Chapter 1

Installation and Creating a Robot

Set your stopwatch! Five minutes from now, the robot playground, called the environment, that you will be using in this book will be up and running and ready for you to have fun in. In this chapter you will learn how to install the environment, become acquainted with its different parts, and begin interacting with the robots that live in this environment. You will learn how to program these robots to accomplish challenging tasks by sending them messages. So let us get started installing the environment and preparing for all the challenges ahead in the rest of the book. If your environment is already installed, then turn off your stopwatch, skip the first section, and plunge directly into the following sections, which give an overview of the environment. After you have acquired some facility with robots in Chapters **??**2 through 4**??**, I will go into more detail on using the environment in Chapter 5 **??**.

1.1 Installing the Environment

The environment used in this book has been developed to run on top of Squeak. Squeak is a rich and powerful Open Source multimedia environment written entirely in Smalltalk and freely available for most computer operating systems at http://www.squeak.org. Note, however, that you will not be using the default Squeak distribution. Rather, you will be using a distribution that I have prepared for use with this book. It can be downloaded from the publisher of this book at http://www.apress.com and Stéf ►*fix*◄ , in the Downloads section.

Squeak runs exactly the same on all platforms. However, to make your life a little easier, I have prepared several platform-dependent compressed files. The principle is exactly the same on a Mac, PC, or any other platform. The only differences are in the tools for file decompression and the way that you will invoke Squeak. Once you have obtained a file named ReadyToUse.zip, you decompress it and then drag the file named Ready.image (Mac) or Ready (PC) onto the Squeak application, and that does it! The file Ready[.image] contains the complete environment used in this book. Note that you may get files with slightly different names, but that should have no effect on how everything works.

Installation on a Macintosh

For installation on a Macintosh, you should have a ZIP archive file named readyToUse.zip. Normally, double clicking on the file's icon should invoke the proper decompression software, such as StuffIt Expander. Once this archive has been decompressed, you should end up with four files, as shown in Figure 1.1. You should identify two files: the file named Ready .image and the *Squeak application* file (the one without a file extension in Figure 1.1; it is named Squeak).

Installation under Windows

For installation under Windows, you should have a ZIP archive file named readyToUse.zip. Once this archive has been decompressed using WinZip, you should end up with four files, as shown in Figure 1.2. You should identify two files: the file named Ready and the *Squeak application* file (the one without a file extension in Figure 1.2; it is named Squeak).

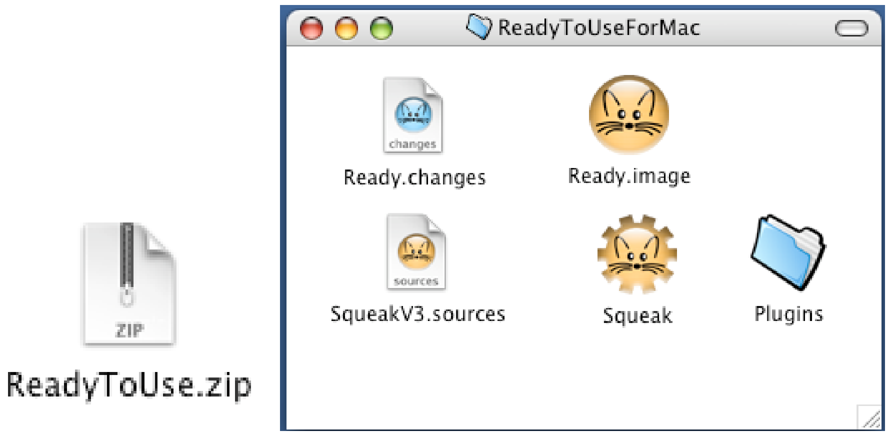


Figure 1.1: Ready-to-use files for the Macintosh. Left:the ZIP archive. Right:the decompressed files.

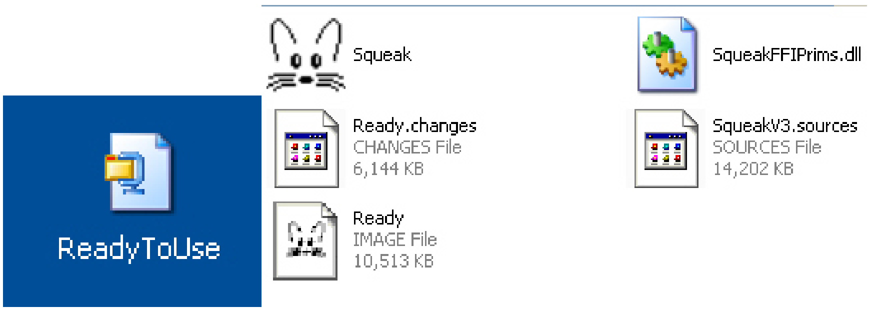


Figure 1.2: Ready-to-use files for Windows. Left:the ZIP archive. Right:the decompressed files.

1.2 Opening the Environment

To open the environment, drag the file Ready[.image]onto the Squeak application,that is, onto the file named Squeak, as shown in Figure 1.3. You should obtain the environment shown in Figure 1.4. If you do not get this environment, then read the section “Installation Trouble- shooting” near the end of this chapter.

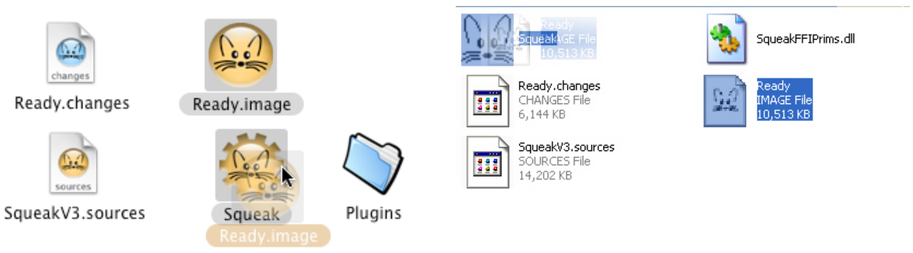


Figure 1.3: Dragging and dropping the *image* file onto the *Squeak application file* opens the environment on a Mac (left) or on PC (right).

Tips for Installation.

The environment can be opened simply by double clicking on the image file. However, there are several disadvantages to this: You may have to identify the Squeak application,and sometimes another application may interfere and try to use the image file. Moreover, you can find yourself in trouble if you have multiple installations of different versions of Squeak. So I suggest that you always open the environment by dragging and dropping the image file onto the Squeak application file or an alias of it.

Note that if you do not have enough space for the installation on your hard drive, you can use an alias to the SqueakV3.sources file, which can be shared among several installations.

Important! To start the environment, drag and drop the file Ready (with the .image extension for Mac) onto the squeak application.

1.3 First Interactions with a Robot

Once you have opened the environment by dragging the file named Ready[.image] onto the Squeak application as explained previously, the environment that you obtain should look something like the one presented in Figure 1.4.

The environment is composed of a robot factory and two flaps. A flap is a drawer contain- ing programming tools. You will not need these for a while, and so I will put off describing them until a later chapter. You should see a small blue robot in the middle of the screen. This is not a robot made of wires and metal, but a software robot, imagined as seen from

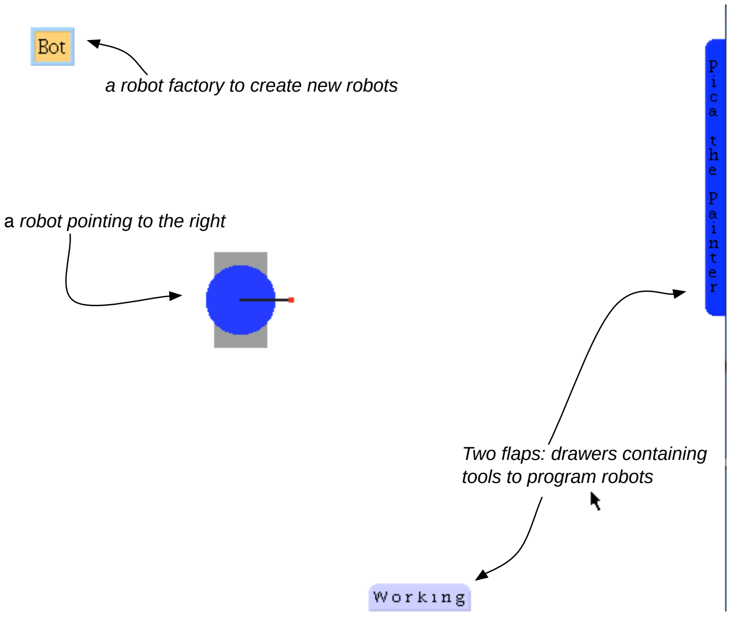

Bot
a robot factory to create new robots
a robot pointing to the right
Pica the Painter
Two flaps: drawers containing tools to program robots
Working

Figure 1.4: The environment is ready to use.

above, pointing toward the right edge of the screen. A robot is a round blue circle; it has two wheels and a small red head that points in its current direction. As you work through this book, you will be sending orders to robots. These orders are called *messages*, and we say that the robots *execute* these messages.

Place the mouse over the robot and wait a second. A balloon pops up with some information about the robot, such as its current location and its direction, as shown in Figure 1.5. Since computer monitors are of varying sizes and resolutions, your robot's position may have other values.

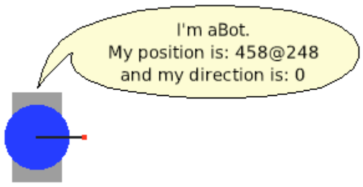


Figure 1.5: Place the mouse over a robot to pop up a balloon with information about the robot.

Sending Messages to a Robot

You can interact directly with a robot by left clicking on the robot with the mouse (or just clicking with a one-button mouse). A messaging balloon pops up, as shown in the left picture in Figure 1.6. In this balloon you can type messages to be sent to the robot. After you type your messages, you send them to the robot by pressing the return key, and the robot then executes them.

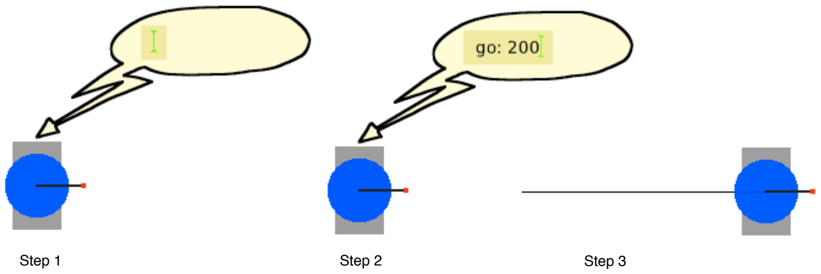


Figure 1.6: Step 1: Left-clicking on a robot causes a messaging balloon to appear. Step 2: You can type a message to the robot to move 200 pixels forward and then press the return key. Step 3: The robot executes the message; it has moved,leaving a trace on the screen behind it.

For example, if you type the message go: 200 followed by the return key, you have told the robot to move forward 200 pixels in its current direction. If you type the message turnLeft: 20 + 70, you are instructing the robot to turn to its left (counterclockwise) 20 + 70 = 90 degrees, as shown in Figure 1.7. This second message is more complex than the previous one, because the value representing the number of degrees that the robot is to turn is itself a message (as I will soon explain), namely, 20 + 70. We will call such messages *compound messages*.

When the message color: Color green is sent to a robot, it changes its color, as shown in Figure 1.8. (You will have to imagine the green color in the grayscale picture.)

You may not understand the format of the messages that I have just presented. Some of them may appear a bit complex. In fact, color: Color green is another compound message. I will explain later how you can develop your own messages. For now, simply type the messages presented to you so that you can become familiar with the robot's environment. If you want to repeat a previous message, you do not have to retype it. Simply use the up and down arrows to navigate over the previous messages that you have

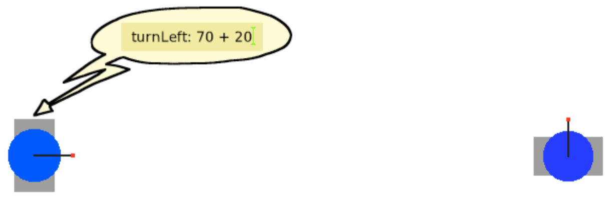


Figure 1.7: Left: sending a compound message. Right: The message has caused the robot to turn to its left by 90 degrees.

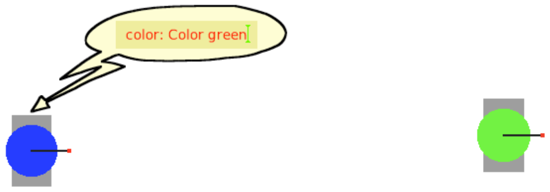


Figure 1.8: Left: Changing the color of a robot to another color. Right: its effect.

sent to the robot. In subsequent chapters, you will learn step by step all the messages that a robot understands, and what is more, you will learn how to define new behaviors for your robots.

Important! To interact with a robot, click on it, type a message, and press the return key.

1.4 Creating a New Robot

The environment already contains a robot, but now I am going to show you how to create new robots. If you are not satisfied with having only one robot, you can create a new one by sending the appropriate message to a robot *factory*. A robot factory is graphically represented as an orange box surrounded by a light blue box, in the middle of which the word Bot is written, as shown in Figure 1.9. In Squeak jargon, and in general in the jargon of object-oriented programming, a robot factory is called a *class*. Classes (factories that produce objects, such as robots) have a name starting with an uppercase letter. Hence this is the class Bot and not bot.

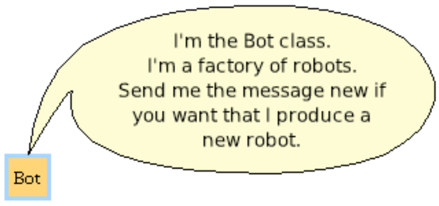

I'm the Bot class.
I'm a factory of robots.
Send me the message new if you want that I produce a new robot.
Bot

Figure 1.9: In Squeak jargon, a robot factory is called a class.Classes produce objects. The Bot class produces new robots.

Just as you did for robots, you can interact with a robot factory by sending it messages. The message to create a new robot is the message new, as shown in Figure 1.10. Note that newly created robots, like your original robot, point to the right of the screen. Each of the two robots has an independent existence, and you can send messages to each of them in turn.

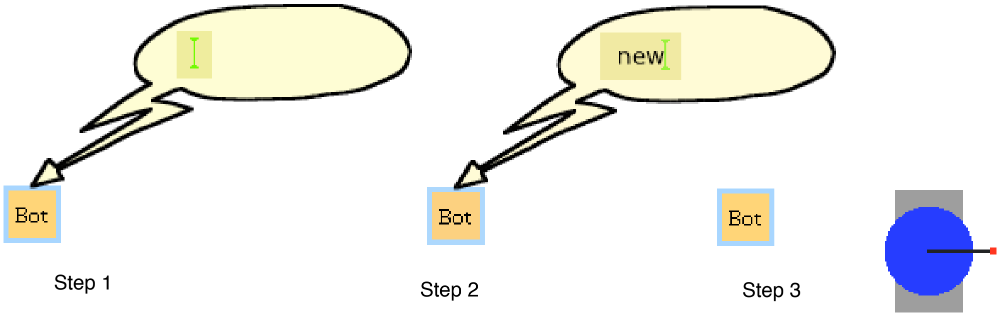


Figure 1.10: Step 1: Start typing a message. Step 2: The message new has been sent to the robot factory. Step 3: In response, the factory has created a robot and delivered it to you.

Important! To create a new robot, send the message *new* to the robot factory, which is the class Bot. When a robot is created, it is always pointing to the east, that is, to the right of the screen.

1.5 Quitting and Saving

The background of the Squeak window application is called the World. The World has a menu offering a number of different options. To display the World menu, just (left) click on the back- ground. You should get a menu similar to the one shown in Figure 1.11. The last group of options consists of all the actions that you can take to quit out of the environment or save your work.

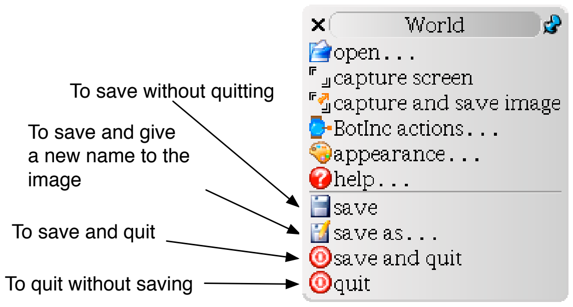


Figure 1.11: The World menu includes actions for quitting and saving.

Selecting the item quit simply quits the environment without saving your work. The result is that the next time you launch the environment, it will be in exactly the same state as the last time you saved it. Selecting the item save saves the complete environment. The next time you start the environment, it will be in exactly the same state as the last time you saved it. Finally, if you select the item save as . . . , the environment asks you to create a new name, and it will then create two new files with that name: one with the extension .image and one with the extension .changes. That is how I created the files Ready[.image] and Ready.changes. To open the environment that you saved with a new name, drag and drop the file with the new name that has the extension .image onto the squeak application file icon as you did to start the environment by dragging and dropping the file Ready[.image].

Installation Troubleshooting

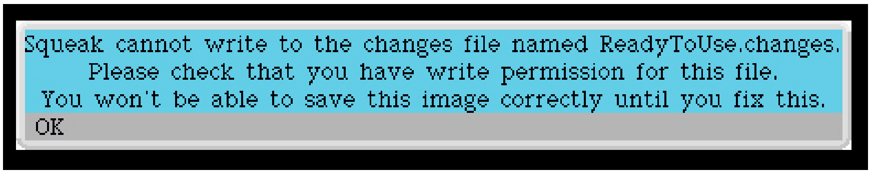
Sometimes things don't proceed just as they should, so in this section I will present some information that should be of help if you encounter problems during installation. First, I will explain the role of the principal files that you obtained when you decompressed the archive. To run the environment provided with this book or with any Squeak distribution, four files are necessary. Knowing about them can help in solving any problems you may encounter.

Image and changes. The file Ready[.image], called simply the image file, and the file Ready.changes, called simply the changes file, contain information about your current Squeak system. These two files are synchronized by Squeak automatically and should be writable (that is, not read-only). Each time you save your environment, these two files are synchronized. You should not edit them with a file editor or change the name of the file manually. If you want to use different names, just use the save as... menu item of the World menu. Squeak will then create a new pair of files for you.

Source. The file named SqueakV3.sources, called the *sources* file, contains the source code of a part of the Squeak environment. You will not need it in working through this book, so do not try to edit it manually. However, this file should always be in the same directory in which the image file is located.

Application. The application files Squeak for Mac and Squeak.exe for PC are the Squeak application. Each of these files is the application that runs when you are programming in Squeak. It should be executable. This file is referred to as the Squeak *application*. In computer-science jargon, this application is called a *virtual machine*, or VM for short.

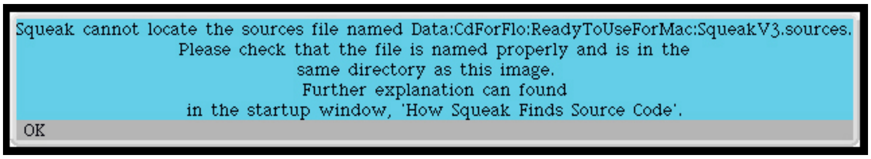
Keep in mind that the image and changes files should be writable. Some operating systems change the properties of files to "read only" when they are copied from an external source. If that happens, Squeak warns you with a message, like that shown in Figure 1.12. If you get such a message, simply quit Squeak without saving, change the property of the file to permit write access, and restart.



```
Squeak cannot write to the changes file named ReadyToUse.changes.
     Please check that you have write permission for this file.
 You won't be able to save this image correctly until you fix this.
```

Figure 1.12: This message appears if the image (Ready.[image]) or changes (Ready.changes) file is not writable.

Another possible problem you may encounter is related to the sources file SqueakV3.sources. This file or an alias pointing to this file should be present in the directory in which the image file is located. If the file itself is not present, you may get the message shown in Figure 1.13. To cure this problem, create an alias to the sources file (SqueakV3.sources) in the directory containing the image file or simply copy the sources file into the directory that contains the image file. You should not have this problem if you are using the distribution for this book.



```
Squeak cannot locate the sources file named Data:CdForFlo:ReadyToUseForMac:SqueakV3.sources.
Please check that the file is named properly and is in the
same directory as this image.
Further explanation can found
in the startup window, 'How Squeak Finds Source Code'.
OK
```

Figure 1.13: Possible messages indicating that the sources file (SqueakV3.sources) is missing from the directory containing the image file.

1.6 Summary

To start the environment, drag and drop the file Ready[.image] or another file that you have saved with the .image extension into the squeak application.

- To send a message to a robot, left click on it, type the message, and press the return key.
- To create a new robot, send the message new to the class Bot, which is your robot factory.
- When a robot is created, it is always pointing to the east, that is, to the right of the screen.
- To obtain the menu for saving the environment, click on the background.