

Module V

Android Animation

- Drawable Animation (Frame Animation)
 - View Animation (Tween Animation)
 - Draw Canvas

Drawable Animation

- ▶ 利用ImageView元件載入res/drawable圖形檔案呈現動畫效果
- ▶ 搭配AnimationDrawable類別來控制動畫播放與停止
- ▶ 也可以利用xml檔做相關設定
 - 使用<animation-list>標籤
 - android:oneshot：是否要重覆播放 (true / false)
 - android:drawable：設定drawable影像
 - android:duration：播放時間 (單位為毫秒)

範例：DrawableAniEx

View Animation (1 / 4)

- ▶ 稱為Tween Animation(補間動畫)，指的是填補兩個圖形之間的變化
- ▶ 補間動畫的類別
 - TranslateAnimation(位移)
 - ScaleAnimation(縮放)
 - RotateAnimation(旋轉)
 - AlphaAnimation(透明)
- ▶ 相關XML屬性請參考下頁：

範例：ViewAniEx

View Animation (2/4)

▶ Animation類別的XML屬性

屬性名稱	說明	屬性值
android:duration	動畫播放時間，單位為毫秒	整數，不可為負值
android:interpolator	動畫運行效果。設定位移補間動畫後，還要指定整個位移過程的效果	linear_interpolator (線性) accelerate_interpolator (加速) decelerate_interpolator (減速) anticipate_interpolator (先退後進) overshoot_interpolator (衝過頭) bounce_interpolator (反彈) cycle_interpolator (以曲線方式加快重覆次數)
android:repeatCount	動畫重複播放次數	整數。-1 代表無限重複播放 預設為0 (不重複播放)
android:repeatMode	動畫重複播放模式	restart：重新播放 (預設) reverse：反向播放
android:startOffset	設定主動畫開始後多久才執行此動畫，單位為毫秒，要播放多個動畫時，可以使用此屬性來指定各個動畫播放的相對時間	整數，不可為負數

View Animation (3/4)

▶ 通用的XML屬性

屬性名稱	說明	屬性值
android:pivotX android:pivotY	元件的哪個位置會發生補間動畫， 指定該位置的x軸/y軸座標 原點(0,0)在左上角，向右x軸值增加，向下y軸值增加	浮點數 50%代表在該元件的中央位置 50代表在父元件的50%位置，也就是父元件的中央位置

▶ 與位移相關的XML屬性

屬性名稱	說明	屬性值
android:fromXDelta android:fromYDelta	位移開始時元件的x軸/y軸座標	浮點數
android:toXDelta android:toYDelta	位移結束時元件的x軸/y軸座標	

View Animation (4/4)

▶ 與縮放有關的XML屬性

屬性名稱	說明	屬性值
android:fromXScale android:fromYScale	縮放開始時元件的水平/垂直尺寸	浮點數 1.0代表維持原來尺寸 < 1.0代表縮小 > 1.0代表放大
android:toXScale android:toYScale	縮放結束時元件的水平/垂直尺寸	

▶ 與旋轉相關的XML屬性

屬性名稱	說明	屬性值
android:fromDegrees	旋轉開始時元件的角度	浮點數 < 0.0代表逆時針 > 0.0代表順時針
android:toDegrees	旋轉結束時元件的角度	

▶ 與透明化有關的XML屬性

屬性名稱	說明	屬性值
android:fromAlpha	動畫開始時元件的透明度	介於0.0 ~ 1.0之間的浮點數 0.0代表完全透明 1.0代表完全不透明
android:toAlpha	動畫結束時元件的透明度	

Draw Canvas

- ▶ Android提供2D繪圖功能，開發者所需套件在android.graphics裡
- ▶ 想在畫面上畫出自訂的圖案，就必須自訂View元件
 - 繼承View類別並改寫onDraw()方法
 - 透過layout檔案作版面配置必須建立帶有AttributeSet參數的建構子，例如RectBar(Context, AttributeSet)
 - 再呼叫View.invalidate()重繪元件
- ▶ 3D繪圖請參考OpenGL ES 3.0相關網路資源或書籍