THE EXPERT'S VOICE® IN JAVA

Beginning Java 8 Games Development

LEARN THE FUNDAMENTALS OF JAVA 8 GAME PROGRAMMING

Wallace Jackson

Apress®

For your convenience Apress has placed some of the front matter material after the index. Please use the Bookmarks and Contents at a Glance links to access them.

Apress®

Contents at a Glance

Introduction

The Java Programming Language is currently the most popular object-oriented (OOP) programming language in the world today. Java runs on everything from SmartWatches to HD Smartphones to Touchscreen Tablets to eBook Readers to Game Consoles to SmartGlasses to Ultra-High Definition (UHD) 4K Interactive Television Sets, with even more types of consumer electronics devices, such as those found in the automotive, appliances, health care, digital signage, security, and the home automation market, increasingly adopting the open source Java platform for use in their hardware devices as time goes on.

Since there are literally billions of Java compatible consumer electronics devices, owned by billions of users all over the world, it stands to reason that developing popular Java 8 Games for all of these people could be an extremely lucrative undertaking, given that you have the right game concept, artwork, game design, and optimization work process, of course.

Java 8 (and its multimedia engine, JavaFX 8) code can run on just about every operating system out there, including Windows XP; Vista, 7, 8, and 9; all Linux distributions; 32-bit Android 4 and 64-bit Android 5; Open Solaris; Macintosh OS/X, iOS; Symbian, and Raspberry Pi – it's only a matter of time before the other popular OSes add support for this popular open source programming language. Additionally, every popular Internet browser has Java built in! Java provides the ultimate flexibility in installing software, as an application, or in the browser as an applet. You can even drag a Java application right out of the browser, and have it install itself on that user's desktop! Java 8 is a truly remarkable technology.

There are a plethora of embedded and desktop hardware support levels currently for Java 8 (and for JavaFX 8.0) including the full Java SE 8, Java SE 8 Embedded, Java ME (Micro Edition) 8, and Java ME 8 Embedded, as well as Java EE 8 for Enterprise Application Development. Talk about being able to "code once, deliver everywhere!" That is the dream of every programmer, and Oracle is making it a reality with the powerful Java 8 multimedia programming platform.

This book will go a long way toward helping you to learn exactly how to go about developing Java 8 games, using the Java programming language in conjunction with the recently added JavaFX 8.0 multimedia engine. These Java 8 game applications will be able to run across a plethora of Java compatible consumer electronics devices. Developing Java 8 game applications that play smoothly across all of these different types of consumer electronics devices requires a very specific work process, including asset design, game code design, and optimization, all of which I will be covering during this book.

I wrote the Beginning Java 8 Game Development title from scratch, using a real-world client game project that I am actually working on, and will be delivering to the public sometime in 2015. I am targeting those readers who are Beginning Game Developers, and who had not coded in Java 8 and JavaFX 8.0. These readers are technically savvy, but they are not that familiar with object-oriented computer programming concepts and techniques. Since Java is now at Version 8u40, this book will be more advanced than many of the other Java books out there. Java 8 has added some very advanced features, such as the JavaFX 8.0 API, which gives Java 8 its own multimedia engine, supporting SVG, 2D, 3D, audio, and video media.

I designed this book to contain a comprehensive overview of the optimal Java 8 game development work process. Most beginning Java application development books only cover the language, however. If you really want to become that well-known Java game application developer that you seek to become, you will have to understand as well as master all of the areas of game design, including multimedia asset creation, user interface design, Java 8 Programming, JavaFX 8.0 class usage, and data footprint, memory, and CPU usage optimization. Once you've mastered these areas – hopefully, by the end of this book, you will be able to create the memorable user experience that will be required to create popular, best-selling Java 8 games. You can do it; I know you can!

Java 8 games are not only developed using the NetBeans 8.0 Integrated Development Environment (IDE) alone, but also in conjunction with the use of JavaFX 8 and several other different types of new media content development software packages. For this reason, this book covers the installation and use of a wide variety of other popular open source software packages, such as GIMP 2.8 and Audacity 2.0.6, in conjunction with developing Java 8 game applications using the NetBeans 8.0 IDE and the JavaFX new media engine, which brings the "wow factor" to the Java programming language.

I am architecting this book in this fashion so that you can ascertain precisely how your usage of new media content development software will fit into your overall Java 8 game development work process. This comprehensive approach will serve to set this unique book title distinctly apart from all of those other Java 8 game application development titles that are currently out on the market. The book starts out in Chapter 1 with downloading and installing the latest Java 8 JDK as well as the NetBeans 8.0 IDE, along with several popular open source content development applications.

In Chapter 2, you will learn about NetBeans 8.0, and create your first Java 8 game application, and look at useful NetBeans features, such as code completion and code profiling. In Chapter 3, you will learn about the fundamentals of the Java 8 programming language, which you'll be implementing to create a Java 8 game during the remainder of the book.

In Chapter 4, you will learn all about the JavaFX 8.0 new media engine (API) and how its impressive features can take your Java 8 game development and place it into the stratosphere. In Chapter 5, you will learn all about the JavaFX 8 FXML (Java FX Markup Language) and about the underlying concepts of developing new media assets such as digital audio, digital images, digital video, 2D scalable vector graphics (SVG), and 3D geometry, for use with Java 8 games. In Chapter 6, you will learn about game design concepts, and create the foundation for your Java 8 game, its user interface, and a splashscreen. Thus the first third of this book is foundational material, which you'll need to be able to understand how NetBeans 8.0, Java 8, JavaFX 8.0, and various new media asset types supported by the JavaFX engine function together as a platform.

In Chapter 7 we will start to create the various game engines, starting with the game play loop 60 FPS timing engine, and we will learn about the JavaFX 8 Animation, Timeline, KeyFrame, KeyValue, Interpolator, and AnimationTimer classes, which allow the Java 8 game to tap into the JavaFX pulse event timing engine that gives Java 8 its multimedia power.

In Chapter 8, we will create your game Actor and Hero Java abstract classes, the Actor engine, if you will, which will allow us to create the different types of game play components that we will need for the Java 8 game. This will teach you how to create custom foundational classes for a game project, and you will look at the Node, SVGPath, Shape, Image, and ImageView classes as we incorporate these JavaFX class (object) types into our Java 8 Game Actor design.

In Chapter 9, you will learn how to add interactivity to your Java 8 Game projects, using event handling. We will add an event processing engine, which will process all of the different types of action, key, mouse, and drag events that you are likely to utilize in your Java 8 game development work process in the future when you create your own custom games.

In Chapter 10, you will learn about Java List, Set, and Array classes. These are called Java collections, and we will create a custom Actor management engine, which we will call the CastingDirector class, during this chapter. This will allow you to automate the task of keeping track of the cast of your game for each level, and will be used for collision detection.

In Chapter 11 we will start coding our primary Actor class for the InvinciBagel character, and add Java 8 code that controls movement on the screen, so that we can start to work on fusing character animation with game player key use so that we can allow our game players to control the InvinciBagel character completely. This involves "wiring up" the Bagel class to the GamePlayLoop (game play timing class created in Chapter 7) class, so we can start working in the fourth dimension of time.

In Chapter 12 you will use your Actor and Hero abstract classes that you created in Chapter 8 to create the InvinciBagel primary character and his Bagel.java class, as well as learn how to implement code that sets the boundaries for your Java 8 game, so that the Actor does not go off the screen, forcing him stay inside of the field of play for the game.

In Chapter 13 you will add different InvinciBagel sprite image states into your Java 8 game, and when these are combined with the movement you coded in Chapters 11 and 12, allow your InvinciBagel character to run, jump, fly, land, wait impatiently to be moved, and even turn sideways to evade bullets.

In Chapter 14, you will create a series of Prop classes that will allow you to place fixed props and obstacles into your Java 8 game levels. You will learn how to use one digital image asset to create four different scenery props, using the JavaFX ability to flip and mirror your image assets around either (or both of) their X and Y axes.

In Chapter 15, you'll implement your Java 8 game audio engine, using the JavaFX AudioClip class, which allows digital audio sequencing to be integrated into your Java 8 game play, taking it an order of magnitude higher, by stimulating the aural senses of your game player. You'll learn how to optimize digital audio assets so well, that you will not have to use any lossy compression, giving you perfect audio samples, and showing you exactly how much of the system's memory your audio assets will be using.

In Chapter 16, we'll start getting into advanced topics, such as designing collision polygons using SVG data and the GIMP 2.8 and PhysicsEditor software packages. We will also learn about the JavaFX Bounds and Node classes, and how collision detection is accomplished for Java 8 game development, using the .getBoundsInLocal() and .getBoundsInParent() method calls, in conjunction with the Node.intersects() and Shape.intersect() method calls.

In Chapter 17, we will pull everything together, and focus solely on implementing your game play. You will create Actor subclasses for Treasure, Projectile, and Enemy, and create an auto-attack engine that will turn a game player's PC or mobile device into his or her adversary. We look at the most advanced topics, such as physics and AI, during this chapter, after which you will have enough of a foundation to create your own Java 8 games, using your own intellectual property!

This book attempts to be the most comprehensive Java 8 game application development programming title on the market, by covering most, if not all, of the major Java 8 and JavaFX classes that will need to be used to create Java 8 Game Applications. Some of these include the Image, ImageView, Group, Node, StackPane, Scene, Stage, Application, ListArray, HashSet, Arrays, AudioClip, MediaPlayer, URL, Button, Shape, HBox, SVGPath, Insets, AnimationTimer, and more.

If you're looking for the most comprehensive, up-to-date overview of the Java 8 programming language for games, including JavaFX 8.0 and NetBeans 8.0 IDE all seamlessly integrated with new media content development work processes, as well as a "soup to nuts" knowledge about how to optimally use these technologies in conjunction with the leading open source new media game content design and development tools, then this book will really be of significant interest to you.

It is the intention of this book to take you from being a Beginner in Java 8 game application development to a solid intermediate knowledge level regarding Java 8, NetBeans 8, and JavaFX 8.0 game application development. Be advised that this book, even though it's ostensibly a Beginner title, contains a significant amount of technical knowledge. All of the work processes that are described during the book may well take more than one read through to assimilate into an application development knowledge base (your quiver of technical knowledge). It will be well worth your time, however, rest assured.

CHAPTER 1

Setting Up a Java 8 Game Development Environment

Welcome to the book *Beginning Java 8 Games Development*! Let's get started by creating a solid development software foundation for use with this book. The core of this foundation will be **Java SDK** (**Software Development Kit**) **8**, also called **JDK (Java Development Kit) 8**. I will also set you up with **NetBeans IDE 8.0** (**Integrated Development Environment**), which will make coding Java 8 games much easier. After that, I will introduce you to the latest open-source new media content creation software packages for digital illustration (Inkscape), digital imaging (GIMP [GNU Image Manipulation Program]), digital video (EditShare Lightworks), digital audio (Audacity), and 3D modeling and animation (Blender). At the end of the chapter, I will also suggest some other professional-level software packages that you should consider adding to the professional game development workstation that you will be creating over the course of this chapter.

To get the best results from all this free, professional-level software, you will want to have a modern, **64-bit** workstation with at least 4GB of system memory (6GB or 8GB would be even better) and a **multicore** processor (central processing unit [CPU]), such as an AMD FX-6300 (hexa-core), AMD FX-8350 (octa-core), or Intel i7 (quad-core). Workstations such as these have become commodity items and can be purchased at Walmart or `Pricewatch.com` at an affordable price.

The first thing that you will do in this chapter is make sure that you have **removed** any of the **outdated versions** of Java, such as Java 7 or Java 6, or any outdated versions of NetBeans, such as NetBeans 7 or NetBeans 6. This involves **uninstalling** (removing or deleting completely) these older development software versions from your workstation.

You will do this using the Windows program management utility **Programs and Features**, which can be found in the Windows operating system (OS) **Control Panel** suite of **Windows OS Management Utilities**. There are similar utilities on the Linux and Mac platforms, if you happen to be using one of these less commonly used OSs. Because most developers use Windows 7, 8, or 9, you will be using the Windows 64-bit platform for the examples in this book.

Next, I will show you where exactly to go on the Internet to get these software packages, so get ready to fire up your speedy Internet connection so that you can download nearly a gigabyte of all-new game content production software! After you download the latest versions of all this software, you will install the programming and content development packages and configure them for use with this book.

The order in which you perform these software installations is important, because Java JDK 8 and Java 8 Runtime Environment (JRE) form the foundation of NetBeans IDE 8.0. This is because NetBeans IDE 8.0 was originally coded using the Java programming language, so you will see just how incredibly professional a piece of software can be using this language. Thus, the Java 8 software will be the first software you install.

After you install Java 8, you will then install NetBeans 8.0, so that you have a graphical user interface (GUI), on top of the Java programming language, which will make the Java software development work process easier. After you have these two primary software development tools installed, you will get a plethora of new media content creation software packages, which you can use in conjunction with Java 8 and NetBeans 8.0 to create 2D and 3D games.

Prepare a Workstation for Java 8 Game Development

Assuming that you already have a professional-level workstation in place for new media content development and game development, you need to remove all the outdated JDKs and IDEs and make sure that you have the latest V8 (not the drink, silly!) Java and NetBeans software installed on your system and ready to go. If you are new to this and do not have a game-appropriate workstation, go to Walmart or Pricewatch.com, and purchase an affordable multicore (use a 4-, 6- or 8-core) 64-bit computer running Windows 8.1 (or 9.0 if it is available) that has 4GB, 6GB, or 8GB of DDR3 (1333 or 1600 memory access speed) system memory at the very least and a 750GB, or even 1TB, hard disk drive.

The way that you remove old software is through the Windows **Control Panel** and its set of utility icons, one of which is the **Programs and Features** icon (Windows 7 and 8), displayed in Figure 1-1. Note that in earlier versions of Windows, this utility icon may be labeled differently, probably as something like **Add or Remove Programs**.

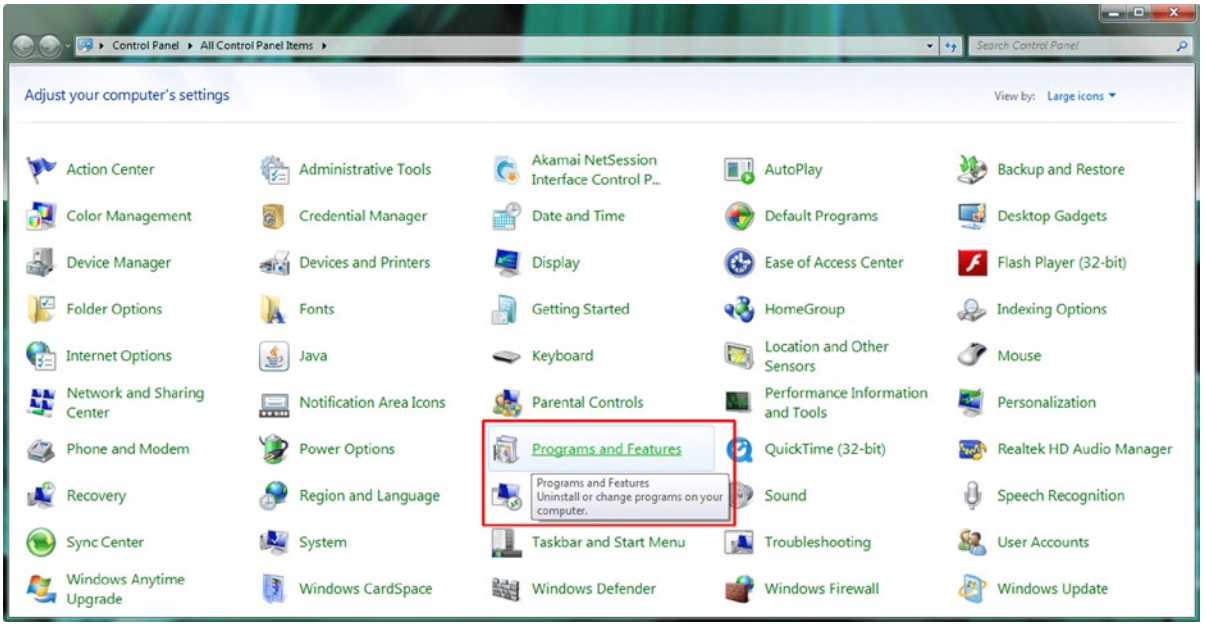

Figure 1-1. *Use the Programs and Features utility icon to uninstall or change programs on your computer workstation*

Click the Programs and Features link, or double-click the icon in previous versions of Windows, to launch the utility. Then, scroll down to see if you have any old versions of the Java development tools (Java 5, Java 6, or Java 7) installed on your workstation. Note that if you have a brand new workstation, you should find no preinstalled versions of Java or NetBeans on your system. If you do find them, return the system, as it may have been used previously!

As you can see in Figure 1-2, on my Windows 7 HTML5 development workstation, I had an older version of Java, Java 7, installed (on November 29, 2013), taking up 344MB of space. To remove a piece of software, **select it** by clicking it (it will turn light blue), and then click the **Uninstall** button, shown at the top of the figure. I left the **tool tip**, which says, "**Uninstall this program,**" showing in the screenshot so that you can see that if you **hover** your mouse over anything in the Programs and Features utility, it will tell you what that feature is used for.

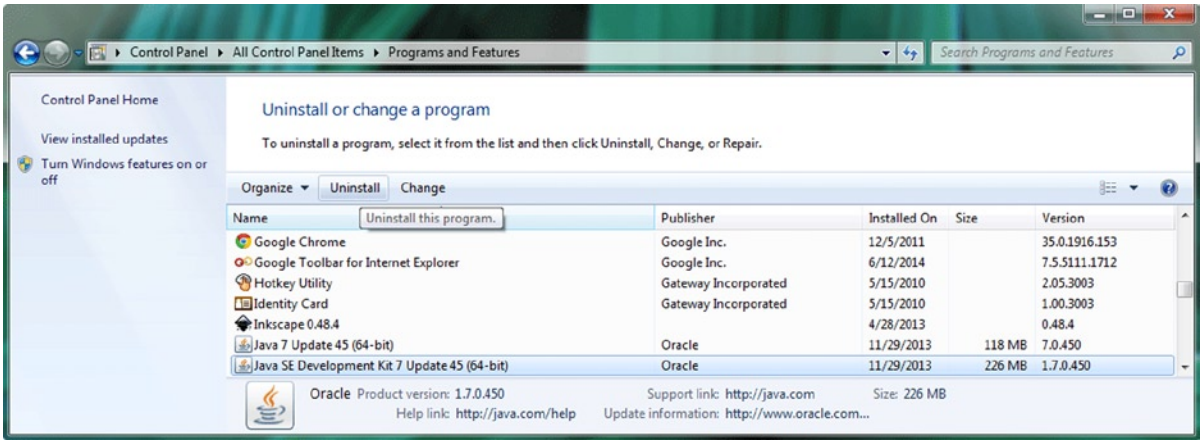

Figure 1-2. *Select any version of Java older than the current version (Java 8), and click the Uninstall button at the top*

Once you click the Uninstall button, the utility will remove the older version of Java. If you want to keep your old Java project files, make sure to back up your Java project files folder (if you have not done so already, that is). Make sure that you back up your workstation's hard disk drive regularly so that you do not lose any of your work.

Also make sure that you uninstall all versions of Java; in my case, there were 64-bit Java 7 update 45 and Java SDK 7u45, used to run or execute IDEs, such as NetBeans (or Eclipse), that were coded using the Java programming language.

Next, you will want to ascertain if there are any older versions of the NetBeans IDE on your workstation. In my case, as you can see in Figure 1-3, there was indeed a NetBeans 7 IDE installation currently on my 64-bit Windows 7 workstation. I selected this for removal and then clicked the **Uninstall/Change** button, shown at left, which brought up a custom **Uninstall Summary** dialog, shown at right.

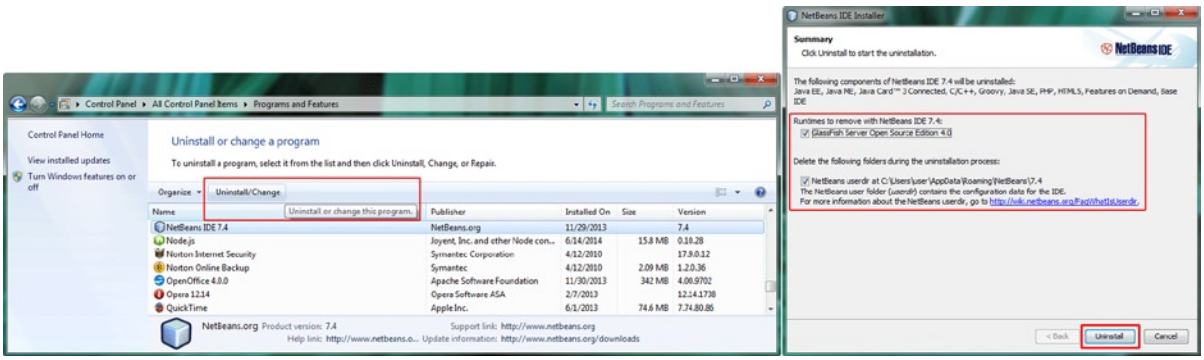

Figure 1-3. *Find and select any version of NetBeans that is older than version 8.0; also, uninstall old GlassFish versions*

Manufacturers (in this case, the NetBeans development team) can create custom Uninstall Summary dialogs for their products to use during the uninstall process, as you can see here. This dialog allows you to select whether you want to uninstall GlassFish Server 4 and the NetBeans **UserDir Configuration** folder. Because you are installing new versions of NetBeans and GlassFish, select both check boxes, and then click the **Uninstall** button.

Downloading Java JDK 8 and NetBeans 8.0

Now that the outdated versions of Java and NetBeans have been removed from your workstation, you will need to go on the Internet, to the **Oracle** and **NetBeans** web sites, respectively, to get the latest development SDKs and IDEs. I will show you how to do this using Google's search engine (I am using this method in case the download links, or URLs, ever change) as well as demonstrate what the direct download URLs are currently, at the time of writing this book.

Let's get Java 8 first, as that is the foundation for everything that you are going to be doing as you read through this book. A Google search for **Java JDK 8** will give you the search result that Oracle's Java **Downloads** page, which is located in the Oracle Technology Network section, as shown at the top of the screenshot in Figure 1-4. The URL for this page is currently **`www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html`**. It is important to note that this URL could change at any time in the future and that you can always use Google Search to find the latest one. Before you can download the 170MB SDK installer file for Windows 7/8 64-bit, you will need to click the **radio button** next to the **Accept License Agreement** option shown at the top left of the Java 8 download table. Once you accept the license agreement, these 11 OS-specific links will become activated for use.

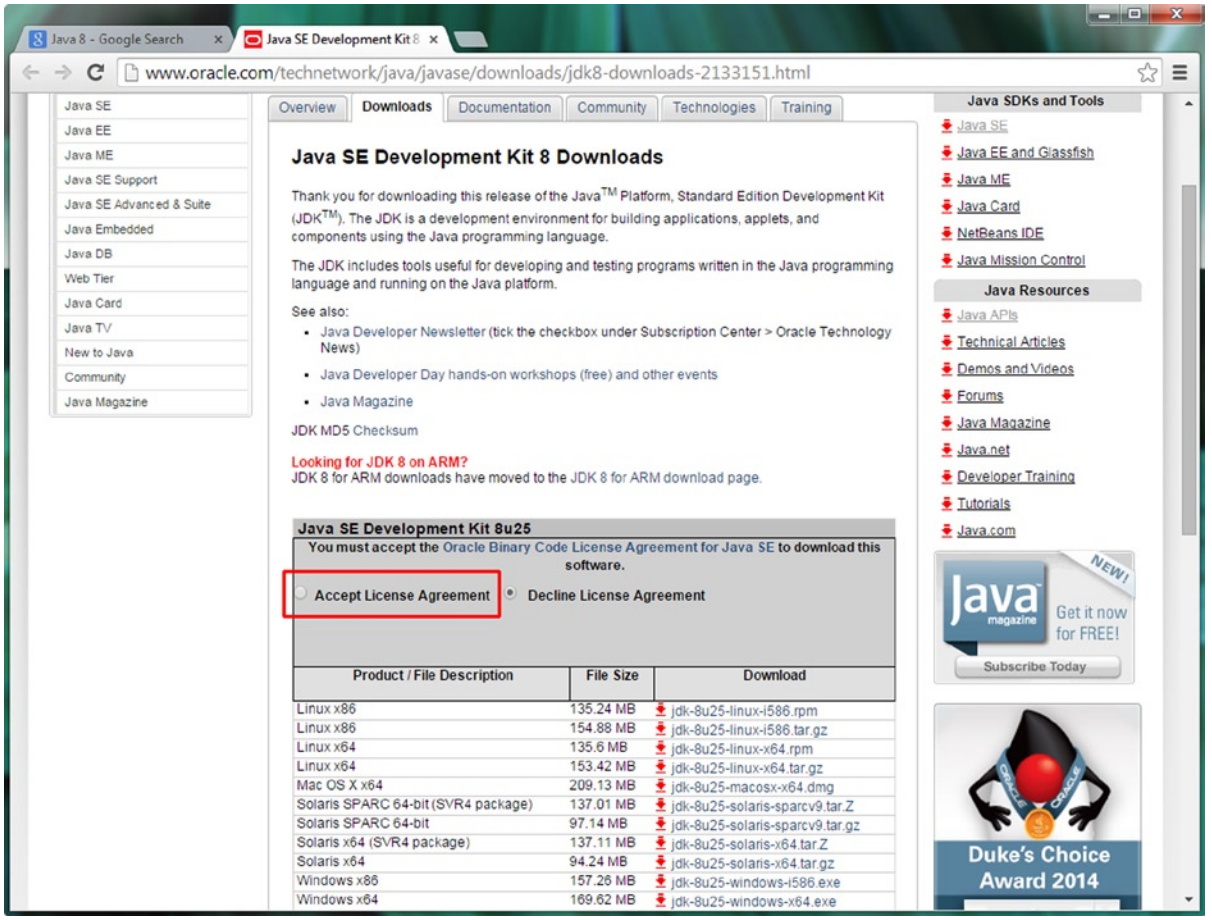

Figure 1-4. *Google the term "Java JDK 8," open the JDK 8 Downloads page, and select Accept License Agreement*

Be sure to match the Java JDK 8 software that you download to your OS and bit level (**x86** signifies a **32-bit**-level OS). Most modern-day workstations use a **64-bit** Linux, Mac, Solaris (Oracle), Windows 7, or Windows 8 OS. This will be specified with the **x64** delineation after the name of the OS.

To find out the bit level of the OS, on **Windows 7**, open the **Start Menu,** right-click the **Computer** entry, and select the **Properties** option, at the bottom of the context-sensitive menu. On Windows 8, you click **Start** (a window pane icon at the bottom left of your desktop if you are in Windows 7 desktop mode) and then the **down-arrow icon** at the bottom left, then click the **PC Settings** purple gear icon and finally the **PC Info** entry at the bottom left of the screen. In both use cases, there should then be a text entry that says **System type** and either **32-bit Operating System** or **64-bit Operating System**.

Now that you have downloaded the Java JDK 8 installer, the next thing that you need to do is download is NetBeans IDE 8.0. Do a Google search for the term **NetBeans 8.0**, as is shown at the top of Figure 1-5, and click the **Download** search result option, which will take you to the NetBeans IDE 8.0.1 Download page (currently **https://netbeans.org/downloads**). If you want to keep both tabs open in the browser, as I did, then right-click the **Download** link, and select the **Open link in new tab** option.

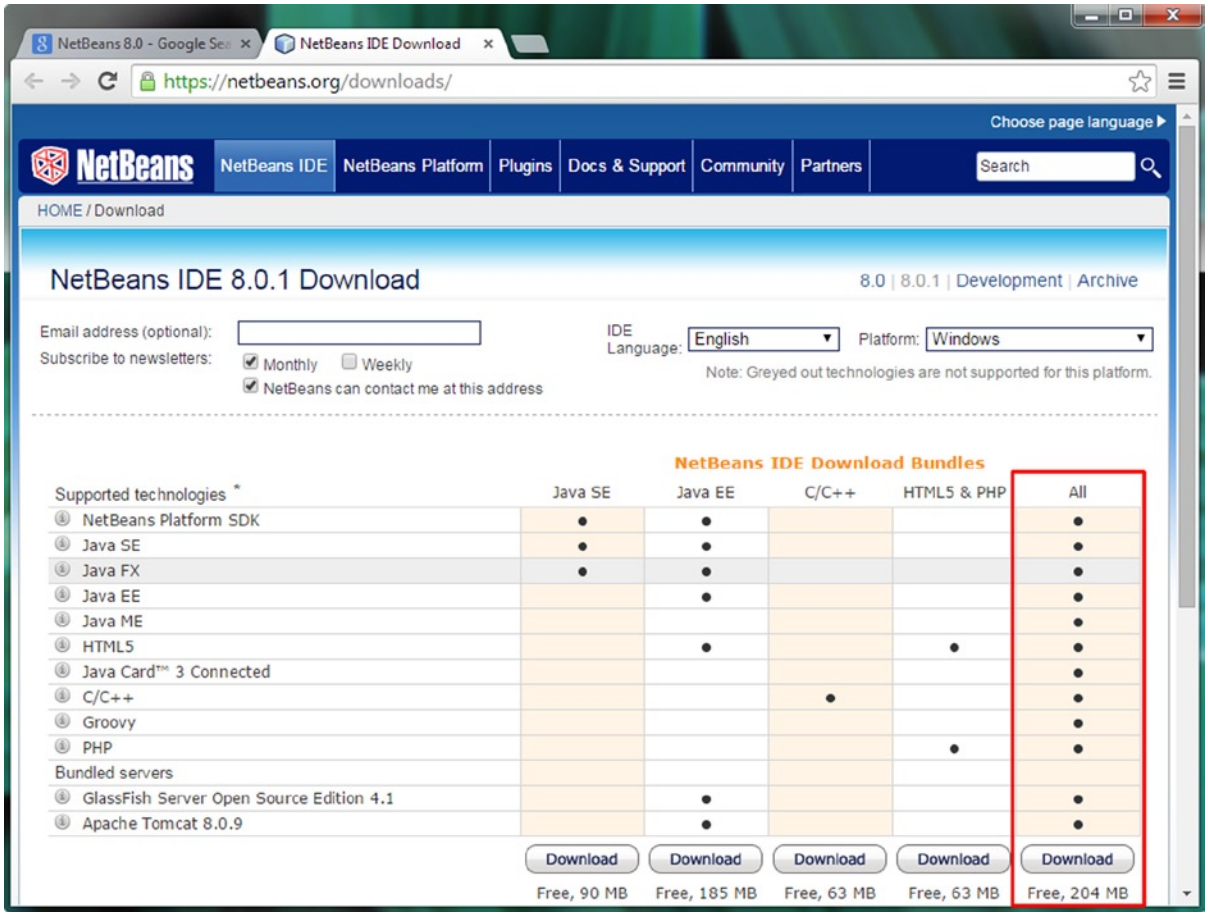

Figure 1-5. *Google the term "NetBeans 8.0," open the NetBeans IDE 8.0.1 Download page, and download all versions*

Once you are on the NetBeans IDE 8.0 Download page, select the **language** and **platform** (OS) that you are using from the drop-down menus at the top right of the page. I chose **English** and **Windows**. Now, you can click one of the three **Download** buttons at the bottom of the page to download a NetBeans IDE 8.0 that supports JavaFX 8 new media (and that will therefore support game development) programming language (application programming interface [API]). You will learn more about what an API is in Chapter 3, when I cover the Java programming language in detail.

If you are only going to develop Java SE (Standard Edition) and JavaFX applications (games) for **individuals**, then click the first button. If you are going to develop Java EE (Enterprise Edition) and JavaFX applications (games) for **enterprise** (business), then click the second button. If you are going to develop both JavaFX and HTML5 applications (games), which is what I do for my business, then you click the fifth **Download** button, and download the "All" version of NetBeans IDE 8.0. This version will allow you to develop in all the programming languages supported by NetBeans!

Because the NetBeans IDE is free, and your workstation hard disk drive can handle huge amounts of data, I recommend that you install this 204MB **All** version of the IDE, in case you ever find that you need any of the other capabilities that NetBeans IDE 8.0 is able to provide for you as a software developer (Java EE, Java ME, PHP, HTML5, Groovy, GlassFish, Tomcat, C++). This is an extra 120MB if you are going to install the client-side, or Java SE IDE, version, but is less than 20MB of extra disk space if you are going to install the server-side, or Java EE IDE, version.

Once you click the **Download** button, the software download will commence. After it is finished, you will be ready to install Java 8 and then NetBeans IDE 8.0. Finally, to complete the setup of your comprehensive Java 8 game development workstation, you will get some ancillary new media content tools. You will be able to use the workstation as you read through this book (and thereafter) to create epic Java 8 game deliverables! This is getting exciting!

Installing the Java 8 Software Development Environment

NetBeans IDE 8.0 requires Java to be installed in order to function (run), so you will need to install the JDK and JRE first. Because you want to develop games using the latest and most feature-filled version of Java, so you are going to be installing Java 8, which was released in 2014. Installing the latest version of software ensures that you have the newest features and the fewest bugs possible. Make sure to check often that you are using the latest version of all your software packages; after all, these are open source and free to download, upgrade, and use!

The first step is to find where you downloaded your installer files to on your system. The default should be set to the **Download** folder in Windows. I downloaded mine to a `C:/Clients/Java8` folder, as you can see in Figure 1-6.

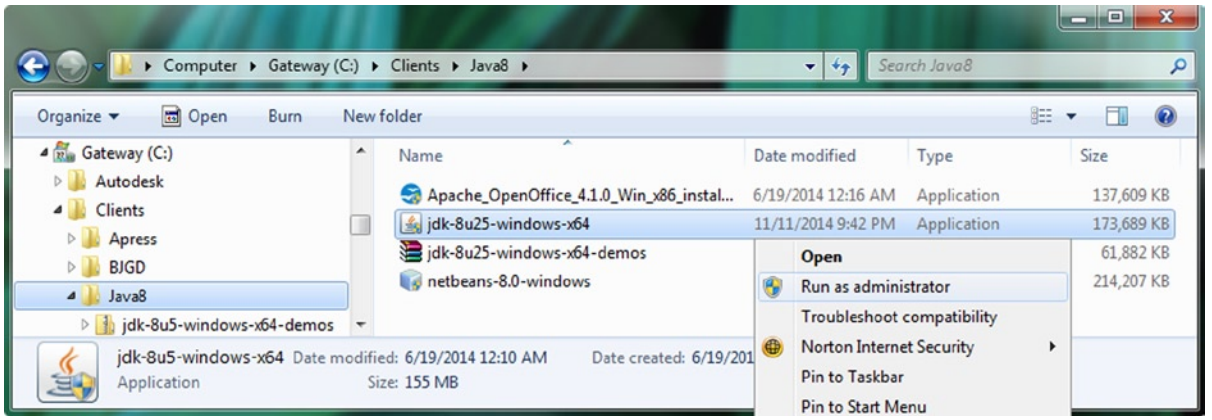

Figure 1-6. *Find the JDK 8 install file on your hard disk drive, right-click it, and select Run as administrator*

The file will be named using the format **jdk-version-platform-bitlevel**, so find the latest version (in this case, it was `jdk-8u25-windows-x64`). Right-click it, and select the **Run as administrator** option so that the installer has all the OS "permissions" that it needs to create folders, transfer files into them, and the like.

Once you launch the installer, you will see the **Welcome** dialog, shown in Figure 1-7 (left). Click the **Next** button to advance to the **Select Features to Install** dialog, shown in Figure 1-7 (right), and accept the defaults.

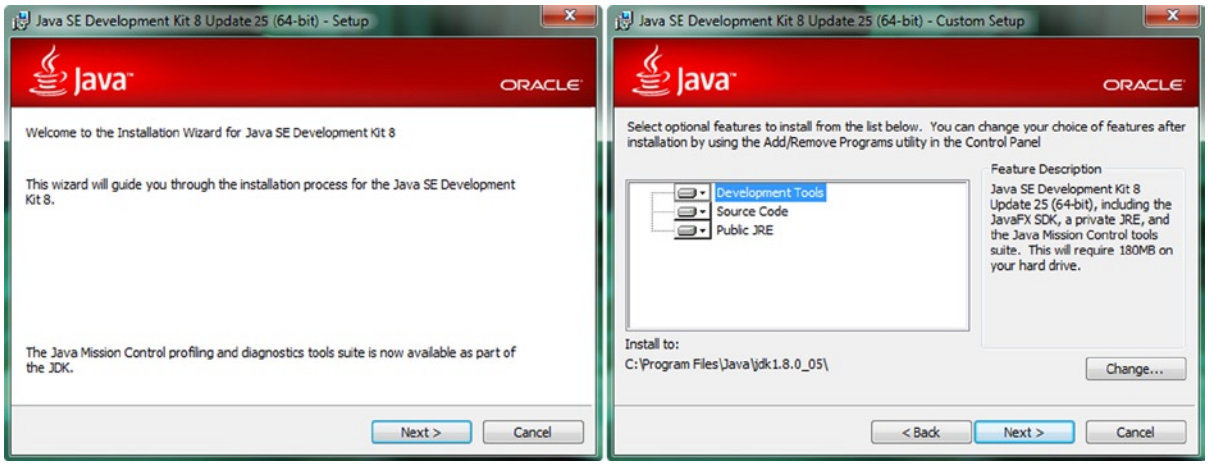

Figure 1-7. *Click Next in the Welcome dialog to advance to the Select Features to Install dialog, and then click the Next button*

As you can see, the installer will install **180MB** of software into the **`C:\ProgramFiles\Java\jdk1.8.0_25`** folder on your workstation. Click the **Next** button to start the installation process, which will extract the installation files and then copy them onto your system, using an animated progress bar, as displayed in Figure 1-8 (left).

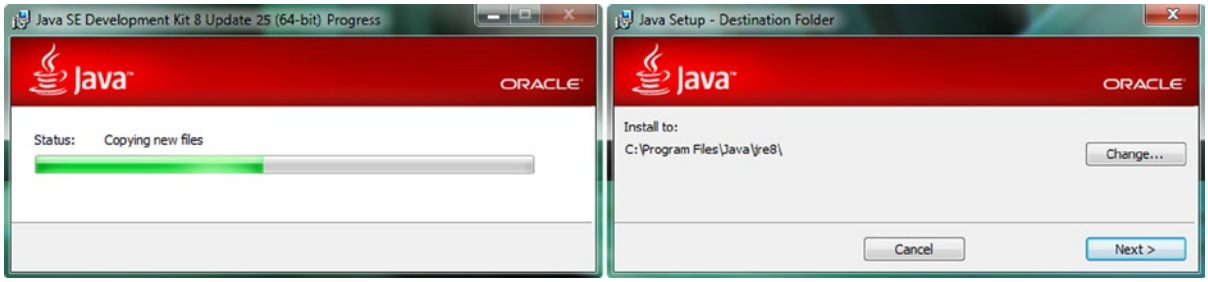

Figure 1-8. *Java 8 installation will extract and copy install files (left) and then suggest the installation directory (right)*

After the Java SDK is installed on your system, you will get the **JRE** installation dialog, which is presented in Figure 1-8 (right). Make sure that you accept the default installation location for this JRE; it should be installed in the **`\Java\jre8`** folder. It is best to allow Oracle (Java SDK) to put the software in an industry standard folder location, as other software packages that you will be using that use this JRE, such as NetBeans IDE 8.0, will be looking for it there first. Click the **Next** button to install the JRE.

The installation will show a progress bar during the install, as seen in Figure 1-9 (left). When it is finished, it will display the **Successfully Installed** dialog, shown in Figure 1-9 (right). If you want to access tutorials, API documentation, developer guides, version release notes, and so on, you can click the **Next Steps** button.

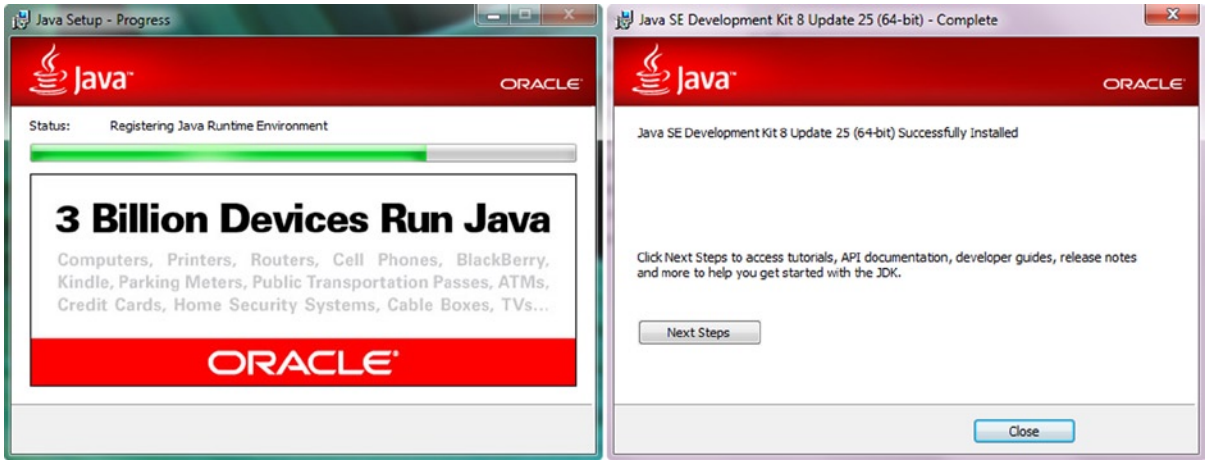

Figure 1-9. *During installation a progress bar shows you what is installing (left) and then gives you a completed dialog (right)*

Installing NetBeans IDE 8.0

Now, you are ready to install NetBeans, so locate your `netbeans-8.0-windows` file (see Figure 1-6. Right-click it, and select the **Run as administrator** option to launch the installer. Once it is launched, you will see the dialog shown in Figure 1-10, which gives you a **Customize** button that you can use to customize the install.

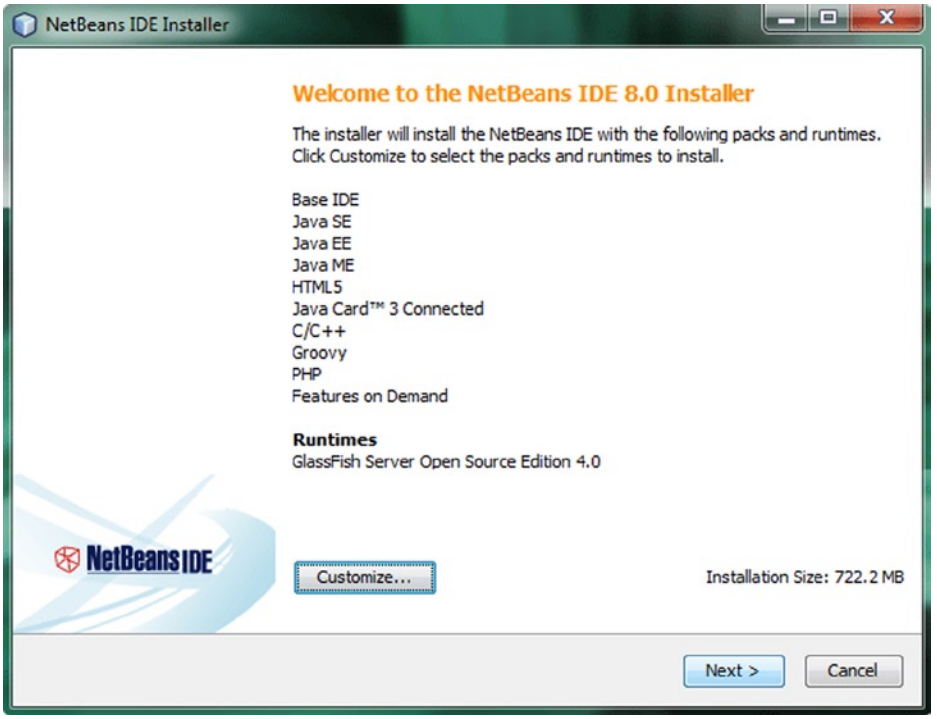

Figure 1-10. *The Welcome to the NetBeans IDE 8.0 Installer dialog*

Click the **Next** button to begin the default (full) installation, and you will get the **NetBeans IDE 8.0 License Agreement** dialog, shown in Figure 1-11 (left). Select the **I accept the terms in the license agreement** check box, and click the **Next** button to advance to the **JUnit License Agreement** dialog, shown in Figure 1-11 (right).

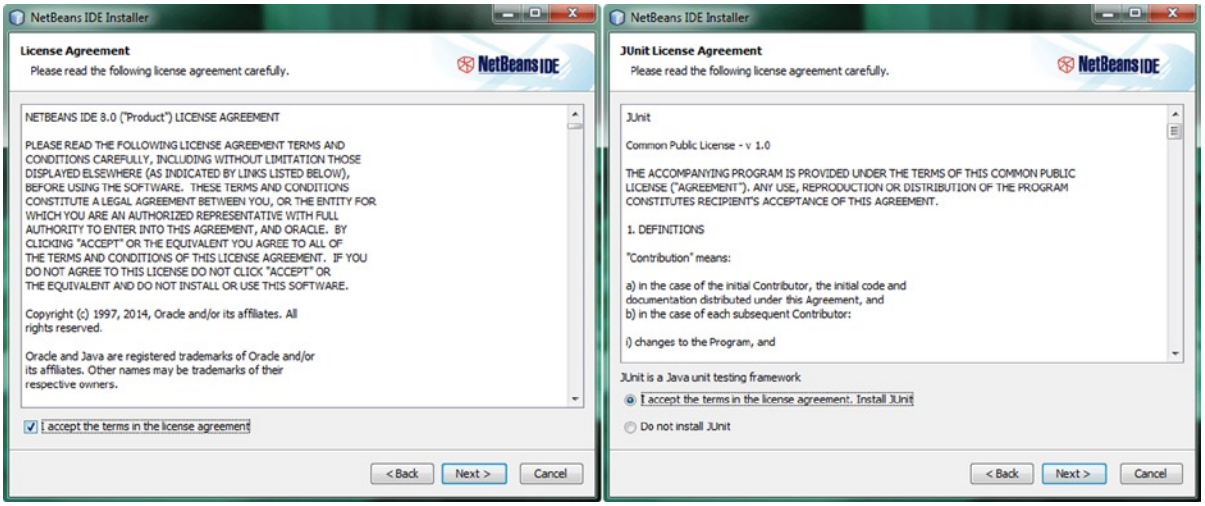

Figure 1-11. *Accept the terms of the license agreement, click the Next button (left), and then do the same for JUnit (right)*

In the JUnit License Agreement dialog, presented in Figure 1-11 (right), click the **radio button** next to the **I accept the terms in the license agreement** statement, and click the **Next** button to proceed with the installation. The next two installer dialogs, illustrated in Figure 1-12, will allow you to specify where NetBeans 8.0 and GlassFish 4.0 will be installed on your system. I suggest accepting the default installation locations in these two dialogs as well. As you will notice, the NetBeans installer has found your Java installation in its default location as well.

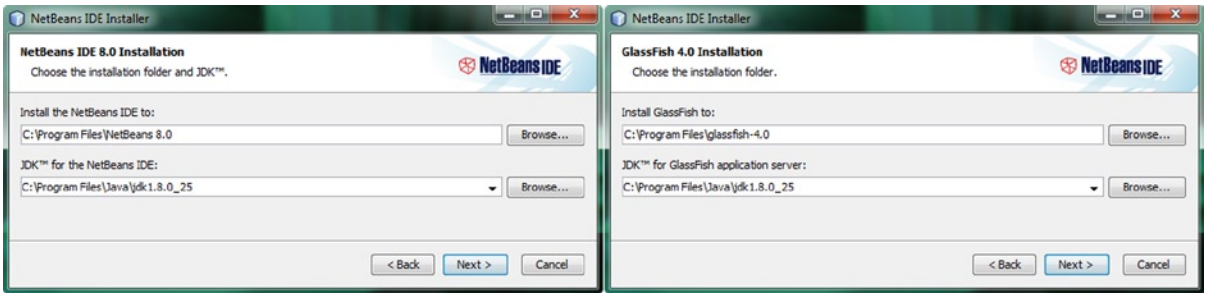

Figure 1-12. *Accept the default installation directory suggestions for NetBeans IDE (left) and GlassFish 4.0 (right)*

Once you accept these default installation locations and click the **Next** button to advance through these dialogs, you will get a **Summary** dialog, shown in Figure 1-13 (left). This dialog contains an **Install** button, which will trigger the installation that you have set up over the previous five NetBeans IDE 8.0 installation dialogs.

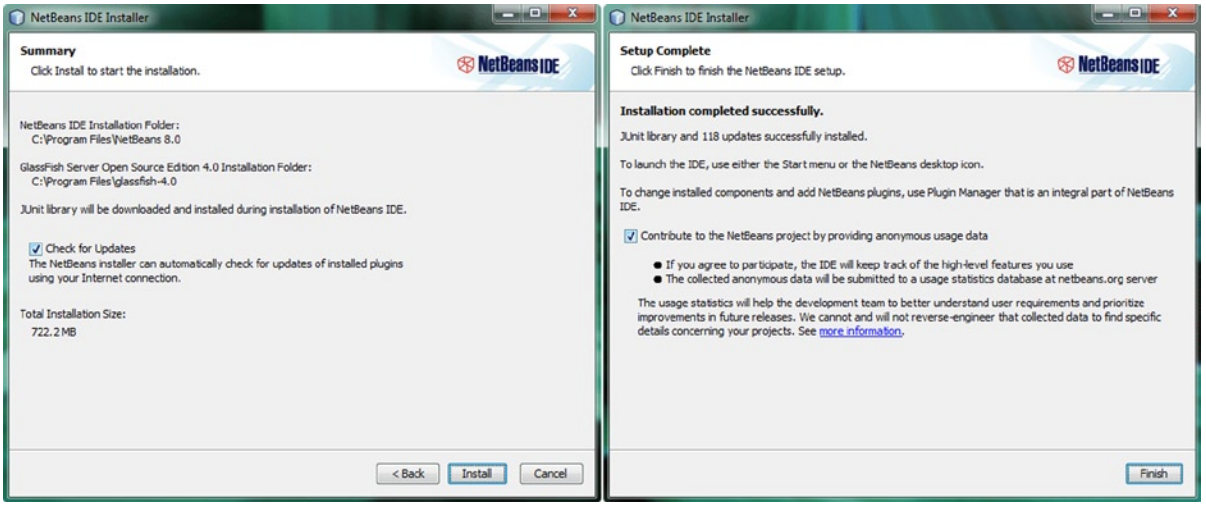

Figure 1-13. *Select the Check for Updates check box, and click the Install button (left) and the Finish button (right)*

During the installation, you will see the **Installation** dialog, and its progress bar, illustrated in Figure 1-14, which will tell you exactly what percentage of the installation has been completed as well as which IDE files are currently being extracted, and installed, on your workstation.

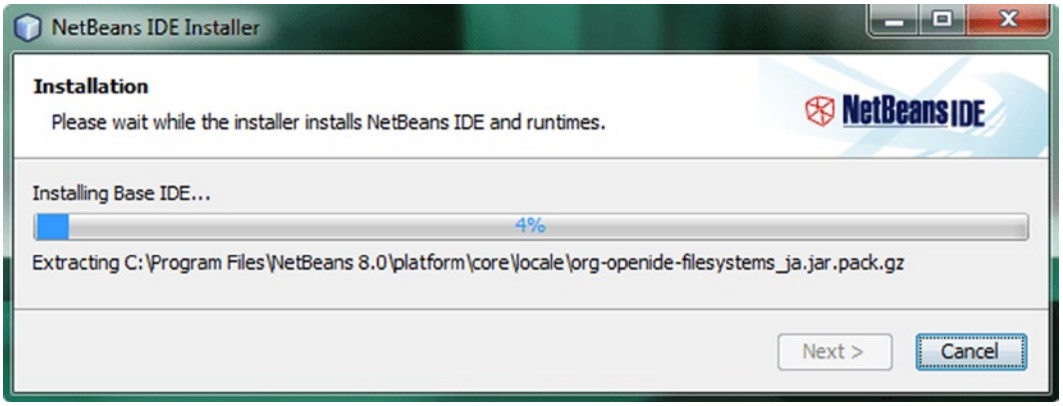

Figure 1-14. *The Installation progress dialog, showing the percentage of install complete*

When the installation process is complete, you will see the **Setup Complete** dialog, which is shown in Figure 1-13 (right). Now, you are ready to develop Java 8 and JavaFX applications (games) on your workstation.

Next, let's download five of the most popular free open-source new media content development software packages so that you have all the tools that you will need for a Java 8 games development business!

After that, you will take a look at some other impressive open-source software that I use on my workstation. That way, if you want to, you can put together the ultimate software development workstation before you have even finished this chapter, creating an incredibly valuable content production workstation for the cost of the hardware (and OS) alone!

Installing New Media Content Production Software

There are a number of "genres" of new media elements, or assets, as I call them, that are supported in JavaFX, which is the new media engine in Java 8 (and Java 7) and thus what you will be using as the foundation for your Java 8 game development. The primary genres of new media, for which you will be installing the leading open-source software in the remainder of this chapter, include digital illustration, digital imaging, digital audio, digital video, and 3D.

Downloading and Installing Inkscape

Because JavaFX supports 2D, or **vector,** technology, commonly used in **digital illustration** software packages, such as Adobe Illustrator and FreeHand, you will download and install the popular open-source digital illustration software package known as **Inkscape**.

Inkscape is available for the Linux, Windows, and Mac OSs, just like all the software packages that you are installing in this chapter, so you can use any platform you like to develop games!

To find the Inkscape software package on the Internet, go to Google Search, and type in **Inkscape**, as shown in Figure 1-15, at the top left. Click the **Download** link (or right-click, and open in a separate tab), and click the icon that represents the OS that you are using. The Penguin signifies Linux (far-left icon), the Window signifies Windows (center icon), and the stylized apple signifies Mac (far-right icon).

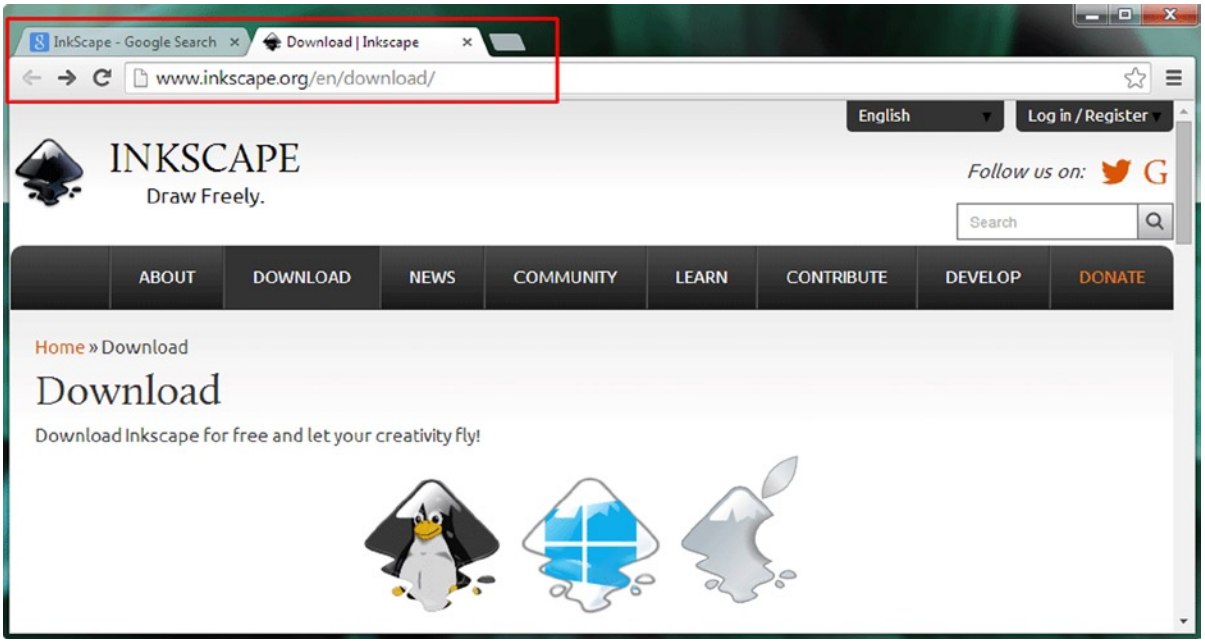

Figure 1-15. *Google the term "InkScape," go to the Inkscape Download page, and click the icon that matches your OS*

If you want to use the 64-bit Windows version of Inkscape, scroll down, and take a look at the text links below these three icons to access that particular OS download. Once you have downloaded the software, right-click it and Run as administrator, and install the software on your workstation. If you have a previous version of Inkscape, the installation will upgrade it to the latest version; you do not need to use the Programs and Features utility that you used earlier in the chapter to uninstall your SDK and IDEs, which do not upgrade previous versions, like new media production software packages tend to do.

After the software is installed, create a Quick Launch icon on your taskbar so that you can launch Inkscape with a single click of the mouse. Next, you will install a popular digital imaging software package, called **GIMP**, which will allow you to create "raster," or pixel-based (bitmap), artwork for your games in JPEG, PNG, or GIF digital image file formats supported by JavaFX. Raster images are different from vector, or shape, illustrations, so you will need GIMP.

Downloading and Installing GIMP

JavaFX also supports 2D images that use **raster** image technology, which represents images as an array of pixels and is commonly used in digital image compositing software packages, such as Adobe Photoshop and Corel Painter. In this section, you will download and install the popular open-source digital image editing and compositing software package called GIMP. This software is available for the Linux, Windows, Solaris, FreeBSD, and Mac OSs.

To find the GIMP software on the Internet, go to Google Search, and type in **GIMP**, as demonstrated in Figure 1-16.

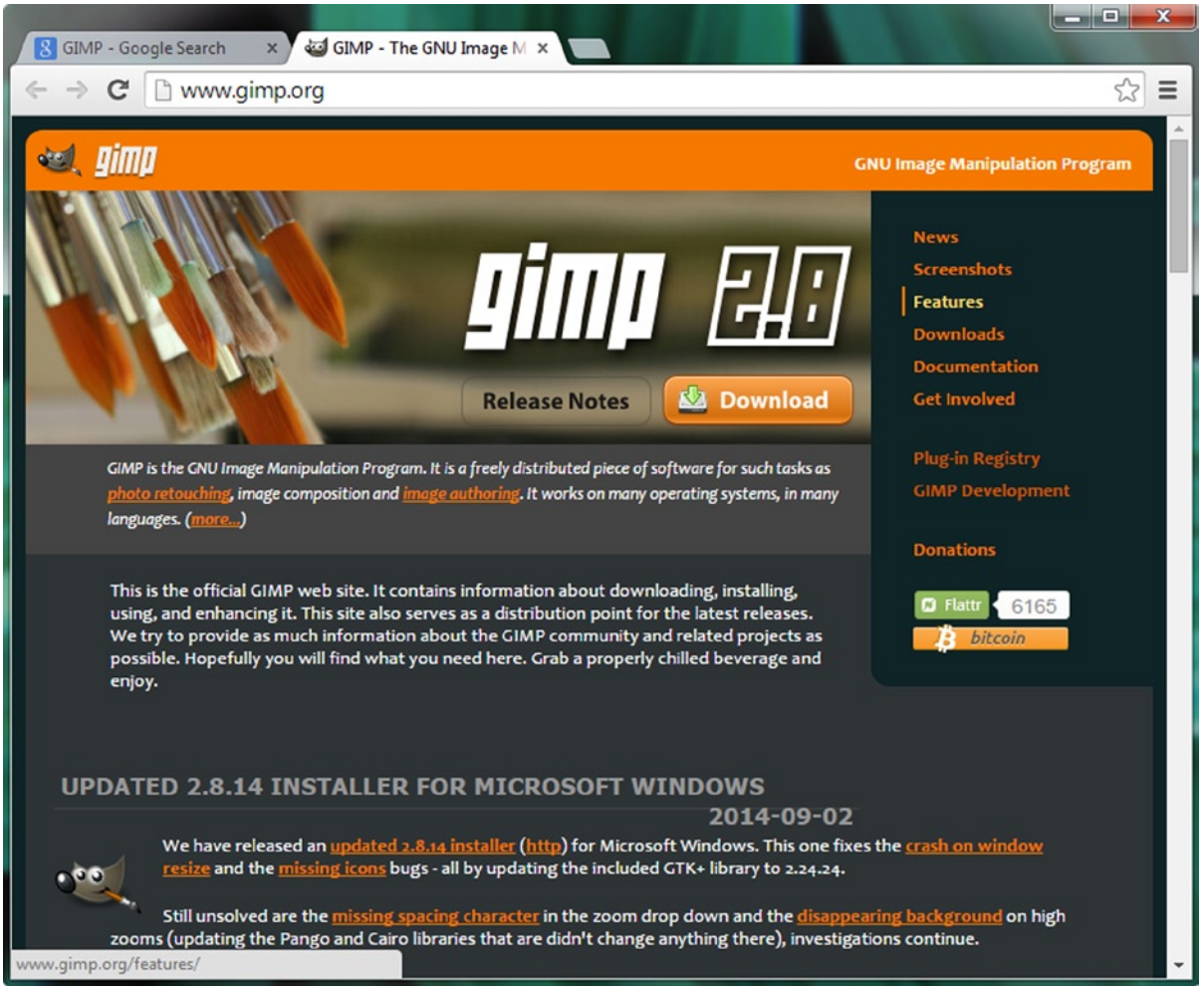

Figure 1-16. *Google the term "GIMP," go to the GIMP Downloads page, and click the Download GIMP link*

Click the **Download** link (or right-click, and open it in a separate tab), and click **Download GIMP 2.8.14** (or the latest version that represents the OS that you are using). The **Downloads** page will automatically detect the OS that you are using and give you the correct OS version; in my case, it is Windows. Download and install the latest version of GIMP, and then create a Quick Launch icon for your workstation taskbar, as you did for Inkscape. Next, you will install a powerful digital audio editing and audio effects software package, called **Audacity**.

Downloading and Installing Audacity

JavaFX supports digital audio sequencing, which uses digital audio technology. Digital audio represents analog audio by taking digital audio **samples**. Digital audio content is commonly created using digital audio composition and sequencer software packages, such as Propellerhead Reason and Cakewalk Sonar. In this section, you will download and install the popular open source digital audio editing and optimization software package known as **Audacity**. Audacity is available for the Linux, Windows, and Mac OSs, so you can use any OS platform that you like to create and optimize digital audio for your Java 8– and JavaFX–based games.

To find the Audacity software package on the Internet, use the Google search engine, and type in **Audacity**, as shown in Figure 1-17, at the top left. Click the **Download** link (or right-click, and open in a separate tab), and click **Audacity for Windows** (or the version that represents the OS that you are using).

Figure 1-17. *Google the term "Audacity," go to the Audacity Download page, and click a link matching your OS*

Download and install the latest version of Audacity (currently, it is 2.0.6), and then create a Quick Launch Icon for your workstation taskbar, as you did for Inkscape and GIMP. Next, you will install a powerful digital video editing and special effects software package, called **EditShare Lightworks**.

Downloading and Installing EditShare Lightworks

JavaFX also supports digital video, which uses **raster** pixel-based motion video technology. Raster represents video as a sequence of **frames**, each of which contains a digital image based on an array of pixels. Digital video assets are usually created using digital video editing and special effects software packages, such as Adobe After Effects and Sony Vegas. In this section, you will download and install open-source digital video editing software known as **Lightworks**.

EditShare's Lightworks used to be a paid software package until it was made open source. You will have to **register** on the Lightworks web site to download and use the software. This package is available for Linux, Windows, and Mac OSs. To find Lightworks on the Internet, go to Google Search, and type in **Lightworks**, as shown in Figure 1-18, at the top left. Click the **Download** link (or right-click, and open in a separate tab), and click the appropriate **Download** button and the tab that represents the OS that you are using. The **Downloads** page will automatically detect the OS that you are using and select the correct OS tab; in my case, Windows.

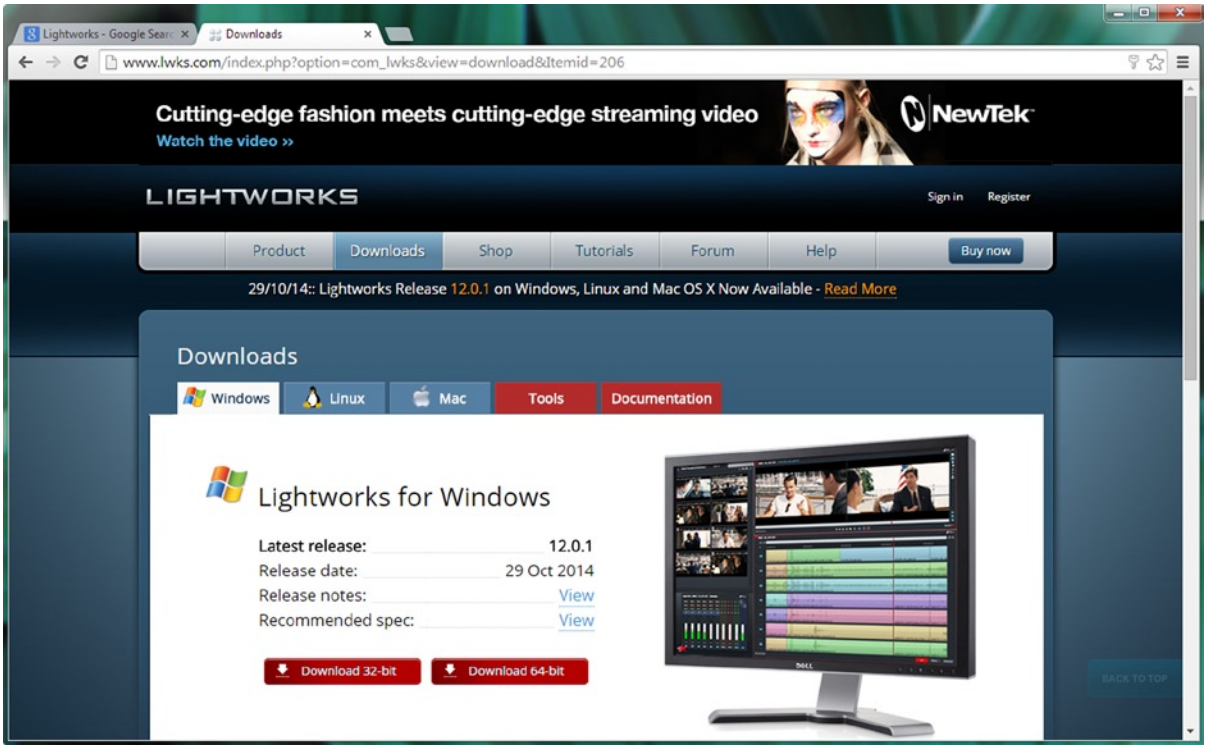

Figure 1-18. *Google the term "Lightworks," go to the Lightworks Downloads page, and click the tab that matches your OS*

Register on the Lightworks web site, if you have not done so already. Once you are approved, you can then download and install the latest version of Lightworks. Install the software, and create a Quick Launch icon for your taskbar, as you did for the other software. Next, you will install a 3D modeling and animation package, called **Blender**.

Downloading and Installing Blender

JavaFX has recently moved to support 3D new media assets that are created outside the JavaFX environment, which means that you will be able to create 3D models, textures, and animation, using third-party software packages, such as Autodesk 3D Studio Max or Maya and NewTek Lightwave 3D. In this section, you will download and install the popular open-source 3D modeling and animation software package known as Blender. Blender is available for the Linux, Windows, and Mac OSs, so you can use any OS platform that you like to create and optimize 3D models, 3D texture mapping, and 3D animation for use in your Java 8 and JavaFX games.

To find the Blender software on the Internet, using the Google search engine, type in **Blender**, as shown in Figure 1-19. Click the correct download link to download and install Blender, and then create the Quick Launch icon.

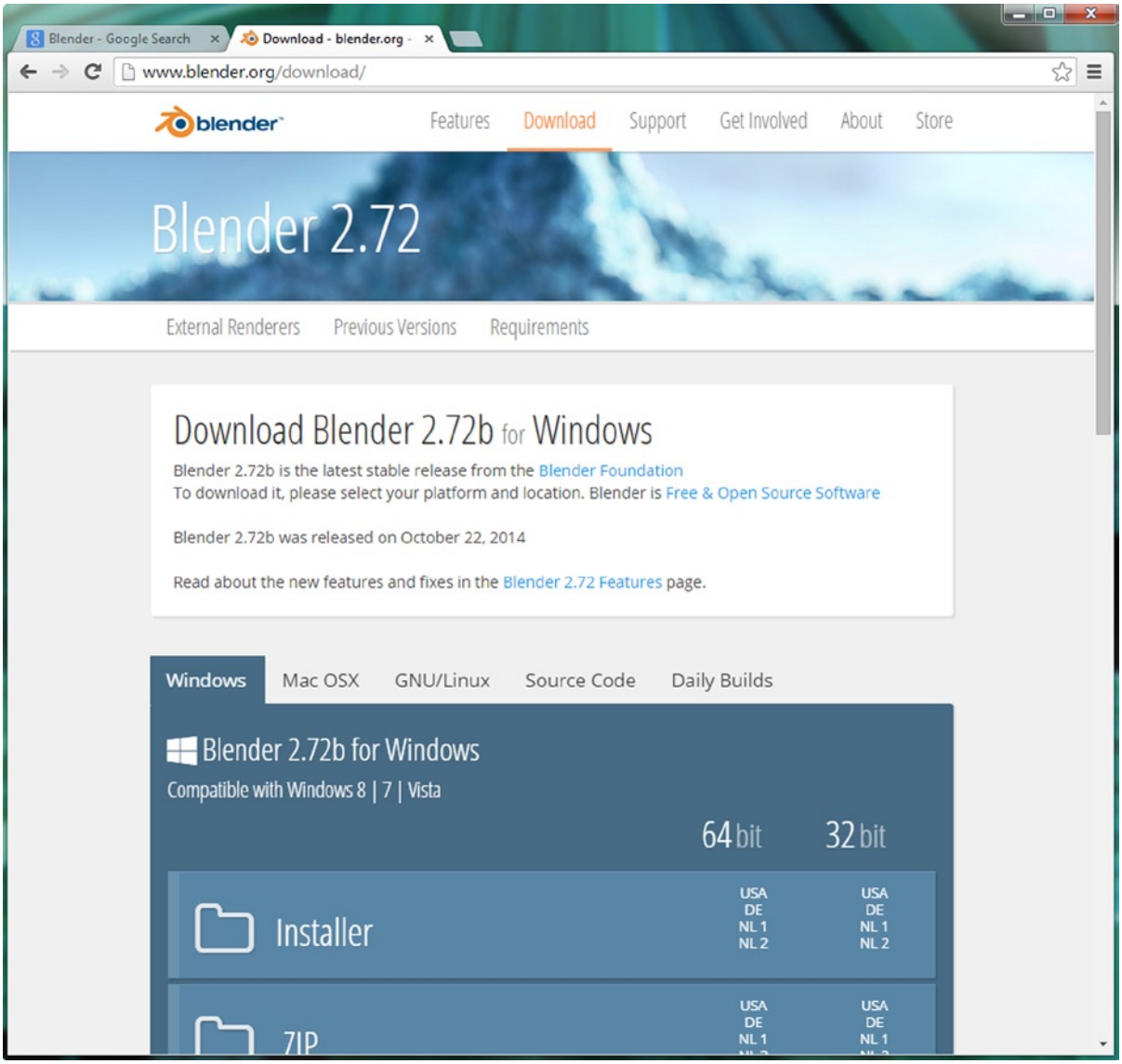

Figure 1-19. *Google the term "Blender," go to the Blender Download page, and click the tab for your OS*

Other Open-Source Software Packages of Interest

There are a number of other professional-level open-source software packages that I use in my new media content production business that I thought I would let you know about, in case you had not heard about them. These will add even more power and versatility to the new media production workstation that you have built up to this point. It is important to note that you have already saved yourself thousands of dollars that would have otherwise been spent on similar paid content production software packages in the process of doing all this extensive downloading and installing. I guess you could say my motto is, "Do it right the first time, and be sure to go all the way," so let me tell you about some of the other free, and even some of the more affordable, new media content production software packages that I have installed on my own content production workstations.

One of the best values in open-source software, aside from the EditShare Lightworks package, which used to cost six figures back in the day, is a **business productivity software suite** that was made open source by **Oracle** after it acquired Sun Microsystems. Oracle transferred its OpenOffice software suite over to the popular Apache open-source project. **OpenOffice 4.1** is an entire office productivity software suite that contains **six** full-fledged business productivity software packages! Because your content production agency is actually a full-fledged business concern, you should probably know about this software, as it is an exceptionally solid open-source software offering. You can find it at **www.openoffice.org**; this popular business software package has been downloaded by savvy professionals such as yourself more than a hundred million times, so it's no joke, as they say!

A great complement to the Audacity digital audio editing software is **Rosegarden** MIDI sequencing and music composition and scoring software, which can be used for music composition and printing out the resulting scores for music publishing. Rosegarden, currently in version 14.02, and being ported from Linux to Windows, can be found via Google Search or at **www.rosegardenmusic.com**.

Another impressive audio, MIDI, and sound design software package is **Qtractor** If you are running the Linux OS, be sure to download and install this professional-level digital audio synthesis software package by doing a Google search or going to **https://Qtractor.SourceForge.net**.

For 3D character modeling and animation, be sure to check out the 3D software packages from **DAZ Studio** (**www.daz3d.com**) when you have the chance. The current version of **DAZ Studio Pro** is 4.6, and yes, it is free! You have to log in and sign up, like you did for EditShare Lightworks, but that is a small price to pay! There is also a free 3D modeling software package on this web site, called **Hexagon 2.5**, and a popular terrain generation software package for less than 20 dollars, called **Bryce 7.1 Pro**. The most expensive software on the DAZ Studio web site is **Carrara** (150 dollars) and **Carrara Pro** (285 dollars). DAZ Studio makes most of its revenue selling **character models** of one type or another, so take a look, as it is a force to be reckoned with in the 3D content (virtual) world!

Another impressive (and free, for the basic version) world generation software package is **Terragen 3.2**, from **Planetside** Software, in the United Kingdom. You can download the basic version from **https://planetside.co.uk** as well as join its forum. I have used this software in a couple of my Android application development books, so I know it works well for multimedia applications and games. It is also used by professional filmmakers, as the level of quality is pristine.

Caligari TrueSpace 7.61 is also excellent, free 3D modeling and animation software. The program, which is "free and alive!" according to the Caligari web site (**https://Caligari.us**), from which you can still download it, used to cost nearly a thousand dollars when it was first developed by Roman Ormandy, the founder of the Caligari Corporation (later purchased by Microsoft). A professional-level 3D modeling and animation software package, this program had millions of users in its heyday. It is a really cool piece of software, with a fun-to-use user interface (UI), so be sure to grab it!

Another 3D rendering software you should take a look at is **POV-Ray** (Persistence of Vision Raytracer), which works with any 3D modeling and animation software package to generate impressive 3D scenes, using advanced ray-traced rendering algorithms. The most recent version on the POV-Ray web site (**www.povray.org**), **3.7**, is **64 bit** and multicore (multithreaded) compatible, and it can be downloaded for free!

Bishop3D is a cool 3D modeling software package that was specifically designed for use with POV-Ray. The software can be used to create custom 3D objects, which can then be imported into POV-Ray (and then into JavaFX) for use in your games. The most recent version, an 8MB download, is **1.0.5.2**, for Windows 7. The software can be found at **www.bishop3d.com** and can currently be downloaded for free!

Yet another free 3D modeling software worth investigating is **Wings 3D**. This software can be used to create custom 3D objects, which can then be imported into JavaFX for use in your games. The most recent version, a **64-bit**, 16MB download, is **1.5.3** and was released in April 2014, for Windows 7, Mac OS X, and Ubuntu Linux. The software can be found at **www.wings3d.com** and can currently be downloaded for free!

For UI design prototyping, the free software package **Pencil 2.0.6**, from **Evolus**, allows you to easily prototype UI designs before you create them in Java, Android, or HTML5. The software is located at **http://pencil.evolus.vn** and is available for Linux, Windows, and Mac OSs.

Next, you will take a look at how I organize some of the basic OS utilities and open-source software on my taskbar.

Organizing Quick Launch Icons in Your Taskbar Area

There are certain OS utilities, such as the calculator, text editor (Notepad), and file manager (Explorer), for which I create Quick Launch icons on my taskbar, as these utilities are used frequently in programming and new media content development work processes. I also keep as Quick Launch icons a wide range of new media development, programming, and office productivity applications. Figure 1-20 displays a dozen of these, including everything that you just installed, in the order in which you installed it, as well as a few others, such as OpenOffice 4.1, DAZ Studio Pro 4.6, and Bryce 7.1 Pro.

Figure 1-20. *Make taskbar Quick Launch icons for key system utilities, NetBeans 8.0, and new media production software*

There are a couple of ways to create these Quick Launch icons: you can drag programs from the start menu and drop them onto the taskbar, or you can right-click icons on the desktop or in the Explorer file manager and select **Pin this program to taskbar** from the context-sensitive menu. Once icons are on the taskbar, you can change their position simply by dragging them to the left or to the right.

Congratulations, you have just set up a new media Java 8 game development workstation that is highly optimized and that will allow you to create any new media Java 8 game that you or your clients can imagine!

Summary

In this first chapter I made sure that you have everything that you need to develop standout Java 8 games, including the latest versions of Java 8, JavaFX, and NetBeans 8.0 as well as all the latest open-source new media software.

You started by downloading and installing the latest Java JDK 8 and NetBeans IDE 8.0 software. Then, you did the same for a plethora of professional open-source new media tools.

In the next chapter, I will show you how to use NetBeans 8.0 to create a Java 8 project.

CHAPTER 2

Setting Up Your Java 8 IDE: An Introduction to NetBeans 8.0

Let's get started here in Chapter 2 by considering **NetBeans IDE 8.0**, because that is the primary piece of software that you will be using to create your Java 8 games. Even though Java JDK 8 is the foundation for your Java 8 games, as well as for NetBeans 8.0, you will start your journey by learning about NetBeans, as it is the "front end," the window through which you look at your Java game project.

NetBeans 8.0 is the **official IDE** for Java JDK 8, and, as such, it is what you will be using for this book. That is not to say you cannot use another IDE, such as Eclipse or IntelliJ, which are the official IDEs for Android 4.x (32 bit) and Android 5.x (64 bit) respectively, but I prefer to use NetBeans 8.0 for my new media application and game development for the Java 8, JavaFX 8, HTML5, CSS3 (Cascading Style Sheets 3), and JavaScript software development markup and programming paradigms.

This is not only because NetBeans 8.0 integrates **JavaFX Scene Builder**, which you will be learning about in Chapter 5 of this book, but also because it is an HTML5 IDE, too, and I create everything I design for my clients using Java 8, JavaFX 8, Android 4.x, or Android 5.x as well as HTML5. I do this so that the content works across (on) closed, or proprietary, OSs and platforms. I prefer open-source software and platforms, as you observed in Chapter 1.

First, you will take a look at what is new in NetBeans 8.0. This version of NetBeans was released at the same time as Java 8, and the version number synchronization is no coincidence. You will discover why you will want to use NetBeans 8.0 rather than an older NetBeans version, such as NetBeans 7.4 or earlier.

Next, you will examine the various attributes of NetBeans IDE 8.0 that make it an invaluable tool for Java 8 game development. You will not be able to get hands-on experience with all its features in the chapter, but you will be exploring all the cool things that it can do for you over the course of this book (you will need to put an advanced code base into place to really give some of the features a workout).

Finally, you will learn how to create your Java 8 and JavaFX project, using NetBeans 8.0 so that you progress toward creating the Java 8 game that you will be developing as you read through this book.

Primary Attributes of NetBeans 8.0: An Intelligent IDE

Assuming that you already have a professional-level workstation in place for new media content and game development, you need to remove all the outdated JDKs and IDEs and make sure that you have the latest V8 Java and NetBeans software installed on your system and ready to go. If you are new to this and do not have a game-appropriate workstation, go to Walmart or PriceWatch.com, and purchase an affordable multicore (use a 4-, 6- or 8-core) 64-bit computer running Windows 8.1 (or 9.0 if it is available) that has 4GB, 6GB, or 8GB of DDR3 (1333 or 1600 memory access speed) system memory at the very least and a 750GB, or even 1TB, hard disk drive.

NetBeans 8.0 Is Smart: Put Your Code Editing into Hyperdrive

Although it is true that an IDE is like a word processor, only geared toward writing code text rather than creating business documents, a programming integrated development environment such as NetBeans does a lot more for your programming work process than a word processor does for your document-authoring work process.

For instance, your word processor does not make suggestions in real time regarding the content that you are writing for your business, whereas the NetBeans IDE will actually look at what you are coding while you are writing that code and will help you write your code statements and constructs.

One of the things that NetBeans will do is finish lines of code for you as well as apply color to the code statements to highlight different types of constructs (classes, methods, variables, constants, references, and the like) (for more details, see Chapter 3). NetBeans will also apply the industry standard for **code indenting** to make your code much easier to read (for both yourself and the other members of your game application development team).

In addition, NetBeans will provide **matching** code structure **brackets**, **colons**, and **semicolons** so that you do not get lost when you are creating complex, deeply nested, or exceptionally dense programming constructs. You will be creating constructs such as these as I take you from Java 8 game beginner to Java 8 game developer, and I will point out Java 8 code that is dense, complex, or deeply nested as you encounter it.

NetBeans can also provide bootstrap code, such as the JavaFX game application bootstrap code that you will be creating a bit later in this chapter (see the section "Creating Your Java 8 Project: The InvinciBagel"), as well as code templates (which you can fill out and customize), coding tips and tricks, and code refactoring tools. As your Java code becomes more complex, it also becomes a better candidate for code refactoring, which can make the code easier to understand, easier to upgrade, and more efficient. NetBeans can also refactor your code automatically.

In case you are wondering, **code refactoring** is changing the structure of existing computer code to make it more efficient or scalable without changing its external behavior, that is, what it accomplishes. For instance, you could take Java 6 or Java 7 code and make it more efficient by implementing Lambda Expressions, using Java 8.

Furthermore, NetBeans offers pop-up helper dialogs of various types, containing **methods**, **constants**, asset **references** (see Chapter 3), and even **suggestions** regarding how to construct the code statement, for example, when it might be appropriate to use the powerful new Java 8 **Lambda Expressions** feature to make your code more streamlined and multithread compatible.

NetBeans 8.0 Is Extensible: Code Editing with Many Languages

Another thing that your word processor cannot do is allow you to add features to it, which NetBeans can do using its **plug-in** architecture. The term that describes this type of architecture is **extensible**, which means that if needed, it can be extended to include additional features. So, if you wanted to extend NetBeans 8.0 to allow you to program using Python, for instance, you could. NetBeans 8.0 can also support older languages, such as COBOL and BASIC, in this fashion as well, although with the majority of popular consumer electronics devices using Java, XML, JavaScript, and HTML5 these days, I am not really sure why anyone would want to take the time do this. I did a Google search for this, however, and there are people coding in Python and COBOL in NetBeans 8.0, so there is real-world proof that the IDE is indeed extensible.

Probably because of its extensibility, NetBeans IDE 8.0 supports a number of popular programming languages, including **C**, **C++**, **Java SE**, **JavaScript**, **XML**, **HTML5**, and **CSS** on the **client side** and **PHP**, **Groovy**, **Java EE**, and **JavaServer Pages** (**JSP**) on the **server side**. Client-side software is run on the device that the end user is holding or using (in the case of an iTV); server-side software runs **remotely,** on a server, and talks to the end user over the Internet or a similar network while the software is running on the server. Client-side software is more efficient, as it is **local** to the hardware device that it is running on and thus is more **scalable**: no server is involved to experience overload as more and more people use the software at any given point in time.

NetBeans 8.0 Is Efficient: Organized Project Management Tools

A good programming IDE needs to be able to manage projects that can grow to become quite massive, involving more than a million lines of code contained in hundreds of folders in the project folder hierarchy and thousands of files and new media assets. For this reason, project management features must be extremely robust in any mainstream IDE. NetBeans 8.0 contains a plethora of project management features that allow you to look at your Java 8 game development project, and its corresponding files and their interrelationships, in a number of different ways.

There are four primary project management views, or "panes" that you can use to see the different types of **interrelationships** in your project. (I call them panes, as the entire IDE is in what I call a window). I jumped ahead (to the end of the chapter, once your Java 8 game project has been created) and created the screenshot presented in Figure 2-1. This screenshot displays the four project management panes opened in this new project so that you can see exactly what they will show you.

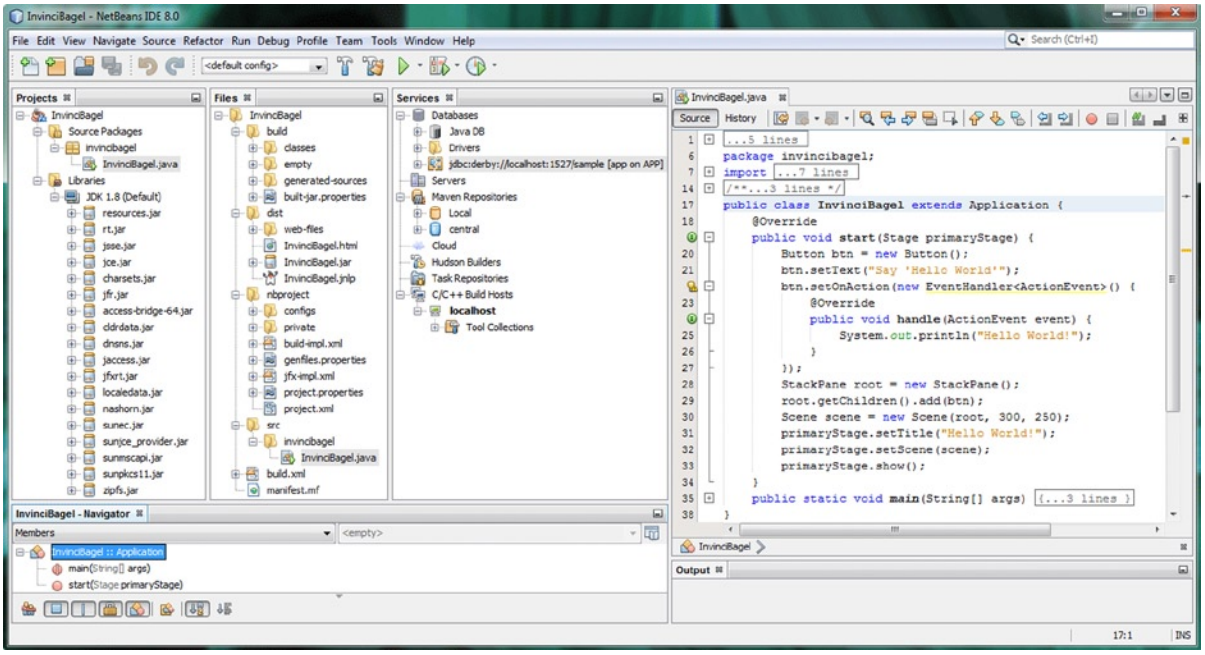

Figure 2-1. *Project management panes, at the left of the IDE, include Projects, Files, Services, and Navigator*

The **Projects** pane, at the left of the screen, shows the Java **Source Packages** and **Libraries** that make up your (game) project. The next pane over is the **Files** pane, which has the **project folder** and **file hierarchy** on your hard disk drive. The **Services** pane contains the databases, servers, repositories, and build hosts, if they are used in the project (these are primarily server-side technologies, and technologies used with a development team, so I am not going to go into these in detail).

The Projects pane should always be left open (as you will see in Figures 2-7 through 2-21). The Projects pane provides you with a primary access point for all the project source code and assets (content) in your Java 8 game project. The Files pane shows not only the project folder and file hierarchy, but also the data and FXML markup (JavaFX) or Java 8 code hierarchy inside each file.

The **Navigator** pane (bottom) shows the **relationships** that exist inside your Java code structures. In this case, these are the `InvinciBagel` class, the `.start()` method, and the `.main()` method (for further information, see Chapter 3).

NetBeans 8.0 Is User Interface Design Friendly: UI Design Tools

NetBeans 8.0 also has **Design a GUI** drag-and-drop design tools for a plethora of platforms, including Java SE, Java EE, Java ME, JavaFX, and Java Swing as well as C, C++, PHP, HTML5, and CSS3. NetBeans provides visual editors that write the application's **UI** code for you, so all you have to do is make the visual on the screen look like what you want it to look like in your game application. Because games use the JavaFX new media (game) engine, you will be learning about the **JavaFX Scene Builder**, an advanced FXML-based visual design editor, in Chapter 5 of this book.

JavaFX has the Prism game engine as well as 3D (using OpenGL ES [**OpenGL for** Embedded Systems]) support, so I will be focusing quite a bit on the JavaFX Scene Graph and JavaFX APIs. The assumption here is that you will want to build the most advanced Java 8 games possible, and leveraging the JavaFX engine, which is now a part of Java 8 (along with Lambda Expressions), is going to be the way to accomplish this. The fastest way to develop a game is to leverage advanced code and programming constructs that the Java 8 and JavaFX environments generously give you for your use in creating cutting-edge applications (in this case, games) that contain powerful new media elements.

NetBeans 8.0 Is not Bug Friendly: Squash Bugs with a Debugger

There is an assumption across all computer programming languages that the negative impact to your programming project of a "bug," or code that does not do exactly what you want it to, increases in magnitude the longer it remains unfixed, so bugs must be squashed as soon as they are "born." NetBeans bug-finding **code analysis** tools, and integrated **NetBeans Debugger**, and integration with the third-party **FindBugs** project, which, as you now know from experience (Audacity), can be found on the SourceForge web site (**`http://findbugs.sourceforge.net`**) (if you want the stand-alone version), all supplement the real-time, "as you type" code-correcting and efficiency tools I discussed earlier (see the section "NetBeans 8.0 Is Smart: Put Your Code Editing into Hyperdrive").

Your Java code will not be very complicated until a bit later in the book, so I will cover how these tools work when you need to use them, once your knowledge base is a bit more advanced.

NetBeans 8.0 Is a Speed Freak: Optimize the Code with a Profiler

NetBeans also has something called a **Profiler**, which looks at your Java 8 code while it is running and then tells you how **efficiently** it uses **memory** and **CPU** cycles. This allows you to refine your code and make it more efficient in its use of key system resources, which is quite important for Java 8 game development, as this will affect the smoothness of game play on systems that are not as powerful (e.g., on single- and dual-core CPUs).

This Profiler is a **dynamic** software analysis tool, as it looks at your Java code **while it is running**, whereas the FindBugs code analysis tool is a **static** software analysis tool, as it simply looks at your code **in the editor**, when it is not compiled and running in system memory. The NetBeans Debugger will allow you to **step** through your code while it is running, so that tool can be viewed as a **hybrid** that ranges from a static (editing) to a dynamic (executing) code analysis mode.

After you create the foundation for your Java 8 (JavaFX) game engine (in the following sections), you will run the Profiler to see how it works inside NetBeans IDE 8.0. I am going to present as many key features of NetBeans as possible up front so that you get comfortable with this software.

Creating Your Java 8 Game Project: The InvinciBagel

Let's get down to business and create the foundation for your game. I am going to demonstrate how to create an original game so that you can see the process involved in developing a game that does not yet exist, as opposed to most game programming books, which replicate games that are already on the market. I got permission from my client **Ira Harrison-Rubin,** cartoonist/author/humorist for the **BagelToons** franchise, to let readers to see the process of creating his **InvinciBagel** cartoon game during the course of this book.

Click the Quick Launch icon on your taskbar (or double-click the icon on your desktop) to launch NetBeans 8.0, and you will see the NetBeans start-up screen, illustrated in Figure 2-2. This screen contains a progress bar (in red) and will tell you what is being done to configure the NetBeans IDE for use. This involves loading the various components of the IDE into your computer system memory so that they can be used smoothly and in real time during development.

Figure 2-2. *Launch NetBeans 8.0, using the Quick Launch icon*

After NetBeans IDE 8.0 has been loaded into your system memory, the NetBeans 8.0 **start page** will be displayed on your screen, as shown in Figure 2-3. Click the "**x**" at the right of the Start Page tab to close this page.

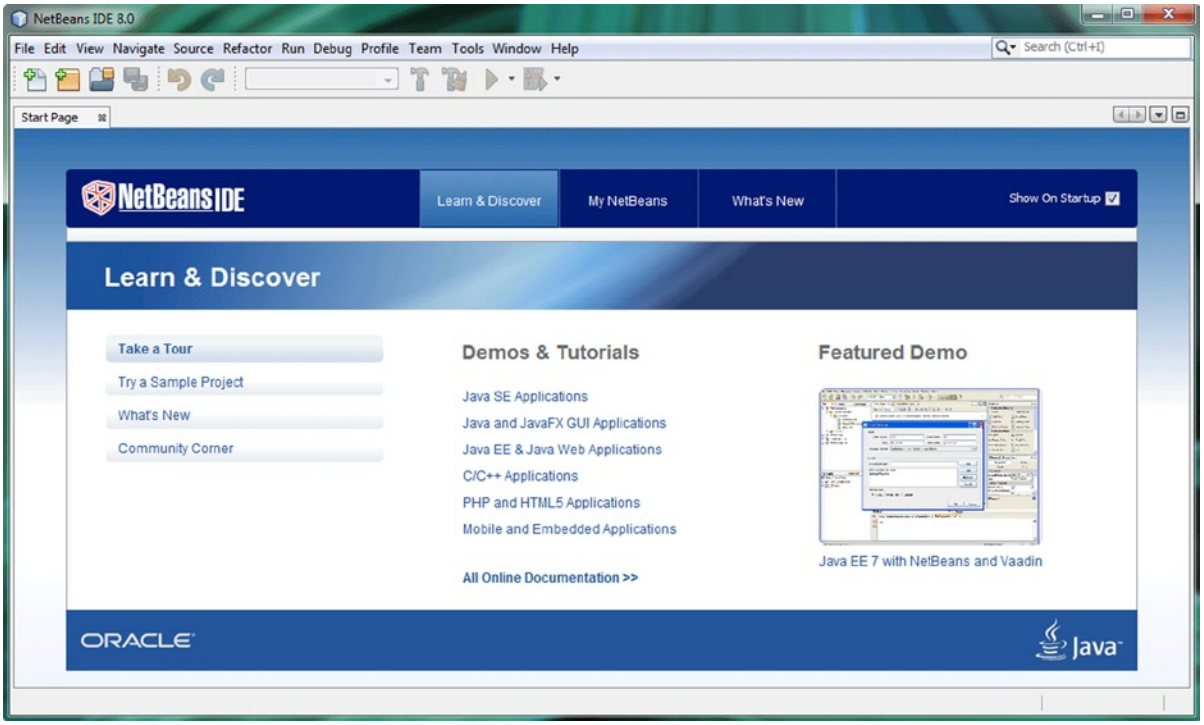

Figure 2-3. *Close the Start Page tab, at the top left of the screen, by clicking the "x" at the right of the tab to reveal NetBeans IDE 8.0*

This will display what I term the virgin IDE, with no projects active. Enjoy this now, as soon you will be filling this IDE with panes for your project components (you can see part of this empty IDE in Figure 2-4, which contains menus and shortcut icons and not much else).

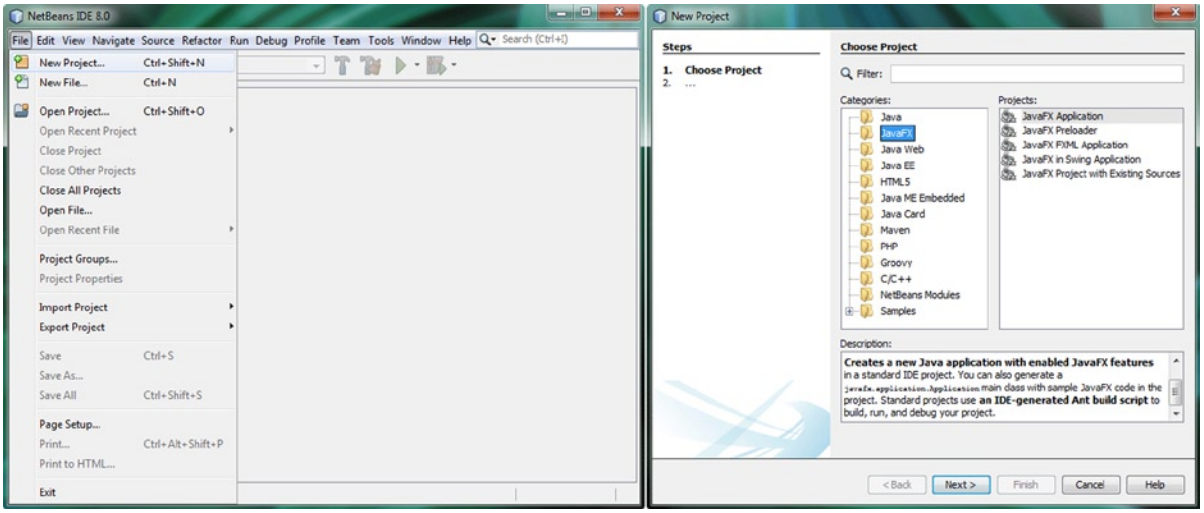

Figure 2-4. *Showing virgin NetBeans 8.0 IDE (left) and a JavaFX New Project dialog (right)*