Session 6: Working with Menus

Assignment 1:

Problem Statement

a) What is menu. Write its importance.

b) What is options menu. When it is activated?

c) What is context menu. When it is activated?

d) What is the alternative of ActionBarActivity as its deprecated.

Solution:

1. Menus are a common user interface component in many types of applications. To provide a familiar and consistent user experience, we should use the Menu APIs to present user actions and other options in your activities.

For all menu types, Android provides a standard XML format to define menu items. Instead of building a menu in your activity's code, you should define a menu and all its items in an XML menu resource. One can then inflate the menu resource in the activity.

Using a menu resource is a good practice for a few reasons:

* It's easier to visualize the menu structure in XML.
* It separates the content for the menu from your application's behavioral code.
* It allows you to create alternative menu configurations for different platform versions, screen sizes, and other configurations by leveraging the app resources framework.

<menu>

Defines a Menu, which is a container for menu items. A <menu> element must be the root node for the file and can hold one or more <item> and <group> elements.

<item>

Creates a MenuItem, which represents a single item in a menu. This element may contain a nested <menu> element in order to create a submenu.

<group>

An optional, invisible container for <item> elements. It allows you to categorize menu items so they share properties such as active state and visibility. For more information, see the section about Creating Menu Groups.

Here's an example menu named game\_menu.xml:

<?xml version="1.0" encoding="utf-8"?>

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

<item android:id="@+id/new\_game"

android:icon="@drawable/ic\_new\_game"

android:title="@string/new\_game"

android:showAsAction="ifRoom"/>

<item android:id="@+id/help"

android:icon="@drawable/ic\_help"

android:title="@string/help" />

</menu>

1. **Option Menus** are the primary menus of android. They can be used for settings, search, delete item etc. The options menu is where one should include actions and other options that are relevant to the current activity context, such as "Search," "Compose email," and "Settings”etc. Where the items in your options menu appear on the screen depends on the version for which one had developed your application.

To specify the options menu for an activity, override onCreateOptionsMenu().In this method, you can inflate your menu resource into the menu provided in the callback. For example:

@Override

public boolean onCreateOptionsMenu(Menu menu) {

MenuInflater inflater = getMenuInflater();

inflater.inflate(R.menu.game\_menu, menu);

return true;

}

When the user selects an item from the options menu (including action items in the app bar), the system calls your activity's onOptionsItemSelected() method. This method passes the MenuItem selected. You can identify the item by calling getItemId(), which returns the unique ID for the menu item. For example:

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// Handle item selection

switch (item.getItemId()) {

case R.id.new\_game:

newGame();

return true;

case R.id.help:

showHelp();

return true;

default:

return super.onOptionsItemSelected(item);

}

}

1. A contextual menu offers actions that affect a specific item or context frame in the UI. You can provide a context menu for any view, but they are most often used for items in a [ListView](https://developer.android.com/reference/android/widget/ListView.html), [GridView](https://developer.android.com/reference/android/widget/GridView.html), or other view collections in which the user can perform direct actions on each item.

There are two ways to provide contextual actions:

1. In a [floating context menu](https://developer.android.com/guide/topics/ui/menus.html#FloatingContextMenu). A menu appears as a floating list of menu items (similar to a dialog) when the user performs a long-click (press and hold) on a view that declares support for a context menu. Users can perform a contextual action on one item at a time.
2. In the [contextual action mode](https://developer.android.com/guide/topics/ui/menus.html#CAB). This mode is a system implementation of [ActionMode](https://developer.android.com/reference/android/view/ActionMode.html) that displays a *contextual action bar* at the top of the screen with action items that affect the selected item(s). When this mode is active, users can perform an action on multiple items at once (if your app allows it).

@Override

public void onCreateContextMenu(ContextMenu menu, View v,

ContextMenuInfo menuInfo) {

super.onCreateContextMenu(menu, v, menuInfo);

MenuInflater inflater = getMenuInflater();

inflater.inflate(R.menu.context\_menu, menu);

}

Implement onContextItemSelected().When the user selects a menu item, the system calls this method so you can perform the appropriate action. For example:

@Override

public boolean onContextItemSelected(MenuItem item) {

AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();

switch (item.getItemId()) {

case R.id.edit:

editNote(info.id);

return true;

case R.id.delete:

deleteNote(info.id);

return true;

default:

return super.onContextItemSelected(item);

}

}

The getItemId() method queries the ID for the selected menu item.

1. ActionBarActivity has been deprecated in favor of the new AppCompatActivity and that AppCompatActivity is base class for activities that use the support library action bar features. AppCompatActivity class is a safe to use backward compatibility class.

AppCompatActivity is a new, more generic implementation which uses AppCompatDelegate class internally. AppCompatActivity is a single consistent Action Bar for all API 7 and higher devices. In revision 21, it took on new responsibility: bringing material color palette, widget tinting, Toolbar support, and more to all API 7+ devices.

AppCompatActivity is the direct child class of FragmentActivity of support v4 and the direct parent class of ActionBarActivity. Here are a few of the key classes included in the v7 appcompat library:

* ActionBar - Provides an implementation of the action bar user interface pattern.
* AppCompatActivity - Adds an application activity class that can be used as a base class for activities that use the Support Library action bar implementation.
* AppCompatDialog - Adds a dialog class that can be used as a base class for AppCompat themed dialogs.
* ShareActionProvider - Adds support for a standardized sharing action (such as email or posting to social applications) that can be included in an action bar.