

1.1 Getting Started - About Blender

Introduction.....	1
Key Features.....	2
Blender's History.....	3
Video: From Blender 1.60 to 2.50.....	4
Version/Revision Milestones.....	5
The start!.....	5
Blender goes Open Source.....	5
Blender 2.5x - The Recode!.....	6
Blender 2.6x to 2.7x - Improvements & Stabalizing.....	6
About Free Software and the GPL.....	7
The Blender Community.....	8
Who uses Blender?.....	8
Independent Sites.....	8
Support.....	9
Development.....	9
IRC Channels.....	9
Other Useful Links.....	9

Introduction



Blender 2.5 with a Big Buck Bunny scene open

Welcome to Blender, the free and open source 3D animation suite.

Blender can be used to create 3D visualizations such as still images, video and real-time interactive video

games.

Blender is well suited to individuals and small studios who benefit from its unified pipeline and responsive development process.

It is cross-platform and runs on Linux, Mac-OSX and MS-Windows systems with a small memory and disk footprint. Its interface uses OpenGL to provide a consistent experience across all supported hardware and platforms.

Key Features

- Blender is a fully integrated 3D content creation suite, offering a broad range of essential tools, including *Modeling*, *Rendering*, *Animation*, *Video Editing*, *VFX*, *Compositing*, *Texturing*, *Rigging*, many types of *Simulations*, and *Game Creation*.
- Cross platform, with an OpenGL GUI that is uniform on all major platforms (and customizable with Python scripts).
- High quality 3D architecture enabling fast and efficient creation work-flow.
- Excellent community support from forums and IRC.
- Small executable size, optionally portable.

You can download the latest version of Blender here.



A rendered image being post-processed

Blender makes it possible to perform a wide range of tasks, and it may seem daunting when first trying to grasp the basics. However, with a bit of motivation and the right learning material, it is possible to familiarize yourself with Blender after a few hours of practice.

This manual is a good start, though it serves more as a reference. There are also many online video tutorials

from specialized websites, and several books and training DVDs available in the Blender Store and on the Blender Cloud.

Despite everything Blender can do, it remains a tool. Great artists do not create masterpieces by pressing buttons or manipulating brushes, but by learning and practicing subjects such as human anatomy, composition, lighting, animation principles, etc.

3D content creation software such as Blender have the added technical complexity and jargon associated with the underlying technologies. Terms like UV maps, materials, shaders, meshes, and subsurf are the mediums of the digital artist, and understanding them, even broadly, will help you to use Blender to its best.

So keep reading this manual, learn the great tool that Blender is, keep your mind open to other artistic and technological areas, and you too can become a great artist.

Blender's History

In 1988 Ton Roosendaal co-founded the Dutch animation studio NeoGeo. NeoGeo quickly became the largest 3D animation studio in the Netherlands and one of the leading animation houses in Europe. NeoGeo created award-winning productions (European Corporate Video Awards 1993 and 1995) for large corporate clients such as multi-national electronics company Philips. Within NeoGeo Ton was responsible for both art direction and internal software development. After careful deliberation Ton decided that the current in-house 3D toolset for NeoGeo was too old and cumbersome to maintain, and needed to be rewritten from scratch. In 1995 this rewrite began and was destined to become the 3D software creation we all know as Blender. As NeoGeo continued to refine and improve Blender it became apparent to Ton that Blender could be used as a tool for other artists outside of NeoGeo.

In 1998, Ton decided to found a new company called Not a Number (NaN) as a spin-off of NeoGeo to further market and develop Blender. At the core of NaN was a desire to create and distribute a compact, cross platform 3D application for free. At the time this was a revolutionary concept as most commercial 3D applications cost thousands of dollars. NaN hoped to bring professional level 3D modeling and animation tools within the reach of the general computing public. NaN's business model involved providing commercial products and services around Blender. In 1999 NaN attended its first SIGGRAPH conference in an effort to more widely promote Blender. Blender's first SIGGRAPH convention was a huge success and gathered a tremendous amount of interest from both the press and attendees. Blender was a hit and its huge potential confirmed!

Following the success of the SIGGRAPH conference in early 2000, NaN secured financing of €4.5M from venture capitalists. This large inflow of cash enabled NaN to rapidly expand its operations. Soon NaN boasted as many as fifty employees working around the world trying to improve and promote Blender. In the summer of 2000, Blender 2.0 was released. This version of Blender added the integration of a game engine to the 3D application. By the end of 2000, the number of users registered on the NaN website surpassed 250,000.

Unfortunately, NaN's ambitions and opportunities didn't match the company's capabilities and the market realities of the time. This over-extension resulted in restarting NaN with new investor funding and a smaller company in April 2001. Six months later NaN's first commercial software product, Blender Publisher was launched. This product was targeted at the emerging market of interactive web-based 3D media. Due to disappointing sales and the ongoing difficult economic climate, the new investors decided to shut down all NaN operations. The shutdown also included discontinuing the development of Blender. Although there were clearly

shortcomings in the then current version of Blender, such as a complex internal software architecture, unfinished features and a non-standard way of providing the GUI, the enthusiastic support from the user community and customers who had purchased Blender Publisher in the past meant that Ton couldn't justify leaving Blender to fade into insignificance. Since restarting a company with a sufficiently large team of developers wasn't feasible, Ton Roosendaal founded the non-profit organization Blender Foundation in March 2002.

The Blender Foundation's primary goal was to find a way to continue developing and promoting Blender as a community-based open source project. In July 2002, Ton managed to get the NaN investors to agree to a unique Blender Foundation plan to attempt to release Blender as open source. The "Free Blender" campaign sought to raise €100,000 so that the Foundation could buy the rights to the Blender source code and intellectual property rights from the NaN investors and subsequently release Blender to the open source community. With an enthusiastic group of volunteers, among them several ex-NaN employees, a fund raising campaign was launched to "Free Blender". To everyone's surprise and delight the campaign reached the €100,000 goal in only seven short weeks. On Sunday October 13, 2002, Blender was released to the world under the terms of the GNU GPL. Blender development continues to this day driven by a team of dedicated volunteers from around the world led by Blender's original creator, Ton Roosendaal.

Video: From Blender 1.60 to 2.50

<https://vimeo.com/8567074>

Version/Revision Milestones

The start!

- 1.00 - January 1994: Blender in development at animation studio NeoGeo.
- 1.23 - January 1998: SGI version published on the web, IrisGL.
- 1.30 - April 1998: Linux and FreeBSD version, port to OpenGL and X11.
- 1.3x - June 1998: NaN founded.
- 1.4x - September 1998: Sun and Linux Alpha version released.
- 1.50 - November 1998: First Manual published.
- 1.60 - April 1999: C-key (new features behind a lock, \$95), MS-Windows version released.
- 1.6x - June 1999: BeOS and PPC version released.
- 1.80 - June 2000: End of C-key, Blender full freeware again.
- 2.00 - August 2000: Interactive 3D and real-time engine.
- 2.10 - December 2000: New engine, physics, and Python.
- 2.20 - August 2001: Character animation system.
- 2.21 - October 2001: Blender Publisher launch.
- 2.2x - December 2001: Mac OSX version.

Blender goes Open Source

- **13 October 2002: Blender goes Open Source, 1st Blender Conference.**
- 2.25 - October 2002: Blender Publisher becomes freely available.

- Tuhopuu1 - Oct 2002: The experimental tree of Blender is created, a coder's playground.
- 2.26 - February 2003: The first true open source Blender release.
- 2.27 - May 2003: The second open source Blender release.
- 2.28x - July 2003: First of the 2.28x series.
- 2.30 - October 2003: Preview release of the 2.3x UI makeover presented at the 2nd Blender Conference.
- 2.31 - December 2003: Upgrade to stable 2.3x UI project.
- 2.32 - January 2004: Major overhaul of internal rendering capabilities.
- 2.33 - April 2004: Game Engine returns, ambient occlusion, new procedural textures.
- 2.34 - August 2004: Particle interactions, LSCM UV mapping, functional YafRay integration, weighted creases in subdivision surfaces, ramp shaders, full OSA, and many many more.
- 2.35 - November 2004: Another version full of improvements: object hooks, curve deforms and curve tapers, particle duplicators and much more.
- 2.36 - December 2004: A stabilization version, much work behind the scene, normal and displacement mapping improvements.
- 2.37 - June 2005: Transformation tools and widgets, softbodies, force fields, deflections, incremental subdivision surfaces, transparent shadows, and multi-threaded rendering.
- 2.40 - December 2005: Full rework of armature system, shape keys, fur with particles, fluids and rigid bodies.
- 2.41 - January 2006: Lots of fixes, and some game engine features.
- 2.42 - July 2006: The nodes release, array modifier, vector blur, new physics engine, rendering, lip sync, and many other features. This was the release following Project Orange.
- 2.43 - February 2007: Multi-resolution meshes, multi-layer UV textures, multi-layer images and multi-pass rendering and baking, sculpting, retopology, multiple additional matte, distort and filter nodes, modeling and animation improvements, better painting with multiple brushes, fluid particles, proxy objects, sequencer rewrite, and post-production UV texturing.
- 2.44 - May 2007: The big news, in addition to two new modifiers and re-awakening the 64-bit OS support, was the addition of subsurface scattering, which simulates light scattering beneath the surface of organic and soft objects.
- 2.45 - September 2007: Serious bug fixes, with some performance issues addressed.
- 2.46 - May 2008: The Peach release was the result of a huge effort of over 70 developers providing enhancements to provide hair and fur, a new particle system, enhanced image browsing, cloth, a seamless and non-intrusive physics cache, rendering improvements in reflections, AO, and render baking, a mesh deform modifier for muscles and such, better animation support via armature tools and drawing, skinning, constraints and a colorful Action Editor, and much more. It was the release following Project Peach.
- 2.47 - August 2008: Bugfix release.
- 2.48 - October 2008: The Apricot release, cool GLSL shaders, lights and GE improvements, snap, sky simulator, shrinkwrap modifier, and Python editing improvements. This was the release following Project Apricot.
- 2.49 - June 2009: Node-based textures, armature sketching (called Etch-a-Ton), boolean mesh operation improvements, JPEG2000 support, projection painting for direct transfer of images to models, and a significant Python script catalogue. GE enhancements included video textures, where you can play movies in-game, upgrades to the Bullet physics engine, dome (fish-eye) rendering, and more API GE calls made available.

Blender 2.5x - The Recode!

- 2.5x - From 2009 to August 2011: This series released four pre-version (from Alpha 0 in November 2009 to Beta in July 2010) and three stable versions (from 2.57 - April 2011 - to 2.59 - August 2011). It is one of the most important development projects, with a total refactor of the software with new functions, redesign of the internal window manager and event/tool/data handling system, and new Python API. The final version of this project was Blender 2.59 in August 2011.

Blender 2.6x to 2.7x - Improvements & Stabalizing

- 2.60 - October 2011: Internationalization of the UI, improvements in animation system and the GE, vertex weight groups modifiers, 3D audio and video, bug fixes, and the UI internationalization.
- 2.61 - December 2011: The Cycles renderer was added in trunk, the camera tracker was added, dynamic paint for modifying textures with mesh contact/approximation, the Ocean Sim modifier to simulate ocean and foam, new add-ons, bug fixes, and more extensions added for the Python API.
- 2.62 - February 2012: The Carve library was added to improve boolean operations, support for object tracking was added, the Remesh modifier was added, many improvements in the GE, matrices and vectors in the Python API were improved, new add-ons, and many bug fixes.
- 2.63 - April 2012: Bmesh was merged to trunk with full support for n-sided polygons, sculpt hiding, a panoramic camera for Cycles, mirror ball environment textures and float precision textures, render layer mask layers, ambient occlusion and viewport display of background images and render layers, new import and export add-ons were added, and 150 bug fixes.
- 2.64 - October 2012: Mask editor, improved motion tracker, OpenColorIO, Cycles improvements, sequencer improvements, better mesh tools (Inset and Bevel were improved), new keying nodes, sculpt masking, Collada improvements, new skin modifier, new compositing nodes backend, and many bugs were fixed.
- 2.65 - December 2012: Fire and smoke improvements, anisotropic shader for Cycles, modifier improvements, bevel tool now includes rounding, new add-ons, and over 200 bug fixes.
- 2.66 - February 2013: Dynamic topology, rigid body simulation, improvements in UI and usability (including retina display support), Cycles now supports hair, the bevel tool now supports individual vertex bevelling, new *Mesh Cache* modifier and the new *UV Warp* modifier, new SPH particle fluid solver. More than 250 bug fixes.
- 2.67 - May 2013: Freestyle was added, paint system improvements, subsurface scattering for Cycles, Ceres library in the motion tracker, new custom python nodes, new mesh modeling tools, better support for UTF8 text and improvements in text editors, new add-ons for 3D printing, over 260 bug fixes.
- 2.68 - July 2013: New and improved modeling tools, three new Cycles nodes, big improvements in the motion tracker, Python scripts and drivers are disabled by default when loading files for security reasons, and over 280 bug fixes.
- 2.69 - October 2013: Even more modeling tools, Cycles improved in many areas, plane tracking is added to the motion tracker, better support for FBX import/export, and over 270 bugs fixed.
- 2.70 - March 2014: Cycles gets basic volumetric support on the CPU, more improvements to the motion tracker, two new modeling modifiers, some UI consistency improvements, and more than 560 bug fixes.
- 2.71 - June 2014: Deformation motion blur and fire/smoke support is added to Cycles, UI popups are now draggable, performance optimizations for sculpting mode, new interpolation types for animation, many improvements to the GE, and over 400 bug fixes.

- 2.72 - October 2014: Cycles gets volume and SSS support on the GPU, pie menus are added and tooltips greatly improved, the intersection modeling tool is added, new sun beam node for the compositor, Freestyle now works with Cycles, texture painting workflow is improved, and more than 220 bug fixes.
- 2.73 - January 2015: Cycles gets improved volumetric support, major upgrade to grease pencil, MS-Windows gets Input Method Editors (IMEs) and general improvements to painting, freestyle, sequencer and add-ons.
- 2.74 - March 2015: Support for custom-normals, viewport compositing and improvements to hair dynamics.
- 2.75 - July 2015: Integrated stereo/multi-view pipeline, corrective smooth modifier and new dependency graph (*enable as a command line option*).
- 2.76 - November 2015: Pixar OpenSubdiv support, Viewport and File Browser performance boost, node auto-offset, and a text effect strip for the Sequencer.

About Free Software and the GPL



When one hears about “free software”, the first thing that comes to mind might be “no cost”. While this is typically true, the term “free software” as used by the Free Software Foundation (originators of the GNU Project and creators of the GNU General Public License) is intended to mean “free as in freedom” rather than the “no cost” sense (which is usually referred to as “free as in free beer” or *gratis*). Free software in this sense is software which you are free to use, copy, modify, redistribute, with no limit. Contrast this with the licensing of most commercial software packages, where you are allowed to load the software on a single computer, are allowed to make no copies, and never see the source code. Free software allows incredible freedom to the end user. Since the source code is universally available, there are also many more chances for bugs to be caught and fixed.

When a program is licensed under the GNU General Public License (the GPL):

- You have the right to use the program for any purpose.
- You have the right to modify the program, and have access to the source codes.
- You have the right to copy and distribute the program.
- You have the right to improve the program, and release your own versions.

In return for these rights, you have some responsibilities if you distribute a GPL'd program, responsibilities that are designed to protect your freedoms and the freedoms of others:

- You must provide a copy of the GPL with the program, so that recipients are aware of their rights under the license.
- You must include the source code or make the source code freely available.
- If you modify the code and distribute the modified version, you must license your modifications available under the GPL (or a compatible license).
- You may not restrict the licensing of the program beyond the terms of the GPL. (you may not turn a GPL'd program into a proprietary product.)

For more on the GPL, check the its page on the GNU Project web site.

Note

The GPL only applies to the Blender application and **not** the artwork you create with it; for more info see the Blender License.

The Blender Community

Being freely available from the start, even while closed source, helped considerably in Blender's adoption. A large, stable, and active community of users has gathered around Blender since 1998. The community showed its support for Blender in 2002 when they helped raise €100,000 in seven weeks to enable Blender to go Open Source under the GNU GPL.

Who uses Blender?

Blender has a wide variety of tools making it suitable for almost any sort of media production. People and studios around the world use it for hobby projects, commercials, feature films, games and other interactive applications like kiosks, games and scientific research.

Check out the User Stories page on the Blender website for more examples.

Independent Sites

There are several independent websites such as forums, blogs, news and tutorial sites dedicated to Blender.

One of the largest community forums is Blender Artists, where Blender users gather to show off their creations, get feedback, ask and offer help and in general discuss Blender.

Support

Blender's community is one of its greatest features, so apart from this user manual there are many different ways to get support from other users, such as IRC and Stack Exchange.

There are also more official sources of support, such as Certified Trainers and the Blender Cloud. If you think you have found an issue with Blender, you can easily report a bug.

More details about support can be found on the support page.

Development

Being open source, some of Blender's development is done by volunteers. Communication between developers is done mostly through three platforms:

- The developer.blender.org system
- Various mailing lists
- The #blendercoders IRC channel (see below)

If you are interested in helping develop Blender, see the [Get Involved](#) page.

IRC Channels

For real-time discussion, there are some Blender IRC channels on the Freenode network. You can join these with your favorite IRC client:

- #blender Community support channel
- #blenderchat For general discussion or offtopic chat
- #blendercoders For developers to discuss Blender development
- #blenderpython For support for developers using the Python API
- #gameblender For discussion on issues related to game creation with the GE
- #blenderwiki For discussion related to Blender's documentation

Note

If you do not have an IRC client, you can access IRC using webchat.

There also several more Blender-related channels not listed here (e.g. channels for speakers of a particular language). We recommend you search Freenode to see them all.

Other Useful Links

- [Blender FAQ \(Can I use Blender commercially? What is GPL/GNU? ...\)](#)
- [Demo and benchmark files](#)
- [Developers Ask Us Anything!](#)