

13



Graphics



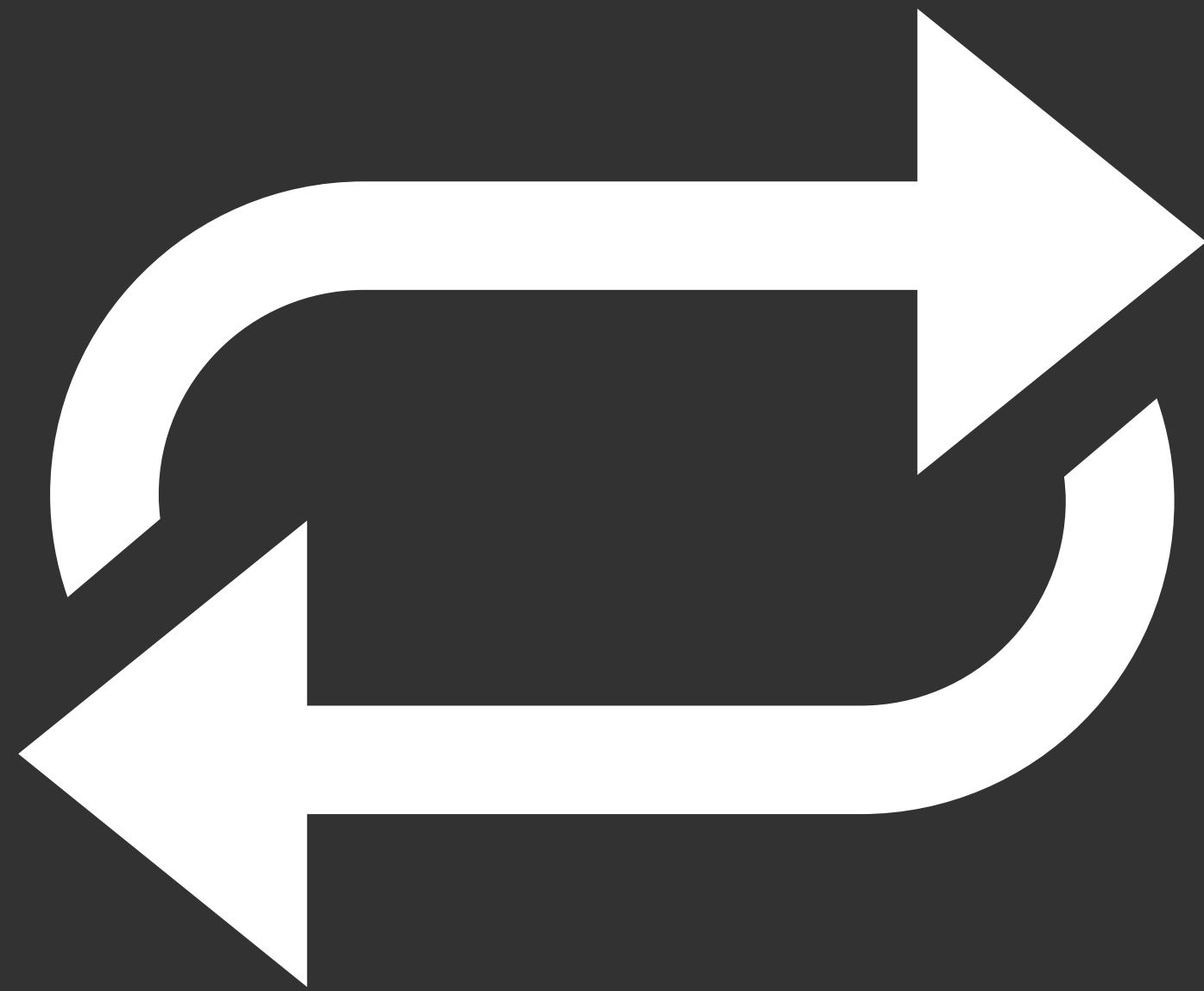
Performance



Chet Haase

Romain Guy





Reordering & merging

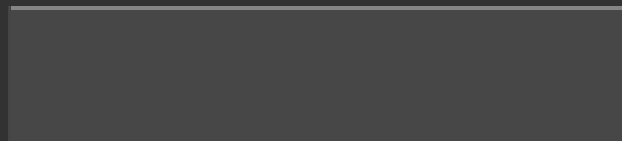
Maximize compatibility

Include metadata

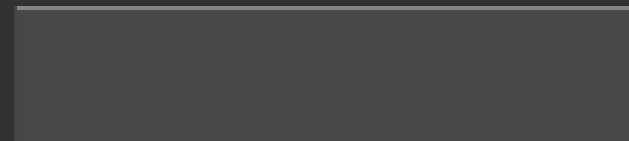
Save

Cancel

Include metadata Maximize compatibility

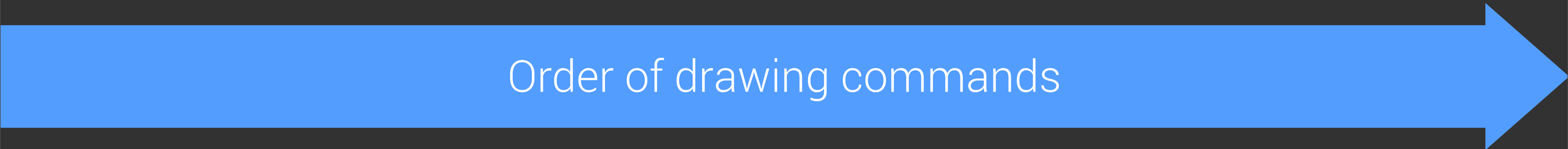


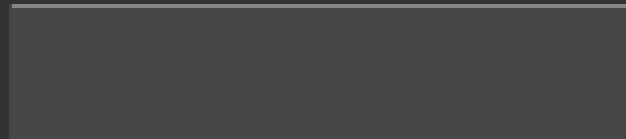
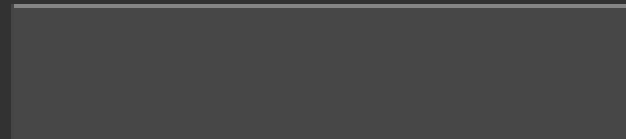
Save



Cancel

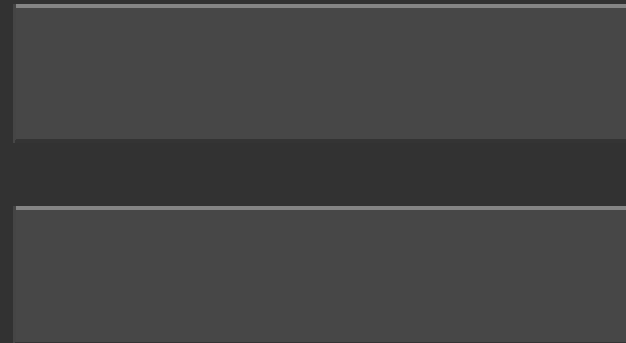
Order of drawing commands





Include metadata Maximize compatibility Save Cancel

1. Re-ordering



Cancel

Include metadata

Maximize compatibility




Save

2. Merging

ABOUT


POSTS

PHOTOS




+156    19

Kevin Coppock Very nice photo! I was just up there on Saturday but I've never...



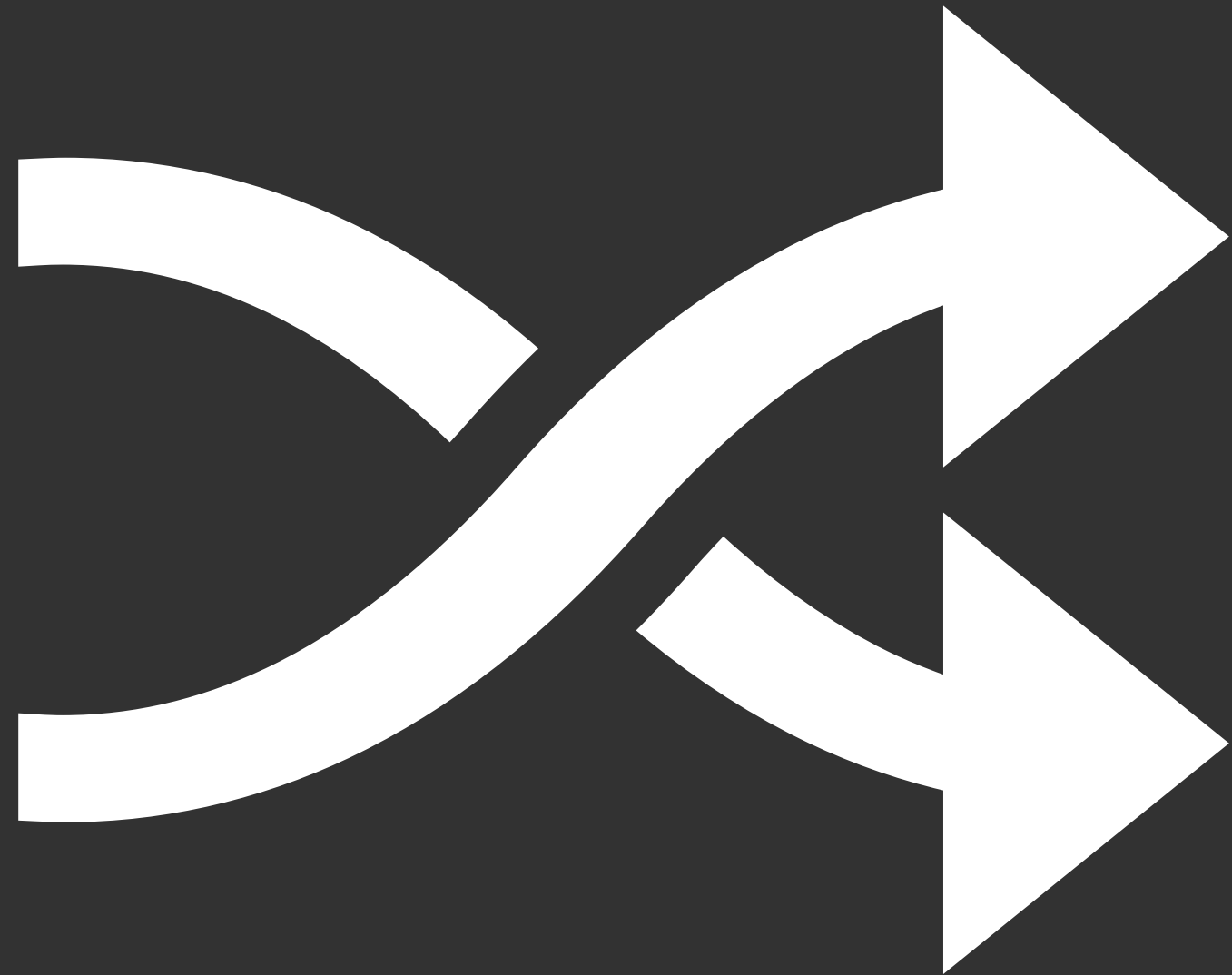
 **Romain...**
February 17 #sunsetphotography

Sunset in Kyoto
Kyoto, Japan

+73    14

Ming Thein Whats the T stop of this thing like?





Multi-threading

Drawing

Drawing

Shadows

Shadows

Shadows

Shadows

Drawing

Shadows

Paths

Shadows

Paths

Shadows

Shadows

ipsum ad amet, con
piscing elit. Sed vitae risus non s
ctum pellentesque at convallis mi. In
agna nisi. Integer lorem odio, blandit
ia malesuada, auctor commodo odio
auris in ipsum leo, egestas imperdie
rabitur adipiscing congue purus, p
ittis sapien faucibus vel. Mauris
s non nulla tristique sed vesti
faucibus. Curabitur vel se
urus. Interdum, nibh sit
h est lobortis risus
um sit amet elit
erdiet aliquam
rsus ju
tur

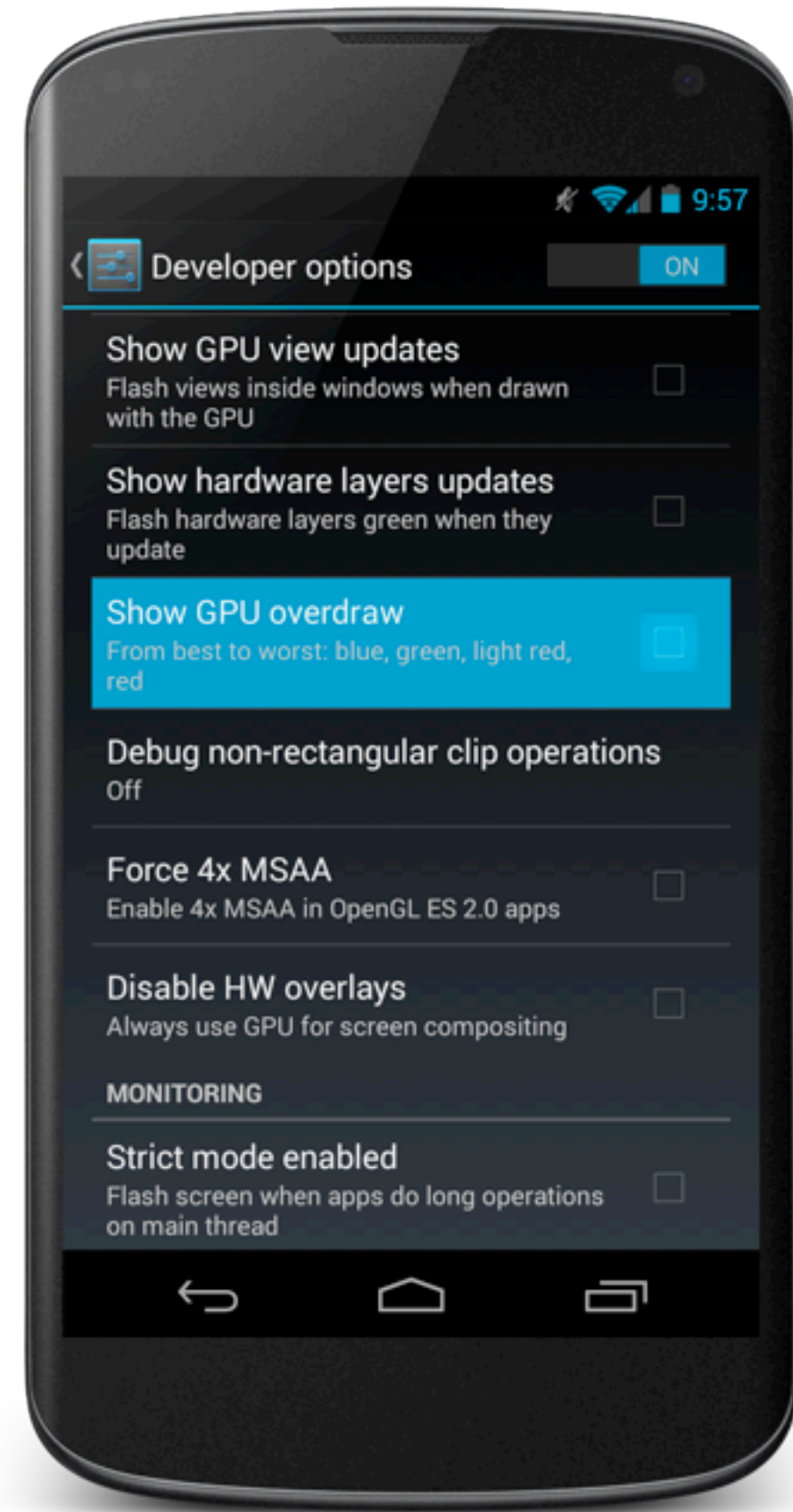
Non-rectangular clipping

```
@Override
protected void onDraw(Canvas canvas) {
    // Clip with a shape
    Path clip = getPath();
    canvas.clipPath(clip);

    // Draw the content
    for (int i = 0; i < mLines.length; i++) {
        TextLine line = mLines[i];
        canvas.drawText(line.text,
            line.x, line.y, mPaint);
    }
}
```

Developer Tools





< Developer options

ON

Show GPU view updates

Flash views inside windows when drawn with the GPU



Show hardware layers updates

Flash hardware layers green when they update



Show GPU overflow

From best to worst: blue, green, light red, red



Debug non-rectangular clip operations

Off

Force 4x MSAA

Enable 4x MSAA in OpenGL ES 2.0 apps



Disable HW overlays

Always use GPU for screen compositing



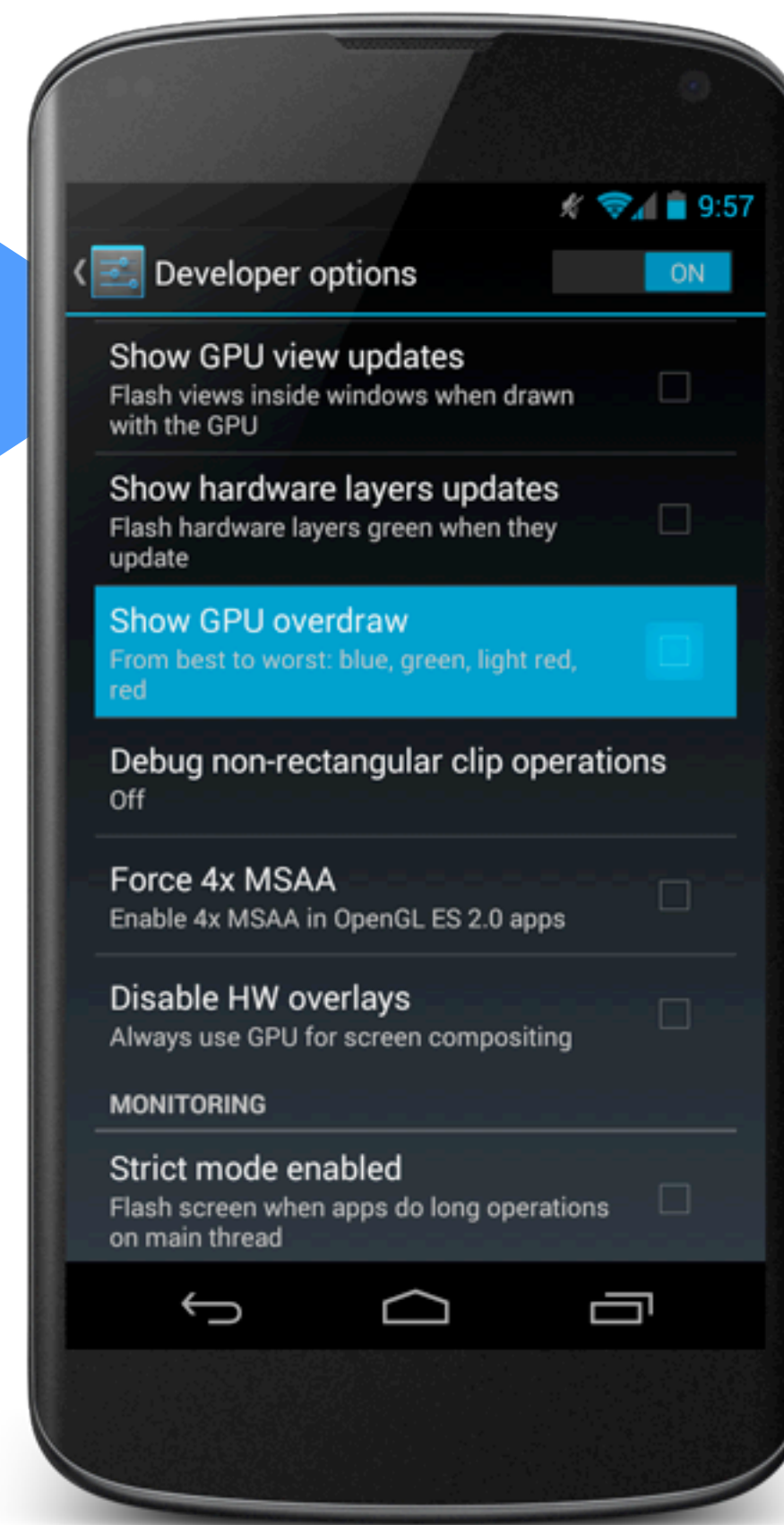
MONITORING

Strict mode enabled

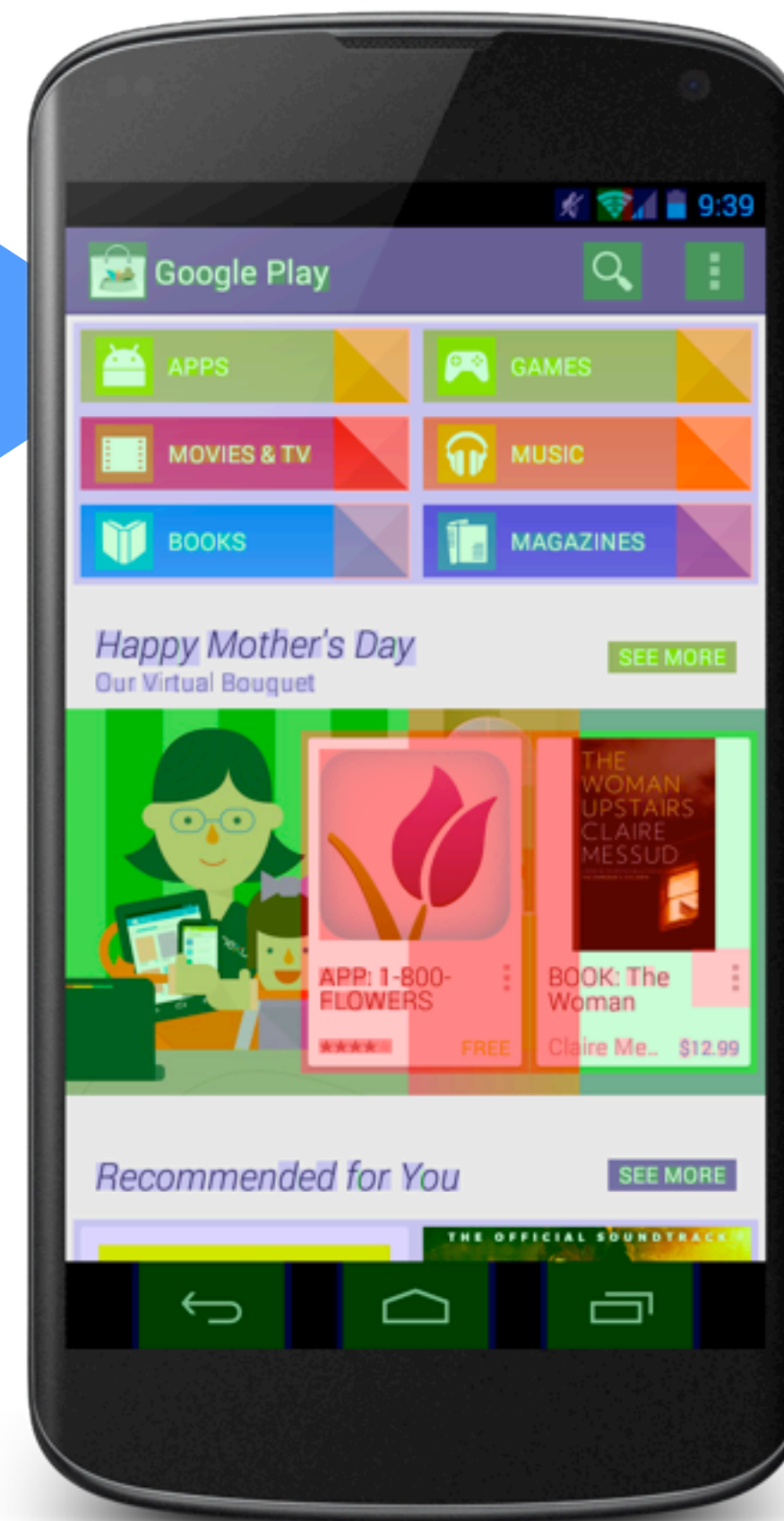
Flash screen when apps do long operations on main thread



Overdraw



Overdraw



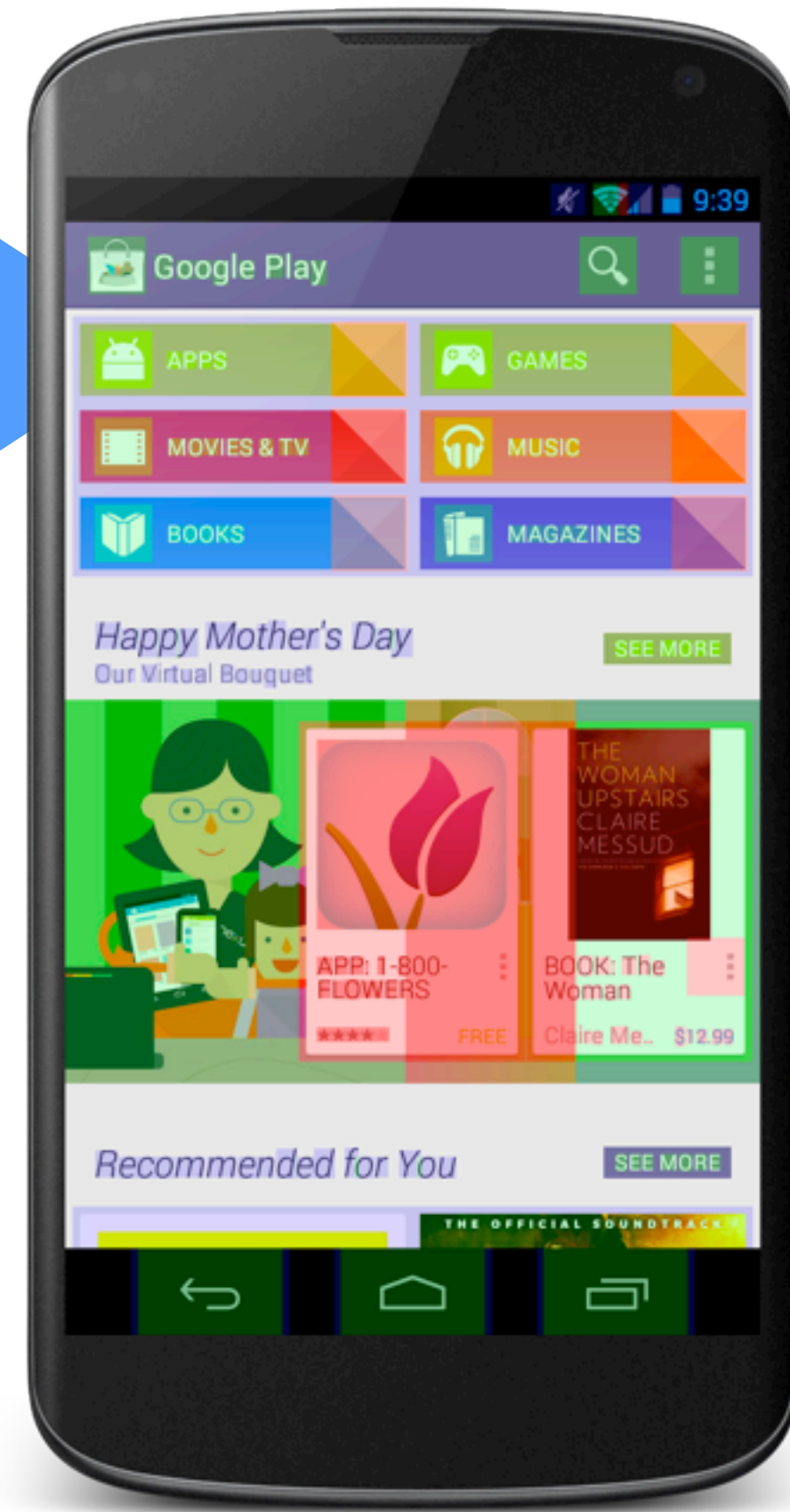
Overdraw

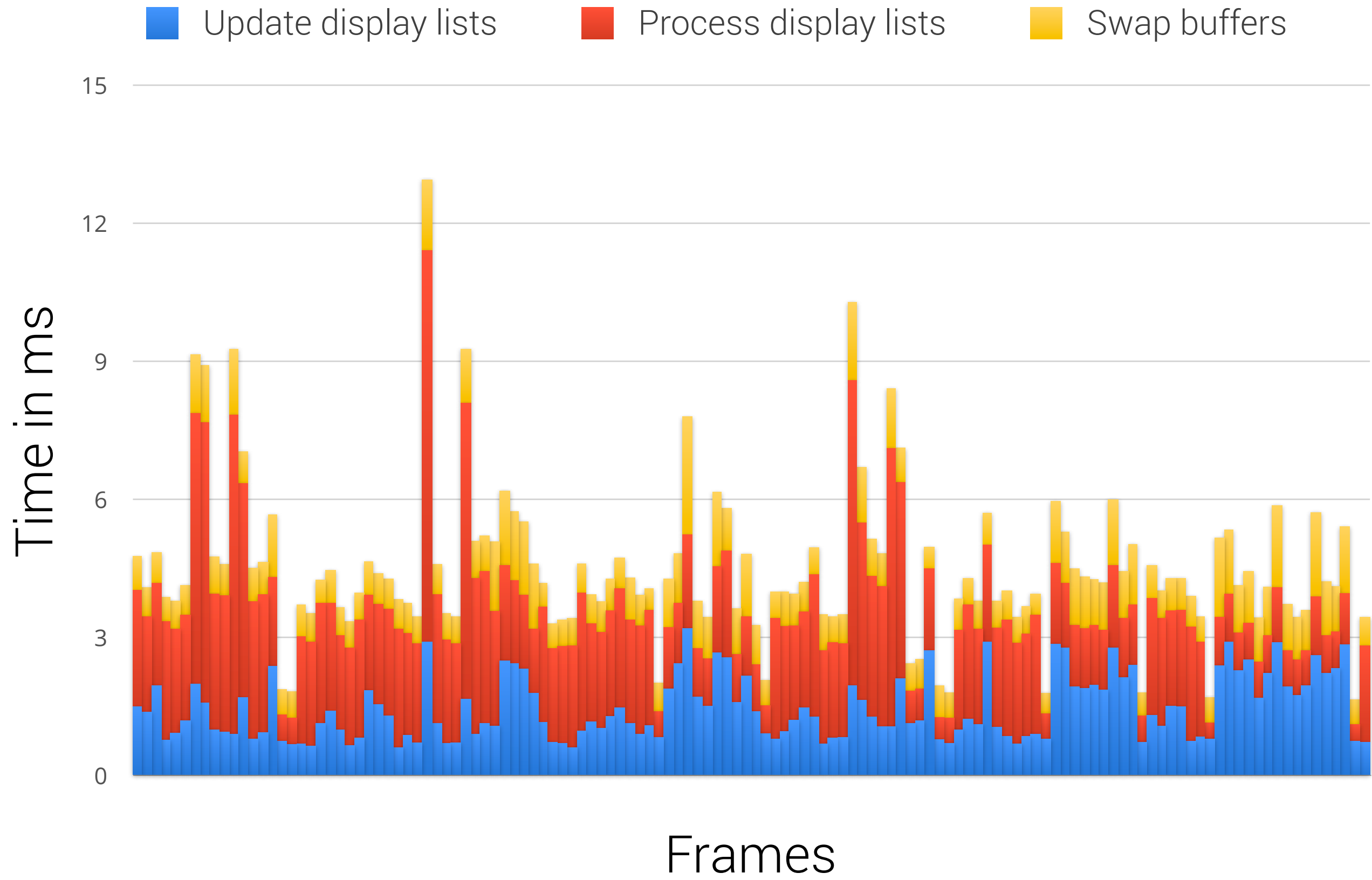
Blue 1x

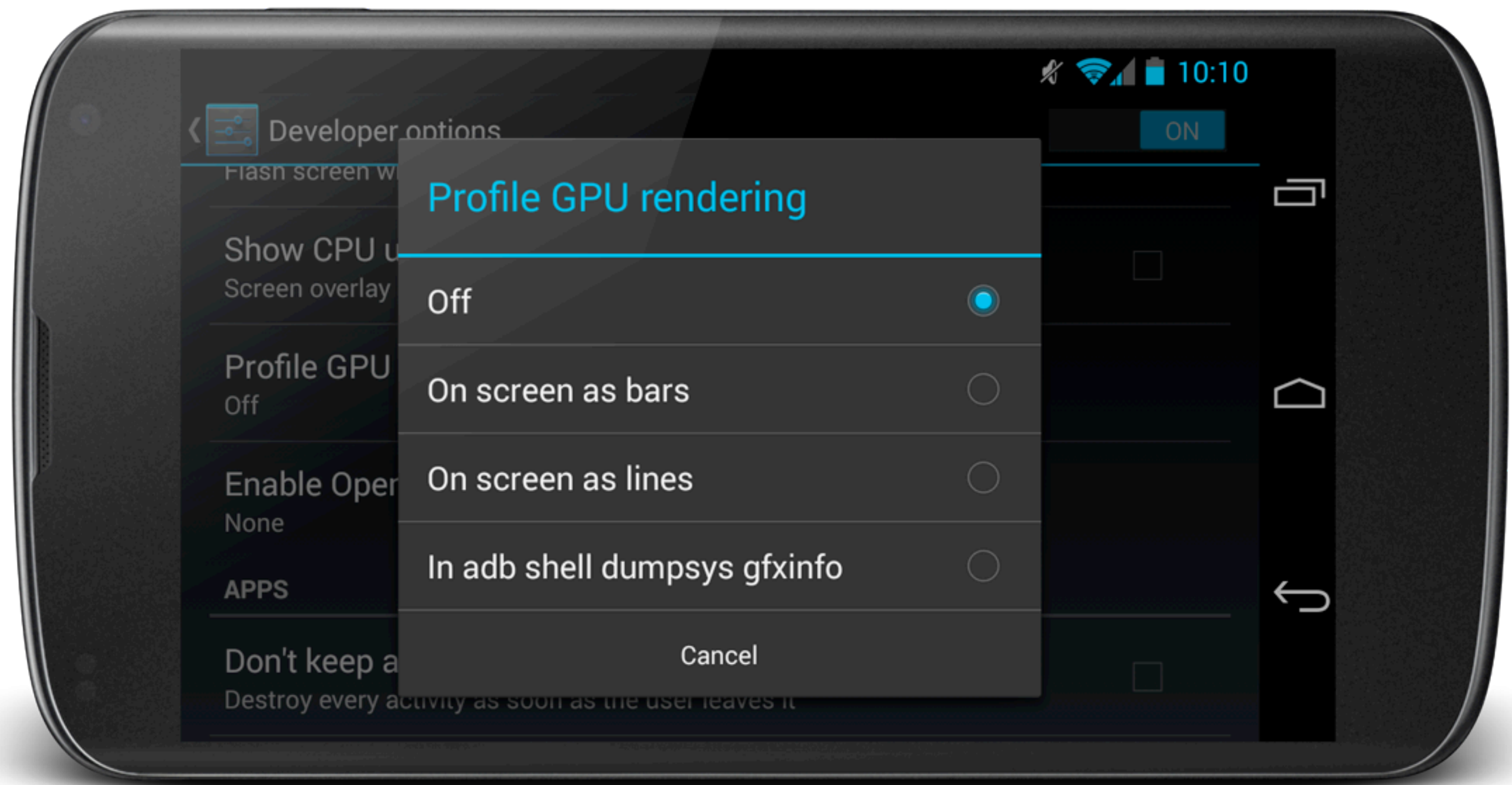
Green 2x

Red 3x

Deep Red 4x







10:10

Developer options

Flash screen w

Show CPU u

Screen overlay

Profile GPU

Off

Enable Oper

None

APPS

Don't keep a

Destroy every activity as soon as the user leaves it

ON

Profile GPU rendering

Off



On screen as bars



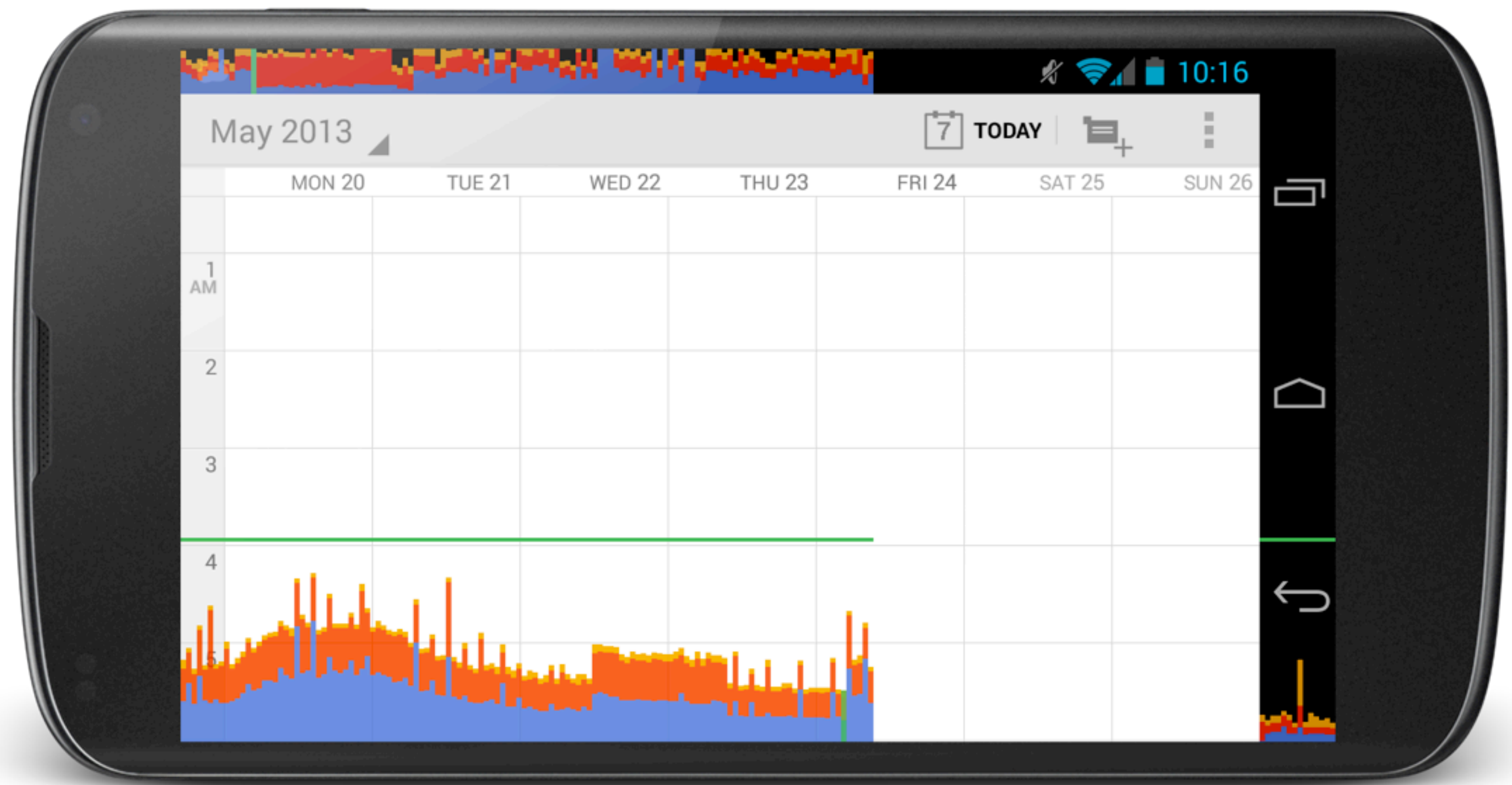
On screen as lines



In adb shell dumpsys gfxinfo



Cancel



May 2013

7 TODAY

MON 20

TUE 21

WED 22

THU 23

FRI 24

SAT 25

SUN 26

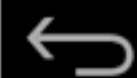
1 AM

2

3

4

5



performTraversals

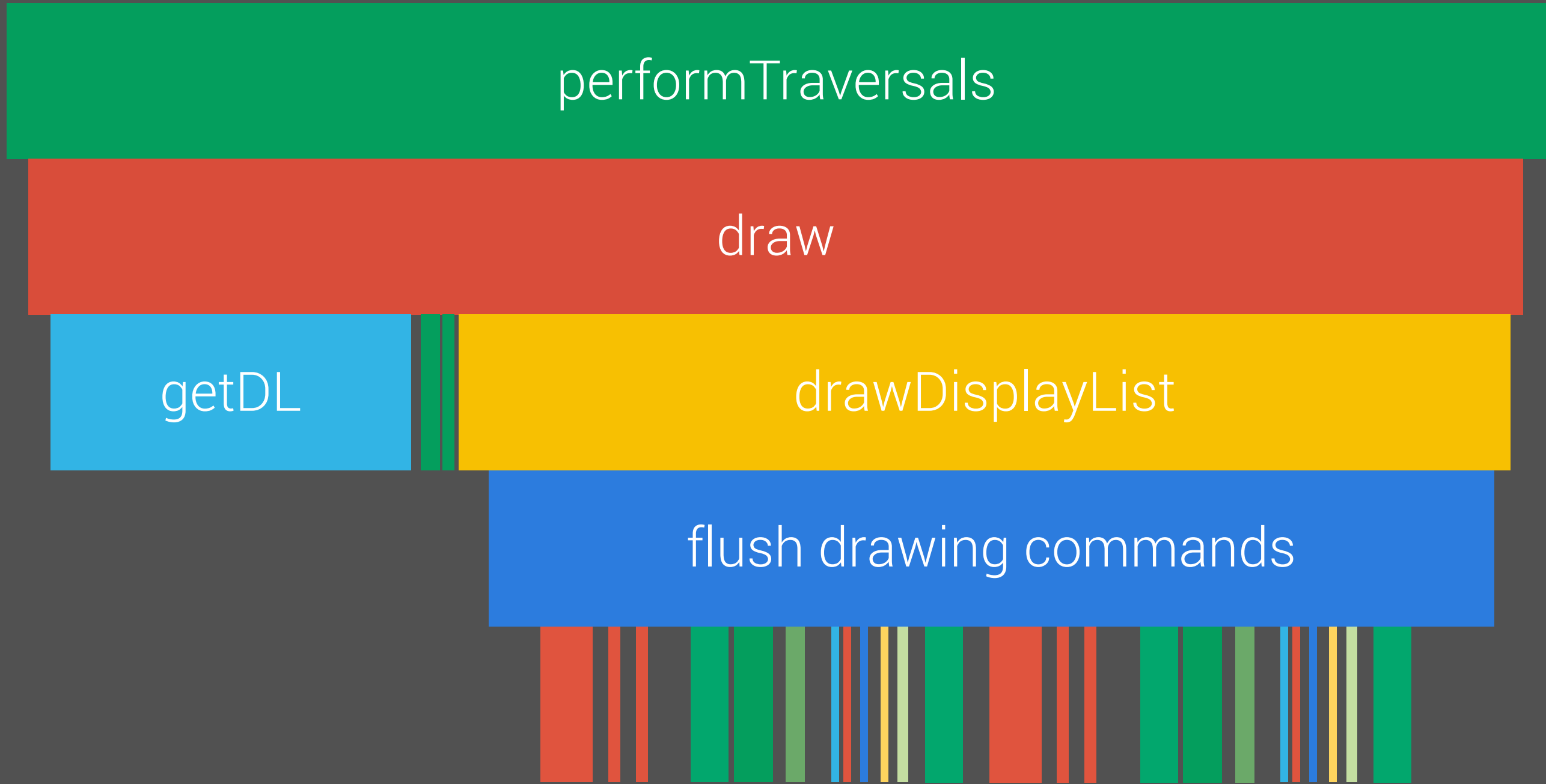
draw

getDL

drawDisplayList

flush drawing commands

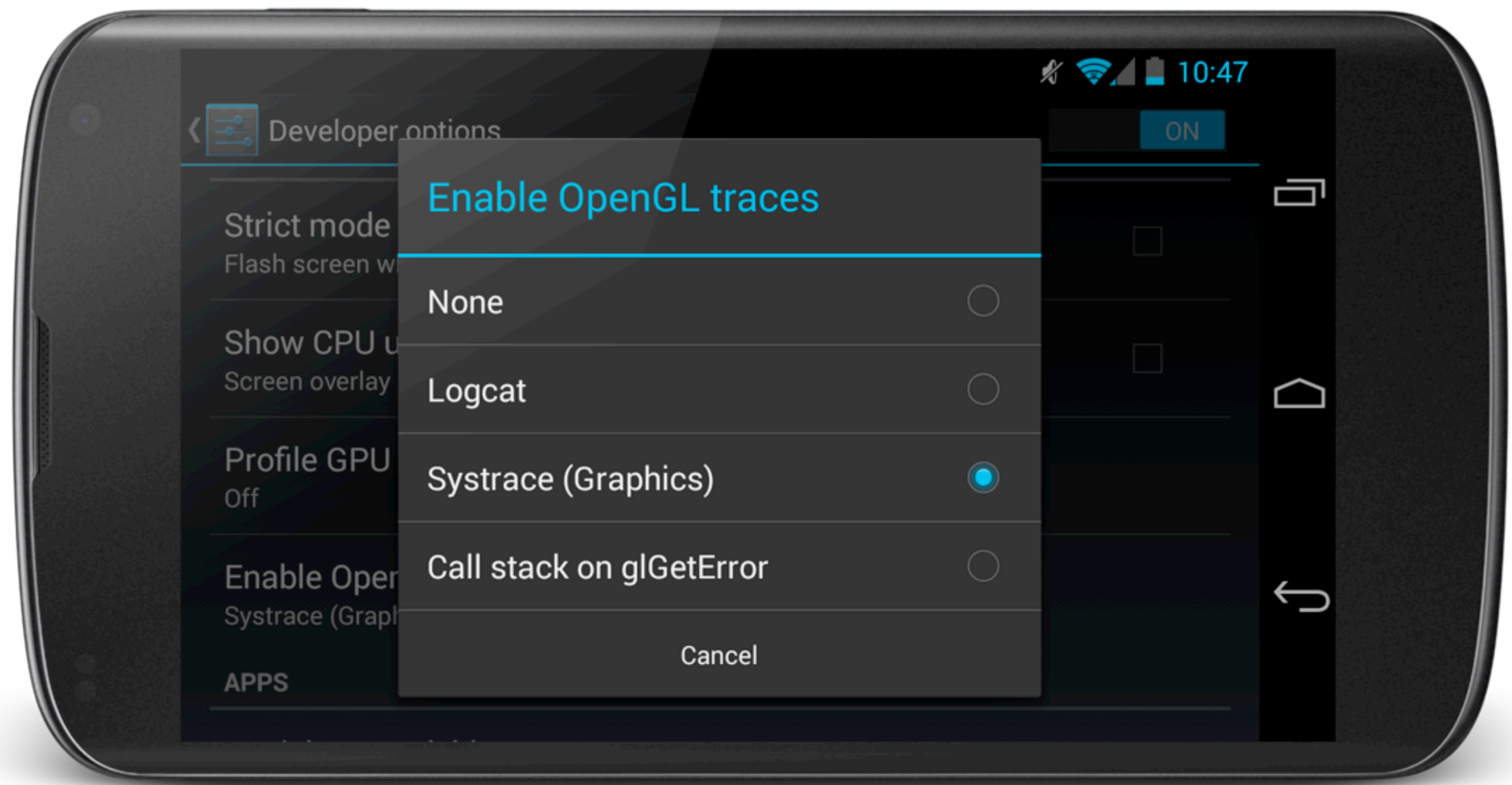
systrace



```
android:sdk $ cd platform-tools/
```

```
android:platform-tools $ ↵
```

```
./systrace.py gfx view freq sched ■
```



10:47

Developer options

ON

Enable OpenGL traces

None

Logcat

Systrace (Graphics)

Call stack on glGetError

Cancel

Strict mode

Flash screen w

Show CPU u

Screen overlay

Profile GPU

Off

Enable Oper

Systrace (Graph

APPS

```
import android.os.Trace;

@Override
public View getView(int pos, View view, ViewGroup parent) {
    Trace.beginSection("getView");
    if (view == null) {
        view = createView();
    }

    // Trace time spent binding data
    Trace.beginSection("bind");
    bindView(pos, view);
    Trace.endSection();

    Trace.endSection();
    return view;
}
```

```
android:sdk $ cd platform-tools/
```

```
android:platform-tools $ ↵
```

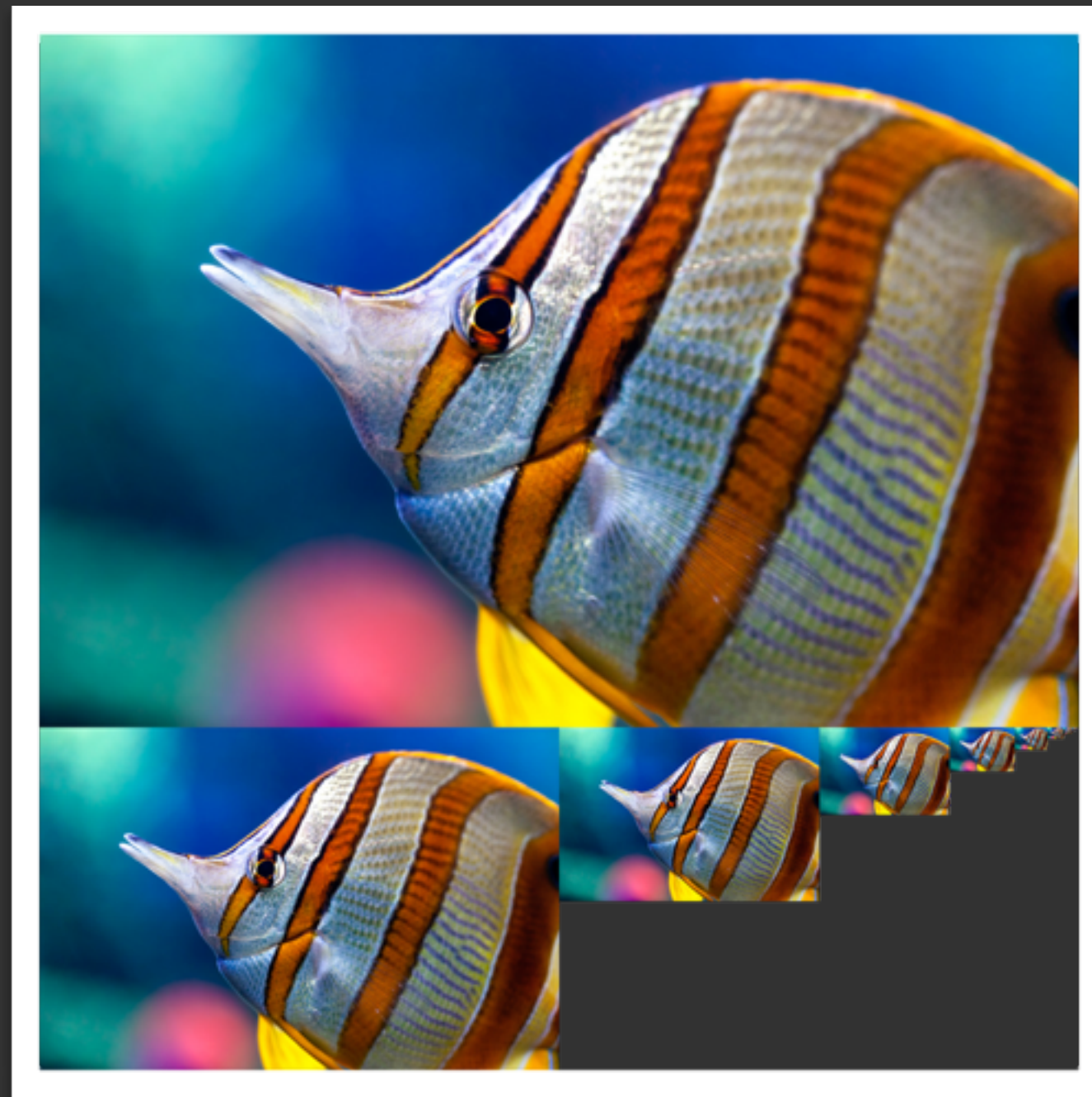
```
./systrace.py -a com.example.myapplication ■
```


Tips & Tricks





Overdraw demo



Trilinear filtering

MipMap

4:54





Off



On

```
private void loadData() {  
    // Load bitmap  
    Bitmap b = getBitmap();  
  
    // Enable trilinear filtering  
    b.setHasMipMap(true);  
}
```

```
<bitmap  
    android:mipMap="true"  
    android:src="@drawable/my_drawable" />
```

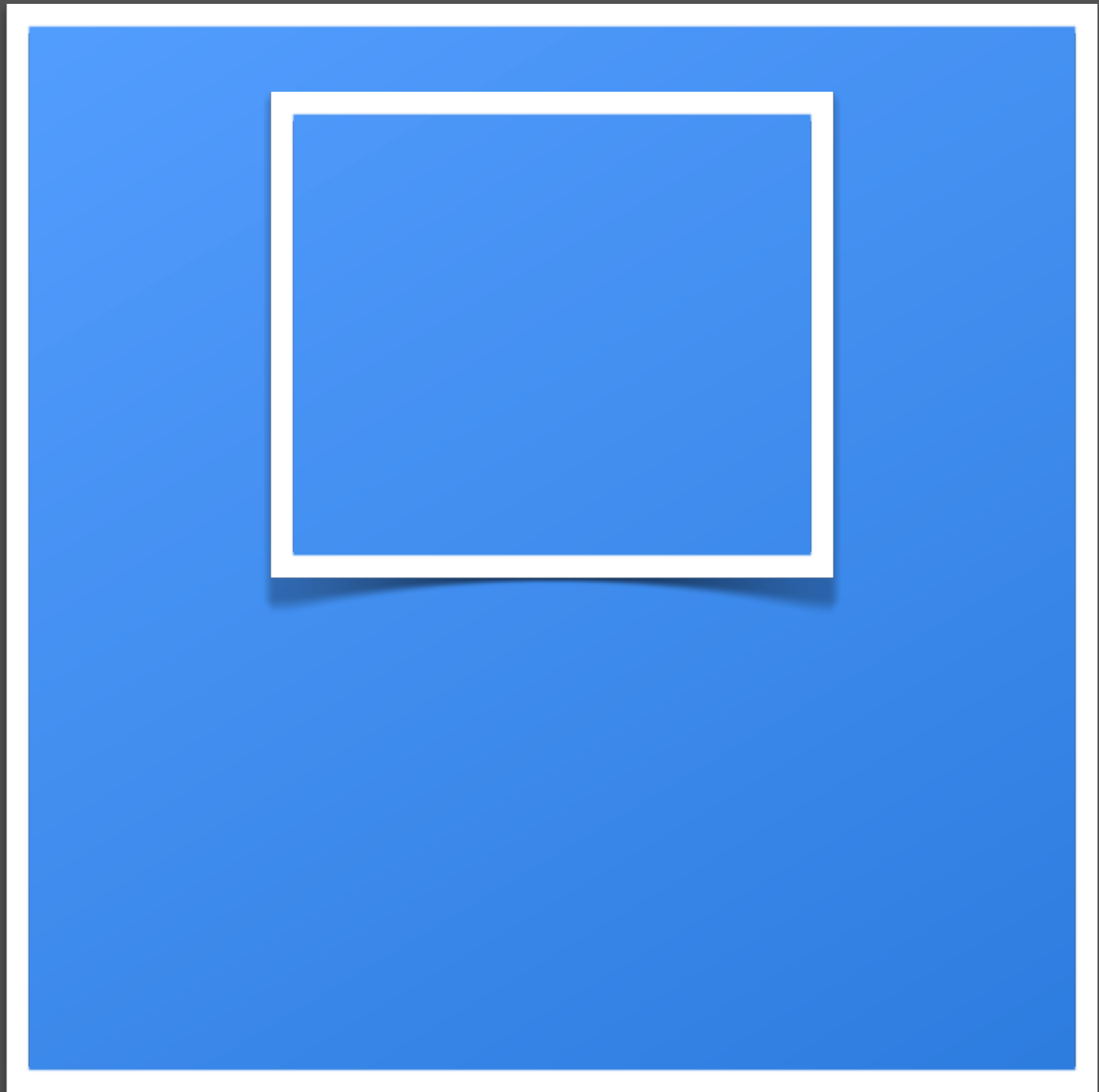


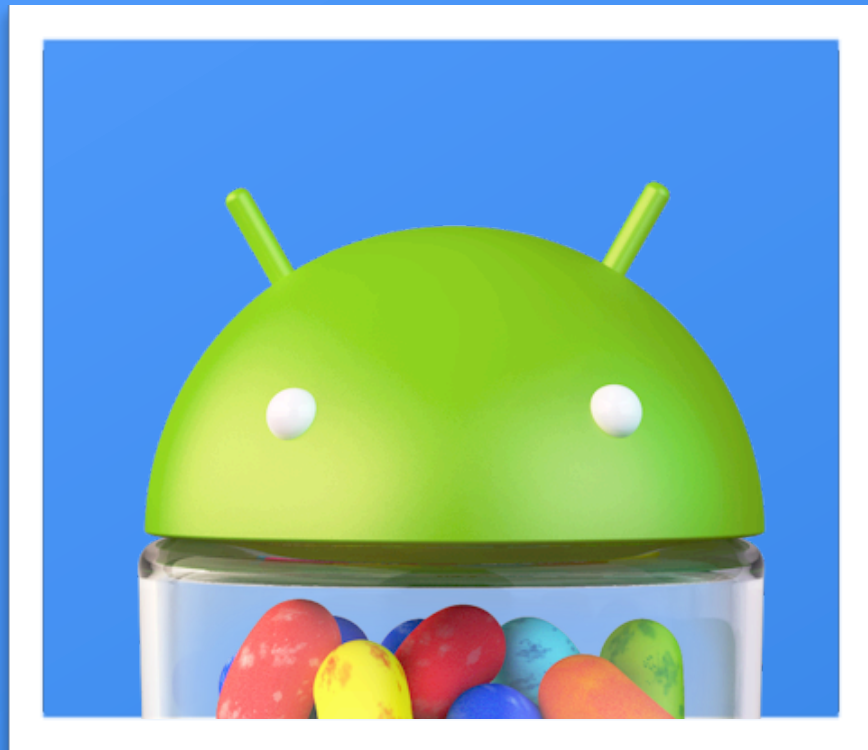
Canvas layers


```
@Override
protected void onDraw(Canvas canvas) {
    // Create a clipped layer
    canvas.save();
    canvas.saveLayer(x, y, width, height,
        Canvas.CLIP_TO_LAYER_SAVE_FLAG);

    // Draw stuff
    canvas.drawBitmap(bugDroid, 0.0f, 0.0f, null);
    canvas.restore();
}
```

saveLayer()

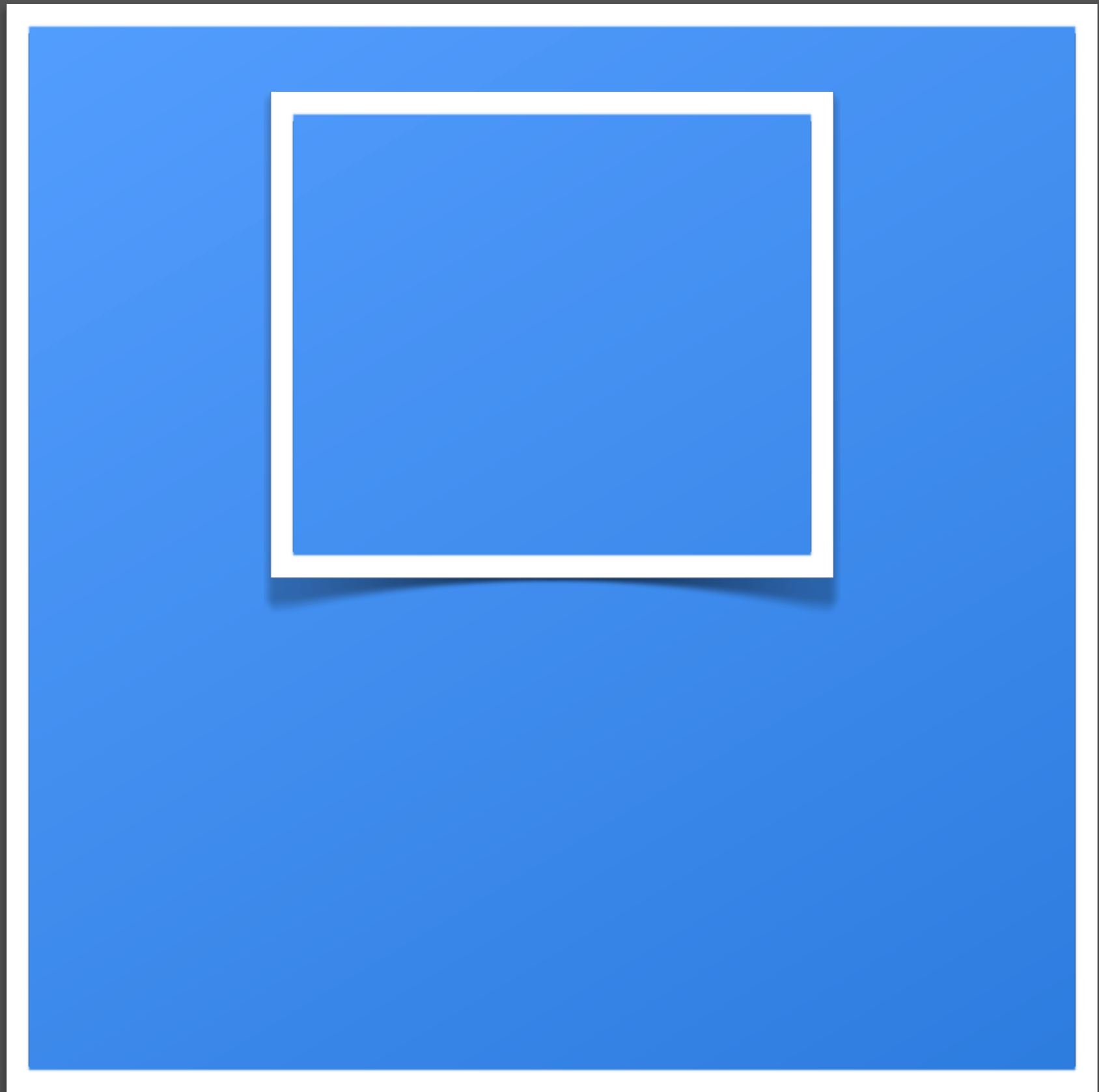




```
@Override
protected void onDraw(Canvas canvas) {
    // Create an unclipped layer
    canvas.save();
    canvas.saveLayer(x, y, width, height, 0);

    // Draw stuff
    canvas.drawBitmap(bugDroid, 0.0f, 0.0f, null);
    canvas.restore();
}
```

saveLayer()







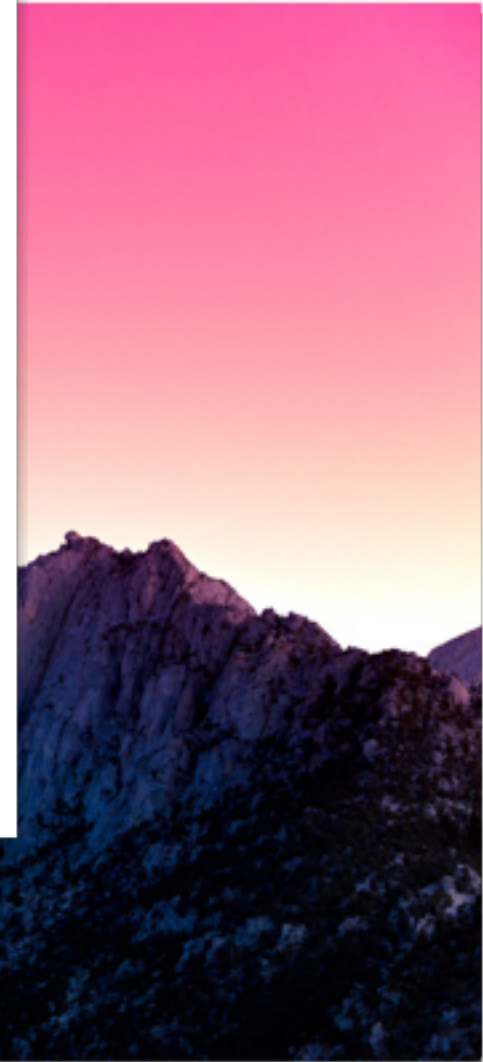
Using alpha with care

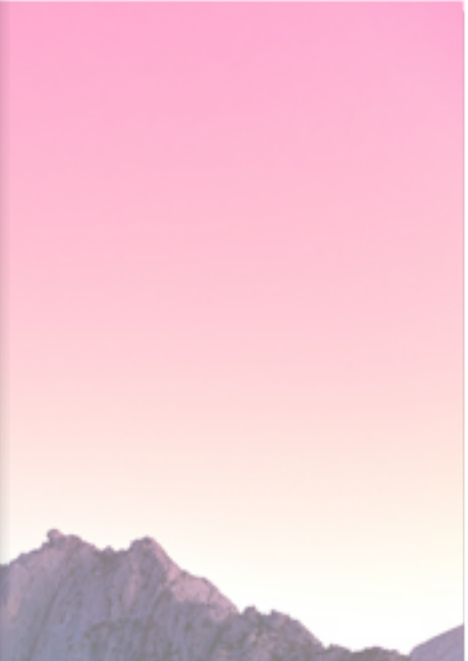

```
view.setAlpha(0.5f);  
View.ALPHA.set(view, 0.5f);  
ObjectAnimation.ofFloat(view, "alpha", 0.5f)  
view.animate().alpha(0.5f);  
view.setAnimation(new AlphaAnimation(1.0f, 0.5f));
```

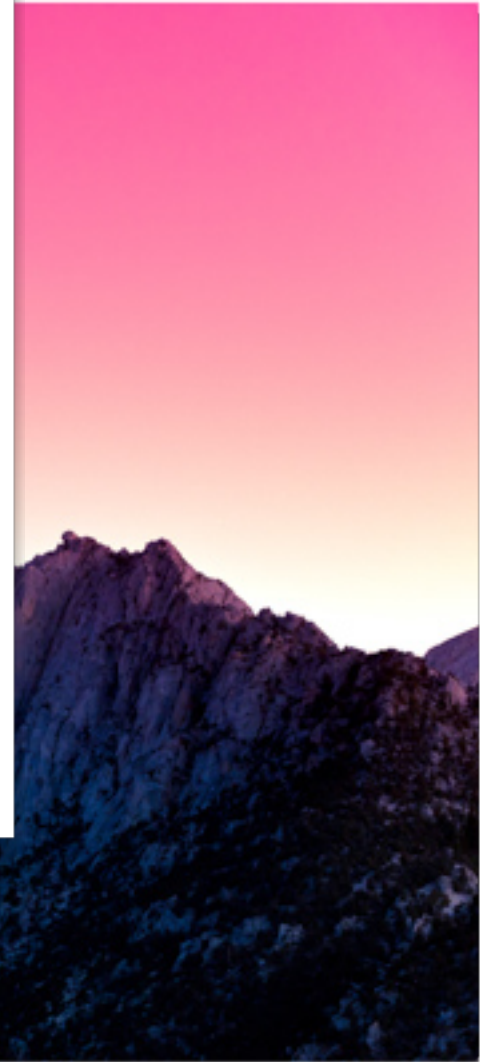
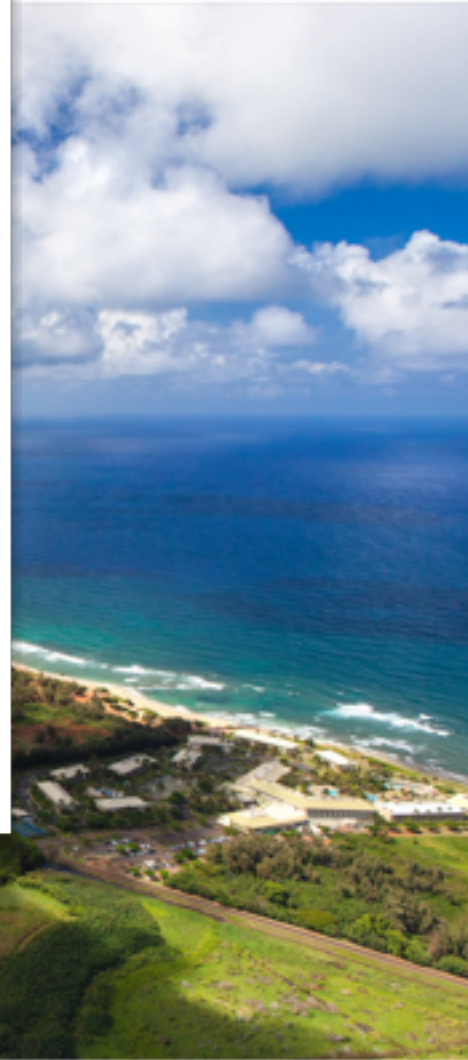
```
view.setAlpha(0.5f);  
View.ALPHA.set(view, 0.5f);  
ObjectAnimation.ofFloat(view, "alpha", 0.5f)  
view.animate().alpha(0.5f);  
view.setAnimation(new AlphaAnimation(1.0f, 0.5f));
```

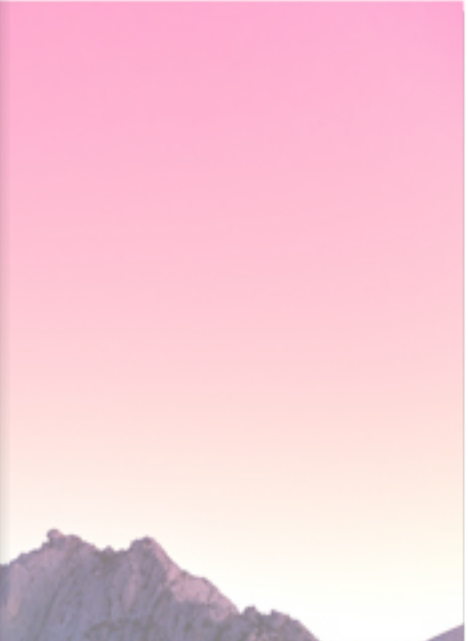
==

```
Canvas.saveLayerAlpha(l, t, r, b, 127,  
Canvas.CLIP_TO_LAYER_SAVE_FLAG);
```









α Alternatives

```
// Not this  
textView.setAlpha(alpha);
```

```
// But this  
int newTextColor = (int) (0xFF * alpha) << 24 |  
    baseTextColor & 0xFFFFFFFF;  
textView.setTextColor(newTextColor);
```

```
// Not this  
imageView.setAlpha(alpha);
```

```
// But this  
imageView.setImageAlpha((int) (alpha * 255));
```

```
// Not this  
customView.setAlpha(alpha);
```

```
// But this  
int alpha = (int)  
    (255 * slider.getProgress() / 100.0f);  
paint.setAlpha(alpha);  
canvas.draw*(..., paint);
```

```
// Or use a layer
view.setLayerType(View.LAYER_TYPE_HARDWARE, null);

// Transient layer
view.animate().alpha(0).withLayer();
```

```
// API level 16+
@Override
public boolean hasOverlappingRendering() {
    // Don't lie to us!
    return false;
}
```

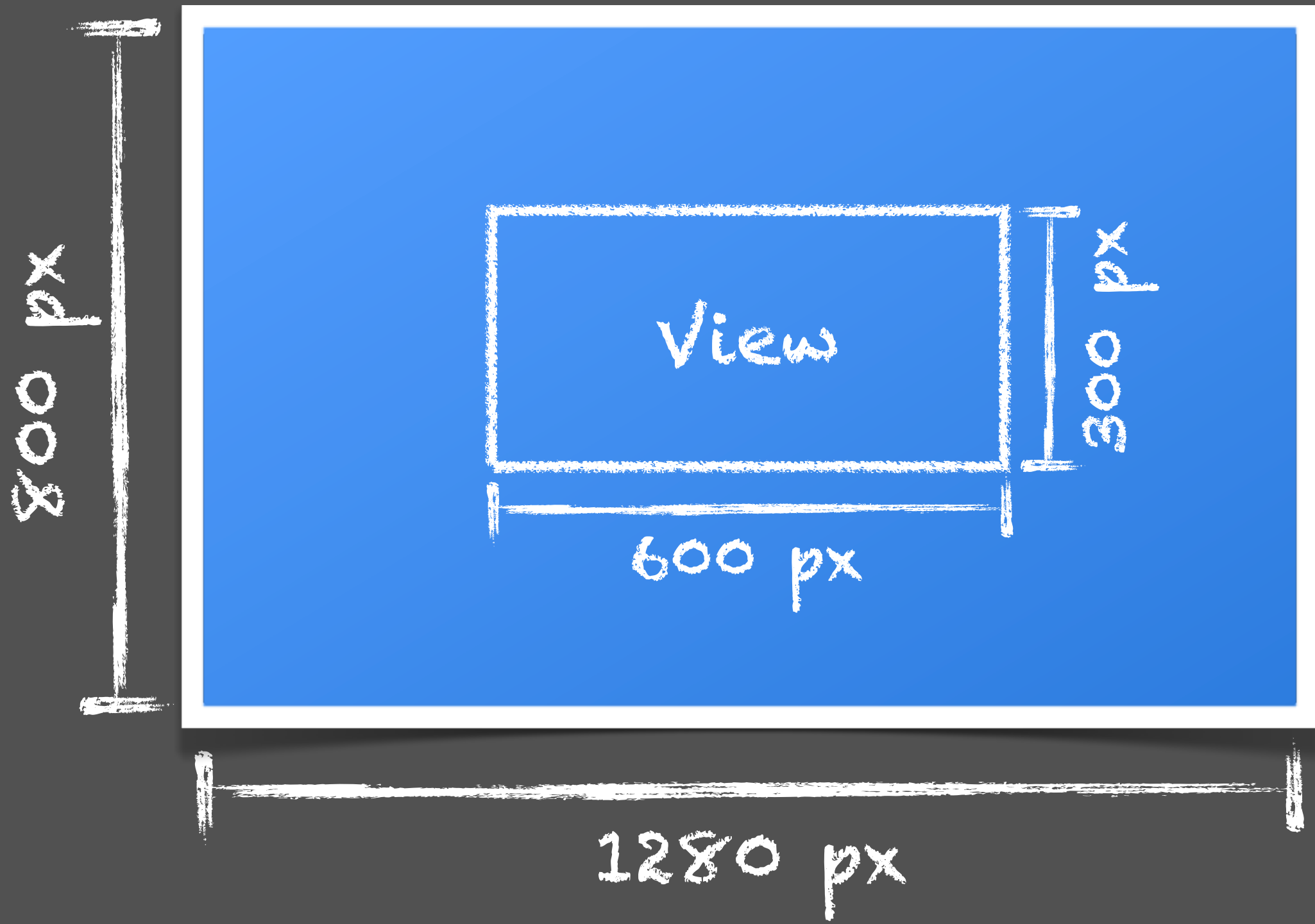
400 dp

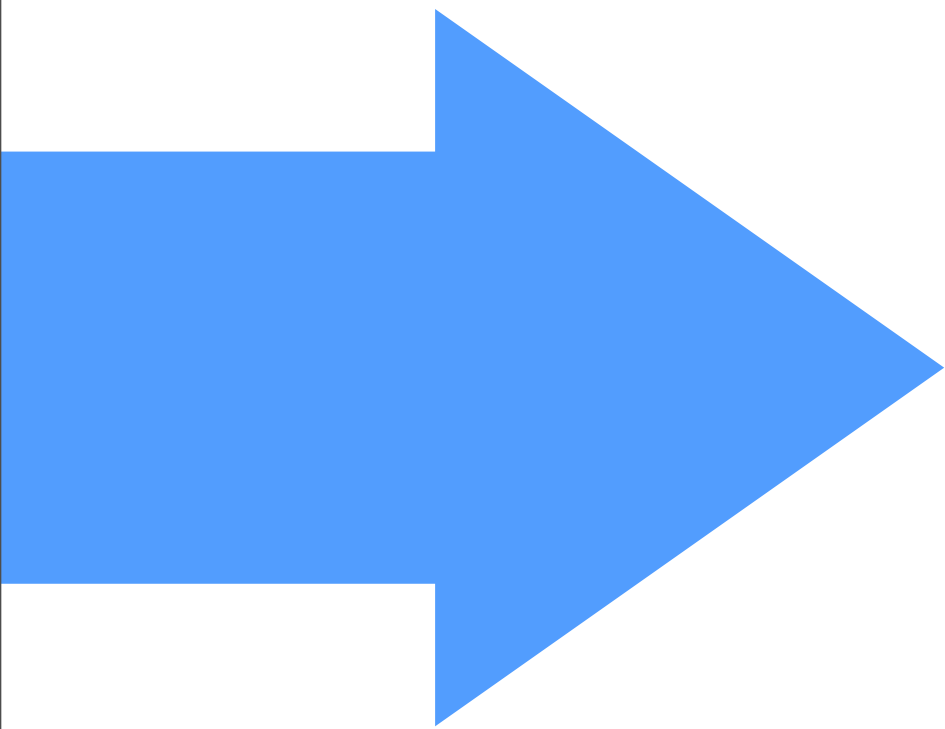
Canvas

640 dp

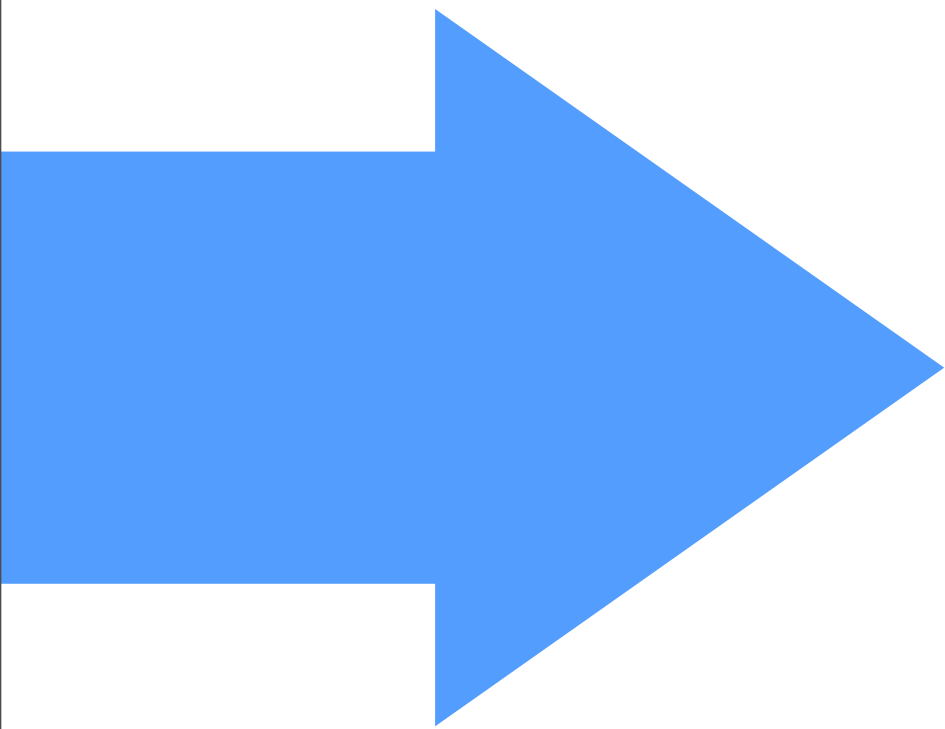
```
@Override
protected void onDraw(Canvas canvas) {
    // Get the dimensions of the Canvas
    int w = canvas.getWidth();
    int h = canvas.getHeight();

    canvas.drawRect(0, 0, w, h, mPaint);
}
```

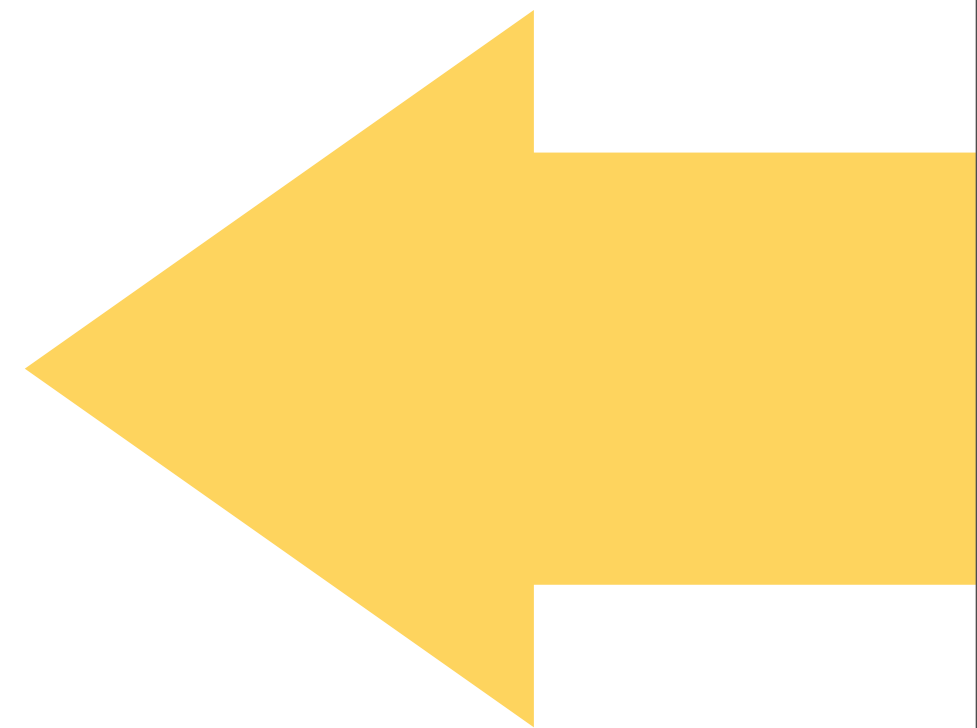


With **hardware** rendering
600x300 px (size of the **View**)



With **hardware** rendering
600x300 px (size of the **View**)

With **software** rendering
1280x800 px
(size of the **window**)



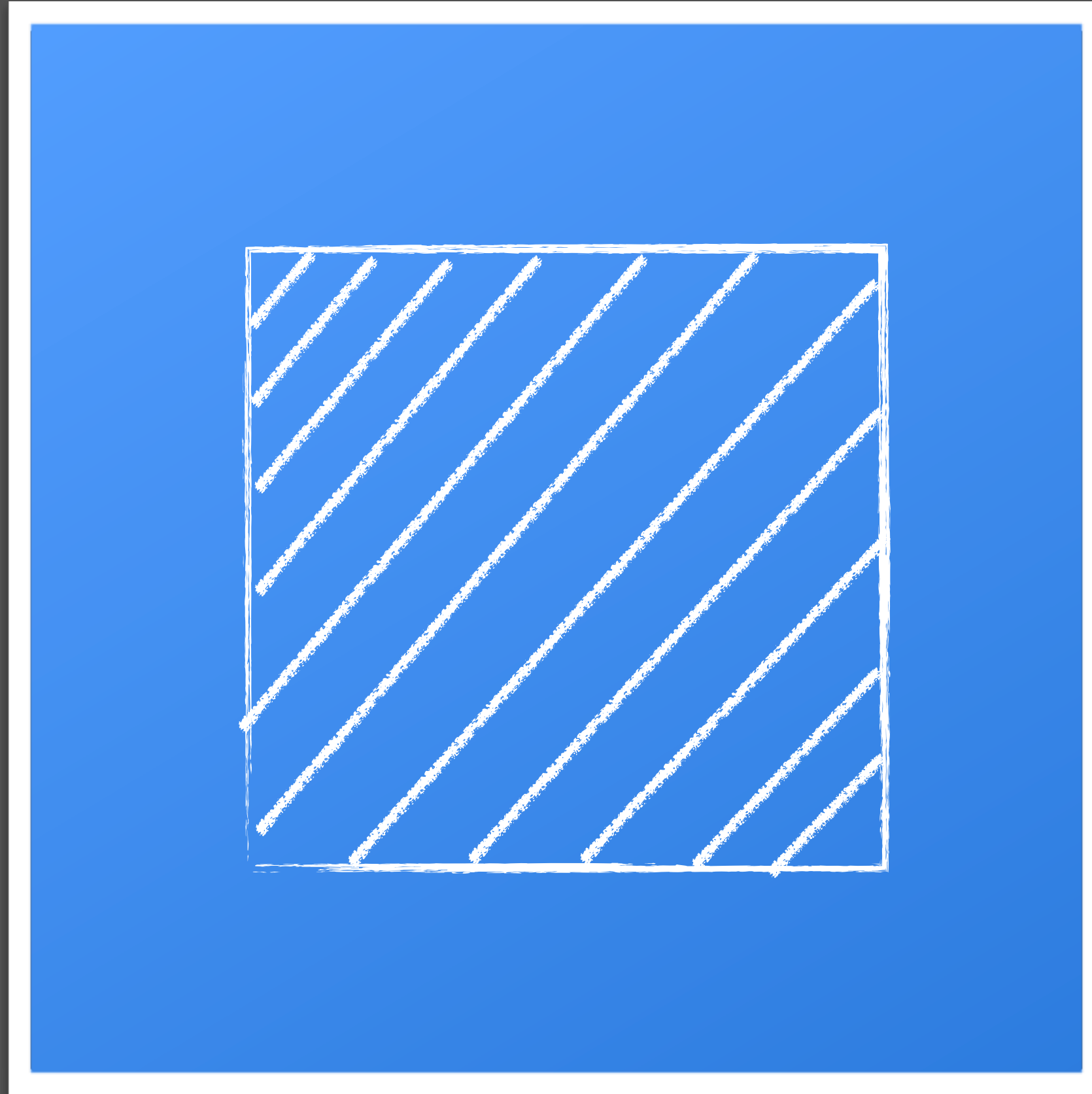


Clipping

```
@Override
protected void onDraw(Canvas canvas) {
    // Keep the jellybeans
    canvas.clipRect(l, t, r, b);
    // Rotate the jar
    canvas.rotate(-30.0f, pX, pY);
    // Draw the jar
    canvas.drawBitmap(mJellyBeans, x, y, null);
}
```



1. Clip





2. Rotate



3. Draw



```
@Override
protected void onDraw(Canvas canvas) {
    // Rotate the jar
    canvas.rotate(-30.0f, pX, pY);
    // Keep the jellybeans
    canvas.clipRect(l, t, r, b);
    // Draw the jar
    canvas.drawBitmap(mJellyBeans, x, y, null);
}
```





1. Rotate



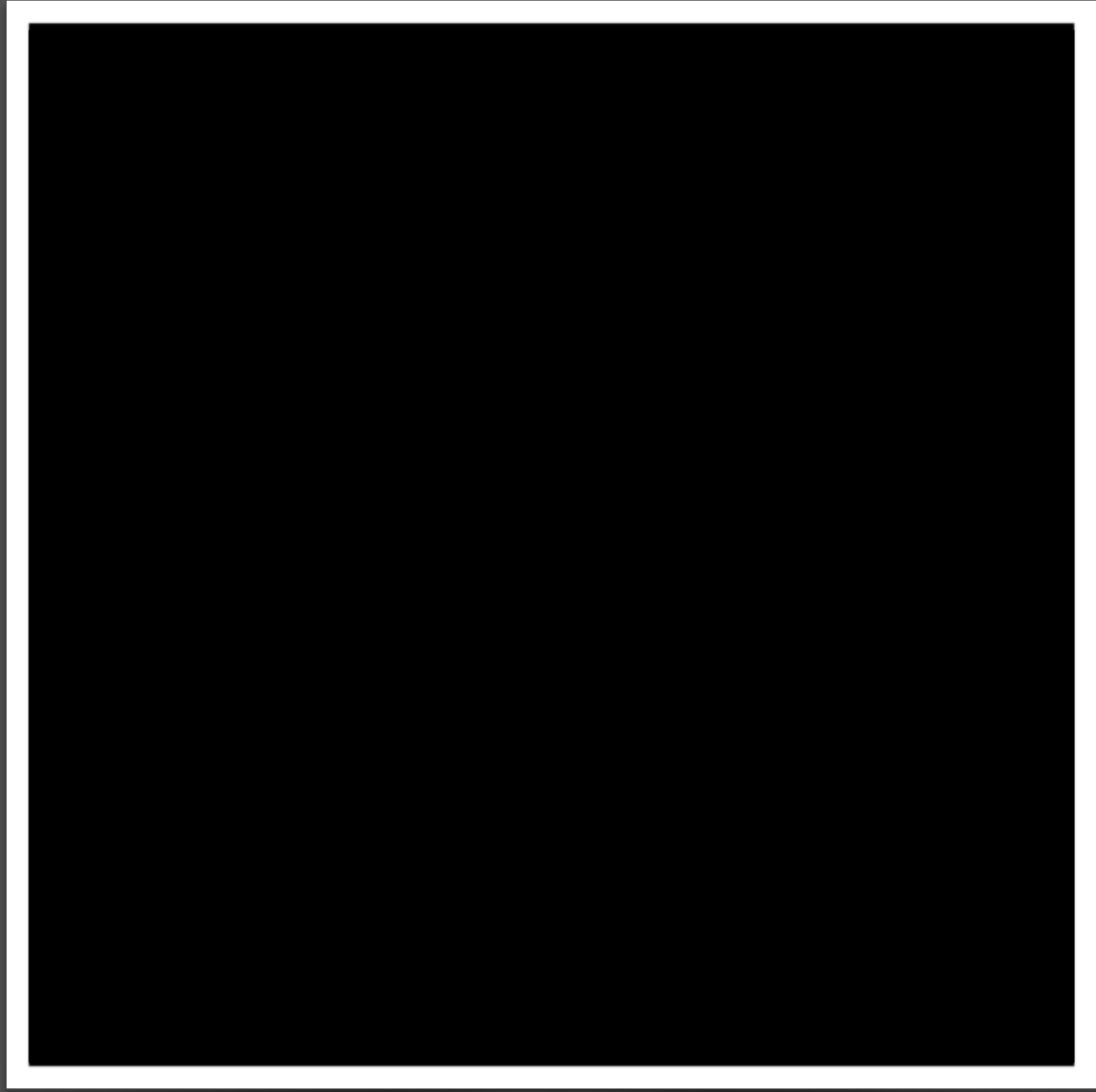
2. Clip



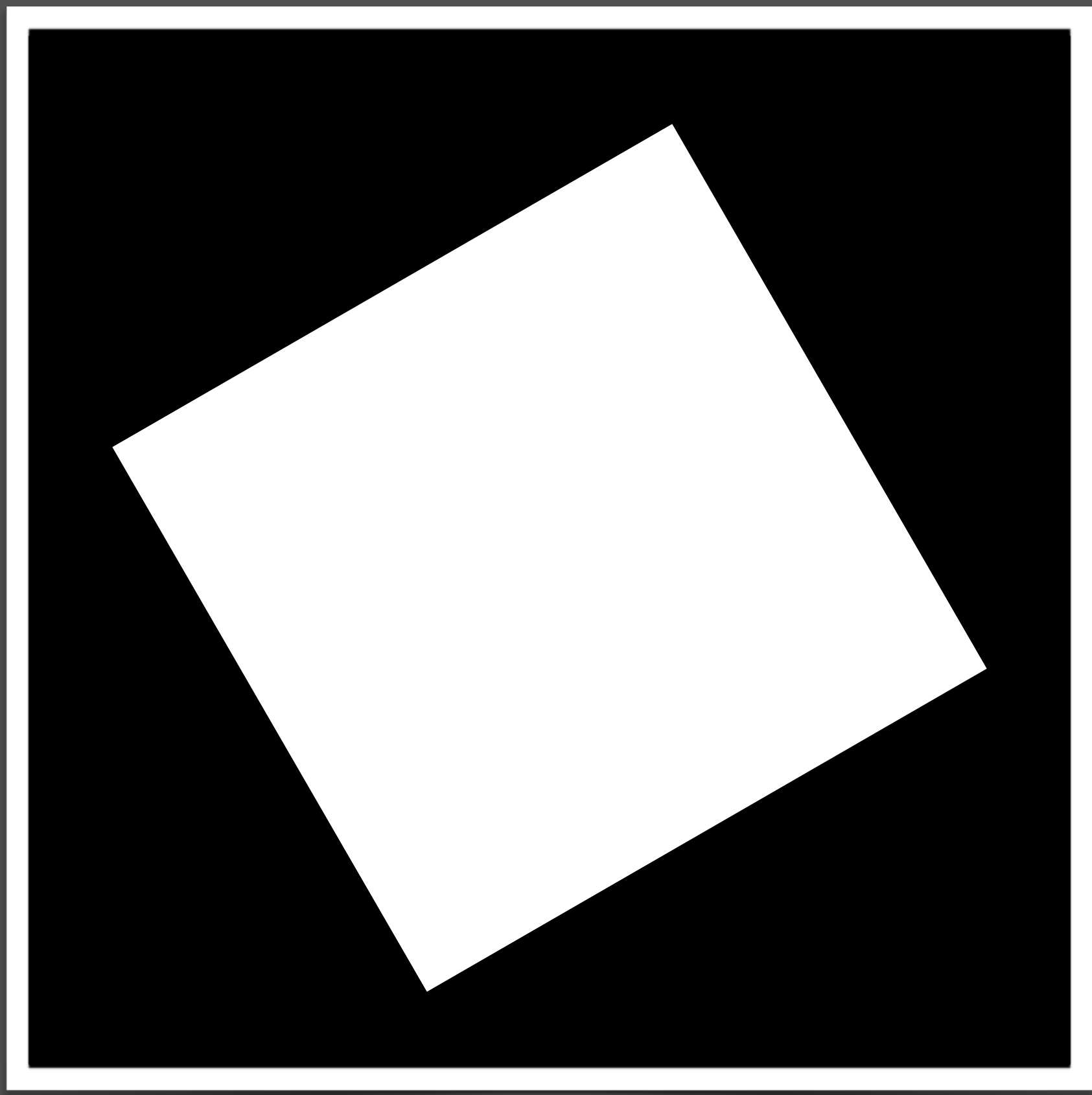
3. Draw



Stencil buffer



Stencil buffer



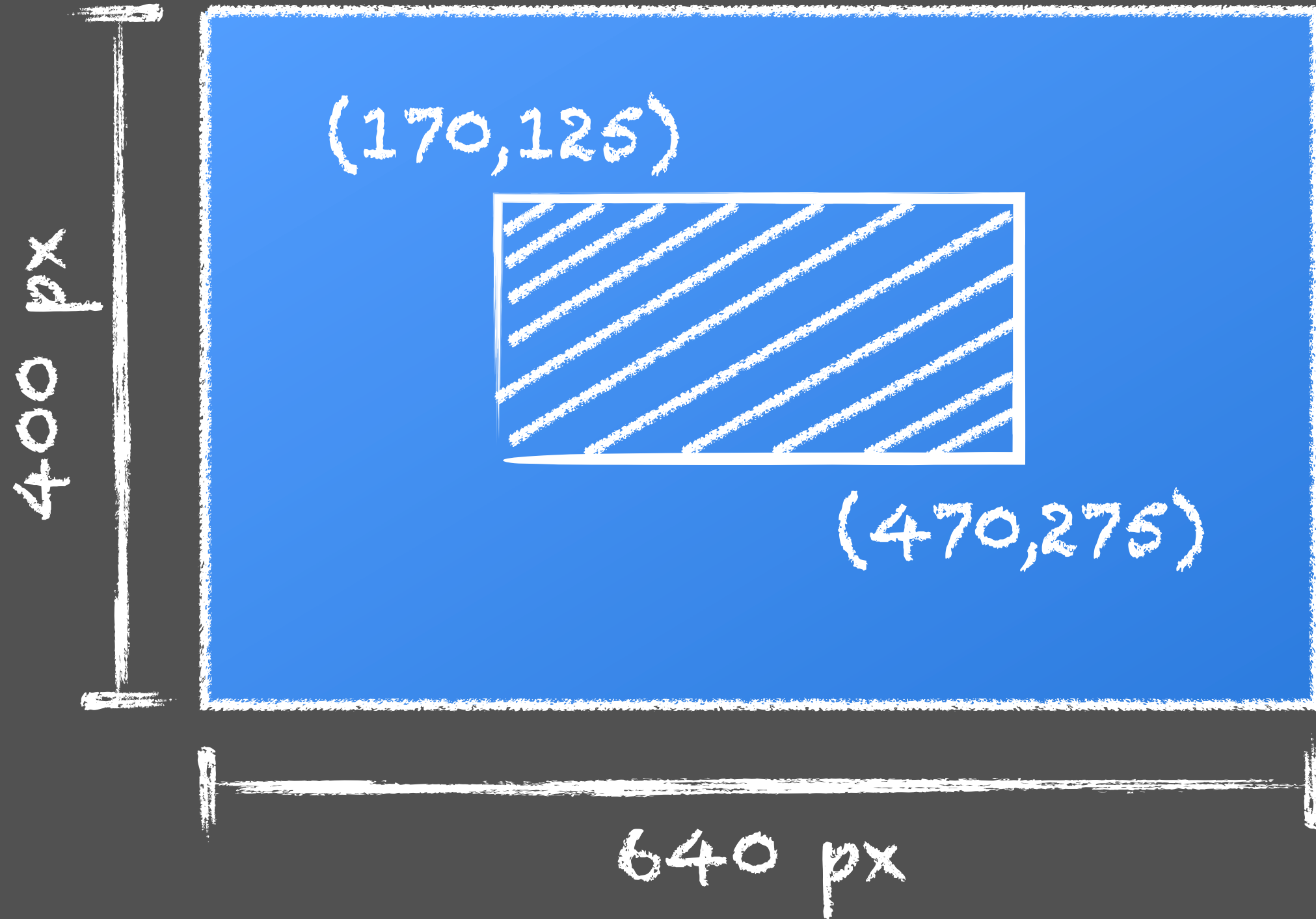
View

400 px

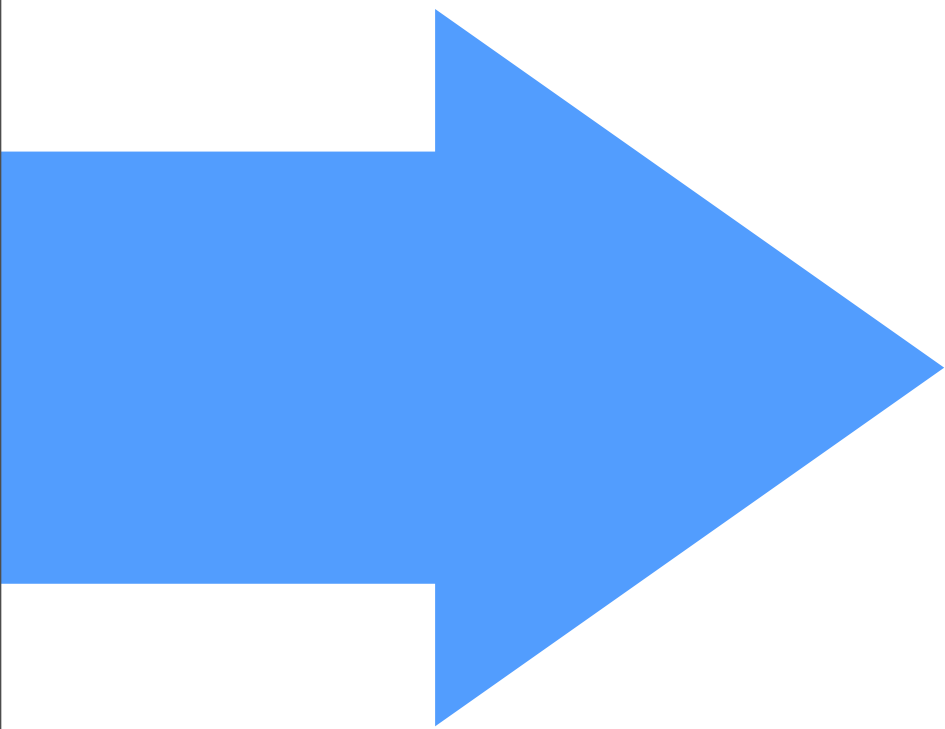
640 px



Invalidate



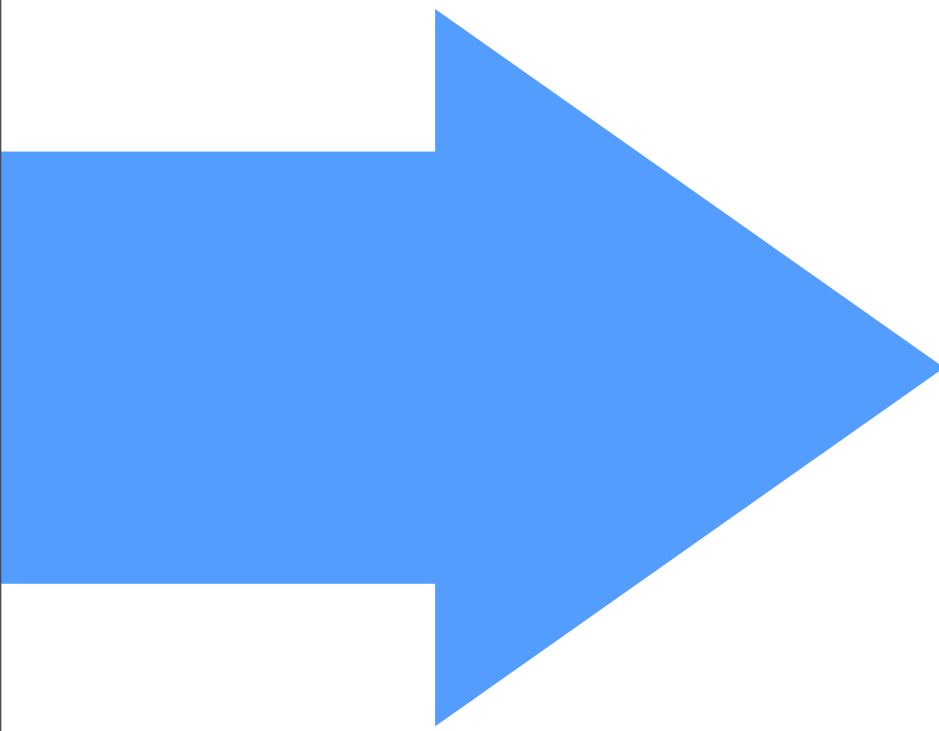
```
@Override
protected void onDraw(Canvas canvas) {
    // Query the current clip
    Rect clip = canvas.getClipBounds();
    // ???
    Log.d("I/O", "clip = " + clip);
}
```

With **hardware** rendering

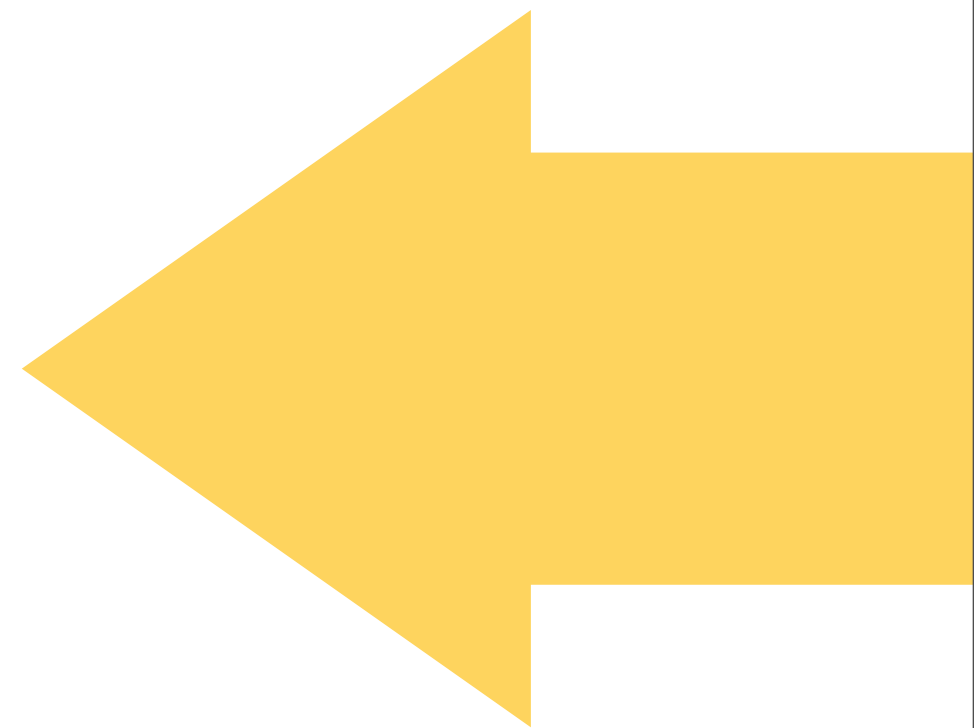
0, 0, 640, 400

(bounds of the **View**)



With **hardware** rendering
0, 0, 640, 400
(bounds of the **View**)

With **software** rendering
170, 125, 470, 275
(bounds of the **dirty rect**)

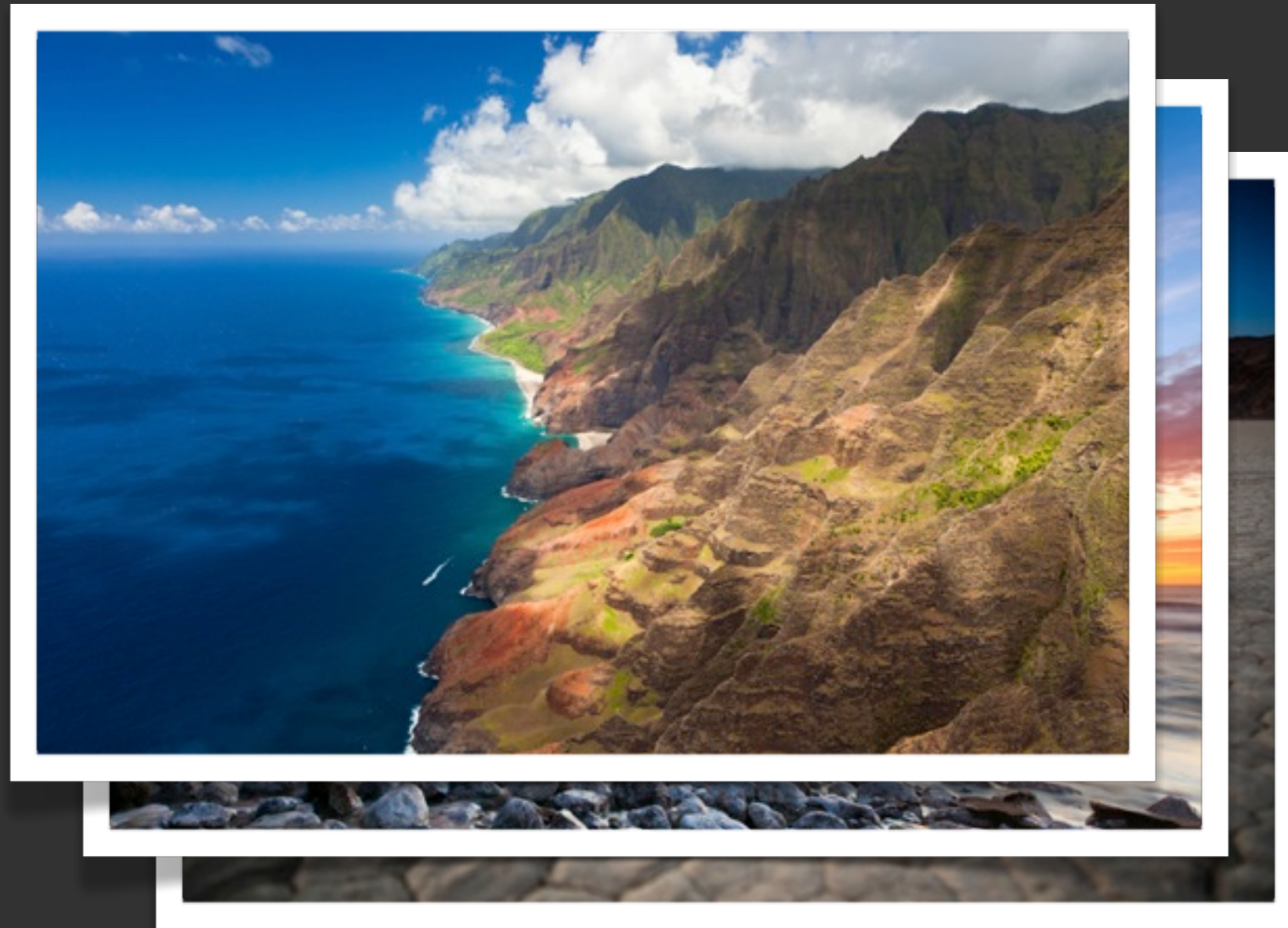




Reordering barriers

ipsum ad amet, con
ipiscing elit. Sed vitae risus non s
ctum pellentesque at convallis mi. In
agna nisi. Integer lorem odio, blandit
ia malesuada, auctor commodo odio
auris in ipsum leo, egestas imperdie
rabitur adipiscing congue purus, p
ittis sapien faucibus vel. Mauris
s non nulla tristique sed vesti
faucibus. Curabitur vel se
urus. Interdum, nibh sit
h est lobortis risus
um sit amet elit
erdiet aliquam
rsus ju
tur

Non-rectangular clips



saveLayer()

Google I/O 2011

Accelerated Android Rendering

Google I/O 2012

For Butter or Worse

Parleys.com

Various Android GUI & performance talks



More info

Android Performance Case Study

goo.gl/y9JZr

Chet's Tips & Tricks

graphics-geek.blogspot.com

Romain's Tips & Tricks

www.curious-creature.org



More info

google.com/+ChetHaase
google.com/+RomainGuy

@chethaase
@romainguy



Q & A



Google
Developers