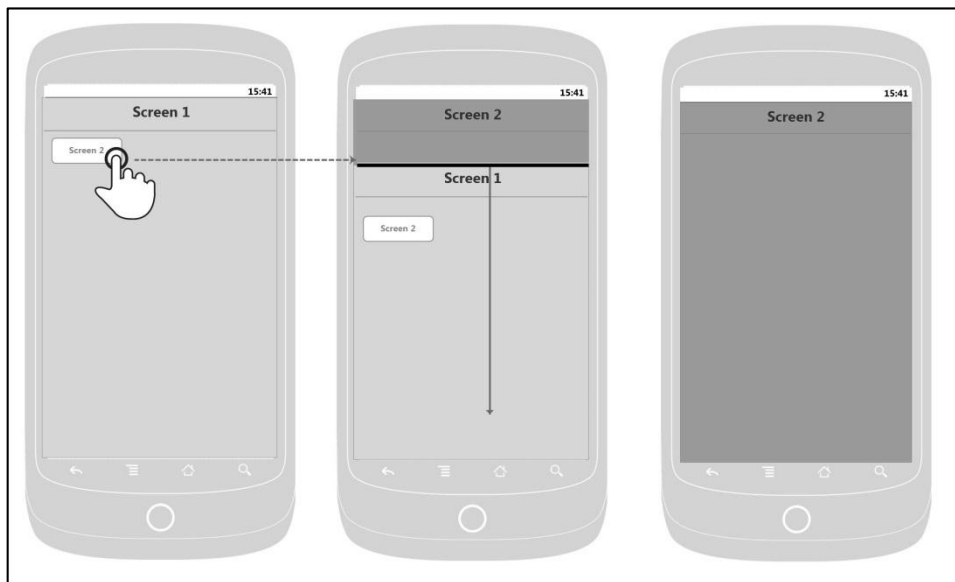


# Bài 5

## CÁC CHUYỂN HOẠT TRONG ỨNG DỤNG

### 1. PROPERTY ANIMATION

- Property Animation là dạng chuyển hoạt hỗ trợ chúng ta trong việc lập trình các chuyển hoạt cho các thuộc tính của một đối tượng nào đó bằng các thay đổi 2 giá trị 0 và 1. Ví dụ, khi ta có thể thay đổi màu sắc, độ trong suốt để làm hiệu ứng xuất hiện và biến mất của đối tượng, hoặc thay đổi các kích thước dòng chữ chẳng hạn.
- Được giới thiệu từ phiên bản Android 3.0 (API 11) có thể nói đây là bộ công cụ mạnh mẽ trong việc lập trình chuyển hoạt bởi vì chúng ta có thể ứng dụng nó vào bất kỳ thuộc tính nào chúng ta muốn.
- Mỗi chuyển hoạt sẽ được chứa trong một tập tin XML riêng biệt trong thư mục res/animator. Giống như Layout và Drawable, tên của tập tin sẽ được sử dụng làm định danh cho chuyển hoạt đó. Việc định nghĩa các chuyển hoạt chỉ đơn giản là gán một giá trị cho một thuộc tính nào đó, sau đó cấp phát cho đối tượng thực hiện chuyển hoạt đó.



*Hình 1.1. Ví dụ mô tả Property Animation.*

Ví dụ sau đây mô tả việc thay đổi thuộc tính độ trong suốt của một đối tượng bằng cách thiết lập thuộc tính alpha, tăng dần giá trị từ 0 đến 1 trong khoảng thời gian là 10 giây.

- Tạo file `alpha_anim.xml` trong thư mục `res/animator`:

```
<?xml version="1.0" encoding="utf-8"?>
<objectAnimator
xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="10000"
    android:propertyName="alpha"
    android:valueFrom="0.0"
    android:valueTo="1.0" >
</objectAnimator>
```

- Tham chiếu và sử dụng trong Java code:

```
TextView txtHello = (TextView) findViewById(R.id.txtHello);
ObjectAnimator animOb = (ObjectAnimator)
AnimatorInflater.loadAnimator(getApplicationContext(),

    R.animator.alpha_anim);
animOb.setTarget(txtHello);
animOb.start();
```

## 2. VIEW ANIMATION

### 2.1. Tween Animation

- Mỗi chuyển hoạt được chứa trong tập tin XML trong thư mục res/anim với các hiệu ứng sau: alpha (ẩn hiện mờ), scale (cân chỉnh tỉ lệ), translate (chuyển động) và rotate (quay góc). Mỗi hiệu ứng bao gồm các cặp thuộc tính và giá trị như bảng sau:

Hiệu ứng	Thuộc tính	Giá trị
Alpha	fromAlpha/toAlpha	Từ 0 đến 1 (số thực)
Scale	fromXScale/toXScale	Từ 0 đến 1 (số thực)
	fromYScale/toYScale	Từ 0 đến 1 (số thực)
	pivotX/pivotY	Tỉ lệ chiều ngang và chiều dọc từ 0% đến 100%

Translate	fromX/toX	Từ 0 đến 1 (số thực)
	fromY/toY	Từ 0 đến 1 (số thực)
Rotate	fromDegrees/toDegrees	Từ 0 đến 360 (số thực)
	pivotX/pivotY	Tỉ lệ chiều ngang và chiều dọc từ 0% đến 100%

- Chúng ta có thể sử dụng cặp thẻ set để thiết lập nhiều hiệu ứng với nhau. Khi đó, các hiệu ứng sẽ được bổ sung các một số thuộc tính để chỉnh sửa thời điểm và cách thức mỗi hiệu ứng sẽ thực thi. Danh sách các thuộc tính như sau:
  - *duration* – thời gian thực thi tất cả hiệu ứng, tính bằng mili giây.
  - *startOffset* – độ trễ trước khi hiệu ứng bắt đầu.
  - *fillBeforetrue* – cho phép biến đổi các hiệu ứng trước khi thực thi chuyển hoạt.
  - *fillAftertrue* – cho phép biến đổi các hiệu ứng sau khi kết thúc chuyển hoạt.
  - *interpolator* – thiết lập tốc độ của hiệu ứng thay đổi theo thời gian, để sử dụng thuộc tính này ta sẽ tham chiếu đến tài nguyên chuyển hoạt của hệ thống android:anim/interpolatorName.
- Nếu không thiết lập thuộc tính startOffset, tất cả hiệu ứng sẽ cùng thực thi đồng bộ với nhau.

Ví dụ sau sẽ mô tả chuyển hoạt quay đổi tượng một góc 360 độ, rung và mờ dần:

- Tạo file shrink\_anim.xml trong thư mục res/anim:

```
<?xml version="1.0" encoding="utf-8"?>
<set android:interpolator="@android:anim/accelerate_interpolator"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <rotate
        android:fromDegrees="0"
        android:toDegrees="360"
        android:pivotX="50%"
        android:pivotY="0%"
        android:startOffset="500"
        android:duration="1000"/>

    <scale
        android:fromXScale="1.0"
        android:toXScale="0.0"
        android:fromYScale="1.0"
        android:toYScale="0.0"
```

```
        android:pivotX="50%"
        android:pivotY="50%"
        android:startOffset="500"
        android:duration="500" />

    <alpha
        android:fromAlpha="1.0"
        android:toAlpha="0.0"
        android:startOffset="500"
        android:duration="500" />
</set>
```

- Tham chiếu và sử dụng trong Java Code:

```
AnimationSet anim_shrink = (AnimationSet)
AnimationUtils.loadAnimation(getApplicationContext(),
R.anim.shrink_anim);
txtHello.startAnimation(anim_shrink);
```

## 2.2. Frame Animation

- Frame animation là sự kiện diễn ra một chuỗi các chuyển hoạt được lồng ghép bởi nhiều hình ảnh khác nhau. Mỗi hình ảnh sẽ được thiết lập một thời gian để thực hiện. Do các chuyển hoạt liên quan đến tài nguyên dạng Drawable nên các tập tin Frame-by-Frame Animation sẽ được lưu trong thư mục tài nguyên res/drawable và tên tập tin sẽ được dùng như định dạng để tham chiếu sử dụng.
- Có thể tạo frame animation trong thư mục res/drawable vì frame animation cũng được xem như một dạng hình ảnh.

Ví dụ sau minh họa về chuyển hoạt đơn giản của các hình ảnh, mỗi hình ảnh sẽ được thể hiện trong một phần tư giây.

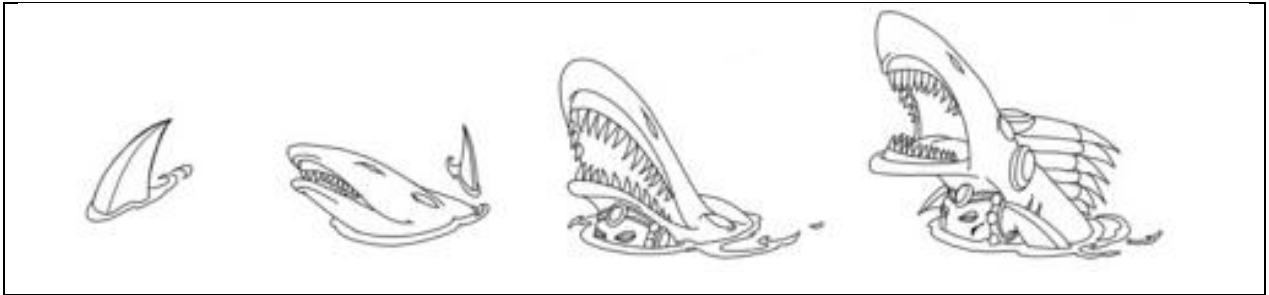
- Tạo file `fizz_utt_animation.xml` trong thư mục res/drawable:

```
<animation-list
xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:drawable="@drawable/shark1"
        android:duration="250"></item>
    <item
        android:drawable="@drawable/shark2"
```

```
android:duration="250"></item>
    <item                                android:drawable="@drawable/shark3"
android:duration="250"></item>
    <item                                android:drawable="@drawable/shark4"
android:duration="250"></item>
    <item                                android:drawable="@drawable/shark5"
android:duration="250"></item>
    <item                                android:drawable="@drawable/shark6"
android:duration="250"></item>
    <item                                android:drawable="@drawable/shark7"
android:duration="250"></item>
    <item                                android:drawable="@drawable/shark8"
android:duration="250"></item>
    <item                                android:drawable="@drawable/shark9"
android:duration="250"></item>
</animation-list>
```

- Tham chiếu và sử dụng trong Java Code:

```
ImageView imgFizz = (ImageView) findViewById(R.id.imgFizz);
imgFizz.setBackgroundResource(R.drawable.fizz_utl_animation);
AnimationDrawable animFizz = (AnimationDrawable)
imgFizz.getBackground();
animFizz.start();
```



Hình 1.2. Ví dụ mô tả Frame Animation.

### 3. DRAWABLE ANIMATION

- Thư mục tài nguyên: res/drawable/filename.xml.
  - o Khai báo các thuộc tính trong XML:  
AnimationDrawable - <animation-list>

```
<animation-list
xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot=["true" | "false"] >
    <item
android:drawable="@[package:]drawabledrawable_resource_name"
        android:duration="integer" />
</animation-list>
```

- Ví dụ khai báo Drawable Animation:  
exam\_drawable\_anim.xml:

```
<animation-list
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot="false">
    <item android:drawable="@drawable/plane01"
        android:duration="200" />
    <item android:drawable="@drawable/plane02"
        android:duration="200" />
    <item android:drawable="@drawable/plane01"
        android:duration="200" />
</animation-list>
```

- Tham chiếu và sử dụng trong Java code:
  - o Thực hiện truy xuất đối tượng AnimationDrawable từ đối tượng View đang hiển thị Drawable, bằng phương thức `getBackgroundResource()`.
  - o Gọi phương thức `start()` từ đối tượng AnimationDrawable.

Ví dụ tham chiếu AnimationDrawable:

```
ImageView planeView = (ImageView) findViewById(R.id.plane);
planeView.setBackgroundResource(R.drawable.exam_drawable_anim);
AnimationDrawable plane =
(AnimationDrawable)planeView.getBackground();
plane.start();
```