droidcon
LONDON 28-29 OCT 2010

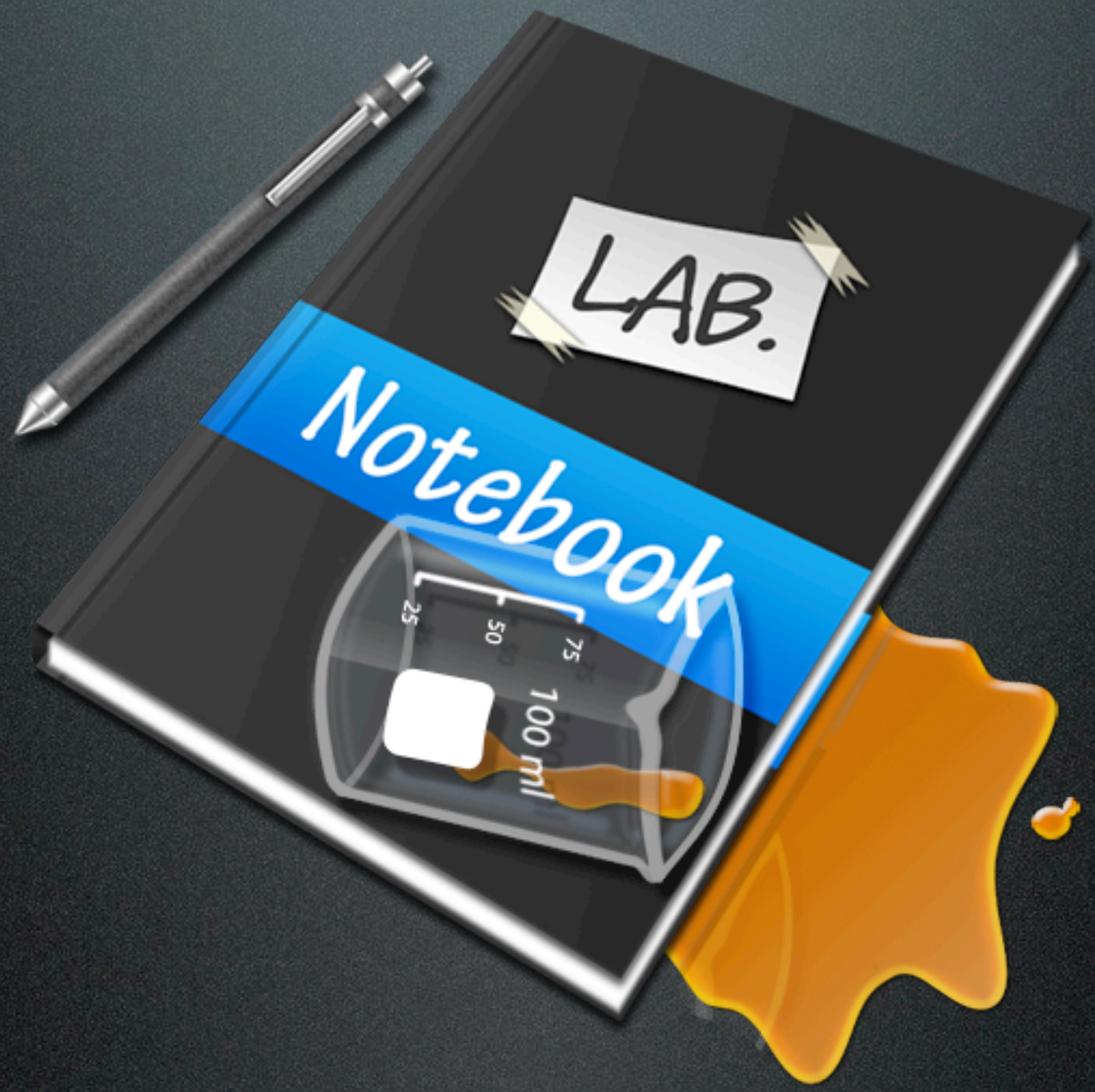
greencopper

Cyril Mottier

Mastering Android Drawables

Drawable are powerful so use them !

Global note



droidcon
LONDON 28-29 OCT 2010

Introduction

Drawables 101

Introduction

What's a Drawable ?

Introduction

What's a Drawable ?

- Entity that can be « drawn »
  - A shape, a gradient, an image, a 9-patch, etc.

Introduction

What's a Drawable ?

- Entity that can be « drawn »
  - A shape, a gradient, an image, a 9-patch, etc.
- Essential component of the Android SDK

Introduction

What's a Drawable ?

- Entity that can be « drawn »
  - A shape, a gradient, an image, a 9-patch, etc.
- Essential component of the Android SDK
- May be created using Java or XML

Introduction

Why is it so important?

- Easy multi-device management
- Abstract representation of a drawable entity
  - Easy to use
  - Way more evolved than Bitmaps!
- UI widget theming/styling without inheritance

Introduction

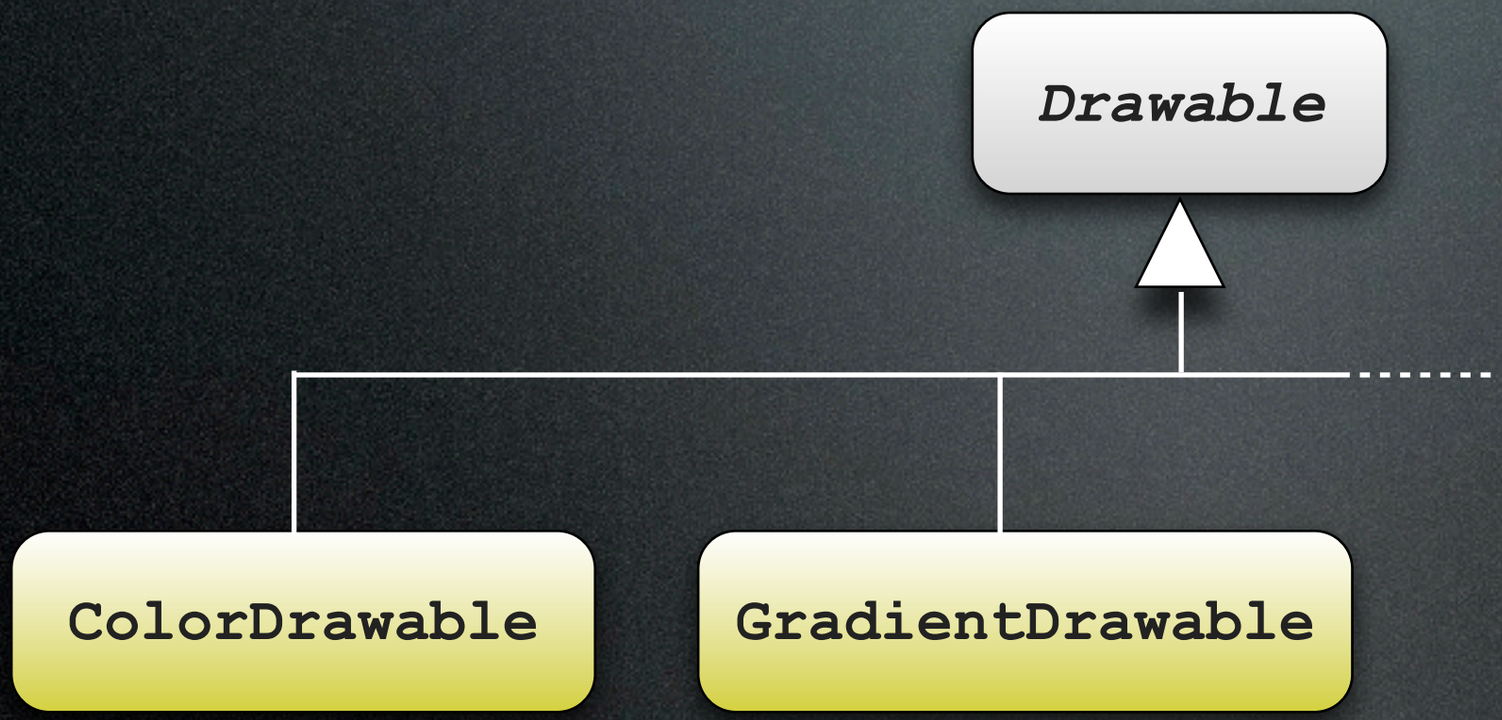
Architecture

Introduction

Architecture

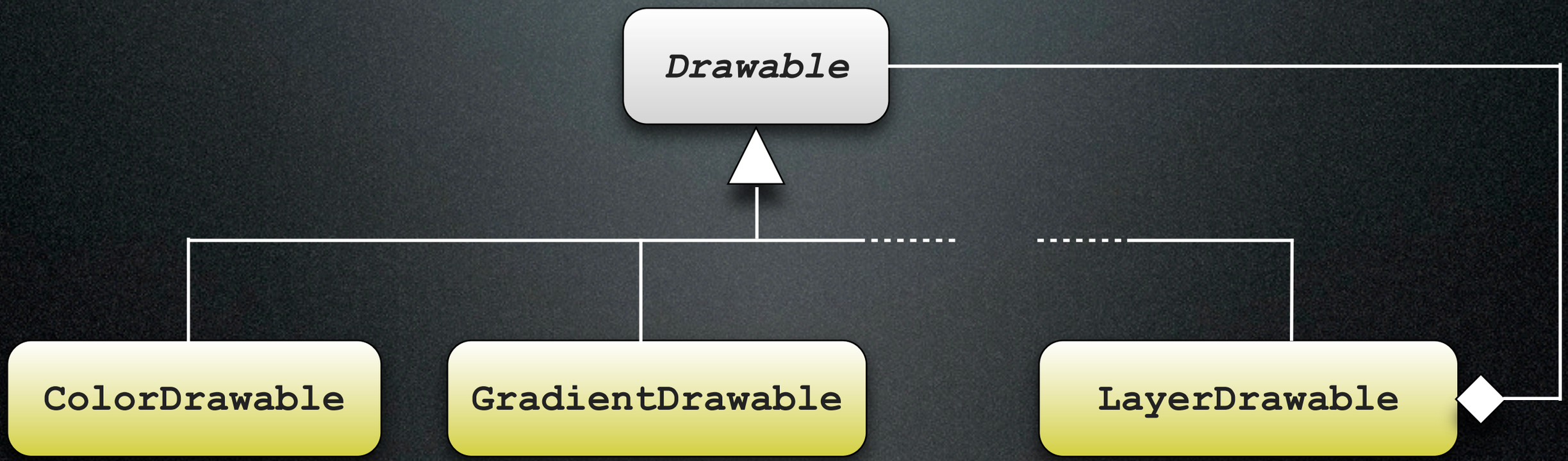
Introduction

Architecture



Introduction

Architecture



droidcon
LONDON 28-29 OCT 2010

Basic « How-to »

Hello Drawables !

Basic « How-to »

First try

```
private Drawable mDrawable;

public DrawableView(Context context, AttributeSet attrs, int defStyle) {
    super(context, attrs, defStyle);
    mDrawable = context.getResources().getDrawable(R.drawable.my_drawable);
}

public void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    mDrawable.draw(canvas);
}
```

Basic « How-to »

What the hell ? It's not working !

Basic « How-to »

What the hell ? It's not working !

- My Canvas is not in a correct state?

Basic « How-to »

What the hell ? It's not working !

- My Canvas is not in a correct state?
- I forgot to attach the view to my layout?

Basic « How-to »

What the hell ? It's not working !

- My Canvas is not in a correct state?
- I forgot to attach the view to my layout?
- .... I haven't set the size of my Drawable

Basic « How-to »

What the hell ? It's not working !

- My Canvas is not in a correct state?
- I forgot to attach the view to my layout?
- .... I haven't set the size of my Drawable
  - setBounds(), getIntrinsic[Width/Height]()

Basic « How-to »

The correct way !

```
private Drawable mDrawable;

public DrawableView(Context context, AttributeSet attrs, int defStyle) {
    super(context, attrs, defStyle);
    mDrawable = context.getResources().getDrawable(R.drawable.my_drawable);
    mDrawable.setBounds(0, 0, mDrawable.getIntrinsicWidth(),
mDrawable.getIntrinsicHeight());
}

public void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    mDrawable.draw(canvas);
}
```

Features

What are Drawables useful for?

Some features

Miscellaneous

- Origin / Size :
  - setBounds(), getIntrinsic[Width/Height](), getMinimum[Width/Height](), etc.
- Current state :
  - setState(), state_[pressed/focused/...], setLevel(), etc.
- Graphic : setAlpha(), setDither(), ...

Some features

Drawable.Callback

- A Drawable should be able to invalidate itself
- The Drawable.Callback interface
- View implements Drawable.Callback
  - Helper method: verifyDrawable()

Some features

Drawable.Callback

```
public class DrawableView extends View {

    private Drawable mDrawable;

    public DrawableView(Context context, AttributeSet attrs, int defStyle) {
        super(context, attrs, defStyle);
        mDrawable = context.getResources().getDrawable(R.drawable.my_drawable);
        mDrawable.setBounds(0, 0, mDrawable.getIntrinsicWidth(),
mDrawable.getIntrinsicHeight());
    }

    @Override
    protected boolean verifyDrawable(Drawable who) {
        return who == mDrawable || super.verifyDrawable(who);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        mDrawable.draw(canvas);
    }
}
```

Constant state principle

Show me the problem

```
private static final int OPAQUE = 255;
private static final int TRANSLUCENT = 70;

Book book = ...;
TextView listItem = ...;

listItem.setText(book.getTitle());

Drawable star = getContext().getResources().getDrawable(R.drawable.star);
if (book.isFavorite()) {
  star.setAlpha(OPAQUE);
} else {
  star.setAlpha(TRANSLUCENT);
}
```

Extracted from http://www.curious-creature.org/2009/05/02/drawable-mutations/

Constant state principle

Phenomenon explanation

- Android factorize Drawables' states
  - Reduce memory foot-print.

Constant state principle

Phenomenon explanation

Constant state principle

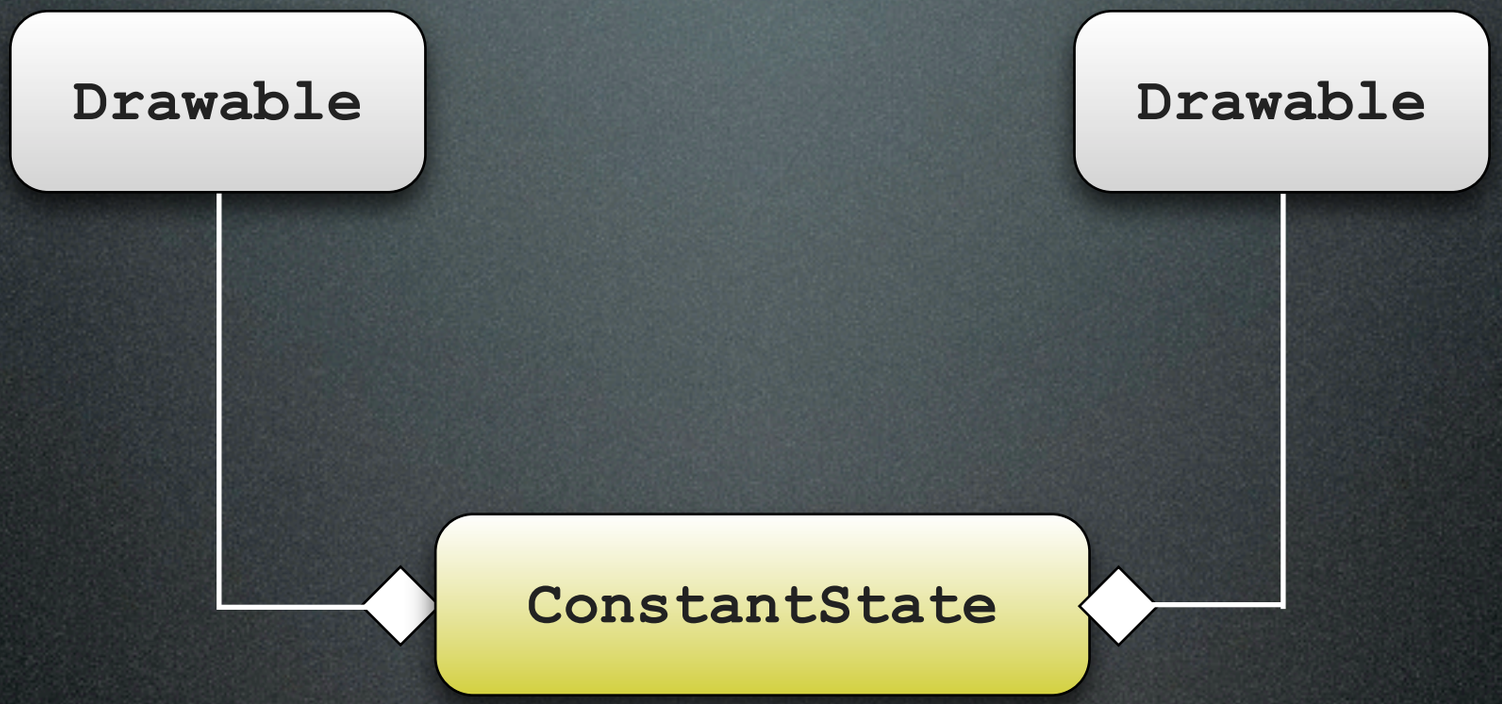
Phenomenon explanation

`Drawable`

`Drawable`

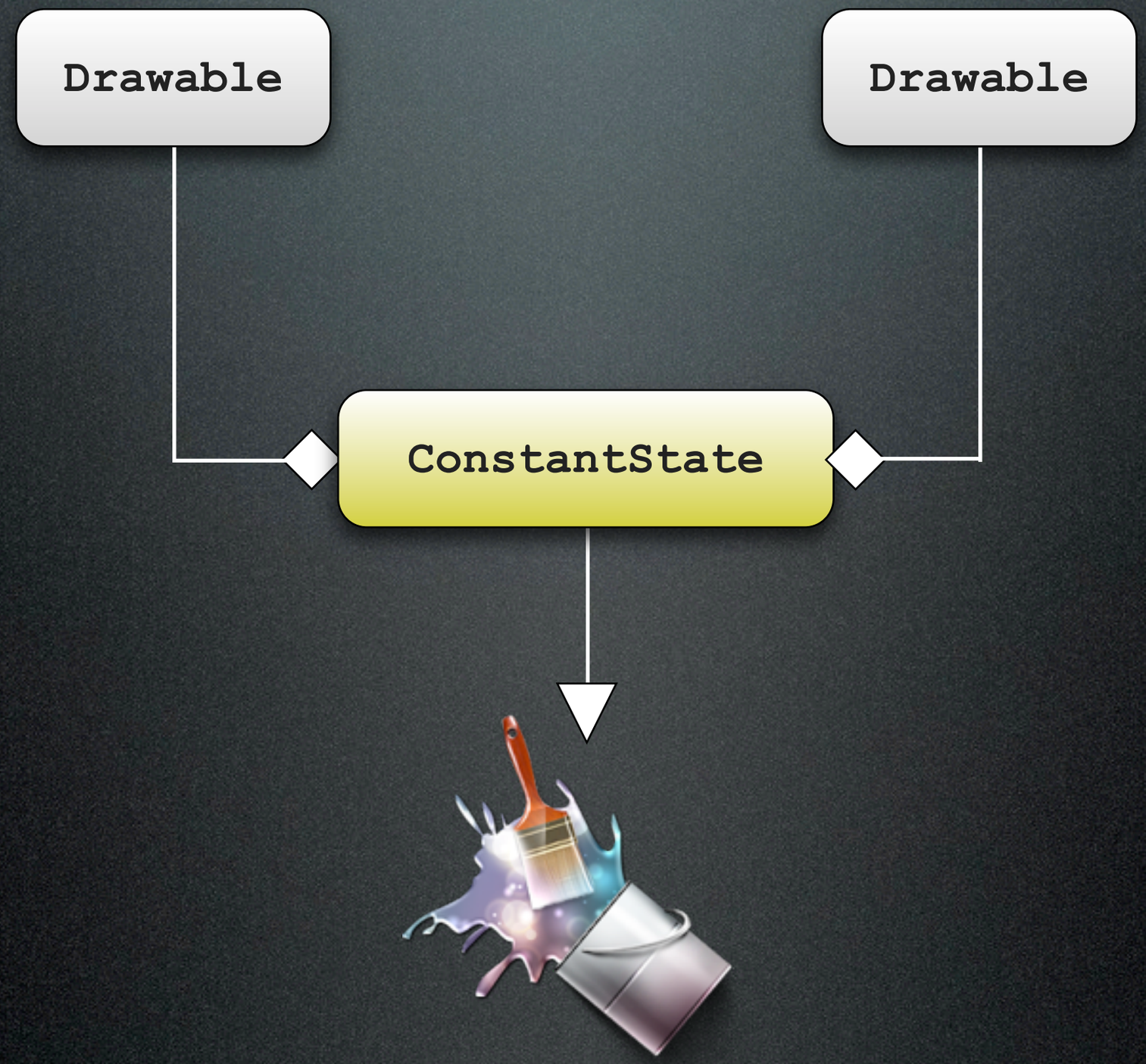
Constant state principle

Phenomenon explanation



Constant state principle

Phenomenon explanation



Constant state principle

Problem resolution

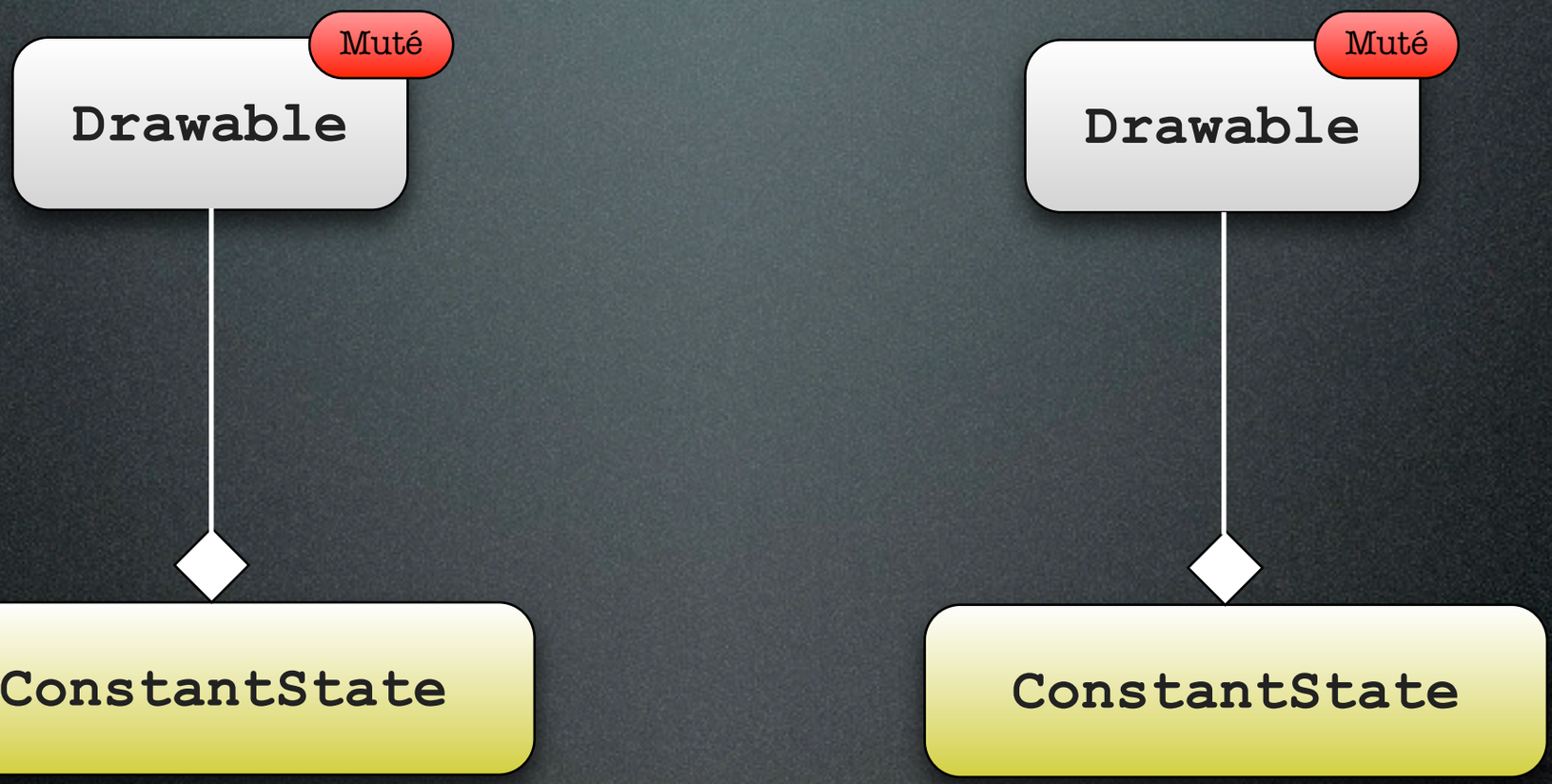
- No « copy-on-write »
- Use mutate() :
  - Copy the constant state
  - Set the mMutated flag to true
    - Multiple calls to mutated() effectless

Constant state principle

Problem resolution

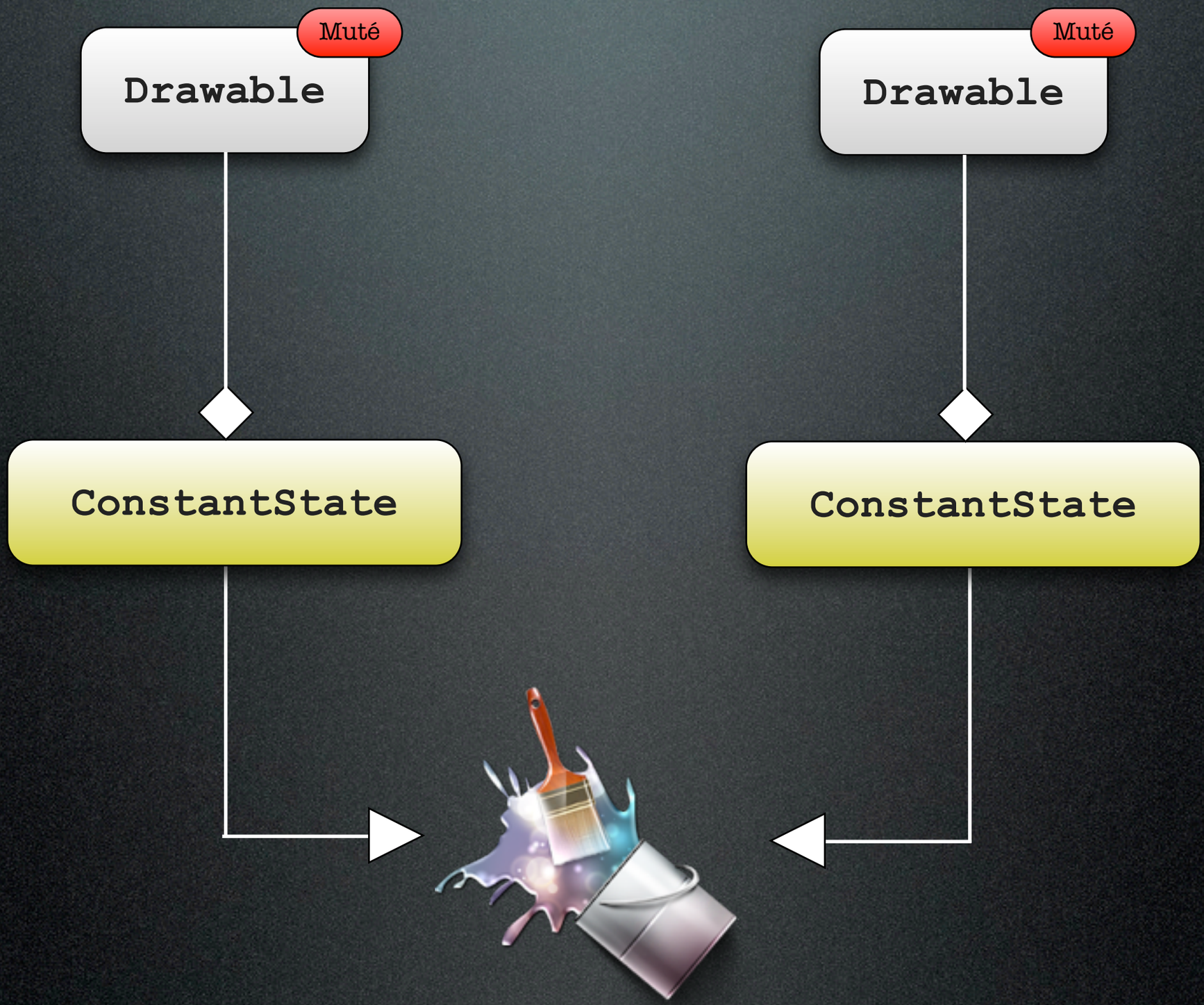
Constant state principle

Problem resolution



Constant state principle

Problem resolution



Constant state principle

Problem resolution

```
private static final int OPAQUE = 255;
private static final int TRANSLUCENT = 70;

Book book = ...;
TextView listItem = ...;

listItem.setText(book.getTitle());

Drawable star = getContext().getResources().getDrawable(R.drawable.star);
if (book.isFavorite()) {
  star.mutate().setAlpha(OPAQUE);
} else {
  star.mutate().setAlpha(TRANSLUCENT);
}
```

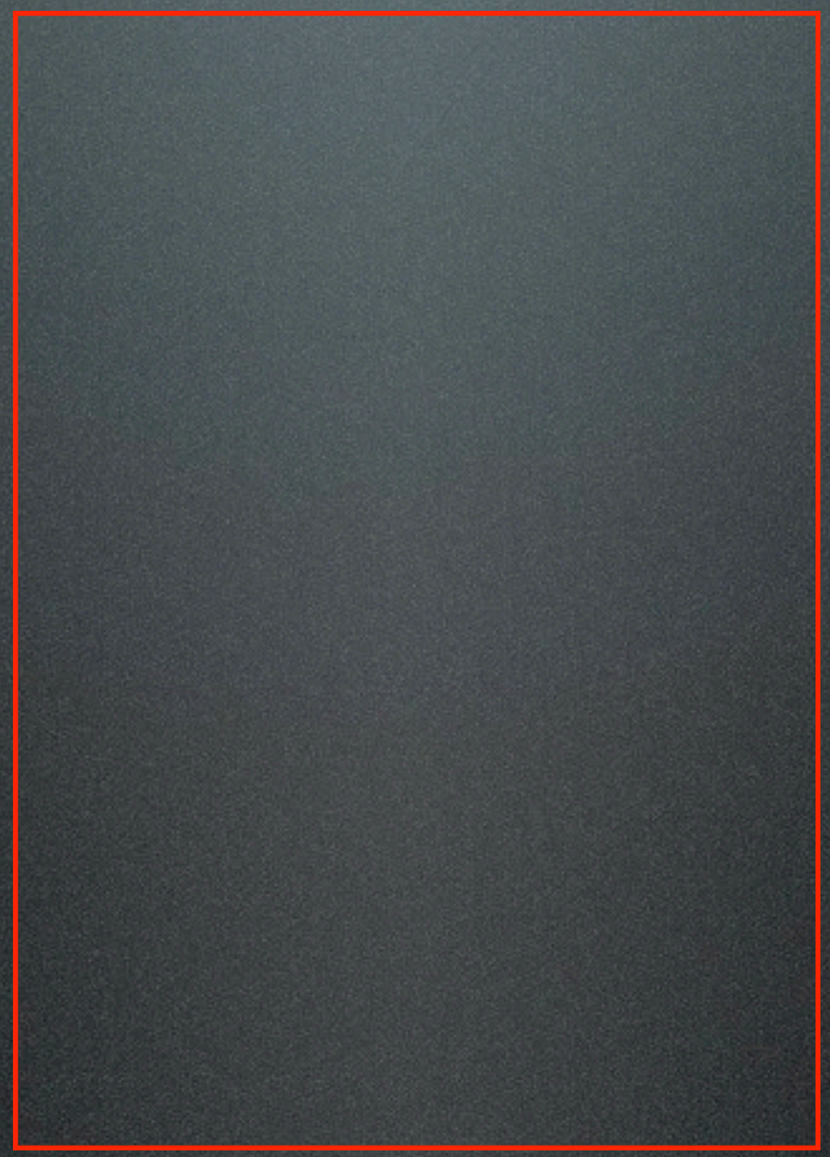
Extracted from http://www.curious-creature.org/2009/05/02/drawable-mutations/

Instantiation

Let's start coding !

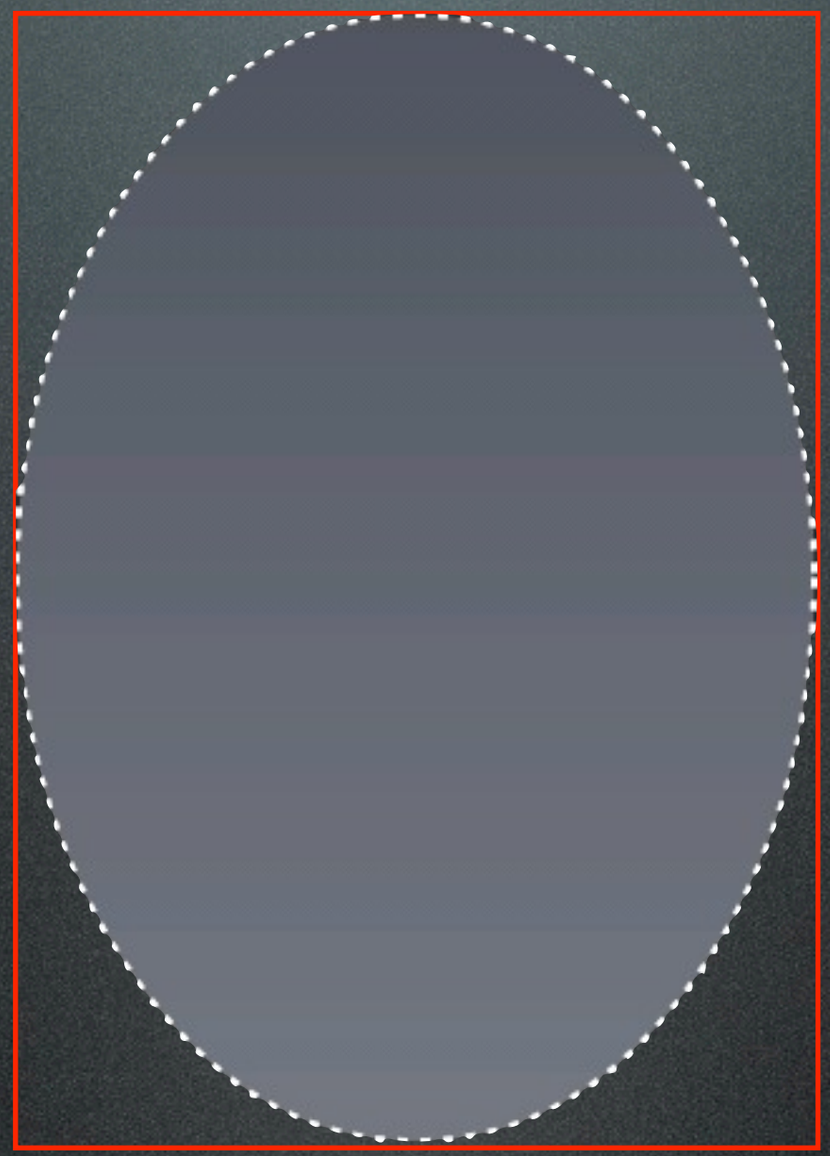
Instantiate a Drawable

What we want



Instantiate a Drawable

What we want



Instantiate a Drawable

Two ways: Java ...

```
float density = getContext().getResources().getDisplayMetrics().density;

// gradient from #4e525c to #31343c
int[] colors = new int[] {Color.rgb(78, 82, 92), Color.rgb(49, 52, 60)};

GradientDrawable gradient = new GradientDrawable(Orientation.BOTTOM_TOP,
colors);
gradient.setShape(GradientDrawable.OVAL);
gradient.setStroke((int) (3 * density), Color.WHITE, 4 * density, 5 *
density);
```

Instantiate a Drawable

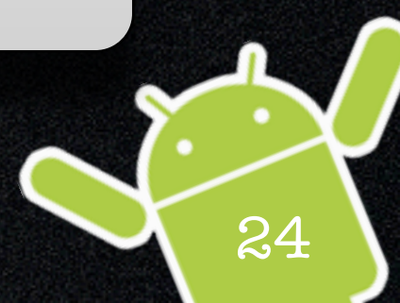
... and XML

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="oval">

    <gradient
        android:startColor="#4e525c"
        android:endColor="#31343c"
        android:angle="90" />

    <stroke
        android:width="3dip"
        android:color="#fff"
        android:dashWidth="4dip"
        android:dashGap="5dip" />

</shape>
```



Drawable Listing

Code available on
http://www.github.com/cyrilmottier/DrawablePresentation

BitmapDrawable

- Widely used
- But unknown / misknown
- Often use « as it »
  - context.getResources().getDrawable(R.drawable.image) (where image is a PNG or JPG file)
- Advanced features available
  - tileMode, gravity, etc.

BitmapDrawable

android:tileMode

```
<?xml version="1.0" encoding="utf-8"?>

<bitmap
    xmlns:android="http://
schemas.android.com/apk/res/android"
    android:src="@drawable/pattern" />
```

droidcon
LONDON 28-29 OCT 2010

BitmapDrawable

android:tileMode

```
<?xml version="1.0" encoding="utf-8"?>

<bitmap
    xmlns:android="http://
schemas.android.com/apk/res/android"
    android:src="@drawable/pattern" />
```

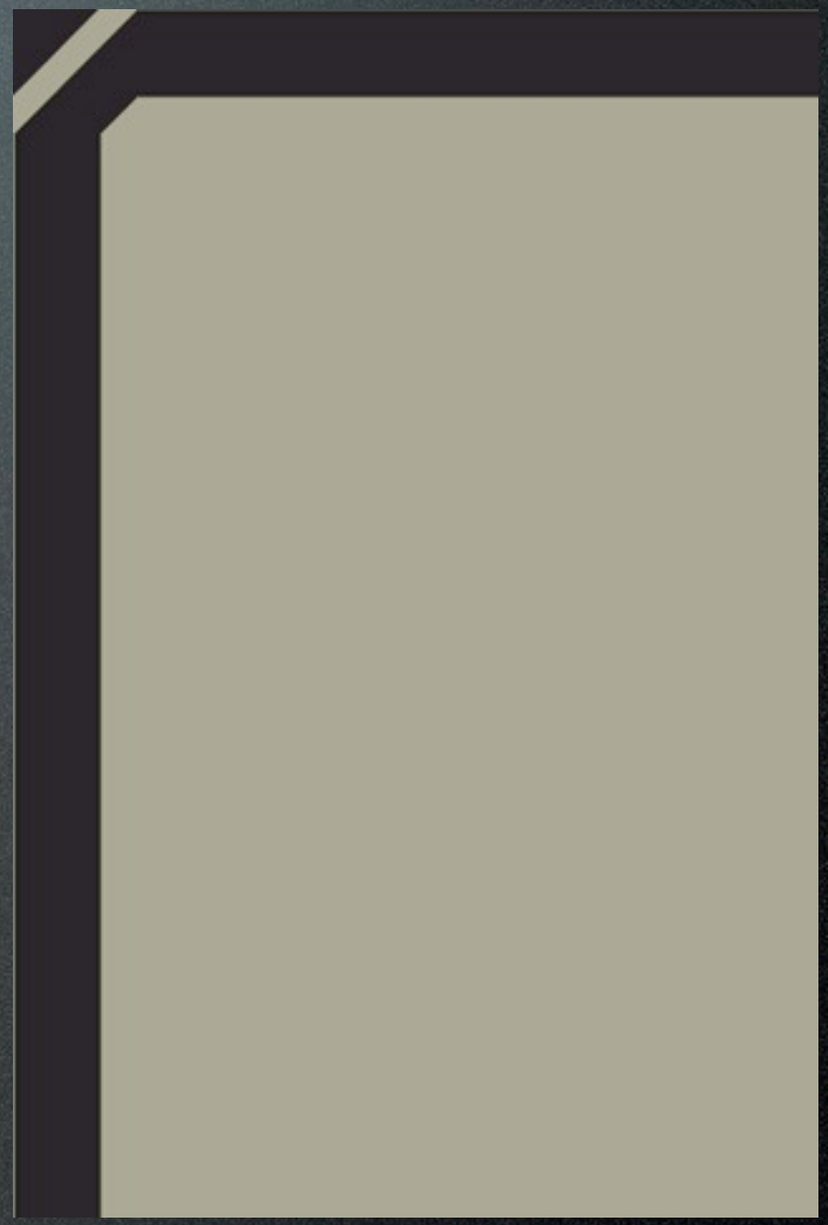
droidcon
LONDON 28-29 OCT 2010

BitmapDrawable

android:tileMode

```xml
<?xml version="1.0" encoding="utf-8"?>

<bitmap
    xmlns:android="http://
schemas.android.com/apk/res/android"
    android:src="@drawable/pattern"
    android:tileMode="clamp" />
```



droidcon
LONDON 28-29 OCT 2010

BitmapDrawable

android:tileMode

```
<?xml version="1.0" encoding="utf-8"?>

<bitmap
    xmlns:android="http://
schemas.android.com/apk/res/android"
    android:src="@drawable/pattern"
    android:tileMode="repeat" />
```

BitmapDrawable

android:tileMode

```
<?xml version="1.0" encoding="utf-8"?>

<bitmap
    xmlns:android="http://
schemas.android.com/apk/res/android"
    android:src="@drawable/pattern"
    android:tileMode="mirror" />
```

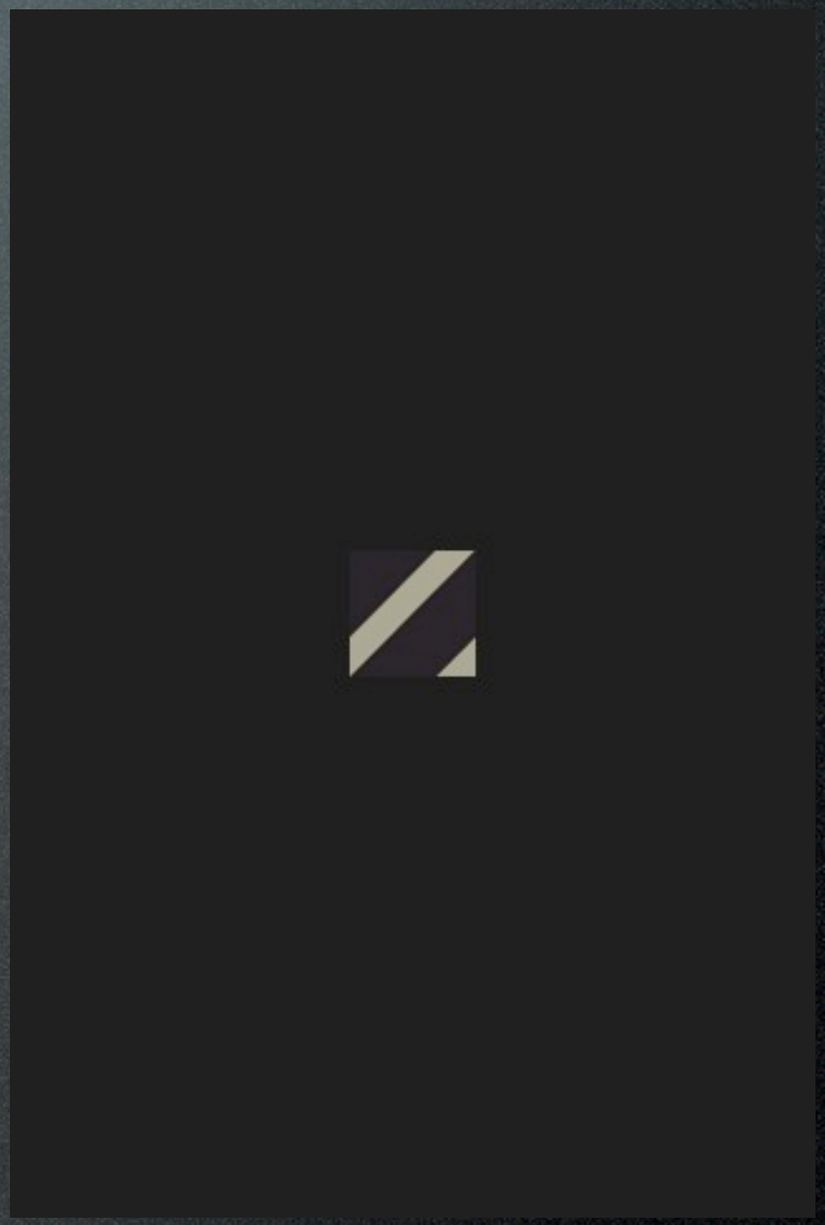
droidcon
LONDON 28-29 OCT 2010

BitmapDrawable

android:gravity

```
<?xml version="1.0" encoding="utf-8"?>

<bitmap
    xmlns:android="http://
schemas.android.com/apk/res/android"
    android:src="@drawable/pattern"
    android:gravity="center" />
```



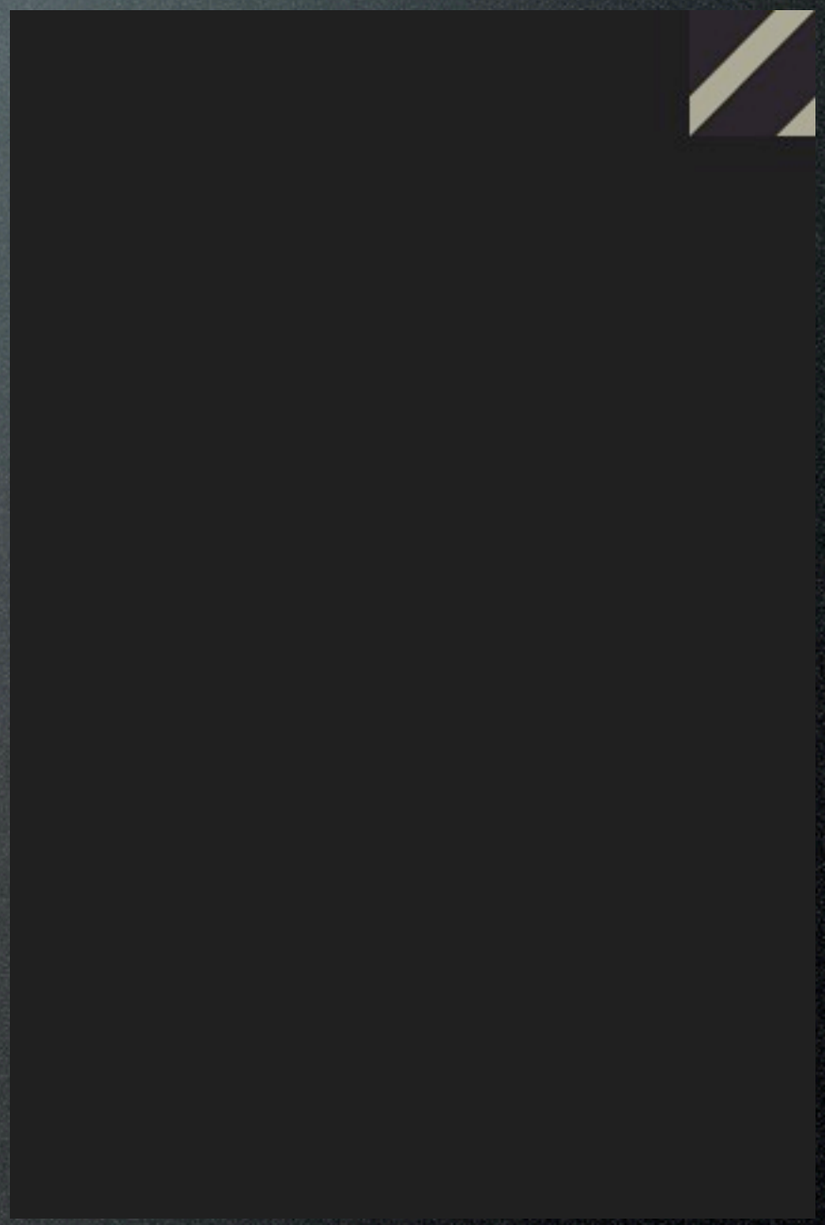
droidcon
LONDON 28-29 OCT 2010

BitmapDrawable

android:gravity

```
<?xml version="1.0" encoding="utf-8"?>

<bitmap
    xmlns:android="http://
schemas.android.com/apk/res/android"
    android:src="@drawable/pattern"
    android:gravity="top|right" />
```



NinePatchDrawable

My favorite one!

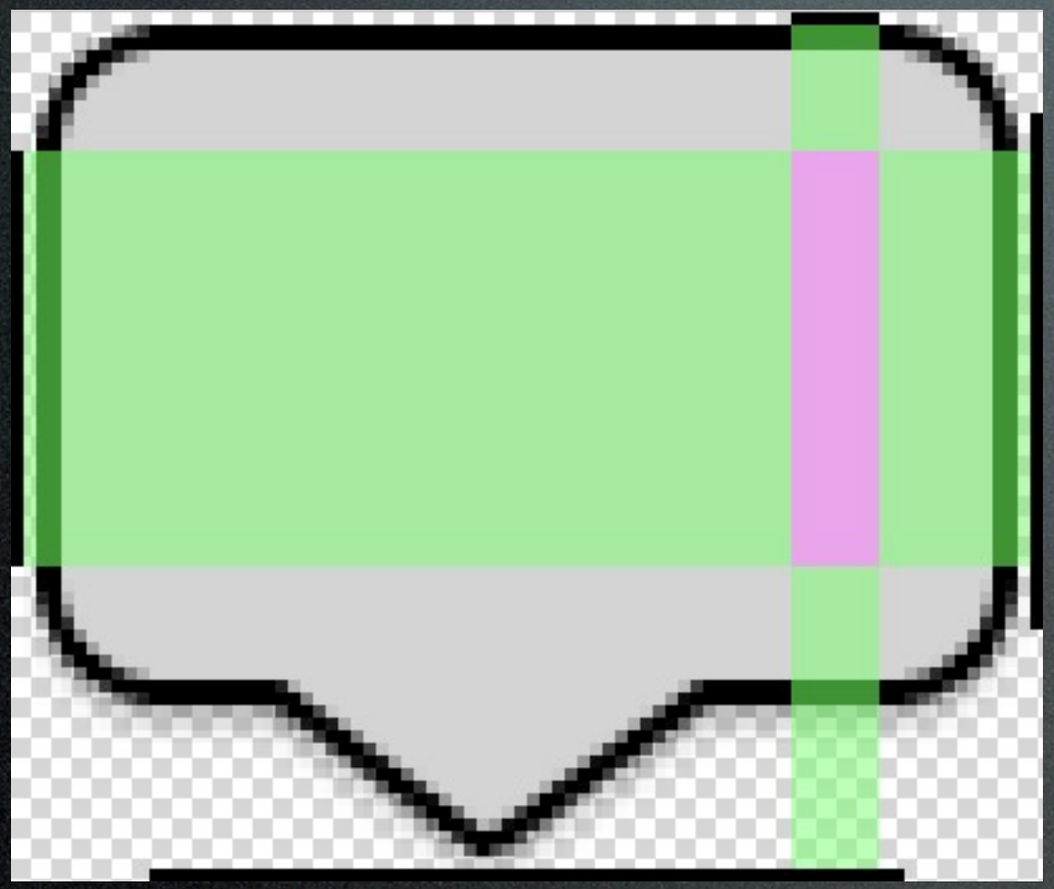
NinePatchDrawable

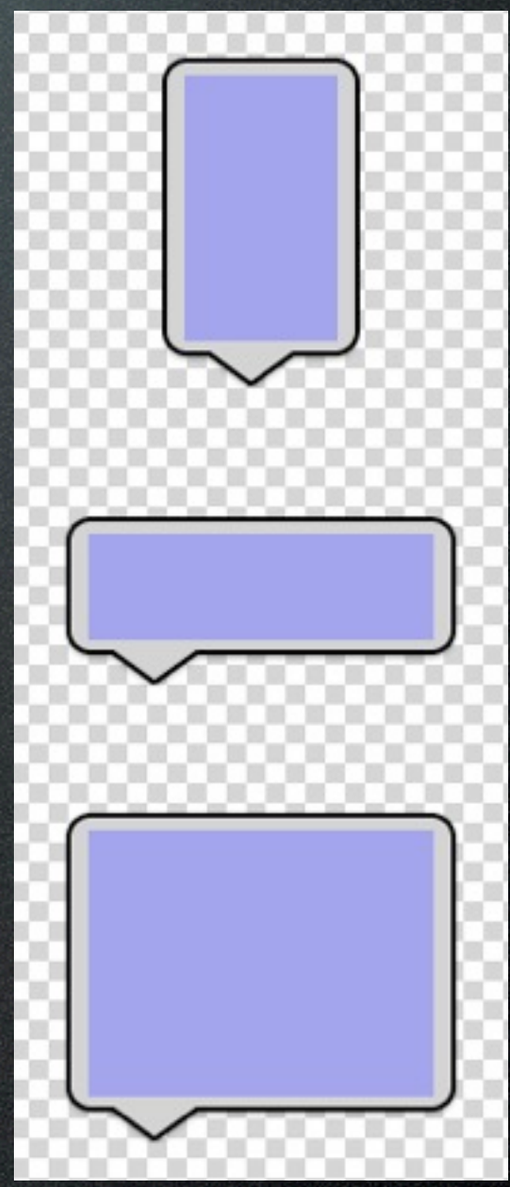
My favorite one!

- Simple instantiation :
  - getDrawable(R.drawable.image) (image with a .9.png extension)
- Stretchable image concept

NinePatchDrawable

My favorite one!





StateListDrawable

- Used everywhere :
  - Buttons
  - Checkboxes
- Change its apperance based on the current state

Final demos

Are those the final fireworks?

droidcon
LONDON 28-29 OCT 2010

A quoi ça sert tout ça?

Flatten Hierarchy

Code available on
http://www.github.com/cyrilmottier/DrawablePresentation

Créer son Drawable

- Étendre de Drawable (ou une classe fille)
- Définir les méthodes abstraites
  - setAlpha, setColorFilter, draw, etc.
- Redéfinir si nécessaire

RemoteDrawable

Code available on
http://www.github.com/cyrilmottier/DrawablePresentation

References

- Know more about it (or more):
  - http://android.cyrilmottier.com
- Source code :
  - http://android.git.kernel.org/
  - http://github.com/cyrilmottier

Some questions?

Do not hesitate ... but please keep in mind
English is not my native language :)