



THIẾT KẾ GIAO DIỆN ANDROID

BÀI 4: BAR VÀ MENU

- ⊙ Kết thúc bài học này bạn có khả năng
 - ⊙ Hiểu được cách sử dụng ActionBar, ToolBar
 - ⊙ Phân biệt được các loại menu
 - ⊙ Biết cách sử dụng từng loại menu



Phần I: ActionBar vs ActionBar

 A ActionBar

 B ToolBar

 C Tìm hiểu thêm

Phần II: Các loại Menu

 D ContextMenu

 E OptionMenu

 F PopupMenu

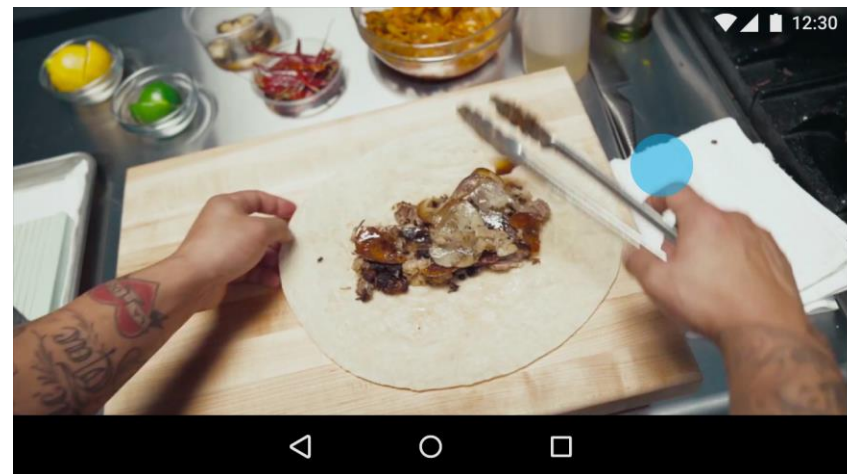




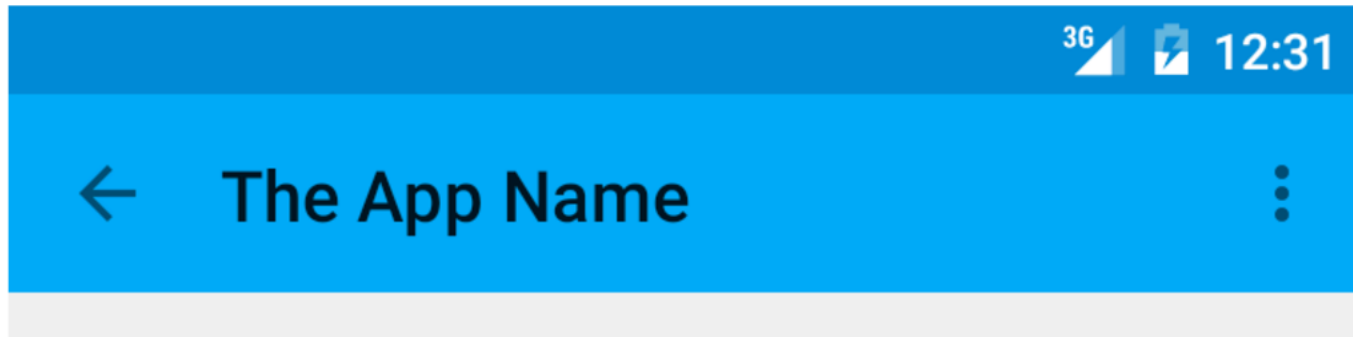
BÀI 4: BAR VÀ MENU

PHẦN 1: BAR

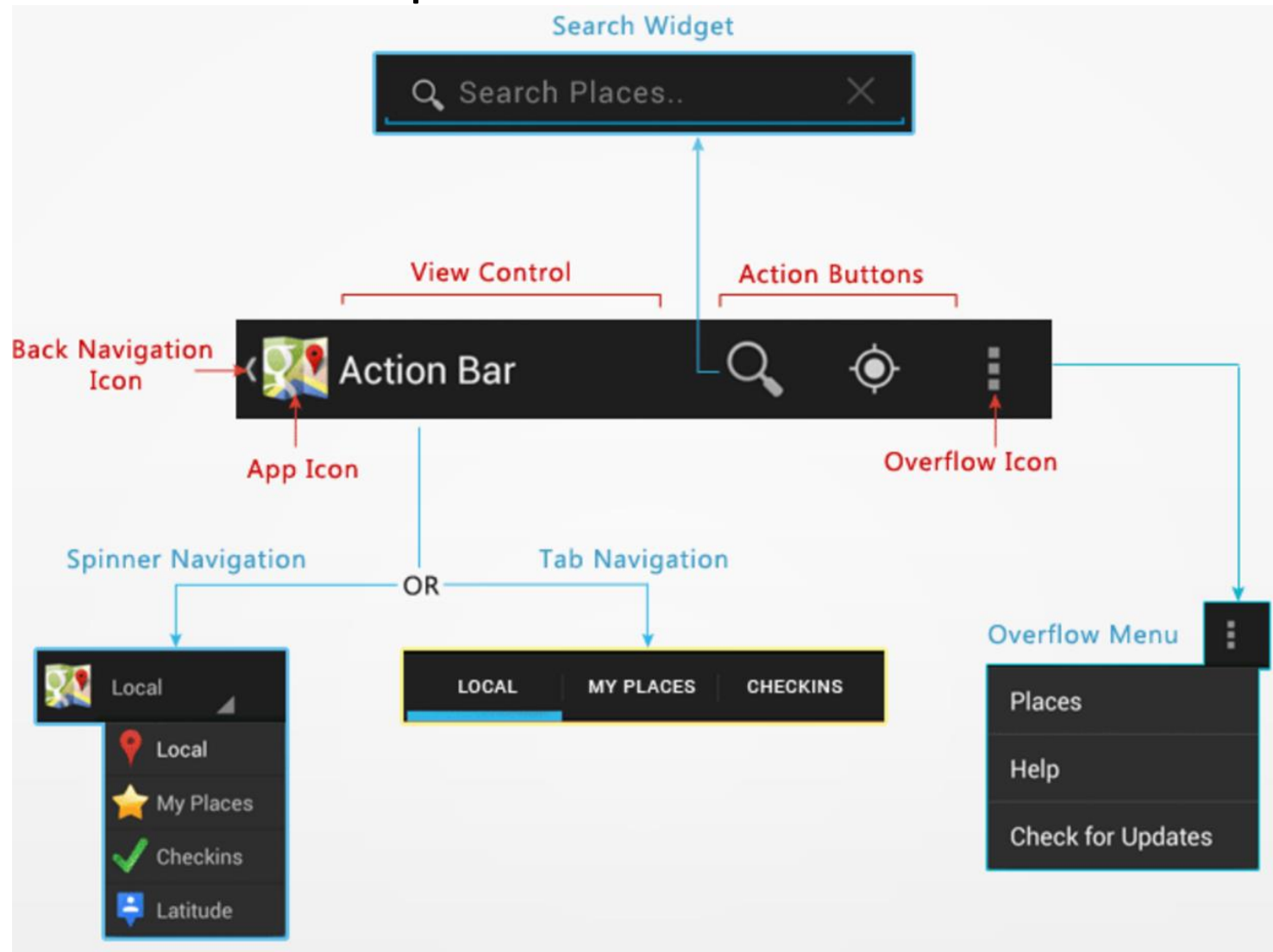
- ❑ StatusBar (Thanh trạng thái) chứa các notification icon và System icon
- ❑ Thông thường ta không tác động đến thanh này, chỉ tác động như đổi màu, ẩn hoặc trong suốt.



- ❑ ActionBar (Thanh tác vụ) là một thanh nằm trên đầu của ứng dụng (bên dưới thanh status bar), nó thường cố định và xuyên suốt trong toàn bộ ứng dụng.



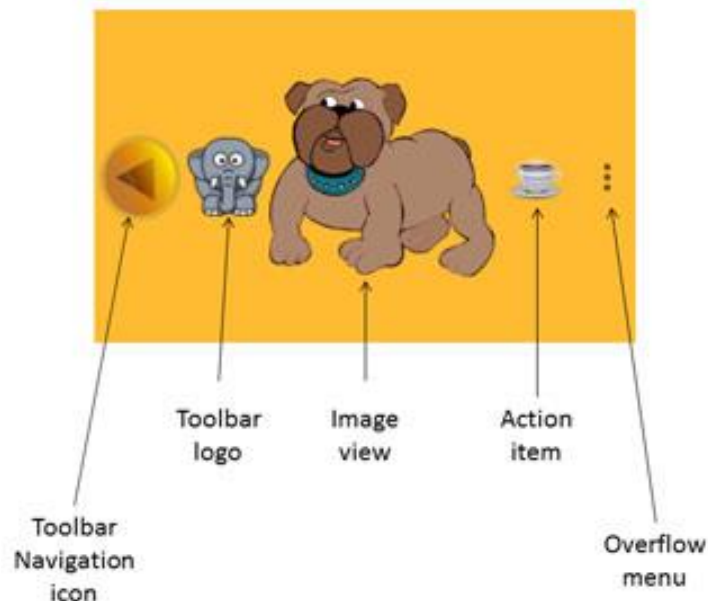
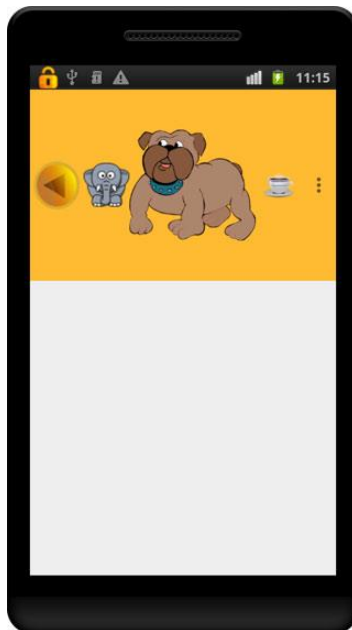
ActionBar có 4 phần



- ❑ App Icon & Back Navigation Icon: hiển thị logo của chương trình và có thể có 1 back Navigation Icon để quay về activity trước đó.
- ❑ ViewControl: Nếu ứng dụng hiển thị trong chế độ xem khác nhau, phần này sẽ cho phép chuyển đổi giữa các chế độ xem. Có thể dùng Spinner Navigation hoặc Tab Navigation để chuyển.
- ❑ Action Button: Icon để thực hiện các hành động quan trọng. Các hành động kém quan trọng hơn sẽ đưa vào Overflow Menu
- ❑ Overflow Menu: Menu chứa các hành động ít quan trọng

❑ Android 5.0 SDK phát hành có thêm AppCompatActivity (ActionBarCompat). Toolbar được giới thiệu với nhiều tính năng nổi trội hơn so với Action Bar.

- ❖ Nổi trội hơn về giao diện
- ❖ Có thể đặt bất kì đâu trên giao diện (vd: top hoặc bottom).
- ❖ Có thể đặt bất kì widget vào trong nó
- ❖ Có thể có nhiều toolbar trên 1 Activity.



- ☐ Tạo project, kiểm tra build.gradle đã có thư viện appcompat-v7 chưa.
- ☐ Kiểm tra file java đã kế thừa từ AppCompatActivity chưa
- ☐ Sửa theme trong AndroidManifest.xml (hoặc trong file styles.xml) từ DarkActionBar thành NoActionBar (sau thao tác này Activity sẽ mất ActionBar)

```
<application  
    android:theme="@style/Theme.AppCompat.Light.NoActionBar"  
/>
```

❑ Thêm ToolBar vào trong layout của Activity

```
<android.support.v7.widget.Toolbar  
    android:id="@+id/my_toolbar"  
    android:layout_width="match_parent"  
    android:layout_height="?attr/actionBarSize"  
    android:background="?attr/colorPrimary"  
    android:elevation="4dp"  
    android:theme="@style/ThemeOverlay.AppCompat.ActionBar"  
    app:popupTheme="@style/ThemeOverlay.AppCompat.Light"/>
```

- ❑ Chú ý: ToolBar như một ViewGroup bình thường có thể bỏ bất cứ View gì vào trong nó (vd: bỏ ImageView, Button, Spinner.... Vào ToolBar)

- ❑ Trong onCreate ánh xạ Toolbar và dùng lệnh `setSupportActionBar` để gán Toolbar thay cho ActionBar

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_my);
    Toolbar myToolbar = (Toolbar) findViewById(R.id.my_toolbar);
    setSupportActionBar(myToolbar);
}
```

- ❑ Gán thêm Logo và Navigation Icon (nút back để quay về Activity trước) cho Toolbar bằng

```
toolbar.setNavigationIcon(R.drawable.ic_action_back);
toolbar.setLogo(R.mipmap.ic_launcher);
```

- ❑ Nếu là đây là Activity con, muốn nhấn Navigation Icon để quay về Activity cha. Trong Manifest, trong Activity con khai báo cho nó biết Activity cha của nó

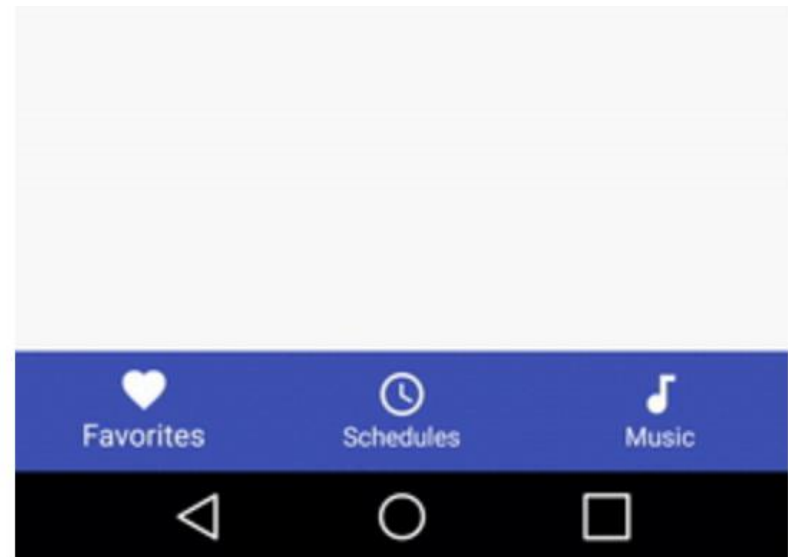
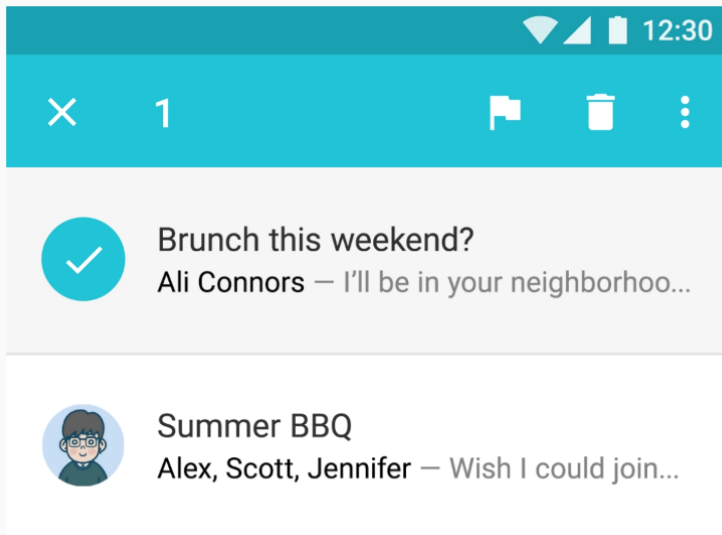
```
<activity android:name=".MainActivity"  
    android:parentActivityName=".ChaoActivity"  
>
```

- ❑ Trong onCreate của Activity con, thêm lệnh để Navigation Icon hoạt động

```
getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
getSupportActionBar().setDisplayShowHomeEnabled(true);
```

❑ Sinh viên nên tìm hiểu thêm về

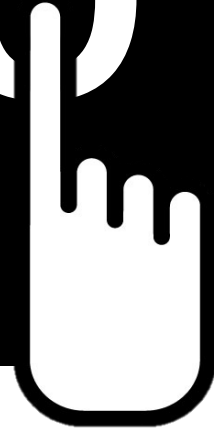
- ❖ Contextual Action Bar
- ❖ Bottom Navigation View





DEMO

ToolBar



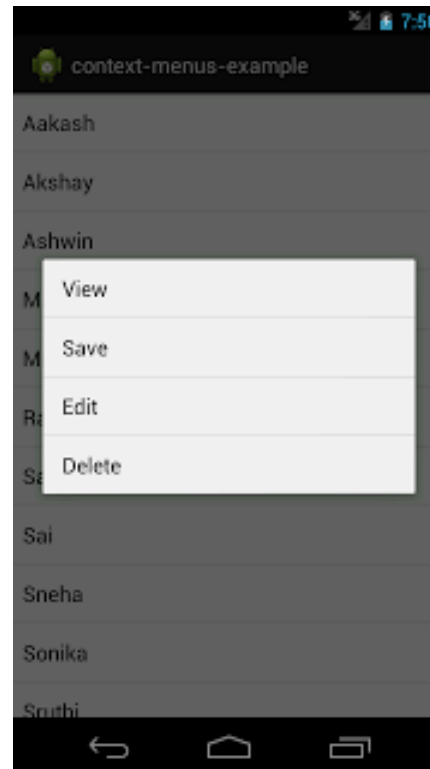
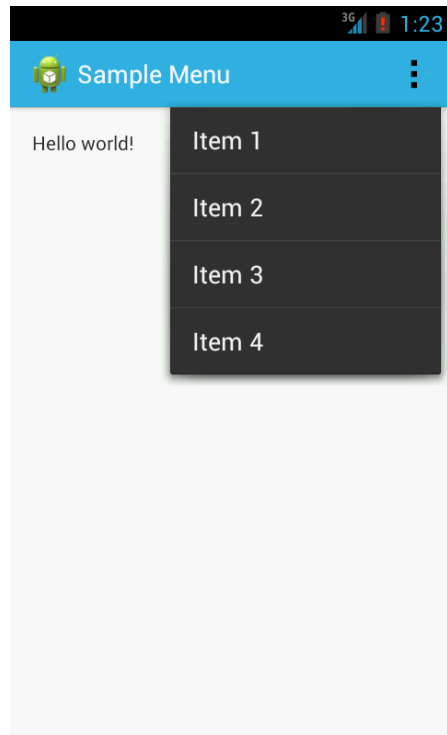


BÀI 4: BAR & MENU

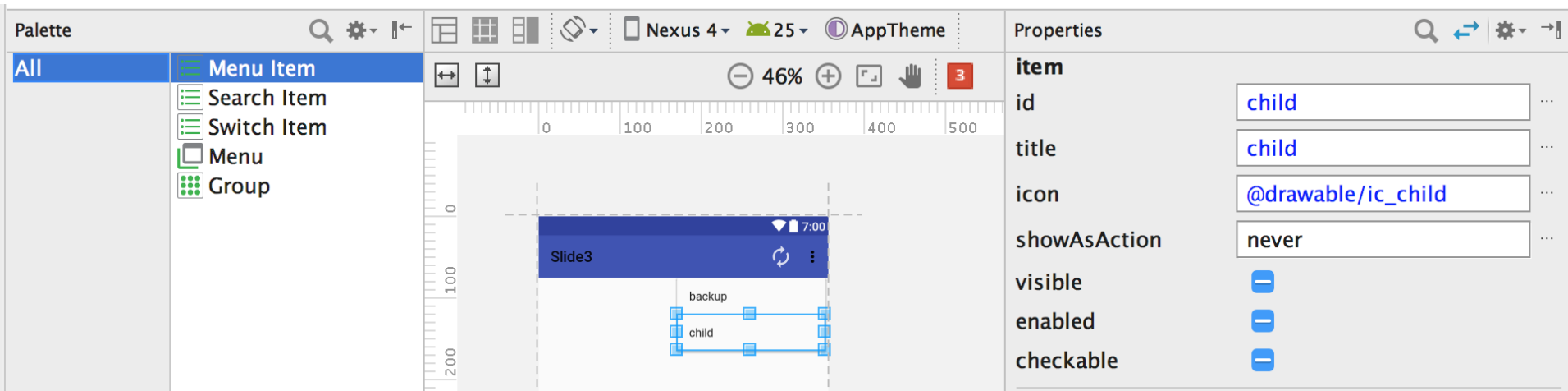
PHẦN 2: MENU

❑ Menu là một danh sách xổ xuống các item cho ta chọn lựa. Menu gồm có:

- ❖ OptionMenu
- ❖ ContextMenu
- ❖ PopupMenu



- ❑ OptionMenu Android 3.0 (API 11) không còn dùng menu nằm bên dưới với nút cứng nữa mà đưa optionMenu lên gốc trên bên phải của ActionBar /ToolBar
- ❑ Tạo OptionMenu đầu tiên tạo file `res/menu/optionmenu.xml`. Dùng thao tác



□ Hoặc bằng code:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:id="@+id/refresh"
    android:icon="@drawable/ic_refresh"
    android:title="Build"
    app:showAsAction="ifRoom" />
  <item
    android:id="@+id/backup"
    android:icon="@drawable/ic_backup"
    android:title="backup"
    app:showAsAction="ifRoom" />
  <item
    android:id="@+id/child"
    android:icon="@drawable/ic_child"
    android:title="child"
    app:showAsAction="never" />
</menu>
```

Chú ý các đối số:

- ☐ Id: id dùng để xử lý trong java
- ☐ Icon: icon của menu item
- ☐ Title: tiêu đề hiển thị lên menu item
- ☐ showAsAction:
 - ❖ ifRoom hiển thị lên ActionBar của Bar nếu đủ chỗ, không đủ sẽ bị đưa vào Overflow Icon.
 - ❖ Always luôn luôn hiện lên ActionBar.
 - ❖ Never luôn hiện trong Overflow Icon
- ☐ Có thể cho menu lồng vào item để tạo thành menu cấp 2, thậm chí cấp 3...

❑ Gắn menu vào ActionBar/ToolBar

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.option_menu, menu);
    return super.onCreateOptionsMenu(menu);
}
```

❑ Bắt sự kiện chọn vào một item của menu

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    if(item.getItemId()==R.id.backup)
        Toast.makeText(this, "backup", Toast.LENGTH_SHORT).show();
    if(item.getItemId()==R.id.refresh)
        Toast.makeText(this, "refresh", Toast.LENGTH_SHORT).show();
    return super.onOptionsItemSelected(item);
}
```

❑ Menu ngữ cảnh là menu được gắn vào AdapterView. Khi ta nhấn giữ lâu trên 1 AdapterView vd như ListView, GridView thì ContextMenu sẽ hiện ra

❑ Tạo file menu tên context_menu.xml

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

❑ Tạo và ánh xạ listview, tạo adapter gán cho

```
ListView lv;  
String mang[]=new String[]{"cam", "xoai", "coc", "oi", "miaghim"};  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    lv=(ListView)findViewById(R.id.listview1);  
    ArrayAdapter<String> adapter=new ArrayAdapter<String>(MainAc  
    lv.setAdapter(adapter);  
}
```

❑ Override onCreateContextMenu để gán file menu vào

```
@Override  
public void onCreateContextMenu(ContextMenu menu, View  
    MenuInflater inf=getMenuInflater();  
    inf.inflate(R.menu.context_menu, menu);  
    super.onCreateContextMenu(menu, v, menuInfo);  
}
```

❑ Tạo và ánh xạ listview, tạo adapter gán cho listview

❑ Override onOptionsItemSelected để bắt sự kiện

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    AdapterView.AdapterContextMenuInfo info=  
        (AdapterView.AdapterContextMenuInfo)item.getMenuInfo();  
    int index=info.position;  
    if(item.getItemId()==R.id.can)  
        Toast.makeText(this, "can "+mang[index], Toast.LENGTH_SHORT).show();  
    if(item.getItemId()==R.id.cap)  
        Toast.makeText(this, "cap "+mang[index], Toast.LENGTH_SHORT).show();  
    if(item.getItemId()==R.id.nhai)  
        Toast.makeText(this, "nhai "+mang[index], Toast.LENGTH_SHORT).show();  
    return super.onOptionsItemSelected(item);  
}
```

❑ Lên onCreate đăng kí lv được nhận menu

```
registerForContextMenu(lv);
```


- ❑ PopupMenu là menu có thể được bung ra từ bất kì View nào
- ❑ Tạo file menu

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    <item
        android:id="@+id/popup1"
        android:title="popup 1" />
    <item
        android:id="@+id/popup2"
        android:title="popup 2" />
</menu>
```

- ❑ Trên giao diện kéo vào 1 button và 1 textview, qua java ánh xạ và bắt sự kiện cho button

❑ Trong hàm onClick của button viết để tạo Popup

```
PopupMenu popup = new PopupMenu(Main2Activity.this,tv);
MenuInflater inflater = popup.getMenuInflater();
inflater.inflate(R.menu.popup_menu, popup.getMenu());
popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener()
    @Override
    public boolean onMenuItemClick(MenuItem item) {
        if(item.getItemId()==R.id.popup1)
            Toast.makeText(Main2Activity.this, "popup 1", Toast.LENGTH_SHORT);
        if(item.getItemId()==R.id.popup2)
            Toast.makeText(Main2Activity.this, "popup 2", Toast.LENGTH_SHORT);
        return false;
    }
});
popup.show();
```

❑ Đối số thứ 2 của tạo đối tượng menu là View anchor (gắn menu vào view nào). Tức là ở đây ta nhấn vào button nhưng menu hiện ở textView

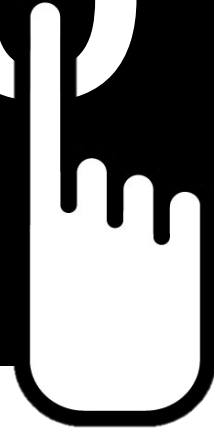
- ☐ ActionBar
- ☐ ToolBar
- ☐ ContextMenu
- ☐ OptionMenu
- ☐ PopupMenu





DEMO

OptionMenu
ContextMenu
PopupMenu





Cảm ơn