



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

Lập trình Android

Bài 25: *Animations*

Phòng LT & Mạng

<http://csc.edu.vn/lap-trinh-va-csdl>





Nội dung

1. Android Animations
2. Property Animations
3. Activity Transitions
4. Fragment Transitions



Android Animations

- Android hỗ trợ **animation** cho cả view và việc chuyển tiếp giữa các activity
- Có 3 hệ thống animation với cách thức hoạt động khác nhau:
 - Property Animations
 - View Animations
 - Transition Animations
- Có thể tạo animation bằng Java code hoặc từ XML resource file



Property Animations

- Các thuộc tính animated phổ biến trên view

Thuộc tính (Property)	Mô tả
alpha	Fade in, fade out (độ trong suốt)
rotation, rotationX, rotationY	Spin, flip (xoay)
scaleX, scaleY	Grow, shrink (độ lớn)
x, y, z	Position (vị trí)
translationX, translationY, translationZ (API 21+)	Offset from Position (độ dời)



Property Animations - Object Animator

- Ví dụ: Tạo hiệu ứng fade out cho textview bằng cách thay đổi độ trong suốt (**alpha**)

```
ObjectAnimator fadeAltAnim = ObjectAnimator.ofFloat(tvHello, View.ALPHA, 1, 0.5f);  
fadeAltAnim.start();
```
- Các thuộc tính bao gồm:
ALPHA, ROTATION, ROTATION_X, ROTATION_Y, SCALE_X, SCALE_Y, TRANSLATION_X, TRANSLATION_Y, TRANSLATION_Z, X, Y, Z
- Thiết lập thời gian và tính lặp (repeat) cho animation:

```
ObjectAnimator scaleAnim = ObjectAnimator.ofFloat(tvHello, View.SCALE_X, 1.0f, 2.0f);  
scaleAnim.setDuration(3000);  
scaleAnim.setRepeatCount(ValueAnimator.INFINITE); // lặp vô tận  
scaleAnim.setRepeatMode(ValueAnimator.REVERSE); // quay ngược lại trạng thái ban đầu  
scaleAnim.start();
```



Property Animations - Interpolation

- Qui định tốc độ, quỹ đạo của animation:

```
ObjectAnimator moveAnim = ObjectAnimator.ofFloat(view, View.Y, 1000);  
moveAnim.setDuration(2000);  
moveAnim.setInterpolator(new BounceInterpolator());  
moveAnim.start();
```

- **Các interpolators:**

- BounceInterpolator: View bị bật ngược trở lại ở cuối animation
- AccelerateInterpolator: Tốc độ chuyển động bắt đầu chậm, sau đó nhanh dần
- DecelerateInterpolator: Tốc độ chuyển động bắt đầu nhanh, sau đó chậm dần
- LinearInterpolator: Tốc độ chuyển động không thay đổi (chuyển động đều)



Property Animations – Animation Lifecycle

- Có thể thêm một **AnimatorListener** để quản lý các events trong suốt lifecycle từ khi bắt đầu cho đến khi kết thúc một animation:

```
anim.addListener(new Animator.AnimatorListener() {  
    @Override  
    public void onAnimationStart(Animator animator) {  
  
    }  
  
    @Override  
    public void onAnimationEnd(Animator animator) {  
  
    }  
  
    @Override  
    public void onAnimationCancel(Animator animator) {  
  
    }  
  
    @Override  
    public void onAnimationRepeat(Animator animator) {  
  
    }  
});
```



Property Animations – AnimatorSet

- Có thể thực hiện đồng thời nhiều **ObjectAnimator** bằng cách sử dụng **AnimatorSet**:

```
AnimatorSet set = new AnimatorSet();  
set.playTogether(  
    ObjectAnimator.ofFloat(btnHello, View.SCALE_X, 1.0f, 2.0f)  
        .setDuration(2000),  
    ObjectAnimator.ofFloat(btnHello, View.SCALE_Y, 1.0f, 2.0f)  
        .setDuration(2000),  
    ObjectAnimator.ofObject(btnHello, "backgroundColor", new ArgbEvaluator(),  
        /*Red*/0xFFFF8080, /*Blue*/0xFF8080FF)  
        .setDuration(2000)  
);  
set.start();
```




Property Animations – AnimatorSet (2)

- Có thể animate các animator sets cùng với nhau:

// Định nghĩa set thứ nhất

```
ObjectAnimator anim1 = ObjectAnimator.ofFloat(btnHello, "scaleX", 2.0f);
```

```
ObjectAnimator anim2 = ObjectAnimator.ofFloat(btnHello, "scaleY", 2.0f);
```

```
AnimatorSet set1 = new AnimatorSet();
```

```
set1.playTogether(anim1, anim2);
```

// Định nghĩa set thứ hai

```
ObjectAnimator anim3 = ObjectAnimator.ofFloat(btnHello, "X", 300);
```

```
ObjectAnimator anim4 = ObjectAnimator.ofFloat(btnHello, "Y", 300);
```

```
AnimatorSet set2 = new AnimatorSet();
```

```
set2.playTogether(anim3, anim4);
```

// Play set thứ hai sau set thứ nhất

```
AnimatorSet set3 = new AnimatorSet();
```

```
set3.playSequentially(set1, set2);
```

```
set3.start();
```



Property Animations - ViewPropertyAnimator

- Một view thực hiện phương thức `animate()` với các thuộc tính như: độ trong suốt, xoay...

```
tvHello.animate().alpha(0.2f).xBy(-100).yBy(100);
```

- Hoặc

```
btnHello.animate().alpha(0.5f).rotation(90f).  
    scaleX(2).xBy(100).yBy(100).setDuration(1000).setStartDelay(10).  
    setListener(new AnimatorListenerAdapter() {  
        @Override  
        public void onAnimationStart(Animator animation) {  
            Toast.makeText(MainActivity.this, "Started...", Toast.LENGTH_SHORT).show();  
        }  
    });
```

- Hoặc

```
ViewCompat.animate(btnHello).alpha(0.2f).xBy(-100).yBy(100);
```





Property Animations - Sử dụng XML file

- Có thể tạo các property animations từ XML file. Ví dụ, tạo file res/animator/fade_out.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<objectAnimator xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="1000"
    android:propertyName="alpha"
    android:valueTo="0" />
```

- Load nội dung XML file vào Activity và thực hiện animation:

```
Animator anim = AnimatorInflater.loadAnimator(this, R.animator.fade_out);
anim.setTarget(btnHello);
anim.start();
```



Property Animations - Sử dụng XML file (2)

- Để thực hiện nhiều animation đồng thời, có thể tạo file XML res/animator/multi_anim.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:ordering="together">
    <objectAnimator
        android:propertyName="alpha"
        android:valueTo="0.5" />
    <objectAnimator
        android:propertyName="rotation"
        android:valueTo="90.0" />
    <objectAnimator
        android:propertyName="scaleX"
        android:valueTo="2.0" />
    <objectAnimator
        android:propertyName="translationX"
        android:valueTo="100.0" />
    <objectAnimator
        android:propertyName="translationY"
        android:valueTo="100.0" />
</set>
```

```
Animator anim = AnimatorInflater.loadAnimator(this,
    R.animator.multi_anim);
anim.setTarget(btnHello);
anim.setDuration(1000);
anim.setStartDelay(10);
anim.start();
```



Activity Transitions – Hiệu ứng chuyển activity

- Khi chuyển từ activity này sang activity khác, có thể thực hiện hiệu ứng chuyển màn hình bằng cách gọi phương thức **overridePendingTransition** sau hàm **startActivity**

```
Intent i = new Intent(MainActivity.this, SecondActivity.class);
```

```
startActivity(i);
```

```
overridePendingTransition(R.anim.right_in, R.anim.left_out);
```

- Với tham số thứ nhất là “enter” animation, tham số thứ hai là “exit” animation

```
<?xml version="1.0" encoding="utf-8"?>
<set
xmlns:android="http://schemas.android.com/apk/res/android"
  android:interpolator="@android:anim/linear_interpolator">
  <translate
    android:duration="500"
    android:fromXDelta="100%p"
    android:toXDelta="0" />
</set>
```

res/anim/right_in.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set
xmlns:android="http://schemas.android.com/apk/res/android"
  android:interpolator="@android:anim/linear_interpolator">
  <translate
    android:duration="500"
    android:fromXDelta="0"
    android:toXDelta="-100%p" />
</set>
```

res/anim/left_out.xml



Activity Transitions – Hiệu ứng chuyển activity (2)

- Xử lý khi nhấn nút **Back** để thêm hiệu ứng quay về màn hình trước đó:

```
public class SecondActivity extends AppCompatActivity {  
    @Override  
    public void onBackPressed() {  
        finish();  
        overridePendingTransition(R.anim.left_in, R.anim.right_out);  
    }  
}
```

- Với các animations ngược lại với các animations trên:

```
<?xml version="1.0" encoding="utf-8"?>  
<set  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:interpolator="@android:anim/linear_interpolator">  
    <translate  
        android:duration="500"  
        android:fromXDelta="0"  
        android:toXDelta="-100%p" />  
    </set>
```

res/anim/right_out.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<set xmlns:android="http://schemas.android.com/apk/res/android"  
    android:interpolator="@android:anim/linear_interpolator">  
    <translate  
        android:duration="500"  
        android:fromXDelta="100%p"  
        android:toXDelta="0" />  
    </set>
```

res/anim/left_in.xml





Fragment Transitions – Hiệu ứng chuyển fragments

- Sử dụng hàm **setCustomAnimations** của **FragmentTransaction** để thiết lập hiệu ứng chuyển fragment:

```
FragmentTransaction fts = getSupportFragmentManager().beginTransaction();
fts.setCustomAnimations(R.anim.slide_in_left, R.anim.slide_out_right);
ft.replace(R.id.fragment_container, newFragment, "fragment");
ft.commit();
```

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <translate
    android:duration="@android:integer/config_mediumAnimTime"
    android:fromXDelta="-50%p"
    android:toXDelta="0" />
  <alpha
    android:duration="@android:integer/config_mediumAnimTime"
    android:fromAlpha="0.0"
    android:toAlpha="1.0" />
</set>
```

res/anim/slide_in_left.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <translate
    android:duration="@android:integer/config_mediumAnimTime"
    android:fromXDelta="0"
    android:toXDelta="50%p" />
  <alpha
    android:duration="@android:integer/config_mediumAnimTime"
    android:fromAlpha="1.0"
    android:toAlpha="0.0" />
</set>
```

res/anim/slide_out_right.xml



