用户界面开发进阶

刘成

ByteDance字节跳动

Animation

Animation

- 意义
- View 动画
- Drawable 动画

意义

- 用户体验
- 视觉反馈
- Impressive
- Focused
- Expressive

View 相关动画

- View Animation: android.view.animation
- Property Animation: android.animation

示例 - PropertyAnimation, XML

```
// res/animator/fade.xml
<objectAnimator xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="1000"
    android:propertyName="alpha"
    android:valueTo="0.1f" />

// activity
Animator anim = AnimatorInflater.loadAnimator(activity, R.animator.fade);
anim.setTarget(v);
anim.start();
```

示例 - PropertyAnimation，Code

```
ObjectAnimator alphaAnimation = ObjectAnimator.ofFloat(alphaButton, View.ALPHA, 0);
alphaAnimation.setRepeatCount(1);
alphaAnimation.setRepeatMode(ValueAnimator.REVERSE);
alphaAnimation.start();
```

示例 - More Property

```
// translationX, translationY

ObjectAnimator transAnim = ObjectAnimator.ofFloat(translateButton, "translationX", 800);

<objectAnimator xmlns:android="http://schemas.android.com/apk/res/android"
    android:propertyName="translationX"
    android:duration="300"
    android:valueTo="800"/>
```

示例 - More Property

```
// rotation, rotationX, rotationY
ObjectAnimator rotateAnim = ObjectAnimator.ofFloat(rotateButton, “rotation”, 360);

<objectAnimator xmlns:android="http://schemas.android.com/apk/res/android"
    android:propertyName="rotation"
    android:valueFrom="0"
    android:duration="300"
    android:valueTo="360"/>
```

示例 - More Property

```
// scaleX, scaleY
PropertyValuesHolder pvhX = PropertyValuesHolder.ofFloat(View.SCALE_X, 2);
PropertyValuesHolder pvhY = PropertyValuesHolder.ofFloat(View.SCALE_Y, 2);
ObjectAnimator scaleAnim = ObjectAnimator.ofPropertyValuesHolder(scaleButton, pvhX,
pvhY);

<set xmlns:android="http://schemas.android.com/apk/res/android">
    <objectAnimator
        android:propertyName="scaleX"
        android:duration="300"
        android:valueTo="2"/>
    <objectAnimator
        android:propertyName="scaleY"
        android:duration="300"
        android:valueTo="2"/>
</set>
```

示例 - AnimatorSet，XML

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:ordering="sequentially">
    <objectAnimator android:propertyName="alpha"
        android:valueTo="0"/>
    <objectAnimator android:propertyName="translationX"
        android:valueTo="800"/>
    <objectAnimator android:propertyName="rotation"
        android:valueFrom="0"
        android:valueTo="360"/>
    <set>
        <objectAnimator android:propertyName="scaleX"
            android:valueTo="2"/>
        <objectAnimator android:propertyName="scaleY"
            android:valueTo="2"/>
    </set>
</set>
```

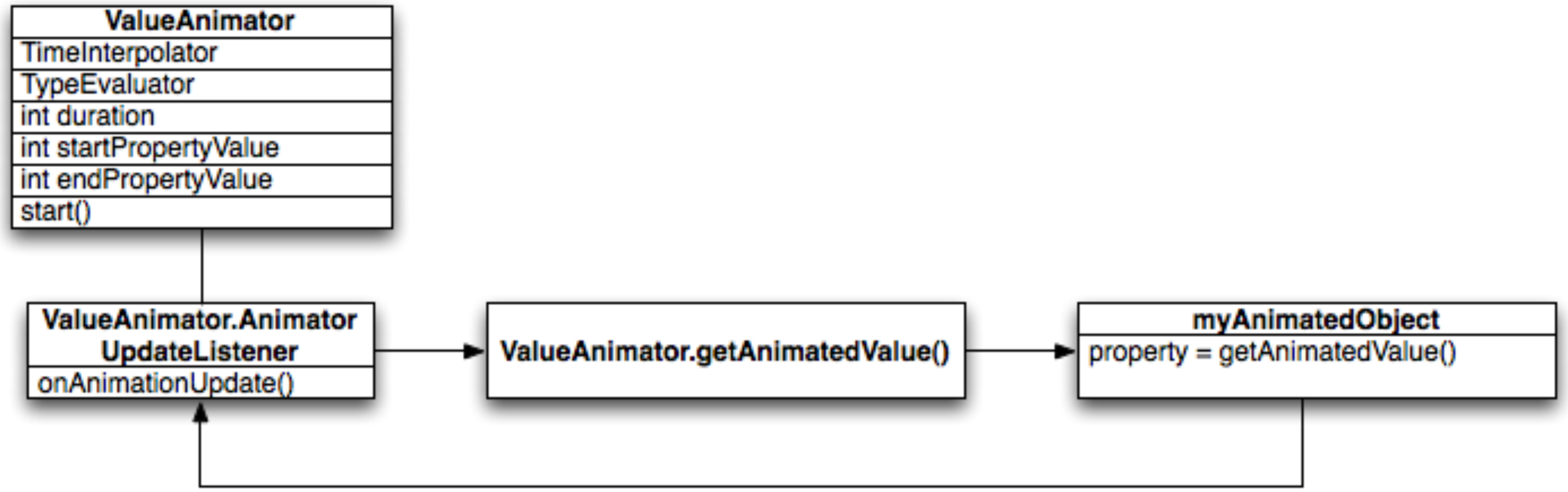
示例 - AnimatorSet，Code

```
AnimatorSet setAnimation = new AnimatorSet();
setAnimation.play(translateAnimation).after(alphaAnimation).before(rotateAnimation);
setAnimation.play(rotateAnimation).before(scaleAnimation);

setAnimation.playSequentially(alphaAnimation, translateAnimation, rotateAnimation,
scaleAnimation);

setAnimation.playTogether(alphaAnimation, translateAnimation, rotateAnimation,
scaleAnimation);
```

基本的类

- Property
- ObjectAnimator
- AnimatorSet

背后 - ValueAnimator



ValueAnimator
TimeInterpolator
TypeEvaluator
int duration
int startPropertyValue
int endPropertyValue
start()

ValueAnimator.Animator UpdateListener
onAnimationUpdate()

ValueAnimator.getAnimatedValue()

myAnimatedObject
property = getAnimatedValue()

背后 - ValueAnimator

```
ValueAnimator animation = ValueAnimator.ofFloat(0f, 100f);
animation.setDuration(1000);
animation.addUpdateListener(new ValueAnimator.AnimatorUpdateListener() {
    @Override
    public void onAnimationUpdate(ValueAnimator updatedAnimation) {
        float animatedValue = (float)updatedAnimation.getAnimatedValue();
        textView.setTranslationX(animatedValue);
    }
});
animation.start();
```

示例 - Crossfade

```
mLoadingView.animate()
        .alpha(0f)
        .setDuration(mShortAnimationDuration)
        .setListener(new AnimatorListenerAdapter() {
          @Override
          public void onAnimationEnd(Animator animation) {
            mLoadingView.setVisibility(View.GONE);
          }
        });
```

Drawable 动画

- AnimationDrawable
- AnimationVectorDrawable
- Lottie

示例 - AnimationDrawable

```
// res/drawable/rocket.xml
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot="true">
    <item android:drawable="@drawable/rocket_thrust1" android:duration="200" />
    <item android:drawable="@drawable/rocket_thrust2" android:duration="200" />
    <item android:drawable="@drawable/rocket_thrust3" android:duration="200" />
</animation-list>

// activity
ImageView rocketImage = (ImageView) findViewById(R.id.rocket_image);
rocketImage.setBackgroundResource(R.drawable.rocket_thrust);
rocketAnimation = (AnimationDrawable) rocketImage.getBackground();
rocketAnimation.start();
```

示例 - Lottie

```
<com.airbnb.lottie.LottieAnimationView
        android:id="@+id/animation_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:lottie_rawRes="@raw/hello_world" // from res/raw
        app:lottie_fileName="hello_world.json" // from assets/
        app:lottie_loop="true"
        app:lottie_autoPlay="true" />
```

Fragment

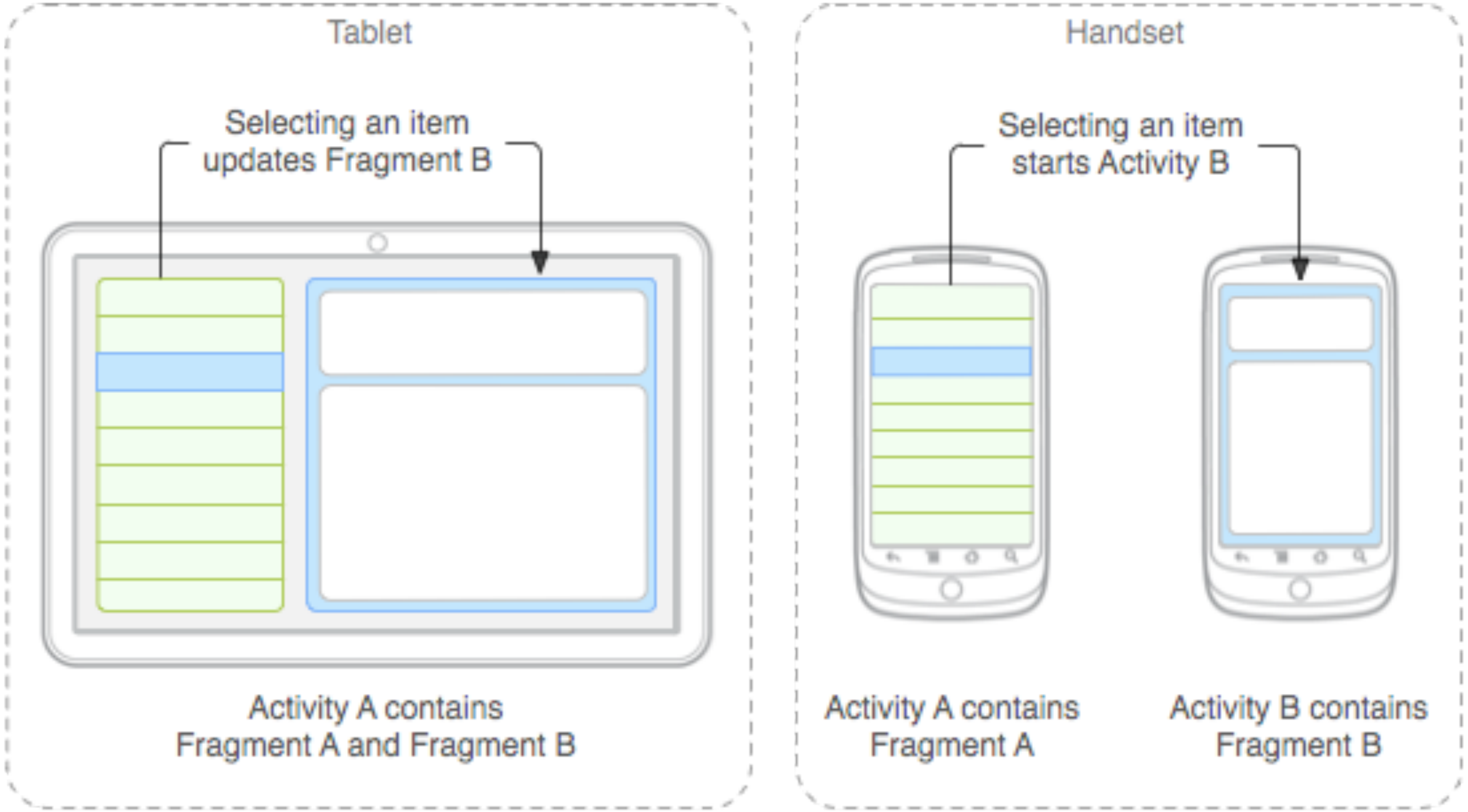
Fragment

- 概念和作用
- 生命周期
- 如何动态添加
- 如何和 Fragment/Activity 通信

Fragment

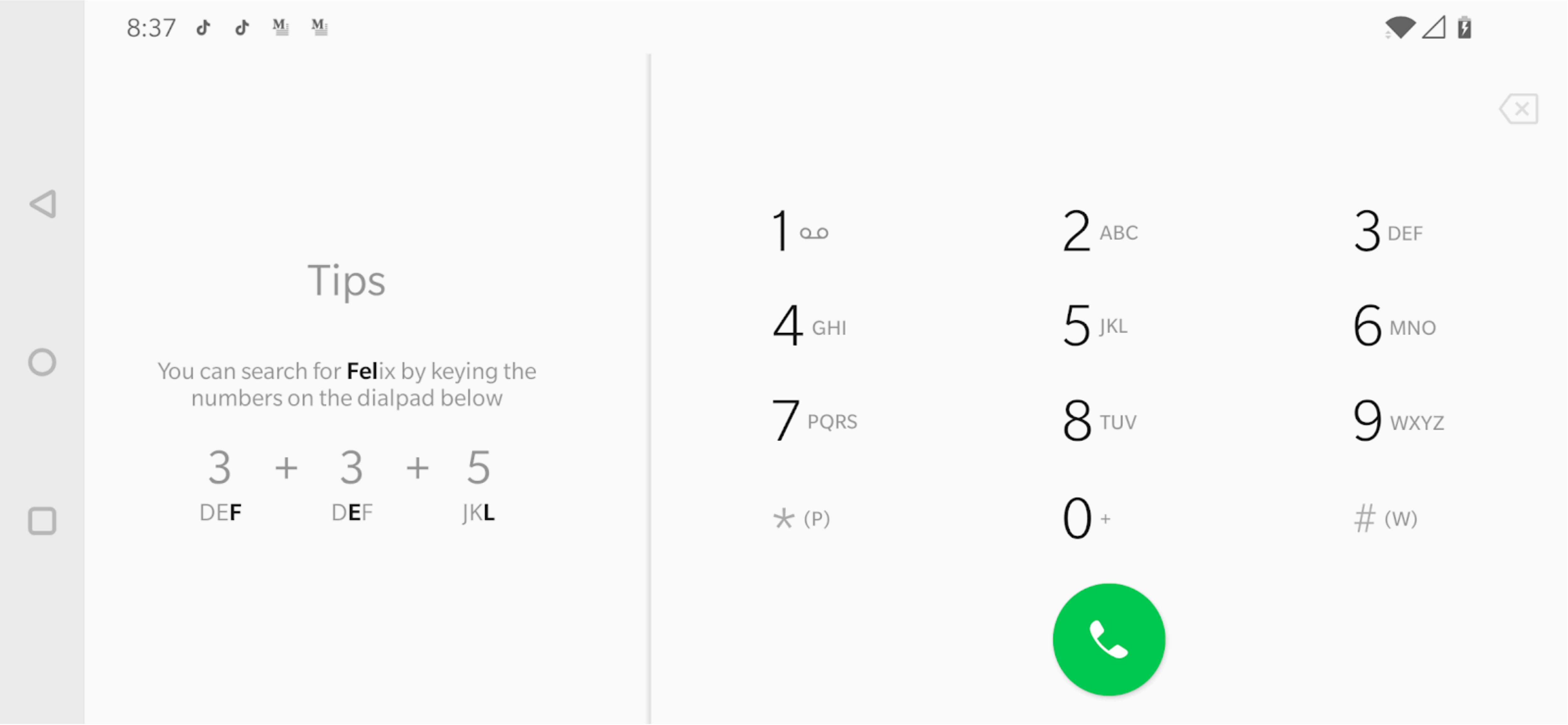
- App component, mini-activity
- 可重用 UI 单元

Fragment - UI 重用



Tablet
Selecting an item updates Fragment B
Activity A contains Fragment A and Fragment B
Handset
Selecting an item starts Activity B
Activity A contains Fragment A
Activity B contains Fragment B

Fragment - Responsive Design



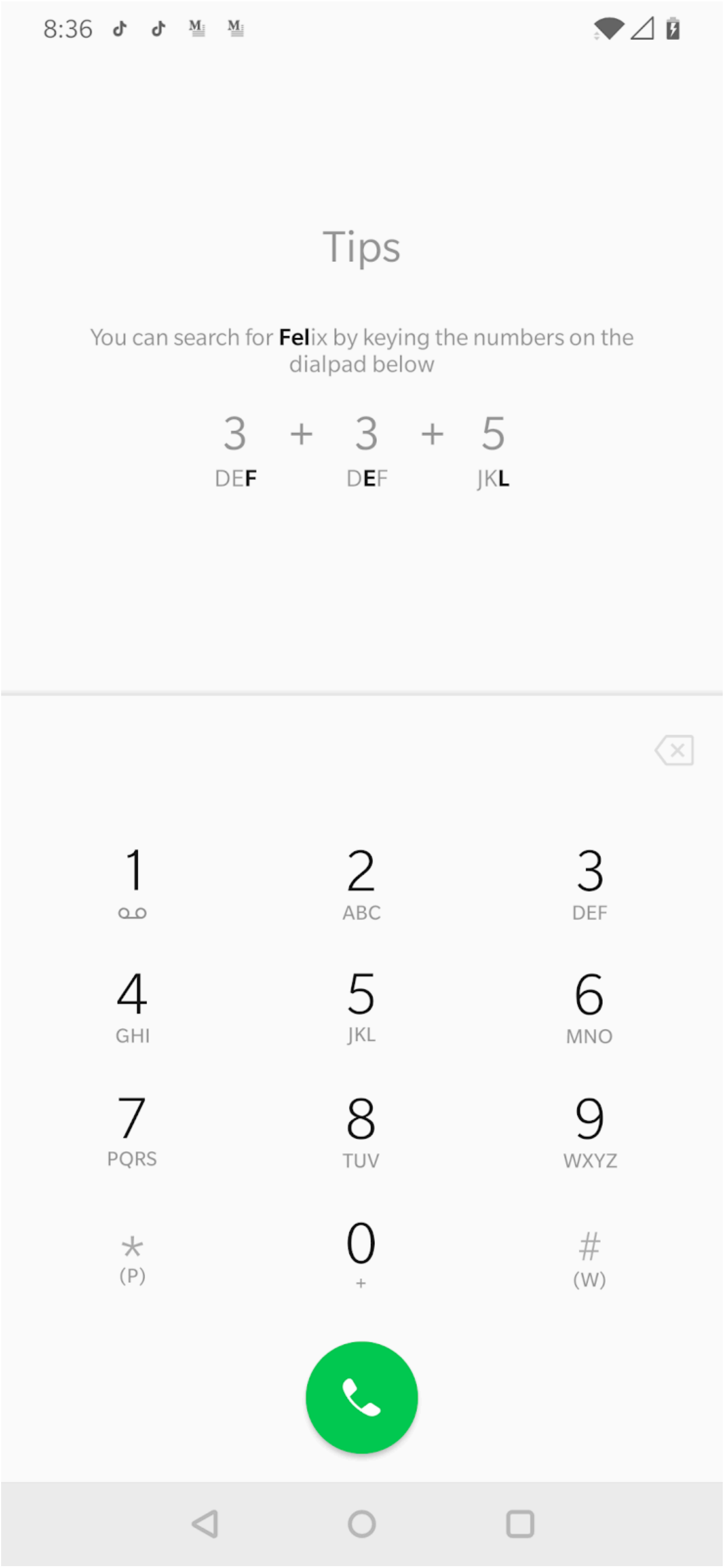
8:37

Tips

You can search for **Fel**ix by keying the numbers on the dialpad below

3 + 3 + 5

DE**F** DE**F** JK**L**



8:36

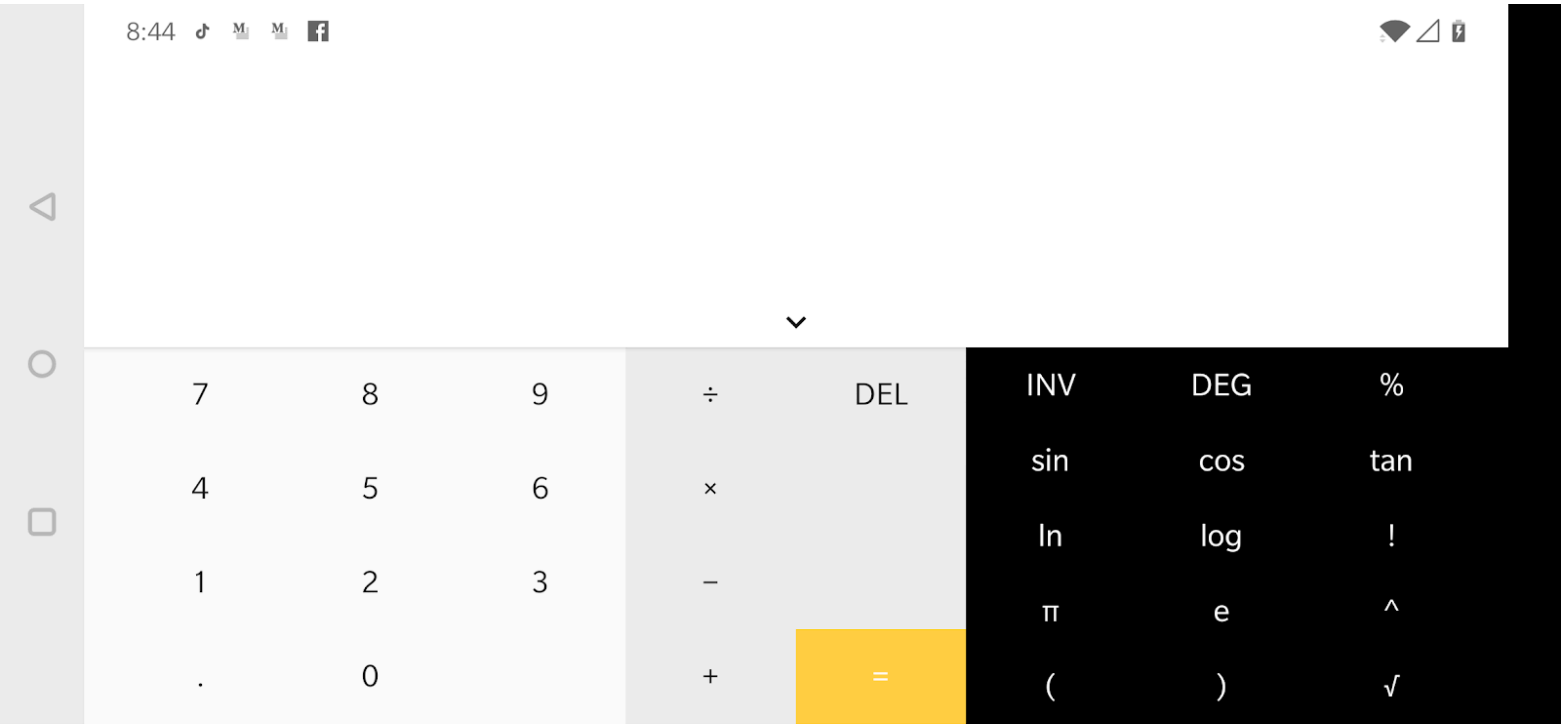
Tips

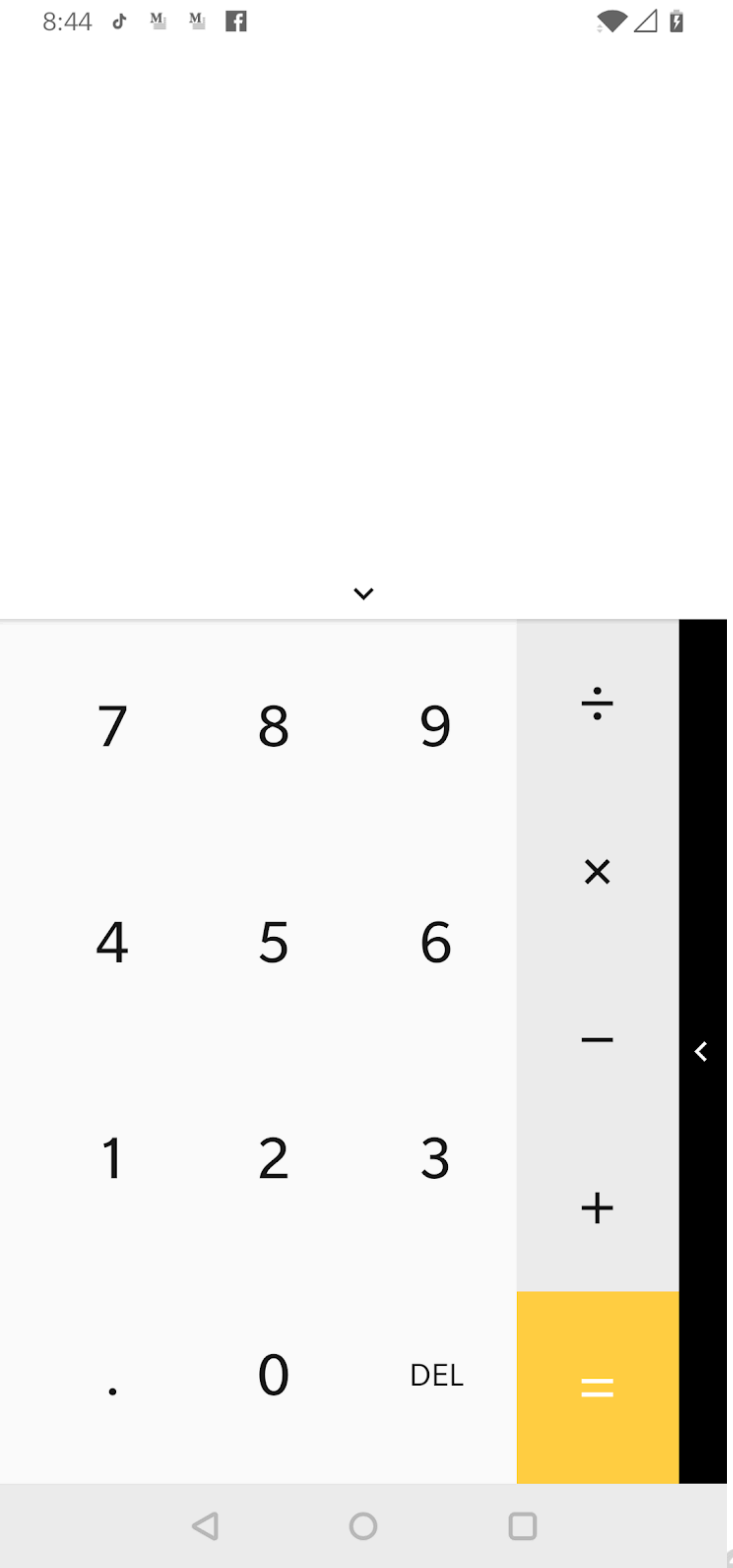
You can search for **Fel**ix by keying the numbers on the dialpad below

3 + 3 + 5

DE**F** DE**F** JK**L**

Fragment - Responsive Design





Fragment - Why

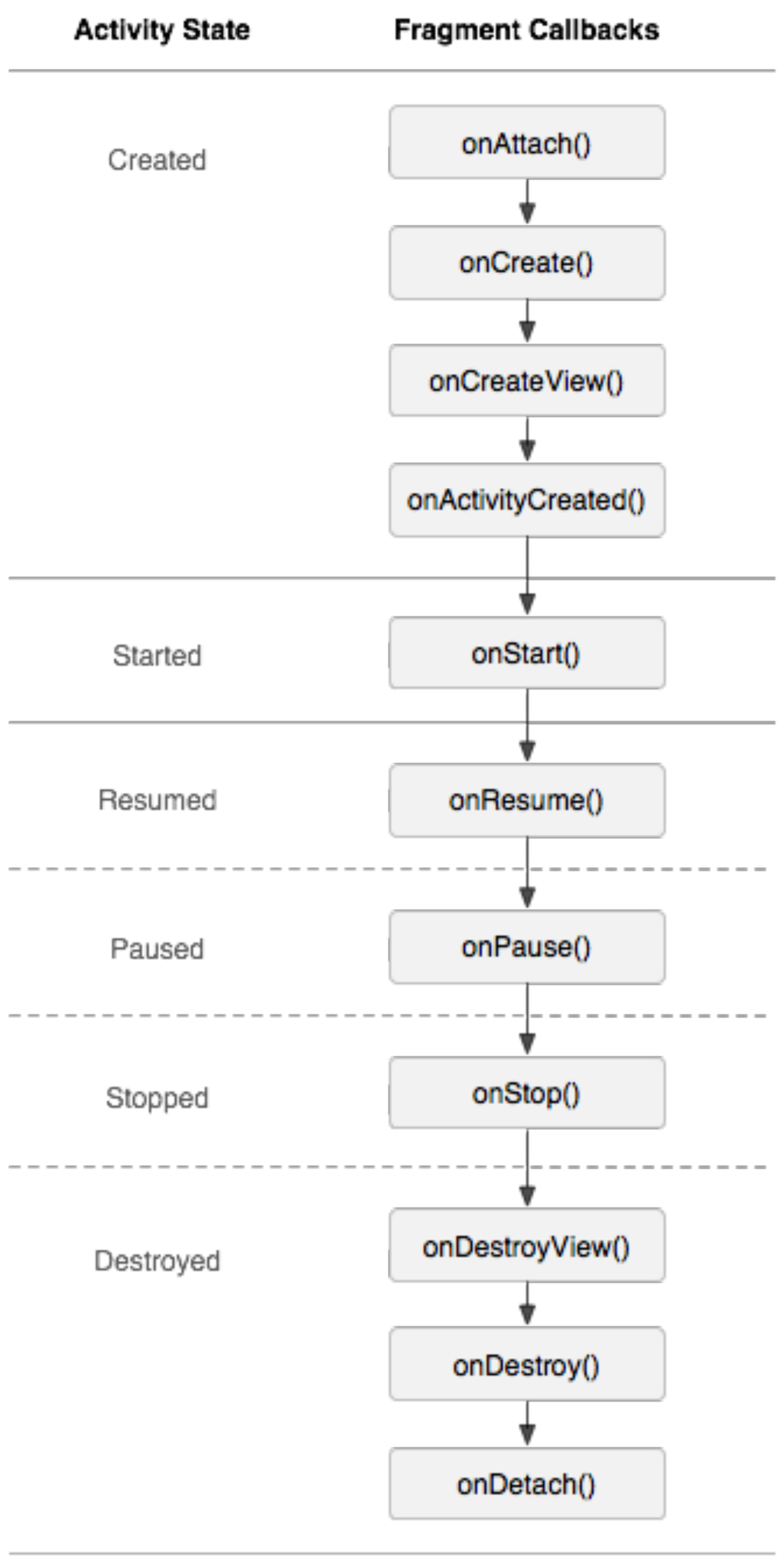
- Activity 模块化
- 相比 View，带有生命周期管理
- 可重用，灵活
- 可以动态添加和删除

使用哪个?

- android.support.v4.app.Fragment
- android.app.Fragment
- androidx.fragment.app.Fragment

生命周期

- onAttach/onDetach
- onCreate/onDestroy
- onCreateView/onDestroyView
- onActivityCreated
- onStart/onStop
- onResume/onPause



示例 - Lifecycle

- 定义 fragment 布局文件
- 定义 fragment 类
- 在 activity 布局文件中嵌入 fragment

示例 - Lifecycle - 1

- **fragment_hello.xml**

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="@string/hello_fragment"/>
</FrameLayout>
```

示例 - Lifecycle - 2

- **HelloFragment.java**

```
public class HelloFragment extends Fragment {
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
            @Nullable ViewGroup container,
            @Nullable Bundle savedInstanceState) {
      return inflater.inflate(R.layout.fragment_hello, container, false);

}
```

示例 - Lifecycle - 3

- **activity_fragment.xml**

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

    <fragment android:name="com.example.fragment.f2.HelloFragment"
        android:id="@+id/hello_fragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</FrameLayout>
```

动态添加/删除 Fragment

- FragmentManager
  - 动态添加/替换/删除 Fragment
  - FragmentTransaction
- Fragment 容器
  - 定义 Fragment 的位置和大小

示例 - 动态修改 Fragment - 1

- 在 activity 布局文件中定义 fragment 容器

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/androi
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <FrameLayout
        android:id="@+id/fragment_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</FrameLayout>
```

示例 - 动态修改 Fragment - 2

- 使用 FragmentTransaction 添加 Fragment

```
getSupportFragmentManager().beginTransaction()
        .add(R.id.fragment_container, new HelloFragment())
        .commit();
```

Fragment/Activity 之间的通信

- 构造 Fragment 时传递参数
- 通过接口和回调

示例 - 通信 - 传参

```
public class StaticColorFragment extends Fragment {
    private static final String KEY_COLOR = "key_color";

    public static StaticColorFragment newFragment(int color) {
        StaticColorFragment fragment = new StaticColorFragment();
        Bundle args = new Bundle();
        args.putInt(KEY_COLOR, color);
        fragment.setArguments(args);
        return fragment;
    }

    public View onCreateView(...) {
        int color = Color.CYAN;
        Bundle args = getArguments();
        if (args != null) {
            color = args.getInt(KEY_COLOR, Color.CYAN);
        }
        View view = new View(inflater.getContext());
        view.setBackgroundColor(color);
        return view;
    }
}
```

示例 - 通信 - Listener

```
public class DynamicColorFragmentActivity extends AppCompatActivity implements OnColorSelectListener {
    ...

    @Override
    public void onAttachFragment(Fragment fragment) {
        super.onAttachFragment(fragment);
        if (fragment instanceof DynamicColorSelectorFragment) {
            ((DynamicColorSelectorFragment) fragment).setOnColorSelectListener(this);
        }
    }

    @Override
    public void onColorSelect(@ColorInt int color) {
        Fragment f = getSupportFragmentManager().findFragmentById(R.id.color_fragment_container);
        if (f instanceof DynamicColorFragment) {
            ((DynamicColorFragment) f).updateColor(color);
        }
    }
}
```

示例 - Master Detail

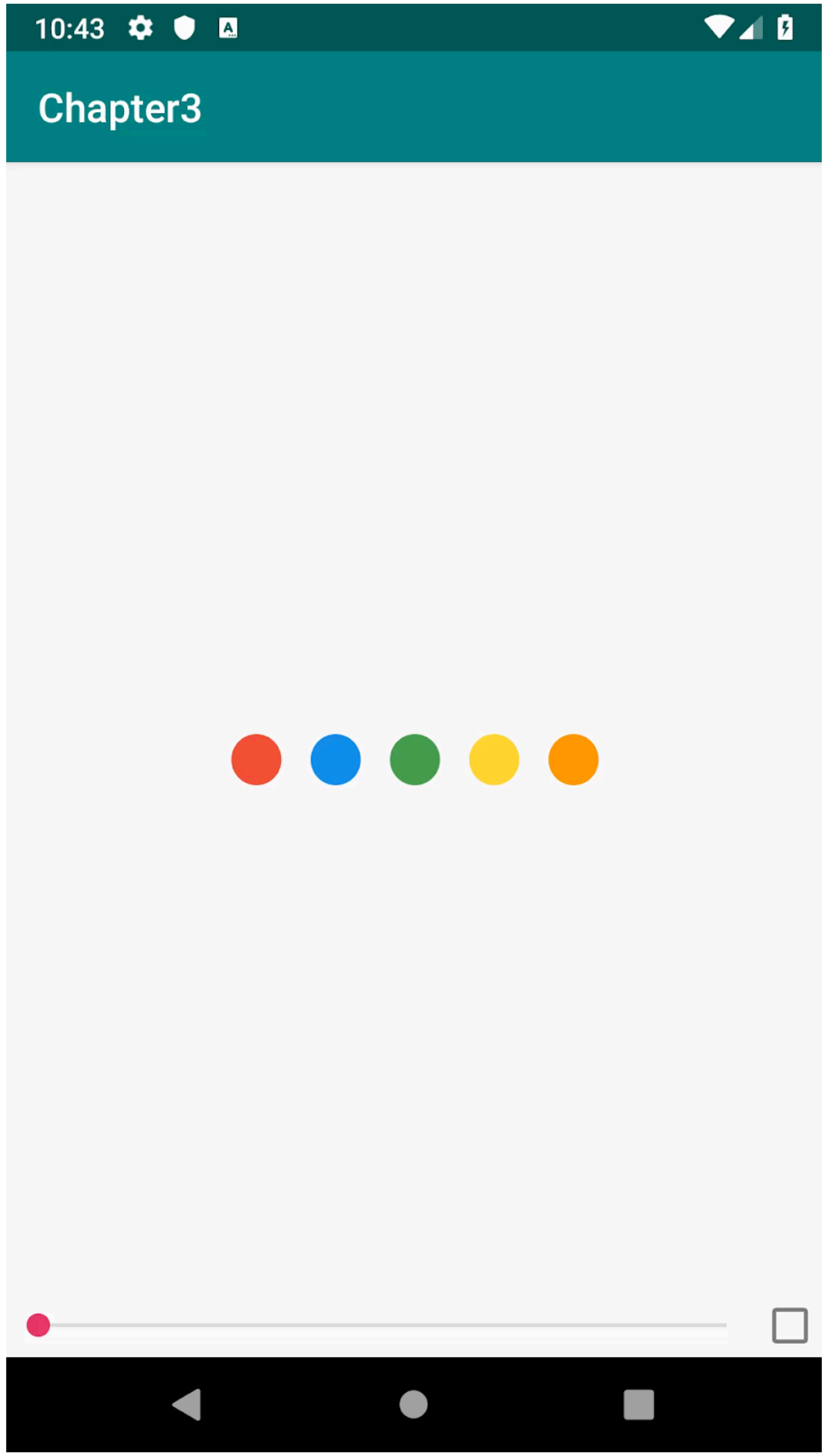
- Portrait
  - Master Activity: Item List
  - Detail Activity: Item Detail
- Landscape
  - One Activity: List & Detail

Exercise

练习1 - ch3ex1

- 引入 Lottie 库实现简单的图标动画
- 1. 在 ch3ex1 的 build.gradle 中引入 lottie 库
- 2. 在 activity_main.xml 中添加 LottieAnimationView
- 3. 在 SeekBar 的回调中修改 LottieAnimationView 的进度

练习1 - ch3ex1

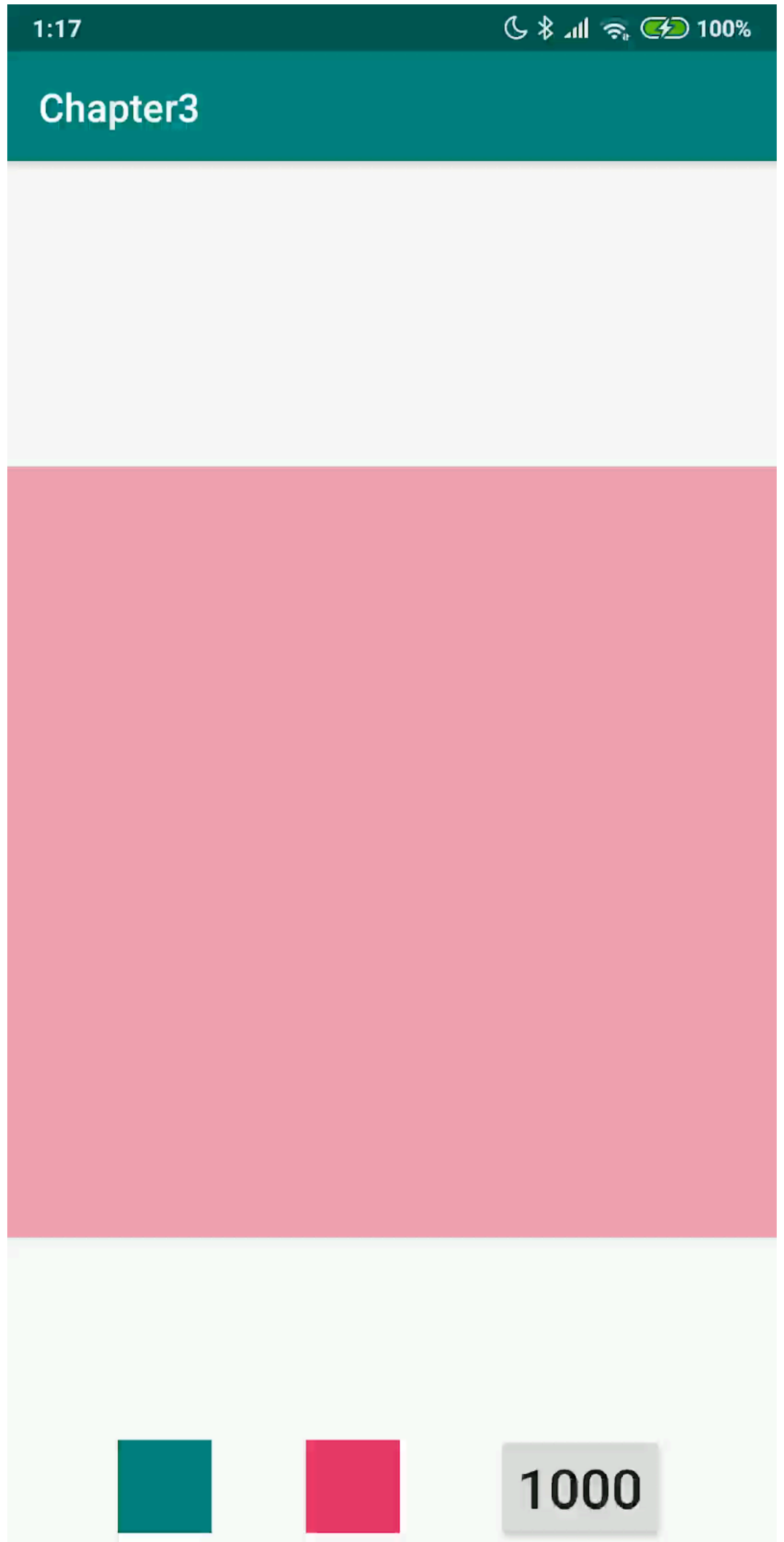

10:43

Chapter3

练习2 - ch3ex2

- 使用属性动画，练习 AnimatorSet 和 scale/fade 等基本动画样式
- 1. 添加 scale 动画
- 2. 添加 alpha 动画
- 3. 组合到 AnimatorSet 中

练习2 - ch3ex2



THANKS.

ByteDance 字节跳动