

Chapter 1

Discovering Blender

In This Chapter

- ▶ Figuring out what Blender is and what it's used for
 - ▶ Understanding Blender's history
 - ▶ Getting familiar with the Blender interface
 - ▶ Becoming a part of the Blender community
-

In the world of 3D modeling and animation software, programs are usually expensive: Like, really, really, thousands-of-dollars-and-maybe-an-arm expensive. And there are *some* valid reasons for that. Software companies spend millions of dollars and countless hours developing these programs. And the large production companies that buy these programs for their staff make enough money to afford the high cost.

But what about us? You and I, the little guys? We are the ambitious dreamers with big ideas, high motivation . . . and tight budgets. How can we bring our ideas to life and our stories to a screen, even if it is our own computer monitors? Granted, we could shell out that cash (and hopefully keep our arms) for the expensive programs that the pros use. But even then, animation is a highly collaborative art and it's difficult to produce anything in a reasonable amount of time without some help.

We need quality software and a strong community to work, grow, and evolve with. Fortunately, Blender can provide us with both of these things. This chapter is an introduction to Blender, its background, its interface, and its community.

Getting to Know Blender

Blender is a free and open source 3D modeling and animation suite. Yikes! What a mouthful, huh? Put simply, Blender is a computer graphics program that allows you to produce high quality still images and animations using three-dimensional geometry. If you've seen one of the recent animated feature films or watched a television show where they explain how they made

an actor look like he's being chased by a giant monster even though he's really just standing in a big green room, you've seen what can be done with 3D computer graphics. In the right hands, Blender is capable of producing this kind of work. With a little patience and dedication, *your* hands can be the right hands.

One of the things that makes Blender different and special compared to other comparable 3D software is the fact that it is free and *open source*. This means that not only can you go to the Blender Web site (www.blender.org) and download the entire program right now without paying a dime, but the code that makes up the program, called the source, is also freely available for download. For most programs, the source code is a heavily guarded and highly protected secret that only certain people can see and modify (mostly programmers hired by the company that distributes the program). Because it's open source, anybody can see Blender's source code and make changes to it. The benefit of this is that, rather than having a small group of paid programmers work on the program, Blender can be improved by programmers all over the world!

Because of these strengths, Blender is an ideal program for small animation companies, freelance 3D artists, independent filmmakers, students beginning to learn about 3D computer graphics, and dedicated computer graphics hobbyists. Blender has a reputation for being difficult to understand for new users, but at the same time, it is also known for allowing experienced users to bring their ideas to life. Fortunately, helping you bridge that gap is the very reason this book exists.

Discovering Blender's History

The Blender we know and love today wasn't always free and open source. It's actually pretty unique in that it's one of the few software applications that was "liberated" from proprietary control by its user community.

Originally, Blender was written as an internal production tool for an award-winning Dutch animation company called NeoGeo, founded by Blender's original (and still lead) developer, Ton Roosendaal. In the late 1990s, NeoGeo started making copies of Blender available for download from their Web site. Slowly but surely, interest grew in this less-than-2MB program. In 1998, Ton spun off a new company, Not a Number (NaN), to try to market and sell Blender as a software product. NaN still distributed a free version of Blender, but also offered an advanced version with more features for a small fee. There was strength in this strategy and by the end of 2000, there were well over 250,000 Blender users worldwide.

Unfortunately, even though Blender was gaining in popularity, NaN was not making enough money to satisfy its investors, especially in the so-called “dot bomb” era that happened around that time. Because of this, in the early part of 2002, NaN shut its doors and stopped working on Blender.

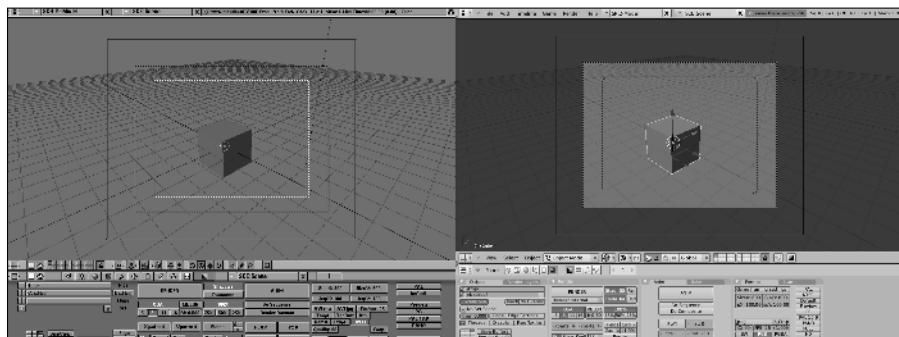
Ironically, this is where the story starts to get exciting.

Even though NaN went under, Blender had developed quite a strong community by this time, and this community was eager to find a way to keep their beloved little program from becoming lost and abandoned. In July of 2002, Ton provided a way. Having established a non-profit called the Blender Foundation, he arranged a deal with the original NaN investors to run the “Free Blender” campaign. The terms of the deal were that, for a price of €100,000, the investors would agree to release Blender’s source to the Blender Foundation for the purpose of making Blender open source. Initial estimations were that it would take as much as six months to raise the necessary funds. Amazingly, the community was able to raise that money in a mere *seven weeks*.

Because of the Blender community’s passion and willingness to put its money where its metaphorical mouth is, Blender was released under the GNU General Public License on October 13, 2002. With the source in the community’s hands, Blender had an avalanche of development and new features added to it in a very short time. We were even finally able to have Undo, a functionality that was conspicuously missing and highly desired since the initial releases of Blender by NeoGeo!

Six years later, the Blender community is larger and stronger than ever, and Blender itself is a powerful modern piece of software that competes with quality on par to similar software that costs thousands of dollars. Not too shabby. Figure 1-1 shows a screenshot of Blender from its early days compared to the Blender of today.

Figure 1-1:
Blender of
old (left)
versus
Blender of
today (right).



Making Open Movies and Games

One of the cool things about the programmers who write Blender is that many of them also use the program regularly. They're writing code not just because they're told to do it, but because they want to improve Blender for their own purposes. Part of this has to do with Blender's open-source nature, but quite a bit also has to do with the fact Blender was an in-house production tool, built for artists, based on their direct input, and often written by the artists themselves.

Seeking to get even more of this direct artist feedback, the Blender Foundation launched "Project Orange" in 2005. The purpose of the project was to create an animated short movie using open source tools, primarily Blender. A team of six members of the community were assembled in Amsterdam, in the Netherlands, to produce the movie. Roughly seven months later, *Elephants Dream* premiered and was released to the public as the first *open movie*, meaning not only that it was created using open-source tools, but all of the production files — 3D models, scenes, character rigs, and so on — were also released under a permissive and open Attribution Creative Commons license.

Due to the success of the Orange project, Ton established the Blender Institute in the summer of 2007 for the expressed purpose of having a permanent space to create open movie and game projects, as well as provide the service of training people in Blender. The first open project to come out of the Blender Institute was Project Peach, which, following the model of Orange, assembled a team of artists who produced a short comedic animation called *Big Buck Bunny*, which premiered in April 2008. Like *Elephants Dream*, all production files for the Peach project were released under an open license. Figure 1-2 shows an image from *Big Buck Bunny*.

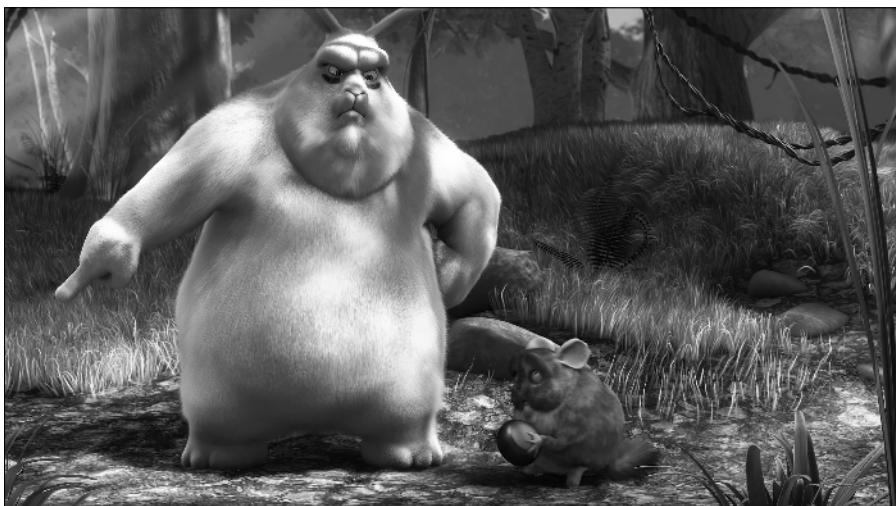


Figure 1-2:
Big Buck Bunny.

Not being inclined to rest on its laurels, the Blender Institute launched Project Apricot as the team creating *Big Buck Bunny* was wrapping up its production. Apricot was a project similar to Orange and Peach, but rather than create an animated movie, the goal here was to create a video game by combining Blender's strengths with the strength of the Crystal Space game engine. The result of this was *Yo Frankie!*, a game based on the "leader squirrel" character from the *Big Buck Bunny* project. The DVD containing this game shipped out in November 2008, containing playable levels in the Blender game engine as well as the Crystal Space engine. And of course, all of the content is freely available under a permissive Creative Commons license.

With the completion of each of these projects, the functionality and stability of Blender increased by a large degree. Orange brought improved animation tools, basic hair, and a node-based compositor. Peach provided enhanced particles for better hair and fur, optimizations for large scenes, improved rendering, and even better animation and rigging tools. Apricot revitalized Blender's internal game engine, which extended to better real-time visuals when modeling and animating in Blender. In fact, much of the content of this book wouldn't even exist without these projects. For example, Chapter 13 starts with using Blender's particle system to do exciting effects along with hair and fur. Half of Chapter 15's content is focused on the node compositor, a way of combining and enhancing still images and animations. In fact, nearly all of Part III is devoted to features that were enhanced or directly added for one of these three projects.

All of these projects continue to exhibit the strength of the Blender community. This is because each of them are financed in a large part by DVD pre-sales from users who understand that regardless of the project's final product, great improvements are the result and everyone benefits from that.

Joining the community

Congratulations! You're part of a community. As a Blender user, you're joining a very diverse group that spans all age ranges, ethnicities, professional backgrounds, and parts of the globe. We are a passionate bunch: proud of this little 3D program and more than willing to help others enjoy using it as much as we do. Have a look at Chapter 18 for a list of community resources that are invaluable, not only for discovering the intricacies of using Blender, but also for improving yourself as an artist. You can find

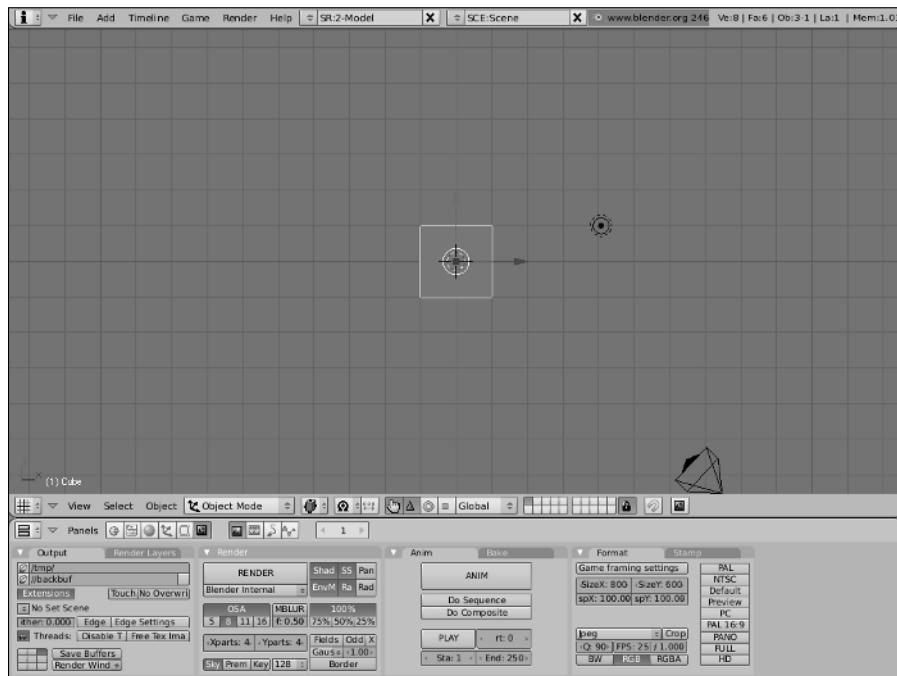
innumerable opportunities for critique, training, discussion, and even collaboration with other artists, some of whom might also be Blender developers. I've made quite a few good friends and colleagues through the Blender community, both through the various community Web sites as well as by attending events like the annual Blender Conference. I go by the name "Fweeb" on these sites and I look forward to seeing you around!

Getting to Know the Interface

Probably one of the most daunting aspects of Blender for newcomers and long-time 3D professionals alike is its unique and somewhat peculiar interface. It's arguably the most controversial feature Blender has. In fact, merely calling it a "feature" might raise the blood pressure of some of you who tried using Blender in the past, but gave up in frustration when it did not behave as you expected. Figure 1-3 shows what you're presented with when you start Blender for the first time. It's been called everything from "brilliantly thought-out" to "thrown together by a pack of monkeys."

This book explains some of the design decisions in Blender's interface and ultimately allows you to be productive with it. Who knows, you might even start to like it and wonder why other programs don't work this way!

Figure 1-3:
The first time you start Blender, the entire world might seem like an alien planet.



Fast to use versus fast to learn

One of the key things to remember is that Blender was originally designed as an in-house tool for commercial production. Working in that industry (especially television production) requires very short turn-around times and extremely tight deadlines. For this reason, 3D artists have to work very quickly to produce high quality work in a short period of time. Blender was built from the ground up to facilitate this need. And because artists worked side-by-side with the developers, they could tailor the whole program to match the way they worked.

The upside to this is that the program evolved with the artists and enabled them to successfully produce great work at a blazingly fast rate. The downside is that, as with most things that are customized, Blender became somewhat difficult to understand for people who had never been exposed to it before. This is what I mean when I say “fast to use versus fast to learn.” You can be extremely productive with Blender after you understand how it thinks. However, your first few projects with Blender might be arduous. Of course, alleviating that potential pain is what this book is all about.

The Blender non-blocking interface

The first thing to understand about Blender’s interface is the concept of a *non-blocking interface*. This means that windows in Blender never overlap one another and working in one window typically won’t restrict you from working in any of the others. As an example, in most software, if you want to open a new file or save your project, a file browser dialog box pops up for you to do this. This is an overlapping window. Not only does it block things behind it from view, but it usually also prevents you from making any changes to your file. This isn’t the case with Blender. In Blender, the file browser shows up in a window just like any other, and it makes perfect sense to be able to make a couple tweaks to your scene before hitting the save button. Figure 1-4 shows what this might look like.



At first, this way of working might seem to be really restrictive. How do you see different types of windows? Can you see them at the same time? Everything looks like it’s nailed in place; is it even possible to change anything? Fortunately, all of these things are possible and you get the benefit of never having your view of one window obstructed by another. This is a great way to be able to see what’s going on in your file at a glance.

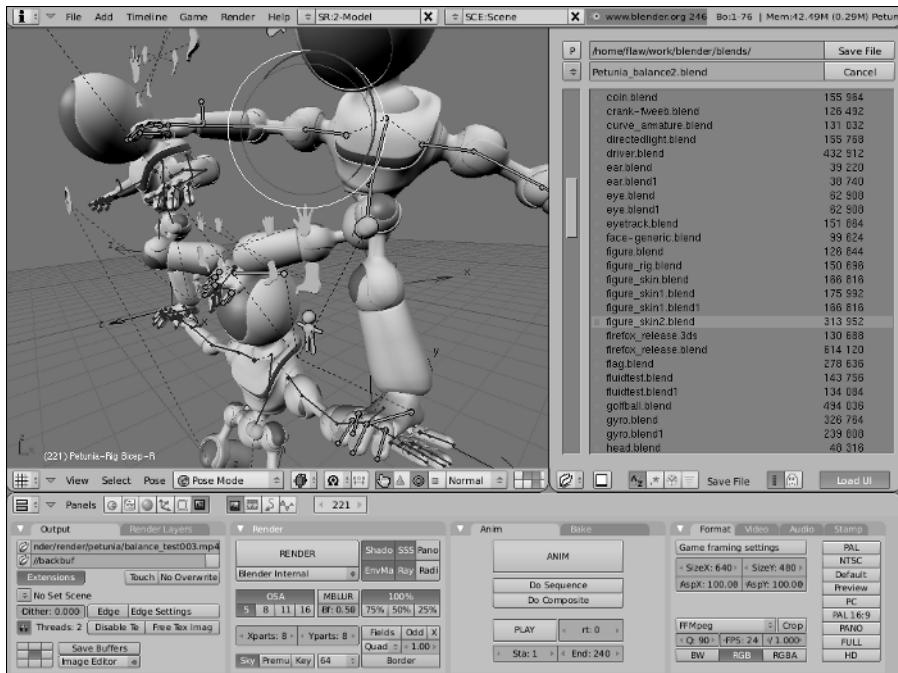


Figure 1-4:
Doing those
last couple
of tweaks
before
finally
saving.

The windows can all be modified and changed in the same way: Change the size of windows by left-clicking the seam between windows and dragging it to a new position. This increases the size of one window while reducing the size of some of those that adjoin it. You can also add new windows by right-clicking the seam and choosing Split Area from the menu that pops up. After you do that, a line appears under your mouse cursor. Move your mouse to where you would like to create a new seam and left-click to confirm. If you decide you no longer wish to have this additional window, you can remove it by right-clicking that seam and selecting Join Areas. This darkens the window your mouse is in and draws an arrow to indicate which window you would like to remove. Figure 1-5 shows the process of splitting an area to create a new window and then removing that window by joining areas. When I work in Blender, I find myself constantly changing the screen layout by splitting and joining new windows as I need them.

When you right-click the seam between windows, you see a third option that says No Header. Choosing this option removes the header bar from the window your mouse cursor was in last. You can put the header back by right-clicking the seam again and choosing Add Header. You can also change the location of the header to either the top or bottom of the window it belongs to. To do so, right-click the header bar and choose one of the three available options: Top, Bottom, or No Header.



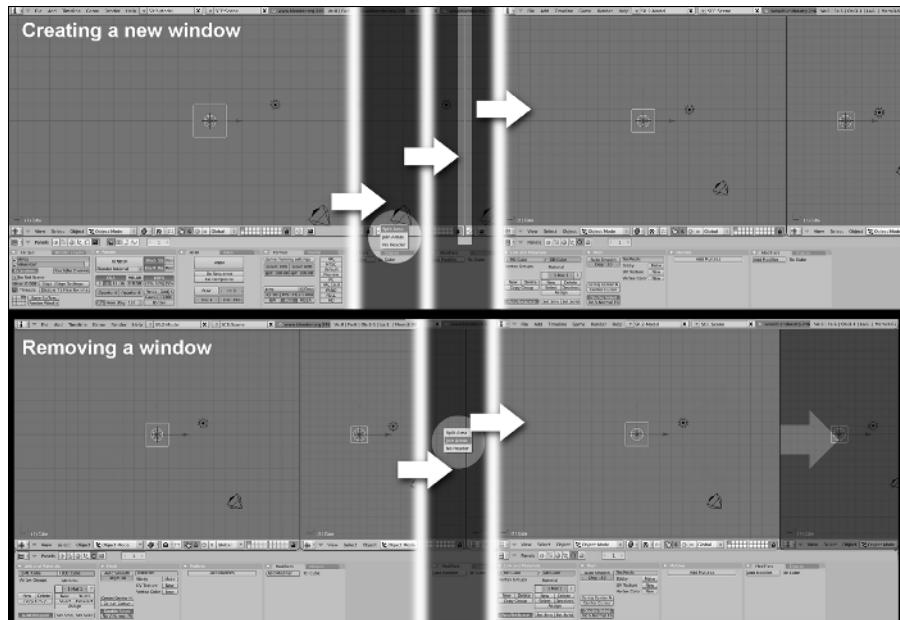


Figure 1-5:
Creating
a new
window
and then
removing
that
window.

When working in Blender, you also occasionally need to maximize a window. This is particularly useful when you're working on a model or scene and you just want to get all of the other windows out of your way so you can use as much screen space as possible. To maximize any window, hover your mouse over that window and press Shift+Spacebar. You can toggle back to the tiled screen layout by pressing Shift+Spacebar again. These options are available in almost all window types by choosing **View**→**Maximize/Tile Window** from that window's header. The only exception to this is the Buttons window. In this window, the option is available by choosing **Panel**s→**Maximize/Tile Window**.



You might notice that the hot keys next to these menu items are Ctrl+UpArrow for maximizing and Ctrl+DownArrow for tiling, rather than Shift+Spacebar. Those hotkeys also work, but I find that I don't have to move my left hand as much to hit Shift+Spacebar, so that's much more convenient for me.



This non-blocking window philosophy, combined with the fact that Blender's entire interface is written in a standardized programming library for graphics called OpenGL, is the precise reason that Blender looks the same, no matter where you run it. Whether you run it from Linux, Windows, a Mac, or even a cell phone, Blender looks and behaves like Blender. There's an additional benefit to being written in a 3D library like OpenGL: Many parts of Blender's interface allow you to zoom in on them. Try it! Place your mouse in the Buttons window (the bottom window) and hold Ctrl while scrolling your mouse wheel. You can make the panels in this window much larger or smaller than they are by default. Pretty cool!

So now you have a taste of how Blender thinks. That's not so bad, is it? Of course not! Of course, this little bit is just a start into the wild world that is Blender and its interface. The next chapter goes into the interface in more detail and gets you started with editing objects in Blender.