

# Animacións

## Sumario

- 1 Introducción
- 2 Caso práctico
  - ◆ 2.1 O XML do Layout
  - ◆ 2.2 Animación Tween
  - ◆ 2.3 Animación frame
  - ◆ 2.4 O código Java

## Introdución

- Android ten 2 mecanismos para crear animacións
  - ◆ **Animación de propiedades:** Crea unha animación modificando os valores das propiedades dos obxectos por un período de tempo.
- - ◆ **Animación de vistas:** que se divide en dous tipos:
    - ◇ **Animacións Tween /Animation':** Crea unha animación de rotación, translación, zoom, transparencia (alpha) sobre calquera vista.
    - ◇ **Animación Frame /AnimationDrawable:** crea unha animación amosando unha secuencia de imaxes. Crease un obxecto da clase: .
- Usaremos estes dous últimos tipos.
- Referencias:
  - ◆ <http://developer.android.com/guide/topics/resources/animation-resource.html>
  - ◆ <http://developer.android.com/guide/topics/graphics/drawable-animation.html>
  - ◆ <http://developer.android.com/reference/android/view/animation/Animation.html>
  - ◆ <http://developer.android.com/reference/android/graphics/drawable/AnimationDrawable.html>

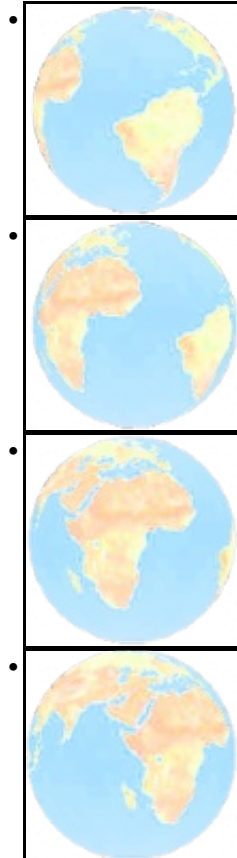
## Caso práctico

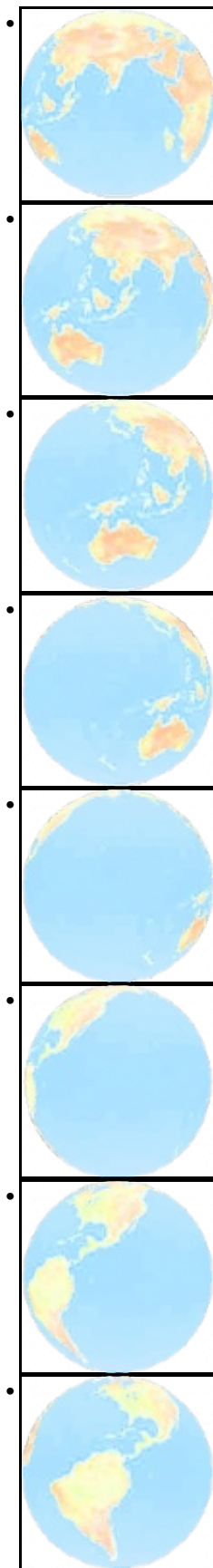
- Comezamos creando unha actividade: **U3\_30\_Animacion**
- A imaxe amosa as animacións que se crearán:



- Ao iniciar a aplicación, o fondo é branco e no espazo de tempo vaise tornando en vermello.
- A imaxe da terra xirando como secuencia das seguintes imaxes.

• Frames da animación





- No seguinte ficheiro pódense descargara as imaxes e colocalas en **/res/drawable** do proxecto.

Media:Drawable\_terra.zip

## O XML do Layout

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/layout_terra"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F00"
    android:gravity="center"
    android:orientation="vertical" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Animación"
        android:textColor="#FFF"
        android:textSize="22sp" />

    <ImageView
        android:id="@+id/imgvwTerra"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:contentDescription="Animación da terra" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onPararIniciarClick"
        android:text="Parar / Iniciar"
        android:textColor="#FFF" />

</LinearLayout>
```

- **Liñas 17-21:** Vista onde se vai amosar a animación da terra

## Animación Tween

- Unha animación Tween define os seus efectos: transicións, rotacións, fading, movement, zoom sobre un gráfico nun ficheiro XML
- Localización do ficheiro: **/res/anim/filename.xml**
- O nome do ficheiro será usado como ID do recurso.
- Neste caso o recurso chámase: **/res/anim/animacion\_aparece\_fundido.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="true"
    >
    <alpha
        android:fromAlpha="0.0"
        android:toAlpha="1.0"
        android:duration="6000"
    />

</set>
```

- Elementos do ficheiro:
  - ◆ **set:** É un contedor para albergar outros elementos de animación (<alpha>, <scale>, <translate>, <rotate>)
  - ◆ **alpha:** Animación fade-in/fade-out (fundido). Usa a transparencia alpha para indicar dende que nivel de transparencia se parte e ata cal se chega e canto tempo debe durar o proceso.
  - ◆ No noso caso partimos dunha transparencia total (0.0) a unha opacidade total (1.0). O proceso de fundido dura 6 segundos.

- Ao iniciar a aplicación o Layout aparece branco (transparente) e vaíse fundindo até chegar a vermello (Opaco) que é o color co que se definiu o Layout.
- Logo no código aplícase a animación e cargase.

Para consultar máis elementos dunha animación Tween:

- **Referencias:**

- ♦ <http://developer.android.com/guide/topics/resources/animation-resource.html#Tween>

## Animación frame

- A animación defínese nun XML que amosa a secuencia de imaxes a amosar nunha orde (Como unha película)
- Localización do ficheiro: **/res/drawable/filename.xml**
- O nome do ficheiro será usado como ID do recurso.

- Neste caso o ID do recurso é: **/res/drawable/animacion\_terra.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot="false"
    >
    <item android:drawable="@drawable/terra1" android:duration="250"/>
    <item android:drawable="@drawable/terra2" android:duration="250"/>
    <item android:drawable="@drawable/terra3" android:duration="250"/>
    <item android:drawable="@drawable/terra4" android:duration="250"/>
    <item android:drawable="@drawable/terra5" android:duration="250"/>
    <item android:drawable="@drawable/terra6" android:duration="250"/>
    <item android:drawable="@drawable/terra7" android:duration="250"/>
    <item android:drawable="@drawable/terra8" android:duration="250"/>
    <item android:drawable="@drawable/terra9" android:duration="250"/>
    <item android:drawable="@drawable/terra10" android:duration="250"/>
    <item android:drawable="@drawable/terra11" android:duration="250"/>
    <item android:drawable="@drawable/terra12" android:duration="250"/>
</animation-list>
```

- **Elementos:**

- ♦ **<animation-list>**: é o elemento raíz e contén elementos <item>
  - ◊ **android:oneshot**: Booleano. "true" para realizar a animación unha soa vez; "false" para estar nun bucle infinito.
- ♦ **<item>**: Indica a secuencia de imaxe a cargar e a duración en milisegundos.

- **Referencias:**

- ♦ <http://developer.android.com/guide/topics/resources/animation-resource.html#Frame>

- A animación será cargada no código.

## O código Java

```
package com.example.u3_30_animacion;

import android.app.Activity;
import android.graphics.drawable.AnimationDrawable;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;

public class U3_30_Animacion extends Activity {
    AnimationDrawable ad;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u3_30__animacion);
    }
}
```

```

cargarAnimacions();
}

private void cargarAnimacions() {
    //Tweek
    View pai = (View) findViewById(R.id.layout_terra);
    Animation fundido = AnimationUtils.loadAnimation(this, R.anim.animacion_aparece_fundido);
    pai.startAnimation(fundido);

    //Frame
    ImageView img = (ImageView) findViewById(R.id.imgvwTerra);
    img.setBackgroundResource(R.drawable.animacion_terra);

    ad = (AnimationDrawable) img.getBackground();
    ad.start();

}

public void onPararIniciarClick(View v) {
    if (ad.isRunning())
        ad.stop();
    else
        ad.start();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.u3_30__animacion, menu);
    return true;
}

}

```

- **Liñas 23-26:** Cargamos os efectos da animación e aplicamos ese efecto (fundido) ao Layout.
- **Liñas 29-34:** Cargamos a animación na vista ImageView (img) da activity e iniciamos a animación
- **Liña 40:**Controlar se a animación esta ou non en marcha.