TextureView

Компонент **TextureView** находится в разделе **Expert** и служит для вывода видеопотока или сцен OpenGL ES.

По сравнению с **SurfaceView/VideoView** данный компонент обладает большей гибкостью. Он не требует отдельного окна для вывода видео, а работает как обычный **View** - его можно поворачивать, менять размеры и прозрачность, перемещать.

Подготовьте видеофайл формата MP4 и разместите его в папке **assets**. Создадим разметку.

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/rootView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context=".MainActivity">

<TextureView

android:id="@+id/textureView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:layout\_alignParentTop="true" />

</RelativeLayout>

При запуске приложения будет проигрываться видеофайл в циклическом режиме. При касании экрана размер компонента будет меняться, а само видео будет играть дальше.

package ru.alexanderklimov.testapril2015;

import android.content.res.AssetFileDescriptor;

import android.graphics.SurfaceTexture;

import android.media.MediaPlayer;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.util.Log;

import android.view.MotionEvent;

import android.view.Surface;

import android.view.TextureView;

import android.view.View;

import android.widget.RelativeLayout;

import java.io.IOException;

public class MainActivity extends AppCompatActivity implements TextureView.SurfaceTextureListener {

private static final String TAG = MainActivity.class.getName();

private static final String FILE\_NAME = "cat.mp4"; // ваш файл

private MediaPlayer mMediaPlayer;

private TextureView mTextureView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mTextureView = (TextureView) findViewById(R.id.textureView);

mTextureView.setSurfaceTextureListener(this);

RelativeLayout rootView = (RelativeLayout) findViewById(R.id.rootView);

rootView.setOnTouchListener(new View.OnTouchListener() {

@Override

public boolean onTouch(View view, MotionEvent motionEvent) {

switch (motionEvent.getAction()) {

case MotionEvent.ACTION\_UP:

updateTextureViewSize((int) motionEvent.getX(), (int) motionEvent.getY());

break;

}

return true;

}

});

}

private void updateTextureViewSize(int viewWidth, int viewHeight) {

mTextureView.setLayoutParams(new RelativeLayout.LayoutParams(viewWidth, viewHeight));

}

@Override

protected void onDestroy() {

super.onDestroy();

if (mMediaPlayer != null) {

mMediaPlayer.stop();

mMediaPlayer.release();

mMediaPlayer = null;

}

}

// ниже представлены методы интерфейса TextureView.SurfaceTextureListener

@Override

public void onSurfaceTextureAvailable(SurfaceTexture surfaceTexture, int i, int i1) {

Surface surface = new Surface(surfaceTexture);

try {

AssetFileDescriptor afd = getAssets().openFd(FILE\_NAME);

mMediaPlayer = new MediaPlayer();

mMediaPlayer

.setDataSource(afd.getFileDescriptor(), afd.getStartOffset(), afd.getLength());

mMediaPlayer.setSurface(surface);

mMediaPlayer.setLooping(true);

mMediaPlayer.prepareAsync();

mMediaPlayer.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {

@Override

public void onPrepared(MediaPlayer mediaPlayer) {

mediaPlayer.start();

}

});

} catch (IOException e) {

Log.d(TAG, e.getMessage());

}

}

@Override

public void onSurfaceTextureSizeChanged(SurfaceTexture surfaceTexture, int i, int i1) {

}

@Override

public boolean onSurfaceTextureDestroyed(SurfaceTexture surfaceTexture) {

return false;

}

@Override

public void onSurfaceTextureUpdated(SurfaceTexture surfaceTexture) {

}

}