertices. The fractal function adds vertices with a little bit of randomness. All we really did was add some interest to the skirt.

## Creating Vertex Group

The vertex group will tell us which vertices will be affected by softbody and which ones won't. The non-affected vertices are given a weight of 1.0. To understand the concept of vertex groups, [Click Here](#), however, it's not necessary for this tutorial.

1. Deselect all vertices by pressing **AKEY**.
2. You're still looking from the front, right? If not, then press 1 on the num pad. Press **z** to enter wireframe mode. This allows you to select vertices that you couldn't see before.
3. Tap the **b** key and the cursor turns into a plus. Now, if you click and drag, you can select all vertices within the box, it's much faster. Anyways, select all the vertices on the top layer.
4. Look in the bottom pane under 'Link and Materials'. You will see that there are two columns of buttons directly underneath the 'Vertex Groups' label. Click the 'New' button. Two boxes appear directly underneath the 'Vertex Groups' label. Change the new vertex group name to something meaningful, like 'softbody'. Change the 'Weight:' to 1, by clicking on the left side and sliding to the right. Now press the 'Assign' button. This assigns the vertices that you have selected (the inner-most circle) to this vertex group.

-Note: The higher the weight, the closer that vertex will stay to its position in edit mode. By making it '1', we essentially

"pinned" the top vertices where they are, but will let the other ones hang loosely.

## Making the Skirt a Soft Body

1. Enter object mode and select the skirt (right-click it)
2. In the buttons window, typically at the bottom of your screen, click the objects button. It looks like three arrows.
3. A new set of buttons appear, click the one that says **physics** when you hover over it. It is a black arrow reverbed on a line (right of the 3 arrows), just at left of the particle button (red and yellow angle with yellow squares coming out of it).
4. At this point there should be three little windows on the bottom. One deals with deflection, one with particles and one with softbody. Click **enable softbody**.
5. Set the **gravity** to 9.81. This is like real earth gravity.
  1. leave mass and friction, but increase speed to about 3. This allows you to see the movement in fewer frames.
  2. for goal, click **enable vertex groups** (right next to the "use Goal" button is a small button indicating up/down. Click it to select your Vertex-group) and assign the one that you made earlier. What this does, is it tells the vertices that were assigned a 1, to stay where they are, and the ones with a zero that they don't have to stay. Of course, you can assign a value of .5, then the vertex will go somewhere between where it began and where gravity wants it to go.
6. Test it out, by placing your cursor over the biggest 3d window and pressing **ALT A**
  1. To exit this animation, press **ESC**
  2. If it runs too slow, exit the animation, press **Z** To enter wireframe mode, then try again.

Noob Note: *Using Version 2.45, I had to create a second vertex group for the bottom side vertices and set their weight to 0. Then, in the Soft Body tab, the 2nd vertex group (the one with weight=0) had to be selected (thus the 1st group was deselected) for the object to move.*

## Extra Practice

This YouTube tutorial might also help: <http://au.youtube.com/watch?v=mgYhZ3hWwTQ> happy animating!

---

# Blender 3D: Noob to Pro/Soft Body with wind

This is the **Blender wind and soft body tutorial**. This tutorial will try and help your knowledge of using Soft Body and the Fields and Deflection panels in Blender 3D. (For best results, I recommend you use Blender version 2.43 or higher). Don't forget to save your work at various points throughout the tutorial.

## Setting up scene

### The plane

Delete the default cube [Delete key, yes] and enter Front view mode [Num. Pad 1]. Add a plane, scale 3 times [S key, 3, Enter] and sub-divide 4 times until you get something like the shape below. If your computer can handle more, and you want more, sub-divide as many times as you like, but if your PC is struggling with this, undo once or twice.

### The wind items

TAB out of edit mode (if you haven't done so already), enter side view [Num Pad 3] and add an empty. Clear the rotation [ALT + R] and rotate it 90° [R, 90]. Place it about -5 from the centre [G, y, -5] and make sure its still in the middle of the X axis. Your scene should look like the picture below:

For the second wind item, add an empty like before and clear its rotation, but DON'T rotate it any more. Move it along the Y axis by 4 places [G, Y, 4] and -10 by the Z [G, Z, -10]. And now your scene should look something like this (with all items selected).

And that's all your items you'll need for this. Onto the harder part.

## Designating each object's job

### The plane

Now, the plane will become a soft body, so enter the Object Panel [F7], click on the Physics Button (beside the three arrows, see second picture below) and click on the soft body button. You will get a load of boxes, but we only need to look at the Grav, Goal, and the bottom parts:

Change the values of these as shown above. Change the Aero part to 732 (if you go up to 1000, it changes the final output by a bit).

Now, click on the Soft Body Collision tab and click on the Self Collision and Deflection buttons. You don't need to change any of the numbers here. The buttons window should look something like this:

Now we will move on to the Empty items which will be the wind.

## Creating the wind

The two empty's we created earlier will be the items that will act like wind, and when you render at the end, they will be invisible, just like the wind. Anyway, select one of the empty's and enter the Physics panel. Click on the box under the Fields and select "Wind". Change the Strength to 20, click on Use MaxDist and change MaxDist to 2.5.

Do the same for the other empty. Your scene should look something like this:

## The movement

This is probably the most complicated part, but shouldn't take long.

### The Empty on the left

In side view [Num pad 3], select the empty on the left and do the following: 1. Press [I key] and then select LocRotScale from the list it gives you. This inserts a key frame, saying that on this frame, keep the Location, Rotation and scale the same throughout the rest of the movie, unless it comes across another key frame, which might tell it to move or stay the same. Now, press the Up Arrow Key until you get to frame 31 (about 3 times). If you don't know where the box is that tells you what frame your on, check the following picture:

Number 1 shows the current frame number and the name of the selected object while number 2 shows the current frame and allows you to change the frame with the arrows on either side.

### Back to the movement

On frame 31, select the empty on the left (the one selected in the picture above) and press [G key]. Now press [Y key] and press "4", then [Enter key]. Now press [I key] and select LocRotScale again. Now, if you press and hold the Left Arrow Key, you should see the Empty move back to its starting place. Now, to check the curve guide (called the "IPO Curve Editor"), right-click on the top panel (number 1 in picture below), then select "Split Area" and click when the line is a bit out from the side (number 2 in picture below).

Now, click on the box (shown below) and select "IPO Curve Editor" from the menu.

This window just shows where the key frames you entered are, and how much the shape changes, moves or rotates. Now, go back to frame 1 and press [ALT]+[A], which plays the animation. You should see the "wind" hit the sheet and the sheet will blow away. Pretty nice, huh? Now, it would be fine like that (I'll show you how to get rid of the blockiness in the last part), but say you want to hit the sheet again and send it upwards? We will use the second empty for this.

Select the second empty (the one on the bottom). Insert a key frame ([I key], LocRotScale) on frame 1 and frame 50. Go to frame 60, grab the empty and move it up 8 places ([G key], [Z key], 8). Insert a key frame ([I key], LocRotScale) and then go back to frame 1. You have completed the basics, and the most of this tutorial. Now, press [ALT]+[A key] and watch your animation.

## Finishing touches

### Blockiness

Now, to sort out the blockiness, go to the Editing buttons panel [F9] and select the "Set Smooth" button.

Now if you preview again ([ALT]+[A]) you should see the plane is smoother, but might be a bit unrealistic (i.e. not acting like real cloth). Now, that can't be helped, the only way to get rid of it is to subdivide the plane several more times, but that will put a LOT of strain on your computer, so I wouldn't recommend doing it unless your computer can handle it.

## Background

First of all, got to the Scene panel [F10] and over at the Format panel, change SizeX to 400 and SizeY to 300. Now, to change the background colour when you render, go to the shading panel [F5] and select the World buttons. Now, in the World panel change HoR to 0, HoG to 0.80 and HoB to 1.0.

All that does is just changes the background colour to a brighter, nicer shade of blue. Now, I'm just getting a bit lazy, so I'll just explain this part with a picture: (don't forget to select the plane).

Change the values and press the buttons shown in the picture. This will change the colour of the plane to white.

The only problem now is the light. Press [F12] and you should see what I mean. There's no light shining on the plane where the camera is, so its just shaded dark. To fix that, select the light and go to the shading panel [F5]. You now have options for the light. Click on the "Sun" button and press [ALT]+[R] to clear rotation. Now, in side view [Num pad 3] press [R] and then type -45. Enter top view [Num pad 7] and press [R] and then -45. This now shines the light in the direction of the plane, but, if like mine, the light is above the plane in top-view, press [G], [Y] and then -5. The line coming out of the light source should be roughly pointing at the centre of the plane. Here's what your window should look like:

The pink line is the light. This was taken in camera view [Num pad 0]. Render the frame and you should see what the final product looks like. Move to any frame you want and press [F12] and it will show the rendered image for that frame. Now, rendering the entire animation might, in fact, will take a long time unless you have a completely new PC, are using a big server like DreamWorks (one of the big PC's, used for rendering images faster) or have a quicker renderer, I don't use any external renderers, and am stuck using a PC that is about 2 years old (in other words, full of junk). It took my PC less than 5 minutes to render 100 frames, and the last 25 were just blank (as the plane just flew above the camera). To change the amount of frames to render, and to render the full animation, here's what you press:

The box that says End:100 lets you select what frame you finish rendering the animation on (NOTE: the Sta:1 button lets you select what frame to start rendering the animation). When you press the ANIM button, it starts rendering the animation. The Box covered in blue, that says Jpeg, lets you change the format of the image in the end.

To export the animation as an AVI format, select the AVI button shown below, then press OK. Now, you need to Animate the sequence again, so press the ANIM button again.

The finished .AVI will be saved in a folder with the rest of your rendered images. It should be saved in a folder called tmp, which (on a Microsoft computer) is in My Computer/tmp. The name of the file will be whatever your start and end frames are, so in this movie, mine is called 0001\_0100.avi. You can now do whatever you want with your movie now, and hopefully you have learned a trick or two that will be helpful later on in your Blender career.

## Final result

Here is an animated .gif of my final result: (imported into Macromedia Flash and exported as an animated .gif) You are required to be an auto confirmed user or uploader.

--Eb264 /

---

# Blender 3D: Noob to Pro/Your First Test

The Blender Game Engine is an interesting feature of Blender. It is basically a 3D environment in which 3D objects move around and react to each other upon contact. One common application is to recreate 3D architectural tours.

In this tutorial, you will learn the basics of object collision within the Blender Game Engine (BGE). From Blender games to use in animations, the bullet physics engine offers a massive number of possibilities. The tutorials found within this wikibook on the subject of the BGE are generally focused on game creation, but the concepts taught within can be applied to a multitude of situations.

As a start, we will teach you to make a ball roll realistically down the hill using Blender's game engine.

## Adding the Hill

First, make a plane, then switch to Edit mode (**TAB**), and multi-subdivide it with 2 cuts (**WKEY** → *Subdivide Multi* → 2). Next, enter face select mode (**CTRL+TAB** → *Faces*) to drag the center face up, in order to form a rough outline of a hill. Add a subsurf modifier (in the edit buttons) to about 3, then apply. You should now have a serviceable, but small, hill. Scale the hill up (**SKEY**) by about 10 times, and we're ready to add the ball.

## Creating the Ball

Now, add an icosphere (**SPACE** → *Add* → *Icosphere*) and relocate it to be just above the hilltop (**GKEY** or use the translate widget by pressing **CTR+ALT+G**). Let's change the color of the sphere so we can differentiate it from the plane. Go to the material buttons (with the sphere selected) and click on the white panel beside the COL value. In the color selection wigeet that appears, change its color to a bright red.

Next we need to make the Physics engine iterate over it. With the sphere selected, go to the logic buttons (the little purple Pacman-icon). You will see a button in the top left that says *Actor*. Press it. Now select from the selection box beside of the "Actor" button *Rigid Body*. This makes the ball roll, instead of staying completely upright the entire time. You will see a bunch of settings available now. Change *Radius* to 2. This changes size the physics engine *thinks* the ball is. You notice a dotted circle around the object; this is the boundary. Now change the *Radius* back to 1. You now have your first Blender game ready to go.

## Testing your game

Now the time has come for the first test of our game.

1. Add a light source well above the maze (**SHIFT+AKEY** → *Lamp* → *Lamp*). Align in front view (**NUM1**)
2. Press **NUM5** to switch to Perspective mode, which gives a realistic view, rather than a view in which objects stay the same size with distance (be sure to switch back to Orthographic view when you are editing again using **NUM5**)
3. Enter textured mode (**ALT+ZKEY** -- press **ZKEY** to switch back to solid view mode)
4. Switch into side view (**NUM3**) and press **NUM8** several times to get a good perspective on the ball.
5. Press **PKEY** to play the game (Make sure you are in Object mode (**TAB**))
6. Press **P** to start testing the game. You should see the red ball drop onto the hill.
7. Press **ESC** to quit testing the game

## Conclusion

With the knowledge acquired in this tutorial, there are many things you could accomplish within the Blender Game Engine, although the majority of them would require more knowledge. So read on, and work your way through the multitudinous seas of tutorials (That is, two).

## Extra Tutorials

Making a Basic Game: Link <sup>[1]</sup>, The State Actuator: Link <sup>[2]</sup>, Blender Game Engine Mouse Follow: Link <sup>[3]</sup>, Blender Bullet Physics: Link <sup>[4]</sup>, Domino Game: Link <sup>[5]</sup>, Rag Doll: Link <sup>[6]</sup>

## References

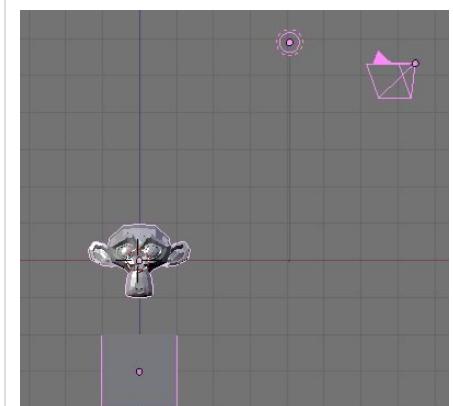
- [1] [http://www.youtube.com/watch?v=WGycLDs\\_dLg](http://www.youtube.com/watch?v=WGycLDs_dLg)
- [2] <http://forum.nystic.com/viewtopic.php?f=9&t=4633>
- [3] <http://forum.nystic.com/viewtopic.php?f=9&t=4743>
- [4] <http://forum.nystic.com/viewtopic.php?f=9&t=4752>
- [5] <http://forum.nystic.com/viewtopic.php?f=9&t=4820>
- [6] <http://forum.nystic.com/viewtopic.php?f=9&t=4987>

# Blender 3D: Noob to Pro/Platformer: Creation and Controls

This tutorial will teach you the basics of the Blender Game Engine, and how to create a platform game, which is a game where you overcome in-game obstacles by controlling the character's motions. (e.g. Super Mario)

## Set Up

To start, switch to the frontal view **NUM1** and move the starting cube to -3.0 along the Z axis, or three spaces down. If your default settings do not have you starting with a cube, create one now. Next, hit **SPACE** → **Add** → **Mesh** → **Monkey** to create Suzanne the Monkey. Any mesh will do, but it's the easiest of the basic meshes to intuitively determine local direction from. Next, go to the logic window (with Suzanne still selected) and toggle on "Actor" and then "Dynamic". Hit **PKEY** to test. Suzanne should fall, then stop upon hitting the cube. Press **ESC** to exit.



How it should look so far.

## Controls

Now we will add the controls. First, forward:

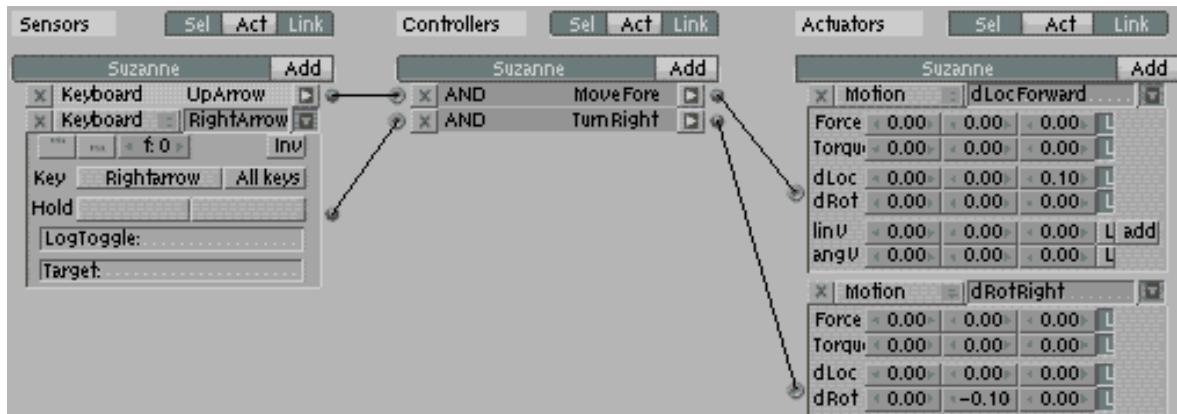
1. Create a Keyboard Sensor, AND Controller, and Motion Actuator. Connect them by dragging lines between the dots next to their names.
2. Select the empty box next to the word "Key" on the Keyboard Sensor and hit the up arrow key on your keyboard.
3. Set the dLoc on the Motion Actuator to 0.10 Z and toggle Local Transformation (may appear as simply an L next to the dLoc row).

Next, backward. Repeat the process used for forward, but set the dLoc to -0.10 Z and the key to the down arrow. You can repeat this for left and right if you so choose to have a linear control scheme. We will not be using such a scheme for this tutorial, but you should still be able to follow along just fine. Instead, the character will rotate left and right and move in the new forward and backward.

To rotate right:

1. As before, create a Keyboard Sensor, AND Controller, and Motion Actuator, then connect them.
2. Set the key to the right arrow and set the dRot to -0.10 Y.

Repeat for left, with the left arrow and 0.10 Y.



Finally, the ubiquitous, and some say defining, Platformer move: the jump. Repeat the previous processes, with the space key as the trigger and the linV of the motion effect as 7.50 Z (not local). This doesn't actually move the character in a certain direction, but sets its velocity. If you've done programming for games before, this will probably be familiar to you. Each frame, the engine adds the velocity of the object to its current position. The gravity portion of the engine subtracts from the upward velocity each time. When it reaches a negative, it's adding a negative number to its altitude, ergo subtracting from it, ergo causing it to fall. This is why using linV to move the character does not cause it to immediately relocate to the target position.

# Blender 3D: Noob to Pro/An aMAZEing game engine tutorial

This tutorial is intended as an intermediate introduction to the Blender game engine, in the form of a maze game, and is the sequel to Platformer: Creation and Controls. It will require a familiarity with the Blender UI, simple commands (such as **AKEY** to select) and basic modeling skills. The game we will create within Blender will have the following features:

1. a protagonist controlled by means of the WASD keys
2. a maze surface without walls
3. death on falling off the maze
4. multiple levels
5. dynamic obstacles
6. a goal within the maze that will transfer you to the next level

This tutorial was written for version 2.44 and has been tested for all later versions (as of April, 2008)

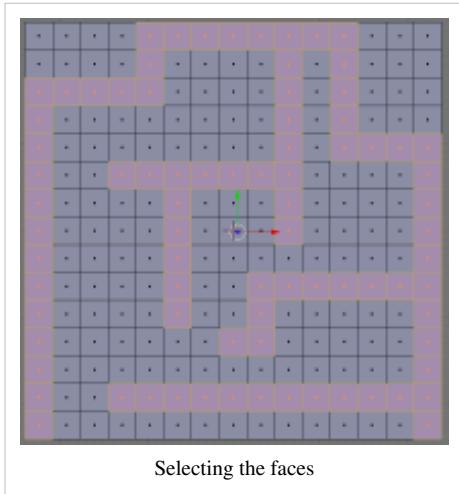
## Maze Surface

Now we'll make the maze. When you are creating the path of the maze surface keep in mind that you need a start and a goal.

1. Clear the scene; Select all (**AKEY**), delete selected items (**XKEY** → **ENTER**)
2. Go into top view (**NUM7**)
3. Add a grid (**SPACE** → **Add** → **Mesh** → *Grid*) with X and Y resolution of 16
4. Scale grid by 24 times (**SKEY**, type 24, **ENTER**)
5. Enter edit mode with the grid selected (**TAB**)
6. Go into face select mode (**CTRL+TAB** → *Faces*)
7. Start selecting faces so as to make a two dimensional maze (use **BKEY** to draw selection boxes)
8. With the faces selected, duplicate those faces (**SHIFT+D**, **RMB**)
9. Move those faces away from the original grid (**GKEY** or the movement widget **CTR+ALT+GKEY**)

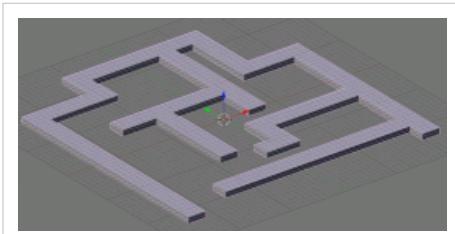
10. Delete original grid
11. Center your maze using **GKEY** or by pressing the **Center** button in the **Mesh** panel of the Editing buttons (**F9**)

You should now have a two dimensional maze. In the game engine, we only see one side of every face, so this maze will not appear to be there when seen from below, and will be obviously two dimensional when seen from the surface of the maze itself. This is undesirable so we will make our maze 3 dimensional by extruding the maze surface down.



1. Go into side view (**NUM3**)
2. Select all the faces (**AKEY**)
3. Extrude down, -1 unit (**EKEY** → *Region*), hold **CTR** to constrain to 1 unit increments, or manually type in the value -1, like you scaled the grid in the beginning.
4. **LMB** to accept the operation, or **RMB** to cancel the move of the faces. If you cancel it, the faces will have extruded, but will simply not have moved. You can undo the operation (**CTRL+Z**), or delete the vertices/faces you have extruded (**WKEY** → *Remove Doubles*).

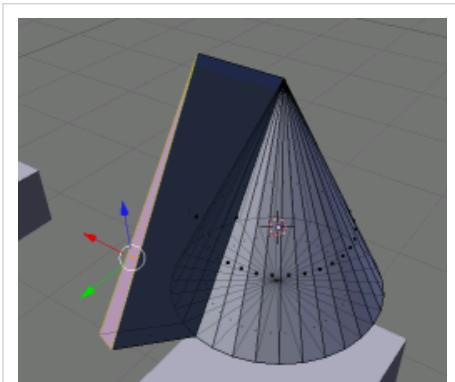
Your maze is now complete, only lacking someone trapped in it...



The extruded maze

## Character

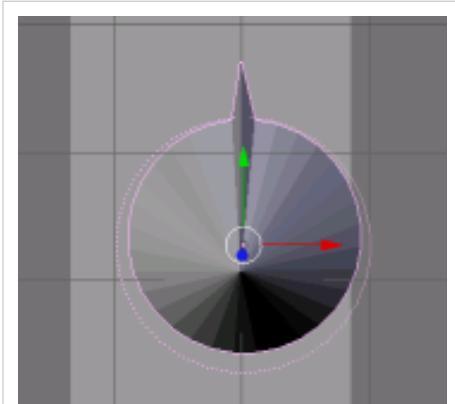
We will create a very basic character sufficient to illustrate the concepts meant to be conveyed in this tutorial. You can make a character as complex or basic as you desire, but it is important to always have an indicator of direction on your protagonist, otherwise the player may become confused. So, we will use a primitive with a 'nose' (i.e. a face or vertex extruded away from the object center in order to provide a point of rotational reference). I'll use a cone for this tutorial, simply because they don't roll very easily.



Extruding a nose

1. Enter top view (**NUM7**)
2. Add the cone (**SPACE** → *Mesh* → *Cone*, use default values)
3. Select cone and enter Edit mode (**TAB**)
4. Enter face select mode (**CTRL+TAB** → *Faces*)
5. Select a face on the side of the cone
6. Extrude it about 1/4 of a unit or less (**EKEY**)
7. Merge the extruded face to make the nose pointy (**WKEY** → *Merge* → *At Center*)
8. Enter object mode (**TAB**)
9. Rotate the cone so the nose is aligned with the positive Y axis (**RKEY**, align with the green arrow)

You've got your character now (Not exactly Pixar-like, but it will do for now). Now we just need to give it motivation.



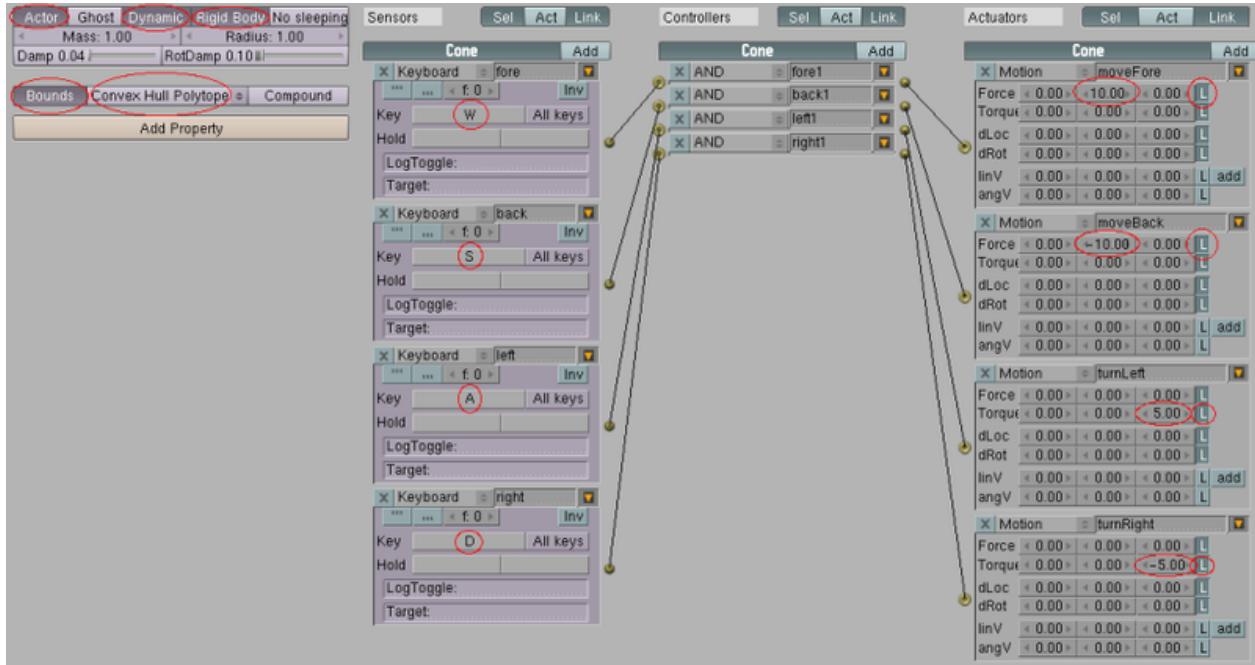
Rotate to the direction character will move

Note: From points 8 to 9 it would seem that the cone should rotated so that the "nose" is aligned with the Y-axis in object mode, but after doing this I encountered the cone still moved in the original Y-axis direction (before rotation). To remedy this I did the alignment of the "nose" to the Y-axis in edit mode instead, this seems to have solved the problem.

## Motion

We'll now make the character move, via the use of the WASD keys. In order to do this, game logic needs to be associated with the WASD keys. Copy the configuration seen in the screenshot, taking care to include the convex hull polytope collision in the top left as well as depressing the 'L' buttons, which make the force and torque local to that object rather than global. If the force was global, the protagonist would continue moving in a straight line, even when turning.

1. Go to the Logic buttons (The purple pac-man icon or F4)
2. Add four Sensors, Controllers and Actuators.
3. Maximize the logic window (**CTRL+UPKEY**)
4. Copy the configuration seen in the screenshot below.



(Note: I made the simulation using Actor->Dynamic since there was no way to use both dynamic and rigid body in 2.49b. After some testing it appears to work correctly.)

### dLoc/dRot vs Force/Torque

The character logic in this tutorial is very similar to that of the previous platformer logic, but it differs in that it uses force and torque rather than dLoc and dRoc, respectively. The difference between these two is that force and torque move and rotate the object within the bullet physics engine, applying friction and collision to the process, whereas dLoc and dRot move and rotate the object without regard for any other 'actors' (objects factored in to the physics engine's calculations). Although dLoc and dRot would serve in most maze games, many other instances demand force and torque. However, most of these instances occur in more complicated games with a multitude of interactions with dynamic objects. Later in this tutorial we will add dynamic objects to our maze, and the reasons for using force will become clear.

## Testing

Now the time has come for the first test of our game.

1. Add a light source well above the maze (**SPACE** → *Lamp* → *Lamp*), align in front view (**NUM1**)
2. Press **NUM5** to enter perspective viewmode, which gives a realistic view, rather than a view in which objects stay the same size with distance. Be sure to switch back to orthographic view when you are editing (**NUM5**)
3. Enter textured mode (**ALT+ZKEY**) -- press **ZKEY** to switch back to solid view mode
4. Switch into top view (**NUM7**) or camera view once we install the camera (**NUM0**)
5. **PKEY** to play the game
6. **ESC** to escape testing the game

If you receive a "No camera" error, disregard it. We'll be adding a camera later.

If your character doesn't move at all make sure you drew the connecting 'wires' between the actuators in the logic panel, as well as setting the "Actor" "Dynamic" and "Rigid Body" buttons to True.

You should now have a nosy cone that speeds around the maze.



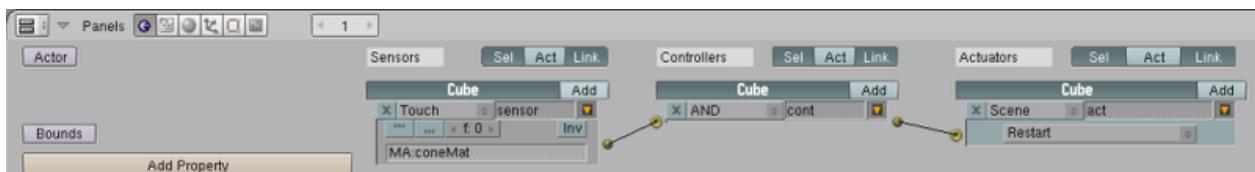
## Falling

We will now implement the "death on falling off maze" feature. In order to accomplish this, we'll add something that allows us to check whether the cone has fallen off the maze or not. The simplest way to do this is with a 'skydome' object the entire maze is within, and although there are a number of more elegant ways to accomplish the goal, we will use the skydome method for its simplicity to replicate. From now on we'll need to test the maze with both layers one and two enabled (**SHIFT+2KEY**)

1. Add a material to your cone called *coneMat* (**F5**, press "Add New", Links and Pipeline, Link to Object, *MA:coneMat*).
2. Add a new cube
3. resize it to gargantuan proportions (**SKEY**, just large enough to encompass the maze, and all within it)
4. Move it to layer 2 (**MKEY**, **2KEY**, **Enter**)

Now that you have a massive cube, go to the logic panel and add one Sensor, Controller, Actuator set, and set them as seen in the screenshot.

1. Sensor: Touch, *coneMat*
2. Controller: And
3. Actuator: Scene, restart



Enable the actor button and press "ghost" this will let the dynamic obstacles, which are added later on, fall through and not get stuck there!

If you test your game now, you should see the game be restarted when you fall off the maze. As we discussed earlier, faces are one-sided in the game engine, so you should not be able to see the cube.

(*Newbie Question 1:* When I did this, my cone just kept on falling. When I moved the cube back to layer 1, it worked fine. Am I missing something?) (newb note: ok i had the same problem everytime i was just in layer 1 but when i was just in 1 then i just kept falling)

(*Answer:* The tutorial says to create the cube in layer 2, although the layer shouldn't matter. Make sure you test the game with both layers one **and** two selected. Also double check the logic for the 'skycube')

(*Answer2:* I had the same problem. I fixed this by replacing the cube with a grid, scaled and placed under the maze. The grid had the touch logic mentioned above.)

(*Answer3:* Same here. I thought I'd remove "coneMat" from the material reference, but that wouldn't do if the bricks fell down and touched the cube, it'd just restart the level... the solution: create a property in the cone, a bool value named "isCone" (=true), then in the cube add a sensor 'collision' that checks for the object's property 'isCone' and keep the rest as shown.)

(*Question 2:* How do I make the box translucent? Right now all I see, also when testing, is the skybox. Also, when I put the box underneath the maze, it works fine, but then none of the objects have shading, while when not showing layer 2 everything looks shaded. Can anybody give a solution? This is in 2.48a)

(*Answer:* To make something transparent- press **F5**; press the "Material Buttons" button; under the Links and Pipeline tab press Halo and ZTransp, unpress Shadbuf; under Material make A 0.000.)

(*Answer 2:* My solution was to press the Invisible button (Logic section, next to Ghost in 2.49))

## Camera

**SPACE** → *Camera* near the protagonist cone. There are two easy ways to do the camera: logic camera and child camera. In this case you might be better off with the logic camera, but it really is a matter of personal preference.

- **Logic Camera:** Add to the camera's logic panel
  1. Sensor: Always
  2. Controller: And
  3. Actuator: Camera, to object "Cone" (or whatever your protagonist is called--look under the object panel when you have the protagonist selected) height 5, min 5, max 10.
- **Child Camera:** Select the camera, then select the protagonist (order is important) then **CTRL+PKEY** to parent the cone to the camera. Align the camera in a view that you like, and test the game.
- **First Person Camera:** Well, this one doesn't really count as another method, but if you're willing to redo your motion to just move the camera instead of the character, (just do everything you did for the character for the camera) you can run around in the first person. Or you can just put the camera in the cone because one of the sides always are invisible (look in texture mode) and parent it to the cone. REMEMBER: To access the ingame camera during gameplay, before you start, hit ctrl 0 or you can go to (view > camera)

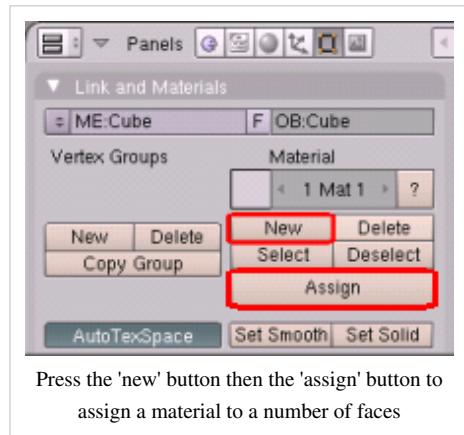
## Beautification

Blender is a great 3D modeling program, and with such a vast number of tools at your disposal, you should be able to make your game look better than this:

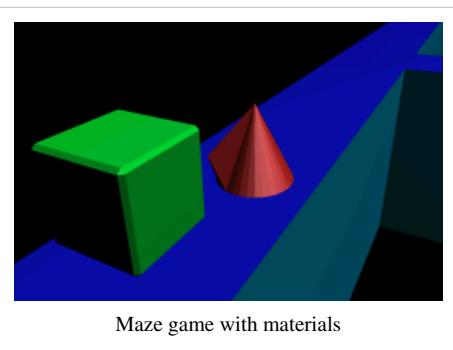


We'll add one or two textures to each of our objects.

1. Select the object you want to texture
2. Go into edit mode if you want your object to have more than one color
3. Select all the faces you are going to assign to a color (**CTRL+TAB** → *Faces*)
4. Go to the edit panel (**F9**) on the buttons window
5. Near the left there should be a panel that says "Link and Materials". Click on the buttons highlighted in the screenshot
6. Adjust the color of that material by going to the material buttons and changing the "Col" (color) value to something other than white

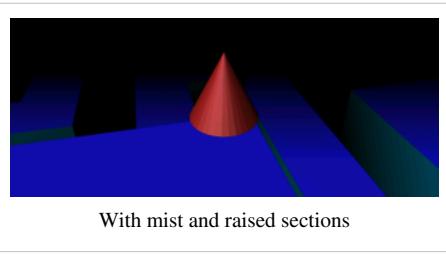


It looks better now, but when the camera is nearly horizontal, you can see where the camera stops rendering the maze. You can fix this by selecting the camera and, in its edit buttons, changing its clipping range... but there is a better way. Mist.

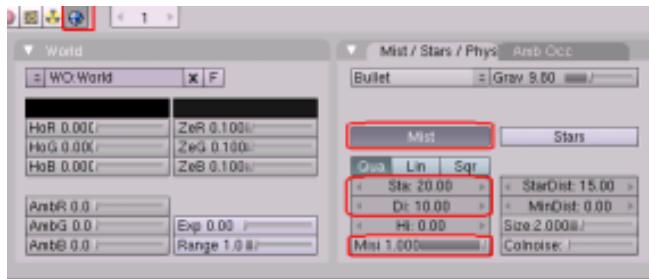


Mist obscures everything a certain distance away, is the same color as the world texture, and can be handy. We'll add some mist to our game so the player can't see too far ahead of him, thus making the game too easy.

We'll make the mist quite close, although you can vary the distance. Go to the Material button, and to the World sub-buttons, then copy the settings in the screenshot. **When you test this out, make sure you are in texture mode to see the mist.**



With mist and raised sections



(Question: I copied the screenshot but i can't get the mist to work. What's going on?)

(Answer: Change Start to a smaller value like 15.)

## Levels

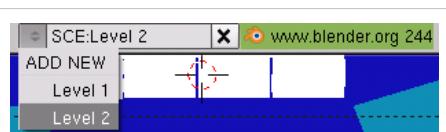
Now all that remains is to create the next level with some obstacles.

1. Add a cube at the end of the maze, just above the floor level so the cone can touch it (**SPACE → Add → Mesh → Cube**)
2. Press the Scene dropdown menu and select "ADD NEW" → *Full copy*
3. You can now go back to the Maze Surface section to make a new maze mesh
4. Go back to level one via the Scene menu
5. Select the End of level cube
6. Change its logic panel to resemble this:
  - Sensor: Touch, coneMat
  - Controller: And
  - Actuator: Scene, Set Scene, Scene.001 (or whatever you named your level 2)

(noob: When I went into level 2 my player name changed to coneMat.001 and I had to change all my goals and the giant restart cube assignments.) Answer: Rename the coneMat.001 to whatever you had the material named for the first scene - this fixed it for me.

(noob: In my level 2 my cone hits the cube and nothing happens!) Answer: Make sure that the Sensors, Controllers and Actuators tabs under the Logic panel (F4) are connected with wires.

To make it so multiple objects have to be touched (or acquired by the character, as it may be), give each object the logic settings described above and change the actuator type to Message with the subject "goalget". If you'd like the object to disappear after it is collected, add an Edit Object actuator to the object of the type "End Object", and link it to the existing AND controller.



The scene button is on the top panel

Next, select the massive cube that restarts the level upon contact with the character. If you chose not to include one such cube, create an empty at the start of the level (so you can easily find it again later). Give the cube or empty the following logical operators:

- Sensor: Message, goalget

- Controller: And
- Actuator: Property, Add, wincon, -1
- Sensor: Property, Equal, wincon, 0
- Controller: And
- Actuator: Scene, Set Scene, Scene.001 (or whatever you named your level 2)

Then click "Add Property" and give it the Type Int, the Name wincon, and a value equal to the number of objects needed to be collected to complete the level (if all of them, make it equal to the number contained in the level). This can be expanded to include other multiple win conditions besides the collection of an item by making the completion of each condition subtract from "wincon".

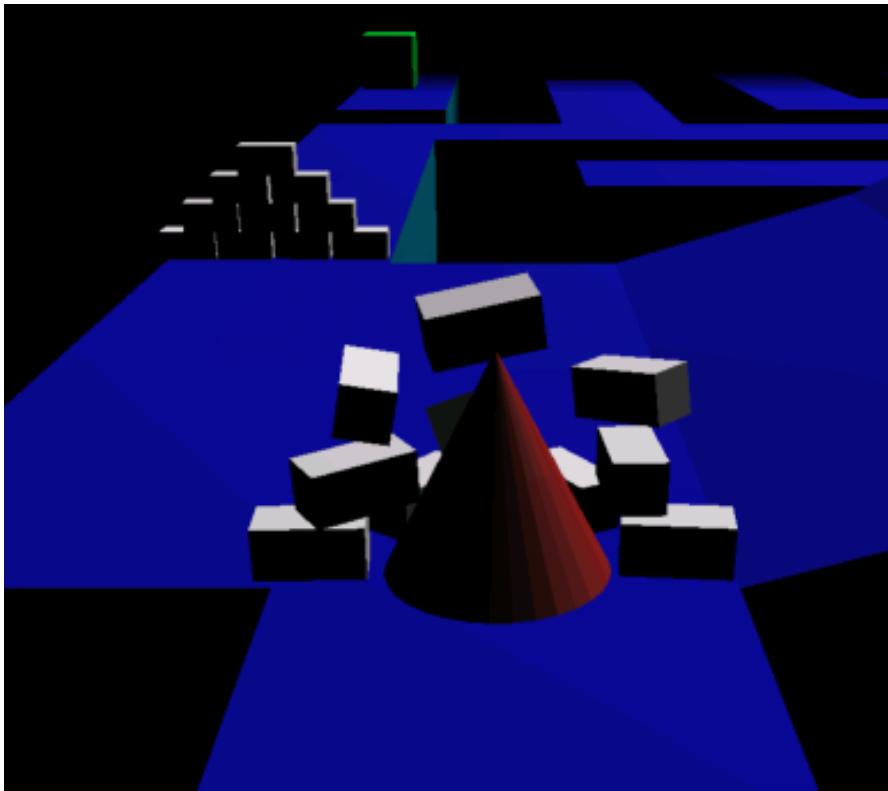
You now have a multilevel maze game. Feel free to elaborate on the gameplay mechanics and models so they look better.

## Dynamic obstacles

This part of the tutorial will assume increased skills and knowledge of the Blender interface and basic commands. As such, this section will be written less specifically, and the shortcuts for all basic commands will not be covered here. From this point on, this tutorial is of an intermediate level. If you cannot follow this part of the tutorial, you can learn the basic functions used below in an earlier tutorial <sup>[1]</sup>.

We will now add some dynamic obstacles. In this case, they will constitute a 'brick' wall that can be broken through by your character.

1. Add a cube
2. Scale it (in edit mode) to the dimensions 1,0,5,0,5
3. Bevel it (WKEY) recursion 1, size 0.03 (press space to enter value manually)
4. In logic panel, set it to Actor, Dynamic, Rigid body, Mass 0.5
5. Go right below that to click Bounds and leave it on "Box". That makes Blender realize that what you added is a box, and should act like it (instead of rolling, it will slide now)
6. Move it to an intersection in your maze (GKEY) and duplicate it until you have a brick wall (SHIFT DKEY, LMB)



You now have a barrier that your character can break through.

By using this

## Conclusion

You have now learned enough of the Blender Game Engine (BGE) to create your own game. If you wish to go very far into Blender games, I recommend you learn Python, as trying to make a full game within the graphical interface for the BGE is like trying to dig a grave with the blunt half of a toothpick. Blindfolded. And the toothpick is glued to your forehead. However, you are now well on your way to becoming a game-maker!

If you have any questions, concerns, or 'noob notes', please post them in the discussion page for this tutorial, and not in the tutorial itself.

## References

- [1] [http://en.wikibooks.org/wiki/Blender\\_3D:\\_Noob\\_to\\_Pro/Learn\\_to\\_Model](http://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Learn_to_Model)

# Blender 3D: Noob to Pro/Platformer: Physics Fixes

This tutorial is intended as an improvement on its prequel, Platformer: Creation and Controls, and will require files created through that tutorial. It is also recommended that you read An aMAZEing game engine tutorial as well before beginning this tutorial. It will require a familiarity with the Blender UI, simple commands (such as AKEY to select) and basic modeling skills. This tutorial will introduce the following features and improvements:

1. only letting the character jump while touching the ground
2. stop the character from bouncing greatly upon hitting an object
3. where to adjust material friction

This tutorial was written for version 2.45.

## Creating Your Hit Test Object

Copy the cube with **SHIFT-D**. Go into edit mode, select the bottom four vertices of the cube and delete them. Go back to object mode. Go to the editing panel in the buttons window. In the Link and Materials section, change the ME: value to "Square" and the OB: value to "FloorHit". In the Mesh section, there will be three buttons with the word "center" in them; Center, Center New, and Center Cursor. Press Center New. This will change the origin of the "Square" mesh to the center of all of its vertices, instead of the center of the cube you made it from. Scale it to 0.99% of the original and move it up 0.01 along Z. Now go to the shading panel and make sure you're in the Material buttons subpanel. Find Links and Pipeline, and make sure "ME is selected. Click the X next to the material selection (under the words "Link to Object"). Then select "OB" and press "Add New" and name this material "FloorHit" (again, no quotes). Make it green colored, so you can easily find it while editing, but turn on "Shadeless" and "No Mist" in the Material section and "OnlyCast" in the Links and Pipeline - Render Pipeline section and turn off "Radio", "Traceable", and "Shad(ow)buffer" in the Render Pipeline and "Shadow" in the Shaders section. This will make it completely invisible at runtime and take up little resources.

(question: it wont make it invisible it just flashes rapidly)

## Making the sensor require floor contact

Now select Suzanne and go to the logic panel. Add a touch sensor called "jumpcol" (for 'jump collision') and connect it to the same AND controller as the jump keyboard sensor. Set the f to 10 and the MA: to FloorHit. This will make it so your character can only jump while it's touching something with the FloorHit material. By that same token you can link this to *all* of your movement-related AND controllers so that the player can't adjust its movements in mid-air. This is a nice physics touch but for most platformer games, where mid-air dexterity is almost essential, it doesn't work. You also have to use Force and not dLoc, otherwise your character won't be able to move while jumping at all, which is useless your character only needs to hit things with his/her head, and not jump over gaps. Note that this still does not keep the character from jumping while touching the hit test object at the sides, so they could still jump if they were touching the side of the collision surface, but at least they can't jump in mid-air or while touching the bottom of the ground. I have yet to find a solution to this problem not involving Python.

## Excessive Bouncing

Another problem with the original model, as you might have noticed, is that when you run into one of the ground cubes you bounce back a great deal. To fix this, turn to the materials panel with one of your green hit test squares selected. Under the color selectors there should be three buttons that say "RGB", "HSV", and "DYN". Select DYN, and turn the restitution most or all the way up. As you can see, you can also find the friction property from here, if you want your ground to be more or less frictional (like mud or ice).

## Final Notes

Always copy your floors, their respective hit test objects, and other generic objects with **ALT-D** and not **SHIFT-D**. This keeps you from duplicating things that remain the same for every one, like the material and mesh, which quickly drain resources if not recycled. Since there's not much further you can take the Platformer without learning Python, I suggest you start looking up information on the Blender API. Some essential information is to be found in the procedural object creation tutorial. A Blend file of the tutorial's finished product is coming soon.

**Note:** A (possibly) simpler method for jump limitation can be found on the discussion page for this tutorial.

# Blender 3D: Noob to Pro/Making exe

---

This tutorial will show you how to make an executable for your game made in Blender.

Note: The methods listed in the "Windows" and "GNU/Linux or Mac OS X" sections work only for the operating system you are on when you create the file. To make it cross platform, use the "BlenderPlayer" method.

## Windows

First create a folder that will hold all your game information. Name it something meaningful, like "Yo Frankie!".

The folder must contain four files that you can copy from your blender installation directory.(under windows it should be "C:\Program Files\Blender Foundation\Blender") They are:

1. SDL.dll
2. python24.dll (*Or other relevant python file; python25.dll, python26.dll etc*)
3. pthreadVC2.dll
4. zlib.dll

(sometimes you will need the following also...)

1. avformat-51.dll
2. avutil-49.dll
3. avcodec-51.dll

They are in the same folder.)

Not quite a Noob Note: If you are using the latest version of Blender you WILL need the following files)

I am running windows xp media edition and blender 2.46, here is a full list of files i needed:

1. avcodec-51.dll
2. avformat-52.dll
3. avutil-49.dll
4. libfaac-0.dll
5. libfaad-0.dll
6. libmp3lame-0.dll
7. libx264-59.dll

8. pthreadVC2.dll
9. python25.dll
10. SDL.dll
11. swscale-0.dll
12. vcomp90.dll
13. xvidcore.dll
14. zlib.dll

On Blender 2.48a (Windows XP Media Center 2002, SP3), these .manifest files are also required:

1. blender.exe.manifest
2. blenderplayer.exe.manifest
3. Microsoft.VC90.CRT.manifest
4. Microsoft.VC90.OpenMP.manifest

Now open blender with the game that you have created and open the file menu. Click on **Save Game as Runtime** and then save to the new folder that you have created and rename the file yourgamenname.exe, and then you can run the game!

## Making A Screensaver (Windows Only)

1. First, save your runtime (that's the .exe addressed above.) via File->Save Game as Runtime. ("Save Runtime" in older versions)
2. Now, go to your .exe game and rename it .scr - for instance, if your game was NotMyGame.exe, rename it to NotMyGame.scr. You can now right click to Install it, and then use it as your regular screensaver by applying it as you would any other screensaver (right click on desktop, properties... You know the drill.)

Just a helpful idea; when distributing my games, I often include a real awesome screen saver so that when Person A walks into Person B's room and sees that amazing thing going, they decide that they want my game. Just a note, though; screensavers are not games, and so they should not accept input. At most, they should be videos showing what your game does. If you just rename your regular game as a .scr, it will be remarkably boring, because your game needs input and screensavers do not.

## Intel screensaver bug

There is a caveat with Intel Integrated Graphics drivers, found in many laptops though. The graphics driver shuts off OpenGL acceleration for screensavers, for some obscure reason. The way to work around this is to rename the .scr extension to a **.sCr** as the driver's algorithm is dependent on case-sensitive characters. If you experience very low framerates in your screensaver, you should attempt this fix, It is tested and reported to work.

## GNU/Linux or Mac OS X

Open blender with the game that you have created and open the file menu. Click on **Save Game as Runtime** and then save to the new folder that you have created and rename the file yourgamenname... and then you can run the game!

## BlenderPlayer

The methods shown above only create an executable for your operating system. Well, BlenderPlayer can fix that.

1. Make a new folder to store all your game data.
2. Then save your .blend file into the directory.
3. You can skip this step and the next step if you do not want a Windows version. For the Windows users, copy blenderplayer.exe to the new folder from a Windows copy of Blender. Then copy all your DLL files for Blender as mentioned for Windows to the folder.
4. Next you have to make an MS-DOS batch file (for \*NIX users, this is the shell script equivalent). In a simple text editor, in CR-LF mode if available (Notepad is always in this mode, and NOT a word processor!), copy and paste this text:

```
blenderplayer.exe yourgamenname.blend
```

Save it as YourGameName-Windows.bat in your game folder.

5. You can skip this step if you do not want a \*NIX (basically Mac OS X and GNU/Linux) port. For GNU/Linux (at least), make a shell script. (A shell script is the \*NIX term for a batch file.) In a simple text editor, in LF mode (unfortunately Notepad can't be used), copy and paste this text:

```
#!/bin/bash
```

```
blenderplayer yourgamenname.blend
```

Save the file as YourGameName-UNIX.sh in your game folder. Note that due to the nature of these systems, BlenderPlayer will need to be pre-installed.

6. Write a readme for your program. This is again best done with a simple text editor like Notepad or gedit, but it does not matter which mode it is in. You should include the name of the game, a description, perhaps a walkthrough or hints, and if you made a \*NIX port, mention that it requires BlenderPlayer, available with Blender.

## Legal Issues

Blender and the BlenderPlayer fall under the GNU General Public License. Blend files are copyrighted to their respected owners and do not fall under the GPL as long as they are not packed in the BlenderPlayer. If a user does not want the blend files to fall under the GPL, it is recommended not to use the "Save Game as Runtime" feature. To put it another way, the user must keep the blend files and the BlenderPlayer in separate files. More information can be found at <http://www.blender.org/education-help/faq/gpl-for-artists/>.

## Proprietary Blend Files

Blend files are open source by nature. Anyone who has Blender can open a blend file. However, there are a few ways to "lock" a blend file. Common methods involve encrypting the blend file and then decrypting it at runtime. This can be accomplished by using python scripts or by using external 3rd party applications (which are feasible under the GPL v2).

# Blender 3D: Noob to Pro/Build a skybox

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 <sup>[2]</sup> or the incredibly cool Autostitch <sup>[1]</sup> 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 <sup>[2]</sup>. 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 <sup>[3]</sup> 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 <sup>[4]</sup>.

Here are the images I'll be using (you'll want to use images with higher resolution): Note that the sky has trees, etc.

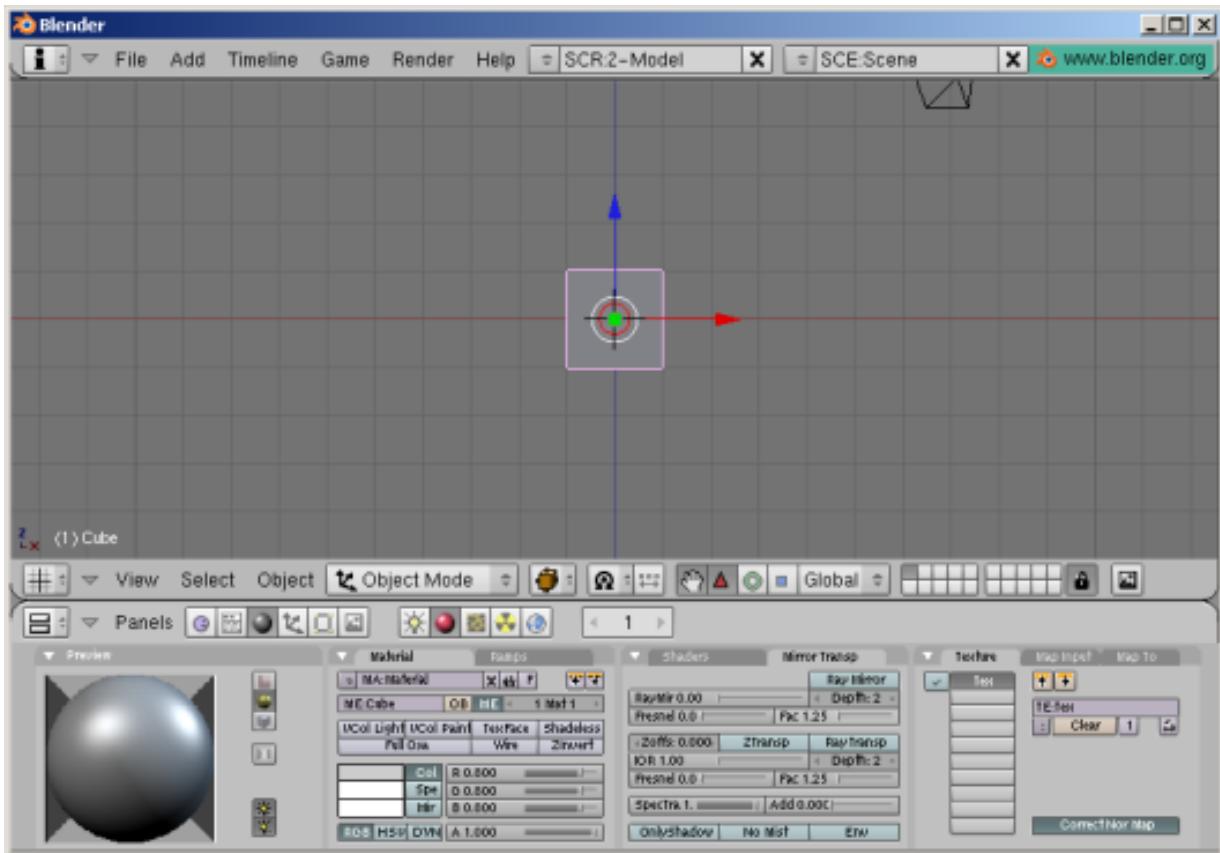




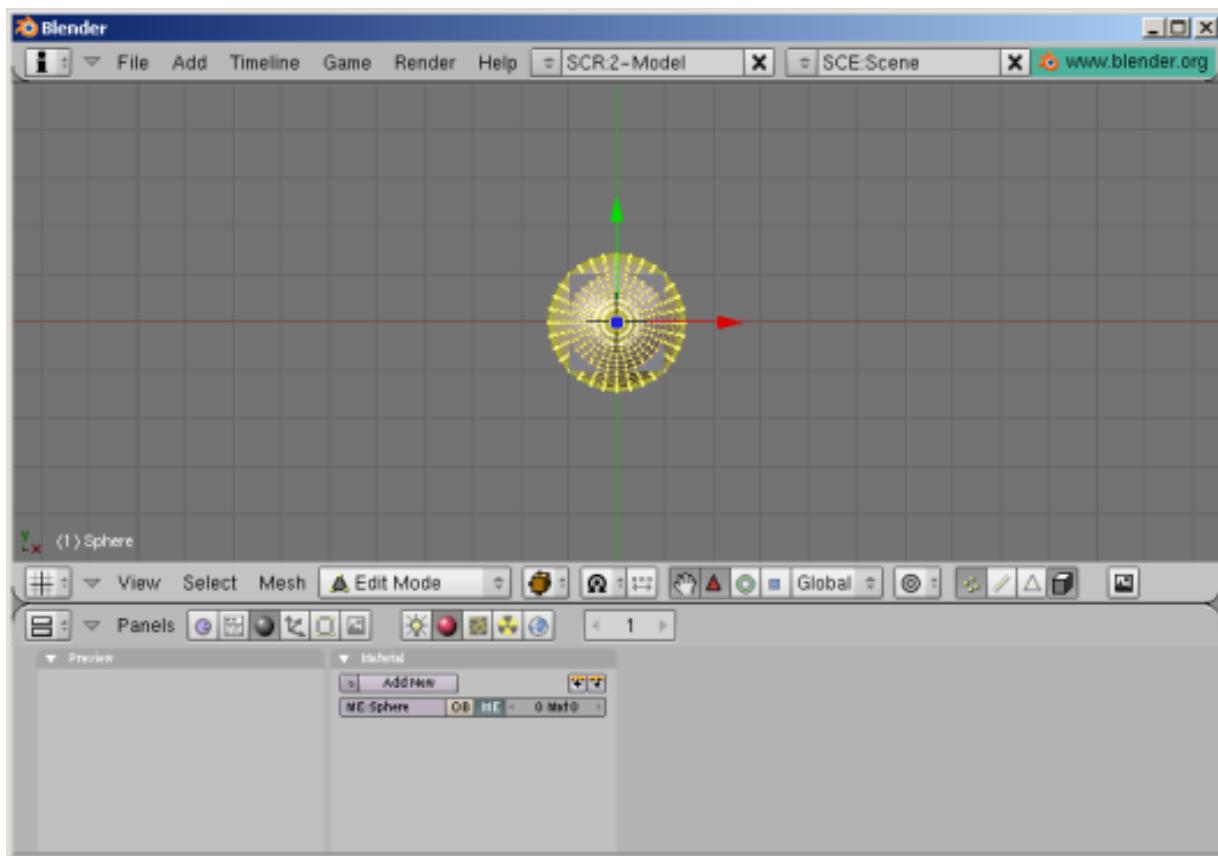
Note: I've outlined the horizon of the ground texture with an alpha channel which will allow me to place the ground mesh right against the sky mesh with a very natural feel.

## Create a dome for the sky

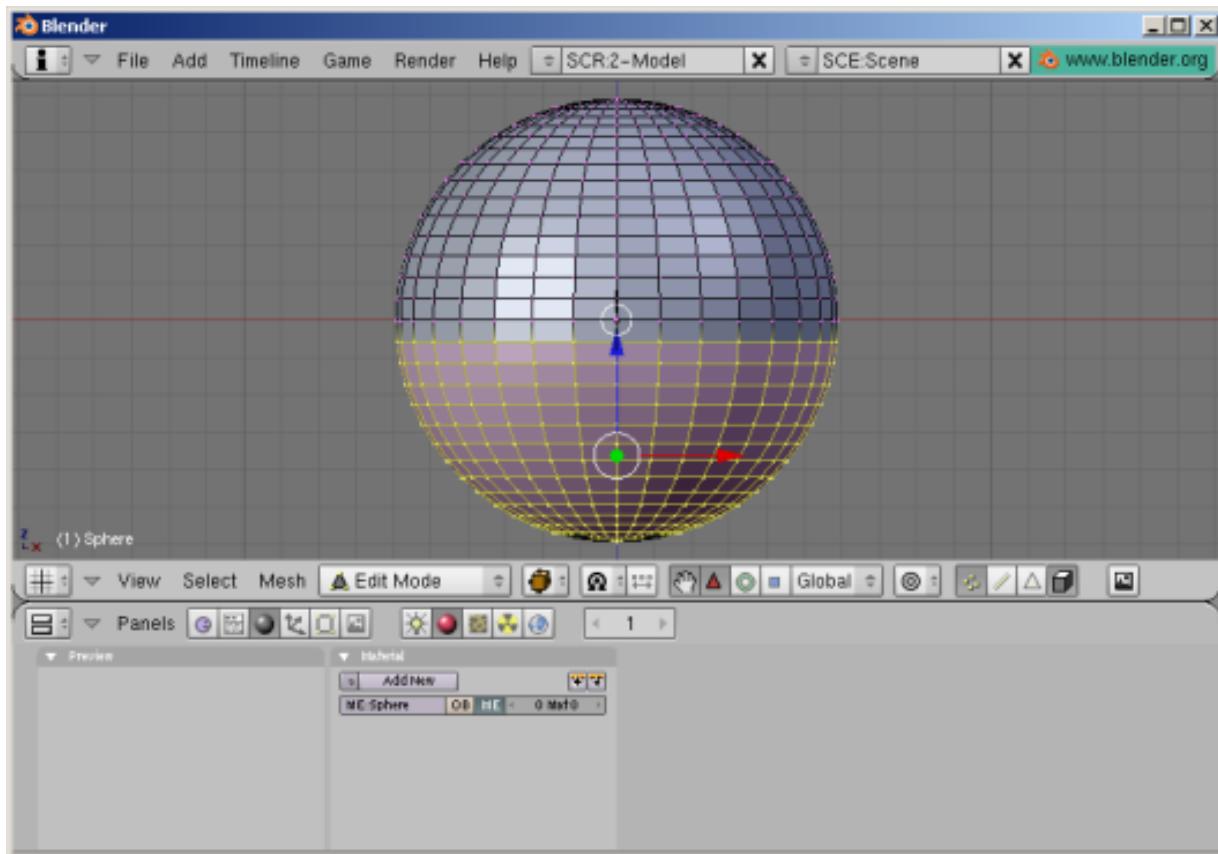
Open a new file in Blender. Your default new file will probably be a two-unit cube in the center of the screen, with a single light source and a camera. You can delete the light source because we won't be needing it. Leave the cube, because that is what will become our skybox.



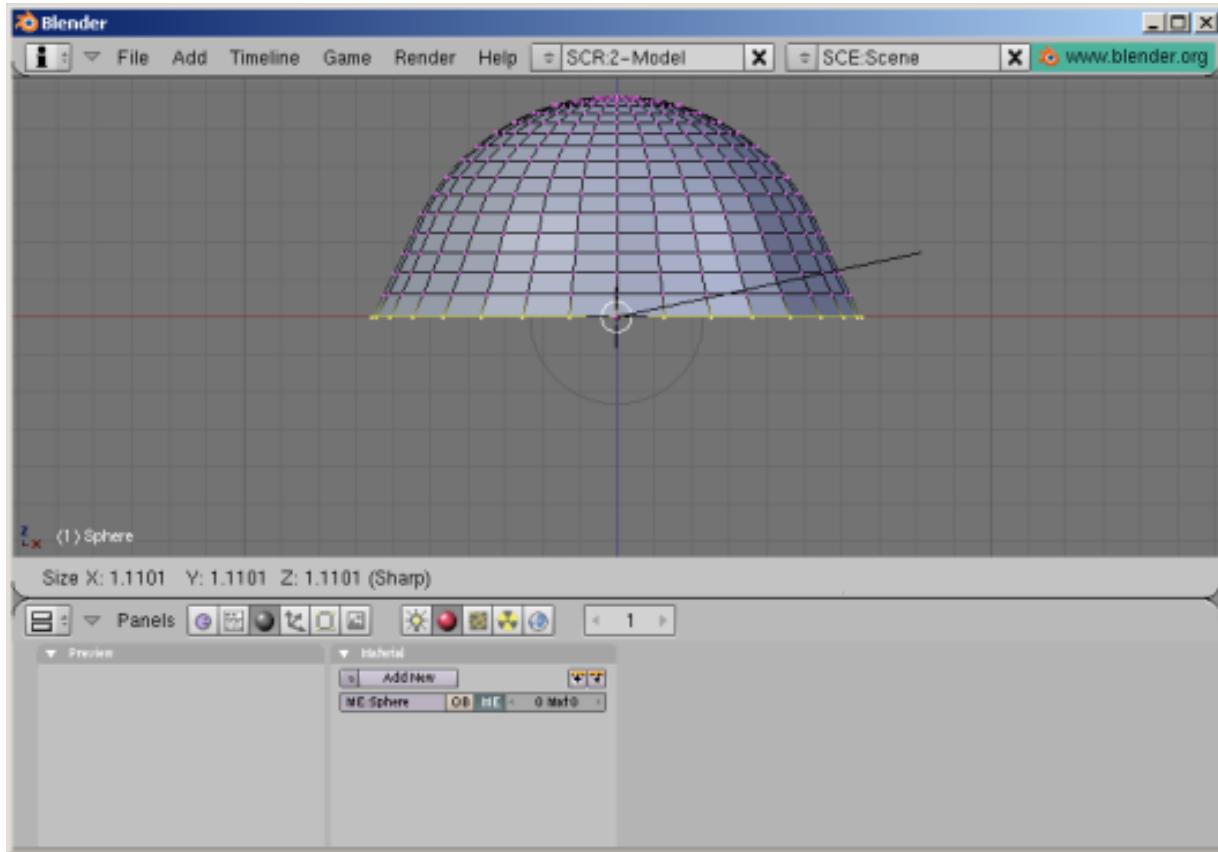
The cube will be the center of our environment, so use Object->Snap->Cursor To Selection if your cursor is not centered. Then, from the top view [KEYPAD-7], Use [KEY-SPACEBAR] to insert a new mesh; make it a UV sphere. I find a 32-segment, 32-ring sphere to be sufficient. We create the sphere from the 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.



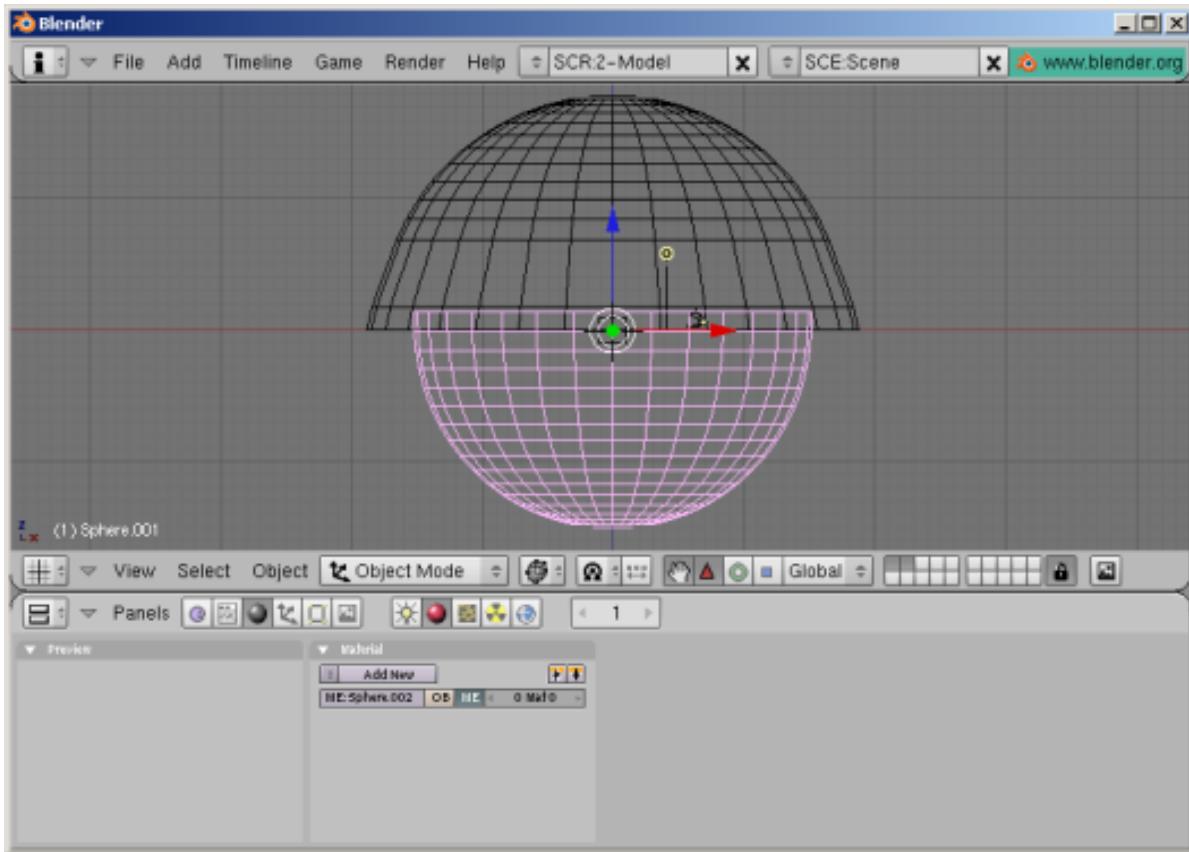
Now you're ready to add your sky texture to this mesh. In the Materials menu, create a new material and a new texture. Be sure to set your material not to receive shadows by clicking the "Shadeless" button. Then, in the Texture menu, set the texture type to Image, and click the Load Image button to insert 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.



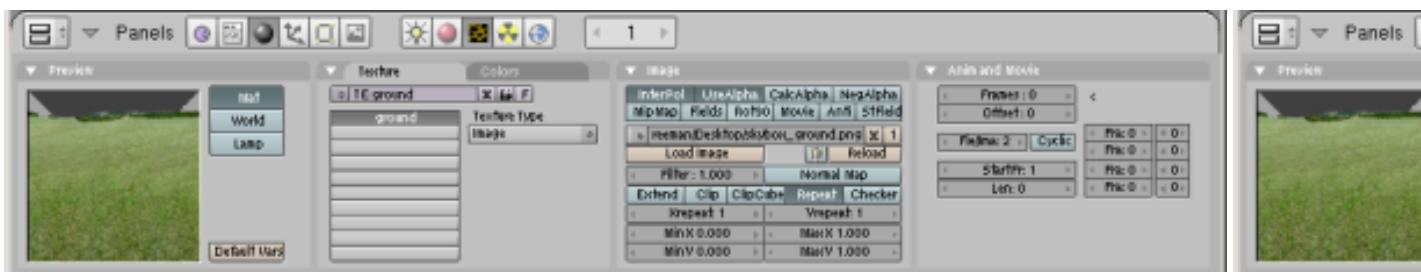
At this point, if you wish, you can reposition the camera and render the scene to see how your sky mesh looks.

## Create a dome for the ground

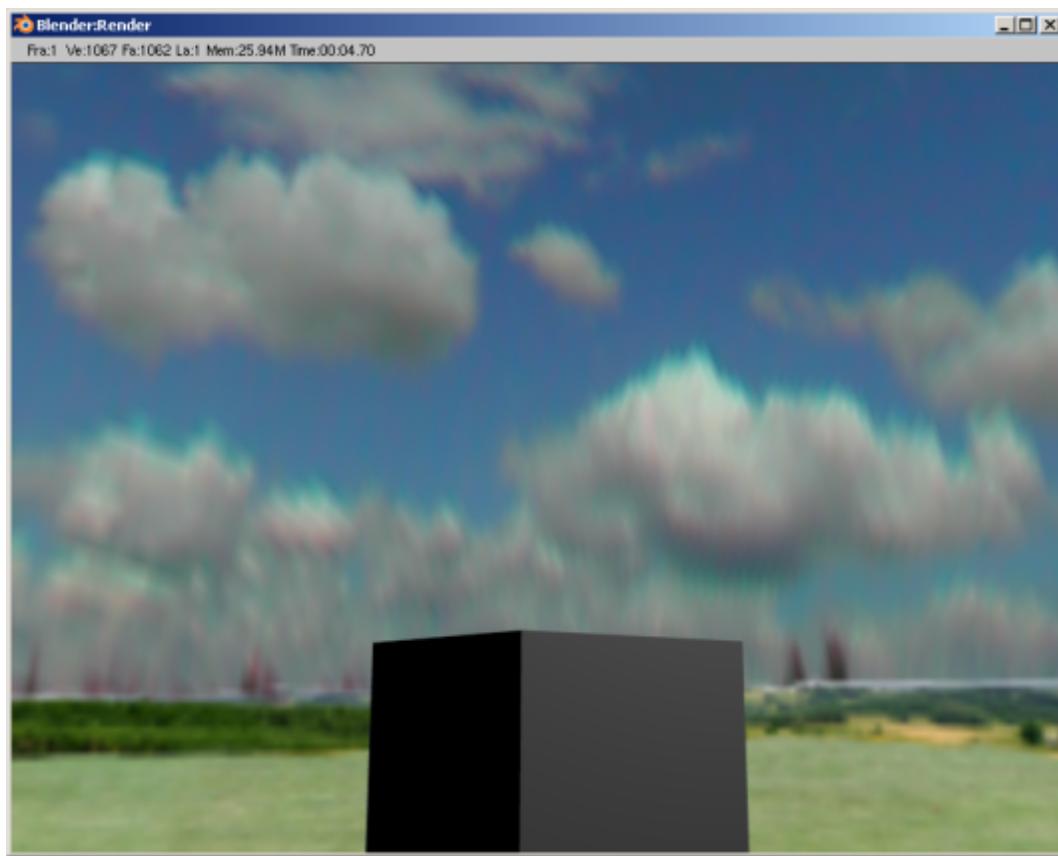
I found it easiest to move the sky dome to a new layer with the [KEY-M] move to layer command. Then you can select the cube, Object->snap cursor to selection if you need to, select the top view [KEYPAD-7] and insert another UV sphere just as before -- except this time, remove the top hemisphere of vertices. I left an extra row of vertices at the "equator", scaled up, to function as a "billboard" to display the 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 [Materials panel | Map Input tab | Tube button]. Because I have an alpha channel on this texture, I click "Use Alpha" in the Texture menu and Map To -> both Col and Alpha buttons [Materials Panel | Map To Tab | -> both Col and Alpha buttons]. You will also need to set ZTransp in the Mirror Transp menu [Materials Panel | Links and Pipelines Tab | ZTransp button] so that your alpha channel shows up in the envmap (which will become your skybox), and Alpha to 0 [Materials panel | Material tab | A slider ] 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.

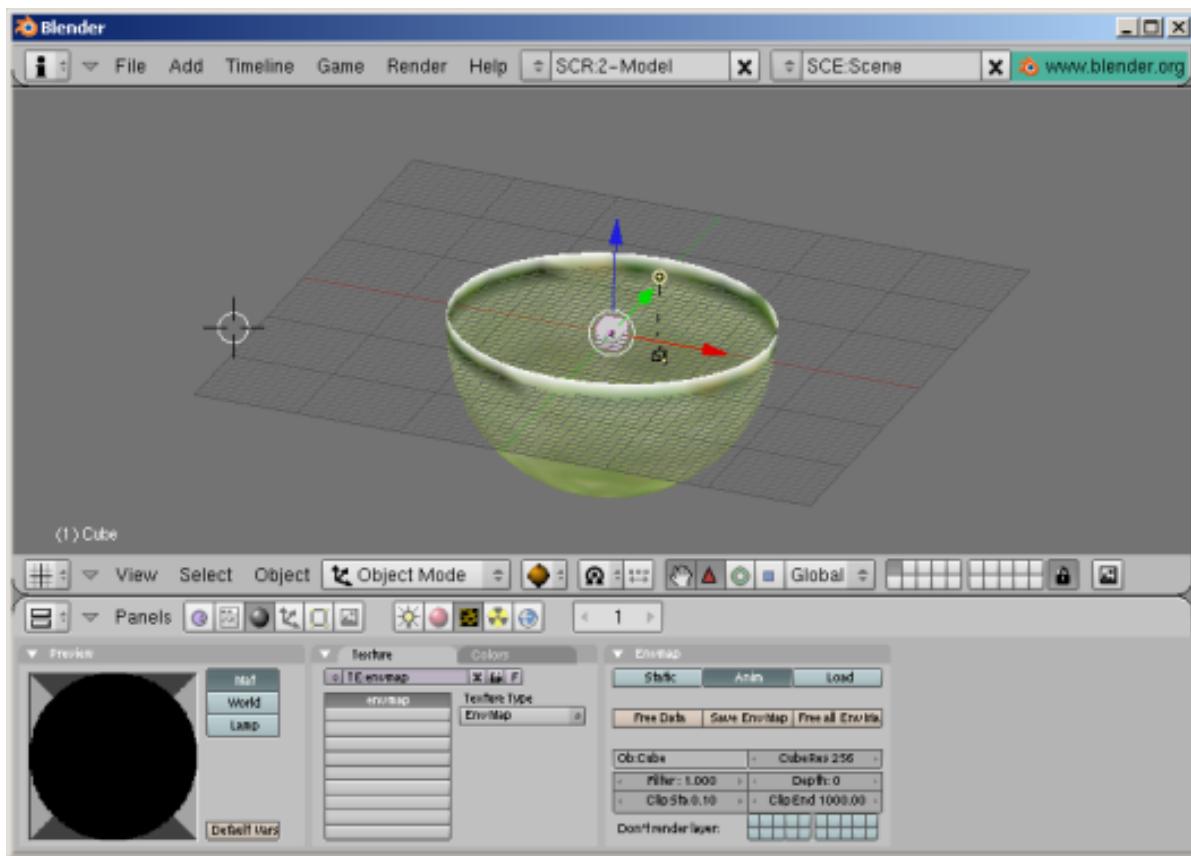


Again, you can reposition the camera and render the scene to make sure everything is properly aligned. Be sure to activate the layer where you moved the sky mesh. Your results will look similar to the following image. Set OSA on in the render screen for best results. Also, use higher resolution images with cleaner alpha channels -- the image below is rather blurry and you can see a halo around the horizon.



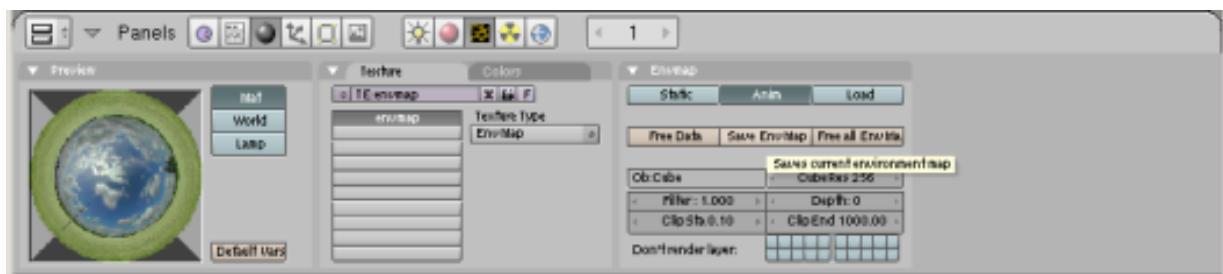
## Render the environment map

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" [Materials Panel | Materials Tab | Shadeless button]. Add a new texture and make its type Envmap. Set the CubeRes [Envmap tab | CubeRes setting] 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.)

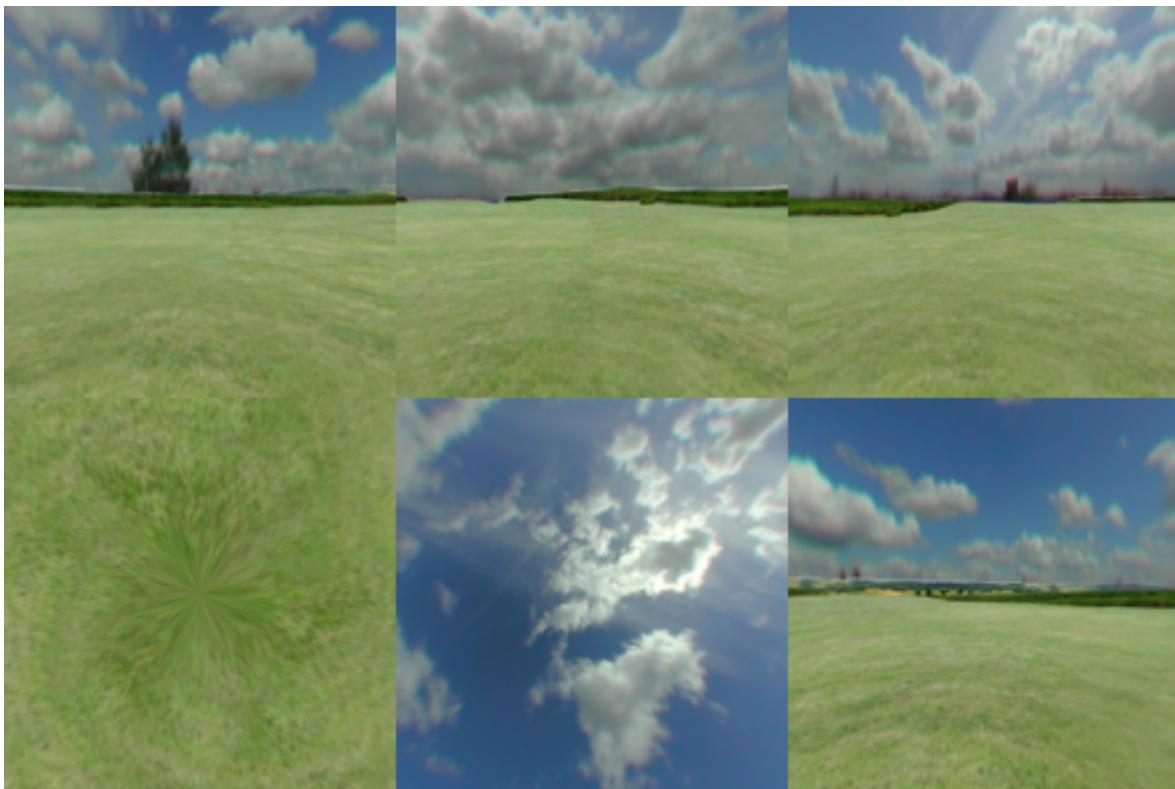


Once you've created the Envmap texture, you should be ready to render the Envmap for your skybox. If you want to set your file format such as JPG or PNG, you should do that first. Then, simply go to the render screen and click "Render." Again, make sure all layers are visible. The rendering window appears. 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:



The cube faces are in the following order.

LEFT (-X)	BACK (-Z)	RIGHT (+X)
<hr/>		
BOTTOM (-Y)	TOP (+Y)	FRONT (+Z)

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 the current active camera object.

## Video Tutorial

Ira Krakow's Blender 2.49 Skybox Tutorial: <http://www.youtube.com/watch?v=azkk3JrM5Es>

## References

- [1] <http://www.autostitch.net>
- [2] <http://www.philohome.com>
- [3] [http://en.wikibooks.org/wiki/Blender\\_3D:\\_Noob\\_to\\_Pro/Mountains\\_Out\\_Of\\_Molehills](http://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Mountains_Out_Of_Molehills)
- [4] <http://www.accustudio.com>

# Blender 3D: Noob to Pro/Basic mouse pointer

---

Making a simple mouse pointer in the game engine. This takes up a lot of resources but it is very simple.

A little Python is involved but it is very easy to use and is only 2 lines of code.

1. Open up blender and split the screen in two.
2. Make the right screen a text editor and add a new text file with **ALT+N**. Type in the following code.

```
import Rasterizer as r
r.showMouse(1)
```

3. Set *TX: showpointer* in the middle of the Text panel menu bar.
4. Select an object that will always be available - preferably a camera.
5. Go to the logic tab, add a "*Property*" sensor, a "*Python*" controller, an "*AND*" controller and a "*Property*" actuator.
6. Activate the *True Level Triggering* (the ““ button), set *Prop: switch* and *Value: 0*. Connect the property sensor to the python and AND controllers by dragging lines between the bullets.
7. For the python controller set *Script: showpointer*. Note: If the value keeps being reverted to blank after setting it, the name you entered is not a legal script name; chances are you did not set the name of your script correctly. Look for the selection menu beginning with "TX:" and make sure it says *TX:showpointer*.
8. Now connect the AND controller to the property actuator. Set *Prop: switch* and *Value: 1*.
9. Select *Actor* and click on *Add Property* make it a *Int* type and set *Name:switch*.

Now press **P** to start the game and now you'll see your mouse pointer.

# Blender 3D: Noob to Pro/Text in BGE

---

There are a lot of tutorials that show how to make text for the Blender Game Engine, for example to use in menus. Most involve editors, graphics programs, targa files, UV mapping, scripting, higher magic and so on. Here is a different approach that takes but a minute.

1. Start Blender.
2. Remove the default cube. (**X**)
3. Add text. (**SPACE** → *Add* → *Text*)
4. You may want to switch to edit mode (**TAB**) and use the variety of features that Blender provides for editing, formatting and laying out text. They are described in the Blender Manual Section <sup>[1]</sup>. For a start just use **BACKSPACE** to delete the letters "Text" and type your own like "This is simple".
5. When finished editing go back to object mode. (**TAB**)
6. Convert your text to a mesh. (**ALT+C** → *Mesh*)
7. Press **P**

This is as simple as it gets. Of course you can do all kinds of laying out with your text before the conversion in step 6 (check the Blender Manual) as well as modeling, texturing and dynamically manipulating thereafter (as you have a normal mesh). Simply remember that you need to make sure the text is final before you convert it.

This tutorial has been tested with Blender 2.8 RC1.

"However, it should be pointed out that while this is so incredibly simple, it is not poly-count efficient. Remember, there is more to making a game than making it look good--making it run well. If you need a cut-and-dry menu, this is the best way. If you need to worry about performance, then it might be worth it to check out some UV Mapping and external program usage. --Bullercruz1 (talk) 23:07, 7 February 2009 (UTC)"

It is also important to note that these fonts can't be dynamic, contrary to bitmap-uvmapped ones...

## References

[1] <http://wiki.blender.org/index.php/Manual/Text>

# Blender 3D: Noob to Pro/Python Platformer: Creation

Note: Python code is placed in a Text Editor window. It might be helpful to split your 3D View window into a separate part, so that you can use a buttons, 3D view, and text editor window simultaneously.

The initial stages of the Python Platformer tutorial series will mostly have to do with replicating what was done in the logic board Platformer tutorials. If you have not read and understood those tutorials, you may not understand exactly what the function of most of this code is. This tutorial details:

1. An explanation of the "code sections" to be modified, rewritten, or redefined frequently throughout the series
2. Creating an object linked to a pre-made mesh
3. Linking an object to a scene

and will detail by completion:

1. How to move an object in response to keyboard triggers

## Code Sections

We will refer to the sections of the code in this manner throughout the series:

### Import Section

The series of import commands at the beginning of the piece of code, like the contents of the head tag in HTML documents. Python has a basic set of commands kept naturally in the language, and the rest are imported so that a large amount of commands aren't loaded when not needed, causing unnecessary memory expenditure. This lends the advantage of being able to extend the language by writing custom sets of commands to be imported into Python, which is how Blender interfaces with Python. For this tutorial, our import section is:

```
import Blender  
import bpy
```

When you import something as something else, it basically creates a variable equal to the imported module's name, so that you don't have to type it out. It is inessential, but it makes the coding go faster. You might want to run a find and replace search on your document afterwards, replacing all instances of the name you imported it as with the real module's name and deleting the part that imports the module by another name, as this can make the code run slightly faster.

### Essential Footer

This is the collection of commands that you must remember to include at the end of the document, regardless of any changes made to the code, for it to work. It must be noted whether you should change the entire footer (as if a feature was removed from the main code-in-progress) or adding a command to your existing footer (if the text deals with the addition of a single feature). Every article in the main tutorial line should contain the full footer as it should look at that point in the series. For this tutorial, our essential footer is:

```
Blender.Redraw()
```

## Adding the Player Object

To start, create a new document and delete the basic cube. Create a Monkey/Suzanne. In the Editing panel with the monkey selected, change the mesh name (ME:) under Link and Materials to "Hero". Click the F next to the input box to preserve the data block even when nothing links to it. Now delete the monkey object.

The following code will create an object called "Player" in the library and link it to the Hero mesh:

```
player = Blender.Object.New("Mesh", "Player")
player.link(bpy.data.meshes["Hero"])
```

In the first line, `Blender.Object.New` obviously references a new object. The "Mesh" variable should not be changed for the purposes of this code. I don't know exactly its function, so I don't want to give out misinformation, but I speculate that it has to do with the object type. If the object were to be a lamp or camera type, for example, you would not be able to apply a mesh to it. The second variable, "Player", is the name of the actual object you're creating. Change it to your liking.

In the second line, we link `player` (which was equated to our new object) to the pre-existing mesh "Hero", which is the Suzanne mesh we dealt with before. Using this method, you can model your character's mesh beforehand but have the actual character created dynamically. Using a complication of this code, you can make `player.link()` link to a variable and not the bit "`bpy.data.meshes[]`". That variable can reference an existing mesh or it can create a new one. As for the meaning of the link's value, `bpy.data.meshes` is an array containing all the meshes in the movie. Likewise, `bpy.data.objects` contains all the objects in the movie. By going to the Scripts window and going to **Scripts->System->Interactive Console** you can gain such information as the contents of these arrays. By entering "`list(bpy.data.objects)`" into the console, you will be rewarded with a list in the format of `[Object "Camera"]`, `[Object "Cube"]`, `[Object "Lamp"]` which is the list of objects in the standard new document set-up. So, to reference the item `[Object "Cube"]` you would use the line `"bpy.data.objects['Cube']"` and so on. For these commands to work, you must make sure to import `bpy` in your import section.

## Appending the Object to the Scene

Like you must append the mesh to the object, you must also append the object to the scene, or it will just be a floating data block and not actually appear anywhere.

```
scene = Blender.Scene.GetCurrent()
scene.link(player)
```

If you were to test this code by pressing **ALT-P** while the mouse is hovering over the Text Editor window, it would create an object named Player with the monkey "Hero" mesh at the origin point of your scene. Make sure you included this tutorial's import section before all of the other code and the essential footer after all of other code.

# Blender 3D: Noob to Pro/Beginning Modeling Final Project

---

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 idiot 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 <sup>[1]</sup> forum on BlenderArtists.org(formerly elysium.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>

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 favourite graphics application. 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 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
- A Weapon

## References

- [1] <http://blenderartists.org/forum/forumdisplay.php?s=b6a3696a3b803274bb7c1b49df9a22a2&f=16>

---

# Unit Three: Python In Blender

---

## Blender 3D: Noob to Pro/Advanced Tutorials

---

Schedule to be re-written from the ground up. --Bullercruz1 (talk) 13:39, 4 April 2009 (UTC)

Author: Anthony Gomez (Extensor) Date: March 5, 2005

Short Description: RVK tutorial

- Letters in brackets i.e.: (z) mean there is addition information at the bottom of the page.

### **Introduction:**

This tutorial is meant to stop all the RVK (Relative Vertex Keys) questions.

### **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 separate 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 overlapping 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 other end of the curve if needed.

Warning! Blender has a limit to the number of verts you can use.

---

Click here to read the advanced animation tutorial guided tour.

# Blender 3D: Noob to Pro/Python Scripting

---

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 usable tool.

The relevant section of the Blender User's Manual is here <sup>[1]</sup>.

Download Python Scripts from the Blender site here <sup>[2]</sup>.

This forum <sup>[3]</sup> on blenderartists.org discusses Python scripting.

here is a remix and update to the great noob tutorial : /Tutorial (Blender 2.4.9)/

## References

[1] [http://wiki.blender.org/index.php/Manual/Python\\_Scripting](http://wiki.blender.org/index.php/Manual/Python_Scripting)

[2] <http://wiki.blender.org/index.php/Extensions:Py/Scripts/Catalog>

[3] <http://blenderartists.org/forum/forumdisplay.php?f=11>

# Blender 3D: Noob to Pro/Advanced Tutorials/ Python Scripting/Introduction

---

For a general introduction to Python Programming, see the official Python tutorial: <http://docs.python.org/tutorial/index.html>

## Introduction

Python is an incredible language, much akin to C, which also offers linking of C libraries, probably why it was chosen to be integrated into Blender. It was designed to incorporate the extremely quick and easy learning of ABC with the more powerful functionality of C, but at the same time keeping syntaxing and naming conventions simple. Arguably its main function is in importing and exporting 3D models, an essential task for a 3D modeling application. With the adaptability of Python, it allows you to create custom import/export scripts easily and quickly for Blender, to service filetypes which Blender doesn't natively support.

Next comes its ability to allow automation, such as the Discombobulator, which is a fantastic (but not often used) tool, which turns simple meshes into futuristic panels.

It is also almost universally used in the BGE (Blender Game Engine), to bring life to games.

And yes, Python was named after *Monty Python*.

## Your First Python Script

### Blender Setup

Python can run equally well outside Blender, however Blender supplies the modules Python requires to interact with it only when it is running. In which case it's probably best to write it within Blender.

### 2.4 and Below

So start up Blender and create a new view panel. Now change its type to 'Text Editor'. The script we will make will be stored within our Blend file itself, so there's no need to save it externally (unless you wish to). On the text editor toolbar find a drop-down arrow on its own (you might have to re-size the window). Click it and select "Text". This is an automatically created text file ready for you to program into. Now check the AB button and the button with a snake on. The AB button turns on colourising, so keywords are highlighted, which makes it easier to program. The snake button turns on plugins, which will allow you to see the autocomplete (the equiv. to IntelliSense) which *really* speeds up programming, and makes it far easier to become accustomed to Python.

### 2.5 Alpha 0

If you are using the Alpha of 2.5 you have a number of cool new features, and also missing ones, as you should have come to expect (releasing without Edge-Creasing?!). For this you can use two windows, so create one horizontally, and then split this vertically. Change the type of the top window to 'Console', which is amazing; and the bottom to 'Text Editor'. In Text Editor, like in 2.4, find the notepad icon next to the 'New' button, click on this and load the default 'Text' file.

## Let's Get Programming

### Useful Tips

- The Dir(...) function is extremely useful. Use dir(<itemname>) to list all the properties and functions of <itemname>. For example: in Blender 2.4-

```
import Blender ; print dir(Blender) #The semi-colon allows for it to be  
on one line, normally commands are separated by newlines
```

- print outputs whatever comes after it on the commandline, as long as it's delimited by speech marks, or encapsulated in a variable, eg: -

```
print "Hello World!"
```

```
A = "SPAM"
```

```
print A
```

- Blender 2.5 uses the newer but inconsequentially different Python 3.0, and so requires brackets or parentheses after print and around the text (just to be difficult and remove the most commonly used keyword in the entire language) e.g: -

```
print ("Hello World!")
```

- Any output unless written to the 3d view or the Python window will be directed to the command line window that appears behind Blender.
- Using the console allows you to use dir() and other functions and their output will be displayed in there (*this is only in 2.5*)
- Hit **Ctrl-Space** to activate the autofill in 2.4 and inside the console in 2.5

- Press **Alt-P** to execute your script

## 2.4 and Below

```
import Blender #imports the module supplied by Blender, which allows
you to manipulate the scene and Blender itself.

a = Blender.Object.Get("Cube") #Select an object from the Scene and
assign it to a variable for easy use
a.LocX = 2 #move the location-x of the cube now encapsulated in our
variable 'a' to 2.
```

This simple script sets the position of your default cube (as long as it's named 'Cube') along its local X-axis to 2, in effect moving it 2 along. If you want to move it 2 from wherever it is, use:

```
a.LocX += 2
```

which is the same as

```
a.LocX = a.LocX + 2
```

## 2.5

2.5 is completely different in that it doesn't have a Blender module. Whether this is consistent with the final release of 2.5, we have the other method to work with.

```
import bpy
Cube = bpy.data.objects["Cube"] #get the object named "Cube" from our
scene data and assign it to the variable 'Cube' for easy access

Cube.delta_location #this is a vector (a list of its current positions
relative to [X,Y,Z] respectively)
Cube.delta_location[0] #List is zero-based (i.e. item one in the the
list is accessed with the index 0, but we want to change the Y-axis,
which is the second item)
Cube.delta_location[1] = 2 #Assign the location of our cube's Y-axis

#OR

Cube.delta_location[1] += 2 #As before, you can simply add 2 to
whatever value there was originally , rather than re-assigning it.
```

we can see all the properties and methods on our object easily in the console.

```
import bpy
dir(bpy) #show all the methods and other classes of bpy
```

This will output a list of items You could also type in

```
bpy.
```

and press **Ctrl-Space** which will autofill for you, and show you the same information in green.

Obviously you can simply type out each line of the code in your script straight into the console and it will run fine.

## Exercises

- Easy
  - Change the script so that it sets the object's location-x to 6
  - Change the script so that it sets the object's location-y to 1
- Intermediate
  - Change the script so that it sets the object's rotation-x to 2
  - Change the script so that it sets the object's rotation-y to 4
- Hard

Look up the time module in the Python reference on the Python website, then using `time.Sleep(...)`:

- Change the script so that it animates the object, moving it along the z-Axis by 10 and rotating along the z-Axis by 180 in a total of 20 seconds.
- Then run it in-game using `bpy.ops.view3d.game_start()` (*2.5 only*)

## Summary

Hopefully, if anything, you will have gotten a feel for the Python engine under Blender's hood. It's an amazing feature which many people forget about, and you could soon find yourself using it often.

# Blender 3D: Noob to Pro/Advanced Tutorials/ Python Scripting/Export scripts

---

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 (2.44 or greater), 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 known 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 Python Console*.

Now, you are ready for the big moment, you are about to execute the first Blender scripting command. Type this and hit **RETURN** (or you could type into the scripts window import Blender on the topmost line, then these lines below it, precede all the 'dir(x)' lines with print and choose file->Execute): [Note:that you shall have typed

```
import bpy
```

before you can use the list function (bpy.data.objects) else will you get the "python error" in blender ]

```
list(bpy.data.objects)
```

As a result, you should see this:

```
Object "Camera"], [Object "Cube"], [Object "Lamp"]
```

Now, what just happened? The line "list(bpy.data.objects)" consists of three words, separated by two dots. The dots separate different things. The first, *bpy*, means to use a function from the *bpy* module. *data* is a sub-module of Blender. And finally *objects* is an iterator of *bpy.data*. The *list()* function is used to loop through all data in *bpy.data.objects* and return that as a list of all available objects. In our case, this is a Camera, a Cube, and a Lamp.

To get more information about an object, you can use the object name as a key in *bpy.data.objects*, and assign it to a variable, like this:

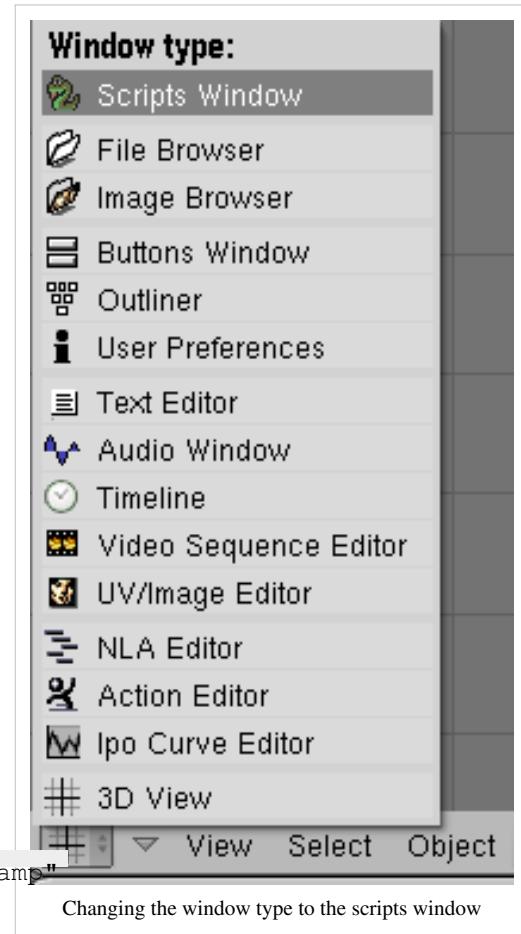
```
camera = bpy.data.objects["Camera"]
cube = bpy.data.objects["Cube"]
lamp = bpy.data.objects["Lamp"]
```

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"]
```



Changing the window type to the scripts window

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(bpy.data.objects)
```

This will print the documentation of the bpy.data.objects 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.blender.org/documentation/244PythonDoc/index.html>

In the documentation, click on bpy, then on data and you can see more examples. 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.type
```

It will tell us that the cube really is a Mesh object in Blender. Look up "type" in the online docs. Since the variable cube holds an Object, and "type" is an attribute of that Object, click on Object. There you find its "type".

Now that we know that the cube is a mesh, let's find out more about the mesh.

```
mesh = cube.getData(mesh=1)
```

Every Blender object has data assigned to it, depending on the type. In the case of a mesh, the data are of type Mesh. In the documentation, go to the top again, and look for the Mesh module. It will contain documentation for the Mesh type. You can also try

```
dir(mesh)
```

to get an idea about the available functions and properties. Try these:

```
list(mesh.verts)  
list(mesh.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 = mesh.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/Mesh, when you click one "MVert" 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 mesh.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 built in text editor can be hard to use if it doesn't have the standard shortcuts of your preferred text editor. To copy text to and from your clipboard in blender, you have to use **CTR+SHIFT+C** and **CTR+SHIFT+V**, otherwise it uses an internal clipboard. Otherwise, it is quite usable. You reach it over the window selector, or by pressing **SHIFT+F10** (**SHIFT+F11** for Blender 2.41 and up). 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 code below into it, and save it. Or alternatively, paste the code below into a file, and open that file with *File → Open* in Blender. As name choose something with the extension .py, for example `wikibooks.py`. Put it into Blender's user scripts path.

For different operating systems this is:

- Linux: `~/.blender/scripts`
- Windows XP: `C:\Program Files\Blender Foundation\Blender\blender\scripts`
- Windows XP (alt): `C:\Documents and Settings\USERNAME\Application Data\Blender Foundation\Blender\blender\scripts`
- Windows Vista: `C:\Users\USERNAME\AppData\Roaming\Blender Foundation\Blender\blender\scripts`
- Mac OS X:
  - 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/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.
  - Right-click (or ctrl-click) the file "blender", and select "Show Package Contents" in the popup-menu. It will display all the hidden files under `blender`'s folder, and select "scripts" folder inside it.
  - To see this folder from the OSX terminal use the `ls -a` command (lists all folders/files even hidden) in the `MacOS` folder of the listed path. It is probably 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.
  - A safer approach than the one above consists in keeping your scripts somewhere in your home folder: with this scheme, there is no risk of deleting your scripts when you upgrade your blender application, as they are not contained within its folder. A method that follows this principle is as follows: create a folder that will contain your scripts (or some of them) inside your own home directory; then, instead of putting your files directly in the `.../.blender/scripts/` folder discussed above, simply add a *link* to your script directory in the `.../.blender/scripts/` folder (for instance with the "`In -s`" Unix command, or by doing "`open`

/Applications/blender-2.37a-OSX-10.3-powerpc/blender.app/Contents/MacOS/.blender/scripts/" [adapted to your version of blender] and then creating a link through the Finder, with File->Make Alias). Blender will now find all the scripts that you put in your home directory: it will follow the link you created in its .../.blender/scripts/ folder and go to the corresponding folder in your own directory, and find all the python scripts you put there.

```
# !BPY

"""
Name: 'Wikibooks'
Blender: 244
Group: 'Export'
Tooltip: 'Wikibooks sample exporter'
"""

import Blender
import bpy

def write(filename):
    out = file(filename, "w")
    sce= bpy.data.scenes.active
    for ob in sce.objects:
        out.write(ob.type + " : " + ob.name + "\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: 244
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
import bpy
```

Remember how we said all functions from the bpy 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 (if you want to directly use functions from the Blender module in a script, you can simply replace the import statement above with "from Blender import \*": no "Blender." prefix is necessary anymore; however, this slows down the loading of your script). So the above simply allows us to use the functions from the Blender module in our script.

the bpy module is new and will replace Blender for data access.

```
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".

```
sce= bpy.data.scenes.active  
for ob in sce.objects:  
    out.write(ob.type + ": " + ob.name + "\n")
```

These three lines are our real export script. You already know what the first line does - first we get the current scene, then get a list of all objects in that scene, the for loop is assigning each one in turn to the variable "ob". 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
import bpy

def write_obj(filepath):
    out = file(filepath, 'w')
    sce = bpy.data.scenes.active
    ob = sce.objects.active
    mesh = ob.getData(mesh=1)
    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 what's going on.

```
sce = bpy.data.scenes.active
ob = sce.objects.active
```

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 it's an easy way to test a new exporter.

```
mesh = ob.getData(mesh=1)
```

This gets the object's linked datablock. At the moment we don't know it's 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 vertex, 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 Mesh and MFace 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.

## Blender 3D: Noob to Pro/Advanced Tutorials/ Python Scripting/Import scripts

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:
        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) # Mesh must be spelled
just this--it is a specific type
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.

# Blender 3D: Noob to Pro/Advanced Tutorials/ Python Scripting/Procedural object creation

## Generating Elements with Python

A quick reference for commands to create elements through a Python script instead of manually. All of these scripts require the following code at the start:

```
import Blender  
from Blender import NMesh
```

We will also, for the tutorial, be working with an object, known as obj. To work with obj, after the import sequence, add the line

```
obj = NMesh.GetRaw()
```

If the name in quotes of an existing object is entered into the parentheses instead of left blank, it will use that object's mesh as a template.

### Vertices

The basic structure of the vertex generation command is:

```
v=NMesh.Vert(x.x,y.y,z.z)
```

Where v is the name of the vertex and x.x, y.y, and z.z are the X, Y, and Z coordinates of v, which must end in a decimal place.

For example, here is how one might create the three vertices of a triangle and attach them to obj's mesh:

```
v=NMesh.Vert(1.0,0.0,0.0)  
obj.verts.append(v)  
v=NMesh.Vert(0.0,1.0,0.0)  
obj.verts.append(v)  
v=NMesh.Vert(0.0,0.0,1.0)  
obj.verts.append(v)
```

Or, for vertices not created under the same name (v), it can be written thusly:

```
a=NMesh.Vert(1.0,0.0,0.0)  
b=NMesh.Vert(0.0,1.0,0.0)  
c=NMesh.Vert(0.0,0.0,1.0)  
obj.verts.append(a)  
obj.verts.append(b)  
obj.verts.append(c)
```

## Faces

The basic structure of the face generation command is:

```
f=NMesh.Face()
```

Where f is the name of a given face to be created.

To append vertices (maximum 4) to a face, use the line

```
f.v.append(obj.verts[x])
```

Where x is the order of the vertex around the perimeter (beginning with 0).

Let us mix the previous vertex assigning conventions in with our example of a creation and appendage of a face to more efficiently demonstrate the degree of flexibility had by the command. This code adds a square to Obj.

```
v=NMesh.Vert(1.0,1.0,0.0)
obj.verts.append(v)
v=NMesh.Vert(1.0,-1.0,0.0)
obj.verts.append(v)
a=NMesh.Vert(-1.0,-1.0,0.0)
b=NMesh.Vert(-1.0,1.0,0.0)
obj.verts.append(a)
obj.verts.append(b)
f.v.append(obj.verts[0])
f.v.append(obj.verts[1])
f.v.append(obj.verts[2])
f.v.append(obj.verts[3])
obj.faces.append(f)
```

## Objects

All of this code hasn't so much modified or created a mesh. It's more like a blueprint for the changes. To actually construct the new object, insert the following line of code at the end of the blueprint:

```
NMesh.PutRaw(obj, "Square", 1)
```

The first argument is the name of the mesh to be used to create "square", the second argument, which is the name of the object. This can be anything you like. The third argument is the renormalization boolean. When set to 1 (true), this forces Blender to recalculate the mesh's normals before generating the object in 3D space. Finally, use this piece of code after all of your object creation and other visual modification scripts to tell Blender to refresh the screen:

```
Blender.Redraw()
```

## Iterative Loops

Possibly the most important advantage of procedural object creation is the work it saves you when creating something with any sort of replicable quality. Here we will give the basic format of a loop in Python, then show some example scripts you can build on. The basic loop is thus:

```
for i in range(a,b,c):
    Four (more) spaces indentation marking all code contained in the
    loop
    Same indententation level as "for i in range" line to exit loop
```

Where i is the iteration number, a is the start value of the iterations, b is the end value of the iterations, and c is the increment. Entering in c is not necessary if the increment is 1. If a is 0, b is 100, and c is 1, it will run the iterative loop 100 times, starting with the 1st iteration, where i=0. If c is 2, it will run the iterative loop, adding two to i each loop, until i=100.(50 times - for iteration 1 i=0, iteration 2 i=2, and iteration 50 i=98). Note that when i=b (end value of the iterations), the loop will end prior to executing the loop code.

If you nest loops, you can perform such tasks as placing a row of ten vertices in with one of the coordinates set to i, so that when i makes another iteration the nested loop makes another row of ten vertices in the new position given by i.

*A loop that produces a vertex at the origin, (1,1,1), and (2,2,2)*

```
for i in range(0,3,1):
    v=NMesh.Vert(i,i,i)
    obj.verts.append(v)
```

*A square-producing loop:*

```
for i in range(0,2,1):
    for j in range(0,2,1):
        v=NMesh.Vert(j,0,i)
        obj.verts.append(v)
```

*A two-square-producing loop:*

```
for i in range(0,2,1):
    for j in range(0,2,1):
        for k in range(0,2,1):
            v=NMesh.Vert(j,k,i)
            obj.verts.append(v)
```

## Links

- Blender tools:
  - <http://www.makehuman.org/>
  - <http://www.reocities.com/blenderdungeon/lSystem/> (*broken link - needs update*)
- 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/>

# Blender 3D: Noob to Pro/Advanced Tutorials/ Python Scripting/Scripts for modifying meshes

---

(to be written)

Also see saltshaker<sup>[1]</sup> a basic but functional python script for blender, page includes details of how it was made.  
[http://jmsoler.free.fr/didacticiel/blender/tutor/python\\_script01\\_en.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/python_script01_en.htm) is a good one for learning about mesh creation.

## References

[1] <http://saltshaker.sourceforge.net/>

# Blender 3D: Noob to Pro/Advanced Tutorials/ Python Scripting/Creating a GUI for your script

---

It is very easy to create a GUI for your script, and that way make it easy to change aspects of it for everyone.

The command to create a Graphical User Interface (GUI) is:

```
Blender.Draw.Register(draw,event,button)
```

This command registers the functions:

- draw - to draw the GUI
- event - to action mouse and key presses
- button - to action GUI button presses

**However, this command will NOT work by itself !!!** You first need to define these 3 functions.

First we will import Blender's library of built in functions:

```
import Blender
```

**Next, we will define the draw function.**

```
def draw():
```

Inside this function we will draw the GUI. Here is an example of a drawing function we can use. It will clear the current window.

```
Blender.BGL.glClear(Blender.BGL.GL_COLOR_BUFFER_BIT)
```

And the next command will draw a button. Note that the first number in the command, '1' identifies the button as **button 1**. We will refer to this button later.

```
Blender.Draw.Toggle("Clear origin",1,10,20,100,20,0,"Tooltip")
```

**Next, we will define the event function.** The code of a key pressed on the keyboard is passed into the function as the variable **evt**.

```
def event(evt,val):
```

Now we will test to see if the escape key is pressed:

```
if evt == Blender.Draw.ESCKEY:
```

If it is pressed, exit the script, and return from the function:

```
    Blender.Draw.Exit()
    return
```

**Next, we will define the button function.** This function will perform an action if the button is pressed.

```
def button(evt):
```

Now test the variable **evt** which holds the button number that we previously identified.

```
if evt == 1:
```

If it is pressed, we will move the selected object in the 3d window back to the centre and redraw the screen:

```
    Blender.Scene.GetCurrent().getActiveObject().loc = (0, 0, 0)
    Blender.Window.Redraw()
```

**Lastly, we can create the Graphical User Interface** by typing the command:

```
Blender.Draw.Register(draw, event, button)
```

That's it !!! To enter the script yourself, type the following into the Text Editor window in Blender, and then press **alt p** to execute the script. Here's the entire script. Everything after the hash # is a comment and can be left out.

```
import Blender # This will import the library of blender functions we
will use

def draw():      # Define the draw function (which draws your GUI).
    Blender.BGL.glClear(Blender.BGL.GL_COLOR_BUFFER_BIT) # This
clears the window
        # Add here drawing commands to draw your GUI, for example:
    Blender.Draw.Toggle("Clear origin",1,10,20,100,20,0,"Tooltip")
        # The line above will draw a toggle button.
        # Note the first number, '1' means this is button number 1

def event(evt,val): # Define mouse and keyboard press events
    if evt == Blender.Draw.ESCKEY: # Example if esc key pressed
        Blender.Draw.Exit()      # then exit script
        return                   # return from the function

def button(evt):      # Define what to do if a button is pressed, for
example:
    if evt == 1: # If button '1' is pressed, set active object to
centre:
        Blender.Scene.GetCurrent().getActiveObject().loc = (0, 0, 0)
        Blender.Window.Redraw() # This will redraw the 3d window.

# You can now run the Graphical User Interface by typing the command:

Blender.Draw.Register(draw, event, button)
```

```
# End of script
```

Noob Note: When I tried to run that script it came up with an error: Attribute Error: 'Scene' object has no attribute 'getActiveObject'. I changed "Blender.Scene.GetCurrent().getActiveObject().loc = (0,0,0)" to "Blender.Scene.GetCurrent().**objects.active**.loc = (0,0,0)" and it worked.

2.5 Update: no Blender module currently exists for Blender 2.5 A0, they could even be removing it altogether.

---

# Unit Four: Advanced Modeling

---

## Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Modeling/index

---

### Advanced Modeling

We have learned quite a few useful techniques that we can now use to make something really cool or fun. But there are plenty of little tricks hidden within Blender that will allow us to make very realistic or intricate 3D graphics with a much lower amount of memory used. It usually takes an extra application or two, and the more advanced the program, the better; however, in the next few tutorials the *Paint* program--or its equivalent on another OS--will be just fine for our purposes.

The purpose of these modeling tutorials will be to lower the poly count on your models--this could range from no effect to cutting out 99.99% of your original faces, and still being able to keep the realistic look you were going for. Make sure you have read over the other tutorials in this book or at least are very familiar with Blender, so you can better understand how to work these modeling techniques.

## Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Modeling/Modeling

---

### Making Better Models

This section will deal with the actual modeling of objects and how to make better, more detailed objects much faster and more efficient. It will also deal with making more accurate models, whether it is cartoonish or realistic. Either way, this tutorial will deal with making better models and using the more advanced modeling systems in Blender.

# Blender 3D: Noob to Pro/HDRI

## Introduction

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 <sup>[1]</sup> 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 usefulness of HDRI, please read the wikipedia link.

Also, visit Paul Debevec's website <sup>[2]</sup> if you've got some more time to spare.

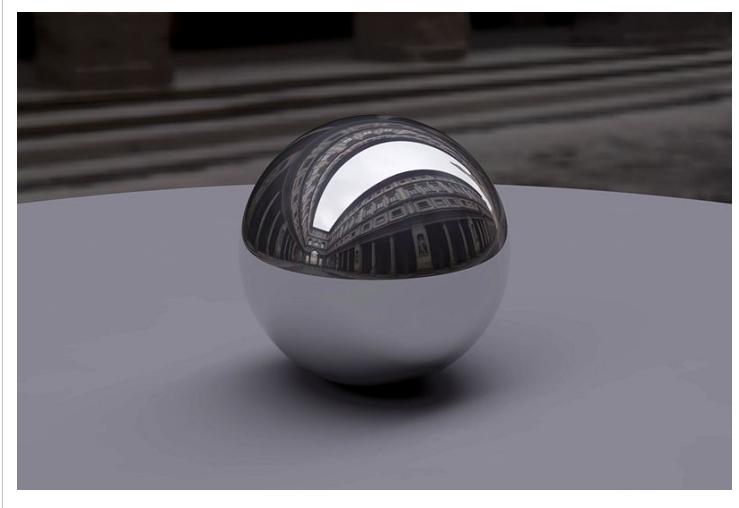
To sum up the excitement of HDR CG, think of it like the hype of the next-generation videogames that are about to come out, except set the stage for 1996 instead of 2006. Paul Debevec pioneered parallax mapping, HDR lighting, image-based modeling, his latest work includes some even more amazing technologies, and for the record he's my hero too.

To use HDRI images for 3D rendering, you need something called a light probe...

## Definitions

### HDRI

HDRI stands for **High Dynamic Range Imaging**, and is basically an image format that contains from the deepest shadow up to the brightest highlight information. While an 'ordinary' digital image contains only 8 bits of information per color (red, green, blue) which gives you 256 gradations per color, the HDR image format stores the 3 colors with floating point accuracy. Thus the 'depth' from dark to light per color is virtually unlimited. Using HDR images in a 3D environment will result in very realistic and convincing shadows, highlights and reflections. This is very important for realistic emulation of chrome for example.



### Light Probe

Light Probe is a HDR image containing 360 by 360 degrees image information. In other words : it's a 360 degree spherical panorama image, not only looking around the horizon, but also up and down. Thus a Light Probe image contains all visible information as can be seen from a specific point, wherever you turn your head.

## Usage

Given that a Light Probe image is an 'all around' image with a high dynamic range, it's the perfect solution for your 'world' background, especially for a 3D animation.

## Quick Tutorial (*for experienced blendies*)

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. (There are even more at <http://blenderartists.org/forum/showthread.php?t=24038> ). I will use the St. Peter's Basilica probe, but any other HDR image will do just fine.

To apply the HDRI environment to your scene:

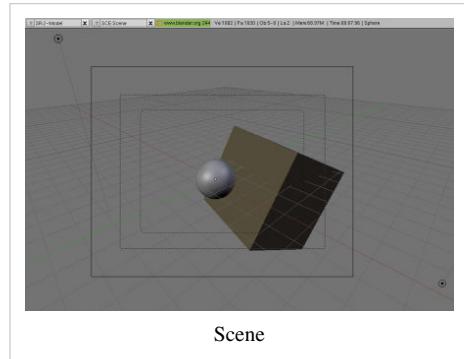
- Go to the shading settings (press F5) and click the World button.
- Enable "Real" to force the horizon to stay still, as opposed to follow the camera
- If you're rendering with Blender Internal (ver. 2.49), go to "Ambient Occlusion" tab and activate "Ambient Occlusion", then activate "Sky Texture". Skip this step if you're using external renderers.
- 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.
- If you're using YafRay or other external renderers, you need to turn on Global Illumination and to set the quality to something other than "none".

## Step-by-step Tutorial

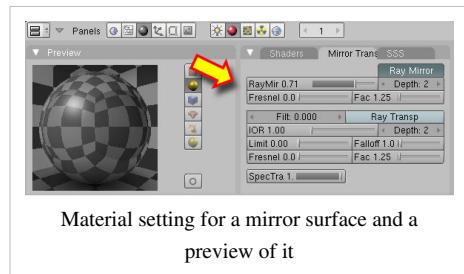
### Build a simple scene

To see the advantage of using a 360 by 360 world image, the simplest example to demonstrate this is a scene with a mirrored sphere.

1. Add a sphere and a cube to your scene and place them in a bit of an interesting position. (note that I added a second lamp to light up the shadow part of the cube)
2. Perhaps give the cube a different color than the default grey.

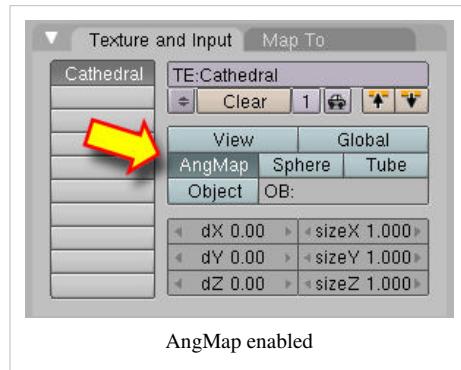


1. Give the sphere a mirrored material : go to the **Shading -> Material** panel (F5) and find the **Mirror Trans** buttons.
2. Check that **Ray Mirror** button is pressed. If it is not, check it.
3. Set the **RayMir** value to a value of 0.5 or higher. Your preview should show the reflection of the checkboard environment.



## Render with HDR (Blender Internal v2.49)

1. Download a HDR image (see Paul Debevec's website)
2. Go to the shading settings (press F5) and click the World button.
1. Go to "Ambient Occlusion" tab and activate "Ambient Occlusion", then make sure "Raytrace" is chosen as gather method, and activate "Sky Texture"
2. Adjust quality settings:
  1. Increasing number of samples reduces noise, but increases render times
  2. Adaptive QMC is faster, but generates more noise than Constant QMC
3. In the "Texture and Input" tab, click "Add New" and "Angmap"
4. Then go to the "Map To" tab and deactivate "Blend" and activate "Hori".
5. Now go to the Texture settings (press F6) and change the "Texture Type" to "Image".
6. Click the "Load Image" button and locate your HDR image.



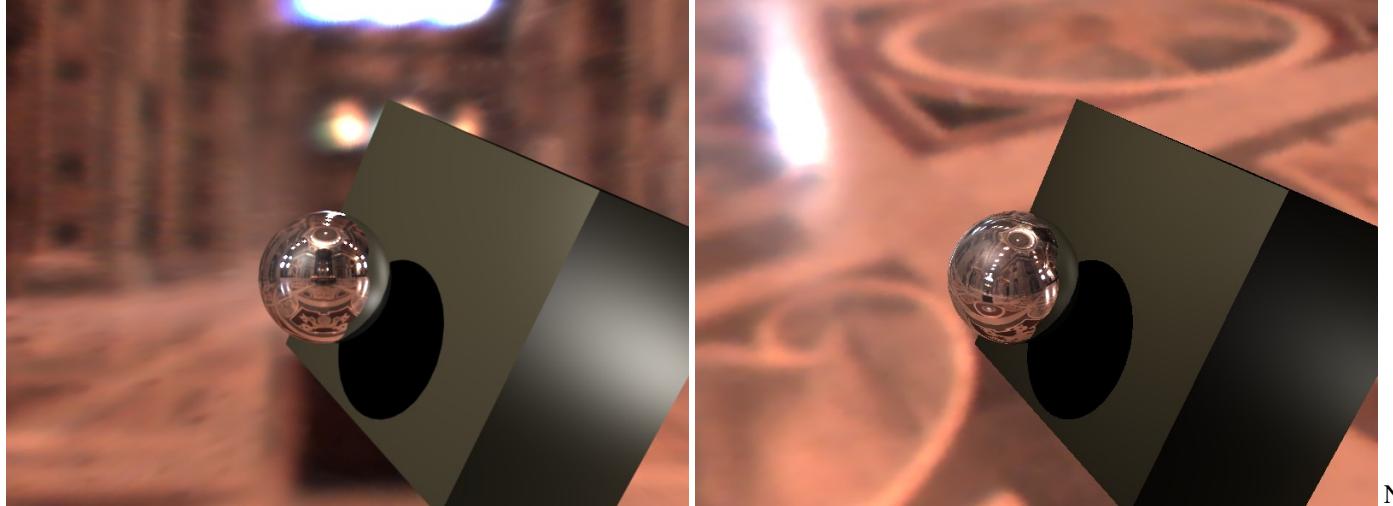
## Render with YafRay

1. Download a HDR image (see Paul Debevec's website).
2. Go to the shading settings (press F5) and click the World button.
3. In the "Texture and Input" tab, click "Add New" and "Angmap". **Note that the Angmap button is the important thing to tell Blender that this file is a Light Probe file !**
4. Then go to the "Map To" tab and deactivate "Blend" and activate "Hori".
5. Now go to the Texture settings (press F6) and change the "Texture Type" to "Image".
6. Click the "Load Image" button and locate your HDR image.

(optional step, as it was not needed for my setup :)

5. Press F10 and change the "Blender Internal" to "YafRay". You need to turn on Global Illumination and to set the quality to something other than "none". Note that the YafRay renderer does not come standard with the Blender installation. You need to download and install this separately.

Your result should look like this: (Rendering: left with Blender, right with YafRay) Click for larger version



that the reflecting ball reflects the whole interior from every angle, even though we added just a single image to the World settings !

## References

- [1] [http://en.wikipedia.org/wiki/High\\_dynamic\\_range\\_imaging](http://en.wikipedia.org/wiki/High_dynamic_range_imaging)
- [2] <http://www.debevec.org>

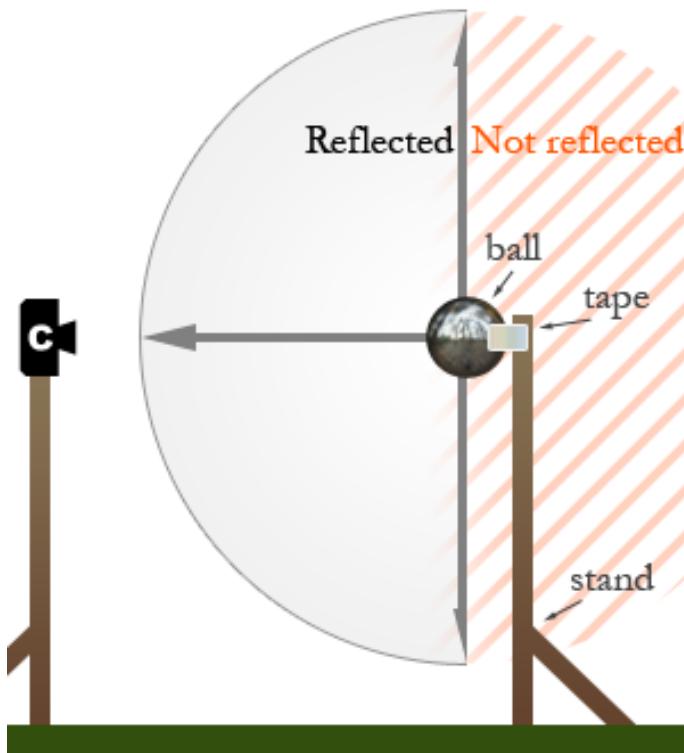
# Blender 3D: Noob to Pro/Creating a Light Probe

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, preferably 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 recommends 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 video camera as was done for 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 a series of images with varying exposure times.
2. Taking 2 sets of pictures, each offset by 90 degrees, will enable you to get better coverage of the background and eliminate the reflection of the camera taking the picture.

Good article <sup>[1]</sup>

## References

[1] <http://gl.ict.usc.edu/HDRShop/tutorial/tutorial5.html>

# Blender 3D: Noob to Pro/Making Landscapes with heightmaps

This tutorial will show you how to make advanced terrain such as mountains using Blender and gimp or any other image editing software. Blender has the ability to use height maps to create meshes. Height maps are black and white images with white representing the highest point and black the lowest.

## Creating the heightmap image

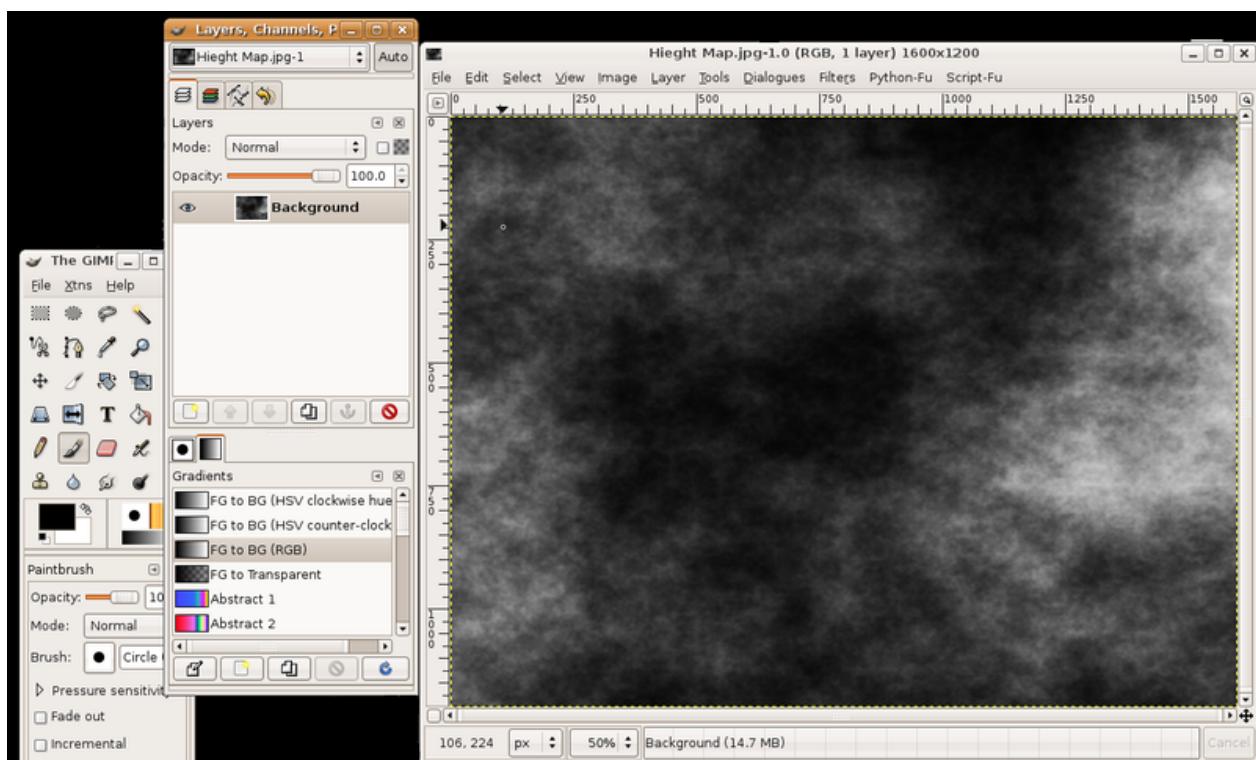
**Note:** This entire Tutorial presumes that you are already familiar with other Editing Software, such as GIMP... and already know how to create "textures" with that software. However, you do not have to use another program to obtain a texture. At the end of the next section, they show you how to generate a random 'cloud' texture which you can use directly. Do not worry about GIMP or Photoshop. Just skip the rest of this section and most of the next one.

To begin with, open your image editing software. This part applies to Gimp only (downloadable at <http://gimp.org>), if you use another program you will have to do it another way.

First use the "New" menu option to create an image 1600 wide by 1200 tall. Go to **Filters** → *Render* → *Clouds* → *Plasma* (In Photoshop, this is Filter > Render > Clouds). For this example just use the default settings, it doesn't really matter. Click OK. You should now have a nice colorful image. We don't want that, we want it in black and white. Make sure your Gradient is set to "FG to BG(RGB)"(this is the default anyway) and that your foreground color is black and the background color is white. Go to **Colors** → *Map* → *Gradient Map*. This will convert it to black and white for you.

Save your image as PNG or JPEG.

It should look something like this



## Create grid and add the image as texture

Open Blender and delete the default cube.

Add a grid (**SPACE** → *Add* → *Mesh* → *Grid*) with resolutions 32 and 32 from top view. Do not scale the grid just yet, zoom instead if you wish to take a closer look at your grid. It will be explained later why you shouldn't scale now.

**Note:** When I followed this tutorial it didn't work - the vertices in the grid were moving sideways rather than vertically. It turns out I needed to **TAB** into object mode after creating the grid, then hit **CTRL+AKEY** to reset the transformation, then **TAB** back into edit mode. If someone could explain why that was necessary, or what it does exactly, that would be helpful. *Response:* You should create the basic shape and outline in object mode, then edit the details in edit mode, kinda like the names imply.

Click **F5** to go to the *Shading panel*. By default the *Material buttons* button should be selected (button with red ball), but if it isn't then click on it. If this window has a panel named *Preview* and it is empty then look to the right for the *Links and Pipeline* panel. Here you should find an *Add New* button beneath *Link to Object*, click on it. 3 more panels should appear when you do this. You should find the *Texture buttons* button, it is just to the right of the *Material button* and is black & yellow. You may also just hit **F6**. Now you should get two new panels titled *Preview* and *Texture*, in the *Texture* panel you'll find an *Add new* button, click it. Where the button just was it should now be a text field saying something like "TE:Tex.001", click in the text field and write "height" instead. Look a bit to the right and down for a dropdown menu allowing you to choose *Texture Type*, click it and select *Image*. Two more panels will appear and the rightmost should be *Image* with a *Load* button.

**Note:** The *Texture buttons* button is not part of the windows that pop up after you click *Add New* button, but it directly to the right of the *Material button* that is a red ball.

**Note:** If you can't find the *Texture Type* menu, press **F6**. Once you press **F6**, or the *Texture buttons* button, you'll see another panel, also called *Texture*. This one, however, shows a *Texture Type* menu after you select *Add new*. Having two very similar panels can be a nasty trap.

Before you load the image take a look at the *Map Image* panel. About in the middle there should be 5 buttons, *Extend*, *Clip*, *ClipCube*, *Repeat* and *Checker*. Select *Clip* or your heightmap may wrap around the grid (if this doesn't make sense then do this tutorial twice, one where you keep *Repeat* selected and one without. In that case make sure there's some bright white spots in your height map on each edge).

Click the *Load* button and load the heightmap you just created in your favourite image editing software.

**If you're going for the random cloud-type image, though, you can make it from within Blender itself - when you add a texture, in the menu where you select 'image,' before actually selecting it, pick 'clouds' instead, and play with the settings.)**



Click the *Material buttons* button again (the red ball button) and look in the *Preview* panel. If this just shows a black ball then click on the uppermost button just to the right of the black ball.



## Use texture as heightmap

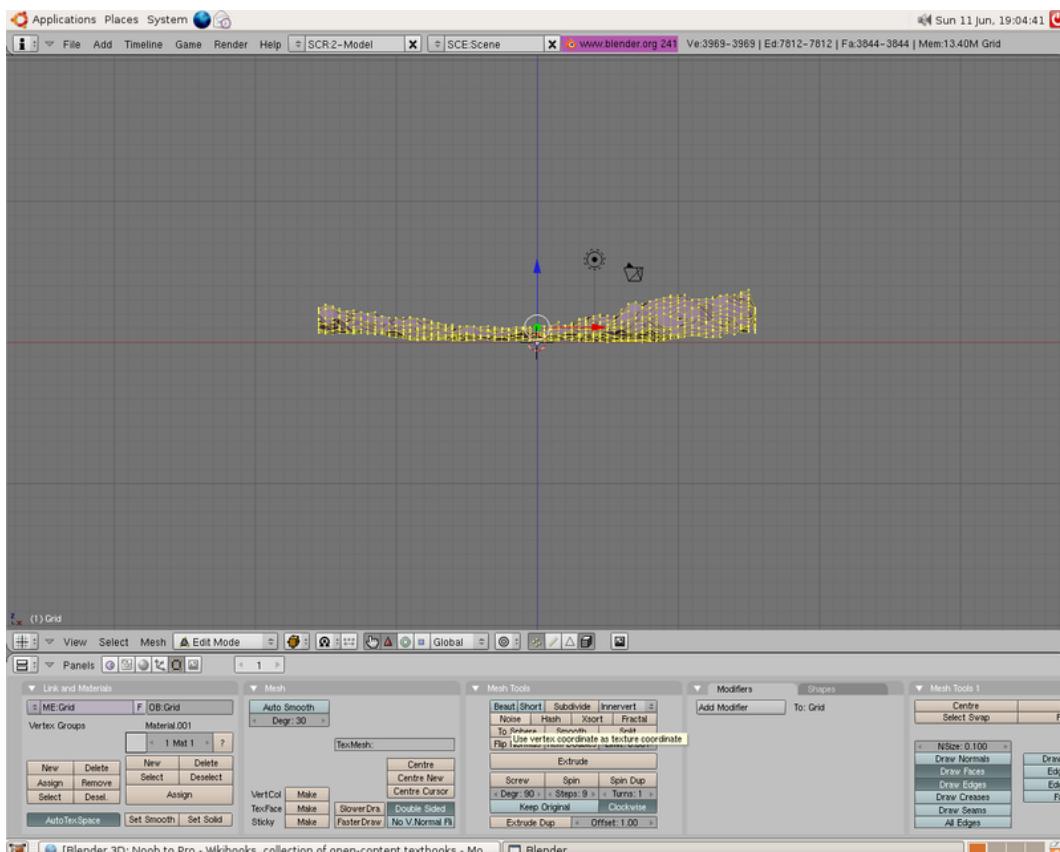
Look to the top right of the rightmost panel, the window should have 3 panes titled *Texture*, *Map Input* and *Map To*. Click on the *Map To* pane. A whole lot of buttons will appear:

- Turn off the *Col* button. "Col" stands for "color," and we don't want our plane to be *colored* by this texture.
- All of these buttons should be turned off: we're not going to be using this texture to "Map To" anything. Instead, we'll be using it to deform the actual geometry.

Go to the *Edit window* (**F9**) and enter edit mode (**TAB**) if you're in *Object Mode*. Go to either front or side view so you can see what you are doing. Click the *Noise* button in the *Mesh Tools*-panel (it's located in the top left corner of this panel) and your grid should start to change shape.

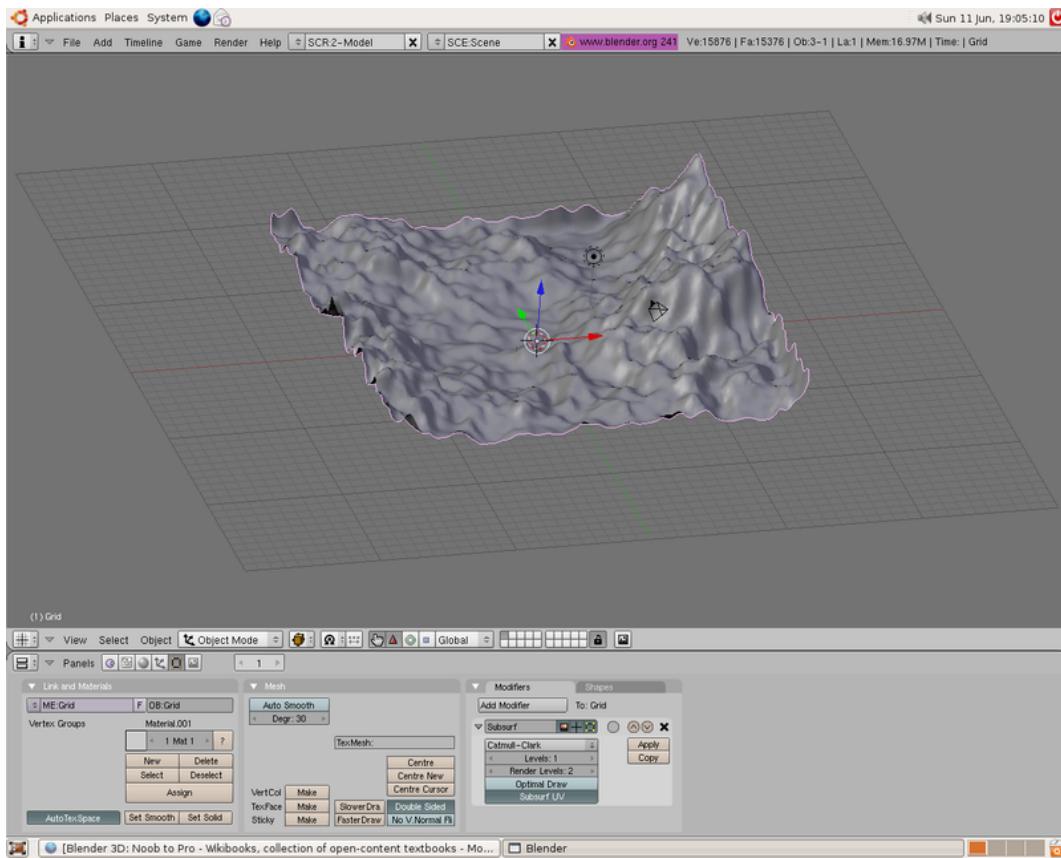
**Note:** make sure all your vertices are selected with AKEY or whatever you want to use to select. If not selected, *Noise* won't do anything!

- (*While you could, theoretically, use this to your advantage ... intentionally de-selecting vertices that you don't want to influence ... in practice that's quite awkward if for some reason you have to do it again. It's much easier to leave any such areas, that you don't want to influence, "completely white" in the image.*)



Continue clicking the button until the terrain has reached the height you want and voilà, you have just made terrain using heightmaps. You may want to subsurf the shape to get a smoother effect. If you wish to scale your new landscape then now is the time. There seems to be a bug or gotcha causing your heightmap to be tiled on your grid

rather than resized if you scale the grid before applying the heightmap. Hopefully someone familiar with blender will clarify how this should be done properly, but until then this method will work.



#### Tips:

- 1.If you want a more jagged landscape then adjust the contrast on the image editing software
- 2.Once you've finished using the *Noise* button in the *Mesh Tools* panel as shown above, the texture is no longer needed: the effect it has made on the geometry of the object is permanent. If texture-resources are tight, you can remove it now. But it's advisable to keep it, unchanged, in case you need to re-do your work at some time in the future. If you want to use other kinds of texture-maps (*Color*, *Normal* maps, and so-forth), you can use the landscape-image as a handy reference (say in a background-layer in your painting program) for correct placement of these features, but you should not alter the landscape-image itself).
- 3.Height maps can also be used to make city terrain. Draw white squares on a black background. If you do this make sure you make the squares perfectly parallel to the sides of the image. This prevents jagged edges to buildings. You can also use gray squares to adjust the buildings to different heights.
- 4.The "Noise" mesh-editing button displaces vertices in the object's Z-Axis and negative Z-Axis only. To deform your mesh's other dimensions, simply rotate your object and "apply rotation," or rotate the vertices in edit mode, and apply Noise. Then, rotate it back again to get your original orientation.
- 5.The "Noise" button permanently modifies your mesh according to the material texture. Each click adds onto the current mesh. For a temporary effect, map the texture to *Disp(lacement)* for a render-time effect (*and do not click the "Noise" button at all*). In object/edit mode your object will appear normal, but will render deformed.
- 6.*Note:* The "Noise" mesh-editing button, as used here, is not the same as the "Noise" texture that is referred to in other tutorials!

# Blender 3D: Noob to Pro/How to Do Procedural Landscape Modeling

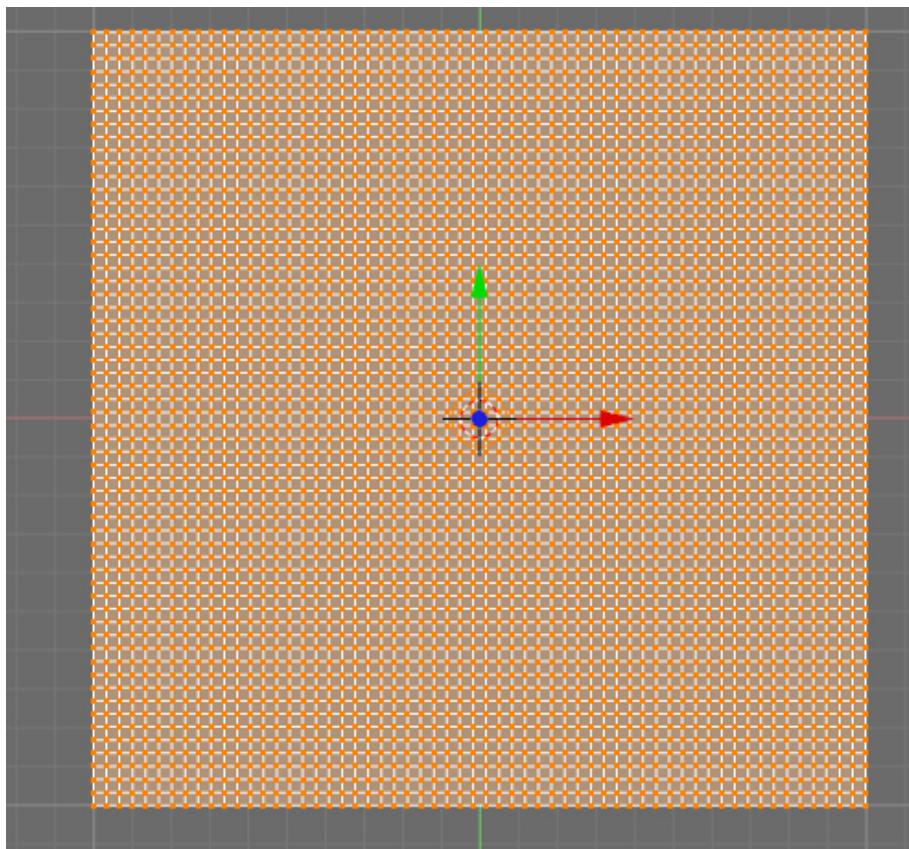
## Mountains

I was playing around in Blender several months ago and messing with cloud textures when I found a great way to make landscapes entirely within Blender (without height maps from The Gimp or some other application) and decided to make a tutorial (yes, it does use the noise tool).

### Making the Mesh

To start, open up Blender and with the cube selected, press the **XKEY** and click Erase Selected Objects.

Add a grid (**SPACE--> Add--> Mesh--> Grid**), using whatever numbers you like, remembering that bigger numbers means more vertices. For this tutorial, I'll use 60 for x and y resolution. Scale the grid up, but only a little bit (I went up to 1.3) as the texture won't make very tall landscape with a very big grid. Press the **F5** key to go to the Materials tab, then add a new material.



## Displace and Shade

Now press the **F6** key to go the Textures panel, and add a cloud texture in the very first slot (slot 0).

After Pressing F6, look at the windows on the bottom. To the left is the "preview" window and next to that (to the right) is a *Texture* window with an *Add* button... click that and then go to the little pull down menu that shows up just below and to the right. From there, find and select "Clouds"... and now, continue!

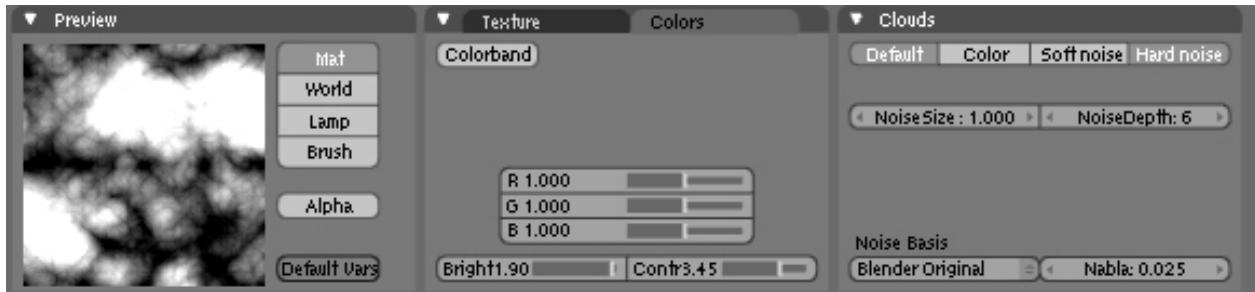
You may be wondering how we are going to make landscape with a cloud texture, when you could just use the Gimp height map plugin, Terragen, or the A.N.T. modeling script right in Blender. Well, we are going to make the landscape with a **height map**. The color determines the height: white = very tall; grey = halfway tall; black = no height. This makes the cloud texture a very good candidate for land.

The next thing we are going to do is adjust the size. You can set the noise size to whatever you want, but I want to have several bigger pieces of landscape, instead of a bunch of little ones so I'm setting noise size to 1.00 and noise depth to 6, as I want a high level of detail. (The *Preview* immediately shows you the effect of your "tweaks" as you make them.)

**Noob Note:** You can hold shift and click the number next to noise size then just type 1 and press enter instead of using the arrows. The same applies for NoiseDepth, almost every other setting in blender changes like this as well.

At the top, we can select *Hard noise* or *Soft noise*. Each gives a different landscape, so experiment with them. Next, go to the *Colors* tab (it's right next to the texture button by default). Adjust the brightness and contrast as required. For "Hard noise," try a brightness of 1.9 and a contrast of 3.45. For "Soft noise," a brightness around 1.0 to 1.2 should be fine.

**Noob Note:** The above noob note applies to the brightness setting as well!



Jump back to material buttons with **F5** and go into the "Map To" panel. Turn off *col* and press *nor* once.

**Noob Note:** I had a difficult time finding the "Map To" so if you are too... This is one of the several small Windows that opens at the bottom of your screen after pressing **F5** and for me, it was on the far, far right, off my screen. I had to use my mouse wheel to scroll the Windows to the side to bring it into view and found it as a Tab on a Window with two other tabs. "Col" and "Nor" are buttons in the *Map To* Tab.

- *What the buttons are for:* The "Map To" settings determine which various attributes of the material will be affected by our "cloudy texture."
  - "Col" refers to "color." We turn it off here, because we don't want our surface to *look* "cloudy."
  - "Nor" refers to "surface normals." This is the angle at which light seems to reflect from the surface. (The so-called "bump map.")
  - Experiment with the other "Map To" buttons, as well! For instance, "Disp" (for "displacement) actually causes a map to deform the geometry of a surface... but doing so only during rendering.

In this tutorial, as you will see, we are going to *combine* several techniques to produce a very rocky surface. Here, we are changing the way that the surface reflects light. Then, we'll actually deform the surface geometry.

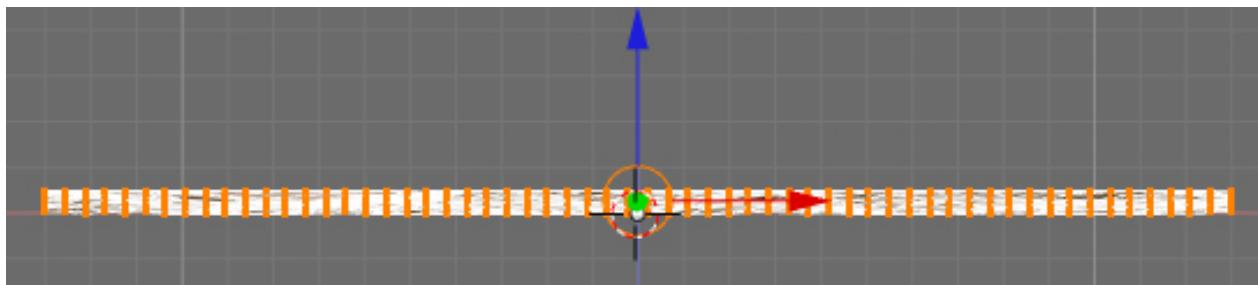
Turn the *nor* value all the way up, to 25 (that will give it a nice rocky look). Go into the *Shaders* tab and then change the "Diffuse Shader" (the top left selector, defaulting to "Lambert") to "Oren-Nayer," and the "Specular Shader" (the 2nd selector, defaulting to "CookTorr") to "Blinn." Adjust the *Rough* value in Oren-Nayer to 1.5 (rock is really rough). Also, set the *Spec* value to 0.01 and *Hard* to 25. Now we are ready to make some mountains!

- *Trivia note:* In computer graphics, a "shader" is a mathematical function, and they're customarily named after the clever people who invented them. These functions determine exactly how the cloudy-texture will "map to" the attributes (*e.g.* "Nor"...) that we selected.



## Making Mountains

Press **TAB** to go into edit mode and press the **AKEY** twice, in case you had any specific vertices selected. Go into the front view and press the "Noise" button (located under mesh tools in **F9**). You should see the vertices jump up a good bit.



If not, you may have scaled your grid up too much, so make it somewhat smaller. Depending on how high you want your mountains, press the *Noise* button to your required amount, though I pressed it 8 more times. The mesh doesn't look very good right now, as it is not only blocky, but it is missing that last bit of random detail.

- **Important note!** The *mesh-editing button* named "Noise" causes a permanent change to the geometry of the object in the "Z"-axis, as provided by the texture. (The vertices actually *moved*, and their new position is permanent.)
  - There is also a *procedural texture* named "Noise," but that is just pure coincidence. The two are unrelated, both in what they do and in how they work.

If you are not in edit mode, go into it and select all the vertices. Press *Fractal* (in the same row as *Noise* under mesh tools) and enter a value of 15 in the random factor. Hopefully, your system can manage all these vertices: if not, then don't do any fractal subdivide. If you want to, you can use a smaller value in the fractal box and do it several more times for more displacement. Be careful, though: too much randomness will cause vertices to separate, causing tears in the mesh that you will have to fix by hand.

Finally, tab out of edit mode and press *Set Smooth*. Then, if you want to make the mesh even more smoother, add a *Subsurf* modifier, with whatever level you want. (I'm going with 1 for faster render times.)

If you desire, you can scale up the mesh however big you want though for close-ups, you may want another level of subsurf or another subdivide. Then, apply colors, set up the lights, and render! A word to the wise, however: this process creates a huge file (about 986kb).

## Discussion

You have just made mountains all in Blender, without having to generate height maps in The Gimp, or any other program. The next step is to go beyond the tutorial. Try using different noise sizes and noise basis, and even try using other textures like musgrave and marble. Or try using two textures and see the conclusion. The possibilities are almost infinite! Have fun.

**Observe** how the technique illustrated *this* tutorial is "the same, yet different" from the one demonstrated in the tutorial for "height maps":

- A *procedural texture* was used to supply changes to the *Normal Mapping* of the object at render time. (This is the angle at which the object reflects light.) (The "height mapping" tutorial did not do this.)
- Then, the *mesh-editing button* (also...) named "Noise" was used to deform the actual geometry of the mesh. (This is the same technique used in the "Height Map" tutorial, although a different kind of texture .. an image .. was used.)
- The texture, once used for the height-mapping (deformations) was left in place, still mapped to the surface "Nor"mals. The result will be a *very* mountainous landscape.

Yep! The techniques shown in all of these terrain-modeling tutorials *can* be combined! And, they frequently are. Very often the most satisfactory results are obtained by combining several different techniques. You could have used an image-map (from the "Height Mapping" tutorial) to deform the geometry, then used the "noisy Normal-mapping" idea from the first part of this tutorial, and even used smoothing. All at the same time. The choice is yours.

When using "Map To," remember that a single material can have several different textures applied to it, each one mapped to the same or to different attributes. This mapping can even be *animated*.

Experiment! The sky's the limit!

# Blender 3D: Noob to Pro/Landscape Modeling I: Basic Terrain

---

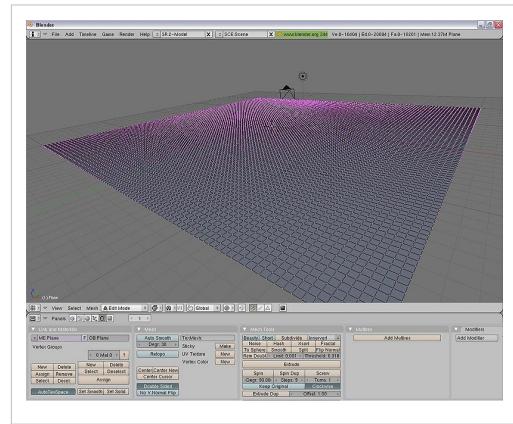
I worked with a group to make a small animation for class and I was responsible for the environment modeling. I really liked the results of the process I used, so I decided to share it with others. By the end of this tutorial you'll have the know-how to create your own flexible, realistic terrain utilizing multiple textures for different ground types. This tutorial assumes that you have the very basic understanding of using Blender (how to add/remove a mesh, how to change views, etc...) I use the following textures in this demo. Feel free to use them if you don't have something on hand to use.



**NOTE:** This is not the actual texture I used in this demo. I couldn't find the original source for the one I used (though I know it was a public image), so to avoid licensing issues I replaced it with one that I did have licensing info on. Sorry.

## Creating the Canvas

The "canvas" that we'll use for our terrain is an evenly spaced grid of vertices. Open a new project, delete the default cube (doesn't everyone do this?:::(yes everyone does this!!!)), and add a grid mesh. Use whatever size you want. The more vertices you have, the more detailed and realistic your landscape will appear, but don't get crazy with it since we'll apply subsurfacing at the end to smooth it out. At the same time, you do want enough vertices to prevent sharp edges, and we'll need them in the second tutorial when we cover texture stenciling. So consider how big you want your landscape and try to find a reasonable balance. I like to start with 100x100. The grid will be pretty small, so scale the entire thing up, let's say by a factor of 20. This grid will be used to build our landscape by pulling hills and mountains out of it.



**Noob Note:** ...delete the default cube (doesn't everyone do this?). If you don't want to delete the default cube every time you are opening a new file just delete the cube once, so you have the blank window (only lamp and cam are left) in front. Then select *File - Save default settings* or hit *Ctrl+U*. And from now on you have always the blank window and no default cube when starting a new Project.

## Molding the Mountains

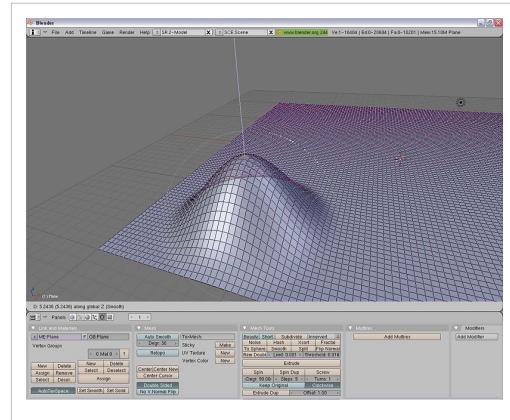
The key to making good mountains is using the proportional edit mode (**OKEY**) and constantly adjusting the radius of influence. If you've already gone through the Mountains Out Of Molehills <sup>[3]</sup> tutorial then this section will be familiar. One major difference is that in this tutorial I recommend rotating the 3D view around so you have a good view of all three axes instead of working in the front or sides view. Using the proportional editing tool affects multiple vertices, and it helps to see what effect your changes are having as you make them.

- Go ahead and turn on proportional editing, either by pressing **OKEY** or clicking on the grey ring on the 3D View header. You have to be in Edit Mode to select this option. Once proportional editing is enabled, the ring will appear orange and a new drop-down menu will appear next to it with different falloff styles. Select Smooth if it is not already selected.



- Select any random vertex and grab it (**GKEY**). You will see a ring around the vertex you are grabbing. This is the radius of influence, and only vertices inside this ring are affected by the transformation. If you are doing this in the orthographic view from the front, side, or top, then this will be obvious. But if you're at a view where you can see all three axes, then it may be less obvious.

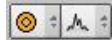
- Restrict movement to the z-axis (**ZKEY**) and translate the vertex upward. Throughout this entire process you ONLY want to translate along the z-axis. If you start moving vertices in the x or y directions, things become distorted and you get some nasty creases. Play around with adjusting the size of the radius of influence (**Mouse Wheel**) to get steeper or flatter hills.



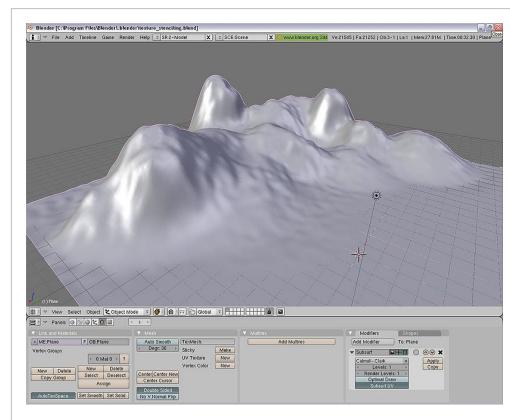
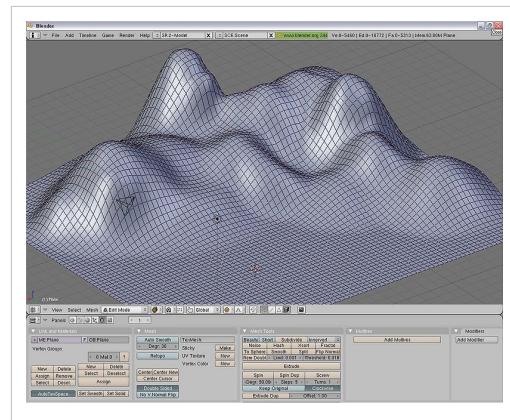
- Keep repeating this process with different size radii and different heights until you have a decent mountain range, but leave an area flat. We'll be using that spot later in the second tutorial. Don't be afraid to occasionally translate some vertices down instead of up to create depressions in the hills. Remember, variety is the spice of life. Very few things in nature are naturally geometric, so mix up your hills and especially make sure they overlap. How often do you see a nice, smooth hill all by itself in nature?

**Note:** You'll notice in my screenshot that I have reduced the size of my grid. For simplicity's sake, I didn't feel like filling an entire 100x100 grid with mountains since this can take some time.

- Well now, that's looking pretty good! Now, there's one problem with our hills so far. They're too smooth! Let's bumpify them a little. Change the falloff type from Smooth to Random.



- Select a single vertex and grab it (**GKEY**). We're still working on the z-axis only, to restrict your movement with the **ZKEY**. Now when you move the vertex up and down, all vertices in the radius of influence will also move but with a random falloff instead of smoothly. It only takes a little movement to get the effect we want, so something around 0.5-1.0 is enough. Mix up moving up and down with different vertices, again to add variety to the scene.
- Once you have your landscape the way you like it, add a subsurf modifier under the Editing tab (**F9**) and select Catmull-Clark. This will smooth out your terrain a little so that it's not too rough. Given the number of vertices you already have, it's not necessary to have a higher render value than 1 unless you just REALLY want it to be smooth, but I don't recommend it. Land is supposed to be bumpy and rocky, we just don't want sharp edges.



**Noob note:** If you're making mountains using Random Falloff and the peaks stick up too much: in Edit mode, select the points in the area around the peak using circle select, then press **WKEY** and click 'Smooth' until you're satisfied (or, in the Editing tab (**F9**), click the 'Smooth' button).

**Useful Tip** Proportional editing can be used on multiple vertices simultaneously. This is especially useful if you're trying to create a cliff face or a river bed. Use the box tool (**BKEY**) to select a group of vertices and then translate them. Keep in mind that the size of the radius of influence determines how many vertices around **EACH VERTEX** will be influenced. So suppose you have a 5 vertex radius, that means that 5 vertices all the way around your selected region will be influenced.

## Texturing the Terrain

Alright! We've got some pretty nice hills now! But there's still a few problems. Hills shouldn't be white, and hills shouldn't be SHINY! Let's dress them up a little, shall we?

- With your landscape selected, go to the Shading Panel (**F5**) and add a new material.
- Under the Shaders tab, drop the Specular value to 0.
- Go to the Texture tab (**F6**) and add a new texture.
- In the Texture Type drop down menu, select Image.
- Two new tabs will appear. In the Image tab, click Load and load a texture from file.
- In the Map Image tab, increase the Xrepeat and Yrepeat. Depending on the size of your terrain and the image that you use (please use something that tiles!), these values will vary. I've used 10 for each in this tutorial.

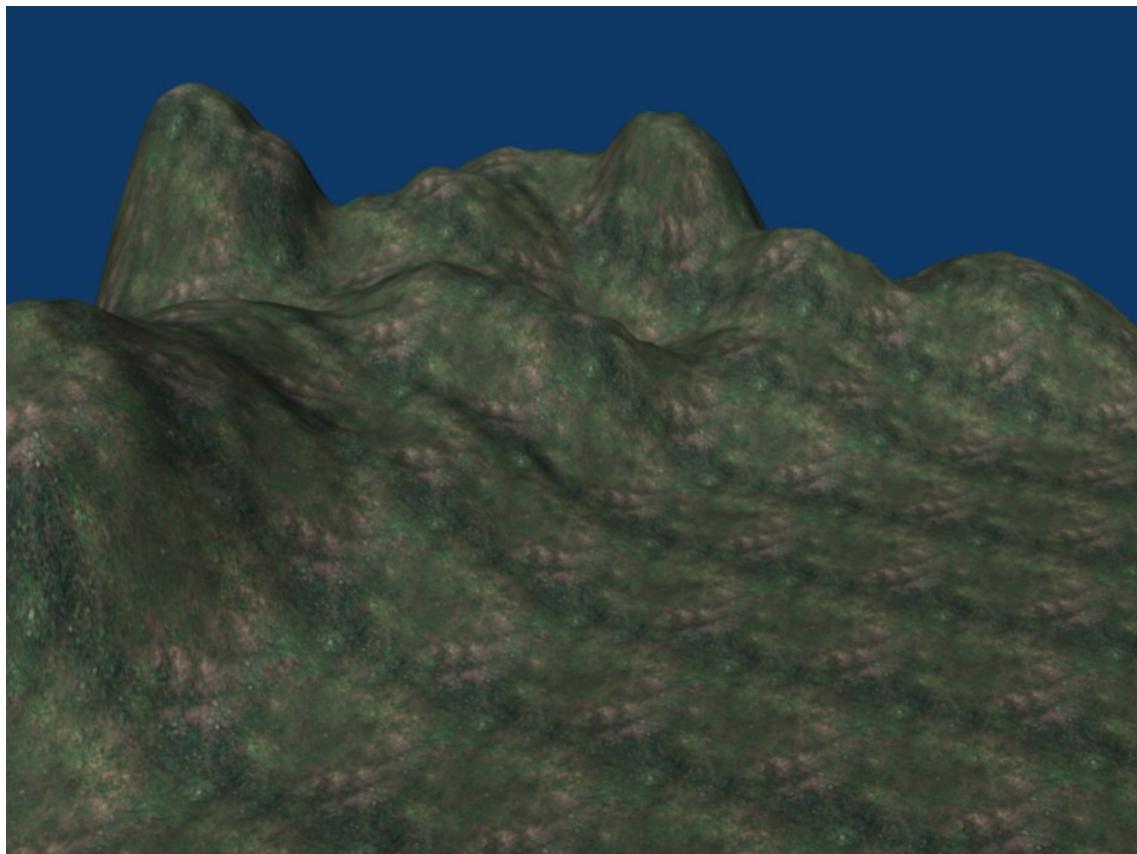


(**Noob note:** Render the scene (**F12**) to see the applied texture) (**Noob note:** A quicker way to do this is by using the render preview tool, 3d view window - render preview, **View--> Render Preview** or press **SHIFT--> PKEY**) (**Noob note:** you may also select shaded in the Viewport Shading menu next to where you select object mode or edit mode to see your texture on your mountains without rendering, however it does slow your computer down some which could make editing frustrating. So only do this if you want to see what your textures look like, then switch back to solid for more editing.)

- Finally, let's do something about the lighting. Go to Object Mode (**TAB**) if you're not already there and select the lamp. Choose the Shading Panel (**F5**) and then click the icon that looks like a light bulb to display the Lamp buttons. Change the lamp to a sun and up the energy to 1.5. You may also need to increase the distance if your terrain is large, or rotate it around if you don't like where it's pointing. The dashed line coming out of the light is the direction. Play around with different angles and energy values for the sun to get different times of day in your scene.

And there you have it! Your landscape is now textured with some nice grass to make it look a little more realistic. Play around with hill sizes and roughness if you're not satisfied with your landscape, but try not to be too picky. Nature shouldn't look too sculpted. Remember, you won't notice a change in texture unless you render your image.

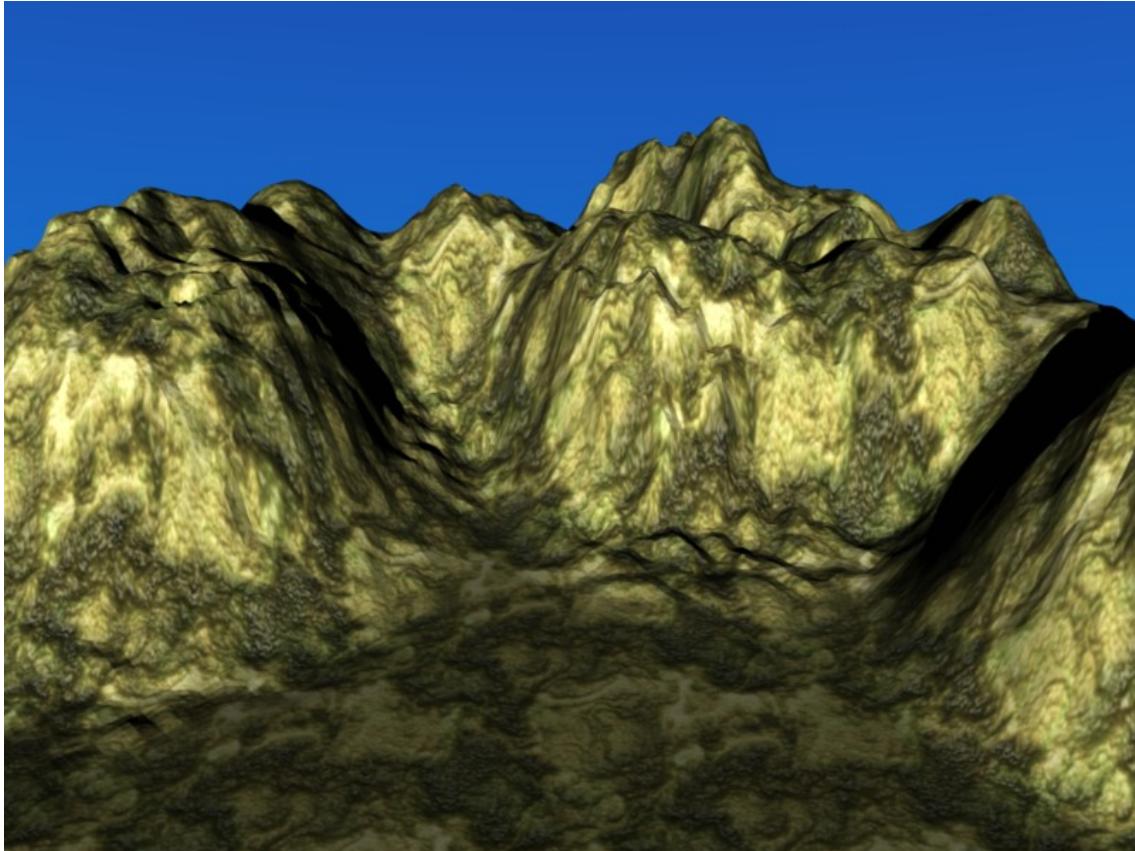
**NOTE** If you went with the suggested 100x100 grid, the rendering process could definitely take some time depending on your system, especially if you have ray tracing and shadows enabled. To boost the rendering speed, go to the Scene panel (**F10**) and under Rendering, disable the buttons that say **Shadow** and **Ray**. Also note that because we've created some hills, your camera may now be under the terrain. Switch to the camera view (**NUM0**) to see what your camera sees and move it if you need to.



Join us next time as we explore how to make the landscape look even better using texture stenciling! And please, if you have any issues with the tutorial, or feedback (positive or negative), drop it in the discussions. I'll try to keep an eye on it to see what people have to say. - Moohasha

## Reader Contributions

I'm mainly doing this so the hills look better and more realistic. Reading this will consume more time than doing it. I'm making everything clear for beginners. It will only take about 8 minutes more for your hills to end up like this:



1. First step to achieve it is to switch from the default lamp to Sun.
  1. You do this by clicking on the default and go to the shading tab (F5)
  2. Click on *Sun*.
  3. Go to *rotate manipulator mode* with Ctrl Alt R.
  4. Rotate the sun until the dotted line is in your desired position. That is where the main energy will go. It will differ if you want to achieve the different time of day. You can make the distance greater or less with *dist..* I kept mine at default 30.
2. Now let's make more realistic, paler sunlight.
  1. In the RGB slider, set R for 1, G for 1, and B for .848.
  2. Set energy for 1.63. This will all differ for different times of day, so set your east and west in your head, and the later into the day, the further the sun to the west and the more orange. For midday, keep the energy on 1.63 and put the sun right above your hills. I set mine for earlier in the morning.
  3. Now click on the picture of a globe to change the background.
  4. Click on the *blend* button for a more realistic looking sky. On the left of the World toolbar, that will be the color of lower down in the sky and the right sliders will be the color of the top of the sky. Naturally, the top should be darker blue than the bottom.
    1. It will differ for different times of the day. So you might want something different from this. But I set up mine for the morning using the settings below:
      - The left HoRGB sliders to 0.50, 0.68, and 1.
      - The right ZeRGB sliders to 0.11, 0.25, and 0.66.
5. Now for the texture. *This* won't differ for the times of day, but it will differ for what type of landscape you want.

1. I got the texture from here <sup>[1]</sup>. The best textures to use would be either Desert Canyon, Hill Country, Rough Terrain, Class M Planet, or Red Planet. Use Red Planet for a Rocky, Mars-like look.
2. I chose Hill Country <sup>[2]</sup>. Click on it, and I know it looks like snot, click on it again, and save the image.
3. Go back to Blender, go to Texture buttons (F6)
4. Click add new
5. Select *image* for the texture type
6. Go to the *image* toolbar, and upload the texture. Blender will only allow you to upload from the Blender documents, unless if you click on the up and down arrows on the top left of the screen and choose the place where you saved the texture (or picture).
7. After uploading, the X and Y repeats should be smallish, like 6x6, so the changes aren't noticeable. It will look ugly in the preview, but it will look nice after it's wrapped.
6. Render, and 1 minute later, voila! Those are nice looking hills!
7. Now boast in front of your friends!

**Noob Note:** You need to have your landscape selected if you are in object mode to change the texture, otherwise it will change the texture of the world.

## Reader Contributions 2

You can have a even better result if you use the texture to "bumpmap" the mountains.

1. Press **F5** until you are into Material Buttons
2. Select the *Map to* tab
3. Click on *Nor* once (the *Col* option must stay selected as well)
4. Slide the *Nor* slider to 5 or more (might differ depending on texture size and repeat options)
5. Render with **F12**

## Reader Contributions 3

Alternately from making the hill model manually, you can use a program like L3DT <sup>[3]</sup>. It uses various algorithms to generate very detailed and realistic terrain heightmaps. It also lets you edit them in a more intuitive way than Blender does. After you have L3DT make your height map, you can export it as a .x file, which you can import into Blender.

**Noob Note:** That can be done by going into File->Import->DirectX(.x). And there you have a very realistic landscape mesh with a lot less work.

## References

- [1] <http://oneodddude.net/FreeTextures.php?action=browse&catid=5>
- [2] <http://oneodddude.net/FreeTextures.php?action=display&id=10&page=1&sort=id&catid=5>
- [3] <http://www.bundysoft.com/L3DT/downloads/standard.php>

# Blender 3D: Noob to Pro/Landscape Modeling II: Texture Stenciling

---

This is a continuation of the previous tutorial, Landscape Modeling I: Basic Terrain. In this tutorial, we will make our terrain look even better by using some texture stenciling to add multiple textures where we want them in the landscape. This tutorial assumes that you have a basic understanding of how to use Blender (how to add/remove a mesh, how to change views, etc...)

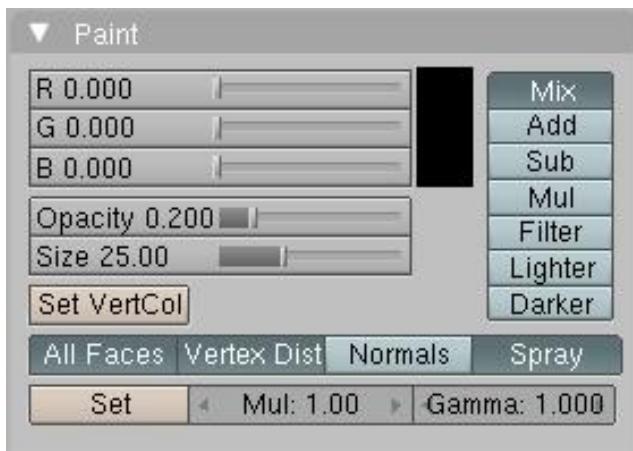
## Creating the Stencil

The landscape from the previous example looks pretty good, but the entire thing has the same texture, so it doesn't look very natural. Let's add some rocks to those hills. If we just add a second texture to our material, it will completely cover the previous one. What we want is to have the first texture only show in certain places and the second texture cover the rest. In order to do this we'll have to create a stencil, which is like a mask that determines where textures appear on the material. The stencil is a black and white image, similar to a heightmap, except the intensity of each pixel determines how much of the next texture will appear (black = 0%, white = 100%). I highly recommend viewing this <sup>[1]</sup> tutorial for a more in depth description of how stencils work. If you think you have the gist, then continue on here. We're going to create our own stencil to determine where we want rock in the landscape by "painting" on the object. **\*IMPORTANT\*** The next few steps will be performing temporary modifications to your scene, so be sure to save your file before you continue so that the changes will not be saved!! Also, if you have a subsurf modifier on the landscape, remove that at this time as it drastically slows down the following process.

- Select the landscape object and switch from "edit mode" or "object mode" to "**Vertex Paint mode**". This is found in the Mode drop down menu, on the 3D View header. This mode lets you paint the object, and thereby change the color value of each vertex. When in Vertex Paint mode, you'll notice your object change color to a pixelated version of the texture that is applied to it. This is because with that texture applied, each vertex will be drawn with the colors you see.

**Noob Note:** Vertex Paint mode is in the drop down menu that contains Object mode and Edit mode. Just select the object while in Object mode, click the drop down menu, and select Vertex Paint mode from the list.

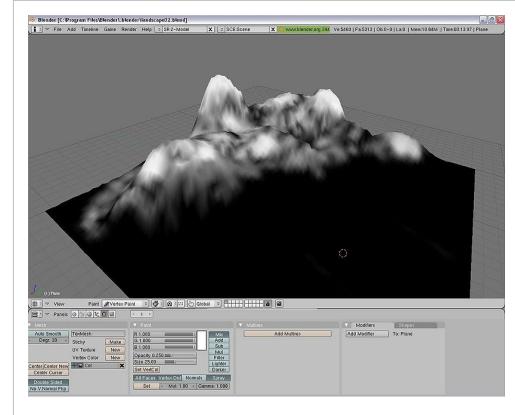
- Switch to the Editing menu (**F9**) and you'll see a new tab called Paint. Notice the sliders labeled Opacity and Size. Opacity determines how much to blend the selected color with the existing colors when you paint. So 0.2 means that when you click, you'll get 20% of the selected color mixed with whatever is currently there. Over on the right are different options to combine to colors. We want to mix, so as we paint the colors will mix together and give us a nice smooth blend. Size is the size of your cursor while painting, and thus how many vertices are affected. Keep in mind that does not change as you zoom in and out, so a size of 10 can actually paint more pixels if you're zoomed way out than a size of 30 if you're zoomed way in.



- Change the color to black and click the button that says Set VertCol (in 2.48 version, that button is below the color square and named 'SetVCol'). This will change the color of your entire object to black. You'll notice that the object is not lit in Vertex Paint mode, so when the entire object is black it can be difficult to see where your hills are. You'll have to move the camera around to see the shapes.

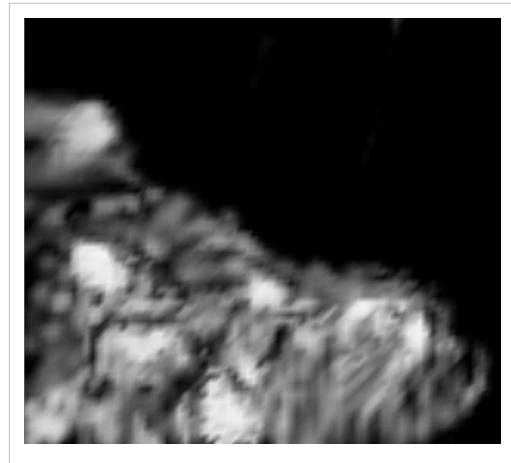
[Another newbie recommends: if you switch to the "object" buttons (f7) and turn on "Wire" under "Draw Extra" in the "Draw" panel/tab you should see a wireframe on top of the vertex colors you are painting, which makes it much easier to see where the hills are. Remember to switch this back off when you have finished painting.]

- Now we're going to paint the places where we want rock to show through. Change the paint color to white. It's less likely that grass will grow on steep slopes, so start painting the tops and edges of the steeper hills white by holding down **LMB** and dragging your mouse (think MSPaint). If you didn't heed my earlier advice and still have a subsurf modifier in effect, the painting will be very choppy, so remove that now.
- As you're painting, try to make some spots pure white by going over them again and again, but don't make any one area of white too large as this will make a huge area be all rock, and we're trying to blend two textures.
- This process can just be trial and error getting the landscape painted the way you want. Once we apply our stencil later you may decide there's too much rock in one area, or not enough in another and go back and change it. A few things to keep in mind:
  - 20% opacity means that most of the original texture (grass, in our case) will show through, so the rock won't be very noticeable if at all. You'll have to go over the same spots a few times to increase the intensity.
  - You don't want a dramatic change from grass to rock, so be sure to blend white areas with black. That's why we're using the 20% opacity instead of just bumping it up to 100%.
  - Make your patches random and spotty. This will end up creating a more realistic effect once we combine the textures.
- Once you're done you should have something that looks like this. Notice the mixture of white and black in some places, how it doesn't just go from pure white to pure black as you move down the hills. This will make a more natural blend and it will cause random rocky areas mixed in when the grass. Now we need to turn this into our stencil.
- Go to the overhead view (**NUM7**) and make the projection orthographic if it's not already (**NUM5**).
- Zoom in/out (**Mouse Wheel** or **NUM+/-**) until the plane almost occupies the entire window.
- Move the 3D cursor and any lamps or cameras away so that they're not over the landscape and take a screenshot.



- Open your favorite image editor (I'm a traditionalist, so I still like MSPaint), and cut out the image of landscape. This is now your stencil. Save it under your favorite format.
- Re-open the saved version of your landscape to undo the changes we made to create the stencil.

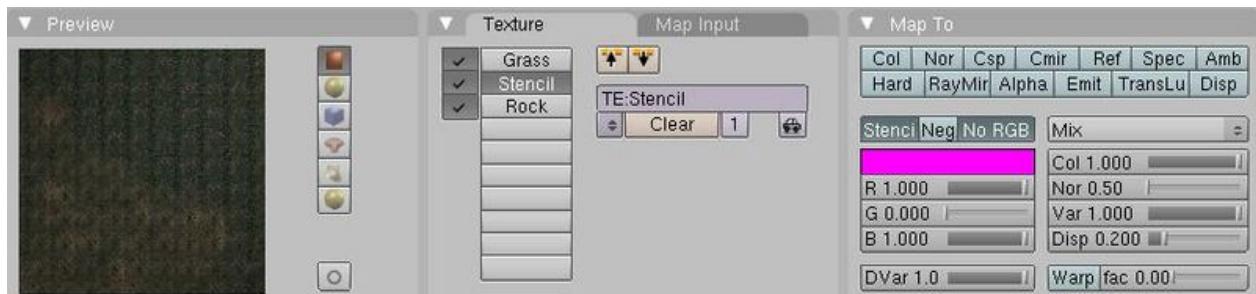
So why did we make our stencil this way? This allowed us to actually paint on our terrain so that we ensure we get the rock exactly where we want it. Using this method, you can customize your stencil to any object, as long as it has enough vertices. For example, if we just had a flat plane made of 4 vertices, this technique would not have worked because we could only paint the corners. That's why in the previous tutorial I recommended using a high (but not too high) number of vertices in your grid.



## Applying the Stencil

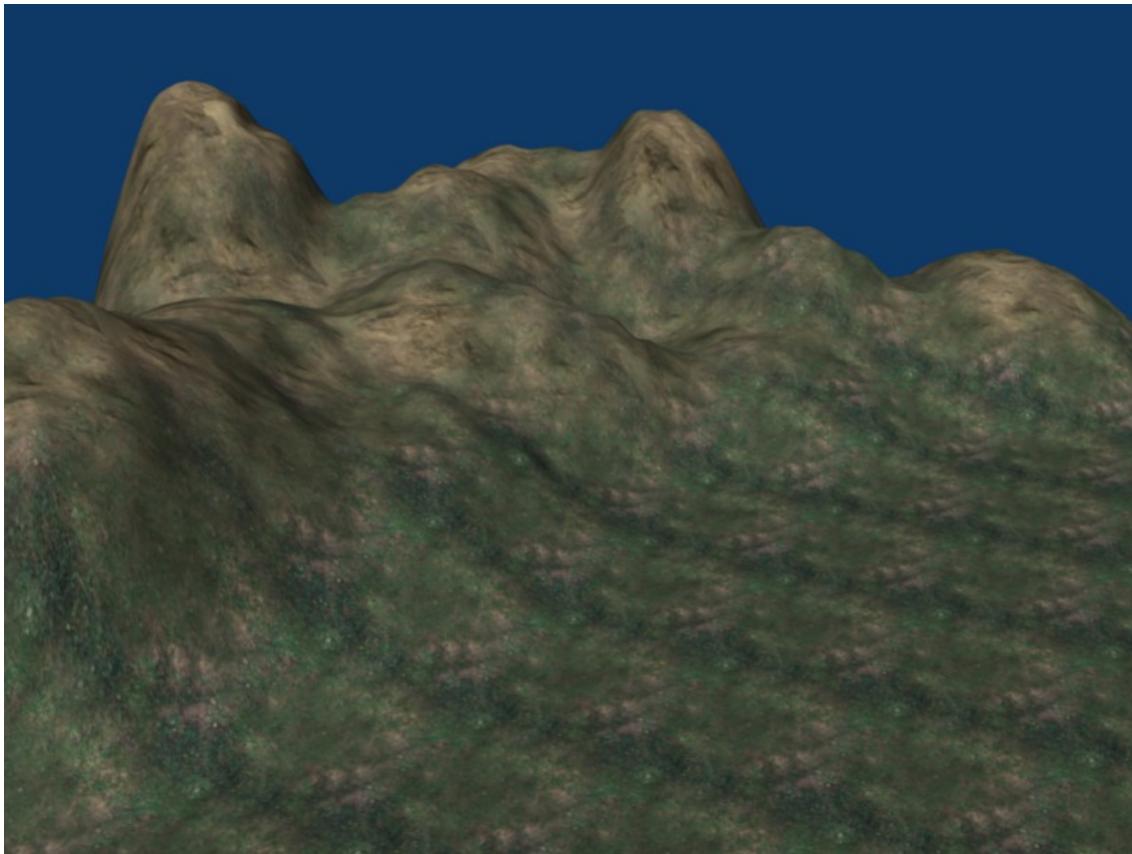
So now we need to apply the stencil to the landscape to mix our grass and rock textures together.

- Return to either Object Mode or Edit Mode (**TAB**), it doesn't matter which.
- Go to the Texture menu (**F6**) and add two new textures in the material (you should already have the grass texture).
- Make texture 2 (the first new one you added) an image and load your stencil from file. Do not repeat this in the x or y directions. We want it to map to the entire object.
- Make texture 3 an image and load your rock texture from file. This one you do want to repeat, just as you did the grass.
- Now return to the Materials menu and select your stencil texture. It's a good idea to name your textures, materials, objectives, etc... so that they're easily identifiable.
- With your stencil texture selected, switch to the tab labeled MapTo. Deselect Col (which maps the texture to the color), and select Stencil and No RGB. Stencil will treat this image as a stencil, and No RGB treats it as a black and white image. If you didn't select this second option your stencil wouldn't work.



- In the preview tab you should notice your material change so that the grass and rock are now mixed. If you render, you'll see a much more realistic landscape than the one from the previous tutorial!

**Noob Info:** Be sure that for the rock - texture *col* is selected because we want to color the landscape with the rock-texture! Otherwise it doesn't work. *Select Rock-Texture -> Map To -> col*



## Adding Snow

Since you already have some nice mountains, why not add some snow to them? Adding snow is easy. All you have to do is add a fourth texture AFTER all the others. Make this texture the same as your stencil.

- Make sure you're in Shading (F5) > Texture Buttons (F6) > Textures.
- Click on an empty panel beside Texture Type
- Click on the little box to the left of Add New
- Choose your stencil from the menu (hint: it helps if you name your textures)
- Render and see the results.

If you made your stencil properly, you should have some nice, snowcapped peaks. It's not much, but it's enough for a quick fix. If it isn't even, you probably didn't make your stencil properly (meaning pretty much all white on the peaks, and not much anywhere else). Of course, there are better ways to do this, ways I wouldn't know about; I discovered Blender merely three days before the time of writing. I leave it up to you to see the effect: I only have so much room and bandwidth for pictures!

This effect works by saturating the underlying color with white, depending on the amount of white in your stencil. A light gray will saturate it only a bit, whereas a bright, 100% white will cover it very clearly and efficiently. If you look closely, you'll see that the light gray on the stencil DOES make a difference, but it's not immediately visible and requires lots of fidgeting with the Render Preview and texture options. You might also try fiddling with the Stencil and NoRGB option, although it worked fine for me with both turned off (I use Blender 2.48a)

Some challenges:

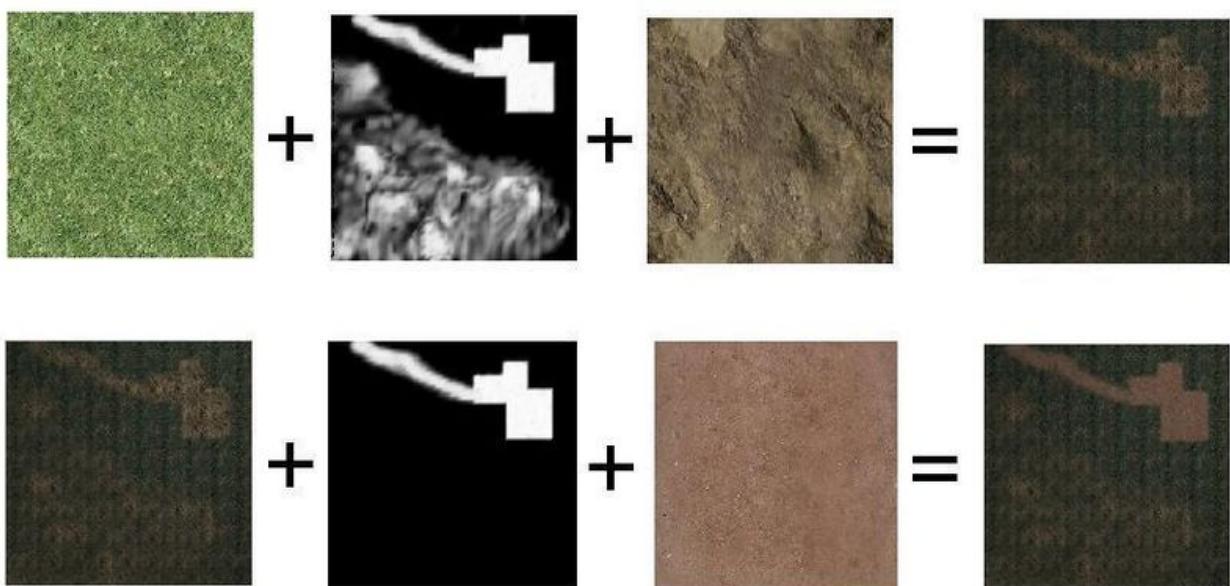
- Try to change the color of the snow. Hint: RGB
- I've yet to do this, but try making snow out of a heightmap.

## Multiple Stencils

Remember in the first tutorial when I said to leave part of the landscape flat, because I'd be using it later? Well it's time to use it. I want to eventually use the landscape as the backdrop for a military base, so let's texture the ground around where the base will go to give it a dirt ground and road leading to it. **\*IMPORTANT\*** Just as before, the next few steps will be performing temporary modifications to your scene, so be sure to save your file before you continue so that the changes will not be saved!!

- Switch to the Vertex Paint mode in the mode menu on the 3D View header.
- Go to an overhead view (**NUM7**) and make the projection orthographic (**NUM5**)
- Once again, paint the entire object black by selecting the color black in the Paint tab and clicking on Set VertCol.
- For the next few steps, you may need to **TAB** back and forth from Vertex Paint mode to Object mode to make sure you're painting in the correct area. Make the opacity of the painter 1.0 and the size 10 so that we can work very precisely.
- Paint an area over the flat terrain in the shape of a generic military base, like you see here.
- Reduce the opacity back down to 0.2 and increase the size to 20.
- Paint the road leading away from the base. We've reduced the opacity and increased the size so that the road will blend more with the landscape around it instead of being more defined like our base.
- Once you have your stencil looking the way you want, take a screen shot and save it. Then revert back to your previous environment (before you started painting on it).
- Go to the Texture menu (**F6**) and add two new textures.
- Load your stencil into the forth texture (the first new one), and the dirt image into the fifth texture. Once again, remember to set the x and y repeat for the dirt texture.
- In the Materials menu (**F5**) select your new stencil and go to the MapTo tab. Deselect Col again, and select Stencil and No RGB. Notice the new preview.
- Wait a second, where's the dirt path?!? Here's the problem and the reason I've included this section: stencils are cumulative. That is, the first stencil defines which portions of ALL successive textures will be drawn, including other stencils! In order for our new dirt path to show up, we need to make sure it's part of the first stencil.
- Open your first stencil (the one you used to add rocks to the hills) in your image editor and combine the new stencil with it. The result should appear somewhat like this. Be sure that when you combine your dirt stencil with your rock stencil that the dirt stencil remains the same size and in the same place. That is, you want the two to be perfectly aligned.
- Reload your first stencil on your material, the one we used for the rocks. Now the preview should look the way we want it to, and if you render you will see your new dirt area on the flat part of the landscape. You can add even more stencils the same way. Here is how all of the textures blend together:





**NOTE** I had to replace the grass texture in the above image since I could not find the licensing information for it. I apologize for the inconsistencies in the diagram that result from this.

Well there you have it! This is a pretty useful method of texturing complex objects. Check out our new landscape and how it compares to the previous, boring, single-textured landscape!



Once again, if you have any issues with the tutorial, or feedback (positive or negative), drop it in the discussions.

## References

- [1] <http://feeblemind.tuxfamily.org/dotclear/index.php/2005/02/27/16-didacticiel-usage-des-masques-de-textures-tutorial-using-texture-stencils>

# Blender 3D: Noob to Pro/Landscape Modeling III: Exporting as a Heightmap

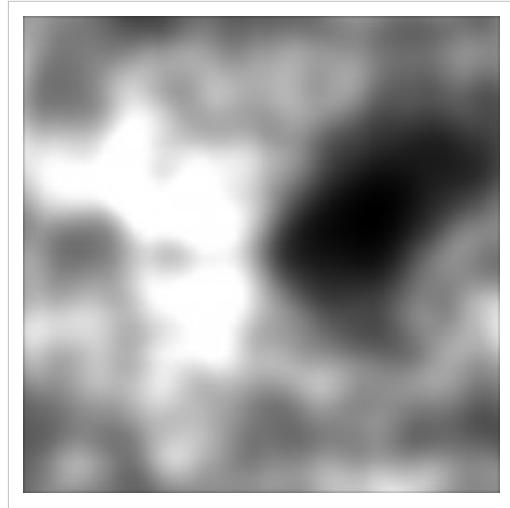
In this tutorial, I will show you how to export your beautiful terrain to a heightmap that can be used in most 3D graphics engines. The benefit to doing this as opposed to just exporting your mesh is that many engines have a special process for dealing with terrains as opposed to regular meshes where it divides the terrain up into different sections so that it can subdivide regions closer to the camera more for greater detail, and also cull (not render) regions outside the camera's field of view, saving precious processing time. This structure is called an oct tree and is a highly optimized way to render large meshes such as terrains.

Before I begin, I must give credit to the Creating a Heightmap from a Plane <sup>[1]</sup> tutorial on the Blender wiki <sup>[3]</sup>. This is how I learned to do this trick, and most of what I will cover came from this tutorial. I'm simply including it here with the other landscaping tutorials I've written for convenience.

## A Word About Heightmaps

I will briefly cover what a heightmap is and how it's useful for those who may not know. If you are familiar with heightmaps, you can skip ahead to the next section.

A heightmap is a grayscale image that uses various shades of gray to represent different elevations across a map. Since the images are 8-bit (with the exception of some RAW formats that are 16-bit), you have 256 different shades of gray, ranging from pure black (0) to pure white (255). Black represents the lowest elevation on the map, while white represents the highest. Many 3D graphics engines already have functions to read in a heightmap and generate terrain from it. The way it does this is it creates a grid of vertices (like we are going to do in Blender in just a moment), and uses the heightmap to determine the elevation of each point on the grid. The more intense the pixel color (the closer it is to white), the higher the elevation of that vertex. In most cases, if the resolution of the heightmap is smaller than the resolution of the terrain, the engine will interpolate the vertices in between those set by the heightmap. For example, if your heightmap is 256x256 and your terrain is 1024x1024, your heightmap will determine the elevation of every fourth vertex, and the three in between will be interpolated.



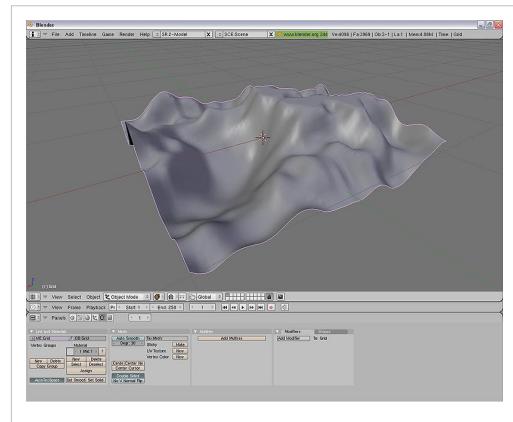
Note that these values do not represent the absolute height of any given pixel, but rather the height relative to the rest of the map. That is, your heightmap does not represent a landscape with altitudes ranging from 0 to 256. When importing a heightmap, you can specify the minimum and maximum altitudes for the map, and the whole thing gets scaled. For example, suppose you decide the minimum is 1000 feet, and the maximum is 5000 feet. When the terrain is rendered, black pixels will represent 1000, white pixels will represent 5000, and a pixel that is exactly in between black and white (128) will represent 3000 feet (half of the sum of min and max). This makes heightmaps relatively flexible.

Unfortunately, as you may have already noticed, there is a limitation of using heightmaps. Since you only have 256 shades of gray, you only have 256 possible elevations. This causes a problem. No matter how precise your vertices may be in the mesh that you model, no matter how smooth you may have it looking, each point will get rounded off to an integer value between 0 and 255. When used to render a terrain, this can cause a "stair step" effect as the terrain goes from one discrete elevation to the next instead of making a smooth transition. But there are ways to avoid this, which I will cover later.

I apologize if you're scratching your head right now saying "Huh?" For a further explanation and example, check the Wikipedia page on heightmaps.

## Creating the Material

To begin this tutorial, I will assume that you already have a landscape that you want to create a heightmap for. If not, read my first tutorial, Landscape Modeling I: Basic Terrain, to quickly create something. **WORD OF CAUTION!!!** If you've already created a landscape, try to remember the exact size of the plane that you used. If you're about to make one, it's okay to scale the plane to have a larger surface to work with, but REMEMBER how much you scale it by. This will make things easier later. I recommend just working with the default 2x2 plane or grid and zooming in on it. You may also want to backup your scene since we'll be changing a few things

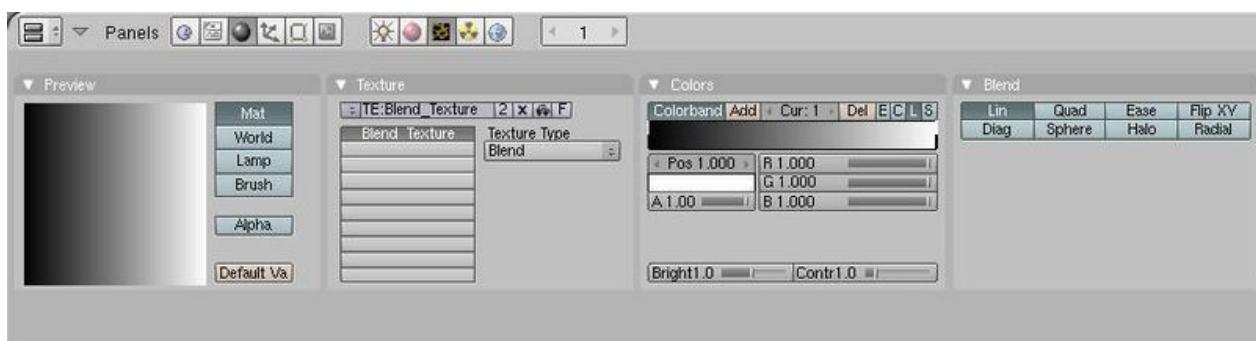


To create the material we need, do the following steps:

- Remove any existing materials that may be applied to the terrain.

**Note:** to remove a material, I think you need to go to the links and pipeline window in the shader section and click the little "X" underneath the words "link to Object"

- Add a new material in the Shading tab (**F5**).
- Go to the Textures tab (**F6**) and add a new texture. This should be the only texture on your material.
- From the Texture Type drop down menu, select Blend.
- In the same window, find the Colors tab and select Colorband. This will let us define the blend pattern, which by default fades from black with full transparency to a solid cyan. Why cyan? I have no idea. It's an odd default.
- Select the black color by clicking on the left side of the colorband. You should see a little black and white bar selected.
- Increase the Alpha all the way up to 1 by adjusting the slider labeled "A".
- Select the cyan color and change this to a solid white.
- What you have now should look like my colorband below.

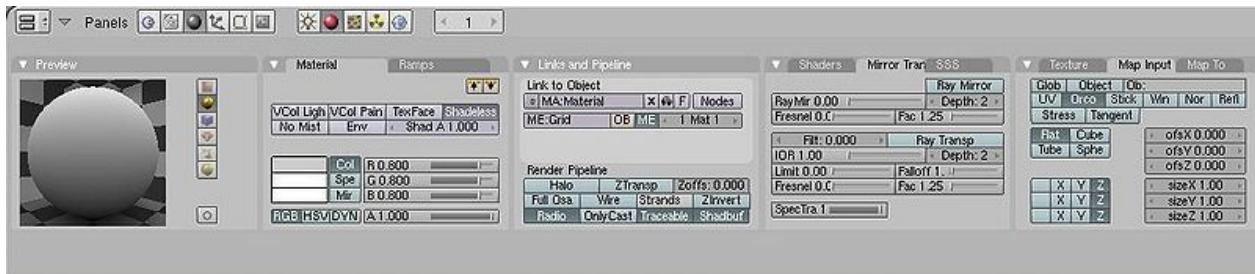


- Return to the Materials tab (the red ball icon next to the cheese looking icon).

- In the Material pane, select the Shadeless option to disable lighting on the material.
- To the right, find the pane labeled Map Input. In the bottom of this pane is a grid that by default reads:

**XYZ****XYZ****XYZ**

- Set all three rows to **Z** so that the texture will map entirely to the Z coordinate.
- In the preview window to the left, if you select the sphere or cube you should see that it goes from black on the bottom to white on the top. That's the effect we want for the terrain. Remember, in a heightmap, black represents the lowest point and white the highest.

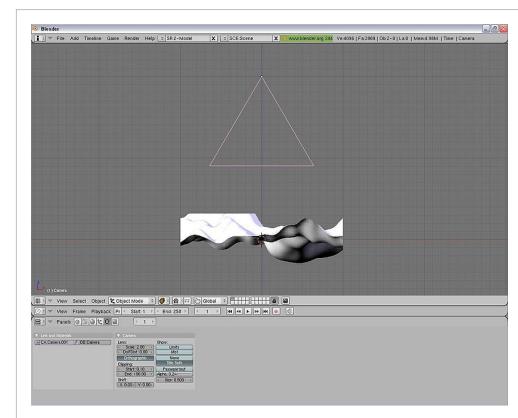


- Finally, switch back to the Editing tab (**F9**) and click the Set Smooth button to enable Gouraud shading so that everything appears smooth.

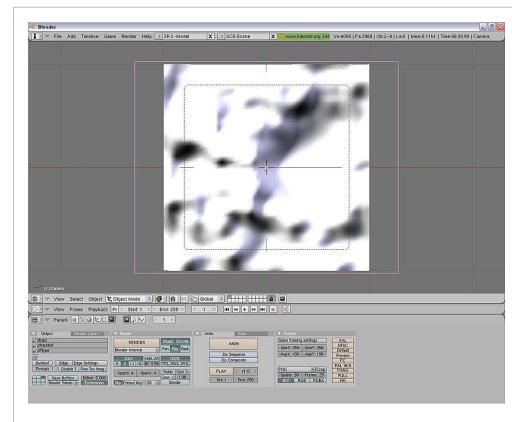
## Setting up the Camera

That's all we need to do for the terrain itself. Now what we're going to do is setup the camera in such a way that when we render the scene we'll get an orthographic projection straight down on the terrain so that we can save it as our heightmap.

- If you still have any lights or cameras in the scene, delete them.
- Switch to the top view (**NUM7**) and place the 3D cursor at the origin (0, 0, 0) by left clicking near the origin, hitting **SHIFT+SKEY** and selecting *Cursor to Grid*.
- Add a new camera, *Space->Add->Camera*. This will add a new camera to the scene that is looking straight down.
- Switch to a side view (**NUM1** or **NUM3**) and move the camera up the Z axis so that it is above the highest point in your terrain. Even though the view is going to be orthographic, and therefore distance doesn't matter, the entire terrain still needs to be in front of the camera.
- Now, with the camera selected, go to the Editing tab (**F9**) and select the "Orthographic" button. This will change the camera to an orthographic projection which basically ignores the z-coordinate for all vertices (except for determining which pixels should be in front of others).
- Just above the orthographic button is a value labeled "Scale". Remember earlier when I told you to remember how big your terrain is? This is why.
- Change the Scale of the lens (default: 6) to match the size of your terrain. If you left the plane the default size, this should be 2. If you scaled the plane up by a factor of 20, as suggested in the "Landscape Modeling I: Basic Terrain" tutorial, this should be 40 (the plane is initially 2 units square).
- Now go to the scene tab (**F10**) and under Format, change the SizeX and SizeY to be the size you want your heightmap to be. For most graphics engines, this is required to be either a power of 2 ( $2^n$ ) or one more than a power of 2 ( $2^{n+1}$ ). So try something like 256 (or 257 if you need it to be  $2^{n+1}$ ).

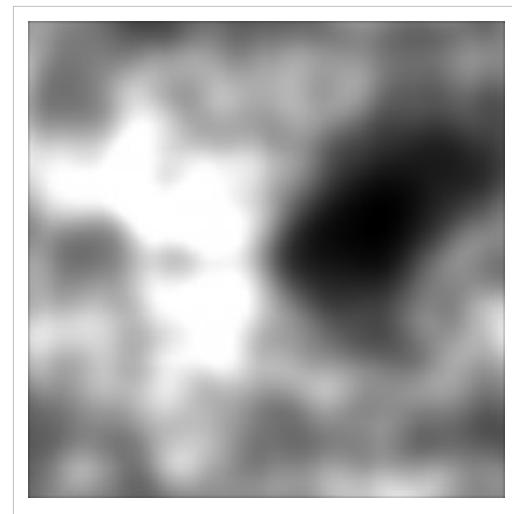
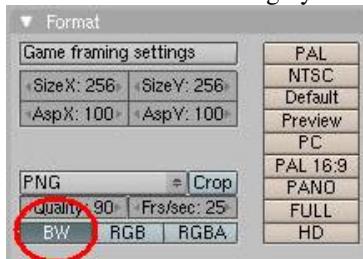


- Now, if you switch to the camera view (**NUM0**) you should see a "flat" square that completely fills the dotted box which represents the camera's field of view.



- Render the scene (**F12**). You should be looking at your heightmap. Hit **F3** to save the rendered image and you're good to go!

Since this image has no color, and some engines require that your heightmap be a grayscale image (only one color channel instead of three), select the BW button in the Format pane before rendering. This will render it as a grayscale image instead of a color image.



## Avoid Stair-Stepping

As mentioned earlier, exporting the heightmap as an 8-bit image (Blender doesn't support any 16-bit formats that I'm aware of) has the drawback that you only have 256 discrete height values which can cause a stair-step effect when the terrain is subdivided. Ideally, you could use an application that can export the image in a 16-bit raw format, such as Terragen. But this is a Blender tutorial, so we'll not go into that. =P

## Reducing the resolution

One way you can "smooth" out the rendered terrain is to actually reduce the resolution of the heightmap. This may sound counter intuitive, but bear with me. Suppose you have a heightmap that is a simple blend from black to white going from left to right. If your heightmap is 1024x1024, you will get 4 columns of each height since you only have 256 different values. In other words, your grayscale will look something like this:

00001111222233334444...

00001111222233334444...

00001111222233334444...

00001111222233334444...

00001111222233334444...

00001111222233334444...

If you try to render terrain from this, you'll get 4 points with 0 altitude, 4 with 1, 4 with 2, and so on, creating the stair-step effect.

Now suppose you make the same heightmap, but reduce its size to 256x256. Now you'll have 1 column for each height, and your grayscale will look something like this:

012345678...

012345678...

012345678...

012345678...

You may be thinking, "But then my terrain will be smaller, or less detailed." But it turns out the opposite is actually true. Consider applying the above heightmap to a terrain that is 1024x1024. The values from your heightmap will be applied to every 4th vertex, and the ones in between will be interpolated. So instead of having something like:

**0.0 0.0 0.0 0.0 1.0 1.0 1.0 2.0**

which is what you would get with the 1024x1024 heightmap, you'll actually get something like this:

**0.0 0.25 0.5 0.75 1.0 1.25 1.5 1.75 2.0**

Where the integers are the values from your heightmap and the fractions are interpolated values. This is assuming that whatever engine you are using will interpolate the values between heights read in from the heightmap.

At the present time, this is the only method I have verified that helps reduce the effects of stair-stepping. As I try different techniques I will update this page with them. For example, I know that Adobe Photoshop can export images in 16-bit raw formats, but I haven't tried it yet so I don't want to suggest it as a solution.

---

As always, I do these tutorials for you, the reader. So if you have any issues with the tutorial, or feedback (positive or negative), drop it in the discussions. I'll try to keep an eye on it to see what people have to say and update accordingly. Moohasha (talk) 19:02, 17 December 2007 (UTC)

## References

[1] [http://wiki.blender.org/index.php/Tutorials/Creating\\_a\\_Heightmap\\_from\\_a\\_Plane](http://wiki.blender.org/index.php/Tutorials/Creating_a_Heightmap_from_a_Plane)

# Blender 3D: Noob to Pro/Realistic Water using Fluid Sim and Yafray

(This was written using Blender v2.43)

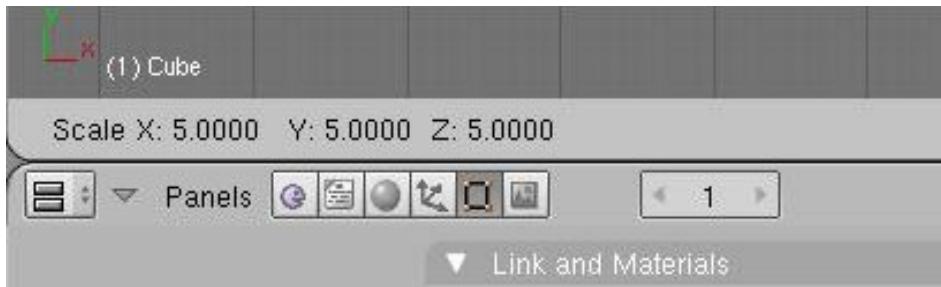
## Using the Fluid Simulator and Yafray to create realistic looking water

This tutorial will (hopefully) successfully walk you through creating semi-realistic looking water using the Fluid Simulator in Blender, then rendering in Yafray.

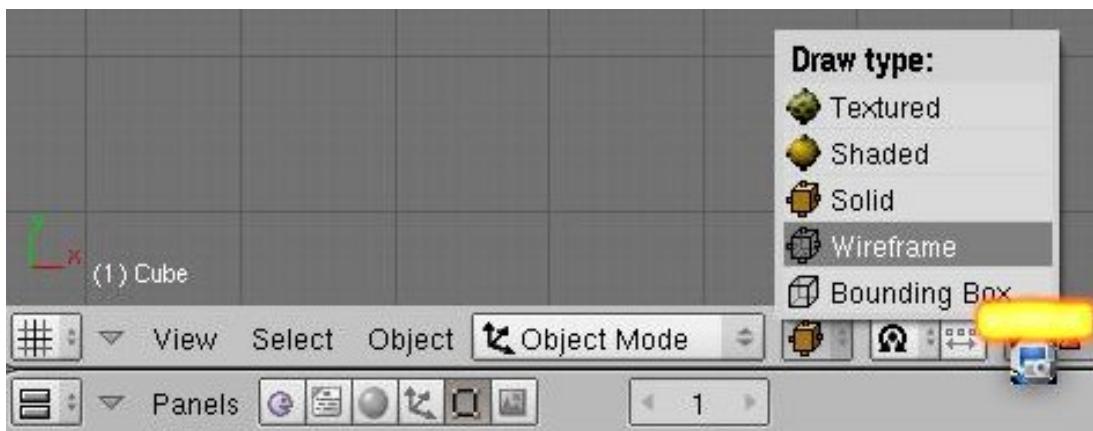
### Getting Started

When you first open blender there will be nothing but a blank cube.

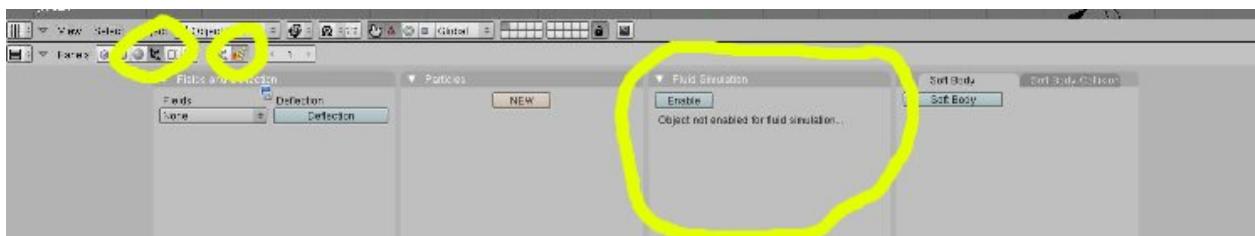
1. Make sure you are in object mode, and scale the cube up to about 5. Do this by pressing S in the 3d view and holding ctrl, move your mouse until the numbers in the bottom left of the screen are 5. Or select the object and press S then 5.



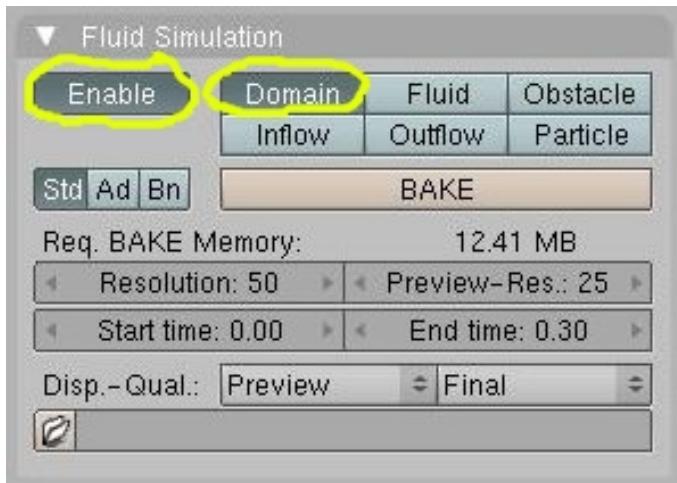
1. Next, in the viewport shading box select wireframe. This lets you see through the cube.



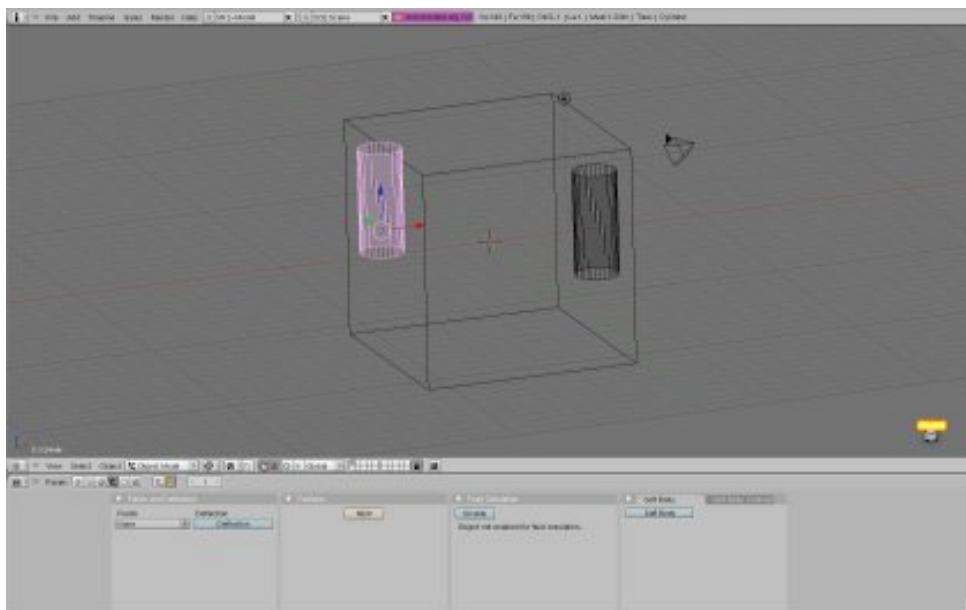
1. We are now going to enable the cube for Fluid Simulation. With the cube still selected in object mode, press F7 (object button) and then press the "Physics Button".



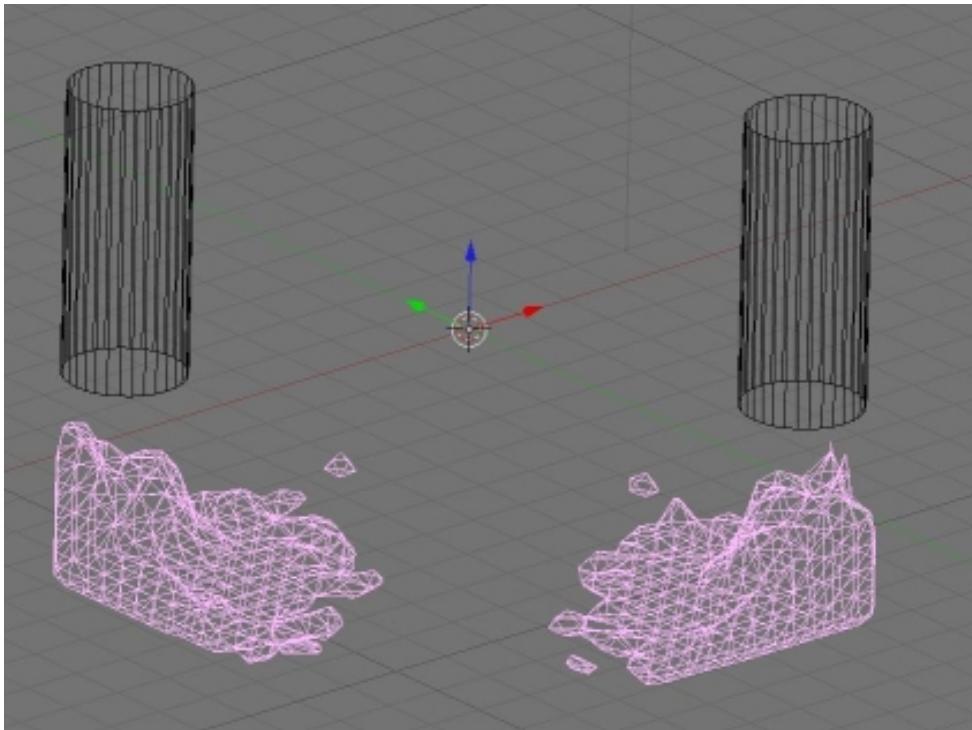
1. You should see a tab labeled "Fluid Simulation" press Enable and then Domain. After pressing Domain several buttons will pop up, these will be used later.



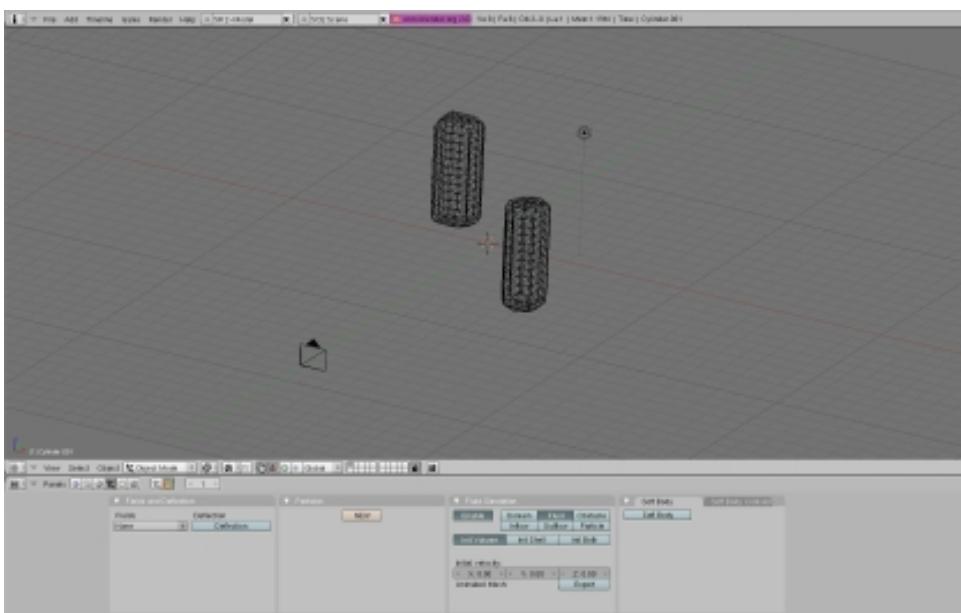
1. Make sure you are in top view now, if not just press NUMPAD7. Next press space>Add>Mesh>Cylinder. 32 vertices is ok.
2. Now middle click and drag in the 3d view to see the cylinder inside your beautiful box. (yay!) What's going to happen is the cylinder is going to be turned into liquid, and the big box is where the liquid can flow about. (it is the "invisible wall")
3. Move the cylinder to a corner of the cube, you can give it some height if you like. Now with the cylinder still selected press Shift-D to duplicate it. Move the new cylinder to the opposite corner.



1. On both of the cylinders go the fluid sim. tab and press Enable then Fluid. This tells blender that these cylinders are fluid.
2. IMPORTANT!!! After this, you MUST subsurf both of the cylinders, but don't apply it. Instead, when the cylinder shows a preview of the subsurf, click the "X" in the top-right of the subsurf preferences.
3. Switch to Object Mode then Right click on the big box (the domain) and press the big BAKE button. A little box will pop up and just click on the top option.
4. Now you should see a blob looking thing appear in the cylinders, and it will fall to the ground and splash around. Your computer is "baking" the liquid. (Calculating where it should go)

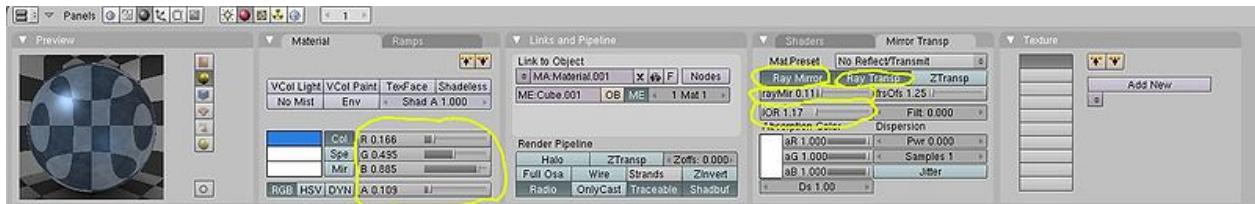


1. As you see now, the liquid isn't very realistic looking. If you do a render, you will see. You would also see that the cylinders are appearing in the image. This is not wanted, so with both of the cylinders selected, press M then click on one of the little boxes that isn't already pressed. This moves them to another layer.
2. Now we are going to move the camera to a different position for a better view. Move the viewport around until it's where you think will give you the best view and press **SHIFT-NUM0** to make the camera match your current view.
3. Drag the newly placed camera back a bit, so the when NUMPAD0 is pressed most of the scene is in view. You should now have something that looks similar to this:



1. Now we are going to rebake the water at a higher resolution for a more detailed result. Select the goo looking mesh, (both blobs should turn purple if either is clicked) and go the fluid simulation tab again. Turn the resolution up to about 100 (or whatever you want) this rebakes the fluid for more detail. If you turn preview mesh up too, you will see it more detailed before rendering. Baking now may take a while.
2. After it's done baking, you are ready to change the material of the water to something better looking.

3. First change the renderer to Yafray. Press F10 then under the big Render button select Yafray. Make sure Yafray is installed first, the option still appears if you don't.
4. Change the background white by pressing F5 then click on the box that has world buttons appear when the cursor is hovered over it.
5. You will see HoR, HoG, and HoB. Turn these all the way over to one. The preview should turn white.
6. Now go to the material buttons (red sphere) and make everything look like the following: (I think I circled all the changed options, but maybe not.)



1. After this is completed look through the frames by pressing right arrow, find one that looks cool then press render and you may get something like this:



Hopefully you got something like that :)

(that's all folks...)

If the render does not look smooth, apply a subsurf.

## Extra Practice

This YouTube tutorial on fluids might also help: Link [1] and this Realistic Water Texture [1]

Video tutorial showing fluid simulator objects in Blender 2.49b: [http://www.youtube.com/watch?v=wX5w1Kmsv\\_A](http://www.youtube.com/watch?v=wX5w1Kmsv_A)

---

**User Question:** When I click Bake there is no box that shows up, I cannot get it to bake my fluids at all, all it does is add a bunch of files to my hard drive and take a long time, What should I do?

**Answer:** Be sure to set the directory that you would like the files to be saved to, the option to do so is in the area where you changed the box to a domain very bottom there is a folder. Choose a directory and all will be good.

**User question:** When i click the bake button it says there can't be more then one domain object. I did everything you said to do but i don't understand why it says that.

**Answer:** The only solution I have found around that is to go to the top header and where it says "SR:2-Model" change that to actor and select the different objects until you find something that doesn't highlight. Delete those. I found that occasionally a domain "disappears" and when you make a new one causes this problem. I have also found it better to set up the domain last around your scene to prevent this annoyance as well.

**Note:** The domain object does NOT disappear - the Domain object IS the object on your screen which represents the fluid itself.

## References

[1] <http://au.youtube.com/watch?v=yMBx2pYiqnY>

# Blender 3D: Noob to Pro/BBB's Particles -- New 2.46 Particle System

---

Note: Particle Documentation is great, and I don't want to discourage anybody. But: I've now just finished the particle documentation [1] and so this page may not be necessary. Maybe the page title should be changed though, and the page moved to another place. --SoylentGreen (talk) 10:46, 29 March 2009 (UTC)

## Introduction

Blenders release of 2.46 -- created for the Blender movie Big Buck Bunny and dubbed so -- BBB's particle system is a stream-lined and much more articulated so particles can be fine-tuned and run in extremely intricate ways. You can even enter a brand new mode --Particle Mode-- and edit each particle individually or in groups, for things such as fur or hair waves. This section will go over the three different types: emitters, reactors, and hair.

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Modeling/Textures

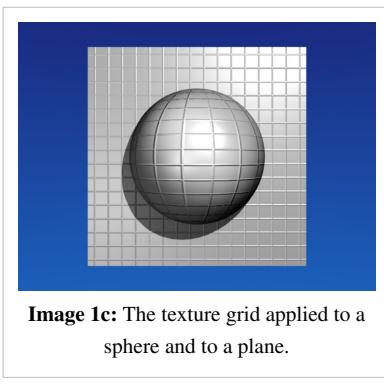
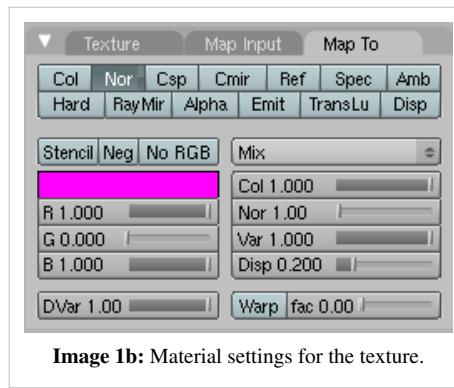
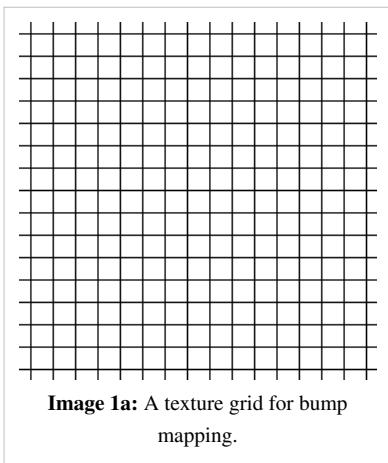
## Modeling Textures

When 3D modeling, textures make or break not only the look of your model, but almost as important--your poly count and workload. There are many great things textures can be used for:

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Modeling/Bump Mapping

*This tutorial was created using Blender v2.49*

Bump Maps are textures that store the relative height of pixels from the viewpoint of the camera. The pixels seem to be moved in the direction of the facenormals, either in direction to or away from the camera. You may either use greyscale pictures or the intensity values of a RGB-Texture (including images).



Bump Maps are easy to apply. They work well on flat surfaces, only to some extend on curved surfaces. On curved surfaces it is more easily noticeable that you don't create real 3D structures. The visible effect depends on factors like lighting, specularity of your material, camera angle, distance and so forth.

- Add a new texture to your object.
- Activate the *Nor* button in the *Map To* panel.
- Set the depth of the bumping with the *Nor* slider.
- Use the texture type *Image* and load your bump map.

Bump maps should contain hard transitions between black and white. A gray wedge (e.g. a linear blend texture) would be hardly visible in the rendering.

## Creating Bump Maps

You can easily create Bump Maps with Blender yourself, this is especially useful if you have modeled some small details on a surface and you realize at the end that your scene will get too complex. You could also use the Bump Maps in a 2D application like Gimp or other, similar programs.

I will create an animated bump map in highest possible quality, it is not always necessary to make such an effort. The goal is to create a wave effect and make an image sequence of it, to be used as a bump map. The original object has 600.000 vertices, the object the map is applied to has 8 vertices.

### Setting up the scene

- Open the default scene and remove the cube.
- Insert a plane and change to edit mode.
- Crease its edges with +1 (**Shift-E**).
- *Subdivide multi* with 11 cuts.
- Change to object mode.
- Add a *SubSurf* modifier with a render level of 6.
- Add a *Wave* modifier.
  - *Time Sta:* 10
  - *Height:* 0.2
  - *Width:* 1.5
  - *Narrow:* 1.5
- Set the camera dead upon the plane.
  - *X/Y/Z: 0/0/5*. You can use the *Transform Properties Panel* to bring the camera to a certain position, the Z-Position is important in this case.
- Set the camera lens to 80.00 (*Editing Buttons->Camera Panel*).

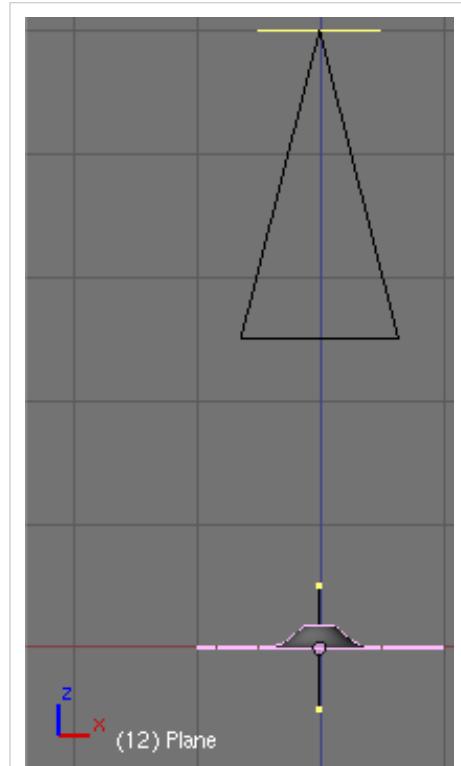
The plane should now fit exactly into the camera view.

- Save your file.

We're going to use the Z-Buffer information to create the bump texture. The Z-Buffer contains the distance from the camera, this is exactly what a bump map is. To render the Z-Buffer information as an image, we're going to use *Composite* nodes. To get the highest possible quality, we will use *Open EXR* as file format, this allows us to store the Z-Buffer information with a numerical accuracy of 32-Bit floating point, instead of a meager 8-Bit value.

### Render settings

- Change to the *Render* buttons.
- Set *End* to 40.
- Activate *Do Composite*.
- Select *Open EXR* in the *Format* panel, set *SizeX/Y* to both 600 (square image for the square plane).
- Set the *Output* directory to *//BumpAnim/*. This creates a subdirectory to the file where the image sequence will be stored. Don't omit the slashes.



**Image 2a:** Setting up camera and object in the 3D view.

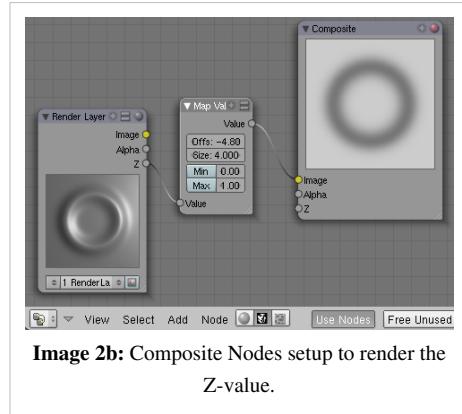
## Node editor

Now the setup for the composite nodes.

- Open a *Node Editor* window.
- Select *Composite Nodes*.
- Activate *Use Nodes*.

A *Render Layer* and a *Composite* node will be inserted automatically.

- *Add->Vector->Map Value*.
- Connect the Z-Output of the *Render Layer* to the *Value* input, and the *Value* output to the *Image* input of the *Composite* node.

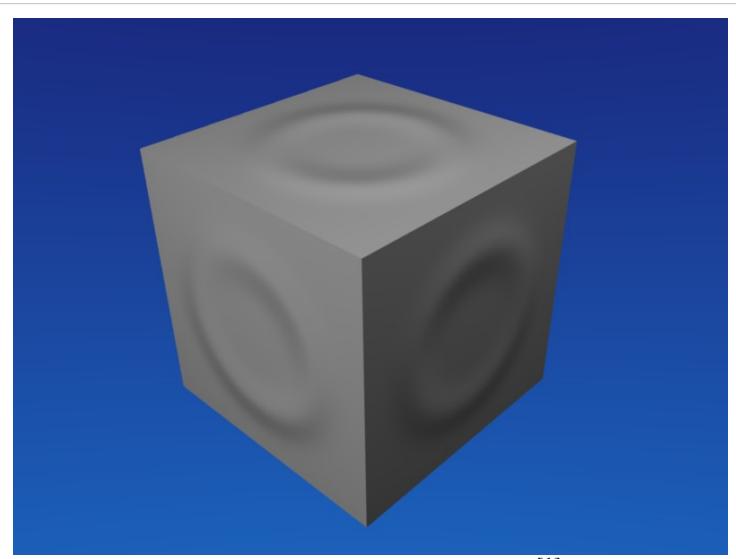


If you render now the image is plain white, we have to talk a bit about the *Map Value* node. The *Offs* value is the distance from the camera where the Z-Buffer should start (in negative BU). It is not so important to get the best range, because we use OpenEXR, but if you would like to use PNG instead, you have to select this value carefully.

- Set *Offs* to -4.8. This is the smallest distance from the camera needed. You can measure this value if you use the *Camera Clipping*, or if you render and move your mouse with pressed **LMB** above the render window.
- Set *Size* to 4. The size value is a bit complicated. This is not directly the value range, instead  $range = \frac{1}{size}$ , and  $size = \frac{1}{range}$ .
  - If you want to include the Z-Values from -4.8 up to -5.05 the range is 0.25, so *Size* is 4.
  - If you want to include the Z-Values from -4.5 to -3.5 the range would be -1, so *Size* would be also -1.
  - If you want to include the Z-Values from -4.5 to -5.5, the range would be 1, so *Size* would be 1.
- Render the animation.

## Applying the animated bump map

- Open a new file.
- Select the default cube and change to the *Texture* buttons.
- Use the texture type *Image*.
- Load the first OpenEXR-Image from the directory where you have stored the sequence.
- Activate *Sequence* in the *Image* panel, set *Frames* to 40.
- Change to the *Material* buttons.
- Activate *Nor* for the texture and set the *Nor* slider to 25 in the *Map To* panel.
- Set the mapping to *Cube* in the *Map Input* panel.



This is it. Pretty much the same look as with the real deformation, but using much less resources.

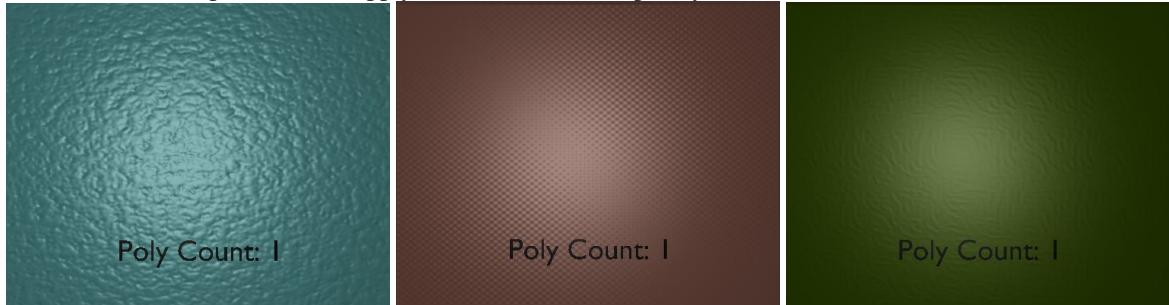
## References

[1] <http://www.vimeo.com/4959751>

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Modeling/Normal Mapping

*Noob Warning: This technique is best used for models and animation, and is not well used and applied in the Blender Game Engine*

Normal Mapping in 3D graphics development is the process of using an RGB color-map to create a three-dimensional relief on a two-dimensional plane. The source for the normal map in blender can either be a texture already installed in Blender, or using an external picture-file (.jpg, .jpeg, .bmp, and so on) and loading it as a normal map. One great aspect of Blender and normal maps is that Blender can very easily be used to **create** normal maps that can be turn around and used in blender to reduce poly count. This tutorial will show the various ways to create a normal map and how to apply different normal maps to your model in Blender.



Sections

within Chapter:

Noob to Pro/Texture Normal Mapping

Blender 3D: Noob to Pro/Color Map Normal Mapping

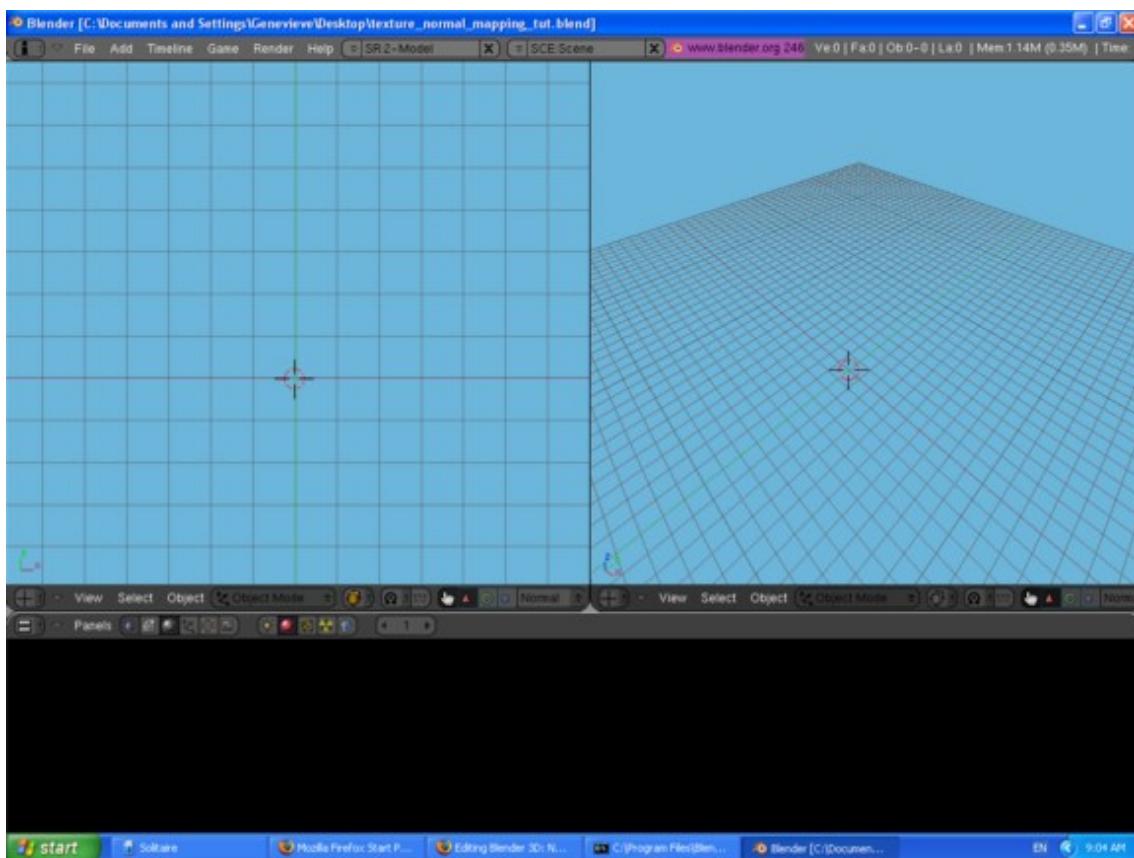
# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Modeling/Texture Normal Mapping

## The Texture-Normal

To familiarize yourself with normal maps and how they work, we will begin by using the texture engine in Blender to create our first normal map. This process is very useful for creating quick and non-specific normal maps for your projects that need a bit of texture.

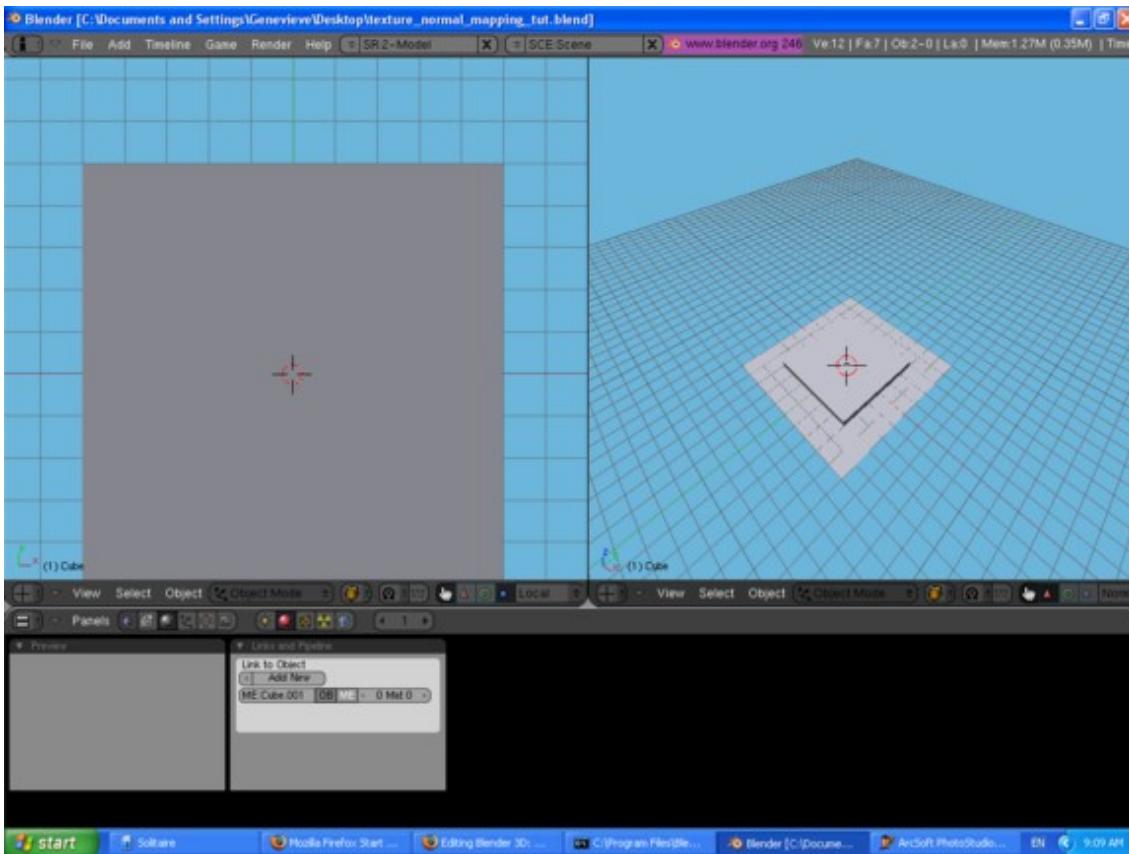
To begin, open blender and whatever your basic load settings are, in the 3D view, delete everything so the field is clear. (**AKEY** to select all, then **XKEY** to delete)

Move the 3D cursor to the center of the X,Y, and Z axis, and switch to top view (**NUM7**) orthographic view (**NUM5**). Hit **Shift-SKEY-4KEY** then **CKEY** to center your view now.



Add a mesh plane to the field, and scale that plane to five times normal size. (**Shift-AKEY**, select **Add->Mesh->Plane**, **TAB** out of edit mode, **SKEY** then **5KEY** to scale five times size.) This plane is going to be our base-it will remain untextured so it can be used as a comparison. Don't give it a material.

Now, with the 3D cursor still on the center, add a mesh cube to the field. Scale this cube 3 times in the X and Y directions, and 0.5 times in the Z direction. (**Shift-AKEY**, select **Add->Mesh->Cube**, **TAB** out of edit mode, **SKEY**, **Shift-XKEY**, then **3KEY** to scale 3 times in X and Y directions, **SKEY**, **ZKEY**, then **.KEY-5KEY** for Z.) This cube will serve as our texture comparison.



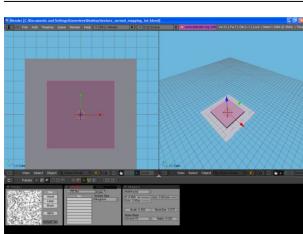
For this tutorial, let's use the "Musgrave" texture to make a 3D relief.

Select the cube, and then in the Buttons Window select the material index. Give the Cube a material, and make this material any light color you wish to differentiate it from the base plane (darker colors won't show the relief that well). Now select the texture icon in the material window. Add a new texture, and make that texture Musgrave.

---

Make the settings for Musgrave the following:

- Type: Multifractal
  - H: 0.5
  - Lacu: 3
  - Octs: 2
  - iScale: 0.45
  - NoiseSize: 0.075
  - NoiseBasis: Voronoi F2
  - Nabla: 0.025
- 



Excellent, now we are ready to show what we have done! If you return to the material window in the material index, then you will probably be wondering what's wrong-your preview window just shows a sphere with a huge purple splotch! Don't worry, we just haven't applied the texture as a normal map yet, it's still just a color map. Click the "Map To" tab in the material window and you will see a whole row of buttons, menus, and sliders. Currently on the

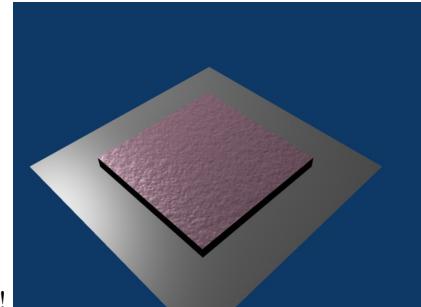
top row of buttons the "Col" button is depressed, which is how the texture knows to apply as a color map. Click the "Col" button to remove the color from your preview window. Now click and depress the "Nor" button and look what happens! An intricate texture embeds itself on the sphere in the preview window. The Musgrave texture has caused a depression texture within the preview. If you happen to click the "Nor" button again, you will see the letters turn yellow and now that means the Musgrave texture is causing an elevation texture. Click it once more and the texture



is turned off. Keep the "Nor" button depressed for this tutorial.

Now that you know how that works, let's see what it did to your cube. First, go to front view (numpad-1) and zoom out a bit so about 10% of the screen is filled with your cube. Select the cube and hit Shift-S-key-4-key then C-key to center your view. In the upper-left corner, add a sun-lamp (Shift-A-key, Add->Lamp->Lamp). Adjust the "Dist" Slider to a very far length, to make sure it reaches the cube. In the upper-right corner (make it a little closer to the cube) add a Camera (Shift-A-key, Add->Camera).

Move the Camera to get a better perspective of your work, by rotating it to look at the cube and the plane. If you want exact directions: Switch to top view (numpad-7). Move -4 Units in Y direction. Rotate -75 degrees on Z rotation. Rotate -45 degrees on Y rotation. Now hit numpad-0 to see how well pointed it is to the cube. Make adjustments to your own work if you'd like.

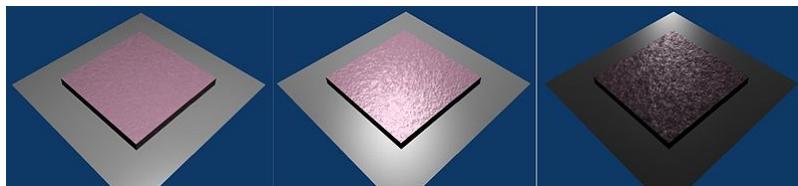


Now you are ready to Render and see how it turned out!

After your first render, I'll let you know that you can adjust the depth of the shading. Select the cube, and in the material index in the "Map To" tab, in the mid-right area, there is a slider called "Nor" that should be set at 0.5. Just to test, set it to 0 and render to see what happens. Then set it to 2 and render to see the results.

After you see it rendered, you probably realized that the edges of the cube were very sharp and didn't have the rocky texture. This is because the 3D relief on the cube is fake--it's not actual 3D, it's the computer just calculating where light should bounce off and in what direction. Imagine how many faces it would have taken to create this texture shading without Normal Mapping.

If you want, you can move the lamp around and see how the light moves around the fake-bumps.



This is best if you just need a non-specific texture, such as asphalt, cement, carpet, or even fabric. Now if you want to do something like make a low-poly brick wall, you will need to a specifically-tailored normal map for such thing. The next section will show you how you can make your own normal map and apply it as well.

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Modeling/Color Map Normal Mapping

## The Color-Map-Normal

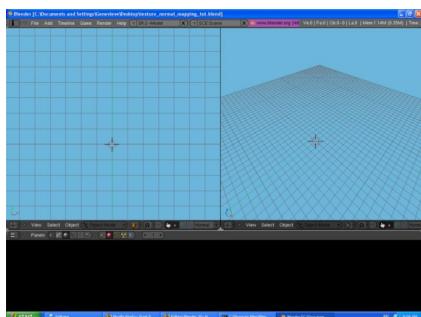
For more advanced Blender users, you probably have at one time or another wanted to create something very detailed and realistic, only to realize that one modeled head or one cool building has taken an enormous amount of memory, and rendering (or worse, animating!) something like that would probably take days! Well there is an awesome trick you can use to create intricate details and design shading, with a relatively low amount of faces. You can use blender to create and apply normal maps, which are RGB colored maps that can be used as a texture to calculate how light will bounce off an object's surface. All you need is blender and photo-editing software to tweek the picture a bit.

## Creating a Normal Map

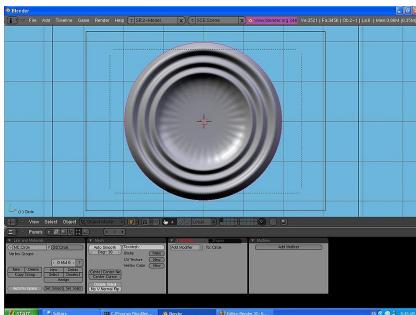
You can very easily create a normal map once you figure out all the necessary settings. You can start by either making a high-poly count version of your object, or making the material that will be applied to the object to make the normal map.

### High-Poly Version

We will first start by making a simple object with a high-poly count. Start up blender, and let's delete everything there first, then go to top-view, orthographic, and start by adding a camera.

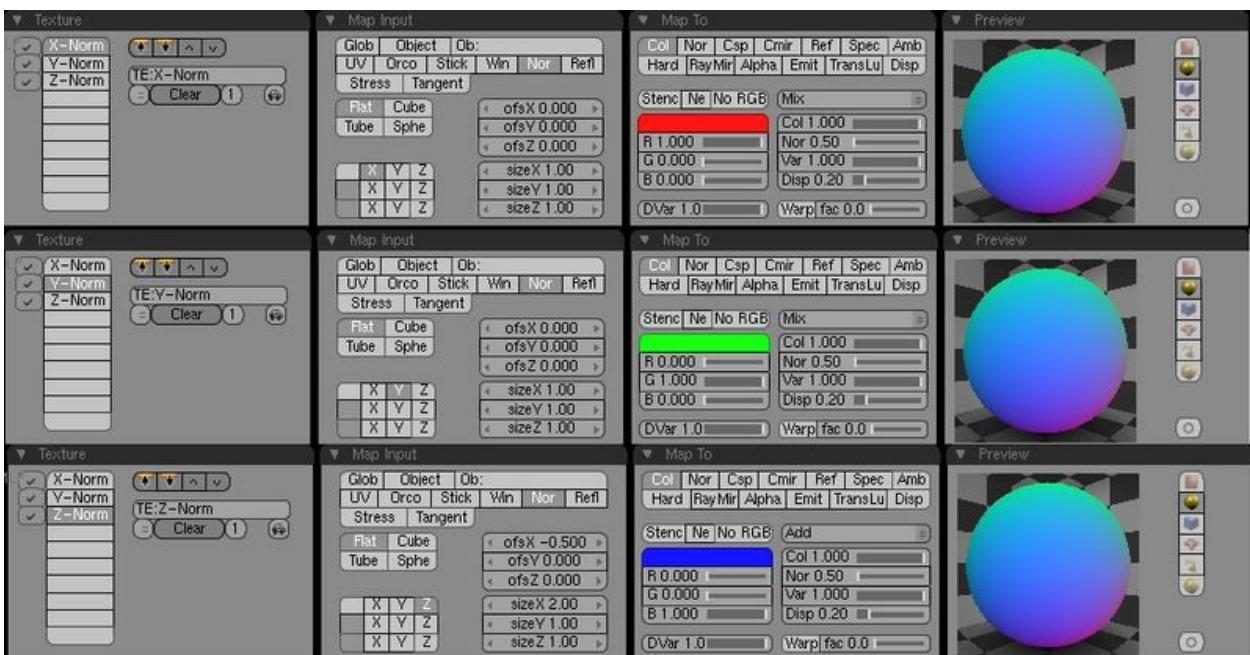


Next, start with a circle, with a radius of 2 and 32 vertices in circumference. Keep it filled. Select the edges of the circle, keeping the inner vertex unselected, and extrude edges only. Scale the new vertices by 1.1. Repeat this with only the outermost vertices 6 more times, until you have a ring of 7 vertices and a center vertex. Now, we will refer to the center vertex as center, the innermost ring as Ring 1, and the outermost ring as Ring 7. Deselect all vertices, then select all vertices of ring 2, 4, and 6 (This can easily be done by holding the Alt and Shift keys, then selecting an edge adjoining two vertices of the same ring for each ring). With these rings selected, move it 0.4 in the Z direction. Now, let's smooth this bowl out. In the editing index, under the modifiers tab, add a subsurf modifier of 2, and then apply the modifier. Then under the links and materials tab, select Set Smooth. Next after the object looks all nice and smooth, select your Camera and move it positive in the Z direction so it's positioned directly above the object looking straight down at it, and edit the camera Lens properties so that it is set to Orthographic, and scale the lens so that it includes all the bowl and fills up the camera space as much as possible.



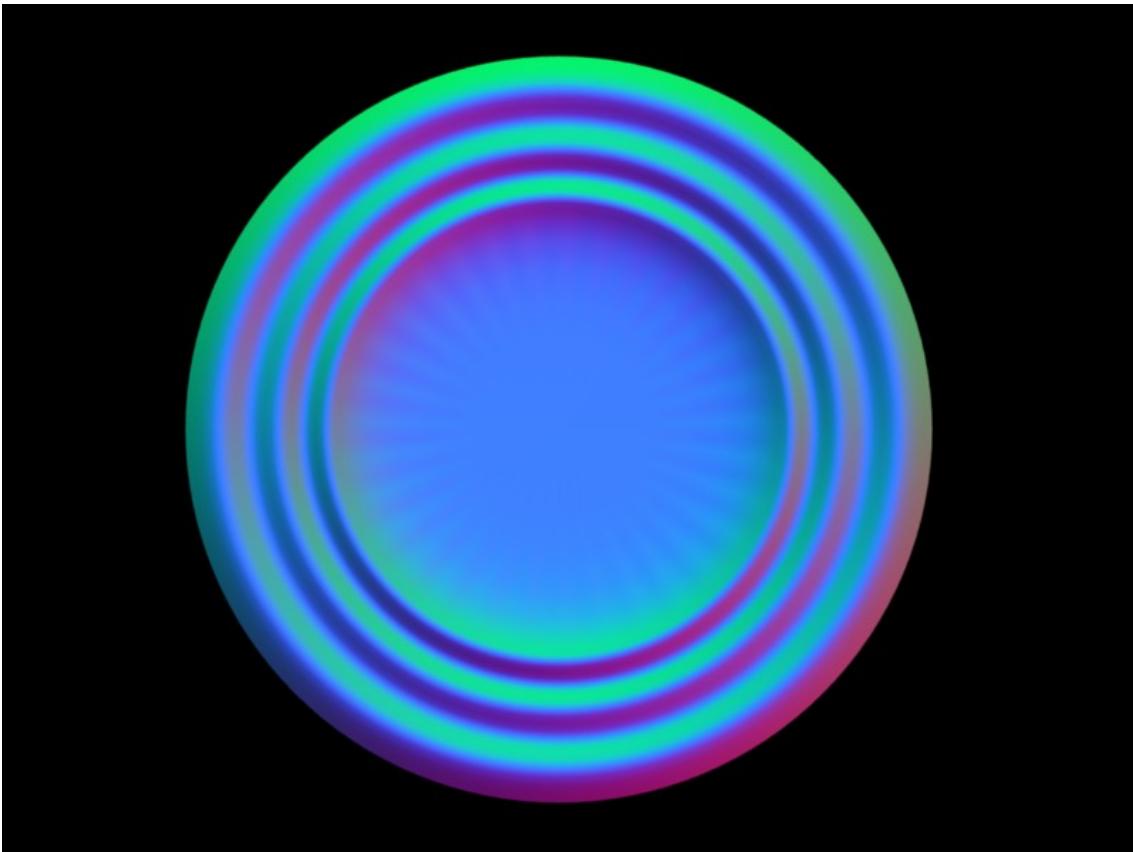
## Creating the Color-Map Material

Now, to be able for the Render to apply the colors on the right areas of the object. For this, we will start by applying a material to the bowl. Make this material 0.0 on the R, G, and B spectra, and depress the shadeless button. Now, go to the textures index and add 3 blend textures to the material. For these 3 textures, name them something to do with either "Red"/"Green"/"Blue" or "X"/"Y"/"Z" in that order, because doing this will ensure you keep the channels in order. The following image will show all settings that need to be set to make the material work.



- [User Question] I have applied these settings, checked and rechecked the values and all I get is a completely white circle for the object in preview and when I render. Using Blender 2.48a.
- [Author Comment] Make sure that the material color is black, and only the shadeless button is depressed.
- [USER NOTE VERY IMPORTANT!!!] Where to choose texture types, set the texture to 'blend' as opposed to none. Solves above issue!!!

Now before your Rendering you want to make the World texture to be completely black, so this way it won't interfere when you apply it as a normal map. Now Render your image, and hit F3 to save. It should come out to something like this.

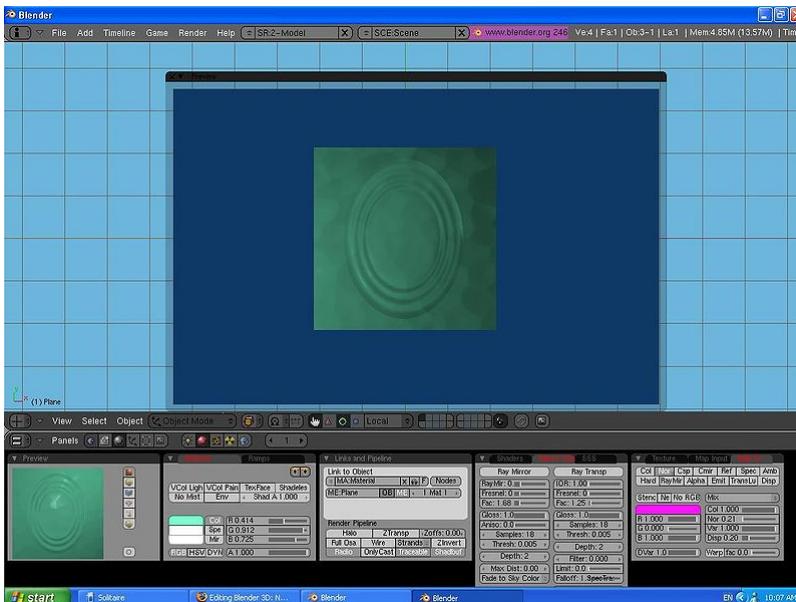


As a final note in this subsection, what I suggest doing is deleting the bowl and saving the file as something like "BlenderForNormals.blend" so if you ever want to make another normal, the area and materials are already set up and all you need to do is import or create the object. Just a suggestion.

### Applying as a Normal Map

Well we got the image, now let's apply it and make it come out as a low poly-count texture. So, let's continue by deleting the high poly bowl and instead add a plane in the middle of the camera. Now, apply a material to the object and make the material hard, soft, rocky, wet, or however you want it to look. After you've made the plane how you want, add a texture to the material, and make it an image texture. Load the image as the texture, depress the normal map button, and set the normal spacing to object. Now, on the materials index under the Map To tab, make sure the Col button is relieved and instead depress the Nor button. Just add a lamp above the plane, and Render! You should now have the same shading effect as the original bowl, with about 0.01% of the original faces--one.

If your Render comes out odd, it may be because the shading is being applied too heavily. You can alter this by adjusting the Nor slider directly below the "Mix" drop-down menu. O used about 0.2 and it came out just fine, like this picture.



Congratulations, that's all there is to normal mapping. It's a simple technique with many great uses!

## Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Modeling/Light Mapping

### Needs Expert

The original writer of this section is not knowledgeable about the subject of "Light Mapping" and hopefully can benefit from someone who has much more experience with it

---

# Unit Five: Advanced Animation

---

## Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/index

---

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. It is assumed that the user has a good understanding of Blender here.

### Index

- **Advanced Animation**

- Introduction
- Guided tour:
  - Armature Object
    - Armature Object in Object Mode
    - Armature Object in Edit Mode
    - Armature Object in Pose mode
  - Mesh Object
    - Connection between Armature and Mesh
    - Envelope
    - Vertex Groups & Weight Paint
    - Shape Key
  - Constraints
    - Copy Location
    - Copy Rotation
    - Track-To
    - Floor
    - Locked Track
    - Follow Path
    - Stretch-To
    - IK Solver
  - Timeline Window
  - IPO Window
    - Data Type
    - Channel
    - Curve Edition
    - Driven IPO
  - Action Window
    - Introduction To Action Data Block
    - Key Edition
  - NLA Window
    - Introduction To NLA Editor

- Key Editor In the NLA
- Strip Edition
- Strip's Properties (NKEY)
- The Stride feature
- Working example: Bird
  - Build The Rig
  - Add Constraints
  - Deform The Mesh
  - Create A Fly Cycle
- Working example: Bob
  - Build The Rig
  - Add Constraints
  - Deform The Mesh
  - Create Shape Key
  - Create A Walk Cycle

## Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Introduction

---

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).

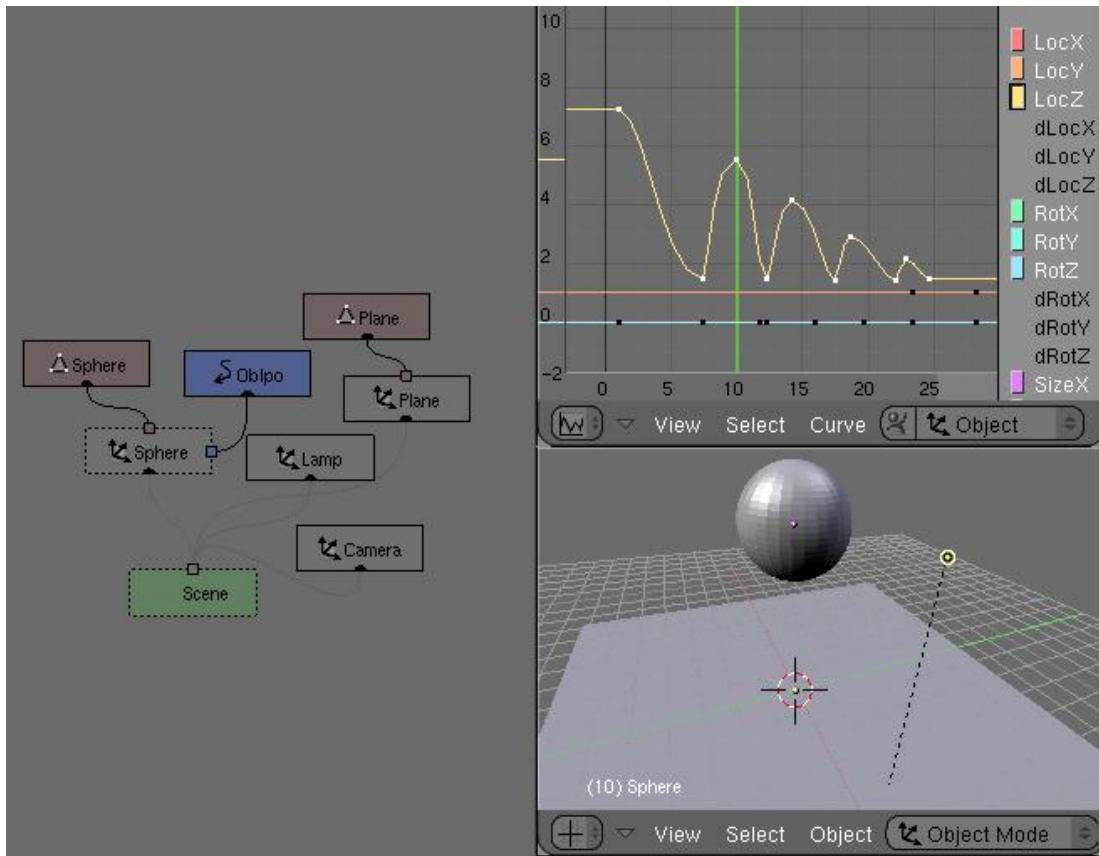
An object links to all of the data you can see in a 3D view such as **mesh**, **curves**, **nurbs**, **lattices**, **armatures**, **metadata**, the **empty** property, text, camera and lamps.

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.

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/index

---

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

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/Armature/ index

---

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 available in Blender 2.40a and later. Refer to the Blender 2.40a release notes <sup>[1]</sup> on Armature draw types and Armature envelopes.

## References

[1] [http://www.blender.org/cms/Blender\\_2\\_40\\_alpha.598.0.html](http://www.blender.org/cms/Blender_2_40_alpha.598.0.html)

---

# Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Armature/object

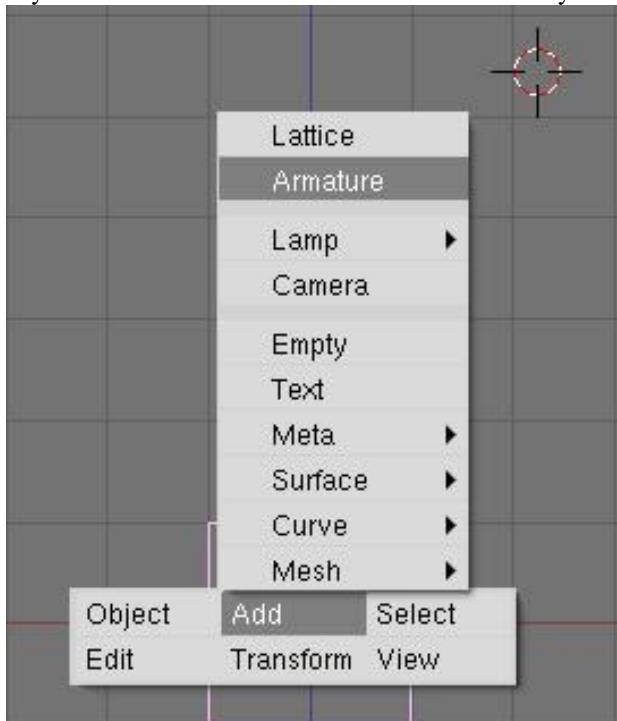
---

## The Armature Object

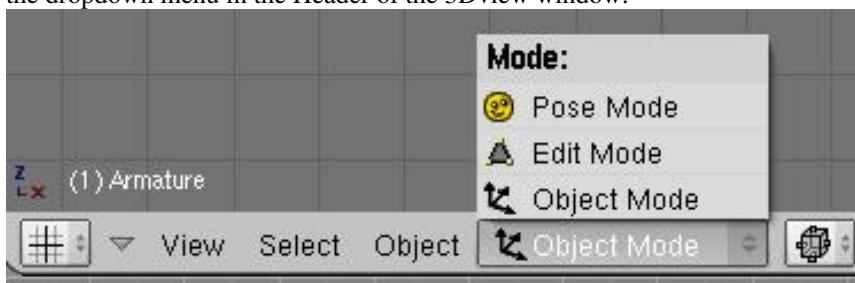
Armature Object is like any other object type:

- 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.

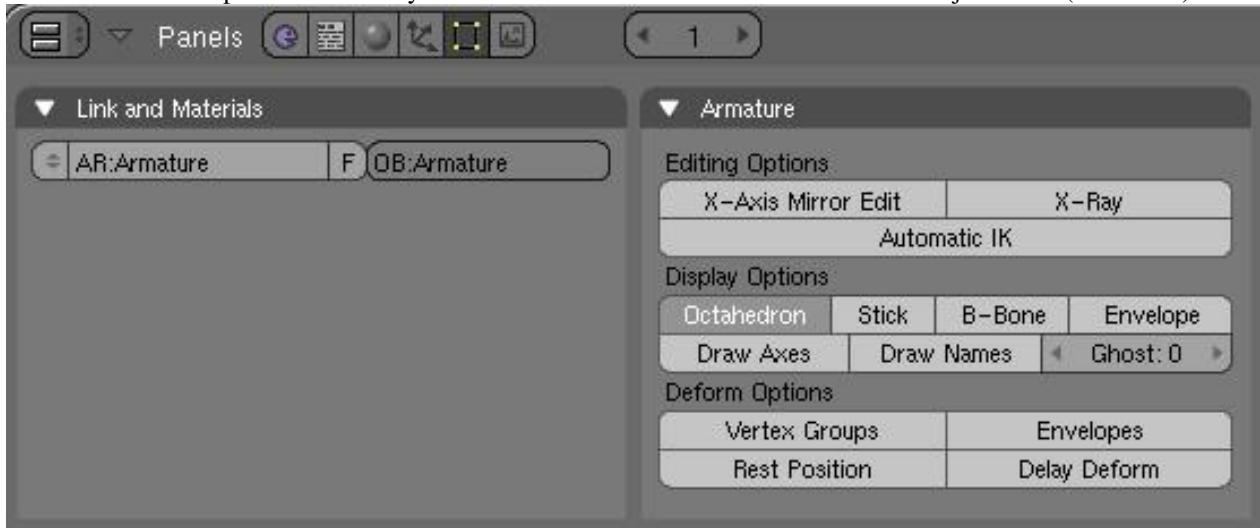


When you add a new armature, you'll enter editmode automatically. To switch between modes, use the TABKEY or the dropdown menu in the Header of the 3Dview window:



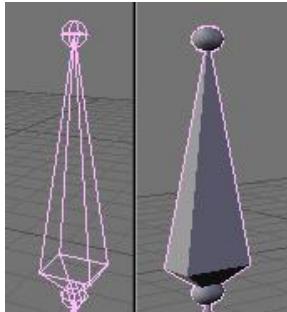
## The Edit Panel When in Object Mode

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.

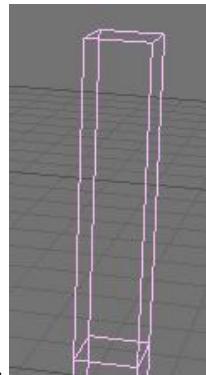


- Stick: This display mode is really useful when you have a lot of bones in your view. It lets you "unclutter"



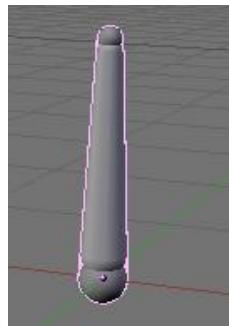
the screen a bit. It draws the bones as tiny sticks.

- 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



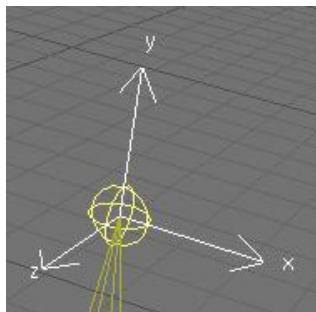
curvy poses. This will be exposed in the following pages.

- 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 your 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

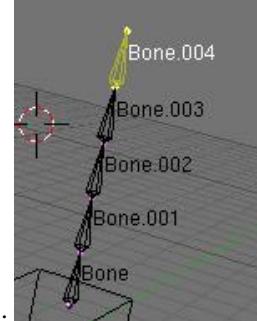


visible in Editmode or Posemode though.

- 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.

## Extra Practice

This YouTube tutorial might also help: Link <sup>[1]</sup> The short tutorial might be a help Link 2 <sup>[2]</sup>

## References

- [1] <http://youtube.com/watch?v=GbFvrlssNo>  
 [2] <http://www.youtube.com/watch?v=Tl-tvJVokr8>

# Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Armature/edit

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 corresponds 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.

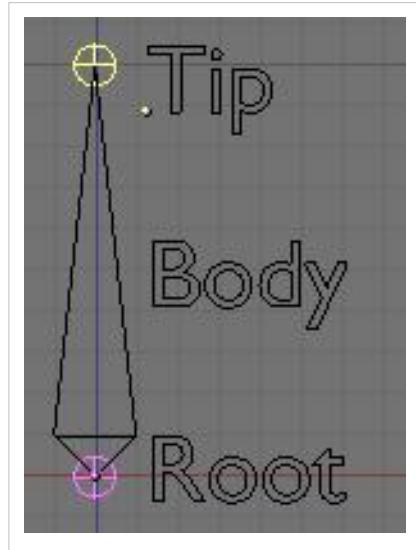
Editing an Armature Object gives you the chance to add, move or connect bones together. Whilst in edit mode, you will see all of the bones within the currently selected Armature.

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.

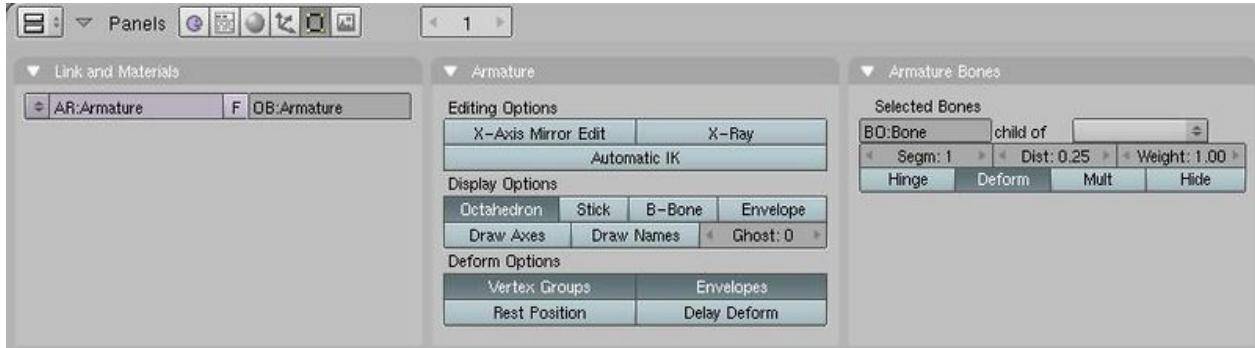
## Now the basics about bones

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 first - 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.

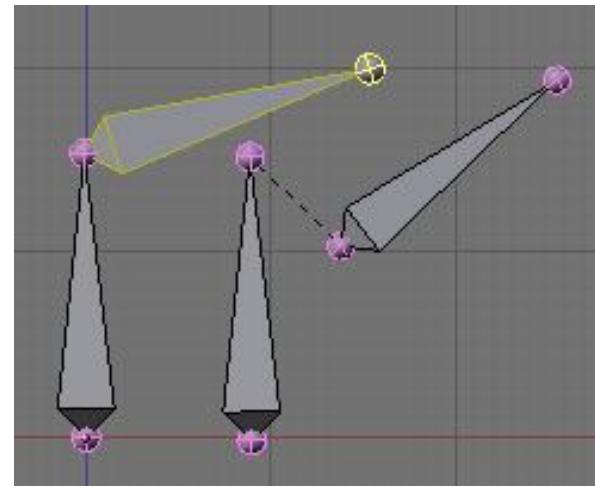


## The edit panel

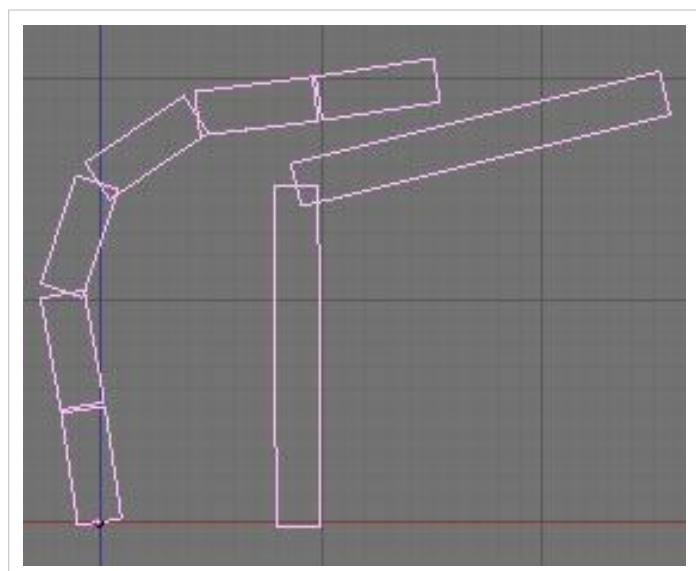


- Armature Bones Panel

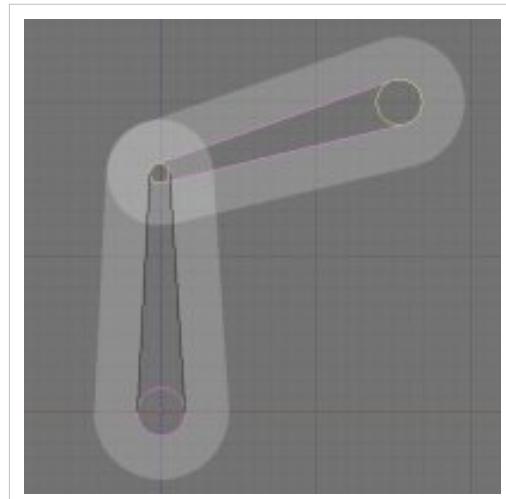
- OB: this field lets you rename your bone.
- "Child of" Dropdown: lets you choose which bone will be the parent of this bone. If a parent is selected, there will be a small button labelled "con", meaning connected. Setting the parent defines the relationship between your bones. When one bone has another 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 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):



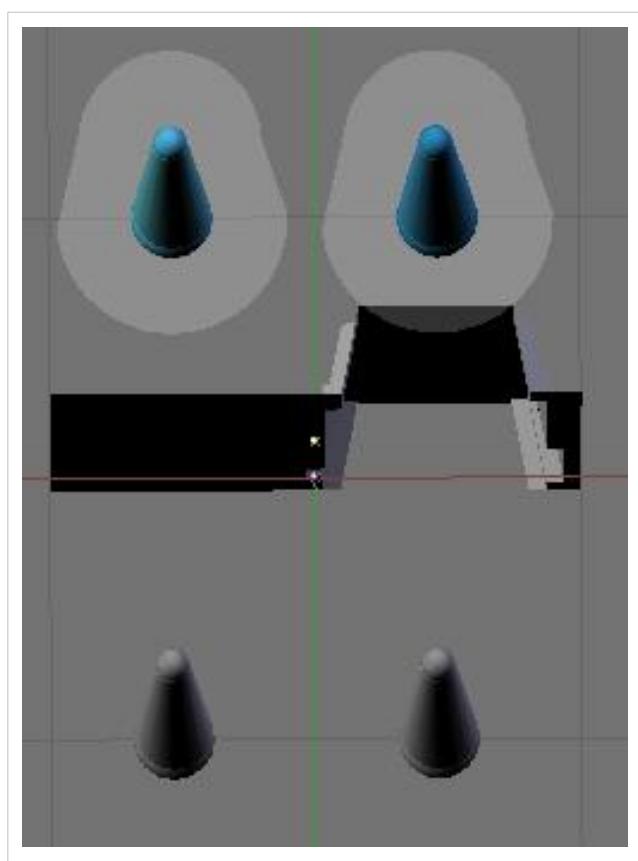
- Dist: This is the area of influence of the bone. It can be visualised using the Envelope display mode. We generally don't touch this field as there is an easier and faster way to change this option. Turn Envelope on and select a bone. Then using ALT-S, you can scale the zone of influence. This has the advantage that you can do it on multiple bones simultaneously, and it works in both editmode and posemode:



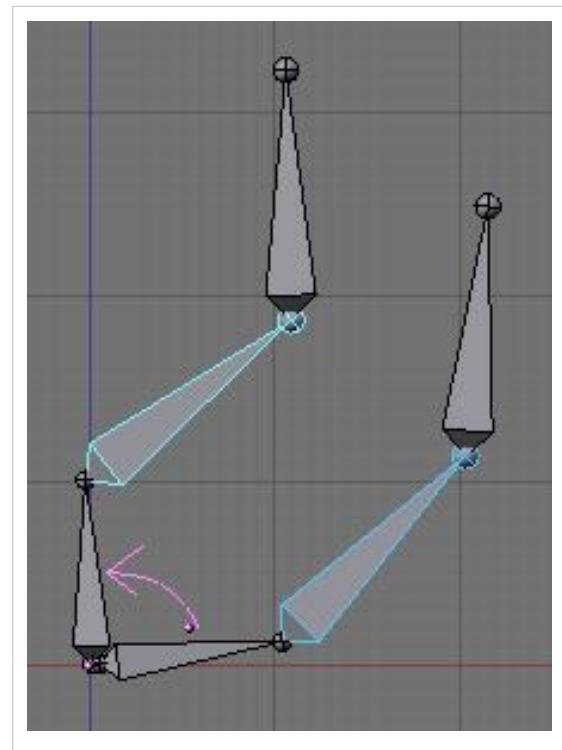
- Weight: This specifies how strongly this bone will influence the geometry around it, relative to the other bones. If two bones crossing each other, both with envelope influence, have the same weight (like 1:1) they will influence the surrounding geometry equally. But if you set one to 0.5, the geometry 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.

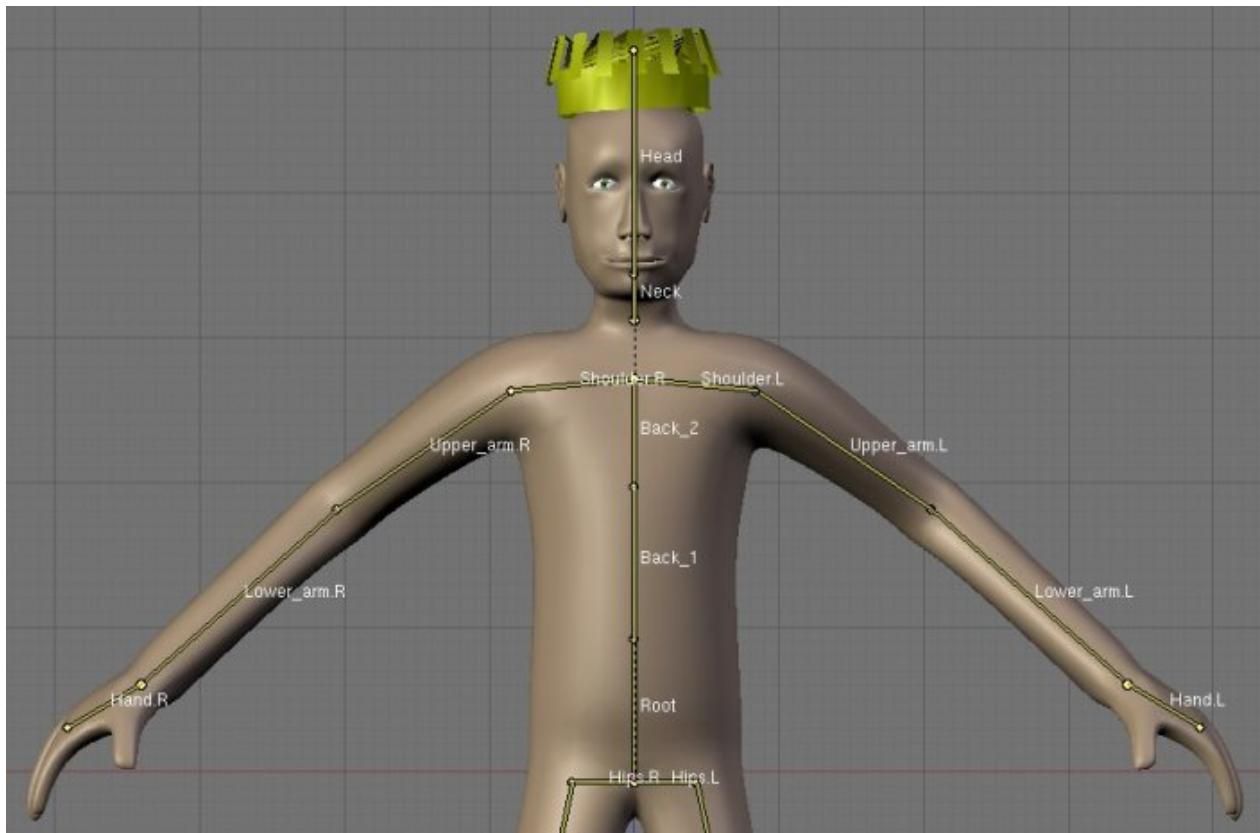


## Naming convention

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:



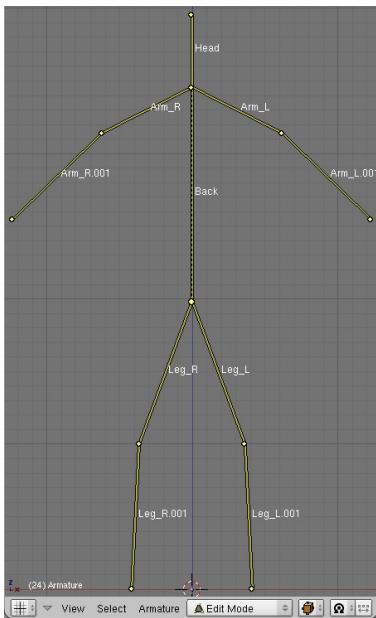
## Mirror Editing

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.

- Create a new, empty scene. Since the mirror editing 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. Add a new armature to the scene (SPACE, Add|Armature). Enable 'Draw names' from the 'Display options' section of the Editbutton panel, so we can see what we're doing.
- Now enable mirror editing by pressing F9 on the buttons window and clicking the **X-Axis Mirror** button in the **Armature** panel under **Editing options**. 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 your mouse over the name field and do **CTRL-CKEY**. You just copied the name of the bone! Select the other bone, hover your 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 another 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 understands 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 legs 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. In the new version, editmode and posemode have an *active bone*. The active bone is the last bone you selected. 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*. For now use *Keep offset*. Do this for the other leg as well.
- 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!

# Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Armature/pose

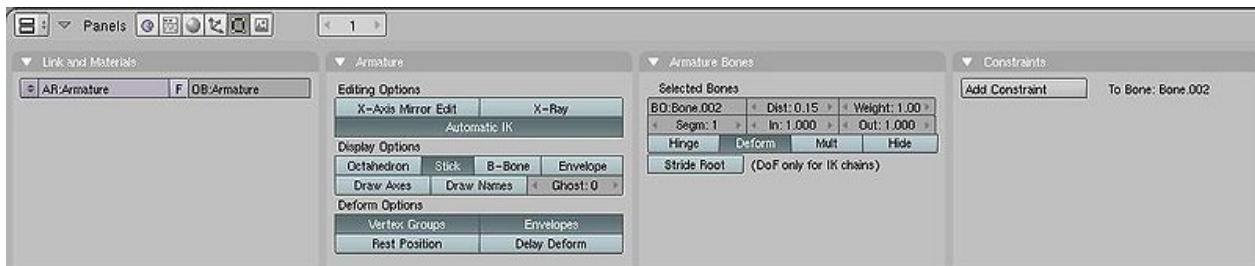
Posemode is a very versatile place where you Animate your character, create and manage constraints and apply your rig to your character.

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?

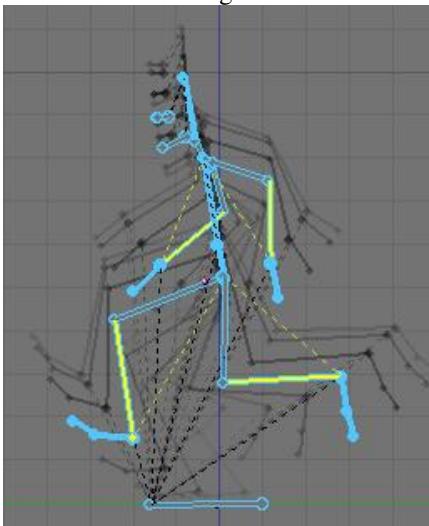
When you are done building your armature, you can go into Posemode to add constraints and start creating actions. There are also some new tools accessible in Posemode that you may want to look at. You can easily get into "pose" mode by selecting the mode from IPO type list box in the left portion of the lower screen.

The panel has changed a bit too:

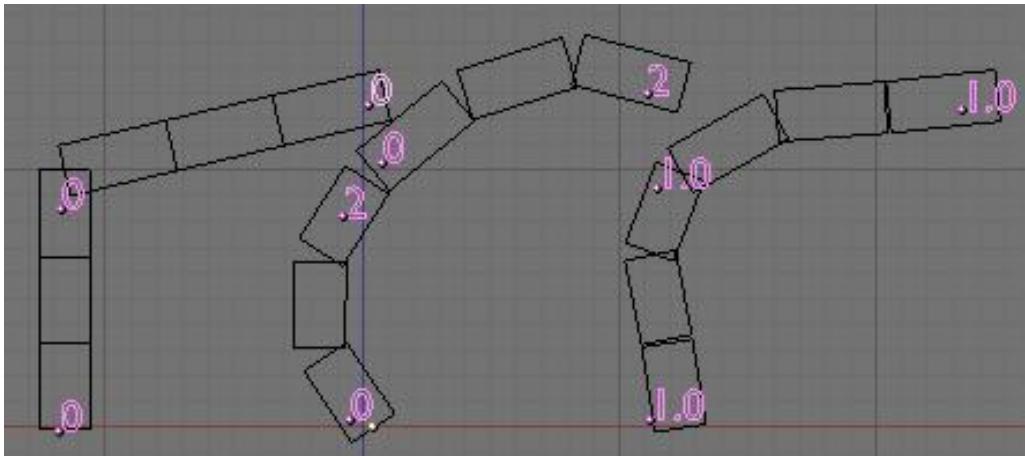


- What's new in the panels?:

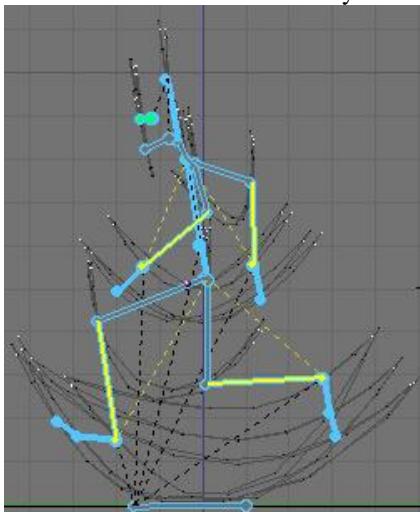
- You can use the Automatic IK feature in the Editbutton(F9) to pose a chain of bones like it was an ik chain. Its usefulness is very limited though. It works well only if there is no other ik solver in the chain, and if your chain is isolated from the rest of the rig.
- Ghost: in the armature panel 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 bezier 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.

## Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/Mesh/index

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.

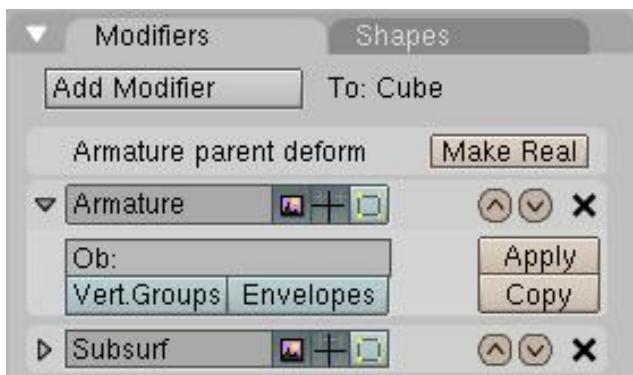
- Connection between Armature and Mesh
- Envelope
- Vertex Groups & Weight Paint
- Shape Key

## Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/Mesh/ Amodif

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. Change the **OB:** field to the name of the armature object that should control the mesh. This step is very important! Without the armature object being defined, Blender won't know how to modify the mesh since there may be multiple armatures within your world. To limit the effect of the modifier, you can enter the name of a vertex group in the **VGroup:** field. This will minimize unwanted distortions in very dense meshes. There are also fields to 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

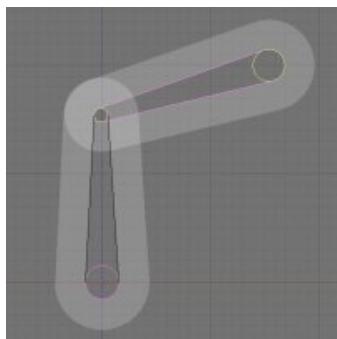
The workflow is very simple. When you are done with the envelope's tweaking and you have gotten the best out of it, delete the Armature modifier and parent the mesh to the armature. To parent it, go to object mode, first select the mesh and then the armature, then press CTRL-PKEY. Select *Create From 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.

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/Mesh/env

## What is Envelope

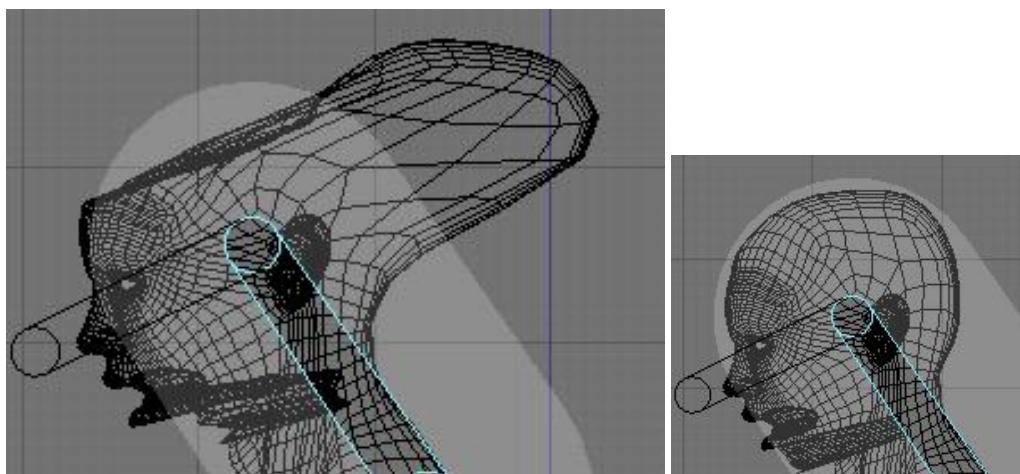
Envelope is a new visual tool to help you rig your characters faster and easier. It can often save you a lot of time. Each bone has a special area around it, allowing you to tell Blender what part of the geometry will follow each bone. This zone is customizable so you can move, scale and blend them together.



## Edit Envelope

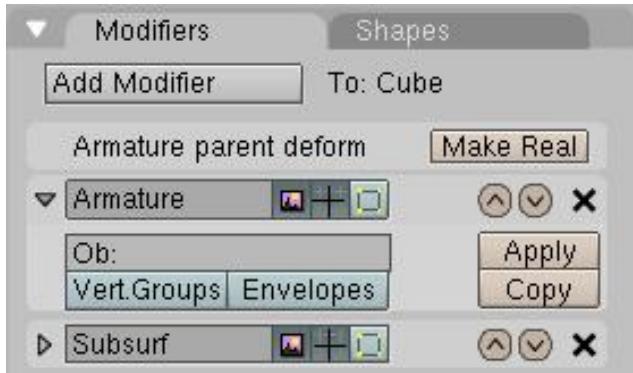
You can edit this white zone in Editmode or posemode by going in Envelope display mode, selecting bones and using SKEY or ALT-SKEY.

*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 transparent area around the center is a zone of influence which loses power as you go away from the center. 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 real time, 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:

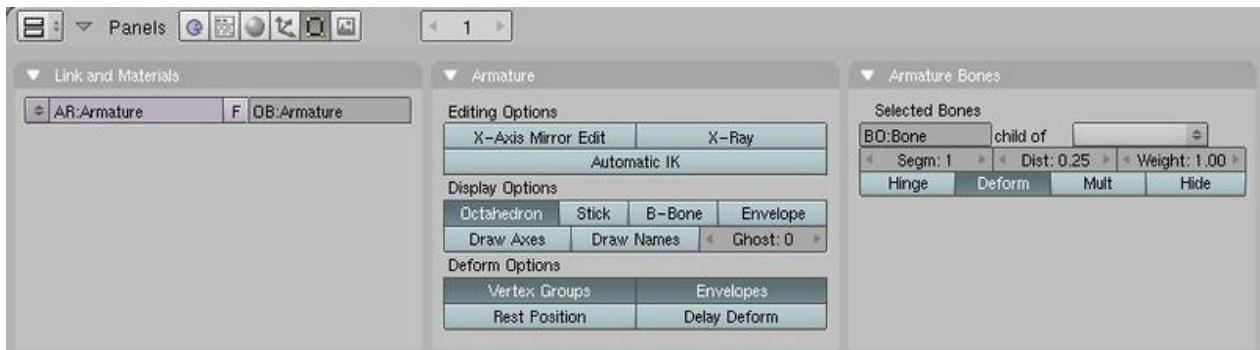


## Envelope Options

It's possible to enable/disable the use of Envelope in the Modifier stack using the "Envelope" toggle.

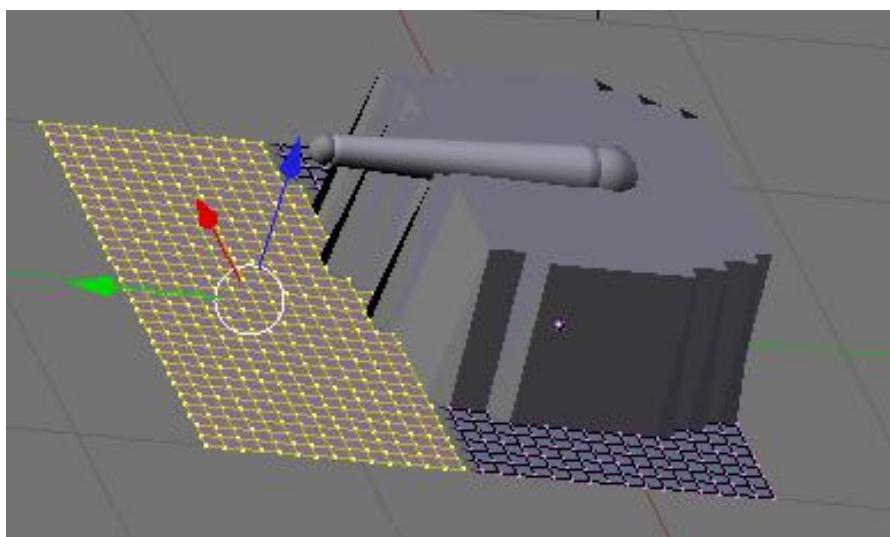


There are also two important buttons in the Armature Bones panel: Deform and Mult.



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 gets from envelope (let's say 0.7) with the weight you painted in weight paint (let's say 0.5). The result will be  $0.5 \times 0.7 = 0.35$  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  $0 \times (\text{something})$  will always give 0. This way you can give custom shape to your envelope. More on weight paint on next page.



In this example of a formerly flat surface you can see that all the selected vertices are not following the bone when it is moved upwards. 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.

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/Mesh/vg

---

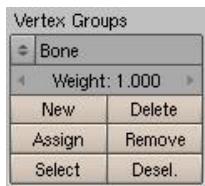
## What Are Vertex Groups?

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:

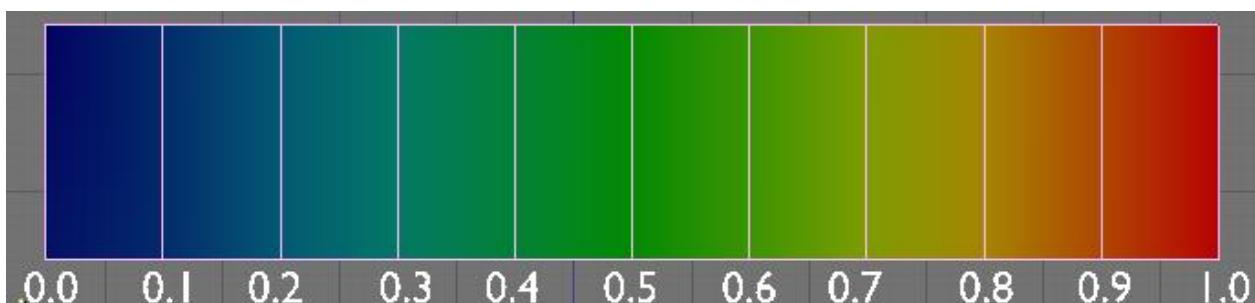
- The dropdown menu lets you select an existing vertex group or rename the current one.
- The weight numfield lets you choose the weight value assigned when you add vertices.
- You can add a new group or delete the current one.
- Assign or remove selected vertices to/from current group.
- Select/deselect 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 vertices 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 your 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 a good effect, use a small opacity value so the weight will increase slowly.
- "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.

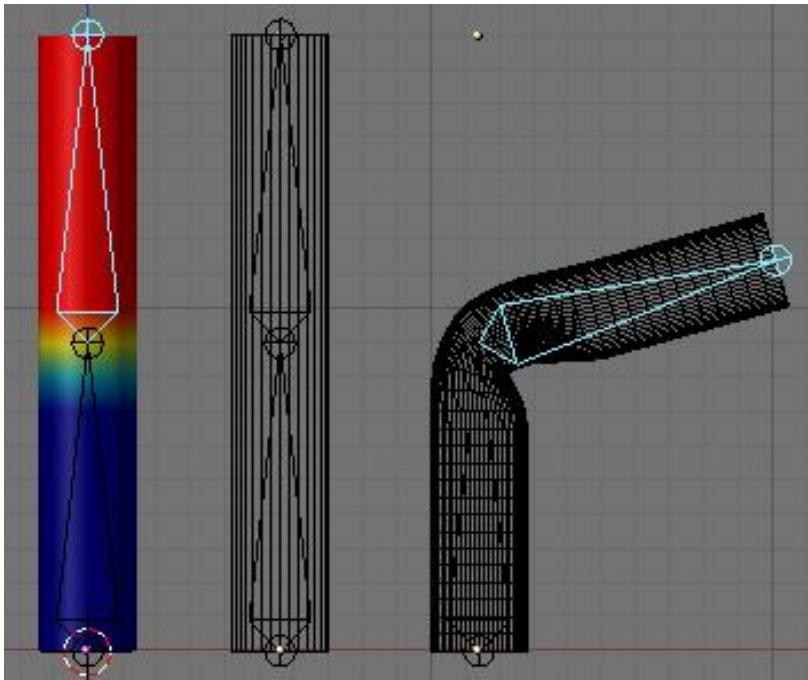
## Vertex Groups and Armatures

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.
- Select the armature and enable Armature Posemode (CTRL-TABKEY).
- Select the mesh and enter Weightpaint mode (CTRL-TABKEY).
- Select the bone you want to paint for with RMB.
- Paint the parts you want that bone to affect.

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:

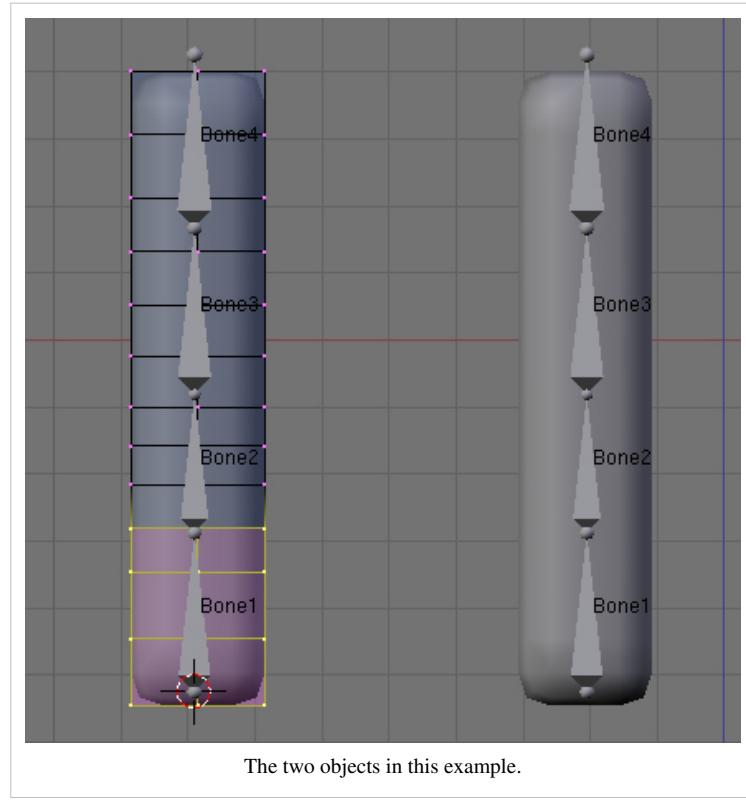


The Groups are painted so the body of each bone is in red and the zone between the two bones are gradually going from 1 to 0. This will bend nicely. If, for a special reason, you want a side to react differently, you can always move the bone while painting and try the new modification you just did. By the way, having Subsurf on while painting can be very cpu expensive. It's a good idea to turn it off.

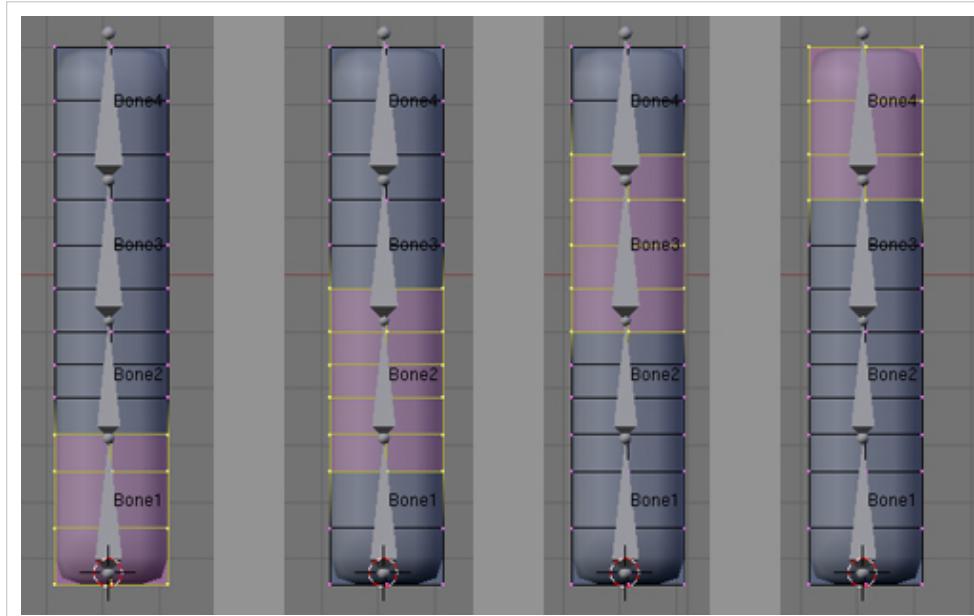
## Using Weight Painting with Armatures

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".



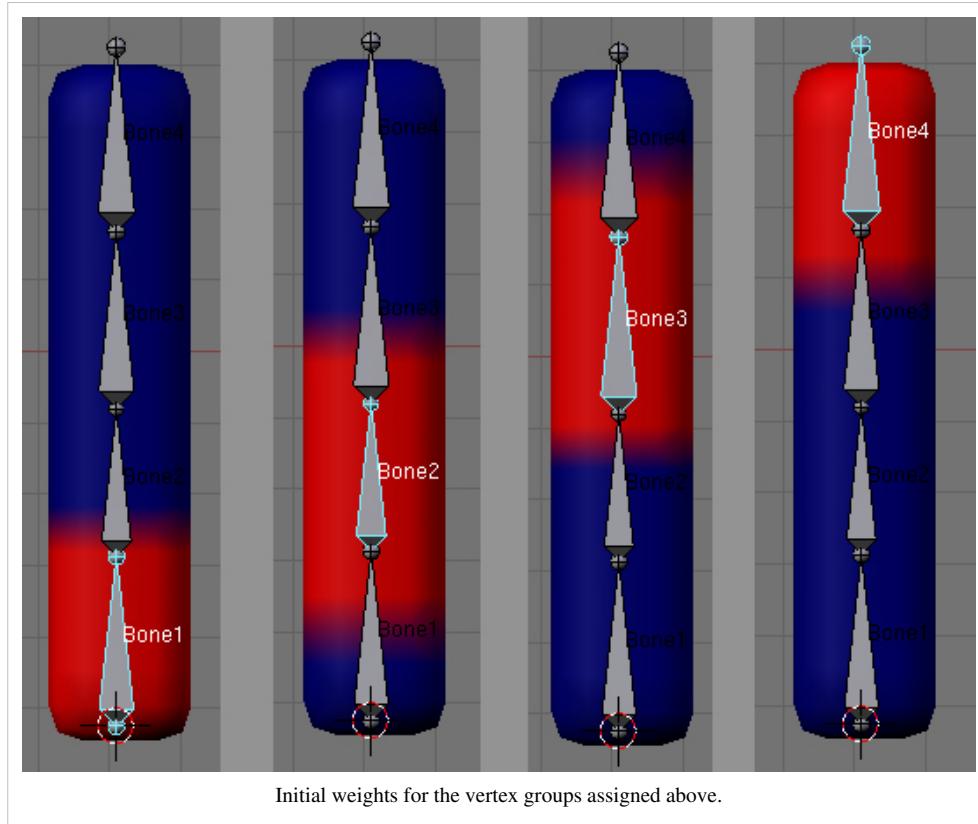
For the object on the left, take a look at the vertex groups as initially assigned (from left to right: Bone1, Bone2, Bone3, and Bone4). These same vertex groups were assigned for the object on the right:



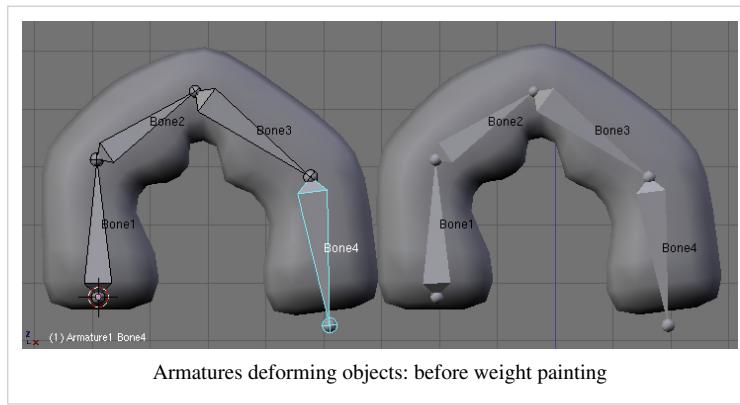
**Important:** A bone in an armature will only act upon those vertices that are in a vertex group with exactly the same name as the bone.

- 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:



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.



# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/Mesh/Shape

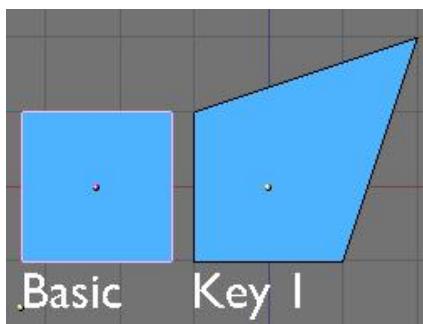
## Shape Key?

Shape keys are modifications of the original mesh that you can animate and mix with each other. Previously called Relative Vertex Keys (RVK), one of their major uses is to create facial expressions.

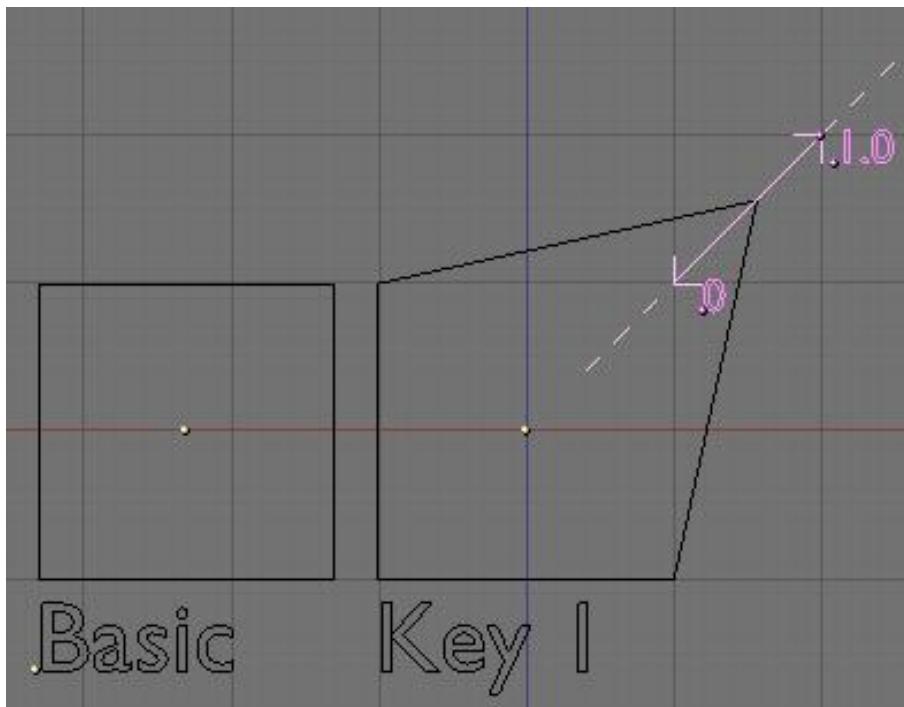


The idea behind Shape keys is simple. The basic, undeformed mesh you start with is the "**Basis**" shape. You can add a variety of different versions of this shape and store each one as a Shape Key for use in an animation (other software sometimes refers to these as "morph targets"). You can't add or delete vertices as Shape Keys only store the positions of vertices and not the creation/deletion of them.

Ok, to start out, take a plane. I'll add a new shape to it and move a vertex away:

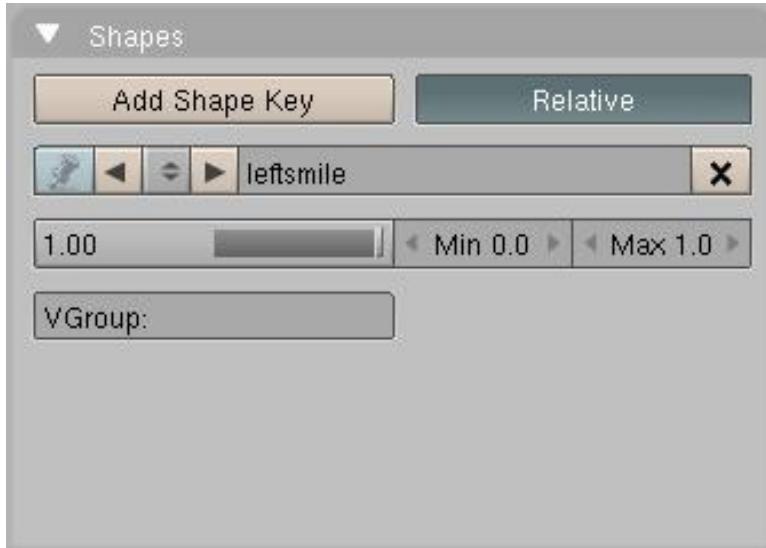


When I play with the influence, this key has over the basic shape, the result will be as follows (0.5 in this example):



If I play with the slider from 0 to 1, Blender will gradually apply the Key 1 shape with a constantly varying amount of deformation. So from 0 to 1, it will slide from the basis shape to the Key 1 shape. The movement of the vertices is linear from start position to end position and you can even set an influence greater than 1 or lower than 0 and Blender will extrapolate the movements of the vertices relative to the basis shape (marked as a dotted line above). This feature allows you to set general shapes such as a smile and then exaggerate the shape without needing to remake it.

## The GUI



## Shape Keys step-by-step

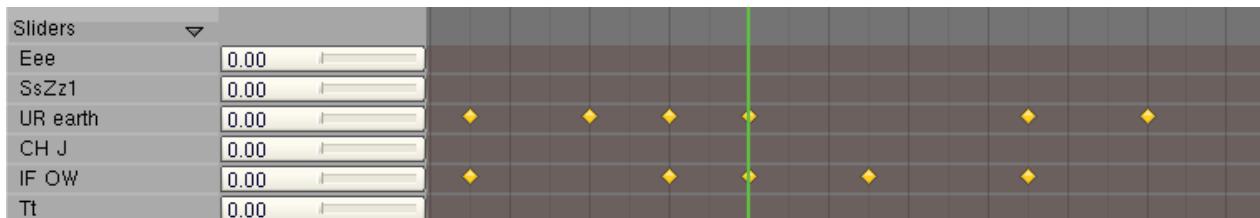
Here's a hand-held walk-through of how shape keys are used.

Start with the default cube or a more interesting mesh of your choice.

- 1: Select your object in **Object mode**. Go to **F9 Editing** window. Find and select the "**Shapes**" panel. Press the "**Add Shape key**" button. This adds a key called "**Basis**" and this stores the "*basic*" mesh in its undeformed state. Make sure the "**Relative**" button is pressed (should be default).
- 2: Press the "**Add Shape key**" button again. It now says "**Key 1**" and you have a slider and some other settings. Go into **Edit Mode**, grab a vertex and move it. **Exit Edit Mode**. The mesh returns to normal but you've just added a real Shape key and Blender has stored it.
- 3: Repeat step 2 for as many different shapes as you like. You can rename the keys to whatever you want. Normally you create each new shape by first selecting the "**Basis**" key but if you select an existing key from the list and immediately press "**Add Shape Key**" then enter Edit Mode, the mesh will already be deformed. This is useful for making similar but unique keys.

## Using Shape Keys

- 1: Starting on frame 1, select each key one by one from the pop-up list (*or go to the Action Window, press the sliders button and select the Keys from the list there*) and click on the slider and move the slider forward then back to zero. Sometimes just clicking on the slider at the zero point is enough to set the key.
- 2: Move forward ten frames, select Key 1 from the list and move the slider to 1.00. You'll see your object deform. Move more frames and slide another key. And so on and so on. You can move them forwards and/or backwards as you move through the frames and they'll just add together if they have to. Each time you move the slider, you set a keyframe for that Shape Key.



*Setting Shape Keys*

### IMPORTANT NOTE:

You should add shape keys to a finished mesh. Don't work on a mirrored mesh or a partially finished model. Adding geometry (vertices, faces, loops, joining etc...) can result in the loss of the shape keys or to unpredictable results. Not always, but probably when you least expect it. There are scripts available to make some of these things possible. That's the basics. Note that shapes will constantly transform between keys so if you set a key at frame 1 to zero and at frame 10 you set the slider to 1 then at frame 5 the slider will be at 0.5 - which you may or may not want. If you want it to hold shape for a while before changing (*e.g. staying at zero until frame 7*), you'll need to set a key at the beginning and end of the time frame you want it to stay the same (*So you would set it a zero at the start, then zero again at frame 7 then to one at frame 10*). You can also edit the IPO curves (*use Shapes pop-up*) to really mess with the transformations.

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/Mesh/Shape/ Sync

## Lip-Sync with Shape Keys

Here I will attempt to explain my recent dealings with using Blender Shape Keys to produce convincing lip-sync (Lip-synchronisation, ie: "speech") for simple, humanoid characters.

This is aimed at people with an understanding of Blender fundamentals like *vertex loops*, *face loops*, *sequencer* and of course, Blender's new *Shape Key* system. If these terms mean nothing to you, then you may well struggle to keep up. If you're familiar with them then I hope this tutorial will prove to be a breeze and your characters will be speaking so fluently you'll have trouble shutting them up!

Other Lip-sync tutorials, if you can find them, recommend using other software like Magpie, Papagayo and others but while I've no doubt they provide a valuable service and maybe make syncing easier, I will be using only Blender for this tutorial. I haven't tried using lip-sync software yet so I can't really say if it's easier or not.

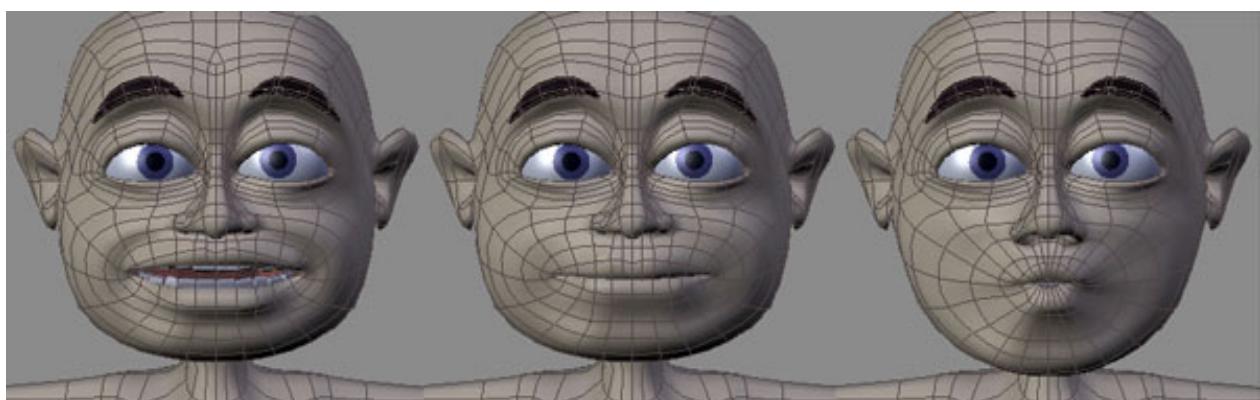
You will find links to interesting audio files you can use for testing animation and lip-sync near the bottom of the page.

## The hard work first

### Setting up for Lip-Sync

First, set up your Blender screen so you have everything you need for this project. You'll need a front 3D Window, an Action Window, the Buttons Window (showing Editing - F9 - panels) and a Timeline Window. If you've got the room, a Side-view 3D Window would be handy too.

The basic sound units are called **phonemes** and the mouth shapes we use to represent these phonemes for lip-sync are called **visemes** (*or sometimes, phoneme shapes*) and there are many references for these around the web. Legendary animator Preston Blair created one of the most popular viseme sets <sup>[1]</sup>, which is great for cartoon-style characters. Other visemes are aimed at more-realistic, humanoid characters. The shapes you choose depend on the style of your model. (*I'll try to provide some useful links later*)



Sample viseme shapes.

The first and most difficult step in good lip-sync is making the shape keys for these visemes. How well these are made ultimately determines the success of the animation and it is worth spending time getting them right, although they can be modified and tweaked later. So choose your favourite set of visemes (*or even look in a mirror and use your own face as a reference*) and start setting your keys. A model with good **topology** - especially well formed **edge**

loops around the mouth area - will be a big help here!

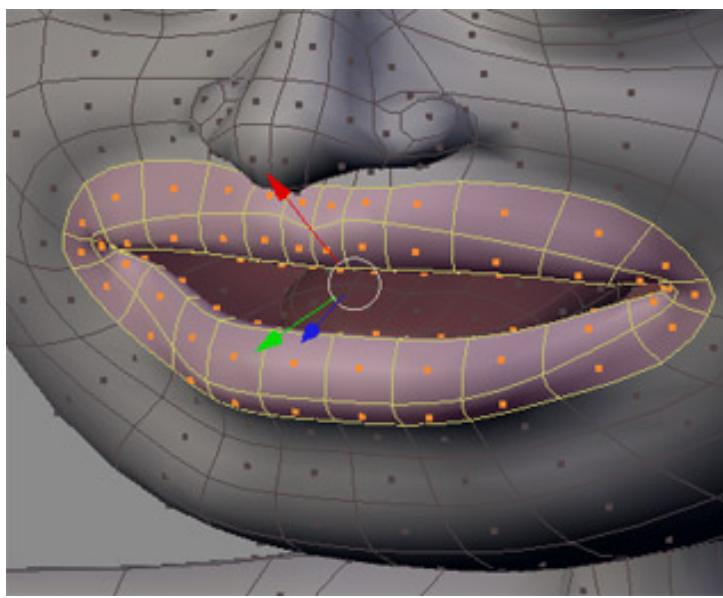
### What on Earth are topology and edge loops?!?!

Topology refers to the way your 3D mesh defines the object, in this case your character's head. Edge loops refers to the flow of consecutive edges around the character's major features. Typically, good edge loops flow around the eyes and around the mouth in a somewhat deformed, "circular" manner. Selecting and deforming a single loop of edges and vertices surrounding a facial feature is much quicker than having to individually select a lot of edges that don't flow naturally around that feature. Similarly, deforming a single loop or collection of related, nested loops is much easier and quicker than trying to deform a seemingly random set of edges. You can easily see these loops in the character screenshots above. The series of edges surrounding the mouth simply stretch from wide ellipses to almost circular to create a useful variety of mouth shapes. The faces defined by paired edge loops are referred to as **face loops**. The close-up image below shows selected face loops.

There are many tutorials on the web about these topics so if you need more information before proceeding then a quick search is probably a good idea.

### Setting the basic viseme shape keys

First, select your undeformed mesh and create your basis key. Go to **Editing** (F9) window and go to the "**Shapes**" panel. Press the "**Add Shape Key**" button. Enter and exit edit mode to set the basis key. Then create your first key **Key 1** and name it "M". Enter edit mode and if your character's mouth isn't already closed (*some people make them that way*) then close it by carefully selecting the faces and loops around the mouth. You will usually use Size-Z plus a bit of grabbing and shifting to achieve this. Don't forget to include the faces on the inside of the lips or the deformation will be unpleasant. When you're happy with the shape, exit Edit mode and there you have it. Your character can now say "*Mmmmmm*" whenever you like. Test it by moving the Shape Key slider back and forth but make sure to leave it at zero before making more keys. (*If you made your character with a closed mouth then you can just add the new "M" key then enter and exit edit mode to set it.*)



#### Selecting face loops

For most new keys, you will select the basis key first then press "**Add Shape Key**" then make the required shape from scratch in edit mode. However, some mouth shapes are very similar, like "OH and OOO" or "EE and SS" and it would be easier if you could start with something close to what you want rather than shifting everything from scratch every time. Luckily, Blender allows you to do just this. Once you've made your "EE" key, for example, you can select it and immediately press "**Add Shape Key**" then enter edit mode and the mouth will already be deformed and you only need to make subtle changes to it to make your "S" shape.

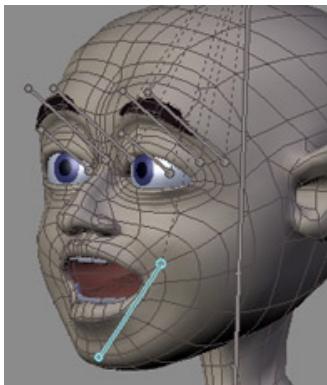
**Remember that the keys you make are for sounds, not letters.**

In general, you'll need shape keys for the following sounds

- **M, B, P**
- **EEE** (*long "E" sound as in "Sheep"*)
- **Err** (*As in "Earth", "Fur", Long-er" - also covers phonemes like "H" in "Hello"*)
- **Eye, Ay** (*As in "Fly" and "Stay"*)
- **i** (*Short "I" as in "it", "Fit"*)
- **Oh** (*As in "Slow"*)
- **OOO, W** (*As in "Moo" and "Went"*)
- **Y, Ch, J** (*"You, Chew, Jalopy"*)
- **F, V**

These can often be used in combination with each other and with jaw bone rotations and tongue actions to also produce other sounds like "S", "R", "T", "Th" and a short "O" as in "Hot" - or these can also be specifically made with their own shapes. This decision depends largely on your character. Start with the essentials and make others if you need them.

**NOTE:** I use one jawbone in my current character and this is also used to control the mouth shapes. It doesn't drive the shapes but it moves the bottom teeth and tongue (*which can also be controlled separately*) and the faces that make up the chin part of the character mesh. For some visemes, I move the jawbone into a logical position before adding the shape key. For example. I open the jaw for the "OH" key and close it for the "M" key. Later, when animating, the jawbone is animated along with the Shapes for a very convincing result.

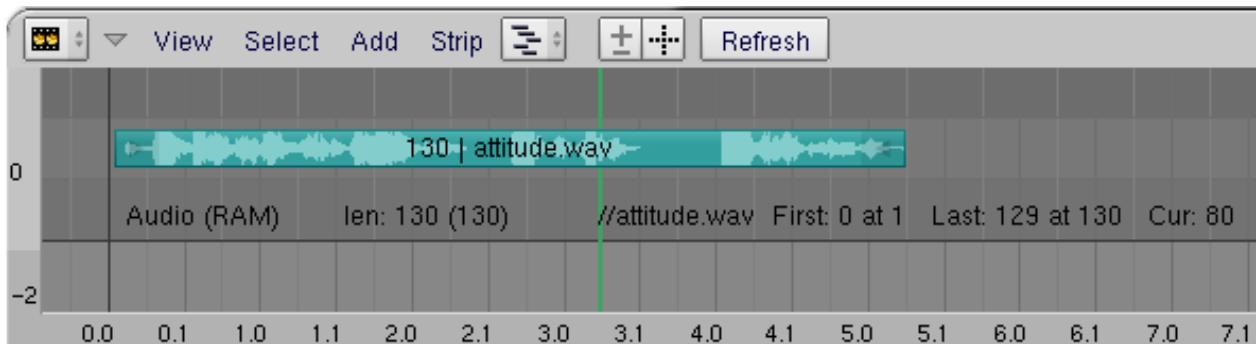


*Using jawbone with shapes*

**Load the audio**

Once all your viseme shapes are set the time comes to get your character to speak. (*Normally you would animate the body first and leave the lip-sync till near the end.*)

If you haven't already done so, load your audio file into a Blender **Video Sequencer Window** and position it where you want it. Currently, Blender only supports 16 bit wav format audio files so you may need some editing or conversion software to process the file if it isn't in this format. "Audacity" is a good, open-source (free) editor that will suit this purpose and much more.



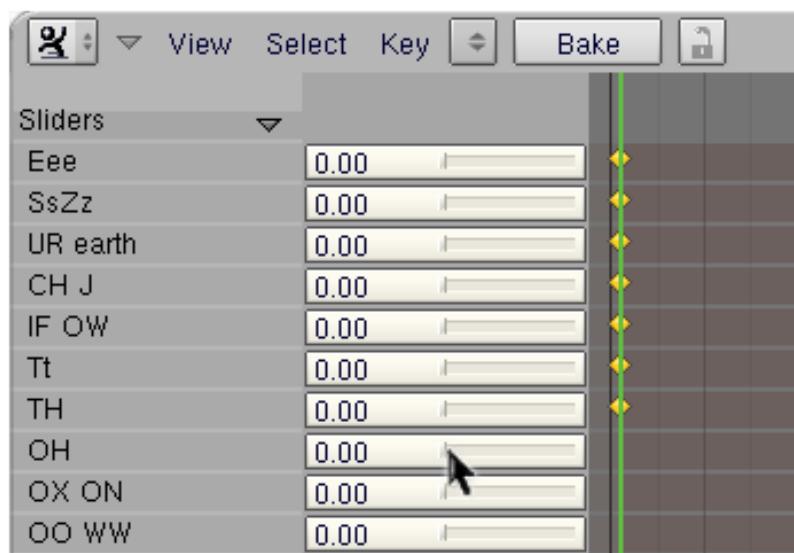
*Blender Video Sequence Editor Window showing loaded Audio Strip*

Go to the Scene Window (F10) and press the "Sound block buttons" button --(in the three buttons near the frame counter. It looks like an audio wave)--. Press the "Sync" and "Scrub" buttons. The "Sync" button makes the playback in the 3D window closely match the audio when you press **Alt-A** (*it does this by dropping image frames if necessary and it generally provides only a rough guide of how things match-up*). The "Scrub" button is important for lip-sync as it means that whenever you change frames, Blender plays the audio at that point (*Currently in some Blender builds you have to press Alt-A at least once to get this feature to start working*).



*Blender Soundblock Panel*

Select your character and your Shape Keys should be listed in the **Action Window**, in the order in which you made them (*I don't think they can be shuffled*). You will see a small triangle button labelled "Sliders" at the top of the list - press it to show the sliders for each shape. If you drag the mouse back and forth in this window, you should hear the audio play as you cross frames. This is how you will identify the main viseme/phoneme key frames.



*Shape Key Sliders*

## Getting down and dirty

Before you begin your lip-sync, you obviously need to know what your character says. It might be worthwhile writing it down and even speaking it over and over, in time with the audio, to get a feeling for the sounds you'll be dealing with. Some sounds are what I'll call "**key sounds**" and others are almost dead, non-descript "in-between" sounds that fill in the gaps between the key sounds. Obvious key sounds are those where the lips close and those where they are tightly pursed or wide and round, other key sounds can differ with every piece of audio. Don't make assumptions about the shapes you'll use based on the words you know are there. What you are animating are the actual sounds - not letters or words (*Keith Lango has much to say about this on his website and I recommend reading it*)

### The Timeline Window

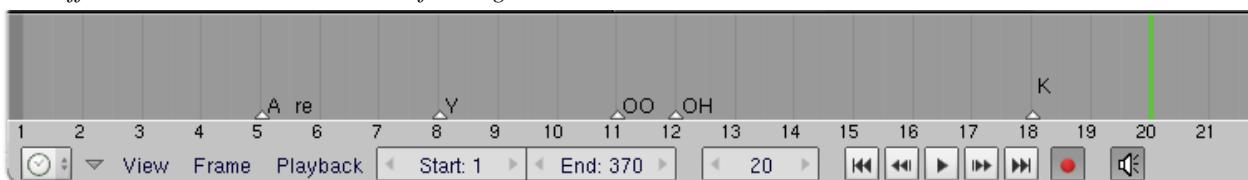
Go to the **Timeline Window**. This window seems to have been largely overlooked in previous documentation yet it provides the basics for timed animation plus a few other goodies to make your animation life a little easier. Here you can turn auto-keying on and off at will (*you'd normally go to the hidden **User Preferences** window which is a pain and easy to forget*), navigate frames, play and pause the animation in the 3D window, turn Audio Sync on and off, set the start and end frames for the animation and **set frame markers**. This last feature is the key to our project.

In recent Blender releases, the audio "Scrub" feature works in most windows. As you scrub through the audio, listen for where the key sounds occur. As you hit a key sound, scrub back and forth to find where that sound commences. In the Timeline Window, press "M" to set a frame marker at this frame. The marker can be labelled by pressing CTRL-M in the Timeline Window while the marker is selected (yellow). Enter a sensible name for the marker that indicates what the marker is for (*like the phoneme sound and/or which word it starts or ends*). Markers are selectable and moveable and can be deleted just like other Blender items.

**NOTE:** Depending on the speed of your system, you may find you get more pleasing audio scrubbing and better 3D window playback if you turn off subsurf for your model and hide any unimportant parts of the scene on different layers. The fewer things Blender has to calculate as you scrub or play-back, the faster it can draw the frames to maintain sync with the audio.



Turn off these buttons to disable subsurf setting in 3D window



Blender Timeline Window

Scrub through the whole audio - or sections of it in a long piece - marking and labelling all the key sounds.

## Setting the keys

Now you have everything you need for your first lip-sync pass. Start at the first frame and click once on all your **Shape Key sliders** in the Action Window to **set them all to zero**. Move through the frames from start to finish setting shape keys where you marked the key sounds in the Timeline Window. If your character has a jaw bone and tongue bone(s), you will need to set these where required as you go.

### Trouble in paradise? A quick lesson in handling Shape Keys

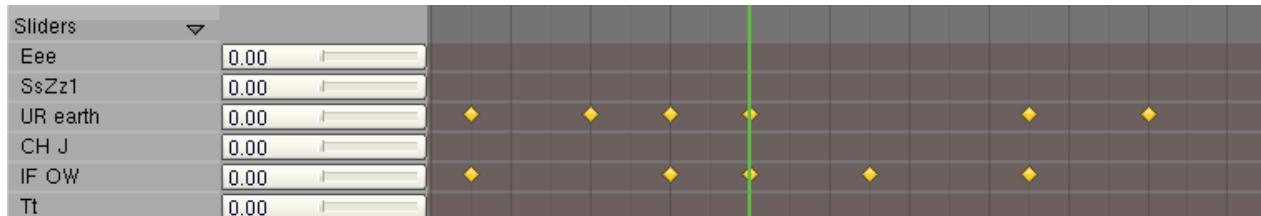
If you haven't set shape keys before you might notice one interesting dilemma - they have great memories! Once you set a slider to any value, it stays at that value until you set another value somewhere. The shapes change in a linear fashion between keys. At first, this appears to be a problem if you want to key "MOO" because after you set the "M" key slider to 1 (one), it will stay there, making it impossible to get your character to say "OO" while his lips are trying to stay shut. So, you have to set the "M" slider on the "M" sound, then as the audio goes into the "OO" sound, you must set the "M" slider back to zero and then set the "OO" slider to its full value.

This introduces another problem. After you set the "OO" key, your "M" sound is messed up because the mouth is now also being affected by the "OO" shape that follows it. So, you must make sure the "OO" sound is set to zero when you want the lips closed in the "M" position.

#### In general, you'll find yourself setting each shape 3 times

- once to determine where you want the mouth to begin moving to this shape (*slider set to zero*)
- once to set the slider at the desired level for this phoneme
- and once more to end this shape and move into the next one (*slider set to zero*)

*The same principle applies to the jaw bone and tongue - 3 keys to each move.*



#### Setting multiple shape keys

As you get more comfortable with this procedure, you'll find you can leave some shapes set longer or adjust them to different levels as they can provide some interesting mouth shapes when combined with other shapes.

## Setting the in-between sounds

Once the key sounds are properly set, you should be able to scrub slowly through the Action Window and watch your character speak in time with the audio. It won't be perfect but it's a start. To fix his speech impediment, you now have to fill in the sounds between the key sounds. Mostly these will be dull vowel sounds ("err, uh, ah") and silence. These shapes are set in exactly the same way but here you'll have to really watch the 3D view as you set the sliders. Try combinations of sliders like "EE" and "OH" to get the perfect shape for each individual sound. You have to be guided by your character. Does he look like he's saying the sound you're hearing? (*Remember that exactly what's being said is not important - it's only the sound that matters*). Test each sound as you set it by scrubbing a few frames over and over and watching your character mouth the sounds.

All that's left, hopefully, is some polishing and tweaking. If it's not perfect then don't despair. Like other areas of animation, it can take a while to get a feeling for lip-sync and as the tools and workflow become more familiar you can pay more attention to the important work.

## Putting it all together

Once you're reasonably happy, it's time to combine the audio and video and watch the result. Since Blender can't do muxing (combined audio/video) you'll need to composite it with the editing software of your choice (OSX users can use recent versions of iMovie, virtualDub is often recommended for Windows users and Avidemux2 is often recommended for Linux users.).

What Blender can do is provide a **fully synced** version of the audio file the same length as the animation - even adding silence at the start and end if need be to maintain the synchronisation. To make this synced audio file, go back to the "Sound block buttons" panel and press "**MIXDOWN**". This saves a .wav file using the filename and location you entered in your output box (*you did didn't you?*) plus a frame count (*something like 'speech.avi0001\_0400.wav'*). Then save your animation by pressing the "**ANIM**" button.

Combine the audio and video in your video editor and export a muxed file. You may find when you play it back that the mouth seems to be just slightly out of sync. This is for two reasons: sound travels much slower than light, so we see the lips move before the sound reaches our ears, and the lag is more the further away the person is (about 3 milliseconds of delay for every metre), and the more exaggerated their expressions as they shout. The second reason is the way the brain processes faces and expressions and mixes it with sounds heard to comprehend speech. This comprehension phenomenon is barely understood and is a common challenge in animation, and some physiologists think our brains read lips and facial expression as a way of setting up to understand the context of sounds received and comprehending the meaning behind language. To solve it, you can go back into Blender, select the audio in the Sequencer Window and move it one or two frames later, then press "MIXDOWN" again to create a new .wav file with a split second of silence at the start. Remix this with your video and watch the results.

From here on it's all a matter of testing and tweaking until you're happy!

## Using Blender to render Audio AND Video TOGETHER.

Ever since Blender 2.3, it has been possible to render video with the audio attached. Once you have you're animation done, and are ready to render. In the buttons window, click scene (F10). In the format tab, choose FFMpeg. There will now be three tabs. Click on Audio. Click Multiplex, and choose your audio codec. Click Video. Choose your video codec (including AVI). Now, in another section of the window, there will be an Anim tab. Click Do Sequence. And click ANIM. When you watch the video, you'll notice that it has sound.

Note: The mac version of blender doesn't have the FFMpeg codec in blender by default.

## Audio Files

You can find some interesting audio files selected for animators at [2]animationmeat. These files come complete with a pre-marked phoneme sheet.

## P.S.

One final note. Watch how the pros do this. When actors are doing voice overs for major releases their actions are recorded and even integrated into the final animation. If you have a digital camera, you may also try recording your own mouth performing your dialogue and approximating its positions to your animation. This can give you a great visual reference, possibly frames for frames if your frame rates match, and save you time.

## References

- [1] [http://www.garycmartin.com/mouth\\_shapes.html](http://www.garycmartin.com/mouth_shapes.html)
- [2] <http://www.animationmeat.com/practice/practice.html>

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/Const/index

## 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.

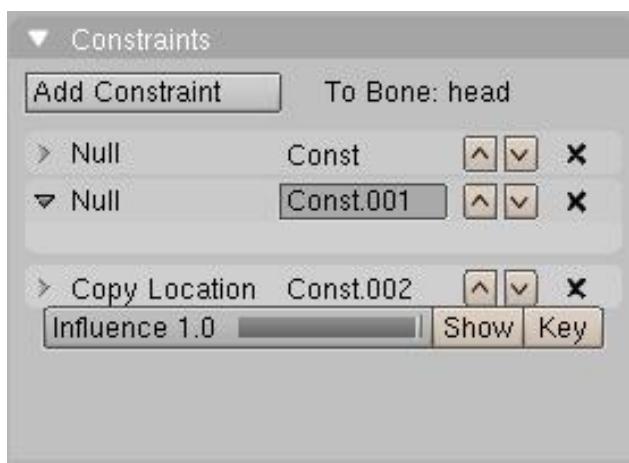
There are no strict rules to follow for when to use constraints. As long as they save you time and make everything work by itself. A constraint should never be "time-consuming" or difficult to use. Think about the animator who is going to work with this rig (it could be you!). So, do everything in a 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 gets added in the stack. The UI is almost the same as the Modifier Stack. Each block represents an entry. You can delete it with "X", move it up or down in the stack, close or open it.

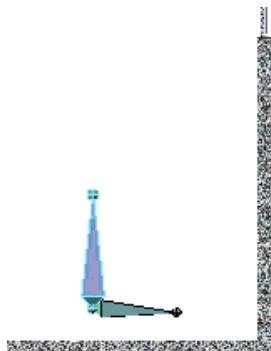
- Constraints are evaluated from first to last. So if you have two Constraints working on the same channel, let say Location, The last one will most probably win the chance to move the object. But...
- Most of the constraints have an influence slider to determine how much it will influence the stack. If the last constraint has 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 slider and adding a key with the **Key** button.
- The **Show** button will display the influence IPO curve in an IPO window for editing. (The IPO window must be opened before pressing the 'show' button).
- 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

# Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/cl

## 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.

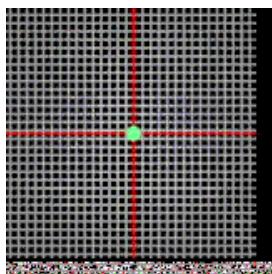
## Where To Use It

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 relationship (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.

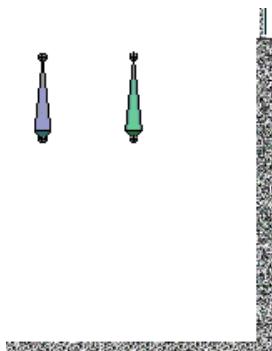
Before Blender 2.40, the above method was a good way of faking parent relationship without rotation. But now we have the hinge option which does the same.

Create this kind of tracking device using the X Y Z toggle button



## Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/Const/cr

### 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

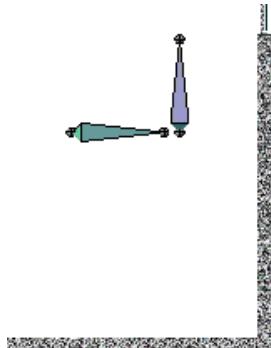
Can be used with Copy Location to fake parent relationship. As you can key the influence you can make a character pickup something and holding it in his hands. Check the .blend for the hand-glass scene.

(?)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.

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/Const/tt

## Track-To



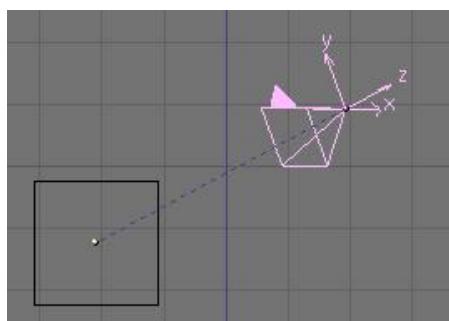
The Track-To constraint lets you influence the Rotation of the constrained object by making it track a target with one of the constrained object's axis.

## The Constraint Panel



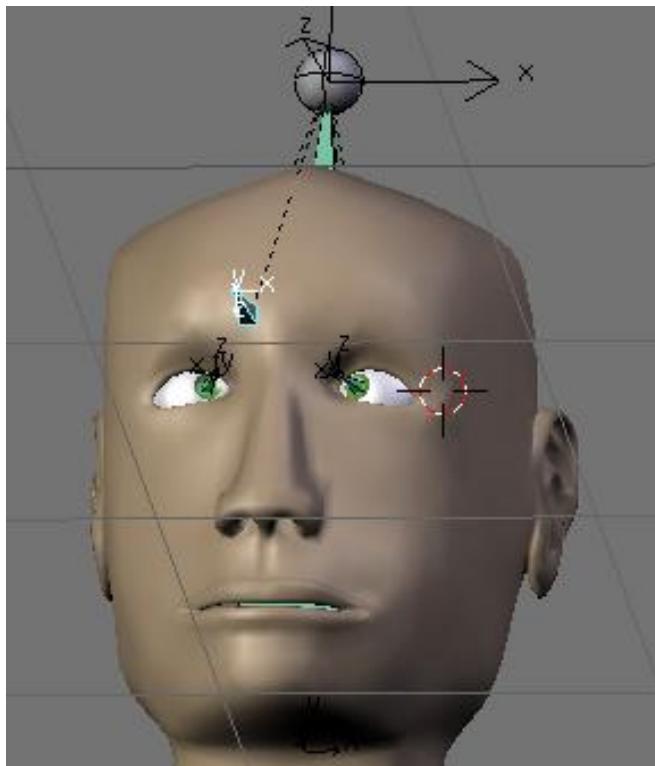
- You can enter the name of the target you want to track.
- You can select which axis is going to track the target.
- You can select which axis is going to stay up.

## 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(?)

## **Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/Const/fl**

---

Using the floor constraint will keep a bone from passing through an object from a given direction. It is useful when making floors of course but also when making walls and other items which are obstacles for the armature.

There is also an offset value which is very useful when say for example you have a foot where the mesh stretches down below the actual tip of the armature you can then use the offset to make the bone stop before it actually reaches the obstacle object.

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/Const/fp

---

Next page: Stretch-To

Locked Track

---

## Follow path

### The Constraint Panel

#### Where To Use It

(?)link to the Blend(?)

---

Next page: Stretch-To

Locked Track

---

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/Const/st

---

Next page: IK Solver

Follow Path

---

## Stretch-To

### Description

Stretch To causes the affected object to scale the Y axis towards a target object. It also has volumetric features, so the affected object can squash down as the target moves closer, or thin out as the target moves farther away. Or you can choose not to make use of this volumetric squash-'n'-stretch feature, by pressing the NONE button. This constraint assumes that the Y axis will be the axis that does the stretching, and doesn't give you the option of using a different one because it would require too many buttons to do so.

This constraint affects object orientation in the same way that Track To does, except this constraint bases the orientation of its poles on the original orientation of the bone! See the page on Tracking for more info. Locked Track also works with this constraint. Options

R

Pressing the R button calculates the rest length as the distance from the centers of the constrained object and its target

### Rest Length

Rest Length determines the size of the object in the rest position

### Volume Variation

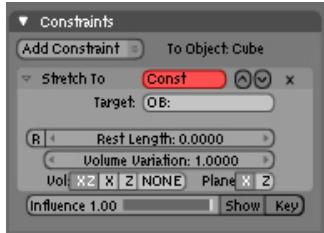
Volume Variation controls the magnitude of the effect

### Vol

The Vol: buttons control along what axes the volume is preserved (if at all)

### Plane

The Plane buttons define which local orientation should be maintained while tracking the target



[Next page: IK Solver](#)

[Follow Path](#)

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/Const/ik

[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).

- 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 during 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.

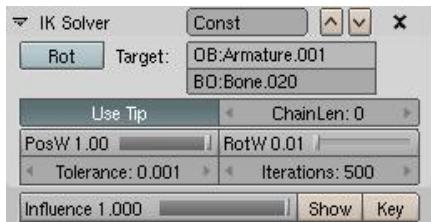
It's also possible to remove all IK constraints from selected objects or bones with '**ALT-IKEY**'.

Q: **ctrl+i** doesn't seem to do anything

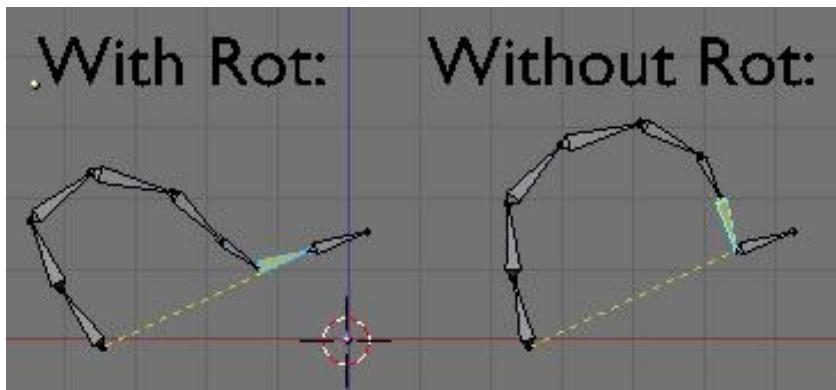
A: Either the 3D Window is out of focus (r-click in empty space to solve) or you're not in Pose Mode (ctrl-tab, selected bone will be magenta in color)

A: in the 2.48 version : the shortcut is '**SHIFT-I**'.

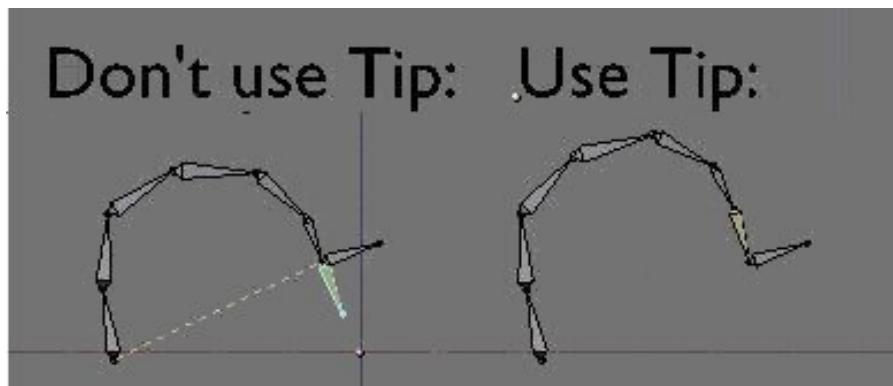
## The Constraint Panel



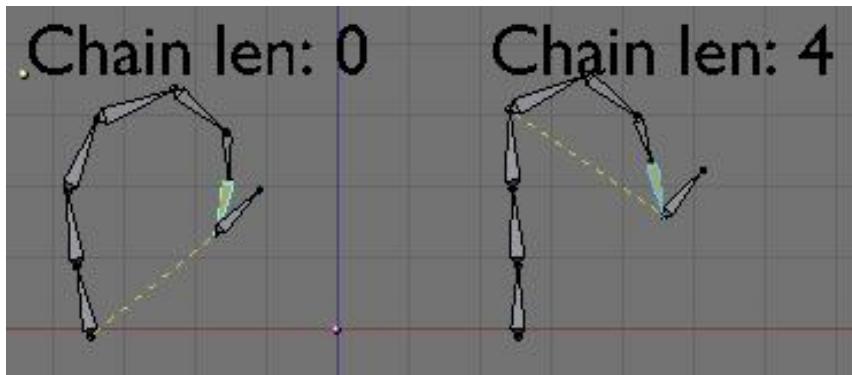
- You can rename the constraint.
- You can select which Object or bone will be the target. Don't forget Tab completion.
- The Rot button let you tell Blender to use the rotation of the target to influence the rest of the chain:



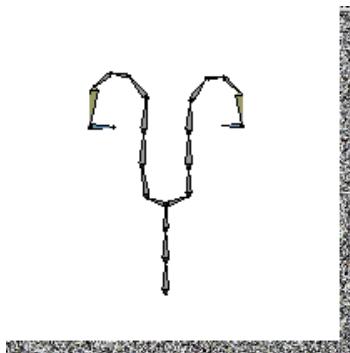
- The Tip button lets you tell Blender which part of the bone is the target, the Tip or the Root. It's interesting to use tip, because this way the Bone holding the IK constraint can be used to deform geometry.



- 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.

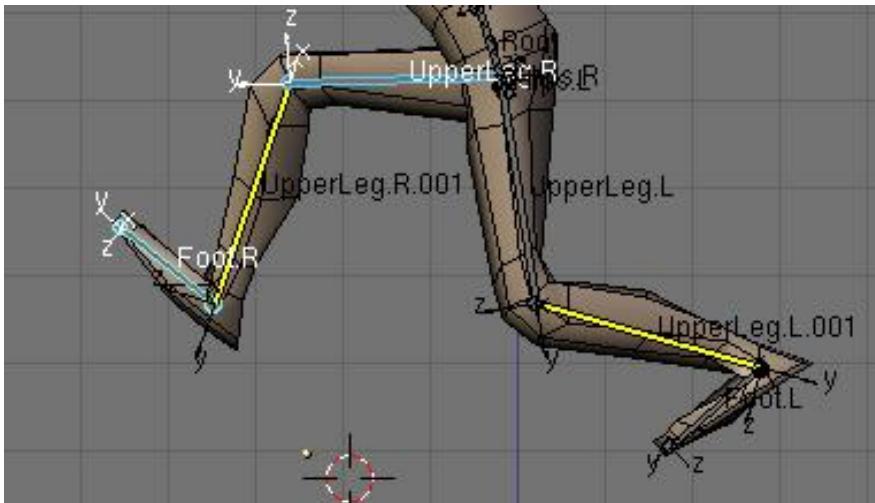


- Also If you set len to 0 and your chain's root is a child of another bone, The IK solver will reach and rotate all the bones until it gets to the end of the parent relationship. If all the bones are linked up to a master root, then all other sub-branches will be affected. If there is another IK target in other sub-branches 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.



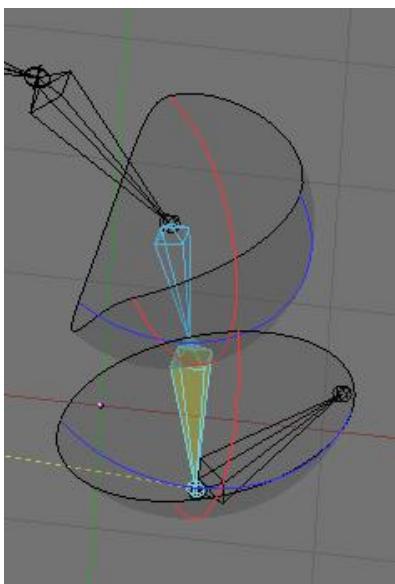
- PosW and RotW let you tell Blender if this IK solver will influence the final result more or less in the case of a Tree IK setup. With these options it's possible to use an IK solver just for location and an other one just for rotation.
- Tolerance and iterations are performance and precision options. 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 ends 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. This is very useful when doing a mechanical rig as you can limit the move or better, lock completely an axis.

Lock X Rot	Lock Y Rot	Lock Z Rot
Stiff X: 0.000		Stiff Z: 0.000
Limit X		Limit Z
Min X: - 94.7		Min Z: - 64.8
Max X: 118.4		Max Z: 88.6

- 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

## **Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/timeline**

---

Next page: IPO Window

Previous page: IK Solver

---

Next page: IPO Window

Previous page: IK Solver

---

## **Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/ipo/index**

---

Next page: Data Type

Previous page: Timeline Window

---

Next page: Data Type

Previous page: Timeline Window

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/ipo/type

---

Next page: Channel

Previous page: IPO Window

---

Next page: Channel

Previous page: IPO Window

---

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/NLA/index

## (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.

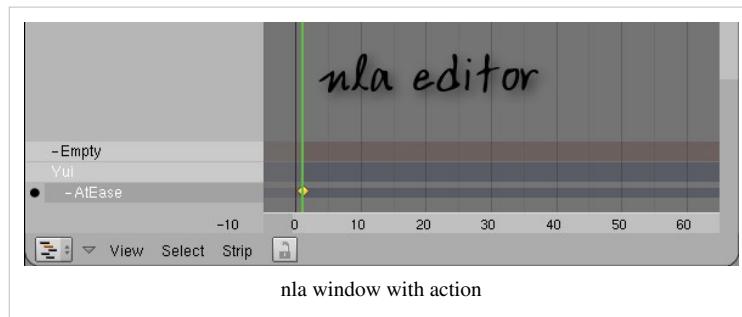
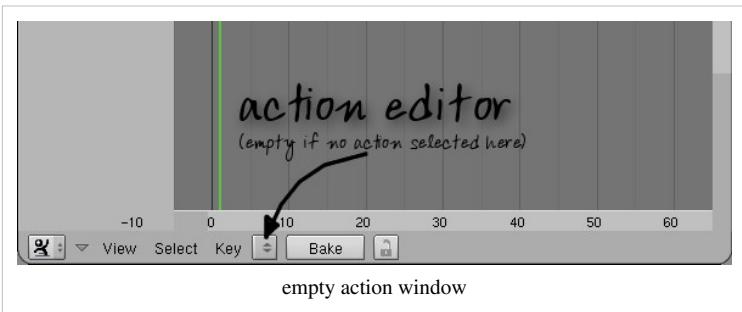
Close any open Actions by clicking the scary X in the Action Editor. If you don't do this, only this action will play. Now in the NLA editor, play the animation.

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)

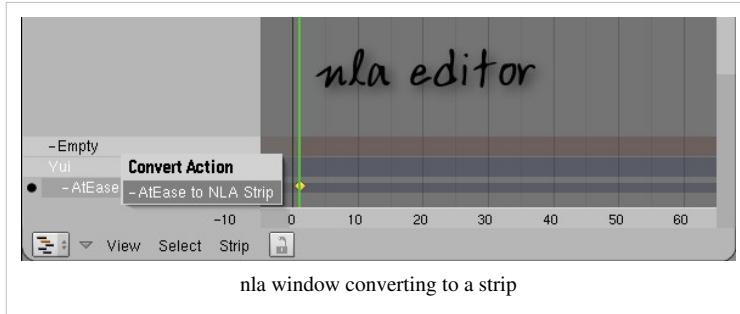
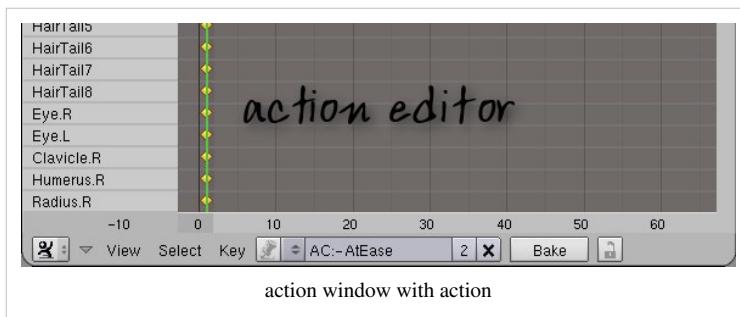
## Walkthrough

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.

Select an Action you've already made. Here, an Armature named **Yui** has bones involved in a one-frame action called **-AtEase**.

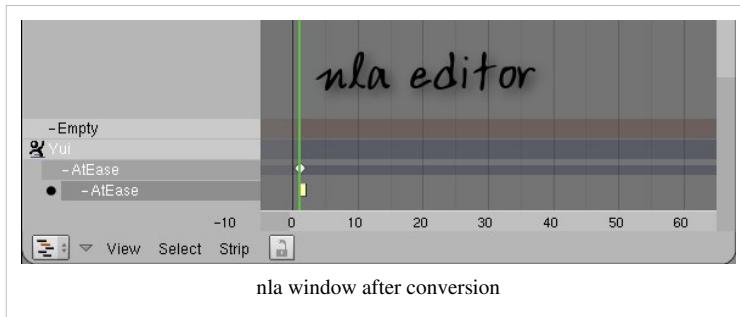


Convert the listed Action to an NLA Strip in the NLA Editor by pressing the CKEY with the mouse hovering over the Action to be converted. No change in the Action Editor; it is still available as an Action.



Once converted, note the changes in the NLA Editor. The Action icon appears next to the Armature's name: Yui . 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.



# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/NLA/intro

---

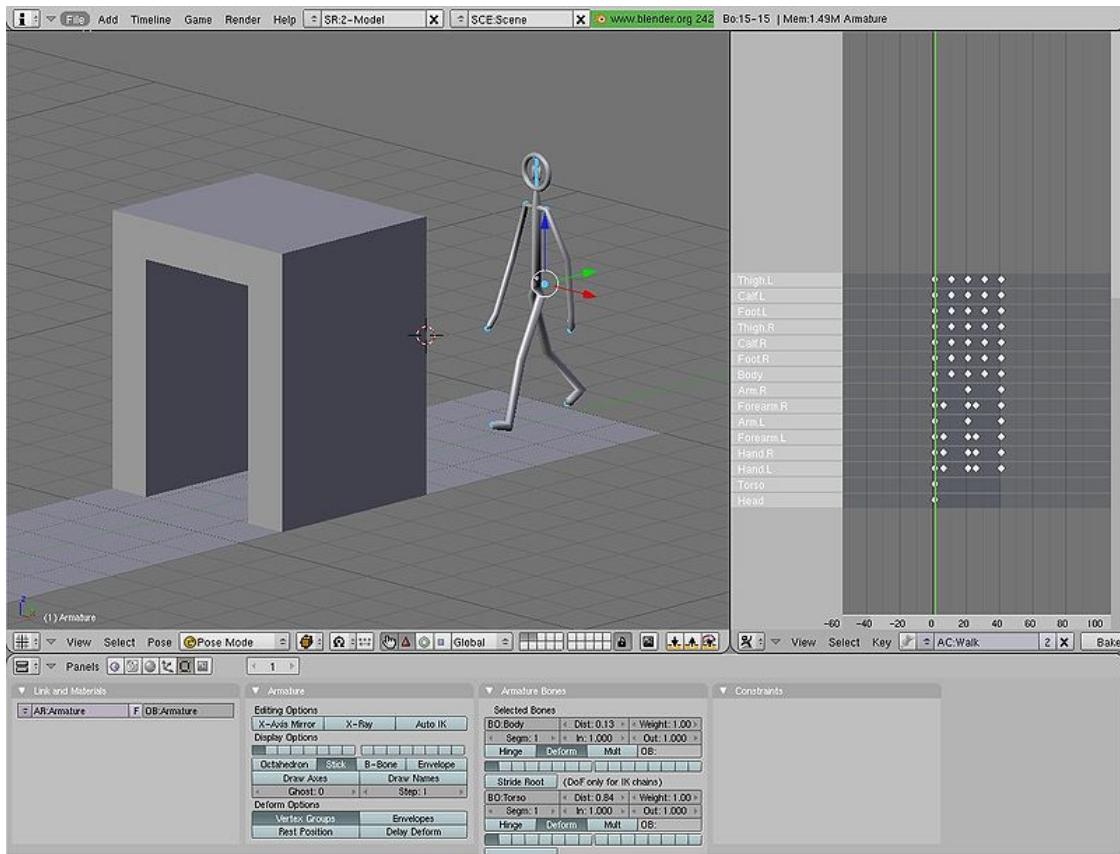
## NLA (Non-Linear Animation)

***WARNING**-This page assumes the reader to understand the IPO window and the Action Editor window, as well as rigging a character with an armature object. This tutorial will make little sense without this previous knowledge.*

Imagine this--yourself, sitting at your computer late at night, beating away at a huge blender animation. You didn't think it would take as long as it did, but you lost your composure staring into the jungle of colorful IPO lines, little white and yellow Action diamonds, and that annoying green current frame line. You know that the NLA window would help you make sense of it all, but you are afraid of opening that Pandora's Box because of the problems that will follow it. Never fear, for this tutorial will clean up all those problems and revolutionize the way you blend for the rest of your life.

## Setting up the Scene

It will be easier for you if you start with a small demo file than if you go straight for the big prize. Give a character or other armature-rigged object a few SEPARATE actions (remember to name them something distinct, like "Walk" or "Run" as you always should with *everything*). For your own sanity, you will want to have a path or an IPO that correlates with the actions in question. All of the blender screenshots I will be using come from a file with a very basic stick person, rigged with an armature skeleton. He has two actions; a normal walk cycle, and then another, hunched over one, as if to pass beneath a low ceiling. If you use this idea in your own test file (and I thoroughly recommend it) do yourself a favor and give them both the same stride length! Give this guy an IPO or a path to follow that keeps his feet from slipping, and make sure that it has a linear interpolation mode. The rest of the scene consists of a floor-plane and an arch too small for him to walk under (hence the crouching walk). Place this arch exactly one walkcycle away for now, we will move it around later.

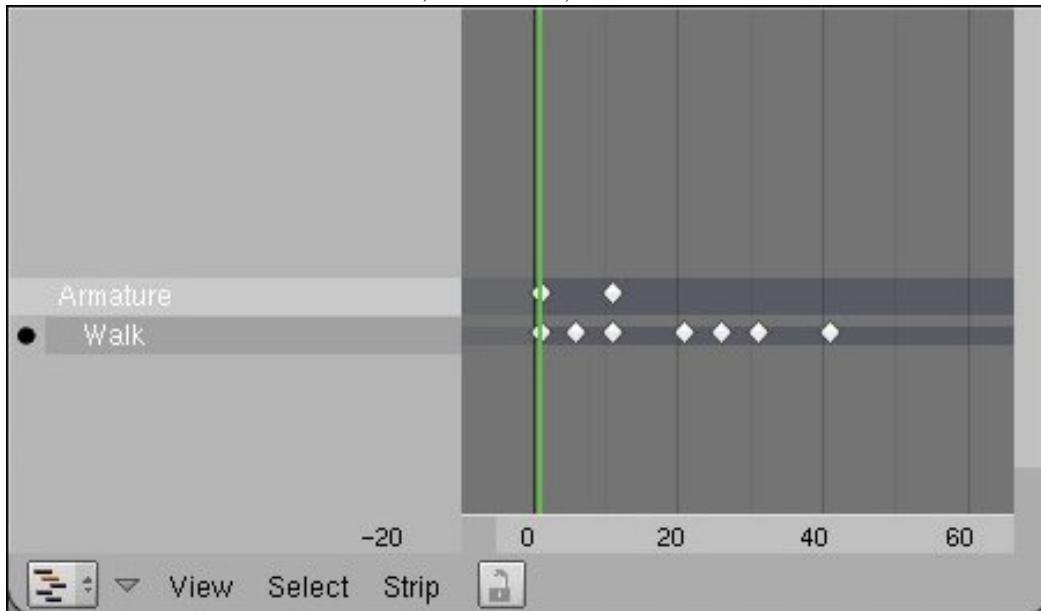


Complete the scene before going on to the next step.

## Adding Action Strips

Now, he moves forwards as you scroll through the frames, but how do we get him to walk forward, duck, and walk under the arch, too? First of all, you need to select the armature object and create a link to the normal walk. By pressing the up or right arrows, or pressing "alt+a", you should be able to see him walk up to the arch, stop walking, but keep sliding through, with his head sticking out the top. As ridiculous as this seems, right now, however, you are on the right track. Split your window now and open the NLA window with .

At first glance, this window looks almost identical to the action window we all know and love. This is not totally off of the mark. The NLA window is, in essence, an abbreviated version of the action and IPO windows.

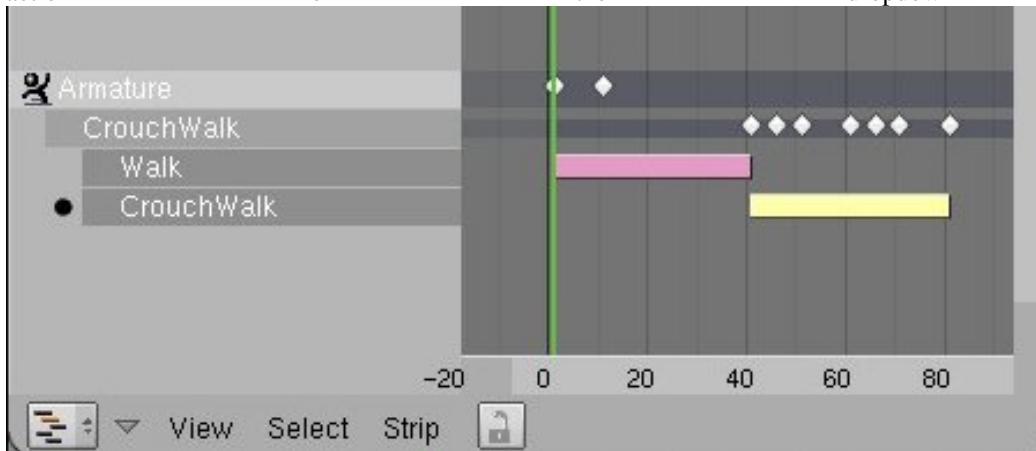


This is what you should see in the NLA window.

You see a space with the name of your armature (named "Armature" in my demo) and in that row you see a few key diamonds. These show the ipo you have put on the person. I made it so the IPO extended on forever, using the first ten frames as a guide. In a subset beneath the armature object, you see the name of an action. The bullet next to it means that the armature is currently linked to that object. In that row, you see key diamonds, and those correlate with the keys you put into the action in question. If you switch actions in the action window, you see that the NLA window also changes. But how do you get it to keep the first and play the second later?

Go to the first frame you want him to walk in (most likely number one) and place your cursor in the NLA window. Press "shift+a" to see a dropdown menu. Pick the name of the first walk cycle. You should see a new subset space appear, with the name of the first walk in it. This space should be occupied by an action strip. Two colors of action strips are yellow and pink, yellow meaning selected and pink meaning deselected, just like IPO vertices. This strip takes up the amount of frames the walk did. The NLA window has now saved your first action. Now to toggle to the next action, move the frame line to the end of the first strip, then press "shift+a" again, this time clicking the other action

on the dropdown menu.



The result.

## Getting the Strips to Play

Now, as cool as this seems so far, if you press "alt+a", you will be disappointed and confused. The animation will only play one of the actions. This is a simple problem to fix. If you look carefully, you will see that the armature is still linked to the action it played. This data is overriding any other data from the NLA window. Press the "X" in the action window right next to the name of the action. Press "alt+a" again. It works!

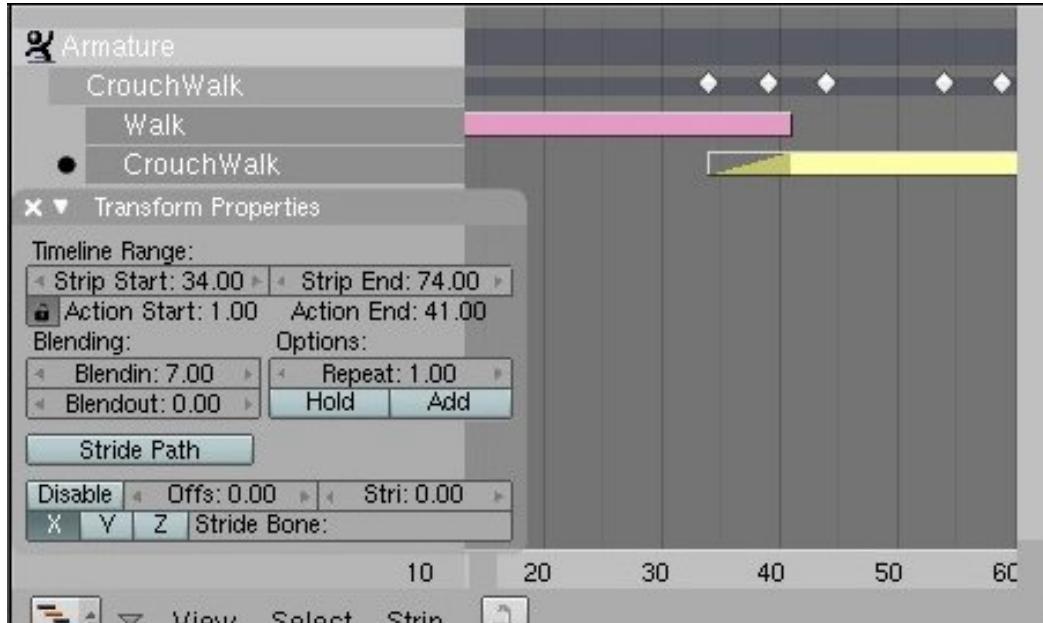
This is, however, a jerky and instantaneous switch from one action to the other. Making it less weird is easy. Press the "N" key with your cursor in the NLA window. You will see a small box with a bunch of functions in it (this will be explained in detail on later pages).



The Properties window.

This little box is the lifeblood of the NLA in blender. It contains all the tools you will need to be able to run this feature smoothly. I will illustrate a few of them right now, but most others will be shown in the section on this window.

Make sure you have highlighted the **second** action strip, and look at the space where it says "Blendin". The number there is the number of frames blender will use to smooth the transitions *in to* the highlighted action. Depending on how fast your character is walking, you may need to change the number a bit, but I set my "blendin" for seven. Whatever number you used, however, move the second strip *back* that same amount of frames. The wedge on the end of the action strip should end as the other strip ends.



The NLA solution.

## Conclusion

You now have the necessary skills to complete much more involved and complex animations. You can, however, increase this even more by continuing to read the other NLA tutorials in this wikibook.

Good Luck!

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/Guided tour/NLA/stride

At the moment, one of the best documentation for Blender Stride features can be found here: Blender Stride Tutorial [1], which takes on where the Official Blender Stride Page [2] takes off. Good luck! A simpler and shorter, but still very good tutorial can be found here: Blender wiki: NLA Editor and Stride Path [3].

## References

- [1] <http://www.telusplanet.net/public/kugyelka/blender/tutorials/stride/stride.html>
- [2] [http://www.blender3d.org/cms/Advanced\\_Stride\\_suppor.720.0.html](http://www.blender3d.org/cms/Advanced_Stride_suppor.720.0.html)
- [3] [http://wiki.blender.org/index.php/NLA\\_Editor\\_and\\_Stride\\_Path#Stride\\_Support](http://wiki.blender.org/index.php/NLA_Editor_and_Stride_Path#Stride_Support)

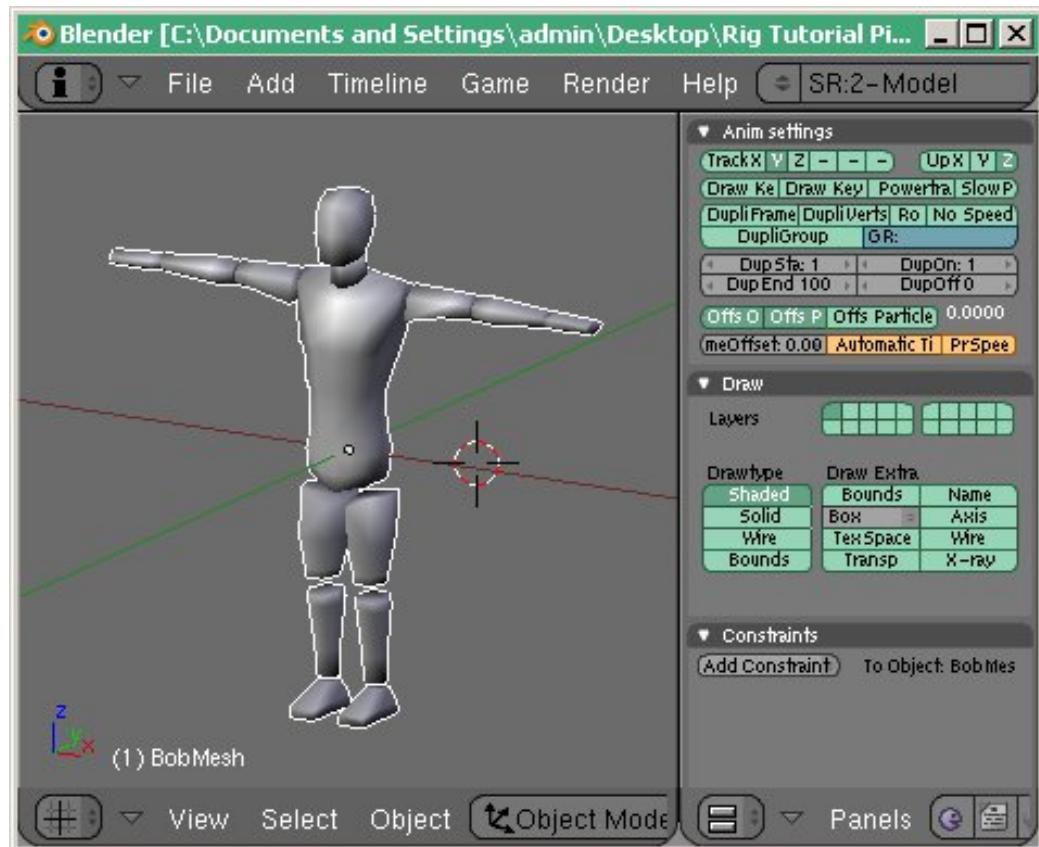
# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/example/bob/index

*Next Page: Build The Rig*

*Previous Page: Fly*

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



bob.blend had an incorrect link and has been taken off until further notice or someone else supplies a model.

- Build the Rig
- Deform the Mesh
- Create a Walk Cycle

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/example/bob/Build rig

*Next Page: Deform the Mesh*

*Previous Page: Working Example: Bob*

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**

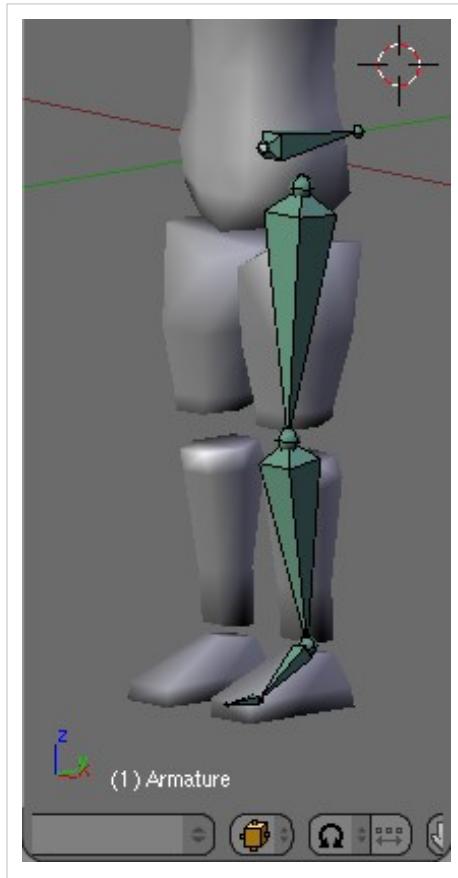
Snap your cursor to the root of this chain (shift+s) and add a bone. Now select the points at the hip joint and the ankle joint, and snap the cursor to the selection. **IMAGE**

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



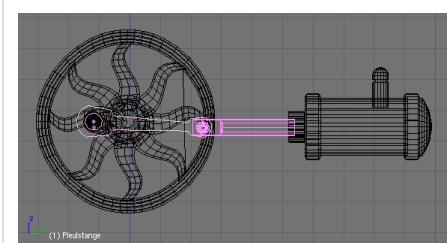
# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/example/ Piston, Rod and Crank

This page relates to Blender v.2.42a

## Piston, Rod and Flywheel

] A setup like that shown in **Image 1** is e.g. typical for a steam engine. It is surprisingly difficult to animate this setup, because one has to translate a rotating movement (of the Flywheel) to a horizontal movement (of the Piston). This alone wouldn't be too difficult, but the Rod has to stay attached to the Piston. The shown setup uses three empties and two armatures, each with two bones.

The horizontal direction is the Y-direction, if this is different in your model you have to exchange Y with your horizontal direction.



**Image 1:** Surprisingly difficult: the animation of Piston [Kolben], Rod [Pleuelstange] and Flywheel [Schwungrad]

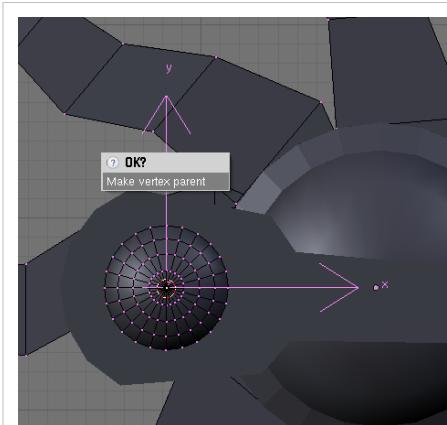
## Setup

- Use **Ctrl-A->Apply scale and rotation** on all mechanical parts in object mode.

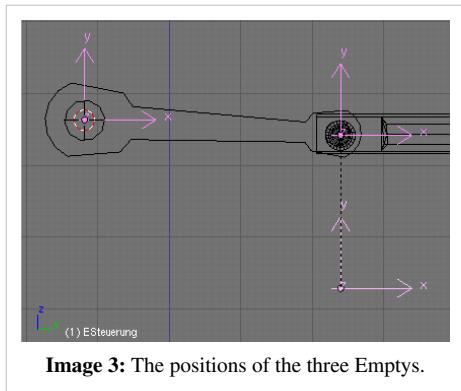
This will spare you from bad surprises if you rotate the flywheel afterwards.

We need an object which follows the rotation of the flywheel. We use an Empty, which is parented to a Vertex.

- Change to edit mode of the flywheel. Select a Vertex in the middle of the crank.
- Move the cursor to the Vertex (**Shift-S->Cursor->Selection**).
- Change to object mode and insert an Empty (**ERotation**).
- Change back to edit mode of the flywheel. The vertex is probably already selected, now select with **Ctrl-RMB** the Empty and press **Ctrl-P->Make vertex parent** (**Image 2**).



**Image 2:** Make vertex parent of an Empty.



**Image 3:** The positions of the three Emptys.

We need two more Emptys:

- Place the second Empty (*EGelenk*) in the center of the joint between Rod and Piston.
- Place the third Empty (*ESteuerung*) near to the joint.
- Now select at first *EGelenk*, than **Shift-RMB** *ESteuerung* and parent *EGelenk* to *ESteuerung* with **Ctrl-P**.

We're going to insert two armatures, each with two bones. The first armature, helps translate the rotation to a horizontal movement. The second armature drives the Piston and the Rod. The cursor should be positioned at the Empty *ERotation* (like in **Image 3**), else set it there.

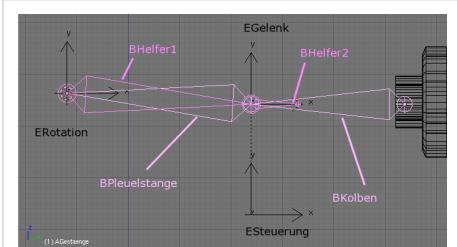
- Add an armature (*AHelper*). The root of the first bone (*BHelper1*) has the same position as *ERotation*, the tip has to be positioned to the Empty *EGelenk* (select Empty first, *Cursor->Selection*, than select bone tip, *Selection->Cursor*).
- Extrude another small bone (*BHelper2*) in horizontal direction.

*BHelper2* is automatically *Child* of *BHelper1*. The second armature that is driving the movement of piston and rod will be inserted in the opposite direction, i.e. from right to left in our example.

- Position the cursor on the piston.
- Add an armature (*AGestaenge*).
- Move the tip of the first bone (*BKolben*) to *EGelenk*.
- Extrude the first bone and place the tip of the second bone (*BPleuelstange*) to *ERotation*.

Connecting the two bones will ensure that the rod cannot disconnect from the piston.

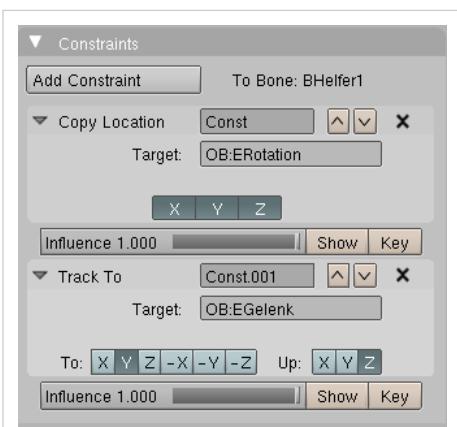
Now we're going to add the "logic".



**Image 4:** The alignment of the two armatures.

- Add a *Copy Location* constraint to *BHelper1* which points to *ERotation*. To do this change to the pose mode of the armature *AHelper* and select *BHelper1*. In the *Constraints* panel click *Add Constraint->Copy Location* and type *ERotation* into the *Ob:* field.
- *BHelper1* needs a second constraint: *Track To* to *EGelenk* (**Image 5**).

Now the root of the bone *BHelper1* follows the rotation of the Flywheel. Its tip follows the joint, but with slight deviations upwards and downwards. The position of *BHelper2* governs the horizontal position of the Piston. We just have to "extract" the horizontal position, for that we're going to use a *CopyLocation* constraint, but only in Y-direction. There is no offset for a copy location constraint - else we could have done with one empty less.



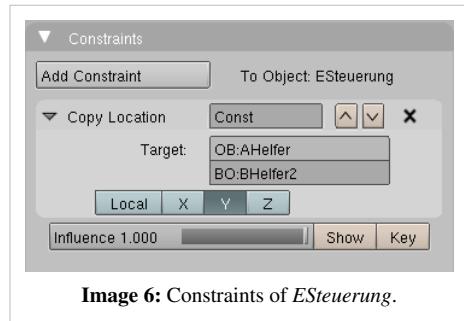
**Image 5:** The constraint settings for *BHelper1*.

- *ESteuerung* gets a *CopyLocation* constraint in Y-direction to *AHelper->BHelper2*. You have to inactivate X and Z (**Image 6**).
- Make *ESteuerung* the parent of *AGestaenge*.

The armature *AGestaenge* gets its horizontal movement from *BHelper2* via *ESteuerung*.

Nearly finished:

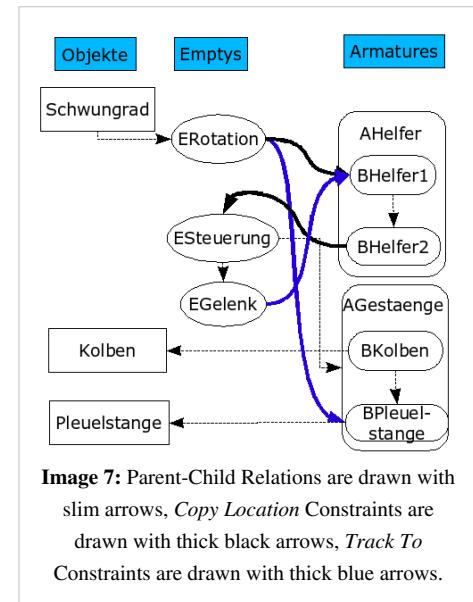
- Give *BPleuelstange* a *Track To* constraint to *ERotation*.
- Parent the piston and the rod (the meshobjects) to their bones.



## Overview about the relations between armature bones and objects

- *Schwungrad* is vertex parent of *ERotation*.
- *BHelper1* has a *Copy Location X/Y/Z* constraint to *ERotation*.
- *BHelper1* has a *Track To Constraint* to *EGelenk*.
- *ESteuerung* has a *Copy Location Y* constraint to *BHelper2*.
- *ESteuerung* is parent of *AGestaenge*.
- *ESteuerung* is parent of *EGelenk*.
- *BPleuelstange* has a *Track To* constraint to *ERotation*.
- *BKolben* is parent of *Kolben*.
- *BPleuelstange* is parent of *Pleuelstange*.

The Parent-Child relations between the bones are inserted automatically through extruding, you have to insert the other relations yourself.



## Finishing Touches

- Animate the rotation of the flywheel.

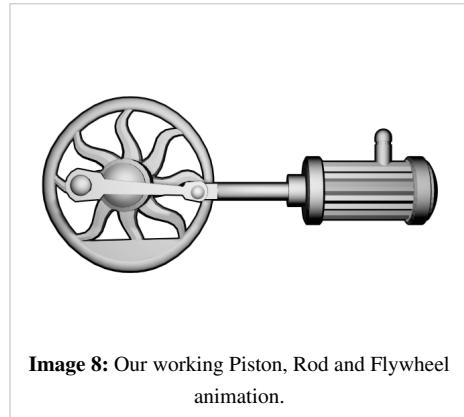
The driving object is the flywheel, all the other movements are synchronised with its rotation.

If you've done everything right, your animation should look like in **Image 8**.

## Links

The german original of this tutorial (same author)

Another method using constraint system <sup>[1]</sup>



## References

[1] <http://blenderartists.org/forum/showthread.php?t=140734>

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Animation/example/ Cutting Through Steel

*This tutorial has been created with Blender v2.48a*

In order to simulate a laser beam which is cutting through a steel plate, we have in Blender (at least) two possibilities:

1. an animated Mesh with many Shapekeys
2. an animated texture

The latter method is considerably easier to implement and the result is nevertheless convincing enough. We will proceed as follows:

- You need an animated alpha mask for the points at which the metal is changed. For this use a text object as a path. A cube follows the path and emits a particle system. The results are rendered as animation and the images stored.



**Figure 1:** Cutting out a logo, the result of this tutorial

- Then load these pictures and use them as an animated texture on a plane.
- With a second particle system the sparks are simulated.
- With a third particle system we simulate the annealing of the weld. Again, this sequence is used as an animated texture.
- Smoke is created with a fourth particle system.

The result is shown in **Fig. 1**.

This guide is based on the well-known tutorial Cutting Through Steel <sup>[1]</sup>, but remade with the current Blender particle system and with constraints. You need to have basic knowledge of Blender.

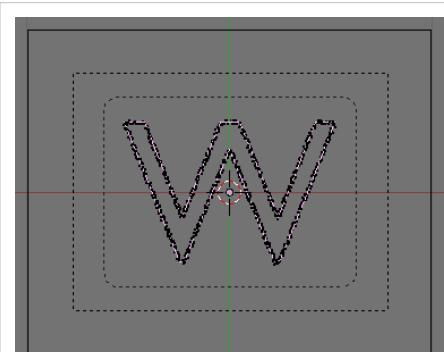
## Creating the Alpha-Map

- Start Blender. Save your file. Delete the cube.

We should save the file right at the beginning, because we will save the animated textures in subdirectories to the current directory. Particle caches are also created as subdirectories.

- Set the *End* value to 150 in the *Anim* panel of the *Render* buttons. This allows you to specify the total length of the animation.
- In the *Output* panel use "*//AlphaMap/*" as output directory. When we render the animation the images are than saved in the subdirectory "AlphaMap" to the current directory.

The animated alpha map consists of gray-scale images whose luminance information is used to calculate the mask. Bright pixels produce transparency, dark pixels remain opaque.



**Figure 2a:** Setup of curve, emitter and camera in the 3D window.

- Use *PNG* as output format. This avoids compression artifacts which occur with the *JPEG* format.

I've left the resolution at 800x600, but you need to set your desired resolution here.

- Create the shape you want to cut out as *Curve* object, or use a *text* object and convert this to a curve. I've done the latter, I've used a *text-object* (a "W"), and converted this to a *Curve* object (*Object-> Convert Object Type ...-> Curve*).
- With the function *Center* in the *Curve and Surface* panel I've brought the object in the middle of the 3D-coordinate space.
- The filling of the object is not needed here, therefore activate the option *3D* for the curve object, even though the object should remain two-dimensional.

Now we need again a cube. The cube will follow the curve. Its particle system will create a glowing trail, that is later used as alpha map.

- Insert a cube.
- Scale the cube as small as the track of the laser beam (the welder) shall become.
- Select the cube, than also select the curve object with **Shift-RMB**.
- **Ctrl-P->Path Constraint** creates a *Follow Path* constraint.
- Select only the cube and activate the option *CurveFollow* in the *Constraints* panel in the *Object* buttons.

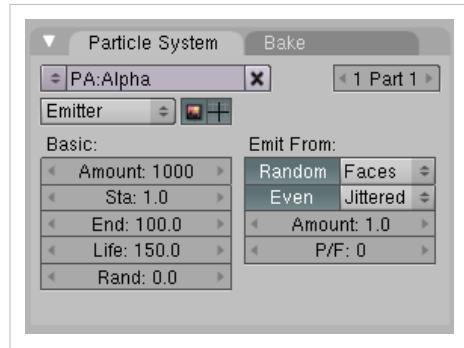
If you press **Alt-A** now and let run the animation, the cube will follow the shape of the curve object, but not necessarily in the desired direction and not from the desired location. So some work needs to be done:

- Change to the *Edit* mode of the curve object.
- *Curve->Toggle Cyclic* opens the curve object, by adding (*Subdivide*) of one or more control points you achieve the desired movement. **W->Switch Direction** reverses the direction of the motion of the cube.

By adding a *Time* Ipocurve to the curve you could set the duration and speed of the animation, but I've left the preset duration of 100 Frames. The last 50 Frames the path will glow a bit and the smoke will vanish.

- Add a particle system to the cube, call this *Alpha*.
- Set the particle lifetime to 150 (or higher) (**Fig. 2b**). The particles will live the entire length of the animation.
- *Bake* the particle animation.
- The cube gets a material:
  - *Halo*
  - RGB 1/1/1, that is completely white
  - *XAlpha*
  - *HaloSize*: 0.03
  - *Hardness*: 40

I want to achieve with these settings, that we will get a grayscale image, that is completely white where the material shall become transparent. We don't need partial transparency, since a relatively sharp cutting edge shall be created. Slight irregularities in the periphery are wanted.



**Figure 2b:** particle system of the cube.

To render the animation, the camera is centered directly over the object.

- *View->Top*
- **Strg-Alt-Num0** sets the camera to the current view. Select the camera.
- With the *Transform Properties* panel in the 3D-window (*Object->Transform Properties*) set the coordinates of the camera exactly vertically above the center of the 3D window (X/Y=0).
- In the *Editing* buttons of the camera set the option *Orthographic* in the *Camera* panel. Scale the size of the animation with the *Scale* value.



**Figure 2c:** A frame of the AlphaMap.

- Set the world color to black in the *World* buttons.
- Render the animation.

Blender now generates an image sequence from 1 to 150 in the subfolder *AlphaMap* (Fig. 2c).

## Welding sparks

Now we'll create a new scene, in which the different effects are joined.

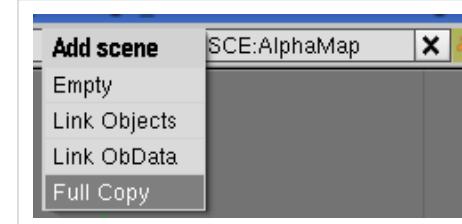
- Rename your current scene to *AlphaMap*.
- Create a new scene as *FullCopy* (Img. 3a). Name this new scene *Cutter*.

The welding sparks are created using a particle system.

- Add a *UVSphere* with 16 segments and 16 rings.
- Delete the lower half of the sphere in Edit Mode and scale it to approximately the same size as the cube.
- Switch to Object Mode.
- Select the cube in addition to the sphere. Copy the *Follow-Path*-Constraint from the cube to the sphere.

Note that you copy attributes from the active to the selected objects. The active object is the one you've selected last

- Create a particle system for the sphere.
  - Name *Sparks*
  - Amount: 4200
  - Life: 5



**Figure 3a:** Adding a new Scene



**Figure 3b:** The particle system for the sparks.

- *Emit From: Random/Verts*
- *Normal: 0.757*
- *Random: 0.397*
- *Rotation Random: 0.116*
- *AccZ: -1.37*

Except for the lifetime the values are relatively arbitrary, I have just experimented with the values until I was halfway satisfied with the results. The sparks have a slight acceleration in negative Z direction to sink down.

- *Visualization:*
  - *Line* (which corresponds as much as possible to the setting *Vect* in the particle system before 2.46)
  - *Speed*
  - *Back: 0.04*
  - *Front: 0.0*

You achieve a long spark with these settings.

- *Bake* the particles. Change to the *Bake* panel and click on the *Bake* button. The particle animation is now permanently saved.

The *Bake* end value is also the final value for the normal particle animation, even if you don't bake. If you need more than 250 frames, you need to increase this value.

## Using the alpha map on an object

The alpha map is now used as texture on a plane. Therefore we have to position the texture exactly like the previously rendered animation, so that the sparks run in sync with the blending out of the plane.

- Add a new plane to your scene (*Add->Mesh->Plane*).
- Scale the plane at about twice it's size - as big as you need it for the final shot.

The main problem is now to fit the texture exactly to the plane. We will give the plane UV coordinates, than fit the plane to the ratio of the texture and finally adjust the position in the 3D window with UV coordinates.

- Switch to *Draw Type Textured* (**Fig. 4a**).
- Split the 3D window, and open in one part the *UV/Image Editor*.
- Switch to edit mode of the plane.
- Unwrap the plane (*(Mesh->UV Unwrap)*).
- Load an image of the alpha map in the image editor (not the first one, so you can see the course of the map).

Normally the texture is not orientated the way I would like it to be.

- Click the padlock icon in the toolbar of the *UV/Image Editor* (*Update other windows in real time*).



**Figure 4a:** Setting the *Draw Type*

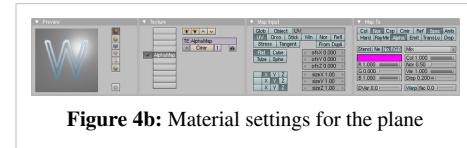
- Rotate the UV vertices in the *UV/Image Editor* until the texture fits (eg, while holding the **Ctrl** key)

- Switch to object mode.
- Add a new material to the plane and name it *Rust*.
- Select the fifth texture slot and add a new texture. Name it *AlphaMap* with the parameters:

- *Map Input: UV*

- *Map To:*

- *Alpha*
- *Spec*
- *Nor*
- *NoRGB*
- *DVar: 0*



**Figure 4b:** Material settings for the plane

With these (**Fig. 4b**) settings we achieve, that the white pixels in the texture set the Alpha and the Spec value to zero (*DVar*). Because the texture delivers RGB colors we have to activate *NoRGB*, because we need intensity values. We don't need *RayTransp*, because the world background is black anyway. We can't use *ZTransp*, because Halos are not rendered in front of materials with *ZTransp* (probably a bug in Version 2.48a). With *Nor* we get a 3D effect.

- Switch to the *Texture* buttons.
- Type: Image Texture
- Load the first image of the alpha map sequence.
  - *Image: Sequence*



**Figure 4c:** Texture settings for the animated alpha map

- *Frames:* 150
- *Map Image: Extend* (**Fig. 4c**)

Now we adjust the UV coordinates.

- *Object->Transform->Scale to Image Aspect Ratio*. This scales the ratio of the plane automatically to the aspect ratio of the image texture.
- Change to a frame where you can judge the particle animation and it's fitting to the texture.
- Switch to edit mode. Scale the UV coordinates in the UV editor until the texture fits to the particles.
- Switch back to object mode.

## Material for the plane

We will now give the plane a better material. To judge the material we will first re-focus the camera, give it a texture and make some material settings. If you already have a good metal material, you can use this of course directly and use the Alpha texture from above on your existing material.

- Select the camera.
- Turn *Orthographic* in the *Camera* buttons off.
- Position the camera as desired.
- Select the plane.
  - I've used the rust texture that is linked at the bottom of the page as image texture.
    - *Map Input: Orco, SizeX/Y each 0.5*. The size change is to use a larger part of the texture on the final image. But this depends on the size of the camera and the texture.
    - *Map To: Col/Nor*
  - In the second texture slot I've mapped a *Stucci* texture to *Spec*.
    - This abridged sentence means: select the texture slot. Set *Map To* to *Spec*. Deselect *Col*. Change to the *Texture* buttons. Select *Texture type Stucci*.
    - The *Specularity* is thus determined by a texture. The spec value is increased by the texture (*DVar=1*). Therefore you have to set the *Specularity* in the *Shaders* panel to the smallest value the material shall have (e.g. 0.08).



**Figure 5:** Material for the plane

If you have done a test render, you'll see that the particle track of the cube is still there.

- We don't need the cube anymore, you may delete it. In the *Scene Cutter* we use the already generated AlphaMap. Texture with OSA (*Antialiasing*) are somewhat blurred, for our purpose we use *CatRom* instead of the default *Gauss* method.
  - Set in the *Render* panel *CatRom* as *AntiAliasing* filter.
- We need a new output directory for the animation, to not overwrite the AlphaMap we've created earlier.
- Set the output directory to "*//Render/*" in the *Output* panel.

## Lighting the torch

The torch itself shall cast light, therefore we parent a Lamp to the *Sphere* and animate the energy of the lamp.

- Change to frame 1.
- Select the *Sphere*.
- Place the 3D cursor on the *Sphere*. *Object->Snap->Cursor to Selection*
- *Add->Lamp->Lamp*
- Select the lamp in addition to the *Sphere*.
- *Object->Parent->Make Parent->OK*
- Select the lamp and move it a bit up (ie in positive z-direction). The lighting shall emit from the sparks, not from the hole.

I find it cumbersome to animate the energy of the lamp by setting ipo-keys in the lamp buttons. Instead I do it directly in the 3D window.

- Change to the *Ipo Curve Editor* in one of the open windows.
- Change the ipo type to *Lamp*.
- Activate the *Energy* channel with **LMB** click in the ipo window.
- By holding **Ctrl** key pressed, you create by left-clicking in the ipo window the points of the ipo curve.

These values may be larger than 1. However, the last maximum value should be reached in frame 100, and decrease significantly to frame 105. The sparks are only issued up to frame 100, and live up to the 105th frame. After that the metal is glowing a bit.

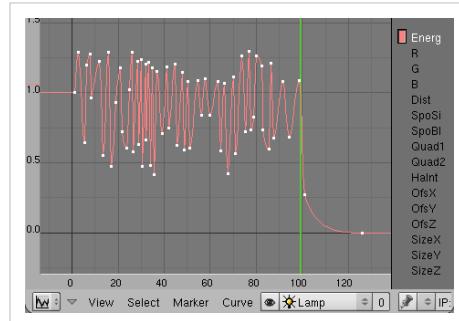
Your result may look somewhat like mine in **Fig. 6**. If the amplitude of the curve is not large enough, you can scale the ipo curve at the y-axis.

## More realistic sparks

You may animate the material for *Line* visualization during the lifetime of each individual particle if you use *Halo* for the material, so you could change the sparks color from white to red.

We will show this in the next step, but confine ourself here to make a better, not animated material (**Fig. 7**).

- Add a material to the *Sphere*, name it *Sparks*.
- Activate *Halo* in the *Links and Pipelines* panel.
- Set the RGB color for *Halo* to white.
- Activate *Lines/X Alpha/Shaded* in the *Shaders* panel.
- *Halo Size:* 0.1
- *Add:* 0.8



**Figure 6:** Ipo curve for the energy of the lamp



**Figure 7:** Better sparks

## Smoke

Now we're going to create an individual material animation for every particle during its own lifetime.

We will use particles with a simple texture and animate the alpha value of the material. We're going to duplicate the *Sphere* and give it a new particle system.

- Duplicate the *Sphere* with *Object->Duplicate*. The new sphere already bears the *Path* constraint.
- Remove the *Sparks* system from the new *Sphere*.
- Create a new particle system. Name it *Smoke*.
- *Amount:* 300
- *Emit from:* *Random/Verts*
- *Initial velocity:*
  - *Object:* 0.02
  - *Normal:* 0.1
  - *Random:* 0.01
- *AccZ:* 0.05
- Bake this particle system.
- Change to the *Material* buttons.
- Create a new material and call it *Smoke*.
- *Alpha:* 0.034. This is the basic visibility for the smoke texture.
- Add an ipo key for *Alpha* in frame 1.
- Change to frame 100.
- Set *Alpha* to 0.0 and add the second ipo key.



**Figure 8a:** Sparks and Smoke

Now the smoke will disappear within the lifetime of each particle, independently of the lifetime of the particle.

- *Halo*
  - *Halo Size:* 0.224
  - *Hard:* 12
  - *Add:* 0.014
  - *Halo Tex* This is the crucial setting to use a texture for the shape of the halo.
  - *X Alpha*
- Create a new texture, you may also call it *Smoke*.
- *Texture Type:* *Image*
- Load an image with random, smoke-like structures, like that from **Fig. 8b.**



**Figure 8b:** Particle texture

If the fading goes to quickly, or the texture is to transparent / opaque, change the animation of the *Alpha* value.

You see, even though we use only a few objects and materials, it is extremely helpful if we designate meaningful names to the objects, materials, textures etc.

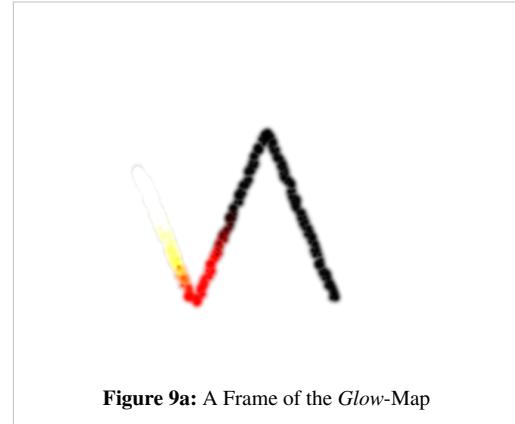
## After Glow

Now at last we will create another animated texture, that shall simulate the annealing of the steel plate after it is being cut through.

- Change to the scene *AlphaMap*.
- Add a new scene again as a *Full Copy*.
- Name this scene *Glow*.
- Enter the output directory "*//Glow/*" in the *Output*-Panel of the *Render* buttons.

The texture shall have nice semi transparencies.

- Therefore we select a different ouput format in the *Format* panel, namely *RGBA* (RGB with Alpha).
- Selecting *Premul* in the *Render* panel the world background will become fully transparent, and the particle trace semi transparent.



**Figure 9a:** A Frame of the *Glow-Map*

We use the existing particle system, but change it's material settings.

- Call the material of the cube *Glow*.
- Increase the *HaloSize* to 0.04. Our new texture has to be larger than the cutout of the plane, else it will not be visible.

Now we're going to animate the material of the halos. Again this works only because we have *Point* visualization of the particles.

- Change to frame 1.
- Set an IPO key for the RGB color (here white).
- Go to 3rd Frame.
- Change the Halo color to 1/1/0.42 (light yellow). Set the next IPO key.
- Go to 6th Frame. Change the Halo color to 1/0/0 (red). Set the next IPO key.
- Go to 11th Frame. Change the Halo color to 0/0/0 (black). Set the next IPO key.

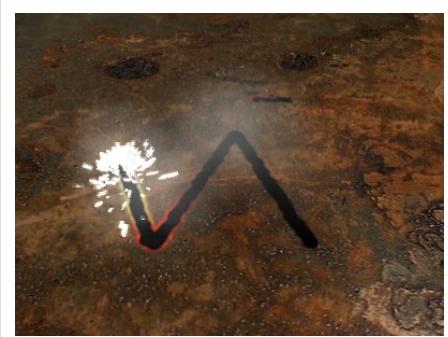
The particle trace will fade out during 11 frames.

- Render the animation.
- Go back to the scene *Cutter*.
- Select the plane.
- Load the newly created texture in the third texture slot as an animated image texture.
  - *Add New*. Name the texture *Glow*.
  - *Map Input: UV*
  - *Map To: Col/Emit*

We need the alpha values in the image for changing the emit value.

Also the alpha blends the glow texture nicely with the plane texture.

- *Texture Type: Image*
  - Select the first image from the subdirectory *Glow*.
  - Activate *Sequence*.
  - *Frames:* 150
  - *Extend*



**Figure 9b:** A finished frame of the animation.

The settings for the texture are identical with those of the texture *AlphaMap* (*Fig. 4c*), except of course for the directory in which the images reside.

In **Fig. 9b**, see a finished frame of the animation. For the video in **Fig. 1** I've changed the animation a bit, scaled the emitter objects a bit smaller, changed the sparks Color and used a different camera perspective.

Suggestions, critics and approval please write on the commentary page of this article. If you find some obvious mistakes you may correct the article directly of course.

## Links

- The Blend File <sup>[2]</sup>



Rust Texture

## References

- [1] <http://www.blendernation.com/tutorials/blender-3d-intermediate-tutorial-cutting-through-steel/>
- [2] <http://www.snapdrive.net/files/180816/CuttingThroughSteel.zip>

---

# Unit Six: Advanced Game Engine

---

## Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Game Engine/index

---

This set of tutorials is just for you Blender users who stumble up on the basics of certain games, such as making a HUD, menu, or even a simple button. This is also a place where users who know cool ways to spice a game up or explain troublesome aspects of making certain parts of a game genre's content work (like how to get Blender to calculate the damage of a bullet shooting).

There are a couple subsections to this, though. One is for those who don't like the messiness of python. Learning python to the point of being able to create a game with it will take months, maybe years, just to learn it--not to mention how long it would take to type the specific code and troubleshoot. So the first section is for those who are much more comfortable with using the GUI Blender screen to make the majority of there game, and the tutorials exist to help them enhance there video game.

**Note:** No matter how much GUI you'd like to use in game creation, there is a certain amount of python you will HAVE to use to be able to really make your game good. Make sure to at least read up on it. Sorry.

Jump to *Game Creating Techniques within a GUI*

Then there is the second subsection. This section is to help those who are comfortable with python learn how to use the code in ways they never thought of, and to show those who don't do so well with it just what python can do. While it's a simple language, it can still be incredibly complicated at times, and very, very picky.

**Note:** DO NOT SUBMIT ANY CODE YOU DO NOT WANT COPIED A MILLION TIMES! This section is for helping the broad and the narrow range of Blender users to learn how to use python better, and there will be no expectation of attribution for submitting code you found out yourself.

Jump to Game Creating Techniques with Python

---

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Game Engine/Game Creating Techniques(GUI)

---

## Making Blender Games Graphically

If you're reading this section, I'm guessing that you have either made a game before or at least are in the process of making one right now. Let's face it--it's probably much harder than you actually imagined it, we all take video games and the work put into them for granted. And there are probably quite a few times in which you wanted to do something so simple--such as a menu or even a button--but didn't even know where to start or what to do.

This is why this section exists.

The next few pages will all be tutorials with different techniques on how to make objects in Blender move and respond to things like your mouse, keyboard (a.k.a. "HotKeys") and making it all look like what a real game does. They will be simple versions of a menu, or a button, in which you can use this and put it into your own idea with your own graphics and your own little tweaks.

There will be a certain amount of Python necessary in the next few tutorials, mainly just to allow the mouse icon to appear on screen. However, you can do the rest of the tutorials in the 3D View Window, and under the Logic Index in the Tab window.

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Game Engine/Game Creating Techniques(GUI)/Creating Pop-Up Menus

---

In this tutorial, we will be making a Title Screen and a Main Menu Screen to come from that Title Screen. While most people hardly pay attention to these areas on a real video game, they do make the game seem much better to play, and gives the player a certain choice of what they can do before they play the game. Before beginning this, make sure you familiarize yourself with several logic sensors and actuators, such as "Scene" and "Property", as they will heavily be relied upon. It is also best if you have several Pictures ready to use for your game or are ready for additional functions with certain programs (a Word Processor Program and Image Editing Program [that has a transparency setting] can help move along your game.)

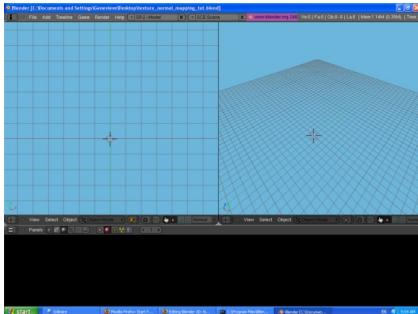
## Making An Easy Title Menu

The Title Screen will be very simple -- it will be the "press Enter to continue" kind. This will give way to a more interactive kind of Main Menu, which you can either use the arrow keys, "HotKeys", or the mouse pointer. This tutorial will not be the most visually pleasing, but then again, you can always add whatever graphics you would like to. Again, it would be wise to also have handy a Word-Processing Program, a Paint Program, and hopefully your keyboard has a "Print Screen" button somewhere (usually next to the F12)

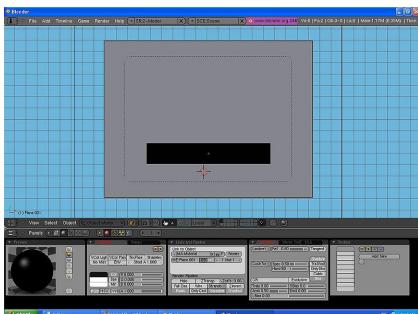
*If you have never used different scenes before, you're about to get a crash course. Also, Blender hotkeys and actions will not be explained with (A-key) or (numpad-6), it is assumed you have already learned at least how to perform basic Blender services.*

## Making the Menu

First, start up Blender and delete everything in the scene. Then, move to top orthographic view (press numpad 7, or if you have not a numpad, go to the hidden top menu and in the System & OpenGL section select "emulate numpad").



The first thing you should do is add a new scene. In the Information view panel at the top, click the SCE: drop-down menu and click on ADD NEW. Label this new scene something like "Main Menu". Now go back to your original scene and name it something like "Title Menu". These names will be important when it comes to changing the scenes around. So, on your "Title Menu" Scene, begin by adding a Camera. Move this Camera in the Z direction about a point, just to get it above the grid. Now, set the camera view to Orthographic. Now switch to Camera View. Add a plane that is scaled to cover the entire surface of the Camera's view-bounds. Move that plane just a little bit in the -Z direction. Then, add another plane, but scale this plane to be a thin rectangle about 2/3rds down the Y-length of the Camera, centered in X-direction.



(Image Shown with Materials for visual purposes, DON'T ADD MATERIALS)

*Now this next step isn't necessary, but if you want a good-looking game you will probably need this step.*

Let's add some UV textures. Open a Word Processing program and type in something that would be appropriate for your game, like "Press Any Key to Continue". Now, highlight the text and copy it, or you can hit "Print Screen" button, then open a good photo-editing program like Photoshop, GIMP, or Microsoft Photo Editor; something that preferably supports Transparency. Either way, open some image-editing program and paste the text. What I do is add lots of extra space and add additional text that I will use in the main menu too, such as "New Game", "Continue", "Story Mode", ect. so it is all in one Image for multiple UV textures. If your editor supports it, set the background color to transparent. If not, set it to black so it will register as 0 on the alpha channel in Blender. Save the image (as a ".gif" if you want to save transparency). Now, in Blender, split the 3D screen into 2 screens, and change one screen over to UV face select. Select your small rectangle, and switch to Edit Mode and Unwrap the face in the 3D window, then open your saved image in the UV face window. Scale your face in the UV window so it holds only the "Press Any Key to Continue" graphic. If you wish, also add a starting Image on the larger rectangle.

## Setting Up The Actions

Now this step is necessary. Select the large background rectangle, in the Buttons Window at the bottom switch it to the Logic Tab. This will be very simple. Add one Sensor, one Actuator, and one Controller. Make the sensor a Keyboard Sensor, depress the "True Pulse" button, and in the key area depress the "All Keys" button. Make the Controller an "AND" controller. Make the Actuator a Scene Actuator. Select the Drop-Down box and make it "Set Scene" with "SCE:Main Menu". **Before you test the game, make sure to save and add a Camera to the "Main Menu" Scene. Otherwise the game WILL crash.**

If you want the option of returning to this scene, instead of one Actuator make two. Make both of them Scene, but make one "Suspend Scene" with "SCE:Title Menu" and make the other Actuator "Add Overlay Scene" with "SCE:Main Menu". Also, that means in the Main Menu Scene later you'll have to add some key to hit or some action to "Remove Scene" "SCE:Main Menu" and "Resume Scene" "SCE:Title Menu".

## Author's Starting Menu Example

*The following contains actual in-progress work by the original Author of this page. The work and ideas are heavily copyrighted and it is highly encouraged (begged) that you do not rip this content off--the author has submitted this work generously to help less advanced users more efficiently.*

The following is my work on a game I am making, and I hope it helps you too.



This will get you to the next menu, which will be on the next section.

## A Main Menu

This menu will be more difficult, using a mouse-python function to use and make the mouse appear when you play your game and a maze of sensors and actuators entangled.

Here is the first thing you need: a python script to make the Mouse show ---

*\*This code is attributed to the author of the chapter "Blender Game Engine/A Simple Mouse Pointer" and this is not my own code.*

```
import Rasterizer as r
r.showMouse(1)
```

Put this in the text editor, and save the text as something simple like "Mouse". Now, on your Main Menu scene, select the camera and add 1 sensor, 2 controllers, 1 actuator, and 1 property. Name the property "Switch" "Int" Type, and set it to 0. Set up your logic as follows:



A change will need to be made in order to make yours work. The python controller should be Script:Mouse, not Script:showpointer. If you can't see the connections:

- Connect the Sensor to the 2 controllers
- Connect the AND controller to the actuator
- Leave the Python controller open-ended

If you play the game, you will see the mouse then! Now, back to the Menu, there are multiple kinds of Menus you can make when it comes to Menus popping up. The one I believe works best is a 3-layered button. To explain what this means, I'll run you through the steps of making one. Let's start with the layers themselves. (For this to work best, make the background of any images you use have transparency, so you can make invisible-yet-clickable faces.) Create a rectangle facing the camera where you want your first menu option to be. Copy the object until you have 3 rectangles, and make each one very slightly in front of the last. Now, Select the back rectangle, and on this one apply the UV face of the option's words. The second rectangle is the UV face for a graphic outline around the words if the option is selected in the game. It's not necessary but then again, it helps the player with selecting options and it looks nice. The top rectangle will be the selectable one--make the UV face applied only over a completely transparent area. Now, for the logic.

Text-Rectangle:

-No Logic Needed

Outline-Rectangle:

-Give it a Property named "Selected", Int Type, starting at 0.

-Make a Property Sensor, True Pulse, "Selected" Property is Equal to 1, Connected to an AND controller, Connected to a Visibility Actuator at "Visible"

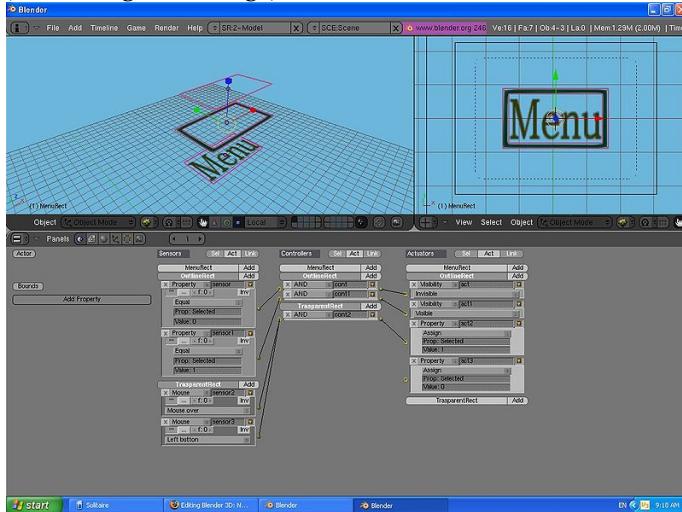
-Make a Property Sensor, True Pulse, "Selected" Property is Equal to 0, Connected to an AND controller, Connected to a Visibility Actuator at "Invisible"

-Make a Property Actuator, "Selected" Property is Assigned Amount 1 -Make a Property Actuator, "Selected" Property is Assigned Amount 0 -Create AND Controller and connect both Mouse Sensors. Make this AND Controller connect to the Outline-Rectangle's Property Actuator "Selected" Assign=1.

Transparent-Rectangle:

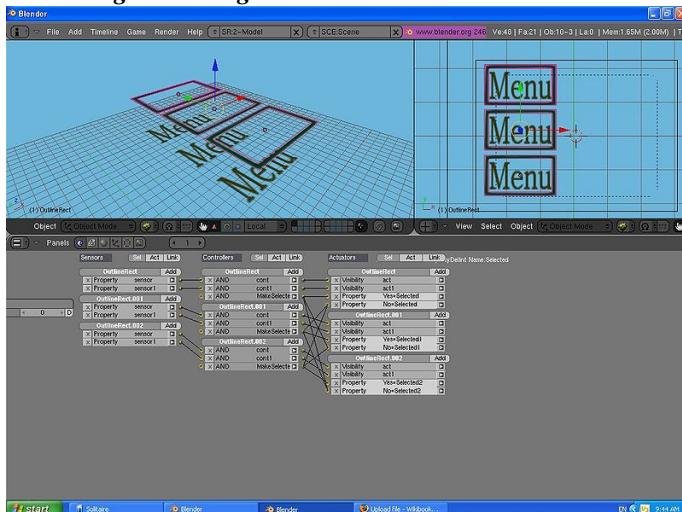
-2 Mouse sensors: "Mouse Over" and "Left Button". Set them to "True Pulse".

**(Click Image to Enlarge)**



Now here is the trickier part to explain. To make things a little more simple, name that last Outline-Rectangle's AND controller connected to the 2 mouse sensors. Select all 3 layers, and duplicate them 3 times. Make each one (in camera view) below the last, making sure to keep them parallel on the Z axis. Now, to make only one selected at a time, take the named controller of the first, and connect it to the Outline-Rectangle's Property Actuator "Selected" Assign=0 of second and third button. Do the same from the second button to the first and third one, and same with the third to the first and second. It should look like the picture:

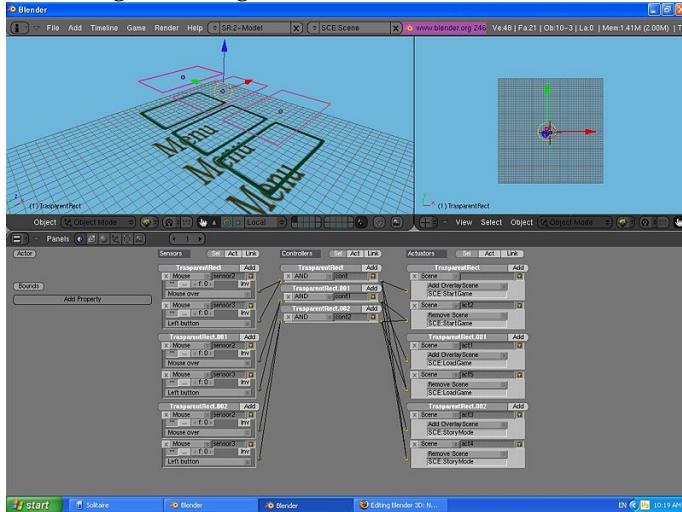
**Click Image to Enlarge**



This logic arrangement should yield (when the game is played) options in which if you click one it will light up, and if you click another, that one will go down and the other will light up.

Now this looks nice asthetically, but the use of it is now all you have to do is make some Scene Actuators. Add 2 more actuators to each of the Transparent Planes. Use the same method as you did with the Properties, but instead of "Property Assign 1/0" make 1 AND controller, 1 Scene Overlay and 1 Scene Remove for each transparent button. Now make 3 additional Scenes, and either make an entirely new menu for each, or to be easier just make colored rectangles on the scene, different colors for each scene -- either way, make sure that, if the scenes overlay, the additional scene won't cover up the buttons of the Main Menu Scene. So, now connect the 2 Mouse Sensors to the corresponding AND Controller, and connect that to the Corresponding "Overlay Scene" Actuator and 2 Opposing "Remove Scene" Actuators. Do this with the other 2 Transparent Rectangles. Make sure your new scenes also have cameras and that they are scaled properly.

*Click Image to Enlarge*



You can use this technique to make Menus and Sub-Menus Galore! It's fairly simple to make multiple sublevels of this and get back to the original scene with this method.

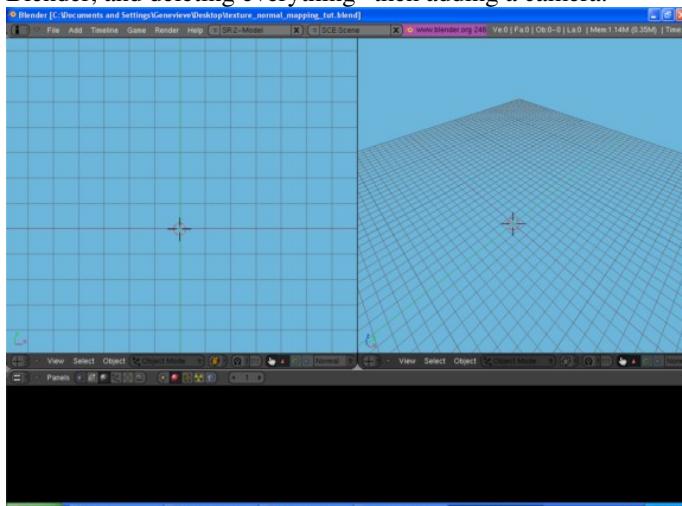
I hope this helped you make your menus work well.

## Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Game Engine/Game Creating Techniques(GUI)/Creating Moving Menus

In this section, you are going to learn a variety of ways to make menus with motions. These menus can be used in more ways than the simple pop-Up menus, such as in-game if you need to select something you can use this to bring up a menu smoothly. In this section there is heavy use of Properties, "Property" Logic Blocks, and especially IPO usage. When you do it once, it will be very much easier to do it again.

### Making Moving Selectors

To kick this tutorial off, let's just do a simple icon that will be next to the selected menu item. There will be 6 possible menu items, which can be controlled with both the arrow keys and the mouse. Let's begin by booting up Blender, and deleting everything--then adding a camera.

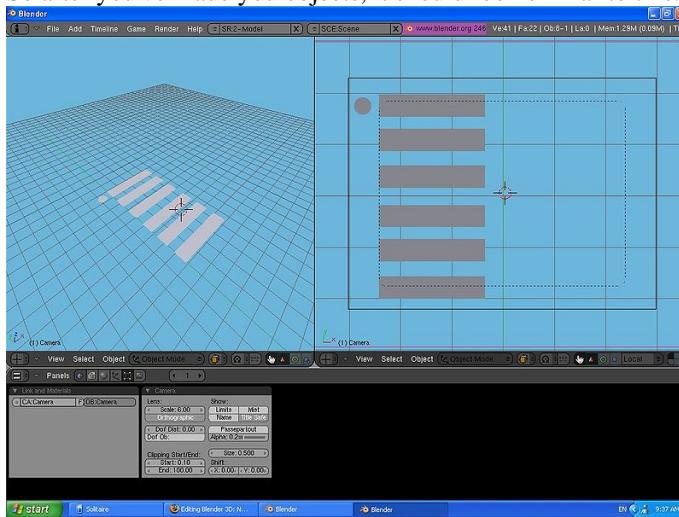


Set this camera to orthographic. Now I like making

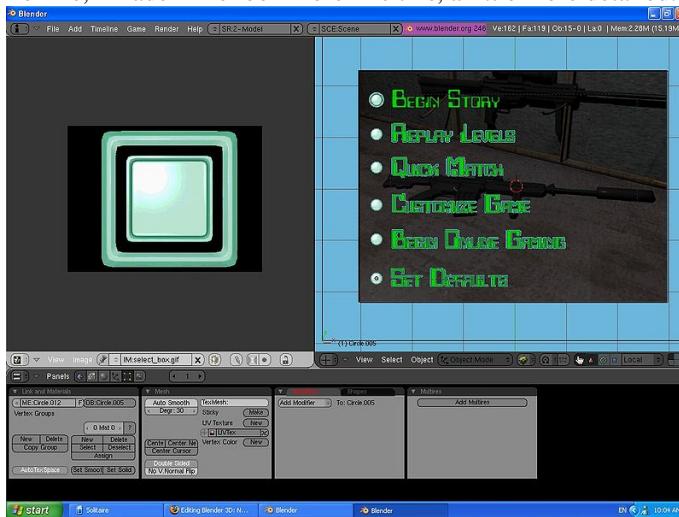
things look nice, but here are the basics of what you will need for this tutorial:

- Make 6 Planes, all different objects, and put them in a vertical list.
- Make a circle next to the first plane. This will act as the Selector.

So after you've made your objects, it should look similar to this:



For me, I made mine look more like this, a little more detailed:



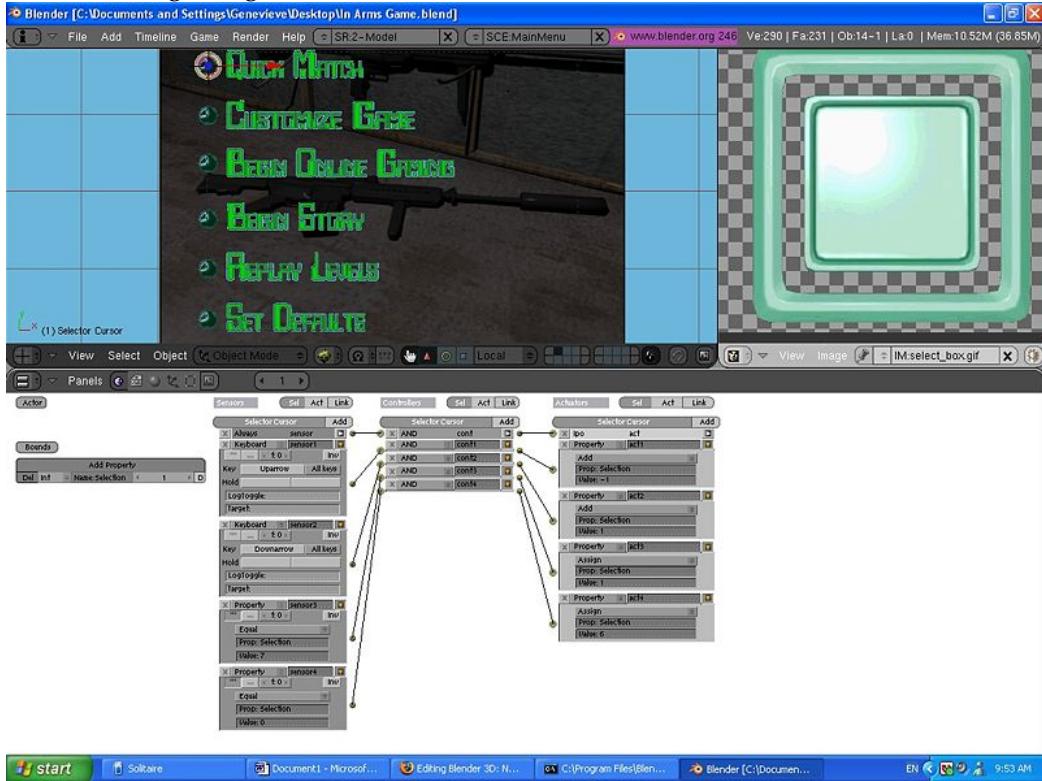
So now that we have our menu set up, it's time to set up the properties that will tell the selector how to move. Select the selector that will be moving and give it a property name "Selection". Make it "Int" type, and start it out at "1". To get this selector to move the way we want it, we have to set up its movements. Make sure you are on frame 1, and put the cursor on the first menu item, then Insert Keyframe for "Loc". Next, move to frame 2, and put the cursor on the second menu item, then insert "Loc" Keyframe, and continue in this fashion for all your menu items. To get this applied in the Game Engine, in the logic bricks, add an "Always" Sensor (True Pulse Enabled), connected to an "AND" connector, and connected to a "IPO" Actuator, set to "Property" with Prop:Selection. Now we can do 3 things with this selector to have it move like a selector in a video game:

#### **Up/Down Arrow Keys**

To make the up and down arrow keys work properly, Select the Selector and add four sensors, controllers, and actuators. Make two sensors Keyboard, with one set to "UpArrow" and one set to "DownArrow" (do not activate pulse triggering). Make the other two sensors Property Sensors, with True Pulse. For the first, make it "Equal", Prop:Selection, Value:0. For the other, make it "Equal", Prop:Selection, Value:7. For the Actuators, make all four "Property" sensors. For two of them, make them "Add", Prop:Selection, and make the first "Value:-1" and the second "Value:1". The other two should be "Assign", Prop:Selection, with the first at "Value:1" and the next as "Value:6".

Now connect them as follows:

**Click To Enlarge Image**



"Keyboard:UpArrow"--"AND"--"Prop:Selection/Add/Value:1"

"Prop:Selection/Equal/7"--"AND"--"Prop:Selection/Assign/Value:1"

"Prop:Selection/Equal/0"--"AND"--"Prop:Selection/Assign/Value:6"

Now you can play your game, and the arrow keys should work just fine!

### HotKeys

If you don't know what a HotKey, it's a way to access menu items or action by pressing a specified letter, instead of just clicking or arrow keys. This method will require a different step when you make your UV text textures, as you will need to somehow change one letter (either the color or possibly underline) to let people know that that is a HotKey action.

The HotKey system is easy to set up right alongside the IPO/Selection property. In each menu item, highlight or make special one unique letter in each menu item. Then, with the selector selected, set up 6 keyboard logic sensors, each one for a different letter. Add 6 controllers, and 6 actuators as well. Make all the actuators "Property", "Assign, Prop:Selection, starting with Value:1 progressing to Value:6"

### Mouse Selection

Our final selection method we can add is the mouse scroll over. Set up the Mouse Sensor to Scroll Over for all 6 invisible selection planes. Then, make the AND Controllers and connect to your original set of "Set Property" Actuators. that matches the Object's IPO property number. It's fairly simple.

### Make the Selection

The final thing to do is, once the selections are set up, make another 8 sensors, 7 controllers, 6 and actuators. Now make the 7th one a keyboard sensor for the ENTER key, as most people usually think they can start a game by pressing ENTER. The 8th Sensor is for another Mouse Sensor for "Left Click" so that if any of them is highlighted and you Left-Click the mouse, then it will be equivalent to pressing the ENTER key. That may be confusing as, unless you set up the dozen extra sensors, controllers, and actuators, you don't need to scroll over the selection to pick it, so that 8th sensor is your choice whether you want to add it. Connect the 7th and 8th sensor to the 7th controller. If you have the 8th sensor, make it an OR controller; otherwise, make the controller AND. As for the first 6 sensors, they should be Property Sensors for "Prop:Selection" "Value:[1-6]". Connect each to their own separate

controller. Now connect the 7th controller to all 6 Actuators, and the first 6 controllers connect to their own Actuator alongside the 7th. The Actuators can be whatever you want them to: whatever the selection causes to happen when you select the item selection.

## Make the Menu Move

So we created a selection-based moving icon on a menu, but what might happen if you select that item and need a sub-menu in-game? You can't very well pause the game every time (unless it's a pause menu). This is where making the menu itself move comes in very handy. In this section, you will make an object clickable, and once clicked it will give out a submenu, which in itself will be able to give out a submenu.

The IPOS

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Game Engine/Game Creating Techniques(GUI)/The "5-Layer" Button

---

## Scheduled Construction

This chapter is scheduled to be created by the Author. Sorry for the inconvenience.

# Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Game Engine/Game Creating Techniques(GUI)/Creating Object Outlines

---

## Scheduled Construction

This chapter is scheduled to be created by the Author. Sorry for the inconvenience.

# **Blender 3D: Noob to Pro/Advanced Tutorials/ Advanced Game Engine/Game Creating Techniques(Python)**

---

## **Needs Expert**

This chapter is not yet created, but it is something that would be very helpful to know about Blender. This book would benefit very much so if readers were to be so generous as to give helpful information on this subject, and expand it further. Thank You.

## **External links**

- [http://en.wikibooks.org/wiki/Blender\\_3D:\\_Blending\\_Into\\_Python](http://en.wikibooks.org/wiki/Blender_3D:_Blending_Into_Python)
- <http://www.theboomshelter.com/hss.html>
- <http://www.youtube.com/watch?v=7hbCEyvVYHI>

---

# Unit Seven: Miscellanea

---

## Blender 3D: Noob to Pro/modelling tips

---

This is not really a tutorial, but a list of handy tricks you will probably use a lot when modeling.

1: Extrusion: to extrude, press EKEY in edit mode with the vertices selected, move and click to place the extrusion. you'll basically need this for every model in blender, without it, most models would be really hard to create

2: Set Smooth: setting smooth will smoothen everything, without increasing the amount of polygons and vertices you can use it for models with a smooth surface. to apply set smooth, select the mesh, go to the editing buttons (F9) and in the Links and Materials tab, press Set Smooth. if you press the Set Solid button, it will go back to normal.

3: Remove Double: when you remove double vertices, they will melt together, this is good for attaching things to eachother. to remove double vertices, select the vertices, go to editing (F9) in edit mode, and in the Mesh tools tab, press Rem Doubt, the limit button next to it is the distance that the vertices have to have to be melted together.

Example use of these techniques:

Extrusion: <http://www.geocities.com/gauravnawani/tutorials/beginner/a-vase-modeling.html> (they explain subsurf here too, but I'd recommend set smooth, because it doesn't increase the number of polygons and subsurf does)  
[http://biorust.com/index.php?page=tutorial\\_detail&tutid=77&lang=en](http://biorust.com/index.php?page=tutorial_detail&tutid=77&lang=en)

Remove Double: [http://biorust.com/index.php?page=tutorial\\_detail&tutid=77&lang=en](http://biorust.com/index.php?page=tutorial_detail&tutid=77&lang=en) (same tutorial as in the extrusion section)

Set smooth: <http://www.blender.org/documentation/html/x2681.html>

If you want to add more handy techniques, please add them.

# Blender 3D: Noob to Pro/Creating Weapons based on 2D Images

Alright, this is my first tutorial here, I made this a while back, but hopefully it will be a good addition to this Wikibook. It was originally made for a game called Fable - The Lost Chapters<sup>[1]</sup>, but I think it is worthy of another place in the world. Please note that this is a WIP and will be updated time to time.

*Screenshots coming soon.*

**You can find a high quality video here.** <sup>[2]</sup>

## Modeling Technique 1

Modeling Technique 1 Part 1 Video <sup>[3]</sup>

### Part 1:

1. After emptying the screen, go to view and click background image, load, then choose a picture of the weapon you want
2. Go to top view and add a square, then delete two of the vertices and place one of the remaining on a point on the picture, and the other out of the way.
3. Select both vertices and subdivide.
4. Move the generated point to the next good-looking spot on the picture
5. With that selected still, select the outside vertex also, and subdivide.

(Note: A much easier way is to select one vertex, move it to where you want it, select the other vertex, move it to the next spot, then extrude the second vertex to the next point, rinse and repeat. This requires a lot less effort.) Another guy's note: For an easier way, just click CTRL+LMB where you want the next point to appear.

1. Continue until you are on your third to last one, then select your outside vertex and move it where you would move the next vertex.
2. Select the first and last vertices and go to mesh>>make edge/face
3. Extrude to half your preferred thickness, then extrude the rest of the way.
4. Select the center vertices on the blade, and scale up.

Modeling Technique 1 Part 2 <sup>[4]</sup>

### Part 2:

cont.)

1. Then move vertices to your liking.
2. Select the vertices of the handle, go to mesh>>vertices>>separate.
3. Add a modifier, choose subSurf, turn up the level until it looks close to what you want without too many vertices. Click apply next to the modifier.
4. Modify vertices (using proportional edit helps) to your liking.
5. Go to object mode, and turn off double-sided on any meshes that are.
6. If black appears on any parts of the mesh, highlight it, go to edit mode, select all vertices, go to mesh>>normals>>recalculate outside.
7. If there is still black, select those faces and go to mesh>>normals>>flip.
8. If there is still black, then you are missing a piece of mesh. Highlight the vertices around the hole, go to mesh>>vertices>>fill.
9. If black did not appear, then select everything (in object mode) and go to object>>join objects and say yes.

## Modeling Technique 2

Video Here <sup>[5]</sup> Download it for high quality.

Blade:

1. Select all with the 'A' key, and delete everything.
2. Go to view and choose background image >> use background image >>load and pick the sword you'll be doing.
3. Close the dialog after you see it in the background (you can change the brightness of it by changing the 'blend' option)
4. Make a bezier circle by pressing spacebar and going to add>>bezier>>bezier circle.
5. put a bar at each main point of your weapon (anywhere the curve changes direction)

Bezier Controls <sup>[6]</sup>

1. Modify the bars until they match.
2. You might want to go into wireframe view.
3. Go down and turn up the bevel depth a little, to give it some sharpness.
4. To lower the poly count you should turn down the DefResolIU number.
5. Go back to object mode, hit spacebar, go to object>>convert object type>>to mesh.
6. Modify vertices to your liking.

For the handle:

1. Do steps 1-7.
2. Turn up the bevel resolution, to give it some roundness.
3. Repeat steps 8-10

For That one thing (see video):

Just watch the video, but here is a summary.

Extrude and flatten a circle, use proportional edit to make it easier on you when you curve the circle.

For hilt guard:

1. Repeat steps 1-6
2. Add some extrusion.
3. Repeat steps 8-10.

Morerunes (talk) 17:52, 18 December 2007 (UTC)

## References

- [1] [http://en.wikipedia.org/wiki/Fable\\_\(video\\_game\)](http://en.wikipedia.org/wiki/Fable_(video_game))
- [2] <http://files.filefront.com/model+technique1+higher+qyavi;/9063237;/fileinfo.html>
- [3] <http://youtube.com/watch?v=isQzBvPGIx8>
- [4] <http://youtube.com/watch?v=R3UgOXYZMX4>
- [5] <http://files.filefront.com/modeling+technique+2+agkavi;/9221627;/fileinfo.html>
- [6] <http://wiki.blender.org/index.php/Manual/Curves#B.C3.A9ziels>

# Blender 3D: Noob to Pro/Match Moving

---

*Next page: Blender 3D: Noob to Pro/HDRi*

*Previous page: Blender\_3D: Noob to Pro/Build a skybox*

Match moving is the art of merging 3D with live action film. If you are unfamiliar with the concept, take a look at the WikiPedia article on Match Moving

Blender cannot perform the match moving itself, you use a 3rd party tool to determine the camera position and the way it moves, then import this data into Blender. While there are many software tools to do this, this page references two free options: Voodoo and Icarus.

## Icarus

Icarus <sup>[1]</sup> is a discontinued University of Manchester project which can be used for non-commercial work. The download links from the official page no longer function, but Windows and MacOS X versions are available from this Icarus video tutorial <sup>[2]</sup> by Colin Levy.

## Voodoo

Voodoo <sup>[3]</sup> is an actively developed free match mover available for Windows and Linux. Here is a tutorial on using Voodoo and Blender <sup>[4]</sup>

*Next page: Blender 3D: Noob to Pro/HDRi*

*Previous page: Blender\_3D: Noob to Pro/Build a skybox*

## References

[1] <http://aig.cs.man.ac.uk/research/reveal/icarus/>

[2] <http://peerlessproductions.com/tuts/pages/Icarus.html>

[3] <http://www.digilab.uni-hannover.de/docs/manual.html>

[4] <http://fictionality.co.uk/doku.php?id=tutorials:matchmoving>

# Blender 3D: Noob to Pro/Motion Tracking with Icarus

**Motion tracking**, also called **Match moving** <sup>[1]</sup>, is an essential element when integrating 3D elements with live footage. Motion tracking software is usually pretty expensive, but the Icarus application (Windows and Mac) is available for free for educational use. Icarus was later replaced by the commercial application PFTrack <sup>[2]</sup> and hasn't been updated in a while, but it's still very powerful. Other popular motion tracking applications are Voodoo <sup>[3]</sup> (for Windows/Linux; free for non-commercial and commercial use), SynthEyes <sup>[3]</sup>, Boujou <sup>[4]</sup> and 3D-Equalizer <sup>[5]</sup> (commercial).

The excellent CG prodigy Colin Levy <sup>[6]</sup> hosts Icarus, the Icarus import script for Blender, as well as a splendid video tutorial (see Download Icarus and Video Tutorial <sup>[7]</sup>). However, I lacked a brief text tutorial about motion tracking, so I decided to write my own. This tutorial is extremely brief and high-level, and requires some previous knowledge on video editing, 3D, and Blender.

**Note:** this tutorial was created using Mac OS X 10.5 Leopard, Blender 2.46, Icarus 2.09, and the Icarus Import Script for Blender <sup>[8]</sup> v1.07e (for Blender 2.41, written by Alfredo de Greef).

## Tutorial

### Phase 1: Preparing the Video Footage

**Note:** this tutorial explains the *Auto-feature Tracking* mode in Icarus. There are other options which gives more user control - see the Icarus **UserGuide.pdf** for more information.

1. Record your video footage. Having the camera on a tripod (thus limiting to just panning/rotating) simplifies the tracking, but Icarus can handle a hand-held camera as well. Filming a background with orthogonal lines (that can align to X/Y/Z dimensions) also helps the tracking.
2. Capture/import your video footage to your computer. Icarus handles video up to DV resolution (720\*576 pixels).
3. Start the Icarus *Calibration* application (there is also a *Distortion* and a *Reconstruction* application).
4. Create a new project (Project->New).
5. Import your video footage (Project->Import Movie).
6. Fill in the *Camera Parameters* information in the window that pops up - especially the *Camera Motion* and the *Pixel Aspect* options.
7. In the left panel, expand the group called *Coordinate Frame*. You should see X Axis, Y Axis, etc.
8. Click the Z Axis tool (blue) and mark vertical lines in your video footage. Use the X Axis (red) and Y Axis (green) tools to mark horizontal lines (up to you to decide which should be X and Y).
9. Estimate the focal length (Camera->Estimate Focal Length).
10. Navigate in time in your video footage using the time slider (beneath the video image). Add more X/Y/Z marker lines on a few key frames, especially as new pieces of the background are revealed when the camera moves.
11. Save your project (Project->Save).
12. Start the tracking process (Camera->Track and Calibrate). This will take some time.
13. Export the results in human-readable form (Project->Export 3D Motion, select *Human Readable (\*.txt)* as file type).

## Phase 2: Importing the Motion Tracking Data into Blender

1. Start Blender, and open a Text Editor view.
2. Open the Icarus import script **ICARUS\_import241.py** (File->Open).
3. Start the script (File->Run Python Script). You should now see the Icarus Import screen.
4. Press the *FSEL* button, and open the results you exported from Icarus.
5. Press the *Create Curves* button. This imports the camera motion from the Icarus data and applies it to the Blender default camera.
6. Press the *Feature Points Mesh* button. This imports 3D shape dots from the Icarus data, which helps as reference when you want to align your own 3D elements to the video footage.

You are now ready to add your own 3D elements to the Blender scene.

## Phase 3: Compositing 3D Elements on top of Video Footage

If you want to easily composite the 3D elements on top of an image, you can add the image as the rendering back buffer in Blender (Scene tab in the Buttons view). However, this doesn't work for videos, so we need another solution.

1. In Blender, switch to *SR:4 - Sequence* in the layout dropdown menu at the top of the screen.
2. In the *Video Sequence Editor* view (middle of screen), add your video file (Add->Movie). Move the new strip to layer 1, frame 1.
3. Add the current scene to the sequence (Add->Scene, Scene). Move the new strip to layer 2, frame 1.
4. Select the scene strip on the second layer (right-click).
5. Open the *Scene* panel in the Buttons view, and then open the *Sequencer* sub-panel.
6. Change the *Blend Mode* dropdown from *Replace* to *Alpha Over*. Your 3D elements should now render over the background video in the top-right preview screen.
7. In the Render panel, enable *Do Sequence* just below the ANIM button. This will enable the background video when rendering.

## Troubleshooting

- If your imported Feature Points Mesh looks a bit spherical, you need to generate camera distortion data using the Icarus *Distortion* application.

## References

- [1] [http://en.wikipedia.org/wiki/Match\\_moving](http://en.wikipedia.org/wiki/Match_moving)
- [2] <http://www.thepixelfarm.co.uk/products/products.aspx?PID=3>
- [3] <http://www.ssontech.com/>
- [4] <http://www.2d3.com/product/?v=1>
- [5] <http://www.sci-d-vis.com>
- [6] <http://www.peerlessproductions.com>
- [7] <http://www.peerlessproductions.com/tuts/pages/Icarus.html>
- [8] [http://www.peerlessproductions.com/tuts/IcarusTutorials/ICARUS\\_import241.zip](http://www.peerlessproductions.com/tuts/IcarusTutorials/ICARUS_import241.zip)

# Blender 3D: Noob to Pro/Create a Clayman

---

## Create a Clayman

(under construction)

**This Tutorial was made using Blender version 2.41**

Upon completing this tutorial, you will have made something like this:

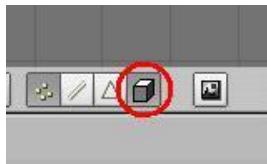


Completed Clayman Render

## Making The Clayman

Upon start, delete the default cube (X-KEY) and switch to front view with NUM1. Now add a mesh plane and under The Edit Buttons Window, Mesh Tools Tab click the subdivide button. Now select the 3 vertices on the left and delete them. (B-KEY for Border Select Tool)

If you can't select the vertices then deselect the Limited Selection to Visible button.



Now select/deselect all vertices with A-KEY (we want them selected) In the Modifiers Tab in the Edit Buttons Window, Click Add Modifier > Mirror You should now see something like this.



I, the copyright holder of this work hereby release it into the **public domain**. This applies worldwide. *In case this is not legally possible, I grant anyone the right to use this work **for any purpose**, without any conditions, unless such conditions are required by law.*

Now select the 3 right vertices and extrude them on the X axis by movement of 1.0 units by pressing E-KEY (*Extrude > Only Edges*), X-KEY (to constrain it to the X axis) and holding CTRL to constrain it to simple movements. Do this twice.

Now extrude the legs by selecting the 3 bottom-right vertices and extruding them on the Z axis by 1 unit. Do this twice as well.

Now extrude the top of the clayman on the Z axis (in units of 1.0) 7 times so that it looks like this.

Now let's make the arms by extruding the 2 boxes that are 3 boxes down from the top. Let's extrude it on the X axis by 1 unit 4 times so that it looks like this.

Now comes the extrusion of the whole body. As you can see, we only have a flat shape of a rough "Block". so lets move our view around with the NUMPAD arrows and extrude it on the Y axis. Select all vertices with A-KEY. *Extrude > Region* on the Y axis until it reaches 2.0 units. After done correctly, it should look like this.

With the Clayman successfully extruded, its now time to add some definition to him by adding a subsurface modifier. In the Edit Buttons window, under Modifiers, click *Add Modifier > Subsurf* and set the Levels bar to 2. It should look somewhat like this.

Now, Select all vertices with A-KEY, and click the Set Smooth button in the Links and Materials tab. Let's get to shaping the Clayman.Using the Border Select Tool (press B to get the Border Select Tool), grab the according vertices to make a shape like this.

after that's done, select the head vertices and grab them on the Z axis and move them up by 0.2 to make it look like this.

Now select all vertices with A-KEY, and scale them down (S-KEY) on the Y axis until it reads 0.6000. And finally, select the underside of the foot (the vertices that make it up) and subdivide it to make ir more flat on the bottom.

## Adding Materials

Start by going to the Material Buttons window (F5) and Clicking on Add New, and going down to the column of 3 rectangles that have one gray one and two blank white ones. Click on the gray one and in the top-right corner of the pop-up window, you'll see a Hex Code. Click on the current hex code and type in this new hex code to get the right color: 73BCF7. Now hit F6 to go to the Texture buttons window and click on Add New. Change the Texture type to Stucci. Now you'll see a new tab appear. On it, for NoiseSize enter 0.150. Now go back to the Materials Buttons window. On the far right there will be three tabs in one. On the Map To tab, Deselect Col and select Nor. Now your example should look like this.

## Inserting Armatures

Now lets move on to inserting armatures in our clayman. Lets start by going into object mode (alternate between Object and Edit mode with TAB) and adding an armature. Press *Space > Add > Armature*. It automatically puts you in Edit mode upon adding the object. Switch to Object mode and grab it along the Y axis by 1.0 units so that it's inside the clayman. Sure, you can't see it now, but turn on X-axis Mirror Edit and X-Ray in the Edit Buttons window.

Now switch back to edit mode, and select the top of the bone. Extrude it on the Z axis by 1.4 units two times so that the tip of the second extrude is right in the center of the two arms. Now extrude it on the X axis by 3.0 units, then grab the tip of it and move it on the Z axis by 0.1700 units. Now extrude it on the X axis by 2.0 units twice. And do the same to the other side. It should look like this.

Now let's extrude the neck. Extrude the top bone along the Z axis by 1.0 units to make a pretty good size neck. Then extrude it up on one side so that the position reads like so. If you can't get it exact then at least try to get it as closest to the number as you can.

Then do the same to the other side so that it looks like 2 antennae. Now let's extrude the hips and legs. Lets start by extruding the bottom bone by 2.0 units on the X axis and for the Z axis, -1.0 units. Then extrude the foot by -2.0 on the Z axis. And last, select a forearm bone (select a whole bone by clicking the middle of it) and in the Edit Buttons window, you'll see a tab called Armature Bones. Deselect the Con button in that tab, and do the same with the other forearm bone. This allows us to be able to move the forearm out, instead of having it locked to another bone.

## Applying The Armature

Ok, now we're going to apply the armature to the mesh so that when we move the bones, the mesh moves along with them. Lets start by switching to object mode and selecting the clayman mesh, and under the Modifiers tab, click apply on the Mirror modifier. You may see a pop-up, just click OK. Not only have we just made room in the Modifiers tab, we applied the mirror modifier to the mesh, so now we will have to edit both sides if we want to change something. But worry not, we wont need to change anything from here on. With that new space in the modifiers tab, let's add an Armature modifier. You may see a line that says "OB:" type in the name of the Armature in here (the default name for an armature is "Armature" unless you have made more than one) also, deselect the Envelopes button,for we will not be using envelopes in this tutorial. Here is what you should have.

Now, select the Armature and press **CTRL + TAB**. If you already have a bone selected, it should be highlighted blue now. We have just switched to Pose Mode. Now, while in Pose Mode, select the clayman mesh (if you cant select it, zoom in closer to get some of the bones out of the way) and press **CTRL + TAB** on the clayman mesh. We have just entered Weight Paint Mode. This allows us to assign vertices to the bones so that the mesh moves with the bones. Let's start by setting up the weight painting. You should see a new tab called Paint in your Edit Buttons window. Adjust your settings so that they conform to this.

Start by selecting the top-right bone of the head and start weight painting the front and back (you'll have to switch your view so that you can see the back to get the painting in there) Weight Paint it according to this.

Now Weight Paint the Neck. (pictures not shown of back view, but be sure to get the back anyway)

Now the Shoulder.

Now the Forearm.

Now The Hand.

And after getting the other side of the body, let's get the upper chest.

Now the middle chest.

Now the lower chest.

Now for the hips.

Now the Feet.

After doing the other side of the leg, switch back to Object Mode with **CTRL + TAB**, and select the Armature. You should already be in Pose Mode with the Armature. Now select a bone and grab it to move it. if you have Weight painted correctly, the Clayman mesh should move with the bone. There's two tips to moving the bones around in Pose Mode.

1) In the Edit Buttons window, whether Automatic IK is turned on or not affects how the bones move.

2) Where your view is affects how the bones rotate. (unless you have it constrained to an axis)

If you find that a part of your clayman isn't moving, just switch back to Weight Paint mode for that bone and repaint it.

Back To Blender 3D: Noob to Pro<sup>[1]</sup>

## References

[1] [http://en.wikibooks.org/wiki/Blender\\_3D:\\_Noob\\_to\\_Pro](http://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro)

---

# Blender 3D: Noob to Pro/Organic Modeling

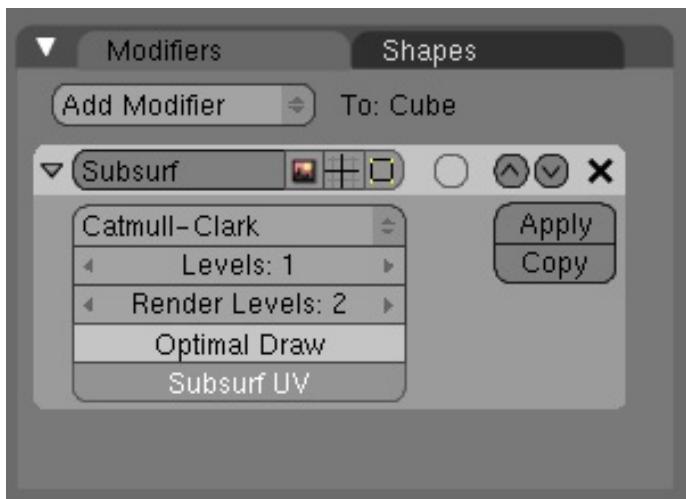
---

## Subsurface modeling

Organic modeling is considered by some the most challenging. Non-organic can mostly be accomplished by extrusion and scaling. Organic modeling on the other hand involves mainly curves, as nature has a thing against straight lines. Because of this, organic modeling is usually done with subsurfaces. To subsurf a mesh, first select it and navigate to the 'Editing' tab. Then go to the modifiers stack and add a new Subsurf modifier.

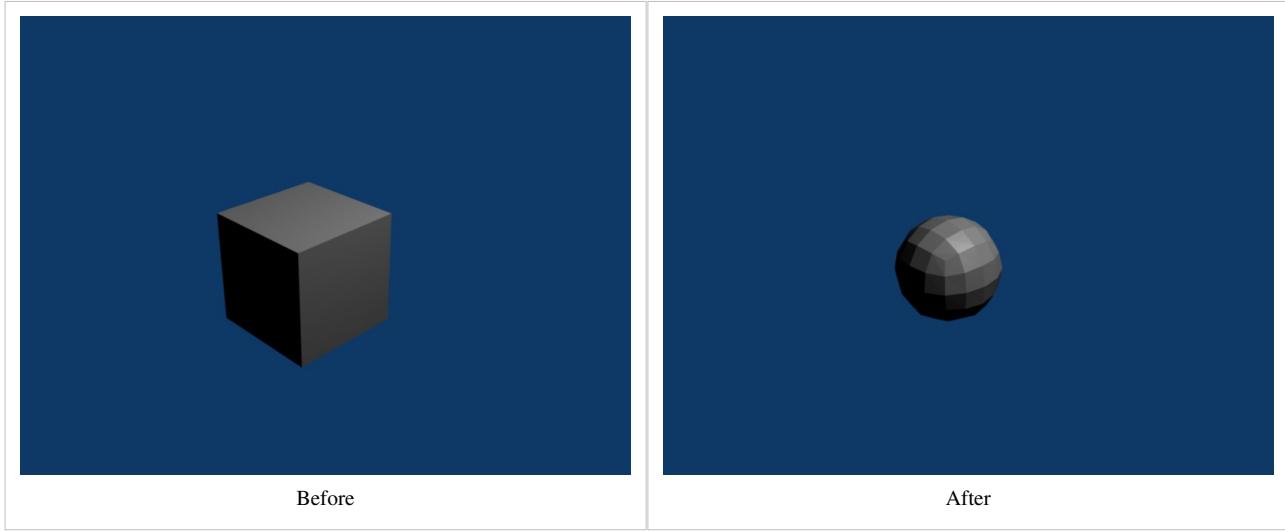


In Blender 2.5 Alpha 0, Subsurf has been removed from the modifiers list. Instead hit **Ctrl**, followed by a number which will specify the subsurf level. For example **Ctrl-1** adds a new Subsurf modifier of level 1. This also works in Blender 2.4. Increasing the level greatly increases the number of verts in your model, so make the level relative to the number of vertices in your original model (pre-subsurfed). Subsurf often works best in conjunction with smoothing, so be sure to set your object to smooth, again in the 'Editing' tab, or in 2.5 A0 under 'Object Tools', which you can bring up by hitting the **T** key in the 3D viewport.



In this way you can create a smoother, higher-poly shape based on that of the original mesh. This is controlled by the vertices, edges and faces of the original. In order to control the rounding of your mesh you can use two methods: loopcuts, which is a very sloppy method as it adds more verts to your mesh, which serve very little purpose and can get in the way of modeling; and Edge Creasing. The latter can work extremely well in most circumstances. By default all edges are declared uncreased, and so allow complete rounding in subsurfing, by creasing the edges using

**Shift-E** you can far better control the amount of rounding on your mesh, and most importantly, without adding extra vertices. However, unfortunately Edge Creasing is not available in Blender 2.5 A0. Yet. And so loopcuts and extra verts are the only option.



Now, start shaping your mesh into an organic shape. There are various tutorials for modeling bodies and faces, but it is often a good idea to use references. Only use quads - that is, shapes with four vertices. Triangles do not subsurface well, they create bad looking rough surfaces - avoid if at all possible. Also avoid Poles, which are vertices connected to anything other than 2 or 4 other vertices, as they create literal creasing in your subsurfed mesh, that is, sudden bunching up and pinching of faces.

To make the object shape properly, you will need good edgeloops and edgeflow. This means that all your vertices and/or faces line up in a continual line or curve around your model. It is possible to select edgeloops all in one, fantastic when you have to resort to them in controlling rounding, by holding down **Alt** when selecting a vert. This really expands into topology, which is an advanced subject. I would recommend visiting <http://www.subdivisionmodeling.com/forums>, although it has just been closed down, which was a great place to start, however it still (at the time of revision) is active and well worth a look if you are struggling with modeling heads. It is best not to have too many vertices to avoid making the subsurfaced shape look rough - in other words, the original shape should be quite low-poly.

# Blender 3D: Noob to Pro/Free Blender Samples

These samples are free to use and distribute, but please check the license first. Please note this page is very small.

## Weapons

M14 KC semi-auto .30-06 caliber rifle by Big Bill57 - Public Domain [1]



## References

[1] <http://dodownload.filefront.com/9560247/>

4edcc1e2764659bf0a30ce8332e11534fd31b052d14c974f95550348430bb02834469984a9dbe79Incomplete

# Blender 3D: Noob to Pro/Understanding the Fluid Simulator

## Understanding the Fluid Simulator

When I first encountered the Fluid Simulator I had a hard time understanding its behavior, especially the *Start time* and *End time* didn't seem to make any sense at all. Going on a Google spree revealed that many people have problems figuring out the secrets behind fluid simulation, and I didn't find any truly helpful guides. In this little guide I'll try to explain it in a newbie friendly way. It may not be entirely correct, although it might help newbies understanding how it works.

## First of all

*Start time* and *End time* **are** in seconds. Don't forget this. Even if your simulation seems to go insanely fast when you set *Start time* to 0 and *End time* to 10 and having 250 frames in your animation with 25 frames per second, there is a good reason for this. For now, just remember this, don't let your mind wander and believe that the values are in milliseconds or that you have to do some wicked math dividing/multiplying with frames and so on.

## Setting up the scene

We'll learn how the fluid works the practical way. plane

- Start with the default box, this simulation will be very simple.
- Let's work in wireframe mode, press the **ZKEY** to turn off solid mode.
- Go in to camera view by pressing **Numpad0**.
- With the box selected, scale it up to two times by pressing **SKEY**, then **2**, then **ENTER**. This fits the camera fairly well.
- Press **Numpad7** for a top view.
- With the box selected, press **Shift-D** (don't move the mouse or else the duplicated box will move) then press **ENTER** to confirm the duplicated box's location.
- While the new box is selected (and in the exact same spot as the other box), press **SKEY**, then **.5**, then **ENTER** to scale it to half size.
- Stay away from the mouse, accuracy is important here and I'll explain why later.

- Now we want to move the new box into one of the upper corners. Press the **GKEY**, then press the **XKEY**, type in **-1**, press **ENTER** and the box should move to the left wall of our larger box.
- We want the box in a corner, so press **GKEY Y 1 Enter**, the box should now be in the top left corner from our current view.
- However, we're in 3 dimensions, not 2 so click **Numpad1** for a side view. This time we'll move the box up along the Z-axis: **G Z 1 Enter**.
- Excellent, our setup is done.

## Setting up the simulation

- Make sure you're in Object Mode, and that you followed the above steps precisely.
- Select the smaller box and click **F7** twice. You should get a panel where the rightmost pane says "Fluid Simulation". Click **Enable**.
- Our small box will be the fluid, so just click the **Fluid** button. That's all there is to do with the small box.
- Now select the large box. The Physics panel should still be visible, click **Enable** in the Fluid Simulation pane and then select **Domain**.
- By default your animation should have 250 frames. Rendering should also be set to 25 frames per second by default, this tutorial assumes this setup.
- Since we got 250 frames and 25 frames per second that means our animation is 10 seconds long. So here comes the tricky part, which actually isn't that tricky at all:
  - Start time is by default set to 0 seconds. This means that on the very first frame the simulation has just begun. You could increase this value to say, 1 second and that would mean that on the first frame the fluid simulation has already run for 1 second. We don't want to do this, so keep it at 0 seconds.
  - End time is by default set to 0.3 seconds. What does this mean? This means that on the 250th frame the simulation has run for 0.3 seconds. However, by default our animation is 250 frames long with 25 frames per second, making those 0.3 seconds stretched over 10 seconds. Basically this means that we're watching the show in slow motion, or slightly less than 1/33 the realtime speed. So now you may think "it looks quite realtime to me!", and yes, it does, but why does it do that? Well, that's hard to explain. Consider this: In a world without friction, how far would a drop of water fall after 1 second? The answer is about 4.9 meters. So, if a drop of water falls from 4.9 meters it will take 1 second before it reaches the ground. How long would it take the waterdrop to reach the ground if it falls from 3 centimeters? About 0.078 seconds. So why do I mention 3 centimeters? Because by default the size of our domain is 3x3x3 centimeters, or **really** small. If you're like me, you were probably thinking that the fluid was flowing around in a bathtub or a barrel, not in the wrapping of a cupcake. Set **End time to 10 seconds**.
- Since our imagination likes big things, let's crank up that cupcake to say, a swimming pool. Make sure the big box is selected in Object Mode In and look at the Fluid Simulation pane. Just to the left of the "BAKE" button there should be 3 other buttons, possibly named "St", "A", "B". Click **A** for advanced options.
- Some new boxes should appear, Gravity (should be -9.81 for the Z-axis, nothing else), Water and the option we're looking for, "Realworld-size". Also Gridlevels and Compressibility, but let's not care about those now.
- The "Realworld-size" value says how large our domain is in meters, and as you can see it's 0.03 meters by default, or 3 centimeters. We want it huge, so crank it up to 10, which is the limit for Blender 2.45. Now our swimming pool is 10x10x10 meters (don't drown!), remember this because scale matters with fluid. Do not think we're playing with cupcakes again :)
- Now click **BAKE**, and read on while your computer is chewing zeroes and ones.
- Remember how I told you to be very accurate about placing that second box? And how I began talking about gravity, falling waterdrops and stuff? Well, now you're going to see why.
- As stated, our "swimming pool" is 10x10x10 meters. The smaller box we added is exactly half the size (well, in terms of length/width/height, not volume), or 5x5x5 meters. Remember that a drop of water would fall 4.9 meters

in 1 second? And that our animation got 25 frames per second? This means that the bottommost part of our blob of water will be exactly 5 meters above the "ground", and that after 25 frames our water should be very close to the ground.

- If you got a fast computer, Blender should be done baking by now. Go to frame 25, for example by using the arrow keys (up/down goes 10 frames forwards/backwards, right/left goes 1 frame forward/backward). Take a close look at the blob, then go forward 1 frame. Notice how the blob hits the ground? Rings a bell, doesn't it? :)
- Although, we're not done! We gotta render our swimming pool. It's easy, but takes time, hit **Ctrl-F12** and go make dinner.
- When the rendering is done, press **Ctrl-F11**, and think of a 10x10 meter large pool. You might want to keep an eye on your kids if your local swimming pool acts this way, though.

## Final notes

Scale matters. It's really difficult to understand fluid dynamics on a very small scale, especially when you don't even know what scale is used. The "Realworld-size" value seems to be left out in many guides, I would recommend you set it to something you can relate to, or you'll end up with simulations that look really slow/fast or having an **End time** value that seemingly makes no sense. Further I'm not a mathematical genius, for all I know I could be way off with my explanation, although this way the values makes sense to me, and I'm able to make fluid simulations without "guessing" on values for **End time**.

## Extra Practice

This YouTube tutorial on fluids might also help: Link <sup>[1]</sup> and this Realistic Water Texture <sup>[1]</sup>

## Links

Youtube: Comparison of different values for resolution and real-world size <sup>[1]</sup>

## References

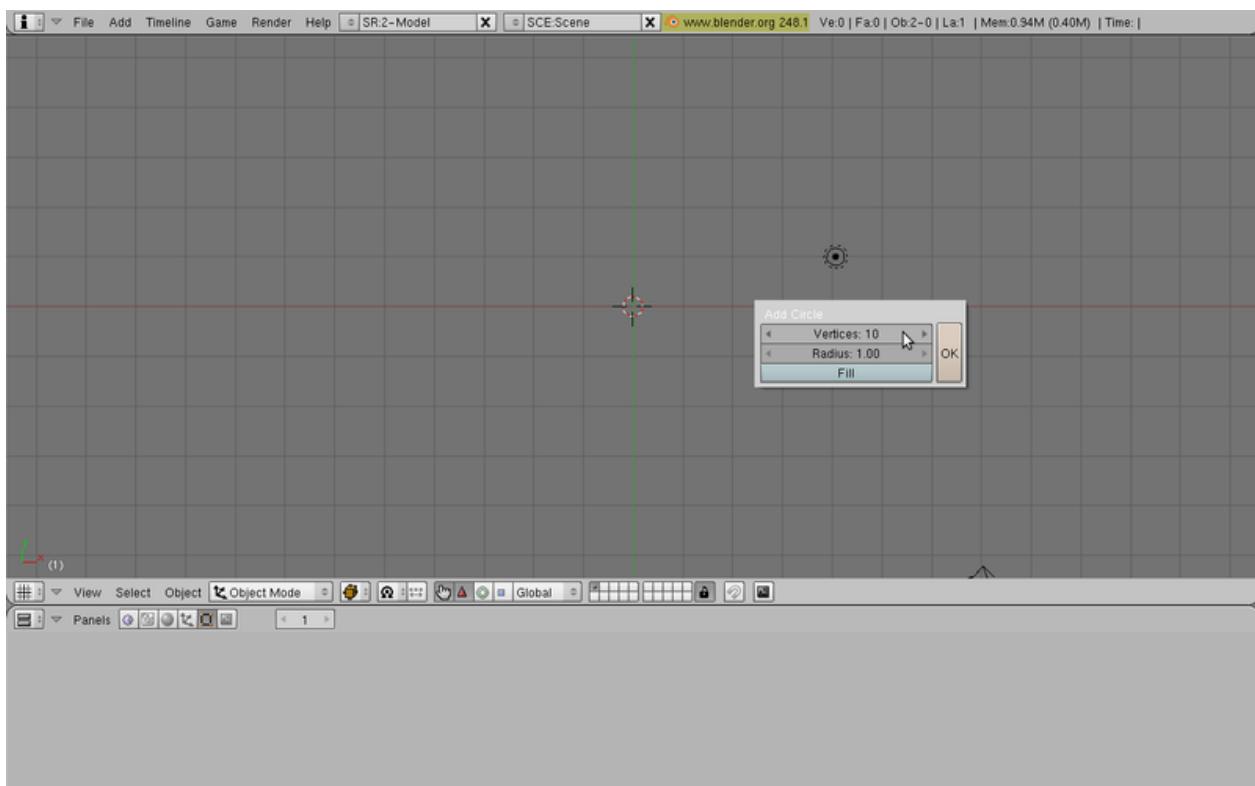
[1] <http://www.youtube.com/watch?v=okHt4z3FGGM&hd=1>

# Blender 3D: Noob to Pro/Creating a jewel in Blender

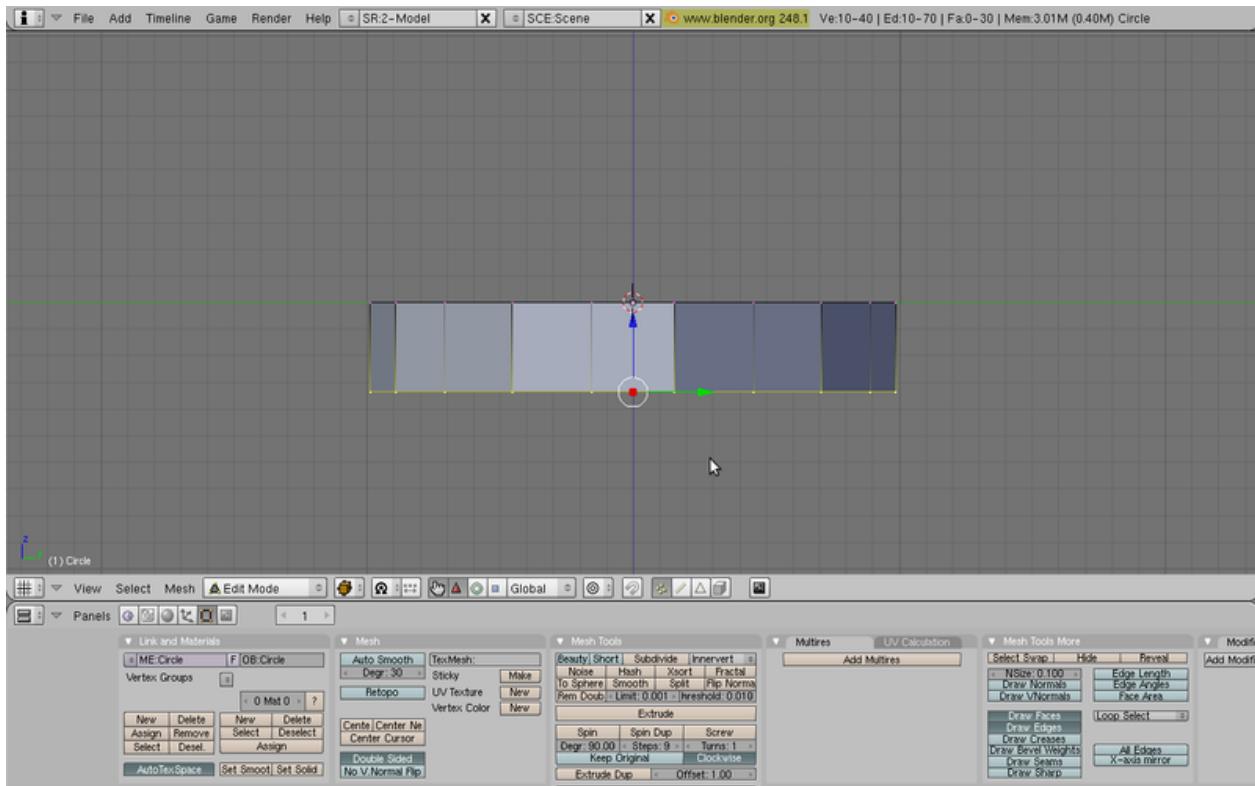
In this tutorial we will create a jewel in Blender. It is fairly simple. I recommend you do this tutorial if you are a noob, because it explains some basic features, but I suggest you read the tutorial syntax and the pages at the very beginning of this Wikibook first.

## Modeling the jewel

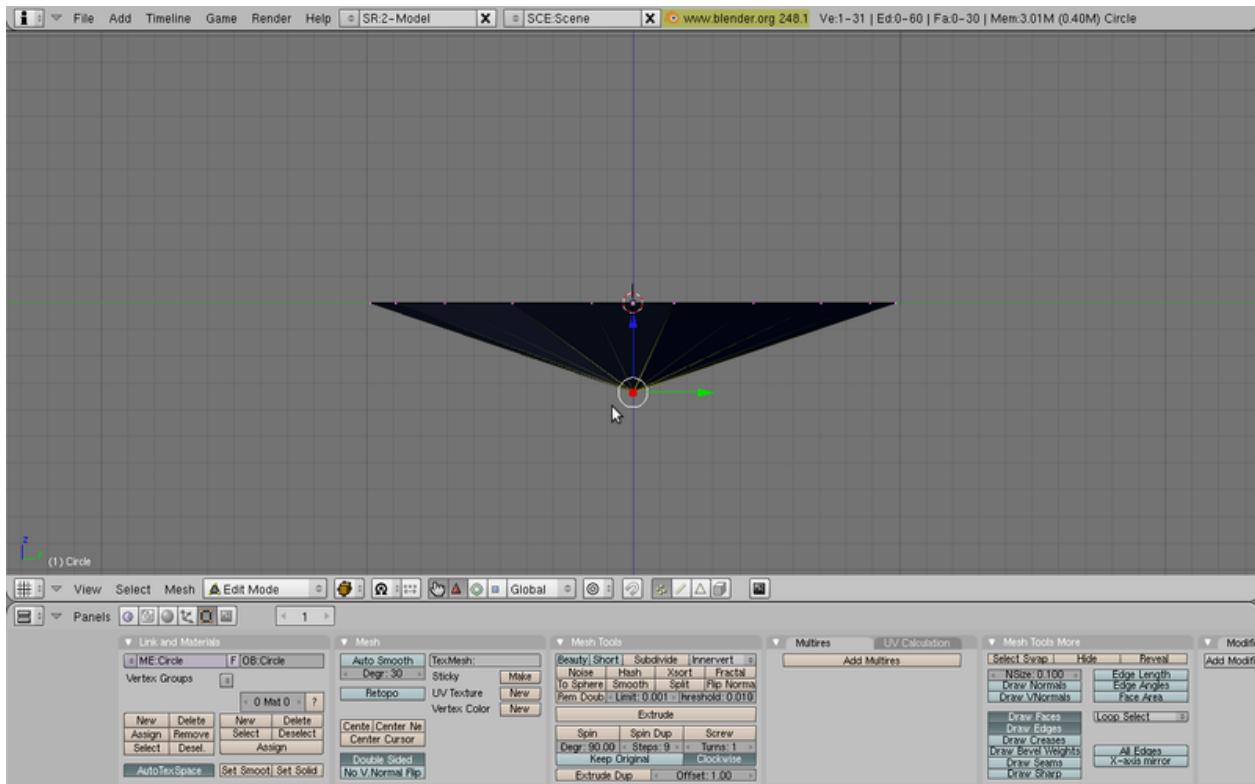
Start up Blender if you haven't already. There is a cube in top view (it looks like a square because it is in top view.) Delete it by pressing the X key. Then, to begin modeling, add a circle by pressing SPACE>>ADD>>MESH>>CIRCLE. Set the vertices to 10 and make sure it is not filled in. Then press OK.



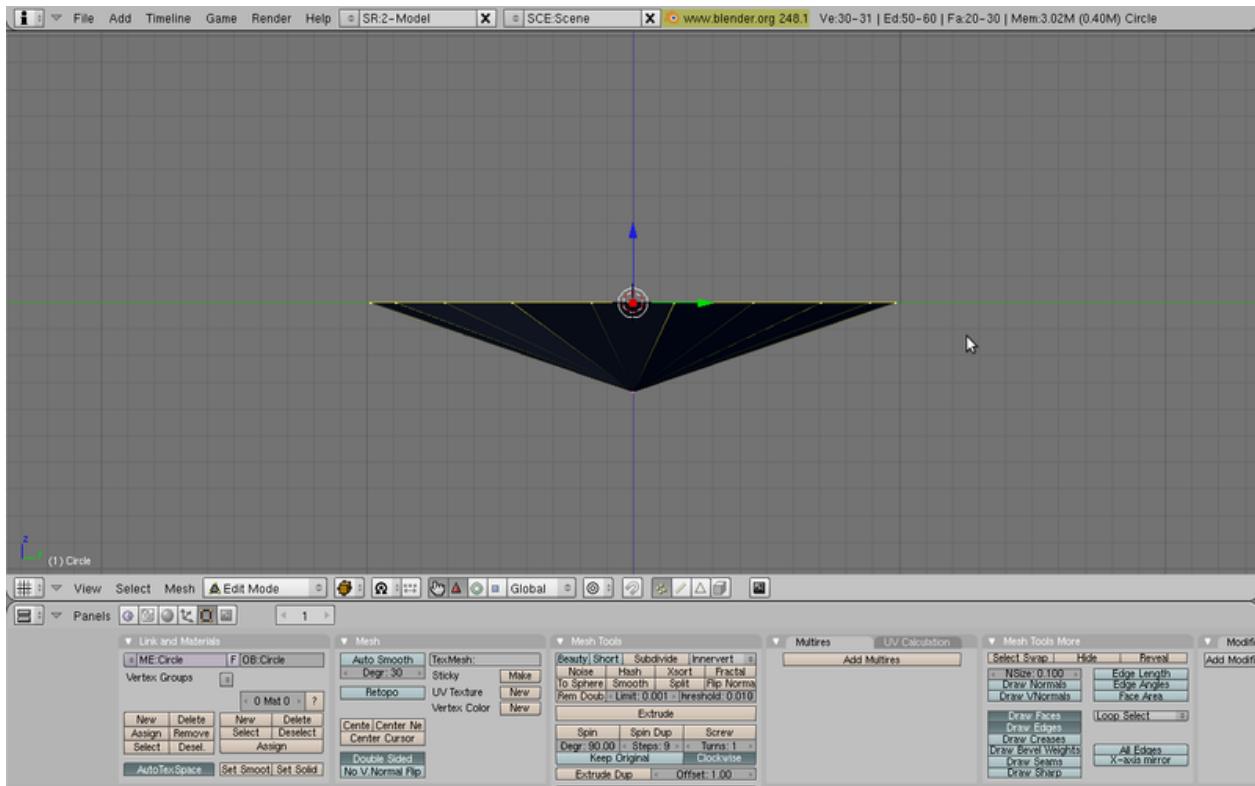
Next, we will extrude the vertices down. First enter side view by pressing Numpad 3 on your keyboard. As you can see, the circle is flat and not filled in. We will give it some depth. Switch to Edit Mode (TAB) and extrude the circle by pressing the E key and selecting "Only Edges". Move the mouse down and click to confirm the position (you may want to limit to Z axis by pressing the Z key).



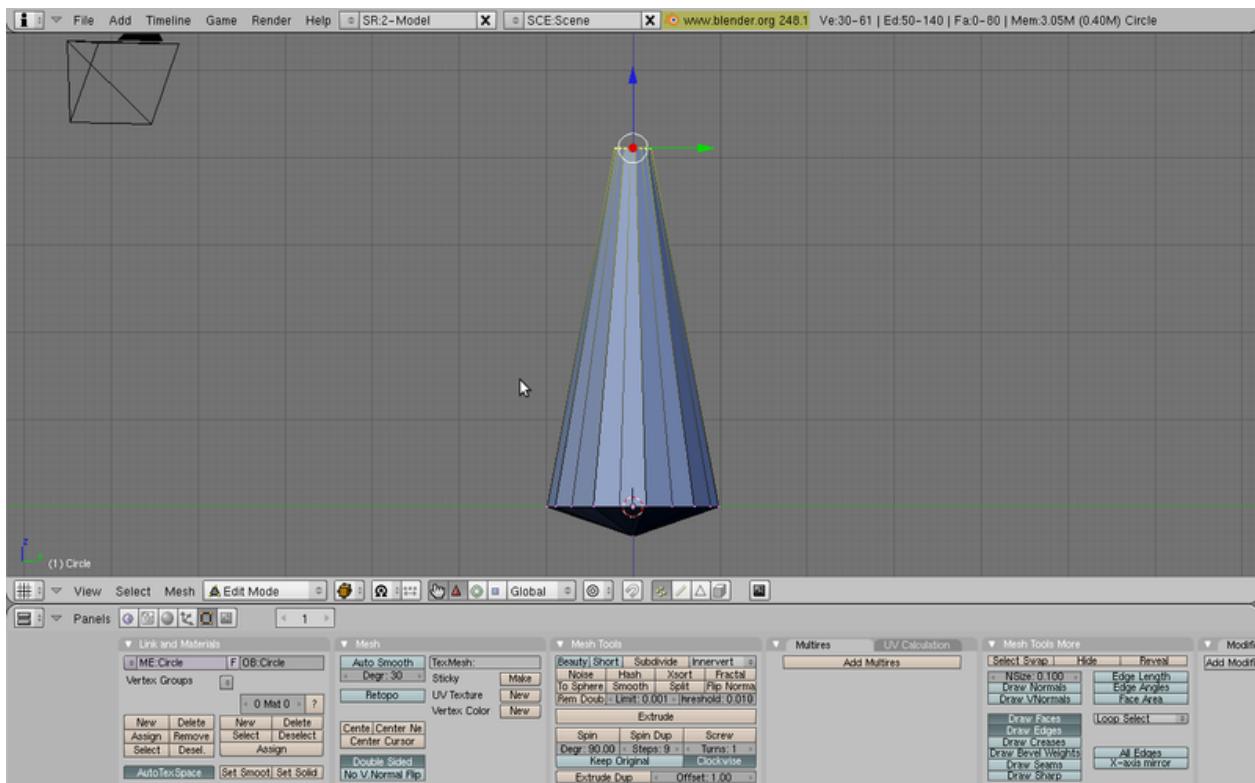
OK, what we just did is turn the circle into a hollow cylinder by extruding. But we don't want a cylinder. We want the bottom to be a nice tip. To do this, press Alt-M. This creates a Merging menu. Select "At Center". Now the bottom is a nice tip, like we want it!



Now we will edit the top of the jewel. Press A to deselect everything and press the B key. This enters Box-Select Mode. Drag a box over the top vertices to select them.

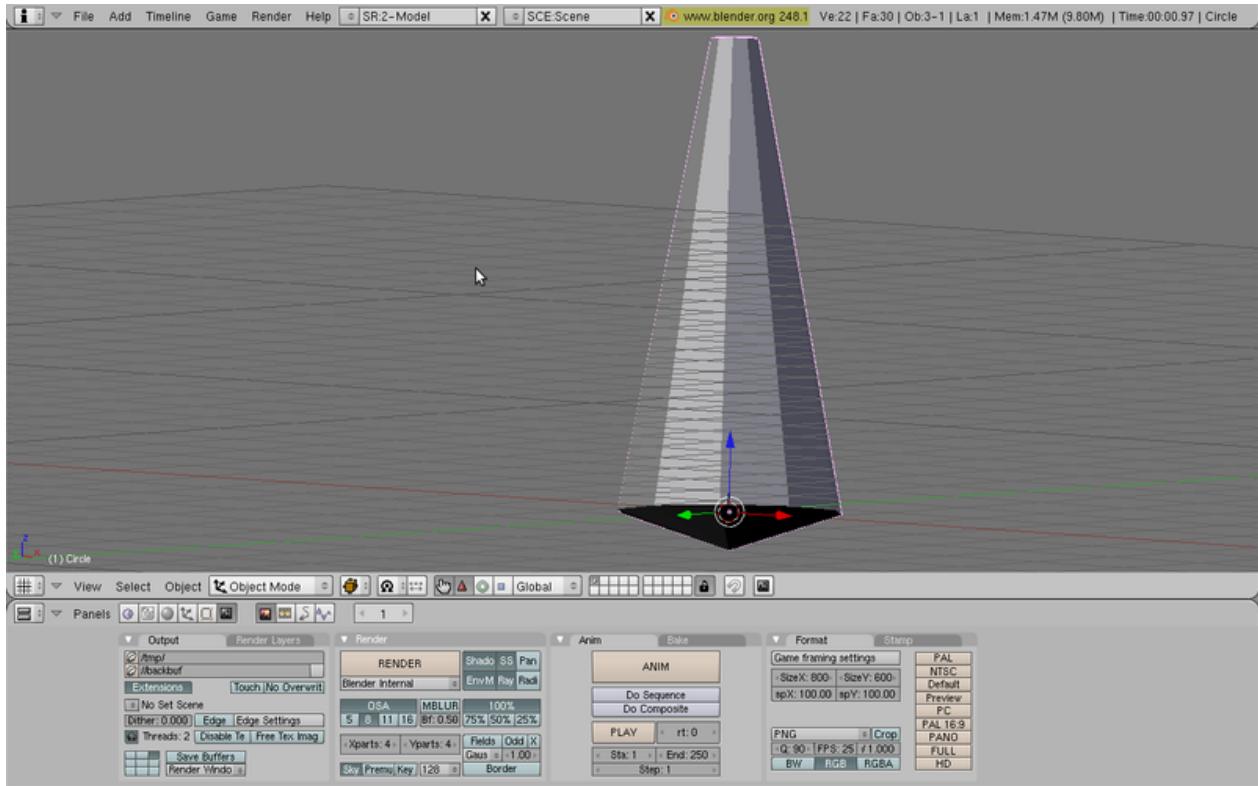


Now extrude the vertices upwards a bit (again: E key and "Only Edges"). Next, we will scale down these vertices to look a bit more like a jewel. Make sure the ring of vertices are selected and press the S key. This enters Scale Mode. Scale down the top vertices a bit and click to confirm.



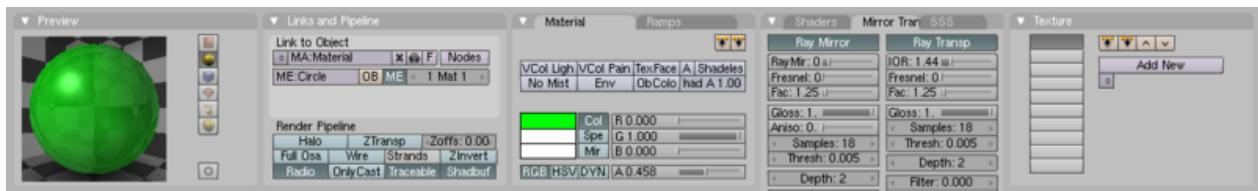
To look around your jewel model use the Middle Mouse Button or Alt-Left Click. Your jewel is looking fantastic! But there's a big gaping hole at the top. We'll fix that. First, enter side view (Numpad 3). Now extrude the topmost ring of vertices but don't move them anywhere with the mouse. If you are feeling doubtful, just press 0 and then Enter (or just click right mouse button). This makes a duplicate of rings placed over the original. Next merge this new ring with Alt-M -> At Center. This merges the duplicate ring, thus filling up your hole.

At last! Your jewel is finished! Feel free to rotate around it and make further adjustments if you are more advanced with using Blender.



## Shading

Switch back to Object Mode (TAB) and show the Shading Panel (F5). Enable the Material Buttons (click on the button with a red ball). Add new material by clicking on "Add New" button in the "Links and Pipeline" tab. Copy the settings in the screenshot below. If you are having trouble setting the precise value to a slider, just click on the number. This allows a manual type-in. First, make it green by setting R to 0, G to 1, and B to 0. Next set the alpha slider to 0.458 (it looks like an A). Now press "Ray Mirror" (raytracing reflection) to turn it on and set it to 0.13. Now press "Ray Transparency" and set the IOR value to set 1.44. When you're done it should look like this:



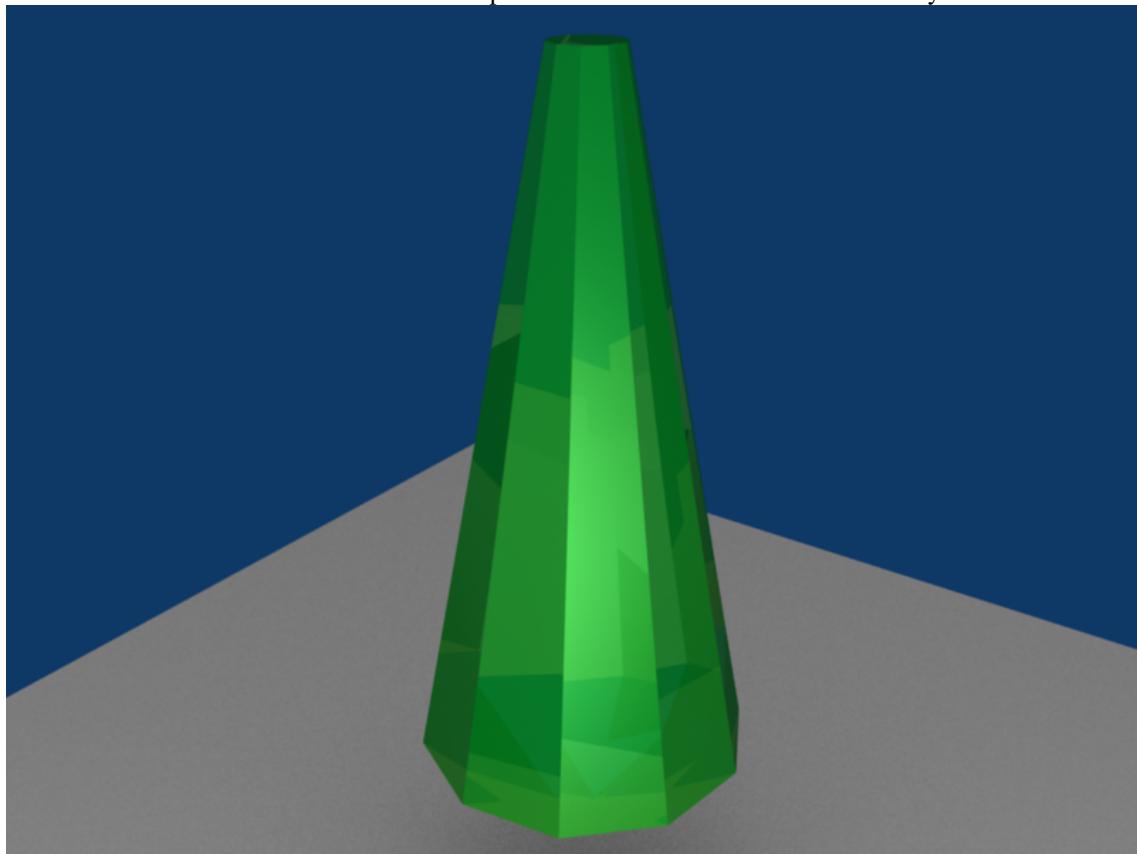
## The finishing touch

To improve the effect, add a plane underneath the jewel. In the Object Mode (TAB) press SPACE>>ADD>>MESH>>PLANE. Scale it 5 times using the S key. Then move it down a bit using the G key (press the Z key to restrict movement to the Z axis).

You may try to render your jewel now. Press the F12 key. You may find that camera doesn't see whole jewel. Move and rotate the camera (using the G and R keys) to set it in the right position. You may want to switch to "Camera View" (Numpad 0) and try out "Camera Fly Mode" (SHIFT F). Try also moving the lamp and see what happens.

I've found that the jewel looks best with Ambient Occlusion on. So go to the Shading window, then the World buttons, click the Amb Occ tab and click the Ambient Occlusion button.

Here is an example of what you can do:



## External links

<http://www.blender.org>

<http://www.blenderartists.org>

# Blender 3D: Noob to Pro/Modeling Realistically

---

Everyone who starts a new career or hobby in 3D Modeling also seems to look around their immediate environment for things to model. Things like your computer monitor, your keyboard, and even the desktop which they are resting are all common "first models." Developing an eye for geometry takes time and this is a great chance to practice.

For this tips and techniques discussion, I'll start with the concepts of furniture and walls, but the ideas really extend to anything and everything you can model in Blender, from spaceships to fantasy demon-women, from your barbecue deck blueprints to biologically accurate squids.

## Assemble Things, Don't Chisel Them

Let's start with the furniture around you.

You may think it's easy to model your desk using a couple of boolean operations on cubes. But it rarely comes out looking quite right. It seems like such a good idea at first. Then you find that the drawers won't open realistically. The corners are all amazingly sharp and crisp. The writing surface seems to be at the wrong scale for thickness. The woodgrain goes in the same direction everywhere. You then realize it will be hard to bevel just the few edges that need beveling. The more you look at it, even with good render settings, the more it looks like dollhouse furniture instead of a real life, full size desk.

Real furniture is not chiseled out of a half-ton cube of mahogany, because trees don't grow that thick, and it's a real waste to chip out the spaces for drawers. Instead, cabinetmakers tend to assemble a piece of furniture from many component parts. Think of those "self-assembly" kits from the local superstore, or even the showroom examples from a nearby unfinished knotty pine store.

Model several kinds of boards and fittings, then copy and assemble them appropriately. You don't need to model every nail and dowel, and you don't even need to know what a dovetail joint is. But you can make realistic furniture when you build things from their natural components.

- Cut lumber in a variety of sizes and thicknesses.
- Measure real example furniture and take note of the construction details.
- Choose varying woodgrains by duplicating materials and adjusting them to be more individual.
- Align those woodgrains or other materials in an appropriate direction and scale for the model.
- Bevel just the parts or edges which are naturally worn or intentionally beveled.
- Any cloth, glass or metal fittings are separate components using their own materials.
- Create an "Empty" type object, give it a descriptive name, and parent all of the pieces to it.

There are a couple more benefits to this form of component modeling:

- Working with things made from multiple real materials is easier: one object, one material.
- The model is more flexible, because they can be reconfigured to other styles more easily.
- Components come in handy for other models: adapt a desk's drawers in kitchen cabinets and bedroom furniture.

## Separate Objects for each Material

A common question for new modelers is how to make one mesh that uses more than one material. For example, a skateboard with metal trucks, rubber tires, and a wooden deck. While Blender does provide an interface to allow such a combination, it is not usually the best approach for modeling realistically.

If you model the skateboard as separate parts, based on the way different real parts are assembled, then you have a lot more control over the final model. You can replace the wheels or use trucks of different styles later. You can adjust the mesh that defines the shape of the deck, without having to re-assign the areas that should be one material or the other. And lastly, you can choose to use subsurface tricks on the various parts, while avoiding them on other parts. Even if the whole model needs subsurfaces, it becomes a bit hard to control where one material starts, and another material ends.



skateboards, modeled in parts

## Removing the Fourth Wall

Just as with furniture, you may model your first building interior as a hollowed-out cube. It seems straightforward to make a big cube, turn it inside out, and call it a living room. There are a lot of drawbacks to working with a room that was carved out of a single chunk of solid rock.

Buildings and rooms are assembled from parts. Just like the furniture modeling tip, design your rooms as a collection of separate pieces.

- You don't need to measure out the internal beams and joists, but remember they're there.
- Each interior or exterior wall should be its own object or assembly.
- Take care to note the correct dimensions and placement of doorways, windows, ceiling.

The benefits of a component-modeled interior include:

- Easier to see inside, just hide some walls on a different layer.
- Walls are much easier to move around to reconfigure the room layout.
- The same modeling work on one area can be copied to other rooms more easily.
- A more realistic architecture with consistent scaling is achieved quickly.

## Proportions and Measurements

For realism, it's vitally important to get the right proportions for familiar objects. If your audience sees a bookshelf with 5" thick shelves, or a human hand which has a 5" long thumb, their brain is going to tell them that something is wrong. They might not be able to decide why, but they'll tell you it looks fake.

Even when you're animating fantasy characters, you should develop an understanding for proportion. Many cartoon characters have very large heads when compared to their bodies; this allows room for greater expression and makes the characters seem younger. Deciding how many "heads high" each character should appear will ensure that your figures all maintain proper proportions to each other and to their environment.

To get good proportions, you need to think about measurements.

- Grab a ruler or find some existing blueprints.
- Think in terms of those measurements for the duration of the project.
- Choose a base scale that will let you work comfortably within a 500x500x500 unit volume.
- Blender limits cameras to 1000 unit distances, for better numerical accuracy.

You can mix and match base units, by using Empty objects as parents. For example, you can model your furniture in inches or centimeters, and model your house in feet or meters. Then when you append your bed.blend's objects, just parent that bed to a new Empty object, and scale the Empty. This will conveniently scale all of the separate component pieces of the object together.

There's only one big wrinkle with working with multiple scales: materials should be designed with a certain base scale in mind. Many material surface properties are scaled according to the units you choose for a given object, or the world units. Just consider this as you design, and check your work critically.



# Blender 3D: Noob to Pro/Modeling a picture

## 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 <sup>[1]</sup>
- Now 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 <sup>[2]</sup>
- 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, in **Curve Tools**, find and click the **Poly** button. 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 <sup>[3]</sup>. 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 across the picture or you'll have a messed up outline).

(user note: hitting **CTRL-LMB** instead of **SHIFT-DKEY** will add a vertex that is already connected.)

- 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 between 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 deselect 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 **CKEY**, 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 <sup>[4]</sup>. 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 instead of a curve. Add a primitive mesh and delete all the vertices in edit mode, then ctrl click to all point outline. Add depth to the surface in a side view (split views so you can see what you're moving). It helps to have 2 or more reference images, but you can

wing it. Usually the final result has to be subsurfed.

(USER EDIT: I accidentally 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 while in Object Mode. Note that this is NOT reversible.)

(user edit: You can delete one vertex of a plane, in order to get a line. You may find easier to outline the picture extruding and moving points of the line you created.)

(user edit: you can use this tracing technique to make solid and symmetric models, else, you would really have to use normal modelling)

## Printing a Rendered Image

Render your image. Exit or minimize the "Blender:Render" window. In blender, go to File -> Save Image... Then save your image. Then you can print it as you would print any other picture, using The Gimp, Paint, Microsoft's Photo Editor, or many others.

## References

- [1] [http://www.masterpiecepumpkins.com/Graphics/Devil2%20\(1.5\).PM.JPG](http://www.masterpiecepumpkins.com/Graphics/Devil2%20(1.5).PM.JPG)
- [2] [http://k.domaindlx.com/genis/images/bg\\_select.jpg](http://k.domaindlx.com/genis/images/bg_select.jpg)
- [3] <http://k.domaindlx.com/genis/images/points.jpg>
- [4] <http://k.domaindlx.com/genis/images/make3D.jpg>

# Blender 3D: Noob to Pro/Creating Ogg-Theora movies using Blender

---

Wikimedia Commons requires that movies be uploaded as Ogg-Theora (OGG) files. As of Blender 2.42a, this is not a builtin feature of Blender. To get OGG files from your finished animation isn't difficult, though. However, you'll need additional software.

There are basically two ways to generate OGG files: you can use one of the many fine video editors or you can use special conversion programs. Video editors like LiVES or Cinerella allow you to load your AVI or your rendered frames, manipulate them, and create the OGG file from it. Please refer to the editor's documentation on how to achieve this.

A disadvantage of a video editor is they are huge pieces of software, duplicating functionality that you already used when you created your animation file/s with blender. It's actually not necessary to install a video editor just for converting your animation to OGG Theora format.

## Converting AVIs directly to Ogg Theora

To convert AVI to OGG, you can use the `ffmpeg2theora` software package<sup>[1]</sup> which is available in source or binary for all relevant systems. Usage example:

```
ffmpeg2theora --optimize my.avi
```

## Converting saved frame picture files to Ogg Theora

It's actually possible to convert frame pictures that you saved before to OGG format movies. All you need to do is create a movie format that the abovementioned `ffmpeg2theora` tool understands, and then use that. On Linux, this is readily achieved by using the `mjpegtools` software package<sup>[2]</sup>. For example, if your frames were saved as PNG, you could convert them to a soundless OGG file with:

```
png2yuv -I p -f 25 /tmp/*.png >my.yuv
ffmpeg2theora --optimize my.yuv
```

Sound is possible too. Please refer to the *mjpegtools* documentation.

## References

[1] <http://www.v2v.cc/~j/ffmpeg2theora/>

[2] <http://mjpeg.sourceforge.net/>

# Blender 3D: Noob to Pro/Creating animated GIFs using Blender and Gimp

This tutorial will guide you through how to make a simple animated Gif Using Blender and Gimp. This is useful for creating Avatars for forums etc. This tutorial assumes basic knowledge of blender and Gimp, see the basic animation tutorial.

To start off you will need an animation, this usually should be no longer than 25 frames long.

1. Open Blender. and delete the default cube and add a UV sphere, the default settings for the sphere work fine.
2. Now set the camera size in the scene buttons (F10) to 50 by 50 (I am creating an avatar for deviant art where the required size is 50 by 50 pixels, you may change this if you want)
3. set the frames to start at frame 1 and end at frame 20.
4. select the sphere and insert a LOC keyframe at frame 1 and 21, then go to frame 11, move the sphere and insert another LOC keyframe. this will create a looping animation.
5. Set the image type to PNG or JPEG (it doesn't matter) and render the animation.

Now to combine the images into an animated GIF using Gimp.

1. Open the first image with gimp.
2. Now click File-> open as Layer or press "Ctrl-Alt-O". Select the next frame and it will be added as a new layer. Repeat this for all of the images, or select all of the images by pressing "Ctrl-A".
3. If you press Filters->Animation->Playback it should play the animation. It will probably have a low frame rate making it "choppy". this will be fixed in the next step.
4. Change the frame rate to 40 ms(25 frames per second). Choose file->Save as-> "Name".gif, then choose Save as Animation.



You should end up with something like this:

# Blender 3D: Noob to Pro/3D Tiling Backgrounds For The Web

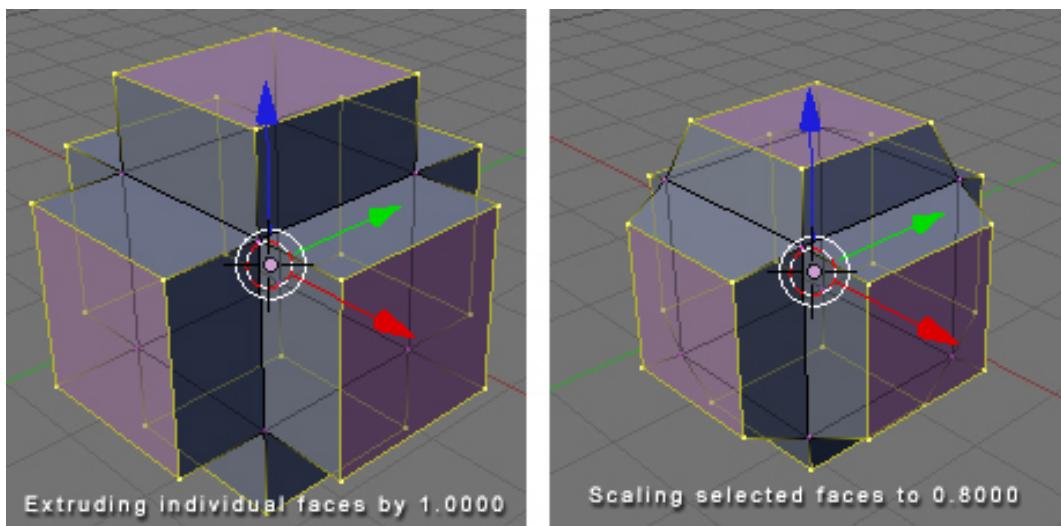
## Overview

This tutorial will guide you through the process of making 3D tiling backgrounds for use with web pages, your desktop, or anything else for that matter. We will be using Blender and a graphic editing program such as Adobe Photoshop or GIMP. I will provide detailed explanation to cater to beginners, however more experienced Blenderists can probably get away with just following the picture diagrams provided. You may also find the diagrams useful if English is not your native language, or if you hate reading instructions.

## Create The Object You Wish To Tile

Start up Blender and look at your lonely cube. We will use this cube as a starting point for the sake of demonstration, but feel free to use any shape you like. Press "Tab" to enter **Edit Mode**. Press the "E" key on your keyboard and the **Extrude** menu will appear. Select "**Individual Faces**". As you then drag your mouse cursor you will see some numbers move in the bottom left in the window and text saying "Shrink/Fatten". I recommend setting it to 1.0000 to keep it simple. Hold down the "Ctrl" key while dragging to do so in set increments.

Next, with the six faces of the cube still selected, press the "S" key to scale those faces. I recommend scaling to 0.8000.

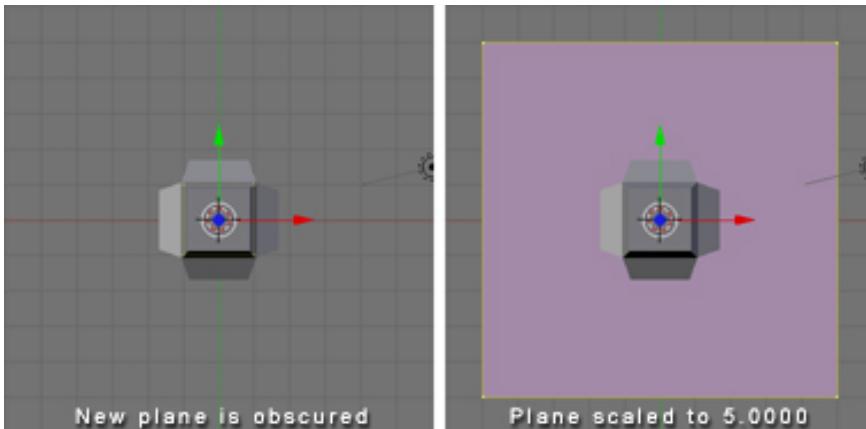


Good. Now that we have created our object we will prepare to tile it.

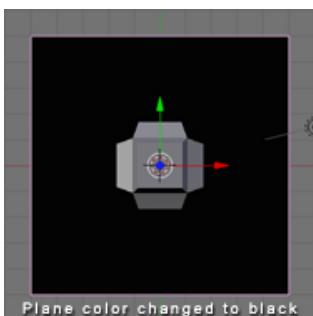
## Specify Your Tiling Area

"Tab" back into **Object Mode** and press "NumPad 7" to go to the Top View, or manually click "**View**" at the bottom of the window and select "**Top**".

Press "**spacebar**" and from the menu that appears select "**Add**" > "**Mesh**" > "**Plane**". The plane has appeared, but is obstructed by our cube object, so press "**S**" to scale the plane to 5.0000.



You'll notice that I've colored my plane black. This is only to make it easier for you to see. You need not bother with this as we will be deleting the face of this plane shortly.



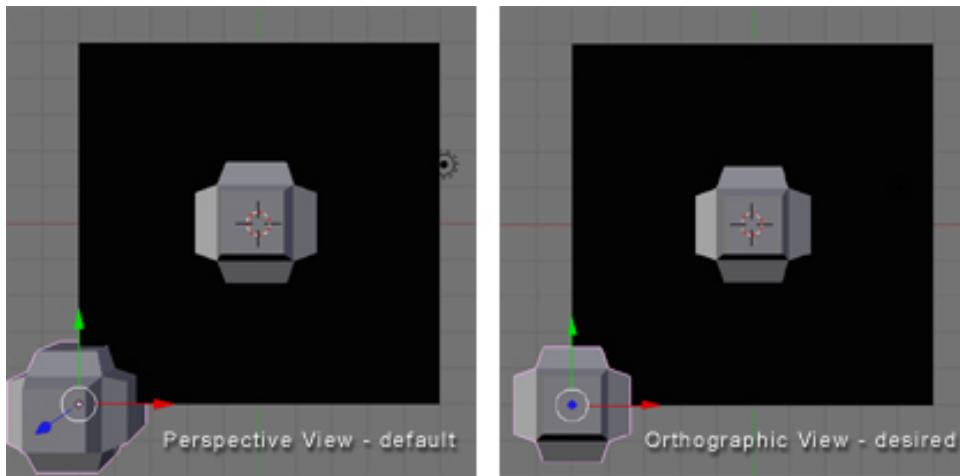
We now have an easily visible boundary representing the area which will be tiled. If at any point we decide we no longer need or want this we can move it to another layer by pressing "**M**" and selecting a layer to move it to.

## Tile Away!

Now the fun part begins. If you have created 2D tiling backgrounds before, this concept will be very familiar to you.

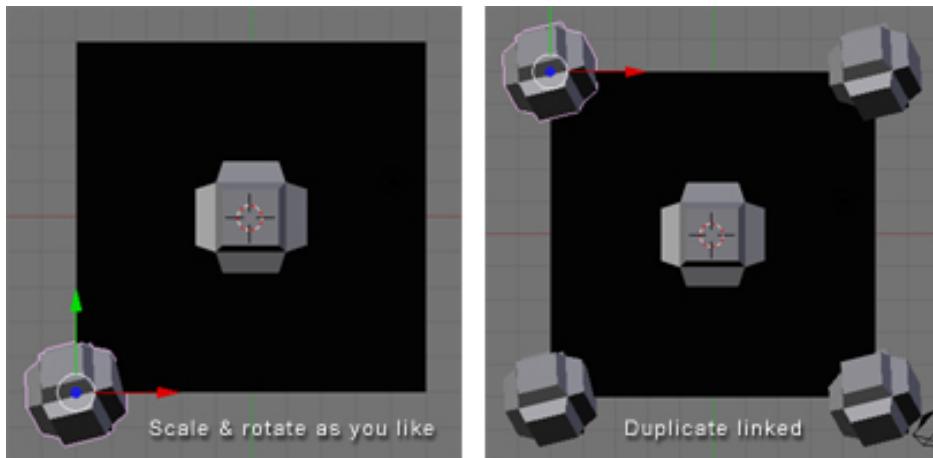
While in **Object Mode**, select your cube object. At this point you have a couple of options. One is to make duplicates of the cube object. Another is to make *linked* duplicates of the cube object. I recommend making linked duplicates because you can later edit the mesh of one object in order to change the mesh of all the duplicated objects. This can be especially useful if you later plan to animate your tiling background (*careful not to distract from the foreground*). It is also useful if you want to experiment with new designs easily without having to re-place all your objects.

With your cube object still selected, press "**Alt+D**" to make a linked duplicate. (*Non-linked duplication is done with "Shift+D"*) Now hold down "**Ctrl**" to move in set increments and move the object to one of the corners of our plane. Make sure the center of the object (represented by a pink dot) is aligned with the very corner of the plane. At this point you may notice it's somewhat difficult to tell whether the center of the object is exactly on the corner of the plane, and that's why we are going to press "**NumPad 5**" or click "**View**" and select "**Orthographic**". For all practical purposes we can spend the rest of the time we are building the tiling pattern in Orthographic View because this view is essential for tiling.

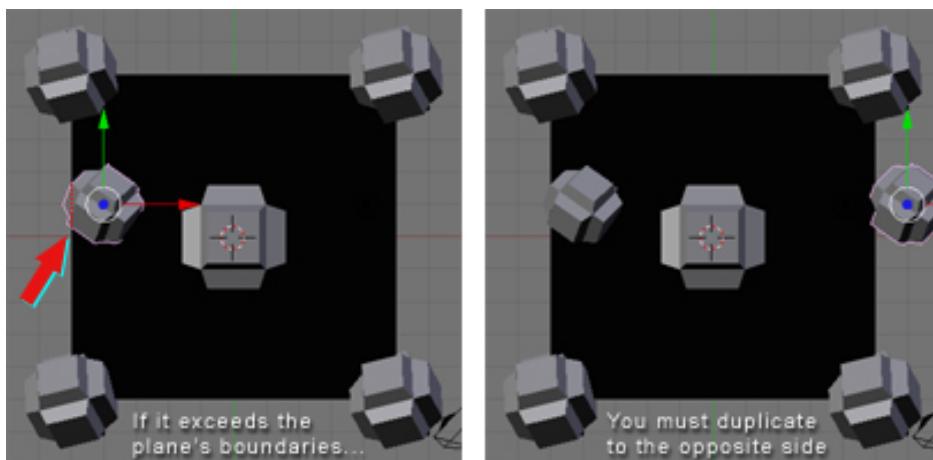


Now with your duplicate object still selected, scale it ("S" key) so that is somewhat smaller than the first. Then rotate it using the "R" key. While rotating you can constrain to a particular axis by typing "X", "Y", or "Z" respectively.

We will now make linked duplicates (**Alt+D**) of our rotated and scaled cube object and place one in each corner exactly on the grid (remember to hold "Ctrl" while dragging). If you place an object in one corner, you must place it in all four corners because corners touch both the X axis and the Y axis.

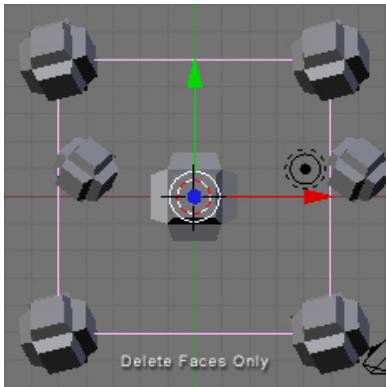


Now as you will notice in the following diagram, I made an object that is crossing over the edge of our plane. This is good. This is how to make natural tiling backgrounds. However! Whenever we do this we must be absolutely certain to do the same on the opposite side, or else our backgrounds will not tile properly.

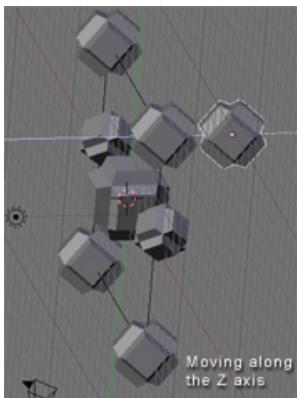


So now we've created a few objects of different sizes and orientations, however they have all been along the same spot on the **Z** axis. So let's give our pattern some depth - after all, that is the joy of working in 3D!

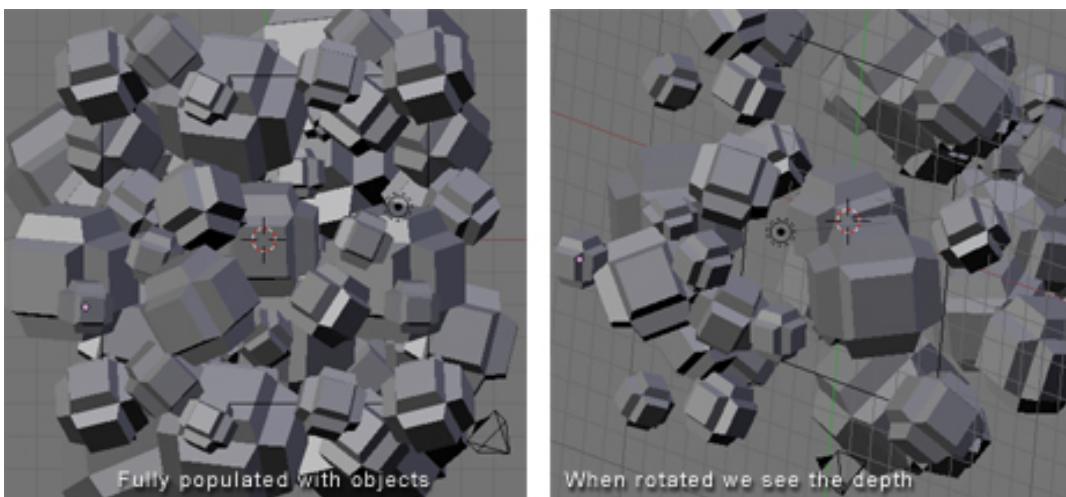
Before we do this, we'll want to get our guide plane out of the way. So **right click** on the plane to select it, then **"Tab"** into **Edit Mode**. Press **"X"** and from the **Erase** menu choose **"Only Faces"**. Now **"Tab"** back into **Object Mode**. There! We now have only the segments of the plane as our guide.



Hold down the **middle mouse button** while dragging the mouse to see what our design looks like in three dimensions. Now right click one of the cube objects, duplicate a linked copy (**Alt+D**), and press **"Z"** to constrain movement to the **Z** axis.



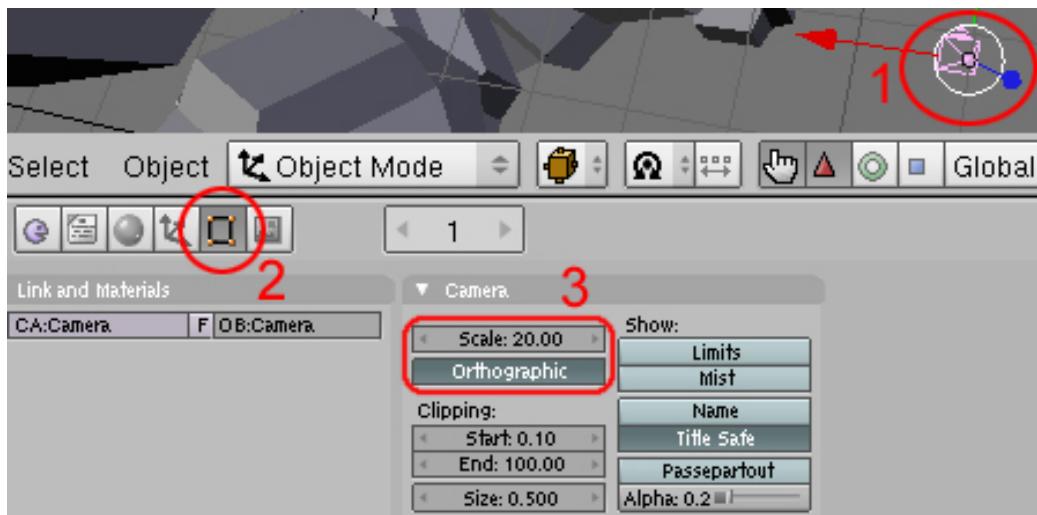
Navigate in this way along the **X** and **Y** axis as well and once you've found a good spot, **scale** and **rotate**. Remember to make a duplicate on the opposite side whenever you cross the outline of the plane guide. You can go on like this and populate your tiling pattern with as many objects as you wish. The grid is your friend during this process, so you should always have your finger on **"Ctrl"** while moving the duplicate of an object that crosses a border of the guide plane. Also, any object that will "tile" across the border must be at the same point on the **Z** axis as its counterpart on the other side of the seam. Press **"NumPad 7"** periodically to see where you are from a 2D standpoint. This is the view from which we will eventually render the image, so this perspective is the one that counts. Beginners, remember that **mouse wheel** zooms in and out, and **"Shift+middle mouse button"** allows you to "drag" your way around.



Keep in mind that in **Orthographic View** we do not perceive the depth of objects. Those far away and those near by appear to be at the same distance. This is what allows us to make a tiled image, but it also limits the apparent depth of the scene. We can compensate for this by scaling down the objects we want to appear further away. Or, as in this example, we can just make each object a different size and place some in front of others.

## Camera Settings

We will now position the camera directly over our cluster of objects. Press "**NumPad 7**", center the view on the cluster using "**Shift+middle mouse button**", and zoom out a couple notches with the **mouse wheel**. From the **3D View Window**'s menu, click "**View**" > "**Align View**" > "**Align Active Camera to View**". Now you may be wondering why everything appears to be distorted. This is because when we changed to the camera's perspective, the view automatically reverted to **Perspective View** because by default the camera is set to that view. But just as we changed to **Orthographic View** in the 3D View Window, we can change the view of the camera as well. While in **Object Mode**, swivel the view until you see the camera. It is represented by a pyramidal wireframe with a black triangle atop the opening. **Right click** to select it. Now press "**F9**" or click the **Editing** button (it's icon is four vertices joined in a square). Now in the **Camera** panel you will see a button labelled "**Orthographic**". Press it. Above the button is a value labelled **Scale**. Set it to around 20. (See figure below)



## Lighting & Materials

Place some lights in your scene, and add some materials to your objects. **Lighting** and **materials** are subjects which demand their own tutorials, so if you don't know how to do these things yet, please consult the Wiki.

**Note:** I advise lighting your scene pretty evenly in order to make the tiling seem contiguous.

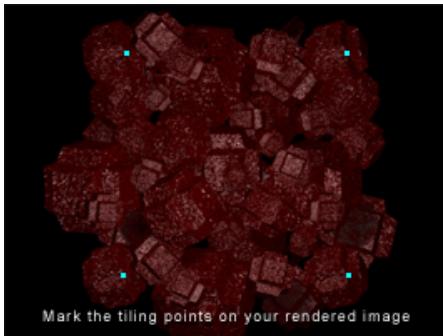
**Note:** You could also only use Sun lamp, so the light is clean all over the mesh!

## Rendering

Now that you've presumably got your lighting and materials as you want them, it's time to render the scene. Under **User Preferences** click "**Render**" > "**Render Settings**" or just press "**F10**". In the **Format** tab you can choose the dimensions of your rendered image, the file format you prefer, and the quality. I'm going with 800x600 at 100% quality.

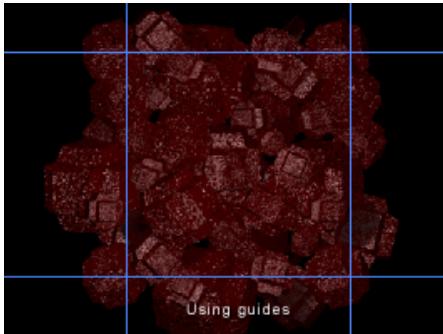
## 'Shopping

And it's time to begin post-production. Open your rendered image in your preferred image editor. I will be using Photoshop in this example. Now find and mark the four corners of your tile (on a new layer of course). Just eyeball it.



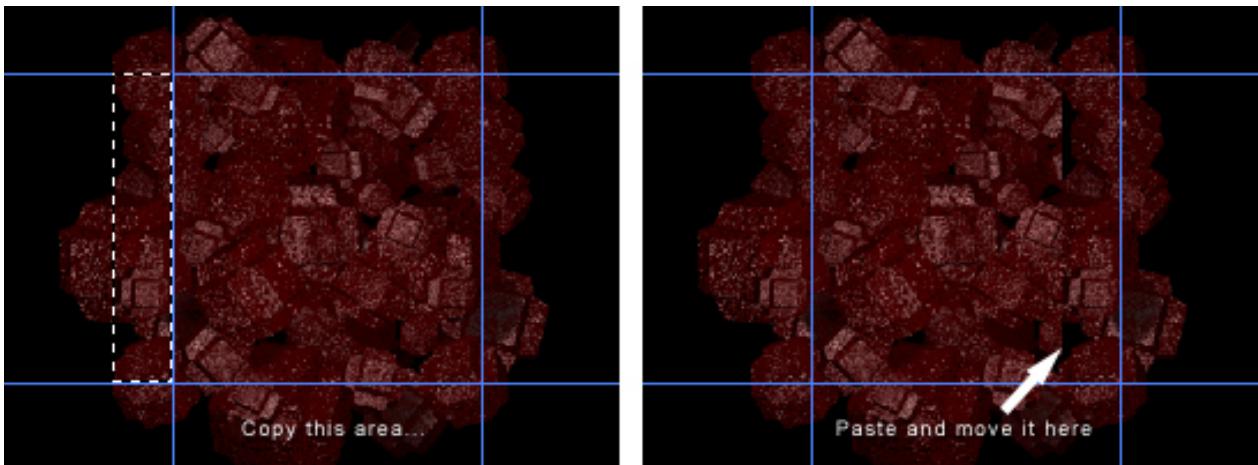
We will use guides to mark the tile boundaries. In Photoshop guides can be inserted by selecting "**View**" > "**New Guide...**". Guides can be positioned with the **Move Tool**. You can zoom in and make sure your guides are accurately positioned by holding "**Ctrl**" while typing "+" or "-". You can now hide or delete the layer with the markers, as that was only to help us get accurate guides.

Should you decide to make an animated tiling background I recommend layering all your frames on top of one another in your image editor before editing in order to keep the position uniform. You can then animate them using Adobe ImageReady (comes with Photoshop) or find decent low-budget and occasionally free GIF animation programs online.

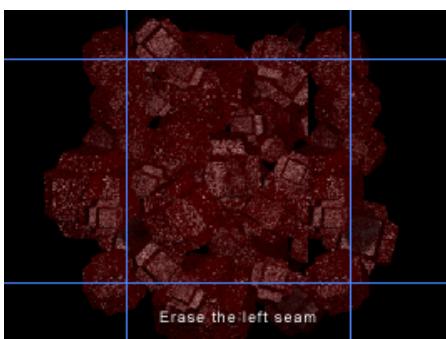


Now select an area just *outside* of the guides with the **Marquee Tool**. Make sure "Feather" is set to **0px** and that the marquee is rectangular and **Style** is set to **Normal** (meaning no Fixed Aspect Ratio or Fixed Size). Also, it will help you to have **Snap** enabled, so go to "**View**" and make sure it is enabled for guides.

Once you have selected the area, copy (**Ctrl+C**) and paste (**Ctrl+V**). Then drag to the right using the **Move Tool** and that piece should snap to the inside of the rightmost vertical guide.



Select the Eraser Tool and choose a brush size. I'm going with a diameter of 65. You'll want to use a soft gradient-like brush for this. Now erase the left edge of the newly pasted selection so that it blends into the picture. Don't erase any of the right edge.



Press "Ctrl+E" or alternatively, select "Layer" > "Merge Down".

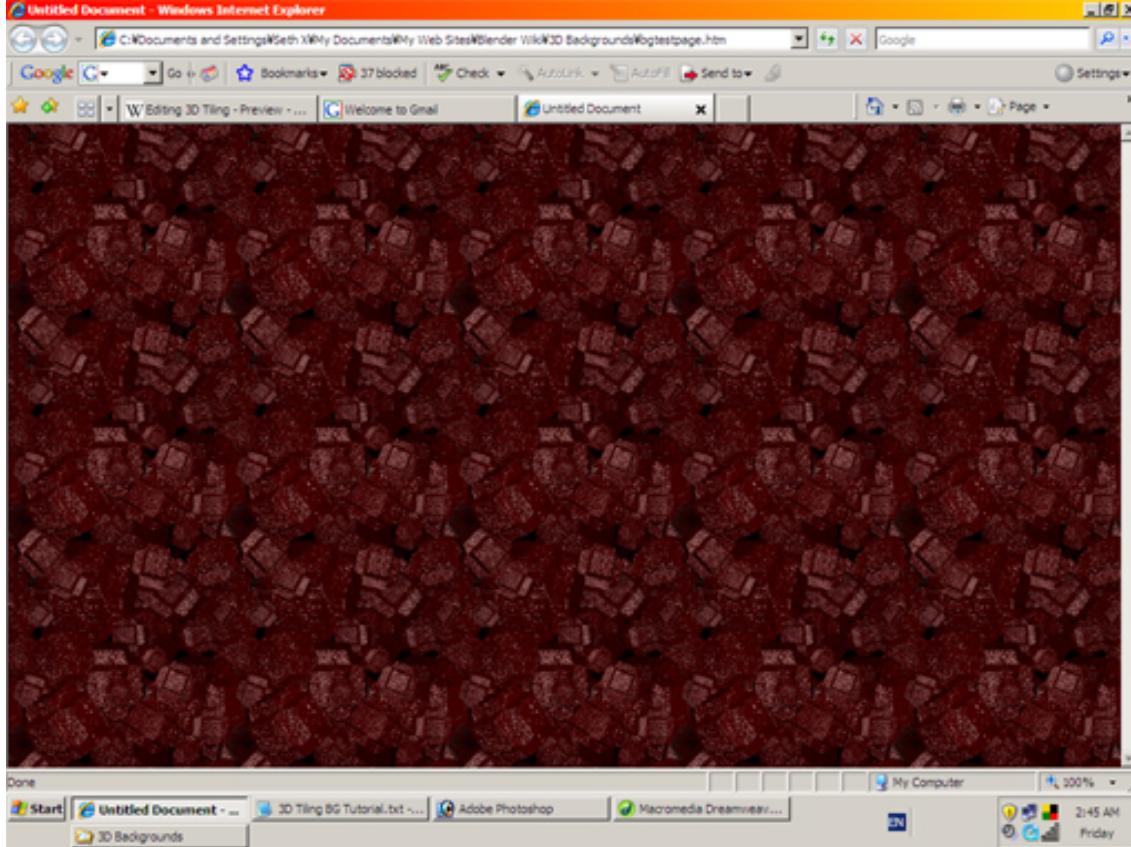
Alright, so that procedure you just performed copying from left to right - now do it bottom to top. Once again, erase the edge. Do a **Save As**.

Finally, select the square center area that will become your tile and do "Image" > "Crop".

Now scale your new tiling background to a web-friendly size. **Save As** a JPEG, GIF, or PNG. **You're done!**

*Note: If your pattern still isn't tiling quite perfectly, load the version with the guides again and repeat the copy-paste-erase process. Remember to merge down your layers each time or the corners may not tile correctly. If still no joy, try tiling the background in a web browser, taking a screenshot, pasting that into your graphics editor, and touching up from there.*

Here is the final result:



## Blender 3D: Noob to Pro/Cool Things

---

Cool things in blender that aren't that obvious. Useful tips and tricks in Blender.

### Attribution

Many of The Following Tips and Tricks have been contributed by members of the CGsociety.org<sup>[1]</sup>, a Public Society for Digital Artists. The tips have been extracted from a CGsociety thread<sup>[2]</sup>. They have been roughly edited to improve readability.

### File Browser Functions

#### Delete, Move, Rename, and Make Directory

When you are inside a file browser for loading or saving something and you want to create a new directory, just add the name to the path on top of the window and confirm 'Makedir'.

you can also delete(x), move(m), or rename(n) a file. you can do action on multiple files by selecting with right click

## Preview images when loading them as a texture

Whenever you are loading images as a texture, you can hold down **Ctrl** while clicking on the 'load' or 'open' button and the original preview window will appear. But be careful. The preview does not like certain file types and may crash. That is the reason why it is hidden .

## Trick for Creating Quick File Revisions

After you have saved a blend file or an image you can then save it in progression that is:

car.blend. or car.JPG

Next time do "Save As" then press the + (plus) key which will advance the blend file by 1 every time it's pressed. example: car.blend becomes car.1blend. press again car.2blend and so on.

The - (minus) key will subtract one. I've gotten into the habit of saving frequently. Yea I know there is now the undo feature but I like this better because it gives you a history in case you need to back a few levels of a build. You get a saved version at the level you choose.

Note: Blender automatically detects the number... meaning it does not need to be in any position. For instance: If you have a file named 001starport.png or .blend or whatever, pressing the + (plus) key will automatically name it 002starport.png. If you want to name it starport1.png, it will change it to starport2.png.

Two rules: The filename has to have a number. It can be 0, or 1, or 3.141569.

If the file number is a negative, pressing + (plus) key will increase the "magnitude" of the negative number. I may have used magnitude wrong, if so, I mean pressing the + (plus) key will make -0.04 drop to -0.05. The - (minus) key will bring you only to 0, and then it will start to eat itself up.

## Open Recent

Control O

## Object/Vertex Manipulation

### Constrain Movement to one Axis or to a plane

when moving objects/vertices or set of objects/vertices ([G] key) if you move in the direction of the global X axis (Up/Down) and then press the **MMB**, movement will be constrained to only move in the X axis, or if you move the vertices in the direction of the Y axis and then press **MMB** it will be constrained to move only in the Y axis. The same is true of the Z axis.

You can achieve the same effect by using the X, Y or Z keys while in grab mode. You simply have to press X key, Y or Z, once to lock to X Y or Z global axis.\*

To Constrain the Movement to two axes (a plane): Press **Ctrl+X** to move in the Z-Y-Plane. **Ctrl+Z=XY**, **Ctrl+Y=XZ**.\*

Alternatively, Select scaling mode and select the axis not to scale with the selecting button at the same time as you hit **Shift**.

- in every case You can hit the X,Y or Z button again to constrain movement to a different set of axes. Normally this different set of axes is local. However you can change the identity of this set to global, local, normal, or view by pressing **Alt+Space**. which cycles through the different identities.

Note: All of these shortcuts work with scaling and Rotating as well

## Shrink/fatten mesh in direction of vertex normals

When you're mesh-editing, **Alt+S** will shrink/fatten the mesh selection in the direction of the vertex normals.

## Vertex Parenting

You can parent object to a Mesh, in that case you are parenting to the center of the mesh.. BUT if the mesh is translated somehow (let's say by an armature'[S] pose) the center remains in the same spot, and thus the child object doesn't receive any transformation at all.

To solve this, you can parent the child object to a vertex (or a face) within the mesh, and any transformation that the vertex receives is passed to the child.

There are only 2 options, to parent to any 3 vertex within the mesh or to parent to just one vertex. If you parent to 1 vertex then only location information is passed, with 3 vertex all transformations (rotation, location and size) are passed to the child.

How to do it? Starting out of edit mode select the child(ren), hold **Shift** and select the parent, enter edit mode, select one or three vertex, press **Ctrl+P**. That's it!

## Work around to welding Verts

For Edge loop (Verts) position both loops together as close as possible then hit W then 4 (not on numkey pad). You can adjust how far the effect of collapse can go in the Edit window (a button on the right labeled Limit: \*\*\*, where \* is a number).

As for individual Verts, eg. Two vertices welded to become one, select both Verts, scale until they are very close then hit W then 4.

you can also do this with the snap combo

select the vertex you want to weld together **Ctrl+S**, Cur -> Sel **Ctrl+S**, Sel -> Cur [W], Remove Doubles

Also, whenever Blender pop up a menu with different options, you can just type in a number to choose one of the options (use the numbers not on the NumPad)

## Make individual objects the camera and Change them back

If you select certain objects and press **Ctrl+0(zero)**it will make them the camera. I use it all the time to align spotlights.

Select your camera and hit **Ctrl NumPad 0** to make it the active camera again

## Ordering Meshes in Vertex Groups

If you are preparing to skin your meshes and you are ready to create the vertex groups, you should pay attention to the order in which you create them, because once they are created there is no way to re-arrange them on the vertex-group list.

That means that if you are a ultra-by-the-book person and you would like the vertex group alphabetically ordered on the list them you must create them in alphabetical order.

.. Or, if you would like them to be ordered according to their function (shoulders, then arms, then forearms, then palms, etc. you must create them in that order.

This may all sound like a stupid thing to care about, but If you have a character with 39 vertex groups you may quickly find that when one of them needs fixing it is a little difficult since they were randomly created.

## Position and scale along face normal,

-**Shift V**: Position camera along face normal, **Alt S**: scale selected vertices along face normal

## Using Fake Users

pressing **Shift+F4** will turn the window into a "Data Select Window" where you can assign and unassign fake users to almost everything by selecting the name and pressing the F key

Creating a fake user allows you to keep useful data blocks (materials, textures, base meshes) at hand even if they are not linked to an object. You can use it to set a default material that would have the shader you like best.

## Align a selection of vertices on a plane

If you want to perfectly align a selection of vertices on a plane, you just have to follow these little steps:

- 1.) Before you are selecting the vertices you want to align, position your 3D cursor in the plane that you want to align to (you could select the 4 vertices of a big plane and hit **Shift+S** / Cursor->Selection for example, but you can position it anywhere)
- 2.) Now select those vertices you want to align
- 3.) Choose "3D Cursor" under "Rotation/Scaling Pivot"
- 4.) Now with the **[S]**-key start scaling mode, hit the key of the axis you want to move the vertices on (X,Y,Z)
- 5.) Holding down the **Ctrl-Key**, you can now move the vertices in one line towards the cursor, until the value for the chosen axis is 0.000. Alternatively, just enter, using the keyboard, the scale factor you want (0 in this case).
- 6.) Hit LMB. The vertices are perfectly aligned along a plane through the 3D cursor.

This even works very well while in perspective view mode, so you can align on the fly and don't have to switch to front/side/top view all the time.

## Welding Vertices

You can weld vertices by selecting them in edit mode and pressing **Alt+M**.

## View wireframe of hidden Verts

to view the wireframe of hidden Verts, make sure you are in WIREFRAME MODE and then turn SubSurf on and change the level to 0 If you already knew about this then

## Select a true loop

shortcut is **Shift + Alt + right Button** of the mouse, serves to select true loop, in vertices as in edges like in faces.

## Selecting one object from a single mesh comprised of multiple objects

If you have more than one object in Edit Mode, you can place your mouse cursor next to one of the Verts in the desired object, then press the "L" key to select all of the Verts linked to that one. "Alt+L" reselects in the same manner.

---

## Precise Zoom and Select/Deselect

Selecting: If you Hold down the **Ctrl+LMB** (left mouse button) and drag the mouse, it will allow you draw a selection as opposed to using the B button which gives you a square.

Deselecting: To draw an area to deselect, Hold down the **Ctrl+ Shift+ LMB**(left mouse button) and drag the mouse,

Zooming: Hold down the **Ctrl+MMB**. Move your mouse vertically to can get a more controlled zoom versus scrolling the Mouse wheel.

This feature may not be present in 2.37 or earlier versions.

## Mouse Gestures

Left click and draw:

- a straight line - moves the selected object.
- a circle - rotates the selected object (note: this must be drawn fairly circle-like).
- a V - scales the selected object.

## Selecting Obscured or Hidden Objects

Say you are in front or side view and you want to select an object, but it is obscured or hidden behind other objects.

If you press **Alt RMB** over a group of objects, a menu will be displayed in the 3D window allowing you to pick the object you wish to select.

## Select or Deselect Multiple Vertices

In Edit mode, when you click the **RMB** near a vertex (face or edge)will be selected, **RMB** again will reselect. By holding the **Shift** key this will allow you to add each selected vertex (face or edge) in that highlighted group.

Pressing the U key (Undo in Edit mode) will also remove the last selection you made.

Alternatively, you can Press B and then draw a box with **MMB**. Anything caught in the box will be deselected. Also works with BB and the draw selection. Draw with **MMB** and you reselect.

## Selecting multiple items

You can use right mouse button to select multiple items to append. Doesn't seem to work with files.

## Measuring, length, distance on an object.

Hit "F9" (editing), you should have split (2) windows. One "3d" the other "buttons" go to the Mesh Tools 1 panel and press the Edge Length, Edge Angles and the dimensions will appear on your selections in the 3d view.

## Adding Connected Vertices

In Edit Mode, if only one vertex is selected, pressing "E" will add a vertex, on a freely defined place, connected to the selected one. As will holding **CTRL** and left clicking the mouse where you want the new vertex to be positioned.

Note: It must be an Image texture and in wireframe mode to be visible.

(2.37a)

## Recalculate Normals

**Ctrl + N** = Recalculate normals outside (you might have to select faces before doing so) **Shift + Ctrl + N** = Recalculate normals inside

These two hotkeys are useful when you extrude some edges and see a kind of seam in between (due to normals pointing in different directions).

Then, after selecting an edge, **Ctrl + NumPad(+)** selects the face associated with this edge. **Ctrl + NumPad(-)** deselects the face.

## Create a Quad from two Tris

**Alt + J** when having two Tris selected makes a Quad.

## Remove Doubles

To Remove Doubles use hotkey: **W**. You can adjust the "limit" option so that "Remove Doubles" has more or less tolerance. This option is located in editing window under the mesh tools panel, (i.e. weld vertices that are further?)

## Combine edit levels on a mesh.

When in Edit mode for a mesh (TAB key) you can choose the level that you wish to edit at. At the bottom of the 3D window, there are four buttons. Vertex, edge, face & back-face cull.

By default the vertex level is selected, if you hold **Shift** and press the edge button, you can use both at once.

## Change Select Mode

to change select mode (vertex, edge or face) you can press **Ctrl+Tab**. But this way you can't use the Ctrl+Key to ADD the select modes.. Still could be useful, if you don't have a header for the window you're working in..

## Precision Warping

When using the warp tool (**Shift+W**) you may find that there are times when you have trouble perfectly closing a 360-degree loop. This is because Blender will warp based on the total width of the selection, which may not necessarily be what you want.

Getting around this is simple, just select two verts to denote the new endpoints, duplicate them (**Shift+D**), scale the two verts times two, so they are just twice as far apart. Then select what you want to warp plus those two marker verts, and warp 720 degrees.

When done, click a vert on your mesh and type **L** to select everything linked to it. Then type **H** to hide it. Once you've hidden all of your mesh, all that remains are the two marker verts. Do a Select All (**A**) and type **X** to delete them. Now unhide your mesh using **Alt+H** and you have just the warped mesh, extra guide verts are all cleaned up.

## Precision Cutting

The Knife Tool can actually be quite precise if you take advantage of the snap feature. Press **K**, choose "**Knife(Exact)**" and then hold the **Ctrl** key while choosing where you wish to cut and you will find that the path to be cut will now snap to nearby vertices.

Keep in mind that the vertices being snapped to don't have to be the ones you are cutting. Say you want to cut the midsection out of a UV sphere and you want the cut to be two grid units in height. While viewing the side of the sphere, add a plane, and scale it to be two grid units tall, and wide enough so that it extends beyond the sphere. Now select the vertices of the sphere (because what we have selected is what gets cut) and when you cut it, snap to the four verts of the plane. Now hit **Enter** to apply your perfect cut.

Using guide geometry such as a plane to cut other geometry "cookie-cutter" style can be extremely useful when accuracy is needed. Remember to align your view before cutting, since your view will determine the angle from which the geometry is cut.

## Working with Meshes

### Turning Sub-Surfed Mesh into Normal Mesh

If you have a sub-surfed mesh you can turn THAT sub-surf mesh into a normal mesh. Just select the sub-surfed mesh and press **Alt+C** (Conversion).

### How to remove (numb) black spots on a Mesh

If a Sub-Surfed mesh becomes (numb) black on some places, that's because of the normals. Select all and press **Ctrl+n** and then confirm. now it should look prettier!

if the above solution does not work , save your Blend file, Quit Blender then restart. Use **Ctrl O** to open the last file and your mesh will have returned to correct shaded state.

### Select all holes in a mesh and fill them

**Shift+M\*** selects all Non-manifold edges/vertices (holes) in a mesh \*\*. Then all you have to do is hit **Ctrl+F** to auto-fill those holes with "beauty fill".

**Shift+M** is an alternative shortcut for 'Select Non-manifold'. You'll find this in the select menu when in mesh edit mode. The listed short cut there is **Ctrl+Alt+Shift+M**. **Shift M** is obviously a lot more comfortable on the fingers though! Although 'Non-manifold' usually refers to holes in your mesh, it is not necessarily only holes eg, an edge with three faces coming out of it is also a non-manifold edge!

### Fill in four or fewer vertices

select the vertexes and press **[F]**, this will fill in the empty space around them

to clean up a filled in space, select all the vertexes for the area, and press **[F]**. choose OK to make FCon For example: add a plane in wireframe mode, extrude it several times, select all vertexes, then hit **[F]**.

## Animation

### Animation Preview in all windows at the same time

It is well known that **Alt+ a** is for previewing an animation on the 3D window. But that's not all of it. Divide your screen into multiple 3D Windows, each from a different point of view.

Press **Alt + Shift + a**

Enjoy!!!

If you have an Action/Ipo Window and 3D windows open, and you issue the **Alt + Shift + a** command from the Action (or the Ipo) window, it will animate both (the action and the 3D) in sync!! Great for visualization of Ipo's effect on your model.

## Choose animation mode, Convert mouse movements to IPO

-T in IPO window: Choose animation mode, ie., linear, bezier, constant -[R] in IPO window: Record mouse movements, and convert to IPO

## Animate procedural textures

To animate procedural textures, press 'i' with the mouse pointer in the materials window, and select the type of Key-Frame you wish to set in the pop up menu. advance a few frames, tweak your materials, and set another key frame.

## UV Mapping, Particles and Texturing

### Blender color picker

Blender has a PhotoShop-esque color picker. Simply click on the color preview next to the sliders to use it. Hit enter when you have the color you want.

### Saving your face groups selections

Regarding UV mapping and Face Groups Selections there seems to be the general misconception that you can't save your face groups selections on Blender.

Most people already know that from within the Face Select Mode (Potato Mode) you can switch into Edit Mode and whatever selection you do while in Edit Mode is passed back to Face Select Mode when you exit the Edit Mode.

Well, Did you ever wonder why Material Index Groups (that are nothing more than face groups with a common material on them) have those little 'Select' and 'Deselect' buttons there? Sure they come handy for later modification of the material index but that is not all about them.

Do this: Before starting the UV unwrapping job, cut your mesh by creating as many material indexes as you need, you can even assign each one a different color so you can be sure that there is no face orphan. Once you have the mesh all cut and sliced (so to speak) you enter in Face Select Mode, then switch into Edit Mode, select the index containing the faces you want to unwrap, press 'Select', leave Edit Mode and Voile! You now have an entirely useful face group waiting for you to unwrap. No more manual (and imprecise) face selection is needed.

If you later need to change the mapping of those faces, don't fear. Just make sure there isn't any face selected on Potato Mode, do as you did first (enter edit mode, select the index, exit edit mode) and there are your very same faces selected again with the UV mapping you already assigned to them.

Note: Another benefit of have precise face selection groups is that, initially, you don't have to worry about UV coordinates overlapping, since you know have the way to select ONLY the faces you want to. For example, you unwrap all your faces by groups and when you are done you can start thinking about scale and position withing the texture map, not like before when you have to solve those things as you go.

## Bulk Texture Change

Consider a scenario in which you have a scene with a 100 mesh objects, and 50 of them have one texture and 50 of them have another.

If you want to change the texture of the first 50, but don't want to change each individually, do the following. Add a Plane out of the view of the camera. Add your new texture (Material) to the plane. Then use the "Copy to material Buffer" button in the header of the Material buttons. Select one mesh object of the same sort that you want to change, open Material buttons and Paste from Material Buffer.

All the mesh objects with the same texture will now have the new texture.

Alternatively, If you have a material that you want to apply to a lot of objects at once:

1. Select all the objects you want to apply the material to.
2. Apply a material. (this only applies to the last selected object)
3. press **Ctrl+L > Materials**. (this links the material of the last selected object to all the other objects)

## Negative Meta-Balls

Add>Metaball as usual. Exit EDIT mode and Add>Metaball. This time before you exit EDIT mode, hit the Negative button in the EDIT buttons window. Then leave edit mode.

If you move the negative Metaball around, you can see the effect it has on the positive metaball.

Be careful though, as negative metaballs are not displayed in the same manner as positive metaballs, you will only see the Pivot point not a meta-mesh.

This is a little test you can try to see the amazing effects negative metaballs can produce.

Make 1 Metaball, make it big. Place three spheres (UV) inside, make them emit particles, one 100, one 200, one 300 particles. Parent three negative metaballs, one to each sphere, and use dupli-vert on each sphere. Make the 100 duplicate metaball quite big, the 200 medium and the 300 small. Hit **Alt A** to run animation in 3D window.

(*Click here for this author's negative-metaball thread... oops, it's not there anymore, well, not the AVI anyway!*)<sup>[3]</sup>

## Maintain the UV layout when moving/scaling/rotating UV co-ords.

When you have the UV image/editor window open and have loaded an image you want to UV map to a mesh, click on the UV tab in the header bar and turn off 'Snap UV to pixels'.

This will help to maintain the UV layout when moving/scaling/rotating UV co-ords.

## Rendering

### Tricks, related to the view ports and the render buffers

First. Switching among screens

So you have your screen made off the 3D window, the buttons window and the info window... but you are doing some fine tuning to the mesh in two places simultaneously, and they both need to zoom in the 3D window. You could scroll or zoom out, translate the view and zoom in again. None of them an elegant solution.

Another situation. You are working on a model and are using an image for reference. You are not tracing over the photo, just take a look at it often to make sure you don't deviate to much from the concept. So you open the photo in a 2D program and keep switching back and forth from Blender.. or you have the photo open in an image window and keep maximizing and minimizing the window... another hassle

Worry no more!!! Blender can handle multiple virtual screen (ala Linux) and you can come and go from them with just one key stroke.

Just press **Ctrl+Left Arrow** or **Ctrl+Right Arrow** (for all of you OS X users, add a **Shift**) and you are switching screens. Go ahead! By default EVERY .blend file comes with 3 screens ... and of course you can add/delete as many as you see fit.

the magic button to add or delete screens is right beside the Tools menu, up there in the info window.

## Using the render buffers

Ok, so you set your scene and press RENDER, a nice window comes up and you see your hard work coming to existence (that's the default behavior, if you change it on the display buttons then this may not work for you).

Do you realize that the window containing your render image is also a render buffer? Actually they are 2 buffers for you to play with. Whenever the render window is open (and you can re-open it by pressing F11 without having to wait again for the render) if you press the J key you can switch from Buffer A and B. (the last active one is what you save when you press F3). You can even switch buffers in the middle of a rendering (but I advice against that when rendering very complex scenes, you have been warned!).

The cool thing about having two separate render buffers is that you can have instant before-and-after images for things that you change in the scene. For example you are searching the perfect position for a light source in a scene, you place it and do a render, place the light in another position, switch to the second buffer and do a new render. Now, with the render window open, just press J to see how the change on the light's position influence your scene and that makes your decision easier.

By the way, the render window can be zoomed (by the normal ways or by pressing Z ) to do a closer inspection of the image.

## Render window Tricks

To zoom into the render window, use the Z key.

To find out what the (Red, Green, Blue, Alpha) values of rendered image are, left click and hold the button of your mouse. this will reveal the RGBA values of the pixel below the mouse cursor. ie R 127, G 255, B 13, A40. The values will appear in the bottom left corner of the render window. You can also hold the button and drag the mouse around. This will display the values of the pixels your mouse pointer passes over.

With the render window open, you can press the A key to view an alpha version of the image.

After rendering, you can use hotkey J to use a spare render. After the second render, you can use it again (hotkey J) to jump back to the first render, and then back to the second, thus making it easier to compare slight changes.

## Working while you Render

If you use a X/X11 based window manager, you do not need to watch blender while it renders, go to a different virtual desktop. Blender doesn't have to keep X informed of what's going on and rendering speed may increase.

## Border Rendering

In the rendering buttons find the buttons marked "Border" and "Crop". If you depress "Border", you can get a rendering of any part of what the camera sees. Just do the following :

Go into a camera view using **NumPad 0**, press **Shift+B**. Then, mark the limits of the rendering as you want them using LMB. Next, render the usual way and the section you marked will be rendered first, then it will be integrated to the complete rendering. If you wish that \*only\* the chosen section would be rendered then click on the button marked "Crop" also.

## Creating a cluster of particles which takes little time to render

Render some particles, and make the picture into an alpha mapped tga in GIMP or whatever graphics editing application you use.

Load the image onto a plane. Adjust the alpha settings accordingly. Parent the plane to the emitter. Press dupli-verts. You now have a cluster of particles which takes relatively little time to render. Of course it doesn't have to be an alpha map of particles. That's entirely up to you.

## Alpha from render view

When cut & pasting stuff from render window to {insert your favorite image editor here} using **Alt+PrtScr**, cut and paste the render first, come back to render window and press "A". It changes the view to alpha and you get black & white mask to cut the background nicely in the {again, favorite image editor}. Nice when you do testing in low res.

## View alpha texture as wire.

**Ctrl+d** in 3d very usefully for preview without rendering.

Also, if you have an object (works best on a mesh plane) with an image texture, you can use **Alt+V** key outside edit mode. This will adjust the object's size values so that the image won't be stretched when projected.

## Miscellaneous

### Restoring your “lost” work.

If you go to the temp folder where you have Blender save its temp .Blend files, reload the most recent one and this will save you losing the whole of your work. You don't even need to save a .Blend file for this to work. You can change the settings for this in the tool window at the top of the Blender screen.

### Built in Hot-Key List

Since the few last versions (since 2.28 I think) blender has a built-in all inclusive hot-key list. Just open a text-editor window, and right besides the option to create a new text-buffer there is a menu option called "KEYLIST.intr". Select that option and the full list is loaded in memory

--Edit: There may be many hot-keys missing, especially the newest ones, like M to mirror a mesh, K for the Knife tool, **Alt+Z** for textured view... but the list is a good start at least.

### Un-compiled PlugIns

Blender can load plug-ins for texturing, sequence editor, etc. . However, Blender comes with a few of such PlugIns un-compiled.

In Linux they are located on the plug-ins sub-directory of the default Blender install, and all that you need is a Make command to compile them. I don't know how to compile them in Windows, but there they are, just waiting for you to awake them!!!

## Blender “Easter Eggs” (Weird things included in blender)

All Publisher versions of Blender have been shipping with a Monkey mesh called Suzanne. Just open the main tool box (**Space Bar**) -> Add -> Mesh -> and right below the other primitives you'll see the Monkey.

Why/What it is for? Only NaN programmers know\*. It is supposed to be a private joke among the blender developing team. Since then, its considered as a sort of mascot for blender. However, it is incredibly useful as a quick complex object for testing textures, shaders, etc..

By the Way, Suzanne isn't the only joke included... but I won't spoil the surprise. You will bump with them on your daily work, that is for sure.

- NaN was the company that originally developed Blender

Try this: run blender with the -Y argument (open the command prompt/terminal, go to your blender executable file folder, and type blender -Y)

## Weird Error Message

Sometimes, when trying to use a boolean on a tri-based mesh, Blender gives the prompt: "Wanna Crash?" >Yes Please!

but clicking it doesn't crash. This is because Blender is a lot more stable now compared to when that 'crash' request was coded. It also used to appear when using beauty fill after face fill. (**Ctrl+F** in edit mode)

Blender 2.37 now has a 'widget', which replicates the red/blue/green axis symbol in 3DSMax in the 3d windows. Rotation scaling and movement of objects/Verts/faces etc, can be manipulated using the widget, or in the usual manner of earlier versions of Blender.

## Turn your blender animation into a screen-saver (Windows)

to turn your blender animation Windows binary file into a Windows screen saver, rename the EXE file into SCR and right click it and install !

## Discover the FPS rate of a Window

If you hit **Ctrl+Alt+T** key in a window, Blender will tell you how much time it takes to render a single frame of that window, in milliseconds. Valuable Benchmark

## ???

using construction widget press **Shift** to get precision mode for fine tuning. Release left mouse button (LMB) and holding **Shift** down press it again, then you'll get moving along another axis.

## Sculpt Mode Hotkeys

- F: an interactive brush resize
- Shift F: an interactive brush strength adjuster
- Ctrl F: in interactive texture angle adjuster for your brush.
- Shift B: a rectangular zoom selection for close-up work
- Alt B: hides all but selected rectangle
- A: toggles airbrush
- S: smooth
- D: draw
- G: grab
- L: layer

- I: inflate
- P: pinch
- V: toggles add and subtract in draw mode
- Use X, Y and Z to toggle axis mirroring.

## References

- [1] <http://www.cgsociety.org/>
- [2] <http://forums.cgsociety.org/showthread.php?t=52560&page=1&pp=15>
- [3] <http://www.elysiun.com/forum/viewtopic.php?t=14058&highlight=metaball>

# Blender 3D: Troubleshooting

---

## ATI Radeon Slowdown Problems

Go to 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.dll\_ 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.

Addendum: I found this solved a problem of crashing on start up after updating my Catalyst driver from 7.2 to 7.8, that is expanding the old 7.2 atioglxx.dll\_ to atioglxx.dll and copying it to the Blender directory.

## 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](#) [1]

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...voilà! Blender runs like it should!

[this also works by renaming **any** .dll to "atioglxx.dll". Try it by copying and renaming any of the .dlls in the blender folder to it, then launching blender. Worked for me!] [This works, because it can't use proper gl calls and uses no hardware acceleration so speed goes crappy...]

## References

[1] [http://www.megagames.com/news/redir.cgi?http://download.nvidia.com/Windows/53.03/53.03\\_winxp2k\\_english\\_whql.exe](http://www.megagames.com/news/redir.cgi?http://download.nvidia.com/Windows/53.03/53.03_winxp2k_english_whql.exe)

# Blender 3D: Noob to Pro/Creating Blender Libraries

---

You've finally made that perfect object, armature or material: a gamepiece, a robot, a fully rigged vampire character, or a millimeter-accurate model of the Empire State Building. Besides using this work in your own artistic project, you have made the courageous decision to share that creation with the world.

Sharing your creative work is a great way to "give back to the community," even if you don't write Blender code and you can't translate the Blender documentation to Swahili.

## How to Make a Library

So how do you make a library? In Blender, you don't need to export to a special format. In fact, you don't need to do anything special beyond saving your regular .blend file. Every .blend is a library file. Users can Append what they like from your .blend file, and ignore parts which they don't need for their own project.

This scheme has some benefits and some drawbacks.

The benefit of using .blend files as a library format is that it's super easy to include extra stuff to help the user see the objects. If the user loads the .blend file directly, it works like the pretty packaging for foods, including a quick and easy way to get a pretty "serving suggestion" rendering of the library contents. What you save in your .blend is what the user will see when they load it, including all your user interface settings, lighting types, and camera positions.

The drawback of using .blend files as a library format is that it's super easy to include *unintended* things, such as extra meshes, unused material and texture channels, and other things which the user will not find helpful. Blender doesn't save things which are no longer referenced anywhere, but it cannot read your mind if you leave a spare mesh on layer 13 which uses some abandoned Material.034.

Also, some people are not accustomed to the way that Blender saves all of the user interface settings along with the .blend file. When they load your mesh, they see your way of working. This can be instructive, but unless that's your intention, it's best to try to stick to a simple and clean user interface setup for your library files.

For best results, you need to apply some discipline to publish a clean and useful library.

## Library Usefulness Checklist

After having used several library .blend files from various sources, I propose that anyone making libraries follow a few suggestions:

- keep to a certain object, theme or area of focus for each library file
- all the test cameras and lights prefixed with a dash; e.g., **-Camera**, **-HemiLight.001**, etc.
- any other components not intended for Appending prefixed with a dash; e.g., **-ground**
- all the test cameras and lights on the last one or two layers (**lower right layer button**)
- any composite object intended for Appending organized in a **group** to hold it together
- any groups, objects, materials, textures intended for Appending given **meaningful names**
- document your unit scheme; e.g., **1 blender unit == 1 imperial foot**, etc.
- any other layer contains test-render-ready objects or scenes
- choose rendering and world **settings which will not take an hour** to make a simple test render
- make visible upon loading the layers required for a camera, a good lighting angle, and a shared object
- make visible upon loading one small text file which lists layers and objects
- make visible upon loading any python script, **with instructions on how to start it** in a big comment
- make your licensing expectations clear: **artistic license**, **creative commons**, etc.
- pack the texture files and other data before saving that final .blend for publishing
- sign your work, stable email address or website url if possible

The dash prefixes for test-rendering cameras, boxes, floors and lights will help the user know at a glance what to Append and what not to Append from your library.

Here's a quick way to throw out all the stuff you really don't need, including extra meshes, materials and user interface complexity.

- save your current .blend (and make a nice backup file too)
- shut down Blender entirely
- open up Blender again, which will load the default settings
- delete the cube and cameras from the default settings
- Append all the useful parts of your library .blend file (including the *useful* test-rendering items)
- select the proper test-rendering camera for users to try out your model quickly (select, then Ctrl+KEYPAD0)
- adjust the views to ensure important things are visible and ready to render
- save the new library .blend, ready to publish

Blender saves the default settings in a file called **.bblend** on your disk. If your own preferred default settings are still too complicated for newcomers to understand, you can move that file away temporarily to get the "factory" built-in default settings which the Blender team produces as a part of each new version of the software. Move the file back again when you want to go back to your individual way of working.

---

## Publishing Your Library

It's helpful to include the .txt file and/or post it separately so that people can read a summary before loading the blend file. This should include any credits, usage notes, layer explanations, and licensing information. For you Unix folks, remember to run it through **unix2dos** to enforce \r\n CRLF newlines, readable by people with less flexible tools such as Windows Notepad.

It's also of immense benefit to put up small test-renders of your library objects or materials. They don't need to be large but they should give an honest view of the work you're sharing before a potential user takes the time to download library files that will not be useful to them.

So, where do you publish your work of artistic genious?

- Blender 3D Model Repository <sup>[1]</sup>
- <http://blenderartists.org/forums>
- <http://www.deviantart.com/>
- your own website

If you post things on your own website, try not to rely on a free site that will over-run your bandwidth limits and disappear two months later. Search engine links will sometimes live on for years, just frustrating those who were hoping to find a millimeter-accurate model of the Empire State Building.

## Beyond Libraries

If you have even more time to spare, consider writing up a tutorial on how you achieved any tricky results!

## Thank You

For every artist who chooses to share their creative works with the community, there are a dozen artists, or even hundreds, who thank you immensely.

## References

[1] <http://www.e2-productions.com/bmr>

---

---

# Appendix: References

---

## Blender 3D: Noob to Pro/Blender Glossary

---

This is a **work-in-progress** glossary of Blender related terms, so add more as you find them.

### Glossary

#### Options

- **Window Types** - In Blender each window can be changed in type. These types are:
  - **Scripts Browser** - This window is used to initiate any Python scripts saved in Blender to add functionality.
  - **File Browser** - Browses the files on your computer in Blender.
  - **Image Browser** - Browses the files on your computer in Blender with visuals of images displayed.
  - **Node Editor** - The enhanced material editing system in Blender in which you can customize the input/output of materials or scenes and composite them together.
  - **Buttons Window** - Displays the Blender scene and object editing options.
  - **Outlines** - Breaks down the entire collection of objects and their properties in an organized list.
  - **User Preferences** - Allows you to change Blender's controls and preferences.
  - **Text Editor** - Simple text editor for any notes/Python code.
  - **Audio Window** - Allows you to upload, enhance, and save audio files into the Blender library.
  - **Timeline** - Shows the animation's beginning, ending, and current frame, as well as a timeline with marks indicating all the key positions of an object.
  - **Video Sequence Editor** - Feature for the composition of sound, scenes, videos and other cinematographic elements.
  - **UV/Image Editor** - Allows you to add and alter image files in Blender and also apply these images onto the faces of meshes in an editable fashion as a material.
  - **NLA Editor** - Allows you to change the key positions of all actions/IPOs.
  - **Action Editor** - Allows you to alter the key positions of an armature and its collection of bones.
  - **IPO Curve Editor** - Allows you to insert saved key points of the positions, transformations, and settings of objects of the course of a series of frames.
  - **3D View** - Window through which you work with, add, and place objects in a 3D world.

#### Terms

- **Mesh *n*.** - A collection of faces, edges, and vertices which can be modeled and manipulated in Edit Mode.
- **Vertex (*pl. vertices*) *n*.** - A 3-dimensional coordinate which in groups to comprise a polygon. With default settings, it is represented in Blender by a purple dot when unselected, or a yellow dot when selected.
- **Edge *n*.** - A wire-like line representing the boundary of 2 adjacent vertices.
- **Face *n*.** - A planar connection of edges representing a boundary, field, or solid surface. Identified individually in Edit Mode as a black box in the middle of the corresponding face.
- **Icosphere *n*.** - A sphere composed of *n* triangular faces composed in a manner to give the best "smoothness" for the lowest memory consumption.
- **UVSphere *n*.** - A sphere composed of *n* square faces arranged in rings to allow the smoothest application of real-time movement of vertices/faces and application of images onto the spectrum of faces.

- **Lamps** - Category of 5 light-emitting objects:
  - Lamp - Light is emitted in the scene with a uniform spherical output which fades with distance, decreasing eventually to 0. The distance can be specified in its options.
  - Sun - Light is emitted in the scene with a uniform spherical output and remains uniform regardless of the distance from it.
  - Spot - Light is emitted in the scene in a predetermined conical area with light properties similar to lamp and also initially calculates the shadow (lack of light) present behind an object (Known as ray shadow).
  - Hemi - Sun-like light emitted in a 180 degree arc. Works just like a Sun lamp, but with less functionality. Best used for quick, broad illumination.
  - Area - Light which works similar to a Spot lamp but shape of the cone is definable.
- **Ray shadows n.** - Method of generating shadows cast by light by calculating the angle and faces of the interfering object to create a smooth, sharp and complete shadow. Memory-intensive, best quality.
- **Buffer Shadows n.** - Method of generating shadows cast by light by calculating the angle of a mesh's edge and creating a set amount of shadow "squares" on the shadow-receiving objects which are calculated from a set buffer-ratio. Memory-friendly, while possibly lower quality.
- **Camera n.** - An object which acts as the viewpoint (when active) for the scene when the scene is to be rendered.
- **Armature n.** - An object which contains 'bones'. Bones are sub-objects which are connected with each other in a chain, each bone-link is parented to the bone before it. The purpose is to parent any objects to the bone, and also you may parent a mesh to the entire armature and have selected vertices of the mesh parented to bone(s) percentage-wise (0% means no effect, 50% means vertices copy half the transformation value, 100% means vertices transformed as full child of bone.)

## Blender 3D: Noob to Pro/Object Mode HotKeys Review

---

### 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.
- **ALT+AKEY** - changes the current Blender window to Animation Playback mode. The cursor changes to a counter.
- **ALT+SHIFT+AKEY** - the current window, plus all 3DWindows go into Animation Playback mode.
- **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.
- **ALT+CKEY** - brings up the convert menu.

**D**

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

**E**

- **ALT+EKEY** - Start/stopEditMode. Alternative hotkey: **TAB**.

**F**

- **FKEY** - In the 3D View, switches to UV Face Select Mode if selected object is a mesh. Pressing FKEY again will bring you back to Object Mode.
- **CTRL+FKEY** - 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).

**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 its Z and Y position. In this mode it will only move along the global X axis.
- **GKEY XKEY XKEY** - Grabs the selection and locks its 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 its Z and X position. In this mode it will only move along the global Y axis.
- **GKEY YKEY YKEY** - Grabs the selection and locks its 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 its X and Y position. In this mode it will only move along the global Z axis.
- **GKEY ZKEY ZKEY** - Grabs the selection and locks its 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** - UNDO 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-F11 F12**

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

# Blender 3D: Noob to Pro/Edit Mode HotKeys Review

---

## 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.
- **ALT+AKEY** - changes the current Blender window to Animation Playback mode. The cursor changes to a counter.
- **ALT+SHIFT+AKEY** - the current window, plus all 3DWindows go into Animation Playback mode.
- **SHIFT+AKEY** - brings up the toolbox.

### B

- **BKEY** - Activates box-select tool.
- **BKEY+BKEY** - Circle Select. If you press BKEY a second time after starting Border Select, Circle Select is invoked. Use NUM+ or NUM- or MW to adjust the circle size. Leave Circle Select with **RMB** or **ESC**.
- **ALT+BKEY** - Select portion of viewing area to only be visible.

### C

- **CKEY** - Centers the 3D View where the 3D cursor currently is.
- **SHIFT+CKEY** - snap the 3D cursor back to the origin

### D

- **DKEY** - Brings up a Draw Type menu.
- **SHIFT-DKEY** - Duplicates an object

### E

- **EKEY** - Extrude selection

### F

- **FKEY** - creates segment/edge/face. 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.
- **ALT+FKEY** - 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+FKEY** - Flip faces, selected triangular faces are paired and common edge of each pair swapped.
- **SHIFT+FKEY** - 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.

**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 its Z and Y position. In this mode it will only move along the global X axis.
- **GKEY XKEY XKEY** - Grabs the selection and locks its 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 its Z and X position. In this mode it will only move along the global Y axis.
- **GKEY YKEY YKEY** - Grabs the selection and locks its 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 its X and Y position. In this mode it will only move along the global Z axis.
- **GKEY ZKEY ZKEY** - Grabs the selection and locks its 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.

**L**

- **LKEY** - Select connected vertices under mouse pointer. (by Noob Lucio Renovato)

**M**

- **MKEY** - Brings up Mirror Axis menu.
- **ALT+MKEY** - Merge selected points.
- **SHIFT+CTRL+ALT+MKEY** - Select unclosed edges / holes in mesh (Edge select mode).

**N**

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

**O**

- **OKEY** - toggles proportional edit mode

**P**

- Enter the Blender 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****U**

- **UKEY** - Opens UV Unwrap Menu
- **SHIFT+UKEY** - N/A

**V**

- **VKEY** - Rip - for example, select one edge of a cube, and press **VKEY** to separate and drag it away from the edges it's attached to.

**W**

- **WKEY** - Boolean Tools menu in Object mode. Specials Menu in Edit mode.

**X**

- **XKEY** - delete the selection.

**Y**

- **CTRL+YKEY** - redo previously undone edit

**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-F10**

- Nothing Really

**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 copy of what is selected at the place clicked.
  - if a single vertex is selected or vertexes on a non-enclosed object are selected, a copy will be created and will be joined to any previously selected vertices by an edge, or edges.
  - if an enclosed mesh (a cube, etc.) is selected, an unconnected copy will be created under the cursor.

**RMB**

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

# Blender 3D: Noob to Pro/Hot Keys

---

**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.

**DOWNDARROW.** 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.

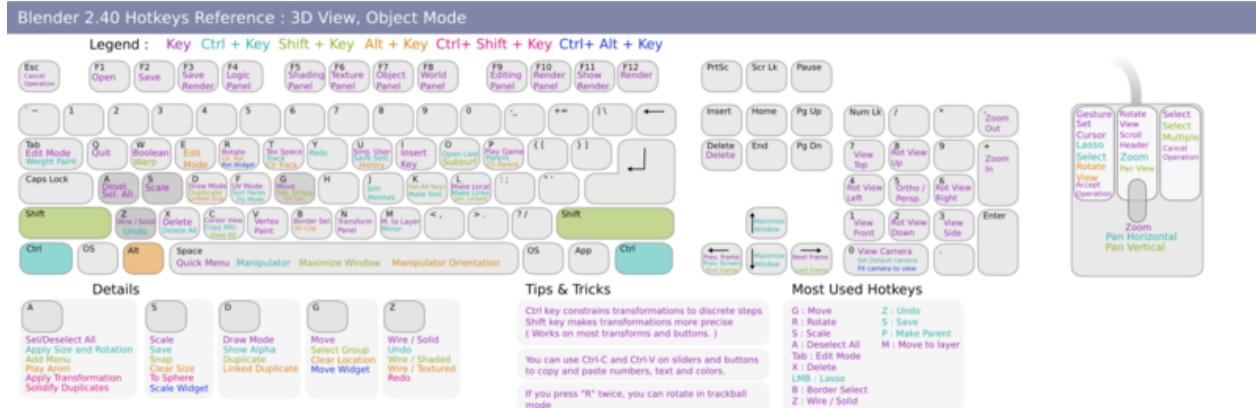
**CTRL+Y.** Redo. Mac users may use CMD+Y.

**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/stopEditMode. 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 on/off. **CTRL+1** to **CTRL+4** switches to the relative SubSurf level for display purposes. Rendering SSubSurf 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 constraints 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 (SeeEditMode 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 toEditMode, 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 be 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 its 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.

---

## Blender 3D: Noob to Pro/FAQ

---

This section of frequently asked questions is intended to cover general information about the goal and structure of Blender 3D: Noob to Pro and the software in is designed to teach, Blender.

### What is Blender?

Blender is a free and open source program available at [blender.org](http://blender.org)<sup>[1]</sup> for Windows (32 and 64-bit XP, Vista, and 7), Mac OS X (PowerPC and Intel 32 and 64-bit), Linux (32 bit and 64-bit/x86\_64/AMD64), Solaris, and Irix. Blender 2.49b is the current stable version with 2.50 alpha 0 being the current developmental testing build. It is designed to be a fully functional 3D content creation suite, allowing all common methods of 3D editing, animation, texture mapping, and rendering. Interested readers may learn more about Blender's history, functionality, and productions which have utilized Blender at its entry on Wikipedia.

### What is this book about?

This book is designed as a series of tutorials, which lead a user to high general proficiency in using Blender. Following the tutorials in order is highly recommended, as tutorials introduce actions which readers will be expected to know about in later sections. This book is not intended as a knowledge compendium or reference manual, although there is a great number of links to external resources for further information on specific topics.

### How accurate is the information in the book?

Blender is in active development, and as such the appearance, function of the user interface, and tools in the program are subject to change. With Blender 2.50, an entirely new graphical user interface and heavily edited internal code will result in a different appearance and possible changes to keyboard shortcuts (hotkeys). Every attempt is made to keep up with the current standards within Blender and most tutorials remain valid between versions.

### I have a question, where can I get help?

BlenderArtists<sup>[1]</sup> and Blender Newbies forum<sup>[2]</sup>, both large English user forums, can be posted on or searched easily and BlenderWiki<sup>[3]</sup> also has a large documentation database. Similar forums in most languages can be found at [blender.org](http://blender.org)'s community page<sup>[4]</sup> and the discussion pages are also useful. If your question turns out to be very common, please take a moment to see if you can add it to the book by integrating it smoothly with an existing page.

## References

[1] <http://www.blenderartists.org/forum>

[2] <http://forums.blendernewbies.com/>

[3] <http://wiki.blender.org/>

[4] <http://www.blender.org/community/user-community/>

---

# Blender 3D: Noob to Pro/Blueprint Links List

---

- <http://www.shipschematics.net/> Starship Schematics Database
- [http://www.suurland.com/blueprints\\_archive.php](http://www.suurland.com/blueprints_archive.php)
- <http://www.the-blueprints.com/>
- <http://www.modeling.net/> NOT 3D modeling, but human models of the fashion type
- <http://www.eplans.com/> Not 3D either, but if you navigate around a bit, there are a lot of great floor plans if that's what you're looking for.

# Blender 3D/Tutorial Links List

---

Here you can find useful **Blender Tutorial Links** written in English. Feel free to add some cool tutorials, but, on this page, only in English. For tutorials in other languages please read About. A tutorial about editing these pages can be found here. New to the 3D world? Read To Those Learning 3D <sup>[1]</sup>.

## Official Blender Documentation

Almost all **Manuals** can be found in the latest official **Blender Manual Books**. For additional tutorials look below this topic.

- Tons of Video Tutorials <sup>[2]</sup>
- Blender MediaWiki <sup>[3]</sup> \* exposed
- Documentation at Blender3D.org <sup>[4]</sup>
- Categorized list of Blender 3D Tutorials <sup>[5]</sup>

## Interface

- The best website to learn Blender 3d <sup>[6]</sup>
- Blender Interface Theme Repository <sup>[7]</sup>
- Basic Editing <sup>[8]</sup>
- Blender Hotkeys
- Blender Hotkeys II \*broken link\* <sup>[9]</sup>
- Basic Blender <sup>[10]</sup>
- Basic Interface <sup>[11]</sup>
- Blender Multimedia <sup>[12]</sup>
- Changing Views <sup>[13]</sup>
- First Impression <sup>[14]</sup>
- Getting Started <sup>[15]</sup>
- Removing Orphan Vertices <sup>[16]</sup>
- Seeing Textures with Alt-Z <sup>[17]</sup>
- Stereoscopic camera for editing <sup>[18]</sup>
- Blueprint Setup : Blender3d Club <sup>[19]</sup>

## Mesh Modeling

- Basic Modeling [20]
- Video - DupliVerts Tutorial [21]
- How to solve Blender's smoothing problems [22]: discussion about smoothing strategies: Smooth/Solid, Auto Smooth, EdgeSplit modifier...
- Heightmaps [23]
- Mesh Editing Techniques [24] *broken link*
- Object Extrusion and Procedural Objects [25] *broken link*
- Loops Corkscrews Problem [26]
- Loopings Problem [27]
- Knots in Curves Problem [28]
- ZeTool [29]
- DupliFrames Modeling I [30]
- DupliFrames Modeling II [31]
- Dupliframes [32]
- Spin Modeling : Blender3d Club [33]
- Spin and Spin Dup [34] - Needs a lot more explanation
- Automatized BevOb [35]
- Extrusion along a Path using BevOb [36]
- Path Extrusion [37]
- DupliVerts [38]
- Limit DupliVerts [39]
- Types of Handles for Bezier Curves [40]
- Beveled Cube [41] (*pdf tutorial*)
- Joining/Separating Parts of a Mesh [42]
- Extrusion Controlled by IPO [43]
- Basic objects [44]
- Modelling Techniques and Strategies [45]
- Easily Remove Orphan Edges and Vertices [46]
- Non-destructive bevel effect [47]
- "Cage" Technique Tutorial [48]
- Creating Fire [49] - *broken link*

## Nurbs and Subsurface Modeling

- Subsurface modeling tutorial [50]
  - IPOs, Lattices, Nurbs & Stuff [51]
  - Cross-Sections [52]
  - Curve Resolution [53]
  - Weight Parameter [54]
  - OrderU, UniformU and EndpointU [55]
  - Curves -n- Bevels [56]
  - Making a Hole in Subsurf [57]
  - Subsurf Edge Rolling - Round [58]
  - Subsurf Edge Rolling - Square [59]
  - Modeling with lattices - a fork [60]
  - Subsurf modeling 2 [61]
-

- Metaballs [62]
- Abstract SubSurf modeling I [63]

## Specific Object Modeling

- Sponge Bob : Blender3d Club [64]
- Double-Helix [65]
- Turtle [66]
- Leaf Shader Tutorial [67]
- Sears-Roebuck Dairy Barn [68]
- Golf Ball [69]
- Celtic Knot [70]
- Simple Tree [71]
- Making a pedestal [72]
- Building a Castle [73] (broken)
- Ice Cube [74] (*pdf tutorial*)
- Water I [75] (*pdf tutorial*)
- Water II [76] (*pdf tutorial*)
- Space Pod [77] (Dead link)
- Grass [78] (Link moved)
- Bongo Creature [79] (Broken Link)
- Spiral Stairs [80] (All blenderwars links dead)
- Waste Basket [81] (Domain Expired)
- Beveled Cube [82] (Domain Expired)
- Simple Box [83] (Domain Expired)
- Fountain with Moving Water [84] (Dead link)
- Track Creation [85] (Broken link)
- Dolphin [86]
- Dice [87]
- Logo [88]
- Cutting through steel [1]
- Roller Coaster [89]
- Roller Coster Cobra [90]
- Modelling a Glass [91]
- Ocean View [92]
- Solar Systems for beginners [93]
- Chair [94]
- Spiral Stair [95] (*pdf tutorial*)
- Lighthouse [96] (Dead link)
- Landscape [97]
- Landscapes Easy [98] (Dead link)
- Cloth Tutorial [99] (dead link) archived version [100]
- Cup [101]
- Cola cane [102] (Dead Link)
- Candle [103]
- Volcano [104]
- Orbital Logo [105]
- Realistic Planet [106] (Dead Link)

- Landscape Cartoon<sup>[107]</sup> (*pdf tutorial*)
- Text<sup>[108]</sup>
- Text Part 1: Manipulate Text<sup>[109]</sup> (Old link. Might this be the aimed for one? Text Part 1: Manipulate Text<sup>[110]</sup>)

## Human

- Hand Palm<sup>[111]</sup>
- Make Human: Modeling a New Target<sup>[112]</sup>
- Eyes<sup>[113]</sup>
- Hair Tutorial<sup>[114]</sup>
- Wikipedia: Human Body Proportions<sup>[115]</sup>

## Cars

- Creating a Toon Car<sup>[116]</sup> (Creating a Toon Car<sup>[117]</sup>, The old, Broken Link)

## Texture Mapping

- Video - Basic Materials<sup>[118]</sup>
- Making a Lightsaber with Halos (link repaired!)<sup>[119]</sup>
- The Unofficial Texturing Tutorial<sup>[120]</sup>
- Material Indice Tutorial<sup>[121]</sup>
- Raytraced Transparency and Refraction<sup>[122]</sup>
- Chrome and Shiny Metal Surfaces<sup>[123]</sup>
- Texturing Tutorial<sup>[124]</sup> (NOTE: For the game Total Annihilation, not really a general tut)
- Textures in OrCo mode<sup>[125]</sup>
- Material Indices<sup>[126]</sup>
- Textures Channels<sup>[127]</sup>
- Textures Mapping<sup>[128]</sup>
- Specular Color<sup>[129]</sup>
- The Color<sup>[130]</sup>
- Materials in Blender<sup>[131]</sup>
- Alpha, Bump, and Specular image textures<sup>[132]</sup>
- Specular Maps : Blender3d Club<sup>[133]</sup>
- Image Textures<sup>[134]</sup>
- Decals<sup>[135]</sup>
- Chrome<sup>[136]</sup>
- Alpha Masks<sup>[137]</sup>
- Textures with Alpha<sup>[138]</sup>
- Textures with Bumpmapping<sup>[139]</sup>
- Using More Than One Color<sup>[140]</sup>
- Adding Color To Your Shapes<sup>[141]</sup>
- Using Textures and Bumpmapping Them<sup>[142]</sup>
- Texture Mapping Tutorial<sup>[143]</sup> (*pictures ok now 18/03/2006*)
- Environment Mapping<sup>[144]</sup> (*pictures ok now 18/03/2006*)
- Mapping Caracter<sup>[145]</sup>
- Creation and Mapping<sup>[146]</sup>
- Displacement Mapping<sup>[147]</sup>
- Game Sprites<sup>[148]</sup>

- Stencilling Textures [149]
- Using Texture Stencils [150]

## UV Mapping

- Video - UV Mapping [151]
- Intermediate and Advanced UV Mapping [152] \* exposed
- UV Mapping & Texturing [153]
- Sub-Surf and UV-Texturing problem [154]
- UV Texture Skinning [155][Dead Link]
- Character UV Textures [156][Dead Link]
- UV Texturing in Blender [157]
- Tuhopuu UV editor mapping [158][Dead Link]
- UV Mapping [159]
- Complex material for a sword [160]
- UV Mapping a road and box [161]
- Create a Software Box (involves UV Mapping) [162]

## Animated Textures

- Video - Changing Colors Of Objects Mid Animation [163]
- Blender TexMesh Tutorial [164] \* exposed
- animating masks for simulating ice freeze [165]
- Animated Procedural Textures [166] (*pdf tutorial*)

## Nodes

- Video - Blender Nodes [167]

## 2D Texture Painting Techniques

This part isn't about using Blender, but about necessary **2D knowledge** for advanced Blender users.

- Steven Stahlberg's Tutorials [168] \* exposed
- Dirty Metal Texture [169]
- Gritty Pipelines [170]
- Digital Painting [171]
- NWN Texturing Tutorials [172]
- Fixing Lighting Irregularities in Self-Tiling Maps [173]
- Texture Map for the Iris of the Eye [174]
- Airplane Texture Tutorial [175]
- Game Skinning Tips [176]
- Game Lighting Basics I [177]
- Game Lighting Basics II [178]
- Texturing with Gimp [179]

## Lighting, Shadows and Rendering

- Light - a detailed tutorial [180]
- Yafray as an Integrated External Renderer [181]
- Basic 3 Point Lighting [182]
- Radiosity I [183]
- Radiosity II [184]
- Ramp Shaders [185]
- Simulating Radiosity [186]
- Caustic Sampler Tutorial [187]
- Shadows Control [188]
- The 'World' buttons within Blender [189]
- Mastering Shadows [190]
- Ambient Occlusion [191]
- Area Lights [192]
- Depth of Field [193] (dead link)
- Soft Lights [194] (*pdf tutorial*)
- Blender's Mist [195]
- Basic Lighting [196]
- Optimizing Renders [197]
- Flares [198]
- Toon Shading [199] Broken Link
- Simple Gold Ring with Caustics [200]

## Armatures and IK

- Video - Basic Armatures [201]
- Dancing Flor [202] \* exposed
- Animation Workshop II [203] \* exposed
- Using Ipo driven shape keys to correct deformations in joints [204] \* exposed
- Animation recode project [205]
- Action Constraints tutorial made easy [206]
- Rigged Character [207]
- Driven Hand Rig [208]
- Constrained Mechanics [209]
- Armatures Tutorial [210] (Dead link)
- Animation using Armatures [211]
- Ikas Blender 3D - Introduction [212]
- Constrained Mechanics [213]
- Using Armatures [214]
- Making and Using Armatures [215]
- Character Animation tools [214] (broken link ?)
- IK Solver Constraint [216]
- Softbody for Rigged Characters [217]
- Rigging limbs that can twist [218]
- Rigging tricks [219]
- IKA Tutorial [220]

## Animation

- Weight Painting : Blender3d Club [\[221\]](#)
- Learn Animation [\[222\]](#)
- Video - Animation Basics [\[1\]](#)
- Simple Animation [\[223\]](#)
- Basic Keyframing [\[224\]](#)
- Effects [\[225\]](#)
- Character Animation [\[226\]](#)
- Run Cycle [\[227\]](#) (Non Blender specific)
- Lip Sync I [\[228\]](#)
- Lip Sync II [\[229\]](#)
- Walking Blues [\[230\]](#)
- Non Linear Action Editor - NLA I [\[231\]](#)
- Non Linear Action Editor - NLA II [\[232\]](#)
- Wave Effect [\[233\]](#)
- Build Effect [\[234\]](#)
- Animation effects [\[235\]](#)
- Plane Following a Path [\[236\]](#)
- Object Following a Path [\[237\]](#)
- Cyclic Animation [\[238\]](#)
- Animating Materials [\[239\]](#)
- Walking Tutorial [\[240\]](#)
- Change Cameras during an Animation [\[241\]](#)
- Camera Tracking With Additional Roll Constraint [\[242\]](#)
- Shape Keys / PDF: Blender3d Club [\[243\]](#)
- Eye Tracking : Blender3d Club [\[244\]](#)
- Simple Armature Animation : Blender3d Club [\[245\]](#)

## Particles

- Everything about particles [\[246\]](#)
- Video - Particle Basics [\[247\]](#)
- Video - Grass with Particles [\[248\]](#)
- Video - Explosions with Particles and the Explode Modifier [\[249\]](#)
- Particle Interaction [\[250\]](#)
- Static Particles [\[251\]](#)
- Static Particle Effect [\[252\]](#)
- Particles Effect [\[253\]](#)
- Making a Fireplace [\[254\]](#) (down 06.08.2006)-(still down 08.15.2009)
- Making a Fountain With Particles [\[84\]](#)
- Dust Particles [\[255\]](#)
- Flames [\[256\]](#)
- Liquid [\[257\]](#)
- Explosions [\[258\]](#)
- Smoke [\[259\]](#)
- Making a Rain Effect [\[260\]](#)
- Particle and field basics [\[261\]](#)

## Fluid Simulation

- Video - Fluid Simulation [1]
- Fluid Simulation Basics [262]
- Basic Blender Fluid Simulation guide [263]

## Compositing

- Matching Real Lighting [264]
- CGI/Live Action Interaction [265]
- Compositing CG and Live Action in Blender [266]

## Game Engine

- Rigid Body Constraints : Blender3d Club [267]
- Video - Blender Game Engine Basics [1]
- Video - Making Dominoes [268]
- Game Engine Development [269]
- New fully integrated game engine? [270]
- Game Blender Documentation [271] (Last modified 13. 07. 2001)
- #GameBlender [6]
- Walkthrough Tutorial [272]
- Overlay Scenes [273]
- Walking Tutorial [240]
- Make a Menu [274]
- Flipper [275] (Broken Link)
- Armatures in the Game Engine [276]
- Multilevel Maze [277]
- Creating Boulders that make the map restart when hit [278]
- Mouse Look 1st Person. Options include capping, invert mouse pitch and adjust mouse sensitivity. [279]
- Viewports. Creating a split screen. [280]
- Python game functions. Sample Code Included. Updated to 2.48 [281]
- Car Setup. Use Bullet Physics vehicle wrapper. [282]
- Using Blender to host/contain a storyboard

## Python and Plugins

- Python Lessons [283] \* exposed dead link
- Make Hair [284] \* exposed
- Povanim Export Script [285]
- AI Path Importer [286]
- 3D-No Plugins; Put your 3D Blender space on web! [287]
- Randomizer Script [288]
- Embedding the 3D web plugin in a web page [289]
- Different Useful Scripts [290]
- AfterGlow, Polyline, Hitchcock ZoomEffect, ... [291]
- Dynamica [292]
- BlendSaber [293]
- Lsystem tree maker [294]

- Python Scripting for Procedural Animation [295]
- Focalblur - A Matter of Depth [296]
- Python API Introduction [297]
- Python API, Making a Square Mesh [298]
- Python API, Iterations [299]
- Python API, Automating Vertex Creation [300]
- Python API, Automating Face Creation [301]
- Python API, Making Potatoid [302]
- Python Script, To build an Empty for EnvMap [303]
- Python Script, Bezier Curves Import [304]
- Python Script, Paths import [305]
- Python Script, Importing Adobe Illustrator Format [286]
- Python Script, Lionhead The Movies .msh import/export [306]
- Python Script, Mesh Explosion [307]
- Python Script, Level Of Detail [308]
- Python Script, Wire Shadows and extrusions [309]
- Python Script, Changing the active camera instantly [310]
- Subsurface Scattering in Blender [311]
- Using the Superficial Scattering Script [312]
- City Block Generator [313]
- Blender Camera Calibration with Live Camera [314]
- Import-export different formats, Different Generators, etc... [315]
- Horn Extrude [316]
- Mesh shaker and tutorial [1]
- Batch STL [317]
- Vertices to a Curve Converter [318]
- Zutils, Z-Buffer Utilities [319]
- Setting Vertex Colors with Python [320]

## Using other Programs with Blender

- YafRay [321]
- Verse [322]
- SPE - Python IDE for Blender [323]
- Wings 3D Subdivision Modeler [324]
- Verse Gimp-Blender Plugin [325]
- Kerkythea Renderer [326]
- Equinox 3D [327]
- Wings3D, a Quick and Accurate UVmapping Tool for Blender [328]
- Povray export [285]
- Adobe Illustrator Paths import [286]
- Inkscape SVG import [329]
- Using Blender content in PowerPoint 2000 [330]
- Ter2Blend [331]
- Batch Processing Images [332]
- BVH Info [333]
- Stereogram [334]
- Importing VRML [335]

- Creating Quicktime VRs [336]
- Blender, Python and Mac OS X [337]
- Stylistic Rendering [338]
- Voodoo Camera Tracker [3]
- Lionhead The Movies [339]
- Using Blender with IMVU [340] IMVU - Information [341]

## Distributed Computing

- BURP - Big and Ugly Rendering Project [342]
- Global Rendering-Farm [343]
- OS X Distributed Blender Network Rendering with Xgrid [344] (Incorrect link?)
- Bfarm Distributed Rendering via Internet [345]
- Powua [346] Blender renderfarm

## Maybe someday ...

- LightRay [347] [broken link]
- Toxic [348]

## Video Tutorials

- Blender Underground [349]
- Official Blender Video Tutorials [350]
- Super3boy's Blender Tutorials [351]
- Greybeard's Blender Video Tutorials [352]
- Series of short VTuts that explain basic details [353]
- lessons on Blender 3d [354] (broken link)

## Blender WikiBooks

- Blender 3D: Blending Into Python
- Blender 3D: HotKeys
- Blender 3D: Import and Render a SolidWorks Model
- Blender 3D: MemoBook \* exposed
- Blender 3D: Noob to Pro
- All Blender WikiBooks Modules

## FAQ

- Blender FAQ<sup>[355]</sup> (Generated on September 24, 2001)
- Blender Tips<sup>[356]</sup>

## Repository

This section is not about tutorials, but you can find here different useful stuff for your Blender.

### Blueprints

- <http://cgworld.ru/modules.php?name=Blueprints> \*exposed
- <http://blueprints.onnovanbraam.com/>(broken link)
- <http://smcars.net/>
- <http://www.boatdesign.net/boat-plans-archive/index.htm> boats
- <http://www.mgussin.freeuk.com/00Plans.htm> old cars
- [http://www.airwar.ru/other/draw\\_1w.html](http://www.airwar.ru/other/draw_1w.html) War Planes
- <http://www.suurland.com/>All sorts of cars
- <http://jpracing.racerplanet.com/Cars, Trucks, Buses>
- <http://www.the-blueprints.com/>

### Materials

- Sonix' Material Library(esp. Cars)<sup>[357]</sup> (down)
- Large Free Repository of Materials<sup>[1]</sup>

### Models

- [katorlegaz.com/3d\\_models](http://katorlegaz.com/3d_models)<sup>[358]</sup>

### Photos

- Human Photo References<sup>[359]</sup>
- Brain scan<sup>[360]</sup>

### Textures

- Textures Library<sup>[361]</sup>
- Mayang's Free Textures Hi-Res<sup>[362]</sup>
- Jeremy Engleman's Public Textures Hi-Res<sup>[363]</sup>
- Image\*After Free Images Hi-Res<sup>[364]</sup>
- Free Textures Mid-Res<sup>[365]</sup>
- Sky Maps<sup>[366]</sup>

## Miscellaneous

**CG artists must see or read:**

- How to succeed in Animation <sup>[367]</sup>
- Bitter History of 3D Business <sup>[368]</sup>
- Blender Art Gallery <sup>[369]</sup>
- Classical Film and Video Knowledge Base <sup>[370]</sup>
- Quick Tips in Design&GFX <sup>[371]</sup>

## Open Movies

- Orange <sup>[372]</sup> [broken link]
- NaNo - Blender Internet Virtual Movie Studio <sup>[373]</sup>

## 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 <sup>[374]</sup>

## Other Lists

Please read more about on talk page.

## Tutorials

- blenderartists.org <sup>[375]</sup> \* exposed
- Categorized list of Blender 3D Tutorials <sup>[5]</sup>
- A Monolingual List of Tutorials <sup>[376]</sup>
- A Multilingual List of Tutorials <sup>[377]</sup>
- Dave Jarvis' Blender Tutorials <sup>[378]</sup>
- Blender Tutorials/Ebooks in PDF format <sup>[379]</sup>
- Dark Scarab Tutorials <sup>[380]</sup>

## Miscellaneous

- Blender3d Club : Tutorials, samples, Links. [381]
- SurfnLearn.com : Blender 3D Tutorials [5]
- Blender 3D Links & Resources [382]
- Blender Heads around the Globe(Where live other Blenders?) [383]
- Blenderart Magazine [384] Mirror 1 [385], M2 [386], M3 [387]
- Blender Classroom Tutorial [388]
- Blender Tutorials Search Engine [389]
- Blender Basics 2nd ed - excellent classroom course [390]
- BlenderMasters [391] - Many resources for blender users
- **LANGUAGES (czech)** [392]

## 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) [393].

- German Blenders should have a look at [http://de.wikibooks.org/wiki/Blender\\_3D:\\_Tutorial\\_Linkliste](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.

## References

- [1] [http://glenmoyes.com/articles/to\\_those\\_learning\\_3d.shtml](http://glenmoyes.com/articles/to_those_learning_3d.shtml)
- [2] <http://nystic.com/blender.php>
- [3] [http://mediawiki.blender.org/index.php/Main\\_Page](http://mediawiki.blender.org/index.php/Main_Page)
- [4] <http://www.blender3d.org/cms/Documentation.628.0.html>
- [5] [http://www.surfnlearn.com/view\\_tutorials/Blender\\_3D/All/](http://www.surfnlearn.com/view_tutorials/Blender_3D/All/)
- [6] <http://www.animationforcreation.com/>
- [7] <http://www.graphicall.org/themes/>
- [8] [http://www.ee.oulu.fi/~kapu/cg\\_course/bigfiles/blender/blendman.html](http://www.ee.oulu.fi/~kapu/cg_course/bigfiles/blender/blendman.html)
- [9] <http://blender.excellentwhale.com/>
- [10] [http://www.planetannihilation.com/terragen/tut\\_basic.shtml](http://www.planetannihilation.com/terragen/tut_basic.shtml)
- [11] [http://www.users.bigpond.net.au/blendage/pages/beginners\\_tutorials/tut1/basic\\_interface.html](http://www.users.bigpond.net.au/blendage/pages/beginners_tutorials/tut1/basic_interface.html)
- [12] <http://www.malefico3d.org/viejos-tutoriales/audiovideo-en.html>
- [13] <http://blender3dfree.fr/anglais/tut2/tut2.htm>
- [14] <http://blender3dfree.fr/anglais/premiere.htm>
- [15] <http://blender3dfree.fr/anglais/tut1/tut1.htm>
- [16] [http://membres.lycos.fr/bobois/Tuts/reperer\\_sommets\\_orphelins/sommets\\_orphilins.html](http://membres.lycos.fr/bobois/Tuts/reperer_sommets_orphelins/sommets_orphilins.html)
- [17] [http://membres.lycos.fr/bobois/Tuts/subdiviser\\_pour\\_conquerir\\_la\\_texture/subdiviser\\_pour\\_conquerir\\_la\\_texture.html](http://membres.lycos.fr/bobois/Tuts/subdiviser_pour_conquerir_la_texture/subdiviser_pour_conquerir_la_texture.html)
- [18] <http://brunettotuxfamily.org/index.php?n=En.BlenderStereo>
- [19] <http://www.blender3dclub.com/index.php?name=News&file=article&sid=6>
- [20] <http://www.animationforcreation.com/2008/09/lesson-two-basic-of-blender.html>
- [21] <http://www.youtube.com/watch?v=M6EPXc8N3qs>
- [22] <http://feeblemind.tuxfamily.org/docclear/index.php/2007/08/13/88-how-to-solve-blenders-smoothing-problems>
- [23] [http://members.tripod.com/~funky\\_munku/tuts/blender/heightmaps.htm](http://members.tripod.com/~funky_munku/tuts/blender/heightmaps.htm)
- [24] <http://vrotvrot.com/xoom/tutorials/Corridor/Corridor.html>
- [25] <http://vrotvrot.com/xoom/tutorials/mineRide/mineride.html>
- [26] [http://membres.lycos.fr/bobois/Tuts/Courbes/tire\\_bouchon/tire-bouchon.html](http://membres.lycos.fr/bobois/Tuts/Courbes/tire_bouchon/tire-bouchon.html)
- [27] <http://membres.lycos.fr/bobois/Tuts/Courbes/Looping/looping.html>
- [28] [http://membres.lycos.fr/bobois/Tuts/Courbes/Aplat/a\\_plat.html](http://membres.lycos.fr/bobois/Tuts/Courbes/Aplat/a_plat.html)
- [29] <http://membres.lycos.fr/bobois/Tuts/Courbes/ZeTool/zetool.html>

- [30] [http://membres.lycos.fr/bobois/Tuts/DupliFrames/Modeling/modeling\\_with\\_duplifames\\_en\\_1.html](http://membres.lycos.fr/bobois/Tuts/DupliFrames/Modeling/modeling_with_duplifames_en_1.html)
- [31] [http://membres.lycos.fr/bobois/Tuts/DupliFrames/Modeling/pizza\\_boxes/stack\\_of\\_boxes\\_en.html](http://membres.lycos.fr/bobois/Tuts/DupliFrames/Modeling/pizza_boxes/stack_of_boxes_en.html)
- [32] [http://membres.lycos.fr/bobois/Tuts/DupliFrames/les\\_bases/les\\_bases\\_en\\_1.html](http://membres.lycos.fr/bobois/Tuts/DupliFrames/les_bases/les_bases_en_1.html)
- [33] <http://www.blender3dclub.com/index.php?name=News&file=article&sid=45>
- [34] [http://membres.lycos.fr/bobois/Tuts/Spin\\_et\\_spin\\_dup/Spin\\_and\\_spin\\_dup\\_tutorial.html](http://membres.lycos.fr/bobois/Tuts/Spin_et_spin_dup/Spin_and_spin_dup_tutorial.html)
- [35] [http://membres.lycos.fr/bobois/Tuts/Extrusion\\_par\\_bev\\_ob/Ext1\\_Ext2/Ext1\\_Ext2\\_en.html](http://membres.lycos.fr/bobois/Tuts/Extrusion_par_bev_ob/Ext1_Ext2/Ext1_Ext2_en.html)
- [36] [http://membres.lycos.fr/bobois/Tuts/Extrusion\\_par\\_bev\\_ob/Alignement\\_des\\_axes/Alignement\\_des\\_axes\\_en.html](http://membres.lycos.fr/bobois/Tuts/Extrusion_par_bev_ob/Alignement_des_axes/Alignement_des_axes_en.html)
- [37] [http://membres.lycos.fr/bobois/Tuts/Extrusion\\_par\\_bev\\_ob/Les\\_bases/Les\\_bases\\_en.html](http://membres.lycos.fr/bobois/Tuts/Extrusion_par_bev_ob/Les_bases/Les_bases_en.html)
- [38] [http://membres.lycos.fr/bobois/Tuts/Dupliverts/Dupliverts/dupliverts\\_en.html](http://membres.lycos.fr/bobois/Tuts/Dupliverts/Dupliverts/dupliverts_en.html)
- [39] [http://membres.lycos.fr/bobois/Tuts/Dupliverts/limiter\\_les\\_dupliverts\\_en.html](http://membres.lycos.fr/bobois/Tuts/Dupliverts/limiter_les_dupliverts_en.html)
- [40] [http://membres.lycos.fr/bobois/Tuts/Courbes/Courbes\\_de\\_bezier/Types\\_de\\_poignees\\_bezier/Types\\_de\\_poignees\\_bezier\\_en.html](http://membres.lycos.fr/bobois/Tuts/Courbes/Courbes_de_bezier/Types_de_poignees_bezier/Types_de_poignees_bezier_en.html)
- [41] <http://w1.185.telia.com/~u18510119/tutorials/makeacube.pdf>
- [42] [http://membres.lycos.fr/bobois/Tuts/joindre\\_et\\_separer/joindre\\_et\\_separer\\_ang.html](http://membres.lycos.fr/bobois/Tuts/joindre_et_separer/joindre_et_separer_ang.html)
- [43] [http://jmsoler.free.fr/didacticiel/blender/tutor/en\\_modelextipo00.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/en_modelextipo00.htm)
- [44] <http://blender3dfr.free.fr/anglais/tut3/tut3.htm>
- [45] <http://www.elysiun.com/forum/viewtopic.php?t=21382>
- [46] [http://jmsoler.free.fr/didacticiel/blender/tutor/modesel\\_somparaz\\_en.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/modesel_somparaz_en.htm)
- [47] <http://blendertips.blogspot.com/2006/03/bevel-modifier-workaround.html>
- [48] <http://www.cyphertxt.com/blendertechniquetut.php>
- [49] <http://www.raivestudios.com/tutorials/blender/fire/>
- [50] <http://www.animationforcreation.com/2008/10/lesson-6-imagesseparating-objects.html>
- [51] [http://dev.newmediaworx.com/johnnyb/bplanet/tut\\_bottle.htm](http://dev.newmediaworx.com/johnnyb/bplanet/tut_bottle.htm)
- [52] <http://www.blendernation.com/tutorials/blender-3d-beginner-tutorial-the-cave-of-torsan-a/>
- [53] [http://membres.lycos.fr/bobois/Tuts/Extrusion\\_par\\_bev\\_ob/DefResolU/DefResolU\\_en.html](http://membres.lycos.fr/bobois/Tuts/Extrusion_par_bev_ob/DefResolU/DefResolU_en.html)
- [54] [http://membres.lycos.fr/bobois/Tuts/Courbes/Courbes\\_nurbs/Poids\\_des\\_voisins/Poids\\_des\\_voisins\\_en.html](http://membres.lycos.fr/bobois/Tuts/Courbes/Courbes_nurbs/Poids_des_voisins/Poids_des_voisins_en.html)
- [55] [http://membres.lycos.fr/bobois/Tuts/Courbes/Courbes\\_nurbs/Influence\\_de\\_order/Influence\\_de\\_order\\_en.html](http://membres.lycos.fr/bobois/Tuts/Courbes/Courbes_nurbs/Influence_de_order/Influence_de_order_en.html)
- [56] <http://www.ingiebee.com/Blendermania/curves%20and%20Bevels.htm>
- [57] [http://membres.lycos.fr/bobois/Tuts/subsurfs/Trouer\\_une\\_surface/trouer\\_une\\_subsurf\\_ang.html](http://membres.lycos.fr/bobois/Tuts/subsurfs/Trouer_une_surface/trouer_une_subsurf_ang.html)
- [58] [http://membres.lycos.fr/bobois/Tuts/subsurfs/rouler\\_un\\_bord/rouler\\_un\\_bord\\_rond\\_ang.html](http://membres.lycos.fr/bobois/Tuts/subsurfs/rouler_un_bord/rouler_un_bord_rond_ang.html)
- [59] [http://membres.lycos.fr/bobois/Tuts/subsurfs/rouler\\_un\\_bord/rouler\\_un\\_bord\\_carre\\_ang.html](http://membres.lycos.fr/bobois/Tuts/subsurfs/rouler_un_bord/rouler_un_bord_carre_ang.html)
- [60] <http://www.blender3d.org/documentation/htmlII/x10279.html>
- [61] <http://blender3dfr.free.fr/anglais/tut5/tut5.htm>
- [62] <http://blender3dfr.free.fr/anglais/tut6/tut6.htm>
- [63] <http://www.deviantart.com/view/26691152/>
- [64] <http://www.blender3dclub.com/index.php?name=News&file=article&sid=61>
- [65] <http://www.deviantart.com/view/27132715/>
- [66] <http://www.elysiun.com/forum/viewtopic.php?t=44686>
- [67] <http://www.elysiun.com/forum/viewtopic.php?t=39810>
- [68] <http://www.harkyman.com/searsbarn01.html>
- [69] <http://www.elysiun.com/forum/viewtopic.php?t=31781&postdays=0&postorder=asc&start=15>
- [70] [http://jmsoler.free.fr/didacticiel/blender/tutor/cpl\\_ceilticknot\\_en.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_ceilticknot_en.htm)
- [71] <http://www.elysiun.com/forum/viewtopic.php?t=32306>
- [72] [http://web.telia.com/~u91121962/tut1\\_1.htm](http://web.telia.com/~u91121962/tut1_1.htm)
- [73] [http://www.blender3d.org/\\_media/education/quickstart/Building\\_Castle.html](http://www.blender3d.org/_media/education/quickstart/Building_Castle.html)
- [74] <http://caron.yann.free.fr/blender/IceCube.pdf>
- [75] <http://www.selleri.org/Blender/tuts/Water.pdf>
- [76] <http://www.selleri.org/Blender/tuts/Water2.pdf>
- [77] [http://dev.newmediaworx.com/johnnyb/bplanet/tut\\_pod.htm](http://dev.newmediaworx.com/johnnyb/bplanet/tut_pod.htm)
- [78] <http://lib.hel.fi/~basse/>
- [79] <http://www.enricovalenza.com/makebongo.html>
- [80] <http://www.blenderwars.com/tut.php?module=stairs>
- [81] <http://www.blenderwars.com/tut.php?module=waste>
- [82] <http://www.blenderwars.com/tut.php?module=bevcube>
- [83] <http://www.blenderwars.com/tut-path.php?module=box1>
- [84] <http://www.users.bigpond.net.au/blendage/pages/tutorials/tut5/fountain.html>
- [85] <http://www.geocities.com/swdoughty/blendertute1.html>
- [86] <http://www.blendernation.com/tutorials/blender-3d-beginner-tutorial-dolphin/>
- [87] <http://www.blendernation.com/tutorials/blender-3d-beginner-tutorial-a-die/>
- [88] <http://www.blendernation.com/tutorials/blender-3d-beginner-tutorial-creating-a-logo/>

- [89] [http://membres.lycos.fr/bobois/Tuts/Courbes/completeRC/foreword\\_en.html](http://membres.lycos.fr/bobois/Tuts/Courbes/completeRC/foreword_en.html)
- [90] <http://membres.lycos.fr/bobois/Tuts/Courbes/Cobra/cobra.html>
- [91] [http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender\\_material/tp1-index-ang.html](http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender_material/tp1-index-ang.html)
- [92] <http://www.ingiebee.com/Blendermania/create%20ocean%20view.htm>
- [93] <http://www.ingiebee.com/Blendermania/Solar%20System.htm>
- [94] [http://membres.lycos.fr/bobois/objets\\_a\\_partager/a\\_simple\\_chair.html](http://membres.lycos.fr/bobois/objets_a_partager/a_simple_chair.html)
- [95] <http://w1.185.telia.com/~u18510119/tutorials/spiralstair.pdf>
- [96] <http://www.users.bigpond.net.au/blendage/pages/tutorials/tut1/lighthouse.html>
- [97] <http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/paysages/index-ang.html>
- [98] <http://www.users.bigpond.net.au/blendage/pages/tutorials/tut2/landscape.html>
- [99] <http://www.free-webspace.biz/sonix/BlenderTutes/QuickClothTute.html>
- [100] <http://web.archive.org/web/20040805214133/http://www.free-webspace.biz/sonix/BlenderTutes/QuickClothTute.html>
- [101] <http://kahuna.clayton.edu/~jbrooks/blender/tutorials/cup/>
- [102] <http://free.hostdepartment.com/B/Blender3D/cantut/>
- [103] [http://download.blender.org/documentation/oldsite/oldsite.blender3d.org/93\\_Blender%20tutorial%20Candle.html](http://download.blender.org/documentation/oldsite/oldsite.blender3d.org/93_Blender%20tutorial%20Candle.html)
- [104] <http://kahuna.clayton.edu/~jbrooks/blender/tutorials/volcano/>
- [105] <http://kahuna.clayton.edu/~jbrooks/blender/tutorials/universal/>
- [106] <http://www.enricovalenza.com/realplan.html>
- [107] <http://www.selleri.org/Blender/tuts/CartoonishLandscape.pdf>
- [108] <http://blender3dfree.fr/anglais/tut4/tut4.htm>
- [109] <http://www.raivestudios.com/tutorials/blender/text/>
- [110] <http://www.faintmedia.com/forums/topic/93-text/>
- [111] <http://jlp.nerim.net/tutorials/hands-modelling/hands-tuto-01.html>
- [112] <http://feeblemind.tuxfamily.org/dotclear/index.php/2005/04/11/23-didacticiel-modelisation-dun-nouveau-target---modeling-a-new-target>
- [113] <http://web.pdx.edu/~wlf/tut.html>
- [114] <http://www.elysiun.com/forum/viewtopic.php?t=16008>
- [115] [http://en.wikipedia.org/wiki/Body\\_proportions](http://en.wikipedia.org/wiki/Body_proportions)
- [116] <http://www.faintmedia.com/forums/topic/84-toon-car/>
- [117] <http://www.raivestudios.com/tutorials/blender/tooncar/>
- [118] <http://www.youtube.com/watch?v=2kk2BkVQkyk>
- [119] <http://blenderartists.org/forum/showthread.php?t=62611>
- [120] <http://www.elysiun.com/forum/viewtopic.php?t=11889>
- [121] <http://feeblemind.tuxfamily.org/dotclear/index.php/2004/12/21/5-blender-indices-materiau---material-indice>
- [122] <http://feeblemind.tuxfamily.org/dotclear/index.php/2004/12/26/7-didacticiel-la-refraction---tutorial-refraction>
- [123] [http://www.ualberta.ca/%7Enwy/blender/blender\\_tut1.htm](http://www.ualberta.ca/%7Enwy/blender/blender_tut1.htm)
- [124] [http://www.planetannihilation.com/terragen/tut\\_buildpics.shtml](http://www.planetannihilation.com/terragen/tut_buildpics.shtml)
- [125] <http://www.enricovalenza.com/textures.html>
- [126] [http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender\\_material/tp2-index-ang.html](http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender_material/tp2-index-ang.html)
- [127] [http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender\\_material/didac4-ang.html](http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender_material/didac4-ang.html)
- [128] [http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender\\_material/didac3-ang.html](http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender_material/didac3-ang.html)
- [129] [http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender\\_material/didac2-ang.html](http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender_material/didac2-ang.html)
- [130] [http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender\\_material/didac1-ang.html](http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender_material/didac1-ang.html)
- [131] [http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender\\_material/index-ang.html](http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/blender_material/index-ang.html)
- [132] <http://www.ingiebee.com/Blendermania/BumpSpecAlphaTex.htm>
- [133] <http://www.blender3dclub.com/index.php?name=News&file=article&sid=51>
- [134] <http://www.ingiebee.com/Blendermania/attach%20image%20file%20to%20object.htm>
- [135] <http://www.ingiebee.com/tutorials/Decal%20Mirror%20Modelling/theeth%20decal.htm>
- [136] <http://www.ingiebee.com/tutorials/Digital-Mark.htm>
- [137] [http://membres.lycos.fr/bobois/Tuts/trouer\\_par\\_texture/trouer\\_par\\_texture.html](http://membres.lycos.fr/bobois/Tuts/trouer_par_texture/trouer_par_texture.html)
- [138] <http://www-users.cs.umn.edu/~mein/blender/tutorials/dust/alpha.html>
- [139] <http://www-users.cs.umn.edu/~mein/blender/tutorials/dust/bump.html>
- [140] [http://www.users.bigpond.net.au/blendage/pages/tutorials/tut9/two\\_colours.html](http://www.users.bigpond.net.au/blendage/pages/tutorials/tut9/two_colours.html)
- [141] [http://www.users.bigpond.net.au/blendage/pages/beginners\\_tutorials/tut3/adding\\_colour.html](http://www.users.bigpond.net.au/blendage/pages/beginners_tutorials/tut3/adding_colour.html)
- [142] [http://www.users.bigpond.net.au/blendage/pages/beginners\\_tutorials/tut4/adding\\_texture.html](http://www.users.bigpond.net.au/blendage/pages/beginners_tutorials/tut4/adding_texture.html)
- [143] [http://dev.newmediaworx.com/johnnyb/bplanet/tut\\_mapping.htm](http://dev.newmediaworx.com/johnnyb/bplanet/tut_mapping.htm)
- [144] [http://dev.newmediaworx.com/johnnyb/bplanet/tut\\_envmap.htm](http://dev.newmediaworx.com/johnnyb/bplanet/tut_envmap.htm)
- [145] <http://otothegardener.free.fr/tutorials/LittleOTO/anonce/anonce.htm>
- [146] [http://membres.lycos.fr/bobois/Tuts/boite\\_de\\_projection/boite\\_de\\_projection005\\_en.html](http://membres.lycos.fr/bobois/Tuts/boite_de_projection/boite_de_projection005_en.html)

- [147] <http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/displacement/index-ang.html>
- [148] <http://www.ingiebee.com/Blendermania/Game%20Sprites.htm>
- [149] <http://www.ingiebee.com/Blendermania/Stencilling.htm>
- [150] <http://feeblemind.tuxfamily.org/dotclear/index.php/2005/02/27/>
  - 16-didacticiel-usage-des-masques-de-textures---tutorial-using-texture-stencils
- [151] <http://www.youtube.com/watch?v=mVhhZ2KRr4Q>
- [152] <http://www.elysiun.com/forum/viewtopic.php?t=25918>
- [153] [http://biorust.com/index.php?page=tutorial\\_detail&tutid=85](http://biorust.com/index.php?page=tutorial_detail&tutid=85)
- [154] [http://barney.gonzaga.edu/~amoore1/uv\\_mapping\\_project/](http://barney.gonzaga.edu/~amoore1/uv_mapping_project/)
- [155] <http://otothegardener.free.fr/tutorials/LittleOTO/littleoto3/littleoto3.htm>
- [156] <http://otothegardener.free.fr/tutorials/Armure/anonce/anonce.htm>
- [157] [http://www.planetannihilation.com/terragen/tut\\_texturing.shtml](http://www.planetannihilation.com/terragen/tut_texturing.shtml)
- [158] <http://users.pandora.be/blendix/blender/uv/>
- [159] <http://membres.lycos.fr/bobois/Tuts/uvmapping/uvmapping01.html>
- [160] [http://www.3dtotal.com/team/Tutorials/swords&daggers/swords&daggers\\_01.asp](http://www.3dtotal.com/team/Tutorials/swords&daggers/swords&daggers_01.asp)
- [161] <http://www.raivestudios.com/tutorials/blender/uvmapping/>
- [162] <http://www.raivestudios.com/tutorials/blender/softwarebox/>
- [163] <http://www.youtube.com/watch?v=Eljv69HOLR4>
- [164] <http://www.telusplanet.net/public/kugyelka/blender/tutorials/txmesh/txmesh.htm>
- [165] <http://www.elysiun.com/forum/viewtopic.php?p=455683#455683>
- [166] <http://www.cogfilms.com/tutorials.html>
- [167] <http://www.youtube.com/watch?v=-tM1spWQ5aw>
- [168] <http://www.androidblues.com/howto.html>
- [169] <http://div.dyndns.org/EK/tutorial/texture/>
- [170] <http://div.dyndns.org/EK/tutorial/gritty/>
- [171] [http://www.planetquake.com/polycount/resources/general/tutorials/HitmanDaz\\_Tut01/DigitalPainting.shtml](http://www.planetquake.com/polycount/resources/general/tutorials/HitmanDaz_Tut01/DigitalPainting.shtml)
- [172] [http://www.btinternet.com/~i.nation/tutorials/tutorial\\_00.htm](http://www.btinternet.com/~i.nation/tutorials/tutorial_00.htm)
- [173] <http://www.3drender.com/light/EqTutorial/tiling.htm>
- [174] <http://www.kandsdesign.com/kim/eyemap-tut.html>
- [175] <http://www.geocities.com/SiliconValley/Haven/2470/txtrtut/txtrtut.html>
- [176] <http://www.planetquake.com/pandemonium/html/skintips.htm>
- [177] <http://www.planetquake.com/pandemonium/html/skintut3.htm>
- [178] <http://www.planetquake.com/pandemonium/html/skintut4.htm>
- [179] <http://otothecleaner.free.fr/tutorials/GimpTexture/gimptexture.htm>
- [180] <http://www.itchy-animation.co.uk/tutorials/light01.htm>
- [181] <http://wiki.yafray.org/bin/view.pl/UserDoc/GauravGuide>
- [182] <http://www.andrew-whitehurst.net/3point.html>
- [183] <http://blenderman.free.fr/tut/radiosity/uk/>
- [184] [http://download.blender.org/documentation/html/chapter\\_rendering\\_radiosity.html](http://download.blender.org/documentation/html/chapter_rendering_radiosity.html)
- [185] [http://www.blender3d.org/cms/Ramp\\_Shaders.348.0.html](http://www.blender3d.org/cms/Ramp_Shaders.348.0.html)
- [186] [http://jmsoler.free.fr/didacticiel/blender/tutor/lumiere\\_radios\\_en.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/lumiere_radios_en.htm)
- [187] <http://www.elysiun.com/forum/viewtopic.php?t=30265>
- [188] [http://dev.newmediaworx.com/johnnyb/bplanet/tut\\_shadows.htm](http://dev.newmediaworx.com/johnnyb/bplanet/tut_shadows.htm)
- [189] <http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/world/index-ang.html>
- [190] <http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/shadows/index-ang.html>
- [191] <http://feeblemind.tuxfamily.org/dotclear/index.php/2005/01/16/10-didacticiel-loccclusion-ambiant---tutorial-ambiant-occlusion>
- [192] <http://feeblemind.tuxfamily.org/dotclear/index.php/2005/01/02/9-didacticiel-les-aires-lumineuses---tutorial-area-lights>
- [193] <http://www.malefico3d.com.ar/tutor/dof-en.html>
- [194] <http://www.selleri.org/Blender/tuts/SoftLight.pdf>
- [195] <http://www-users.cs.umn.edu/~mein/blender/tutorials/dust/mist.html>
- [196] [http://www.katorlegaz.com/tutorials/basic\\_3d\\_lighting.php](http://www.katorlegaz.com/tutorials/basic_3d_lighting.php)
- [197] <http://www.katorlegaz.com/index.php?a=article&display=1078185600.php>
- [198] <http://www.centralsource.com/blender/flares/>
- [199] <http://free.hostdepartment.com/a/aner/TUTORIAL.html>
- [200] [http://www.geocities.com/sound\\_man\\_dave/caustics.htm](http://www.geocities.com/sound_man_dave/caustics.htm)
- [201] <http://www.youtube.com/watch?v=GbFvrilssNo>
- [202] <http://otothecleaner.free.fr/tutorials/Flor/flor.html>
- [203] <http://www.elysiun.com/forum/viewtopic.php?p=511798>
- [204] <http://kokcito.tk/rvk/ipo.html>

- [205] [http://www.blender.org/cms/How\\_Armatures\\_work.634.0.html](http://www.blender.org/cms/How_Armatures_work.634.0.html)
- [206] <http://www.elysiun.com/forum/viewtopic.php?t=49603>
- [207] <http://www.elysiun.com/forum/viewtopic.php?t=31839>
- [208] <http://www.elysiun.com/forum/viewtopic.php?t=19347>
- [209] <http://argoslabs.com/%7Emalefico/tutor/mecano-en.html>
- [210] <http://www.anycities.com/turbog/Tutorial1.html>
- [211] [http://www.users.bigpond.net.au/blendage/pages/tutorials/tut4/animatin\\_armatures.html](http://www.users.bigpond.net.au/blendage/pages/tutorials/tut4/animatin_armatures.html)
- [212] <http://mmaigrot.free.fr/didac-blender/ikas/ikas-eng/ik-intro.php>
- [213] <http://www.malefico3d.com.ar/tutor/mecano-en.html>
- [214] <http://www.elysiun.com/tutorials/animation/>
- [215] <http://www.users.bigpond.net.au/blendage/pages/tutorials/tut3/armatures.html>
- [216] <http://www.elysiun.com/forum/viewtopic.php?t=42310>
- [217] <http://www.enricovalenza.com/softb.html>
- [218] <http://kokcito.tk/tut/rig3.html>
- [219] <http://kokcito.tk/tut/rig1.html>
- [220] <http://kahuna.clayton.edu/~jbrooks/blender/tutorials/IKA/>
- [221] <http://www.blender3dclub.com/index.php?name=News&file=article&sid=21>
- [222] <http://www.animationforcreation.com/2008/10/lesson-10blenders-animation-tools.html>
- [223] <http://feeblemind.tuxfamily.org/dotclear/index.php/2005/03/08/>
  - 17-didacticiel-animations-simples-1ere-partie---tutorial-simple-animations-1st-part
- [224] [http://www.users.bigpond.net.au/blendage/pages/beginners\\_tutorials/tut2/Keyframeanimation.html](http://www.users.bigpond.net.au/blendage/pages/beginners_tutorials/tut2/Keyframeanimation.html)
- [225] <http://download.blender.org/documentation/html/c1585.html>
- [226] [http://dev.newmediaworx.com/johnnyb/bplanet/tut\\_charanim.htm](http://dev.newmediaworx.com/johnnyb/bplanet/tut_charanim.htm)
- [227] [http://rodri.aniguild.com/tuto\\_run/run\\_en.php](http://rodri.aniguild.com/tuto_run/run_en.php)
- [228] <http://www.meloware.com/blender/lipsync.htm>
- [229] [http://dev.newmediaworx.com/johnnyb/bplanet/tut\\_charanim2.htm](http://dev.newmediaworx.com/johnnyb/bplanet/tut_charanim2.htm)
- [230] <http://argoslabs.com/~malefico/tutor/walking.html>
- [231] <http://mmaigrot.free.fr/didac-blender/nla/eng/index.php>
- [232] <http://argoslabs.com/~malefico/tutor/nla-en.html>
- [233] [http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/animation\\_effects/didac2-ang.html](http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/animation_effects/didac2-ang.html)
- [234] [http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/animation\\_effects/didac1-ang.html](http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/animation_effects/didac1-ang.html)
- [235] [http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/animation\\_effects/index-ang.html](http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/animation_effects/index-ang.html)
- [236] <http://www.ingiebee.com/Blendermania/plane%20follow%20path.htm>
- [237] <http://www.ingiebee.com/tutorials/Path/theeth%20paths.htm>
- [238] [http://membres.lycos.fr/bobois/Tuts/animation\\_cyclique/animation\\_cyclique\\_01\\_ang.html](http://membres.lycos.fr/bobois/Tuts/animation_cyclique/animation_cyclique_01_ang.html)
- [239] <http://www.users.bigpond.net.au/blendage/pages/tutorials/tut7/colourchanging.html>
- [240] <http://www.fortuneicity.com/skyscraper/true/947/id13.htm>
- [241] <http://www.elysiun.com/forum/viewtopic.php?t=42840&highlight=change+camera>
- [242] <http://www.elysiun.com/forum/viewtopic.php?t=46181>
- [243] <http://www.blender3dclub.com/index.php?name=News&file=article&sid=33>
- [244] <http://www.blender3dclub.com/index.php?name=News&file=article&sid=32>
- [245] <http://www.blender3dclub.com/index.php?name=News&file=article&sid=24>
- [246] <http://www.animationforcreation.com/2008/10/lesson-11blender-particle-system.html>
- [247] [http://www.youtube.com/watch?v=4yIN9yjCP\\_4](http://www.youtube.com/watch?v=4yIN9yjCP_4)
- [248] <http://youtube.com/watch?v=M9aG0xBi9IM>
- [249] <http://www.youtube.com/watch?v=At3hIc-423c>
- [250] [http://blender3d.com/cms/Particle\\_interaction.349.0.html](http://blender3d.com/cms/Particle_interaction.349.0.html)
- [251] <http://www.elysiun.com/forum/viewtopic.php?t=41356>
- [252] [http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/animation\\_effects/didac4-ang.html](http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/animation_effects/didac4-ang.html)
- [253] [http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/animation\\_effects/didac3-ang.html](http://www.linuxgraphic.org/section3d/blender/pages/didacticiels/animation_effects/didac3-ang.html)
- [254] <http://www.geocities.com/blengine/fireplace1.html>
- [255] <http://www-users.cs.umn.edu/~mein/blender/tutorials/dust/particle.html>
- [256] <http://www.centralsource.com/blender/flames/>
- [257] <http://www.centralsource.com/blender/liquid/>
- [258] <http://www.centralsource.com/blender/explosions/>
- [259] <http://www.centralsource.com/blender/smoke/>
- [260] <http://feeblemind.tuxfamily.org/dotclear/index.php/2004/12/24/4-blender-faire-pleuvoir---making-rain>
- [261] <http://www.deviantart.com/view/27104504/>
- [262] <http://www.deviantart.com/view/27479896/>

- [263] <http://www.xedium3d.com/tutorials/fluid/>
- [264] [http://www.andrew-whitehurst.net/fx\\_light.html](http://www.andrew-whitehurst.net/fx_light.html)
- [265] [http://st102.startlogic.com/~blenderw/cms/index.php?option=com\\_content&task=view&id=212&Itemid=42](http://st102.startlogic.com/~blenderw/cms/index.php?option=com_content&task=view&id=212&Itemid=42)
- [266] <http://www.weirdhat.com/blender/compositing2/>
- [267] <http://www.blender3dclub.com/index.php?name=News&file=article&sid=19>
- [268] [http://youtube.com/watch?v=3kb\\_utKDz3o](http://youtube.com/watch?v=3kb_utKDz3o)
- [269] <http://wiki.blender.org/index.php/GameEngineDev>
- [270] <http://www.blender.org/forum/viewtopic.php?t=6409&sid=868ed3b00fd68157b704e87f0cc1684a>
- [271] <http://web.archive.org/web/20011207173733/http://www.blender.nl/gameBlenderDoc/book1.html>
- [272] <http://www.blender.org/education-help/tutorials/tutorial-folder/3d-walkthrough/>
- [273] <http://www.fortunecity.com/skyscraper/true/947/id16.htm>
- [274] <http://www.fortunecity.com/skyscraper/true/947/id17.htm>
- [275] <http://otothegardener.free.fr/tutorials/Flipper/flipper.htm>
- [276] <http://www.users.bigpond.net.au/blendage/pages/tutorials/tut8/realtimemarmatures.html>
- [277] <http://www.users.bigpond.net.au/blendage/pages/tutorials/tut6/maze.html>
- [278] [http://www.users.bigpond.net.au/blendage/pages/tutorials/tut10/adding\\_traps.html](http://www.users.bigpond.net.au/blendage/pages/tutorials/tut10/adding_traps.html)
- [279] [http://www.tutorialsforblender3d.com/Game\\_Engine/MouseLook/MouseLook\\_First\\_1.html](http://www.tutorialsforblender3d.com/Game_Engine/MouseLook/MouseLook_First_1.html)
- [280] [http://www.tutorialsforblender3d.com/Game\\_Engine/Viewports/Viewport\\_1.html](http://www.tutorialsforblender3d.com/Game_Engine/Viewports/Viewport_1.html)
- [281] <http://www.tutorialsforblender3d.com/GameFunctions/GameLogicPage1.html>
- [282] [http://www.tutorialsforblender3d.com/Game\\_Engine/Vehicle/Vehicle\\_1.html](http://www.tutorialsforblender3d.com/Game_Engine/Vehicle/Vehicle_1.html)
- [283] [http://www.sutabi.tk/timmeh/index.php?option=com\\_content&task=category&sectionid=4&id=20&Itemid=26](http://www.sutabi.tk/timmeh/index.php?option=com_content&task=category&sectionid=4&id=20&Itemid=26)
- [284] <http://www.dedalo-3d.com/index.php?filename=SXCOL/makehair/abstract.html>
- [285] [http://jmsoler.free.fr/util/blenderfile/fr/povanim\\_en.htm](http://jmsoler.free.fr/util/blenderfile/fr/povanim_en.htm)
- [286] [http://jmsoler.free.fr/didacticiel/blender/tutor/cpl\\_import\\_ai\\_en.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_import_ai_en.htm)
- [287] <http://www.thoro.de/portfolio/verschiedenes/3DNP.html>
- [288] <http://www.elysiun.com/forum/viewtopic.php?t=37400>
- [289] [http://www.blender3d.org/Education/index\\_old.php?sub=TutorialEmbedplugin](http://www.blender3d.org/Education/index_old.php?sub=TutorialEmbedplugin)
- [290] <http://www.selleri.org/Blender/scripts/text.html>
- [291] <http://www.hgb-leipzig.de/~daniel/blender/>
- [292] <http://www.centralsource.com/blender/dynamica/>
- [293] <http://www.blenderwars.com/tut.php?module=blendersaber>
- [294] [http://marief.soler.free.fr/Monsite/lsystem\\_en.htm](http://marief.soler.free.fr/Monsite/lsystem_en.htm)
- [295] <http://www.ingiebee.com/Blendermania/Genos%20Spring.htm>
- [296] <http://www.elysiun.com/tutorials.php?id=2>
- [297] [http://jmsoler.free.fr/didacticiel/blender/tutor/english/python\\_script00.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/english/python_script00.htm)
- [298] [http://jmsoler.free.fr/didacticiel/blender/tutor/english/python\\_script01.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/english/python_script01.htm)
- [299] [http://jmsoler.free.fr/didacticiel/blender/tutor/english/python\\_script02.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/english/python_script02.htm)
- [300] [http://jmsoler.free.fr/didacticiel/blender/tutor/english/python\\_script03.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/english/python_script03.htm)
- [301] [http://jmsoler.free.fr/didacticiel/blender/tutor/english/python\\_script04.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/english/python_script04.htm)
- [302] [http://jmsoler.free.fr/didacticiel/blender/tutor/english/python\\_script05.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/english/python_script05.htm)
- [303] [http://jmsoler.free.fr/didacticiel/blender/tutor/english/eng\\_scriptmirror.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/english/eng_scriptmirror.htm)
- [304] [http://jmsoler.free.fr/didacticiel/blender/tutor/cpl\\_curvesimport\\_en.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_curvesimport_en.htm)
- [305] [http://jmsoler.free.fr/didacticiel/blender/tutor/cpl\\_paths\\_import\\_en.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_paths_import_en.htm)
- [306] [http://www.dcmodding.com/main/index.php?option=com\\_docman&task=cat\\_view&Itemid=&gid=36&orderby=dmdatecounter&ascdesc=DESC](http://www.dcmodding.com/main/index.php?option=com_docman&task=cat_view&Itemid=&gid=36&orderby=dmdatecounter&ascdesc=DESC)
- [307] [http://jmsoler.free.fr/didacticiel/blender/tutor/cpl\\_meshexplosion\\_en.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_meshexplosion_en.htm)
- [308] [http://jmsoler.free.fr/didacticiel/blender/tutor/cpl\\_LOD\\_en.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_LOD_en.htm)
- [309] [http://jmsoler.free.fr/didacticiel/blender/tutor/python\\_wireshadows\\_en.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/python_wireshadows_en.htm)
- [310] [http://jmsoler.free.fr/didacticiel/blender/tutor/cpl\\_changerdecamera\\_en.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_changerdecamera_en.htm)
- [311] [http://www.dedalo-3d.com/index.php?filename=SXCOL/experiments/ss\\_scattering\\_python.html](http://www.dedalo-3d.com/index.php?filename=SXCOL/experiments/ss_scattering_python.html)
- [312] <http://feeblemind.tuxfamily.org/dotclear/index.php/2005/04/25/39-mh-tutorial-using-the-superficial-scattering-script>
- [313] <http://www.elysiun.com/forum/viewtopic.php?t=16217>
- [314] <http://www.elysiun.com/forum/viewtopic.php?t=32715>
- [315] <http://www.redrival.com/scorpius/blender-plugins.htm>
- [316] <http://www3.sympatico.ca/emilio.aguirre/hornextrude.html>
- [317] [http://blender.formworks.co.nz/index.pl?p=2z/batch\\_stl.html](http://blender.formworks.co.nz/index.pl?p=2z/batch_stl.html)
- [318] <http://www.elysiun.com/forum/viewtopic.php?t=45576>
- [319] <http://www.elysiun.com/forum/viewtopic.php?t=35355>
- [320] <http://www.blender3dclub.com/index.php?name=News&file=article&sid=54>

- [321] <http://www.yafra.org>
- [322] <http://blender.org/modules/verse/>
- [323] <http://pythonide.stani.be>
- [324] <http://www.wings3d.com/>
- [325] <http://users.pandora.be/blendix/verse/old/demo.html>
- [326] <http://www.softlab.ece.ntua.gr/~jpanta/Graphics/Kerkythea/>
- [327] <http://www.equinox3d.com>
- [328] <http://mywebpages.comcast.net/jmandmc/uvmap/uvmap.html>
- [329] [http://jmsoler.free.fr/didacticiel/blender/tutor/cpl\\_import\\_svg\\_en.htm](http://jmsoler.free.fr/didacticiel/blender/tutor/cpl_import_svg_en.htm)
- [330] [http://www.blender3d.org/Education/index\\_old.php?sub=TutorialPowerpoint](http://www.blender3d.org/Education/index_old.php?sub=TutorialPowerpoint)
- [331] <http://users.skynet.be/sky33676/ter2blend1.html>
- [332] [http://membres.lycos.fr/bobois/Tuts/traitement\\_par\\_lot/traitement\\_par\\_lot\\_ang.html](http://membres.lycos.fr/bobois/Tuts/traitement_par_lot/traitement_par_lot_ang.html)
- [333] <http://www.centralsource.com/blender/bvh/>
- [334] <http://www.geocities.com/blenderlab/tutorial1.html>
- [335] [http://www.blender3d.org/Education/index\\_old.php?sub=TutorialVrml](http://www.blender3d.org/Education/index_old.php?sub=TutorialVrml)
- [336] [http://www.katorlegaz.com/index.php?filename=/articles/blender/Quicktime\\_VR.php](http://www.katorlegaz.com/index.php?filename=/articles/blender/Quicktime_VR.php)
- [337] [http://www.katorlegaz.com/index.php?filename=/articles/blender/Python\\_and\\_OS\\_X.php](http://www.katorlegaz.com/index.php?filename=/articles/blender/Python_and_OS_X.php)
- [338] <http://www.flippyneck.com/temp/NPR.htm>
- [339] [http://www.dcmodding.com/main/index.php?option=com\\_smf&Itemid=79&topic=180.0](http://www.dcmodding.com/main/index.php?option=com_smf&Itemid=79&topic=180.0)
- [340] <http://neo5842.starlihost.info/my%20video%20tuts%2001.html>
- [341] [http://www.imvu.com/catalog/web\\_landing.php?p=power&userId=10209797](http://www.imvu.com/catalog/web_landing.php?p=power&userId=10209797)
- [342] <http://burp.boinc.dk/>
- [343] <http://renderworld.futureware.at/>
- [344] [http://www.katorlegaz.com/index.php?filename=/articles/blender/Using\\_Blender\\_with\\_Xgrid.php](http://www.katorlegaz.com/index.php?filename=/articles/blender/Using_Blender_with_Xgrid.php)
- [345] <http://geocities.com/tronovan3d/>
- [346] <http://www.powua.com/>
- [347] <http://www.tacc.utexas.edu/~cburns/lightray/lightray.php>
- [348] <http://www.toxicengine.org/>
- [349] <http://blenderunderground.com/video-tutorial-list/>
- [350] <http://www.blender.org/tutorials-help/video-tutorials/>
- [351] [http://www.youtube.com/profile\\_videos?user=super3boy](http://www.youtube.com/profile_videos?user=super3boy)
- [352] <http://www.ibiblio.org/bvidtute/>
- [353] <http://www.blendernation.com/2006/03/01/videotutorials-blender-3d-workshop/>
- [354] <http://www.animationforcreation.comFree>
- [355] [http://www.museum.state.il.us/ismdeps/library/linuxguides/blender/blender\\_faq\\_0.html](http://www.museum.state.il.us/ismdeps/library/linuxguides/blender/blender_faq_0.html)
- [356] [http://dev.newmediaworx.com/johnnyb/bplanet/tut\\_blendertips.htm](http://dev.newmediaworx.com/johnnyb/bplanet/tut_blendertips.htm)
- [357] <http://www.free-webspace.biz/sonix/Cars/Blender234CarMaterialLibraryR1.html>
- [358] [http://katorlegaz.com/3d\\_models/index.php/](http://katorlegaz.com/3d_models/index.php/)
- [359] <http://www.3d.sk>
- [360] <http://www.med.harvard.edu/AANLIB/cases/caseNA/pb9.htm>
- [361] <http://textures.z7server.com/>
- [362] <http://www.mayang.com/textures/index.htm>
- [363] [http://www.art.net/~jeremy/photo/public\\_texture\\_frameset.html](http://www.art.net/~jeremy/photo/public_texture_frameset.html)
- [364] <http://www.imageafter.com/>
- [365] <http://digitalcraftsman.com/textureBin/textureBin.htm>
- [366] <http://www.elysiun.com/forum/viewtopic.php?t=24738>
- [367] <http://genedecht.awn.com/>
- [368] <http://home.nordwest.net/Adger/tips/tip077.html>
- [369] <http://centralsource.com/blenderart/index.php>
- [370] <http://www.rondexter.com/>
- [371] <http://www.atpm.com/9.07/design.shtml>
- [372] [http://orange.blender.org/cms/The\\_Movie.555.0.html](http://orange.blender.org/cms/The_Movie.555.0.html)
- [373] <http://nano.prods.free.fr/>
- [374] <http://www.eofw.org/index.php>
- [375] <http://blenderartists.org/cms/index.php?id=38>
- [376] <http://www.zoo-logique.org/3D.Blender/index.php3?zoo=dif>
- [377] [http://membres.lycos.fr/bobois/Liens\\_Links/big\\_list.html](http://membres.lycos.fr/bobois/Liens_Links/big_list.html)
- [378] <http://www.davidjarvis.ca/blender/>
- [379] <http://blendercourse.com/English.aspx>

- [380] <http://www.darkscarab.com/tutorials>
- [381] <http://www.blender3dclub.com>
- [382] [http://www.katorlegaz.com/index.php?filename=links/blender\\_3D.php](http://www.katorlegaz.com/index.php?filename=links/blender_3D.php)
- [383] <http://www.frappr.com/blenderheadsaroundtheglobe>
- [384] <http://s12.yousendit.com/d.aspx?id=14ZVYHLTFOAK2V3YTCWU38RQ3>
- [385] [http://www.apollux-designs.com/BA-Magazine/blenderart\\_mag-nov-05.pdf.zip](http://www.apollux-designs.com/BA-Magazine/blenderart_mag-nov-05.pdf.zip)
- [386] <http://blendertestbuilds.de/index.php?dir=Blenderart/20051115/>
- [387] [http://www.intellidesign.org/blenderart\\_mag/](http://www.intellidesign.org/blenderart_mag/)
- [388] <http://www.statikonline.com/Blender/>
- [389] <http://www.Blender3DTutorials.com>
- [390] <http://www.cdschools.org/54223045235521/blank/browse.asp?A=383&BMDRN=2000&BCOB=0&C=55205>
- [391] <http://www.blendermasters.com>
- [392] <http://www.youtube.com/watch?v=fsHW7J9dTNE>
- [393] <http://www.elysiun.com/forum/viewtopic.php?t=13380>

# Article Sources and Contributors

**Blender 3D: Noob to Pro** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1728241> *Contributors:* 01mf02, 1983, 492star, AM088, AdRiley, Admijr, Adolfhohm, Adrignola, Aidan.Sullivan, Alabandit, Alerante, Allefant, AndyD, Aragan Jarosalam, Arch dude (usurped), Argento, Astronouth7303, AttilaMH, AverieZen, Az1568, BANZ111, BeHE, Ben Russell, Big Bill57, Bmud, BobbyMatt, Bootaleg, Boyage, Bullercruz1, Camel, Camp, Cbh, Charybdis, Chelseafan528, Chw333, Cleobod, Crawmm, Crouch, Cürps, Dallas1278, Damyeon, Dan-aka-jack, Dan13, DanAvner, DanMan, Darklama, Darkone, David Osborn, DavidCary, Derbeth, Dirk Gently, Dobz116, Dokuro, Dragontamer, DuLithgow, EatMyShortz, Eb264, Electro, Embri, EvanCarroll, Fr, Freakazo v2.1, Furrykef, Gabio, Gabriel, GeForce3, Geekboy2000, Geocacher nemesis, Goeland86, Gogreen, Goodgirl4, Grapw the Giant, Greensweater, Hagindaz, Halley, Harry107, Herbythyme, Hrakaro, JstLmdAloAboutStargate@WTP, Ideasman42, Ikarsik, Iluvblender, Iodrome, Jackechan, James.Denholm, Jamin3D, Jawboot, JLee, Jguk, Jmartens, Joe042293, Jogai, Jomegat, Jonon, Joystick Game, Joystick Tutorial, Justthisguy, Kellen, Kernigh, Klowner, Koos Jol, LION, Lazy-lump, Leinad13, LightningIsMyName, LokiClock, MSK61, Mach1723, Magicink, Malaz, ManuelGR, Mastermind 007, Mata\_Halldin, Mdd4696, Michaelx153, Mike.lifeguard, Milky, Miyazaki, Mkn, Mooasha, Moreejit, Morerunes, Morrolan, Mrkat, NCS Digiboy, Neo5842, Neoptolemus, Nick.anderegg, Nivix, Noob1230, Notmyopinion, Nuclearpidgeon, Null Point, OekeIWm, Omegatron, Oracleoftruth, Orbisontrum, Ordigdug, Osxrules, Panic2k4, ParallaxTZ, Pi zero, Pietrodi, Pld, Popski, Qrc, Quantum Anomaly, Radialronnie, Ramac, Randolph Richardson, Read-write-services, Recent Runes, Remi, Ricvelozo, Rm5248, Robert Horning, Robert Vafik, RobinH, Rocastelo, Romainbehar, Rosver, Rwt, Sailer, Saysaknow, SchmS, Scottpledger, Serge, SergeantOreo, Sigma 7, Skyguy, Snarius, Soeb, SoylentGreen, Soylentgreen, Spiderworm, StarryTG, Sumoncil, Super3boy, Sureshgn, Surphaze, Swift, TBOL3, Tannersf, Tedka, Thecheet123, ThomasShepard, Tjb0607, Tom soderlund, Trident, Tualha, Twentytortures, TwistedWeasel, Tyram monkey, Uncle G, Vault, Venar303, Wavez, Wesc, Webaware, WeirdHat, Wentam, Whiteknight, Whitepaw, Wikibooks is Communism, Wwwolf, Wynnnmc, Xania, Xemkis12, Xerol, Yosofun, Zakanfien, ZeroOne, Zeviion, Zilch, 667 anonymous edits

**Blender 3D: Noob to Pro/Beginner Tutorials** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1647255> *Contributors:* Acsteitz, Adrignola, Akhram, Argento, Buckbeak, Bullercruz1, Crouch, Danhash, Derbeth, Lordmyth, Mdd4696, PhilippWeissenbacher, Popski, Qrc, Shellrie, Spiderworm, Tedka, ThomasShepard, Woodsjay, Zoohouse, 74 anonymous edits

**Blender 3D: Noob to Pro/Unit 1: Knowing Before Making** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1719837> *Contributors:* Bullercruz1, Electro, John.delannoy, M.Nießen, Metahop, Nanobug, Sigma 7, SmileyCactus, 5 anonymous edits

**Blender 3D: Noob to Pro/Avoiding Headaches Early On: Good 3D Modeling Practices** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617073> *Contributors:* Adrignola, Bullercruz1, Fluffgar, 4 anonymous edits

**Blender 3D: Noob to Pro/Know What You're Doing** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1718693> *Contributors:* Bullercruz1, Electro, John.delannoy, Sigma 7, 25 anonymous edits

**Blender 3D: Noob to Pro/Cheat the 3D** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1719866> *Contributors:* Adrignola, Bullercruz1, Nanobug, 27 anonymous edits

**Blender 3D: Noob to Pro/Performance vs. Quality** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1707450> *Contributors:* Bullercruz1, Velociostrich, 6 anonymous edits

**Blender 3D: Noob to Pro/Introduction** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1492619> *Contributors:* BANZ111, Bullercruz1, Electro

**Blender 3D: Noob to Pro/Understanding coordinates** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1366832> *Contributors:* BANZ111

**Blender 3D: Noob to Pro/Understanding Coordinates** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1492770> *Contributors:* BANZ111, Electro, Null Point, SoylentGreen, 4 anonymous edits

**Blender 3D: Noob to Pro/Orthographic Projections** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1501213> *Contributors:* BANZ111, Electro, Ghiraddje

**Blender 3D: Noob to Pro/Isometric Projections** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1492628> *Contributors:* BANZ111, Electro

**Blender 3D: Noob to Pro/Perspective Views** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1492629> *Contributors:* BANZ111, Electro, SoylentGreen, Zilch, 3 anonymous edits

**Blender 3D: Noob to Pro/Coordinate Spaces in Blender** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1689546> *Contributors:* Bullercruz1, Electro, SoylentGreen, Squaremo, Tigerstickfigure, 16 anonymous edits

**Blender 3D: Noob to Pro/Tutorial Syntax** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1648015> *Contributors:* Acsteitz, Adrignola, Bullercruz1, Casimir, Clemens, Crouch, DavidJaquay, Dennis G. Jerz, Derbeth, EvanCarroll, JLee, Jessemerriiman, Jguk, Justin6300, Loplin, Mecalith, N comme Nul, Ordigdug, Polluks, Popski, Qrc, Rabenschwinge, Romainbehar, Rylz, Slady, Spiderworm, ThomasShepard, Tyaedalis, 98 anonymous edits

**Blender 3D: Noob to Pro/Non-standard equipment** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1710177> *Contributors:* Adrignola, EvanCarroll, JLee, Jomegat, Mxn, Spartacus3d, Tyaedalis, Velociostrich, 22 anonymous edits

**Blender 3D: Noob to Pro/Operating System specific notes** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616964> *Contributors:* Adrignola, Brian 252, Dlr, EvanCarroll, Soeb, Spartacus3d, Tyaedalis, 13 anonymous edits

**Blender 3D: Noob to Pro/Blender Interface** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616481> *Contributors:* Adrignola, Crouch, Hagindaz, N comme Nul, Popski, Qrc, Soeb, Spiderworm, Tedesson, ZeroOne, Zoohouse, 20 anonymous edits

**Blender 3D: Noob to Pro/Blender Windowing System** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1705524> *Contributors:* AM088, Adrignola, Andrewtrefethen, Argento, Az1568, Crouch, DariusWiles, DavidJaquay, Derbeth, Diblidabidu, Edmetric, El imp, Filaminae, Furrykef, Gavimusic, Geigerho, Geocacher nemesis, Gijs.peek, Goncalopp, Ideasman42, Iluvica, Jamesnelli, Jguk, Lambda, Lynx7725, MacPhyle, Mdd4696, MechaBlue, Null Point, Oracleoftruth, Pedro Fonini, Popski, Priit K., Qrc, Rjh009, Rm5248, Ruyven, Soeb, Spiderworm, Sterre, Theslandlord, Tualha, UsedHONDA, Winterbourne, Xlinuxkdex, ZeroOne, 262 anonymous edits

**Blender 3D: Noob to Pro/The User Preferences Window** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1686789> *Contributors:* 5o, Bekenn, Blenderizer, Hxcan, Jomegat, Lazy-lump, Ramac, Soeb, Spiderworm, Yhevhe, 23 anonymous edits

**Blender 3D: Noob to Pro/The Buttons Window** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1686793> *Contributors:* Adrignola, Arch dude (usurped), Crouch, Geocacher nemesis, Imaginationac, JiBril, MechaBlue, Mouse among men, Popski, Qrc, Savage Beef, Snarius, Spiderworm, TBOL3, Tedesson, ThomasShepard, Tualha, Webaware, Withinfocus, Yes Man, ZeroOne, 78 anonymous edits

**Blender 3D: Noob to Pro/The 3D Viewport Window** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1699595> *Contributors:* 2DCube, AM088, Acsteitz, Adrignola, Andrewtrefethen, Atarun, Banzai, Bekenn, Borisblue, Cale, Crouch, Dagnode, Dandellion, DariusWiles, Darkfire, David Cochran, Dougc, EN-Eagle 101, Entera voss, EvanCarroll, FriedChiliDog1, Gairlochan, Garmon Surrix, Herbythyme, Itsimike, Ivucica, Jackechan, Jeremy, Jollyroger, Jonon, Lazy-lump, Marty one, Michael Coupland, Odie5533, Popski, Qrc, Rjh009, Sestina, Sevczik, Soeb, Spiderworm, ThomasShepard, ThomasTenCate, WesleyJL, Wgoelkel, YesPoX, Zbrown, 350 anonymous edits

**Blender 3D: Noob to Pro/Other Windows** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616968> *Contributors:* Adrignola, Allefant, Crouch, Cspurrier, Darkfire, Miken700, Popski, Qrc, SoylentGreen, Spiderworm, 19 anonymous edits

**Blender 3D: Noob to Pro/Learn to Model** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1718961> *Contributors:* Adrignola, Az1568, Bullercruz1, Crouch, Darkfire, Herbythyme, MichaelFrey, Miken700, Null Point, PhilippWeissenbacher, Popski, Qrc, Slady, Someguywithnothingtodo, Spiderworm, WolfeBay, ZeroOne, Zilch, 18 anonymous edits

**Blender 3D: Noob to Pro/Introducing Meshes** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1727697> *Contributors:* AsalaW, Bullercruz1, Capitancornelio, Cyganski, Daworm, Deadguy, Jimbley, Megaman en m, Sigma 7, Squaremo, Starr, Thespeedylife, Timfrost, Tundra010, 39 anonymous edits

**Blender 3D: Noob to Pro/Beginning Tips** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1720053> *Contributors:* AdRiley, Adrignola, Allefant, Asrekdal, Az1568, Bullercruz1, Capitancornelio, Christian.mercat, Dahak, DariusWiles, Darkfire, EvanCarroll, Fr, Hagindaz, Iamunknown, Infiner, Ivucica, Jguk, Jomegat, Lit-urly, Loplin, Ma5abre, Marco Guardigli, NitishP, Pld, PuzzleMeister, Romainbehar, Sevczik, Slady, Squaremo, Tannersf, TimDumol, Tualha, Tyaedalis, Woodstockeet, 170 anonymous edits

**Blender 3D: Noob to Pro/Quickie Model** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1709998> *Contributors:* Adrignola, Andrewtrefethen, Argento, Beetlemaniac, Cale, Crouch, Darkfire, Ebswift, Geocachernemesis, Ivucica, JoelGodin, Jomegat, Marco Guardigli, Mdd4696, Obiesel, Oracleoftruth, Popski, Qrc, Rdp, Romainbehar, Siggimund, Slady, Soeb, Spiderworm, Squaremo, ThomasTenCate, ZeroOne, Zilch, Zoohouse, 109 anonymous edits

**Blender 3D: Noob to Pro/Quickie Render** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1686574> *Contributors:* Adrignola, Argento, Astronouth7303, Blondandy, Cale, Coffeeparis, Darkfire, Ebswift, Gumba gumba, JoelGodin, Jomegat, Michaelnelson, Nowic, Null Point, Popski, Qrc, Romainbehar, Rylz, SergeantOreo, Soeb, Spiderworm, Spongebob88, Super Dave, ThomasTenCate, Thzi, TimDumol, Tualha, ZeroOne, 76 anonymous edits

**Blender 3D: Noob to Pro/Modeling a Simple Person** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1704303> *Contributors:* 1983, AM088, AdRiley, Adrignola, Allefant, Aphax, Argav, Atimmer, Az1568, Bijee, Bonapon, Borisblue, Cale, CaptBobo, CharlesLambert, Citizen428, Crouch, Darkfire, DavidBaird, Ditw0101, DuLithgow, Ebswift, Geocachernemesis, GeordieMcBain, Gilgamesh007, HWV258, Herbythyme, IamInnocent, Ivucica, James.Denholm, Jeromerobert, Jguk, JoelGodin, Jomegat, Jyril, Kernigh, Lebigot, Mats Halldin, Mdd4696, Mikahl, Mike.lifeguard, Mosquito, Mystictim, N comme Nul, Nessup, NitishP, Null Point, Panic2k4, Pedro Fonini, Popski, Pquijal, Pwd, QuiteUnusual, Randolph Richardson, Rdp, Reece, Rm5248, Robsterberling, Romainbehar, Rreutel, Rylz, Schultz.Ryan, Shinmoos, Sigma 7, Slady, Spiderworm, Squaremo, Tayewiki, TimDumol, Wikidieter, Wimjongman, Xania, YesPoX, Yosofun, ZeroOne, 329 anonymous edits

**Blender 3D: Noob to Pro/Detailing Your Simple Person 1** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1687997> *Contributors:* AM088, Adrignola, Argento, Astronouth7303, Atimmer, Az1568, Bacon, Bronzed bison, Crouch, DavidLaquaay, Dirk gently, DuLithgow, Ebswift, FishFace, Gumba gumba, Hippopotamuses, James.Denholm, Jguk, Lapinbleu, Lebigot, Maveric149, Mike.lifeguard, Mr Me, Phaedriel the Vandal, Rm5248, Sjc, Someguywithnothingtodo, Spiderworm, Withinfocus, Yosofun, 135 anonymous edits

**Blender 3D: Noob to Pro/Detailing Your Simple Person 2** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1648709> *Contributors:* AM088, Adrignola, Argento, Bgks, Bonapon, Burnip, CodeSnorter, Cosworth, Crouch, Derbeth, Dessydes, EatMyShortz, Ebswift, Enselic, Fede.Campagna, Geocachernemesis, Geri, Gilgamesh007, Happy2pester, Hippopotamuses, Jguk, Joda, Jomegat, Jtoni, Matt Howard, Mdd4696, Mercator, Mike.lifeguard, Mikeg8r, N comme Nul, Nick.anderegg, Null Point, Oracleoftruth, Panic2k4, Pikpus, Pinkmouse, Pootworm, Popski, QuiteUnusual, Ramac, ReKast, Reece, Reolithion, Rheinhardt2, Rm5248, Romainbehar, Satrbe101, Simeon, Simon W, SixFt12, Spiderworm, Spiffyandy, Tannersf, Thunder, Twofingers, Webaware, Winters, Yosofun, ZeroOne, Zilch, 273 anonymous edits

**Blender 3D: Noob to Pro/Creating a Simple Hat** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1697527> *Contributors:* 3DSki, Adrignola, Argento, Burnip, CodeSnorter, Crouch, Derbeth, Digitalartist, Ebswift, Esben, Fede.Campagna, Gilgamesh007, Hagindaz, Jackechan, Javac, Jguk, LethalReflex, Lionelb, Michaelnelson, Miga, Mike.lifeguard, Mimooh, Mr Speaker, N comme Nul, Oracleoftruth, Pinzo, Popski, Quantufinity@gmail.com, Romainbehar, Sledz41, Spiderworm, Swift, Thunder, TiMoch, Wwwwolf, YesPoX, Yosofun, ZeroOne, 193 anonymous edits

**Blender 3D: Noob to Pro/Putting Hat on Person** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1656768> *Contributors:* Adrignola, Astronouth7303, Bullercruz1, Dalhamir, Entera voss, Jguk, Mdd4696, Miga, N comme Nul, Nick.anderegg, Oracleoftruth, Phonuz, Popski, SatanClaus, Socratesone, Spiderworm, Twofingers, Winters, Withinfocus, Yosofun, Zilch, 94 anonymous edits

**Blender 3D: Noob to Pro/Bones** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1703465> *Contributors:* Adrignola, AndrewBuck, Argav, Bullercruz1, Derbeth, Dogwynn, Jake Scordato, Jguk, Jmartens, Lordelph, Mikenube, Mortennobel, Niks1024, Oracleoftruth, Rm5248, Root, SchmS, SixFt12, ThomasB, 58 anonymous edits

**Blender 3D: Noob to Pro/Mountains Out Of Molehills** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1696310> *Contributors:* AM088, Adrignola, Argav, Ascen, Beetlemaniac, Blueskies, Bonapon, Bullercruz1, Camel, Darkfire, Daworm, Derbeth, Fede.Campagna, Hendric, Iamunknow, Jackechan, Joeblends, Khono, Kindeller, Lirtferk, Manx61, Mdd4696, Megaloman, Moreejt, N comme Nul, N00b Henry, Neko18, Newbienmiss, Obiesel, Pedro Fonini, QuiteUnusual, Radiat-r, Rajeshja, Skeep, Spiderworm, Spongebob88, Tannersf, ThomasTenCate, Tpower, Twofingers, Valid User, ZeroOne, Zilch, 226 anonymous edits

**Blender 3D: Noob to Pro/Modeling a Gingerbread Man** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1688314> *Contributors:* 80063r, AM088, Adrignola, Argento, Astronouth7303, Bonapon, Brylie, Bullercruz1, Camp, DDD3x, Danielsimlind, Darkone, Firecentaur, Greggreg, Hagindaz, Jomegat, Khono, Knight13, Majko02, Matthewr100, Mike.lifeguard, Milky, Moreejt, Mrwilley, N comme Nul, NathanStocks, Orbisonitrum, Popski, SatanClaus, SchmS, SergeantOreo, Shinjin, Spiderworm, Swap i, TBOL3, Tannersf, Thorack, Tyedalis, Yosefm, 135 anonymous edits

**Blender 3D: Noob to Pro/Penguins from spheres** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1726305> *Contributors:* Adamp, Adrignola, Argento, Ascen, Beginner, Boyage, Brandished, Cleobod, CodeSnorter, DRNZ, Domthedude001, Donlemay, Eruantalon, Esben, Fidoda, Funky weasel, Gilgamesh007, Herbythyme, Jawboot, Jguk, Jogai, Jomegat, Josellis, Koltimaro, Mast3rlinx, Mecalith, Mike.lifeguard, N comme Nul, Norleaf, Opa God, Orbisonitrum, Physicsduke, ProfoX, Reece, Rekov, Romainbehar, Shythinker, Spongebob88, TBOL3, Twofingers, 233 anonymous edits

**Blender 3D: Noob to Pro/Die Another Way** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1726241> *Contributors:* Adrignola, Antcodd, Argento, Arturotena, Bacon, Badalia, Benwulf, Boyage, Chelseafan528, Chris0917, Cleobod, Darkfire, Dbroguy, Donlemay, Dooglus, Epicor, Fede.Campagna, Fezz, Funky weasel, Herbythyme, Iamunknow, Intuitas, IqAndreas, Itinerantrader, Jguk, Khono, Kisubo, Krand, Lethargo, MSK61, Mdd4696, Milahu, Milky, Mochimo, Moreejt, Mouse is back, Mouzi, N comme Nul, Orbisonitrum, Osxrules, QuiteUnusual, Raicundi, Ramac, Reece, Root, Shythinker, Spongebob88, Tannersf, ThomasTenCate, Tommy g003, Toscic, Trezero, Xwriter, Yashnz, Yewbow, Zarathoustra, 160 anonymous edits

**Blender 3D: Noob to Pro/Die Easy 2** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1726310> *Contributors:* Adrignola, CamStone, Dmansouth, Fezz, MSK61, Mast3rlinx, Mecalith, Nk.warren77, Null Point, Romainbehar, ThomasTenCate, Wynnm, 35 anonymous edits

**Blender 3D: Noob to Pro/Model a Silver Goblet** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1715048> *Contributors:* Abedaa, Adrignola, Argav, Ascen, AverieZen, Az1568, Brylie, Camp, Chuey, Clone Dad, Dooglus, Eyeonus, Helderpc, Jguk, Laeleena, Lsmoura, MSK61, Nanodeath, Pedro Fonini, Prydain55, Ramac, RavenWhitehawk, Sandothebrate, ThomasTenCate, Wynnm, Yoshi1476, 82 anonymous edits

**Blender 3D: Noob to Pro/Model a Silver Goblet cylinder** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1645039> *Contributors:* Adrignola, Dooglus, MSK61, Ma5abre, Ramac, Wynnm, 13 anonymous edits

**Blender 3D: Noob to Pro/Simple Vehicle** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617002> *Contributors:* AM088, Adrignola, AverieZen, Derbeth, Dooglus, Jguk, Mdd4696, Mrfelis, Oracleoftruth, Pengo, Popski, Reptiles extintos, SchmS, Sean.flynn, Spiderworm, Withinfocus, ZeroOne, 32 anonymous edits

**Blender 3D: Noob to Pro/Simple Vehicle: Wheel** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1701746> *Contributors:* Adrignola, Argav, AverieZen, Chelseafan528, CodeSnorter, Eyeonus, Feureau, Fidoda, IqAndreas, MSK61, Ma5abre, Nerdabidmonkey, Nielmo, RavenWhitehawk, Richard 27, Spiderworm, Tannersf, ThomasTenCate, Xander9899, 108 anonymous edits

**Blender 3D: Noob to Pro/Simple Vehicle: Seat** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1679557> *Contributors:* Adrignola, AverieZen, Carpetsmoker, Feureau, Josellis, MSK61, Odalcer, Outbacksam34, Spiderworm, Yoshi1476, 39 anonymous edits

**Blender 3D: Noob to Pro/Simple Vehicle: Rocket Launcher** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1666358> *Contributors:* Adrignola, AverieZen, Biomeksensei, Eruantalon, Feureau, IqAndreas, KentTong, Mecalith, Mgiv, NODJKQR, Odalcer, Ramac, Sephlaira, Spiderworm, TBOL3, Yoshi1476, 21 anonymous edits

**Blender 3D: Noob to Pro/Simple Vehicle: Body** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1699081> *Contributors:* Adrignola, Alexwill84, AverieZen, Derbeth, Falconx1096, Henrybissonnette, Inventor, JCrawford, Luke321321, Newbie, PaulJC, Ramac, RavenWhitehawk, Samnsparky, Spellmaker, Spiderworm, 29 anonymous edits

**Blender 3D: Noob to Pro/Simple Vehicle: Some Assembly Required** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1719120> *Contributors:* 00o0ok, Adrignola, Ascen, AverieZen, DDD3x, Falconx1096, Mgiv, Radialronnie, Spiderworm, Telmer6, 24 anonymous edits

**Blender 3D: Noob to Pro/Modeling a 3D Parachute in Blender** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1724877> *Contributors:* Daworm, Null Point, Radialronnie, 8 anonymous edits

**Blender 3D: Noob to Pro/Model a Low Poly Head** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1718890> *Contributors:* Adrignola, DEAFCODE, Daworm, Jguk, Jomegat, MSK61, Radialronnie, Ravichandar84, Rm5248, Sandsmertz, Sumoncil, Tommciver, Wynnm, YMS, 20 anonymous edits

**Blender 3D: Noob to Pro/Building a House** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1718698> *Contributors:* Adrignola, Az1568, Brandished, Bruce89, Bullercruz1, CFrey, MSK61, Mast3rlinx, Neilwrightman, POZ, Ramac, Reece, Sandsmertz, Slush15player, SoylentGreen, Waximo, Wynnm, YMS, 35 anonymous edits

**Blender 3D: Noob to Pro/Pipe joints** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1723818> *Contributors:* Adrignola, Begeun, Camp, SoylentGreen, Vaakmeisster, WithAlligators, 9 anonymous edits

**Blender 3D: Noob to Pro/Creating Models With Photo Assistance** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1715537> *Contributors:* AM088, Adrignola, Alpha0, AverieZen, Badalia, Bullercruz1, Camp, Dmwick, DuLithgow, Efren, Goatofdeath, Hagindaz, Herraotic, Jguk, Lebigot, Lirtferk, Longsword, MSK61, Mdd4696, Moohasha, Moreejt, Mudpuddle, Norleaf, Null Point, Paperclip, Pedro Fonini, Ramac, Random9q, Sandsmertz, Skatanist, SoylentGreen, Spiderworm, Stefantalpalaru, Stormy, Twofingers, Tyaedalis, YMS, Yewbow, Zilch, 161 anonymous edits

**Blender 3D: Noob to Pro/2D Image (logo) to a 3D Model** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1723448> *Contributors:* Adrignola, Allefant, Astharoth, Az1568, Bullercruz1, Danielsimlind, Derbeth, Geocachernemesis, Goatofdeath, Gumanoid, Iamunknow, Jawboot, Jguk, Johno, Jshadias, Kieran, Lazy-lump, Mike.lifeguard, Mrfelis, Null Point, Popski, SchmS, Shadowimmage, Spellmaker, Spiderworm, Sterre, Sunny256, Tannersf, Yjvyas, Yoyotweak, 111 anonymous edits

**Blender 3D: Noob to Pro/2D Image (logo) to a 3D Model Part 2** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1624755> *Contributors:* Adrignola, Bullercruz1, Darkonc, Jguk, Mrfelis, SchmS, Shokuku, Tannersf, Thorack, Twofingers, 35 anonymous edits

**Blender 3D: Noob to Pro/Curve and Path Modeling** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616500> *Contributors:* Adrignola, Atallcostsky, Bullercruz1, Lazy-lump, Popski, SergeantOreo, Spiderworm, Tmandry, Zoohouse, 6 anonymous edits

**Blender 3D: Noob to Pro/Intro to Bezier Curves** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1697823> *Contributors:* Adrignola, Atallcostsky, IqAndreas, Lazy-lump, Neurotic, Spellmaker, 12 anonymous edits

**Blender 3D: Noob to Pro/Curves In 3d** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616501> *Contributors:* Adrignola, Embri, Geekboy2000, Lazy-lump, 1 anonymous edits

**Blender 3D: Noob to Pro/Deforming Meshes using the Curve Modifier** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1724144> *Contributors:* Bullercruz1, Embri, Encesil, Lazy-lump, Mike.lifeguard, Null Point, WiTr, 14 anonymous edits

**Blender 3D: Noob to Pro/Making Your Creation Smoother** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1661078> *Contributors:* Adrignola, Bullercruz1, Gairlochan, Null Point, Xemkis12, 16 anonymous edits

**Blender 3D: Noob to Pro/Materials and Textures** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616659> *Contributors:* Accurrent, Adrignola, Bullercruz1, Crouch, Geocachernemesis, Merrilee, Popski, SchmS, SoylentGreen, Spiderworm, 8 anonymous edits

**Blender 3D: Noob to Pro/Quickie Material** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1677139> *Contributors:* Adrignola, Az1568, Brinman, Crouch, Edr, Fezz, Gregggreg, Jonon, Knight13, Malvineous, Mdd4696, Merrilee, NullPtr, Pete xp, Popski, Qrc, Reverzitative, SchmS, Siesta, Sinistra D32, SoylentGreen, Spellmaker, Spiderworm, Uncle G, Wikibooks is Communism, 55 anonymous edits

**Blender 3D: Noob to Pro/Quickie Texture** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1723785> *Contributors:* 01mf02, AM088, Adrignola, Allefant, Bonapon, Brinman, Crouch, John Nagle, Jonon, Kenkaku, Malvineous, Nuclearpidgeon, NullPtr, Plasmasphere, Popski, SchmS, Siesta, SoylentGreen, Spiderworm, Xlinuxkdex, 26 anonymous edits

**Blender 3D: Noob to Pro/Procedural Textures** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616977> *Contributors:* AM088, Adrignola, Bullercruz1, Derbeth, Gregggreg, Jguk, Mfoxdogg, Nuclearpidgeon, SchmS, Soriyath, Spiderworm, 10 anonymous edits

**Blender 3D: Noob to Pro/Halo Materials** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1639035> *Contributors:* Adrignola, Nuclearpidgeon, SoylentGreen, 3 anonymous edits

**Blender 3D: Noob to Pro/Image Textures** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1638524> *Contributors:* Adrignola, Bullercruz1, MSK61, Rm5248, SchmS, SoylentGreen, Spiderworm, Tjb0607, 10 anonymous edits

**Blender 3D: Noob to Pro/Creating Basic Seawater** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1661063> *Contributors:* AM088, Accurrent, Adrignola, Bullercruz1, DanMan, Daworm, Dobz116, Iluvblender, Khan, Popski, SchmS, SoylentGreen, Spiderworm, Sterre, Tannersf, 16 anonymous edits

**Blender 3D: Noob to Pro/Mountains Out Of Molehills 2** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1686244> *Contributors:* AM088, Adrignola, Arcticfreeze83, Ask, AverieZen, BeRaw, Bullercruz1, Dobz116, SchmS, Spiderworm, 9 anonymous edits

**Blender 3D: Noob to Pro/Basic Carpet Texture** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1638060> *Contributors:* Adrignola, AverieZen, Bonapon, Bullercruz1, Dan13, Leinad13, Mochimo, Nick.anderegg, SchmS, SergeantOreo, 17 anonymous edits

**Blender 3D: Noob to Pro/The Rusty Ball** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1725577> *Contributors:* AM088, Adrignola, Argento, Baadtaste, Brylie, Bullercruz1, DanMan, Mochimo, SchmS, Soriyath, Spiderworm, WithAlligators, 19 anonymous edits

**Blender 3D: Noob to Pro/Procedural Eyeball** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1725698> *Contributors:* 00Anme00, Bonapon, Bullercruz1, Hagindaz, Jmartens, Jorge Morais, Lordelph, Marinalan, Radialronnie, Spellmaker, Tdunn7, WithAlligators, Wovenmoon, 17 anonymous edits

**Blender 3D: Noob to Pro/Creating Pixar-looking eyes in Blender** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1727096> *Contributors:* Adrignola, Az1568, Bach, Badalia, Bmarnz, Bullercruz1, Chinmay gautam, DanMan, Davidwilliams, Furyrek, Harkin, Hrakarow, Icky-wiki, Irakrakow, Jguk, Leedev, Kystoman, Mike.lifeguard, Mochimo, Null Point, PKHG, Pengo, Philipmac, QuiteUnusual, Rdp, Reverzitative, SergeantOreo, Tannersf, ThomasB, Thud, WeirdHat, WithAlligators, Yosofun, 81 anonymous edits

**Blender 3D: Noob to Pro/UV Map Basics** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1724074> *Contributors:* Adrignola, Anonymous Dissident, Arcticfreeze83, Argento, Arru, Az1568, Bullercruz1, DDD3x, HHBones, Irakrakow, Jamin3D, Jonon, LokiClock, LordOfer, Mico, Mohax, Nowic, Null Point, Popski, Remi, SchmS, SergeantOreo, Sestina, SoylentGreen, Spiderworm, TBOL3, YMS, ZeroOne, 128 anonymous edits

**Blender 3D: Noob to Pro/Realistic Eyes In Blender** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1648783> *Contributors:* Bullercruz1, Nick chang207, Null Point, Quantum Anomaly, ScaroDj, 29 anonymous edits

**Blender 3D: Noob to Pro/Light a Silver Goblet** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1727723> *Contributors:* Adrignola, AverieZen, Bonapon, Brinman, Bullercruz1, RavenWhitehawk, WithAlligators, 13 anonymous edits

**Blender 3D: Noob to Pro/Every Material Known to Man** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1722162> *Contributors:* AM088, Adrignola, Argento, Bullercruz1, Chinmay gautam, Dapoulter, Davidfg, Dragonkingme, Electrice, Geri, JaccoG, Jguk, Joe 042293, Jonon, Khan, Kitbluear, Mdd4696, Oracleoftruth, Popski, Rct36, Sandsmertz, Soylentgreen, Spiderworm, Super3boy, Vault, 38 anonymous edits

**Blender 3D: Noob to Pro/Beginning Lighting** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616478> *Contributors:* Adrignola, Bullercruz1, Crouch, ParallaxTZ, Popski, Spiderworm, 5 anonymous edits

**Blender 3D: Noob to Pro/Faked Gi with Blender internal** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1696016> *Contributors:* Adrignola, MidnightLightning, Null Point, Orfo, Radialronnie, 6 anonymous edits

**Blender 3D: Noob to Pro/Render Settings** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616985> *Contributors:* AdemoS, Adrignola, Argento, Digivation, Null Point, ParallaxTZ, Popski, Radialronnie, Rm5248, SergeantOreo, Soeb, 25 anonymous edits

**Blender 3D: Noob to Pro/Output Format Options** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616969> *Contributors:* Adrignola, Argento, Enochsdream, Jguk, ParallaxTZ, Rm5248, Rwst, Saysaknow, SergeantOreo, Soeb, 13 anonymous edits

**Blender 3D: Noob to Pro/OSA** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616963> *Contributors:* 01mf02, Adrignola, Crouch, Goeland86, Jguk, Jonon, ParallaxTZ, Popski, Saysaknow, Soeb, ZeroOne, 8 anonymous edits

**Blender 3D: Noob to Pro/Looking All Around - Panorama Settings** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1677580> *Contributors:* Adrignola, Allefant, Mike.lifeguard, Tarot, 7 anonymous edits

**Blender 3D: Noob to Pro/Other Important Render Options** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616967> *Contributors:* Adrignola, Allefant, Crouch, ParallaxTZ, Popski, Rm5248, Saysaknow, Tarot, Terrywallwork, Tmr232, Xlinuxkdex, 9 anonymous edits

**Blender 3D: Noob to Pro/Adding Lamps** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1689557> *Contributors:* AM088, Adrignola, Cspurrier, Lazy-lump, Mike.lifeguard, Null Point, Ordigdug, ParallaxTZ, Popski, SergeantOreo, Spellmaker, Spiderworm, Totoph64, 51 anonymous edits

**Blender 3D: Noob to Pro/Yafray Render Options** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617040> *Contributors:* Adrignola, Argento, Charybdis, Exo, Jguk, Popski, Soeb, Taye, Xiephoire, 44 anonymous edits

**Blender 3D: Noob to Pro/Stamping** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617100> *Contributors:* Adrignola, Tictoon, 5 anonymous edits

**Blender 3D: Noob to Pro/Basic Animation** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1644243> *Contributors:* Adrignola, AndyD, Crouch, EatMyShortz, Jclee, ParallaxTZ, Peteturtle, Popski, Rm5248, SergeantOreo, TBOL3, 30 anonymous edits

**Blender 3D: Noob to Pro/Basic Animation/ScreenLayout** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617074> *Contributors:* Adrignola, Brinman, Dkarpp, Jackechan, Peteturtle, TBOL3, 7 anonymous edits

**Blender 3D: Noob to Pro/Basic Animation/Lattice** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1654803> *Contributors:* Adrignola, AndyD, Brinman, Jackechan, Jguk, Jomegat, Peteturtle, Rm5248, 31 anonymous edits

**Blender 3D: Noob to Pro/Basic Animation/Bounce** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1698251> *Contributors:* Adrignola, AndrewBuck, AndyD, Brinman, Herbythyme, Jguk, Lionelb, Malaz, Miika22, Mike.lifeguard, Mstram, TBOL3, Teddygage, Xania, 31 anonymous edits

**Blender 3D: Noob to Pro/Creating Basic Water animation** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1726356> *Contributors:* AndrewBuck, Bonapon, Crystal Visions, Jomegat, Joncomp12, Malaz, Nick.anderegg, Null Point, Ramac, SergeantOreo, Soeb, Vaxine19, Woodstockeet, Zeviion, 39 anonymous edits

**Blender 3D: Noob to Pro/Flying Through A Canyon** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1695449> *Contributors:* AndrewBuck, Arcticfreeze83, Bonapon, Dogwynn, Electus, Funky weasel, GeniXPro, Null Point, Psychogears, Tosnic, 9 anonymous edits

**Blender 3D: Noob to Pro/Using the sequencer to compile frames into an animation** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1618936> *Contributors:* Mike.lifeguard, Null Point, Professional noob, Radialhronnie

**Blender 3D: Noob to Pro/Particle Systems** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1693838> *Contributors:* Adrignola, AndyD, Danthemango, Funky weasel, Jguk, MarkM, Null Point, ParallaxTZ, Romainbehar, SergeantOreo, SoylentGreen, TBOL3, 15 anonymous edits

**Blender 3D: Noob to Pro/Making Fire** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1648268> *Contributors:* Adrignola, AndyD, Chinmay gautam, Chw333, Derbeth, Dimzy Doodle, Electus, Funky weasel, Jguk, Jomegat, Michaelsobers, Mstram, ParallaxTZ, Romainbehar, SoylentGreen, Teddygage, Venar303, 57 anonymous edits

**Blender 3D: Noob to Pro/Furry** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1702323> *Contributors:* Adrignola, ScaroDj, SoylentGreen, 14 anonymous edits

**Blender 3D: Noob to Pro/Fireworks** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1683267> *Contributors:* 00Anne00, AndrewBuck, Der Künstler, Digitalcircuit36939, Mrmaginee2759, SoylentGreen, ZeroOne, 11 anonymous edits

**Blender 3D: Noob to Pro/Soft Body Animation** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1651725> *Contributors:* Adrignola, AndrewBuck, Arcticfreeze83, Argento, Atallcostsky, Jguk, Porlando, SoylentGreen, Tosnic, Venar303, ZeroOne, 23 anonymous edits

**Blender 3D: Noob to Pro/Simple Cloth Animation** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617015> *Contributors:* Adrignola, Argento, Bluincher, Hagindaz, Jguk, Malaz, Null Point, Scottpledger, Selfhiding, SergeantOreo, Super3boy, That.1.guy, Tosnic, Venar303, 30 anonymous edits

**Blender 3D: Noob to Pro/Soft Body with wind** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1646507> *Contributors:* Eb264, Null Point, Rukaru, ZeroOne, 4 anonymous edits

**Blender 3D: Noob to Pro/Your First Test** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617041> *Contributors:* 5o, Adrignola, Allefant, Hammurderer, Jomegat, Malaz, Null Point, Spiderworm, 34 anonymous edits

**Blender 3D: Noob to Pro/Platformer: Creation and Controls** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1150239> *Contributors:* LokiClock, Malaz, Neoptolemus, 1 anonymous edits

**Blender 3D: Noob to Pro/An aMAZEing game engine tutorial** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1711087> *Contributors:* AttilaMH, Electus, Kylstoman, LokiClock, Malaz, Sanekur, Seraphrevan, Thenub314, Thud, 48 anonymous edits

**Blender 3D: Noob to Pro/Platformer: Physics Fixes** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1425169> *Contributors:* LokiClock, Malaz, Null Point, 1 anonymous edits

**Blender 3D: Noob to Pro/Making exe** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1707504> *Contributors:* Arcticfreeze83, AttilaMH, Chelseafan528, DeltaSpeeds, Derbeth, Dogwynn, Freakazo v2.1, GeForce3, Jackred, Jguk, Kylstoman, Malaz, Muimi, Venar303, 20 anonymous edits

**Blender 3D: Noob to Pro/Build a skybox** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1724076> *Contributors:* Adrignola, Allefant, AttilaMH, Glome, Greensweater, Irakrakow, Jguk, Malaz, Null Point, 19 anonymous edits

**Blender 3D: Noob to Pro/Basic mouse pointer** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616476> *Contributors:* Adrignola, Atallcostsky, AttilaMH, Freakazo v2.1, Jguk, Justthisguy, Malaz, MrThreepwood, Null Point, 7 anonymous edits

**Blender 3D: Noob to Pro/Text in BGE** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617024> *Contributors:* Adrignola, AttilaMH, Bullercruz1, Null Point, 3 anonymous edits

**Blender 3D: Noob to Pro/Python Platformer: Creation** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1425187> *Contributors:* LokiClock, Null Point

**Blender 3D: Noob to Pro/Beginning Modeling Final Project** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616479> *Contributors:* Adrignola, Bullercruz1, Icky-wiki, Jguk, MSK61, Mattb112885, Popski, Rm5248, Sheridan, Spiderworm, Yiyvas, 15 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616409> *Contributors:* Adrignola, Allefant, Bullercruz1, Jguk, Joystick Tutorial, Popski, Quaddux, Remi0o, Van der Hoorn, 12 anonymous edits

**Blender 3D: Noob to Pro/Python Scripting** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1680532> *Contributors:* Adrignola, Allefant, Argento, Iluvblender, Popski, 9 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Introduction** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1707162> *Contributors:* Admijr, Adrignola, Allefant, Argento, Ggolem1, Ideasman42, Jguk, Justlost, Robstafarian, Xlinuxkdex, 11 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Export scripts** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1712099> *Contributors:* Admijr, Adrignola, Allefant, Arakade, Argento, AttilaMH, Darknessknight, David Osborn, Icky-wiki, Ideasman42, Igumaraesf, Jguk, KaiRyu, Lebigot, Levis501, Nicoladie, Rebol, Roxorz42, Sckoobs, 65 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Import scripts** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1672757> *Contributors:* Adrignola, Argento, David Osborn, Jguk, Jomegat, Rdp, 6 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Procedural object creation** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1717055> *Contributors:* Adrignola, Allefant, Argento, AttilaMH, David Osborn, Jguk, Johndb, LokiClock, Nanki, 8 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Scripts for modifying meshes** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616470> *Contributors:* Adrignola, Allefant, Jguk, Rdp, 4 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Python Scripting/Creating a GUI for your script** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1690912> *Contributors:* Admijr, Adrignola, Allefant, AttilaMH, Jguk, Professional noob, Xlinuxkdex, 12 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/index** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616454> *Contributors:* Adrignola, Bullercruz1, 3 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/Modeling** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616452> *Contributors:* Adrignola, Bullercruz1

**Blender 3D: Noob to Pro/HDRi** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1710488> *Contributors:* Adrignola, Argento, Bmud, Bullercruz1, Chelseafan528, Dan-aka-jack, ErikBongers, Maebe, QuiteUnusual, 15 anonymous edits

**Blender 3D: Noob to Pro/Creating a Light Probe** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1405300> *Contributors:* Allefant, Bmud, Bullercruz1, Nick.anderegg, 4 anonymous edits

**Blender 3D: Noob to Pro/Making Landscapes with heightmaps** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1686450> *Contributors:* Adrignola, Aidan.Sullivan, Alex.L33, Ascen, Bullercruz1, Chelseafan528, CodeSnorter, Demoathon, Ekriceyote, Jackechan, Jguk, Johnstarbird, Kirona, Lazy-lump, Lorhc, Noob.sks.dk, Pedro Fonini, QuiteUnusual, Root, Sandalle, ShadoCrytr, Ubai, UnknownGuy, Wikidieter, Xerol, 60 anonymous edits

**Blender 3D: Noob to Pro/How to Do Procedural Landscape Modeling** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1378101> *Contributors:* AndrewBuck, Ma5abre, Moohasha, Pedro Fonini, Roli, SergeantOreo, Sketlover, 38 anonymous edits

**Blender 3D: Noob to Pro/Landscape Modeling I: Basic Terrain** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1660641> *Contributors:* Adrignola, AndrewBuck, Bladerman, KentTong, Kindeller, Moohasha, QuiteUnusual, Rekov, Root, Zilla, 35 anonymous edits

**Blender 3D: Noob to Pro/Landscape Modeling II: Texture Stenciling** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616649> *Contributors:* Adrignola, Mike.lifeguard, Moohasha, UnknownGuy, 28 anonymous edits

**Blender 3D: Noob to Pro/Landscape Modeling III: Exporting as a Heightmap** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616650> *Contributors:* Adrignola, Bullercruz1, Moohasha, Ubai, 18 anonymous edits

**Blender 3D: Noob to Pro/Realistic Water using Fluid Sim and YafraY** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1646143> *Contributors:* Atallcostsky, Bullercruz1, Charybdis, Joncomp12, Remi, SergeantOreo, SoylentGreen, T.C.Fire, Vapor, 19 anonymous edits

**Blender 3D: Noob to Pro/BBB's Particles -- New 2.46 Particle System** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1454769> *Contributors:* Bullercruz1, Nick.anderegg, SoylentGreen, 1 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/Textures** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616453> *Contributors:* Adrignola, Bullercruz1, Nick.anderegg

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/Bump Mapping** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1508600> *Contributors:* Bullercruz1, SoylentGreen, 4 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/Normal Mapping** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1508602> *Contributors:* Bullercruz1, Nick.anderegg, SoylentGreen

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/Texture Normal Mapping** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1508604> *Contributors:* Bullercruz1, Nick.anderegg, SoylentGreen

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/Color Map Normal Mapping** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1508606> *Contributors:* Bullercruz1, Markharper80, SoylentGreen, 1 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Modeling/Light Mapping** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616451> *Contributors:* Adrignola, Bullercruz1

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/index** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616444> *Contributors:* Adrignola, Bullercruz1, Gabio, Jguk, 7 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Introduction** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616438> *Contributors:* Adhemar, Adrignola, Gabio, Jguk, Pamtango, Thinkingman, Zachary Murray, 6 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/index** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616434> *Contributors:* Adhemar, Adrignola, Gabio, Jguk, 1 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Armature/index** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616412> *Contributors:* Adhemar, Adrignola, Gabio, Jguk, Pamtango, 9 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Armature/object** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1686486> *Contributors:* Adrignola, Allefant, Gabio, Jguk, Jmartens, Pamtango, Super3boy, 22 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Armature/edit** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1631035> *Contributors:* Adrignola, Allefant, Gabio, Jguk, LokiClock, Madcello, Soylentgreen, 56 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Armature/pose** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616415> *Contributors:* Adrignola, Gabio, Jguk, Soylentgreen, 17 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Mesh/index** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616430> *Contributors:* Adrignola, Gabio, Jguk, 3 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Mesh/Amodif** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1637506> *Contributors:* Adrignola, Dreas, Gabio, Jguk, Willemd, 18 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Mesh/env** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616429> *Contributors:* Adrignola, Gabio, Jguk, 19 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Mesh/vg** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616431> *Contributors:* Adrignola, Gabio, Jguk, Madcello, Stalepanda, 32 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Mesh/Shape** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616427> *Contributors:* Adrignola, AndyD, Gabio, Jguk, 10 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Mesh/Shape/Sync** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1696201> *Contributors:* Adrignola, AndyD, Jguk, Nick.anderegg, QuiteUnusual, TBOL3, 19 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/index** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1691584> *Contributors:* Adrignola, Allefant, AndyD, Dipingo, Gabio, Jguk, Mstram, Soylentgreen, 7 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/cl** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616416> *Contributors:* Adrignola, Derbeth, Gabio, Iluvblender, Jguk, LokiClock, 5 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/cr** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616417> *Contributors:* Adrignola, Gabio, Jguk, 4 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/tt** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616425> *Contributors:* Adrignola, Gabio, Jguk, Webaware, 5 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/f1** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1640024> *Contributors:* Adrignola, Allefant, Gabio, Jguk, 14 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/fp** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616419> *Contributors:* Adrignola, Gabio, Jguk, Scorepon13

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/st** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616424> *Contributors:* Adrignola, Gabio, Jguk, SergeantOreo, 1 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/Const/ik** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616420> *Contributors:* Adrignola, Allefant, Gabio, Jguk, Jomegat, SergeantOreo, 16 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/timeline** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616437> *Contributors:* Adrignola, Gabio, Jguk

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/ipo/index** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616435> *Contributors:* Adrignola, Gabio, Jguk

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/ipo/type** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616436> *Contributors:* Adrignola, Gabio, Jguk

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/NLA/index** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616432> *Contributors:* Adrignola, Cspurrier, Halley, Jguk, Tannersf, 6 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/NLA/intro** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616669> *Contributors:* Adrignola, Darklama, Zeviion, 2 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/Guided tour/NLA/stride** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616433> *Contributors:* Adrignola, Jguk, Jguk, 3 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/bob/index** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616442> *Contributors:* Adrignola, Jguk, Wavez, 18 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/bob/Build rig** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616440> *Contributors:* Adrignola, Jguk, Wavez, 1 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/Piston, Rod and Crank** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1639631> *Contributors:* Adrignola, Darklama, Soylentgreen, 10 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Animation/example/ Cutting Through Steel** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617065> *Contributors:* Adrignola, Null Point, SoylentGreen

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Game Engine/index** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616448> *Contributors:* Adrignola, Bullercruz1, Miyazaki, Xerol

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Game Engine/Game Creating Techniques(GUI)** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617068> *Contributors:* Adrignola, Bullercruz1, 1 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Game Engine/Game Creating Techniques(GUI)/Creating Pop-Up Menus** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616447> *Contributors:* Adrignola, Bullercruz1, Muimi, 7 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Game Engine/Game Creating Techniques(GUI)/Creating Moving Menus** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616446> *Contributors:* Adrignola, Bullercruz1, Van der Hoorn

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Game Engine/Game Creating Techniques(GUI)/The "5-Layer" Button** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617069> *Contributors:* Adrignola, Bullercruz1

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Game Engine/Game Creating Techniques(GUI)/Creating Object Outlines** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617067> *Contributors:* Adrignola, Bullercruz1, 3 anonymous edits

**Blender 3D: Noob to Pro/Advanced Tutorials/Advanced Game Engine/Game Creating Techniques(Python)** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617070> *Contributors:* Adrignola, Bullercruz1, 3 anonymous edits

**Blender 3D: Noob to Pro/modelling tips** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616332> *Contributors:* Adrignola, SoylentGreen, 2 anonymous edits

**Blender 3D: Noob to Pro/Creating Weapons based on 2D Images** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1644339> *Contributors:* Chelseafan528, Embri, Mike.lifeguard, Morerunes, Null Point, Root, SoylentGreen, Spellmaker, Swift, Totalwriter0331, 6 anonymous edits

**Blender 3D: Noob to Pro/Match Moving** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1635059> *Contributors:* Adrignola, Dan-aka-jack, Lordelph, SergeantOreo, 7 anonymous edits

**Blender 3D: Noob to Pro/Motion Tracking with Icarus** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617112> *Contributors:* Adrignola, Pixelstuff, Tom soderlund, 1 anonymous edits

**Blender 3D: Noob to Pro/Create a Clayman** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1441013> *Contributors:* Aschoeke, Bootaleg, Bullercruz1, Cptlongshlong, Darklama, Herbythyme, Iamunknow, Nick.anderegg, Reece, SBJohnny, Swift, Tannersf, 5 anonymous edits

**Blender 3D: Noob to Pro/Organic Modeling** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1691017> *Contributors:* Admirj, Adrignola, Jguk, Panic2k4, Pyro guy, Tdunn7, Venar303, 2 anonymous edits

**Blender 3D: Noob to Pro/Free Blender Samples** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1605044> *Contributors:* Big Bill57, Chelseafan528, Nick.anderegg, 7 anonymous edits

**Blender 3D: Noob to Pro/Understanding the Fluid Simulator** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1691389> *Contributors:* Adrignola, Orfo, 9 anonymous edits

**Blender 3D: Noob to Pro/Creating a jewel in Blender** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617093> *Contributors:* Adrignola, CommonsDelinker, GeForce3, Jensenns, Mike.lifeguard, Orfo, Sapalskimichal, Sendoshin, Webaware, Wildern, 5 anonymous edits

**Blender 3D: Noob to Pro/Modeling Realistically** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616665> *Contributors:* Adrignola, Argento, Halley, Jguk, Tybee, 9 anonymous edits

**Blender 3D: Noob to Pro/Modeling a picture** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616953> *Contributors:* AM088, Adrignola, Az1568, Bmarnz, Bullercruz1, Faux, JacKDURdEn, Jguk, Reverzitative, Rm5248, SchmS, SoylentGreen, Withinfocus, 35 anonymous edits

**Blender 3D: Noob to Pro/Creating Ogg-Theora movies using Blender** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616670> *Contributors:* Adrignola, Rwst

**Blender 3D: Noob to Pro/Creating animated GIF's using Blender and Gimp** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1707444> *Contributors:* Adrignola, Aidan.Sullivan, Chelseafan528, Darklama, DrDixie, Quibik, Socratesone, Soeb, 1 anonymous edits

**Blender 3D: Noob to Pro/3D Tiling Backgrounds For The Web** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617063> *Contributors:* Adrignola, Quantum Anomaly, Red4tribe, 2 anonymous edits

**Blender 3D: Noob to Pro/Cool Things** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616493> *Contributors:* Adrignola, Fangiotopia, Iluvblender, Jguk, Jomegat, Jose da vinci, Milky, Mstram, Reece, Tybee, 37 anonymous edits

**Blender 3D: Troubleshooting** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616316> *Contributors:* Adrignola, Dallas1278, Jguk, Thenub314, 17 anonymous edits

**Blender 3D: Noob to Pro/Creating Blender Libraries** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616495> *Contributors:* Adrignola, Argento, Halley, Jguk, 9 anonymous edits

**Blender 3D: Noob to Pro/Blender Glossary** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1617092> *Contributors:* Adrignola, Bullercruz1, Electro, 6 anonymous edits

**Blender 3D: Noob to Pro/Object Mode HotKeys Review** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616521> *Contributors:* AM088, Adrignola, Jguk, Mochimo, Mosquito, Pquijal, Rm5248, Spiderworm, Wavez, Zilch, 40 anonymous edits

**Blender 3D: Noob to Pro/Edit Mode HotKeys Review** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616516> *Contributors:* Adrignola, Jguk, Spiderworm, Zilch

**Blender 3D: Noob to Pro/Hot Keys** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1636505> *Contributors:* AM088, Adrignola, Aleksev, Bullercruz1, Cardinal4, Crouch, Dany 123, Popski, Spiderworm, ZeroOne, 22 anonymous edits

**Blender 3D: Noob to Pro/FAQ** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1712482> *Contributors:* Adrignola, Alabandit, Alex G, Furrykef, Geoff Plourde, Illusionist, Jguk, Jomegat, LeftClicker, Malaz, Rosver, SETH HIKARU, Sigma 7, SoylentGreen, Ysangkok, 31 anonymous edits

**Blender 3D: Noob to Pro/Blueprint Links List** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616483> *Contributors:* Adrignola, Argento, Jguk, ThriiDæ, 3 anonymous edits

**Blender 3D/Tutorial Links List** *Source:* <http://en.wikibooks.org/w/index.php?oldid=1616326> *Contributors:* 1983, Adrignola, AfroToad, Argento, Chelseafan528, Comwizz2, Cthames, Dbreece, Derbeth, Doj^, EatMyShortz, GustavTheMushroom, Herbythyme, IamInnocent, Jmartens, Joystick Game, Kok, Laleena, Lechtiseb, Luckofbuck, Mach1723, Michaelx153, NuclearWarfare, Ok, Popski, PowuaTeam, Reddogzip, Remi0o, RogerWickes, Super3boy, Teddygage, Thangalin, Vrabec, WeirdHat, WolfprintFX, ZeroOne, Zilch, 285 anonymous edits

# Image Sources, Licenses and Contributors

**File:CoverArt.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:CoverArt.jpg> *License:* Public Domain *Contributors:* Jeremy B., Uploaded by image: 00%.svg *Source:* <http://en.wikibooks.org/w/index.php?title=File:00%.svg> *License:* Public Domain *Contributors:* User:Siebrand

**Image:Blender Koordinatensystem.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_Koordinatensystem.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_Koordinatensystem.jpg) *License:* GNU Free Documentation License *Contributors:* b:de:Benutzer:BorishollasBorishollas

**Image:Rechte-hand-regel.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Rechte-hand-regel.jpg> *License:* GNU Free Documentation License *Contributors:* User:Abdull

**Image:Blender3D DirectionOfRotation.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_DirectionOfRotation.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_DirectionOfRotation.png) *License:* Public Domain *Contributors:* User:SoylentGreen

**Image:Orthographic.svg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Orthographic.svg> *License:* Public Domain *Contributors:* Raylton P. Sousa

**Image:Orthographic2.svg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Orthographic2.svg> *License:* Public Domain *Contributors:* Raylton P. Sousa

**Image:Isometric.svg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Isometric.svg> *License:* Public Domain *Contributors:* Original uploader was BANZ111, at en.wikibooks

**Image:Perspective-1point.svg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Perspective-1point.svg> *License:* Public Domain *Contributors:* Original uploader was Marco Polo at en.wikipedia

**Image:Perspective-2point.svg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Perspective-2point.svg> *License:* Public Domain *Contributors:* Original uploader was Marco Polo at en.wikipedia

**Image:Perspective-3point.svg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Perspective-3point.svg> *License:* Public Domain *Contributors:* Original uploader was Marco Polo at en.wikipedia

**Image:Blender3D WhatIsA3DSpace.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_WhatIsA3DSpace.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_WhatIsA3DSpace.jpg) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**File:Blender3d ParentChildCoordinateAnimation.gif** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3d\\_ParentChildCoordinateAnimation.gif](http://en.wikibooks.org/w/index.php?title=File:Blender3d_ParentChildCoordinateAnimation.gif) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**Image:Blender3D ParentChildCoordinates.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_ParentChildCoordinates.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_ParentChildCoordinates.png) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**Image:Blender3D ViewCoordinatesProjectionPlane.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_ViewCoordinatesProjectionPlane.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_ViewCoordinatesProjectionPlane.jpg) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**Image:Blender3d ViewCoordinates.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3d\\_ViewCoordinates.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3d_ViewCoordinates.jpg) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**Image:Blender3d NormalKoordinates.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3d\\_NormalKoordinates.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3d_NormalKoordinates.jpg) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**Image:Blender3d NormalEdgeKoordinates.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3d\\_NormalEdgeKoordinates.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3d_NormalEdgeKoordinates.jpg) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**Image:Blender3D-firstRun.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3D-firstRun.png> *License:* GNU General Public License *Contributors:* Blender Foundation

**Image:Blender3D-firstRun.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3D-firstRun.jpg> *License:* unknown *Contributors:* User:Spiderworm

**Image:Blender3D-changingWindowType.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3D-changingWindowType.jpg> *License:* unknown *Contributors:* b:en:User:Spiderworm

**Image:Blender3D-windowBorder.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3D-windowBorder.jpg> *License:* unknown *Contributors:* b:en:User:Spiderworm

**Image:Blender3D-splittingWindows.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3D-splittingWindows.jpg> *License:* Public Domain *Contributors:* Raylton P. Sousa

**Image:Blender3D-joiningWindows.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3D-joiningWindows.jpg> *License:* Public Domain *Contributors:* Raylton P. Sousa

**Image:Blender3D-userPrefsExtended.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3D-userPrefsExtended.png> *License:* Public Domain *Contributors:* Raylton P. Sousa

**Image:Blender3D-userPrefs-AutoSave.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3D-userPrefs-AutoSave.jpg> *License:* Public Domain *Contributors:* Raylton P. Sousa

**Image:Blender3D-userPrefs-EditMethods.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3D-userPrefs-EditMethods.jpg> *License:* Public Domain *Contributors:* Raylton P. Sousa

**Image:blender3d\_emulate\_numpad.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3d\\_emulate\\_numpad.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3d_emulate_numpad.jpg) *License:* Public Domain *Contributors:* Raylton P. Sousa

**Image:Blender3D-userPrefs-SaveSettings.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3D-userPrefs-SaveSettings.png> *License:* GNU General Public License *Contributors:* Yhevhe

**Image:Blender3D-buttonsWindow-editButtons.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3D-buttonsWindow-editButtons.jpg> *License:* Public Domain *Contributors:* Raylton P. Sousa

**Image:Blender3D-buttonsWindow-logicButtons.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3D-buttonsWindow-logicButtons.jpg> *License:* Public Domain *Contributors:* Raylton P. Sousa

**Image:Blender3D-buttonsWindow-scriptButtons.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3D-buttonsWindow-scriptButtons.jpg> *License:* Public Domain *Contributors:* Raylton P. Sousa

**File:Blender 3D Camera.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_3D\\_Camera.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_3D_Camera.jpg) *License:* unknown *Contributors:* Original uploader was Odie5533 at en.wikibooks

**File:Blender 3D Lamp.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_3D\\_Lamp.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_3D_Lamp.jpg) *License:* GNU Free Documentation License *Contributors:* Original uploader was Odie5533 at en.wikibooks

**File:Blender 3D Cube.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_3D\\_Cube.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_3D_Cube.jpg) *License:* GNU Free Documentation License *Contributors:* Original uploader was Odie5533 at en.wikibooks

**Image:Blender\_view\_numpad.svg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_view\\_numpad.svg](http://en.wikibooks.org/w/index.php?title=File:Blender_view_numpad.svg) *License:* GNU General Public License *Contributors:* Original uploader was 2DCube at en.wikibooks

**File:CursorToSelection.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:CursorToSelection.png> *License:* GNU General Public License *Contributors:* Andrewtrefethen

**Image:Layers\_grid.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Layers\\_grid.png](http://en.wikibooks.org/w/index.php?title=File:Layers_grid.png) *License:* Public Domain *Contributors:* Original uploader was Lazy-lump at en.wikibooks

**Image:Object-edit\_mode.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Object-edit\\_mode.png](http://en.wikibooks.org/w/index.php?title=File:Object-edit_mode.png) *License:* GNU Free Documentation License *Contributors:* Original uploader was Soeb at en.wikibooks

**Image:Blender3D window type selector.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_window\\_type\\_selector.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_window_type_selector.png) *License:* unknown *Contributors:* Arturo Reina, Bayo, OsamaKAWB

**Image:MeshModeling.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:MeshModeling.png> *License:* Public Domain *Contributors:* Rurik

**File:Blender Screenshot for Beginner Tutorial.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_Screenshot\\_for\\_Beginner\\_Tutorial.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_Screenshot_for_Beginner_Tutorial.jpg) *License:* GNU General Public License *Contributors:* Original uploader was Daworm at en.wikibooks

**File:Blender 3D-View Bottom Border.JPG** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_3D-View\\_Bottom\\_Border.JPG](http://en.wikibooks.org/w/index.php?title=File:Blender_3D-View_Bottom_Border.JPG) *License:* GNU General Public License *Contributors:* Original uploader was Daworm at en.wikibooks

**File:BlenderExampleEditingScreenshot.JPG** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderExampleEditingScreenshot.JPG> *License:* GNU General Public License *Contributors:* Original uploader was Daworm at en.wikibooks

**Image:Blender\_object\_mode\_1.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_object\\_mode\\_1.png](http://en.wikibooks.org/w/index.php?title=File:Blender_object_mode_1.png) *License:* GNU General Public License *Contributors:* Arturo Reina, Bayo

**Image:Information icon.svg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Information\\_icon.svg](http://en.wikibooks.org/w/index.php?title=File:Information_icon.svg) *License:* unknown *Contributors:* El T

**Image:Blender\_panel\_modifiers.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_panel\\_modifiers.png](http://en.wikibooks.org/w/index.php?title=File:Blender_panel_modifiers.png) *License:* unknown *Contributors:* Fr

**Image:Blenderediting.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderediting.png> *License:* unknown *Contributors:* Arturo Reina, Bayo, OsamaKAWB

**File:Blender edit buttons.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_edit\\_buttons.png](http://en.wikibooks.org/w/index.php?title=File:Blender_edit_buttons.png) *License:* GNU General Public License *Contributors:* Adrignola, Darklama, Totophe64, ZeroOne

**Image:Blender\_tips\_cube\_subsurfed.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_tips\\_cube\\_subsurfed.png](http://en.wikibooks.org/w/index.php?title=File:Blender_tips_cube_subsurfed.png) *License:* unknown *Contributors:* LuizAlvarez

**Image:Blender\_tips\_cube\_subsurfed2.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_tips\\_cube\\_subsurfed2.png](http://en.wikibooks.org/w/index.php?title=File:Blender_tips_cube_subsurfed2.png) *License:* unknown *Contributors:* LuizAlvarez

**Image:Blender\_cube\_creature\_0.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_cube\\_creature\\_0.png](http://en.wikibooks.org/w/index.php?title=File:Blender_cube_creature_0.png) *License:* unknown *Contributors:* Fr

**Image:Blenderface.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderface.png> *License:* unknown *Contributors:* DariusWiles

**Image:Blender\_cube\_creature\_1.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_cube\\_creature\\_1.png](http://en.wikibooks.org/w/index.php?title=File:Blender_cube_creature_1.png) *License:* unknown *Contributors:* Fr

**Image:Blender\_cube\_creature\_2.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_cube\\_creature\\_2.png](http://en.wikibooks.org/w/index.php?title=File:Blender_cube_creature_2.png) *License:* unknown *Contributors:* Fr

**Image:Blender3D\_3D\_viewport.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_3D\\_viewport.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_3D_viewport.png) *License:* GNU General Public License *Contributors:* Arturo Reina, Bayo, OsamaKAWB

**Image:Blender\_edit\_mode\_1.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_edit\\_mode\\_1.png](http://en.wikibooks.org/w/index.php?title=File:Blender_edit_mode_1.png) *License:* unknown *Contributors:* Arturo Reina, Bayo, OsamaKAWB

**Image:Handicon\_blender.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Handicon\\_blender.png](http://en.wikibooks.org/w/index.php?title=File:Handicon_blender.png) *License:* unknown *Contributors:* Az1568, Beetlemaniac, Webaware, 1 anonymous edits

**Image:Handicon\_blender\_2.49a.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Handicon\\_blender\\_2.49a.png](http://en.wikibooks.org/w/index.php?title=File:Handicon_blender_2.49a.png) *License:* Public Domain *Contributors:* Adrignola, Raylton P. Sousa

**Image:Blender3DNobToPro-SimpleHat-SelectModes.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3DNobToPro-SimpleHat-SelectModes.png> *License:* GNU General Public License *Contributors:* Arturo Reina, Bayo, Liso

**image:Blender\_first\_quickierender.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_first\\_quickierender.png](http://en.wikibooks.org/w/index.php?title=File:Blender_first_quickierender.png) *License:* Public Domain *Contributors:* Raylton P. Sousa

**Image:Menu\_Render\_Current\_Frame.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Menu\\_Render\\_Current\\_Frame.png](http://en.wikibooks.org/w/index.php?title=File:Menu_Render_Current_Frame.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image:Blender simple person.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_simple\\_person.png](http://en.wikibooks.org/w/index.php?title=File:Blender_simple_person.png) *License:* unknown *Contributors:* Arturo Reina, Ausis, Bayo, OsamaKAWB

**Image:Object-edit\_mode.JPG** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Object-edit\\_mode.JPG](http://en.wikibooks.org/w/index.php?title=File:Object-edit_mode.JPG) *License:* unknown *Contributors:* Rm5248

**Image:Blender3D\_Simple\_Person\_First\_Selection.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Simple\\_Person\\_First\\_Selection.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Simple_Person_First_Selection.png) *License:* unknown *Contributors:* Arturo Reina, Bayo, OsamaKAWB

**Image:Make\_selection\_visible.JPG** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Make\\_selection\\_visible.JPG](http://en.wikibooks.org/w/index.php?title=File:Make_selection_visible.JPG) *License:* unknown *Contributors:* Rdp

**Image:Blender3D\_Edge-select-mode\_Button.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Edge-select-mode\\_Button.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Edge-select-mode_Button.png) *License:* GNU General Public License *Contributors:* Dave Mason

**Image:Blender3D\_Vertex-select-mode\_Button.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Vertex-select-mode\\_Button.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Vertex-select-mode_Button.png) *License:* unknown *Contributors:* Arturo Reina, Bayo, OsamaKAWB

**Image:Blender\_Ortographic\_selection.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_Ortographic\\_selection.png](http://en.wikibooks.org/w/index.php?title=File:Blender_Ortographic_selection.png) *License:* GNU General Public License *Contributors:* YesPoX

**Image:Blender3D\_Simple\_Person\_Step1.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Simple\\_Person\\_Step1.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Simple_Person_Step1.png) *License:* unknown *Contributors:* Arturo Reina, Bayo, OsamaKAWB

**Image:Blender3D\_Simple\_Person\_Step2.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Simple\\_Person\\_Step2.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Simple_Person_Step2.png) *License:* unknown *Contributors:* Arturo Reina, Bayo, OsamaKAWB

**Image:Blender3D\_Simple\_Person\_Step3.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Simple\\_Person\\_Step3.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Simple_Person_Step3.png) *License:* unknown *Contributors:* Arturo Reina, Bayo, OsamaKAWB

**Image:Blender3D\_Simple\_Person\_Step4.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Simple\\_Person\\_Step4.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Simple_Person_Step4.png) *License:* unknown *Contributors:* Arturo Reina, Bayo, OsamaKAWB

**Image:Blender3D\_Simple\_Person\_Step5.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Simple\\_Person\\_Step5.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Simple_Person_Step5.png) *License:* unknown *Contributors:* Arturo Reina, Bayo, OsamaKAWB

**Image:Blender-Edit-Mode.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender-Edit-Mode.png> *License:* Public Domain *Contributors:* Romain Behar

**Image:Blender3D\_editing\_panel\_button.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_editing\\_panel\\_button.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_editing_panel_button.png) *License:* unknown *Contributors:* Arturo Reina, Duesentrieb, OsamaKAWB

**File:Blender3D\_sbsurf\_buttons (1).png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_sbsurf\\_buttons\\_\(1\).png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_sbsurf_buttons_(1).png) *License:* unknown *Contributors:* Original uploader was Spiderworm at en.wikibooks. Later version(s) were uploaded by Argento at en.wikibooks.

**Image:Apply-modifier-to-edit-mode.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Apply-modifier-to-edit-mode.png> *License:* GNU General Public License *Contributors:* Jeffmilner

**Image:Simple\_person.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Simple\\_person.png](http://en.wikibooks.org/w/index.php?title=File:Simple_person.png) *License:* unknown *Contributors:* Arturo Reina, Bayo, Duesentrieb

**Image:Blender3D\_SelectionModes.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_SelectionModes.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_SelectionModes.png) *License:* GNU General Public License *Contributors:* Argento

**Image:Blender3D\_SimplePerson\_Snap\_Pelvis.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_SimplePerson\\_Snap\\_Pelvis.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_SimplePerson_Snap_Pelvis.png) *License:* Public Domain *Contributors:* Bgks

**Image:Blender-Pivot\_Median\_Point.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender-Pivot\\_Median\\_Point.png](http://en.wikibooks.org/w/index.php?title=File:Blender-Pivot_Median_Point.png) *License:* Public Domain *Contributors:* Romain Behar

**Image:Blender3D\_SimplePerson\_Scale\_Pelvis.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_SimplePerson\\_Scale\\_Pelvis.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_SimplePerson_Scale_Pelvis.png) *License:* unknown *Contributors:* Bgks

**Image:Blender\_Proportional\_edit\_button.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_Proportional\\_edit\\_button.png](http://en.wikibooks.org/w/index.php?title=File:Blender_Proportional_edit_button.png) *License:* GNU General Public License *Contributors:* YesPoX

**Image:Blender3D\_SimplePerson\_ScaleArmpits.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_SimplePerson\\_ScaleArmpits.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_SimplePerson_ScaleArmpits.png) *License:* unknown *Contributors:* Bgks

**Image:Blender3D\_RotationScalingPivot.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_RotationScalingPivot.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_RotationScalingPivot.png) *License:* GNU Free Documentation License *Contributors:* Bgks

**Image:Blender3D\_SimplePerson\_ScaleBelly\_lockZ.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_SimplePerson\\_ScaleBelly\\_lockZ.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_SimplePerson_ScaleBelly_lockZ.png) *License:* unknown *Contributors:* Bgks

**Image:Blender3D\_EraseArm.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_EraseArm.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_EraseArm.jpg) *License:* Public Domain *Contributors:* User:Null Point

**Image:BlenderArmless.JPG** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderArmless.JPG> *License:* unknown *Contributors:* Matt Howard

**Image:Blender3D\_Extruding\_again\_m.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Extruding\\_again\\_m.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Extruding_again_m.jpg) *License:* GNU Free Documentation License *Contributors:* Arturo Reina, Bayo

**Image:Blender3D\_Giving\_a\_face\_again\_m.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Giving\\_a\\_face\\_again\\_m.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Giving_a_face_again_m.jpg) *License:* GNU Free Documentation License *Contributors:* Arturo Reina, Bayo

**Image:Handsup.JPG** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Handsup.JPG> *License:* Public Domain *Contributors:* WikiWhatWikiWho

**Image:Blender3D\_SimplePerson\_SubdivideFeet.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_SimplePerson\\_SubdivideFeet.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_SimplePerson_SubdivideFeet.png) *License:* unknown *Contributors:* Bgks

**Image:feet-4.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:feet-4.jpg> *License:* Public Domain *Contributors:* WikiWhatWikiWho

**Image:Blender3DNobToPro-SimpleHat-Circle.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3DNobToPro-SimpleHat-Circle.png> *License:* unknown *Contributors:* MichaelFrey, Wwwwolf, ZeroOne

**Image:Blender3DNobToPro-SimpleHat-ThreeFacesSelected.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3DNobToPro-SimpleHat-ThreeFacesSelected.png> *License:* unknown *Contributors:* MichaelFrey, Wwwwolf, ZeroOne

**Image:Blender3DNoobToPro-SimpleHat-Profile.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Blender3DNoobToPro-SimpleHat-Profile.png> License: unknown Contributors: MichaelFrey, Wwwwolf, ZeroOne

**Image:Blender3DNoobToPro-SimpleHat-HatSpun.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Blender3DNoobToPro-SimpleHat-HatSpun.png> License: unknown Contributors: MichaelFrey, Wwwwolf, ZeroOne

**Image:Blender3DNoobToPro-SimpleHat-HatExtrude.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Blender3DNoobToPro-SimpleHat-HatExtrude.png> License: unknown Contributors: MichaelFrey, Wwwwolf, ZeroOne

**Image:Wiki-hat2.jpg** Source: <http://en.wikibooks.org/w/index.php?title=File:Wiki-hat2.jpg> License: GNU Free Documentation License Contributors: Quantufinity@gmail.com

**Image:Blender32\_hatadjust.jpg** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender32\\_hatadjust.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender32_hatadjust.jpg) License: GNU General Public License Contributors: User:Null Point

**File:Blender\_setup.jpg** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender\\_setup.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_setup.jpg) License: unknown Contributors: Adrignola, Rm5248

**Image:An armature.JPG** Source: [http://en.wikibooks.org/w/index.php?title=File:An\\_armature.JPG](http://en.wikibooks.org/w/index.php?title=File:An_armature.JPG) License: unknown Contributors: Adrignola, Rm5248

**Image:In body.JPG** Source: [http://en.wikibooks.org/w/index.php?title=File:In\\_body.JPG](http://en.wikibooks.org/w/index.php?title=File:In_body.JPG) License: unknown Contributors: Adrignola, Rm5248

**Image:Boned.JPG** Source: <http://en.wikibooks.org/w/index.php?title=File:Boned.JPG> License: unknown Contributors: Adrignola, Rm5248

**Image:Naming bones.JPG** Source: [http://en.wikibooks.org/w/index.php?title=File:Naming\\_bones.JPG](http://en.wikibooks.org/w/index.php?title=File:Naming_bones.JPG) License: unknown Contributors: Adrignola, Rm5248

**Image:Pose example.jpg** Source: [http://en.wikibooks.org/w/index.php?title=File:Pose\\_example.jpg](http://en.wikibooks.org/w/index.php?title=File:Pose_example.jpg) License: unknown Contributors: Rm5248

**Image:Blender3DNoobToPro-SimplePlane.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Blender3DNoobToPro-SimplePlane.png> License: unknown Contributors: Megaloman, Whiteknight, ZeroOne

**Image:Blender-Noob-to-Pro Proportional Menu.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender-Noob-to-Pro\\_Proportional\\_Menu.png](http://en.wikibooks.org/w/index.php?title=File:Blender-Noob-to-Pro_Proportional_Menu.png) License: GNU General Public License Contributors: Original uploader was Lirtferk at en.wikibooks

**Image:Blender3DNoobToPro-FirstMountain.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Blender3DNoobToPro-FirstMountain.png> License: GNU General Public License Contributors: Original uploader was Megaloman at en.wikibooks

**Image:Blender3DNoobToPro-Sharpfalloff.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Blender3DNoobToPro-Sharpfalloff.png> License: GNU General Public License Contributors: Original uploader was Megaloman at en.wikibooks

**Image:Blender3DNoobToPro-SharpVsSmoothFalloff.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Blender3DNoobToPro-SharpVsSmoothFalloff.png> License: GNU General Public License Contributors: Original uploader was Megaloman at en.wikibooks

**Image:Blender\_proportional\_smooth\_before\_rotation.jpg** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender\\_proportional\\_smooth\\_before\\_rotation.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_proportional_smooth_before_rotation.jpg) License: unknown Contributors: User:Hendric

**Image:Blender\_proportional\_smooth\_after\_rotation.jpg** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender\\_proportional\\_smooth\\_after\\_rotation.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_proportional_smooth_after_rotation.jpg) License: unknown Contributors: User:Hendric

**Image:Blender3DNoobToPro-RoundedMtns.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Blender3DNoobToPro-RoundedMtns.png> License: GNU General Public License Contributors: Original uploader was Megaloman at en.wikibooks

**Image:Blender3DNoobtoPro-SharpMtnCreasing.PNG** Source: <http://en.wikibooks.org/w/index.php?title=File:Blender3DNoobtoPro-SharpMtnCreasing.PNG> License: GNU General Public License Contributors: Original uploader was Skeep at en.wikibooks

**Image:editing.gif** Source: <http://en.wikibooks.org/w/index.php?title=File:Editing.gif> License: GNU General Public License Contributors: Darklama, Gregggreg, ZeroOne

**Image:BlenderCubeEdit.png** Source: <http://en.wikibooks.org/w/index.php?title=File:BlenderCubeEdit.png> License: GNU Free Documentation License Contributors: Original uploader was SchmS at en.wikibooks

**Image:BlenderSubdividedCube.png** Source: <http://en.wikibooks.org/w/index.php?title=File:BlenderSubdividedCube.png> License: GNU Free Documentation License Contributors: Original uploader was SchmS at en.wikibooks

**Image:Blender3D-Noob-To-Pro-SelectionModesInSolidMode.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Blender3D-Noob-To-Pro-SelectionModesInSolidMode.png> License: unknown Contributors: SatanClaus

**Image:Blender3D-Noob-To-Pro-GingerBreadManSelectedFaces3D.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Blender3D-Noob-To-Pro-GingerBreadManSelectedFaces3D.png> License: unknown Contributors: SatanClaus

**Image:ExtrudedCube1.png** Source: <http://en.wikibooks.org/w/index.php?title=File:ExtrudedCube1.png> License: GNU General Public License Contributors: Argento

**Image:ExtrudedCube2.png** Source: <http://en.wikibooks.org/w/index.php?title=File:ExtrudedCube2.png> License: GNU General Public License Contributors: Argento

**Image:Blender3D-Noob-To-Pro-GingerBreadManExtrudedCube3D.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Blender3D-Noob-To-Pro-GingerBreadManExtrudedCube3D.png> License: unknown Contributors: SatanClaus

**Image:GingerbreadMan-LeftLeg01.png** Source: <http://en.wikibooks.org/w/index.php?title=File:GingerbreadMan-LeftLeg01.png> License: unknown Contributors: N comme Nul

**Image:GingerbreadMan-LeftLeg02.png** Source: <http://en.wikibooks.org/w/index.php?title=File:GingerbreadMan-LeftLeg02.png> License: unknown Contributors: N comme Nul

**Image:BlenderExtrudedLeg.png** Source: <http://en.wikibooks.org/w/index.php?title=File:BlenderExtrudedLeg.png> License: GNU Free Documentation License Contributors: Original uploader was SchmS at en.wikibooks

**Image:Oneleg-ginger.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Oneleg-ginger.png> License: GNU Free Documentation License Contributors: User:Darkonc

**File:BlenderGingerBreadManOneLegGinger3D.png** Source: <http://en.wikibooks.org/w/index.php?title=File:BlenderGingerBreadManOneLegGinger3D.png> License: GNU General Public License Contributors: Original uploader was SatanClaus at en.wikibooks Later versions were uploaded by Argento at en.wikibooks.

**Image:Twoleg-ginger.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Twoleg-ginger.png> License: GNU Free Documentation License Contributors: User:Darkonc

**Image:Twoleg-ginger-pulled.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Twoleg-ginger-pulled.png> License: GNU Free Documentation License Contributors: User:Darkonc

**Image:Blender3D-Noob-To-Pro-GingerBreadMan2LegPulled.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Blender3D-Noob-To-Pro-GingerBreadMan2LegPulled.png> License: unknown Contributors: SatanClaus

**Image:Twoleg-ginger-head.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Twoleg-ginger-head.png> License: GNU General Public License Contributors: Argento, Darkonc, ZeroOne

**image:ginger-zscale.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Ginger-zscale.png> License: GNU Free Documentation License Contributors: User:Darkonc

**image:ginger-smoothbody.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Ginger-smoothbody.png> License: GNU Free Documentation License Contributors: User:Darkonc

**image:ginger-smoothall.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Ginger-smoothall.png> License: GNU Free Documentation License Contributors: User:Darkonc

**Image:BlenderGingerRender.png** Source: <http://en.wikibooks.org/w/index.php?title=File:BlenderGingerRender.png> License: Public Domain Contributors: Original uploader was Argento at en.wikibooks

**File:LinksPipeAddNew.png** Source: <http://en.wikibooks.org/w/index.php?title=File:LinksPipeAddNew.png> License: GNU General Public License Contributors: User:DDD3x

**File:TextureAddNew.png** Source: <http://en.wikibooks.org/w/index.php?title=File:TextureAddNew.png> License: GNU General Public License Contributors: User:DDD3x

**File:StucciType.png** Source: <http://en.wikibooks.org/w/index.php?title=File:StucciType.png> License: GNU General Public License Contributors: User:DDD3x

**File:SizeTurbulence.png** Source: <http://en.wikibooks.org/w/index.php?title=File:SizeTurbulence.png> License: GNU General Public License Contributors: User:DDD3x

**File:LookForMapTo.png** Source: <http://en.wikibooks.org/w/index.php?title=File:LookForMapTo.png> License: GNU General Public License Contributors: User:DDD3x

**File:MapToChange.png** Source: <http://en.wikibooks.org/w/index.php?title=File:MapToChange.png> License: GNU General Public License Contributors: User:DDD3x

**File:MapInputChange.png** Source: <http://en.wikibooks.org/w/index.php?title=File:MapInputChange.png> License: GNU General Public License Contributors: User:DDD3x

**File:MaterialChangePreview.png** Source: <http://en.wikibooks.org/w/index.php?title=File:MaterialChangePreview.png> License: GNU General Public License Contributors: User:DDD3x

**File:BlenderGingerTexturedRender.png** Source: <http://en.wikibooks.org/w/index.php?title=File:BlenderGingerTexturedRender.png> License: Public Domain Contributors: User:DDD3x

**Image:Blender\_3D-cursor.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender\\_3D-cursor.png](http://en.wikibooks.org/w/index.php?title=File:Blender_3D-cursor.png) License: unknown Contributors: Arturo Reina, Bayo, Friedy-peach, OsamaKAWB

**Image:Blenderpenguin1.gif** Source: <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin1.gif> License: GNU Free Documentation License Contributors: Original uploader was Shythinker at en.wikibooks

**Image:Blender3D\_Limit\_selection\_to\_visible.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Limit\\_selection\\_to\\_visible.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Limit_selection_to_visible.png) *License:* unknown *Contributors:* Romain Behar

**File:Blenderpenguin2a.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin2a.png> *License:* GNU General Public License *Contributors:* Screenshot of Blender by me.

**Image:Blenderpenguin2b.gif** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin2b.gif> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:Blenderpenguin2c.gif** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin2c.gif> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:Blenderpenguin3c.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin3c.png> *License:* GNU General Public License *Contributors:* Roanoke

**Image:Blenderpenguin4b.gif** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin4b.gif> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:Penguin\_from\_sphere - wing start.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Penguin\\_from\\_sphere\\_-\\_wing\\_start.png](http://en.wikibooks.org/w/index.php?title=File:Penguin_from_sphere_-_wing_start.png) *License:* Public Domain *Contributors:* Romain Behar

**Image:Blenderpenguin5b.gif** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin5b.gif> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:Blenderpenguin4-02.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin4-02.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:Blenderpenguin4-03.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin4-03.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:Blender - Penguinto spheres - wing smoothing.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_-\\_Penguinto\\_spheres\\_-\\_wing\\_smoothing.png](http://en.wikibooks.org/w/index.php?title=File:Blender_-_Penguinto_spheres_-_wing_smoothing.png) *License:* Public Domain *Contributors:* Romain Behar

**Image:Blenderpenguin4-04.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin4-04.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:Blenderpenguin6b.gif** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin6b.gif> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:Blender - Penguins to spheres - wing complete.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_-\\_Penguins\\_to\\_spheres\\_-\\_wing\\_complete.png](http://en.wikibooks.org/w/index.php?title=File:Blender_-_Penguins_to_spheres_-_wing_complete.png) *License:* Public Domain *Contributors:* Romain Behar

**Image:Blenderpenguin5-01.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin5-01.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:Blenderpenguin7.gif** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin7.gif> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:Blender - Penguins to spheres - feet start.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_-\\_Penguins\\_to\\_spheres\\_-\\_feet\\_start.png](http://en.wikibooks.org/w/index.php?title=File:Blender_-_Penguins_to_spheres_-_feet_start.png) *License:* Public Domain *Contributors:* Romain Behar

**Image:Blender - Penguins to spheres - feet.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_-\\_Penguins\\_to\\_spheres\\_-\\_feet.png](http://en.wikibooks.org/w/index.php?title=File:Blender_-_Penguins_to_spheres_-_feet.png) *License:* Public Domain *Contributors:* Romain Behar

**Image:Blenderpenguin5-05.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin5-05.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:Blenderpenguin5-06.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin5-06.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:Blenderpenguin5-08.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin5-08.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:Blenderpenguin8.gif** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin8.gif> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:Blender - Penguins to spheres - feet selection.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_-\\_Penguins\\_to\\_spheres\\_-\\_feet\\_selection.png](http://en.wikibooks.org/w/index.php?title=File:Blender_-_Penguins_to_spheres_-_feet_selection.png) *License:* Public Domain *Contributors:* Romain Behar

**Image:Blenderpenguin5-09.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin5-09.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:Blenderpenguin5-10.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blenderpenguin5-10.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:Blender - Penguins to spheres - tail.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_-\\_Penguins\\_to\\_spheres\\_-\\_tail.png](http://en.wikibooks.org/w/index.php?title=File:Blender_-_Penguins_to_spheres_-_tail.png) *License:* Public Domain *Contributors:* Romain Behar

**Image:Blender - Penguins to spheres - subsurf.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_-\\_Penguins\\_to\\_spheres\\_-\\_subsurf.png](http://en.wikibooks.org/w/index.php?title=File:Blender_-_Penguins_to_spheres_-_subsurf.png) *License:* Public Domain *Contributors:* Romain Behar

**Image:Penguin\_orbisonitrum.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Penguin\\_orbisonitrum.jpg](http://en.wikibooks.org/w/index.php?title=File:Penguin_orbisonitrum.jpg) *License:* Public Domain *Contributors:* Orbisonitrum

**Image:peng.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Peng.jpg> *License:* Public Domain *Contributors:* Jesse S. (Bnty)

**Image:Penguin.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Penguin.jpg> *License:* Public Domain *Contributors:* Bayo, EugeneZelenko, Liso, WhiplashEffigy

**File:Blender3D\_Die\_Another\_Way.ogv** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Die\\_Another\\_Way.ogv](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Die_Another_Way.ogv) *License:* Public Domain *Contributors:* User:Itinerantrader

**Image:DieAnotherWay2.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:DieAnotherWay2.png> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:DieAnotherWay3b.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:DieAnotherWay3b.png> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:9\_posiciones.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:9\\_posiciones.jpg](http://en.wikibooks.org/w/index.php?title=File:9_posiciones.jpg) *License:* Public Domain *Contributors:* Reptiles extintos

**Image:muescas\_y\_espacios.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Muescas\\_y\\_espacios.jpg](http://en.wikibooks.org/w/index.php?title=File:Muescas_y_espacios.jpg) *License:* Public Domain *Contributors:* Reptiles extintos

**Image:BlenderDieanotherway3-00.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderDieanotherway3-00.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:DieAnotherWay10a.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:DieAnotherWay10a.png> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:DieAnotherWay10b.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:DieAnotherWay10b.png> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:BlenderCommonLSTV-on.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderCommonLSTV-on.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:DieAnotherWay11b.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:DieAnotherWay11b.png> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:DieAnotherWay12c.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:DieAnotherWay12c.png> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:DieAnotherWay13.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:DieAnotherWay13.png> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:DieAnotherWay14.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:DieAnotherWay14.png> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:DieAnotherWay15.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:DieAnotherWay15.png> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:BlenderDieanotherway5-01.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderDieanotherway5-01.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:BlenderCommonButtonswindow.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderCommonButtonswindow.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:BlenderCommonEditing-on.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderCommonEditing-on.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:DieAnotherWay17.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:DieAnotherWay17.png> *License:* Public Domain *Contributors:* Arturotena

**Image:DieAnotherWay17b.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:DieAnotherWay17b.png> *License:* Public Domain *Contributors:* Arturotena

**Image:BlenderCommonShading-on.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderCommonShading-on.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:BlenderCommonMaterialbuttons-on.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderCommonMaterialbuttons-on.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:BlenderDieanotherway5-04.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderDieanotherway5-04.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:BlenderDieanotherway5-04a.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderDieanotherway5-04a.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:BlenderDieanotherway5-05.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderDieanotherway5-05.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:DieAnotherWay18.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:DieAnotherWay18.png> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:BlenderCommonEditmode.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderCommonEditmode.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:BlenderCommonFaceselect-on.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderCommonFaceselect-on.png> *License:* GNU Free Documentation License *Contributors:* Boyage. Original uploader was Boyage at en.wikibooks

**Image:DieAnotherWay18.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:DieAnotherWay18.png> *License:* GNU Free Documentation License *Contributors:* Original uploader was Shythinker at en.wikibooks

**Image:bevel-done.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bevel-done.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:Blender-Die Easy 2-Knife.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender-Die\\_Easy\\_2-Knife.png](http://en.wikibooks.org/w/index.php?title=File:Blender-Die_Easy_2-Knife.png) *License:* Public Domain *Contributors:* Romain Behar

**Image:front-face-9v-cuts.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Front-face-9v-cuts.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:select-edge-mode.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Select-edge-mode.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:limit-selection-to-visible.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Limit-selection-to-visible.png> *License:* unknown *Contributors:* Webaware, Wynnmc

**Image:second-cut-selected.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Second-cut-selected.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:second-cut-removed.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Second-cut-removed.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:fifth-cut-selected.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Fifth-cut-selected.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:fifth-cut-removed.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Fifth-cut-removed.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:eighth-cut-selected.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Eighth-cut-selected.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:eighth-cut-removed.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Eighth-cut-removed.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:front-face-9h-cuts.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Front-face-9h-cuts.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:front-face-h-cuts-finished.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Front-face-h-cuts-finished.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:all-cuts-done.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:All-cuts-done.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:shading-panel-icon.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Shading-panel-icon.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:material-ramps-panel.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Material-ramps-panel.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:Manual-Part-I-Interface-ColorSelector-Dialog.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Manual-Part-I-Interface-ColorSelector-Dialog.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:face-select-mode.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Face-select-mode.png> *License:* GNU Free Documentation License *Contributors:* Wynnmc

**Image:alt-m-one-spot.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Alt-m-one-spot.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:alt-m-five-spots.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Alt-m-five-spots.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:alt-m-five-spots-grabbed.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Alt-m-five-spots-grabbed.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:face-converted-to-triangles.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Face-converted-to-triangles.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:face-spots-subdivided.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Face-spots-subdivided.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:five-spots-extruded.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Five-spots-extruded.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:alt-m-all-spots-done.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Alt-m-all-spots-done.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:ctrl-t-all-spots-done.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Ctrl-t-all-spots-done.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:Editing-panel-modifiers-subsurf.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Editing-panel-modifiers-subsurf.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Webaware, Wynnmc

**Image:alt-m-subsurf-level-2-set-smooth.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Alt-m-subsurf-level-2-set-smooth.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:ctrl-t-subsurf-level-2-set-smooth.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Ctrl-t-subsurf-level-2-set-smooth.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:creasing-five-spots-selected.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Creasing-five-spots-selected.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:creasing-five-spots-done.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Creasing-five-spots-done.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc

**Image:alt-m-all-spots-creased.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Alt-m-all-spots-creased.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmc





**Image:bl21.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl21.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks. Later version(s) were uploaded by Adrignola at en.wikibooks.

**Image:bl22.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl22.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks. Later version(s) were uploaded by Adrignola at en.wikibooks.

**Image:bl24.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl24.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks. Later version(s) were uploaded by Adrignola at en.wikibooks.

**Image:bl25.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl25.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks. Later version(s) were uploaded by Adrignola at en.wikibooks.

**Image:bl26.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl26.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks. Later version(s) were uploaded by Adrignola at en.wikibooks.

**Image:bl27.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl27.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks. Later version(s) were uploaded by Adrignola at en.wikibooks.

**Image:bl28.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl28.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks. Later version(s) were uploaded by Adrignola at en.wikibooks.

**Image:bl30.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl30.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks

**Image:bl31.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl31.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks. Later version(s) were uploaded by Adrignola at en.wikibooks.

**Image:bl32.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl32.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks. Later version(s) were uploaded by Adrignola at en.wikibooks.

**Image:bl33.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl33.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks. Later version(s) were uploaded by Adrignola at en.wikibooks.

**Image:bl34.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl34.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks. Later version(s) were uploaded by Adrignola at en.wikibooks.

**Image:bl35.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl35.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks. Later version(s) were uploaded by Adrignola at en.wikibooks.

**Image:bl36.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl36.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks. Later version(s) were uploaded by Adrignola at en.wikibooks.

**Image:bl37.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl37.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks. Later version(s) were uploaded by Adrignola at en.wikibooks.

**Image:bl39.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bl39.jpg> *License:* unknown *Contributors:* Original uploader was Sumoncil at en.wikibooks. Later version(s) were uploaded by Adrignola at en.wikibooks.

**Image:split-windows.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Split-windows.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image:split-windows-left-right.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Split-windows-left-right.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image:split-windows-complete.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Split-windows-complete.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image:duplicate-cube.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Duplicate-cube.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image:top-four-vertices-selected.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Top-four-vertices-selected.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image:roof-scaled.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Roof-scaled.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image:roof-overhang-all.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Roof-overhang-all.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image:roof-overhang-xy-only.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Roof-overhang-xy-only.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image:ob-cube.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Ob-cube.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image:shading-icon.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Shading-icon.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image:house-roof-coloured.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:House-roof-coloured.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image:chimney-in-position.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Chimney-in-position.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image>window-and-door-added.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Window-and-door-added.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image:front-fence-palings-done.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Front-fence-palings-done.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image:fence-complete.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Fence-complete.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image:house-complete.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:House-complete.png> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Wynnmce

**Image:Blender3D\_Roehrenabzweig.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Roehrenabzweig.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Roehrenabzweig.png) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**Image:Blender3D\_6TubesJoint.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_6TubesJoint.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_6TubesJoint.png) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**Image:Blender3D\_3TubesJoint.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_3TubesJoint.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_3TubesJoint.png) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**Image:Blender3D\_LargeSmallTube.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_LargeSmallTube.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_LargeSmallTube.png) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**Image:Blender3D\_LargeSmallTubeResult.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_LargeSmallTubeResult.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_LargeSmallTubeResult.png) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**Image:squaretriangle.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Squaretriangle.png> *License:* Public Domain *Contributors:* Original uploader was Lirtferk at en.wikibooks

**Image:triangleexemplar.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Triangleexemplar.png> *License:* Public Domain *Contributors:* Iamunknow, Lirtferk

**Image:Importing\_Image.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Importing\\_Image.jpg](http://en.wikibooks.org/w/index.php?title=File:Importing_Image.jpg) *License:* unknown *Contributors:* Badalia

**Image:1st\_vertex.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:1st\\_vertex.jpg](http://en.wikibooks.org/w/index.php?title=File:1st_vertex.jpg) *License:* unknown *Contributors:* Badalia

**Image:1st\_line.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:1st\\_line.jpg](http://en.wikibooks.org/w/index.php?title=File:1st_line.jpg) *License:* unknown *Contributors:* Badalia

**Image:1st\_face.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:1st\\_face.jpg](http://en.wikibooks.org/w/index.php?title=File:1st_face.jpg) *License:* unknown *Contributors:* Badalia

**Image:sidetwo.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Sidetwo.jpg> *License:* GNU Free Documentation License *Contributors:* Moreejt

**Image:Toywolf-construction-left.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Toywolf-construction-left.jpg> *License:* unknown *Contributors:* Dmwick

**Image:Toywolf-construction-top.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Toywolf-construction-top.jpg> *License:* unknown *Contributors:* Dmwick

**Image:Toywolf-construction-front.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Toywolf-construction-front.jpg> *License:* unknown *Contributors:* Dmwick

**Image:Toywolf-left.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Toywolf-left.jpg> *License:* unknown *Contributors:* Dmwick

**Image:Toywolf-front.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Toywolf-front.jpg> *License:* unknown *Contributors:* Dmwick

**Image:Toywolf-top.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Toywolf-top.jpg> *License:* unknown *Contributors:* Dmwick

**Image:Wolftut1.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Wolftut1.png> *License:* GNU General Public License *Contributors:* AerieZen

**Image:Wolftut2.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Wolftut2.png> *License:* GNU General Public License *Contributors:* AerieZen

**Image:Wolftut3.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Wolftut3.png> *License:* GNU General Public License *Contributors:* AerieZen

**Image:Wolftut4.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Wolftut4.png> *License:* GNU General Public License *Contributors:* AerieZen

**Image:Wolftut5.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Wolftut5.png> *License:* GNU General Public License *Contributors:* AerieZen

**Image:Wolftut6.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Wolftut6.png> *License:* GNU General Public License *Contributors:* AerieZen

**Image:Wolftut7.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Wolftut7.png> *License:* GNU General Public License *Contributors:* AerieZen

**Image:Wolftut8.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Wolftut8.png> *License:* GNU General Public License *Contributors:* AerieZen

**Image:Wolf building side.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Wolf\\_building\\_side.jpg](http://en.wikibooks.org/w/index.php?title=File:Wolf_building_side.jpg) *License:* GNU Free Documentation License *Contributors:* Az1568, Norleaf, 1 anonymous edits

**Image:Wolf building front.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Wolf\\_building\\_front.jpg](http://en.wikibooks.org/w/index.php?title=File:Wolf_building_front.jpg) *License:* GNU Free Documentation License *Contributors:* Norleaf

**Image:Wolf finished.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Wolf\\_finished.jpg](http://en.wikibooks.org/w/index.php?title=File:Wolf_finished.jpg) *License:* GNU Free Documentation License *Contributors:* Norleaf

**Image:blender-background-dialog.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender-background-dialog.jpg> *License:* unknown *Contributors:* Mrfelis, Whiteknight, ZeroOne

**Image:Blender\_background\_image\_dialog.JPG** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_background\\_image\\_dialog.JPG](http://en.wikibooks.org/w/index.php?title=File:Blender_background_image_dialog.JPG) *License:* unknown *Contributors:* Yoyotweak

**Image:Blender\_use\_background\_image\_button.JPG** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_use\\_background\\_image\\_button.JPG](http://en.wikibooks.org/w/index.php?title=File:Blender_use_background_image_button.JPG) *License:* unknown *Contributors:* Yoyotweak

**Image:Blender\_use\_background\_image\_button\_v2p43.JPG** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_use\\_background\\_image\\_button\\_v2p43.JPG](http://en.wikibooks.org/w/index.php?title=File:Blender_use_background_image_button_v2p43.JPG) *License:* unknown *Contributors:* Yoyotweak

**Image:Blender\_image\_select.JPG** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_image\\_select.JPG](http://en.wikibooks.org/w/index.php?title=File:Blender_image_select.JPG) *License:* unknown *Contributors:* Yoyotweak

**Image:Blender\_image\_select\_button1.JPG** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_image\\_select\\_button1.JPG](http://en.wikibooks.org/w/index.php?title=File:Blender_image_select_button1.JPG) *License:* unknown *Contributors:* Yoyotweak

**Image:Blender\_image\_select\_button2.JPG** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_image\\_select\\_button2.JPG](http://en.wikibooks.org/w/index.php?title=File:Blender_image_select_button2.JPG) *License:* unknown *Contributors:* Yoyotweak

**Image:Blender\_image\_select\_textbox.JPG** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_image\\_select\\_textbox.JPG](http://en.wikibooks.org/w/index.php?title=File:Blender_image_select_textbox.JPG) *License:* unknown *Contributors:* Yoyotweak

**Image:Blender\_image\_remove.JPG** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_image\\_remove.JPG](http://en.wikibooks.org/w/index.php?title=File:Blender_image_remove.JPG) *License:* unknown *Contributors:* Yoyotweak

**Image:Blender\_refresh.JPG** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_refresh.JPG](http://en.wikibooks.org/w/index.php?title=File:Blender_refresh.JPG) *License:* unknown *Contributors:* Yoyotweak

**Image:Blender\_users\_count.JPG** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_users\\_count.JPG](http://en.wikibooks.org/w/index.php?title=File:Blender_users_count.JPG) *License:* unknown *Contributors:* Yoyotweak

**Image:blender-bezier-curve-initial.PNG** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender-bezier-curve-initial.PNG> *License:* unknown *Contributors:* Mrfelis, Whiteknight, ZeroOne

**Image:blender-bezier-curve-sample.PNG** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender-bezier-curve-sample.PNG> *License:* unknown *Contributors:* Mrfelis, Whiteknight, ZeroOne

**Image:blender-curve-tools-convert.PNG** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender-curve-tools-convert.PNG> *License:* unknown *Contributors:* Mrfelis, Whiteknight, ZeroOne

**Image:blender-2dlogo-rough-trace-points.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender-2dlogo-rough-trace-points.png> *License:* unknown *Contributors:* Mrfelis, Whiteknight, ZeroOne

**Image:blender-2dlogo-rough-trace-final.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender-2dlogo-rough-trace-final.png> *License:* unknown *Contributors:* Mrfelis, Whiteknight, ZeroOne

**Image:bezier\_circle1.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Bezier\\_circle1.png](http://en.wikibooks.org/w/index.php?title=File:Bezier_circle1.png) *License:* GNU Free Documentation License *Contributors:* Kieran

**Image:bezier\_circle2.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Bezier\\_circle2.png](http://en.wikibooks.org/w/index.php?title=File:Bezier_circle2.png) *License:* GNU Free Documentation License *Contributors:* Kieran

**Image:bezier\_circle3.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Bezier\\_circle3.png](http://en.wikibooks.org/w/index.php?title=File:Bezier_circle3.png) *License:* GNU Free Documentation License *Contributors:* Kieran

**Image:bezier\_circle4.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Bezier\\_circle4.png](http://en.wikibooks.org/w/index.php?title=File:Bezier_circle4.png) *License:* GNU Free Documentation License *Contributors:* Kieran

**Image:bezier\_circle\_5.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Bezier\\_circle\\_5.png](http://en.wikibooks.org/w/index.php?title=File:Bezier_circle_5.png) *License:* GNU Free Documentation License *Contributors:* Kieran

**Image:bezier\_circle\_6.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Bezier\\_circle\\_6.png](http://en.wikibooks.org/w/index.php?title=File:Bezier_circle_6.png) *License:* GNU Free Documentation License *Contributors:* Kieran

**Image:bezier\_circle\_8.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Bezier\\_circle\\_8.png](http://en.wikibooks.org/w/index.php?title=File:Bezier_circle_8.png) *License:* GNU Free Documentation License *Contributors:* Kieran

**Image:bezier\_circle\_9b.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Bezier\\_circle\\_9b.png](http://en.wikibooks.org/w/index.php?title=File:Bezier_circle_9b.png) *License:* GNU Free Documentation License *Contributors:* Kieran

**Image:bezier\_circle\_10b.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Bezier\\_circle\\_10b.png](http://en.wikibooks.org/w/index.php?title=File:Bezier_circle_10b.png) *License:* GNU Free Documentation License *Contributors:* Kieran

**Image:logo\_final.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Logo\\_final.jpg](http://en.wikibooks.org/w/index.php?title=File:Logo_final.jpg) *License:* GNU Free Documentation License *Contributors:* Kieran

**Image:Johno\_logofinished.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Johno\\_logofinished.jpg](http://en.wikibooks.org/w/index.php?title=File:Johno_logofinished.jpg) *License:* unknown *Contributors:* Johno

**Image:Johno\_logocircle1.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Johno\\_logocircle1.jpg](http://en.wikibooks.org/w/index.php?title=File:Johno_logocircle1.jpg) *License:* unknown *Contributors:* Johno

**Image:Johno\_logocircle2.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Johno\\_logocircle2.jpg](http://en.wikibooks.org/w/index.php?title=File:Johno_logocircle2.jpg) *License:* unknown *Contributors:* Johno

**Image:Johno\_logocircle3.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Johno\\_logocircle3.jpg](http://en.wikibooks.org/w/index.php?title=File:Johno_logocircle3.jpg) *License:* unknown *Contributors:* Johno

**Image:Johno\_logocircle4.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Johno\\_logocircle4.jpg](http://en.wikibooks.org/w/index.php?title=File:Johno_logocircle4.jpg) *License:* unknown *Contributors:* Johno

**Image:Johno\_diag\_1.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Johno\\_diag\\_1.jpg](http://en.wikibooks.org/w/index.php?title=File:Johno_diag_1.jpg) *License:* Public Domain *Contributors:* Johno

**image:Firstcircle.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Firstcircle.png> *License:* GNU Free Documentation License *Contributors:* User:Darkonc

**image:LogoTwoface.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:LogoTwoface.png> *License:* GNU Free Documentation License *Contributors:* User:Darkonc

**image:CircleSquare2.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:CircleSquare2.png> *License:* GNU Free Documentation License *Contributors:* User:Darkonc

**image:RingBar1.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:RingBar1.png> *License:* GNU Free Documentation License *Contributors:* User:Darkonc

**image:RingBar2.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:RingBar2.png> *License:* GNU Free Documentation License *Contributors:* User:Darkonc

**image:RingBar3.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:RingBar3.png> *License:* GNU Free Documentation License *Contributors:* User:Darkonc

**image:RingBar4.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:RingBar4.png> *License:* GNU Free Documentation License *Contributors:* User:Darkonc

**Image>New\_Bezier\_Curve.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File>New\\_Bezier\\_Curve.png](http://en.wikibooks.org/w/index.php?title=File>New_Bezier_Curve.png) *License:* Public Domain *Contributors:* Lazy-lump

**Image:Face\_Bezier\_Curve.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Face\\_Bezier\\_Curve.png](http://en.wikibooks.org/w/index.php?title=File:Face_Bezier_Curve.png) *License:* Public Domain *Contributors:* Lazy-lump

**Image:Curve\_and\_Surface\_Box.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Curve\\_and\\_Surface\\_Box.png](http://en.wikibooks.org/w/index.php?title=File:Curve_and_Surface_Box.png) *License:* Public Domain *Contributors:* Lazy-lump

**Image:Face\_Bezier\_Curve\_Extruded.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Face\\_Bezier\\_Curve\\_Extruded.png](http://en.wikibooks.org/w/index.php?title=File:Face_Bezier_Curve_Extruded.png) *License:* Public Domain *Contributors:* Lazy-lump

**Image:Bezier\_Face\_Bevel.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Bezier\\_Face\\_Bevel.png](http://en.wikibooks.org/w/index.php?title=File:Bezier_Face_Bevel.png) *License:* Public Domain *Contributors:* Lazy-lump

**Image:Bezier\_Face\_Bevel\_4.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Bezier\\_Face\\_Bevel\\_4.png](http://en.wikibooks.org/w/index.php?title=File:Bezier_Face_Bevel_4.png) *License:* Public Domain *Contributors:* Lazy-lump

**Image>New\_Bezier\_Curve\_3D.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File>New\\_Bezier\\_Curve\\_3D.png](http://en.wikibooks.org/w/index.php?title=File>New_Bezier_Curve_3D.png) *License:* Public Domain *Contributors:* Lazy-lump

**Image:ConeCurve1.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:ConeCurve1.jpg> *License:* GNU General Public License *Contributors:* Embri

**Image:ConeCurve2.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:ConeCurve2.jpg> *License:* GNU General Public License *Contributors:* Embri

**Image:ConeCurve3.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:ConeCurve3.jpg> *License:* GNU General Public License *Contributors:* Embri

**Image:ConeCurve4.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:ConeCurve4.jpg> *License:* GNU General Public License *Contributors:* Embri

**File:Blender3D\_MaterialButtons-2.49.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_MaterialButtons-2.49.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_MaterialButtons-2.49.png) *License:* GNU General Public License *Contributors:* User:SoylentGreen

**Image:delete.gif** *Source:* <http://en.wikibooks.org/w/index.php?title=File>Delete.gif> *License:* unknown *Contributors:* Gregggreg, Reece, ZeroOne

**Image:dropdown.gif** *Source:* http://en.wikibooks.org/w/index.php?title=File:Dropdown.gif *License:* unknown *Contributors:* Greggreg, Reece, ZeroOne

**File:Blender3D\_AutoNamingButton-2.49.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender3D\_AutoNamingButton-2.49.png *License:* GNU General Public License *Contributors:* User:SoylentGreen

**Image:materialrgb.gif** *Source:* http://en.wikibooks.org/w/index.php?title=File:Materialrgb.gif *License:* GNU General Public License *Contributors:* Darklama, Greggreg, ZeroOne

**Image:materialColorPicker.gif** *Source:* http://en.wikibooks.org/w/index.php?title=File:MaterialColorPicker.gif *License:* GNU General Public License *Contributors:* Darklama, Greggreg, ZeroOne

**File:Blender3D\_color\_picker.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender3D\_color\_picker.jpg *License:* GNU Free Documentation License *Contributors:* User:Toni\_grappa

**Image:FirstMaterialPreview.PNG** *Source:* http://en.wikibooks.org/w/index.php?title=File:FirstMaterialPreview.PNG *License:* GNU Free Documentation License *Contributors:* Siesta

**Image:FirstMaterialSettings.PNG** *Source:* http://en.wikibooks.org/w/index.php?title=File:FirstMaterialSettings.PNG *License:* GNU Free Documentation License *Contributors:* Siesta

**File:Blendertexture.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blendertexture.png *License:* unknown *Contributors:* Arturo Reina, Bayo, OsamaKAWB

**File:Blender3D\_BasicCloudsTexture-2.49.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender3D\_BasicCloudsTexture-2.49.png *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D\_MapToPanelStreichholz.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender3D\_MapToPanelStreichholz.png *License:* GNU General Public License *Contributors:* User:SoylentGreen

**Image:CloudTexturePreview.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:CloudTexturePreview.png *License:* GNU Free Documentation License *Contributors:* Siesta

**File:Blender-DefaultSceneRender.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender-DefaultSceneRender.jpg *License:* unknown *Contributors:* N/A

**File:Blender-DefaultSceneRender\_Halo.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender-DefaultSceneRender\_Halo.jpg *License:* unknown *Contributors:* N/A

**File:Blender-Halo\_Navigation1.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender-Halo\_Navigation1.png *License:* unknown *Contributors:* N/A

**File:Blender-Halo\_Navigation2.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender-Halo\_Navigation2.png *License:* unknown *Contributors:* N/A

**File:Blender-Halo\_Navigation3.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender-Halo\_Navigation3.png *License:* unknown *Contributors:* N/A

**File:Blender-Halo\_Navigation4.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender-Halo\_Navigation4.png *License:* unknown *Contributors:* N/A

**File:Blender-Halo\_Navigation5.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender-Halo\_Navigation5.png *License:* unknown *Contributors:* N/A

**File:Blender-Halo\_Navigation6.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender-Halo\_Navigation6.png *License:* unknown *Contributors:* N/A

**Image:Blender-seawater.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender-seawater.png *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* User:Dobz116

**Image:Ground1.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ground1.jpg *License:* Creative Commons Attribution 2.5 *Contributors:* GeorgHH, Serman, 2 anonymous edits

**File:Blender\_Hill.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender\_Hill.jpg *License:* Public Domain *Contributors:* Adrignola, Ask

**Image:Basic-scene.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Basic-scene.jpg *License:* GNU Free Documentation License *Contributors:* Original uploader was Dan13 at en.wikibooks

**Image:Step1-scr.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Step1-scr.jpg *License:* unknown *Contributors:* Dan13

**Image:Step1-base-shader.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Step1-base-shader.jpg *License:* unknown *Contributors:* Dan13

**Image:Texture1-tex-panel.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Texture1-tex-panel.jpg *License:* unknown *Contributors:* Dan13

**Image:Texture1-mat-panel.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Texture1-mat-panel.jpg *License:* unknown *Contributors:* Dan13

**Image:Texture2-tex-panel.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Texture2-tex-panel.jpg *License:* unknown *Contributors:* Dan13

**Image:Texture2-mat-panel.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Texture2-mat-panel.jpg *License:* unknown *Contributors:* Dan13

**Image:Texture2-render.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Texture2-render.jpg *License:* unknown *Contributors:* Dan13

**Image:Texture3-tex-panel.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Texture3-tex-panel.jpg *License:* unknown *Contributors:* Dan13

**Image:Texture3-mat-panel.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Texture3-mat-panel.jpg *License:* unknown *Contributors:* Dan13

**Image:Texture3-render.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Texture3-render.jpg *License:* unknown *Contributors:* Dan13

**Image:Texture4-tex-panel.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Texture4-tex-panel.jpg *License:* unknown *Contributors:* Dan13

**Image:Texture4-mat-panel.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Texture4-mat-panel.jpg *License:* unknown *Contributors:* Dan13

**Image:Carpet-Material-Final.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Carpet-Material-Final.jpg *License:* unknown *Contributors:* Dan13

**Image:Pixareyes-iris\_blueeye.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Pixareyes-iris\_blueeye.png *License:* Public Domain *Contributors:* Original uploader was Hrakaroo at en.wikibooks

**Image:Pixareyes-Eye\_mesh.gif** *Source:* http://en.wikibooks.org/w/index.php?title=File:Pixareyes-Eye\_mesh.gif *License:* Public Domain *Contributors:* Derbeth, Rdp, WeirdHat, ZeroOne

**Image:Cornea-example.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Cornea-example.jpg *License:* GNU General Public License *Contributors:* Original uploader was Icky-wiki at en.wikibooks

**Image:Pixareyes-iriscoloredit.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Pixareyes-iriscoloredit.png *License:* GNU General Public License *Contributors:* Original uploader was Hrakaroo at en.wikibooks

**Image:Iris.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Iris.png *License:* GNU General Public License *Contributors:* Original uploader was ThomasB at en.wikibooks

**Image:settings1.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Settings1.jpg *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Original uploader was Chinmay gautam at en.wikibooks

**Image:blueeyes.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blueeyes.jpg *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Original uploader was Chinmay gautam at en.wikibooks

**File:Greeneyes(1).jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Greeneyes(1).jpg *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Original uploader was Chinmay gautam at en.wikibooks

**Image:redeyes.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Redeyes.jpg *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* Original uploader was Chinmay gautam at en.wikibooks

**Image:Pixareyes-iris\_blueeye.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Pixareyes-iris\_blueeye.png *License:* Public Domain *Contributors:* Original uploader was Hrakaroo at en.wikibooks

**Image:Pixareyes-iris\_greeneye.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Pixareyes-iris\_greeneye.png *License:* Public Domain *Contributors:* Original uploader was Hrakaroo at en.wikibooks

**Image:Pixareyes-iris\_redeye.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Pixareyes-iris\_redeye.png *License:* Public Domain *Contributors:* Original uploader was Hrakaroo at en.wikibooks

**Image:Blender\_icosphere.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender\_icosphere.png *License:* unknown *Contributors:* User:ZeroOne

**Image:Blender\_icosphere\_with\_marked\_seams.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender\_icosphere\_with\_marked\_seams.png *License:* unknown *Contributors:* User:ZeroOne

**Image:Blender icosphere and UV Image Editor.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender\_icosphere\_and\_UV\_Image\_Editor.png *License:* unknown *Contributors:* User:ZeroOne

**Image:Blender icosphere LSCM.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender\_icosphere\_LSCM.png *License:* unknown *Contributors:* User:ZeroOne

**Image:BlueMarble-2001-2002.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:BlueMarble-2001-2002.jpg *License:* Public Domain *Contributors:* User:Apoc2400

**Image:Blender icosphere textured.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender\_icosphere\_textured.png *License:* unknown *Contributors:* User:ZeroOne

**Image:Blender icosphere TexFace.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender\_icosphere\_TexFace.png *License:* unknown *Contributors:* User:ZeroOne

**Image:Blender Blue Marble.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender\_Blue\_Marble.jpg *License:* unknown *Contributors:* User:ZeroOne

**Image:Blender map input.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_map\\_input.png](http://en.wikibooks.org/w/index.php?title=File:Blender_map_input.png) *License:* GNU General Public License *Contributors:* Argento

**Image:Falloff\_Edit.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Falloff\\_Edit.jpg](http://en.wikibooks.org/w/index.php?title=File:Falloff_Edit.jpg) *License:* Public Domain *Contributors:* Quantum Anomaly

**Image:Cornea\_Curve.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Cornea\\_Curve.jpg](http://en.wikibooks.org/w/index.php?title=File:Cornea_Curve.jpg) *License:* Public Domain *Contributors:* Quantum Anomaly

**Image:UV\_SubSurf.gif** *Source:* [http://en.wikibooks.org/w/index.php?title=File:UV\\_SubSurf.gif](http://en.wikibooks.org/w/index.php?title=File:UV_SubSurf.gif) *License:* Public Domain *Contributors:* Quantum Anomaly

**Image:Cornea\_Triangles.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Cornea\\_Triangles.jpg](http://en.wikibooks.org/w/index.php?title=File:Cornea_Triangles.jpg) *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* Quantum Anomaly

**Image:SubDivide\_Smooth.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:SubDivide\\_Smooth.jpg](http://en.wikibooks.org/w/index.php?title=File:SubDivide_Smooth.jpg) *License:* Public Domain *Contributors:* Quantum Anomaly

**Image:UV\_Seams.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:UV\\_Seams.jpg](http://en.wikibooks.org/w/index.php?title=File:UV_Seams.jpg) *License:* Public Domain *Contributors:* Quantum Anomaly

**Image:UV\_Unwrap.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:UV\\_Unwrap.jpg](http://en.wikibooks.org/w/index.php?title=File:UV_Unwrap.jpg) *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* Quantum Anomaly

**Image:UV\_Layout>Edit\_A.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:UV\\_Layout>Edit\\_A.jpg](http://en.wikibooks.org/w/index.php?title=File:UV_Layout>Edit_A.jpg) *License:* Public Domain *Contributors:* Quantum Anomaly

**Image:UV\_Layout>Edit\_B.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:UV\\_Layout>Edit\\_B.jpg](http://en.wikibooks.org/w/index.php?title=File:UV_Layout>Edit_B.jpg) *License:* Public Domain *Contributors:* Quantum Anomaly

**Image:UV\_Untangling.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:UV\\_Untangling.jpg](http://en.wikibooks.org/w/index.php?title=File:UV_Untangling.jpg) *License:* Public Domain *Contributors:* Quantum Anomaly

**Image:UV\_Finished.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:UV\\_Finished.jpg](http://en.wikibooks.org/w/index.php?title=File:UV_Finished.jpg) *License:* Public Domain *Contributors:* Quantum Anomaly

**Image:Save\_UV\_Face\_Layout.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Save\\_UV\\_Face\\_Layout.jpg](http://en.wikibooks.org/w/index.php?title=File:Save_UV_Face_Layout.jpg) *License:* GNU General Public License *Contributors:* Darklama, Mike.lifeguard, Quantum Anomaly

**Image:Marking\_The\_Middle.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Marking\\_The\\_Middle.jpg](http://en.wikibooks.org/w/index.php?title=File:Marking_The_Middle.jpg) *License:* Public Domain *Contributors:* Quantum Anomaly

**Image:Texture\_Bake\_MapTo.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Texture\\_Bake\\_MapTo.jpg](http://en.wikibooks.org/w/index.php?title=File:Texture_Bake_MapTo.jpg) *License:* GNU General Public License *Contributors:* Darklama, Quantum Anomaly

**Image:Texture\_Bake\_Material.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Texture\\_Bake\\_Material.jpg](http://en.wikibooks.org/w/index.php?title=File:Texture_Bake_Material.jpg) *License:* GNU General Public License *Contributors:* Darklama, Quantum Anomaly

**Image:Texture\_Bake\_Preview.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Texture\\_Bake\\_Preview.jpg](http://en.wikibooks.org/w/index.php?title=File:Texture_Bake_Preview.jpg) *License:* GNU General Public License *Contributors:* Darklama, Quantum Anomaly

**Image>New\_Image.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File>New\\_Image.jpg](http://en.wikibooks.org/w/index.php?title=File>New_Image.jpg) *License:* GNU General Public License *Contributors:* Darklama, Quantum Anomaly

**Image:Bake\_And\_Save.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Bake\\_And\\_Save.jpg](http://en.wikibooks.org/w/index.php?title=File:Bake_And_Save.jpg) *License:* unknown *Contributors:* Original uploader was Quantum Anomaly at en.wikibooks

**Image:Bcup7.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bcup7.png> *License:* GNU General Public License *Contributors:* AverieZen

**Image:Bcup8.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bcup8.png> *License:* GNU General Public License *Contributors:* AverieZen

**Image:Bcup9.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bcup9.png> *License:* GNU General Public License *Contributors:* AverieZen

**Image:Bcup10.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bcup10.png> *License:* GNU General Public License *Contributors:* AverieZen

**Image:Bcup11.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bcup11.png> *License:* GNU General Public License *Contributors:* AverieZen

**Image:Bcup12.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bcup12.png> *License:* GNU General Public License *Contributors:* AverieZen

**Image:Bcup13.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bcup13.png> *License:* GNU General Public License *Contributors:* AverieZen

**Image:Bcup14.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bcup14.png> *License:* GNU General Public License *Contributors:* AverieZen

**Image:Bcup15.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bcup15.png> *License:* GNU General Public License *Contributors:* Adrignola, AverieZen, 1 anonymous edits

**Image:cup.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Cup.jpg> *License:* GNU Free Documentation License *Contributors:* Clone Dad, ZeroOne

**Image:RenderResult.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:RenderResult.jpg> *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* Radialronnie

**Image:TutFlipNorms-20081103140255.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:TutFlipNorms-20081103140255.png> *License:* unknown *Contributors:* Radialronnie

**Image:EnableDupVerts.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:EnableDupVerts.png> *License:* unknown *Contributors:* Radialronnie

**Image:SetAAO-2008110322151223.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:SetAAO-2008110322151223.png> *License:* unknown *Contributors:* Radialronnie

**Image:RenderScene-20081103151408.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:RenderScene-20081103151408.png> *License:* unknown *Contributors:* Radialronnie

**Image:SetWireDrawType.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:SetWireDrawType.png> *License:* unknown *Contributors:* Radialronnie

**Image:Render\_Settings.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Render\\_Settings.png](http://en.wikibooks.org/w/index.php?title=File:Render_Settings.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image:Output\_tab.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Output\\_tab.png](http://en.wikibooks.org/w/index.php?title=File:Output_tab.png) *License:* unknown *Contributors:* Soeb

**Image:File locator icon.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:File\\_locator\\_icon.png](http://en.wikibooks.org/w/index.php?title=File:File_locator_icon.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image:Render\_Layers\_tab.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Render\\_Layers\\_tab.png](http://en.wikibooks.org/w/index.php?title=File:Render_Layers_tab.png) *License:* unknown *Contributors:* Soeb

**Image:Render settings backbuf row.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Render\\_settings\\_backbuf\\_row.png](http://en.wikibooks.org/w/index.php?title=File:Render_settings_backbuf_row.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image>Lastbuttonrow.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File>Lastbuttonrow.png> *License:* unknown *Contributors:* Soeb, Webaware

**Image:Render tab.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Render\\_tab.png](http://en.wikibooks.org/w/index.php?title=File:Render_tab.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image:Anim\_tab.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Anim\\_tab.png](http://en.wikibooks.org/w/index.php?title=File:Anim_tab.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image:Bake\_tab.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Bake\\_tab.png](http://en.wikibooks.org/w/index.php?title=File:Bake_tab.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image:Format tab.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Format\\_tab.png](http://en.wikibooks.org/w/index.php?title=File:Format_tab.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image:Format\_settings\_sizeX.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Format\\_settings\\_sizeX.png](http://en.wikibooks.org/w/index.php?title=File:Format_settings_sizeX.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image:Format\_settings\_sizeY.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Format\\_settings\\_sizeY.png](http://en.wikibooks.org/w/index.php?title=File:Format_settings_sizeY.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image:Format\_settings\_aspXaspY.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Format\\_settings\\_aspXaspY.png](http://en.wikibooks.org/w/index.php?title=File:Format_settings_aspXaspY.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image:Format\_settings\_drop-down\_box.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Format\\_settings\\_drop-down\\_box.png](http://en.wikibooks.org/w/index.php?title=File:Format_settings_drop-down_box.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image:Format\_settings\_crop\_button.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Format\\_settings\\_crop\\_button.png](http://en.wikibooks.org/w/index.php?title=File:Format_settings_crop_button.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image:Format\_settings\_palette\_selection.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Format\\_settings\\_palette\\_selection.png](http://en.wikibooks.org/w/index.php?title=File:Format_settings_palette_selection.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image:Format\_settings\_pre-sets.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Format\\_settings\\_pre-sets.png](http://en.wikibooks.org/w/index.php?title=File:Format_settings_pre-sets.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image:Render\_settings\_OSA\_row.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Render\\_settings\\_OSA\\_row.png](http://en.wikibooks.org/w/index.php?title=File:Render_settings_OSA_row.png) *License:* unknown *Contributors:* Soeb, Webaware

**Image:OSA\_comparison.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:OSA\\_comparison.png](http://en.wikibooks.org/w/index.php?title=File:OSA_comparison.png) *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* Soeb

**Image:BlenderPanoramaA.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderPanoramaA.png> *License:* unknown *Contributors:* Allefant, Whiteknight, ZeroOne

**Image:BlenderPanoramaB.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderPanoramaB.png> *License:* unknown *Contributors:* Allefant, Whiteknight, ZeroOne

**Image:BlenderPanoramaC.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderPanoramaC.png> *License:* unknown *Contributors:* Allefant, Whiteknight, ZeroOne

**Image:BlenderButtonsShadingWorld.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderButtonsShadingWorld.png> *License:* unknown *Contributors:* Allefant, Whiteknight, ZeroOne

**Image:BlenderPanoramaD.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderPanoramaD.png> *License:* unknown *Contributors:* Allefant, Whiteknight, ZeroOne

**Image:BlenderCameraSettings.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderCameraSettings.png> *License:* unknown *Contributors:* Allefant, Whiteknight, ZeroOne

**Image:BlenderPanoramaE.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderPanoramaE.jpg> *License:* unknown *Contributors:* Allefant, Whiteknight, ZeroOne

**Image:BlenderPanoramaF.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderPanoramaF.jpg> *License:* unknown *Contributors:* Allefant, Whiteknight, ZeroOne

**Image:Standardlamp.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Standardlamp.jpg> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* SergeantOreo

**Image:Arealamp.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Arealamp.jpg> *License:* GNU Free Documentation License *Contributors:* Original uploader was SergeantOreo at en.wikibooks

**Image:Spotlightlamp.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Spotlightlamp.jpg> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* SergeantOreo

**Image:Sunlamp.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Sunlamp.jpg> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* SergeantOreo

**Image:Hemilamp.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Hemilamp.jpg> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* SergeantOreo

**Image:Blender\_SetSmooth.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_SetSmooth.png](http://en.wikibooks.org/w/index.php?title=File:Blender_SetSmooth.png) *License:* Public Domain *Contributors:* Lazy-lump

**Image:TopPlane.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:TopPlane.png> *License:* unknown *Contributors:* Reece, Totophe64, ZeroOne

**Image:CameraView.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:CameraView.png> *License:* GNU General Public License *Contributors:* Darklama, Totophe64, ZeroOne

**Image:MakeTrack.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:MakeTrack.png> *License:* GNU General Public License *Contributors:* Darklama, Totophe64, ZeroOne

**Image:CamTracked.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:CamTracked.png> *License:* GNU General Public License *Contributors:* Darklama, Totophe64, ZeroOne

**Image:Lamp.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Lamp.png> *License:* GNU General Public License *Contributors:* Darklama, Totophe64, ZeroOne

**Image:Settings.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Settings.png> *License:* unknown *Contributors:* Reece, Totophe64, ZeroOne

**Image:Coulours.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Coulours.png> *License:* GNU General Public License *Contributors:* Darklama, Totophe64, ZeroOne

**Image:BufShad.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BufShad.png> *License:* unknown *Contributors:* Reece, Totophe64, ZeroOne

**File:Blender Render1.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_Render1.png](http://en.wikibooks.org/w/index.php?title=File:Blender_Render1.png) *License:* GNU Free Documentation License *Contributors:* Adrignola, Mike.lifeguard, Totophe64, ZeroOne

**Image:Render2.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Render2.png> *License:* GNU Free Documentation License *Contributors:* Mike.lifeguard, Totophe64, ZeroOne

**Image:Spec.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Spec.png> *License:* unknown *Contributors:* MichaelFrey, Totophe64, ZeroOne

**Image:Render3.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Render3.png> *License:* GNU Free Documentation License *Contributors:* Mike.lifeguard, Totophe64, ZeroOne

**File:Blender3D\_StampedNia.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_StampedNia.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_StampedNia.jpg) *License:* GNU Free Documentation License *Contributors:* User:Tictoon

**File:Blender3D\_DefaultScene-Blender2.48.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_DefaultScene-Blender2.48.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_DefaultScene-Blender2.48.png) *License:* GNU General Public License *Contributors:* TictoonTictoon

**File:Blender3D\_SceneButton-Blender2.48.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_SceneButton-Blender2.48.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_SceneButton-Blender2.48.png) *License:* GNU General Public License *Contributors:* User:Tictoon

**File:Blender3D\_FormatPanel-Blender2.48.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_FormatPanel-Blender2.48.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_FormatPanel-Blender2.48.png) *License:* GNU General Public License *Contributors:* User:Tictoon

**File:Blender3D\_EnableStampButton-Blender2.48.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_EnableStampButton-Blender2.48.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_EnableStampButton-Blender2.48.png) *License:* GNU General Public License *Contributors:* User:Tictoon

**File:Blender3D\_StampingPanel-Blender2.48.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_StampingPanel-Blender2.48.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_StampingPanel-Blender2.48.png) *License:* GNU General Public License *Contributors:* User:Tictoon

**Image:ScreenLayout.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:ScreenLayout.png> *License:* GNU General Public License *Contributors:* TBOL3

**image:AnimationSetup.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:AnimationSetup.png> *License:* unknown *Contributors:* Original uploader was TBOL3 at en.wikibooks

**Image:Lattice\_AddModifier.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Lattice\\_AddModifier.jpg](http://en.wikibooks.org/w/index.php?title=File:Lattice_AddModifier.jpg) *License:* unknown *Contributors:* AndyD

**Image:Lattice\_UVWsets.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Lattice\\_UVWsets.jpg](http://en.wikibooks.org/w/index.php?title=File:Lattice_UVWsets.jpg) *License:* GNU General Public License *Contributors:* AndyD

**Image:Lattice\_UVWtypes.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Lattice\\_UVWtypes.jpg](http://en.wikibooks.org/w/index.php?title=File:Lattice_UVWtypes.jpg) *License:* GNU General Public License *Contributors:* AndyD

**Image:Lattice\_complex.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Lattice\\_complex.jpg](http://en.wikibooks.org/w/index.php?title=File:Lattice_complex.jpg) *License:* GNU General Public License *Contributors:* AndyD

**Image:Lattice\_Samples.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Lattice\\_Samples.jpg](http://en.wikibooks.org/w/index.php?title=File:Lattice_Samples.jpg) *License:* GNU General Public License *Contributors:* AndyD

**Image:Lattice\_AddKey.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Lattice\\_AddKey.jpg](http://en.wikibooks.org/w/index.php?title=File:Lattice_AddKey.jpg) *License:* unknown *Contributors:* AndyD

**Image:Lattice\_revkey.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Lattice\\_revkey.jpg](http://en.wikibooks.org/w/index.php?title=File:Lattice_revkey.jpg) *License:* GNU General Public License *Contributors:* AndyD

**Image:Lattice\_sliders.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Lattice\\_sliders.jpg](http://en.wikibooks.org/w/index.php?title=File:Lattice_sliders.jpg) *License:* GNU General Public License *Contributors:* AndyD

**Image:Lattice\_Animkeys.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Lattice\\_Animkeys.jpg](http://en.wikibooks.org/w/index.php?title=File:Lattice_Animkeys.jpg) *License:* GNU General Public License *Contributors:* AndyD

**Image:Lattice\_eyes\_without.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Lattice\\_eyes\\_without.jpg](http://en.wikibooks.org/w/index.php?title=File:Lattice_eyes_without.jpg) *License:* GNU General Public License *Contributors:* AndyD

**Image:Lattice\_eyes\_with.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Lattice\\_eyes\\_with.jpg](http://en.wikibooks.org/w/index.php?title=File:Lattice_eyes_with.jpg) *License:* GNU General Public License *Contributors:* AndyD

**Image:BounceballlatticeSS.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BounceballlatticeSS.jpg> *License:* GNU General Public License *Contributors:* AndyD

**Image:Bouncelatmod2SS.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bouncelatmod2SS.jpg> *License:* GNU General Public License *Contributors:* AndyD

**Image:Bouncelatmod1SS.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bouncelatmod1SS.jpg> *License:* GNU General Public License *Contributors:* AndyD

**Image:BouncepivotSS.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BouncepivotSS.jpg> *License:* GNU General Public License *Contributors:* AndyD

**Image:BouncebasekeysSS.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BouncebasekeysSS.jpg> *License:* GNU General Public License *Contributors:* AndyD

**Image:BounceipoSS.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BounceipoSS.jpg> *License:* GNU General Public License *Contributors:* AndyD

**Image:BounceipokesSS.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BounceipokesSS.jpg> *License:* GNU General Public License *Contributors:* AndyD

**Image:Bouncewiki.gif** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Bouncewiki.gif> *License:* unknown *Contributors:* AndyD

**Image:Blender\_fluid-demo.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_fluid-demo.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_fluid-demo.jpg) *License:* unknown *Contributors:* User:Malaz

**Image:1\_SetOutputFormat.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:1\\_SetOutputFormat.jpg](http://en.wikibooks.org/w/index.php?title=File:1_SetOutputFormat.jpg) *License:* unknown *Contributors:* Original uploader was Radialronnie at en.wikibooks

**Image:2\_SetSaveLocation.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:2\\_SetSaveLocation.jpg](http://en.wikibooks.org/w/index.php?title=File:2_SetSaveLocation.jpg) *License:* unknown *Contributors:* Radialronnie

**Image:3\_SetEndFrame.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:3\\_SetEndFrame.jpg](http://en.wikibooks.org/w/index.php?title=File:3_SetEndFrame.jpg) *License:* unknown *Contributors:* Radialronnie

**Image:4\_RenderAnimButton.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:4\\_RenderAnimButton.jpg](http://en.wikibooks.org/w/index.php?title=File:4_RenderAnimButton.jpg) *License:* unknown *Contributors:* Radialronnie

**Image:5\_Change3D\_WinToSeqWin.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:5\\_Change3D\\_WinToSeqWin.jpg](http://en.wikibooks.org/w/index.php?title=File:5_Change3D_WinToSeqWin.jpg) *License:* unknown *Contributors:* Radialronnie

**Image:6\_SeqWinAddImages.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:6\\_SeqWinAddImages.jpg](http://en.wikibooks.org/w/index.php?title=File:6_SeqWinAddImages.jpg) *License:* unknown *Contributors:* Radialronnie

**Image:7\_SeqImagesPlacement.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:7\\_SeqImagesPlacement.jpg](http://en.wikibooks.org/w/index.php?title=File:7_SeqImagesPlacement.jpg) *License:* unknown *Contributors:* Radialronnie

**Image:8\_SetMovieFormat.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:8\\_SetMovieFormat.jpg](http://en.wikibooks.org/w/index.php?title=File:8_SetMovieFormat.jpg) *License:* unknown *Contributors:* Radialronnie

**Image:9\_DoSeqAndAnimate.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:9\\_DoSeqAndAnimate.jpg](http://en.wikibooks.org/w/index.php?title=File:9_DoSeqAndAnimate.jpg) *License:* unknown *Contributors:* Radialronnie

**File:Blender3D\_AddAParticleSystem-2.48a.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_AddAParticleSystem-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_AddAParticleSystem-2.48a.png) *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D\_TheVeryFirstParticleSystem-2.49.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_TheVeryFirstParticleSystem-2.49.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_TheVeryFirstParticleSystem-2.49.png) *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D\_FreeNote.gif** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_FreeNote.gif](http://en.wikibooks.org/w/index.php?title=File:Blender3D_FreeNote.gif) *License:* GNU Lesser General Public License *Contributors:* User:SoylentGreen

**File:Blender3D\_TheVeryFirstParticleSystem\_NormalEmission-2.49.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_TheVeryFirstParticleSystem\\_NormalEmission-2.49.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_TheVeryFirstParticleSystem_NormalEmission-2.49.png) *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D\_TheVeryFirstParticleSystem\_Material-2.49.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_TheVeryFirstParticleSystem\\_Material-2.49.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_TheVeryFirstParticleSystem_Material-2.49.png) *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D\_TheVeryFirstParticleSystem\_Render-2.49.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_TheVeryFirstParticleSystem\\_Render-2.49.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_TheVeryFirstParticleSystem_Render-2.49.jpg) *License:* Public Domain *Contributors:* User:SoylentGreen

**File:Blender3D\_SimpleParticleFire.ogv** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_SimpleParticleFire.ogv](http://en.wikibooks.org/w/index.php?title=File:Blender3D_SimpleParticleFire.ogv) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**File:Blender3D\_SimpleFireSetup-2.48a.jpg** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_SimpleFireSetup-2.48a.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_SimpleFireSetup-2.48a.jpg) License: GNU Free Documentation License  
Contributors: User:SoylentGreen

**File:Blender3D\_SimpleFireParticleSystem-2.48a.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_SimpleFireParticleSystem-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_SimpleFireParticleSystem-2.48a.png) License: GNU General Public License  
Contributors: User:SoylentGreen

**File:Blender3D\_SimpleFireNoMaterial-2.48a.jpg** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_SimpleFireNoMaterial-2.48a.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_SimpleFireNoMaterial-2.48a.jpg) License: GNU Free Documentation License  
Contributors: User:SoylentGreen

**File:Blender3D\_SimpleFireMaterial-2.48a.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_SimpleFireMaterial-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_SimpleFireMaterial-2.48a.png) License: GNU General Public License  
Contributors: User:SoylentGreen

**File:Blender3D\_Lagerfeuer\_Textur.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Lagerfeuer\\_Textur.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Lagerfeuer_Textur.png) License: GNU General Public License  
Contributors: bde:User:EilexeEilexe

**File:Blender3D\_SimpleFireMaterialAnimation-2.48a.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_SimpleFireMaterialAnimation-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_SimpleFireMaterialAnimation-2.48a.png) License: GNU General Public License  
Contributors: User:SoylentGreen

**File:Blender3D\_FurWithParticles-Finished-2.48a.jpg** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_FurWithParticles-Finished-2.48a.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_FurWithParticles-Finished-2.48a.jpg) License: GNU Free Documentation License  
Contributors: User:SoylentGreen

**File:Blender3D\_ParticleSystemForFur-2.48a.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_ParticleSystemForFur-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_ParticleSystemForFur-2.48a.png) License: GNU General Public License  
Contributors: User:SoylentGreen

**File:Blender3D\_FurWithParticles-NoMaterial-2.48a.jpg** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_FurWithParticles-NoMaterial-2.48a.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_FurWithParticles-NoMaterial-2.48a.jpg) License: Public Domain  
Contributors: User:SoylentGreen

**File:Blender3D\_StrandsShaderSettings.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_StrandsShaderSettings.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_StrandsShaderSettings.png) License: GNU General Public License  
Contributors: User:SoylentGreen

**File:Blender3D\_FurTexture\_for\_FurTutorial.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_FurTexture\\_for\\_FurTutorial.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_FurTexture_for_FurTutorial.png) License: Creative Commons Attribution-Sharealike 2.5  
Contributors: User:SoylentGreen

**File:Blender3D\_FurWithParticles-ColorTexture-2.48a.jpg** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_FurWithParticles-ColorTexture-2.48a.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_FurWithParticles-ColorTexture-2.48a.jpg) License: GNU Free Documentation License  
Contributors: User:SoylentGreen

**File:Blender3D\_StrandFadeSettings.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_StrandFadeSettings.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_StrandFadeSettings.png) License: GNU General Public License  
Contributors: User:SoylentGreen

**File:Blender3D\_FurWithParticles-TexturePainting-2.48a.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_FurWithParticles-TexturePainting-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_FurWithParticles-TexturePainting-2.48a.png) License: GNU General Public License  
Contributors: User:SoylentGreen

**File:Blender3D\_FurWithParticles-ParticleLengthSettings-2.48a.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_FurWithParticles-ParticleLengthSettings-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_FurWithParticles-ParticleLengthSettings-2.48a.png) License: GNU General Public License  
Contributors: User:SoylentGreen

**File:Blender3D\_FurWithParticles-ParticleLengthTexture-2.48a.jpg** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_FurWithParticles-ParticleLengthTexture-2.48a.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_FurWithParticles-ParticleLengthTexture-2.48a.jpg) License: GNU Free Documentation License  
Contributors: User:SoylentGreen

**File:Blender3D\_FurWithParticles-Combing-2.48a.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_FurWithParticles-Combing-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_FurWithParticles-Combing-2.48a.png) License: GNU Free Documentation License  
Contributors: User:SoylentGreen

**File:Blender3D\_Fireworks-2.48a.ogv** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Fireworks-2.48a.ogv](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Fireworks-2.48a.ogv) License: GNU Free Documentation License  
Contributors: User:SoylentGreen

**File:Blender3D AddingAParticleSystem.png** Source: <http://en.wikibooks.org/w/index.php?title=File:Blender3D AddingAParticleSystem.png> License: GNU General Public License  
Contributors: User:SoylentGreen

**File:Blender3D\_Fireworks-Reactor1-2.48a.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Fireworks-Reactor1-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Fireworks-Reactor1-2.48a.png) License: GNU General Public License  
Contributors: User:SoylentGreen

**File:Blender3D\_Fireworks-Reactor1-Material-2.48a.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Fireworks-Reactor1-Material-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Fireworks-Reactor1-Material-2.48a.png) License: GNU General Public License  
Contributors: User:SoylentGreen

**File:Blender3D\_Fireworks-Reactor2-2.48a.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Fireworks-Reactor2-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Fireworks-Reactor2-2.48a.png) License: GNU General Public License  
Contributors: User:SoylentGreen

**File:Blender3D\_Fireworks-Reactor2-Material-2.48a.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_Fireworks-Reactor2-Material-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_Fireworks-Reactor2-Material-2.48a.png) License: GNU General Public License  
Contributors: User:SoylentGreen

**Image:tut.sb.1.jpg** Source: <http://en.wikibooks.org/w/index.php?title=File:Tut.sb.1.jpg> License: unknown  
Contributors: Porlando, Webaware, 1 anonymous edits

**Image:tut.sb.2.jpg** Source: <http://en.wikibooks.org/w/index.php?title=File:Tut.sb.2.jpg> License: unknown  
Contributors: Porlando, Webaware

**Image:tut.sb.3.jpg** Source: <http://en.wikibooks.org/w/index.php?title=File:Tut.sb.3.jpg> License: unknown  
Contributors: Porlando, Webaware

**Image:tut.sb.4.jpg** Source: <http://en.wikibooks.org/w/index.php?title=File:Tut.sb.4.jpg> License: Public Domain  
Contributors: Porlando

**Image:BlenderNobProPlatformerSetUp.jpg** Source: <http://en.wikibooks.org/w/index.php?title=File:BlenderNobProPlatformerSetUp.jpg> License: GNU Free Documentation License  
Contributors: LokiClock

**Image:BlenderLogicActor\_Dyn.jpg** Source: [http://en.wikibooks.org/w/index.php?title=File:BlenderLogicActor\\_Dyn.jpg](http://en.wikibooks.org/w/index.php?title=File:BlenderLogicActor_Dyn.jpg) License: GNU Free Documentation License  
Contributors: LokiClock

**image:BlenderLogic\_ForRgt\_Optm.gif** Source: [http://en.wikibooks.org/w/index.php?title=File:BlenderLogic\\_ForRgt\\_Optm.gif](http://en.wikibooks.org/w/index.php?title=File:BlenderLogic_ForRgt_Optm.gif) License: GNU Free Documentation License  
Contributors: LokiClock

**Image:MazeTutorial.01\_grid-select.png** Source: [http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.01\\_grid-select.png](http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.01_grid-select.png) License: Public Domain  
Contributors: User:Malaz

**Image:MazeTutorial.14\_maze-extruded.png** Source: [http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.14\\_maze-extruded.png](http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.14_maze-extruded.png) License: Public Domain  
Contributors: User:Malaz

**Image:MazeTutorial.02\_cone-extrude.png** Source: [http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.02\\_cone-extrude.png](http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.02_cone-extrude.png) License: Public Domain  
Contributors: User:Malaz

**Image:MazeTutorial.03\_cone-align.png** Source: [http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.03\\_cone-align.png](http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.03_cone-align.png) License: Public Domain  
Contributors: User:Malaz

**Image:MazeTutorial.04\_character-logic.png** Source: [http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.04\\_character-logic.png](http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.04_character-logic.png) License: Public Domain  
Contributors: User:Malaz

**Image:MazeTutorial.05\_mazegame-test.png** Source: [http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.05\\_mazegame-test.png](http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.05_mazegame-test.png) License: Public Domain  
Contributors: User:Malaz

**Image:MazeTutorial.06\_cube-touch.png** Source: [http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.06\\_cube-touch.png](http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.06_cube-touch.png) License: Public Domain  
Contributors: User:Malaz

**Image:MazeTutorial.09\_no-textures.png** Source: [http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.09\\_no-textures.png](http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.09_no-textures.png) License: Public Domain  
Contributors: User:Malaz

**Image:MazeTutorial.10\_edit-material.png** Source: [http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.10\\_edit-material.png](http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.10_edit-material.png) License: Public Domain  
Contributors: User:Malaz

**Image:MazeTutorial.11\_with-materials.png** Source: [http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.11\\_with-materials.png](http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.11_with-materials.png) License: Public Domain  
Contributors: User:Malaz

**Image:MazeTutorial.12\_with-mist.png** Source: [http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.12\\_with-mist.png](http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.12_with-mist.png) License: Public Domain  
Contributors: User:Malaz

**Image:MazeTutorial.13\_mist-buttons.png** Source: [http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.13\\_mist-buttons.png](http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.13_mist-buttons.png) License: Public Domain  
Contributors: User:Malaz

**Image:MazeTutorial.15\_scene-button.png** Source: [http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.15\\_scene-button.png](http://en.wikibooks.org/w/index.php?title=File:MazeTutorial.15_scene-button.png) License: Public Domain  
Contributors: User:Malaz

**Image:Skybox\_sky.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Skybox\\_sky.png](http://en.wikibooks.org/w/index.php?title=File:Skybox_sky.png) License: Public Domain  
Contributors: Greensweater

**Image:skybox\_ground.png** Source: [http://en.wikibooks.org/w/index.php?title=File:skybox\\_ground.png](http://en.wikibooks.org/w/index.php?title=File:skybox_ground.png) License: Public Domain  
Contributors: Greensweater

**Image:Skybox\_tut1.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Skybox\\_tut1.png](http://en.wikibooks.org/w/index.php?title=File:Skybox_tut1.png) License: unknown  
Contributors: Greensweater, Webaware, ZeroOne

**Image:Skybox\_tut2.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Skybox\\_tut2.png](http://en.wikibooks.org/w/index.php?title=File:Skybox_tut2.png) License: unknown  
Contributors: Greensweater, Webaware, ZeroOne

**Image:Skybox\_tut3.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Skybox\\_tut3.png](http://en.wikibooks.org/w/index.php?title=File:Skybox_tut3.png) License: unknown  
Contributors: Greensweater, Webaware, ZeroOne

**Image:Skybox\_tut4.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Skybox\\_tut4.png](http://en.wikibooks.org/w/index.php?title=File:Skybox_tut4.png) License: unknown  
Contributors: Greensweater, Webaware, ZeroOne

**Image:Skybox\_tut5.png** Source: [http://en.wikibooks.org/w/index.php?title=File:Skybox\\_tut5.png](http://en.wikibooks.org/w/index.php?title=File:Skybox_tut5.png) License: unknown  
Contributors: Greensweater, Webaware, ZeroOne

**Image:Skybox\_tut6.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Skybox\\_tut6.png](http://en.wikibooks.org/w/index.php?title=File:Skybox_tut6.png) *License:* unknown *Contributors:* Greensweater, Webaware, ZeroOne  
**Image:Skybox\_tut7.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Skybox\\_tut7.png](http://en.wikibooks.org/w/index.php?title=File:Skybox_tut7.png) *License:* unknown *Contributors:* Greensweater, Webaware, ZeroOne  
**Image:Skybox\_tut8.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Skybox\\_tut8.png](http://en.wikibooks.org/w/index.php?title=File:Skybox_tut8.png) *License:* unknown *Contributors:* Greensweater, MichaelFrey, ZeroOne  
**Image:Skybox\_tut9.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Skybox\\_tut9.png](http://en.wikibooks.org/w/index.php?title=File:Skybox_tut9.png) *License:* unknown *Contributors:* Greensweater, MichaelFrey, ZeroOne  
**Image:Skybox\_tut10.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Skybox\\_tut10.png](http://en.wikibooks.org/w/index.php?title=File:Skybox_tut10.png) *License:* unknown *Contributors:* Greensweater, Webaware, ZeroOne  
**Image:Skybox\_tut11.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Skybox\\_tut11.png](http://en.wikibooks.org/w/index.php?title=File:Skybox_tut11.png) *License:* unknown *Contributors:* Greensweater, Webaware, ZeroOne  
**Image:Skybox\_tut12.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Skybox\\_tut12.png](http://en.wikibooks.org/w/index.php?title=File:Skybox_tut12.png) *License:* unknown *Contributors:* Greensweater, Webaware, ZeroOne  
**Image:Skybox\_tut13.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Skybox\\_tut13.png](http://en.wikibooks.org/w/index.php?title=File:Skybox_tut13.png) *License:* Public Domain *Contributors:* Greensweater, ZeroOne  
**Image:Blender\_env\_order.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_env\\_order.png](http://en.wikibooks.org/w/index.php?title=File:Blender_env_order.png) *License:* Public Domain *Contributors:* Glome  
**Image:BlenderScriptsWindow.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderScriptsWindow.png> *License:* unknown *Contributors:* Allefant, Whiteknight, ZeroOne  
**File:Ball3 1\_2.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Ball3\\_1\\_2.jpg](http://en.wikibooks.org/w/index.php?title=File:Ball3_1_2.jpg) *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* Wayne Harrison  
**File:Blender 3D Scene.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_3D\\_Scene.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_3D_Scene.jpg) *License:* Public Domain *Contributors:* Adrignola, ErikBongers  
**Image:Blender\_HDR\_Mirror.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_HDR\\_Mirror.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_HDR_Mirror.jpg) *License:* Public Domain *Contributors:* ErikBongers  
**Image:Blender\_HDR\_AngMap.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_HDR\\_AngMap.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_HDR_AngMap.jpg) *License:* Public Domain *Contributors:* ErikBongers  
**Image:Blender\_HDR\_Result.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_HDR\\_Result.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_HDR_Result.jpg) *License:* Public Domain *Contributors:* ErikBongers  
**Image:Blender\_HDR\_YafRay\_Result.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_HDR\\_YafRay\\_Result.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_HDR_YafRay_Result.jpg) *License:* Public Domain *Contributors:* ErikBongers  
**Image:Lightprobe\_hardware\_diagram.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Lightprobe\\_hardware\\_diagram.png](http://en.wikibooks.org/w/index.php?title=File:Lightprobe_hardware_diagram.png) *License:* Creative Commons Attribution-Sharealike 2.5  
*Contributors:* Bmud  
**File:Blender heightmap screenshot.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_heightmap\\_screenshot.png](http://en.wikibooks.org/w/index.php?title=File:Blender_heightmap_screenshot.png) *License:* GNU General Public License *Contributors:* Adrignola, Aidan.Sullivan, Dal308, Swift  
**Image:Blender3d-heightmap-1.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3d-heightmap-1.jpg> *License:* Public Domain *Contributors:* Original uploader was Ascen at en.wikibooks  
**Image:Blender3d-heightmap-2.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3d-heightmap-2.png> *License:* Public Domain *Contributors:* Original uploader was Ascen at en.wikibooks  
**File:Blender-landscape\_with\_height\_maps.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender-landscape\\_with\\_height\\_maps.png](http://en.wikibooks.org/w/index.php?title=File:Blender-landscape_with_height_maps.png) *License:* GNU General Public License  
*Contributors:* Original uploader was Aidan.Sullivan at en.wikibooks  
**File:Blender-landscape\_with\_height\_maps(1).png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender-landscape\\_with\\_height\\_maps\(1\).png](http://en.wikibooks.org/w/index.php?title=File:Blender-landscape_with_height_maps(1).png) *License:* GNU General Public License  
*Contributors:* Original uploader was Aidan.Sullivan at en.wikibooks  
**Image:Scaledgrid.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Scaledgrid.jpg> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* SergeantOreo  
**Image:Cloudsettings.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Cloudsettings.jpg> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* SergeantOreo  
**Image:Shadersettings.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Shadersettings.jpg> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* SergeantOreo  
**Image:Noisevertices.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Noisevertices.jpg> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* SergeantOreo  
**Image:Blender3DNobobToPro-Grass.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Grass.jpg> *License:* Public Domain *Contributors:* azdesignsstudio  
**Image:Blender3DNobobToPro-Creating\_The\_Canvas.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Creating\\_The\\_Canvas.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Creating_The_Canvas.jpg) *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* User:Moohasha  
**Image:Blender3DNobobToPro-Molding\_the\_Mountains\_01.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Molding\\_the\\_Mountains\\_01.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Molding_the_Mountains_01.jpg) *License:* unknown  
*Contributors:* Moohasha, Reece  
**Image:Blender3DNobobToPro-Molding\_the\_Mountains\_02.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Molding\\_the\\_Mountains\\_02.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Molding_the_Mountains_02.jpg) *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* User:Moohasha  
**Image:Blender3DNobobToPro-Molding\_the\_Mountains\_03.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Molding\\_the\\_Mountains\\_03.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Molding_the_Mountains_03.jpg) *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* User:Moohasha  
**Image:Blender3DNobobToPro-Molding\_the\_Mountains\_04.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Molding\\_the\\_Mountains\\_04.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Molding_the_Mountains_04.jpg) *License:* unknown  
*Contributors:* Moohasha, Reece  
**Image:Blender3DNobobToPro-Molding\_the\_Mountains\_05.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Molding\\_the\\_Mountains\\_05.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Molding_the_Mountains_05.jpg) *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* User:Moohasha  
**Image:Blender3DNobobToPro-Texturing\_the\_Terrain\_02.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Texturing\\_the\\_Terrain\\_02.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Texturing_the_Terrain_02.jpg) *License:* unknown  
*Contributors:* Moohasha, Reece  
**Image:Blender3DNobobToPro-Texturing\_the\_Terrain\_03.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Texturing\\_the\\_Terrain\\_03.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Texturing_the_Terrain_03.jpg) *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* User:Moohasha  
**Image:Blender\_hills.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_hills.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_hills.jpg) *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* Bladerman  
**Image:Blender3DNobobToPro-Creating\_the\_Stencil\_02.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Creating\\_the\\_Stencil\\_02.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Creating_the_Stencil_02.jpg) *License:* unknown  
*Contributors:* Moohasha  
**Image:Blender3DNobobToPro-Creating\_the\_Stencil\_03.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Creating\\_the\\_Stencil\\_03.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Creating_the_Stencil_03.jpg) *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* User:Moohasha  
**Image:Blender3DNobobToPro-Creating\_the\_Stencil\_04.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Creating\\_the\\_Stencil\\_04.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Creating_the_Stencil_04.jpg) *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* User:Moohasha  
**Image:Blender3DNobobToPro-Applying\_the\_Stencil\_01.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Applying\\_the\\_Stencil\\_01.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Applying_the_Stencil_01.jpg) *License:* unknown  
*Contributors:* Moohasha  
**Image:Blender3DNobobToPro-Applying\_the\_Stencil\_02.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Applying\\_the\\_Stencil\\_02.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Applying_the_Stencil_02.jpg) *License:* Public Domain  
*Contributors:* User:Moohasha  
**Image:Blender3DNobobToPro-Multiple\_Stencils\_01.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Multiple\\_Stencils\\_01.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Multiple_Stencils_01.jpg) *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* User:Moohasha  
**Image:Blender3DNobobToPro-Multiple\_Stencils\_02.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Multiple\\_Stencils\\_02.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Multiple_Stencils_02.jpg) *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* User:Moohasha  
**Image:Blender3DNobobToPro-Multiple\_Stencils\_03.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Multiple\\_Stencils\\_03.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Multiple_Stencils_03.jpg) *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* User:Moohasha  
**Image:Blender3DNobobToPro-Multiple\_Stencils\_04.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Multiple\\_Stencils\\_04.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Multiple_Stencils_04.jpg) *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* User:Moohasha  
**Image:Blender3DNobobToPro-ExampleHeightmap.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-ExampleHeightmap.jpg> *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* User:Moohasha  
**Image:Blender3DNobobToPro-Exporting\_as\_a\_Heightmap01.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Exporting\\_as\\_a\\_Heightmap01.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Exporting_as_a_Heightmap01.jpg) *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* User:Moohasha  
**Image:Blender3DNobobToPro-Exporting\_as\_a\_Heightmap02.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Exporting\\_as\\_a\\_Heightmap02.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Exporting_as_a_Heightmap02.jpg) *License:* GNU Free Documentation License  
*Contributors:* User:Moohasha  
**Image:Blender3DNobobToPro-Exporting\_as\_a\_Heightmap03.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Exporting\\_as\\_a\\_Heightmap03.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3DNobobToPro-Exporting_as_a_Heightmap03.jpg) *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* User:Moohasha

**Image:Blender3DNobbToPro-Exporting\_as\_a\_Heightmap\_04.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender3DNobbToPro-Exporting\_as\_a\_Heightmap\_04.jpg  
*License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* User:Moohasha

**Image:Blender3DNobbToPro-Exporting\_as\_a\_Heightmap\_05.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender3DNobbToPro-Exporting\_as\_a\_Heightmap\_05.jpg  
*License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* User:Moohasha

**Image:Blender3DNobbToPro-Exporting\_as\_a\_Heightmap\_06.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender3DNobbToPro-Exporting\_as\_a\_Heightmap\_06.jpg  
*License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* User:Moohasha

**Image:Watertut01.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Watertut01.jpg *License:* GNU Free Documentation License *Contributors:* Charybdis

**Image:Watertut02.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Watertut02.jpg *License:* GNU Free Documentation License *Contributors:* Charybdis

**Image:Watertut03.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Watertut03.jpg *License:* GNU Free Documentation License *Contributors:* Charybdis

**Image:Watertut07.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Watertut07.jpg *License:* GNU Free Documentation License *Contributors:* Charybdis

**Image:Watertut06.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Watertut06.jpg *License:* GNU Free Documentation License *Contributors:* Charybdis

**Image:Watertut08.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Watertut08.jpg *License:* GNU Free Documentation License *Contributors:* Charybdis

**Image:Watertut09.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Watertut09.jpg *License:* GNU Free Documentation License *Contributors:* Charybdis

**Image:Watertut10.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Watertut10.jpg *License:* GNU Free Documentation License *Contributors:* Charybdis

**Image:0100.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:0100.jpg *License:* GNU Free Documentation License *Contributors:* Charybdis

**File:Blender3D BW Grid 256.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender3D\_BW\_Grid\_256.png *License:* Public Domain *Contributors:* User:SoylentGreen

**File:Blender3D BumpMapSettings-2.49.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender3D\_BumpMapSettings-2.49.png *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D BW-Testgrid-Applied.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender3D\_BW-Testgrid-Applied.jpg *License:* Public Domain *Contributors:* User:SoylentGreen

**File:Blender3D CreatingABumpMapCamera-2.49.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender3D\_CreatingABumpMapCamera-2.49.png *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D CreatingABumpMapCompositeNodes-2.49.png** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender3D\_CreatingABumpMapCompositeNodes-2.49.png *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D AnimatedBumpMapApplied0022-2.49.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender3D\_AnimatedBumpMapApplied0022-2.49.jpg *License:* Public Domain *Contributors:* User:SoylentGreen

**Image:example\_normal\_texture\_1.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Example\_normal\_texture\_1.jpg *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* Jeremy B.

**Image:example\_normal\_texture\_2.JPG** *Source:* http://en.wikibooks.org/w/index.php?title=File:Example\_normal\_texture\_2.JPG *License:* GNU Free Documentation License *Contributors:* Jeremy B.

**Image:example\_normal\_color\_map.JPG** *Source:* http://en.wikibooks.org/w/index.php?title=File:Example\_normal\_color\_map.JPG *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* Jeremy B.

**Image:Clear\_Blender.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Clear\_Blender.jpg *License:* Public Domain *Contributors:* Jeremy B.

**Image:Blender\_w\_cube.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender\_w\_cube.jpg *License:* Public Domain *Contributors:* Jeremy B.

**Image:blender\_w\_cube\_musgrave.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender\_w\_cube\_musgrave.jpg *License:* Public Domain *Contributors:* Jeremy B.

**Image:Texture\_material\_tabs.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Texture\_material\_tabs.jpg *License:* Public Domain *Contributors:* Jeremy B.

**Image:Blender\_render\_texture.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender\_render\_texture.jpg *License:* Public Domain *Contributors:* Jeremy B.

**Image:Extra\_examples.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Extra\_examples.jpg *License:* Public Domain *Contributors:* Jeremy B.

**Image:Blender\_bowl.JPG** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender\_bowl.JPG *License:* GNU General Public License *Contributors:* Jeremy B.

**Image:3\_norms\_material.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:3\_norms\_material.jpg *License:* unknown *Contributors:* Jeremy B.. Original uploader was Bullercruz1 at en.wikibooks

**Image:Ripple\_disc\_norm.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ripple\_disc\_norm.jpg *License:* Public Domain *Contributors:* Jeremy B.

**Image:Blender\_plus\_norm.JPG** *Source:* http://en.wikibooks.org/w/index.php?title=File:Blender\_plus\_norm.JPG *License:* GNU General Public License *Contributors:* Jeremy B.

**Image:OOPS bouncing.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:OOPS\_bouncing.jpg *License:* GNU General Public License *Contributors:* Gabio, OsamaK, ZeroOne

**Image:Add\_Armature\_obj.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Add\_Armature\_obj.jpg *License:* unknown *Contributors:* Original uploader was Gabio at en.wikibooks

**Image:Armature\_3D\_Header.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Armature\_3D\_Header.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Armature\_button\_obj.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Armature\_button\_obj.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Octahedron\_bone.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Octahedron\_bone.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**File:Stick\_bone.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Stick\_bone.jpg *License:* Public Domain *Contributors:* Adrignola, Gabio, ZeroOne

**Image:Bbones.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Bbones.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Env\_bone.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Env\_bone.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Axis\_drawing.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Axis\_drawing.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Name\_drawing.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Name\_drawing.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Bone\_ana.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Bone\_ana.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image>Edit\_panel\_bones.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File>Edit\_panel\_bones.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_Parent\_bones.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_Parent\_bones.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_bbones.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_bbones.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_envelope.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_envelope.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_Weight.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_Weight.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_hinge.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_hinge.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_bonesname.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_bonesname.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_stick\_guy.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_stick\_guy.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Pose\_panel.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Pose\_panel.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_Ghost.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_Ghost.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_bboneinout.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_bboneinout.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_path.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_path.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_Arm\_parent.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_Arm\_parent.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_arm\_parent\_menu.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_arm\_parent\_menu.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_little\_zone.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_little\_zone.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_ok\_zone.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_ok\_zone.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_Arm\_parent.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_Arm\_parent.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:edit\_panel\_bones.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File>Edit\_panel\_bones.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Ie\_Mult.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Ie\_Mult.jpg *License:* Public Domain *Contributors:* Gabio, ZeroOne

**Image:Vgroups\_but.jpg** *Source:* http://en.wikibooks.org/w/index.php?title=File:Vgroups\_but.jpg *License:* unknown *Contributors:* Gabio, MichaelFrey, ZeroOne

**Image:Weight\_spec.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Weight\\_spec.jpg](http://en.wikibooks.org/w/index.php?title=File:Weight_spec.jpg) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:Paint\_panel.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Paint\\_panel.jpg](http://en.wikibooks.org/w/index.php?title=File:Paint_panel.jpg) *License:* unknown *Contributors:* Gabio, Webaware, ZeroOne  
**Image:le\_paint\_tube.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:le\\_paint\\_tube.jpg](http://en.wikibooks.org/w/index.php?title=File:le_paint_tube.jpg) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:two\_objects.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Two\\_objects.jpg](http://en.wikibooks.org/w/index.php?title=File:Two_objects.jpg) *License:* GNU Free Documentation License *Contributors:* Stalepanda, ZeroOne  
**Image:Vertex\_groups.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Vertex\\_groups.jpg](http://en.wikibooks.org/w/index.php?title=File:Vertex_groups.jpg) *License:* GNU Free Documentation License *Contributors:* Stalepanda, ZeroOne  
**Image:initial\_weights.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Initial\\_weights.jpg](http://en.wikibooks.org/w/index.php?title=File:Initial_weights.jpg) *License:* GNU Free Documentation License *Contributors:* Stalepanda, ZeroOne  
**Image:objects\_bent\_initial.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Objects\\_bent\\_initial.jpg](http://en.wikibooks.org/w/index.php?title=File:Objects_bent_initial.jpg) *License:* GNU Free Documentation License *Contributors:* Stalepanda, ZeroOne  
**Image:Generi\_RVK.gif** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Generi\\_RVK.gif](http://en.wikibooks.org/w/index.php?title=File:Generi_RVK.gif) *License:* Public Domain *Contributors:* Gabio  
**Image:First\_key\_1.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:First\\_key\\_1.jpg](http://en.wikibooks.org/w/index.php?title=File:First_key_1.jpg) *License:* Public Domain *Contributors:* Gabio  
**Image:First\_key\_2.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:First\\_key\\_2.jpg](http://en.wikibooks.org/w/index.php?title=File:First_key_2.jpg) *License:* Public Domain *Contributors:* Gabio  
**Image:Shape\_panel.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Shape\\_panel.jpg](http://en.wikibooks.org/w/index.php?title=File:Shape_panel.jpg) *License:* unknown *Contributors:* Gabio, Webaware  
**Image:SyncSliderKeys.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:SyncSliderKeys.png> *License:* GNU General Public License *Contributors:* AndyD  
**Image:SyncSample.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:SyncSample.jpg> *License:* unknown *Contributors:* AndyD  
**Image:SyncLoops.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:SyncLoops.jpg> *License:* unknown *Contributors:* AndyD  
**Image:SyncJaw.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:SyncJaw.jpg> *License:* unknown *Contributors:* AndyD  
**Image:SyncSeqEditor.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:SyncSeqEditor.png> *License:* GNU General Public License *Contributors:* AndyD  
**Image:SyncSoundblock.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:SyncSoundblock.png> *License:* GNU General Public License *Contributors:* AndyD  
**Image:SyncSliders.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:SyncSliders.png> *License:* GNU General Public License *Contributors:* AndyD  
**Image:SyncSubSurf.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:SyncSubSurf.jpg> *License:* GNU General Public License *Contributors:* AndyD  
**Image:SyncTimeline.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:SyncTimeline.png> *License:* GNU General Public License *Contributors:* AndyD  
**Image:le\_constraint.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:le\\_constraint.jpg](http://en.wikibooks.org/w/index.php?title=File:le_constraint.jpg) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:C\_loc.gif** *Source:* [http://en.wikibooks.org/w/index.php?title=File:C\\_loc.gif](http://en.wikibooks.org/w/index.php?title=File:C_loc.gif) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:C\_loc\_panel.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:C\\_loc\\_panel.jpg](http://en.wikibooks.org/w/index.php?title=File:C_loc_panel.jpg) *License:* GNU General Public License *Contributors:* Darklama, Gabio, ZeroOne  
**Image:C\_loc\_track.gif** *Source:* [http://en.wikibooks.org/w/index.php?title=File:C\\_loc\\_track.gif](http://en.wikibooks.org/w/index.php?title=File:C_loc_track.gif) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:C\_rot.gif** *Source:* [http://en.wikibooks.org/w/index.php?title=File:C\\_rot.gif](http://en.wikibooks.org/w/index.php?title=File:C_rot.gif) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:C\_rot\_panel.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:C\\_rot\\_panel.jpg](http://en.wikibooks.org/w/index.php?title=File:C_rot_panel.jpg) *License:* unknown *Contributors:* Gabio, Webaware, ZeroOne  
**Image:C\_track.gif** *Source:* [http://en.wikibooks.org/w/index.php?title=File:C\\_track.gif](http://en.wikibooks.org/w/index.php?title=File:C_track.gif) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:C\_track\_panel.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:C\\_track\\_panel.jpg](http://en.wikibooks.org/w/index.php?title=File:C_track_panel.jpg) *License:* unknown *Contributors:* Gabio, Webaware, ZeroOne  
**Image:le\_track\_cam.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:le\\_track\\_cam.jpg](http://en.wikibooks.org/w/index.php?title=File:le_track_cam.jpg) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:le\_track\_eye.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:le\\_track\\_eye.jpg](http://en.wikibooks.org/w/index.php?title=File:le_track_eye.jpg) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:Stretch-to constraint.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Stretch-to\\_constraint.jpg](http://en.wikibooks.org/w/index.php?title=File:Stretch-to_constraint.jpg) *License:* Creative Commons Attribution-Sharealike 2.5 *Contributors:* SergeantOreo  
**Image:C\_ik.gif** *Source:* [http://en.wikibooks.org/w/index.php?title=File:C\\_ik.gif](http://en.wikibooks.org/w/index.php?title=File:C_ik.gif) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:C\_IK\_panel.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:C\\_IK\\_panel.jpg](http://en.wikibooks.org/w/index.php?title=File:C_IK_panel.jpg) *License:* unknown *Contributors:* Gabio, Webaware, ZeroOne  
**Image:le\_IK\_chain\_rot.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:le\\_IK\\_chain\\_rot.jpg](http://en.wikibooks.org/w/index.php?title=File:le_IK_chain_rot.jpg) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:le\_IK\_chain\_Tip.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:le\\_IK\\_chain\\_Tip.jpg](http://en.wikibooks.org/w/index.php?title=File:le_IK_chain_Tip.jpg) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:le\_IK\_chain\_Len.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:le\\_IK\\_chain\\_Len.jpg](http://en.wikibooks.org/w/index.php?title=File:le_IK_chain_Len.jpg) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:C\_ik\_tree.gif** *Source:* [http://en.wikibooks.org/w/index.php?title=File:C\\_ik\\_tree.gif](http://en.wikibooks.org/w/index.php?title=File:C_ik_tree.gif) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:le\_IK\_1.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:le\\_IK\\_1.jpg](http://en.wikibooks.org/w/index.php?title=File:le_IK_1.jpg) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:le\_IK\_chain\_limit.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:le\\_IK\\_chain\\_limit.jpg](http://en.wikibooks.org/w/index.php?title=File:le_IK_chain_limit.jpg) *License:* Public Domain *Contributors:* Gabio, ZeroOne  
**Image:le\_IK\_limit\_buttons.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:le\\_IK\\_limit\\_buttons.jpg](http://en.wikibooks.org/w/index.php?title=File:le_IK_limit_buttons.jpg) *License:* unknown *Contributors:* Gabio, Webaware, ZeroOne  
**image:action-1.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Action-1.jpg> *License:* GNU Free Documentation License *Contributors:* Halley  
**image:nla-2.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Nla-2.jpg> *License:* GNU Free Documentation License *Contributors:* Halley  
**image:action-2.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Action-2.jpg> *License:* GNU Free Documentation License *Contributors:* Halley  
**image:nla-3.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Nla-3.jpg> *License:* GNU Free Documentation License *Contributors:* Halley  
**image:nla-4.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Nla-4.jpg> *License:* GNU Free Documentation License *Contributors:* Halley  
**Image:toggle-nla-pose.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Toggle-nla-pose.jpg> *License:* GNU Free Documentation License *Contributors:* Halley  
**Image:toggle-nla-strips.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Toggle-nla-strips.jpg> *License:* GNU Free Documentation License *Contributors:* Halley  
**Image:NLA1.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:NLA1.jpg> *License:* GNU Free Documentation License *Contributors:* Zeviion  
**Image:NLA2.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:NLA2.jpg> *License:* GNU Free Documentation License *Contributors:* Zeviion  
**Image:NLA3.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:NLA3.jpg> *License:* unknown *Contributors:* Zeviion  
**Image:NLA4.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:NLA4.jpg> *License:* GNU Free Documentation License *Contributors:* Zeviion  
**Image:NLA5.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:NLA5.jpg> *License:* GNU Free Documentation License *Contributors:* Zeviion  
**Image:NLA6.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:NLA6.jpg> *License:* GNU Free Documentation License *Contributors:* Zeviion  
**Image:Getting\_Started\_With\_Bob.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Getting\\_Started\\_With\\_Bob.jpg](http://en.wikibooks.org/w/index.php?title=File:Getting_Started_With_Bob.jpg) *License:* GNU General Public License *Contributors:* Derbeth, Wavez  
**Image:Rig\_Step1.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Rig\\_Step1.jpg](http://en.wikibooks.org/w/index.php?title=File:Rig_Step1.jpg) *License:* GNU General Public License *Contributors:* Wavez  
**Image:Blender3D\_KolbenZylinderExample.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_KolbenZylinderExample.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_KolbenZylinderExample.png) *License:* GNU Free Documentation License *Contributors:* User:Soylentgreen  
**Image:Blender3D\_MakeVertexParentSchwungrad.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_MakeVertexParentSchwungrad.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_MakeVertexParentSchwungrad.png) *License:* GNU Free Documentation License *Contributors:* User:Soylentgreen  
**Image:Blender3D\_KolbenZylinderEmptyPositions.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_KolbenZylinderEmptyPositions.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_KolbenZylinderEmptyPositions.png) *License:* GNU Free Documentation License *Contributors:* User:Soylentgreen  
**Image:Blender3D\_KolbenZylinderArmaturePositions.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_KolbenZylinderArmaturePositions.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_KolbenZylinderArmaturePositions.png) *License:* GNU Free Documentation License *Contributors:* User:Soylentgreen  
**Image:Blender3D\_KolbenZylinderBHelper1Constraint.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_KolbenZylinderBHelper1Constraint.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_KolbenZylinderBHelper1Constraint.png) *License:* GNU Free Documentation License *Contributors:* User:Soylentgreen  
**Image:Blender3D\_KolbenZylinderESteuerungConstraint.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_KolbenZylinderESteuerungConstraint.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_KolbenZylinderESteuerungConstraint.png) *License:* GNU Free Documentation License *Contributors:* User:Soylentgreen  
**Image:Blender3D\_kolben\_zylinder\_relations.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_kolben\\_zylinder\\_relations.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_kolben_zylinder_relations.png) *License:* GNU Free Documentation License *Contributors:* User:Soylentgreen  
**Image:Blender3D\_KolbenZylinderAnimation.gif** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_KolbenZylinderAnimation.gif](http://en.wikibooks.org/w/index.php?title=File:Blender3D_KolbenZylinderAnimation.gif) *License:* GNU Free Documentation License *Contributors:* User:Soylentgreen

**File:CuttingThroughSteel.log** *Source:* <http://en.wikibooks.org/w/index.php?title=File:CuttingThroughSteel.log> *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**File:Blender3D CuttingThroughSteel-ObjectCamera-2.48a.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_CuttingThroughSteel-ObjectCamera-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_CuttingThroughSteel-ObjectCamera-2.48a.png) *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D CuttingThroughSteel-AlphaParticleSystem-2.48a.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_CuttingThroughSteel-AlphaParticleSystem-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_CuttingThroughSteel-AlphaParticleSystem-2.48a.png) *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D CuttingThroughSteel-AlphaMap-0026.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_CuttingThroughSteel-AlphaMap-0026.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_CuttingThroughSteel-AlphaMap-0026.png) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**File:Blender3D CuttingThroughSteel-AddScene-2.48a.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_CuttingThroughSteel-AddScene-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_CuttingThroughSteel-AddScene-2.48a.png) *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D CuttingThroughSteel-SparksParticles-2.48a.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_CuttingThroughSteel-SparksParticles-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_CuttingThroughSteel-SparksParticles-2.48a.png) *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D DrawType.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_DrawType.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_DrawType.png) *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D CuttingThroughSteel-AlphaMapMaterial-2.48a.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_CuttingThroughSteel-AlphaMapMaterial-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_CuttingThroughSteel-AlphaMapMaterial-2.48a.png) *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D CuttingThroughSteel-AlphaMapTexture-2.48a.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_CuttingThroughSteel-AlphaMapTexture-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_CuttingThroughSteel-AlphaMapTexture-2.48a.png) *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D CuttingThroughSteel-PlaneMaterial-Frame34.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_CuttingThroughSteel-PlaneMaterial-Frame34.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_CuttingThroughSteel-PlaneMaterial-Frame34.jpg) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**File:Blender3D CuttingThroughSteel-LampEnergieIpo-2.48a.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_CuttingThroughSteel-LampEnergieIpo-2.48a.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_CuttingThroughSteel-LampEnergieIpo-2.48a.png) *License:* GNU General Public License *Contributors:* User:SoylentGreen

**File:Blender3D CuttingThroughSteel-Sparks-Frame34.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_CuttingThroughSteel-Sparks-Frame34.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_CuttingThroughSteel-Sparks-Frame34.jpg) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**File:Blender3D CuttingThroughSteel-SparksSmoke-Frame34.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_CuttingThroughSteel-SparksSmoke-Frame34.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_CuttingThroughSteel-SparksSmoke-Frame34.jpg) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**File:Blender3D CuttingThroughSteel Smoke.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_CuttingThroughSteel\\_Smoke.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_CuttingThroughSteel_Smoke.jpg) *License:* Public Domain *Contributors:* User:SoylentGreen

**File:Blender3D CuttingThroughSteel GlowMap-34.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_CuttingThroughSteel\\_GlowMap-34.png](http://en.wikibooks.org/w/index.php?title=File:Blender3D_CuttingThroughSteel_GlowMap-34.png) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**File:Blender3D CuttingThroughSteel-SparksSmokeGlow-Frame34.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender3D\\_CuttingThroughSteel-SparksSmokeGlow-Frame34.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender3D_CuttingThroughSteel-SparksSmokeGlow-Frame34.jpg) *License:* GNU Free Documentation License *Contributors:* User:SoylentGreen

**File:Rust\_and\_dirt.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Rust\\_and\\_dirt.jpg](http://en.wikibooks.org/w/index.php?title=File:Rust_and_dirt.jpg) *License:* GNU Free Documentation License *Contributors:* User:Roger McLassus

**Image:main\_menu\_starting.JPG** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Main\\_menu\\_starting.JPG](http://en.wikibooks.org/w/index.php?title=File:Main_menu_starting.JPG) *License:* GNU General Public License *Contributors:* Jeremy B.

**Image:In\_Arms\_Startung\_Menu.JPG** *Source:* [http://en.wikibooks.org/w/index.php?title=File:In\\_Arms\\_Startung\\_Menu.JPG](http://en.wikibooks.org/w/index.php?title=File:In_Arms_Startung_Menu.JPG) *License:* GNU General Public License *Contributors:* User:bullecrusu1

**Image:showpointer.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File>Showpointer.jpg> *License:* GNU General Public License *Contributors:* Jeremy B.

**Image:Blender\_Button\_arraignment.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_Button\\_arraignment.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_Button_arraignment.jpg) *License:* GNU General Public License *Contributors:* Jeremy B.

**Image:Blender\_Multi\_Button.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_Multi\\_Button.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_Multi_Button.jpg) *License:* GNU General Public License *Contributors:* Jeremy B.

**Image:Blender Linking Menus.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_Linking\\_Menus.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_Linking_Menus.jpg) *License:* GNU General Public License *Contributors:* Jeremy B.

**File:Clear\_Blender.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Clear\\_Blender.jpg](http://en.wikibooks.org/w/index.php?title=File:Clear_Blender.jpg) *License:* Public Domain *Contributors:* Jeremy B.

**File:Blender basic menu.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_basic\\_menu.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_basic_menu.jpg) *License:* GNU General Public License *Contributors:* Jeremy B.

**File:Blender Graphic Menu.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Blender\\_Graphic\\_Menu.jpg](http://en.wikibooks.org/w/index.php?title=File:Blender_Graphic_Menu.jpg) *License:* GNU Free Documentation License *Contributors:* Jeremy B.

**File:Menu icon selector1.jpg** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Menu\\_icon\\_selector1.jpg](http://en.wikibooks.org/w/index.php?title=File:Menu_icon_selector1.jpg) *License:* GNU General Public License *Contributors:* Jeremy B.

**Image:Clayman.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Clayman.jpg> *License:* Creative Commons Attribution-Sharealike 2.0 *Contributors:* Bootaleg, Lcarsdata, Webaware, 1 anonymous edits

**Image:LSTV.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:LSTV.jpg> *License:* Creative Commons Attribution-Sharealike 2.0 *Contributors:* Bootaleg, Webaware, 1 anonymous edits

**Image:PD-icon.svg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:PD-icon.svg> *License:* Public Domain *Contributors:* User:Duesentrieb, User:Rfl

**File:BlenderAddSubsurf.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderAddSubsurf.jpg> *License:* unknown *Contributors:* Blender Foundation

**File:BlenderSubsurfModifier.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderSubsurfModifier.jpg> *License:* unknown *Contributors:* Blender Foundation

**File:BlenderCube.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderCube.jpg> *License:* Public Domain *Contributors:* User:Admijr

**File:BlenderCubeSubsurfed.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderCubeSubsurfed.jpg> *License:* GNU General Public License *Contributors:* Blender Foundation

**Image:M14kc.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:M14kc.jpg> *License:* Public Domain *Contributors:* Big Bill57, Chelsean528

**File:StepOne.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:StepOne.png> *License:* Public Domain *Contributors:* Michal Sapalski (username: sapalskimichal)

**File:StepTwo.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:StepTwo.png> *License:* Public Domain *Contributors:* Michal Sapalski (username: sapalskimichal)

**File:StepThree.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:StepThree.png> *License:* Public Domain *Contributors:* Michal Sapalski (username: sapalskimichal)

**File:StepFour.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:StepFour.png> *License:* Public Domain *Contributors:* Michal Sapalski (username: sapalskimichal)

**File:StepFive.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:StepFive.png> *License:* Public Domain *Contributors:* Michal Sapalski (username: sapalskimichal)

**File:JewelFinished.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:JewelFinished.png> *License:* Public Domain *Contributors:* Michal Sapalski (username: sapalskimichal)

**File:Jewel\_Materials.png** *Source:* [http://en.wikibooks.org/w/index.php?title=File:Jewel\\_Materials.png](http://en.wikibooks.org/w/index.php?title=File:Jewel_Materials.png) *License:* Public Domain *Contributors:* Michal Sapalski (username: sapalskimichal)

**File:JewelExample.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:JewelExample.png> *License:* Public Domain *Contributors:* Michal Sapalski (username: sapalskimichal)

**Image:skateboard-example.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Skateboard-example.jpg> *License:* Public Domain *Contributors:* Halley

**Image:soda-can-example.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Soda-can-example.jpg> *License:* Public Domain *Contributors:* Halley

**Image:Tut.gif** *Source:* <http://en.wikibooks.org/w/index.php?title=File:Tut.gif> *License:* Public Domain *Contributors:* Aidan.Sullivan

**Image:3dbg-extrude&scale.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:3dbg-extrude&scale.jpg> *License:* GNU Free Documentation License *Contributors:* Quantum Anomaly

**Image:3dbg-planescale.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:3dbg-planescale.jpg> *License:* GNU Free Documentation License *Contributors:* Quantum Anomaly

**Image:3dbg-planeblack.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:3dbg-planeblack.jpg> *License:* GNU Free Documentation License *Contributors:* Quantum Anomaly

**Image:3dbg-persp-ortho.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:3dbg-persp-ortho.jpg> *License:* GNU Free Documentation License *Contributors:* Quantum Anomaly

**Image:3dbg-scale&dup.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:3dbg-scale&dup.jpg> *License:* GNU Free Documentation License *Contributors:* Quantum Anomaly

**Image:3dbg-boundaries.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:3dbg-boundaries.jpg> *License:* GNU Free Documentation License *Contributors:* Quantum Anomaly

**Image:3dbg-deletefaces.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:3dbg-deletefaces.jpg> *License:* GNU Free Documentation License *Contributors:* Quantum Anomaly

**Image:3dbg-zmove.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:3dbg-zmove.jpg> *License:* GNU Free Documentation License *Contributors:* Quantum Anomaly

**Image:3dbg-populated.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:3dbg-populated.jpg> *License:* GNU Free Documentation License *Contributors:* Quantum Anomaly

**Image:3dbg-orthocamera.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:3dbg-orthocamera.jpg> *License:* GNU Free Documentation License *Contributors:* Quantum Anomaly

**Image:3dbg-tilepoints.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:3dbg-tilepoints.jpg> *License:* GNU Free Documentation License *Contributors:* Quantum Anomaly

**Image:3dbg-guides.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:3dbg-guides.jpg> *License:* unknown *Contributors:* Quantum Anomaly

**Image:3dbg-copypaste.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:3dbg-copypaste.jpg> *License:* unknown *Contributors:* Quantum Anomaly

**Image:3dbg-eraseseam.gif** *Source:* <http://en.wikibooks.org/w/index.php?title=File:3dbg-eraseseam.gif> *License:* GNU Free Documentation License *Contributors:* Quantum Anomaly

**Image:3dbg-finaltile.jpg** *Source:* <http://en.wikibooks.org/w/index.php?title=File:3dbg-finaltile.jpg> *License:* GNU Free Documentation License *Contributors:* Quantum Anomaly

**Image:BlenderHotkeysObjectMode.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderHotkeysObjectMode.png> *License:* Public Domain *Contributors:* Cardinal4

**Image:BlenderHotkeysEditMode.png** *Source:* <http://en.wikibooks.org/w/index.php?title=File:BlenderHotkeysEditMode.png> *License:* Public Domain *Contributors:* Cardinal4

---

# License

---

Creative Commons Attribution-Share Alike 3.0 Unported  
<http://creativecommons.org/licenses/by-sa/3.0/>