

ANDROID PROGRAMMING

LESSON 7

Agenda

- Bottom Navigation trong Android
- Tạo project với Bottom Navigation (01)
- Floating Action Button
- Ví dụ (02)

Android Bottom Navigation

- Android Bottom Navigation ở cuối màn hình cung cấp điều hướng giữa các chế độ views ở top-level trong ứng dụng. Điều này được giới thiệu trong design support library với khả năng tương thích ngược.
- Bottom Navigation có thể dễ dàng thêm bằng cách sử dụng thành phần BottomNavigationView. Chúng ta phải sử dụng gravitation hoặc các thuộc tính tương đối để làm cho nó xuất hiện ở dưới cùng của màn hình.

BottomNavigationView

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/navigation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:itemTextColor="@color/color_navigation"
        app:itemIconTint="@color/color_navigation"
        android:layout_alignParentBottom="true"
        app:menu="@menu/menu_navigation_bottom"
    />
</RelativeLayout>
```

Tạo project (01)

- BottomNavigationView
- TabLayout
- ViewPager
- Fragment
- FragmentStatePagerAdapter

07-01-NavigationBottom > app > src > main > res > layout > activity_main.xml

Android

com.example.a07_01_nav

com.example.a07_01_nav

java (generated)

res

color

drawable

culi.jpg

ic_add.xml

ic_home.xml

ic_launcher_backgrou

ic_launcher_foregrou

New

Link C++ Project with Gradle

Cut Ctrl+X

Copy Ctrl+C

Copy Paths Ctrl+Shift+C

Copy References Ctrl+Alt+Shift+C

Paste Ctrl+V

Find Usages Alt+F7

Analyze

Refactor

Add to Favorites

Kotlin File/Class

Drawable Resource File

Sample Data Directory

File

Scratch File Ctrl+Alt+Shift+Insert

Directory

C++ Class

C/C++ Source File

C/C++ Header File

Image Asset

Vector Asset

Kotlin Script

drawable

culi.jpg

ic_add.xml

ic_home.xml

ic_launcher_background.xml

ic_launcher_foreground.xml

ic_local_cafe.xml

ic_notifications.xml

ic_search.xml

menu_navigation_bottom.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:title="Home"
          android:icon="@drawable/ic_home"
          android:id="@+id/mHome"/>
    <item android:title="Notification"
          android:icon="@drawable/ic_notifications"
          android:id="@+id/mNoti"/>
    <item android:title="Search"
          android:icon="@drawable/ic_search"
          android:id="@+id/mSearch"/>
    <item android:title="Cafe"
          android:icon="@drawable/ic_local_cafe"
          android:id="@+id/mCafe"/>
</menu>
```

Các Fragment và color_navigation.xml

- fragment_home.xml FragmentHome.java
- fragment_notification.xml FragmentNotification.java
- fragment_search.xml FragmentSearch.java
- fragment_cafe.xml FragmentCafe.java

```
<selector
xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:color="@color/colorWhite"
        android:state_checked="false"/>
    <item android:color="@color/colorYellow"
        android:state_checked="true"/>
</selector>
```

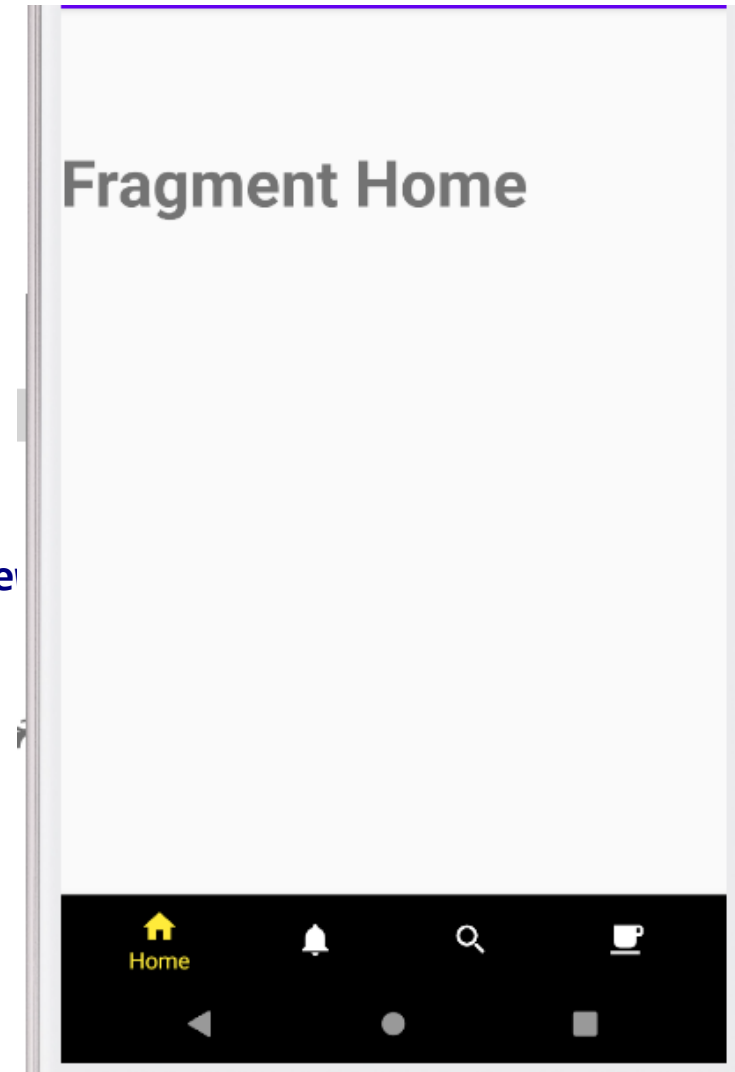


```
public class ViewPagerAdapter extends FragmentStatePagerAdapter {
    private int pageNum;
    public ViewPagerAdapter(@NonNull FragmentManager fm, int behavior) {
        super(fm, behavior);
        this.pageNum=behavior;
    }
    @NonNull
    @Override
    public Fragment getItem(int position) {
        switch(position){
            case 0: return new FragmentHome();
            case 1: return new FragmentNotification();
            case 2: return new FragmentSearch();
            case 3: return new FragmentCafe();
            default: return new FragmentHome();
        }
    }
    @Override
    public int getCount() {
        return pageNum;
    }
}
```

main_activity.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.viewpager.widget.ViewPager
        android:id="@+id/viewPager"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@id/navigation"/>
    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/navigation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:itemTextColor="@color/color_navigation"
        app:itemIconTint="@color/color_navigation"
        android:layout_alignParentBottom="true"
        app:menu="@menu/menu_navigation_bottom"
    />
</RelativeLayout>
```



3 Fragments

- fragment_moka.xml
- fragment_robusta.xml
- fragment_culi.xml

FragmentMoka.java

FragmentRobusta.java

FragmentCuli.java

```
public class CafeViewPagerAdapter extends FragmentStatePagerAdapter {
    public CafeViewPagerAdapter(@NonNull FragmentManager fm, int behavior) {
        super(fm, behavior);
    }
    @NonNull
    @Override
    public Fragment getItem(int position) {
        switch(position){
            case 0:return new FragmentMoka();
            case 1:return new FragmentRobusta();
            case 2:return new FragmentCuli();
            default:return new FragmentMoka();
        }
    }
    @Override
    public int getCount() {return 3;}
    @Nullable
    @Override
    public CharSequence getPageTitle(int position) {
        switch(position){
            case 0:return "Moka";
            case 1:return "Robusta";
            case 2:return "Culi";
            default:return "Moka";
        }
    }
}
```

FragmentCafe.java

```
public class FragmentCafe extends Fragment {
    private TabLayout tabLayout;
    private ViewPager viewPager;
    private View mView;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        mView= inflater.inflate(R.layout.fragment_cafe, container,
                                false);
        tabLayout=mView.findViewById(R.id.tabLayout);
        viewPager=mView.findViewById(R.id.viewPager);
        CafeViewPagerAdapter adapter=new CafeViewPagerAdapter(getChildFragmentManager(),3);
        viewPager.setAdapter(adapter);
        tabLayout.setupWithViewPager(viewPager);
        return mView;
    }
}
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private ViewPager viewPager;
    private BottomNavigationView navigationView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        viewPager=findViewById(R.id.viewPager);
        navigationView=findViewById(R.id.navigation);
        ViewPagerAdapter adapter=new ViewPagerAdapter(getSupportFragmentManager(),
            4);
        viewPager.setAdapter(adapter);
        viewPager.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {
            @Override
            public void onPageScrolled(int position, float positionOffset, int
positionOffsetPixels) {
            }
        })
    }
}
```

```
@Override
public void onPageSelected(int position) {
    switch(position){
        case 0:navigationView getMenu().findItem(R.id.mHome).setChecked(true);
            break;
        case 1:navigationView getMenu().findItem(R.id.mNoti).setChecked(true);
            break;
        case 2:navigationView getMenu().findItem(R.id.mSearch).setChecked(true);
            break;
        case 3:navigationView getMenu().findItem(R.id.mCafe).setChecked(true);
            break;
    }
}

});
```

```
navigationView.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean onNavigationItemSelectedListener(@NonNull MenuItem item) {
        switch(item.getItemId()){
            case R.id.mHome:viewPager.setCurrentItem(0);
                break;
            case R.id.mNoti:viewPager.setCurrentItem(1);
                break;
            case R.id.mSearch:viewPager.setCurrentItem(2);
                break;
            case R.id.mCafe:viewPager.setCurrentItem(3);
                break;
        }
        return true;
    }
});
}
```


Fragment Home



Home



MOKA

ROBUSTA

CULI

Fragment Cafe



Cafe

Floating Action Button

- **Sử dụng Floating Action Button (FAB)**
 - FAB là một nút bấm đặc biệt, chúng thường được biểu diễn bởi hình tròn với biểu tượng, nó nổi trên bề mặt các giao diện thiết kế, thường thì tùy ngữ cảnh bấm vào nó để thực hiện các hành động tương ứng.
 - Nguyên tắc thiết kế giao diện có sử dụng FAB theo triết lý Material Design được Google đề xuất ở đây: FAB Design
 - Thường sử dụng FAB tùy thuộc ngữ cảnh, và FAB là thao tác chủ đạo đưa ra cho người dùng thực hiện. Ví dụ đang ở màn hình xem danh sách email, có thể xuất hiện FAB để bấm vào đó tạo một Email mới

Floating Action Button

- Icon cho FAB
 - Ngoài cách tự thiết kế vẽ ICON sử dụng cho FAB, có rất nhiều ICON chuẩn của Google phù hợp tại material.io, bạn chỉ việc tìm Icon phù hợp, tải về và lưu vào res/drawable
 - Hoặc ở: android.R.drawable Icon Resources, ví dụ <http://www.hellohupten.com/2013/06/25/436/>
 - @android:drawable/ic_dialog_email

- Để sử dụng FAB ta cần thêm vào build.gradle thư viện:
- implementation
'com.android.support.design:X.X.X' với X là phiên bản muốn dùng.
- Ví dụ
implementation
'com.google.android.material:material:1.1.0'
implementation
'com.android.support.design:29.1.0'

- FAB có thể thêm vào các file layout với cú pháp dạng:

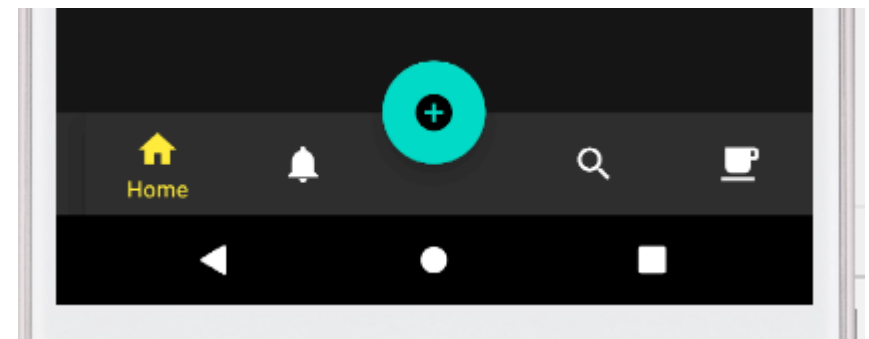
```
<com.google.android.material.floatingactionbutton.FloatingActionButton  
    android:id="@+id/fab"  
    android:src="@drawable/face"  
    app:fabSize="normal"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
>
```

- Với fabSize có thể nhận giá trị normal thì FAB ở cỡ chuẩn, và giá trị mini FAB ở cỡ nhỏ.

```
<com.google.android.material.floatingactionbutton.FloatingActionButton  
    android:id="@+id/fab"  
    android:layout_gravity="bottom|end"  
    android:layout_margin="20dp"  
    android:src="@drawable/ic_add"  
    app:backgroundTint="#BC54"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```



```
<com.google.android.material.bottomappbar.BottomAppBar
    android:id="@+id/bottomAppBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom"
>
<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/navigation"
    app:menu="@menu/menu_navigation_bottom"
    android:layout_width="match_parent"
    app:itemTextColor="@color/color_navigation"
    app:itemIconTint="@color/color_navigation"
    android:layout_height="match_parent"/>
</com.google.android.material.bottomappbar.BottomAppBar>
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_add"
    app:layout_anchor="@id/bottomAppBar"
/>
```



- Để định vị FAB một cách linh động, chúng ta sẽ sử dụng CoordinatorLayout để trình bày Layout, vì loại View này giúp cho việc tương tác giữa các phần tử View con của nó được dễ dàng hơn
- Sử dụng một AppBarLayout để chứa ActionBar (Toolbar) vì AppBarLayout hỗ trợ tính năng cuộn. Khi sử dụng AppBarLayout như code XML trên, ta cần thiết lập không sử dụng Toolbar mặc định bằng cách thiết lập theme mà Activity sử dụng
- Ví dụ, tạo theme có thiết lập không sử dụng ActionBar hệ thống mặc định trong styles.xml

```
<style name="AppTheme" parent="Theme.MaterialComponents">
```


- End of Lesson 8



Thank you!