

3. Writing Your First Android Application

You should now have a workable Android development environment set up on your computer. Hopefully, you also have an Android device as well. Now it's time for you to start writing some Android code. In this chapter, you learn how to add and create Android projects in Eclipse and verify that your Android development environment is set up correctly. You also write and debug your first Android application in the software emulator and on an Android device.



Note

The Android development tools are updated frequently. We have made every attempt to provide the latest steps for the latest tools. However, these steps and the user interfaces described in this chapter may change at any time. Please review the Android development website (<http://d.android.com/sdk>) and our book website (<http://androidbook.blogspot.com>) for the latest information.

Testing Your Development Environment

The best way to make sure you configured your development environment correctly is to take an existing Android application and run it. You can do this easily by using one of the sample applications provided as part of the Android SDK in the `samples` subdirectory found where your Android SDK is installed.

Within the Android SDK sample applications, you will find a classic game called *Snake* (<http://goo.gl/wRojX>). To build and run the Snake application, you must create a new Android project in your Eclipse workspace based on the existing Android sample project, create an appropriate Android Virtual Device (AVD) profile, and configure a launch configuration for that project. After you have everything set up correctly, you can build the application and run it on the Android emulator and on an Android device. By testing your development environment with a sample application, you can rule out project configuration and coding issues and focus on determining whether the tools are set up properly for Android development. After this fact has been established, you can move on to writing and compiling your own applications.

Adding the Snake Project to Your Eclipse Workspace

The first thing you need to do is add the Snake project to your Eclipse workspace. To do this, follow these steps:

1. Click **File** > **New** > **Project**.

1. Choose File, New, Other....
2. Choose Android, Android Sample Project (see [Figure 3.1](#)). Click Next.

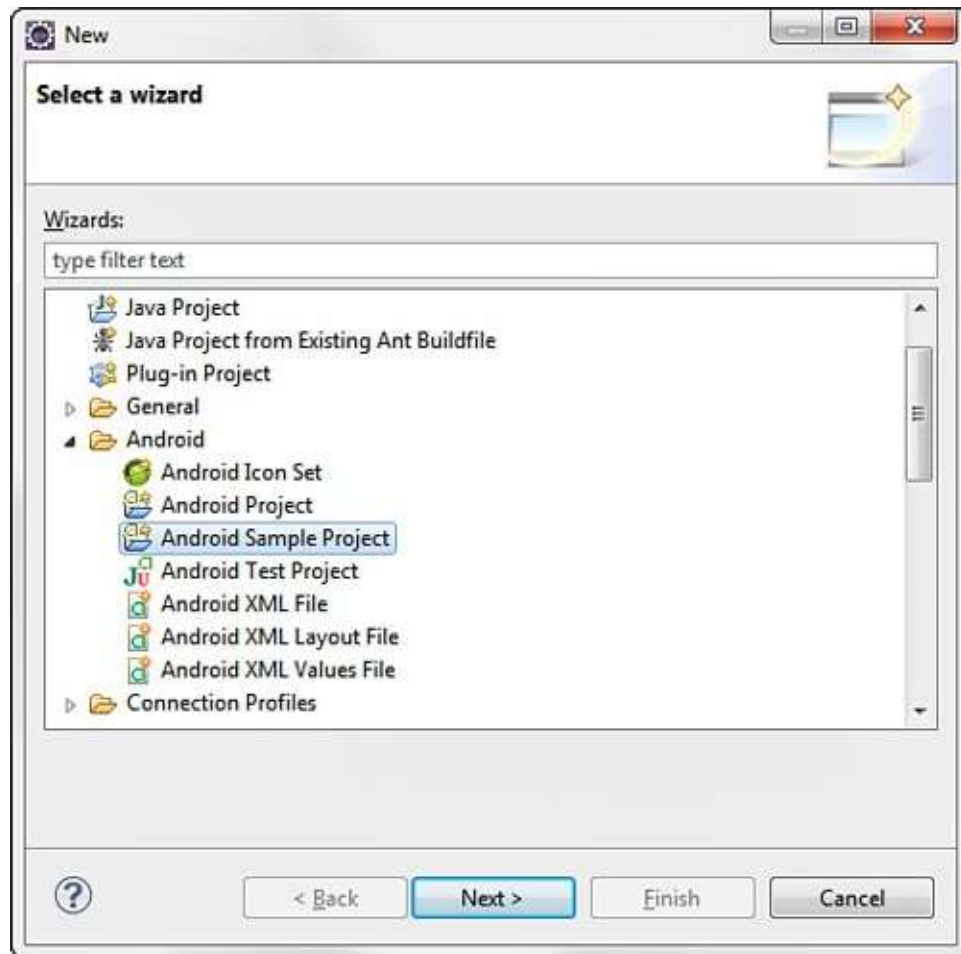


Figure 3.1 Creating a new Android project.

3. Choose your build target (see [Figure 3.2](#)). In this case, we've picked Android 4.0, API Level 14, from the Android Open Source Project. Click Next.

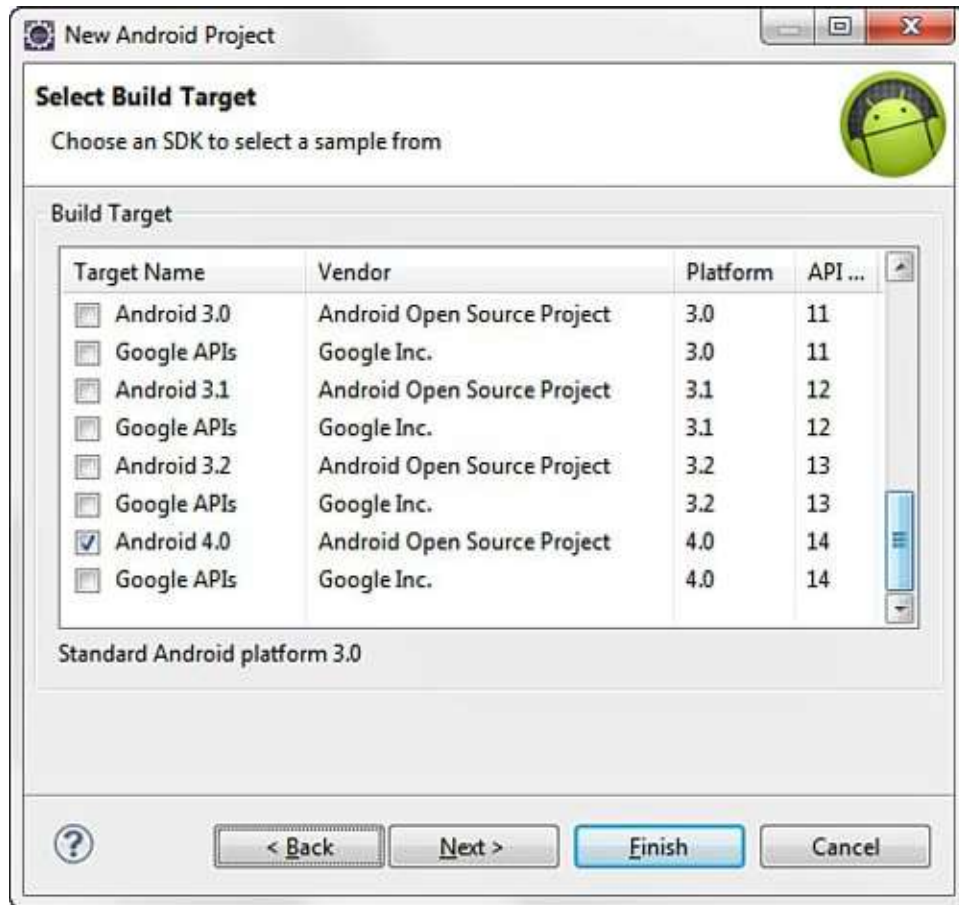


Figure 3.2 Choose an API level for the sample.

4. Next, select which sample you want to create (see [Figure 3.3](#)). Choose Snake.

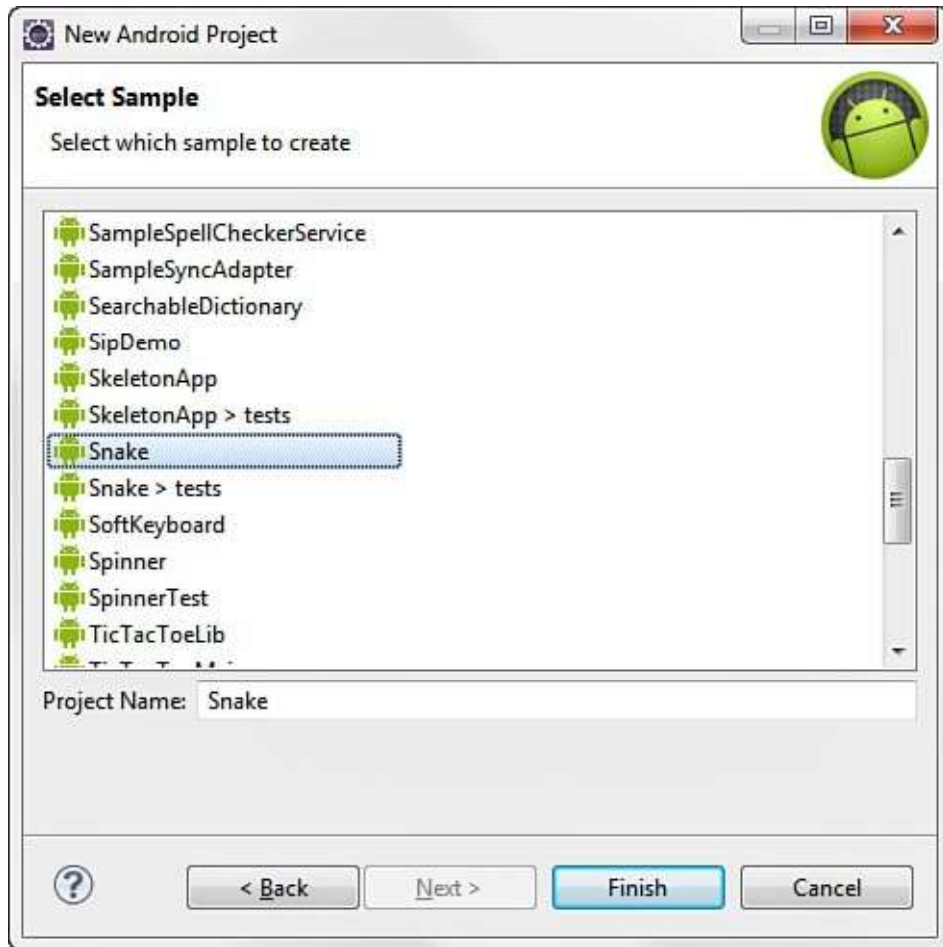


Figure 3.3 Picking the sample project.

5. Click Finish. You now see the Snake project files in your workspace (see [Figure 3.4](#)).

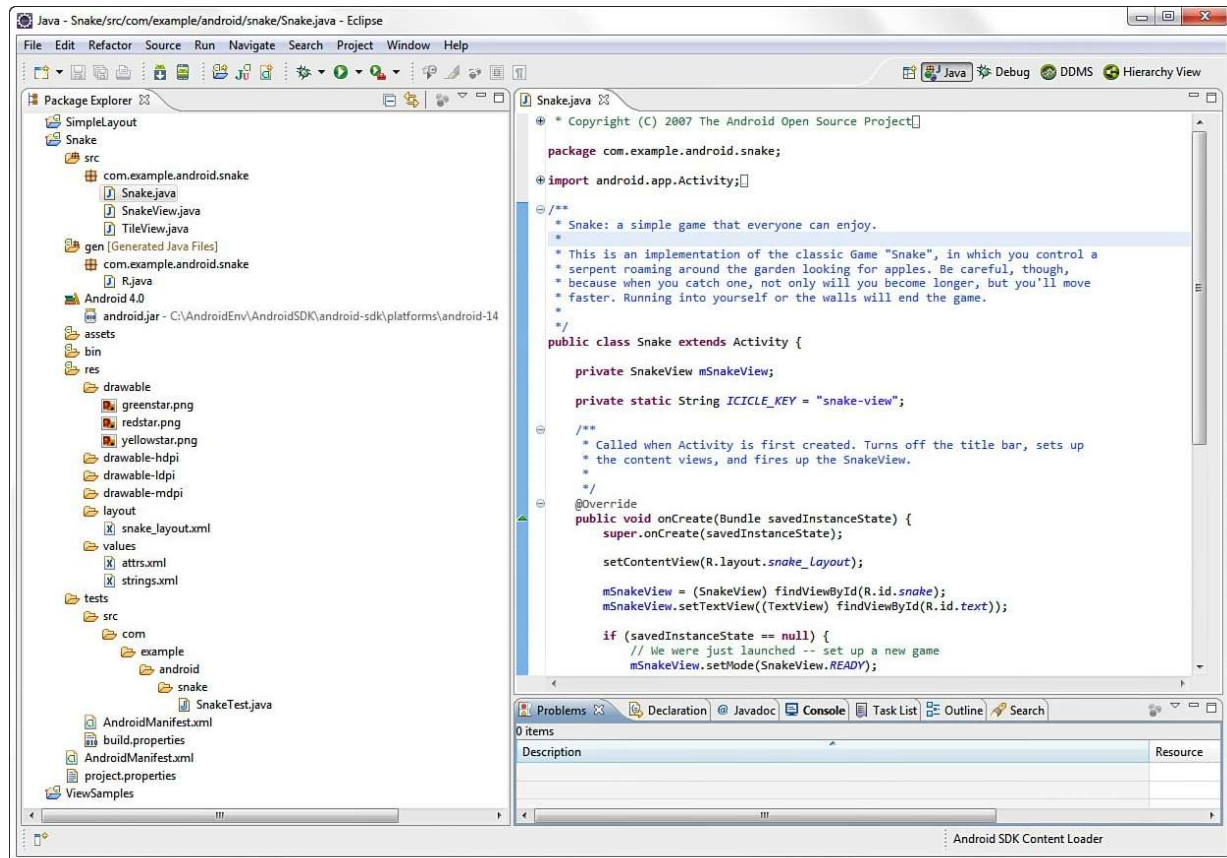


Figure 3.4 The Snake project files.



Warning


Occasionally Eclipse shows the error “Project ‘Snake’ is missing required source folder: gen” when you’re adding an existing project to the workspace. If this happens, navigate to the project file called R.java under the /gen directory and delete it. The R.java file is automatically regenerated and the error should disappear. Performing a Clean operation followed by a Build operation does not always solve this problem.

Creating an Android Virtual Device (AVD) for Your Snake Project

The next step is to create an AVD that describes what type of device you want to emulate when running the Snake application. This AVD profile describes what type of device you want the emulator to simulate, including which Android platform to support. You do not need to create new AVDs for each application, only for each device you want to emulate. You can specify different screen sizes and orientations, and you can specify whether the emulator has an SD card and, if it does, what capacity the card is.

For the purposes of this example, an AVD for the default installation of Android

2.3.3 suffices. Here are the steps to create a basic AVD:

1. Launch the Android Virtual Device Manager from within Eclipse by clicking the little Android device icon on the toolbar (). If you cannot find the icon, you can also launch the manager through the Window menu of Eclipse.
 2. Click the New button.
 3. Choose a name for your AVD. Because we are going to take all the defaults, give this AVD a name of `Android_Vanilla4.0`.
 4. Choose a build target. We want a typical Android 4.0 device, so choose Google APIs (Google Inc.) – API Level 14 from the drop-down menu. This will include the Google Android applications, such as the Maps application, as part of the platform image.
 5. Choose an SD card capacity, in either kibibytes or mibibytes. Not familiar with kibibytes? See this Wikipedia entry: <http://goo.gl/N3Rdd>. This SD card image will take up space on your hard drive, so choose something reasonable, such as 1024MiB.
 6. Seriously consider enabling the Snapshot feature. This feature greatly improves emulator startup performance. See [Appendix A](#), “[The Android Emulator Quick-Start Guide](#),” for details.
-



Warning

As of this writing, there's a known issue with the Android Tools R14 emulator that prevents the snapshot feature from working correctly. See <http://goo.gl/pnMt0> for more information on the current known issues.

7. Choose a skin. This option controls the different resolutions of the emulator. In this case, we use the recommended WVGA800 screen. (The emulator in Tools R14 has known performance issues with running the standard Android 4.0 screen resolution of WXGA720.) This skin most directly correlates to the popular 4.0 devices. Feel free to choose the most appropriate skin to match the Android device on which you plan to run the application.

Your project settings will look like [Figure 3.5](#).

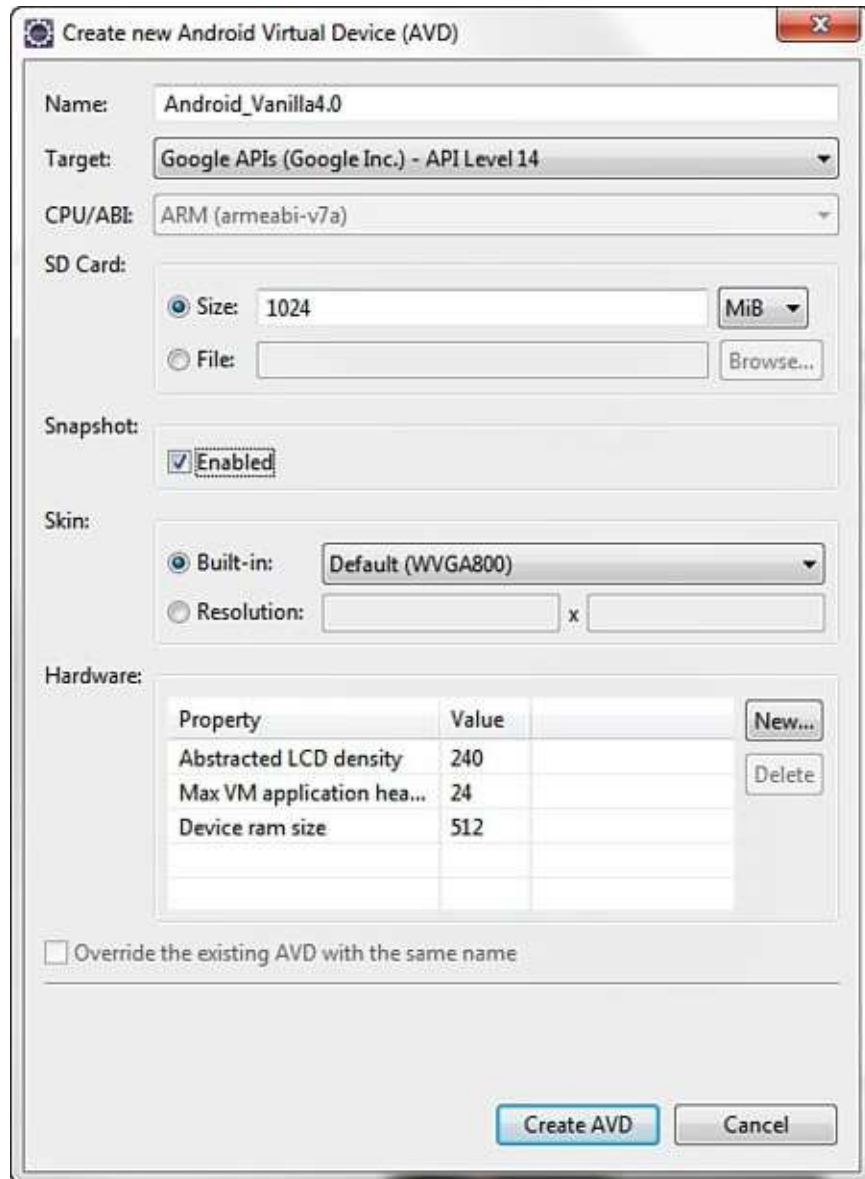


Figure 3.5 Creating a new AVD.

8. Click the Create AVD button and then wait for the operation to complete.
9. Click Finish. Because the AVD manager formats the memory allocated for SD card images, creating AVDs with SD cards could take a few moments.

For more information on creating different types of AVDs, check out [Appendix A](#).

Creating a Launch Configuration for Your Snake Project

Next, you must create a launch configuration in Eclipse to configure under what circumstances the Snake 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, each with different options. These configurations are created under the Run menu in Eclipse (Run, Run Configurations and Run, Debug Configurations). Follow these steps to create a basic Debug configuration for the Snake application:

1. Choose Run, Debug Configurations.
2. Double-click Android Application to create a new configuration.
3. Name your Debug configuration `SnakeDebugConfiguration`.
4. Choose the project by clicking the Browse button and choosing the Snake project.
5. Switch to the Target tab and, from the preferred AVD list, choose the `Android_Vanilla4.0` AVD created earlier, as shown in [Figure 3.6](#).

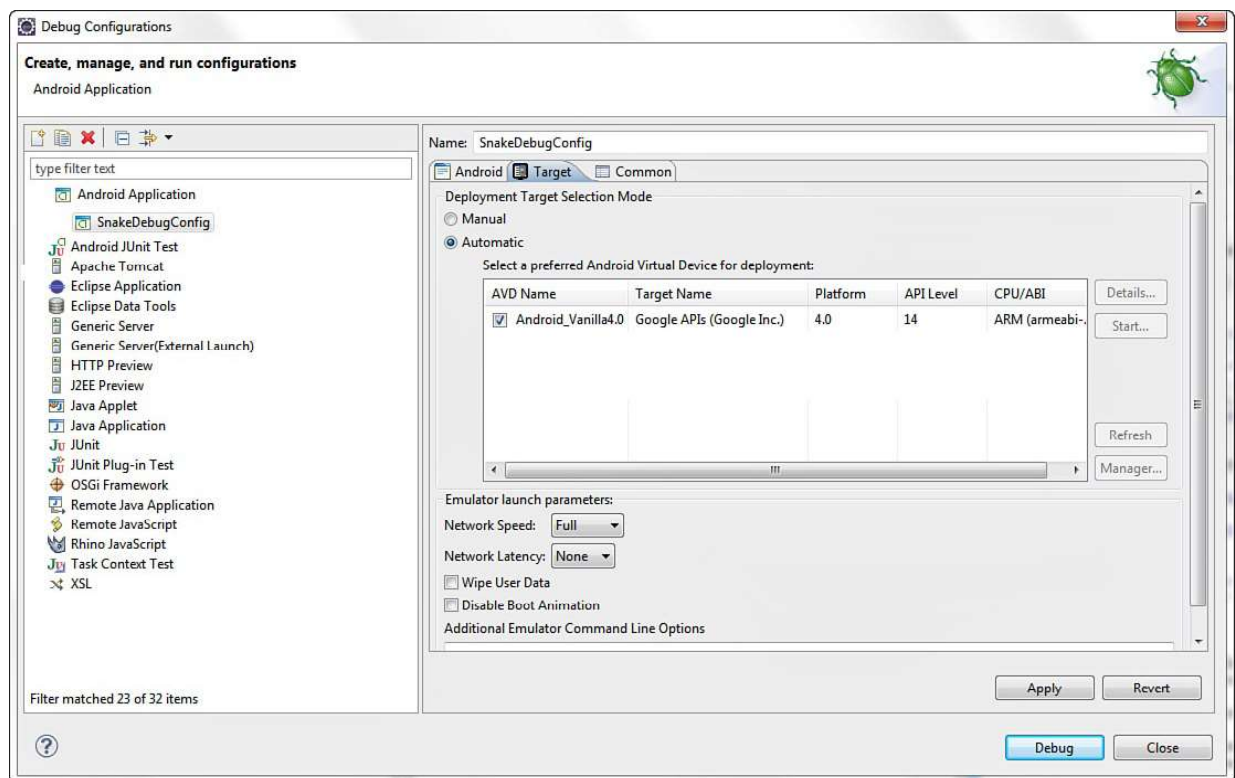


Figure 3.6 The Snake application Debug configuration in Eclipse.

You can set other emulator and launch options on the Target and Common tabs, but for now we are leaving the defaults as they are.

Running the Snake Application in the Android Emulator

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

1. Choose the Debug As icon drop-down menu on the toolbar ()

2. Pull the drop-down menu and choose the `SnakeDebugConfiguration` you created. If you do not see your new configuration listed, find it in the Debug Configurations listing and click the Debug button. Subsequent launches can be initiated from the little bug drop-down.
 3. The Android emulator starts up; this might take a few moments to initialize. Then the application will be installed or reinstalled onto the emulator.
-



Tip

It can take a long time for the emulator to start up, even on very fast computers. You might want to leave it around while you work and reattach to it as needed. The tools in Eclipse handle reinstalling the application and re-launching the application, so you can more easily keep the emulator loaded all the time. This is another reason to enable the Snapshot feature for each AVD. You can also use the Start button on the Android Virtual Device Manager to load up an emulator before you need it. Launching the AVD this way also gives you some additional options such as screen scaling (see [Figure 3.7](#)), which can be used to either fit the AVD on your screen if it's very high resolution or more closely emulate the size it might be on real hardware.

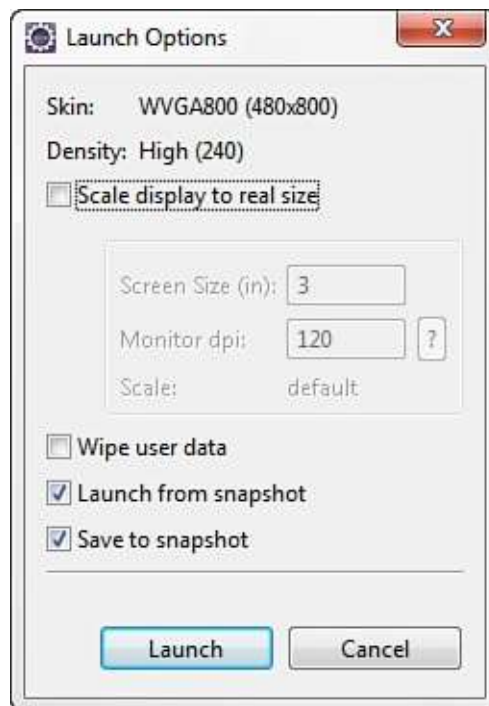


Figure 3.7 Configuring AVD launch options.

4. If necessary, swipe the screen from left to right to unlock the emulator, as shown in [Figure 3.8](#).

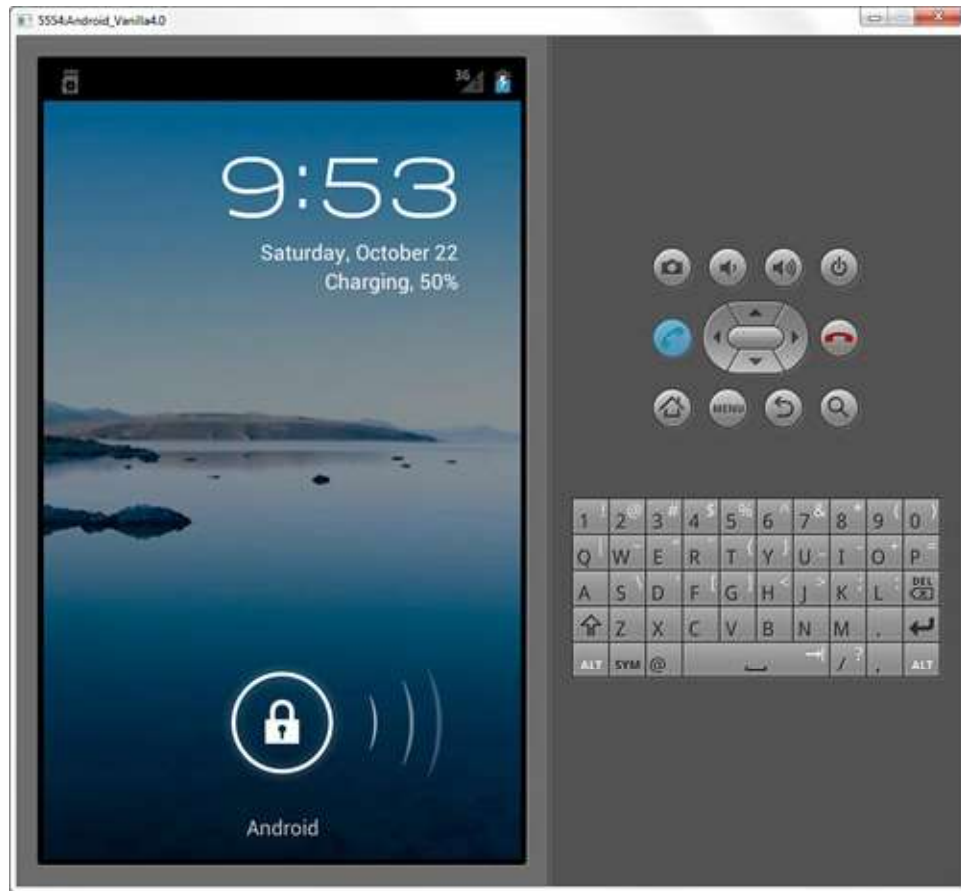


Figure 3.8 The Android emulator launching (locked).

5. The Snake application starts and you can play the game, as shown in [Figure 3.9](#).