

Module 3 : UI Design and Data storage

Using Menus with Views

Menus are useful for displaying additional options that are not directly visible on the main UI of an application. There are two main types of menus in Android:

➤➤ **Options menu** — Displays information related to the current activity. In Android, you activate the options menu by pressing the MENU key.

➤➤ **Context menu** — Displays information related to a particular view on an activity. In Android, to activate a context menu you tap and hold on to it.

Figure 5-11 shows an example of an options menu in the browser application. The option menu is displayed whenever the user presses the MENU button. The menu items displayed vary according to the current activity that is running.

Figure 5-12 shows a context menu that is displayed when the user presses and holds on a hyperlink displayed on the page. The menu items displayed vary according to the component or view currently selected. To activate the context menu, the user selects an item on the screen and either taps and holds it or simply presses the centre button on the directional keypad.



Figure 5.11



Figure 5-12

Creating the Helper Methods

Before you go ahead and create your options and context menus, you need to create two helper methods. One creates a list of items to show inside a menu, while the other handles the event that is fired when the user selects an item inside the menu.

```
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);
    }
    private void CreateMenu(Menu menu)
    {
```

```
MenuItem mnu1 = menu.add(0, 0, 0, "Item 1");
{
mnu1.setAlphabeticShortcut('a');
mnu1.setIcon(R.drawable.icon);
}
MenuItem mnu2 = menu.add(0, 1, 1, "Item 2");
{
mnu2.setAlphabeticShortcut('b');
mnu2.setIcon(R.drawable.icon);
}
MenuItem mnu3 = menu.add(0, 2, 2, "Item 3");
{
mnu3.setAlphabeticShortcut('c');
mnu3.setIcon(R.drawable.icon);
}
MenuItem mnu4 = menu.add(0, 3, 3, "Item 4");
{
mnu4.setAlphabeticShortcut('d');
}
menu.add(0, 3, 3, "Item 5");
menu.add(0, 3, 3, "Item 6");
menu.add(0, 3, 3, "Item 7");
}
private boolean MenuChoice(MenuItem item)
{
switch (item.getItemId()) {
case 0:
Toast.makeText(this, "You clicked on Item 1",
Toast.LENGTH_LONG).show();
return true;
case 1:
Toast.makeText(this, "You clicked on Item 2",
Toast.LENGTH_LONG).show();
return true;
```

```

case 2:
Toast.makeText(this, "You clicked on Item 3",
Toast.LENGTH_LONG).show();
return true;
case 3:
Toast.makeText(this, "You clicked on Item 4",
Toast.LENGTH_LONG).show();
return true;
case 4:
Toast.makeText(this, "You clicked on Item 5",
Toast.LENGTH_LONG).show();
return true;
case 5:
Toast.makeText(this, "You clicked on Item 6",
Toast.LENGTH_LONG).show();
        return true;
case 6:
Toast.makeText(this, "You clicked on Item 7",
Toast.LENGTH_LONG).show();
return true;
}
return false;
}
    }

```

How It Works

The preceding example creates two methods: `CreateMenu()` and `MenuChoice()`. The `CreateMenu()` method takes a `Menu` argument and adds a series of menu items to it. To add a menu item to the menu, you create an instance of the `MenuItem` class and use the `add()` method of the `Menu` object.

```

MenuItem mnu1 = menu.add(0, 0, 0, "Item 1");
{
mnu1.setAlphabeticShortcut('a');
mnu1.setIcon(R.drawable.icon);
}

```

The four arguments of the add() method are as follows:

- groupId — The group identifier that the menu item should be part of. Use 0 if an item is not in a group.
- itemId — Unique item ID
- order — The order in which the item should be displayed
- title — The text to display for the menu item

You can use the setAlphabeticShortcut() method to assign a shortcut key to the menu item so that users can select an item by pressing a key on the keyboard. The setIcon() method sets an image to be displayed on the menu item. The MenuChoice() method takes a MenuItem argument and checks its ID to determine the menu item that is clicked. It then displays a Toast message to let the user know which menu item was clicked.