

Module 3 : UI Design and Data storage

Options Menu

You are now ready to modify the application to display the options menu when the user presses the MENU button on the Android device.

```
package net.learn2develop.Menus;
import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.Toast;
public class MainActivity extends Activity {
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
super.onCreateOptionsMenu(menu);
CreateMenu(menu);
return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item)
{
return MenuChoice(item);
}
private void CreateMenu(Menu menu)
{
//...
```

```
}  
private boolean MenuChoice(MenuItem item)  
{  
//...  
}  
}
```

Figure 5-13 shows the options menu that pops up when you click the MENU button. To select a menu item, either click on an individual item or use its shortcut key (A to D; applicable only to the first four items).

How It Works

To display the options menu for your activity, you need to override two methods in your activity: `onCreateOptionsMenu()` and `onOptionsItemSelected()`. The `onCreateOptionsMenu()` method is called when the MENU button is pressed. In this event, you call the `CreateMenu()` helper method to display the options menu. When a menu item is selected, the `onOptionsItemSelected()` method is called. In this case, you call the `MenuChoice()` method to display the menu item selected (and do whatever you want to do). Observe the icons displayed for menu items 1, 2, and 3. Also, if the options menu has more than six items, a “More” menu item will be displayed to indicate the additional options. Figure 5-14 shows the additional menu items displayed as a list, after the user has pressed the More menu item.



Figure 5-13

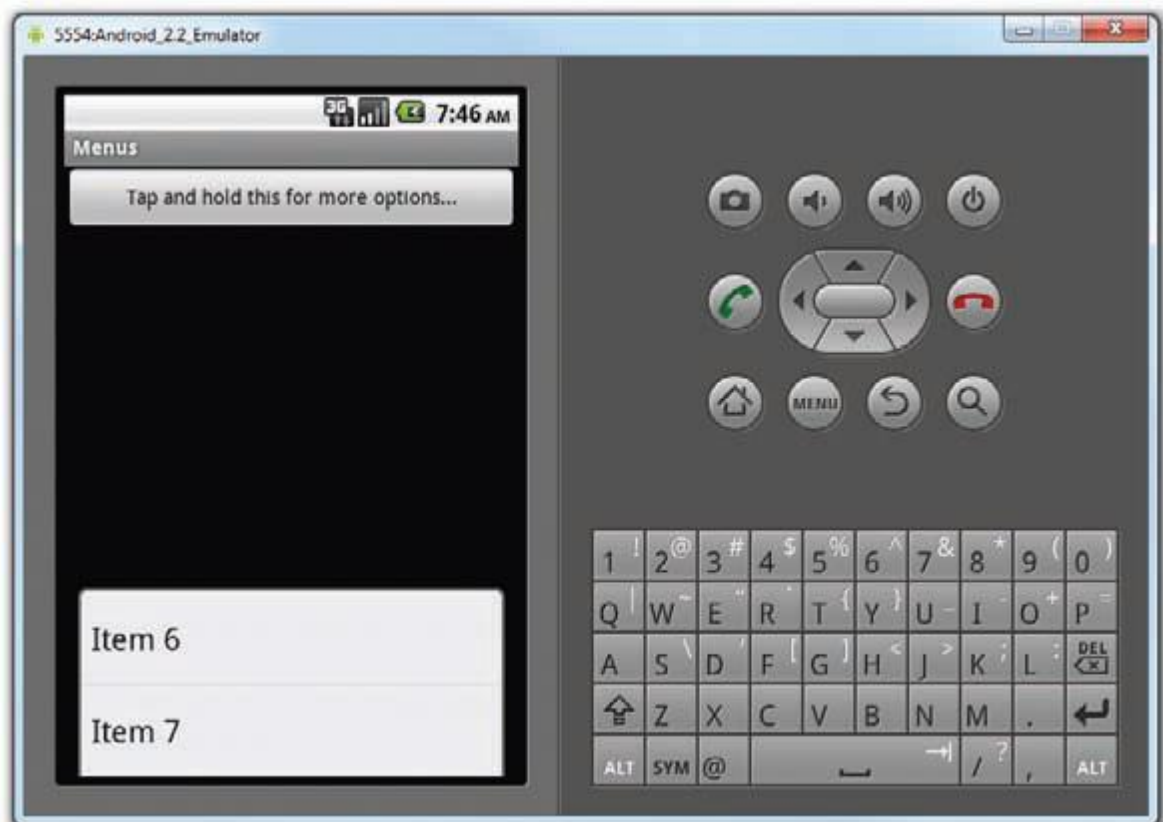


Figure 5-14

Context Menu

A context menu is usually associated with a view on an activity, and it is displayed when the user long clicks an item. For example, if the user taps on a Button view and hold it for a few seconds, a context menu can be displayed. If you want to associate a context menu with a view on an activity, you need to call the `setOnCreateContextMenuListener()` method of that particular view.

The following Try It Out shows how you can associate a context menu with a Button view.

```
package net.learn2develop.Menus;
import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.Toast;
import android.view.View;
import android.view.ContextMenu;
import android.view.ContextMenu.ContextMenuInfo;
public class MainActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button btn = (Button) findViewById(R.id.btn1);
        btn.setOnCreateContextMenuListener(this);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        super.onCreateOptionsMenu(menu);
        CreateMenu(menu);
        return true;
    }
    @Override
```

```

public boolean onOptionsItemSelected(MenuItem item)
{
return MenuChoice(item);
}
@Override
public void onCreateContextMenu(ContextMenu menu, View view,
ContextMenuItem menuInfo)
{
super.onCreateContextMenu(menu, view, menuInfo);
CreateMenu(menu);
}
@Override
public boolean onOptionsItemSelected(MenuItem item)
{
return MenuChoice(item);
}
private void CreateMenu(Menu menu)
{
//...
}
private boolean MenuChoice(MenuItem item)
{
//...
}
}

```

2. Press F11 to debug the application on the Android Emulator. Figure 5-15 shows the context menu that is displayed when you long-click the Button view.



Figure 5-15

How It Works

When the user long-clicks the Button view, the `onCreateContextMenu()` method is called. In this method, you call the `CreateMenu()` method to display the context menu. Similarly, when an item inside the context menu is selected, the `onContextItemSelected()` method is called, where you call the `MenuItem()` method to display a message to the user. Notice that the shortcut keys for the menu items do not work. To enable the shortcuts keys, you need to call the `setQueryMode()` method of the `Menu` object, like this:

```
private void CreateMenu(Menu menu)
{
    menu.setQwertyMode(true);
    MenuItem mnu1 = menu.add(0, 0, 0, "Item 1");
    {
        mnu1.setAlphabeticShortcut('d');
        mnu1.setIcon(R.drawable.icon);
    }
    //...
}
```

Doing so will enable the shortcut key (see Figure 5-16).

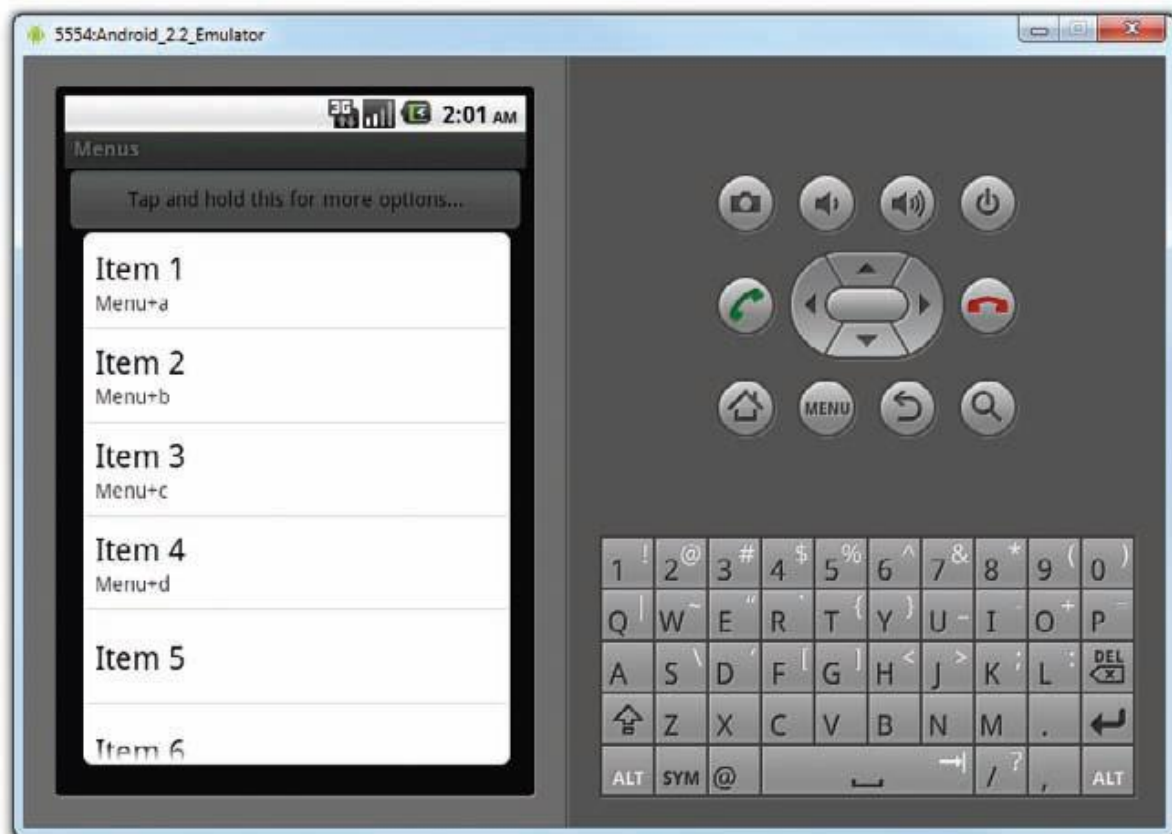


Figure 5-16