

8/7/2024

Slide 1 of 33

Session 5

Menus and Contextual Action Mode

Trình bày : **Đỗ Phú Huy**
BM Tin học-CĐCN



Modules



Introduction to a Menus



Introduction to a Contextual Action Mode



8/7/2024




Slide 3 of 33

Module 9

Introduction to a Menus



Topics

-  **Introduction to a Option Menu**
-  **Introduction to a Context Menu**
-  **Introduction to a Popup Menu**



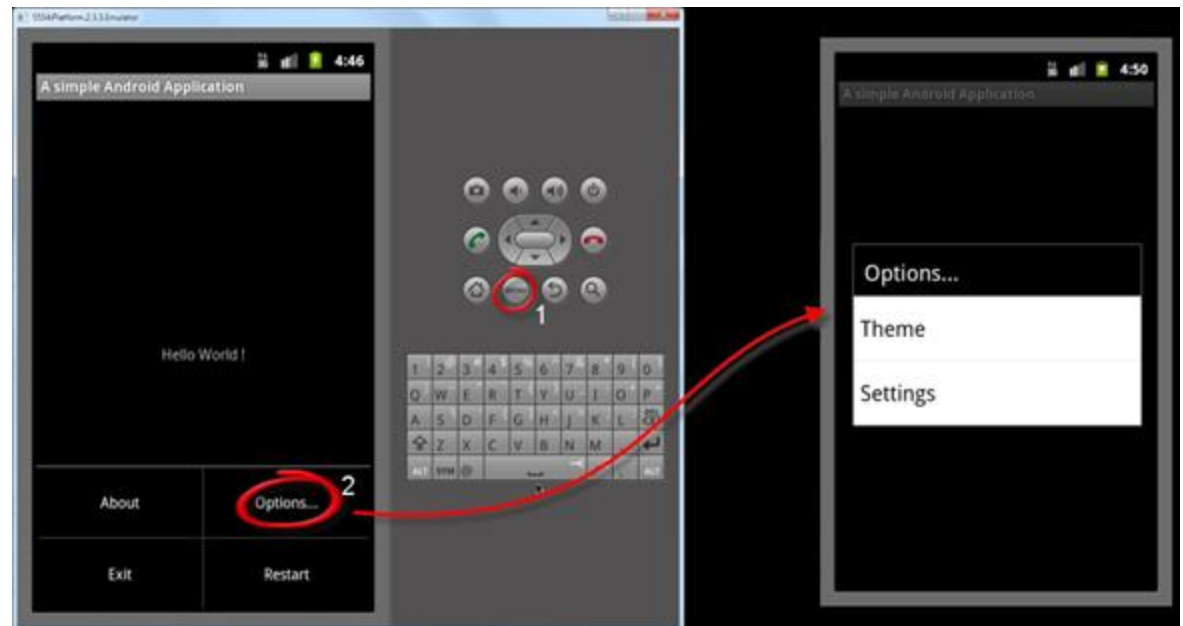
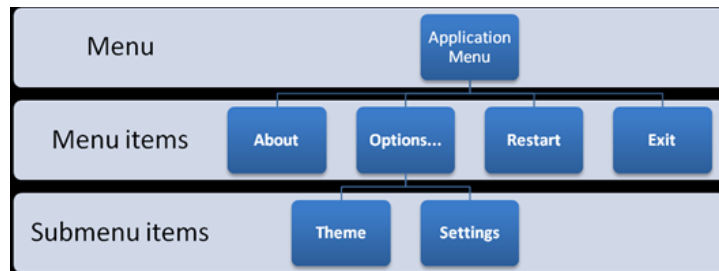
What's Menu ?

Menu là một trong những thành phần quan trọng trong các ứng dụng Android, cung cấp cho người dùng cách để thực hiện các tác vụ trên ứng dụng dễ dàng nhất.

Menu là cách nhanh nhất để cho người dùng có thể thực hiện chức năng của ứng dụng mà không bị mất khoảng không gian trên màn hình. Mỗi Activity có thể chỉ định riêng menu dành cho Activity đó, mà sẽ hiển thị khi nút Menu trên thiết bị di động được nhấn.



Structure of Menu ?



Option Menu

Trong nhiều chương trình cần đến menu để thực hiện các chức năng gì đó mà không cần phải thả các button đâu đó trên màn hình người dùng sử dụng option menu.

Vd: trong các ứng dụng cần đến nhiều diện tích màn hình, như là các ứng dụng sử dụng camera.

Menu có ích cho các ứng dụng chạy trên các thiết bị có kích thước màn hình bé.

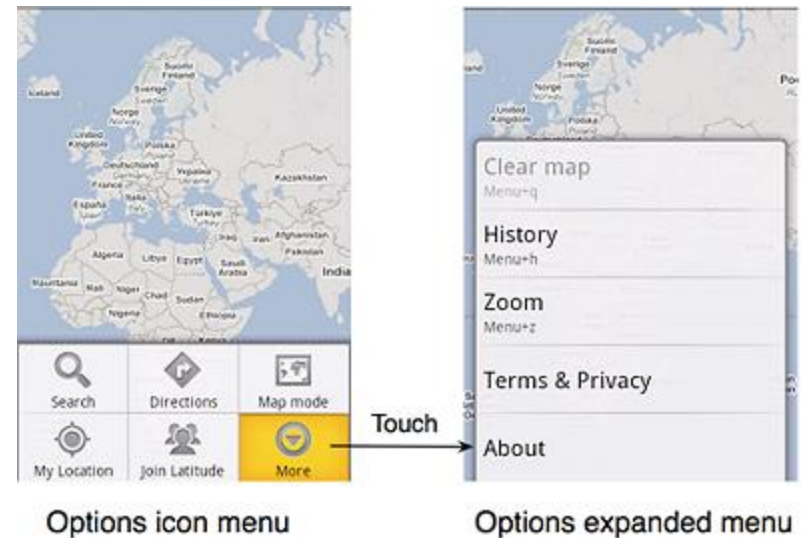


<http://developer.android.com/reference/android/view/Menu.html>



Expanded Menu :

Menu này sẽ được kích hoạt khi người dùng nhấn nút More trên Icon Menu. Nó sẽ hiển thị danh sách các mục Menu nhưng không có icon. Expanded Menu hiển thị văn bản đầy đủ, phím tắt hay các checkbox và radiobutton.



How to create Option Menu ?

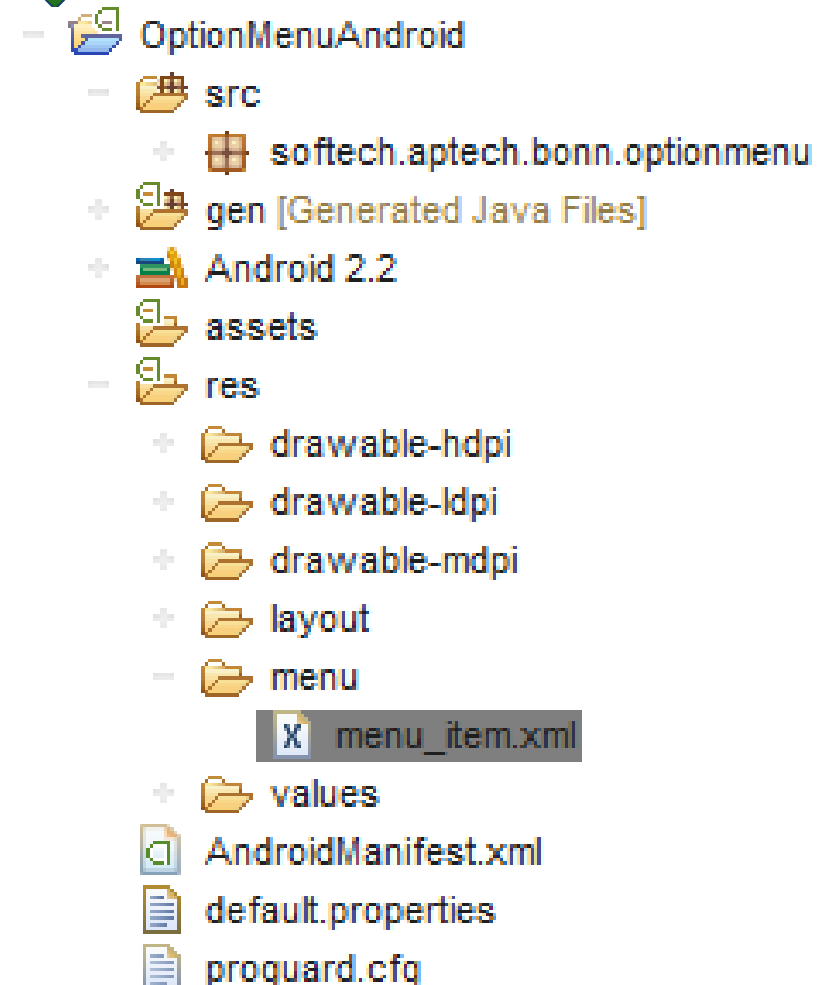
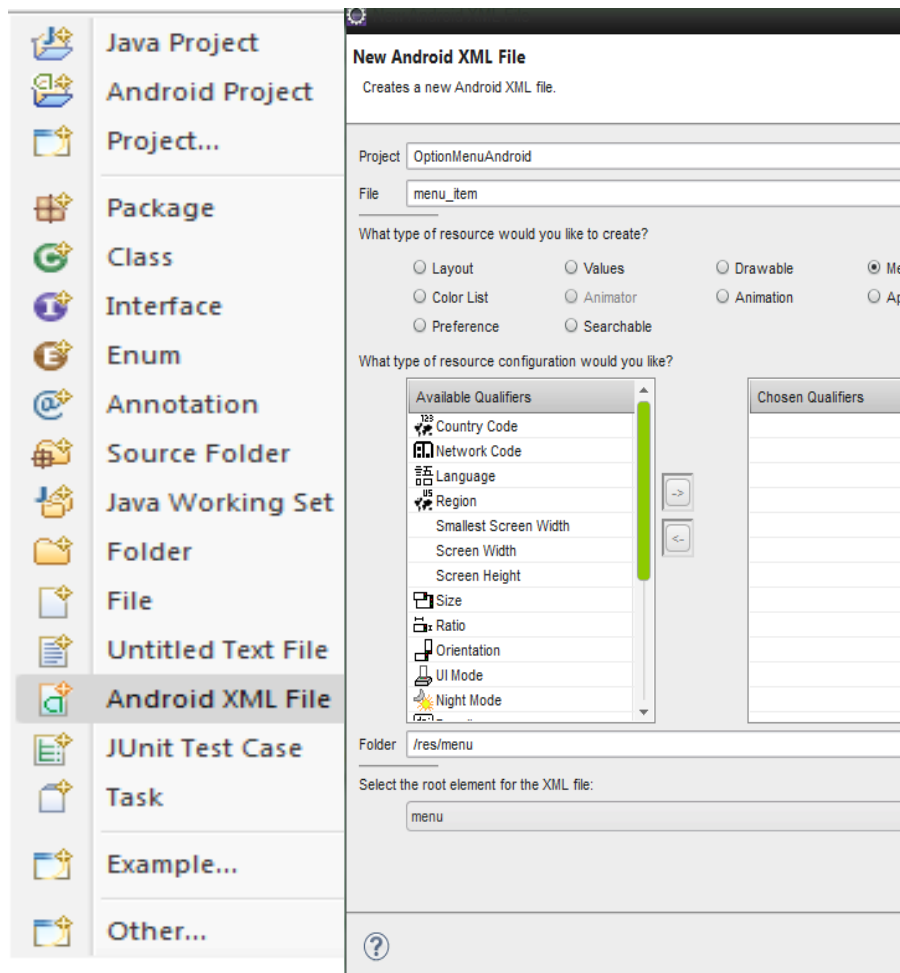


Use XML Code

Use Java code



Use XML Code



Use XML Code (con't)



```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/about" android:title="About" android:icon="@drawable/icon" />
    <item android:id="@+id/help" android:title="Help" android:icon="@drawable/icon" />
</menu>
```

Add Items to Your Menu and Create Icons for Your Menu Items

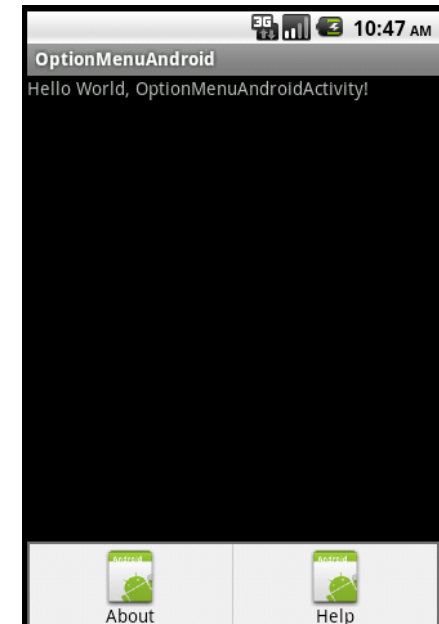


Use XML Code (con't)

```
import android.app.Activity;

public class OptionMenuAndroidActivity 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) {
        // TODO Auto-generated method stub
        // return super.onCreateOptionsMenu(menu);
        MenuInflater inflaterMenu = getMenuInflater();
        inflaterMenu.inflate(R.menu.menu_item, menu);
        return true;
    }
}
```



ANDROID

Inflate Your Menu Resource



Use Java Code



```
import android.app.Activity;

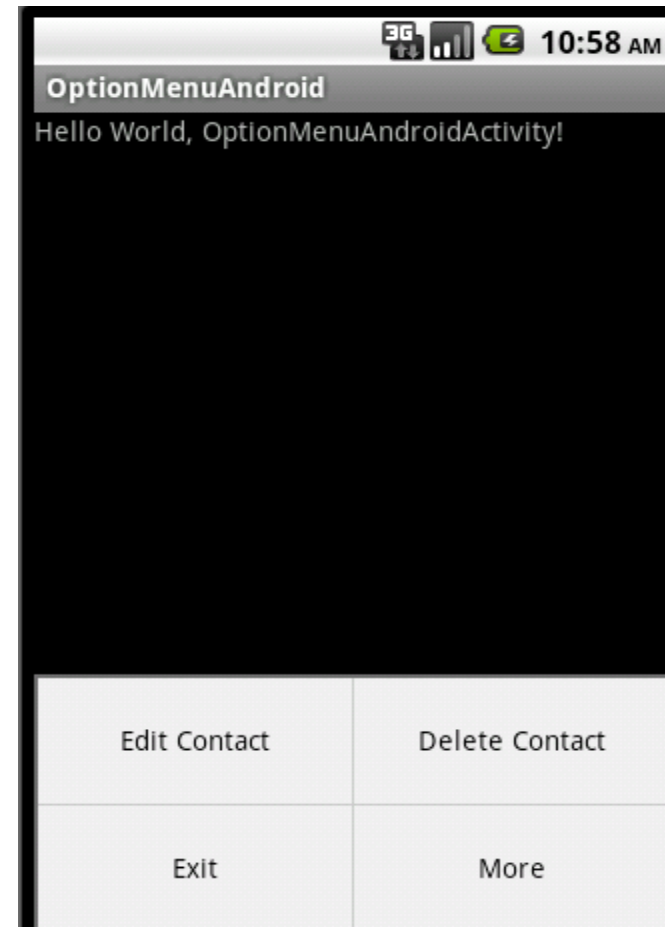
public class OptionMenuAndroidActivity extends Activity {
    /** Called when the activity is first created. */

    //ID for the menu item
    private final int EDIT_CONTACT = 1;
    private final int DELETE_CONTACT = 2;
    private final int EXIT = 3;
    private final int MORE = 4;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // TODO Auto-generated method stub
        // return super.onCreateOptionsMenu(menu);

        // MenuInflater inflaterMenu = getMenuInflater();
        // inflaterMenu.inflate(R.menu.menu_item, menu);
        menu.add(0, EDIT_CONTACT, 0, "Edit Contact");
        menu.add(0, DELETE_CONTACT, 0, "Delete Contact");
        menu.add(0, EXIT, 0, "Exit");
        menu.add(0, MORE, 0, "More");
        return true;
    }
}
```



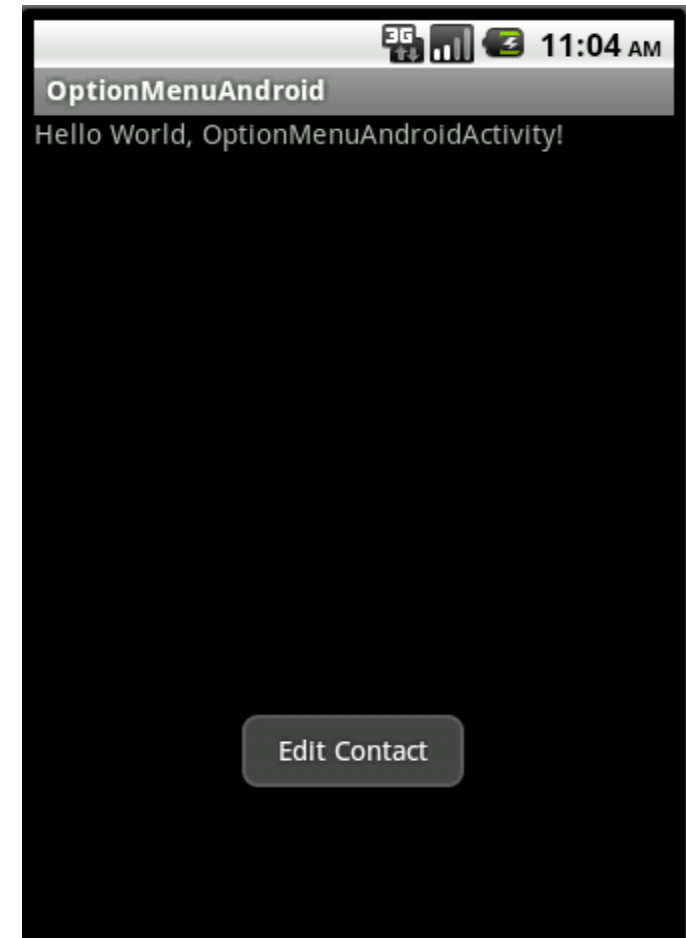
Handling click events



```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // TODO Auto-generated method stub
    // return super.onOptionsItemSelected(item);
    // Handle item selection
    switch (item.getItemId()) {
        case EDIT_CONTACT:
            Toast.makeText(this, item.getTitle(), Toast.LENGTH_SHORT).show();
            return true;
        case DELETE_CONTACT:
            Toast.makeText(this, item.getTitle(), Toast.LENGTH_SHORT).show();
            return true;
        case EXIT:
            Toast.makeText(this, item.getTitle(), Toast.LENGTH_SHORT).show();
            return true;
        case MORE:
            Toast.makeText(this, item.getTitle(), Toast.LENGTH_SHORT).show();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```



ANDROID



Context Menu

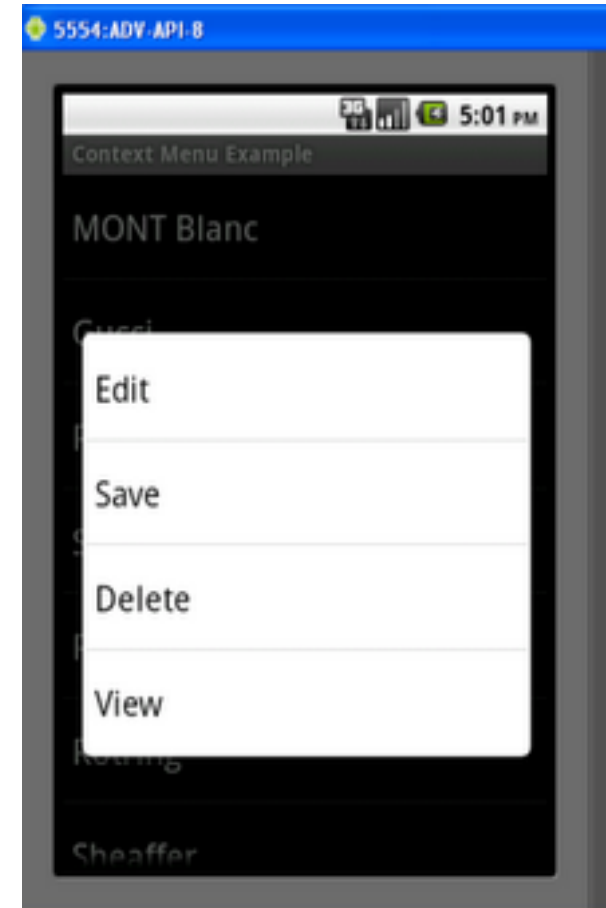


Android hỗ trợ các menu ngữ cảnh (context menu) có thể gán cho bất cứ View nào bên trong một Activity.

Menu ngữ cảnh của View thường được kích hoạt khi người dùng nhấn giữ View trên màn hình khoảng 3 giây.

Activity và các menu ngữ cảnh cũng hỗ trợ các menu con, checkbox, radiobutton, phím tắt và biểu tượng.

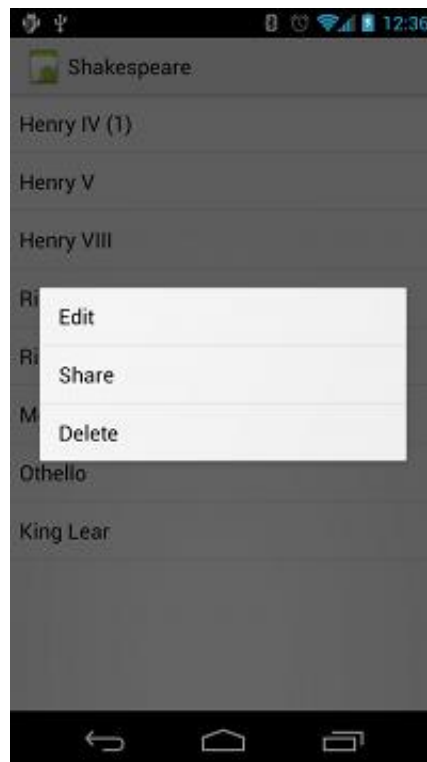
<http://developer.android.com/reference/android/view/ContextMenu.html>



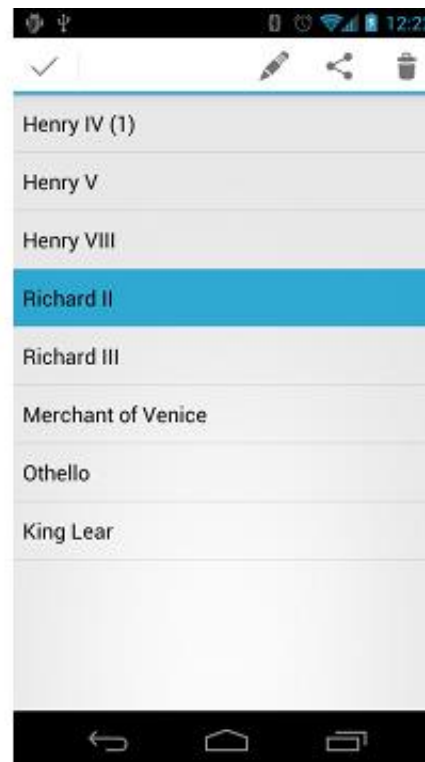
How to provide contextual actions



Floating context menu



Contextual action mode.



ANDROID



Floating Context Menu



```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/about" android:title="About" android:icon="@drawable/icon" />
    <item android:id="@+id/help" android:title="Help" android:icon="@drawable/icon" />
</menu>
```

Add Items to Your Menu and Create Icons for Your Menu Items



Floating Context Menu (con't)



```
import android.app.Activity;

public class FloatingContextMenuActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button btnShowContextMenu = (Button) findViewById(R.id.btnShowContextMenu);
        registerForContextMenu(btnShowContextMenu);
    }

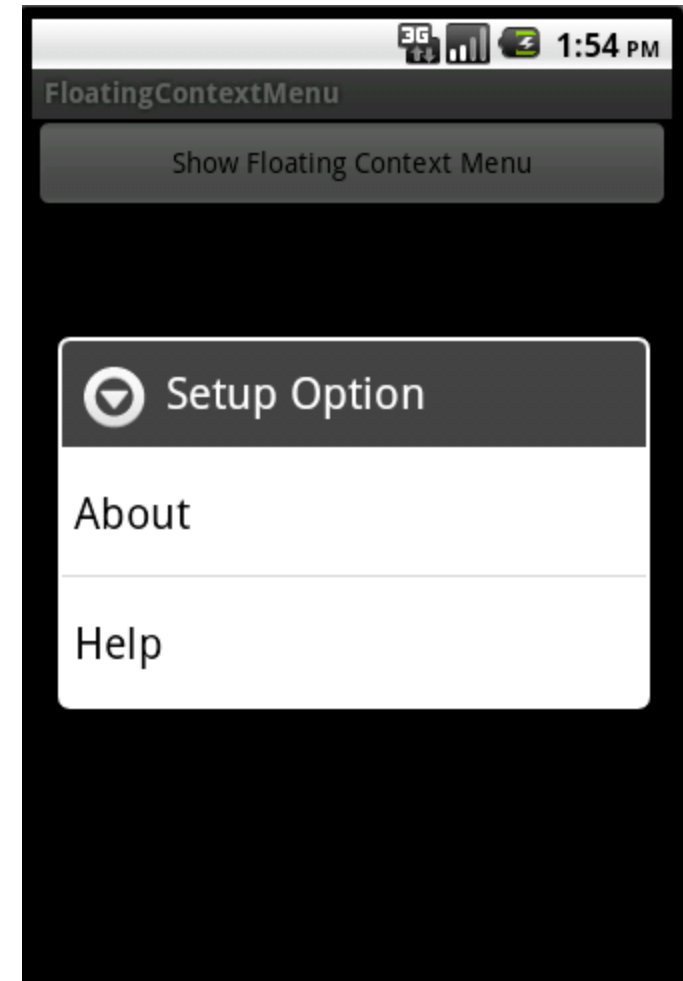
    @Override
    public void onCreateContextMenu(ContextMenu menu, View v,
        ContextMenuInfo menuInfo) {
        super.onCreateContextMenu(menu, v, menuInfo);
        menu.setHeaderTitle("Setup Option");
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu_item, menu);
    }
}
```



Floating Context Menu (con't)



```
@Override
public boolean onContextItemSelected(MenuItem item) {
    // AdapterContextMenuInfo info = (AdapterContextMenuInfo) item
    // .getMenuInfo();
    switch (item.getItemId()) {
        case R.id.about:
            Toast.makeText(this, item.getTitle(), Toast.LENGTH_SHORT).show();
            return true;
        case R.id.help:
            Toast.makeText(this, item.getTitle(), Toast.LENGTH_SHORT).show();
            return true;
        default:
            return super.onContextItemSelected(item);
    }
}
```



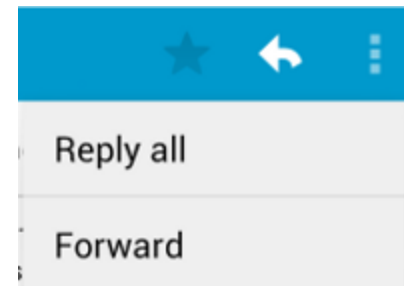
Popup Menu



PopupMenu hiển thị một Menu trong hộp thoại mô hình popup được cố định trên một View.

Chạm ra ngoài popup để thoát popup này.

<http://developer.android.com/reference/android/widget/PopupMenu.html>



Popup Menu



- Là menu kiểu overflow dành cho các hành động liên quan đến một nội dung cụ thể (ví dụ các tiêu đề email trong Gmail).
- Cung cấp một widget kiểu thả xuống (drop-down) tương tự [Spinner](#) và cho phép không giữ lại một chọn lựa cố định nào.



Step by Step create a Popup Menu



If you define your menu in XML, here's how you can show the popup menu:

1. Instantiate a [PopupMenu](#) with its constructor, which takes the current application Context and the View to which the menu should be anchored.
2. Use [MenuInflater](#) to inflate your menu resource into the Menu object returned by [PopupMenu.getMenu\(\)](#). On API level 14 and above, you can use [PopupMenu.inflate\(\)](#) instead.
3. Call [PopupMenu.show\(\)](#).



Step by Step create a Popup Menu



```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/menu_archive" android:title="Archive" />
    <item android:id="@+id/menu_delete" android:title="Delete" />
</menu>
```



Step by Step create a Popup Menu

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    Button btnShowPopup = (Button) findViewById(R.id.btnShowPopup);
    btnShowPopup.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            showMenu(v);
        }
    });
}

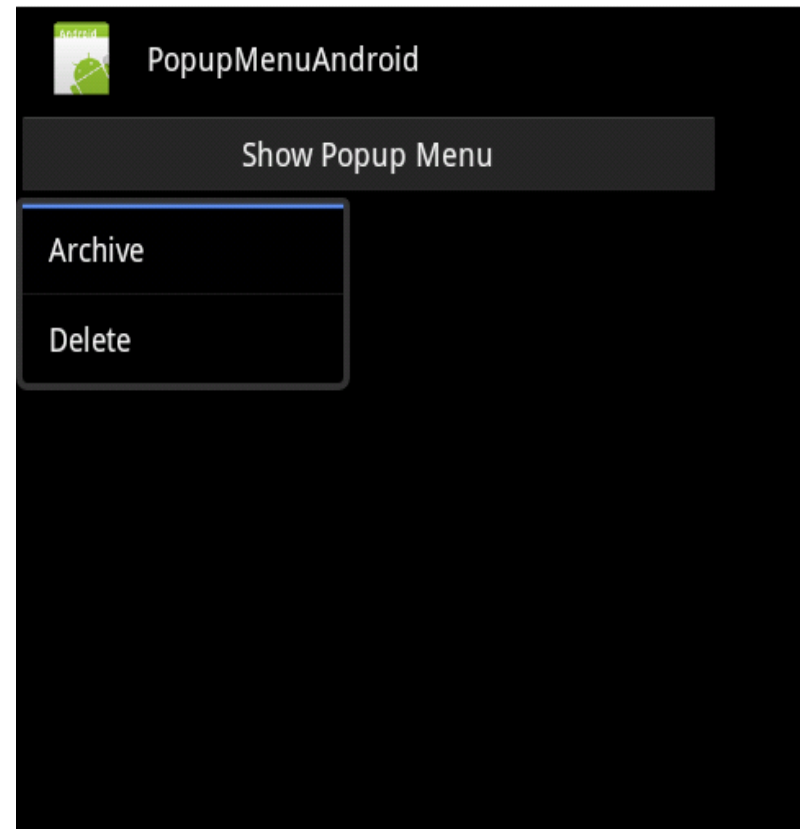
public void showMenu(View v) {
    PopupMenu popup = new PopupMenu(this, v);
    MenuInflater inflater = popup.getMenuInflater();
    inflater.inflate(R.menu.actions, popup.getMenu());
    popup.show();
}
```



Step by Step create a Popup Menu



```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.menu_archive:  
            Toast.makeText(this, item.getTitle(), Toast.LENGTH_SHORT).show();  
            return true;  
        case R.id.menu_delete:  
            Toast.makeText(this, item.getTitle(), Toast.LENGTH_SHORT).show();  
            return true;  
        default:  
            return false;  
    }  
}
```



8/7/2024

Slide 26 of 33

Module 10

Introduction to a Contextual Action Mode



Topics



What's Contextual Action Mode ?



Step by step using Contextual Action Mode.



What's Contextual Action Mode?



- ActionBar nằm ở phía trên của Activity, là nơi hiển thị tiêu đề Activity, menu chuyển hướng hoặc các đối tượng có thể tương tác khác ...
- Một Contextual Action Mode là một ActionBar tạm thời được kích hoạt theo ngữ cảnh. Nó sẽ tạm thời che đi ActionBar của ứng dụng trong suốt thời gian nó xuất hiện (một task con nào đó).
- Một Contextual Action Mode thông thường được kích hoạt bằng cách nhấp vào một item hoặc long-click lên nó.



Step By Setup



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <EditText
        android:id="@+id/myView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" >
        <requestFocus />
    </EditText>

</LinearLayout>
```

main.xml



Step By Setup



```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/toast" android:title="Toast">
  </item>
</menu>
```

contextual.xml



Step By Setup

```
protected Object mActionMode;
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    // Define the contextual action mode
    View view = findViewById(R.id.myView);
    view.setOnLongClickListener(new View.OnLongClickListener() {
        // Called when the user long-clicks on someView
        public boolean onLongClick(View view) {
            if (mActionMode != null) {
                return false;
            }

            // Start the CAB using the ActionMode.Callback defined above
            mActionMode = ActionBarUseActionModeActivity.this
                .startActionMode(mActionModeCallback);
            view.setSelected(true);
            return true;
        }
    });
}
```



android



Step By Setup

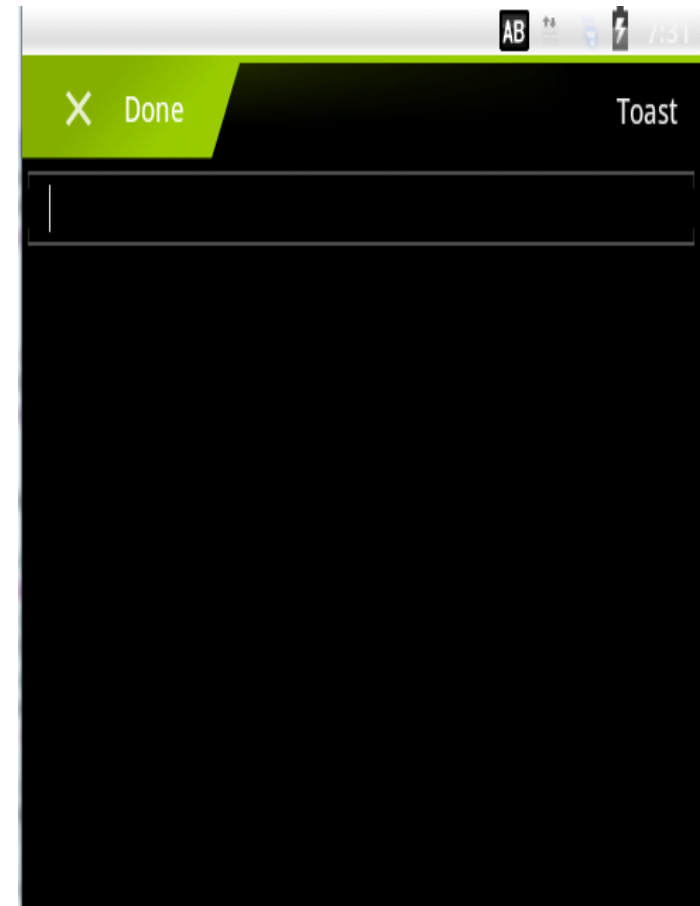
```
private ActionMode.Callback mActionModeCallback = new ActionMode.Callback() {

    // Called when the action mode is created; startActionMode() was called
    public boolean onCreateActionMode(ActionMode mode, Menu menu) {
        // Inflate a menu resource providing context menu items
        MenuInflater inflater = mode.getMenuInflater();
        // Assumes that you have "contextual.xml" menu resources
        inflater.inflate(R.menu.contextual, menu);
        return true;
    }

    // Called each time the action mode is shown. Always called after
    // onCreateActionMode, but
    // may be called multiple times if the mode is invalidated.
    public boolean onPrepareActionMode(ActionMode mode, Menu menu) {
        return false; // Return false if nothing is done
    }

    // Called when the user selects a contextual menu item
    public boolean onActionItemClicked(ActionMode mode, MenuItem item) {
        switch (item.getItemId()) {
            case R.id.toast:
                Toast.makeText(ActionBarUseActionModeActivity.this, "Selected menu",
                    Toast.LENGTH_LONG).show();
                mode.finish(); // Action picked, so close the CAB
                return true;
            default:
                return false;
        }
    }

    // Called when the user exits the action mode
    public void onDestroyActionMode(ActionMode mode) {
        mActionMode = null;
    }
};
```



8/7/2024

Slide 33 of 33



Q&A



8/7/2024

Slide 34 of 33



Thanks you every one 😊

