

Mobile Application Development



04 – Menu

Objectives

- To learn how to construct Menus in your application.
- To learn the different kind of menus provided by Android

Menus

- What are Menu for?
 - To be used as the secondary method to expose an application's functionality.
 - What is the primary method?

Menus and Menu Items

- Menus are containers for
 - Menu items
 - Sub menu
- Menu items may include the following
 - Text, Icon, Radio button, Check box
- Implementation
 - XML and / or Kotlin code

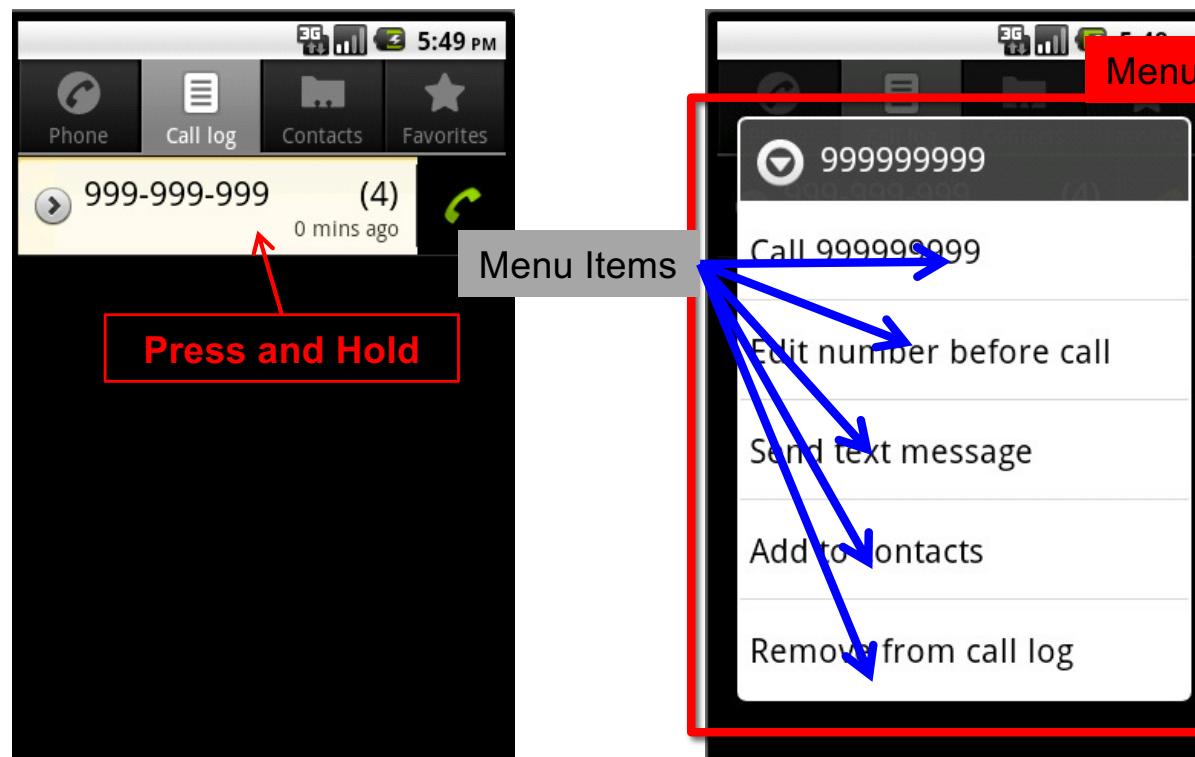
Kinds of Menu

- Option Menu
 - For application that is built for Android 2.3.x or lower the contents of your option menu appear at the bottom of the screen when the user presses the Menu Button
 - For Android 3.0 and above, items from the options menu are available in the action bar
- Contextual Menus
 - Floating contextual
 - Contextual action mode
 - Available in Android 3.0 and above
 - The preferred way

MENUS

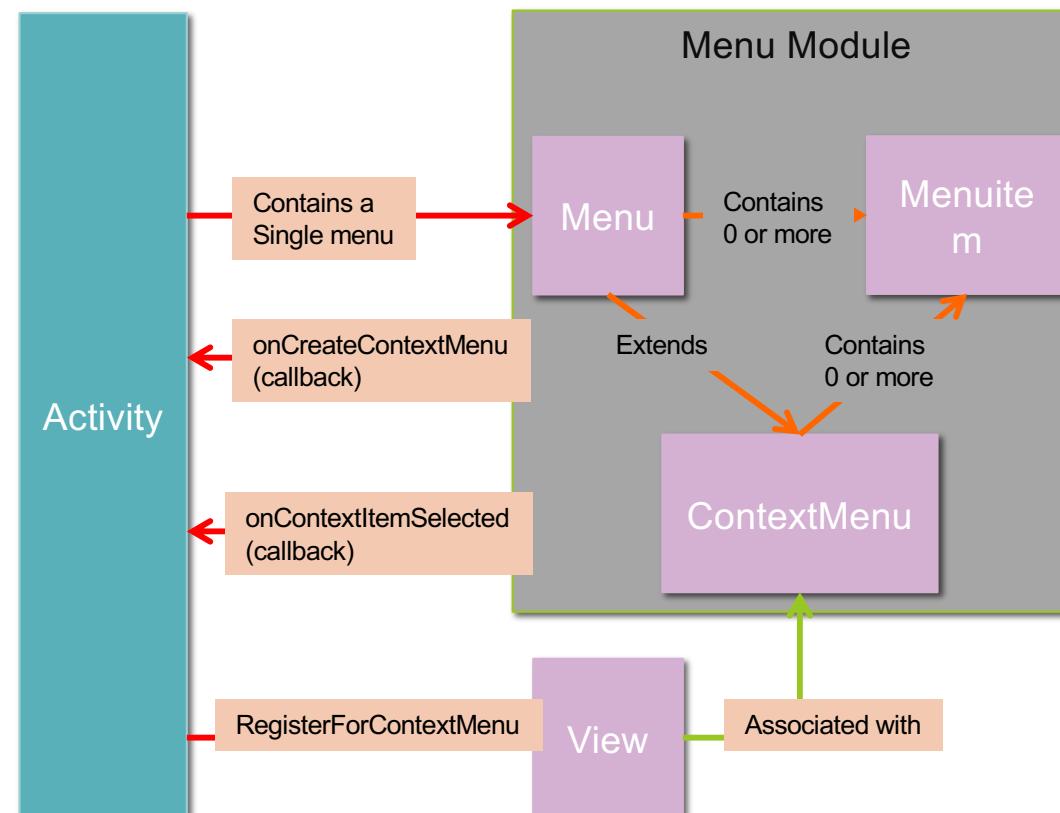
Context Menu

Context Menu

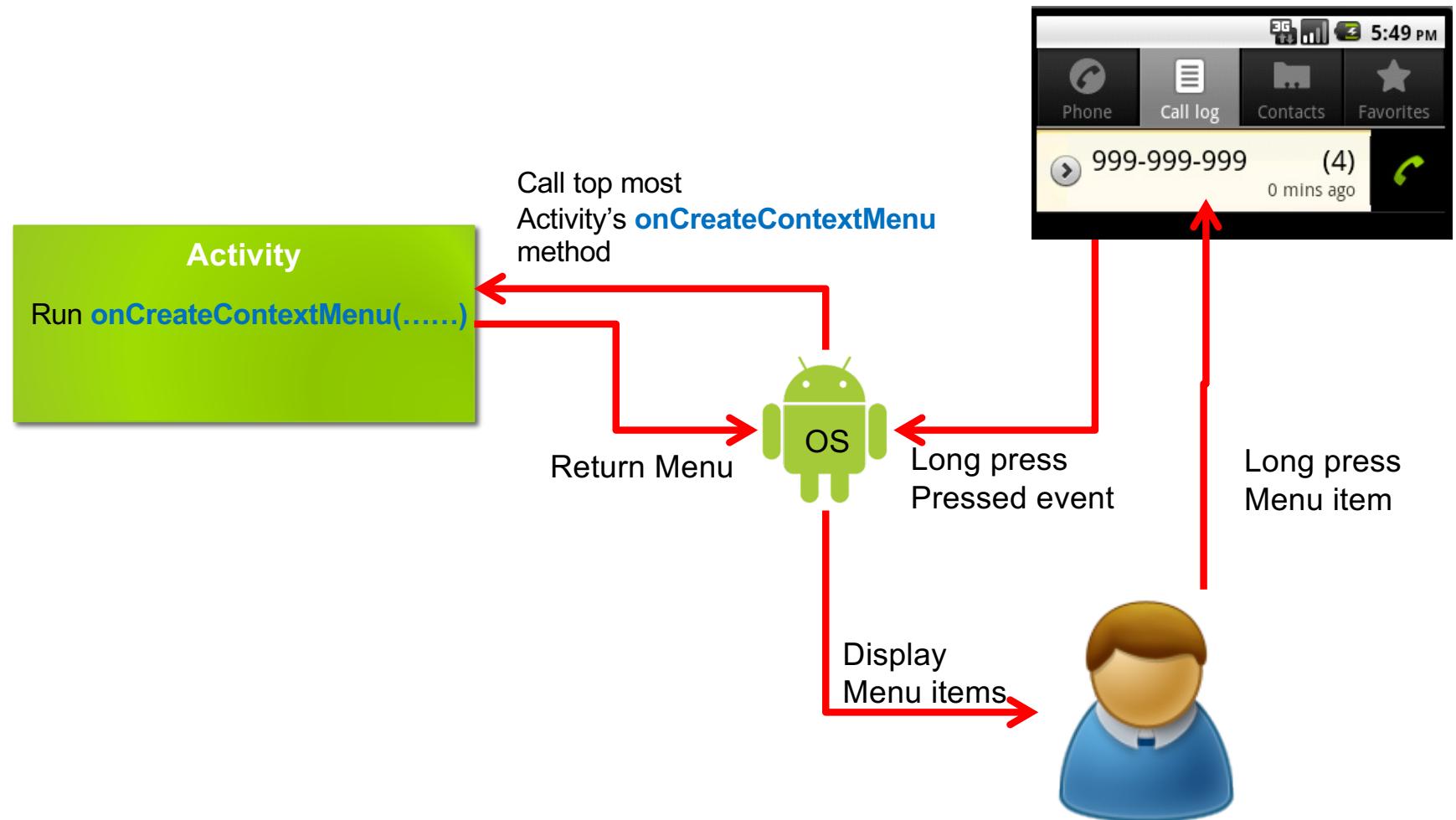


Context Menus

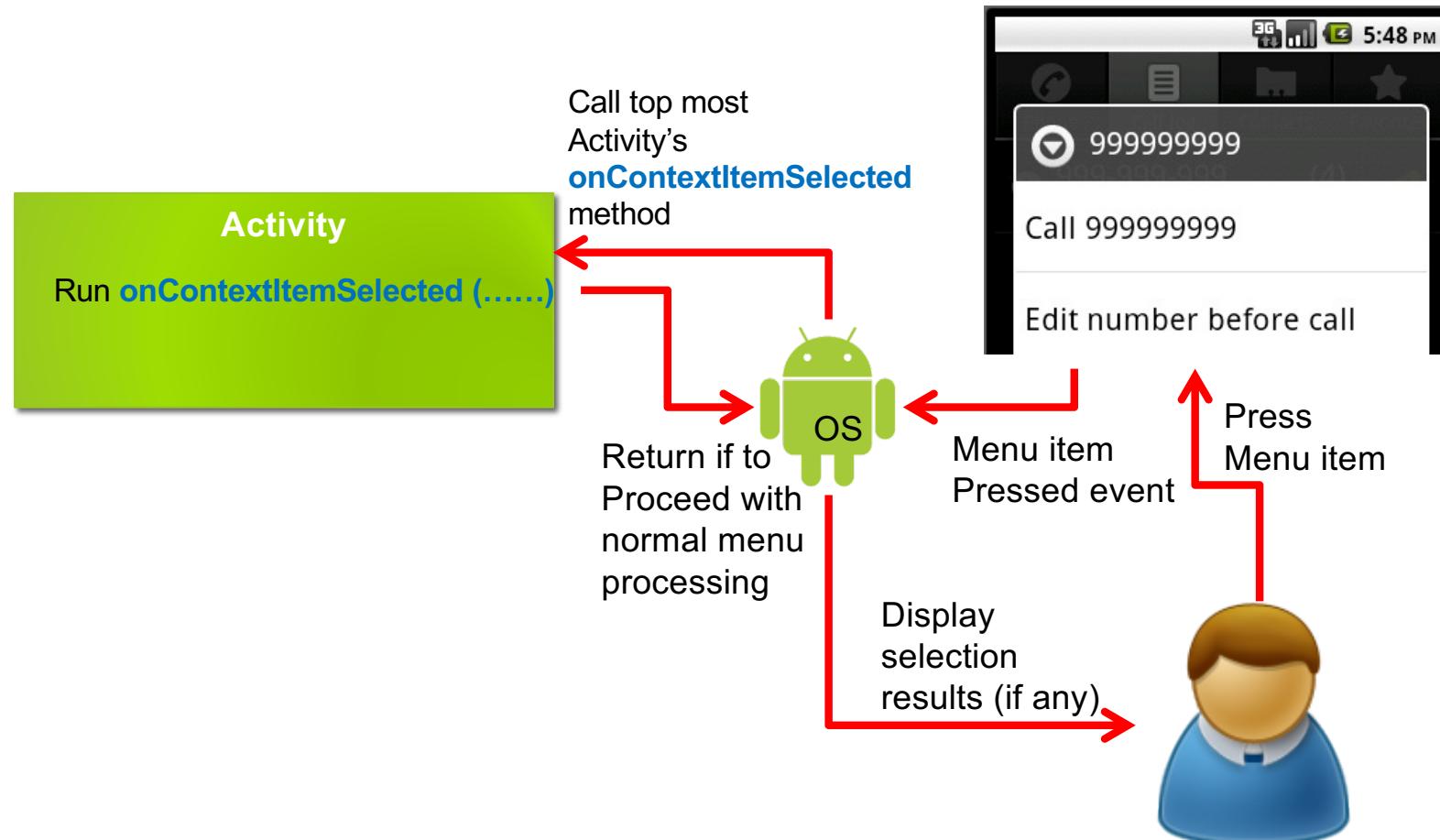
- Owned by a **View** not by the activity
 - Can have 3 views but only 1 of them owns a context menu
- Method to populate the context menu is in the **Activity** class
- ContextMenuInfo contains extra information from the view to the create method
- No sub-menu



Menu Creation



Menu Item Selection



Populating Menu - Kotlin

```
override fun onCreateContextMenu(menu: ContextMenu?, v:  
View?, menuInfo: ContextMenu.ContextMenuItem?) {  
    super.onCreateContextMenu(menu, v, menuInfo)  
  
    if(v?.id == R.id.tvDemo) {  
        menu?.add(1, 2001, 1, "Context Item 1")  
        menu?.add(1, 2002, 2, "Context Item 2")  
    }  
}
```

Populating Menu - Kotlin

```
override fun onContextItemSelected(item: MenuItem?): Boolean {  
  
    if(item?.itemId == 2001 || item?.itemId == 2002)  
    {  
        var itemid = item?.itemId  
  
        Toast.makeText(this,"Context Item $itemid selected ",Toast.LENGTH_SHORT).show()  
  
    }  
  
    return super.onContextItemSelected(item)  
}
```

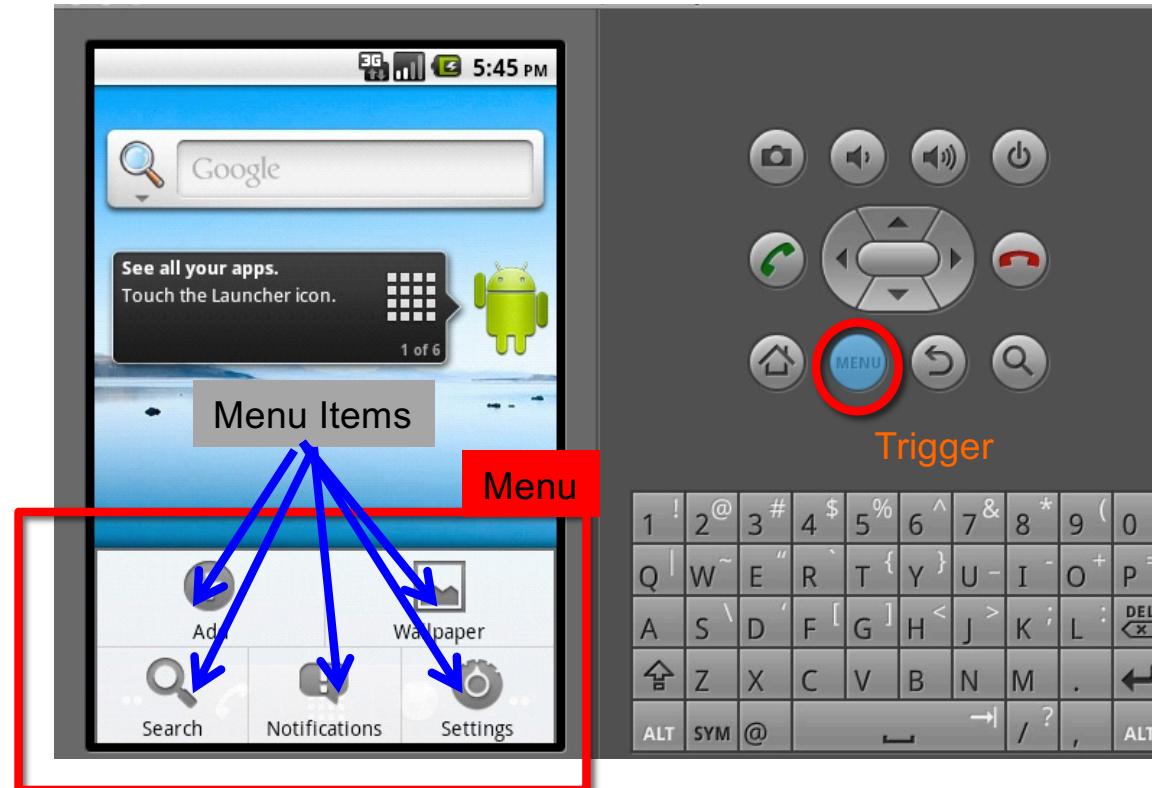
Populating Menu - Kotlin

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
  
    registerForContextMenu(tvDemo)  
}
```

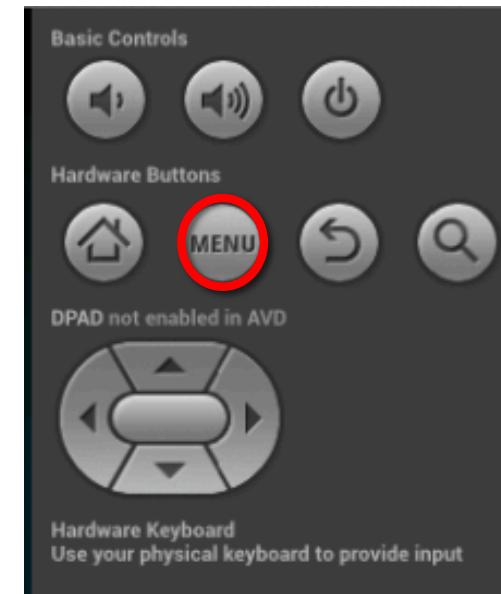
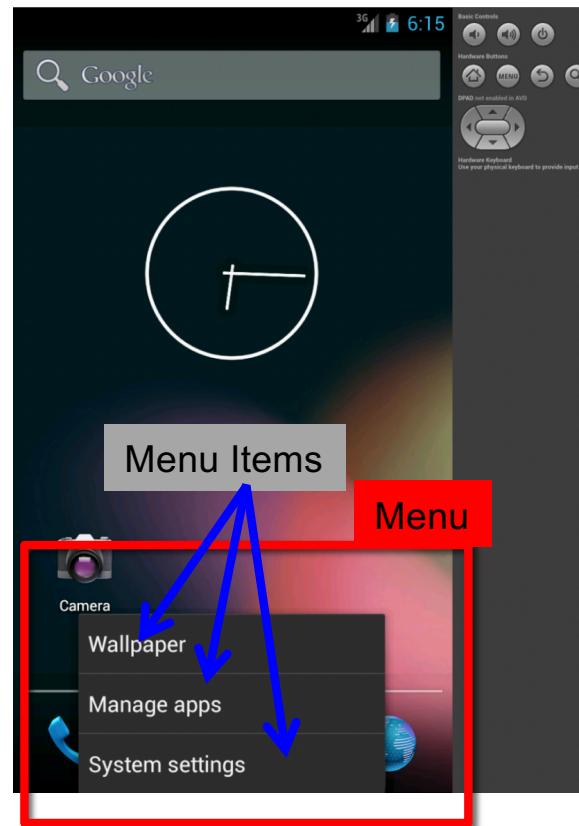
MENUS

OptionsMenu

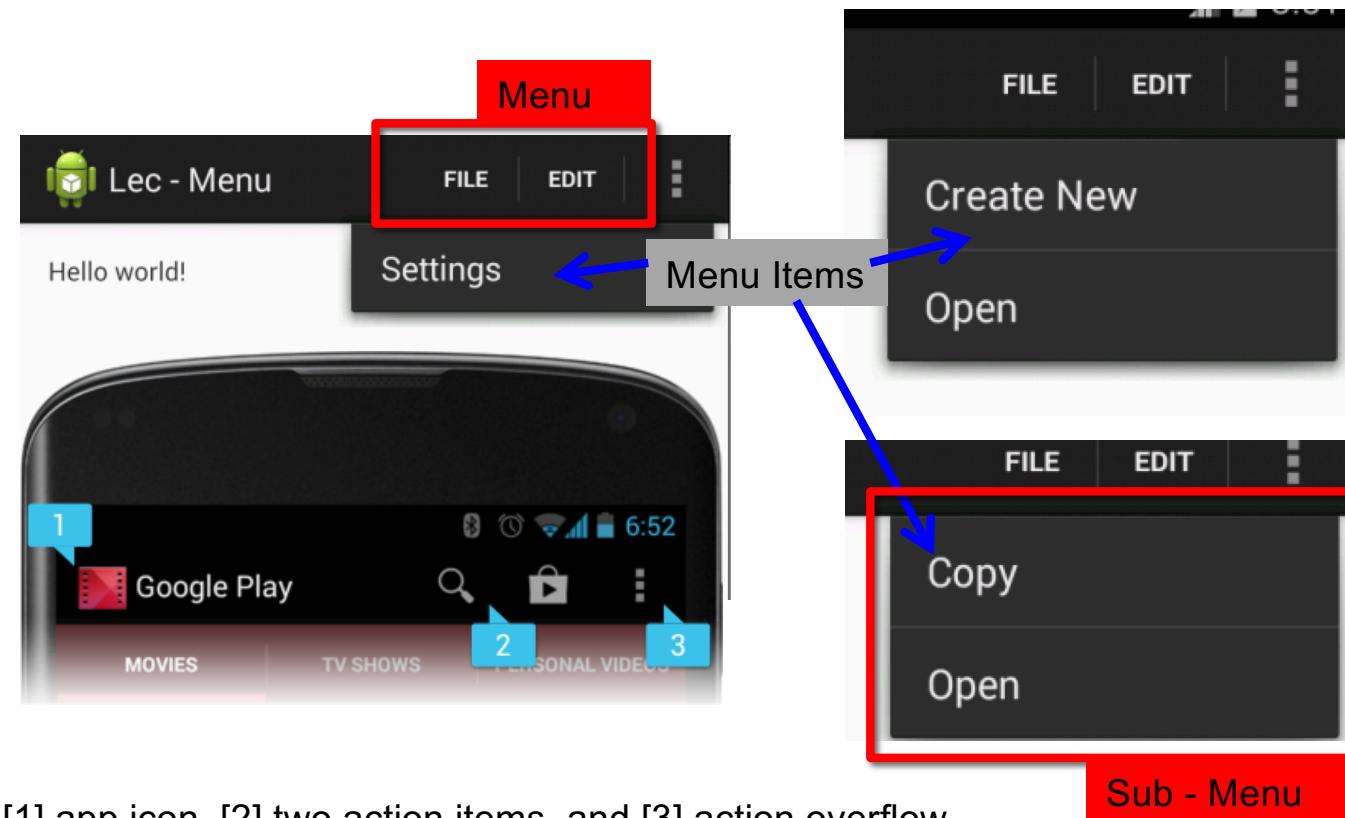
Option Menu



Option Menu



Action Bar

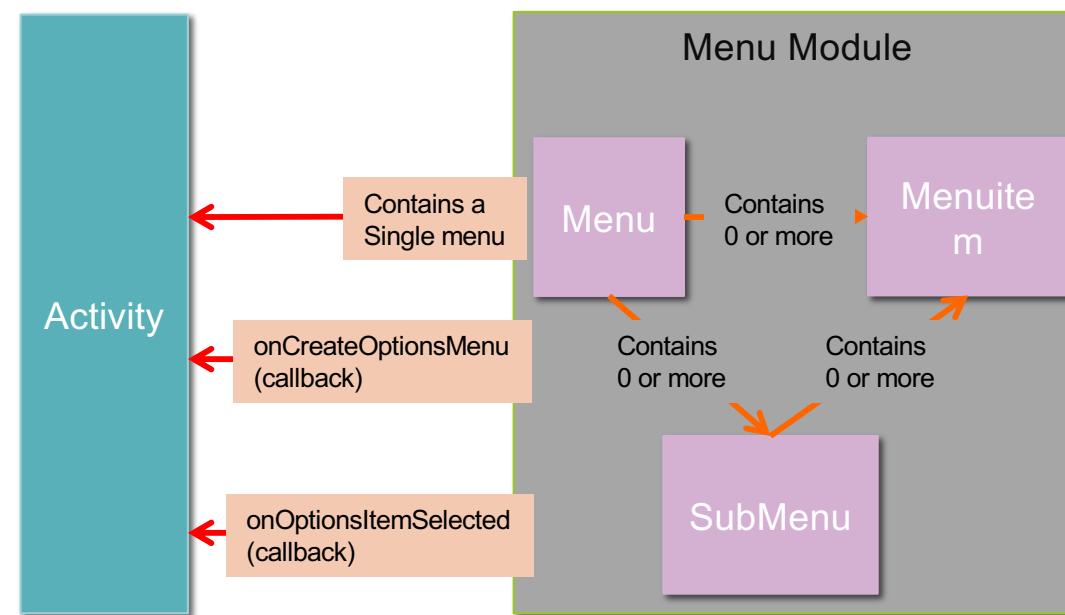


[1] app icon, [2] two action items, and [3] action overflow.

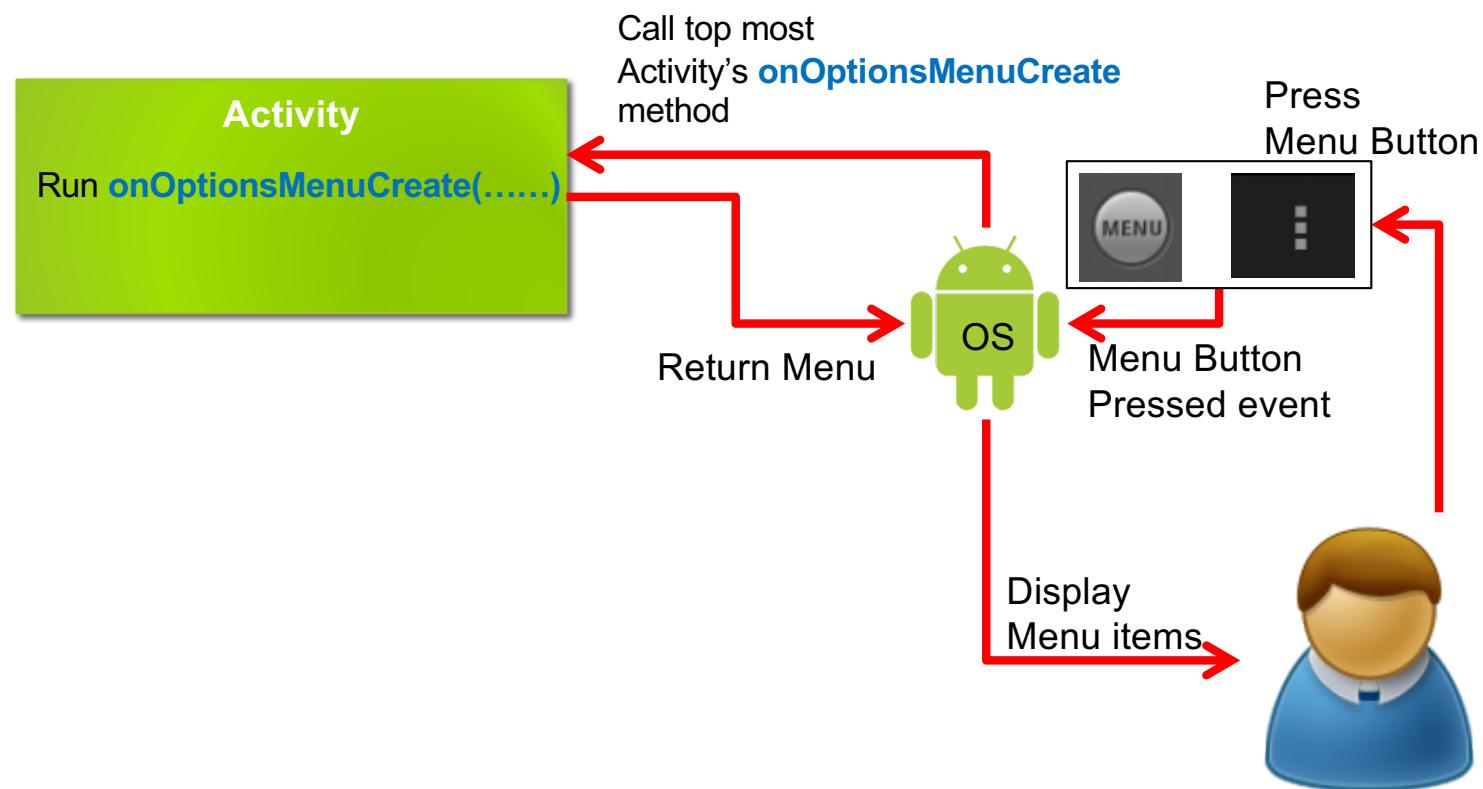
Sub - Menu

Options Menu

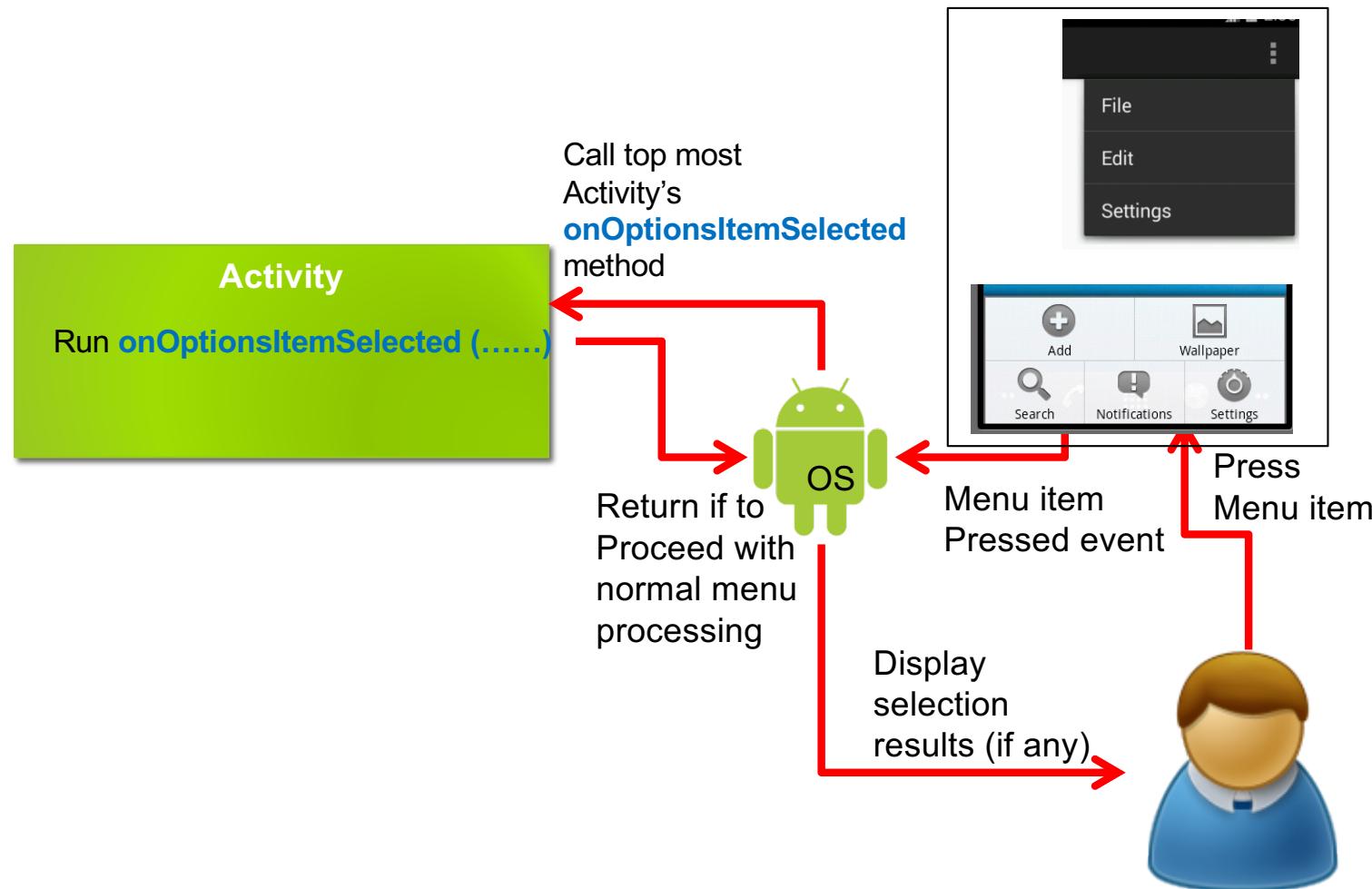
- Key class
`android.view.Menu`
- Associated with an Activity.
- Can contain a number of menu and menu items.
- Call back to activity
 - When menu is to be created.
 - When menu item is selected.



Menu Creation



Menu Item Selection



Populating Options Menu

```
override fun onCreateOptionsMenu(menu: Menu?):  
Boolean {  
  
    menu?.add(1,1001,1,"Options Item 1")  
    menu?.add(1,1002,2,"Options Item 2")  
  
    return super.onCreateOptionsMenu(menu)  
}
```

Populating Menu - Kotlin

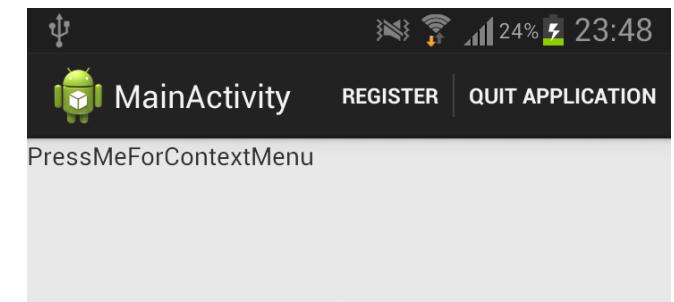
```
override fun onOptionsItemSelected(item: MenuItem?): Boolean {  
  
    if(item?.itemId == 1001 || item?.itemId == 1002)  
    {  
        var itemid = item?.itemId  
  
        Toast.makeText(this,"Options Item $itemid selected ",Toast.LENGTH_SHORT).show()  
    }  
  
    return super.onOptionsItemSelected(item)  
}
```

MENU

XML inflation

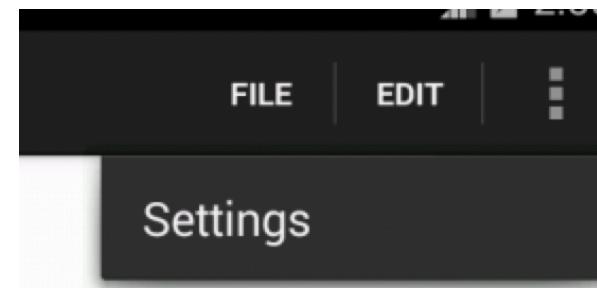
Menu Creation - XML

```
<menu  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto">  
  
    <item  
        android:id="@+id/menu_login"  
        app:showAsAction="withText"  
        android:title="Login"/>  
    <item  
        android:id="@+id/menu_register"  
        app:showAsAction="always"  
        android:title= " Register "  
    <item  
        android:id="@+id/menu_quitapp"  
        app:showAsAction="ifRoom"  
        android:title="Quit Application"/>  
</menu>
```



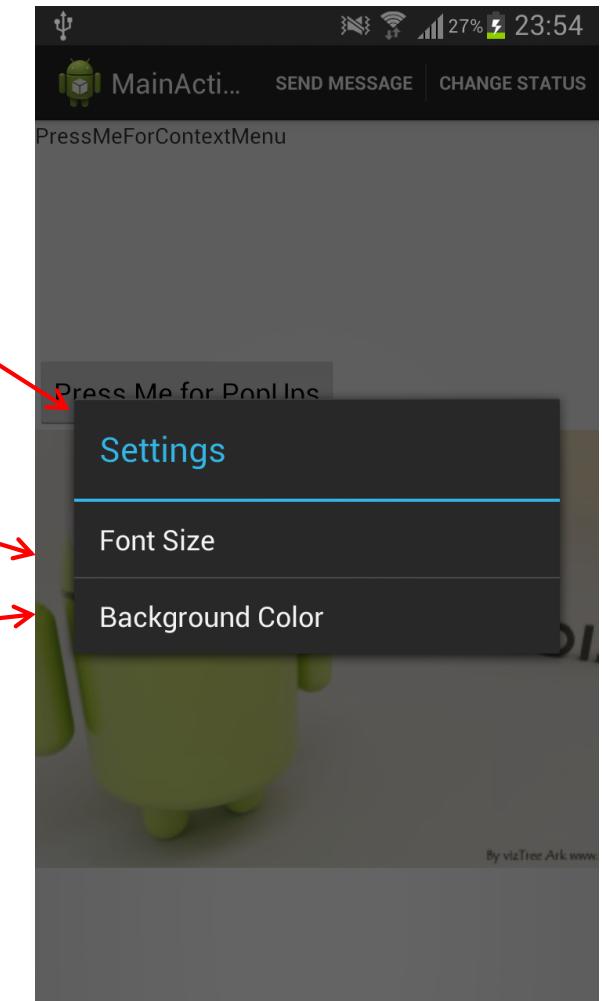
Menu - ActionBar

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app=http://schemas.android.com/apk/res-auto
    tools:context="com.nyp.lec_menu.MainActivity" >
    <item
        android:id="@+id/action_settings"
        android:orderInCategory="100"
        app:showAsAction="never"
        android:title="@string/action_settings"/>
    <item
        android:id="@+id/file
        app:showAsAction="ifRoom"
        android:title="File">
    </item>
    <item
        android:id="@+id/file
        app:showAsAction="ifRoom"
        android:title="Edit">
    </item>
</menu>
```



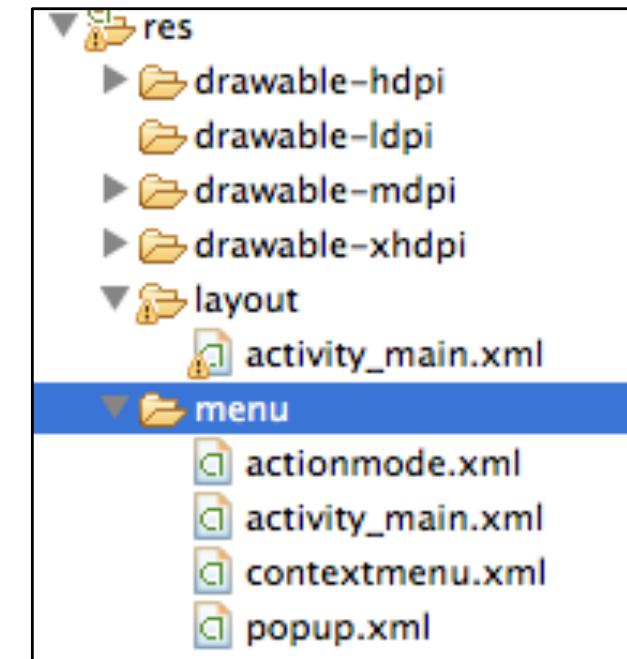
Menu Creation - XML

```
<menu  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:android="http://schemas.android.com/apk/res/android" >  
  
    <item  
        android:id="@+id/menu_settings" android:icon="@drawable/settings"  
        android:orderInCategory="101" app:showAsAction="ifRoom"  
        android:title="Settings">  
        <menu>  
            <item  
                android:id="@+id/menu_settings_font"  
                android:orderInCategory="110" app:showAsAction="ifRoom"  
                android:title="Font Size"/>  
            <item  
                android:id="@+id/menu_settings_bgcolor"  
                android:orderInCategory="110"  
                app:showAsAction="ifRoom"  
                android:title="Background Color"/>  
        </menu>  
    </item>  
</menu>
```



Menu Creation - XML

- Inflate XML file in
 - onCreateOptionsMenu Menu
 - onCreate ContextMenu
- Menu Resource
`menuInflator?.inflate(R.menu.main,menu)`



Changing options menu items at runtime

- Options Menu
 - `onCreateOptionsMenu` is only called ONCE during the Activity lifecycle.
 - System retains an instance of the Menu that was populated.
 - It should only be used to create the initial state.
 - To modify during the Activity lifecycle, make use of the `onPrepareOptionsMenu`.
 - On Android 2.3.x and lower, the system calls `onPrepareOptionsMenu` each time the user opens the options menu.
 - On Android 3.0 and above, the options menu is considered open all the time when menu items are presented in the actionbar. Call `invalidateOptionsMenu()` to request that the system call `onPrepareOptionsMenu`.

Menu

- Creation
 - Done by the operating system
 - Triggered by the user
 - Developers only need to indicate what sub-menus or menu items are to be created to populate the menus
- Populating
 - When ?
 - onCreateOptionMenu
 - onPrepareOptionMenu
 - onCreateContextMenu
 - How ?
 - XML
 - Kotlin

Summary

- Options and Context Menu
- Menu Items and Sub menu items
- XML and Java development of Menu items
- Action bar and Overflow menu