

getBounds()

The Drawables Story

Jamie Huson + Lisa Neigut | 10 Avril 2015

Drawables!

"something that can be drawn."

-Drawable.java JavaDocs

/

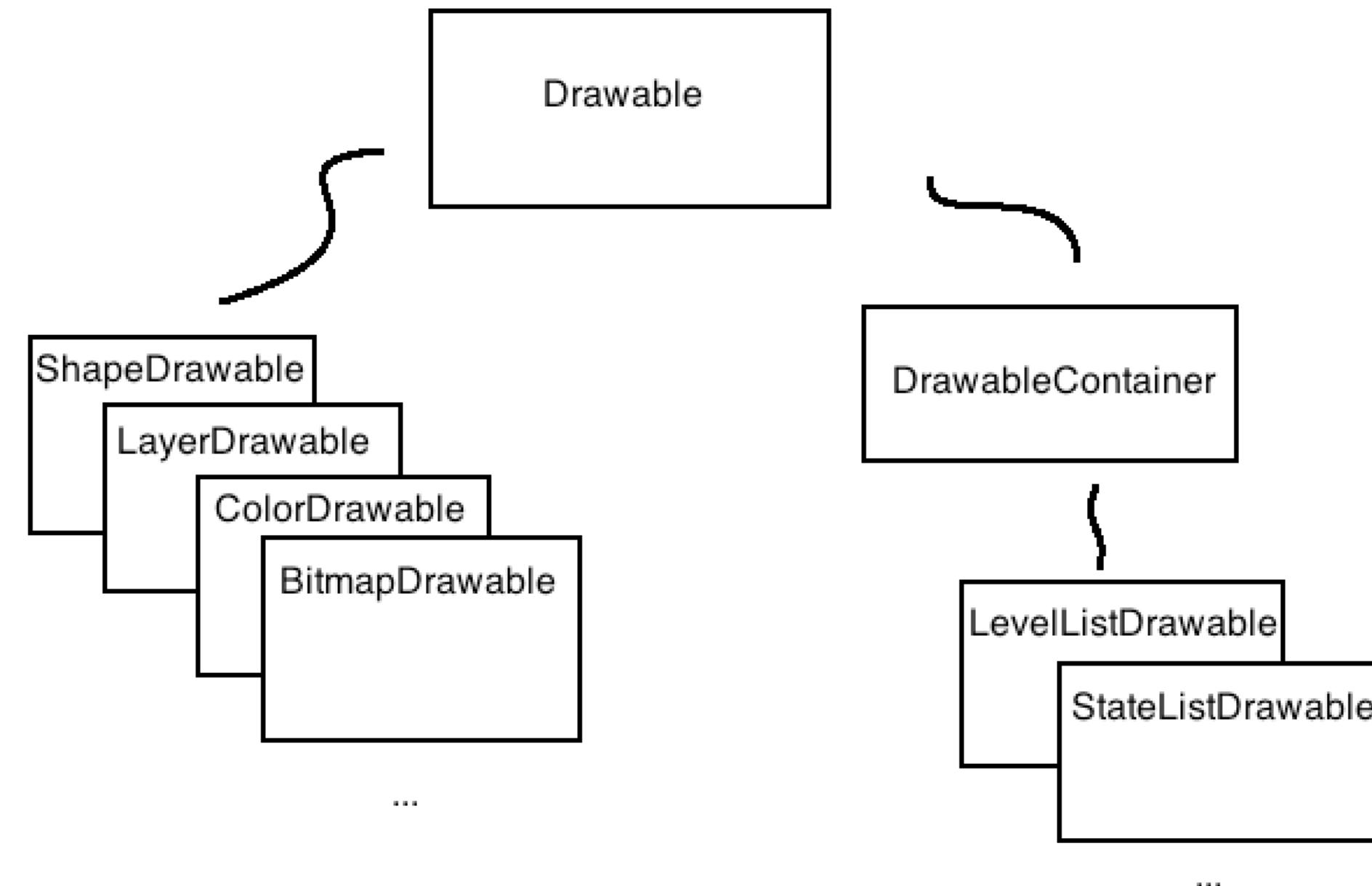
"a positioned entity

that is to be drawn on a canvas."

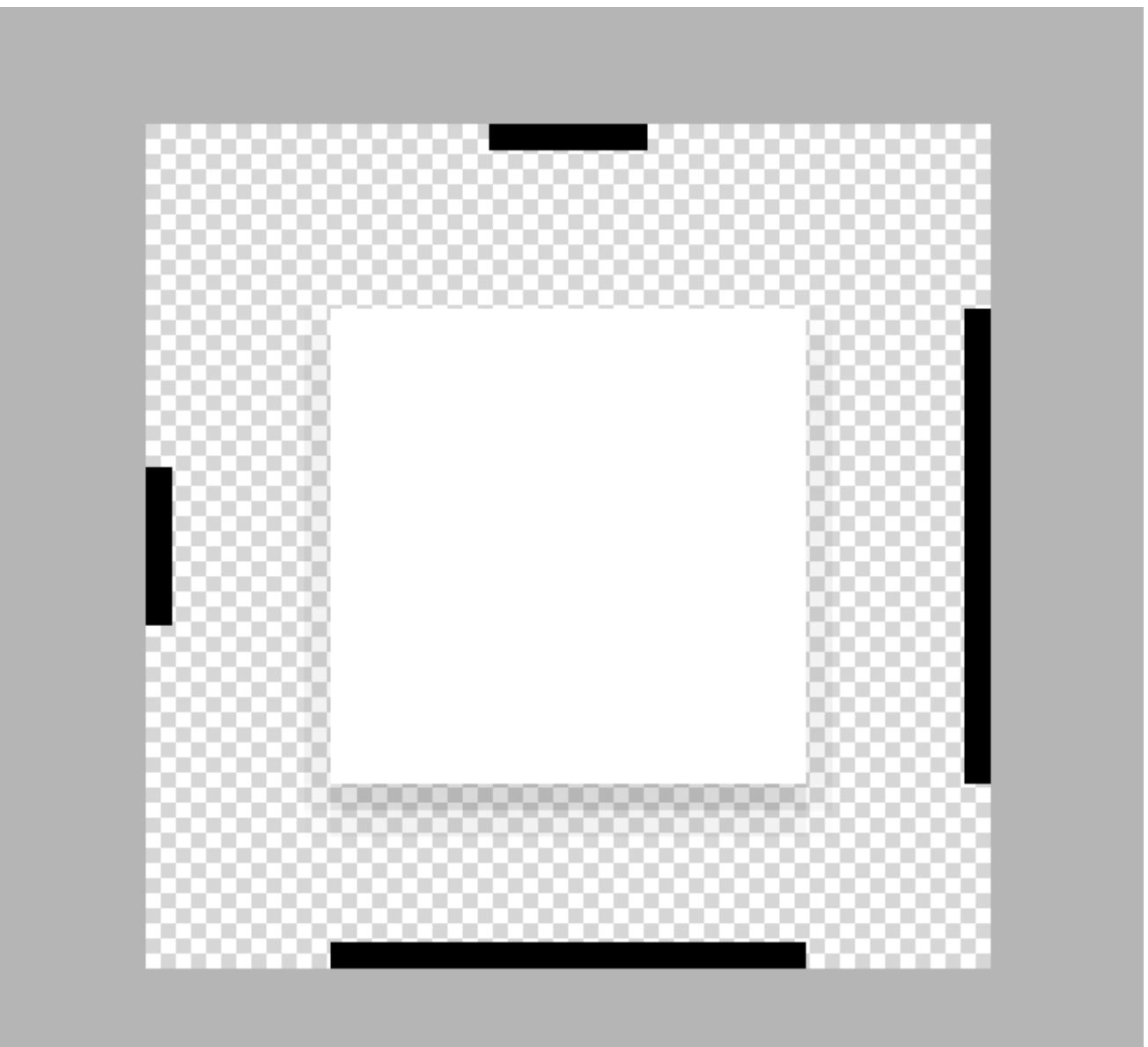
-Cyril Mottier

What Makes Drawables Awesome:

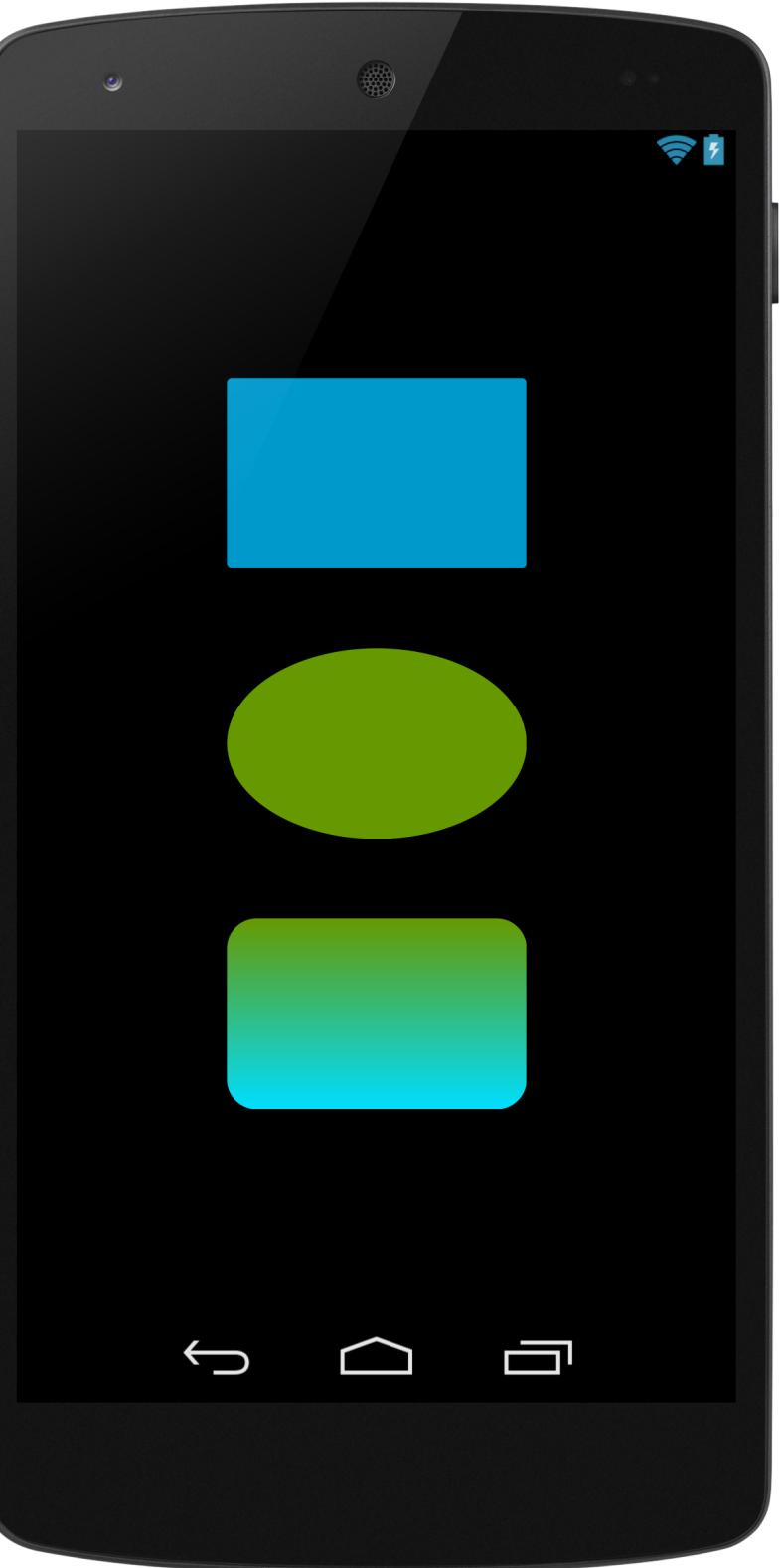
- Drawing is fun!
- Handling State Changes
- Separation of View Logic from Code
- XML
- Versatility



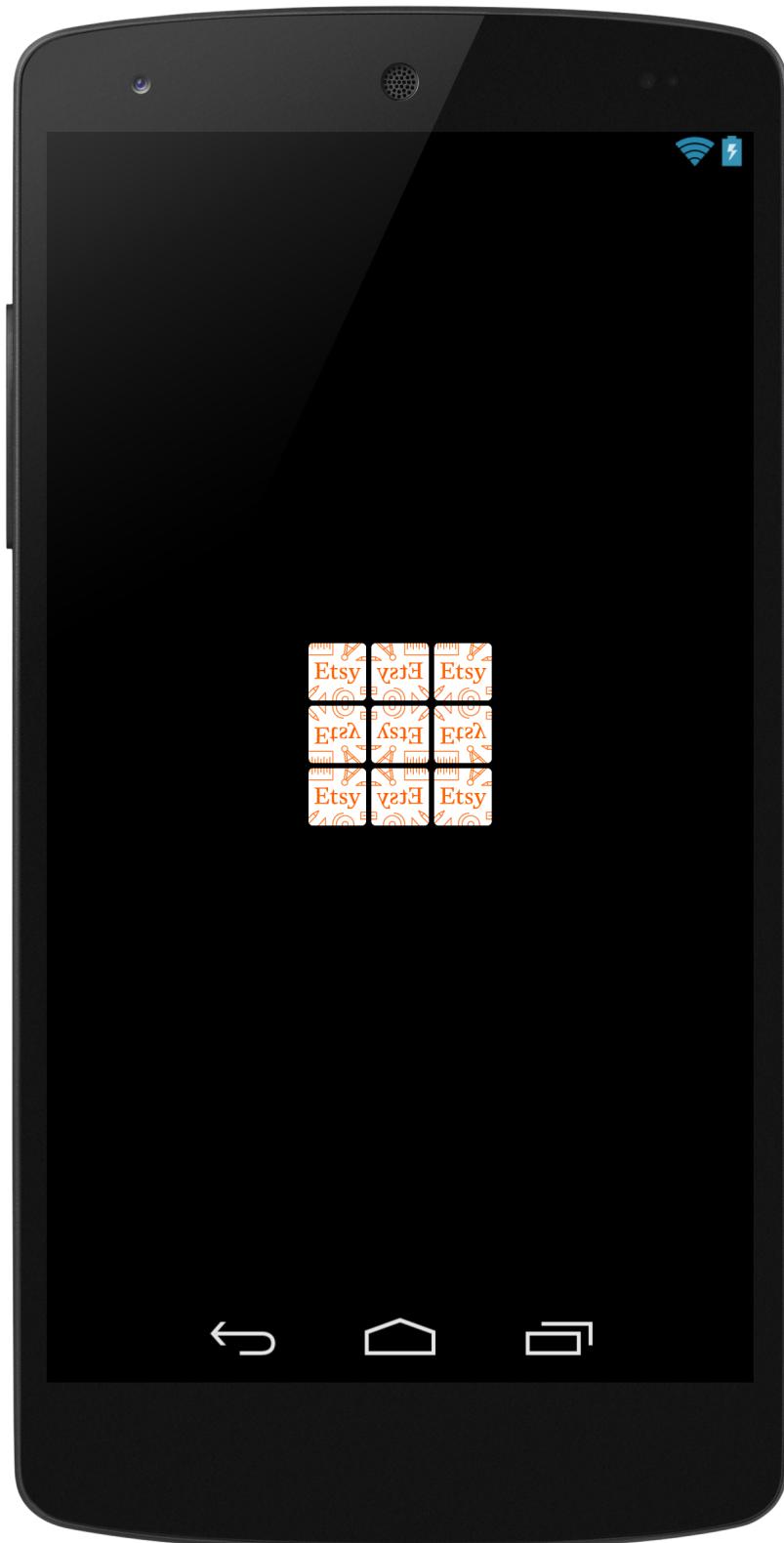
NinePatchDrawable



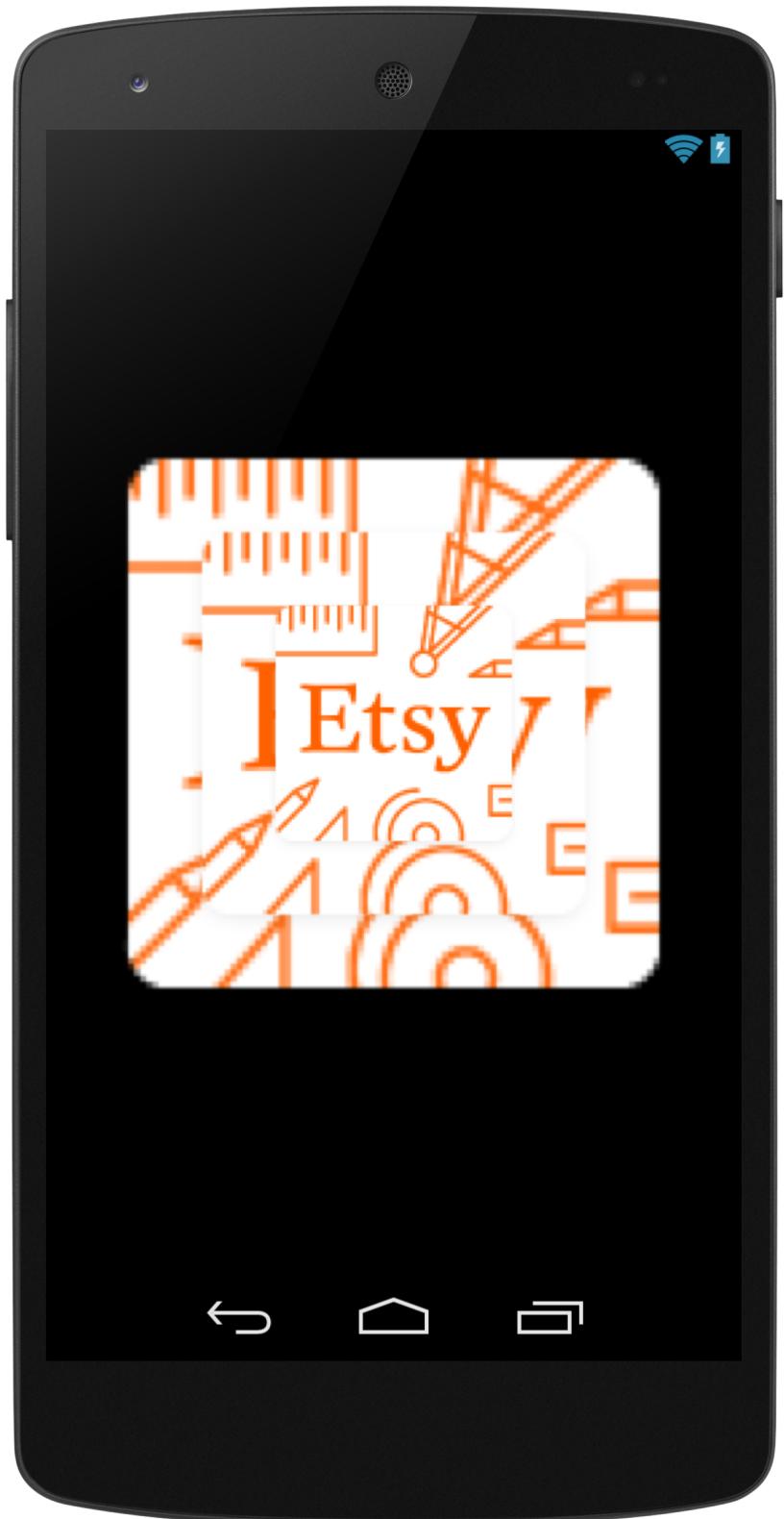
ShapeDrawable



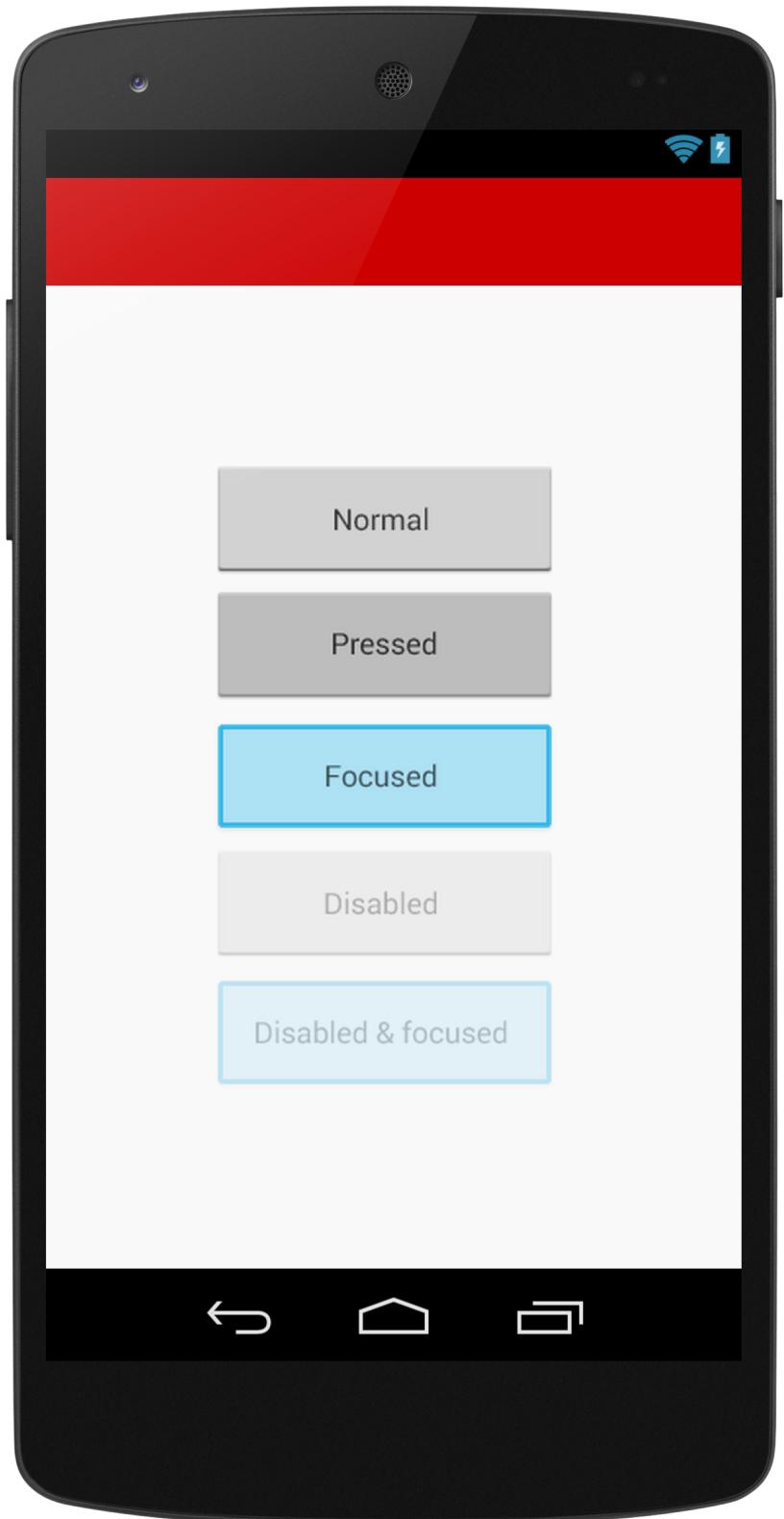
BitmapDrawable



LayerDrawable

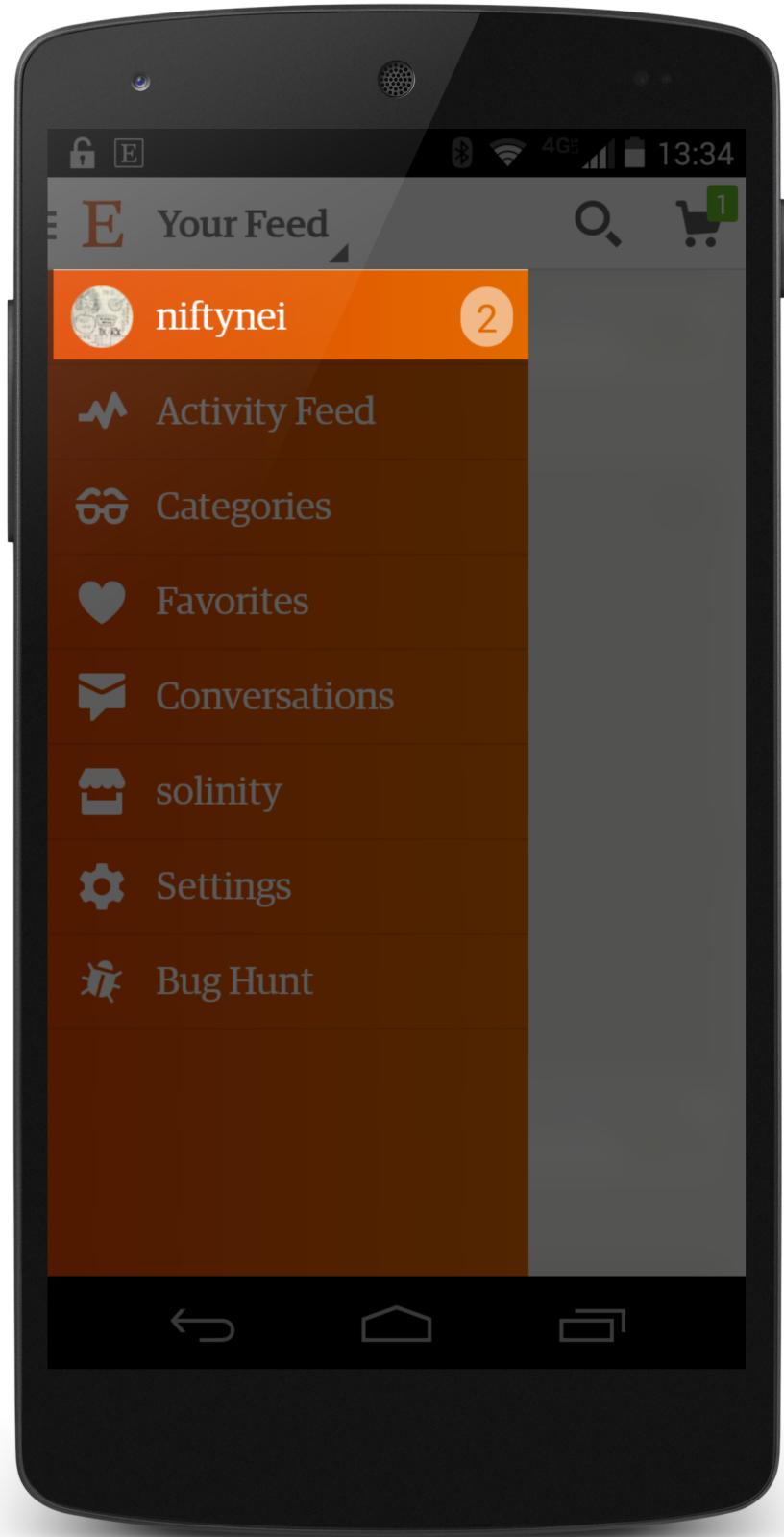


StateListDrawable

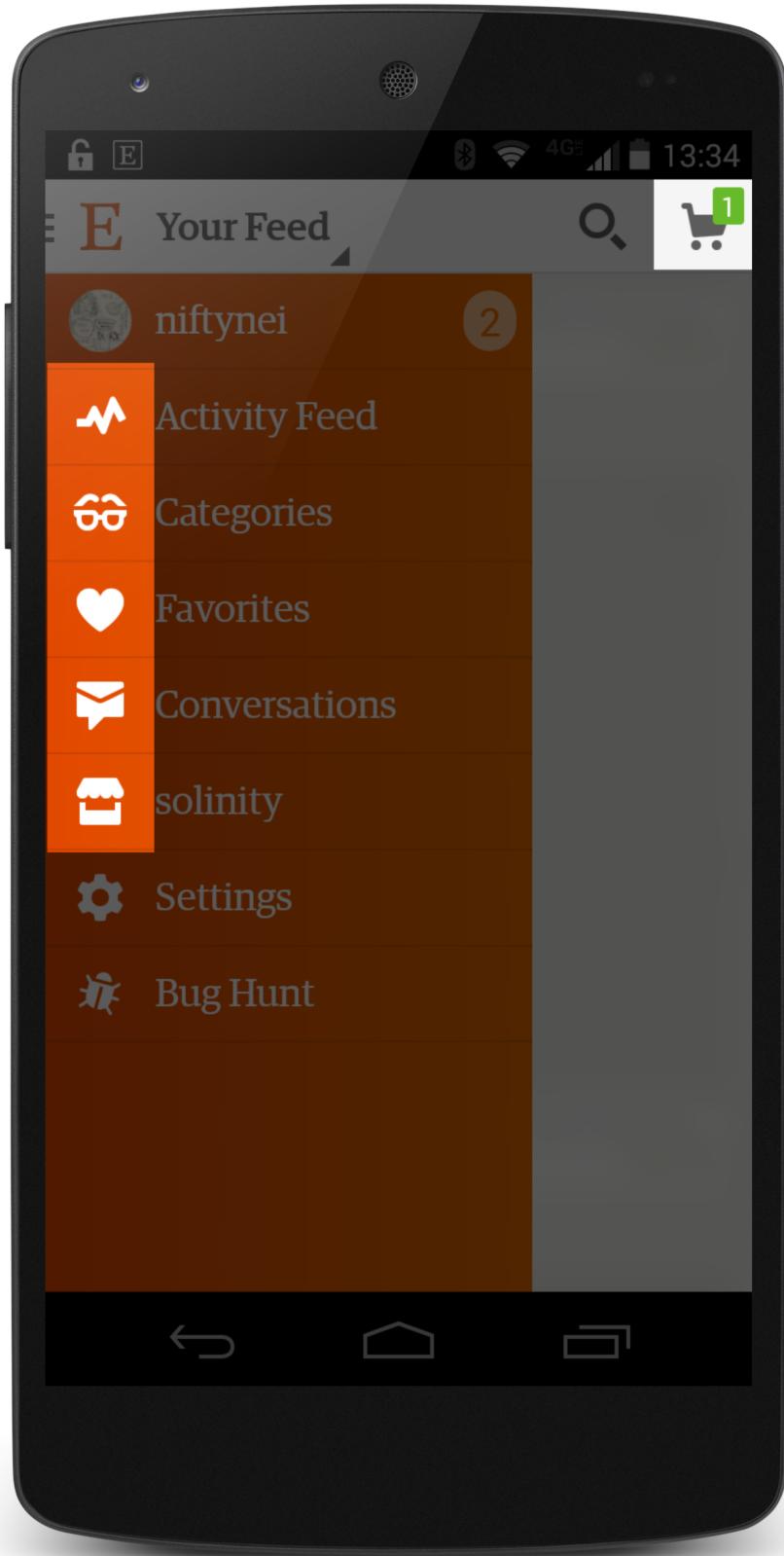


Custom Drawables

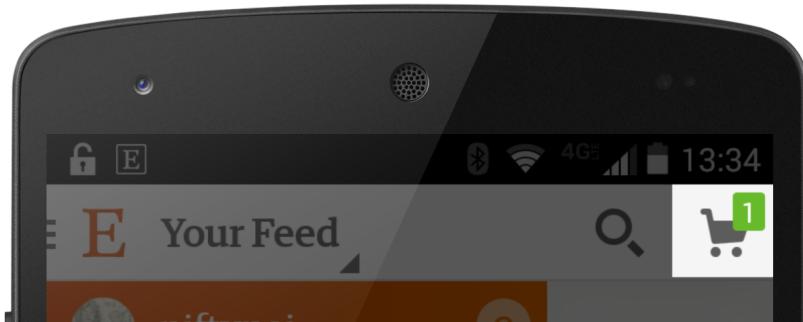
BadgeDrawable



FontDrawable



FontDrawable



.ss-shop
=🏪

shop



.ss-followshop
 

followshop
follow shop *



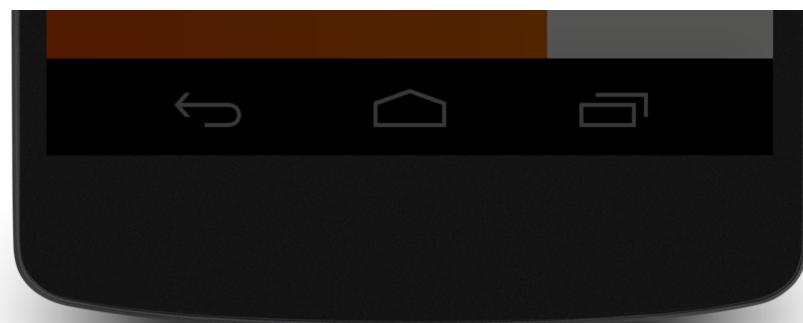
.ss-followingshop
 

following shop
following shop *



.ss-items
 

items



DRAW ALL THE
THINGS!



Markinbo 222

/res

.. /drawable

.. layered_drawable.xml

```
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <shape android:shape="oval">
            <solid android:color="@color/purple"/>
        </shape>
    </item>
    ...

```

layout.xml

```
<TextView  
    ...  
    android:background="@drawable/layered_drawable"/>
```

TOTAL JAVA CODE WRITTEN: \$0

Drawable Redux, vCustom.0

/res

..../drawable

..custom_drawable.xml

```
<com.droidcon.drawables.custom
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:icon="@+id/ic_etsy_e"
    app:color="@color/etsy_orange" >
</com.droidcon.drawables.custom>
```

layout.xml

```
<TextView  
    ...  
    android:background="@drawable/custom_drawable"/>
```

~~Process system is not responding~~

Unfortunately, System UI has stopped.

OK

Runtime Error

Caused by: org.xmlpull.v1.XmlPullParserException:
Binary XML file line #2: __invalid drawable tag custom__

Drawable.java

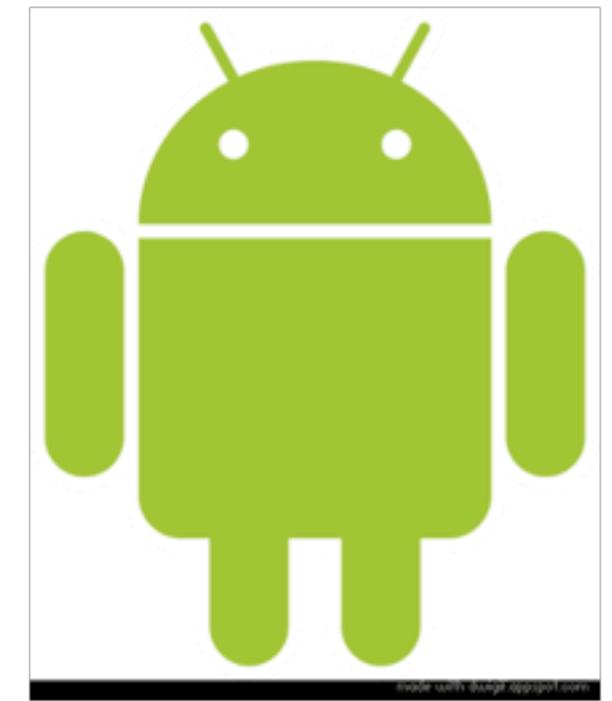
```
public static Drawable createFromXmlInner(Resources r, XmlPullParser parser, AttributeSet attrs)
throws XmlPullParserException, IOException {
    Drawable drawable;
```

```
final String name = parser.getName();

if (name.equals("selector")) {
    drawable = new StateListDrawable();
} else if (name.equals("layer-list")) {
    drawable = new LayerDrawable();
} else if (name.equals("shape")) {
    drawable = new GradientDrawable();
} else if ...
```

```
...
} else {
    throw new XmlPullParserException(parser.getPositionDescription() +
": invalid drawable tag " + name);
}
```

Seriously?



```
CustomDrawable drawable = new CustomDrawable();
drawable.setColorFilter(Color.BLACK, PorterDuff.Mode.DST);
mView.setBackground(drawable);
```

The Drawable API

```
public class NinjaDrawable extends ColorDrawable {
```

Methods you definitely want to @Override

- draw(Canvas canvas)

```
@Override  
draw(Canvas canvas) (onDraw)
```

- Just like a View's drawing.
- Use getBounds() to determine drawing area

Methods, good to know

- `getOpacity()`
- `getIntrinsicWidth()`
- `getMinimumWidth()`

getOpacity()

```
public int getOpacity() {  
    switch (mState.mUseColor >>> 24) {  
        case 255:  
            return PixelFormat.OPAQUE;  
        case 0:  
            return PixelFormat.TRANSPARENT;  
    }  
    return PixelFormat.TRANSLUCENT;  
}
```

getIntrinsicWidth() (onMeasure)

```
public int getIntrinsicWidth() {  
    return -1;  
}
```

.

```
public int getIntrinsicWidth() {  
    return mBitmap.getWidth();  
}
```

`getMinimumHeight\Width()` (`getSuggestedMinimumHeight()`)

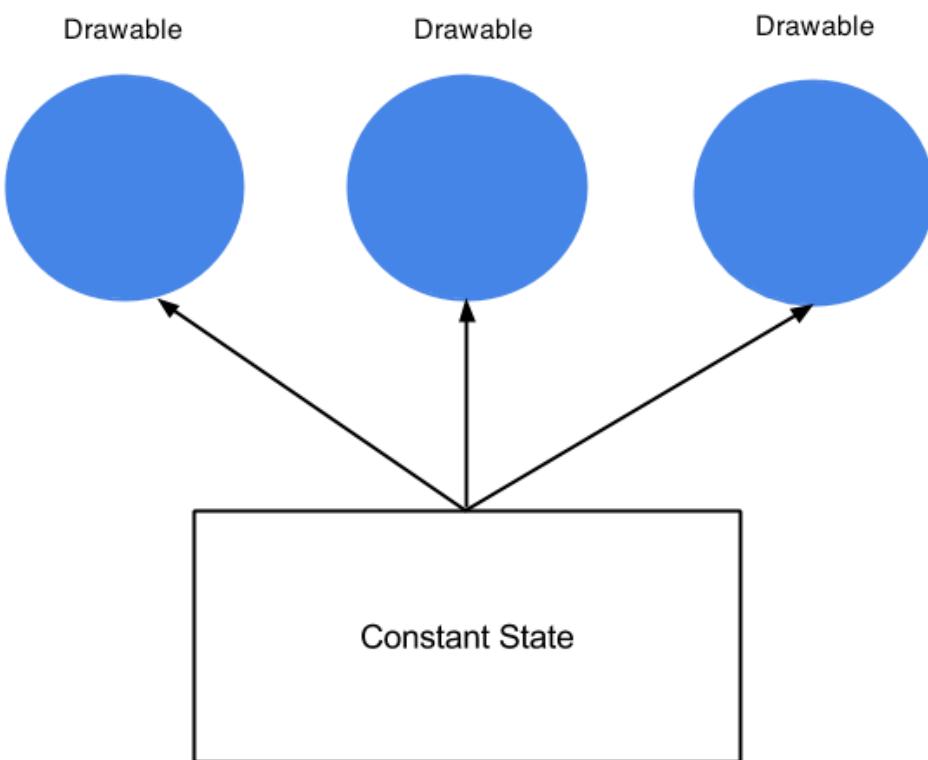
`getMinimumHeight\Width()` (`getSuggestedMinimumHeight()`)

Bottom line: implement `getIntrinsic`, not `getMinimum`.

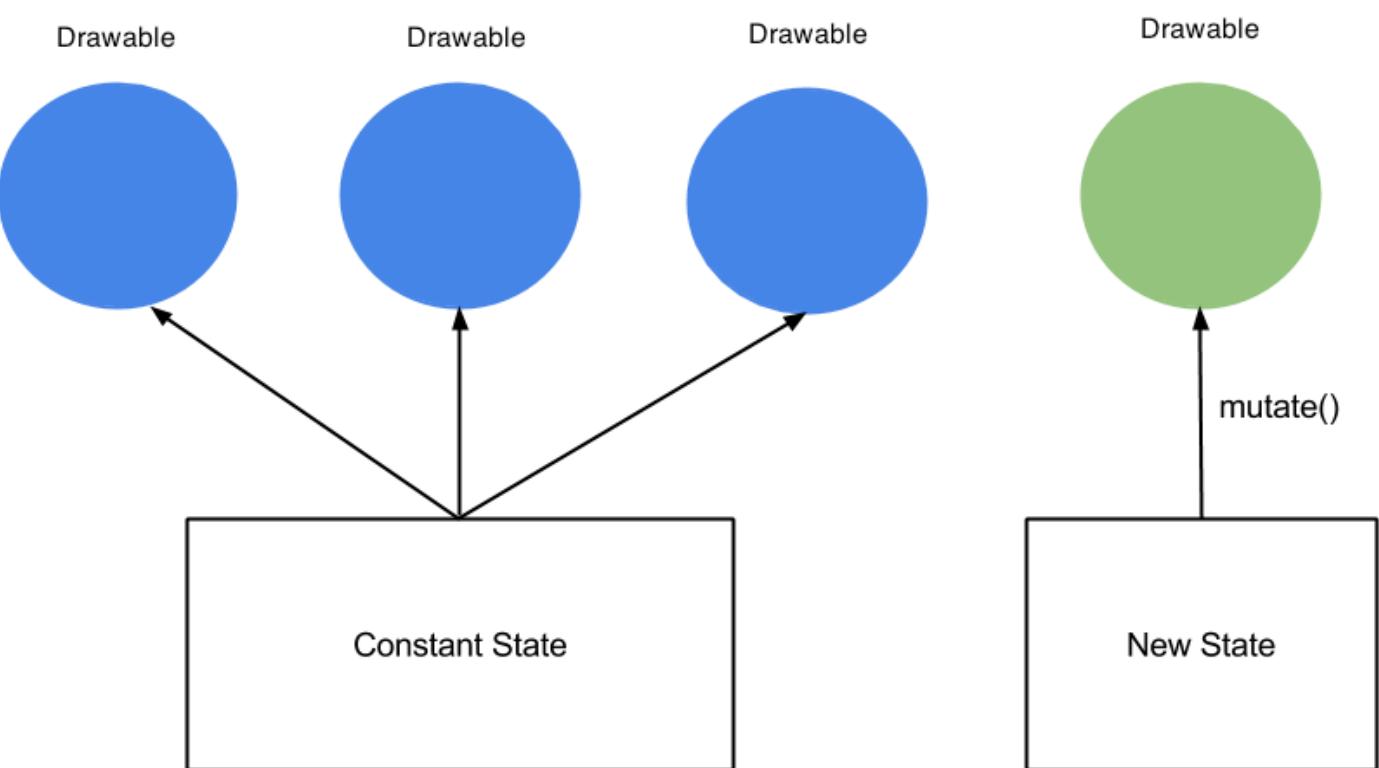
Methods, ninja level

- `getConstantState()` and `mutate()`

getConstantState()



mutate()



Drawables + Views

Drawable.Callback

```
public static interface Callback {  
  
    public void invalidateDrawable(Drawable who);  
  
    public void scheduleDrawable(Drawable who, Runnable what, long when);  
  
    public void unscheduleDrawable(Drawable who, Runnable what);  
  
}
```

```
public class View implements Drawable.Callback {
```

```
public void setBackground(Drawable d) {  
    mBackground = d;  
    d.setCallback(this);  
}
```

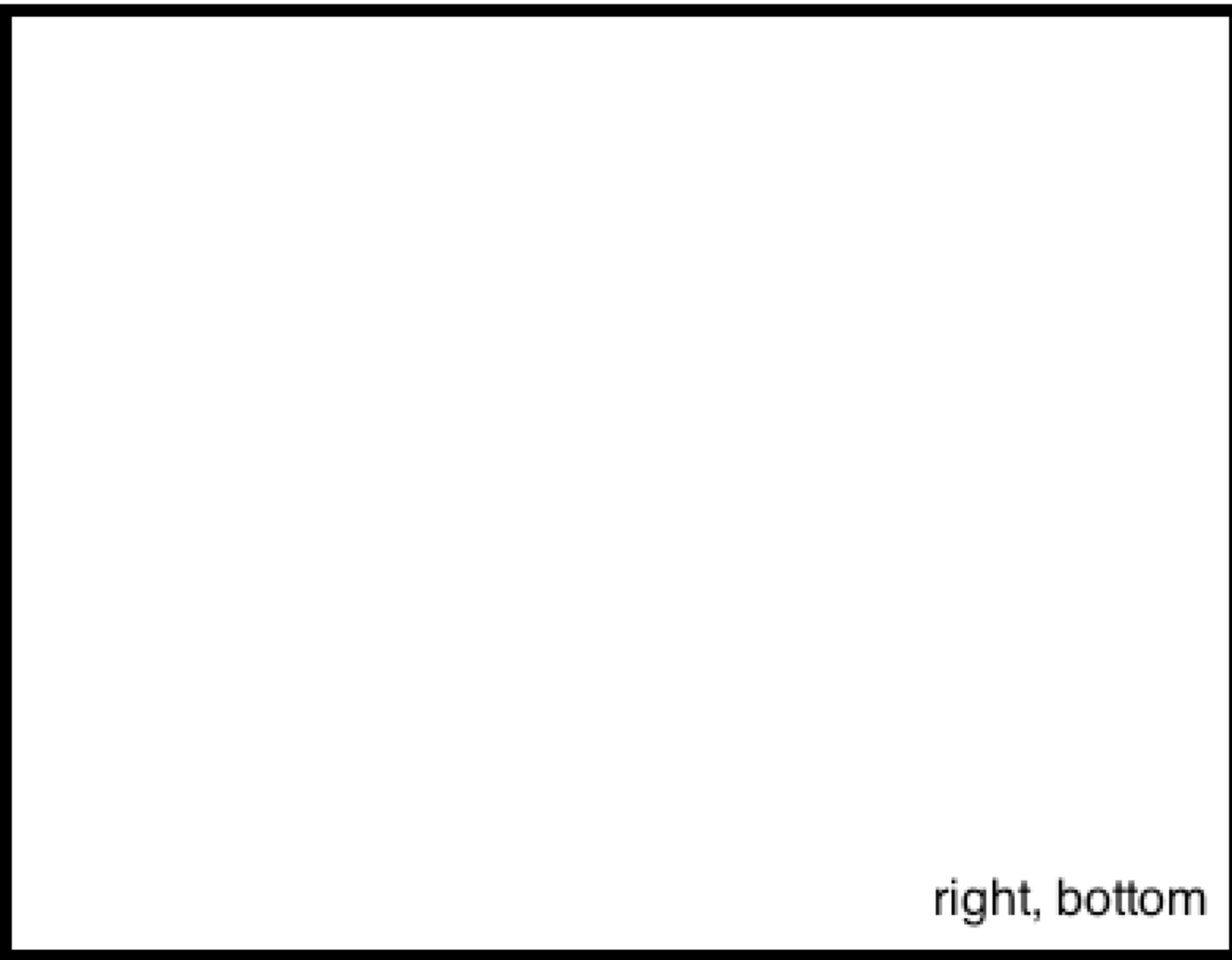
```
public void invalidateDrawable(Drawable drawable) {  
    ...  
    final Rect dirty = drawable.getBounds();  
    invalidate(dirty.left, dirty.top,  
              dirty.right, dirty.bottom);  
    ...
```

Drawable Bounds

Drawable.java

```
setBounds(Rect rect)
setBounds(int left, int top, int right, int
bottom)
```

0, 0



right, bottom

View

0, 0

10, 5

30, 45

Drawable

right, bottom

View

Binding it All Together

- Measurement
- Layout
- Draw

- ***Measurement***

- Layout

- ***Draw***

During **Measurement**

- View -> mDrawable.getIntrinsic*
- Calculates dimens, calls children, etc.
- View Measurements -> ParentView

Sometime before ***Draw***

- Calculates bounds for drawable
- `mDrawable.setBounds(...)`

During **Draw**

- `mDrawable.draw(canvas)`

Caveat Emptor

ColorDrawable.java

```
getIntrinsicHeight() { returns -1; }
```

CompoundButton.java

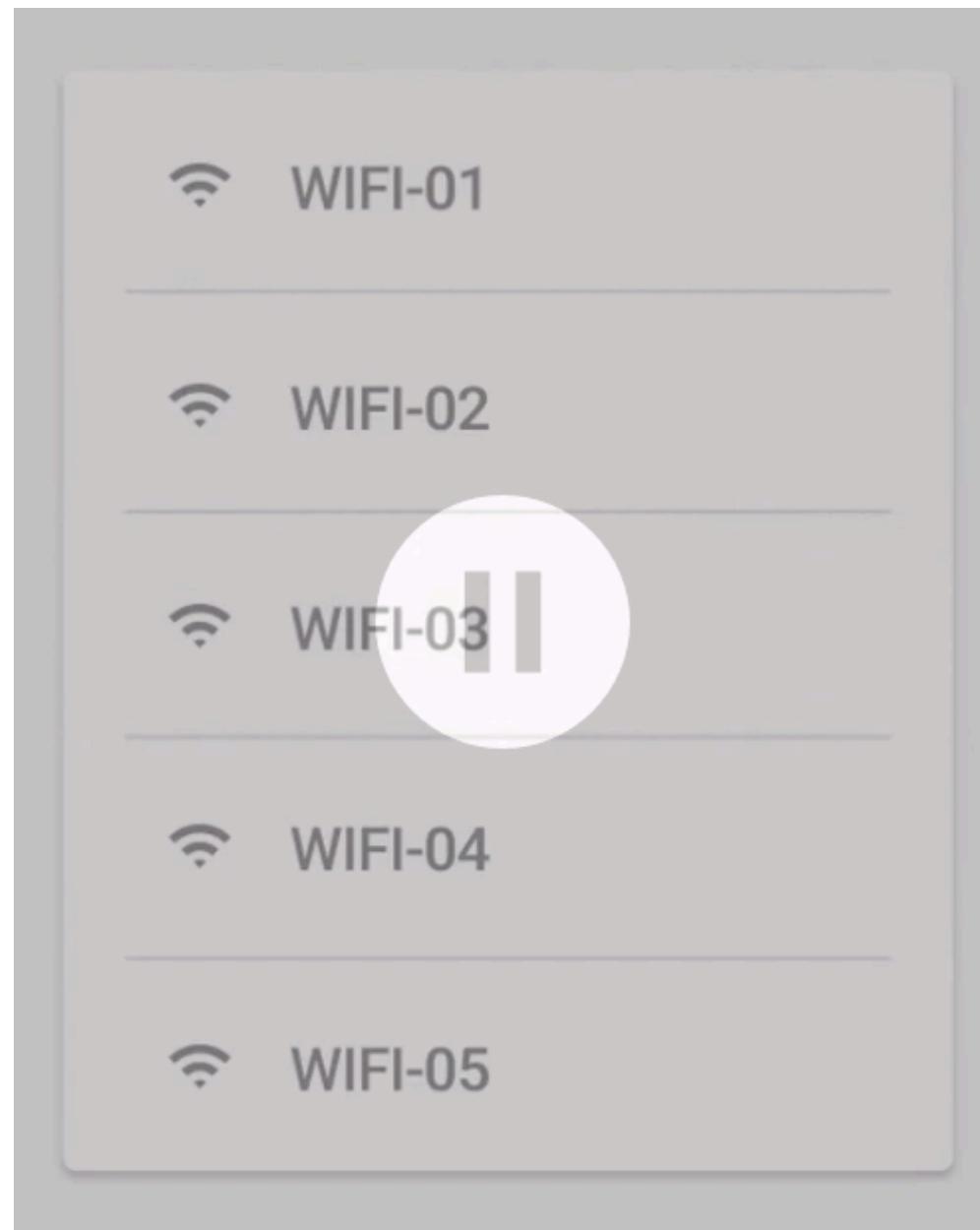
```
public void onDraw(Canvas canvas) {
    final int verticalGravity = getGravity() & Gravity.VERTICAL_GRAVITY_MASK;
    final int drawableHeight = buttonDrawable.getIntrinsicHeight();
    final int drawableWidth = buttonDrawable.getIntrinsicWidth();
    int top = 0;
    switch (verticalGravity) {
        case Gravity.BOTTOM:
            top = getHeight() - drawableHeight;
            break;
        case Gravity.CENTER_VERTICAL:
            top = (getHeight() - drawableHeight) / 2;
            break;
    }
    int bottom = top + drawableHeight;
    int left = isLayoutRtl() ? getWidth() - drawableWidth : 0;
    int right = isLayoutRtl() ? getWidth() : drawableWidth;
    buttonDrawable.setBounds(left, top, right, bottom);
    buttonDrawable.draw(canvas);
}
```

ImageView.java

```
if (dwidth <= 0 || dheight <= 0 || ScaleType.FIT_XY == mScaleType) {  
    /* If the drawable has no intrinsic size, or we're told to  
     * scaleToFit, then we just fill our entire view.  
    */  
    mDrawable.setBounds(0, 0, vwidth, vheight);  
    mDrawMatrix = null;  
} else {  
    // We need to do the scaling ourself, so have the drawable  
    // use its native size.  
    mDrawable.setBounds(0, 0, dwidth, dheight);  
  
    ...  
}
```

Lollipop

Ripple





Lollipop - Touch Events

```
public class Drawable {  
    ...  
  
    public void setHotspot(float x, float y) {}  
  
    public void setHotspotBounds(int, int, int, int) {}  
  
    ...  
}
```

Lollipop - Touch Events

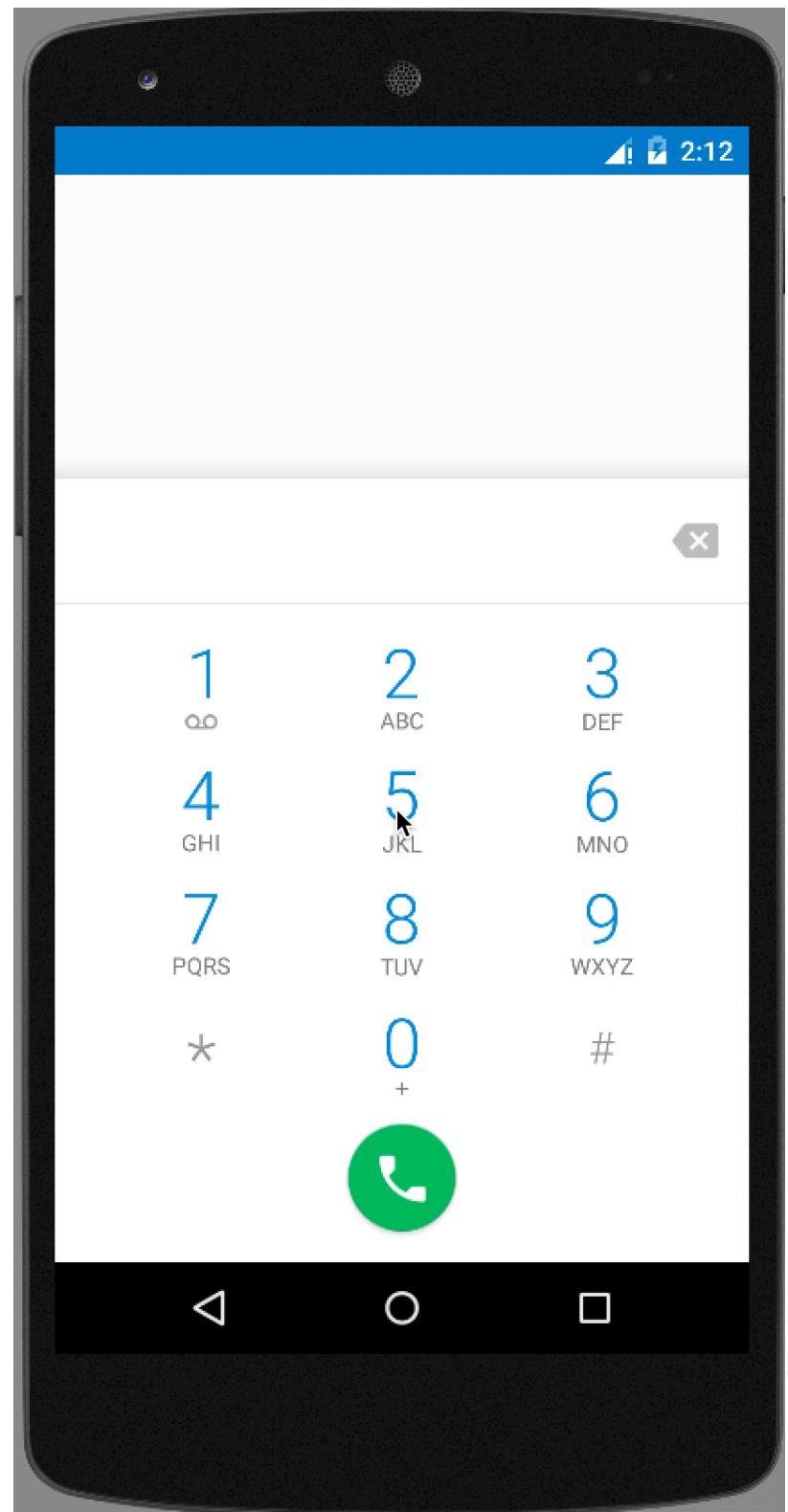
```
public class View {  
  
    ...  
  
    @Override  
    public void onTouchEvent(MotionEvent event) {  
        switch(event.getAction()) {  
            case MotionEvent.ACTION_MOVE:  
                drawableHotspotChanged(x, y);  
  
                ...  
            break;  
        }  
    }  
}
```

Lollipop - Outline

```
public class Drawable {  
    . . .  
  
    public void getOutline(Outline outline) {}  
  
    . . .  
}
```

Lollipop - Dirty Region

```
public class Drawable {  
    . . .  
  
    public Rect getDirtyBounds() {  
        return getBounds();  
    }  
  
    . . .  
}
```



Lollipop - Theme

```
public class Drawable {  
    ...  
  
    public void applyTheme(Theme) {}  
  
    public boolean canApplyTheme() {}  
  
    public Drawable createFromXml(Resources, XmlPullParser, Theme)  
  
    public Drawable createFromXmlInner(Resources, XmlPullParser, AttributeSet, Theme)  
  
    public void inflate(Resources, XmlPullParser, AttributeSet, Theme)  
    ...  
}
```

Lollipop - Tint

```
public class Drawable {  
    ...  
  
    public void setTint(int)  
  
    public void setTintList(ColorStateList list)  
  
    public void setTintMode(Mode)  
  
    ...  
}
```

Lollipop - ConstantState

```
public class ConstantState {  
  
    public boolean canApplyTheme()  
  
    public Drawable newDrawable(Resources, Theme)  
  
}
```

VectorDrawable

Backwards Compatibility

MrVector <https://github.com/telly/MrVector>

Ripple



Ripple

```
public class RippledImageView extends ImageView {  
    ...  
  
    @Override  
    public boolean onTouchEvent(MotionEvent event) {  
        updateHotspot(event.getX(), event.getY());  
        return super.onTouchEvent(event);  
    }  
  
    private void updateHotspot(float x, float y) {  
  
        Drawable background = getBackground();  
        if (background != null && background instanceof RippleDrawableCompat) {  
            ((RippleDrawableCompat) background).setHotspot(x, y);  
        }  
  
        Drawable src = getDrawable();  
        if (src != null && src instanceof RippleDrawableCompat) {  
            ((RippleDrawableCompat) src).setHotspot(x, y);  
        }  
    }  
}
```

Ripple

```
@Override  
public void draw(@NonNull Canvas canvas) {  
    // Clip to the dirty bounds, which will be the drawable bounds if we  
    // have a mask or content and the ripple bounds if we're projecting.  
    final Rect bounds = getDirtyBounds();  
    final int saveCount = canvas.save(Canvas.CLIP_SAVE_FLAG);  
    canvas.clipRect(bounds);  
  
    drawContent(canvas);  
    drawBackgroundAndRipples(canvas);  
  
    canvas.restoreToCount(saveCount);  
}
```

Ripple

```
public boolean draw(Canvas c, Paint p) {
    final boolean canUseHardware = c.isHardwareAccelerated();
    if (mCanUseHardware != canUseHardware && mCanUseHardware) {
        // We've switched from hardware to non-hardware mode. Panic.
        cancelHardwareAnimations(true);
    }

    mCanUseHardware = canUseHardware;
    final boolean hasContent;
    if (canUseHardware && (mHardwareAnimating || mHasPendingHardwareExit)) {
        hasContent = drawHardware((HardwareCanvas) c, p);
    } else {
        hasContent = drawSoftware(c, p);
    }

    return hasContent;
}
```

Ripple

Search:

"Android Graphics Pipeline Button to FrameBuffer"

AOSP:

<https://source.android.com/devices/graphics/index.html>

- ▼  **res**
 -  **anim**
 -  **drawable**
 -  **drawable-hdpi**
 -  **drawable-ldpi**
 -  **drawable-mdpi**
 -  **drawable-sw600dp-hdpi**
 -  **drawable-sw600dp-xhdpi**
 -  **drawable-v21**
 -  **drawable-xhdpi**
 -  **drawable-xxhdpi**
 -  **drawable-xxxhdpi**

Thank You!

@JamieHuson ^.^ @niftynei