



Android

编程经典案例解析

(英文版)

Analysis of Classic
Programming Cases in Android

钟元生 高成珍 等 编著



清华大学出版社

Analysis of Classic
Programming Cases in Android

Android

编程经典案例解析 (英文版)

钟元生 高成珍 黄婧 高必梵 何英 吴微微 编著

清华大学出版社
北京

内 容 简 介

全书的 17 个 Android 编程案例,包括 TextView 特效、手机屏幕区域划分、我的课表、闪烁霓虹灯、简易计算器设计、页面滑动切换效果、图片定时滑动播放效果、搜索关键字提示、仿画廊视图效果、城市景点介绍、高校新闻等。为引导读者理解、掌握和灵活运用,编者通过图解分析、代码展示、技术剖析,由浅入深融会贯通。书中的例子稍加改动就可直接应用于实际的项目中。与本书配套的中文版教材经几千人试用,学习效果不错。

本书既可作为国内大学国际班的 Android 或 Java 类实践课程教材,也可供希望从事国际 APP 项目开发的程序员自学参考。掌握本书内容,计算机、软件工程专业的学生的就业机会将会大大增加。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Android 编程经典案例解析:英文/钟元生等编著.--北京:清华大学出版社,2016
ISBN 978-7-302-43409-2

I. ①A… II. ①钟… III. ①移动终端—应用程序—程序设计—英文 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2016)第 075238 号

责任编辑:袁勤勇 李 晔

封面设计:傅瑞学

责任校对:时翠兰

责任印制:杨 艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:16.75

字 数:382 千字

版 次:2016 年 7 月第 1 版

印 次:2016 年 7 月第 1 次印刷

印 数:1~1000

定 价:39.00 元

产品编号:066672-01

Preface by Authors

With the popularity of the Android phones and Android applications, the market demand for Android developer has grown rapidly. The lavish salary of Android programmer has attracted more and more students of colleges or universities to learn programming in Android. Many Chinese universities have now established a new course, Android Programming.

Therefore, we wrote a textbook, Android Application Development, based on the Android programming training handouts for helping college teachers in Jiangxi province to guide students to take part in the Mobile Application Software Design Contest. This book was published in January 2013 in Jiangxi College Press. The book was selected as textbook by many colleges and universities, such as Jiangxi Normal University, Jiangxi University of Finance and Economics, East China Institute of Technology University, Jiangxi Normal University of Science and Technology, University of Jinggangshan, Gannan Normal University, Jiujiang College; Jiangxi College of Applied Technology, Nanchang Institute of Technology, Jiangxi Vocational College of Environmental Engineering, etc. Besides Jiangxi province, there are several colleges in other provinces in China also selected this book as textbook of Android programming course, such as universities of Tianjin Sino-and-German Vocational and Technical College, School of Software at Xiamen University of Institute of Technology. At the same time, this book also has been selected as training book of Software College in Nanchang University, the Android training class at Nanchang Yiyou Company. The publication of this book, has promoted the process to start a new course of Android programming at some universities while attracted a number of users to study Android programming online.

Many teachers and learners think this book is very practical, suitable to learn how to program in Android vary from simple example to complex example step by step, suitable for self-study, easy to understand, especially suitable for teaching in classroom in the university. Many learners hope we could publish another book with commonly used cases which focus on the analysis of the function or effect that often used in actual Android App development to improve the reader's ability of developing Android App.

In the survey of job requirements related to Android programming, many enterprises have expressed the willing that hope to cooperate with us, so that we could help them to recruit Android App developers and test the skill of job applicants. Based on this, we have developed an online system for it and developed a set of architecture to test the user's programming skill in Android, including the primary, intermediate and advanced levels. In order to help the

examiner and examinee to clarify the contents and skills of this kind of test, we also provide some typical cases as a reference.

In order to improve student's ability to develop App used in Android Phone, to test whether the student master the basic skills, to help companies to recruit the Android developers who had certain project experience and can immediately participate in the project, we had carefully divided the professional skills and basic knowledge in a rational way, eventually designed and programmed a textbook named "*Analysis of Classic Programming Cases in Android*" combined with our own teaching experience in Colleges or universities and the actual Android project development experience. These cases based on the original knowledge of Android programming and inserted some functions and other running effects into them, mainly inspect whether the student is able to master the programming method of Android, and whether they have self-learning ability. The design of these cases is mainly considered the following aspects.

(1) Practical. Simulate the common functions and effects in Android application development;

(2) Comprehensive. Each case involves a number of knowledge points, the student need to use them flexible;

(3) Pay attention to the case analysis. There are many Android program source codes on the Internet, but lack of the detailed analysis of development process. Besides, these codes had less comment and coding style is very different. These cases are not so easy to use in readers' programming after downloaded them on the Internet that we pay attention to the detailed analysis of the case while designing and writing this book.

This book analyzes the development process of 17 typical Android cases in detail. Each case has higher practical value, and can be used in readers' development projects with little modify by learner. At the same time, it introduces the common errors and the methods for debugging program in Android development. The corresponding Android quizzes are provided. After studying this book, readers will have the ability to develop Android application by themselves.

The writing work of this book is arranged as follows. Gao Chengzhen served as the chief editor, he is responsible for the case selection and the writing of most of the chapters, Zhong Yuansheng served as joint chief editor, specifically responsible for writing guidance, design style, compiles, peer review and quality assurance. Division of labor of each chapter are as follows: Gao Chengzhen is responsible for the Chapter 5th, 6th, 7th, 8th, 9th, 10th, 11th, 12th, 13th, 14th, 15th, 16th, 17th and 18th, Zhong Yuansheng is responsible for Chapter 1st, 2nd and 19th, Gao Bifan is responsible for the Chapter 3rd, and He Ying is responsible for the Chapter 4th. Yang Xu, Zhang Wen, Chen Haijun, Wu Weiwei, Huang Jing, Cao Quan and others joined in other works such as the draft discussion, editing the book and developing the supported teaching courseware.

Due to the limitation of our abilities, there may be some defects or errors in the book. We hope you can give us your valuable suggestion when you find out errors or unsuitable contents.

The authors of Chinese book
October 2014

Preface

Mobile application development has attracted more and more students to program in Android or IOS in these years all over the world. Some of them in China may work in the companies in USA, UA or other developed countries serving as an App programmer. In the meantime, there is huge demand among companies outside China to recruit App programmers. The demand cannot be satisfied often in the developed countries because there are no many students to pursuit degree with computer science relative subjects. Some of professors in the universities in the developed countries also want to recruit graduate students major in computer science relative subjects. Thus, some Chinese students proficient in English are recruited by these companies or universities.

However, many good App programmer especial male college students are not proficient in English under current situation in China. These students interested in programming are often very good at programming. According to some experiencing example of professors or foreign companies, the programmer can fulfil the programming task very good in the foreign companies although their English proficiencies is on the ordinary level.

With the popularity of Android mobile application, there are great gap between demand and supply in the marketplace of mobile programmer in the developed countries. Under current situation, many companies cannot find out this kind of Chinese programmer with good programming skills but English proficiency. Thus, there is low possibility to recruit them as a programmer. This situation is the same for the professors in universities to recruit graduate student for their research project with mobile applications.

On the other hand, there are many college students outside China with good English proficiency interested in China because there is great development chance in China these years. This trend will be more and clearer with the development of China for a long time. Those students can often understand few of Chinese words. The students hope to communicate with some Chinese students and set up some kind of friendship relationship. Although there are other ways to fulfill it, we think a new platform with new technology contest among students all around the world will be more suitable one to some person. So, we launch a new competition, named Mobile Application Development Contest. This idea has been communicated with many professor or experts outside China. Many positive responses give us encouragement to make it become true.

The contest provides a way to attract students from colleges and universities worldwide, to learn skills and knowledge how to program on mobile phones, and help companies developing mobile software to get in touch with best mobile software student programmers. It will also promote to interested companies original mobile software applications developed by college students.

Two “tracks” of competition are planned. They will be devoted to Android Application Programming Skills and Mobile Application Software Development.

The first track named Application Programming skill Contest (Android only). The aim of track is to test programming skills necessary to develop intelligent Android applications. The testing will consist of a 3-hour test results of which will be submitted to the contest server. The contestant's score will consist of 40% for basic knowledge, and 60% for programming skills. The mark will be decided on the basis of correctness of submission and the total time of finishing the test. The answer will be blind reviewed by the experts designated by the Academic Committee.

The second track named Mobile Application Software Works Contest (No platform limitation, includes Android, iOS, or others). Applications developed for the competition must show creativity and practicality. There is no limit concerning the application area. The submission must include not only the executable program, but also the source code. Furthermore, the application design report, including PPT presentation and instruction manual must also be submitted. The complete set of files must be uploaded to the official competition website. Applications will be blind reviewed by the experts designated by the Academic Committee.

This book referenced another Chinese book authored by Gao Chengzhen, Zhong Yuansheng, He Ying and Gao Bifan published in January 2015. This book can use as a reference to this contest.

The translation work of this book is arranged as follows. Zhong Yuansheng are the head, he is responsible for the arrangement of translator, discussion of the translation style, translation quality control, overviewing and correcting the draft version, and the translation work of Chapter 1st, 2nd, 3rd and 4th. He also joints in the translation work of all other chapters and appendix. Gao Chengzhen is responsible for translation of the case code especially the electronic version of each Android project case in this book and translation of all relative graphs in Chinese to version in English, and translation of Chapter 15th, 16th and 17th. He Ying is responsible for the translation of Chapter 7th, 8th and 10th. Gao Bifan is responsible for the translation of Chapter 9th, 12th and 13th. Huang Jing is responsible for the translation of Chapter 5th, 6th, 11th and 14th. Wu Weiwei is responsible for the translation of Chapter 17th and appendix. Zhao Shenglun, Chen Haijun, and Ding Yu are involved in other translation works such as the draft discussion, editing the book and developing the supported teaching

courseware.

Due to the limitation of our abilities, there may be some defects or errors in the book. We hope you can give us your valuable suggestion when you find out errors or unsuitable contents.

The authors

August 2015

In this book, we assumed that you have got some basic knowledge of Android such as the basic use of Android Application and some common views. If you have no knowledge about the development of Android, it is better for you to learn our textbook *Android Application Development Guide* or series of videos we recorded for teaching how to develop Android application step by step on Android 4.1. The resource website is <http://www.XS096.com/book/>.

In this book, there are many cases, every case includes several source code files. We won't give you the program analysis of the example and only list some of the key code in the appendixes to focus on the key point and introduce more knowledge in the final pages. If you want to view all the codes, you can download these codes in our website: <http://www.XS096.com/book/> and run as Android application. You can get the effect in the textbook.

We strongly recommended that you complete the program by yourself based on the description and the key code listed in the book, while reading rather than directly copy the program to see the running result. Only when repeatedly failed several times, you can refer the codes.

In order to facilitate teaching, line number for each section of source code is shown and some key points for some of key phrases are given out. Example source code is shown in the Appendix.

```

1 // MainActivity extends AppCompatActivity
2 // MainActivity implements MainActivity
3 // MainActivity implements MainActivity
4 // MainActivity implements MainActivity
5 // MainActivity implements MainActivity
6 // MainActivity implements MainActivity
7 // MainActivity implements MainActivity
8 // MainActivity implements MainActivity
9 // MainActivity implements MainActivity
10 // MainActivity implements MainActivity
11 // MainActivity implements MainActivity
12 // MainActivity implements MainActivity
13 // MainActivity implements MainActivity
14 // MainActivity implements MainActivity
15 // MainActivity implements MainActivity
16 // MainActivity implements MainActivity
17 // MainActivity implements MainActivity
18 // MainActivity implements MainActivity
19 // MainActivity implements MainActivity
20 // MainActivity implements MainActivity
21 // MainActivity implements MainActivity
22 // MainActivity implements MainActivity
23 // MainActivity implements MainActivity
24 // MainActivity implements MainActivity
25 // MainActivity implements MainActivity
26 // MainActivity implements MainActivity
27 // MainActivity implements MainActivity
28 // MainActivity implements MainActivity
29 // MainActivity implements MainActivity
30 // MainActivity implements MainActivity
31 // MainActivity implements MainActivity
32 // MainActivity implements MainActivity
33 // MainActivity implements MainActivity
34 // MainActivity implements MainActivity
35 // MainActivity implements MainActivity
36 // MainActivity implements MainActivity
37 // MainActivity implements MainActivity
38 // MainActivity implements MainActivity
39 // MainActivity implements MainActivity
40 // MainActivity implements MainActivity
41 // MainActivity implements MainActivity
42 // MainActivity implements MainActivity
43 // MainActivity implements MainActivity
44 // MainActivity implements MainActivity
45 // MainActivity implements MainActivity
46 // MainActivity implements MainActivity
47 // MainActivity implements MainActivity
48 // MainActivity implements MainActivity
49 // MainActivity implements MainActivity
50 // MainActivity implements MainActivity
51 // MainActivity implements MainActivity
52 // MainActivity implements MainActivity
53 // MainActivity implements MainActivity
54 // MainActivity implements MainActivity
55 // MainActivity implements MainActivity
56 // MainActivity implements MainActivity
57 // MainActivity implements MainActivity
58 // MainActivity implements MainActivity
59 // MainActivity implements MainActivity
60 // MainActivity implements MainActivity
61 // MainActivity implements MainActivity
62 // MainActivity implements MainActivity
63 // MainActivity implements MainActivity
64 // MainActivity implements MainActivity
65 // MainActivity implements MainActivity
66 // MainActivity implements MainActivity
67 // MainActivity implements MainActivity
68 // MainActivity implements MainActivity
69 // MainActivity implements MainActivity
70 // MainActivity implements MainActivity
71 // MainActivity implements MainActivity
72 // MainActivity implements MainActivity
73 // MainActivity implements MainActivity
74 // MainActivity implements MainActivity
75 // MainActivity implements MainActivity
76 // MainActivity implements MainActivity
77 // MainActivity implements MainActivity
78 // MainActivity implements MainActivity
79 // MainActivity implements MainActivity
80 // MainActivity implements MainActivity
81 // MainActivity implements MainActivity
82 // MainActivity implements MainActivity
83 // MainActivity implements MainActivity
84 // MainActivity implements MainActivity
85 // MainActivity implements MainActivity
86 // MainActivity implements MainActivity
87 // MainActivity implements MainActivity
88 // MainActivity implements MainActivity
89 // MainActivity implements MainActivity
90 // MainActivity implements MainActivity
91 // MainActivity implements MainActivity
92 // MainActivity implements MainActivity
93 // MainActivity implements MainActivity
94 // MainActivity implements MainActivity
95 // MainActivity implements MainActivity
96 // MainActivity implements MainActivity
97 // MainActivity implements MainActivity
98 // MainActivity implements MainActivity
99 // MainActivity implements MainActivity
100 // MainActivity implements MainActivity

```

In the left is the line number, the right is the code (source code).

Reader Guide

In this book, we assumed that you have got some basic knowledge of Android such as the structure of Android Application and some common views. If you have no knowledge about Android application developing, it is better for you to learn our textbook *Android Application Development Guide* or series of videos we recorded for teaching how to develop Android application step by step on Android 4.1. The resources website is <http://www.XS360.cn/book/>.

In this book, there are many cases; every case includes several source code files. We concerned on the program analysis of the example and only list some of the key code in the context in order to focus on the key point and introduce more knowledge in the limit pages. If you want to view all the codes, you can download these codes in our website. Importing the code into your IDE and run as android application, you can get the effect in the textbook.

It is strongly recommended that you complete the program by yourself based on the interpretation, description and the key code listed in the book while reading rather than directly run the case program to see the running result. Only when repeatedly failed several times, you can view the codes.

In order to facilitate teaching, line number for each section of source code is shown and some comments for some of key phrases are given out. Example source code is shown in the following.

1	public class MainActivity extends Activity {
2	public void onCreate(Bundle savedInstanceState) {
3	super.onCreate(savedInstanceState);
	// invoked the same method in parent class
4	setContentView(R.layout.activity_main);
	// setting the layout of the Activity
5	}
6	public boolean onCreateOptionsMenu(Menu menu) {
	//Creating Option Menu
7	getMenuInflater().inflate(R.menu.activity_main, menu);
	// specify the menu resource
8	return true;
9	}
10	}

In the code, 1, 2, 3... in the left is the line number, the “super.onCreate (savedInstanceState);”

in the middle is the real content of the program code. The symbol "/" and the following content "invoked the same method in parent class" indicates the comment of the middle code.

In order to facilitate learning, communication, resources sharing, we developed a website to download the appropriate resources including source code, courseware, papers and so on. The URL is: <http://www.XS360.cn/book/>.

If you have any questions or any good suggestions during your study or reading the book, you can contact us by QQ group: 314753495 or email: 1281147324@qq.com.

In this book, we assumed that you have got some basic knowledge of Android such as the structure of Android Application and some common views. If you have no knowledge about Android application development, it is better for you to learn our textbook *Android Application Development Guide* or series of videos we recorded for teaching how to develop Android application step by step on Android 4.1. The resource website is <http://www.XS360.cn/book/>.

In this book, there are many cases, every case includes several source code files. We are based on the program analysis of the example and only list some of the key code in the context in order to focus on the key point and introduce more knowledge in the final pages. If you want to view all the codes, you can download them codes in our website, importing the code into your IDE and run as Android application, you can get the effect in the textbook.

It is strongly recommended that you complete the program by yourself based on the introduction, description and the key code listed in the book while reading rather than directly run the case program to see the running result. Only when repeatedly failed several times, you can view the codes.

In order to facilitate teaching, line number for each section of source code is shown and some comments for some of key phrases are given. All Example source code is shown in the following.

1	public class MainActivity extends Activity {
2	public void onCreate(Bundle savedInstanceState) {
3	super.onCreate(savedInstanceState);
4	invoked the same method in parent class
5	setContentView(R.layout.activity_main);
6	to setting the layout of the Activity
7	}
8	private boolean onCreateOptionsMenu() {
9	return true;
10	onOptionsItemSelected() {
11	return true;
12	}

In the code 1, 2, 3... in the left is the line number, the "super.onCreate(savedInstanceState)"

Contents

Chapter 1 Special TextView Effects >>>1

1.1 Case Overview.....	1
1.2 Key Code	1
1.3 Code Analysis	3
1.3.1 The effect of scrolling text in the TextView	3
1.3.2 Display various colors in the same text.....	4
1.3.3 Set picture orientation.....	4
1.3.4 Automatic link	5
1.4 Expansion of Knowledge.....	5
1.4.1 The difference between android: gravity and android: layout_gravity ...	5
1.4.2 The difference between android: padding and android: layout_margin	6
1.4.3 The representation of color in Android	6
1.5 Thinking and Exercises.....	7

Chapter 2 Phone Screen Division >>>8

2.1 Case Overview.....	8
2.2 Key Code	8
2.3 Code Analysis	10
2.3.1 LinearLayout	10
2.3.2 Proportionally split screen	10
2.4 Extension of Knowledge.....	11
2.5 Thinking and Exercises.....	11

Chapter 3 My Course Table-TableLayout >>>13

3.1 Case Overview.....	13
3.2 Key Code	13
3.3 Code Analysis	16
3.3.1 Class schedule interface analysis	16
3.3.2 TableLayout	17
3.3.3 Add borders to TextView.....	18

3.3.4	Definition of style.....	18
3.4	Expansion of Knowledge.....	19
3.5	Thinking and Exercises.....	19
Chapter 4 Images Around Text—RelativeLayout		>>>21
4.1	Case Overview.....	21
4.2	Key Code.....	21
4.3	Code Analysis.....	23
4.3.1	Interface analysis.....	23
4.3.2	RelativeLayout.....	23
4.4	Extension of Knowledge.....	24
4.5	Thinking and Practice.....	25
Chapter 5 Flashing Neon—FrameLayout		>>>26
5.1	Case Overview.....	26
5.2	Key Code.....	26
5.3	Code Analysis.....	29
5.3.1	Interface analysis.....	29
5.3.2	FrameLayout.....	29
5.3.3	The timer.....	29
5.3.4	Handler message passing.....	30
5.4	Extension of Knowledge.....	31
5.5	Thinking and Practice.....	31
Chapter 6 Design Calculator—Use Multiple Layout		>>>32
6.1	Case Overview.....	32
6.2	Key Code.....	32
6.3	Code Analysis.....	36
6.3.1	Interface analysis.....	36
6.3.2	Define style.....	37
6.4	Extension of Knowledge.....	37
6.5	Thinking and Practice.....	40
Chapter 7 Page Slide Show		>>>41
7.1	Case Overview.....	41
7.2	Key Code.....	42
7.3	Code Analysis.....	52
7.3.1	Interface analysis.....	52

7.3.2	ViewPager.....	52
7.4	Expansion of Knowledge.....	53
7.4.1	Event handler based on listening.....	53
7.4.2	Page full screen.....	55
7.5	Thinking and Exercises.....	56
Chapter 8 Images Switch Automatically >>>57		
8.1	Case Overview.....	57
8.2	Key Code	57
8.3	Code Analysis	63
8.3.1	Interface analysis	63
8.3.2	Custom MyImageTopView widget.....	63
8.4	Expansion of Knowledge.....	65
8.4.1	Custom widget.....	65
8.4.2	Gesture Detection	65
8.5	Thinking and Exercises.....	66
Chapter 9 Keyword Search Tips >>>67		
9.1	Case Overview.....	67
9.2	Key Code	67
9.3	Code Analysis	70
9.3.1	Smart tips to complete the input.....	70
9.3.2	Intelligent update the data source	71
9.4	Extension of Knowledge.....	72
9.4.1	ArrayAdapter	72
9.4.2	Dialog	72
9.5	Thinking and Exercises.....	73
Chapter 10 Simulate Gallery >>>75		
10.1	Case Overview.....	75
10.2	Key Code	76
10.3	Code Analysis	78
10.3.1	Interface analysis	78
10.3.2	ImageSwitcher introduction	79
10.4	Expansion of Knowledge.....	80
10.5	Thinking and Exercises	80

Chapter 11 Android Books List >>>82

11.1 Case Overview	82
11.2 Key Code	82
11.3 Code Analysis	88
11.3.1 Interface analysis	88
11.3.2 ListView	89
11.3.3 SimpleAdapter	90
11.3.4 ClipDrawable	91
11.4 Extension of Knowledge	92
11.4.1 The raw directory	92
11.4.2 Activity overview	93
11.5 Thinking and Practice	94

Chapter 12 BBC News—ListView Delay Load >>>96

12.1 Case Overview	96
12.2 Key Code	97
12.3 Code Analysis	102
12.3.1 ListView lazy loading principle	102
12.3.2 Introduction of SQLite database	103
12.4 Extension of Knowledge	107
12.5 Thinking and Exercises	107

Chapter 13 BBC News—Drop Down Refresh ListView >>>109

13.1 Case Overview	109
13.2 Key Code	110
13.3 Code Analysis	119
13.4 Extension of Knowledge	121
13.5 Thinking and Exercises	121

Chapter 14 ExpandableListView Widget >>>122

14.1 Case Overview	122
14.2 Key Code	122
14.3 Code Analysis	125
14.4 Extension of Knowledge	126
14.5 Thinking and Practice	129

Chapter 15 Product Category —Custom Multi-level List >>>130

15.1 Case Overview	130
--------------------------	-----

15.2	Key Code	131
15.3	Code Analysis	137
15.4	Extension of Knowledge	138
15.5	Thinking and Exercises	144
Chapter 16	College Introduction—TabHost	>>>145
16.1	Case Overview	145
16.2	Key Code	145
16.3	Code Analysis	153
16.3.1	TabHost introduction	153
16.3.2	Fragment introduction	155
16.3.3	Change the picture according to state	156
16.4	Extension of Knowledge	157
16.4.1	Communicating with the activity	157
16.4.2	Switch page through ActionBar	161
16.5	Thinking and Exercises	163
Chapter 17	The Sound of Music—Music Player	>>>164
17.1	Case Overview	164
17.2	Key Code	165
17.3	Code Analysis	198
17.3.1	Main functions of music player	198
17.3.2	ContentProvider	200
17.3.3	Service	201
17.3.4	BroadcastReceiver	203
17.4	Extension of Knowledge	205
17.4.1	MediaPlayer	205
17.4.2	Notifications	209
17.5	Thinking and Exercises	210
Appendix A	The Common Errors and Debugging Methods	>>>212
Appendix B	The Knowledge of Android Programming	>>>221
Appendix C	The Practice of Android Programming	>>>234
Appendix D	The Informal Test of Android Programming	>>>242

Chapter 1 Special TextView Effects

1.1 Case Overview

This case mainly describes the special effects about the TextView, such as scrolling the text, setting a couple of text colors in the same TextView, displaying pictures around the text, and automatically identifying various links in the text. when our application running, it looks like Figure 1-1. (The TextView program is operating in Figure 1-1)



Figure 1-1 the figure of the running results

1.2 Key Code

Layout file: 01\TextViewEffect\res\layout\activity_main.xml

1	<FrameLayout xmlns:android= "http://schemas.android.com/apk/res/android"
2	xmlns:tools="http://schemas.android.com/tools"