

## 8

# Android Using Menus

Victor Matos  
Cleveland State University

Notes are based on:

The Busy Coder's Guide to Android Development  
by Mark L. Murphy  
Copyright © 2008-2009 CommonsWare, LLC.  
ISBN: 978-0-9816780-0-9  
&  
Android Developers  
<http://developer.android.com/index.html>



8. Android – UI – Using Menus

## Using Menus

Menus expose apps functionality without consuming 'much' view space.

Android provides two types of menu known as: **options menu** and **context menu**.

1. The **options menu** is triggered by pressing the hardware **Menu** button on the device, while
2. the **context menu** is raised by a *tap-and-hold* on the widget associated to the menu.

8. Android - UI - Using Menus

## Using Menus

**Example:** Using an *option menu*

Options available in this context

Press **Menu** button

A max of six entries per menu. Excess will be displayed as part of the **More** option

3

8. Android - UI - Using Menus

## Using Menus

**Example:**  
Using an *option menu*

Available Options in this context

Press **Menu** button

4

8. Android - UI - Using Menus

## Using Menus

**Example :**  
Dealing with SMS (text-messages) by using the built-in *Messaging* app's **context menu**

Available options

Tap-&-Hold

5

8. Android - UI - Using Menus


## Using Menus

**Observation:**

**Option** and **Context** Menus may include:

1. Text
2. Icons
3. Radio Buttons
4. Check Boxes
5. Sub-menus
6. Short-cut keys

6

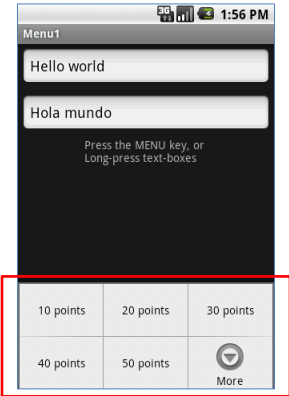

8. Android – UI – Using Menus


## Using Menus

**Example 1: Using an Option Menu**

In this application the Option menu offers a way of changing the text size (on both EditText boxes)


Click on **Menu** button to see size options





After choosing option:  
**50 points**

7

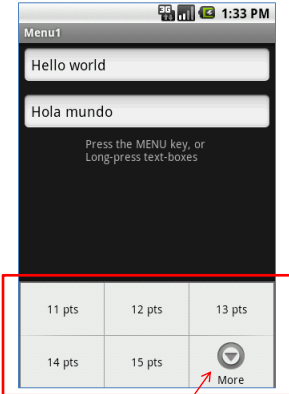

8. Android – UI – Using Menus

## Using Menus

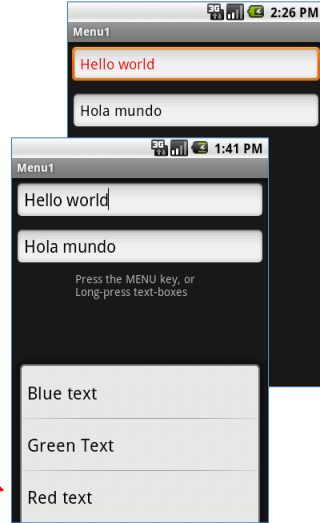
**Example 1: Using an Option Menu**

Up to six options will be displayed on the Option menu.

If there are more than six options pushing the **More** button will show the rest




Click the **More** button to see additional options



Additional choices of the Option menu

8

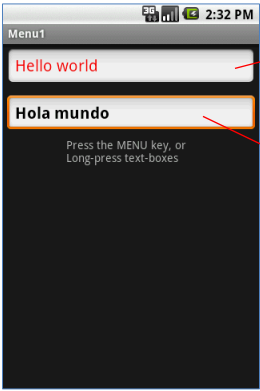


8. Android - UI - Using Menus

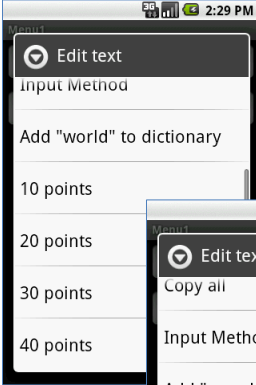
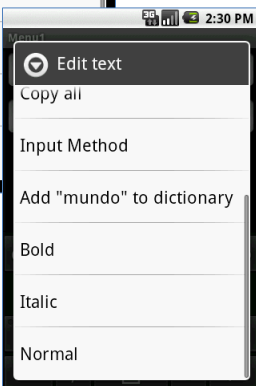
## Using Menus

**Example 1: Using a Context Menu**


Each view could have an associated Context Menu



Long-press a textbox to invoke its Context Menu

9



8. Android - UI - Using Menus

## Using Menus

**Example 1: Using Option and Context Menu**

The app shows two text boxes. Menus are used to change text's size, color, and style.

```

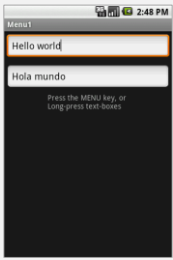
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <EditText
        android:id="@+id/etMessage1"
        android:text="Hello world"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_margin = "5dp" />

    <EditText
        android:id="@+id/etMessage2"
        android:text="Hola mundo"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_margin = "5dp" />

    <TextView
        android:text="Press the MENU key, or \nLong-press text-boxes"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center" />
</LinearLayout>

```



10



8. Android - UI - Using Menus

# Using Menus

## Example 1: Using Option and Context Menu

The app shows two text boxes. Menus are used to change text's size, color, and style.

```
package cis493.matos.menu;
// using Menus (Option & Context)
import android.app.Activity; ...

public class MenuAct1 extends Activity {
    EditText etMessage1;
    EditText etMessage2;
    Integer[] arrayPointSize = {10, 20, 30, 40, 50};

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        etMessage1 = (EditText) findViewById(R.id.etMessage1);
        etMessage2 = (EditText) findViewById(R.id.etMessage2);

        // you may register an individual context menu for each view
        registerForContextMenu(etMessage1);
        registerForContextMenu(etMessage2);
    } //onCreate
```

11



8. Android - UI - Using Menus

# Using Menus

## Example 1: Using Option and Context Menu

The app shows two text boxes. Menus are used to change text's size, color, and style.

```
// set the option menu for the current activity
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // only one Option menu per activity
    populateMyFirstMenu(menu);
    return super.onCreateOptionsMenu(menu);
}

// detect what view is calling and create its context menu
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
    ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    // decide what context menu needs to be made
    if (v.getId() == etMessage1.getId())
        // create a menu for etMessage1 box
        populateMyFirstMenu(menu);
    if (v.getId() == etMessage2.getId()) {
        // create a menu for etMessage2 box
        populateMySecondMenu(menu);
    }
} //onCreateContextMenu
```

12



8. Android - UI - Using Menus

# Using Menus

## Example 1: Using Option and Context Menu

The app shows two text boxes. Menus are used to change text's size, color, and style.

```
private void populateMyFirstMenu(Menu menu) {
    int groupId = 0; int order= 0;
    //arguments: groupId, optionId, order, title
    menu.add(groupId, 1, 1, "10 points");
    menu.add(groupId, 2, 2, "20 points");
    menu.add(groupId, 3, 3, "30 points");
    menu.add(groupId, 4, 4, "40 points");
    menu.add(groupId, 5, 5, "50 points");

    menu.add(groupId, 6, 8, "Red text");
    menu.add(groupId, 7, 7, "Green Text");
    menu.add(groupId, 8, 6, "Blue text");
} //populateMyMenu

private void populateMySecondMenu(Menu menu){
    int groupId = 0; int order= 0;
    //arguments: groupId, optionId, order, title
    menu.add(groupId, 9, 1, "Bold");
    menu.add(groupId, 10, 2, "Italic");
    menu.add(groupId, 11, 3, "Normal");
} //populateMySecondMenu
```

13



8. Android - UI - Using Menus

# Using Menus

## Example 1: Using Option and Context Menu

The app shows two text boxes. Menus are used to change text's size, color, and style.

```
@Override
public boolean onContextItemSelected(MenuItem item) {
    return (applyMenuOption(item));
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    return (applyMenuOption(item));
}
```

14



8. Android - UI - Using Menus

# Using Menus

## Example 1: Using Option and Context Menu

```
private boolean applyMenuOption(MenuItem item){
    int menuItemId = item.getItemId(); // 1, 2, 3, ...11
    String strMsg2 = etMessage2.getText().toString();

    if (menuItemId <= 5) {
        // first five option are for setting text size
        int newPointSize = arrayPointSize[menuItemId - 1];
        etMessage1.setTextSize(newPointSize);
        etMessage2.setTextSize(newPointSize);
    }
    else {
        // either change color on text1 or style on text2
        if (menuItemId == 6)
            etMessage1.setTextColor(0xffff0000); // red
        else if (menuItemId == 7)
            etMessage1.setTextColor(0xff00ff00); // green
        else if (menuItemId == 8)
            etMessage1.setTextColor(0xff0000ff); // blue
        else if (menuItemId == 9)
            etMessage2.setText(beautify(strMsg2, "BOLD")); //bold
        else if (menuItemId == 10)
            etMessage2.setText(beautify(strMsg2, "ITALIC")); //italic
        else if (menuItemId == 11)
            etMessage2.setText(beautify(strMsg2, "NORMAL")); //normal
    }
    return false;
} //applyMenuOption
```

15



8. Android - UI - Using Menus

# Using Menus

## Example 1: Using Option and Context Menu

The app shows two text boxes. Menus are used to change text's size, color, and style.

```
// changing text style using HTML formatting
// Spanned is text to which you could add formatting features
private Spanned beautify (String originalText, String selectedStyle){
    Spanned answer = null;
    if (selectedStyle.equals("BOLD"))
        answer = Html.fromHtml("<b>" + originalText + "</b>");
    else if (selectedStyle.equals("ITALIC"))
        answer = Html.fromHtml("<i>" + originalText + "</i>");
    else if (selectedStyle.equals("NORMAL"))
        answer = Html.fromHtml("<normal>" + originalText + "</normal>");

    return answer;
} //beautify

} //MenuAct1
```

16





8. Android – UI – Using Menus

# Using Menus

## Comments on Creating an Option & Context Menu

### Step1.

Indicate which widget(s) on your activity have context menus. To do this, call `registerForContextMenu(theWidget)` supplying the View that is the widget needing a context menu.

### Step2.

Implement `onCreateContextMenu(...)`, populate your menu adding text, icons, etc. to the different options. Use input menu parameter to determine which menu to build (assuming your activity has more than one).

The `onCreateContextMenu()` method gets the `ContextMenu` itself, the `View` the context menu is associated with, and a `ContextMenu.ContextMenuInfo`, which tells you which item in the list the user did the tap-and-hold over, in case you want to customize the context menu based on that information

17



8. Android – UI – Using Menus

# Using Menus

## Comments on Creating an Option & Context Menu

- `onCreateContextMenu()` is called each time the context menu is requested.
- Unlike the *options menu* (which is only built once per activity), *context menus* are discarded once they are used or dismissed.
- To find out when a context menu choice was chosen, implement `onContextItemSelected()` on the activity.

18



# Using Menus

## Comments on Creating an Option & Context Menu

Our Example1 implementations of  
*onOptionsItemSelected()* (for options menu selections) and  
*onContextItemSelected()* (for context menu selections)


delegate to the *applyMenuChoice(...)* method where actions associated to the choices made is implemented

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    return (applyMenuChoice(item) );
}

@Override
public boolean onContextItemSelected(MenuItem item) {
    return (applyMenuChoice(item));
}
```

19

20



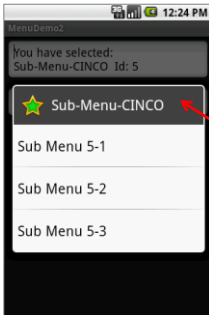
8. Android - UI - Using Menus

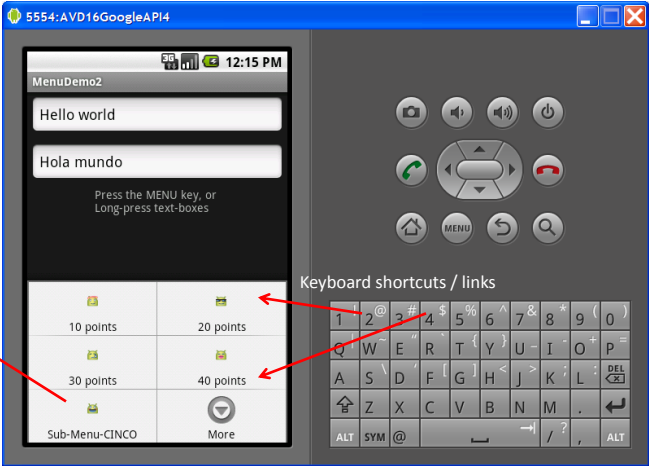
# Using Menus

Extending Example1.

**Example 2: Enhancing Option/Context Menu**

This extension of the previous example adds **icons** and **links** to the Option Menu entries, as well as a **sub-menu list** tied to one the option menu buttons.






Keyboard shortcuts / links

Click on **SubMenu** to see additional options

21



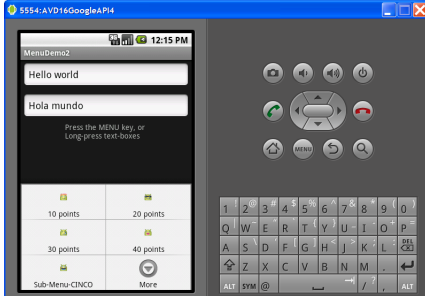
8. Android - UI - Using Menus

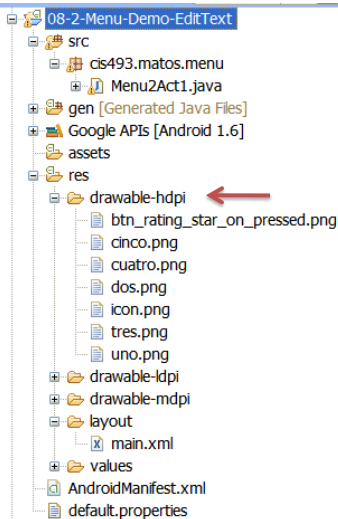
# Using Menus

Extending Example1.

**Example 2: Enhancing Option/Context Menu**

This extension of the previous example adds **icons** and **links** to the Option Menu entries, as well as a **sub-menu list** tied to one the option menu buttons.





22



8. Android - UI - Using Menus

# Using Menus

Extending Example1.

## Example 2: Enhancing Option/Context Menu

Replace the method **populateMyFirstMenu** with the following code

```
private void populateMyFirstMenu(Menu menu) {
    int groupId = 0;
    //arguments: groupId, optionId, order, title
    MenuItem item1 = menu.add(groupId, 1, 1, "10 points");
    MenuItem item2 = menu.add(groupId, 2, 2, "20 points");
    MenuItem item3 = menu.add(groupId, 3, 3, "30 points");
    MenuItem item4 = menu.add(groupId, 4, 4, "40 points");
    //MenuItem item5 = menu.add(groupId, 5, 5, "50 points");

    MenuItem item6 = menu.add(groupId, 6, 8, "Red text");
    MenuItem item7 = menu.add(groupId, 7, 7, "Green Text");
    MenuItem item8 = menu.add(groupId, 8, 6, "Blue text");

    //set icons
    item1.setIcon(R.drawable.uno);
    item2.setIcon(R.drawable.dos);
    item3.setIcon(R.drawable.tres);
    item4.setIcon(R.drawable.cuatro);

    // shortcuts using device's keyboard-keypad
    // on a G1 open slide open the keyboard and
    // type letter u (same as pressing menu UNO)
    item1.setShortcut('1', '1');
    item2.setShortcut('2', '2');
    item3.setShortcut('3', '3');
    item4.setShortcut('4', '4');
}
```

Remove this line from previous version

Figures used in this example were taken from:  
C:\android-sdk-windows\platforms\android-4\data/res/drawable



23



8. Android - UI - Using Menus

# Using Menus

Extending Example1.

## Example 2: Enhancing Option/Context Menu

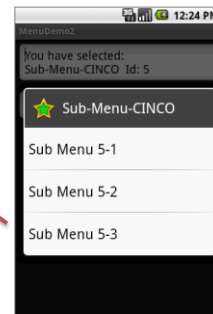
Continuation...  
Replace the method **populateMyFirstMenu** with the following code

```
// adding a sub-menu as fifth entry of this menu
// .addSubMenu(int groupId, int itemId, int order, CharSequence title)
int smGroupId = 0; // don't care, same as Menu.NONE
int smItemId = 5; // fifth element
int smOrder = 5; // don't care, same as Menu.NONE

SubMenu mySubMenu5 = menu.addSubMenu(smGroupId, smItemId, smOrder, "Sub-Menu-CINCO");
mySubMenu5.setHeaderIcon(R.drawable.btn_rating_star_on_pressed);
mySubMenu5.setIcon(R.drawable.cinco);
// .add(int groupId, int itemId, int order, CharSequence title)

MenuItem sub51 = mySubMenu5.add(smGroupId, 5, 1, "Sub Menu 5-1");
MenuItem sub52 = mySubMenu5.add(smGroupId, 5, 2, "Sub Menu 5-2");
MenuItem sub53 = mySubMenu5.add(smGroupId, 5, 3, "Sub Menu 5-3");

} //populateMyFirstMenu
```



24



8. Android - UI - Using Menus

# Using Menus

Extending Example1.

## Example 2: Enhancing Option/Context Menu

Continuation...  
Replace the method  
**applyMenuOption**  
with the following code

```
private boolean applyMenuOption(MenuItem item) {

    int menuItemId = item.getItemId(); //1, 2, 3, ...11

    String strMsg2 = etMessage2.getText().toString();

    if (menuItemId < 5) {
        // first four options are for setting text size
        int newPointSize = arrayPointSize[menuItemId - 1];
        etMessage1.setTextSize(newPointSize);
        etMessage2.setTextSize(newPointSize);
    }
    else if (menuItemId == 5) {
        // the sub-menu (attached to 5th item) is processed here
        etMessage1.setText ( "You have selected: \n" +
            item.getTitle() + " Id: " + menuItemId );
    }
    // either change color on text1 or style on text2
    else if (menuItemId == 6)
        etMessage1.setTextColor(0xffff0000); // red
}
```

← Same as before

← Take care of sub-menu here

25



8. Android - UI - Using Menus

# Using Menus

Extending Example1.

## Example 2: Enhancing Option/Context Menu

Continuation...  
Replace the method  
**applyMenuOption**  
with the following code

```
else if (menuItemId == 7)
    etMessage1.setTextColor(0xff00ff00); // green
else if (menuItemId == 8)
    etMessage1.setTextColor(0xff0000ff); // blue
else if (menuItemId == 9)
    etMessage2.setText(beatify(strMsg2, "BOLD")); //bold
else if (menuItemId == 10)
    etMessage2.setText(beatify(strMsg2, "ITALIC")); //italic
else if (menuItemId == 11)
    etMessage2.setText(beatify(strMsg2, "NORMAL")); //normal

return false;
} //applyMenuOption
```

← Same as before

26




8. Android – UI – Using Menus

## Using Menus

# Questions ?

27

28

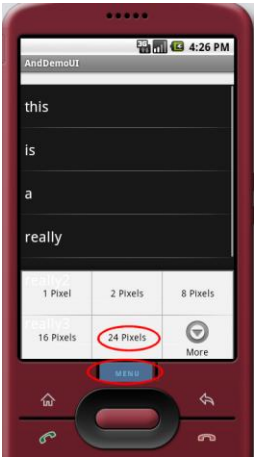


# Using Menus

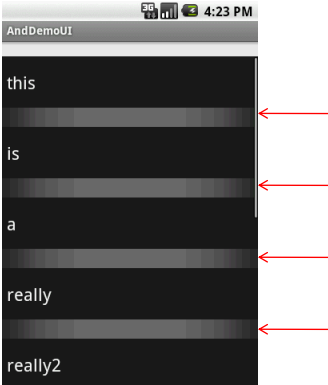
Very similar to Example1.  
You may want to skip it

**Example 2: Using Option/Context Menu**

Observe how the selection made from the Menu changes the **inter-row spacing** of the ListView.




Click on **Menu** to see spacing options



After choosing option: **24 Pixels**  
of row separation

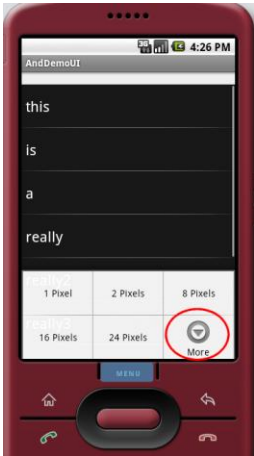
29



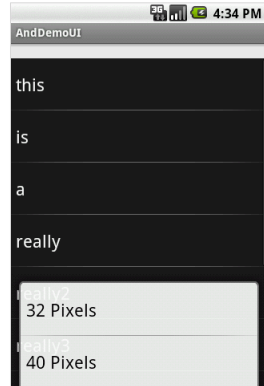
# Using Menus

Very similar to Example1.  
You may want to skip it

**Example 2: Using an Option Menu**




Click on **Menu** to see spacing options



After choosing option: **More**  
a new menu is shown

30




8. Android - UI - Using Menus

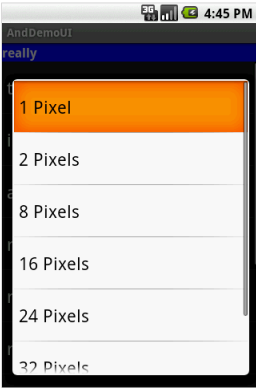
# Using Menus

*Very similar to Example1.  
You may want to skip it*

## Example 2: Using a Context Menu


**Tap & Hold**  
any row to bring  
Context Menu





Choices listed in a  
floating  
ListView

31



8. Android - UI - Using Menus

# Using Menus

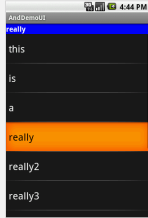
*Very similar to Example1.  
You may want to skip it*

## Example 2: Creating an Option & Context Menu

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <TextView
        android:id="@+id/selection"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#eeeeee"/>
    <ListView
        android:id="@android:id/list"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:drawSelectorOnTop="false"
        />
</LinearLayout>

```



32





8. Android - UI - Using Menus

# Using Menus

Very similar to Example1.  
You may want to skip it

## Example 2: Creating an Option & Context Menu

```
package cis493.demoui;

import android.os.Bundle;
import android.app.ListActivity;
import android.view.ContextMenu;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
```

33



8. Android - UI - Using Menus

# Using Menus

Very similar to Example1.  
You may want to skip it

## Example 2: Creating an Option & Context Menu

```
public class AndDemoUI extends ListActivity {

    TextView selection;
    // data to be displayed in the ListView
    String[] items = { "this", "is", "a", "really", "really2",
                      "really3", "really4", "really5", "silly", "list" };

    public static final int EIGHT_ID          = Menu.FIRST + 1;
    public static final int SIXTEEN_ID       = Menu.FIRST + 2;
    public static final int TWENTY_FOUR_ID  = Menu.FIRST + 3;
    public static final int TWO_ID          = Menu.FIRST + 4;
    public static final int THIRTY_TWO_ID   = Menu.FIRST + 5;
    public static final int FORTY_ID        = Menu.FIRST + 6;
    public static final int ONE_ID          = Menu.FIRST + 7;
```

34



8. Android - UI - Using Menus

# Using Menus

Very similar to Example1.  
You may want to skip it

## Example 2: Creating an Option & Context Menu

```
@Override
public void onCreate(Bundle icle) {
    super.onCreate(icle);
    setContentView(R.layout.main);

    //show array data into list
    setListAdapter(new ArrayAdapter<String>(
        this,
        android.R.layout.simple_list_item_1,
        items));
    selection = (TextView) findViewById(R.id.selection);

    //attach a ContextMenu to the Activity's ListView
    registerForContextMenu(getListView());
}

public void onItemClick(ListView parent, View v,
    int position, long id) {
    selection.setText(items[position]);
}
```

35



8. Android - UI - Using Menus

# Using Menus

Very similar to Example1.  
You may want to skip it

## Example 2: Creating an Option & Context Menu

```
@Override
public void onCreateContextMenu( ContextMenu menu, View v,
    ContextMenu.ContextMenuInfo menuInfo) {
    // if comment it out - no context menu is created
    populateMenu(menu);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // if comment it out - no option menu is created
    populateMenu(menu);
    return (super.onCreateOptionsMenu(menu));
}
```

36



8. Android - UI - Using Menus

# Using Menus

Very similar to Example1.  
You may want to skip it

## Example 2: Creating an Option & Context Menu

```
// we will react to the events: either a click on the Menu button
// or a long 'tap-and-hold' to trigger Option Menu or Context Menu

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    return ( applyMenuChoice(item) );
}

@Override
public boolean onContextItemSelected(MenuItem item) {
    return ( applyMenuChoice(item) );
}
```

37



8. Android - UI - Using Menus

# Using Menus

Very similar to Example1.  
You may want to skip it

## Example 2: Creating an Option & Context Menu

```
private void populateMenu(Menu menu) {

    menu.add(Menu.NONE, ONE_ID, Menu.NONE, "1 Pixel");
    menu.add(Menu.NONE, TWO_ID, Menu.NONE, "2 Pixels");
    menu.add(Menu.NONE, EIGHT_ID, Menu.NONE, "8 Pixels");
    menu.add(Menu.NONE, SIXTEEN_ID, Menu.NONE, "16 Pixels");
    menu.add(Menu.NONE, TWENTY_FOUR_ID, Menu.NONE, "24 Pixels");

    menu.add(Menu.NONE, THIRTY_TWO_ID, Menu.NONE, "32 Pixels");
    menu.add(Menu.NONE, FORTY_ID, Menu.NONE, "40 Pixels");
}
```

A max of six entries  
per menu. Excess will  
be displayed as part of  
the **More** option

1 Pixel	2 Pixels	8 Pixels
really2 16 Pixels	24 Pixels	 More

38



8. Android - UI - Using Menus

# Using Menus

Very similar to Example1.  
You may want to skip it

## Example 2: Creating an Option & Context Menu

```
private boolean applyMenuChoice(MenuItem item) {

    switch (item.getItemId()) {
        case ONE_ID:
            getListView().setDividerHeight(1);
            return (true);

        case EIGHT_ID:
            getListView().setDividerHeight(8);
            return (true);

        case SIXTEEN_ID:
            getListView().setDividerHeight(16);
            return (true);

        case TWENTY_FOUR_ID:
            getListView().setDividerHeight(24);
            return (true);
    }
}
```

39



8. Android - UI - Using Menus

# Using Menus

Very similar to Example1.  
You may want to skip it

## Example 2: Creating an Option & Context Menu

```
case TWO_ID:
    getListView().setDividerHeight(2);
    return (true);

case THIRTY_TWO_ID:
    getListView().setDividerHeight(32);
    return (true);

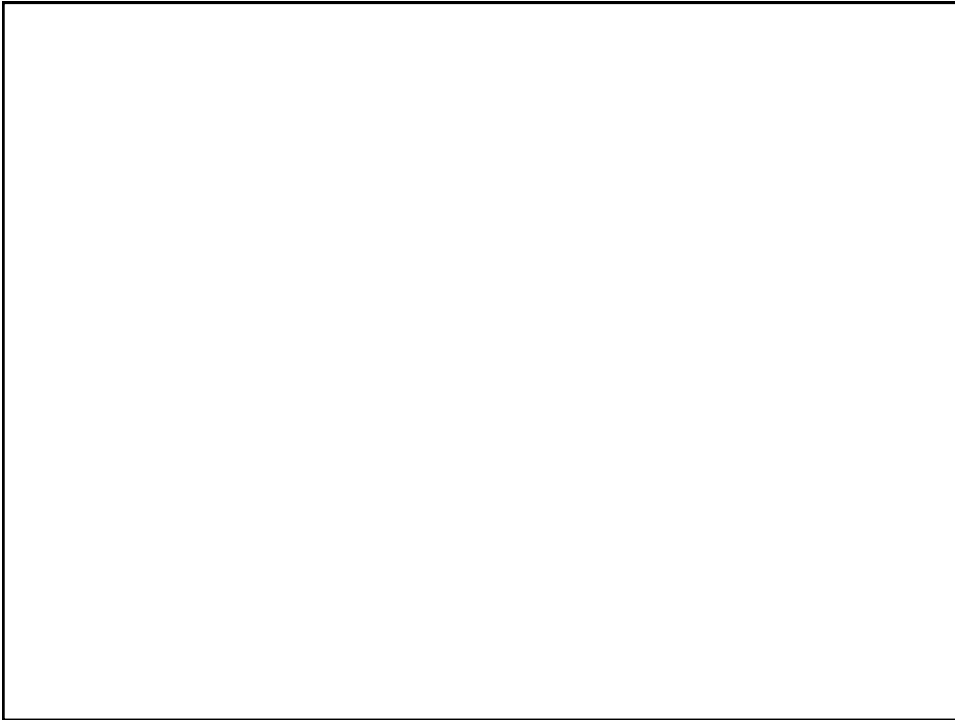
case FORTY_ID:
    getListView().setDividerHeight(40);
    return (true);
}


return (false);

} // applyMenuChoice

} // class
```

40





8. Android – UI – Using Menus

# Using Menus

*Very similar to Example1.  
You may want to skip it*

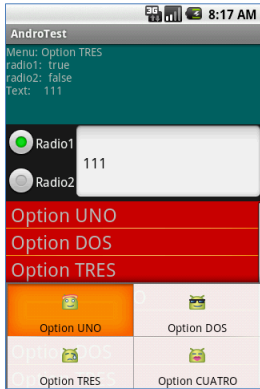
  

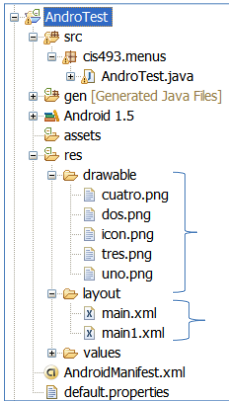
**Example 3:** Another App Using Menus that include Icons & Text

This is a simple application designed around a TableLayout.

1. The **first row** shows a summary of the data collected from the screen.
2. The **second row** exposes a RadioGroup and a textBox.
3. The **third row** shows a list of the calls made to the option and context menus.

This Option Menu shows its options using text & icons





icons

layouts

42



8. Android - UI - Using Menus

# Using Menus

Very similar to Example1.  
You may want to skip it

## Example 3: Another App Using Menus that include Icons & Text

### Application's Layout

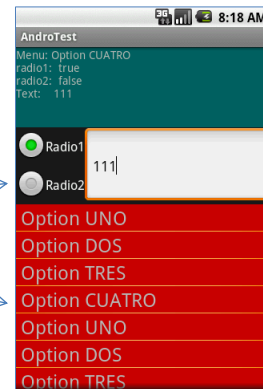
#### TableLayout

Row1: TextView

Row2: RelativeLayout holding



Row3: ListView (acting as a Scrolling Container)



43



8. Android - UI - Using Menus

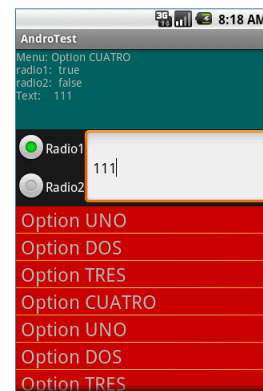
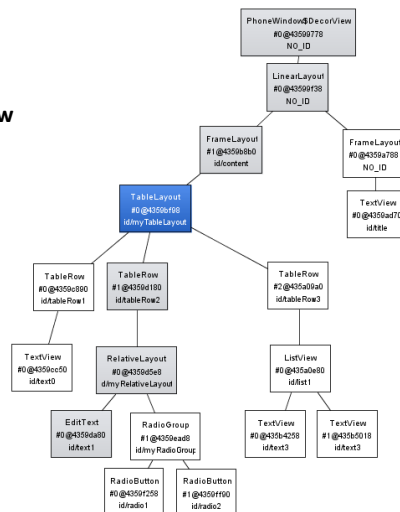
# Using Menus

Very similar to Example1.  
You may want to skip it


## Example 3: Another App Using Menus that include Icons & Text

### Application's Layout.

### Hierarchy View



44



8. Android - UI - Using Menus

## Example 3: Main Layout


### Using Menus

```

<?xml version="1.0" encoding="utf-8"?>
<TableLayout
  android:id="@+id/myTableLayout"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical"
  android:stretchColumns="1,2,3"
  xmlns:android="http://schemas.android.com/apk/res/android"
>
  <TableRow
    android:id="@+id/tableRow1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
  >
    <TextView
      android:id="@+id/text0"
      android:layout_width="fill_parent"
      android:layout_height="100px"
      android:background="#ff006666"
      android:text="TextView"
      android:layout_span="3"
    >
    </TextView>
  </TableRow>

```

Row 1



8. Android - UI - Using Menus

## Example 3: Main Layout


### Using Menus

```

<TableRow
  android:id="@+id/tableRow2"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:orientation="horizontal"
>
  <RelativeLayout
    android:id="@+id/myRelativeLayout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
  >
    <EditText
      android:id="@+id/text1"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:hint="Type something..."
      android:textSize="18sp"
      android:layout_alignTop="@+id/myRadioGroup"
      android:layout_alignBottom="@+id/myRadioGroup"
      android:layout_toRightOf="@+id/myRadioGroup"
      android:padding="10px"
      android:paddingLeft="20px"
    >
    </EditText>

```

Row 2



8. Android - UI - Using Menus

## Example 3: Main Layout Using Menus


```

<RadioGroup
    android:id="@+id/myRadioGroup"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
>
    <RadioButton
        android:id="@+id/radio1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Radio1"
    >
    </RadioButton>
    <RadioButton
        android:id="@+id/radio2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Radio2"
    >
    </RadioButton>
</RadioGroup>
</RelativeLayout>
</TableRow>

```

← Row 2 cont.

47



8. Android - UI - Using Menus

## Example 3: Main Layout Using Menus

```

<TableRow
    android:id="@+id/tableRow3"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="#ffff9900"
    android:orientation="vertical"
    android:layout_weight="2"
>
    <ListView
        android:id="@+id/list1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:isScrollContainer="true">
    </ListView>
</TableRow>
</TableLayout>

```

← Row 3

48





8. Android - UI - Using Menus

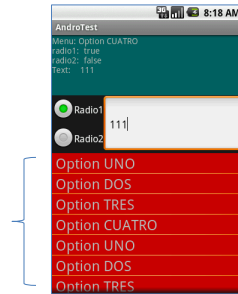
# Using Menus

Very similar to Example1.  
You may want to skip it

## Example 3. Menu Using Icons & Text

This custom layout (*main1.xml*) is used to show the **ListView** rows (shown as red rows – big font)

```
<?xml version="1.0" encoding="UTF-8"?>
<TextView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/text3"
    android:background="#ffcc0000"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize="24sp"
    android:gravity="center_vertical"
    android:paddingLeft="6dip"
/>
```



49



8. Android - UI - Using Menus

# Using Menus

Very similar to Example1.  
You may want to skip it

## Example 3. Menu Using Icons & Text

```
public class AndroTest extends Activity {

    TextView text0;
    EditText text1;
    RadioButton radio1;
    RadioButton radio2;
    ListView list1;
    ArrayList<String> arrOptions;
    ArrayAdapter<String> adaOptions;

    final int PICK1 = Menu.FIRST;
    final int PICK2 = Menu.FIRST + 1;
    final int PICK3 = Menu.FIRST + 2;
    final int PICK4 = Menu.FIRST + 3;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        text0 = (TextView) findViewById(R.id.text0);
        text1 = (EditText) findViewById(R.id.text1);
        radio1 = (RadioButton) findViewById(R.id.radio1);
        radio2 = (RadioButton) findViewById(R.id.radio2);
        list1 = (ListView) findViewById(R.id.list1);
```

Binding XML UI  
controls to Java code



8. Android - UI - Using Menus

# Using Menus

## Example 3. Menu Using Icons & Text

```
//show history of all selected options
arrOptions = new ArrayList<String>();
adaOptions = new ArrayAdapter<String>(
    this,
    R.layout.main1,
    arrOptions);

list1.setAdapter(adaOptions);

//long-press menu for list and textbox
registerForContextMenu(list1);
registerForContextMenu(text1);

} // onCreate
```

A list will record  
Menu selections

Bind context menu to  
EditBox and List

51



8. Android - UI - Using Menus

# Using Menus

## Example 3. Menu Using Icons & Text



```
@Override
public boolean onCreateOptionsMenu(Menu menu) {

    super.onCreateOptionsMenu(menu);
    MenuItem item1 = menu.add(0, PICK1, Menu.NONE, "Option UNO");
    MenuItem item2 = menu.add(0, PICK2, Menu.NONE, "Option DOS");
    MenuItem item3 = menu.add(0, PICK3, Menu.NONE, "Option TRES");
    MenuItem item4 = menu.add(1, PICK4, Menu.NONE, "Option CUATRO");
    //set icons
    item1.setIcon(R.drawable.uno);
    item2.setIcon(R.drawable.dos);
    item3.setIcon(R.drawable.tres);
    item4.setIcon(R.drawable.cuatro);

    //shortcuts using device's keyboard-keypad
    item1.setShortcut('1', 'u');
    item2.setShortcut('2', 'd');
    item3.setShortcut('3', 't');
    item4.setShortcut('4', 'c');
    return true;
} // onCreateOptionsMenu
```

Make Option Menu  
Create items, icons, shortcuts

2



8. Android - UI - Using Menus

# Using Menus

## Example 3. Menu Using Icons & Text

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    try {
        super.onOptionsItemSelected(item);
        int itemID = item.getItemId();
        //title to display is taken from current menu item
        String msg= item.getTitle().toString();
        //add selection to historical array and show in listview
        arrOptions.add(msg);
        adaOptions.notifyDataSetChanged();

        //values in the green TextView box include:
        msg += "\n" + "radio1: " + Boolean.toString(radio1.isChecked());
        msg += "\n" + "radio2: " + Boolean.toString(radio2.isChecked());
        msg += "\n" + "Text: " + text1.getText();
        text0.setText("Menu: " + msg);
    }
    catch (Exception e) {
        text0.setText(e.getMessage());
    }
    return false;
} //onOptionsItemSelected
```

Put summary on  
green textBox



8. Android - UI - Using Menus

# Using Menus

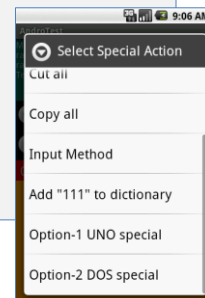
## Example 3. Menu Using Icons & Text

```
//this is a floating context menu that appears when user
//clicks down for a while (about 2 sec) on the textbox or list


@Override
public void onCreateContextMenu(ContextMenu menu, View v,
    ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);

    //add a couple of options to the context menu
    menu.setHeaderTitle("Select Special Action");
    menu.add(0, PICK1, Menu.NONE, "Option-1 UNO special");
    menu.add(0, PICK2, Menu.NONE, "Option-2 DOS special");
} //onCreateContextMenu
}
```

Add two custom options to  
context menu



54




8. Android - UI - Using Menus

# Using Menus

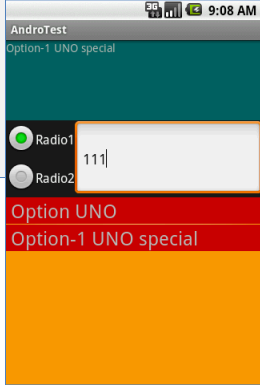
## Example 3. Menu Using Icons & Text

```
// selecting options from the context menu
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    super.onOptionsItemSelected(item);

    String msg = item.getTitle().toString();
    text0.setText(msg);
    arrOptions.add(msg);
    adaOptions.notifyDataSetChanged();
    return false;
} //onOptionsItemSelected
```




Listen to Context  
Menu selections



Produced by option menu →

Produced by context menu →

55



8. Android - UI - Using Menus

# Using Menus

## Comments on Creating Submenus

Submenus are displayed as regular *Menu Items*. When selected they just display more options.

To avoid cluttered UIs, submenus are not displayed using three-like structure, instead they appear as a simple *floating window* presenting a list of items.

Android does not support nested submenus.

56



8. Android - UI - Using Menus

# Using Menus

## Comments on Creating Submenus

### Example:

Extend previous example by adding a fifth entry to the existing menu list. The new entry is of the *SubMenu* type. It should offer three options of its own.

```
// adding a sub-menu as fifth entry of this menu
SubMenu mySubMenu5 = menu.addSubMenu(0, 0, Menu.NONE, "Sub-Menu-CINCO");

mySubMenu5.setHeaderIcon(R.drawable.cinco);
mySubMenu5.setIcon(R.drawable.cinco);

MenuItem sub51 = mySubMenu5.add(0, 0, Menu.NONE, "Sub Menu 5-1");
MenuItem sub52 = mySubMenu5.add(0, 0, Menu.NONE, "Sub Menu 5-2");
MenuItem sub53 = mySubMenu5.add(0, 0, Menu.NONE, "Sub Menu 5-3");
```

**Note:** add this code to the end of the: `onCreateOptionsMenu(...)` method.

57



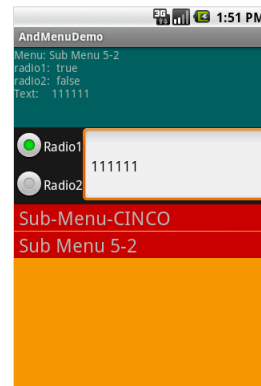
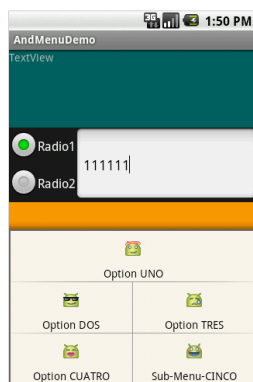
8. Android - UI - Using Menus

# Using Menus

## Comments on Creating Submenus

### Example:

Extend previous example by adding a fifth entry to the existing menu list. The new entry is of the *SubMenu* type. It should offer three options of its own.



58