Creando una animación sencilla en Android

Hay muchas formas de crear animaciones en Android, algunas más básicas y por tanto, sencillas y otras mucho más elaboradas y complejas. La que aquí veremos es una simple animación que constará de varias imágenes secuenciales que al mostrarse una tras de otra crean el efecto de animación.

Esto nos servirá para indicar al usuario que se está realizando un proceso en segundo plano, ya sea descargar una imagen, almacenar algo en la base de datos, recogiendo información de internet, etc..

Vamos a crear un ejemplo práctico de animación y al final pondré un pequeño video que muestra el funcionamiento final.

Pues bien, lo primero es crear un proyecto al que para el ejemplo he puesto de nombre "Hello Animacion Simple", una vez creado y configurado el proyecto empezaremos por crear la animación, necesitaremos 5 imágenes

Con esas imágenes dentro de la carpeta "drawable", vamos a crear el fichero animacion.xml (también dentro de "drawable"), que contendrá lo siguiente:

```
<?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/saving_image_white"
android:duration="150" />
    <item android:drawable="@drawable/saving_image_white_1"
android:duration="150" />
    <item android:drawable="@drawable/saving_image_white_2"
android:duration="150" />
    <item android:drawable="@drawable/saving_image_white_3"
android:duration="150" />
    <item android:drawable="@drawable/saving_image_white_4"
android:duration="150" />
</animation-list>
```

Lo que crea una animación con cada una de las cinco imágenes por el orden deseado e indica que se reproducirá continuamente (android:oneshot="false", si estuviera a true sólo se reproduciría una única vez), además de una duración de cada imagen de 150 milisegundos, podemos aumentarlo a reducirlo para hacer que la animación sea más lenta o más rápida.

Con esto ya tenemos nuestra animación preparada, así de sencillo, sólo quedaría inicializarla en el código principal de la actividad.

2.- Creando el Layout principal de nuestra Activity

En la carpeta "res/layout" editamos el fichero activity_main.xml que se habrá creado al hacer nuestro nuevo proyecto (a no ser que le hayamos cambiado el nombre, que tendrá el nombre indicado) y le añadimos dos botones, uno para iniciar la animación y otro para detenerla y un ImageView que será el contenedor para nuestra animación

```
<LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingLeft="@dimen/activity_horizontal_margin"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin"
        android:paddingBottom="@dimen/activity_vertical_margin"
        tools:context=".MainActivity"
        android:background="@android:color/darker_gray"
        android:orientation="vertical">

    <Button
            android:id="@+id/btn_iniciar_animacion"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="@string/iniciar"/>

    <ImageView
            android:id="@+id/iv_animacion"
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:layout_gravity="center"
            android:gravity="center"/>
```

```
    <Button
            android:id="@+id/btn_detener_animacion"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="@string/detener"/>
</LinearLayout>
```

3.- Lógica principal de la aplicación

Aquí básicamente pondremos los Listener de los botones con su funcionalidad y asignaremos la animación al ImageView. Cómo ya hemos dicho antes, el botón btn_iniciar comenzará a reproducir la animación al ser pulsado, mientras que el botón btn_detener será el encargado de detenerla:

```
public class MainActivity extends Activity implements View.OnClickListener{
    private Button btnIniciar, btnDetener;
    private ImageView ivAnimacion;
    private AnimationDrawable savingAnimation;

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

        btnIniciar = (Button)findViewById(R.id.btn_iniciar_animacion);
        btnDetener = (Button)findViewById(R.id.btn_detener_animacion);

        btnIniciar.setOnClickListener(this);
        btnDetener.setOnClickListener(this);

        ivAnimacion = (ImageView)findViewById(R.id.iv_animacion);
        ivAnimacion.setBackgroundResource(R.drawable.animacion);
        savingAnimation = (AnimationDrawable)ivAnimacion.getBackground();
    }


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

    @Override
    public void onClick(View v) {
        switch(v.getId()){
            case R.id.btn_iniciar_animacion:
                savingAnimation.start();
                break;
            case R.id.btn_detener_animacion:
                savingAnimation.stop();
                break;
        }
    }
}
```

Lo primero es crear las variables necesarias, una para cada botón y la del ImageView, después en el onCreate, inicializamos cada una de las variables y con:

```
        ivAnimacion.setBackgroundResource(R.drawable.animacion);
        savingAnimation = (AnimationDrawable)ivAnimacion.getBackground();
```

Estamos poniéndole al ImageView nuestra animación como recurso y recogiéndolo en savingAnimation como AnimationDrawable (la animación en sí, que contiene todos los métodos para operar con ella).

Y por último el onClick identifica cuál de los botones ha sido pulsado e inicia la animación si se pulsa el de iniciar o la detiene si se pulsa el botón de detener.

De esta forma tan sencilla tenemos una animación bastante decente como para poder indicar al usuario que se está realizando una tarea en segundo plano. Como ejemplo, esto mismo puede usarse con un AsyncTask de manera que iniciando la animación antes de ejecutarlo y deteniéndola cuando termine él sólo mostrará la animación mientras la tarea en segundo plano se esté realizando.