Chronometer. Cliclo de vida I (finish())

Sumario

Introdución

- Un control **Chronometer** implementa en Android un cronómetro simple.
- Iníciase con **start()**
- Párase con stop(). Pero ... el segue contando por detrás. Isto é, se se executa de novo start() sen parámetros porá o cronómetro no mesmo tempo que se non se parase.
- Cando se chama a **start()**, o control colle o tempo que leva acceso o móbil (**elapsedRealtime()**) como tempo base e vai contando dende aí.
- Se non se lle dá un tempo base el colle o tempo no cal se chama a **start()**.
- Para establecer un tempo base úsase **setBase()**
- Por defecto amosa o tempo en formato "MM:SS". Pódese usar **setFormat(String)** para cambiar o formato.
- A clase **Chronometer** herda directamente da clase **View**.
- Para establecer os valores de tempos base apoiámonos na clase **SystemClock**.
- **Referencias**:
  - Clase Chronometer: http://developer.android.com/reference/android/widget/Chronometer.html
  - Clase SystemClock: http://developer.android.com/reference/android/os/SystemClock.html

Caso práctico

- Comezamos creando o proxecto **U2_14_Chronometer**
- Creamos unha aplicación que cando ten un cronómetro que hai que iniciar e que pasado un tempo se non se parou a aplicación autodestrúese.

Autodestrucción

Ao iniciar a aplicación o crono está parado. Iniciámolo.

O crono está en funcionamento e se non se para autodestruirase a aplicación en X seg. O valor de partida para a autodestrucción establécese nunha variable no código.

Parouse o crono. Se se preme en Start, é como comezar de novo.

XML do Layout

- Observar como se crea o control Chronometer e os métodos aos que chaman os botóns.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/tv_autodestruccion"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Autodestrucción en: ?? seg" />

    <Chronometer
        android:id="@+id/cronometro"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:textSize="20sp" />

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

        <Button
            android:id="@+id/start"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="onStartClick"
            android:text="Start" />

        <Button
            android:id="@+id/stop"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="onStopClick"
            android:text="Stop" />
    </LinearLayout>

</LinearLayout>
```

Código Java

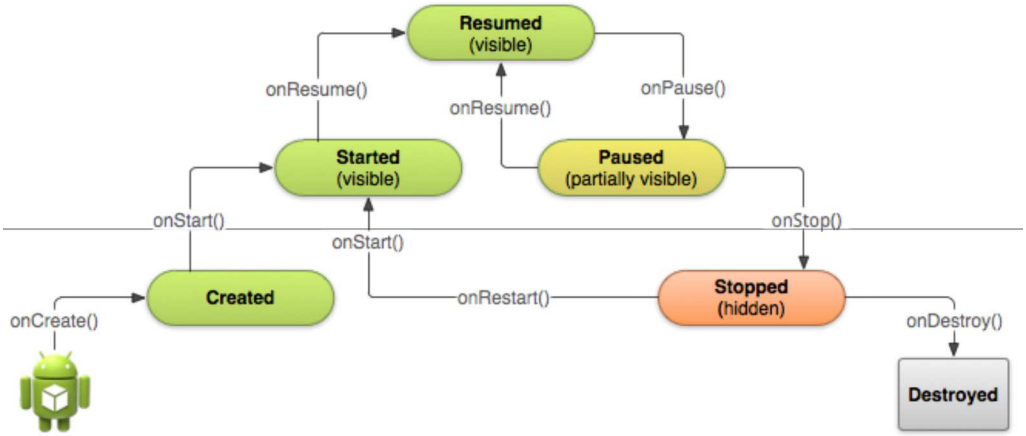
```
package com.example.u2_14_chronometer;

import android.app.Activity;
import android.os.Bundle;
import android.os.SystemClock;
import android.view.Menu;
import android.view.View;
import android.widget.Chronometer;
import android.widget.Chronometer.OnChronometerTickListener;
import android.widget.TextView;

public class U2_14_Chronometer extends Activity {
    Chronometer crono;
    TextView tvAutodestrucion;
    int tempoAutodestrucion;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u2_14__chronometer);
        crono = (Chronometer) findViewById(R.id.cronometro);
        tvAutodestrucion = (TextView) findViewById(R.id.tv_autodestruccion);

        crono.setOnChronometerTickListener(new OnChronometerTickListener() {
            @Override
            public void onChronometerTick(Chronometer chronometer) {
                // TODO Auto-generated method stub

                long tempoPasado = SystemClock.elapsedRealtime()
                        - chronometer.getBase();
                int tempoSeg = (int) tempoPasado / 1000;
                if (tempoSeg == tempoAutodestrucion)
                    finish();

                tvAutodestrucion.setText("Autodestrucción en: "
                        + (tempoAutodestrucion - tempoSeg) + " seg");
            }
        });
    }

    public void onStartClick(View v) {
        tempoAutodestrucion = 20;
        crono.setBase(SystemClock.elapsedRealtime());
        crono.start();
    }

    public void onStopClick(View v) {
        crono.stop();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.u2_14__chronometer, menu);
```

```
55          return true;
56      }
57
58  }
```

- **Liña 42**: Definimos o tempo para a autodestrucción da Activity. Cando se vexan os menús, veremos que este valor podería ser configurado polo usuario que usa a aplicación, como unha opción da mesma.
- **Liña 43**: establécese o tempo base a partir do cal o cronómetro comeza a contar.
  - SystemClock.elapsedRealtime(): devolve o tempo, en milisegundos, que leva aceso o móbil
- **Liña 44**: o cronómetro comeza a contar, non dende cero, senón dende o tempo tomado como base (o tempo que leva acceso o móbil).
- **Liña 48**: párase o cronómetro.
- **Liña 24**: establécese o Listener do cronómetro. Este método vai ser chamado cada vez que o cronómetro cambia de valor.
- **Liñas 29 e 30**: O tempo que pasou dende que se iniciou o cronómetro é a diferencia entre o tempo que leva acceso o dispositivo e o tempo no cal se iniciou o cronómetro.
- **Liña 31**: Tempo en segundos.
- **Liña 33**: Se chegamos ao tempo para autodestrucción hai que matar a Activity. Sería o mesmo que se prememos o botón da botonera "Retroceso ou Back".

Ciclo de vida dunha actividade

- Afondarase máis adiante sobre o ciclo de vida dunha actividade.
- Unha actividade desde que se lanza pasa por moitos estados.
- Cando unha actividade non está en primeira liña da pantalla non está destruída senón que está en estado de Stop (Oculta) esperando na pila a ser chamada, ou se está moi abaixo na pila e se precisan os seus recursos entón o sistema pode destruíla.



- No noso exemplo a actividade destrúese explicitamente se o cronómetro chega a un tempo determinado
- Para iso úsase o método **finish()**.
- Ver liña 33 do código.

-- Ángel D. Fernández González e Carlos Carrión Álvarez -- (2015).

Obtenido de «https://manuais.iessanclemente.net/index.php?title=Chronometer._Cliclo_de_vida_I_(finish())&oldid=57132»

Esta página se editó por última vez el 28 jul 2015 a las 09:09.

El contenido está disponible bajo la licencia Creative Commons: CC-BY-NC-SA, a menos que se indique lo contrario.