





Android 任务驱动式教程







设计用户界面



- 闪屏设计
- 工具栏设计
- 搜索面板设计
- 主菜单设计
- 软件信息设计
- 更新 AndroidManifest.xml

任务实现





Geogle

○ 界面布局

- 01 <?xml version="1.0" encoding="utf-8"?>
- 02 <LinearLayout
- 03 xmlns:android="http://schemas.android.com/apk/res/android"
- 04 android:layout width="fill parent"
- 05 android:layout height="fill parent"
- 06 android:gravity="center">
- 07 <ImageView android:src="@drawable/splash"
- 08 android:layout_width="wrap_content"
- 09 android:layout_height="wrap_content" />
- 10 </LinearLayout>



欢迎使用文件管理器

(c)njcit@msn.com 2011





○ 关键代码分析

● 全屏功能

09	<pre>public void onCreate(Bundle savedInstanceState) (</pre>
10	<pre>super.onCreate(savedInstanceState);</pre>
11	<pre>setFullScreen();</pre>
12	<pre>setContentView(R.layout.splash);</pre>
13	
14	<pre>startSplashTimer();</pre>
15	
16	}
17	
18	public void setFullScreen() (
19	requestWindowFeature(Window.FEATURE_NO_TITLE);
20	int flag = WindowManager.LayoutParams.FLAG_FULLSCREEN;
21	Window myWindow = this.getWindow();
22	<pre>myWindow.setFlags(flag, flag);</pre>
23	}





○ 关键代码分析

● 闪屏时钟

25	public void startSplashTimer() (
26	<pre>splashTimer = new Thread() (</pre>
27	public void run() (
28	try {
29	long ms = 0;
30	while (isSplashActive && ms < splashTime) {
31	sleep(100);
32	if (!isPaused)
33	ms += 100;
34	}
35	
36	<pre>startActivity(new Intent("com.google.app.CLEARSPLASH"));</pre>
37) catch (Exception ex) {
38	<pre>Log.e("Splash", ex.getMessage());</pre>
39) finally (
40	finish();
41	}
42)
43	};
44	<pre>splashTimer.start();</pre>
45	}





○ 关键代码分析

● 键盘响应

59	00verride	
60	public boolean onKeyDown(int keyCode, KeyEvent event) (
61	<pre>super.onKeyDown(keyCode, event);</pre>	
62	switch (keyCode) (
63	case KeyEvent.KEYCODE_DPAD_DOWN:	
64	case KeyEvent.KEYCODE_DPAD_LEFT:	
65	case KeyEvent.KEYCODE_DPAD_RIGHT:	
66	case KeyEvent.KEYCODE_DPAD_UP:	
67	case KeyEvent.KEYCODE_DPAD_CENTER:	
68	isSplashActive = false;	
69	break;	
70	case KeyEvent.KEYCODE_BACK:	
71	<pre>android.os.Process.killProcess(android.os.Process.myPid());</pre>	
72	break;	
73	default:	
74	break;	
75	}	
76	return true;	
77	}	







11	<linearlayout <="" android:id="0+id/toolbar" th=""></linearlayout>
12	android:layout_width="fill_parent"
13	android:layout_height="wrap_content"
14	android:orientation="horizontal"
15	android:background="@drawable/toolbar_background"
16	<imagebutton <="" android:id="0+id/home_button" td=""></imagebutton>
17	android:layout_width="60dip"
18	android:layout_height="60dip"
19	android:layout_weight="1"
20	android:layout_alignParentLeft="true"
21	android:layout_marginLeft="Odip"
22	android:src="@drawable/toolbar_home" />







○ 关键代码分析

● 按钮绑定

```
01 public void bindView() {
       homeButton = (ImageButton) findViewById(R.id.home button);
02
       viewButton = (ImageButton) findViewById(R.id.gridview button);
03
04
       searchButton = (ImageButton) findViewById(R.id.search button);
       upButton = (ImageButton) findViewById(R.id.up button);
05
       operateButton = (ImageButton) findViewById(R.id.operate button);
06
       setButton = (ImageButton) findViewById(R.id.settings button);
07
08
<u>09</u>
       homeButton.setOnClickListener(this);
10
       viewButton.setOnClickListener(this);
11
       searchButton.setOnClickListener(this);
12
       upButton.setOnClickListener(this);
13
       operateButton.setOnClickListener(this);
14
       setButton.setOnClickListener(this);
15 }
```





○ 关键代码分析

● 按钮事件

01	0 Override	
02	public void onClick(View v) (
03	switch (v. <u>getId()</u>) (
04	case R.id.home_button:	
05	if (isHomeDirctory) {	
06	homeButton. <u>setImageResourc</u> e(R.drawable.toolbar_favorite);	
07	isHomeDirctory = false;	
08) else (
09	homeButton.setImageResource(R.drawable.toolbar_home);	
10	isHomeDirctory = true;	
11	}	
12		
13	break;	



搜索面板设计



- 搜索面板
 - Spinner



● 自定义搜索
 ■ EditText





搜索面板设计

○ 关键代码分析

● 搜索面板显示 / 隐藏动画

- 01 <?xml version="1.0" encoding="utf-8"?>
- 02 <set xmlns:android="http://schemas.android.com/apk/res/android">
- 03 <translate
- 04 android:fromYDelta="-100%"
- 05 android:toYDelta="0%"
- 06 android:duration="1000"
- 07 />
- 08 </set>

11 Animation searchPanelIn = AnimationUtils.loadAnimation(this,

- 12 R.anim.search_panel_in);
- 13





关键代码分析 显示搜索面板

14	if (rearchPanel == null) {			
15	rearchPanel = layoutInflater.inflate(R.layout.search_panel_layout,			
16	null);			
17	rearchPanel.startAnimation(searchPanelIn);			
18	<pre>relativeLayout.addView(rearchPanel);</pre>			
19	} else (
20	if (!isShowSearchPanel) {			
21	<pre>rearchPanel.setVisibility(View.VISIBLE);</pre>			
22	<pre>rearchPanel.startAnimation(searchPanelIn);</pre>			
23)			
24	}			





○ 关键代码分析◎ Spinner 绑定

26	searchButtonofPanel = (Button) rearchPanel
27	.findViewById(R.id.search_btnofpanel);
28	<pre>searchButtonofPanel.setOnClickListener(listener_search);</pre>
29	
30	<pre>typeSpinner = (Spinner) rearchPanel.findViewById(R.id.selecttype);</pre>
31	typeAdapter = new ArrayAdapter <string>(this,</string>
32	android.R.layout.simple_spinner_item, type);
33	typeAdapter.setDropDownViewResource(
34	android.R.layout.simple_spinner_dropdown_item);
35	typeSpinner.setAdapter(typeAdapter);
36	typeSpinner.setOnItemSelectedListener(
37	new Spinner.OnItemSelectedListener() {
38	<pre>public void onItemSelected(AdapterView<?> arg0, View arg1,</pre>
39	int arg2, long arg3) (
40	<pre>currentSearchType = type[arg2];</pre>
41	}
42	
43	<pre>public void onNothingSelected(AdapterView<?> arg0) (</pre>
44)
45));





○ 资源 menu.xml

01 <?xml version="1.0" encoding="utf-8"?> 02 <menu xmlns:android="http://schemas.android.com/apk/res/android"> 03 <item android:id="@+id/add" android:title="@string/add" 04 05 android:icon="@drawable/add"/> <item android:id="0+id/settings" 06 android:title="@string/settings" 07 android:icon="@drawable/settings"/> 08 <item android:id="0+id/about" 09 android:title="@string/about" 10 android:icon="@drawable/about"/> 11 12 <item android:id="0+id/exit" android:title="@string/exit" 13 14 android:icon="@drawable/exit"/> 15 </menu>







关键代码分析 创建主菜单

- 01 @Override
- 02 public boolean onCreateOptionsMenu(Menu menu) {
- 03 super.onCreateOptionsMenu(menu);
- 04 MenuInflater inflater = getMenuInflater();
- 05 inflater.inflate(R.menu.menu, menu);
- 06 return true;
- 07 }





○ 关键代码分析 ● 主菜单响应

01	00verride	
02	<pre>public boolean onOptionsItemSelected(MenuItem item) (</pre>	
03	3 super.onOptionsItemSelected(item);	
04	<pre>switch (item.getItemId()) (</pre>	
05	case R.id.add:	
06		
07	break;	
08	case R.id.settings:	
09		
10	break;	
11	case R.id.about:	
12	<pre>Intent i = new Intent(this, About.class);</pre>	
13	<pre>startActivity(i);</pre>	
14	break;	
15	case R.id.exit:	
16	<pre>isExit();</pre>	
17	break;	
18	default:	
19	break;	
20)	
21	return true;	
22		
23	}	



更新 Android Manifest.xml



15	<activity <="" android:name=".MainActivity" th=""></activity>
16	android:label="@string/app_name">
17	<intent-filter></intent-filter>
18	<action android:name="com.google.app.CLEARSPLASH"></action>
19	
20	<category android:name="android.intent.category.DEFAULT"></category>
21	
22	
23	
24	<activity <="" android:name=".About" td=""></activity>
25	android:label="@string/about"
26	android:theme="@android:style/Theme.Dialog">
27	







- 🖤 资源文件的引用
- ᠮ Style 和 Theme
- 常用组件介绍
- 🗹 菜单设计
- 🖤 对话框设计







○ 常见布局

- 线性布局
- 绝对布局
- ◎ 相对布局
- 单帧布局
- 表格布局
- 列表布局
- ◎ 网格布局
- 标签布局







ViewGroup 以线性方向显示它的子视图元素,即后一个元素垂直或水平显示在上一个子元素之后。

布局管理

- 重要属性
 - orientation
 - layout_weight
 - gravity





○ 线性布局 ◎ 示例

Views/Layouts/LinearL	🏭 📶 📧 7:01 ам .ayout/10. Backgrou
From:	
То:	







ViewGroup 以绝对方式显示它的子视图元素,即以坐标的方式来定位子视图在屏幕上位置。屏幕的左上角为坐标O(0,0)。

布局管理

● 重要属性

android:layout_x

android:layout_y





○ 绝对布局
 ◎ 示例



I love Goolge I love the world <?xml version="1.0" encoding="utf-8"?> <AbsoluteLayout

xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="wrap_content"
android:layout_height="wrap_content">
<ImageView</pre>

android:layout_width="wrap_content" android:layout_height="wrap_content" android:src="@drawable/google" android:layout_x="22px" android:layout_y="60px"/>

<TextView

android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="I love Android" android:layout_x="100px" android:layout_y="175px"/>

<TextView

android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="I love Goolge" android:layout_x="120px" android:layout_y="195px"/>

<TextView

android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="I love the world"
android:layout_x="140px"
android:layout_y="215px"/>
</ AbsoluteLayout>





ViewGroup以相对位置显示它的子视图元素,一个视图可以指定相对于它的兄弟视图的位置(例如在给定视图的左边或者下面)或相对于 RelativeLayout 的特定区域的位置。

布局管理

- 重要属性
 - layout_below
 - Iayout_toLeftOf
 - layout_alignTop
 - Iayout_centerHrizontal





○ 相对布局 ◎ 示例



<?xml version="1.0" encoding="utf-8"?> <RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent">
<AnalogClock</pre>

android:id="@+id/aclock"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerInParent="true"/>
<DigitalClock</pre>

android:id="@+id/dclock"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/aclock"
android:layout_alignLeft="@id/aclock"
android:layout_marginLeft="40px"/>
<TextView</pre>

android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="当前时间:" android:layout_toLeftOf="@id/dclock" android:layout_alignTop="@id/aclock"/> </RelativeLayout>





 ViewGroup 将它的子视图元素都放在布局的左上角, 并覆盖在前一子控件的上层。如果绝对布局不指定坐标 位置,子视图将会类似于单帧布局那样层叠放置。

布局管理





 ViewGroup 以表格显示它的子视图元素,即行和列标 识一个视图的位置。Android 的表格布局和 HTML 中 的表格布局非常类似,TableRow 就像 HTML 表格的 标记。

布局管理

- 重要属性
 - Iayout_column
 - layout_span
 - collapseColumns
 - stretchColumns





● あ格布局
 ● 示例
 ● 示例
 ● 6:58 AM
 Views/Layouts/TableLayout/10. Simple Form
 User
 Password
 Login

xml version="1.0" encoding="utf-8"?
<tablelayout <="" td="" xmlns:android="http://schemas.android.com/apk/res/android"></tablelayout>
android:layout_width="match_parent"
android:layout_height="match_parent"
android:stretchColumns="1">
<tablerow></tablerow>
<textview <="" android:text="@string/ user" td=""></textview>
android:textStyle="bold" android:gravity="right"
android:padding="3dip" />
<edittext <="" android:id="@+id/username" td=""></edittext>
android:text="@string/username_text"
android:padding="3dip"
android:scrollHorizontally="true" />
<tablerow></tablerow>
<textview <="" android:text="@string/password" td=""></textview>
android:textStyle="bold" android:gravity="right"
android:padding="3dip" />
<edittext <="" android:id="@+id/password" td=""></edittext>
android:text="@string/password_text"
android:password="true"
android:padding="3dip"
android:scrollHorizontally="true" />
<tablerow .<="" td=""></tablerow>
android:gravity="right">
<button <="" android:id="@+id/cancel" td=""></button>
android:text="@string/cancel" />
<button <="" android:id="@+id/login" td=""></button>
android:text="@string/login" />





○ 其他布局











- o id
- Iayout_width/layout_height
- padding
- Iayout_marginBottom
- Iayout_gravity
- ◎值

px (pixels) sp (scaled pixels) in (inches)

dip (device independent pixels) pt (points) mm (millimeters)

布局管理







```
public void showSearchPanel() {
    if (layoutInflater == null) {
        layoutInflater = (LayoutInflater) getSystemService(LAYOUT INFLATER SERVICE);
    }
    if (relativeLayout == null) {
        relativeLayout = (RelativeLayout) findViewById(R.id.main layout);
    }
    Animation searchPanelIn = AnimationUtils.loadAnimation(this,
            R.anim.search panel in);
    if (rearchPanel == null) {
        rearchPanel = layoutInflater.inflate(R.layout.search panel layout,
                null):
        rearchPanel.startAnimation(searchPanelIn);
        relativeLayout.addView(rearchPanel);
    } else {
        if (!isShowSearchPanel) {
            rearchPanel.setVisibility(View.VISIBLE);
            rearchPanel.startAnimation(searchPanelIn);
        - }
    }
```



资源文件的引用



● 在 xml 中引用

@string/string_name

◎ 在代码中引用

R.string.string_name

● 在代码中获取字符串:

getResources().getString(R.string.string_name) getResources().getText (R.string.string_name)



○ 图片资源

● 在 xml 中引用

@drawable/image_name

◎ 在代码中引用

R.drawable.image_name

在代码中获取字符串 getResources().getDrawable(R.dra wable.image_name)

getResources().openRawResource(R.drawable.image_name)

资源文件的引用



资源文件的引用



● 在 xml 中引用

@color/color_name

- 在代码中引用 R.string.color_name
- 在代码中获取字符串 .color_name)

getResources().getColor(R.string



○ 系统资源

● 字符串

android.R.string.httpErrorBadUrl @android:string/httpErrorBadUrl 资源文件的引用

• 颜色

android.R.darker_gray android.graphics.Color.darker_gray @android:color/darker_gray



资源文件的引用



◎ 图片

android.R.drawable.ic_media_pause @android:drawable/ic_media_pause

◎ 动画

android.R.anim.fade_out

◎ 布局

android.R.layout.simple_dropdown_item_1line



Style

 Style 是一个包含一种或者多种格式化属性的集合,针 对的是窗体中的元素,可以将其作为一个单位用在布局 xml 的单个元素当中,用于改变指定控件或者 Layout 的样式。

Style 和 Theme





● 步骤

- 在 res/values 目录下新建一个名叫 style.xml 的文件;
- 使用 <style> 元素并结合一个全局唯一的名字设计一个风格或 主题,通过这个名字来应用风格,而可选择的父类属性标识了 当前风格是继承于哪个风格;

Style 和 Theme

- 在 <style> 元素内部,申明一个或者多个 <item>,每个 <item
 > 定义了一个名字属性,并且在元素内部定义了这个风格的值;
- 在 xml 布局中使用定义的 Style 资源。



Style 和 Theme



■定义

01	xml version="1.0" encoding="UTF-8"?	
02	<resources></resources>	
03	<style na<="" td=""><td>me="SpecialText" parent="@style/Text"></td></tr><tr><td>04</td><td><item</td><td>name="android:textSize">18sp</item></td></tr><tr><td>05</td><td><item</td><td>name="android:textColor">#008</item></td></tr><tr><td>06</td><td><item</td><td>name="android:shadowColor">@android:color/black</item></td></tr><tr><td>07</td><td><item</td><td>name="android:shadowRadius">2.0</item></td></tr><tr><td>08</td><td></style>	
09		

01 <TextView style="0style/SpecialText"

- 02 android:layout_width="fill_parent"
- 03 android:layout_height="wrap_content"
- 04 android:text="@string/helloandroid" />





● 步骤

■ 在 res/values 目录下新建一个名叫 theme.xml 的文件。

Style 和 Theme

■ 和 Style 设计类似,设计一些样式属性。

■ 在 Java 代码或 AndroidManifest.xml 中使用 Theme。



Style 和 Theme

Style ◎ 示例

■定义

01	xml version="1.0" encoding="UTF-8"?
02	<resources></resources>
03	<style name="Theme" parent="android:Theme.Light"></td></tr><tr><td>04</td><td><item name="android:windowFullscreen">true</item></td></tr><tr><td>05</td><td><item name="android:windowTitleSize">60dip</item></td></tr><tr><td>06</td><td><item name="android:windowTitleStyle">@style/WindowTitle</item></td></tr><tr><td>07</td><td></style>
08	
09	<style name="WindowTitle" parent="android:WindowTitle"></td></tr><tr><td>10</td><td><item name="android:singleLine">true</item></td></tr><tr><td>11</td><td><item name="android:shadowColor">#BB000000</item></td></tr><tr><td>12</td><td><item name="android:shadowRadius">2.75</item></td></tr><tr><td>13</td><td></style>
14	



Style 和 Theme



- ◎ 示例
 - ■引用

setTheme(R.style.theme);

■ 在 AndroidManifest.xml 中设置 Theme

Attributes	for .testSt	yle (Activ	vity)

A The <u>activity</u> tag declares an **android.app.Activity** class that is available as part of the package's application components, implementing a part of the application's user interface.

testStyle	Browse
@style/Theme	Browse
	@style/Theme





docs\reference\android\widget\TextView.html

- 常用方法
 - getText()
 - setText()





docs\reference\android\widget\EditText.html

- 常用方法
 - selectAll()
 - setEllipsize()





docs\reference\android\widget\Button.html

- 监听方法
 - setOnClickListener()
 - ■三种形式
 - ▶形式1:适用于少量按钮

01	<pre>button.setOnClickListener (new Button.OnClickListener()</pre>	{
02	public void onClick(View v) {	
03	// ТОРО	
04	}	
05	<pre>});</pre>	





docs\reference\android\widget\Button.html

- 监听方法
 - setOnClickListener()
 - ■三种形式
 - ▶形式2:适用于对话框按钮

```
01 button.setOnClickListener(listener_qry);
02 ...
03 Button.OnClickListener listener_qry = new Button.OnClickListener() {
04 public void onClick(View arg0) {
05 // TODO
06 }
07 };
```







■三种形式

▶形式3:适用于多个按钮

01	<pre>button.setOnClickListener(this);</pre>
02	
03	00verride
04	void setOnclick(Button button) (
05	<pre>button.setOnClickListener(new OnClickListener() {</pre>
06	public void onClick(View v) (
07	switch (v.getId()) {
08	case R.id.button_1:
09	// TODO
10	break;
11	case R.id.button_2:
12	// TODO
13	break;
14	default:
15	break;
16	}
17	}
18	<pre>>);</pre>
19)



ImageView

docs\reference\android\widget\ImageView.html

常用组件

● 重要属性

src

adjustViewBounds

● 重要方法

getDrawable()

setImageBitmap(Bitmap bm)

setImageDrawable(Drawable drawable)

setImageResource(int resId)



OheckBox 与 RadioGroup

API

docs\reference\android\widget\CheckBox.html

docs\reference\android\widget\RadioGroup.html

常用组件

- 重要方法
 - isChecked()

setChecked()

setOnCheckedChangeListener()



O Spinner

API

docs\reference\android\widget\Spinner.html

- 使用过程
 - 生成一个 ArrayAdapter ,用于 Spinner 下拉列表的内容。
 - 通过 adapter.setDropDownViewResource() 方法设置下拉列 表样式。

- 使用 Spinner.setAdapter(adapter) 方法将数据源绑定
- 使用 Spinner.setOnItemSelectedListener() 方法响应下拉列 表的选择。





○ 两种菜单

Options Menu

通过 MENU 键激活

Context Menu

当用户长按一个元素时激活

Menu都可以加入子菜单, 但子菜单不能嵌套子菜单。





Options Menu







- ◎ 三种使用方法
 - onCreateOptionsMenu (Menu menu) 创建 Options Menu ,这个方法只会在 menu 第一次显示时调用。

菜单设计

- onPrepareOptionsMenu (Menu menu) 更新改变 Options M enu 的内容,这个方法会在 menu 每次显示时调用。
- onOptionsItemSelected (MenuItem item) 处理选中的菜单 项。





Context Menu

🗟 📶 🕒 10:18 /	AM	🖥 📶 🖵 10:18 AM
Dialer Call log Contacts Favorit	tes	🕞 Obi Wan Kenobi
Oars in the Water Mobile 505-987-6543		View contact
Oats And Feed Mobile 392-123-4567	Touch &	Call Mobile
Obi Wan Kenobi Mobile 294-234-5678	Hold	Send SMS/MMS
Oceans Eleven Mobile 497-876-5432		Add to favorites
Odd Couples Ltd Mobile 492-765-4321		Edit contact
Oedipus Complexities		Delete contact
List View		Context Menu



Context Menu

- ◎ 三种使用方法
 - registerForContextMenu(View view) 为某个 view 注册 Context Menu, 一般在 Activity::onCreate 里面调用。
 - onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) 创建 context menu,和 Options Menu不同, context meun 每次显示时 都会调用这个方法。

菜单设计

onContextItemSelected(MenuItem item) 处理选中的菜单 项。



○ 创建 Menu

◎ 通过 Layout 来添加静态菜单

■ 在 res/menu/ 中创建 menu.xml

■ 将当前的 Activity 与指定的 Menu XML 相关联

getMenuInflater().inflate(R.menu.menu, menu);

■ 捕捉菜单触发事件

onOptionsItemSelected(MenuItem item)

■ 更改 Menu Items 的属性

onPrepareOptionsMenu(Menu menu)







● 菜单资源

res/menu/menu_name.xml
<menu>
 <item>...</item>
 <group>
 <item>...</item>
 </group>
 </group>
 </menu>





● 在 Activity 类中动态创建 Menu

■ 定义 Menu Item 识别序号

public static final MENU_EDIT = 0;

■创建菜单项

■ 捕捉菜单触发事件

■ 注册 Context Menu

registerForContextMenu(getListView());

菜单设计





- Toast
- Notification

对话框设计

AlertDialog





● 构造方法

Toast(Context context) 如: Toast.makeText(getApplicationContext(), "Text toast test!", Toast.LENGTH_LONG).show();

Text toast test!



ONOTIFICATION

NotificationManager 有三个公共方法

- cancel(int id) 取消以前显示的一个通知。如果是一个短暂的 通知,视图将隐藏,如果是一个持久的通知,将从状态条中移 走。
- cancelAll() 取消以前显示的所有通知。
- notify(int id, Notification notification) 把通知持久的发送到 状态条上。





AlertDialog

- 创建 AlertDialog 的一般步骤
 - 通过 AlertDialog.Builder(Context) 获取一个构造器
 - 设置对象的属性
 - 通过 show() 方法显示 AlertDialog
 - 通过 dismiss() 方法消除对话框 (也可通过对话框按钮消除对话 框)。





O AlertDialog

- ◎ 常见方法
 - setSingleChoiceltems() 设置为单选项对话框
 - setMultiChoiceltems() 设置为多选项对话框
 - setItems() 设置为选项对话框,不区分多选单选
 - setPositiveButton()、 setNegativeButton()和 setNeutralBu tton()为对话框设置按钮

- setCustomTitle(View customTitleView) 设置对话框标题为 自定义的视图
- setView(View view) 设置对话框内容为自定义的视图





打开 Android 样例程序

- 查看 Android 源代码
- 界面设计器 DroidDraw





打开 Android 样例程序



- 在 Eclipse 中,运行" File / New / Project / Android Project" 菜单命令,打开" New Android Project" 对 话框。
- 在向导中输入 Project 的名字,选择"Create project f rom existing sample",单击 Browse 按钮,导航至 A ndroid SDK \ samples \ android-#,选择 Snake,返 回向导。
- 选择合适的 Target 后,单击 Finish 按钮完成样例项目 的打开。



查看 Android 源代码



• 下载源文件,地址为

http://rgruet.free.fr/public/android-2.0-eclair-src.zip 解压到平台的类似如下的路径下: D:\android\p latforms\android-8\sources。



界面设计器 DroidDraw



 DroidDraw 是一个基于 Java Swing 的 Android 界面设 计器,可以通过它来生成复杂的 Android Layout XML 文件。



界面设计器 DroidDraw



◎ 设计器下载地址

http://droiddraw.googlecode.com/files/droiddraw-r1b 14.zip

● 模拟器下载地址

http://www.droiddraw.org/AnDroidDraw.apk



界面设计器 DroidDraw

○ 界面







✓ 练习使用 DroidDraw 设计应用程序所使用的布局资源,并设计主界面 main.xml。



项目实践