

2ND EDITION

Android Programming

THE BIG NERD RANCH GUIDE

Bill Phillips, Chris Stewart, Brian Hardy
& Kristin Marsicano



Android Programming: The Big Nerd Ranch Guide

by Bill Phillips, Chris Stewart, Brian Hardy and Kristin Marsicano

Copyright © 2015 Big Nerd Ranch, LLC.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, contact

Big Nerd Ranch, LLC.
200 Arizona Ave NE
Atlanta, GA 30307
(770) 817-6373
<http://www.bignerdranch.com/>
book-comments@bignerdranch.com

The 10-gallon hat with propeller logo is a trademark of Big Nerd Ranch, Inc.

Exclusive worldwide distribution of the English edition of this book by

Pearson Technology Group
800 East 96th Street
Indianapolis, IN 46240 USA
<http://www.informit.com>

The authors and publisher have taken care in writing and printing this book but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

ISBN-10 0134171497
ISBN-13 978-0134171494

Second edition, first printing, August 2015
Release D.2.1.1

Dedication

To God, or to whatever it is that you personally have faith in. Reader, I hope that you find the many explanations in this book useful. Please don't ask me how they got here, though. I once thought that I was responsible. Fortunately for you, I was wrong.

— B.P.

To my dad, David, for teaching me the value of hard work. To my mom, Lisa, for pushing me to always do the right thing.

— C.S.

For Donovan. May he live a life filled with activities and know when to use fragments.

— B.H.

*To my dad, Dave Vadas, for inspiring and encouraging me to pursue a career in computing. And to my mom, Joan Vadas, for cheering me on through all the ups and downs (and for reminding me that watching an episode of *The Golden Girls* always makes things better).*

— K.M.

Acknowledgments

We feel a bit sheepish having our names on the cover of this book. The truth is that without an army of collaborators, this book could never have happened. We owe them all a debt of gratitude.

- Our co-instructors and members of our Android development team, Andrew Lunsford, Bolot Kerimbaev, Brian Gardner, David Greenhalgh, Jason Atwood, Josh Skeen, Kurt Nelson, Matt Compton, Paul Turner, and Sean Farrell. We thank them for their patience in teaching work-in-progress material, as well as their suggestions and corrections. If we could give ourselves additional brains to do with as we pleased, we would not. We would just put the new brains in a big pile, and share them with our colleagues. We trust them at least as much as we trust our own selves.
- Special thanks to Sean Farrell for graciously updating many screen shots as Android Studio evolved, and to Matt Compton for publishing all of our sample apps to the Google Play Store.
- Kar Loong Wong and Zack Simon, members of Big Nerd Ranch's amazing design team. Kar made BeatBox look intimidating and polished, and provided advice and imagery for the material design chapter. Zack took time out of his schedule to design MockWalker for us. Kar and Zack's design abilities seem like unknowable superpowers to us. We thank them, and bid them fond returns to their home planet.
- Our technical reviewers, Frank Robles and Roy Kravitz, who helped us find and fix flaws.
- Thanks to Aaron Hillegass. Aaron's faith in people is one of the great and terrifying forces of nature. Without it, we would never have had the opportunity to write this book, nor would we ever have completed it. (He also gave us money, which was very friendly of him.)
- Our editor, Elizabeth Holaday, who many times saved us from going down rabbit holes. She kept our writing focused on what our readers actually care about and spared you all from confusing, boring, and irrelevant detours. Thank you, Liz, for being organized and patient, and for being a constant supportive presence, even though you live many miles away.
- Ellie Volckhausen, who designed our cover.
- Simone Payment, our copy-editor, who found and smoothed rough spots.
- Chris Loper at IntelligentEnglish.com, who designed and produced the print book and the EPUB and Kindle versions. His DocBook toolchain made life much easier, too.

Finally, thanks to our students. We wish that we had room to thank every single student who gave us a correction or opinion on the book as it was shaping up. It is your curiosity we have worked to satisfy, your confusions we have worked to clarify. Thank you.

Table of Contents

Learning Android	xvii
Prerequisites	xvii
What's New in the Second Edition?	xvii
How to Use This Book	xviii
How This Book is Organized	xviii
Challenges	xix
Are you more curious?	xix
Code Style	xix
Typographical Conventions	xx
Android Versions	xx
The Necessary Tools	xxi
Downloading and Installing Android Studio	xxi
Downloading Earlier SDK Versions	xxi
An Alternative Emulator	xxii
A Hardware Device	xxii
1. Your First Android Application	1
App Basics	2
Creating an Android Project	2
Navigating in Android Studio	8
Laying Out the User Interface	9
The view hierarchy	13
Widget attributes	14
Creating string resources	15
Previewing the layout	15
From Layout XML to View Objects	16
Resources and resource IDs	18
Wiring Up Widgets	20
Getting references to widgets	21
Setting listeners	22
Making Toasts	23
Using code completion	25
Running on the Emulator	26
For the More Curious: Android Build Process	29
Android build tools	31
2. Android and Model-View-Controller	33
Creating a New Class	34
Generating getters and setters	34
Model-View-Controller and Android	37
Benefits of MVC	38
Updating the View Layer	39
Updating the Controller Layer	41
Running on a Device	46
Connecting your device	46
Configuring your device for development	47
Adding an Icon	48

Adding resources to a project	49
Referencing resources in XML	52
Challenges	53
Challenge: Add a Listener to the TextView	53
Challenge: Add a Previous Button	54
Challenge: From Button to ImageButton	55
3. The Activity Lifecycle	57
Logging the Activity Lifecycle	58
Making log messages	58
Using LogCat	60
Rotation and the Activity Lifecycle	63
Device configurations and alternative resources	64
Saving Data Across Rotation	68
Overriding onSaveInstanceState(Bundle)	69
The Activity Lifecycle, Revisited	70
For the More Curious: Testing onSaveInstanceState(Bundle)	72
For the More Curious: Logging Levels and Methods	73
4. Debugging Android Apps	75
Exceptions and Stack Traces	76
Diagnosing misbehaviors	77
Logging stack traces	78
Setting breakpoints	79
Using exception breakpoints	82
Android-Specific Debugging	84
Using Android Lint	84
Issues with the R class	85
5. Your Second Activity	87
Setting Up a Second Activity	88
Creating a new activity	89
A new activity subclass	92
Declaring activities in the manifest	92
Adding a Cheat! button to QuizActivity	93
Starting an Activity	95
Communicating with intents	96
Passing Data Between Activities	97
Using intent extras	98
Getting a result back from a child activity	101
How Android Sees Your Activities	106
Challenge	109
6. Android SDK Versions and Compatibility	111
Android SDK Versions	111
Compatibility and Android Programming	112
A sane minimum	112
Minimum SDK version	114
Target SDK version	114
Compile SDK version	114
Adding code from later APIs safely	114
Using the Android Developer Documentation	117

Challenge: Reporting the Build Version	119
7. UI Fragments and the Fragment Manager	121
The Need for UI Flexibility	122
Introducing Fragments	123
Starting CriminalIntent	124
Creating a new project	126
Fragments and the support library	128
Adding dependencies in Android Studio	129
Creating the Crime class	132
Hosting a UI Fragment	133
The fragment lifecycle	133
Two approaches to hosting	134
Defining a container view	135
Creating a UI Fragment	136
Defining CrimeFragment's layout	136
Creating the CrimeFragment class	138
Adding a UI Fragment to the FragmentManager	142
Fragment transactions	143
The FragmentManager and the fragment lifecycle	145
Application Architecture with Fragments	146
The reason all our activities will use fragments	147
For the More Curious: Why Support Fragments are Superior	148
For the More Curious: Using Built-In Fragments	148
8. Creating User Interfaces with Layouts and Widgets	149
Upgrading Crime	149
Updating the Layout	150
Wiring Widgets	153
More on XML Layout Attributes	154
Styles, themes, and theme attributes	154
Screen pixel densities and dp and sp	155
Android's design guidelines	156
Layout parameters	157
Margins vs. padding	157
Using the Graphical Layout Tool	158
Creating a landscape layout	160
Adding a new widget	161
Editing attributes in properties view	161
Reorganizing widgets in the component tree	162
Updating child layout parameters	163
How android:layout_weight works	164
The graphical layout tool and you	165
Widget IDs and multiple layouts	166
Challenge: Formatting the Date	166
9. Displaying Lists with RecyclerView	167
Updating CriminalIntent's Model Layer	168
Singletons and centralized data storage	168
An Abstract Activity for Hosting a Fragment	171
A generic fragment-hosting layout	171

An abstract Activity class	172
RecyclerView, Adapter, and ViewHolder	176
ViewHolders and Adapters	177
Using a RecyclerView	180
Implementing an Adapter and ViewHolder	182
Customizing List Items	185
Creating the list item layout	185
Using a custom item view	188
Responding to Presses	190
For the More Curious: ListView and GridView	191
For the More Curious: Singletons	192
10. Using Fragment Arguments	193
Starting an Activity from a Fragment	193
Putting an extra	194
Retrieving an extra	195
Updating CrimeFragment's view with Crime data	196
The downside to direct retrieval	197
Fragment Arguments	197
Attaching arguments to a fragment	198
Retrieving arguments	199
Reloading the List	200
Getting Results with Fragments	202
Challenge: Efficient RecyclerView Reloading	203
For the More Curious: Why Use Fragment Arguments?	204
11. Using ViewPager	205
Creating CrimePagerActivity	206
ViewPager and PagerAdapter	207
Integrating CrimePagerActivity	208
FragmentManagerAdapter vs. FragmentPagerAdapter	211
For the More Curious: How ViewPager Really Works	212
For the More Curious: Laying Out Views in Code	213
12. Dialogs	215
The AppCompatActivity Library	216
Creating a DialogFragment	217
Showing a DialogFragment	220
Setting a dialog's contents	221
Passing Data Between Two Fragments	224
Passing data to DatePickerFragment	225
Returning data to CrimeFragment	226
Challenge: More Dialogs	233
Challenge: A Responsive DialogFragment	233
13. The Toolbar	235
AppCompatActivity	235
Using the AppCompatActivity library	236
Menus	238
Defining a menu in XML	239
Creating the menu	244
Responding to menu selections	246

Enabling Hierarchical Navigation	248
How hierarchical navigation works	249
An Alternative Action Item	249
Toggling the action item title	251
“Just one more thing...”	252
For the More Curious: Toolbar vs Action Bar	254
Challenge: Deleting Crimes	255
Challenge: Plural String Resources	255
Challenge: An Empty View for the RecyclerView	255
14. SQLite Databases	257
Defining a Schema	257
Building Your Initial Database	258
Debugging database issues	261
Gutting CrimeLab	262
Writing to the Database	263
Using ContentValues	263
Inserting and updating rows	264
Reading from the Database	266
Using a CursorWrapper	267
Converting to model objects	269
For the More Curious: More Databases	271
For the More Curious: The Application Context	272
Challenge: Deleting Crimes	272
15. Implicit Intents	273
Adding Buttons	274
Adding a Suspect to the Model Layer	276
Using a Format String	278
Using Implicit Intents	279
Parts of an implicit intent	280
Sending a crime report	281
Asking Android for a contact	283
Checking for responding activities	287
Challenge: ShareCompat	289
Challenge: Another Implicit Intent	289
16. Taking Pictures with Intents	291
A Place for Your Photo	291
Including layout files	292
External Storage	294
Designating a picture location	296
Using a Camera Intent	297
External storage permission	298
Firing the intent	299
Scaling and Displaying Bitmaps	301
Declaring Features	304
For the More Curious: Using Includes	304
Challenge: Detail Display	305
Challenge: Efficient Thumbnail Load	305
17. Two-Pane Master-Detail Interfaces	307

Adding Layout Flexibility	308
Modifying SingleFragmentActivity	309
Creating a layout with two fragment containers	309
Using an alias resource	311
Creating tablet alternatives	312
Activity: Fragment Boss	314
Fragment callback interfaces	314
For the More Curious: More on Determining Device Size	323
18. Assets	325
Why Assets, Not Resources	326
Creating BeatBox	326
Importing Assets	329
Getting at Assets	331
Wiring Up Assets for Use	333
Accessing Assets	336
For the More Curious: Non-Assets?	337
19. Audio Playback with SoundPool	339
Creating a SoundPool	339
Loading Sounds	340
Playing Sounds	341
Unloading Sounds	343
Rotation and Object Continuity	344
Retaining a fragment	345
Rotation and retained fragments	346
For the More Curious: Whether to Retain	348
For the More Curious: More on Rotation Handling	349
20. Styles and Themes	353
Color Resources	353
Styles	354
Style inheritance	355
Themes	357
Modifying the theme	357
Adding Theme Colors	359
Overriding Theme Attributes	360
Theme spelunking	361
Modifying Button Attributes	365
For the More Curious: More on Style Inheritance	367
For the More Curious: Accessing Theme Attributes	368
Challenge: An Appropriate Base Theme	368
21. XML Drawables	369
Making Uniform Buttons	369
Shape Drawables	371
State List Drawables	372
Layer List Drawables	374
For the More Curious: Why Bother with XML Drawables?	376
For the More Curious: 9-Patch Images	376
For the More Curious: Mipmap Images	381
22. More About Intents and Tasks	383

Setting Up NerdLauncher	384
Resolving an Implicit Intent	386
Creating Explicit Intents at Runtime	391
Tasks and the Back Stack	393
Switching between tasks	393
Starting a new task	395
Using NerdLauncher as a Home Screen	397
Challenge: Icons	398
For the More Curious: Processes vs. Tasks	398
For the More Curious: Concurrent Documents	401
23. HTTP & Background Tasks	405
Creating PhotoGallery	406
Networking Basics	409
Asking permission to network	411
Using AsyncTask to Run on a Background Thread	411
You and Your Main Thread	413
Beyond the main thread	414
Fetching JSON from Flickr	415
Parsing JSON text	419
From AsyncTask Back to the Main Thread	422
Cleaning Up AsyncTasks	425
For the More Curious: More on AsyncTask	426
For the More Curious: Alternatives to AsyncTask	427
Challenge: Gson	428
Challenge: Paging	428
Challenge: Dynamically Adjusting the Number of Columns	428
24. Loopers, Handlers, and HandlerThread	429
Preparing RecyclerView to Display Images	429
Downloading Lots of Small Things	432
Communicating with the Main Thread	432
Assembling a Background Thread	433
Messages and Message Handlers	435
Message anatomy	435
Handler anatomy	436
Using handlers	437
Passing handlers	441
For the More Curious: AsyncTask vs. Threads	447
Challenge: Preloading and Caching	447
For the More Curious: Solving the Image Downloading Problem	448
25. Search	449
Searching Flickr	449
Using SearchView	455
Responding to SearchView user interactions	458
Simple Persistence with Shared Preferences	460
Polishing Your App	464
Challenge: Polishing Your App Some More	465
26. Background Services	467
Creating an IntentService	467

What Services are For	469
Safe background networking	470
Looking for New Results	471
Delayed Execution with AlarmManager	473
Being a good citizen: using alarms the right way	475
PendingIntent	477
Managing alarms with PendingIntent	477
Controlling Your Alarm	478
Notifications	481
Challenge: Notifications on Android Wear	483
For the More Curious: Service Details	483
What a service does (and does not) do	483
A service's lifecycle	484
Non-sticky services	484
Sticky services	484
Bound services	485
For the More Curious: JobScheduler and JobServices	486
For the More Curious: Sync Adapters	488
Challenge: Using JobService on Lollipop	490
27. Broadcast Intents	491
Regular Intents vs. Broadcast Intents	491
Receiving a System Broadcast: Waking Up on Boot	492
Creating and registering a standalone receiver	492
Using receivers	495
Filtering Foreground Notifications	496
Sending broadcast intents	497
Creating and registering a dynamic receiver	497
Limiting broadcasts to your app using private permissions	500
Passing and receiving data with ordered broadcasts	502
Receivers and Long-Running Tasks	507
For the More Curious: Local Events	507
Using EventBus	507
Using RxJava	508
For the More Curious: Detecting the Visibility of Your Fragment	509
28. Browsing the Web and WebView	511
One Last Bit of Flickr Data	511
The Easy Way: Implicit Intents	514
The Harder Way: WebView	516
Using WebChromeClient to spruce things up	520
Proper Rotation with WebView	522
Dangers of handling configuration changes	523
For the More Curious: Injecting JavaScript Objects	523
For the More Curious: KitKat's WebView Overhaul	524
Challenge: Using the Back Button for Browser History	524
Challenge: Supporting Non-HTTP Links	525
29. Custom Views and Touch Events	527
Setting Up the DragAndDraw Project	527
Setting up DragAndDrawActivity	528

Setting up DragAndDrawFragment	528
Creating a Custom View	530
Creating BoxDrawingView	530
Handling Touch Events	532
Tracking across motion events	534
Rendering Inside onDraw(...)	536
Challenge: Saving State	538
Challenge: Rotating Boxes	538
30. Property Animation	539
Building the Scene	539
Simple Property Animation	542
View transformation properties	544
Using different interpolators	546
Color evaluation	546
Playing Animators Together	548
For the More Curious: Other Animation APIs	550
Legacy animation tools	550
Transitions	550
Challenges	550
31. Locations and Play Services	551
Locations and Libraries	551
Google Play Services	552
Creating Locatr	552
Play Services and Location Testing on Emulators	553
Mock location data	554
Building out Locatr	556
Setting Up Google Play Services	559
Location permissions	560
Using Google Play Services	561
Flickr Geosearch	563
Getting a Location Fix	564
Find and Display an Image	566
Challenge: Progress	569
32. Maps	571
Importing Play Services Maps	571
Mapping on Android	571
Maps API Setup	572
Getting a Maps API Key	572
Setting Up Your Map	574
Getting More Location Data	576
Working with Your Map	579
Drawing on the map	582
For the More Curious: Teams and API Keys	584
33. Material Design	587
Material Surfaces	587
Elevation and Z values	589
State list animators	590
Animation Tools	591

Circular reveal	591
Shared element transitions	593
View Components	597
Cards	597
Floating action buttons	598
Snackbars	600
More on Material Design	601
34. Afterword	603
The Final Challenge	603
Shameless Plugs	603
Thank You	604
Index	605

Learning Android

As a beginning Android programmer, you face a steep learning curve. Learning Android is like moving to a foreign city. Even if you speak the language, it will not feel like home at first. Everyone around you seems to understand things that you are missing. Things you already knew turn out to be dead wrong in this new context.

Android has a culture. That culture speaks Java, but knowing Java is not enough. Getting your head around Android requires learning many new ideas and techniques. It helps to have a guide through unfamiliar territory.

That's where we come in. At Big Nerd Ranch, we believe that to be an Android programmer, you must:

- *write* Android applications
- *understand* what you are writing

This guide will help you do both. We have trained hundreds of professional Android programmers using it. We lead you through writing several Android applications, introducing concepts and techniques as needed. When there are rough spots, when some things are tricky or obscure, you will face them head on, and we will do our best to explain why things are the way they are.

This approach allows you to put what you have learned into practice in a working app right away rather than learning a lot of theory and then having to figure out how to apply it all later. You will come away with the experience and understanding you need to get going as an Android developer.

Prerequisites

To use this book, you need to be familiar with Java, including classes and objects, interfaces, listeners, packages, inner classes, anonymous inner classes, and generic classes.

If these ideas do not ring a bell, you will be in the weeds by page 2. Start instead with an introductory Java book and return to this book afterward. There are many excellent introductory books available, so you can choose one based on your programming experience and learning style.

If you are comfortable with object-oriented programming concepts, but your Java is a little rusty, you will probably be OK. We will provide some brief reminders about Java specifics (like interfaces and anonymous inner classes). Keep a Java reference handy in case you need more support as you go through the book.

What's New in the Second Edition?

This second edition shows how to use the Android Studio integrated development environment to write practical applications for Android 5.1 (Lollipop) that are backwards-compatible through Android 4.1 (Jelly Bean). It includes updated coverage of the fundamentals of Android programming as well as new Lollipop tools like the toolbar and material design. It also covers new tools from the support libraries, like **RecyclerView** and Google Play Services, plus some key standard library tools, like **SoundPool**, animations, and assets.

How to Use This Book

This book is not a reference book. Its goal is to get you over the initial hump to where you can get the most out of the reference and recipe books available. It is based on our five-day class at Big Nerd Ranch. As such, it is meant to be worked through from the beginning. Chapters build on each other and skipping around is unproductive.

In our classes, students work through these materials, but they also benefit from the right environment – a dedicated classroom, good food and comfortable board, a group of motivated peers, and an instructor to answer questions.

As a reader, you want your environment to be similar. That means getting a good night's rest and finding a quiet place to work. These things can help, too:

- Start a reading group with your friends or coworkers.
- Arrange to have blocks of focused time to work on chapters.
- Participate in the forum for this book at <http://forums.bignerdranch.com>.
- Find someone who knows Android to help you out.

How This Book is Organized

As you work through this book, you will write eight Android apps. A couple are very simple and take only a chapter to create. Others are more complex. The longest app spans 11 chapters. All are designed to teach you important concepts and techniques and give you direct experience using them.

<i>GeoQuiz</i>	In your first app, you will explore the fundamentals of Android projects, activities, layouts, and explicit intents.
<i>CriminalIntent</i>	The largest app in the book, <i>CriminalIntent</i> lets you keep a record of your colleagues' lapses around the office. You will learn to use fragments, master-detail interfaces, list-backed interfaces, menus, the camera, implicit intents, and more.
<i>BeatBox</i>	Intimidate your foes with this app while you learn more about fragments, media playback, themes, and drawables.
<i>NerdLauncher</i>	Building this custom launcher will give you insight into the intent system and tasks.
<i>PhotoGallery</i>	A Flickr client that downloads and displays photos from Flickr's public feed, this app will take you through services, multithreading, accessing web services, and more.

<i>DragAndDraw</i>	In this simple drawing app, you will learn about handling touch events and creating custom views.
<i>Sunset</i>	In this toy app, you will create a beautiful representation of a sunset over open water while learning about animations.
<i>Locatr</i>	This app lets you query Flickr for pictures around your current location and display them on a map. In it, you will learn how to use location services and maps.

Challenges

Most chapters have a section at the end with exercises for you to work through. This is your opportunity to use what you have learned, explore the documentation, and do some problem solving on your own.

We strongly recommend that you do the challenges. Going off the beaten path and finding your way will solidify your learning and give you confidence with your own projects.

If you get lost, you can always visit <http://forums.bignerdranch.com> for some assistance.

Are you more curious?

There are also sections at the ends of chapters labeled “For the More Curious.” These sections offer deeper explanations or additional information about topics presented in the chapter. The information in these sections is not absolutely essential, but we hope you will find it interesting and useful.

Code Style

There are two areas where our choices differ from what you might see elsewhere in the Android community:

We use anonymous inner classes for listeners.

This is mostly a matter of opinion. We find it makes for cleaner code in the applications in this book because it puts the listener’s method implementations right where you want to see them. In high-performance contexts or large applications, anonymous inner classes may cause problems, but for most circumstances they work fine.

After we introduce fragments in Chapter 7, we use them for all user interfaces.

Fragments are not an absolutely necessary tool but we find that, when used correctly, they are a valuable tool in any Android developer’s toolkit. Once you get comfortable with fragments, they are not that difficult to work with. Fragments have clear advantages over activities that make them worth the effort, including flexibility in building and presenting your user interfaces.

Typographical Conventions

To make this book easier to read, certain items appear in certain fonts. Variables, constants, and types appear in a fixed-width font. Class names, interface names, and method names appear in a bold, fixed-width font.

All code and XML listings are in a fixed-width font. Code or XML that you need to type in is always bold. Code or XML that should be deleted is struck through. For example, in the following method implementation, you are deleting the call to `makeText(...)` and adding the call to `checkAnswer(true)`.

```
@Override
public void onClick(View v) {
    Toast.makeText(QuizActivity.this, R.string.incorrect_toast,
    Toast.LENGTH_SHORT).show();
    checkAnswer(true);
}
```

Android Versions

This book teaches Android development for all widely used versions of Android. As of this writing, that is Android 4.1 (Jelly Bean) - Android 5.1 (Lollipop). While there is a small amount of market-share on older versions of Android, we find that for most developers the amount of effort required to support those versions is not worth the reward. For more info on the support of versions of Android earlier than 4.1 (in particular, Android 2.2 and Android 2.3), see the first edition of this book.

As Android releases new versions, the techniques you learn in this book will continue to work thanks to Android's backwards compatibility support (see Chapter 6 for details). We will keep track of changes at <http://forums.bignerdranch.com> and offer notes on using this book with the latest version.

The Necessary Tools

To get started with this book, you will need Android Studio. Android Studio is an integrated development environment used for Android development that is based off of the popular IntelliJ IDEA.

An install of Android Studio includes:

Android SDK

the latest version of the Android SDK

Android SDK tools and platform-tools

tools for debugging and testing your apps

A system image for the Android emulator

lets you create and test your apps on different virtual devices

As of this writing, Android Studio is under active development and is frequently updated. Be aware that you may find differences between your version of Android Studio and what you see in this book. Visit <http://forums.bignerdranch.com> for help with these differences.

Downloading and Installing Android Studio

Android Studio is available from Android's developer site at <https://developer.android.com/sdk/>.

If you do not already have it installed, you will need to install the Java Development Kit (JDK7), which you can download from <http://www.oracle.com>.

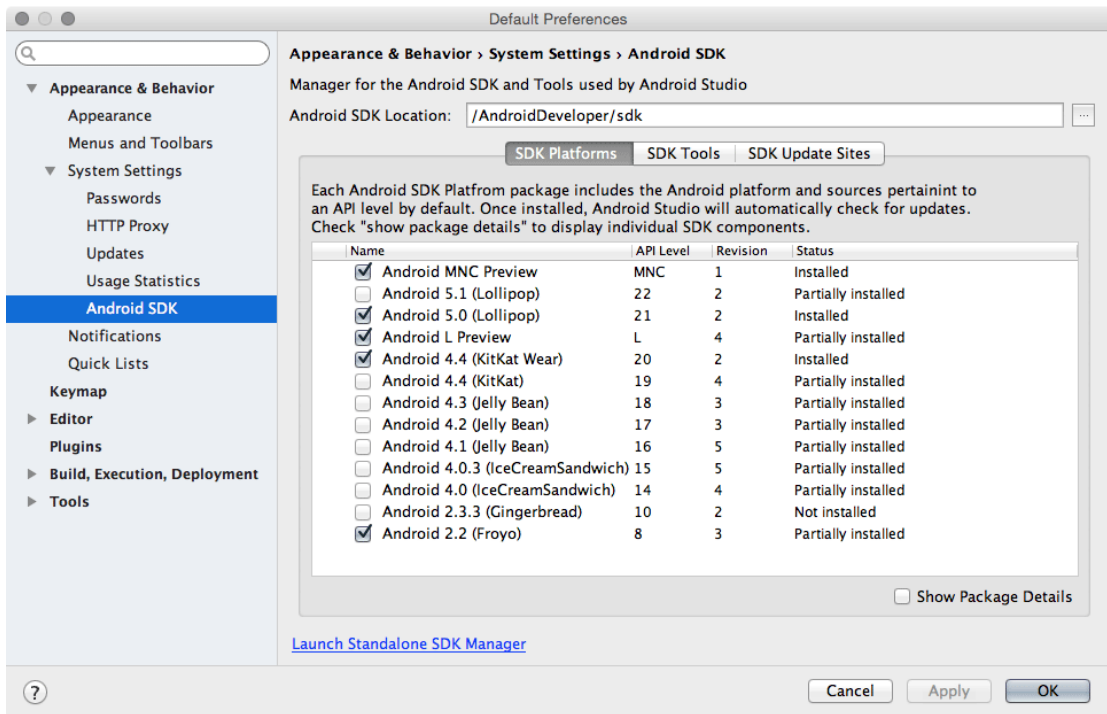
If you are still having problems, return to <https://developer.android.com/sdk/> for more information.

Downloading Earlier SDK Versions

Android Studio provides the SDK and the emulator system image from the latest platform. However, you may want to test your apps on earlier versions of Android.

You can get components for each platform using the Android SDK Manager. In Android Studio, select Tools → Android → SDK Manager. (You will only see the Tools menu if you have a project open. If you have not created a project yet, you can instead access the SDK Manager from the Android Setup Wizard screen. Under the Quick Start section, select Configure → SDK Manager, as shown in Figure 1.)

Figure 1 Android SDK Manager



Select and install each version of Android that you want to test with. Note that downloading these components may take a while.

The Android SDK Manager is also how to get Android's latest releases, like a new platform or an update of the tools.

An Alternative Emulator

The speed of the Android emulator has improved significantly over time and it is a reasonable way to run the code that you write in this book.

As an alternative, Genymotion is a popular, third-party Android emulator. You will occasionally see references to the Genymotion emulator in this book. For more information on Genymotion, visit <http://genymotion.com/>.

A Hardware Device

The emulator and Genymotion are useful for testing apps. However, they are no substitute for an actual Android device when measuring performance. If you have a hardware device, we recommend using that device at times when working through this book.

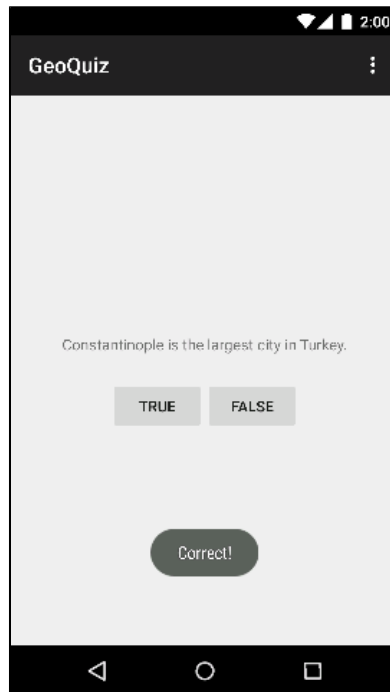
Your First Android Application

This first chapter is full of new concepts and moving parts required to build an Android application. It is OK if you do not understand everything by the end of this chapter. You will be revisiting these ideas again and in greater detail as you proceed through the book.

The application you are going to create is called GeoQuiz. GeoQuiz tests the user's knowledge of geography. The user presses True or False to answer the question on screen, and GeoQuiz provides instant feedback.

Figure 1.1 shows the result of a user pressing the False button:

Figure 1.1 (It's Istanbul, not Constantinople)



App Basics

Your GeoQuiz application will consist of an *activity* and a *layout*:

- An *activity* is an instance of **Activity**, a class in the Android SDK. An activity is responsible for managing user interaction with a screen of information.

You write subclasses of **Activity** to implement the functionality that your app requires. A simple application may need only one subclass; a complex application can have many.

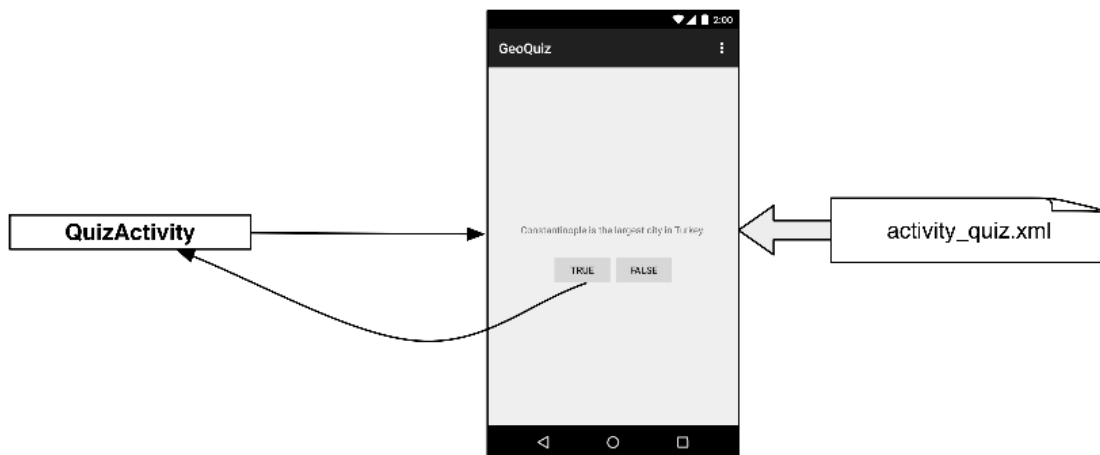
GeoQuiz is a simple app, so it will have a single **Activity** subclass named **QuizActivity**. **QuizActivity** will manage the user interface shown in Figure 1.1.

- A *layout* defines a set of user interface objects and their position on the screen. A layout is made up of definitions written in XML. Each definition is used to create an object that appears on screen, like a button or some text.

GeoQuiz will include a layout file named `activity_quiz.xml`. The XML in this file will define the user interface shown in Figure 1.1.

The relationship between **QuizActivity** and `activity_quiz.xml` is diagrammed in Figure 1.2.

Figure 1.2 **QuizActivity** manages what `activity_quiz.xml` defines



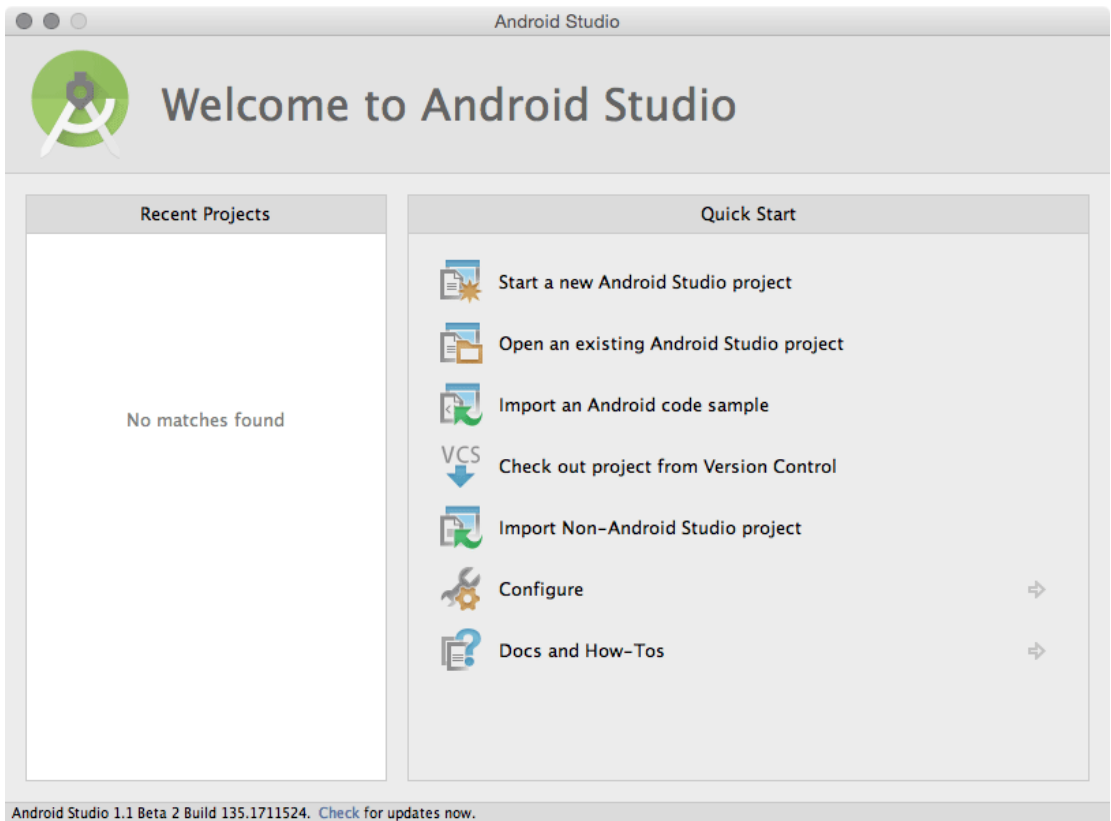
With those ideas in mind, let's build an app.

Creating an Android Project

The first step is to create an Android *project*. An Android project contains the files that make up an application. To create a new project, first open Android Studio.

If this is your first time running Android Studio, you will see the Welcome dialog, as in Figure 1.3.

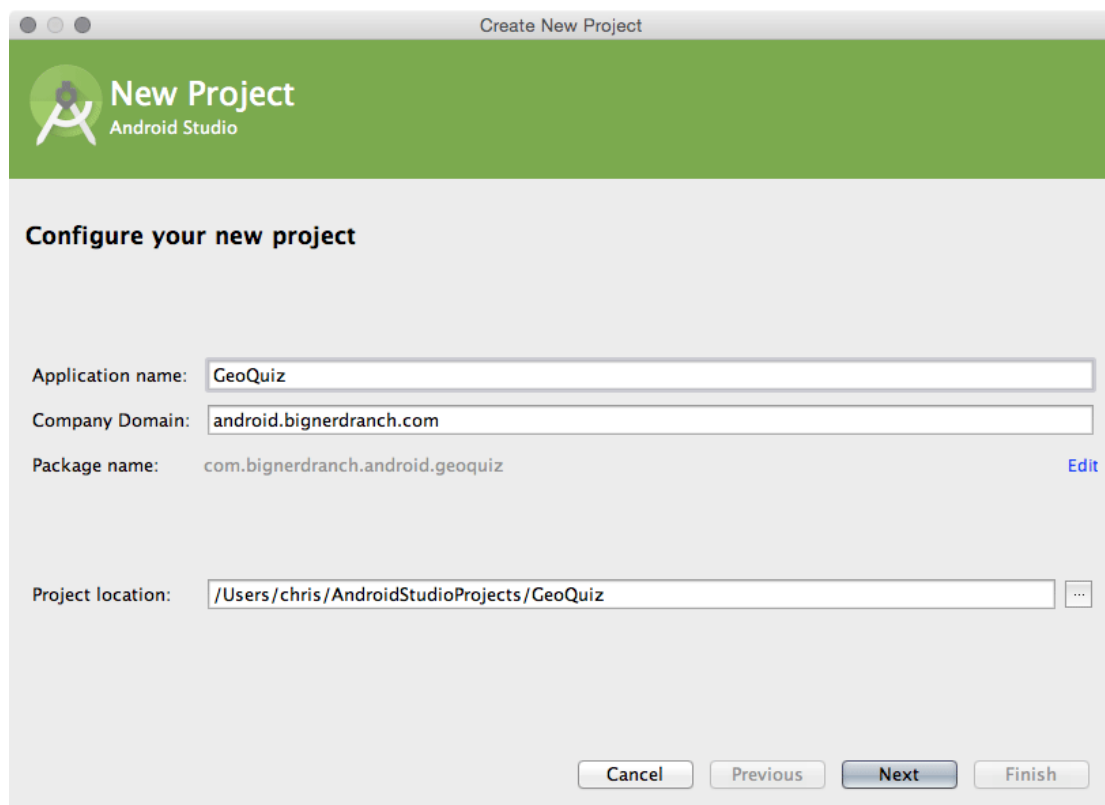
Figure 1.3 Welcome to Android Studio



From the dialog, choose Start a new Android Studio project. If you do not see the dialog, you may have created projects before. In this case, choose File → New Project...

You should see the new project wizard. In the first screen of the wizard, enter GeoQuiz as the application name (Figure 1.4). For the Company Domain, enter android.bignerdranch.com. As you do this, you will see the generated Package name change to com.bignerdranch.android.geoquiz. For the Project location, you can use any location on your filesystem that you want.

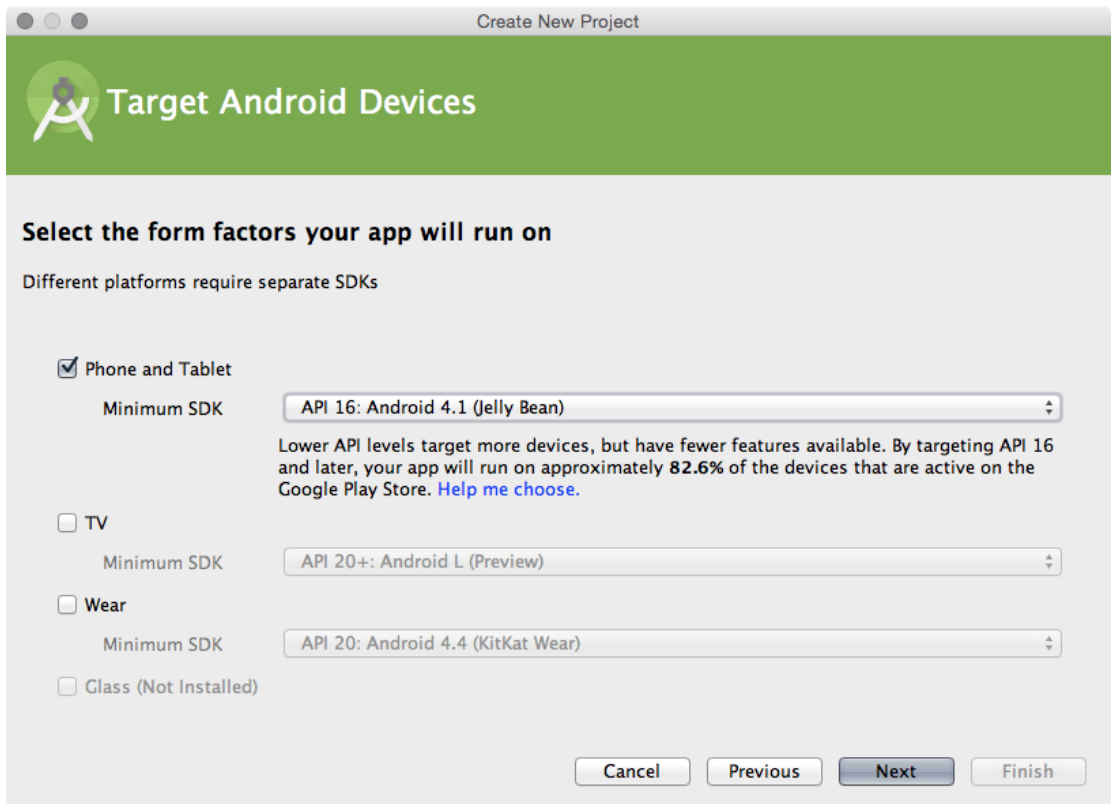
Figure 1.4 Creating a new application



Notice that the package name uses a “reverse DNS” convention in which the domain name of your organization is reversed and suffixed with further identifiers. This convention keeps package names unique and distinguishes applications from each other on a device and on Google Play.

Click Next. The next screen allows you to specify details about which devices you want to support. GeoQuiz will only support phones, so just check Phone and Tablet. Select a Minimum SDK version of API 16: Android 4.1 (Jelly Bean) (Figure 1.5). You will learn about the different versions of Android in Chapter 6.

Figure 1.5 Specifying device support

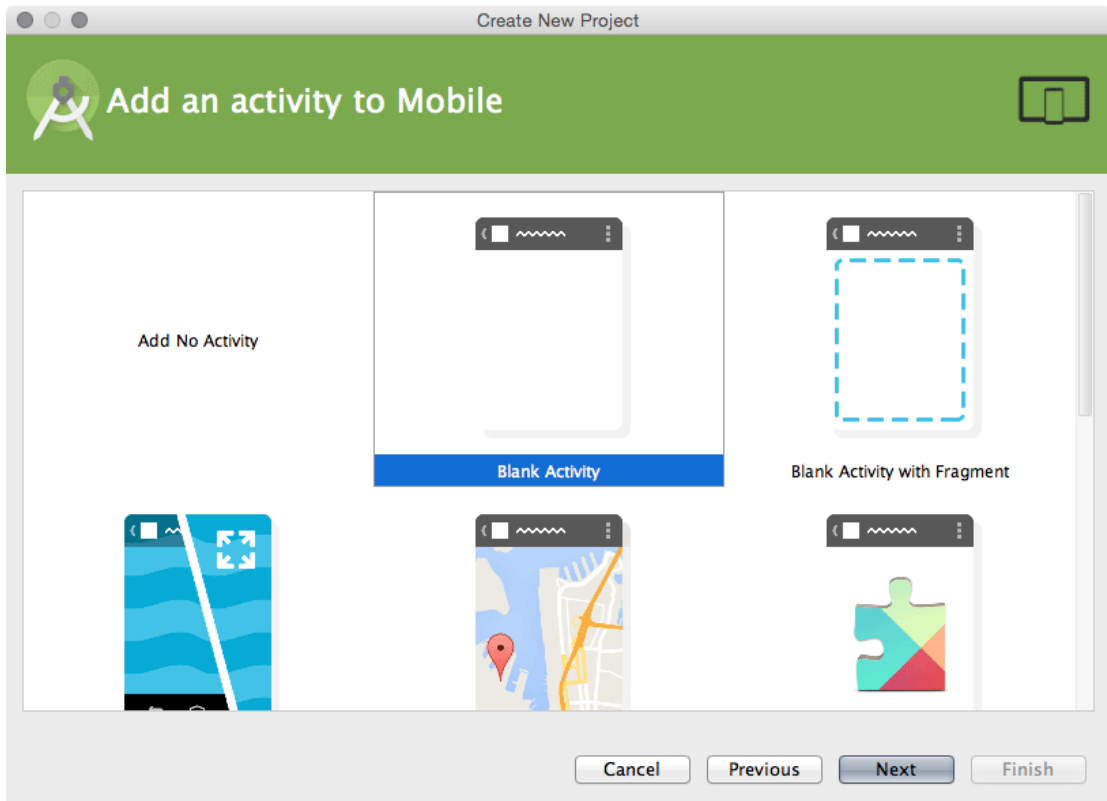


(Android Studio updates regularly, so your wizard may look slightly different from what we are showing you. This is usually not a problem; the choices should be similar. If your wizard looks very different, then the tools have changed more drastically. Do not panic. Head to this book's forum at forums.bignerdranch.com and we will help you navigate the latest version.)

Click Next.

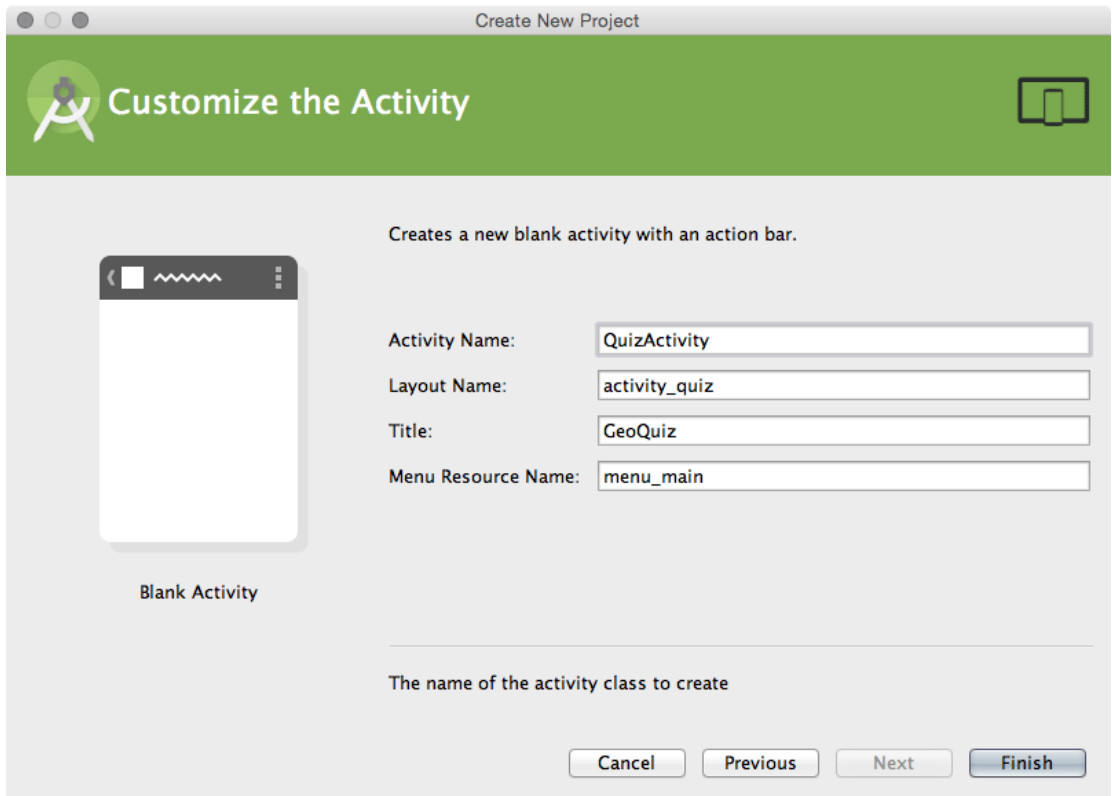
In the next screen, you are prompted to choose a template for the first screen of GeoQuiz (Figure 1.6). Choose Blank Activity and click Next.

Figure 1.6 Choosing a type of Activity



In the final dialog of this wizard, name the activity subclass **QuizActivity** (Figure 1.7). Notice the **Activity** suffix on the class name. This is not required, but it is an excellent convention to follow.

Figure 1.7 Configuring the new activity



The layout name will automatically update to `activity_quiz` to reflect the activity's new name. The layout name reverses the order of the activity name, is all lowercase, and has underscores between words. This naming style is recommended for layouts as well as other resources that you will learn about later.

For the Title, enter `GeoQuiz` to match the name of the app. Leave the Menu Resource Name as is and click Finish. Android Studio will create and open your new project.



Lituz.com

**To'liq qismini
Shu tugmani
bosish orqali
sotib oling!**