

# Android Programing

Gavrilut Dragos

# Style

- ▶ Used to define a template for other XML define objects
- ▶ Located in “res\values-xxx\styles.xml”

```
<resources xmlns:android="http://schemas.android.com/apk/res/android">
<style name="myStyle" parent="">
    <item name="android:background">#RGB</item>
    <item name="property_name">property_value</item>
    <item name="property_name">property_value</item>
    ...
    <item name="property_name">property_value</item>
</style>
<style name="another style">
    ...
</style>
...
</resources>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <Button
        style="@style/myStyle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button1" />
</LinearLayout>
```

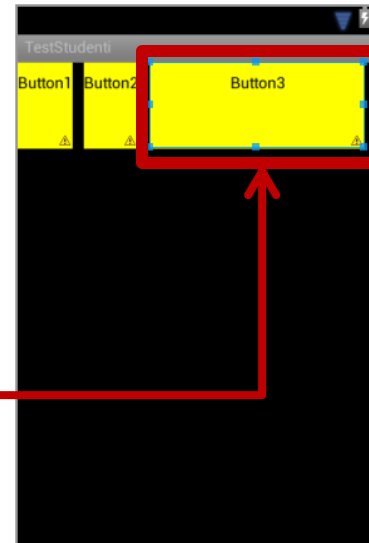
# Style

```
<resources xmlns:android="http://schemas.android.com/apk/res/android">
<style name="myStyle">
  <item name="android:background">#FFFF00</item>
  <item name="android:foreground">#000000</item>
  <item name="android:layout_width">wrap_content</item>
  <item name="android:layout_height">wrap_content</item>
  <item name="android:layout_marginRight">10px</item>
  <item name="android:paddingTop">10px</item>
  <item name="android:paddingBottom">50px</item>
</style>
</resources>
```

```
<LinearLayout
  android:layout_width="fill_parent"
  android:layout_height="fill_parent" >

  <Button style="@style/myStyle"
    android:text="Button1" />
  <Button style="@style/myStyle"
    android:text="Button2" />
  <Button style="@style/myStyle"
    android:layout_width="fill_parent"
    android:text="Button3" />

</LinearLayout>
```



# Style - Calculator

```
<resources xmlns:android="http://schemas.android.com/apk/res/android">

<style name="number">
    <item name="android:background">#808080</item>
    <item name="android:foreground">#000000</item>
    <item name="android:layout_width">70dp</item>
    <item name="android:layout_height">70dp</item>
    <item name="android:layout_marginRight">5dp</item>
    <item name="android:layout_marginLeft">5dp</item>
    <item name="android:layout_marginBottom">10dp</item>
    <item name="android:textSize">30sp</item>
</style>

<style name="operatii" parent="number">
    <item name="android:background">#C0C0C0</item>
</style>

<style name="misc" parent="number">
    <item name="android:background">#C0D0F0</item>
</style>

</resources>
```

# Style - Calculator

```
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:text = "0" android:gravity="right" android:layout_width="fill_parent"
        android:textSize="30sp"
        android:layout_height="30dp"/>

    <TableLayout android:layout_width="match_parent" android:layout_height="match_parent" >
        <TableRow android:layout_width="wrap_content" android:gravity="center_horizontal">
            <Button style="@style/number" android:text="1" />
            <Button style="@style/number" android:text="2" />
            <Button style="@style/number" android:text="3" />
            <Button style="@style/operatii" android:text="+" />
        </TableRow>
        <TableRow android:layout_width="wrap_content" android:gravity="center_horizontal">
            <Button style="@style/number" android:text="4" />
            <Button style="@style/number" android:text="5" />
            <Button style="@style/number" android:text="6" />
            <Button style="@style/operatii" android:text="-" />
        </TableRow>
        <TableRow android:layout_width="wrap_content" android:gravity="center_horizontal">
            <Button style="@style/number" android:text="7" />
            <Button style="@style/number" android:text="8" />
            <Button style="@style/number" android:text="9" />
            <Button style="@style/operatii" android:text="*" />
        </TableRow>
        <TableRow android:layout_width="wrap_content" android:gravity="center_horizontal">
            <Button style="@style/number" android:text="0" android:Layout_span="2"/>
            <Button style="@style/operatii" android:text="/" />
            <Button style="@style/operatii" android:text="." />
        </TableRow>
        <TableRow android:layout_width="wrap_content" android:gravity="center_horizontal">
            <Button style="@style/misc" android:text="C"/>
            <Button style="@style/misc" android:text="=" android:Layout_span="3"/>
        </TableRow>
    </TableLayout>
</LinearLayout>
```



# Theme

- ▶ A style apply to the entire activity and not just some View derived objects
- ▶ Added directly in AndroidManifest.xml when declaring an Activity

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.teststudenti"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="8" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/myStyle" >
        <activity
            android:name="com.example.teststudenti.MainActivity"
            android:label="@string/app_name" >
        </activity>
    </application>
</manifest>
```

# Menus

- ▶ There are 3 types of menus:
  - ▶ Options menus. These menus are associated with the applications (Triggered by the pressed of the “Menu” key (up to version 2.3) or the Menu icon from the action bar starting from 3.0)
  - ▶ Contextual menus. Usually asociated with a long-click over an View derived object
  - ▶ Popup menus

# Menus

- ▶ Can be define using XML (res\menu folder)
- ▶ Usually, it has one of the following items:
  - ▶ **<menu>** → identifies a menu containter
  - ▶ **<item>** → identifies a item within a menu container
  - ▶ **<group>** → used to define a group of several items. All of the items in a group share some properties (visibility, active state, ...)



# Menus

- ▶ Every item in a menu has the following XML attributes:

XML Attribute	Description
android:id	Item id (used by <code>findViewById</code> function)
android:onClick	Name of public method “ <u><b>public void &lt;method&gt; (MenuItem)</b></u> ”
android:orderInCategory	Specifies the position/order of the item in a menu/group
android:title	Menu text
android:titleCondensed	If menu text is too long, titleCondensed is used
android:icon	Menu icon
android:alphabeticShortcut	Alphabetic shortcut (char)
android:numericShortcut	Numeric shortcut (integer)
android:checkable	Boolean. True if the item is checkable
android:checked	Boolean. True if the item is checked
android:visible	Boolean. True if the item is visible
android:enabled	Boolean. True if the item is enabled

# Options Menu

## ► Create

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
```

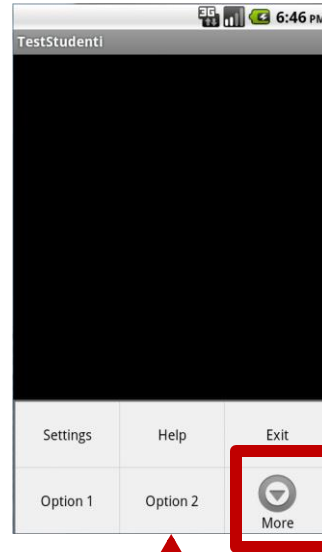
## ► Handle events

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

## ► Or specific event function for each item

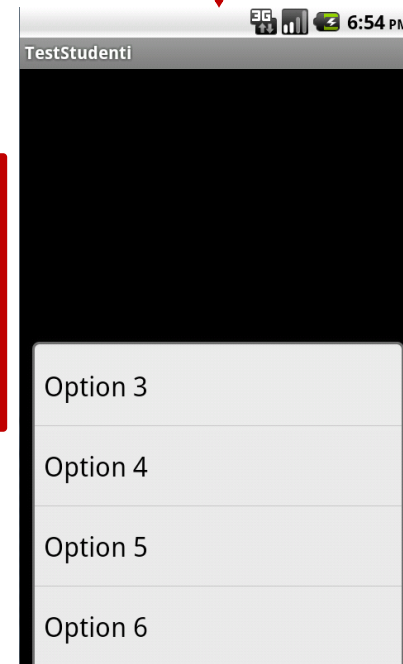
# Options Menu

```
<menu>
  <item android:title="Settings"/>
  <item android:title="Help" />
  <item android:title="Exit" />
</menu>
```



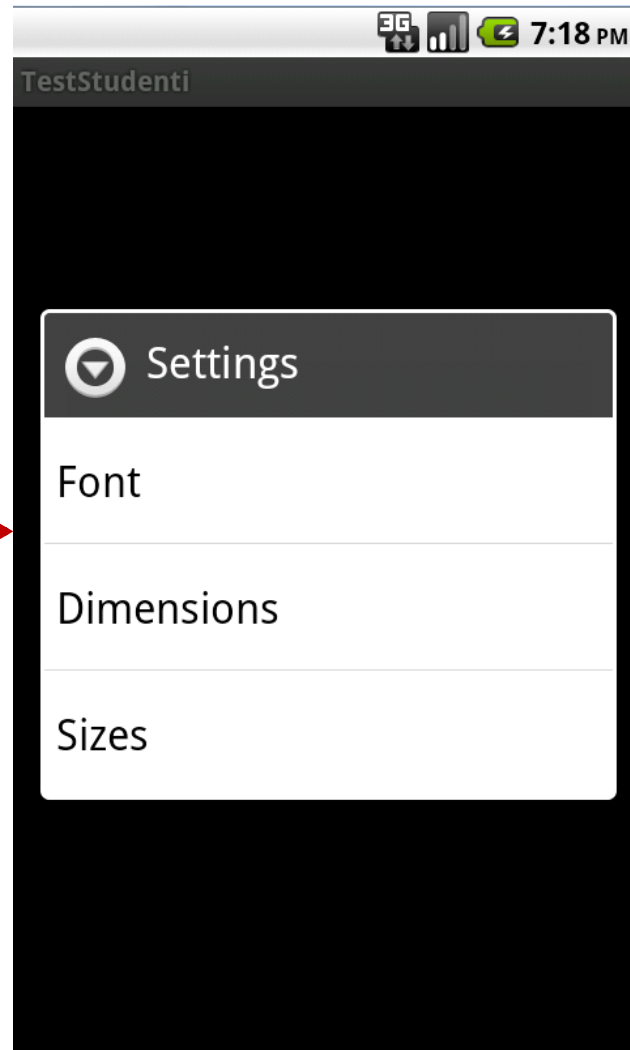
```
<menu>

  <item android:title="Settings"/>
  <item android:title="Help" />
  <item android:title="Exit" />
  <item android:title="Option 1" />
  <item android:title="Option 2" />
  <item android:title="Option 3" />
  <item android:title="Option 4" />
  <item android:title="Option 5" />
  <item android:title="Option 6" />
</menu>
```



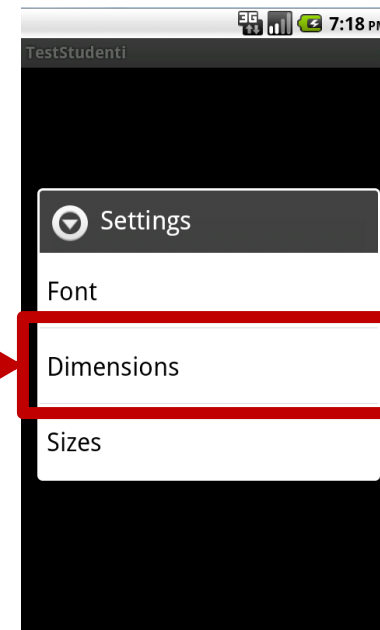
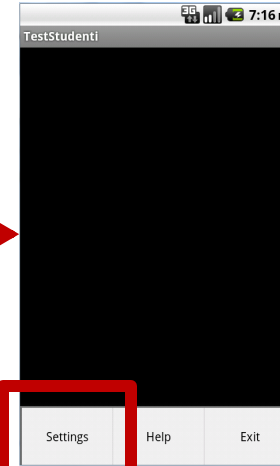
# Options Menu (submenus)

```
<menu>
  <item android:title="Settings">
    <menu >
      <item android:title="Font" />
      <item android:title="Dimensions" />
      <item android:title="Sizes" />
    </menu>
  </item>
  <item android:title="Help" />
  <item android:title="Exit" />
</menu>
```



# Options Menu (submenus)

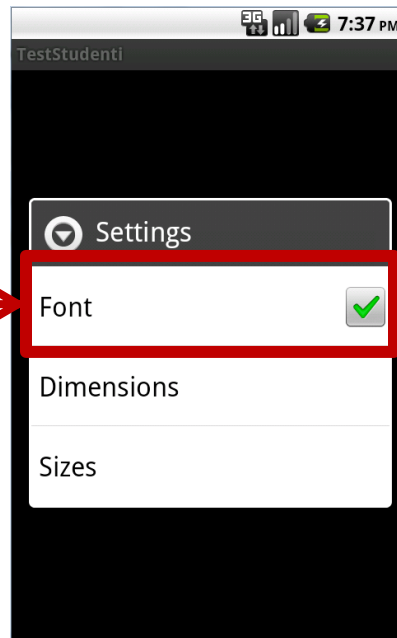
```
<menu>
  <item android:title="Settings">
    <menu>
      <item android:title="Font" />
      <item android:title="Dimensions">
        <menu>
          <item android:title="Pixels" />
          <item android:title="Inchs" />
          <item android:title="Mm" />
          <item android:title="Km" />
          <item android:title="Wrap" />
        </menu>
      </item>
      <item android:title="Sizes" />
    </menu>
  </item>
  <item android:title="Help" />
  <item android:title="Exit" />
</menu>
```



java.lang.UnsupportedOperationException:  
Attempt to add a sub-menu to a sub-menu.

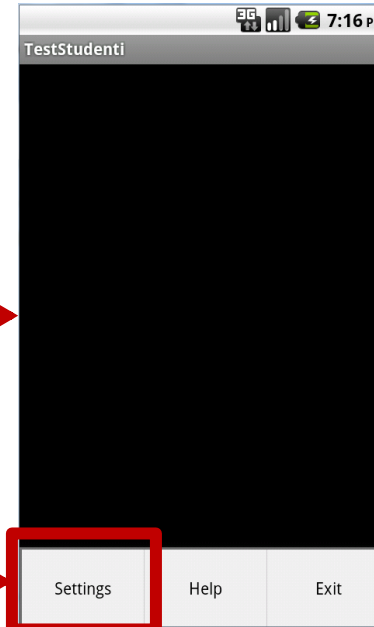
# Options Menu (submenus)

```
<menu>
  <item android:title="Settings">
    <menu>
      <item android:title="Font"
        android:checkable="true"
        android:checked="true" />
      <item android:title="Dimensions" />
      <item android:title="Sizes" />
    </menu>
  </item>
  <item android:title="Help" />
  <item android:title="Exit" />
</menu>
```



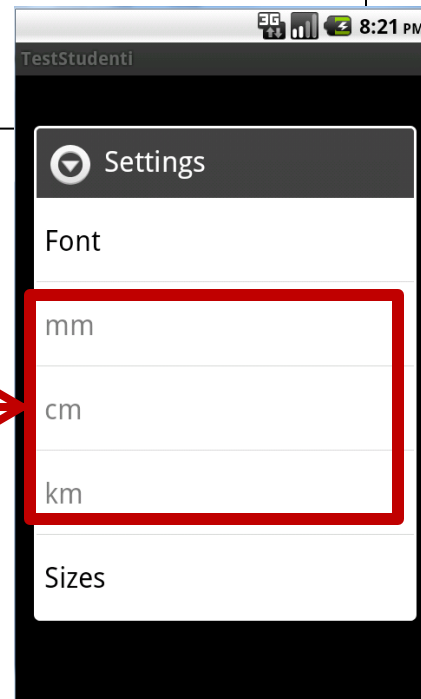
# Options Menu (groups)

```
<menu>
  <item android:title="Settings">
    <menu>
      <item android:title="Font" />
      <group android:checkableBehavior="single">
        <item android:title="mm" />
        <item android:title="cm" />
        <item android:title="km" />
      </group>
      <item android:title="Sizes" />
    </menu>
  </item>
  <item android:title="Help" />
  <item android:title="Exit" />
</menu>
```



# Options Menu (groups)

```
<menu>
  <item android:title="Settings">
    <menu>
      <item android:title="Font" />
      <group android:enabled="false">
        <item android:title="mm" />
        <item android:title="cm" />
        <item android:title="km" />
      </group>
      <item android:title="Sizes" />
    </menu>
  </item>
  <item android:title="Help" />
  <item android:title="Exit" />
</menu>
```





# Options Menu

## ► Handle menu events

```
<menu>
  <item
    android:id = "@+id/MenuSettings"
    android:title="Settings" />
  <item
    android:id = "@+id/MenuHelp"
    android:title="Help" />
  <item
    android:id = "@+id/MenuExit"
    android:title="Exit" />
</menu>
```

```
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item)
    {
        switch (item.getItemId())
        {
            case R.id.MenuSettings:
                Log.v("tag", "settings pressed");
                break;
            case R.id.MenuHelp:
                Log.v("tag", "help pressed");
                break;
            case R.id.MenuExit:
                Log.v("tag", "exit pressed");
                break;
        }
        return true;
    }
}
```

# Options Menu

## ► Handle menu events

```
<menu>
  <item
    android:id = "@+id/MenuSettings"
    android:title="Settings" />
  <item
    android:id = "@+id/MenuHelp"
    android:title="Help" />
  <item
    android:id = "@+id/MenuExit"
    android:onClick="OnExit"
    android:title="Exit" />
</menu>
```

```
public class MainActivity extends Activity {

    @Override
    public boolean onOptionsItemSelected(MenuItem item)
    {
        switch (item.getItemId())
        {
            case R.id.MenuSettings:
                Log.v("tag", "settings pressed");
                break;
            case R.id.MenuHelp:
                Log.v("tag", "help pressed");
                break;
            case R.id.MenuExit:
                Log.v("tag", "exit pressed");
                break;
        }
        return true;
    }

    public boolean OnExit(MenuItem item)
    {
        Log.v("tag", "OnExit pressed");
        return true;
    }
}
```

API Level  $\geq 11$

# Contextual Menus

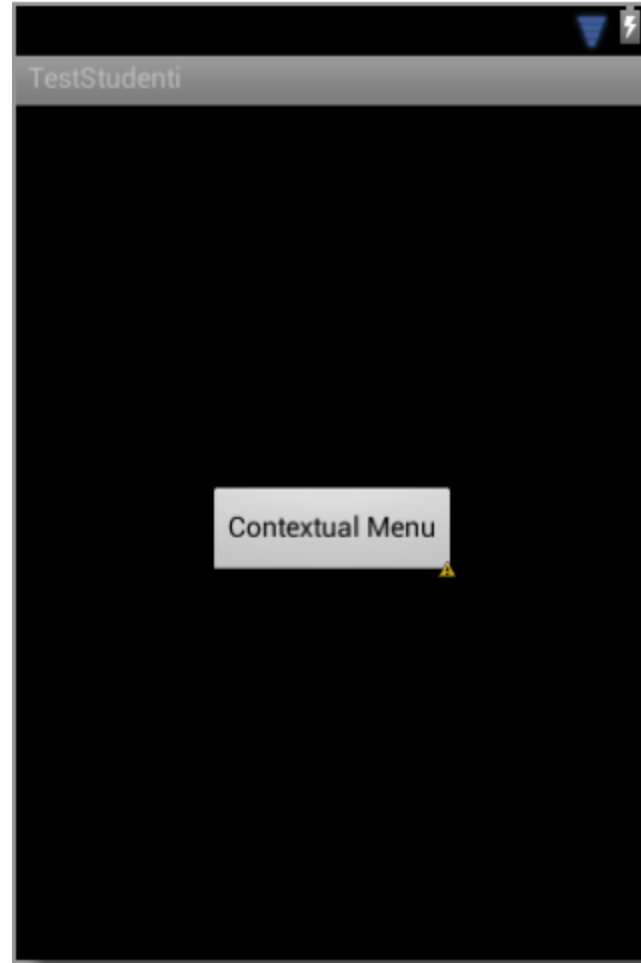
- ▶ Menus associated with a View derived object
- ▶ To create a contextual menu 3 steps must be performed
  1. Register a View derived object to receive a contextual menu (using `registerForContextMenu` api function)
  2. Overwrite `onCreateContextMenu` function (this function will be called for each registered View to set the Menu associated with it)
  3. Overwrite `onContextItemSelected` function to handle menu selection events

# Contextual Menus - Example

```
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:gravity="center" >

    <Button
        android:id="@+id/MyButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Contextual Menu"
        android:onClick="ShowContextualMenu" />

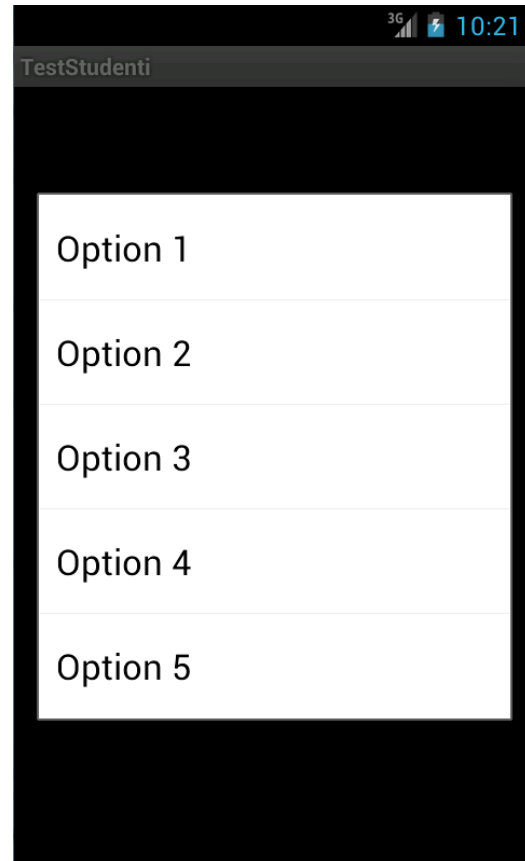
</LinearLayout>
```



# Contextual Menus - Example

- Create a new XML menu (res\menu\contextual\_menu.xml)

```
<menu>
  <item
    android:id = "@+id/context_opt1"
    android:title="Option 1" />
  <item
    android:id = "@+id/context_opt2"
    android:title="Option 2" />
  <item
    android:id = "@+id/context_opt3"
    android:title="Option 3" />
  <item
    android:id = "@+id/context_opt4"
    android:title="Option 4" />
  <item
    android:id = "@+id/context_opt5"
    android:title="Option 5" />
</menu>
```



# Contextual Menus - Example

```
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button b = (Button)findViewById(R.id.MyButton);
        registerForContextMenu(b);
    }
    @Override
    public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo) {
        super.onCreateContextMenu(menu, v, menuInfo);
        getMenuInflater().inflate(R.menu.contextual_menu, menu);
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.context_opt1:
                Log.v("tag", "contextual menu 1 pressed");
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }
    public void ShowContextualMenu(View btn) {
        btn.showContextMenu();
    }
}
```

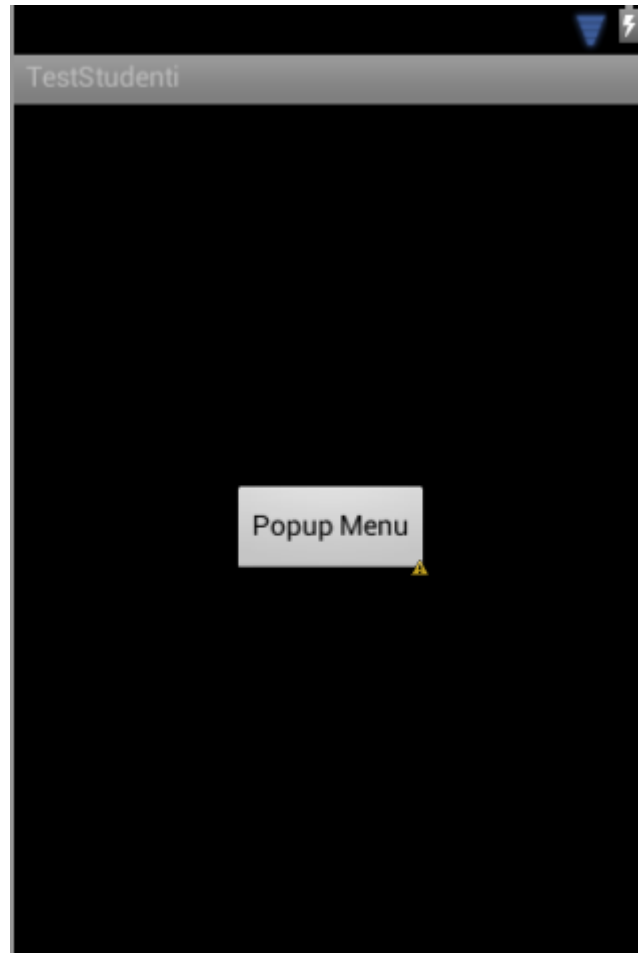
# Popup menus

## ► Menus linked to a View (API Level $\geq 11$ )

```
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:gravity="center" >

    <Button
        android:id="@+id/MyButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Popup Menu"
        android:onClick="ShowPopupMenu" />

</LinearLayout>
```



# Popup menus

```
public class MainActivity extends Activity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
    public void ShowPopupMenu(View btn) {  
        PopupMenu popup = new PopupMenu(this, btn);  
        popup.getMenuInflater().inflate(R.menu.contextual_menu, popup.getMenu());  
        popup.setOnMenuItemClickListener(new OnMenuItemClickListener() {  
            @Override  
            public boolean onMenuItemClick(MenuItem item) {  
                // perform action for selected menu item  
                return false;  
            }  
        });  
        popup.show();  
    }  
}
```

