

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>





Using Menus

Menus usually increase the functionality of an app by providing additional operations on a small overlapping panel.

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

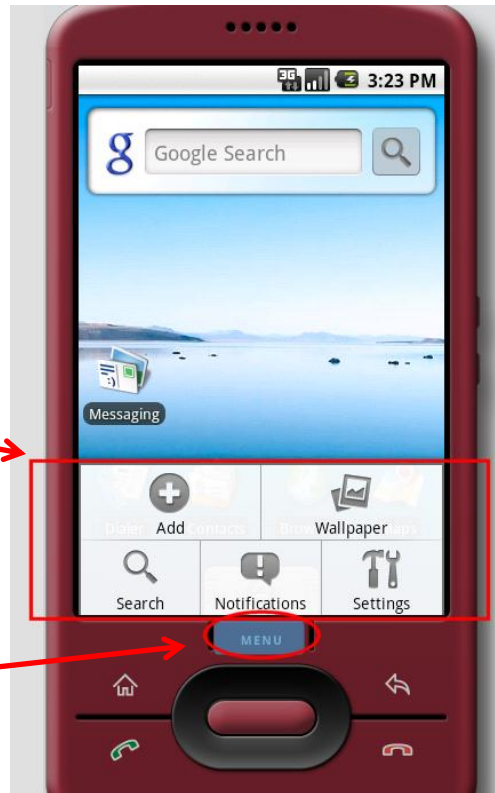
1. The **options menu** is triggered by pressing the hardware **Menu** button on the device, while
2. the **context menu** is raised by a *tap-and-hold* on the widget associated to the menu.

Using Menus

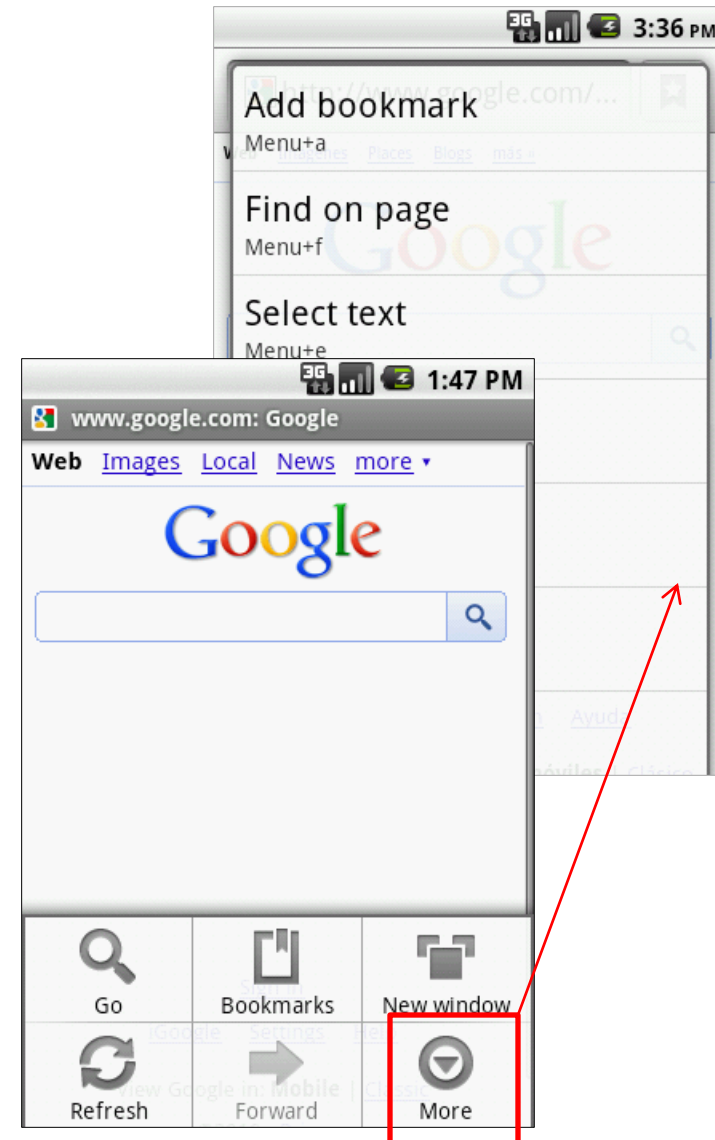
Example: Using an *option menu*

Options
available in
this context

Press **Menu**
button



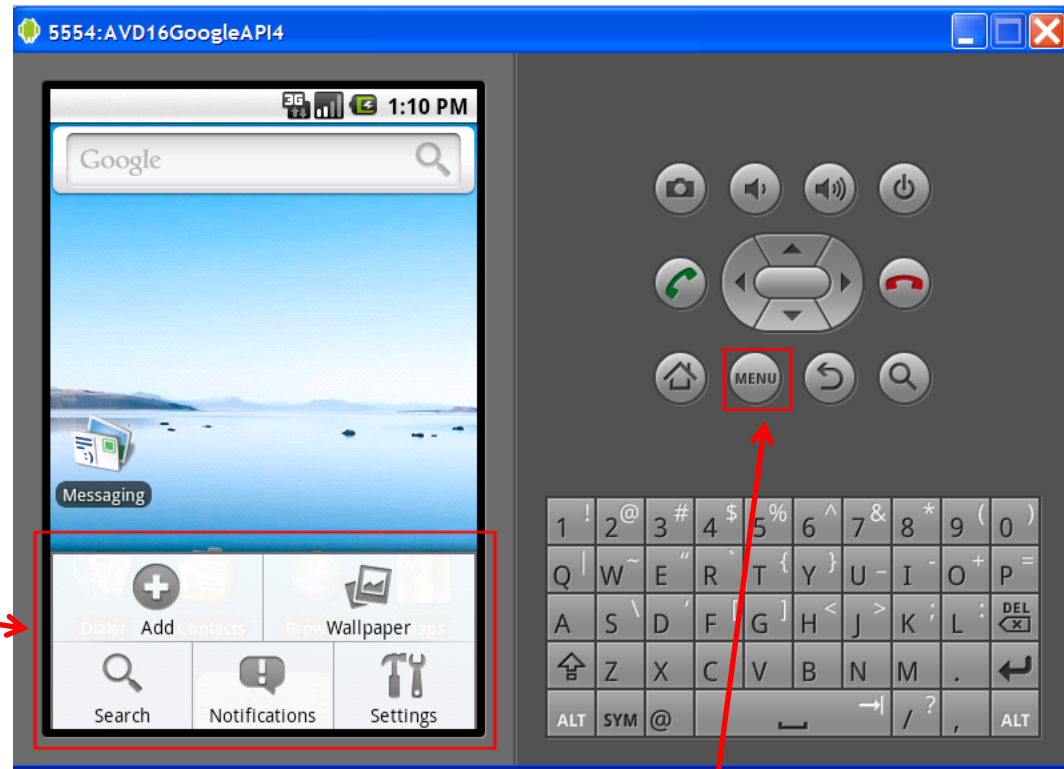
A max of six entries per
menu. Excess will be displayed
as part of the **More** option



Using Menus

Example:
Using an *option menu*

Five available
Options in
this context

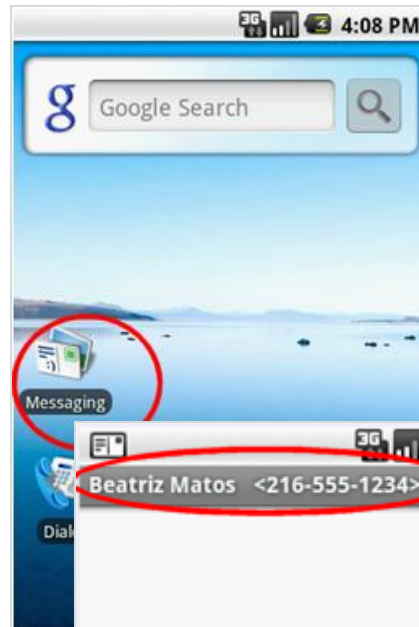


Press **Menu**
button

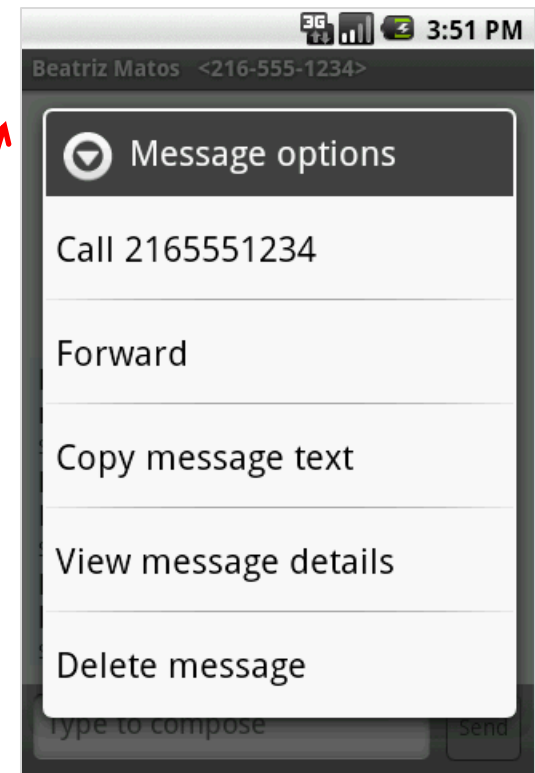
Using Menus

Example :

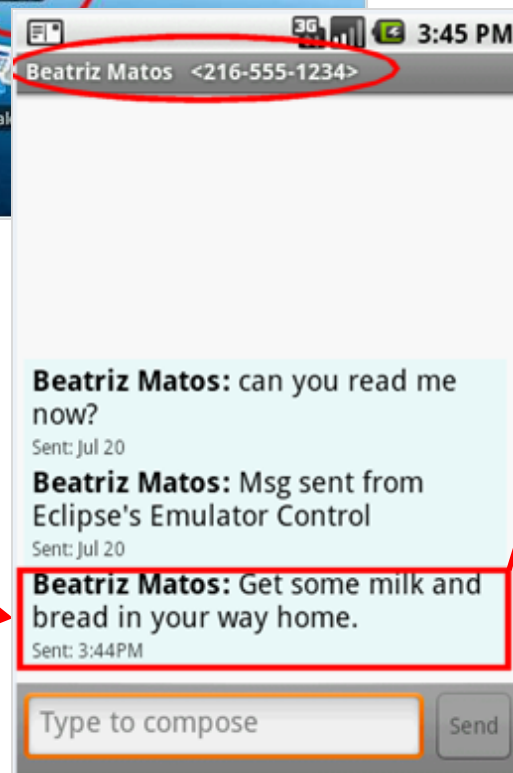
Dealing with SMS (text-messages) by using the built-in *Messaging* app's **context menu**



Available options



Tap-&-Hold





Using Menus

Observation:

Option and **Context** Menus may include:

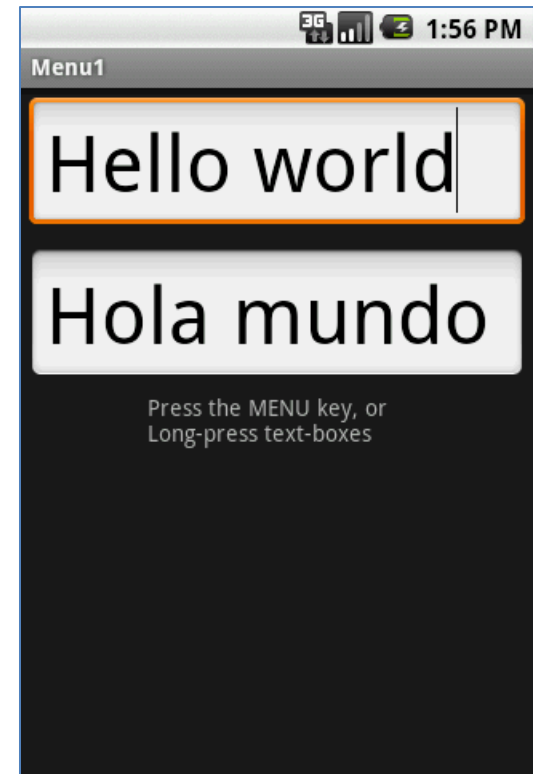
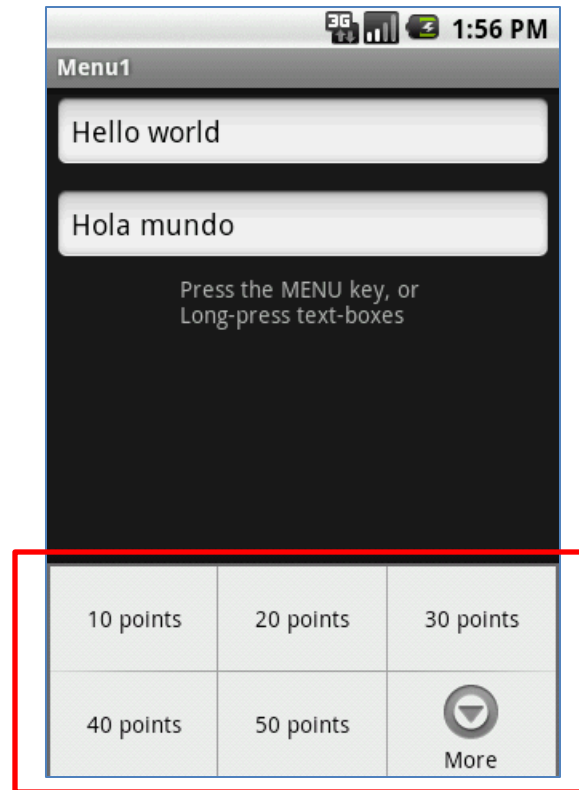
1. Text
2. Icons
3. Radio Buttons
4. Check Boxes
5. Sub-menus
6. Short-cut keys

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



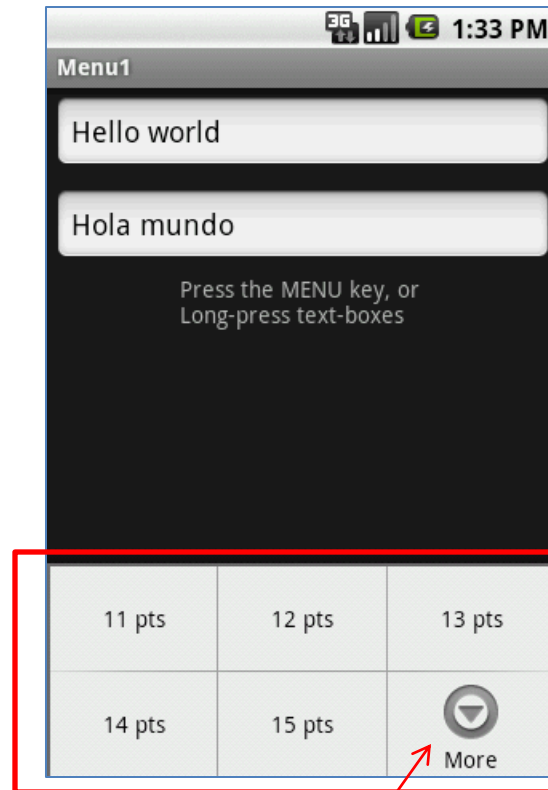
After choosing option:
50 points

Using Menus

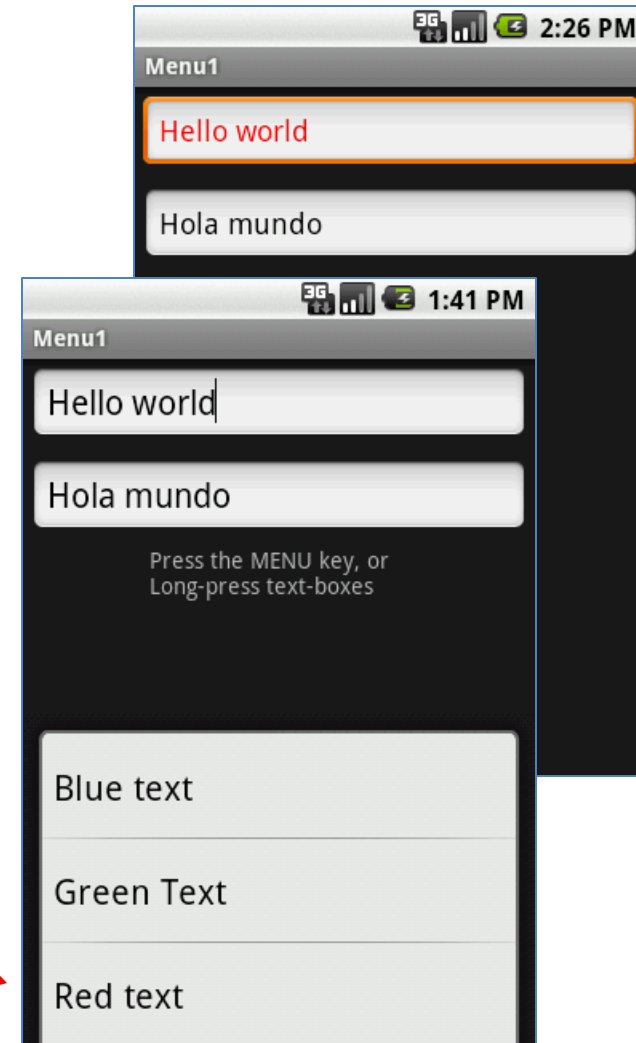
Example 1: Using an Option Menu

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

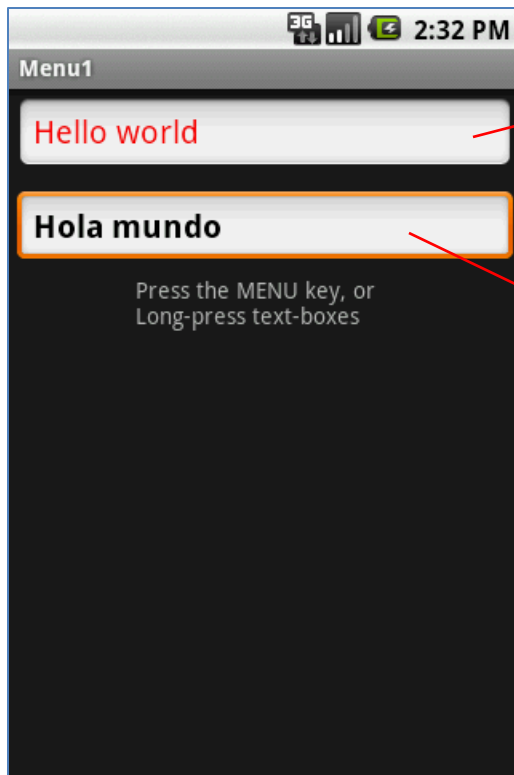


Additional choices of the Option menu

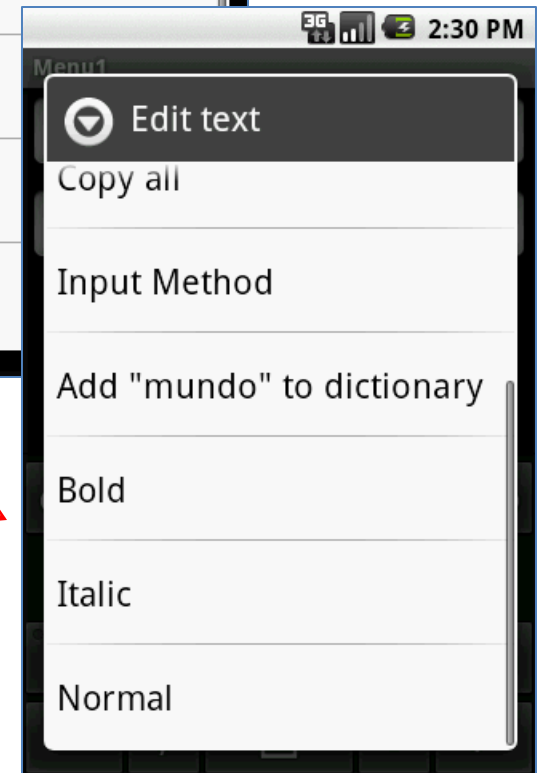
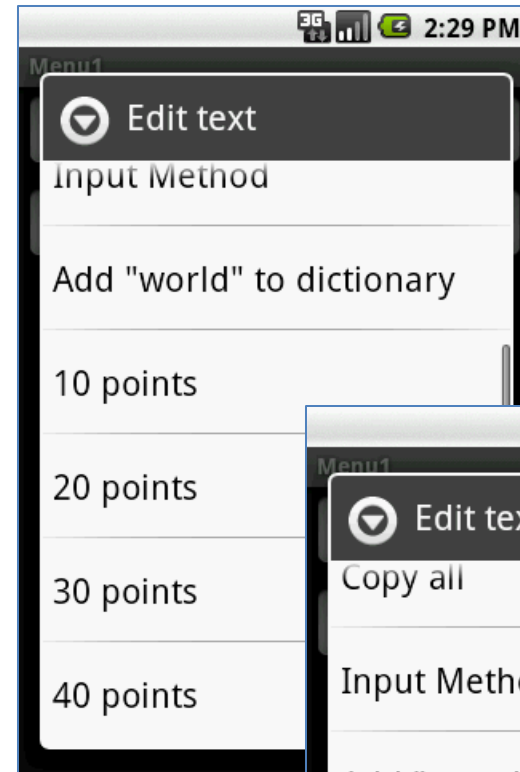
Using Menus

Example 1: Using a **Context Menu**

Each view could have an associated Context Menu



Long-press a
textbox to
invoke its
Context Menu



Using Menus

Example 1: Using Option and Context Menu

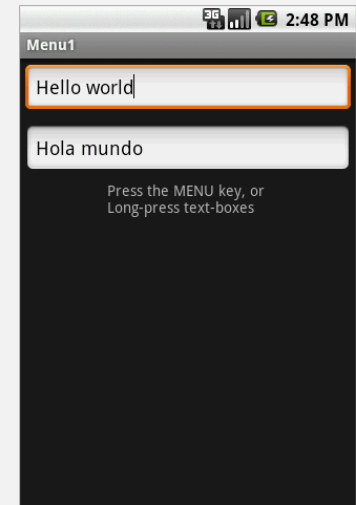
The app shows two text boxes. Menus are used to change text's size, color, and style.

```
<?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>
```



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 Menu1Act1 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
```



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
```



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
```

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.

```
// called whenever an item in your context menu is selected
@Override
public boolean onContextItemSelected(MenuItem item) {
    return (applyMenuOption(item) ||
            super.onContextItemSelected(item) );
}

// called whenever an item in your options menu is selected
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    return (applyMenuOption(item) ||
            super.onOptionsItemSelected(item) );
}
```

Note: `boolean` event observers of the type `onEvent(...)` by convention return **true** to indicate the triggered event has been consumed by the method, when **false** is returned the event is still alive and could be consumed by others.

Using Menus

Example 1: Using Option and Context Menu

```
// apply the action associated to selected item
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 box text1 or style on text2
        if (menuItemId == 6)
            etMessage1.setTextColor(color.background_dark | Color.RED);
            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
```



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

} //Menu1Act1
```




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



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.



Using Menus

Comments on Creating an Option & Context Menu

In Example1 the event observers:

onOptionsItemSelected() (for options menu selections) and
onContextItemSelected() (for context menu selections)

delegate on the *applyMenuChoice(...)* method the performing of the actions associated to the choices made by the user.

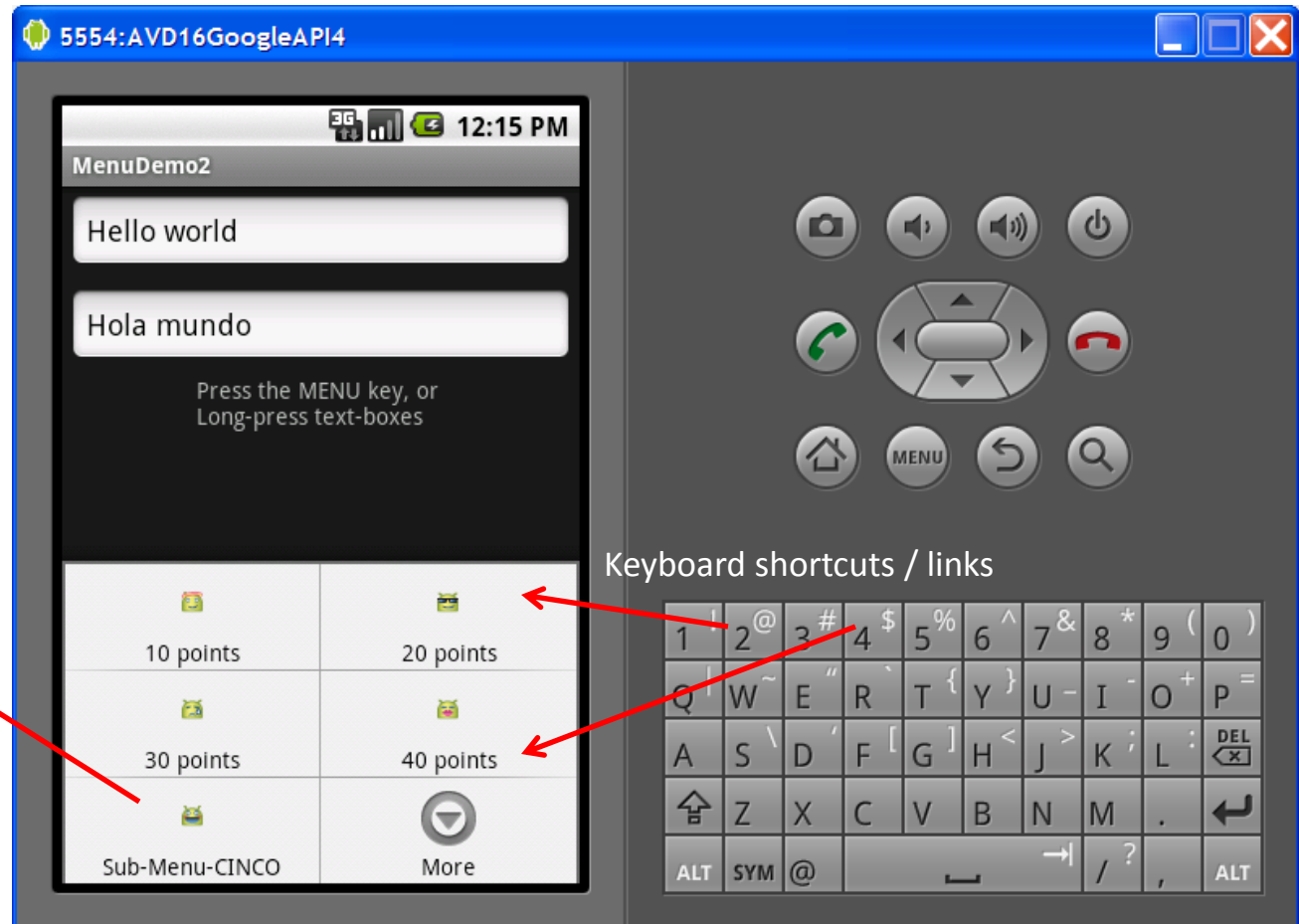
```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    return (applyMenuChoice(item) );
}

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

Using Menus

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.

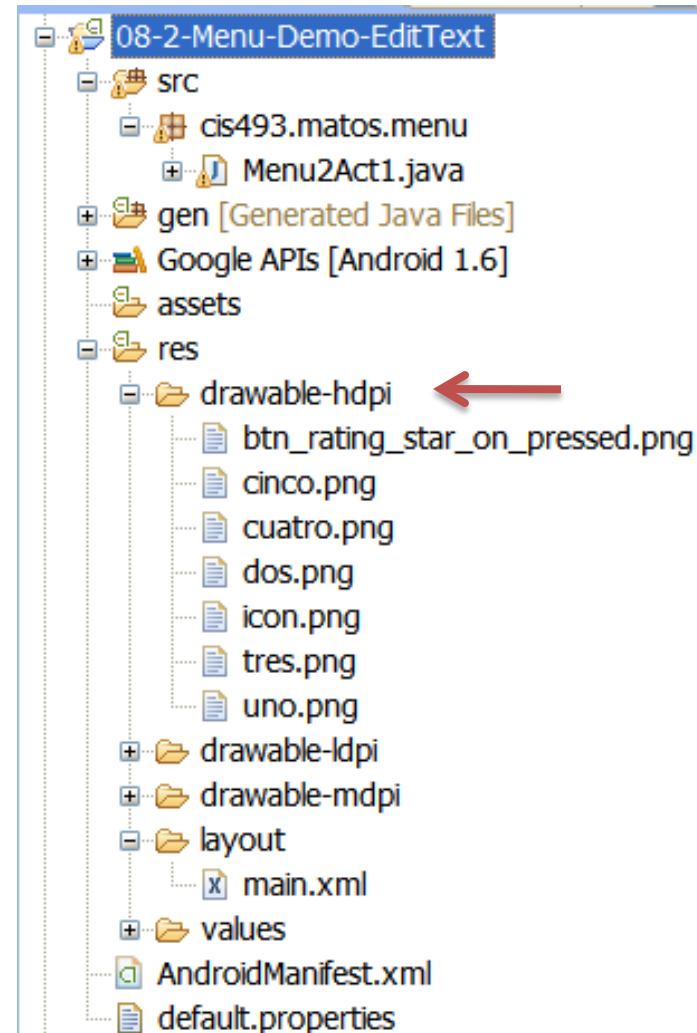
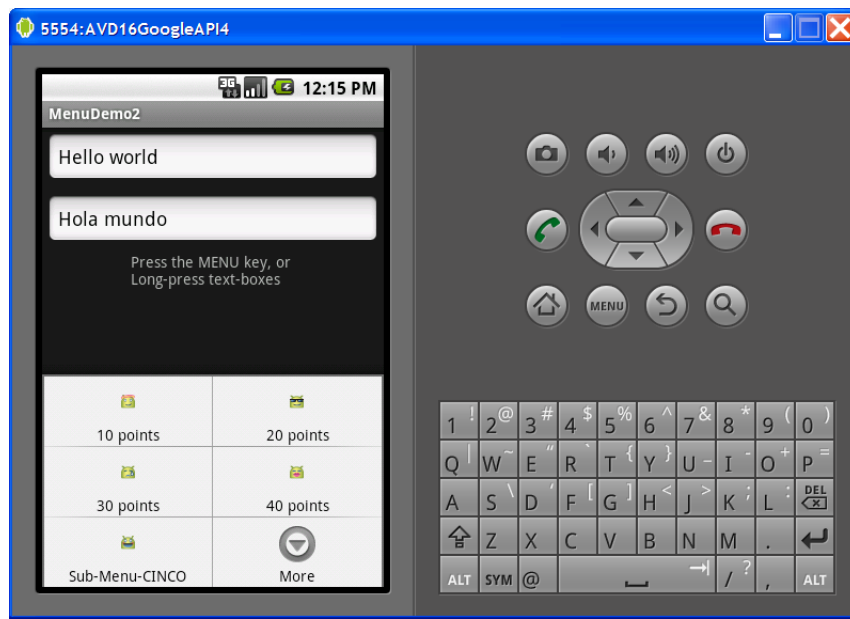


Click on **SubMenu** to see additional options

Using Menus

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.



Using Menus

Example 2: Enhancing Option/Context Menu

```
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');
```

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

Remove this line from previous version

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



Using Menus

Example 2: Enhancing Option/Context Menu

Continuation...

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

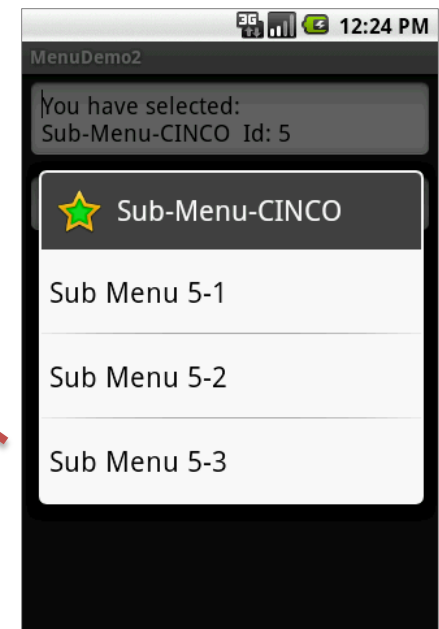
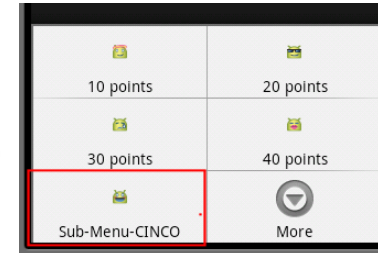


```
// adding a sub-menu as fifth entry of this menu
// .addSubMenu(int groupId, int itemId, int order, CharSequence title)
int smGroupId = 0; // don't care, same as Menu.NONE
int smItemId = 5;  // fifth element
int smOrder = 5;   // don't care, same as Menu.NONE
```

```
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
```



Using Menus

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()
            + "\nId: " + menuItemId
            + " order: " + item.getOrder() );
    }

    // 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

Using Menus

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(uglify(strMsg2, "BOLD")); //bold
else if (menuItemId == 10)
    etMessage2.setText(uglify(strMsg2, "ITALIC")); //italic
else if (menuItemId == 11)
    etMessage2.setText(uglify(strMsg2, "NORMAL")); //normal

return false;

} //applyMenuOption

```

←
Same as
before



Using Menus

Questions ?