Tip

The code examples provided in this chapter are taken from the MyFirstAndroidApp application. The source code for the MyFirstAndroidApp application is provided for download on the book's websites.

Creating and Configuring a New Android Project

You can create a new Android application in much the same way as when you added the Snake application to your Eclipse workspace.

The first thing you need to do is create a new project in your Eclipse workspace. The Android Project Wizard creates all the required files for an Android application. Follow these steps within Eclipse to create a new project:

1. Choose File, New, Android Project, or choose the Android Project creator icon, which looks like a folder (), on the Eclipse toolbar.
2. Choose a project name, as shown in Figure 3.10. In this case, name the project `MyFirstAndroidApp`.

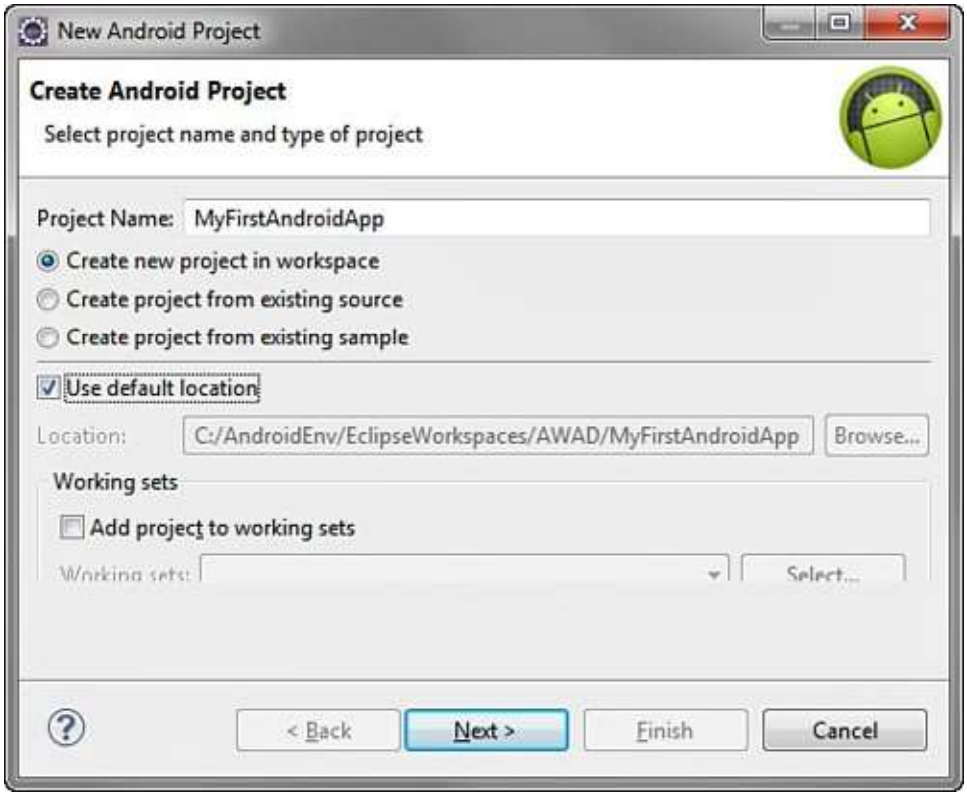

Figure 3.10 Configuring a new Android project.

3. Choose a location for the project files. Because this is a new project, select

the Create New Project in Workspace radio button. Check the Use Default Location check box or change the directory to wherever you want to store the source files. Click Next.

4. Select a build target for your application, as shown in Figure 3.11. Choose a target that is compatible with the Android devices you have in your possession. For this example, you might use the Android 2.3.3 target or, for Ice Cream Sandwich devices, Android 4.0 (API Level 14). Click Next.

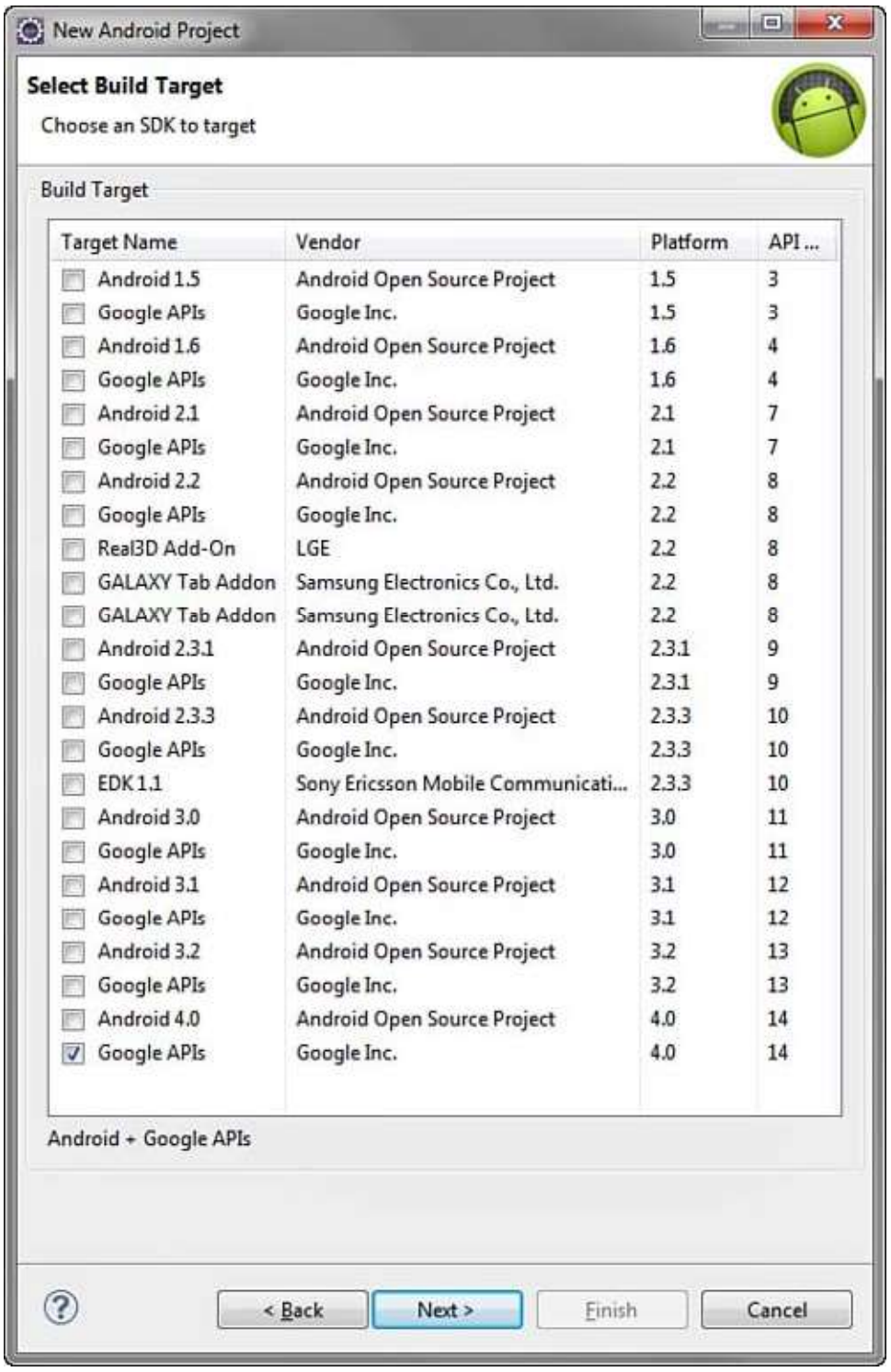

Figure 3.11 Choosing a build target for a new Android project.

5. Configure your application information. Choose an application name. The application name is the “friendly” name of the application and the name shown with the icon on the application launcher. In this case, the application

name is “My First Android App.”

6. Choose a package name. Here you should follow standard package namespace conventions for Java. Because all our code examples in this book fall under the `com.androidbook.*` namespace, we will use the package name `com.androidbook.myfirstandroidapp`, but you are free to choose your own package name.
7. Check the Create Activity check box. This instructs the wizard to create a default launch activity for the application. Call this `Activity` class `MyFirstAndroidAppActivity`.
8. Set the minimum SDK version. This value should be the same or lower than the target SDK API level. Because our application will be compatible with just about any Android device, you can set this number low (like to 4 to represent Android 1.6) or at the target API level to avoid annoying warnings in Eclipse. Make sure you set the minimum SDK version to encompass any test devices you have available so you can successfully install the application on them.

   Your project settings should look like Figure 3.12.

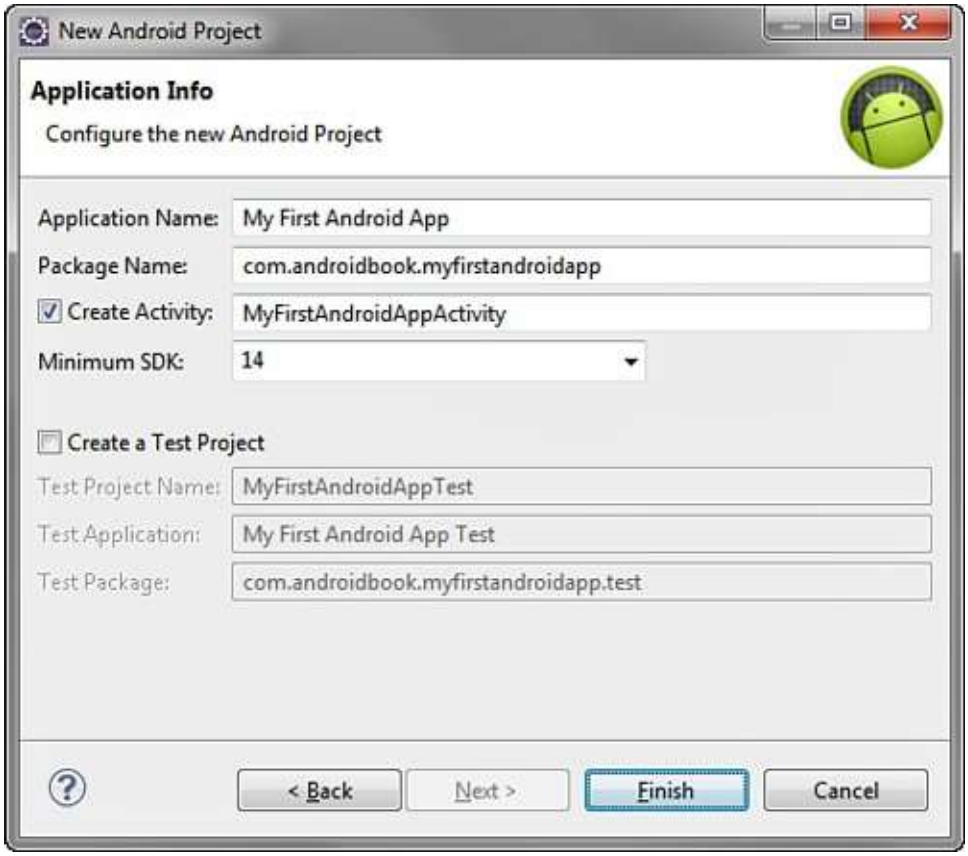

Figure 3.12 Configuring My First Android App using the Android Project Wizard

Wizard.

9. Finally, click the Finish button.

Core Files and Directories of the Android Application

Every Android application has a set of core files that are created and used to define the functionality of the application. The following files are created by default with a new Android application:

- `AndroidManifest.xml`—The central configuration file for the application. It defines your application's capabilities and permissions as well as how it runs.
- `project.properties`—A generated build file used by Eclipse and the Android ADT plug-in. It defines your application's build target and other build system options, as required. Do not edit this file.
- `proguard.cfg`—A generated build file used by Eclipse, ProGuard, and the Android ADT plug-in. Edit this file to configure your code optimization and obfuscation settings for release builds.
- `/src` **folder**—Required folder for all source code.
- `/src/com/androidbook/myfirstandroidapp/MyFirstAndroidAppActivity.java`—Main entry point to this application, named `MyFirstAndroidAppActivity`. This activity has been defined as the default launch activity in the Android manifest file.
- *`gen`*`com/androidbook/myfirstandroidapp/R.java`—A generated resource management source file. Do not edit this file.
- `/assets` **folder**—Required folder where uncompiled file resources can be included in the project. Application assets are pieces of application data (files, directories) that you do not want managed as application resources.
- `/res` **folder**—Required folder where all application resources are managed. Application resources include animations, drawable graphics, layout files, data-like strings and numbers, and raw files.
- *`res`*`drawable-*`—Application icon graphic resources are included in several sizes for different device screen resolutions.
- *`res`*`layout/main.xml`—Layout resource file used by `MyFirstAndroidAppActivity` to organize controls on the main application screen.
- *`res`*`values/strings.xml`—The resource file where string resources are defined.

A number of other files are saved on disk as part of the Eclipse project in the workspace. However, the files and resource directories included in the list here are the important project files you will use on a regular basis.

Creating an AVD for Your Project

The next step is to create an AVD that describes what type of device you want to emulate when running the application. For this example, we can use the AVD we created for the Snake application. An AVD describes a device, not an application. Therefore, you can use the same AVD for multiple applications. You can also create similar AVDs with the same configuration but different data (such as different applications installed and different SD card contents).

Creating a Launch Configuration for Your Project

Next, you must create a Run and Debug launch configuration in Eclipse to configure the circumstances under which the MyFirstAndroidApp application builds and launches. The launch configuration is where you configure the emulator options to use and the entry point for your application.

You can create Run configurations and Debug configurations separately, with different options for each. Begin by creating a Run configuration for the application. Follow these steps to create a basic Run configuration for the MyFirstAndroidApp application:

1. Choose Run, Run Configurations (or right-click the project and choose Run As).
2. Double-click Android Application.
3. Name your Run configuration `MyFirstAndroidAppRunConfig`.
4. Choose the project by clicking the Browse button and choosing the MyFirstAndroidApp project.
5. Switch to the Target tab and set the Device Target Selection Mode to Manual.

Tip

If you leave the Device Target Selection Mode set to Automatic when you choose Run or Debug in Eclipse, your application is automatically installed and run on the device if the device is plugged in. Otherwise, the application starts in the emulator with the specified AVD. By choosing Manual, you are always prompted for whether (a) you want your application to be launched in an existing emulator; (b) you want your application to be launched in a new emulator instance and are allowed to specify an AVD; or (c) you want your application to be launched on the device (if it's plugged in). If any emulator is already running, the device is then plugged in, and the mode is set to Automatic, you see this same prompt, too.

Now create a Debug configuration for the application. This process is similar to

creating a Run configuration. Follow these steps to create a basic Debug configuration for the MyFirstAndroidApp application:

1. Choose Run, Debug Configurations (or right-click the project and choose Debug As).
2. Double-click Android Application.
3. Name your Debug configuration `MyFirstAndroidAppDebugConfig`.
4. Choose the project by clicking the Browse button and choosing the MyFirstAndroidApp project.
5. Switch to the Target tab and set the Device Target Selection Mode to Manual.
6. Click Apply and then click Close.

You now have a Debug configuration for your application.

Running Your Android Application in the Emulator

Now you can run the MyFirstAndroidApp application using the following steps:

1. Choose the Run As icon drop-down menu on the toolbar ().
2. Pull the drop-down menu and choose the Run configuration you created. (If you do not see it listed, choose the Run Configurations... item and select the appropriate configuration. The Run configuration shows up on this drop-down list the next time you run the configuration.)
3. Because you chose the Manual Target Selection mode, you are now prompted for your emulator instance. Change the selection to Launch a New Android Virtual Device and then select the AVD you created, as shown in Figure 3.13. Here you can choose from an already-running emulator or launch a new instance with an AVD that is compatible with the application settings.

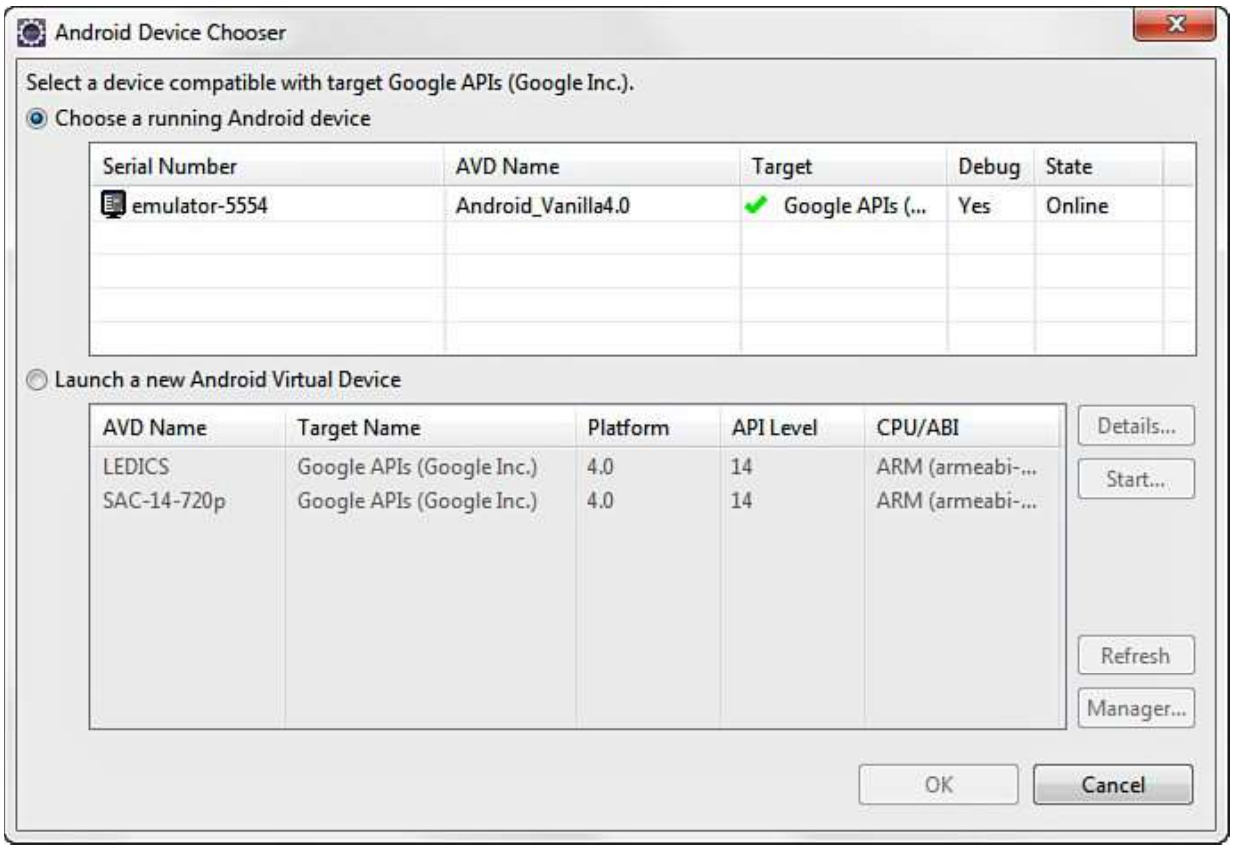

Figure 3.13 Manually choosing a target selection mode.

4. The Android emulator starts up, which might take a moment.
5. Click the Menu button or push the slider to the right to unlock the emulator.
6. The application starts, as shown in Figure 3.14.

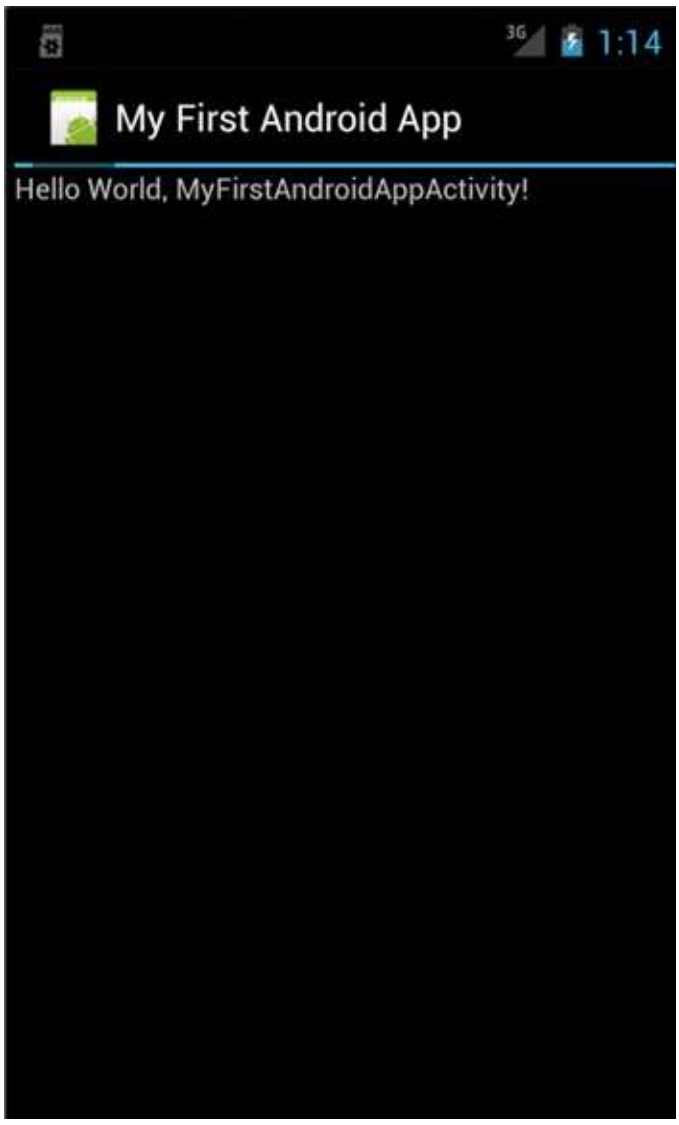

Figure 3.14 My First Android App running in the emulator.

7. Click the Back button in the Emulator to end the game or click Home to suspend it.
8. Click the grid button to browse all installed applications. Your screen looks something like Figure 3.15.

Figure 3.15 The My First Android App icon shown in the application listing.

9. Click the My First Android Application icon to launch the application again.

Debugging Your Android Application in the Emulator

Before we go any further, you need to become familiar with debugging in the emulator. To illustrate some useful debugging tools, let's manufacture an error in the My First Android Application.

In your project, edit the source file called `MyFirstAndroidApp.java`. Create a new method called `forceError()` in your class and make a call to this method in your