

TURING

图灵程序设计丛书

Big
nerd
ranch

- Amazon榜首畅销书全新升级
Android入门进阶不二之选
- 全面覆盖Android开发知识点
通过实战项目手把手教你逐步写Android应用

Android 编程权威指南

(第2版)

[美] Bill Phillips Chris Stewart Brian Hardy Kristin Marsicano 著
王明发 译

Android Programming
The Big Nerd Ranch Guide, Second Edition



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

Android 编程权威指南 (第2版)

[美] Bill Phillips Chris Stewart Brian Hardy Kristin Marsicano 著
王明发 译

Android Programming
The Big Nerd Ranch Guide, Second Edition

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Android编程权威指南 / (美) 菲利普斯
(Phillips, B.) 等著 ; 王明发译. -- 2版. -- 北京 :
人民邮电出版社, 2016.5
(图灵程序设计丛书)
ISBN 978-7-115-42246-0

I. ①A… II. ①菲… ②王… III. ①移动终端—应用
程序—程序设计 IV. ①TN929.53

中国版本图书馆CIP数据核字 (2016) 第083123号

内 容 提 要

Big Nerd Ranch 是美国一家专业的移动开发技术培训机构。本书主要以其 Android 训练营教学课程为基础, 融合了几位作者多年的心得体会, 是一本完全面向实战的 Android 编程权威指南。全书共 34 章, 详细介绍了 8 个 Android 应用。通过这些精心设计的应用, 读者可掌握很多重要的理论知识和开发技巧, 获得最前沿的开发经验。

如果你熟悉 Java 语言, 或者了解面向对象编程, 那就立刻开始 Android 编程之旅吧!

◆ 著 [美] Bill Phillips Chris Stewart Brian Hardy
Kristin Marsicano

译 王明发
责任编辑 朱 巍
执行编辑 杨 琳
责任印制 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷

◆ 开本: 800×1000 1/16
印张: 35.5
字数: 839千字 2016年5月第2版
印数: 24 001~28 000册 2016年5月北京第1次印刷
著作权合同登记号 图字: 01-2015-8300号

定价: 109.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广字第 8052 号

目 录

第 1 章 Android 开发初体验.....	1	2.10 挑战练习：从按钮到图标按钮.....	47
1.1 应用开发基础.....	1	第 3 章 Activity 的生命周期.....	49
1.2 创建 Android 项目.....	2	3.1 日志跟踪理解 Activity 生命周期.....	50
1.3 Android Studio 使用导航.....	6	3.1.1 输出日志信息.....	50
1.4 用户界面设计.....	7	3.1.2 使用 LogCat.....	52
1.4.1 视图层级结构.....	10	3.2 设备旋转与 Activity 生命周期.....	55
1.4.2 组件属性.....	11	3.3 设备旋转前保存数据.....	59
1.4.3 创建字符串资源.....	12	3.4 再探 Activity 生命周期.....	61
1.4.4 预览界面布局.....	12	3.5 深入学习：测试 onSaveInstanceState(Bundle)方法.....	62
1.5 从布局 XML 到视图对象.....	13	3.6 深入学习：日志记录的级别与方法.....	64
1.6 组件的实际应用.....	17	第 4 章 Android 应用的调试.....	65
1.6.1 引用组件.....	18	4.1 异常与栈跟踪.....	66
1.6.2 设置监听器.....	18	4.1.1 诊断应用异常.....	67
1.7 创建提示消息.....	20	4.1.2 记录栈跟踪日志.....	68
1.8 使用模拟器运行应用.....	23	4.1.3 设置断点.....	69
1.9 深入学习：Android 编译过程.....	25	4.1.4 使用异常断点.....	72
第 2 章 Android 与 MVC 设计模式.....	29	4.2 Android 特有的调试工具.....	73
2.1 创建新类.....	29	4.2.1 使用 Android Lint.....	73
2.2 Android 与 MVC 设计模式.....	32	4.2.2 R 类的问题.....	75
2.3 更新视图层.....	34	第 5 章 第二个 activity.....	76
2.4 更新控制层.....	36	5.1 创建第二个 activity.....	77
2.5 在设备上运行应用.....	40	5.1.1 创建新的 activity.....	78
2.5.1 连接设备.....	40	5.1.2 创建新的 activity 子类.....	81
2.5.2 配置设备用于应用开发.....	41	5.1.3 在 manifest 配置文件中声明 activity.....	81
2.6 添加图标资源.....	42	5.1.4 为 QuizActivity 添加 Cheat 按钮.....	82
2.6.1 向项目中添加资源.....	43	5.2 启动 activity.....	84
2.6.2 在 XML 文件中引用资源.....	45		
2.7 关于挑战练习.....	46		
2.8 挑战练习：为 TextView 添加监听器.....	46		
2.9 挑战练习：添加后退按钮.....	46		

5.3 activity 间的数据传递	86	7.7 采用 fragment 的应用架构	130
5.3.1 使用 intent extra	86	7.8 深入学习：为什么应优先使用 支持库版 fragment	131
5.3.2 从子 activity 获取返回结果	89	7.9 深入学习：使用操作系统内置版 fragment	131
5.4 activity 的使用与管理	94		
5.5 挑战练习	97		
第 6 章 Android SDK 版本与兼容	98	第 8 章 使用布局与组件创建用户界面	132
6.1 Android SDK 版本	98	8.1 升级 Crime 类	132
6.2 Android 编程与兼容性问题	99	8.2 更新布局	133
6.2.1 比较合理的版本	99	8.3 生成并使用组件	135
6.2.2 SDK 最低版本	101	8.4 深入探讨 XML 布局属性	136
6.2.3 SDK 目标版本	101	8.4.1 样式、主题及主题属性	136
6.2.4 SDK 编译版本	101	8.4.2 dp、sp 以及屏幕像素密度	137
6.2.5 安全添加新版本 API 中的 代码	101	8.4.3 Android 开发设计原则	138
6.3 使用 Android 开发者文档	104	8.4.4 布局参数	139
6.4 挑战练习：报告编译版本	106	8.4.5 边距与内边距	139
第 7 章 UI fragment 与 fragment 管理器	107	8.5 使用图形布局工具	140
7.1 UI 设计的灵活性需求	108	8.5.1 创建水平模式布局	141
7.2 fragment 的引入	108	8.5.2 添加新组件	142
7.3 着手开发 CriminalIntent	109	8.5.3 在属性视图中编辑组件属性	143
7.3.1 创建新项目	112	8.5.4 在框架视图中重新组织组件	144
7.3.2 fragment 与支持库	113	8.5.5 更新子组件的布局参数	145
7.3.3 在 Android Studio 中增加依赖 关系	114	8.5.6 android:layout_weight 属性 的工作原理	146
7.3.4 创建 Crime 类	117	8.5.7 图形布局工具使用总结	147
7.4 托管 UI fragment	118	8.5.8 组件 ID 与多种布局	148
7.4.1 fragment 的生命周期	118	8.6 挑战练习：日期格式化	148
7.4.2 托管的两种方式	119		
7.4.3 定义容器视图	119	第 9 章 使用 RecyclerView 显示列表	149
7.5 创建 UI fragment	120	9.1 升级 CriminalIntent 应用的 模型层	150
7.5.1 定义 CrimeFragment 的布局	121	9.2 使用抽象 activity 托管 fragment	153
7.5.2 创建 CrimeFragment 类	122	9.2.1 通用的 fragment 托管布局	153
7.6 添加 UI fragment 到 Fragment- Manager	125	9.2.2 抽象 activity 类	154
7.6.1 fragment 事务	126	9.3 RecyclerView、Adapter 和 ViewHolder	158
7.6.2 FragmentManager 与 fragment 生命周期	129	9.3.1 ViewHolder 和 Adapter	159
		9.3.2 使用 RecyclerView	161
		9.3.3 实现 Adapter 和 ViewHolder	163
		9.4 定制列表项	166

9.4.1 创建列表项布局.....	166	12.3.1 传递数据给 DatePicker- Fragment.....	203
9.4.2 使用定制列表项视图.....	168	12.3.2 返回数据给 Crime- Fragment.....	204
9.5 响应点击.....	170	12.4 挑战练习：更多对话框.....	211
9.6 深入学习：ListView 和 GridView.....	171	12.5 挑战练习：按设备类型展现 DialogFragment.....	212
9.7 深入学习：单例.....	171		
第 10 章 使用 fragment argument.....	173	第 13 章 工具栏.....	213
10.1 从 fragment 中启动 activity.....	173	13.1 AppCompat.....	213
10.1.1 附加 extra 信息.....	174	13.2 工具栏菜单.....	216
10.1.2 获取 extra 信息.....	175	13.2.1 在 XML 文件中定义菜单.....	217
10.1.3 使用 Crime 数据更新 CrimeFragment 视图.....	176	13.2.2 创建菜单.....	221
10.1.4 直接获取 extra 信息的缺点.....	177	13.2.3 响应菜单项选择.....	223
10.2 fragment argument.....	177	13.3 实现层级式导航.....	225
10.2.1 附加 argument 给 fragment.....	178	13.4 可选菜单项.....	226
10.2.2 获取 argument.....	179	13.4.1 切换菜单项标题.....	227
10.3 刷新显示列表项.....	180	13.4.2 “还有个问题”.....	229
10.4 通过 fragment 获取返回结果.....	182	13.5 深入学习：工具栏与操作栏.....	231
10.5 挑战练习：实现高效的 RecyclerView 刷新.....	183	13.6 挑战练习：删除 crime 记录.....	231
10.6 深入学习：为何要用 fragment argument.....	183	13.7 挑战练习：优化字符串资源显示.....	231
		13.8 挑战练习：用于 RecyclerView 的 空视图.....	232
第 11 章 使用 ViewPager.....	185	第 14 章 SQLite 数据库.....	233
11.1 创建 CrimePagerActivity.....	186	14.1 定义 Schema.....	233
11.1.1 ViewPager 与 PagerAdapter.....	187	14.2 创建初始数据库.....	234
11.1.2 整合并配置使用 CrimePagerActivity.....	188	14.3 修改 CrimeLab 类.....	238
11.2 FragmentStatePagerAdapter 与 FragmentPagerAdapter.....	190	14.4 写入数据库.....	239
11.3 深入学习：ViewPager 的工作原理.....	192	14.4.1 使用 ContentValues.....	239
11.4 深入学习：以代码的方式创建布局.....	193	14.4.2 插入和更新记录.....	240
第 12 章 对话框.....	194	14.5 读取数据库.....	242
12.1 使用 AppCompat 兼容库.....	195	14.5.1 使用 CursorWrapper.....	243
12.2 创建 DialogFragment.....	196	14.5.2 创建模型层对象.....	244
12.2.1 显示 DialogFragment.....	198	14.6 深入学习：数据库高级主题介绍.....	247
12.2.2 设置对话框的显示内容.....	199	14.7 深入学习：应用上下文.....	248
12.3 fragment 间的数据传递.....	202	14.8 挑战练习：删除 Crime 记录.....	248
		第 15 章 隐式 intent.....	249
		15.1 添加按钮组件.....	250

15.2	添加嫌疑人信息至模型层	252	18.7	深度学习：什么是 non-assets	308
15.3	使用格式化字符串	254	第 19 章	使用 SoundPool 播放音频	309
15.4	使用隐式 intent	255	19.1	创建 SoundPool	309
15.4.1	隐式 intent 的组成	255	19.2	加载音频文件	310
15.4.2	发送消息	256	19.3	播放音频	311
15.4.3	获取联系人信息	259	19.4	释放音频	313
15.4.4	检查可响应任务的 activity	263	19.5	设备旋转和对对象保存	314
15.5	挑战练习：ShareCompat	265	19.5.1	保留 fragment	315
15.6	挑战练习：又一个隐式 intent	265	19.5.2	旋转和已保留 fragment	316
第 16 章	使用 intent 拍照	266	19.6	深度学习：是否要保留	318
16.1	布置照片	266	19.7	深度学习：设备旋转处理再探	318
16.2	外部存储	269	第 20 章	样式与主题	321
16.3	使用相机 intent	272	20.1	颜色资源	321
16.3.1	外部存储使用权限	272	20.2	样式	322
16.3.2	触发拍照	273	20.3	主题	324
16.4	缩放和显示位图	274	20.4	添加主题颜色	327
16.5	功能声明	277	20.5	覆盖主题属性	328
16.6	深度学习：使用 include 标签	278	20.6	修改按钮属性	332
16.7	挑战练习：优化照片显示	278	20.7	深度学习：样式继承拾遗	334
16.8	挑战练习：优化缩略图加载	279	20.8	深度学习：引用主题属性	335
第 17 章	Master-Detail 用户界面	280	20.9	挑战练习：创建多版本主题	335
17.1	增加布局灵活性	281	第 21 章	XML drawable	336
17.1.1	修改 SingleFragment-Activity	282	21.1	统一按钮样式	337
17.1.2	创建包含两个 fragment 容器的布局	283	21.2	shape drawable	338
17.1.3	使用别名资源	284	21.3	state list drawable	340
17.1.4	创建平板设备专用可选资源	285	21.4	layer list drawable	341
17.2	Activity：fragment 的托管者	286	21.5	深度学习：为什么要用 XML drawable	342
17.3	深度学习：设备屏幕尺寸的确定	295	21.6	深度学习：使用 9-patch 图像	343
第 18 章	Assets	297	21.7	深度学习：使用 Mipmap 图像	347
18.1	为何使用 assets	297	第 22 章	深度学习 intent 和任务	348
18.2	创建 BeatBox 应用	298	22.1	创建 NerdLauncher 项目	348
18.3	导入 assets	301	22.2	解析隐式 intent	351
18.4	处理 assets	303	22.3	在运行时创建显式 intent	355
18.5	使用 Assets	305	22.4	任务与后退栈	357
18.6	访问 Assets	308	22.4.1	在任务间切换	357
			22.4.2	启动新任务	358

22.5 使用 NerdLauncher 应用作为设备 主屏幕.....	361
22.6 挑战练习：应用图标.....	362
22.7 深入学习：进程与任务.....	362
22.8 深入学习：并发文档.....	364
第 23 章 HTTP 与后台任务	367
23.1 创建 PhotoGallery 应用.....	368
23.2 网络连接基本.....	371
23.3 使用 AsyncTask 在后台线程上运行 代码.....	373
23.4 线程与主线程.....	374
23.5 从 Flickr 获取 JSON 数据.....	376
23.6 从 AsyncTask 回到主线程.....	383
23.7 清理 AsyncTask.....	386
23.8 深入学习：AsyncTask 再探.....	387
23.9 深入学习：AsyncTask 的替代方案.....	388
23.10 挑战练习：Gson.....	388
23.11 挑战练习：分页.....	388
23.12 挑战练习：动态调整网格列.....	389
第 24 章 Looper、Handler 和 HandlerThread	390
24.1 配置 RecyclerView 以显示图片.....	390
24.2 批量下载缩略图.....	393
24.3 与主线程通信.....	393
24.4 创建并启动后台线程.....	394
24.5 Message 与 message handler.....	396
24.5.1 消息的剖析.....	397
24.5.2 Handler 的剖析.....	397
24.5.3 使用 handler.....	398
24.5.4 传递 handler.....	402
24.6 深入学习：AsyncTask 与线程.....	407
24.7 挑战练习：预加载以及缓存.....	407
24.8 深入学习：解决图片下载问题.....	408
第 25 章 搜索	409
25.1 搜索 Flickr 网站.....	410
25.2 使用 SearchView.....	414
25.3 使用 shared preferences 实现轻量级 数据存储.....	419
25.4 优化应用.....	422
25.5 挑战练习：深度优化 PhotoGallery 应用.....	423
第 26 章 后台服务	424
26.1 创建 IntentService.....	424
26.2 服务的作用.....	427
26.3 查找最新返回结果.....	428
26.4 使用 AlarmManager 延迟运行服务.....	430
26.4.1 合理控制服务启动的频率.....	432
26.4.2 PendingIntent.....	433
26.4.3 使用 PendingIntent 管理 定时器.....	434
26.5 控制定时器.....	434
26.6 通知信息.....	437
26.7 挑战练习：可穿戴设备上的通知.....	439
26.8 深入学习：服务细节内容.....	440
26.8.1 服务的能与不能.....	440
26.8.2 服务的生命周期.....	440
26.8.3 non-sticky 服务.....	440
26.8.4 sticky 服务.....	441
26.8.5 绑定服务.....	441
26.9 深入学习：JobScheduler 和 JobService.....	442
26.10 深入学习：Sync Adapter.....	445
26.11 挑战练习：在 Lollipop 设备上 使用 JobService.....	446
第 27 章 broadcast intent	447
27.1 一般 intent 和 broadcast intent.....	447
27.2 接收系统 broadcast：重启后唤醒.....	448
27.2.1 standalone receiver.....	448
27.2.2 使用 receiver.....	450
27.3 过滤前台通知消息.....	452
27.3.1 发送 broadcast intent.....	452
27.3.2 动态 broadcast receiver.....	453
27.3.3 使用私有权限.....	455
27.3.4 使用有序 broadcast.....	458
27.4 receiver 与长时运行任务.....	462
27.5 深入学习：本地事件.....	462
27.5.1 使用 EventBus.....	463

27.5.2 使用 Rxjava.....	463	第 31 章 地理位置和 Play 服务.....	505
27.6 深入学习：检测 fragment 的状态.....	464	31.1 地理位置和定位类库.....	505
第 28 章 网页浏览.....	466	31.2 创建 Locatr 项目.....	506
28.1 最后一段 Flickr 数据.....	466	31.3 Play 服务和模拟器.....	507
28.2 简单方式：隐式 intent.....	469	31.4 创建 Locatr 应用.....	510
28.3 较难方式：使用 WebView.....	470	31.5 配置 Google Play 服务.....	512
28.4 处理 WebView 的设备旋转问题.....	476	31.6 使用 Google Play 服务.....	514
28.5 深入学习：注入 JavaScript 对象.....	477	31.7 基于地理位置的 Flickr 搜索.....	516
28.6 深入学习：KitKat 的 WebView.....	478	31.8 获取定位数据.....	517
28.7 挑战练习：使用后退键浏览历史 网页.....	478	31.9 寻找并显示图片.....	519
28.8 挑战练习：非 HTTP 链接支持.....	479	31.10 挑战练习：进度指示器.....	521
第 29 章 定制视图与触摸事件.....	480	第 32 章 使用地图.....	522
29.1 创建 DragAndDraw 项目.....	480	32.1 导入 Play 地图服务库.....	522
29.1.1 创建 DragAndDraw- Activity.....	481	32.2 Android 上的地图服务.....	522
29.1.2 创建 DragAndDraw- Fragment.....	481	32.3 地图 API 设置.....	523
29.2 创建定制视图.....	482	32.4 创建地图.....	525
29.3 处理触摸事件.....	484	32.5 获取更多地理位置数据.....	526
29.4 onDraw(...)方法内的图形绘制.....	488	32.6 使用地图.....	529
29.5 挑战练习：设备旋转问题.....	490	32.7 深入学习：团队开发和 API key.....	534
29.6 挑战练习：旋转矩形框.....	490	第 33 章 material design.....	536
第 30 章 属性动画.....	492	33.1 material surface.....	536
30.1 建立场景.....	492	33.1.1 elevation 和 Z 值.....	538
30.2 简单属性动画.....	495	33.1.2 state list animator.....	539
30.2.1 视图属性转换.....	498	33.2 动画工具.....	540
30.2.2 使用不同的 interpolator.....	500	33.2.1 circular reveal.....	541
30.2.3 色彩渐变.....	500	33.2.2 shared element transition.....	542
30.3 播放多个动画.....	502	33.3 新的视图组件.....	545
30.4 深入学习：其他动画 API.....	504	33.3.1 card.....	545
30.4.1 传统动画工具.....	504	33.3.2 floating action button.....	547
30.4.2 转场.....	504	33.3.3 snackbar.....	548
30.5 挑战练习.....	504	33.4 深入学习 material design.....	549
		第 34 章 编后语.....	550
		34.1 终极挑战.....	550
		34.2 关于我们.....	551
		34.3 致谢.....	551

第1章

Android开发初体验

1

本章介绍编写Android应用需掌握的一些新的概念和UI组件。学完本章，如果没能理解全部内容，也不必担心。后续章节还会涉及这些概念并有更加详细的讲解。

马上要编写的首个应用名为GeoQuiz，它能测试用户的地理知识。用户单击TRUE或FALSE按钮来回答屏幕上的问题，GeoQuiz可即时反馈答案正确与否。

图1-1显示了用户点击FALSE按钮的结果。

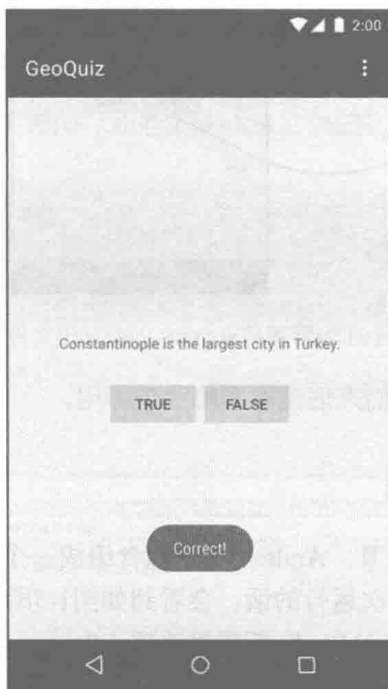


图1-1 正确答案应该是伊斯坦布尔（Istanbul），而不是君士坦丁堡

1.1 应用开发基础

GeoQuiz应用由一个activity和一个布局（layout）组成。

- activity是Android SDK中Activity类的一个具体实例，负责管理用户与信息屏的交互。应用的功能是通过编写一个个Activity子类来实现的。简单的应用可能只需一个子类，而复杂的应用则会有多个。

GeoQuiz是个简单应用，因此它只有一个名为QuizActivity的Activity子类。QuizActivity管理着图1-1所示的用户界面。

- 布局定义了一系列用户界面对象以及它们显示在屏幕上的位置。组成布局的定义保存在XML文件中。每个定义用来创建屏幕上的一个对象，如按钮或文本信息。

GeoQuiz应用包含一个名为activity_quiz.xml的布局文件。该布局文件中的XML标签定义了图1-1所示的用户界面。

QuizActivity与activity_quiz.xml文件的关系如图1-2所示。

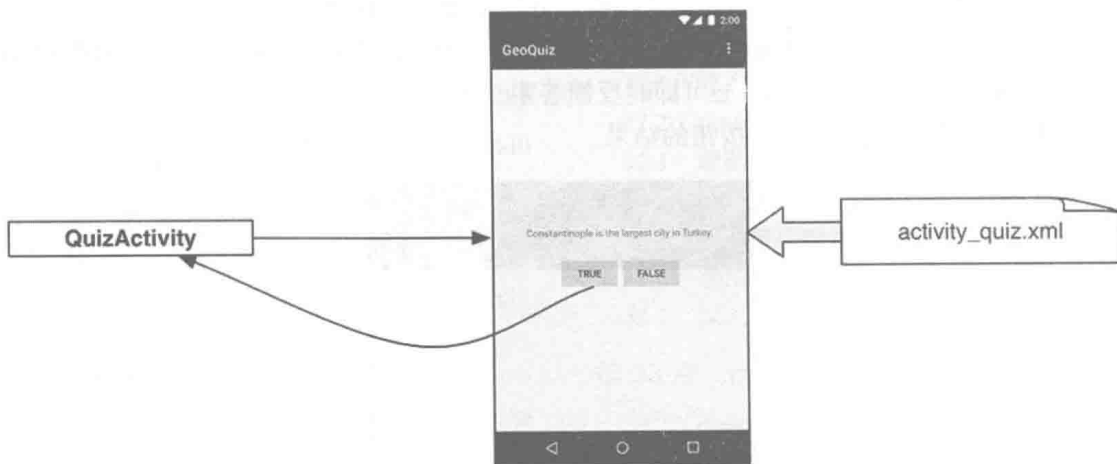


图1-2 QuizActivity管理着activity_quiz.xml文件定义的用户界面

学习了这些基本概念后，我们来创建本书第一个应用。

1.2 创建 Android 项目

首先我们创建一个Android项目。Android项目包含组成一个应用的全部文件。

启动Android Studio程序，首次运行的话，会看到如图1-3所示的欢迎对话框。

在欢迎界面，选择创建Android Studio新项目选项（Start a new Android Studio project）；非首次运行的话，选择File → New Project...菜单项即可。

现在，你应该打开了新建项目向导界面。在此界面的应用名称（Application name）处输入GeoQuiz，如图1-4所示。在公司域名（Company Domain）处输入android.bignerdranch.com。此时自动产生的包名称（Package name）会变为com.bignerdranch.android.geoquiz。至于项目存储位置（Project location），这就看个人喜好了。

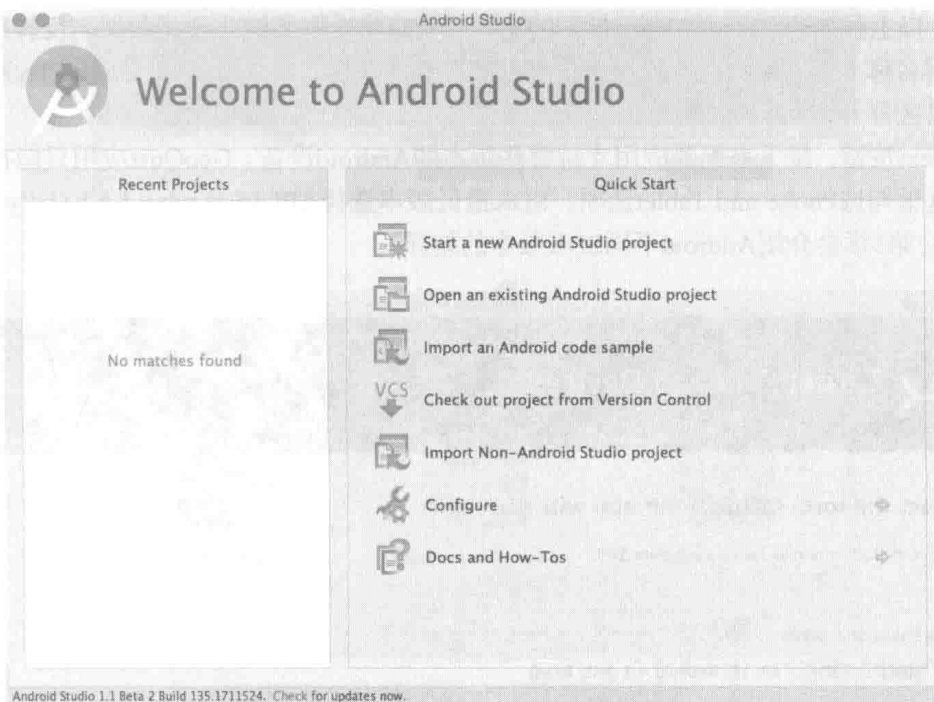


图1-3 欢迎来到Android Studio

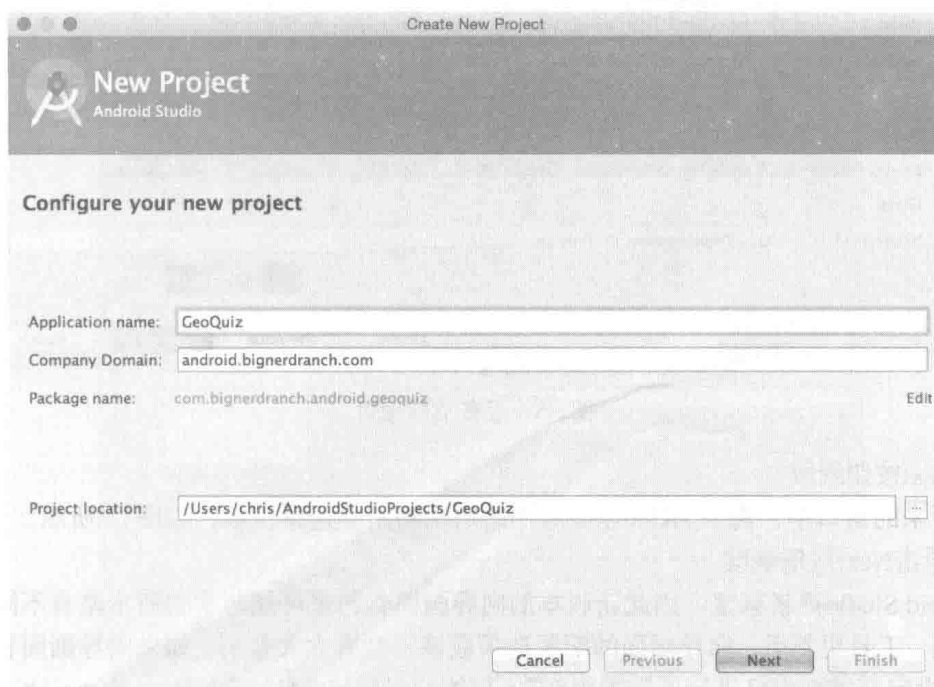


图1-4 创建新项目

注意，以上包名遵循了“DNS反转”约定，也就是将企业组织或公司的域名反转后，在尾部附加上应用名称。遵循此约定可以保证包名的唯一性，这样，同一设备和Google Play商店的各类应用就可以区分开来。

单击Next按钮，接下来配置应用支持哪些版本的Android设备。GeoQuiz应用只能在手机上运行，所以这里勾选Phone and Tablet选项。SDK最低版本选择API 16: Android 4.1 (Jelly Bean)，如图1-5所示。第6章会介绍Android不同SDK版本的差异。

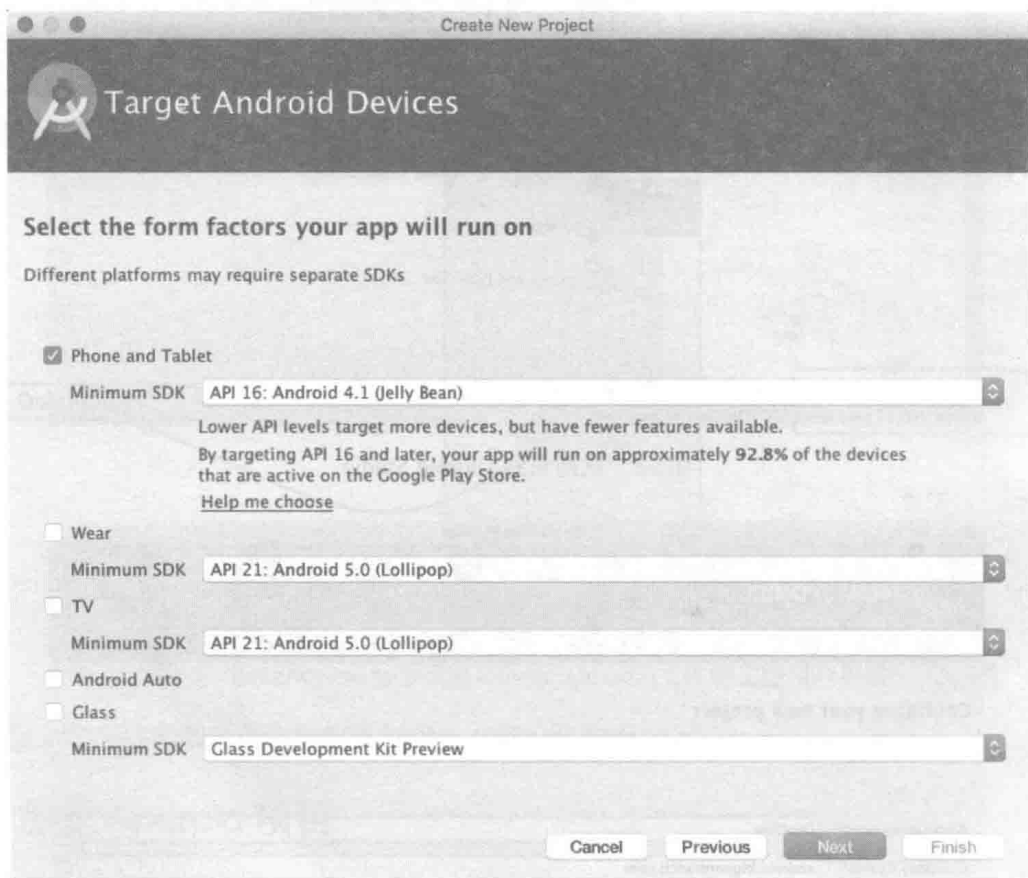


图1-5 设备支持配置

单击Next按钮继续。

在接下来的窗口中，需要为GeoQuiz应用的启动初始屏选择模板，如图1-6所示。选择Empty Activity后单击Next按钮继续。

（Android Studio更新频繁，因此新版本的向导画面看起来可能与本书所示略有不同。通常，这不是问题，工具更新后，向导画面的配置选项应该不会有太大差别。如果向导画面看起来大不相同，可以肯定开发工具已进行了重大更新。不要担心，请访问本书论坛<http://forums.bignerdranch.com>，获取最新版本开发工具的使用信息。）



图1-6 选择activity种类（空activity）

在应用向导的最后一个窗口，命名activity子类为QuizActivity，如图1-7所示。注意子类名的Activity后缀。尽管不是必需的，但我们建议遵循这种规范的命名约定。

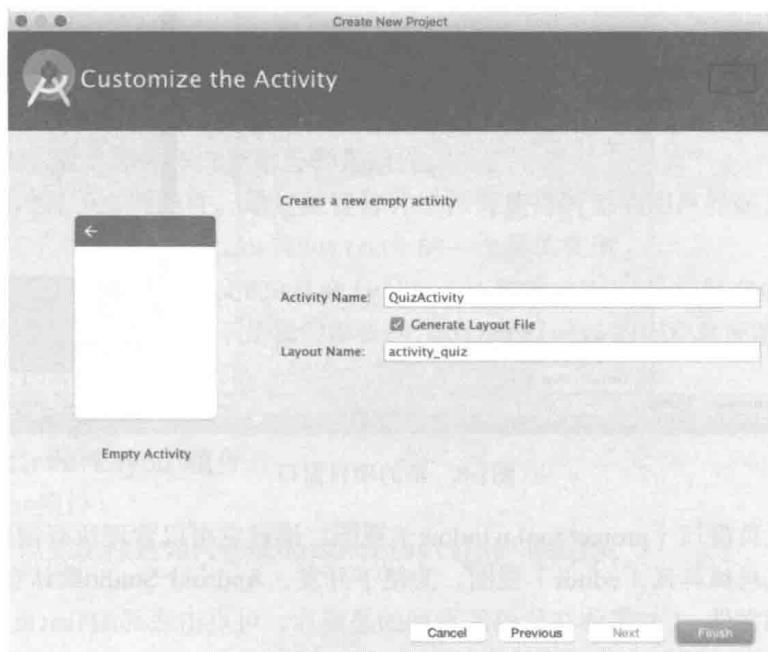


图1-7 配置新建的activity

保持Generate Layout File处于选中状态。为体现布局与activity间的对应关系，布局名（Layout Name）会自动更新为activity_quiz。布局的命名规则是：将activity名称的单词顺序颠倒过来并全部转换为小写字母，然后在单词间添加下划线。对于后续章节中的所有布局以及将要学习的其他资源，建议统一采用这种命名风格。

如果在你的Android Studio版本中还有其他选项，保持默认选择不变。单击Finish按钮，Android Studio会完成创建并打开新的项目。

1.3 Android Studio 使用导航

如图1-8所示，Android Studio已在工作区窗口里打开新建项目。整个工作区窗口分为不同的区域，这里统称为工具窗口（Tool Window）。

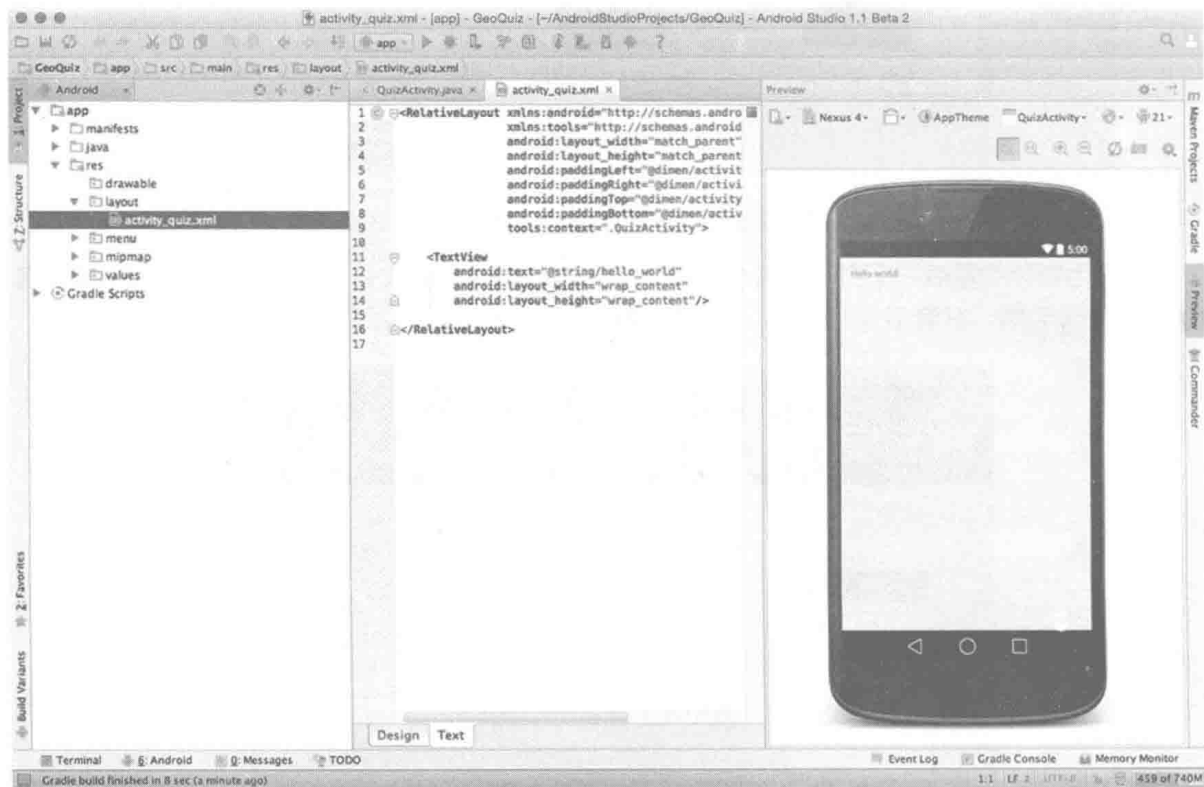


图1-8 新的项目窗口

左边是项目工具窗口（project tool window）视图，通过它可以管理所有项目相关的文件。

中间部分是代码编辑区（editor）视图。为便于开发，Android Studio默认在代码编辑区打开了activity_quiz.xml文件。（如果你在代码区看到的是图片，可点击底部的Text页切换。）当然，你也可以设置在右边窗口预览当前编辑的文件。

点击工作区窗口左边、右边以及底部标有各种名称的工具按钮区域，可显示或隐藏各类工具窗口。当然，也可以直接使用它们对应的快捷键。假如看不到某个工具按钮的话，可以点击左下角的灰色方形区域或单击View → Tool Buttons菜单项找到它。

1.4 用户界面设计

当前，activity_quiz.xml文件定义了默认的activity布局。应用默认的XML布局文件内容经常改变，但总是与代码清单1-1所示文件相似。

代码清单1-1 默认的activity布局（activity_quiz.xml）

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".QuizActivity">

    <TextView
        android:text="@string/hello_world"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

应用activity的默认布局定义了两个组件（widget）：RelativeLayout和TextView。

组件是用户界面的构造模块。组件可以显示文字或图像，与用户交互，甚至布置屏幕上的其他组件。按钮、文本输入控件和选择框等都是组件。

Android SDK内置了多种组件，通过配置各种组件可获得所需的用户界面及行为。每一个组件都是View类或其子类（如TextView或Button）的一个具体实例。

图1-9展示了代码清单1-1中定义的RelativeLayout和TextView是如何在屏幕上显示的。

不过，图1-9所示的默认组件并不是我们需要的，QuizActivity的用户界面需要下列五个组件：

- 一个垂直LinearLayout组件；
- 一个TextView组件；
- 一个水平LinearLayout组件；
- 两个Button组件。

图1-10展示了以上组件是如何构成QuizActivity用户界面的。

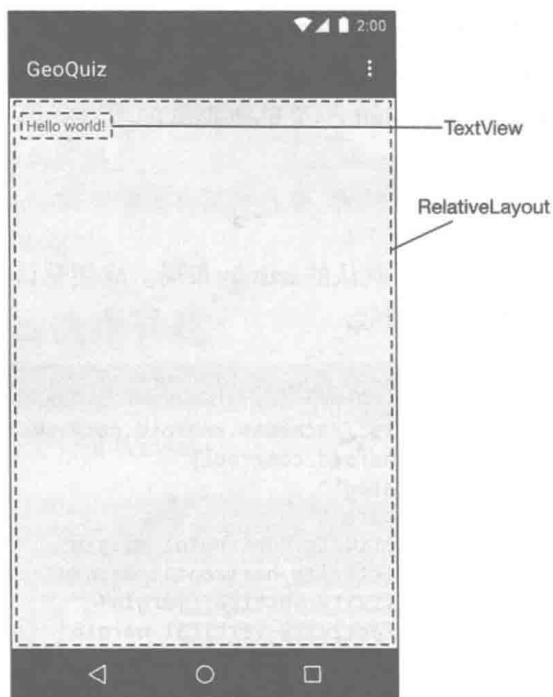


图1-9 显示在屏幕上的默认组件

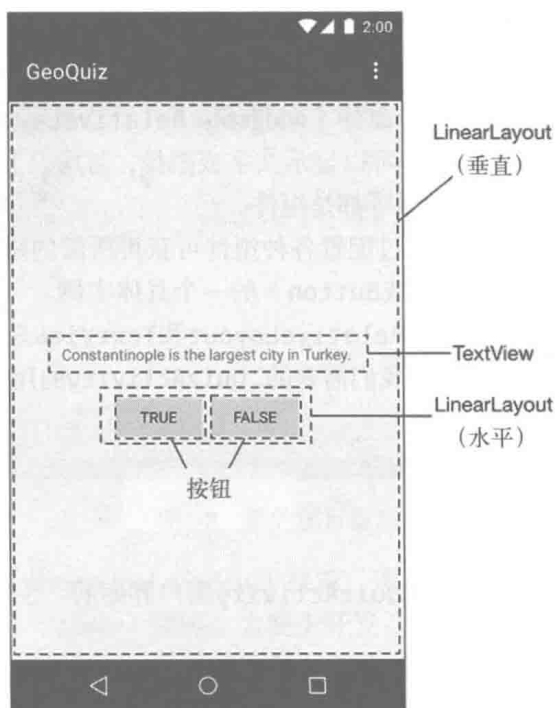


图1-10 布置并显示在屏幕上的组件