Android動畫

程式碼：<https://github.com/stone0705/AndroidAnimatorDemo>

動畫類型分類：

常用的主要有：

1.Tween Animation:

功能較少x﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽，普通的漸變動畫。

2.Frame Animation:  
透過圖片的不斷改變產生動畫，類似GIF。

3.Property Animation:

功能較多，可以完成較為複雜的動畫。

以下主要介紹Property Animation:

Property Animation主要使用兩種方法完成：

ValueAnimator

優點：並不要求一定要有set get方法，靈活性較高，需實做update。

缺點：程式碼稍長

ObjectAnimator

優點：程式碼較短，使用簡單，為ValueAnimator子類別不需實做update。

缺點：想要產生動畫的屬性必須有get set方法(可以自己實作)，靈活性略低。

兩種Animator都可以設定想要的Interpolator，Interpolator用於選擇動畫的速度曲線，Interpolator用到的參數是fraction

fraction是 t / duration 當前時間除以總時間，

fraction介於0跟1，0為開始1為結束，類似動畫的進度，以下用x表示。

Interpolator分為五種：

LinearInterpolator:

f(x) = x

AccelerateDecelerateInterpolator:

先加速後減速

f(x) = (Cos(x+1)\*PI / 2) – 0.5

AccelerateInterpolator :

預設值為等加速度運動

f(x) = x^2

微分一次為V = 2x

微分二次為 A = 2

表示為加速度為2的等加速度運動

可添加參數改變X的次方變成變加速度運動

CycleInterpolator:

是一個sin函數，會依照輸入數字重複運動

DecelerateInterpolator:

預設為等減速度運動

f(x) = 1 – (1-x) \*(1-x)

也可以自己實作想要的Interpolator來完成想要的物理公式，﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽。

當我們希望動畫改變的屬性不只一個的時候我們可以使用AnimatorSet來完成。

AnimatorSet可以自由選擇動畫播放順序，是否同時播放，延遲時間等。

ObjectAnimator使用上較為簡單

ObjectAnimator.ofFloat(view1,"ScaleX",0.6f,0f,1f).setDuration(200).start();

上面這行表示view1要改變ScaleX從0.6f到0f再到1f，持續時間為200毫秒。

要注意的地方有ScaleX必須要有get跟set的方法，後面的數字可以有很多個，可以表現出比較複雜的動畫。

ValueAnimator需要實作update才能表現出動畫效果

ValueAnimator anim = ValueAnimator.ofFloat(0.6f,0f,1f);

anim.setTarget(view);

anim.setDuration(200);

從上面可以發現到你選擇了view跟設定了改變的數字跟持續時間，但是沒有選擇要改變的屬性，所以你還要addUpdateListener來改變屬性。

ValueAnimator還可以設定TypeEvaluator，正常我們都使用ofFloat等等之類的來設定值，但是如果我們希望自己做一個class來設定如何得值的時候，可以使用TypeEvaluator來設定。

除此之外你可以幫動畫加上listener用以在動畫開始或結束或取消等等的時間點產生其他效果。﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽時間拆開來併用:。

DEMO程式

1.MainActivity畫面

主畫面按下Action會有動畫效果，兩邊按鈕動畫呈現一樣，但是Interpolation是使用ObjectAnimator完成，listener按鈕是使用ValueAnimator完成。

按下Interpolation按鈕會到Interpolation的畫面我實作了﷽﷽﷽﷽﷽﷽﷽﷽﷽ation﷽﷽

按下listener的按鈕會到listener的畫面我實作了﷽﷽﷽﷽﷽﷽﷽﷽﷽ation﷽﷽。

2.InterpolationActivity畫面

在Interpolation的畫面，底下有5個按鈕。

1.custom:

左邊的按鈕我使用我自己實作的Interpolation，基本上設定是跟AccelerateInterpolator一樣的值。

右邊的按鈕我使用linearInterpolation，但是我實作了TypeEvaluator，因此雖然是使用linear的速度曲線，我還是能產生加速度的效果。

2.ADe

左邊的按鈕使用AccelerateDecelerateInterpolator，產生先加速後減速的效果。

右邊的按鈕使用LinearInterpolator，作為對照組。

你可以發現到達底部時間是一樣的。

3.A

左邊使用AccelerateInterpolator，為加速效果。

右邊一樣是LinearInterpolator。

4.Cycle

左邊使用CycleInterpolator，我設定的數字是5，所以會產生5次repeat。

右邊一樣是LinearInterpolator。

5.De

左邊使用DecelerateInterpolator，為減速效果。

右邊一樣使用LinearInterpolator。

同樣的按下XML按鈕會到XML的頁面，按下animatorSet會到animatorSet的頁面。

3.AnimationSetActivity畫面

這個頁面將會展現出動畫組合的效果。

當你點下noTogether，你可以看到10個動畫分開來照順序表現的效果。

當你點下playtogether，你可以得到組合動畫的效果。

4.xmlAnemateActivity畫面

在這裏，動畫是寫在xml檔裡面，由java來讀取xml檔產生動畫。

按下xmlSet可以得到由xml做成的組合動畫效果。

按下xml則是單一效果。

5.listenTestActivity畫面

在這裏，呈現動畫listener的效果。

按下onStart的按鈕，在動畫開始的時候會出現一個toast。

按下onFinish的按鈕，動畫結束時會出現一個toast。

按下floatButton按鈕會到FloatButton的頁面。

6.FloatActionButtonActivity畫面

在這個頁面主要展現了動畫的一些應用方式，我把動畫的持續時間拆開來並且使用不同的Interpolation表現出快慢感覺。