

Animaciones de botones y sonidos.

En este caso, he decidido usar el proyecto de la reescalación de una imagen según lo introducido en un cuadro de texto para implementar las funciones pedidas.

Para esta práctica, añadí dos botones que permiten cambiar la imagen que aparece en el ImageView que cambia de escala en la práctica original.

Al layout original, le he añadido los dos botones dentro de un LinearLayout al final del contenido que ya había, y un elemento space entre ellos para la distancia, aun que también lo podría haber hecho con un atributo padding left.



```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center">

    <Button
        android:id="@+id/botonIzquierdo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Cambiar a imagen1"
        android:onClick="cambiarImagen1"
        android:onHover="aumentarTamano"
        android:onPress="animacionApretar"
        android:soundOnClick="@raw/audio1"/>

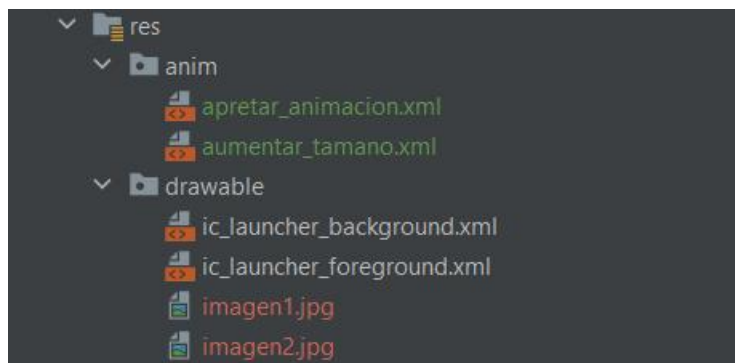
    <Space
        android:layout_width="16dp"
        android:layout_height="1dp"/>

    <Button
        android:id="@+id/botonDerecho"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Cambiar a imagen2"
        android:onClick="cambiarImagen2"
        android:onHover="aumentarTamano"
        android:onPress="animacionApretar"
        android:soundOnClick="@raw/audio2"/>

</LinearLayout>

```

Además, los botones tienen id “botonIzquierdo” y “botonDerecho” respectivamente.



En cuanto a archivos, he añadido dos imágenes (siendo la imagen1 la que aparece por defecto en el layout) y dos animaciones, además de un archivo de audio.



La clase MainActivity ha sido modificada para añadir los nuevos botones, y para crear el MediaPlayer en el método onCreate, pasándole el archivo.mp3 alojado en el directorio raw.

```
1 usage  Aaron Castro *
public class MainActivity extends Activity {
    2 usages
    private EditText editText;
    6 usages
    private ImageView imageView;
    1 usage
    private Button botonIzquierdo;
    1 usage
    private Button botonDerecho;

    8 usages
    private MediaPlayer mediaPlayer;

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

        editText = findViewById(R.id.editText);
        imageView = findViewById(R.id.imageView);
        botonIzquierdo = findViewById(R.id.botonIzquierdo);
        botonDerecho = findViewById(R.id.botonDerecho);

        mediaPlayer = MediaPlayer.create(context: this, R.raw.audio1);
    }
}
```

Estos son los métodos para cambiar la imagen. Estos métodos han sido declarados en la sección onClick de los atributos de los botones.

```
1 usage  new *
public void cambiarImagen1(View view) {
    imageView.setImageResource(R.drawable.imagen1);
    reproducirSonido(R.raw.audio1);
}

1 usage  new *
public void cambiarImagen2(View view) {
    imageView.setImageResource(R.drawable.imagen2);
    reproducirSonido(R.raw.audio1);
}

onClick  cambiarImagen1  ▾  }
```

Para añadir la animación, he introducido en el metodo del onClick una llamada al método de la animación:

```

1 usage new *
public void cambiarImagen1(View view) {
    imageView.setImageResource(R.drawable.imagen1);
    animacionApretar(view);
    reproducirSonido(R.raw.audio1);
}

1 usage new *
public void cambiarImagen2(View view) {
    imageView.setImageResource(R.drawable.imagen2);
    animacionApretar(view);
    reproducirSonido(R.raw.audio1);
}

2 usages new *
public void animacionApretar(View view) {
    Animation anim = AnimationUtils.loadAnimation(context: this, R.anim.apretar_animacion);
    view.startAnimation(anim);
}

```

```

<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true">
    <scale
        android:duration="200"
        android:fromXScale="1.0"
        android:fromYScale="1.0"
        android:toXScale="0.8"
        android:toYScale="0.8"/>
</set>

```

Esta, coge el tamaño original del botón y lo encoje un 20% durante 200ms.

```

2 usages new *
private void reproducirSonido(int rawId) {
    // Reproducir el sonido asociado al botón
    if (mediaPlayer != null) {
        mediaPlayer.release();
    }

    mediaPlayer = MediaPlayer.create( context: this, rawId);
    mediaPlayer.start();
}

new *
@Override
protected void onDestroy() {
    // Liberar recursos del MediaPlayer cuando la actividad se destruye
    if (mediaPlayer != null) {
        mediaPlayer.release();
        mediaPlayer = null;
    }
    super.onDestroy();
}

```

Por último, estos son los métodos que permiten reproducir el sonido y destruir el mediaPlayer según se dio en clase.