Android graphics Picture

Класс **Picture** записывает вызовы методов рисования в одном месте, а затем их воспроизводит на холсте. По уверению документации, такой способ работает быстрее чем обычные отдельные вызовы графических методов. Но на практике использования данного класса не встречал. То ли разработчики не знают об этом классе, то ли он не даёт никаких преимуществ.

При работе с классом **Picture** нужно включить аппаратное ускорение в манифесте для **application** или **activity**.

android:hardwareAccelerated="true"

Создадим отдельный класс **PictureView**, который послужит нам в качестве тестовой площадки.

package ru.alexanderklimov.picture;

import android.content.Context;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.Paint;

import android.graphics.Path;

import android.graphics.Picture;

import android.view.View;

public class PictureView extends View {

private Paint mPaint;

private Path mPath;

private Picture mPicture;

public PictureView(Context context) {

super(context);

mPaint = new Paint(Paint.ANTI\_ALIAS\_FLAG);

mPicture = new Picture();

// Начинаем запись

Canvas canvas = mPicture.beginRecording(400, 400);

// Сначала рисуем прямоугольник

mPaint.setColor(Color.BLUE);

canvas.drawRect(10, 10, 150, 200, mPaint);

// Затем круг

mPaint.setColor(Color.MAGENTA);

canvas.drawCircle(250, 100, 90, mPaint);

// И ещё фигуру

mPaint.setColor(Color.CYAN);

mPath = new Path();

mPath.moveTo(450, 10);

mPath.lineTo(550, 200);

mPath.lineTo(350, 200);

mPath.close();

canvas.drawPath(mPath, mPaint);

// Конец записи

mPicture.endRecording();

}

@Override

protected void onDraw(Canvas canvas) {

super.onDraw(canvas);

canvas.drawColor(Color.YELLOW);

// Выводим записанные операции

canvas.drawPicture(mPicture);

}

}

Выводим созданный компонент на экран. На всякий случай проверяем включение аппаратного ускорения и включаем его также программно. Некоторые устройства даже после этих шагов не включают ускорение, поэтому проверяем логи.

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

// Включаем программно аппаратное ускорение

getWindow().setFlags(WindowManager.LayoutParams.FLAG\_HARDWARE\_ACCELERATED,

WindowManager.LayoutParams.FLAG\_HARDWARE\_ACCELERATED);

PictureView pictureView = new PictureView(this);

setContentView(new PictureView(this));

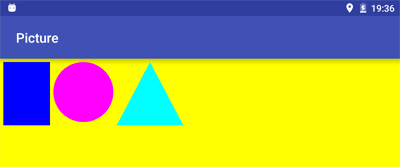
// Проверяем, включено ли ускорение

boolean isHWAccelerated = pictureView.isHardwareAccelerated();

Log.i("accelerator", String.valueOf(isHWAccelerated));

}

Напоминает запись на диктофон - включили запись, а затем его воспроизвели.



Мы выводили последовательность действий через метод **drawPicture()** с одним аргументом. Есть перегруженная версия с двумя аргументами. Во втором аргументе можно указать размер и записанная последовательность будет размещена внутри заданного размера. Добавим код.

@Override

protected void onDraw(Canvas canvas) {

super.onDraw(canvas);

canvas.drawColor(Color.YELLOW);

// Выводим записанные операции

canvas.drawPicture(mPicture);

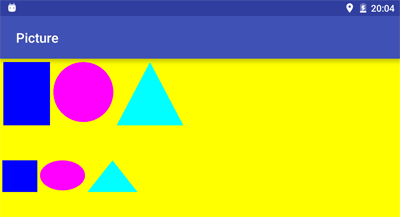
// Смещаемся вниз

canvas.translate(0, 300);

canvas.drawPicture(mPicture, new Rect(0, 0, 300, 200));

}

В результате записанная последовательность будет выведена чуть ниже оригинальной записи, при этом фигуры получатся сплюснутыми.



Р